Video Processing Engine

This chapter describes the video processing engine (VPE) for the device.

Topic **Page**

10.1 VPE Overview

VPE Features:

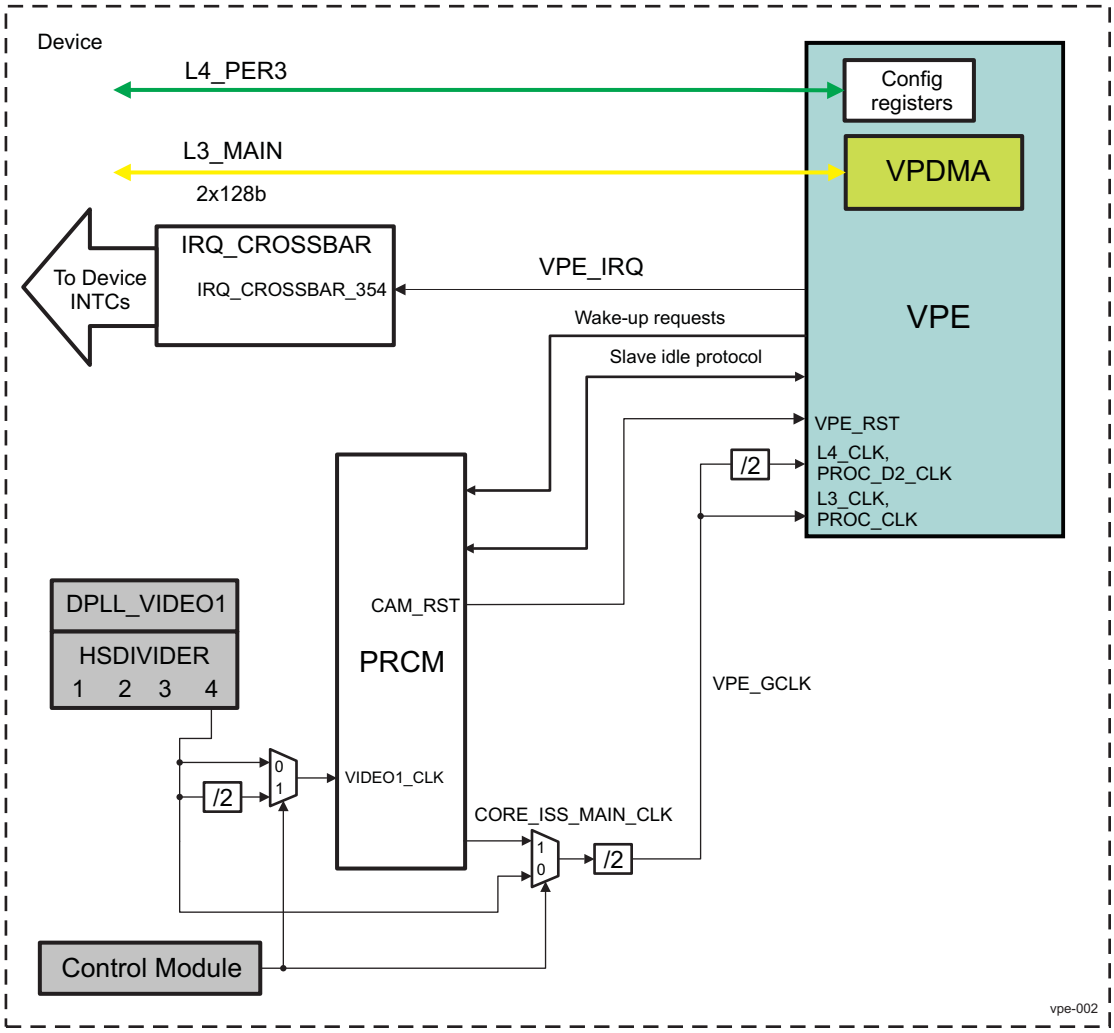
- Supports memory to memory operations only.
- VPE consist of a single memory to memory path which can perform the following operations:
  - Read of raster or tiled YUV420 coplanar, YUV422 coplanar or YUV422 interleaved video
  - Deinterlacing of the input video using a 4 field motion based algorithm
  - Scaling of the input video up to 1080p (1920x1080) resolution
  - Write of the resulting video in YUV420 coplanar (raster or tiled), YUV422 coplanar (raster or tiled), YUV422 interleaved (raster or tiled), YUV444 single plane (raster only) or RGB888 (raster only)
  - Deinterlacing up to two 1080i video sources.
  - The single data path performs operations in the following order
    - Chroma Upsampling from 420 to 422 (if needed)
    - Deinterlacing of 422 video from interlaced to progressive (if needed)
    - Scaling of 422 video after deinterlace
    - Conversion of 422 video to 420, 444 or RGB (if needed)
  - VC-1 Range Mapping and Range Reduction support on input video before Chroma Upsampling (if needed)
- Chroma Upsampling Features
  - 4 line Catmull-Rom based implementation
  - Programmable coefficients for interlaced or progressive conversion. Separate coefficients can be provided for top and bottom fields
- Deinterlacer Features
  - 8-bit, YCbCr 4:2:2
  - Motion-adaptive deinterlacing (MDT)
    - Motion detection is based on Luma only
    - 4-field data is used
    - Motion values adaptive to the frequency of luma texture
  - Edge-Directed Interpolation (EDI)
    - Edge detection using luma pixels in a 2x7 window
    - Seven edge vectors: -1.5, -1, -0.5, 0, 0.5, 1, 1.5
    - Edge-directed chroma interpolation
    - Soft-switch between edge directed interpolation and vertical interpolation depending on the confidence factor
  - Film Mode Detection (FMD)
    - 3-2 pull down detection
    - 2-2 pull down detection
    - Hysteresis controls how fast FMD can enter/exit film mode (software function)
    - Bad Edit Detection (BED)
  - Progessive Input
    - For Progressive Input, the module passes input to output. No internal processing is performed. This is essentially a bypass mode
  - Interlace Bypass
    - For Interlace Input, the module can pass the inputs data directly to the outputs in a bypass configuration. No internal processing is performed
- Scaler Features
  - Vertical and horizontal up/down scaling
    - Polyphase filter upscaling

  - Running average vertical down scaling for memory optimization
  - Decimation and polyphase filtering for horizontal scaling
  - Non-linear scaling for stretched/compressed left and right sides
  - Input image trimmer for pan/scan support
  - Pre-scaling peaking filter for enhanced sharpness
  - Scale field as frame
  - Interlacing of scaled output
  - Full 1080p input and output support
  - YCbCr422 input and output
  - Minimum horizontal scaling ratio = 1/8x
  - Maximum horizontal scaling ratio – limited by output line buffer (2014 pixels)
  - Scaling filter Coefficient memory download
- Chroma Downsampler Features
  - Simple two-line averager capable of converting from YUV422 to YUV420 space
- 422 to 444 Features
  - Catmull-Rom based filter
  - 4 pixel, fixed coefficient
- Color Space Converter Features
  - Fully programmable 3x3 matrix multiplier with offset control

10.2 VPE Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests. Figure 10-1 summarizes the integration of the module in the device.

Figure 10-1. VPE Integration



NOTE: Alternative clocking to the VPE module is provided from DPLL_VIDEO1. For more information about configuring DPLL_VIDEO1, see Section 11.1.2.1, *Display Subsystem Clocks*. Source clock selection is done with CTRL_CORE_SMA_SW_1[8] VPE_CLK_DIV_BY_2_EN register from the Control Module. For more information, see Chapter 18, *Control Module*.

Table 10-1 and Table 10-2 list the integration attributes and clock and resets, respectively.

Table 10-1. VPE Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
VPE	PD_VPE	L4_PER3 for configuration L3_MAIN for data (through VPDMA module)

Table 10-2. VPE Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
VPE	L3_CLK PROC_CLK	VPE_GCLK	PRCM	L3_CLK is the clock used to drive and receive data over the bus to L3_MAIN. The VPDMA uses this clock to send and receive external data and transfer this data to internal processing. PROC_CLK is the clock used to drive data processing within the VPE subsystem.
	L4_CLK PROC_D2_CLK	VPE_GCLK/2	PRCM	L4_CLK is the interface clock for Memory Mapped Registers (MMR) configuration bus PROC_D2_CLK is an additional clock used by the DEI within the DEI Subsystem. Inputs and outputs of the DEI operate on PROC_CLK, but internally, the data paths within the DEI operate on this clock.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
VPE	VPE_RST	VPE_RST	PRCM	VPE Reset

Table 10-3. VPE Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
VPE	VPE_IRQ	IRQ_CROSSBAR_354	N/A	VPE interrupt requests. These IRQ source signals are not mapped by default to any device INTC.

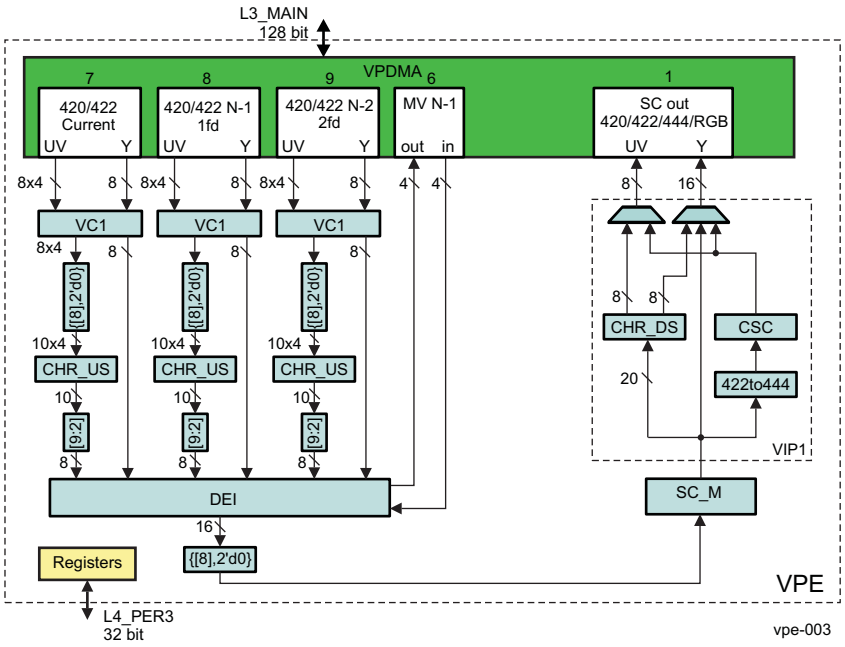
NOTE: The "**Default Mapping**" column in Table 10-3 *VPE Hardware Requests* shows the default mapping of module IRQ source signals. These module IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module, respectively.
For more information about the IRQ_CROSSBAR module, see Section 18.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 18, *Control Module*.
For more information about the device interrupt controllers, see Chapter 17, *Interrupt Controllers*.

10.3 VPE Functional Description

10.3.1 VPE Block Diagram

Figure 10-2 shows the internal structure of the VPE module in the device.

Figure 10-2. VPE Block Diagram



10.3.2 VPE VC1 Range Mapping/Range Reduction

VC1 range mapping and range reduction is implemented prior to the Chroma Upsampler (CHR_US) in the Primary Input Paths of the module.

Range Mapping is performed by setting VPE_CLKC_RANGE_MAP[6] RANGE_MAP_PRIM_ON bit to '1'. The output for every component is calculated by the follwing formulas:

Y[n] = (((Y[n] – 128) * (RANGE_MAPY_PRIM + 9) + 4) >> 3) + 128

Cb[n] = (((Cb[n] – 128) * (RANGE_MAPUV_PRIM + 9) + 4) >> 3) + 128

Cr[n] = (((Cr[n] – 128) * (RANGE_MAPUV_PRIM + 9) + 4) >> 3) + 128

In the above, VPE_CLKC_RANGE_MAP[2:0] RANGE_MAPY_PRIM and VPE_CLKC_RANGE_MAP[5:3] RANGE_MAPUV_PRIM are defined as in the register descriptions for each instantiation of this function.

Range Reduction is performed based by setting VPE_CLKC_RANGE_MAP[28] RANGE_REDUCTION_PRIM_ON bit to '1'. The output for the color components is calculated by the follwing formulas:

Y[n] =(Y[n] – 128) * 2 + 128

Cb[n] =(Cb[n] – 128) * 2 + 128

Cr[n] = (Cr[n] – 128) * 2 + 128

NOTE: The block performs Range Mapping first, and the output of Range Mapping drives Range Reduction. Although Range Mapping and Range Reduction are supposed to be mutually exclusive, the implementation allows both to be done simultaneous.

10.3.3 VPE Deinterlacer (DEI)

10.3.3.1 Functional Description

Figure 10-3 illustrates the block-diagram of motion-adaptive Deinterlacer. The general concept behind motion adaptive deinterlacing is that spatial filtering works very well for images with motion, while temporal filtering works very well for static images. So, the intuitive way is to combine them together. Motion detection is used to switch or fade between the use of spatial deinterlacing and temporal deinterlacing, as shown in the following formula:

$$\hat{y}(j, i, n) = \alpha y_{spat}(j, i, n) + (1 - \alpha) y_{temp}(j, i, n)$$

where $y_{spat}(j, i, n)$ is the spatial interpolation output, $y_{temp}(j, i, n)$ is temporal interpolation output, α is the motion detection output ranging from 0 to 1, $\hat{y}(j, i, n)$ is the final output from deinterlacer, and j, i, n are the vertical, horizontal, and temporal indexes, respectively. From the previous formula, the final output is controlled by the motion detector output, α. The higher the motion, the higher value of α, and the output favors spatial interpolation. If the motion is absent or very low, the temporal interpolation has higher weight.

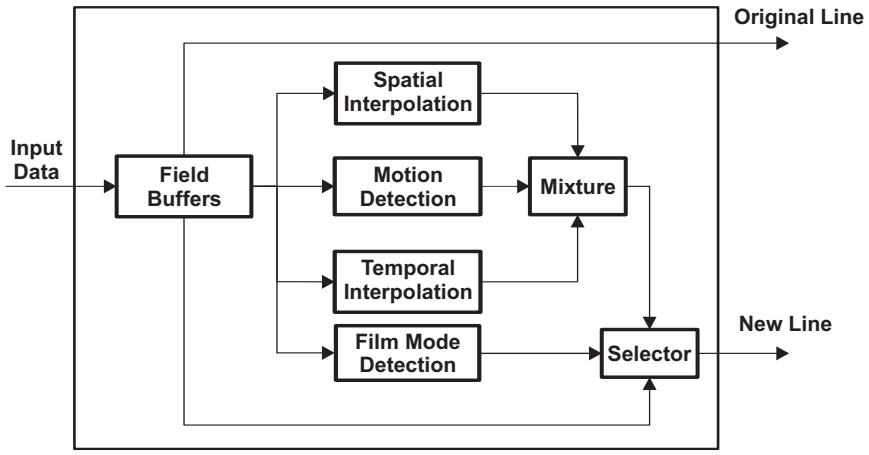
Temporal interpolation can be disabled by programmable control registers. In that case we have:

$$\hat{y}(j, i, n) = y_{spat}(j, i, n)$$

Chroma interpolation is handled in the same manner as luma with regards to the above equations. There is a separate control for disabling temporal interpolation for chroma, such that luma can perform the full mixture, and chroma can only be spatial. If luma temporal interpolation is disabled, chroma temporal interpolation is also disabled.

Figure 10-3 provides a simple description of how a motion-adaptive deinterlacer operates. The actual interpolation used is edge directed interpolation. Edge directed interpolation is only performed spatially prior to the output mixing.

Figure 10-3. Block Diagram of Motion-Adaptive Deinterlacer



The Deinterlacer consists of:

- MMR Configuration Register Block
  - – - Used to program configuration items for the Deinterlacer
- VPDMA Interface Block
  - – Used to read source video/motion data from VPDMA
  - – Used to write generated motion data to VPDMA
- Motion Detection (MDT) Block
  - – Examines 3 fields of input video data (luma only) and calculates a 4 bit motion vector to drive the Edge Directed Interpolation Block

- Edge Directed Interpolation (EDI) Block
  - Performs the motion based edge directed interpolation on Luma and Chroma inputs to generate the missing line in the interlaced source
- Film Mode Detection (FMD) Block
  - Field Difference : Accumulated difference between the spatial interpolated frame and the adjacent input field of opposite field ID
  - Frame Difference : Accumulated difference between two fields with the same field ID
  - Combing Artifacts : Accumulated sum-of-difference between two adjacent fields
- Output Multiplexor (MUX) Block
- Line Buffer Block

10.3.3.2 Bypass Mode

The DEI can be operated in bypass mode and deinterlacer mode.

In the bypass mode, input luma and chroma are buffered and sent to the stage after DEI without processing. To bypass the DEI module registers VPE_DEI_REG0[31] PROGRESSIVE_BYPASS or VPE_DEI_REG0[29] INTERLACE_BYPASS must be set to '1' for progressive and interlaced source input respectivly.

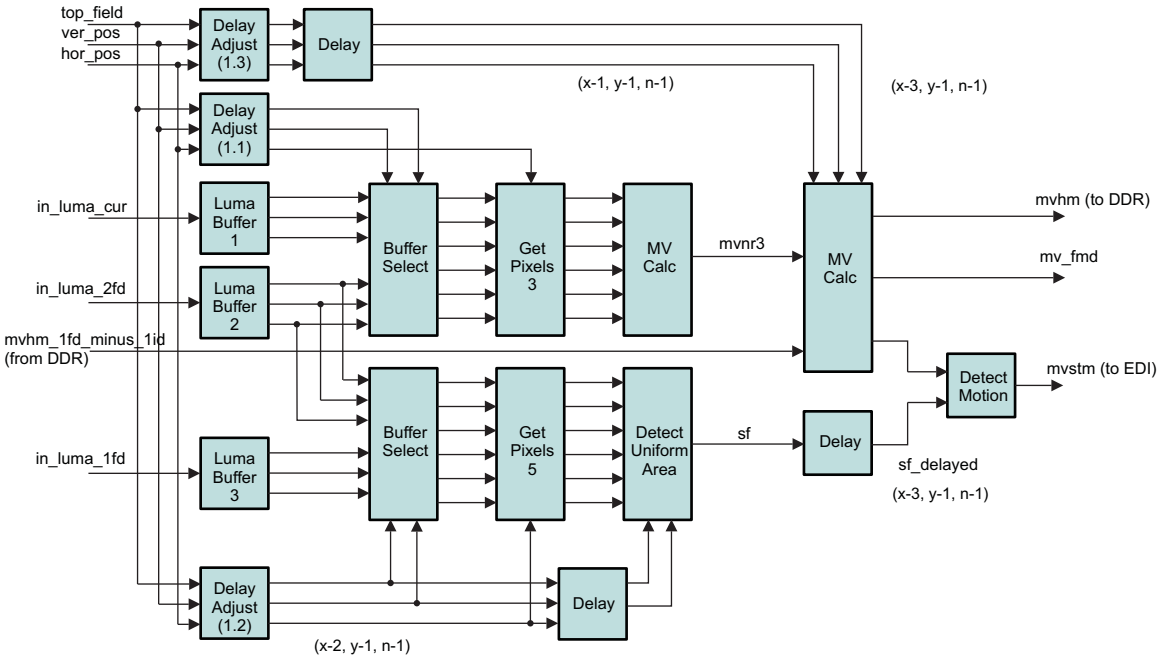
10.3.3.2.1 VPDMA Interface

The VPDMA interface handles transactions between the internal core of the design and the VPI interface protocol, which is used for both external module input and motion vector output. All input data is pixel aligned, but has different request and ready signals. Motion Output is not pixel aligned with input.

10.3.3.2.2 MDT

A block diagram representing the Motion Detection (MDT) Block is shown on Figure 10-4.

Figure 10-4. Motion Detection (MDT) Block Diagram



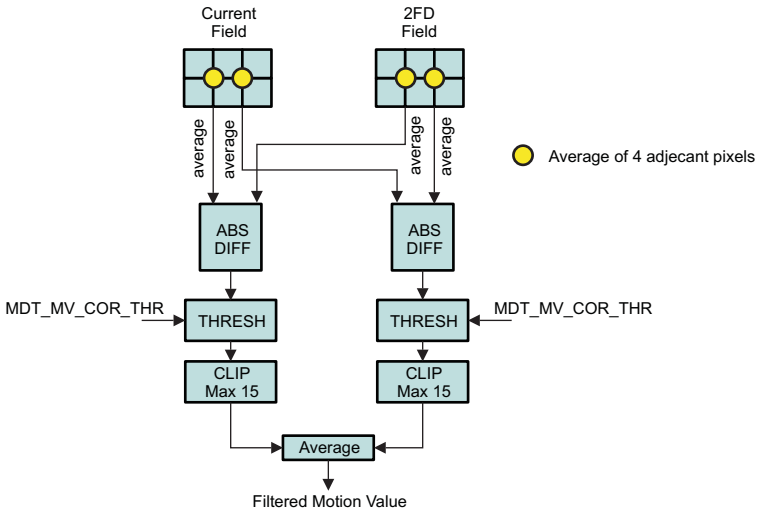
The Motion Detection block takes as input 3 adjacent fields of Luma data, motion data calculated from the previous 3 fields, and the current X, Y and field ID indicators and generates motion vector outputs which drive the Edge Directed Interpolation (EDI), Film Mode (FMD) and output motion values to DDR

There are two parallel paths - Motion Vector Calculation and Uniform Area Detection.

The Motion Vector Calculation path generates the FMD motion vector and the DDR output motion vector. The combination of the Motion Vector and Uniform Area Detection is used to calculate the EDI motion vector value.

The Motion Vector Calculation Path operates using a sliding 3 pixel wide by 2 pixel tall window over 2 adjacent fields of the same field ID polarity (2 top fields or 2 bottom fields). The first stage is to calculate the filtered motion value (MV Calc). The below diagram shows the dataflow for calculating the Filtered Motion Value from MV Calc:

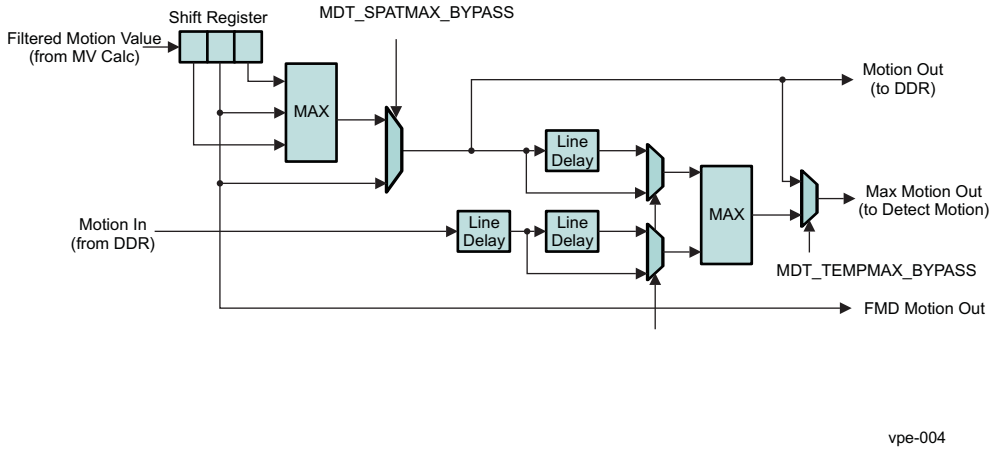
Figure 10-5. Motion Detection (MDT) MV Calc Data Path



The Filtered Motion Value is then operated on by the Max Filter operation, which performs a maximum operation between the last 3 filtered motion values and the last frame's motion value. The process is to calculate the maximum value of the last 3 filter motion values, and this value is written to DDR to be read back in on the next frame. The maximum value is taken between the previous frame's maximum value as read back from DDR, and the maximum value between the current last 3 motion values, and this is passed to the Detect Motion block which will drive the EDI module.

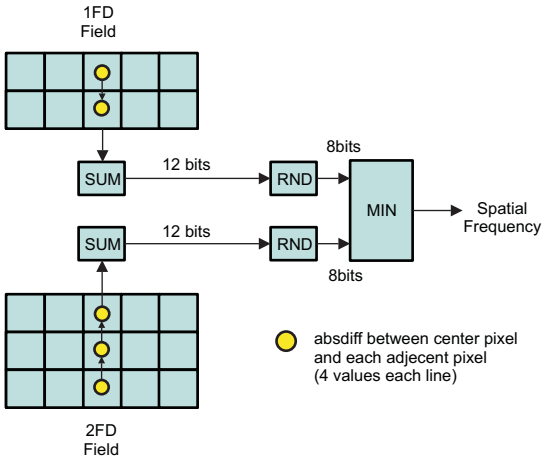
The Film Mode motion value output is just the output from the MV Calc data path. The following diagram shows the data path. Please note the line buffers, which are used to adjust for different polarity of fields (2 motion value fields are of opposite polarity, meaning a 1 line difference in one or the other field):

Figure 10-6. Motion Detection (MDT) Max Filter Data Path



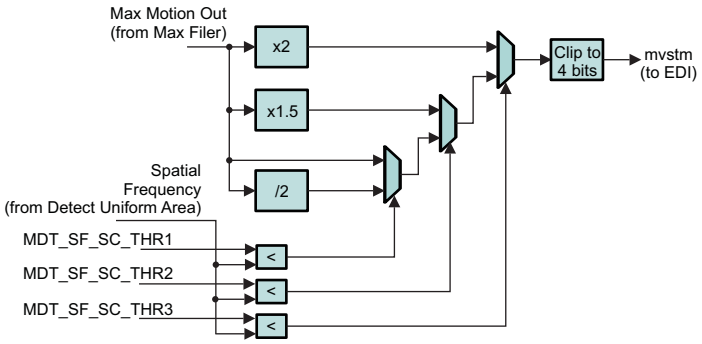
The Uniform Area Detection Path operates using a sliding 5 pixel wide by 3 pixel tall window over 2 adjacent fields of opposite field ID polarity (1 field delay and 2 field delay) to calculate a spatial frequency component. The following diagram shows how this component is calculated within each pixel window:

Figure 10-7. Motion Detection (MDT) Uniform Area Data Path



The spatial frequency output is used to scale the motion vector generated by the Maximum Filter operation based on thresholding against the MMR values VPE_DEI_REG2[7:0] MDT_SF_SC_THR1, VPE_DEI_REG2[15:8] MDT_SF_SC_THR2 and VPE_DEI_REG2[23:16] MDT_SF_SC_THR3. The following diagram (in the Detect Motion block) performs this thresholding to generate the final motion vector output to the EDI module:

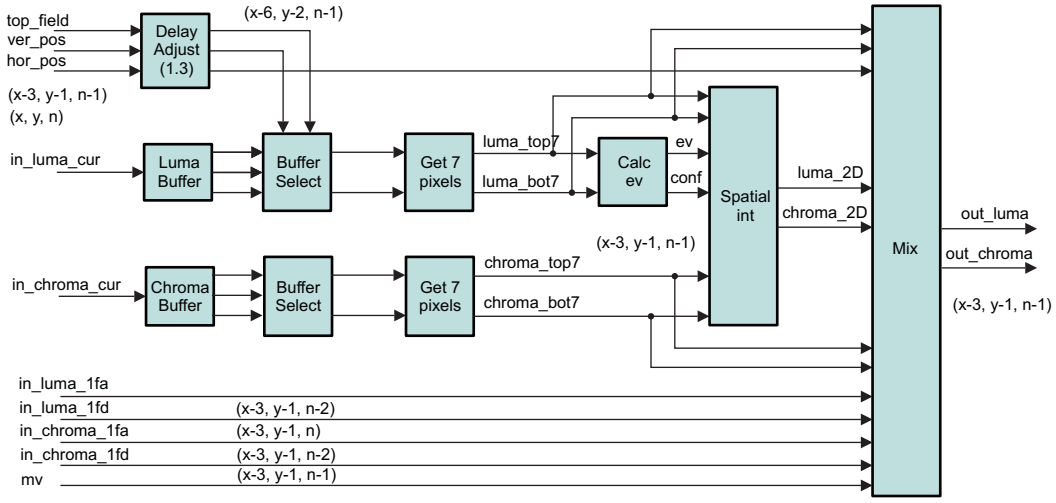
Figure 10-8. Motion Detection (MDT) Detect Motion Data Path



10.3.3.2.3 EDI

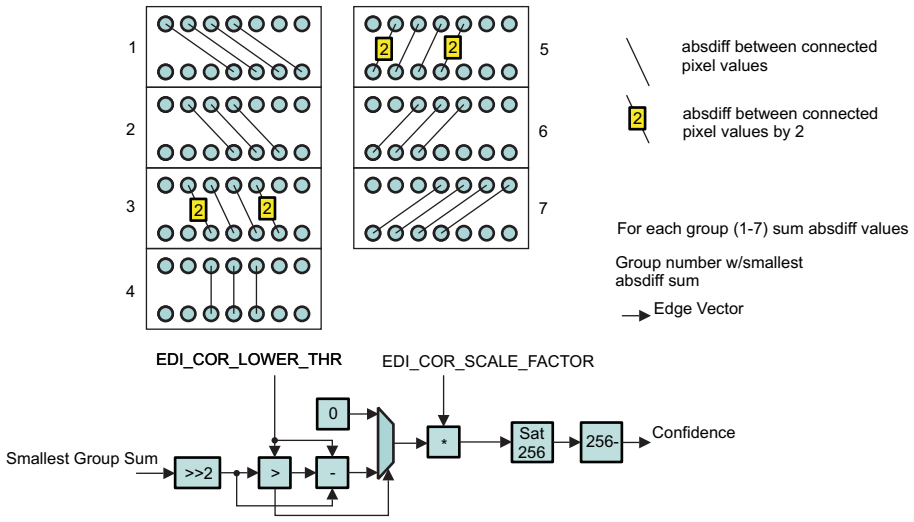
A block diagram representing the Edge Directed Interpolation (EDI) Block is shown on Figure 10-9

Figure 10-9. Edge Directed Interpolation (EDI) Block Diagram



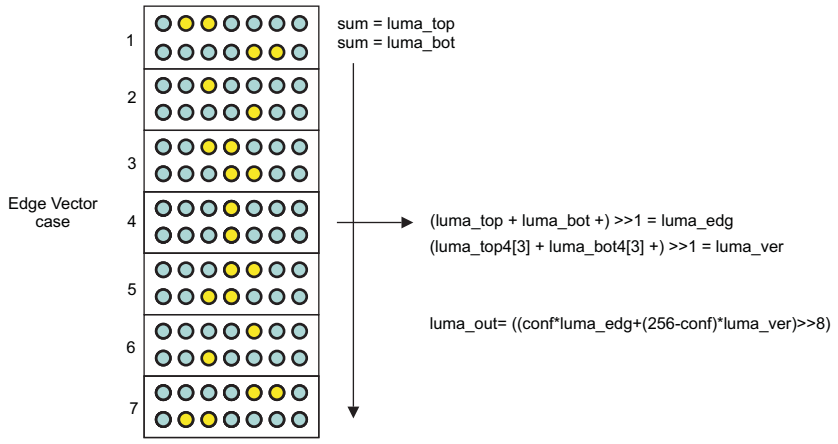
The Edge Directed Interpolation module operates on a 7x2 window (7 pixels wide, 2 in height) in both Luma and Chroma. Luma data is used to calculate an edge vector and edge vector confidence. Correlation scaling factor is set with VPE_DEI_REG3[31:24] EDI_COR_SCALE_FACTOR register. A diagram showing this calculation follows (Calc ev):

Figure 10-10. Edge Directed Interpolation Edge Vector Calculation



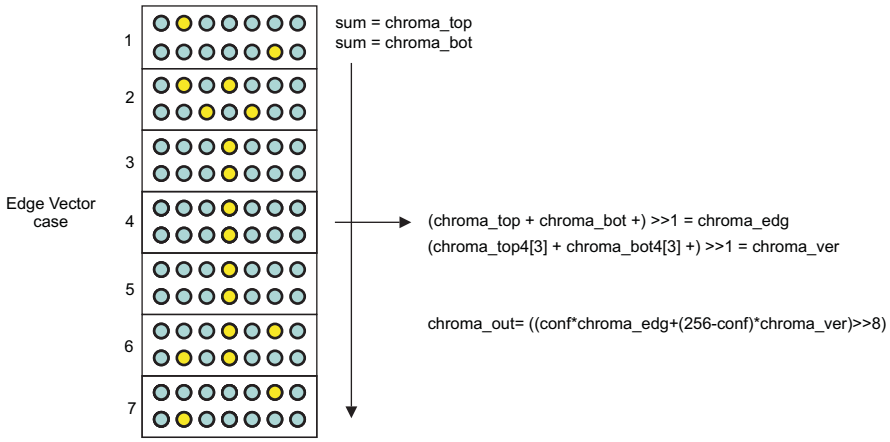
The edge vector and edge vector confidence are passed to the Spatial Interpolation module, which performs edge directed spatial interpolation. Following is a diagram showing how Luma is interpolated:

Figure 10-11. Edge Directed Interpolation Luma Interpolation Calculationn



The following diagram shows how Chroma is interpolated:

Figure 10-12. Edge Directed Interpolation Chroma Interpolation Calculation



Each of the cases shown above is on the 7x2 input pixel window for Luma and Chroma, based on the edge vector that was calculated. Each output equation (luma_out and chroma_out) is a combination of a straight vertical interpolation (luma/chroma_ver) and edge directed interpolation (luma/chroma_edge) using the Confidence factor output from the Edge Vector Calculation module. In the case of chroma interpolation, if the register VPE_DEI_REG3[1:0] EDI_INP_MODE = “11”, the chroma_out calculated for spatial interpolation is forced to the the vertical chroma output.

The Mix module does that actual mixing of spatial and temporal interpolation to produce the final result. 4 different interpolation modes are supported: Line double, Field double, 3D interpolation and 2D interpolation.

- Line double averages the top and bottom pixel to produce the interpolated pixel. Setting of MDT mode has no effect on output pictures.
- Field double averages the previous and next frame pixel to produce the interpolated pixel. In other words, if the current field is a top field, the interpolated bottom field picture is created by averaging pixels from bottom field pictures before and after the current field
- 2D interpolation uses the Edge Directed interpolation result only as the interpolated pixel
- 3D interpolation uses the EDI LUT table values with the MDT motion vector as an index to blend between the 2D result and the value from the previous field temporally.

NOTE: 3D processing is enabled with VPE_DEI_REG3[2] EDI_ENABLE_3D register

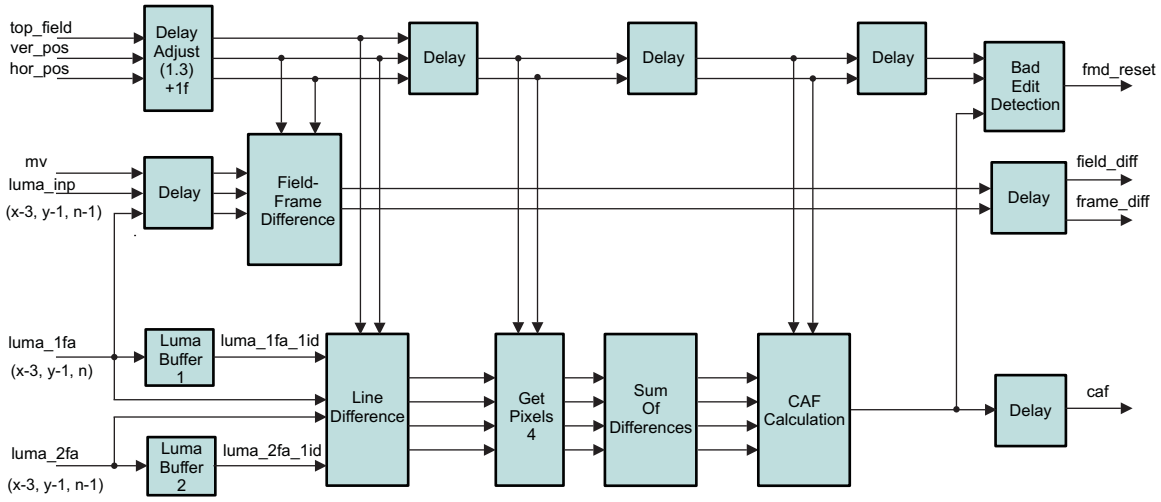
The EDI Lut value selected based on the motion value will perform a blend between temporal and edge directed spatial interpolation using the equation:

$\hat{y}(j, i, n) = \alpha y_{spat}(j, i, n) + (1 - \alpha) y_{temp}(j, i, n)$

10.3.3.2.4 FMD

The following shows the block diagram of the Film Mode Detection (FMD) module:

Figure 10-13. Film Mode Detection (FMD) Block Diagram

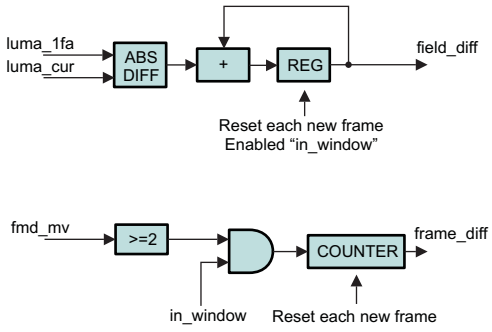


Film Mode Detection calculates four parameters which are used in conjuction with software to determine if the source of the video is film (3:2 or 2:2 sequences). This block is not part of the main data path, but calculates statistics based on the main data path for each frame generated, then interrupts the processor to read the information and then set to '1' the VPE_DEI_REG10[1] FMD_LOCK (lock to film mode) and VPE_DEI_REG10[2] FMD_JAM_DIR (direction of field jam) registers, to set the design to film mode operation.

All of the calculations are performed within a window defined by VPE_DEI_REG8[10:0] FMD_WINDOW_MINX, VPE_DEI_REG8[26:16] FMD_WINDOW_MAXX, VPE_DEI_REG9[10:0] FMD_WINDOW_MINY, VPE_DEI_REG9[26:16] FMD_WINDOW_MAXY. All calculations are reset on each new input frame.

Field difference is calculated as the absolute difference (absdiff) between the current Luma input and the previous field Luma input. Value can be read on VPE_DEI_REG13[27:0] FMD_FIELD_DIFF register. Frame difference is calculated using the motion input from the MDT. Value can be read on VPE_DEI_REG14[19:0] FMD_FRAME_DIFF register. Figure 10-14 is showing how these are calculated is shown below:

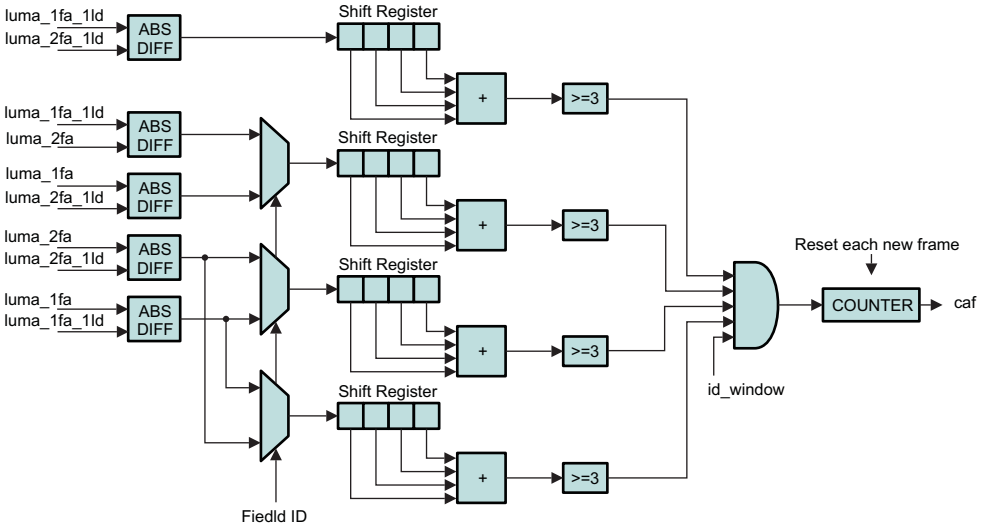
Figure 10-14. Film Mode Detection (FMD) Frame/Field Difference Calculation



NOTE: Value of FMD_WINDOW_MAXX must be less then WIDTH and value of FMD_WINDOW_MAXY must be less then ½ HEIGHT.

Combing Artifacts value is calculated as shown in Figure 10-15. Value can be read on VPE_DEI_REG12[20:0] FMD_CAF register.

Figure 10-15. Film Mode Detection (FMD) Combing Artifacts Calculation



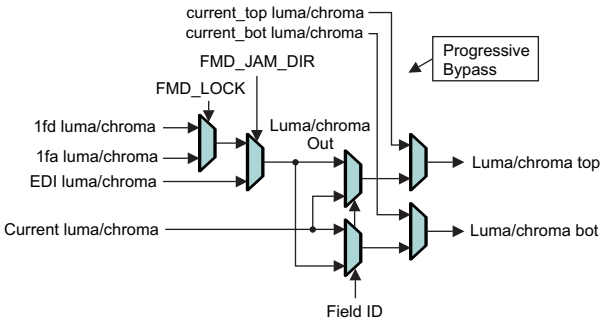
FMD_RESET is set to tell software when the Film Mode Detection should be unlocked back to normal operation. The logic controlling this is based on the caf output, and is set based on caf being above a threshold set by VPE_DEI_REG11[19:0] FMD_CAF_THR and where within the window the current x/y position is. If the position is over ¾ in height, and caf > FMD_CAF_THR OR if over ½ in height and caf > ¾*FMD_CAF_THR OR if over ¼ in height and caf > ¼*FMD_CAF_THR, then VPE_DEI_REG12[24] FMD_RESET must be set (it will clear at the start of the next field input).

The Film Mode Interrupt (fmd_int) will be generated at the end of the window defined for FMD operation and tells software to read the above parameters. When the interrupt fires, the above parameters are latched into the DEI Control Interface (MMR) so they can be read.

10.3.3.2.5 MUX

The MUX block generates the proper outputs based whether the design is in standard Deinterlacer modes (interpolated inputs from the EDI), Film Mode Deinterlacer mode (direct inputs) or progressive bypass mode. The following diagram shows this data path:

Figure 10-16. Film Mode Detection (FMD) Combing Artifacts Data Path



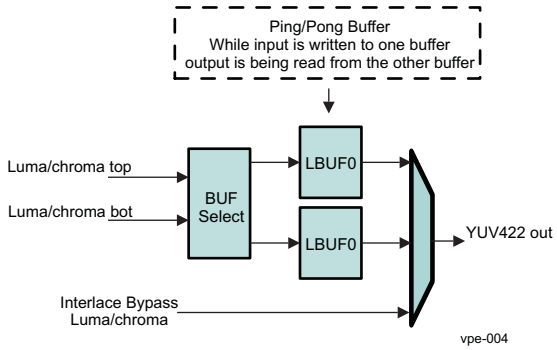
When in Film Mode (VPE_DEI_REG10[1] FMD_LOCK = '1'), the output of the DEI depends only on the input field sequences. Depending on the Jam Direction , either the previous or next field data is output. If not in Film Mode, the interpolated result from the EDI is output.

Because the DEI produces the missing line in an interlaced frame, what this part of the data path is selecting is the “interpolated” line. The other line output is the current line. Together, the interpolated line and the current line form the actual output of the main part of the DEI data path. Depending on if the current field is a top field or bottom field (as selected by Field ID), the output “top” of the MUX is either the current or interpolated lines, and the output “bot” is the opposite.

10.3.3.2.6 LINE BUFFER

The Line Buffer takes two lines of data (top line and bottom line) and serializes this into a single stream. The following diagram shows this data path:

Figure 10-17. Output Data Path



The sequencing is such that LBUF0 is being written while LBUF1 is being read. When LBUF0 has been completely written, it will try to write to LBUF1 next. If on the output side LBUF1 has not been completely read, then the data path will stall waiting for the output to finish reading LBUF1 and switch to reading LBUF0 (and visa versa)

10.3.4 VPE Scaler (SC)

This section is used for driver development for the highly optimized video resizers, SC (scalers), in the VPE module.

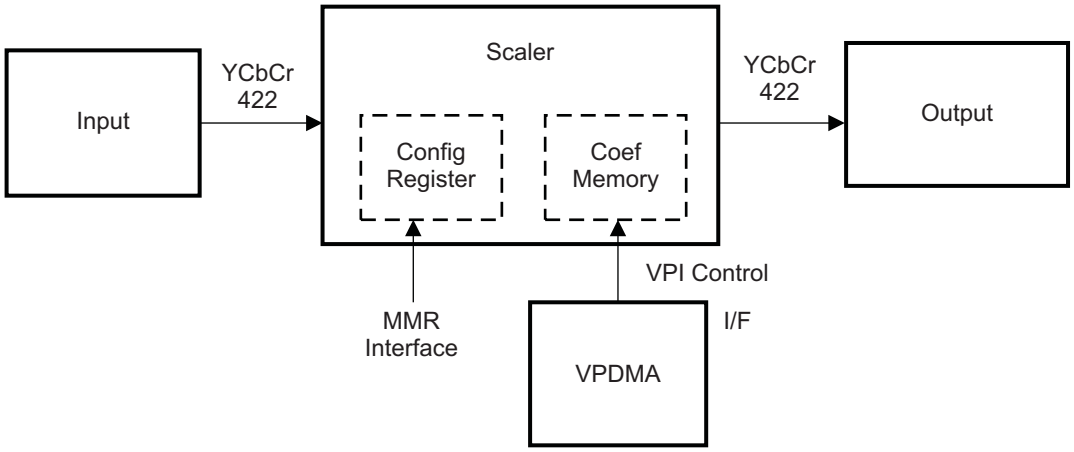
10.3.4.1 SC Features

- Independent vertical and horizontal up and down scaling
- Running average vertical down scaling for memory optimization
- Decimation and polyphase filtering for horizontal scaling
- Non-linear scaling for stretched/compressed left and right sides
- Input image trimmer for pan/scan support
- Pre-scaling peaking filter for enhanced sharpness
- Scale field as frame
- Interlacing of scaled output
- Full 1080p input and output support
- YCbCr422 input and output
- Maximum horizontal scaling ratio only limited by output line buffer (2047 pixels)
- Scaling filter Coefficient memory download via VPI (Video Port Interface) Control interface

10.3.4.2 SC Functional Description

Scaler takes in a 10-bit YCbCr 422 video frame from an upstream module, performs vertical/horizontal scaling and outputs a YCbCr422 scaled image to a next downstream module. All configurations are done via the MMR interface except for the scaler coefficient memory configuration that is done via the common VPI control interface bus by VPDMA. Figure 10-18 shows the high-level block diagram of the scaler module.

Figure 10-18. High Level Block Diagram

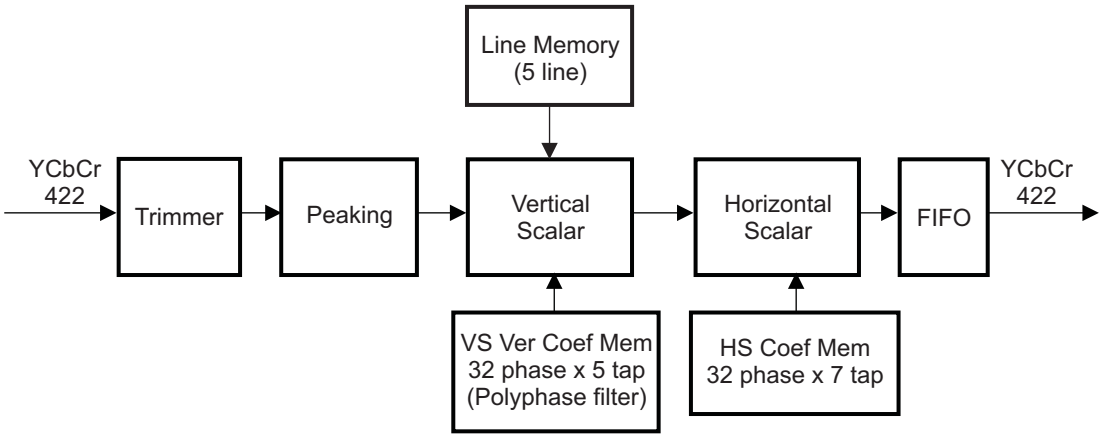


The SC is used in the video path and in all other video write-back data paths in the VPE module.

Scaling is performed in following three steps:

1. Trimming and Pre-peaking filtering
2. Vertical Scaling (Polyphase/Running Average Filter)
3. Horizontal polyphase scaling

Figure 10-19. SC Block Diagram



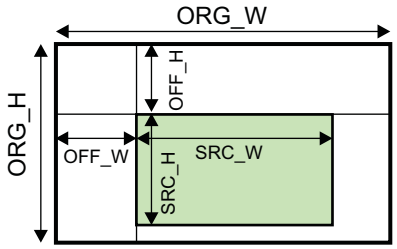
10.3.4.2.1 Trimmer

The trimmer can be programmed to re-define a new source image within the input video frame sent from an upstream module before it is sent to the scaling sub-modules. This feature enables small area zoom out, source pan/scan, or removal of unwanted area in the video (such as black box / curtains / noisy line-21 video) without modifying the VPDMA parameters.

Horizontal and vertical offset is set through VPE_CFG_SC25[26:16] CFG_OFF_W and VPE_CFG_SC25[10:0] CFG_OFF_H registers.

Width and height are set through VPE_CFG_SC24[26:16] CFG_ORG_W and VPE_CFG_SC24[10:0] CFG_ORG_H registers.

Figure 10-20. Input Image Trimming



NOTE: Width and height of the source image are global parameters and are set with VPE_CFG_SC5[22:12] CFG_SRC_W and VPE_CFG_SC5[10:0] CFG_SRC_H registers.

It is required that the input image frame (CFG_SRC_W x CFG_SRC_H) to be at least 32 × 32 (after trimming, if the trimming is enabled) to properly fill the input filter stage pipelines.

10.3.4.2.2 Peaking

The peaking block increases the amplitude of high frequency luminance information in horizontal direction to increase the sharpness of a video image before it is scaled. As shown in Figure 10-21, the high-frequency luminance is increased using an 11-tap High-Pass filter with adjustable gain. The non-linear coring function removes low-level noise and the modified luminance is then added to the original luminance signal. The implementation details are shown in Figure 10-21.

Figure 10-21. Filter Implementation and Parameter Description

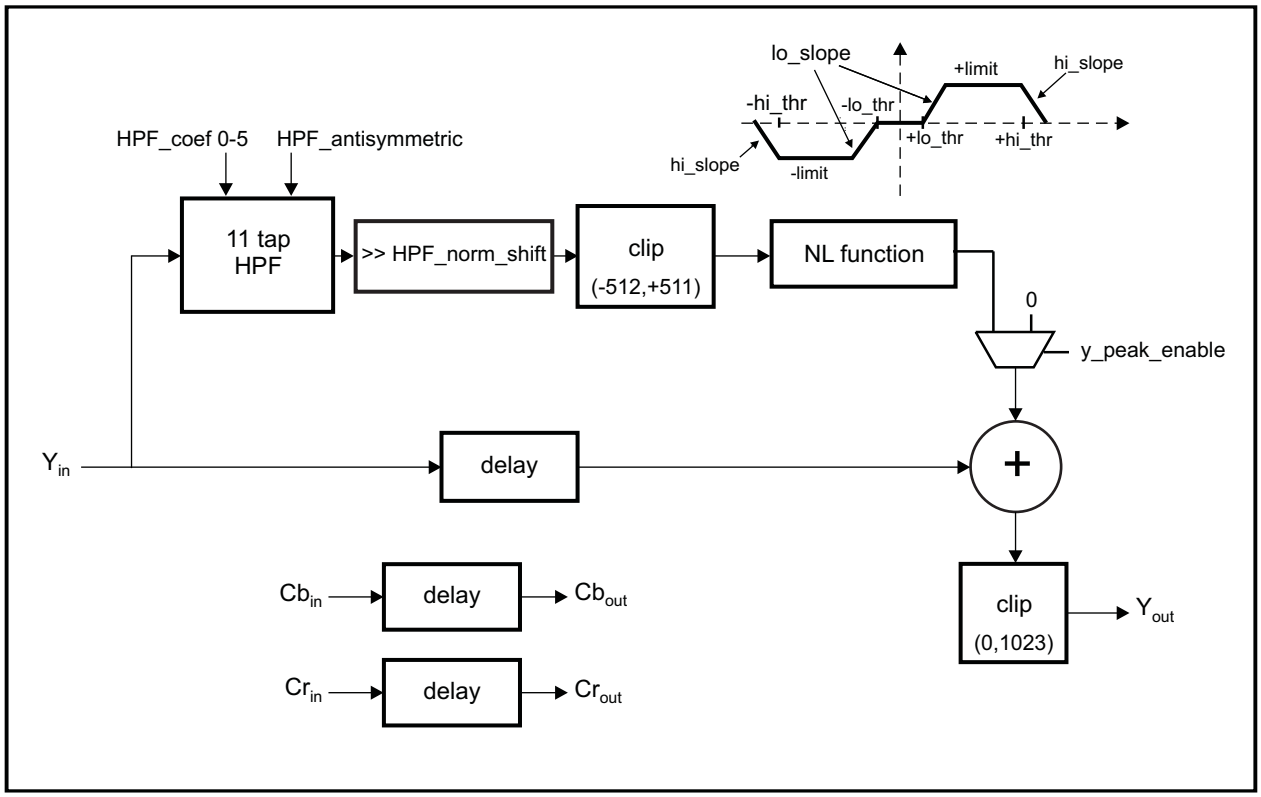
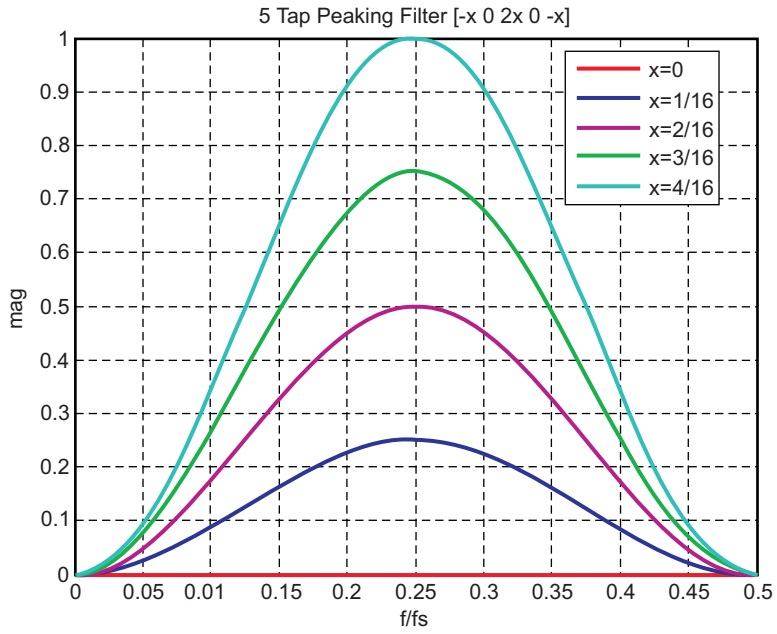


Table 10-4. Parameter Description

Parameter	Description	Bits	Default
VPE_CFG_SC19[7:0] CFG_HPF_COEF0 to VPE_CFG_SC20[15:8] CFG_HPF_COEF5	FIR coefficients	8	[0 0 0-4 0 8]
VPE_CFG_SC20[18:16] CFG_HPF_NORM_SHIFT	Right shift	3	4
VPE_CFG_SC21[8:0] CFG_NL_LO_THR	Coring threshold	9	16
VPE_CFG_SC22[8:0] CFG_NL_HI_THR	High threshold	9	400
VPE_CFG_SC21[23:16] CFG_NL_LO_SLOPE	Lo slope= - CFG_NL_LO_SLOPE/16	8	16
VPE_CFG_SC22[18:16] CFG_NL_HI_SLOPE_SHIFT	Hi slope = $2^{(CFG\_NL\_HI\_SLOPE\_SHIFT-3)}$	3	4
VPE_CFG_SC20[28:20] CFG_NL_LIMIT	Clipping limit	9	200
VPE_CFG_SC0[14] CFG_Y_PK_EN	Control	1	0

Parameters for the Peaking filters are defined in VPE_CFG_SC19 through VPE_CFG_SC22 registers. The frequency responses of the peaking-filter with different sets of coefficients are shown in Figure 10-22. If the source of the input video is NTSC or PAL format, the peaking filter can be configured to reject the color subcarrier frequency.

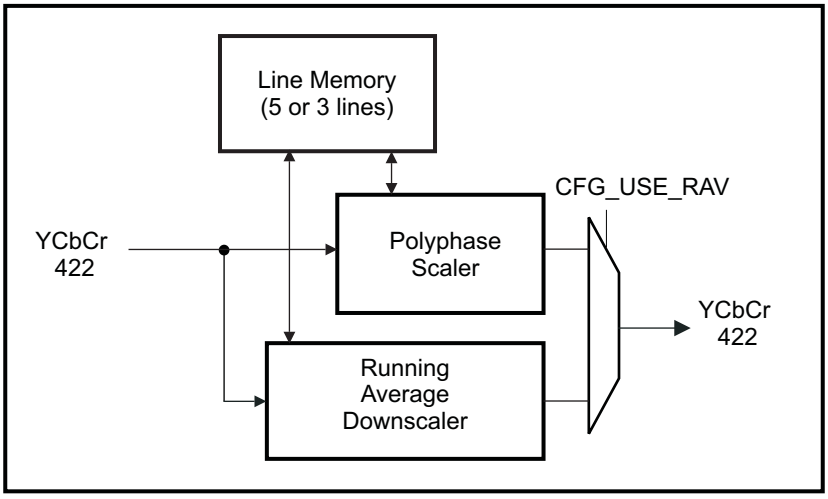
Figure 10-22. Peaking Filter at fs/4



10.3.4.2.3 Vertical Scaler

The vertical scaler has a polyphase (32-phase × 5-tap) filter and a running average filter as shown in Figure 10-23. While the polyphase filter can be used for any up-scaling and preferably downscaling to 3/16 scale factor, the running average filter is used only for downscaling to a ½ or less size. Selection between these two scalers is based on the user setting of VPE_CFG_SC0[4] CFG_USE_RAV parameter (CFG_USE_RAV= '0' for poliphase filter, and CFG_USE_RAV= '1' for running average filter), according to the user preference of the tradeoff between sharpness preserving and introduced artifacts.

Figure 10-23. Vertical Scaler Block Diagram



10.3.4.2.3.1 Running Average Filter

When a poly-phase filter is used, usually it has to have many taps in order to achieve acceptable quality for very small downscaling ratio, which requires the use of many line buffers. In VPE, there is a weighted Running Average filter for downscaling when the scaling factor is small (for example, when the scaling ratio is less than 0.5). This highly optimized design requires only one line buffer for luma and one for chroma, which still achieving acceptable quality. The output of the running average filter is based on weighted average of pixels in the current and previous rows in vertical direction. Initializations of accumulators affect the weights.

10.3.4.2.3.2 Vertical Scaler Configuration Parameters

Table 10-5. Vertical Scaler Configuration Parameters

Parameter	Typical Value	Controls	Description
VPE_CFG_SC0[10] CFG_INTERLACE_I		Frame or Field	0 = progressive, 1 = interlace
VPE_CFG_SC0[0] CFG_INTERLACE_O			0 = progressive 1 = interlace
VPE_CFG_SC0[3] CFG_INVT_FID			Invert field ID input
VPE_CFG_SC0[4] CFG_USE_RAV		Scaler Mode	0 = use polyphase scaler 1 = use running average scaler
VPE_CFG_SC1[26:0] CFG_ROW_ACC_INC		Bilinear & Polyphase Scalers	For progressive in/progressive out: round(2^{16}*(srcH-1)/(tarH - 1)) For progressive_in/interlace_out: round(2^{16}*2*(srcH-1)/(2*tarH - 1)) For interlace_in/progressive_out: round(2^{16}*(2*srcH-1)/(2*(tarH - 1))) For interlace_in/interlace_out: round(2^{16}*(2*srcH - 1)/(2*tarH - 1)) For interlace in/out, srcH/tarH are number of field lines as specified in VPE_CFG_SC4/VPE_CFG_SC5 descriptions.
VPE_CFG_SC2[27:0] CFG_ROW_ACC_OFFSET	0		Initial row accumulator value for progressive frame and top field
VPE_CFG_SC3[27:0] CFG_ROW_ACC_OFFSET_B	0		Initial row accumulator value for bottom field
VPE_CFG_SC13[27:24] CFG_DELTA_CHROMA_THR	4	Bilinear Scaler	Range for chroma soft switch based on pixel differences (max limit = 8)

Table 10-5. Vertical Scaler Configuration Parameters (continued)

Parameter	Typical Value	Controls	Description
VPE_CFG_SC13[21:12] CFG_CHROMA_INTP_THR	64		Threshold used in chroma soft switch based on pixel differences
VPE_CFG_SC13[9:0] CFG_SC_FACTOR_RAV		Running Average Scaler	Scale factor = round(1024 × tarH/srcH)
VPE_CFG_SC6[9:0] CFG_ROW_ACC_INIT_RAV			Initial row accumulator value for progressive frame and top field
VPE_CFG_SC6[19:10] CFG_ROW_ACC_INIT_RAV_B			Initial row accumulator value for bottom field

NOTE: Bi-linear scaler is not present in this device

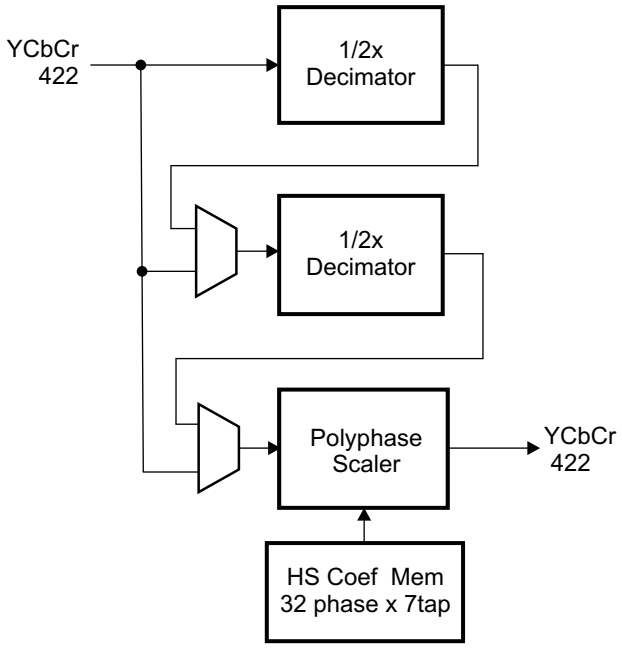
10.3.4.2.4 Horizontal Scaler

The Horizontal scaler is implemented using a 32-phase × 7-tap polyphase filter preceded by two sets of 1/2x decimators. The general configuration of the Horizontal scaler is performed as follows:

- For up scaling, the input video is interpolated using the polyphase scaler.
- For downscaling, it is recommended that input video is decimated by 2 until the modified scale factor falls between 1/2 and 1. Then, a polyphase filter is configured with coefficients selected based on the mod_scale_factor calculated as shown below from one of nine different sets of coefficients: (8,9,10,11,12,13,14,15,16)/16.

```
if  (scale_factor>=1/2)  {
   mux=0;  mod_scale_factor=scale factor;
} else if  (scale_factor>=1/4) {
   mux=1; mod_scale_factor=2*scale_factor;
} else {
   mux=2; mod_scale_factor=4*scale_factor;
}
```

Figure 10-24. Horizontal Scaler Block Diagram



In auto mode (CFG_AUTO_HS = '1'), scaler will operate as per above recommendation. In addition to this, for (CFG_AUTO_HS = '1'), polyphase filtering will be bypassed when (scale_factor = '1') or (scale_factor = ½) or (scale_fator == ¼). If CFG_AUTO_HS = '0') is used, user must provide proper values for dcm_2x, dcm_4x, proper values for all inputs to polyphase filter in the registers as well as appropriate coefficient sets. There is no constraint on polyphase filtering in terms of scaling ratio. However, there are constraints on input and output image width for scaler. Frame dimensions are limited from 64x64 to 2047x2047.

10.3.4.2.4.1 Half Decimation Filter

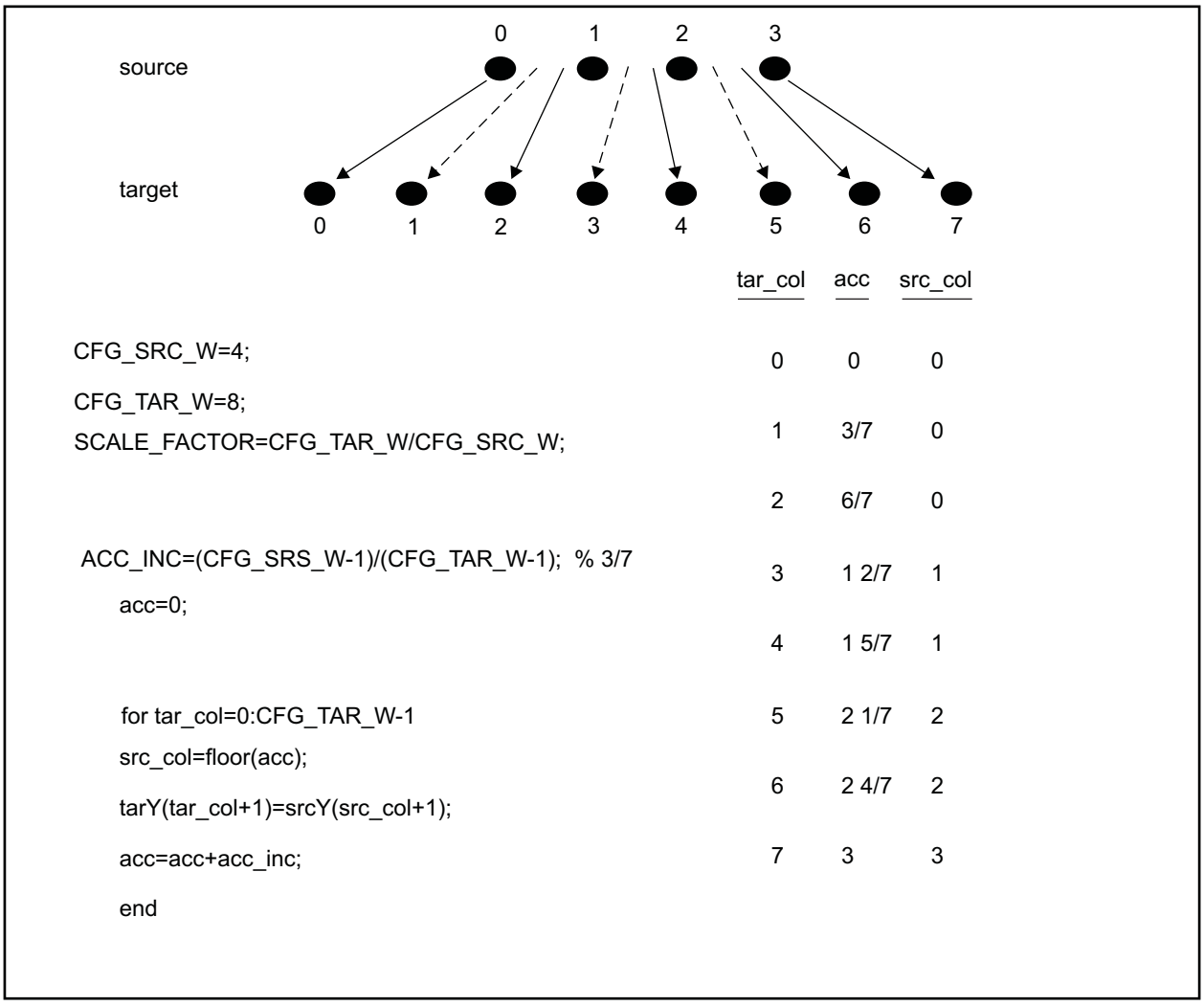
The half-decimation filter is an 11-tap filter with following coefficients: (14, 0, -29, 0, 79, 128, 79, 0, -29, 0, 14). For processing left and right edge pixels, the first and last data are repeated to pre-fill and extend the filter data pipeline respectively. These coefficients are hard-coded into scaler design and user cannot modify these.

10.3.4.2.4.2 Polyphase Filter

The horizontal scaling is done by stepping across the target row and interpolating target pixels based on source pixels. Accumulator points to the source pixel that corresponds to the target pixel.

Figure 10-25 shows an up-scaling example.

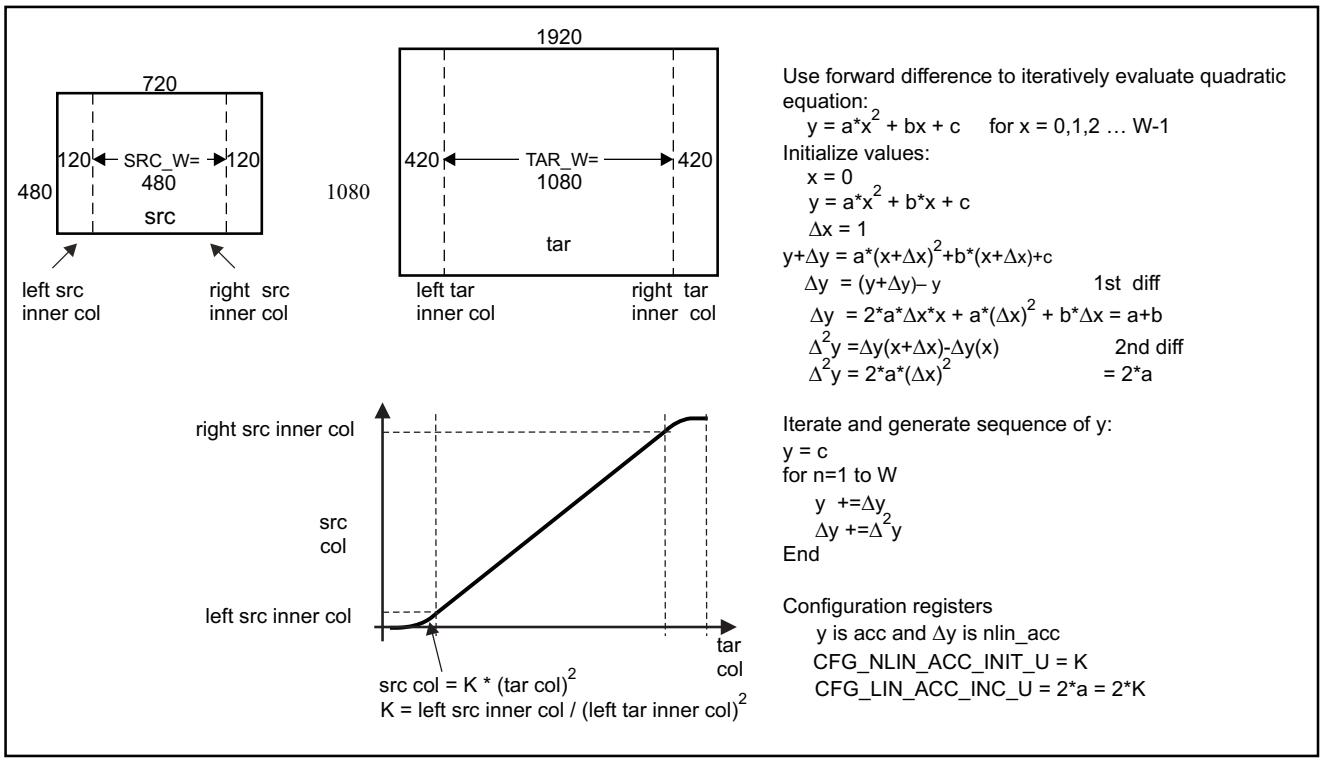
Figure 10-25. Polyphase Filtering Example



10.3.4.2.4.3 Nonlinear Horizontal Scaling

The horizontal scaler supports non-linear scaling to maintain the same aspect ratio for inner picture while stretching picture edges to fill the required resolution when capturing a 4 × 3 picture and fetching it as a 16 × 9 to memory. Non-linear scaling parameters are set with VPE_CFG_SC4[30:28] CFG_NLIN_ACC_INIT_U and VPE_CFG_SC4[26:24] CFG_LIN_ACC_INC_U registers. When setting up a non-linear scaling application, the inner picture is defined as the center square portion of the image with width and height equal to the height of the source image. Remaining side regions are then scaled non-linearly. Figure 10-26 shows a non-linear scaling case.

Figure 10-26. Non-linear Scaling Example



10.3.4.2.4.4 Horizontal Scaler Configuration Registers

Table 10-6. Register Group 1

Parameter	Controls	Description			
VPE_CFG_SC4[10:0] CFG_TAR_H	Image Dimension	Source Width			
VPE_CFG_SC4[22:12] CFG_TAR_W		Target Width			
VPE_CFG_SC0[1] CFG_LINEAR	Scaler Mode	If (linear == 1) SRC_Wi = SRC_W and TAR_Wi = TAR_W Else SRC_W= SRC_H and TAR_W = TAR_H			
VPE_CFG_SC0[2] CFG_SC_BYPASS		0 = enable scaler, 1 = bypass scaler			
VPE_CFG_SC0[6] CFG_AUTO_HS		CFG_AUTO_HS	CFG_DCM_2X	CFG_DCM_4X	Definition
		0	0	0	Polyphase scaling
VPE_CFG_SC0[7] CFG_DCM_2X		0	0	1	Horizontal decimation by 4 and polyphase scaling
		0	1	0	Horizontal decimation by 2 and polyphase scaling
VPE_CFG_SC0[8] CFG_DCM_4X		1	-	-	Automatic (selection of decimation filter is automatic)

Table 10-7. Register Group 2

Scale Factor	Decimation Usage	Control Register Bit
< 1/4	Decimation by 4	VPE_CFG_SC0[8] CFG_DCM_4X (set to 1 to enable decimation; disabled by default)
== 1/4	Decimation by 4	
1/4 < and < 1/2	Decimation by 2	VPE_CFG_SC0[7] CFG_DCM_2X (set to 1 to enable decimation; disabled by default)
== 1/2	Decimation by 2	
1/2 < and < 1	Bypassed	VPE_CFG_SC0[7] CFG_DCM_2X and CFG_DCM_4X (set to 0 to disable decimation; default value)
1	Bypassed	
> 1	Bypassed	

Table 10-8. Register Group 3

Parameter	Controls	Description
VPE_CFG_SC9[26:24] CFG_LIN_ACC_INC	Polyphase Scaler	if upscaling then CFG_LIN_ACC_INC = round(2^{24}*(srcWi-1)/(tarWi-1)) elseif downscaling CFG_LIN_ACC_INC= round(2^{24}*(srcWi/n-1)/(tarWi-1)) where n=2 or 4
VPE_CFG_SC8[10:0] CFG_NLIN_LEFT		if linear==1 CFG_NLIN_LEFT = 0 else CFG_NLIN_LEFT = (tarW - tarWi)/2
VPE_CFG_SC8[22:12] CFG_NLIN_RIGHT		if linear==1 CFG_NLIN_RIGHT = tarW-1 else CFG_NLIN_RIGHT = Ltar + tarWi – 1

VPE_CFG_SC5[26:24] CFG_NLIN_ACC_INC_U		if tarW/srcW >= 1 then
		d = 0
		if Ltar!=0 K = round[2^{24}*Lsrc/(Ltar*Ltar)] where Lsrc = (srcW-srcWi)/2
		else K = 0
		else
		d = (tarW-1)/2
		if Ltar!=0 K = round[2^{24} * Lsrc / (Ltar*(Ltar-2d))]
		where Lsrc= (srcW-srcWi)/(2n) and n=1,2 or 4
		else K = 0
		CFG_LIN_ACC_INC = 2*K (negative for downscaling)

VPE_CFG_SC4[30:28] CFG_NLIN_ACC_INIT_U		CFG_LIN_ACC_INC = K*(1-2*d)

NOTE: Source width and height variables for the polyphase filter are internally set with trimmer and decimation filter adjusted values.

10.3.4.2.5 Basic Configurations

Table 10-9 shows how the scaler should be configured based on the scale factor and the input/output mode.

Table 10-9. Scaler Configuration

Interlace		Mode[1]	VPE_CFG_SC5 [10:0] CFG_SRC_H mod_srcH	VPE_CFG_SC4 [10:0] CFG_TAR_H mod_tarH	Scale Factor
In	Out				
0	0	p->p	CFG_SRC_H	CFG_TAR_H	CFG_TAR_H/CFG_SRC_H
0	1	p->i	CFG_SRC_H	CFG_TAR_H/2	CFG_TAR_H/CFG_SRC_H
1	0	i->p	CFG_SRC_H/2	CFG_TAR_H	CFG_TAR_H/(CFG_SRC_H/2)
1	1	i->i	CFG_SRC_H/2	CFG_TAR_H/2	(CFG_TAR_H/2)/(CFG_SRC_H/2)

(1) p = progressive; i = interlaced

Table 10-10 shows how the vertical scaler should be configured based on the scale factor and the input/output mode.

Table 10-10. Vertical Scaler Configuration

Interlace		Mode[1]	VPE_CFG_SC9[26:24] CFG_ROW_ACC_INC/ 216	VPE_CFG_SC6[9 :0] CFG_ROW_ACC _INIT_RAV/216	VPE_CFG_SC6[19:10] CFG_ROW_ACC_INIT _RAV_B/216
In	Out			Top	Bot
0	0	p->p	(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	0
0	1	p->i	2*(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	(CFG_SRC_H-1)/(CFG_TAR_H-1)
1	0	i->p	1/2*(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	-0.5

(1) p = progressive; i = interlaced

Table 10-10. Vertical Scaler Configuration (continued)

1	1	i->i	(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	[(CFG_SRC_H-1)/(CFG_TAR_H-1)-1]/2

10.3.4.2.6 Coefficient Memory

10.3.4.2.6.1 Overview

The scaler requires initialization of eight coefficient SRAMS prior to processing a video frame. These coefficients are stored in the external DDR memories and downloaded by the VPDMA. The VPDMA writes the coefficient data through the VPI Control Interface bus.

The eight coefficient SRAMs are:

- HS horizontal polyphase scaler, Luma and Chroma (7tap)
- VS vertical polyphase scaler, Luma and Chroma (5tap or 3tap)

10.3.4.2.6.2 Physical Coefficient SRAM Layout

Each of the six legacy coefficient SRAMs is 32 phases × 91 bits. A single coefficient value is 13 bits. Therefore, one word of the SRAM contains 7 coefficient values as shown in Figure 10-27, and 224 coefficient values are stored in each SRAM.

Figure 10-27. SRAM Layout for 7tap Coefficient

Phase 0	C6	C5	C4	C3	C2	C1	C0
Phase 31	C223	C222	C221	C220	C219	C218	C217

The two vertical polyphase SRAMs are 32 phases × 65 bits. A single coefficient is 13 bits. Therefore, one word of SRAM contains 5 coefficient values as shown in Figure 10-28.

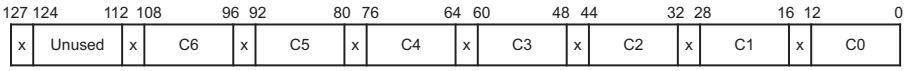
Figure 10-28. SRAM Layout for 5tap Coefficient

Phase 0	C4	C3	C2	C1	C0
Phase 31	C221	C220	C219	C218	C217

10.3.4.2.6.3 Scaler Coefficients Packing on 128-bit VPI Control I/F

A coefficient value is 13 bits wide and takes a single half word. Thus, one VPI Control write carries one phase's worth of coefficients. The last half-word in a quad-word is not used for 7tap coefficients.

Figure 10-29. VPI Control I/F Coef Data Format (7tap)



The Vertical Polyphase scaler coefficients have only five or three values per phase, so the last three (or five) entries are unused. The Vertical Polyphase coefficient packing is shown in Figure 10-30 and Figure 10-31.

Figure 10-30. VPI Control I/F Coef Data Format (5tap)

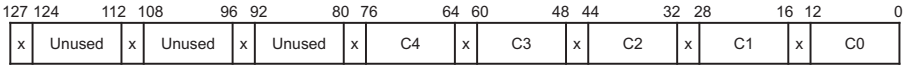
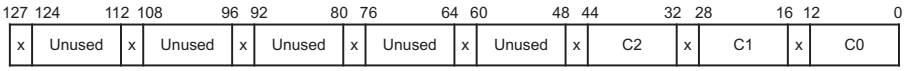


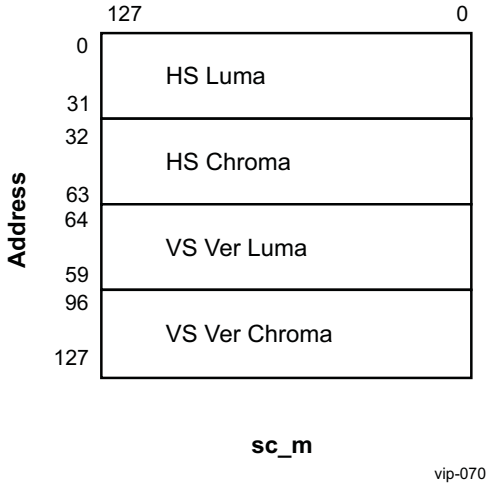
Figure 10-31. VPI Control I/F Coef Data Format (3tap)



10.3.4.2.6.4 VPI Control I/F Memory Map for Scaler Coefficients

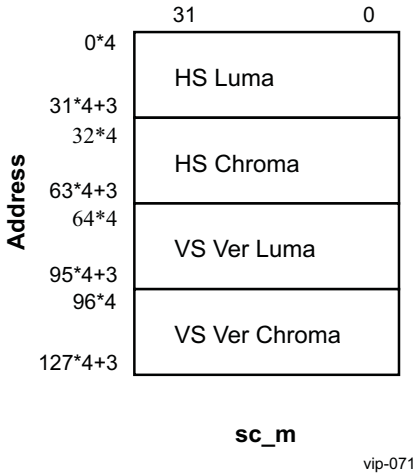
The memory map of the VPI Control I/F for the Scaler coefficients is shown in Figure 10-32. All coefficients can be filled up by a single VPI Control write command. The update address feature of VPI Control I/F enables individual access to a certain location of memories.

Figure 10-32. VPI Control I/F Memory Map (Write)



The module supports the VPI Control Read to read back the contents in the coefficient memories for debug purpose. Figure 10-33 shows the memory map of the VPI Control Read. As the VPI Control Read has only 32 bit data bus, it requires the word addressing while the write does the quad-word addressing.

Figure 10-33. VPI Control I/F Memory Map (Read)



10.3.4.2.6.5 VPI Control Interface

VPDMA is used to configure the coefficient memories of scaler through VPI control interface. Since the coefficient memories are not shadowed (unlike the memory mapped registers), VPI control write access needs to be done only during the gap between video frame processing times. If a write request is made while the Scaler is active, the access will be held off by the hardware until the last data of the currently processed frame is sent out. Care must be given to the order of the DMA descriptors so that blocking of VPI control bus does not occur.

10.3.4.2.6.6 Coefficient Table Selection Guide

The scaler filter coefficient tables are pre-generated using a MATLAB® program for various scaling factor ranges. Table 10-11 provides a general selection guide table for coefficient data files.

The mentioned .dat files are available in Section 10.3.4.4.

Table 10-11. Coefficient Data Files

Scaler	Scale Factor	Coeff table
HS Polyphase Filter	All upscaling	Section 10.3.4.4.1.1, *ppfcoef_scael_eq_1_32_phases_flip.dat*
	½ or ¼ down scaling	Section 10.3.4.4.1.1, ppfcoef_scale_eq_1_32_phases_flip.dat
	> 15/16	Section 10.3.4.4.1.9, *ppfcoef_scale_eq_15div16_32_phases_flip.dat*
	> 14/16	Section 10.3.4.4.1.8, *ppfcoef_scale_eq_14div16_32_phases_flip.dat*
	> 13/16	Section 10.3.4.4.1.7, *ppfcoef_scale_eq_13div16_32_phases_flip.dat*
	> 12/16	Section 10.3.4.4.1.6, *ppfcoef_scale_eq_12div16_32_phases_flip.dat*
	> 11/16	Section 10.3.4.4.1.5, *ppfcoef_scale_eq_11div16_32_phases_flip.dat*
	> 10/16	Section 10.3.4.4.1.4, *ppfcoef_scale_eq_10div16_32_phases_flip.dat*
	> 9/16	Section 10.3.4.4.1.3, *ppfcoef_scale_eq_9div16_32_phases_flip.dat*
	> 8/16	Section 10.3.4.4.1.2, *ppfcoef_scale_eq_8div16_32_phases_flip.dat*(1)

(1) HS Scaler has two sets of ½ decimator to perform downscaling ratios below ½ and ¼.

Table 10-11. Coefficient Data Files (continued)

Scaler	Scale Factor	Coeff table
VS Polyphase Filter	Upscaling	Section 10.3.4.4.2.1, *ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat*
	> 15/16	Section 10.3.4.4.2.6.8, *ppfcoef_scale_eq_15div16_32_phases_ver_5tap_flip.dat*
	> 14/16	Section 10.3.4.4.2.6.7, *ppfcoef_scale_eq_14div16_32_phases_ver_5tap_flip.dat*
	> 13/16	Section 10.3.4.4.2.6.6, *ppfcoef_scale_eq_13div16_32_phases_ver_5tap_flip.dat*
	> 12/16	Section 10.3.4.4.2.6.5, *ppfcoef_scale_eq_12div16_32_phases_ver_5tap_flip.dat*
	> 11/16	Section 10.3.4.4.2.6.4, *ppfcoef_scale_eq_11div16_32_phases_ver_5tap_flip.dat*
	> 10/16	Section 10.3.4.4.2.6.3, *ppfcoef_scale_eq_10div16_32_phases_ver_5tap_flip.dat*
	> 9/16	Section 10.3.4.4.2.6.2, *ppfcoef_scale_eq_9div16_32_phases_ver_5tap_flip.dat*
	> 8/16	Section 10.3.4.4.2.6.1, *ppfcoef_scale_eq_8div16_32_phases_ver_5tap_flip.dat*
	else	For down-scaling <8/16, RAV filter is recommended. In this case, coeffcients for vertical scaling need to be loaded.

10.3.4.3 SC Code

10.3.4.3.1 Generate Coefficient Memory Image

The following Perl script is used to generate a coefficient memory image for a given set of scaling factors.

```
#!/usr/local/bin/perl

#$dir="coef/";                    # directory which contains the coef files
#$cfg_file="sc_config1.cfg";      # configuration file name
#$spl_file="sc_config_supl.cfg"; # supplemental configuration file name
$cfg_file=$ARGV[0];               # configuration file name
$spl_file=$ARGV[1];               # supplemental configuration file name
$dir=$ARGV[2];                    # directory which contains the coef files

$coef_width=13;  # coef bit width
$coef_ntap=7;    # coef tap
$coef_nphase=32; # coef phase
$coef_norm=11;   # coef norm

#-------------------------------------------------------------
# read config file to get srcH/tarH/interlace_i/interlace_o
#-------------------------------------------------------------
open(INFILE, "<$cfg_file") or die "### ERROR: Cannot open $cfg_file";
while(<INFILE>){
  if (m/([-0-9]+) +\/\/ +srcW/) {
    $srcW = $1;
  } elsif (m/([-0-9]+) +\/\/ +srcH/) {
    $srcH = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarW/) {
    $tarW = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarH/) {
    $tarH = $1;
  } elsif (m/([-0-9]+) +\/\/ +interlace_in/) {
    $interlace_i = $1;
  } elsif (m/([-0-9]+) +\/\/ +interlace_out/) {
    $interlace_o = $1;
```

```
  }
}
close(INFILE);

#-------------------------------------------------------------
# read supplemental config file to get srcWi/tarWi from
#-------------------------------------------------------------
open(INFILE,"<$spl_file") or die "### ERROR: Cannot open $spl_file";
while(<INFILE>){
  if (m/([-0-9]+) +\/\/ +srcWi/) {
    $srcWi = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarWi/) {
    $tarWi = $1;
  } elsif (m/([-0-9]+) +\/\/ +profile/) {
    $profile = $1; # 0:HIGH,1:MEDIUM,2:LOW
  }
}
close(INFILE);
#-------------------------------------------------------------
# determine coef file based on the width/height
#-------------------------------------------------------------
#VS
#$vsc_file0 = "mod_ppfcoef_scale_eq_1_32_phases_flip.dat";
$vsc_file0  = "ppfcoef_scale_eq_1_32_phases_flip_PPF3_peak5_gain_eq_1_25.dat";
#VS VER
$mod_tarH = ($interlace_i == 0 && $interlace_o == 1)   $tarH<<1 : $tarH; if ($profile==2) {
  # LOW profile
  if ($mod_tarH >= $srcH) {
    $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_3tap_flip.dat";
} else {
    if ($mod_tarH >=($srcH>>1)) {
      $n = int(16.0*$mod_tarH/$srcH);
      $vsc_ver_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_ver_3tap_flip.dat",$n);
    } else {
      $n = 0;
      $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_3tap_flip.dat";
    }
  }
} else {
  if ($mod_tarH >= $srcH) {
    $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat";
  } else {
      $n = int(16.0*$mod_tarH/$srcH);
      $vsc_ver_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_ver_5tap_flip.dat",$n);
  }
}

# HS
if ($tarWi >= $srcWi) {
  $hsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip.dat";
} elsif ( ($tarWi == ($srcWi>>1)) || ($tarWi == ($srcWi>>2)) ) {
  $hsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip.dat";
} else {
if ($tarWi > ($srcWi>>1)) {
    $n = int(16.0*$tarWi/$srcWi);
  } elsif ($tarWi > ($srcWi>>2)) {
    $n = int(16.0*$tarWi/($srcWi>>1));
  } elsif ($tarWi >=($srcWi>>3)) {
    $n = int(16.0*$tarWi/($srcWi>>2));
  } else {
    $n = 0;
  }
  $hsc_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_flip.dat",$n);
}
#-------------------------------------------------------------
# write out the coef hex file
```

```
#-----------------------------------------------------------
&write_coef($hsc_file0);
&write_coef($hsc_file0);
&write_coef($vsc_ver_file0);
&write_coef($vsc_ver_file0);
&write_coef($vsc_file0);
&write_coef($vsc_file0);
sub write_coef {

  my ($filename) = @_;

  open(INFILE, "<$dir/$filename") or die "### ERROR: Cannot open $dir/$filename";

  $line=<INFILE>;
@val=split(' ',$line);
  $ntap=$val[0];
  $nphase=$val[1];
  $norm=$val[2];
for ($p=0;$p<$nphase;$p++) {
    $line=<INFILE>;@val=split(' ',$line);
    for($i=0;$i<$ntap;$i++) {
      if ($val[$i]<0) {
        $val[$i]+=(1<<$coef_width);
      }
    }
    undef(@coef);
    unshift(@coef, sprintf("%04x",$val[0]));
    unshift(@coef, sprintf("%04x",$val[1]));
    unshift(@coef, sprintf("%04x",$val[2]));
    unshift(@coef, sprintf("%04x",$val[3]));
    unshift(@coef, sprintf("%04x",$val[4]));
    unshift(@coef, sprintf("%04x",$val[5]));
    unshift(@coef, sprintf("%04x",$val[6]));
    unshift(@coef, sprintf("%04x",0));
    $coef=join("",@coef);
    print "$coef\n";
  }

  close(INFILE);
}
```

10.3.4.3.2 Scaler Configuration Calculation

The following C-code shows how configuration parameters are calculated:

```
// ======================================
// Required Input Parameter
// ======================================
//   srcW, srcH, tarW, tarH, srcWi, tarWi
//   input/output scan modes
//   Note:  srcH and tarH refer to number of lines in the frame even for interlace in/out
//          scaling.  Based on scaling scan mode input/output scan mode option,
//          heights are adjusted during internal calculations see mod_srcH and mod_tarH.


pixel_scale_factor=4;  // 10 bit pixel
hor_pixel_offset =0.0

  // ================================================================================================
  // Peaking Filter Configuration
  // ================================================================================================
  // ----------------------------------------------------------------
  // HPF Coef
  // ----------------------------------------------------------------
    y_peak_enable     =   0;
```

```
        peak_select=0; // 0=peak at fs/4  1=NTSC  2=PAL
        switch(peak_select) {
        case 0: {// peak at fs/4 and gain = 1
                 HPF_coef0         =   0;
                 HPF_coef1         =   0;
                 HPF_coef2         =   0;
                 HPF_coef3         =  -4;
                 HPF_coef4         =   0;
                 HPF_coef5         =   8; // mid tap
                 HPF_norm_shift    =   4;
             break;
    }
        case 1: {// NTSC: peak at 0.133*fs and gain=1
                 HPF_coef0         =  -2;
                 HPF_coef1         =  -8;
                 HPF_coef2         =  -8;
                 HPF_coef3         =  -2;
                 HPF_coef4         =  12;
                 HPF_coef5         =  16; // mid tap
                 HPF_norm_shift    =   6;
                 break;
        }
        case 2: {// PAL: peak at 0.163*fs and gain=1
                 HPF_coef0         =   2;
                 HPF_coef1         =  -4;
                 HPF_coef2         = -11;
                 HPF_coef3         =  -7;
                 HPF_coef4         =   9;
                 HPF_coef5         =  22; // mid tap
                 HPF_norm_shift    =   6;
                 break;
        }
        }

      // -----------------------------------------------------------------
      // NonLinear Coring Function typical values
      // -----------------------------------------------------------------
        NL_coring_thr     =  16;
        NL_limit          = 200;
        NL_lo_slope       =  16;
        NL_hi_thr         = 400;
        NL_hi_slope_shift =   4;



      // ================================================================================================
      // Edge Detection Configuration
      // ================================================================================================
        // edge detection
        confidence_default = 0; // 0  =use 5 tap polyphase filter for SC with ev_enable =0

        min_Gy_thr         =  64; // 64
        min_Gy_thr_range   =   3; // 3 power of 2
        gradient_thr       = 200; // 200
    gradient_thr_range =   6; // 6 power of 2

        ev_thr = int(4.0*3.111+0.5); // edge vector soft switch threshold (3.2)


      // ================================================================================================
      // vertical scaler configuration
      // ================================================================================================
      // -----------------------------------------------------------------
      // vertical scaler typical parameters
      // -----------------------------------------------------------------
        invert_field_ID           =  0; // invert field ID input
```

```
    delta_ev_thr              =  1; // edge vector soft switch range
    ver_pixel_offset          =  0.0;
    uv_intp_thr               =  pixel_scale_factor*16;
    delta_y_thr               =  4; // luma soft switch range
    delta_uv_thr              =  4; // chroma soft switch range
  //
  //
  // ------------------------------------------------------------------
  // Vertical Scaler Mode Determination
  // ------------------------------------------------------------------
  //
  //  interlace
  //   in   out   mode mod_srcH mod_tarH         scale
  // ----- ------ ---- -------- --------   -----------------
  //   0      0   p->p   srcH     tarH           tarH/srcH
  //   0      1   p->i   srcH     tarH>>1        tarH/srcH
  //   1      0   i->p   srcH>>1  tarH           tarH/(srcH/2)
  //   1      1   i->i   srcH>>1  tarH>>1    (tarH/2)/(srcH/2)

  if (interlace_in)  mod_srcH=srcH>>1; // interlace
  else               mod_srcH=srcH;    // progressive
  if (interlace_out) mod_tarH=tarH>>1; // interlace
  else               mod_tarH=tarH;    // progressive

  // determine vertical scaler
  if ((interlace_in==0)&&(interlace_out==1)) {
    if (tarH>((1+srcH)>>1)) use_rav = 0;   // 1=use RAV scaler 0=use polyphase scaler
    else                    use_rav = 1;
  } else {
    if (mod_tarH>((1+mod_srcH)>>1)) use_rav = 0; // 1=use RAV scaler  0=use polyphase scaler
    else                            use_rav = 1;
}



  // ------------------------------------------------------------------
  // RAV or Polyphase parameters
  // ------------------------------------------------------------------
  if (use_rav) {  // downscale
    // ------------
    // --- RAV ----
    // ------------
    if (use_internal_defaults) enable_edge_detection =   0;

    if ((interlace_in==0)&&(interlace_out==1)) scale = double(tarH)/double(srcH);
    else                                       scale = double(mod_tarH)/double(mod_srcH);
    sc_factor_rav = int(1024.0*scale+0.5);
// Peter's method
    delta     = (1.0/scale-1.0)/2.0;
    int_part  = floor(delta);
    frac_part = delta-int_part;

    row_acc_init_rav   = int(1024*(scale+(1.0-
scale)/2.0)+0.5);                                        // top field
    row_acc_init_b_rav = int(1024*(scale+(1.0-2.0*frac_part)*(1.0-
(1.0+2.0*int_part)*scale)/2.0)+0.5); // bottom field

    row_acc_inc        = 0; // polyphase scaler
    row_acc_offset     = 0; // polyphase scaler
    row_acc_offset_b   = 0; // polyphase scaler

  } else { // upscale using polyphase scaler
    // ------------
    // --- PPF ----
    // ------------
    if (use_internal_defaults) enable_edge_detection =   1;
```

```
    sc_factor_rav      = 0;
    delta_rav          = 0;
    row_acc_init_rav   = 0;
    row_acc_init_b_rav = 0;

    // upscaler
    // interlace                              row acc init value
        //  in   out mode      row acc inc      top          bottom
    // ---- ---- ---- -------------------- ---  ----------------------
    //   0    0  p->p     (srcH-1)/(tarH-1)  0            0
    //   0    1  p->i   2*(srcH-1)/(tarH-1)  0    (srcH-1)/(tarH-1)
    //   1    0  i->p 1/2*(srcH-1)/(tarH-1)  0           -0.5
    //   1    1  i->i     (srcH-1)/(tarH-1)  0   [(srcH-1)/(tarH-1)-1]/2

    row_acc_offset   = int(65536.0*ver_pixel_offset +0.5);  // progressive or top field
    if (interlace_in) {
if (interlace_out) {
         row_acc_inc      =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5);
        row_acc_offset_b =  (int(65536.0/2.0*(double(srcH-1)/(double(tarH-1))-
                              1.0)+0.5))+row_acc_offset;
       } else { // progressive out
          row_acc_inc      = int(65536.0*double(srcH-1)/(2.0*double(tarH-1))+0.5);
            if ((-0.5+row_acc_offset)<0.0) round_factor=-0.5;
            else                           round_factor= 0.5;
            row_acc_offset_b = int(65536.0*(-0.5)+round_factor)+row_acc_offset;
       }
    } else {   // progressive in
       if (interlace_out) {
        row_acc_inc      = int(65536.0*2.0*double(srcH-1)/double(tarH-1)+0.5);
        row_acc_offset_b =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5)+row_acc_offset;
       } else { // progressive out
          row_acc_inc      =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5);
           row_acc_offset_b = row_acc_offset;
       }
    }
  }
// ===============================================================================================
  // Horizontal Scaler configuration
  // ===============================================================================================
  // ---------------------------------------------------------------
  // horizontal scaler mode determination
  // ---------------------------------------------------------------
  auto_hs                = 1;
  dcm_2x                 = 0;
  dcm_4x                 = 0;
  hp_bypass              = 0;
  if (srcWi==srcW) linear = 1;
  else             linear = 0;

  // hor scaler parameters
  if (tarW>srcW) {               // upscale
    mod_srcW  = srcW;
    mod_srcWi = srcWi;
} else if (tarW<=(srcW>>2)) {  // downscale by <=1/4
    mod_srcW  = srcW>>2;
    mod_srcWi = srcWi>>2;
  } else if (tarW<=(srcW>>1)) {  // downscale by <=1/2
    mod_srcW  = srcW>>1;
    mod_srcWi = srcWi>>1;
  } else {                       // downscale by <=1
    mod_srcW  = srcW;
    mod_srcWi = srcWi;
  }

  // Not used any more:
```

```
  // hs_factor       = int(16.0*double(tarWi)/double(mod_srcWi)+0.5);  // hor scale factor (6.4)

  // -------------------------------------------------------------
  // Horizontal PolyPhase Settings  --
  // -------------------------------------------------------------
  lin_acc_inc     = int(16777216.0*double(mod_srcWi-1)/double(tarWi-1)+0.5);
  col_acc_offset  = int(16777216.0*hor_pixel_offset +0.5);
  nlin_left       = (tarW-tarWi)>>1;
  nlin_right      = nlin_left+tarWi-1;
  if (linear) {
    nlin_acc_inc    = 0;
    nlin_acc_init   = 0;
  } else {
     // ---------------------------------
     // Non-linear scaling configuration
     // ---------------------------------
    nlin_left_src   = (mod_srcW-mod_srcWi)>>1;

        if (tarWi>=srcWi) {    // upscale
      d             =  0.0;
      round_factor  =  0.5;
        } else {                // downscale
      d             = (double(tarW)-1.0)/2.0;
      round_factor  = -0.5;
    }

      K             = 16777216.0*double(nlin_left_src)/(double(nlin_left)*double(nlin_left-
2.0*d));
    nlin_acc_inc  = int(2.0*K+round_factor);
    nlin_acc_init = int(K*(1.0-2.0*d)+0.5);
  }
nlin_left_tar  = nlin_left;
  nlin_right_tar = nlin_right;

  // ============================================================================================
  // Bypass Determination
  // ============================================================================================
  // bypass
  if ((srcW==tarW)&&(srcWi==tarWi)&&(mod_srcH==mod_tarH)) sc_bypass = 1;
  else                                                   sc_bypass = 0;
  //

}
```

10.3.4.3.3 Typical Configuration Values

The following is the list of all scaler register fields that are set to constant values, representing typical settings:

VPE_CFG_SC0[3] CFG_INVT_FID = 0 (Field ID will be used without inversion)

VPE_CFG_SC0[5] CFG_ENABLE_EV = 1 (Field ID will be used without inversion)

VPE_CFG_SC0[6] CFG_AUTO_HS = 1 (The hardware will automatically decide, if current operation is up or down scaling. In down-scaling, it will also decide, if 2X or 4X decimation filter is needed)

VPE_CFG_SC0[7] CFG_DCM_2X = 0 (The 2X decimation filter is disabled)

VPE_CFG_SC0[8] CFG_DCM_4X = 0 (The 4X decimation filter is disabled)

VPE_CFG_SC0[11] CFG_ENABLE_SIN2_VER_INTP = 1 (Modified bilinear interpolation is used)

VPE_CFG_SC0[14] CFG_Y_PK_EN = 0 (Luma peaking is disabled)

VPE_CFG_SC0[15] CFG_TRIM= 1 (Trimming is enabled)

VPE_CFG_SC12[24:0] CFG_COL_ACC_OFFSET = 0 (No horizontal offset is involved)

VPE_CFG_SC13[21:12] CFG_CHROMA_INTP_THR = 64 (If the difference is less than this threshold, the interpolation of chroma should be done along edge direction. Otherwise, the interpolation of chroma should be done vertically)

VPE_CFG_SC13[27:24] CFG_DELTA_CHROMA_THR = 4 (max limit=8)

VPE_CFG_SC18[24:16] CFG_CONF_DEFAULT = 0x100 (Defines confidence factor when edge detection is disabled (VPE_CFG_SC0[5] CFG_ENABLE_EV bit = 0))
VPE_CFG_SC19 = 0xFC000000
VPE_CFG_SC20 = 0x0C840800
VPE_CFG_SC21 = 0x00100010
VPE_CFG_SC22 = 0x00040190

10.3.4.4 SC Coefficient Data Files

10.3.4.4.1 HS Polyphase Filter Coefficients

10.3.4.4.1.1 ppfcoef_scale_eq_1_32_phases_flip.dat

```
 7  32 11
31 -112  210 1790  210 -112   31
28  -98  159 1787  264 -126   34
25  -84  111 1779  320 -140   37
22  -71   65 1767  379 -154   40
19  -58   23 1750  439 -168   43
16  -45  -17 1728  502 -181   45
14  -33  -53 1701  565 -193   47
11  -22  -86 1670  631 -205   49
 9  -11 -116 1635  696 -216   51
 7   -1 -142 1594  763 -225   52
 5    8 -166 1551  830 -233   53
 3   16 -186 1504  898 -240   53
 2   23 -204 1455  965 -245   52
 1   30 -218 1401 1031 -248   51
 0   35 -230 1345 1097 -249   50
-1   40 -238 1286 1162 -248   47
44 -244 1224 1224 -244   44    0
47 -248 1162 1286 -238   40   -1
50 -249 1097 1345 -230   35    0
51 -248 1031 1401 -218   30    1
52 -245  965 1455 -204   23    2
53 -240  898 1504 -186   16    3
53 -233  830 1551 -166    8    5
52 -225  763 1594 -142   -1    7
51 -216  696 1635 -116  -11    9
49 -205  631 1670  -86  -22   11
47 -193  565 1701  -53  -33   14
45 -181  502 1728  -17  -45   16
43 -168  439 1750   23  -58   19
40 -154  379 1767   65  -71   22
37 -140  320 1779  111  -84   25
34 -126  264 1787  159  -98   28
```

10.3.4.4.1.2 ppfcoef_scale_eq_8div16_32_phases_flip.dat

```
7  32 11
  -28   61  542  898  542   61  -28
  -27   52  523  899  560   70  -29
  -26   44  505  898  578   79  -30
  -25   37  487  895  595   89  -30
  -24   30  468  892  613  100  -31
  -22   23  450  887  630  111  -31
  -21   17  432  883  647  122  -32
  -20   11  414  877  664  134  -32
  -19    6  396  871  680  146  -32
  -18    1  378  864  695  159  -31
  -16   -4  360  856  711  172  -31
  -15   -8  343  847  726  185  -30
  -14  -12  325  838  740  200  -29
```

```
  -13  -15  308  828  754  214  -28
  -12  -18  292  816  768  229  -27
  -10  -21  275  805  780  244  -25
  -23  258  789  789  258  -23    0
  -25  244  780  805  275  -21  -10
  -27  229  768  816  292  -18  -12
  -28  214  754  828  308  -15  -13
  -29  200  740  838  325  -12  -14
  -30  185  726  847  343   -8  -15
  -31  172  711  856  360   -4  -16
  -31  159  695  864  378    1  -18
  -32  146  680  871  396    6  -19
  -32  134  664  877  414   11  -20
  -32  122  647  883  432   17  -21
  -31  111  630  887  450   23  -22
  -31  100  613  892  468   30  -24
  -30   89  595  895  487   37  -25
  -30   79  578  898  505   44  -26
  -29   70  560  899  523   52  -27
```

10.3.4.4.1.3 ppfcoef_scale_eq_9div16_32_phases_flip.dat

```
7  32 11
  -33    8  547 1004  547    8  -33
  -31    0  525 1003  570   16  -35
  -29   -7  503 1001  592   25  -37
  -27  -13  481  998  614   34  -39
  -26  -19  459  995  636   44  -41
  -24  -25  437  990  658   55  -43
  -22  -29  414  983  679   67  -44
  -20  -34  393  976  700   79  -46
  -18  -38  371  968  721   91  -47
  -17  -41  350  959  742  104  -49
  -15  -44  330  948  761  118  -50
  -13  -46  309  936  780  133  -51
  -12  -48  289  924  799  148  -52
  -11  -50  270  911  817  163  -52
   -9  -51  250  897  833  180  -52
   -8  -52  232  882  850  196  -52
  -52  213  863  863  213  -52    0
  -52  196  850  882  232  -52   -8
  -52  180  833  897  250  -51   -9
  -52  163  817  911  270  -50  -11
  -52  148  799  924  289  -48  -12
  -51  133  780  936  309  -46  -13
  -50  118  761  948  330  -44  -15
  -49  104  742  959  350  -41  -17
  -47   91  721  968  371  -38  -18
  -46   79  700  976  393  -34  -20
  -44   67  679  983  414  -29  -22
  -43   55  658  990  437  -25  -24
  -41   44  636  995  459  -19  -26
  -39   34  614  998  481  -13  -27
  -37   25  592 1001  503   -7  -29
  -35   16  570 1003  525    0  -31
```

10.3.4.4.1.4 ppfcoef_scale_eq_10div16_32_phases_flip.dat

```
7  32 11
  -30  -46  542 1116  542  -46  -30
  -28  -52  515 1115  570  -39  -33
  -25  -57  488 1113  597  -32  -36
  -23  -62  462 1109  624  -24  -38
  -20  -65  435 1104  650  -15  -41
  -18  -69  409 1097  678   -5  -44
  -16  -71  383 1089  704    6  -47
```

```
  -14  -74  358 1081  730   17  -50
  -12  -75  333 1070  756   29  -53
  -11  -76  309 1058  782   42  -56
   -9  -77  285 1045  806   56  -58
   -8  -77  262 1030  831   71  -61
   -6  -77  239 1015  855   86  -64
   -5  -76  218  997  877  103  -66
   -4  -75  196  980  899  120  -68
   -3  -74  176  961  920  138  -70
  -72  156  940  940  156  -72    0
  -70  138  920  961  176  -74   -3
  -68  120  899  980  196  -75   -4
  -66  103  877  997  218  -76   -5
  -64   86  855 1015  239  -77   -6
  -61   71  831 1030  262  -77   -8
  -58   56  806 1045  285  -77   -9
  -56   42  782 1058  309  -76  -11
  -53   29  756 1070  333  -75  -12
  -50   17  730 1081  358  -74  -14
  -47    6  704 1089  383  -71  -16
  -44   -5  678 1097  409  -69  -18
  -41  -15  650 1104  435  -65  -20
  -38  -24  624 1109  462  -62  -23
  -36  -32  597 1113  488  -57  -25
  -33  -39  570 1115  515  -52  -28
```

10.3.4.4.1.5 ppfcoef_scale_eq_11div16_32_phases_flip.dat

```
7  32 11
  -19  -94  522 1230  522  -94  -19
  -17  -98  490 1230  555  -90  -22
  -14 -100  458 1227  587  -85  -25
  -12 -102  427 1223  620  -79  -29
  -10 -103  397 1217  652  -73  -32
   -8 -104  367 1209  685  -65  -36
   -6 -104  337 1199  717  -56  -39
   -4 -103  309 1187  749  -47  -43
   -3 -102  281 1174  781  -36  -47
   -1 -100  253 1159  812  -24  -51
    0  -98  227 1142  843  -11  -55
    1  -96  201 1124  874    3  -59
    1  -93  177 1105  903   18  -63
    2  -90  153 1084  932   34  -67
    2  -87  131 1062  961   51  -72
    3  -83  109 1038  987   69  -75
  -79   89 1014 1014   89  -79    0
  -75   69  987 1038  109  -83    3
  -72   51  961 1062  131  -87    2
  -67   34  932 1084  153  -90    2
  -63   18  903 1105  177  -93    1
  -59    3  874 1124  201  -96    1
  -55  -11  843 1142  227  -98    0
  -51  -24  812 1159  253 -100   -1
  -47  -36  781 1174  281 -102   -3
  -43  -47  749 1187  309 -103   -4
  -39  -56  717 1199  337 -104   -6
  -36  -65  685 1209  367 -104   -8
  -32  -73  652 1217  397 -103  -10
  -29  -79  620 1223  427 -102  -12
  -25  -85  587 1227  458 -100  -14
  -22  -90  555 1230  490  -98  -17
```

10.3.4.4.1.6 ppfcoef_scale_eq_12div16_32_phases_flip.dat

```
7  32 11
   -3 -132  486 1346  486 -132   -3
```

```
  -1 -132  449 1345  524 -131   -6
   1 -131  413 1342  562 -130   -9
   3 -130  378 1336  600 -127  -12
   4 -128  343 1328  639 -123  -15
   5 -125  309 1319  677 -119  -18
   6 -122  277 1306  716 -113  -22
   7 -118  245 1292  754 -106  -26
   8 -114  214 1276  793  -98  -31
   8 -109  185 1257  831  -89  -35
   9 -105  156 1237  869  -78  -40
   9 -100  130 1214  906  -66  -45
   9  -94  104 1190  942  -53  -50
   9  -89   79 1165  978  -38  -56
   8  -83   56 1138 1012  -22  -61
   8  -78   35 1108 1046   -4  -67
 -72   15 1081 1081   15  -72    0
 -67   -4 1046 1108   35  -78    8
 -61  -22 1012 1138   56  -83    8
 -56  -38  978 1165   79  -89    9
 -50  -53  942 1190  104  -94    9
 -45  -66  906 1214  130 -100    9
 -40  -78  869 1237  156 -105    9
 -35  -89  831 1257  185 -109    8
 -31  -98  793 1276  214 -114    8
 -26 -106  754 1292  245 -118    7
 -22 -113  716 1306  277 -122    6
 -18 -119  677 1319  309 -125    5
 -15 -123  639 1328  343 -128    4
 -12 -127  600 1336  378 -130    3
  -9 -130  562 1342  413 -131    1
  -6 -131  524 1345  449 -132   -1
```

10.3.4.4.1.7 ppfcoef_scale_eq_13div16_32_phases_flip.dat

```
7  32 11
  14 -154  435 1458  435 -154   14
  15 -150  393 1458  477 -157   12
  16 -146  353 1454  521 -160   10
  16 -141  314 1447  565 -161    8
  17 -135  276 1436  609 -161    6
  17 -129  239 1425  654 -161    3
  17 -123  204 1410  699 -159    0
  16 -116  170 1393  745 -156   -4
  16 -109  137 1373  790 -151   -8
  16 -102  107 1350  835 -146  -12
  15  -94   77 1325  879 -138  -16
  14  -87   50 1298  924 -130  -21
  13  -80   24 1269  968 -119  -27
  12  -72    0 1238 1010 -107  -33
  11  -65  -22 1204 1053  -94  -39
  10  -58  -43 1169 1093  -78  -45
 -52  -62 1138 1138  -62  -52    0
 -45  -78 1093 1169  -43  -58   10
 -39  -94 1053 1204  -22  -65   11
 -33 -107 1010 1238    0  -72   12
 -27 -119  968 1269   24  -80   13
 -21 -130  924 1298   50  -87   14
 -16 -138  879 1325   77  -94   15
 -12 -146  835 1350  107 -102   16
  -8 -151  790 1373  137 -109   16
  -4 -156  745 1393  170 -116   16
   0 -159  699 1410  204 -123   17
   3 -161  654 1425  239 -129   17
   6 -161  609 1436  276 -135   17
   8 -161  565 1447  314 -141   16
  10 -160  521 1454  353 -146   16
```

```
   12 -157  477 1458  393 -150   15
```

10.3.4.4.1.8 ppfcoef_scale_eq_14div16_32_phases_flip.dat

```
 7  32 11
   27 -158  370 1570  370 -158   27
   27 -150  324 1568  417 -165   27
   26 -142  281 1563  465 -172   27
   25 -133  238 1555  515 -178   26
   24 -124  198 1543  565 -183   25
   23 -115  159 1527  616 -186   24
   22 -106  122 1510  667 -189   22
   21  -97   87 1489  719 -191   20
   19  -87   54 1464  772 -191   17
   18  -78   23 1437  824 -190   14
   16  -69   -6 1407  876 -187   11
   15  -60  -32 1373  927 -182    7
   13  -52  -57 1339  979 -176    2
   12  -44  -79 1300 1030 -168   -3
   11  -36  -99 1261 1079 -159   -9
    9  -28 -117 1218 1128 -147  -15
  -21 -134 1179 1179 -134  -21    0
  -15 -147 1128 1218 -117  -28    9
   -9 -159 1079 1261  -99  -36   11
   -3 -168 1030 1300  -79  -44   12
    2 -176  979 1339  -57  -52   13
    7 -182  927 1373  -32  -60   15
   11 -187  876 1407   -6  -69   16
   14 -190  824 1437   23  -78   18
   17 -191  772 1464   54  -87   19
   20 -191  719 1489   87  -97   21
   22 -189  667 1510  122 -106   22
   24 -186  616 1527  159 -115   23
   25 -183  565 1543  198 -124   24
   26 -178  515 1555  238 -133   25
   27 -172  465 1563  281 -142   26
   27 -165  417 1568  324 -150   27
```

10.3.4.4.1.9 ppfcoef_scale_eq_15div16_32_phases_flip.dat

```
 7  32 11
   33 -143  294 1680  294 -143   33
   31 -132  246 1678  345 -155   35
   30 -121  199 1671  398 -165   36
   27 -109  154 1661  452 -175   38
   25  -97  112 1647  508 -185   38
   23  -86   72 1629  564 -193   39
   21  -75   35 1607  622 -201   39
   19  -64    0 1580  681 -207   39
   17  -53  -32 1551  740 -213   38
   15  -43  -61 1518  799 -217   37
   13  -33  -88 1481  859 -219   35
   11  -24 -113 1442  919 -220   33
    9  -15 -134 1399  978 -219   30
    8   -7 -153 1354 1036 -217   27
    6    0 -170 1307 1094 -212   23
    5    7 -184 1257 1150 -205   18
   13 -196 1207 1207 -196   13    0
   18 -205 1150 1257 -184    7    5
   23 -212 1094 1307 -170    0    6
   27 -217 1036 1354 -153   -7    8
   30 -219  978 1399 -134  -15    9
   33 -220  919 1442 -113  -24   11
   35 -219  859 1481  -88  -33   13
   37 -217  799 1518  -61  -43   15
   38 -213  740 1551  -32  -53   17
```

```
  39 -207  681 1580    0  -64   19
  39 -201  622 1607   35  -75   21
  39 -193  564 1629   72  -86   23
  38 -185  508 1647  112  -97   25
  38 -175  452 1661  154 -109   27
  36 -165  398 1671  199 -121   30
  35 -155  345 1678  246 -132   31
```

10.3.4.4.2 VS Polyphase Filter Coefficients

10.3.4.4.2.1 ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat

```
5  32 11
  -47  177 1788  177  -47
  -40  133 1785  225  -55
  -33   91 1778  276  -64
  -27   53 1765  330  -73
  -21   18 1747  386  -82
  -15  -13 1722  445  -91
  -11  -41 1693  507 -100
   -7  -66 1660  570 -109
   -3  -88 1622  635 -118
    0 -107 1579  703 -127
    2 -122 1532  771 -135
    4 -135 1482  839 -142
    5 -145 1428  909 -149
    6 -153 1371  978 -154
    7 -158 1310 1047 -158
    7 -161 1247 1116 -161
 -162 1186 1186 -162    0
 -161 1116 1247 -161    7
 -158 1047 1310 -158    7
 -154  978 1371 -153    6
 -149  909 1428 -145    5
 -142  839 1482 -135    4
 -135  771 1532 -122    2
 -127  703 1579 -107    0
 -118  635 1622  -88   -3
 -109  570 1660  -66   -7
 -100  507 1693  -41  -11
  -91  445 1722  -13  -15
  -82  386 1747   18  -21
  -73  330 1765   53  -27
  -64  276 1778   91  -33
  -55  225 1785  133  -40
```

10.3.4.4.2.2 ppfcoef_scale_eq_3_32_phases_flip.dat

```
5,   32,   11,
  130,  515,  758,  515,  130,
  121,  503,  757,  528,  139,
  113,  490,  756,  541,  148,
  105,  477,  755,  553,  158,
   97,  464,  753,  566,  168,
   90,  451,  751,  578,  178,
   83,  437,  749,  590,  189,
   76,  424,  746,  602,  200,
   69,  411,  743,  614,  211,
   63,  398,  739,  626,  222,
   57,  386,  734,  637,  234,
   52,  373,  729,  648,  246,
   46,  360,  725,  659,  258,
   41,  347,  719,  670,  271,
   37,  335,  713,  680,  283,
   32,  322,  707,  690,  297,
  314,  710,  710,  314,    0,
```

```
  297,  690,  707,  322,   32,
  283,  680,  713,  335,   37,
  271,  670,  719,  347,   41,
  258,  659,  725,  360,   46,
  246,  648,  729,  373,   52,
  234,  637,  734,  386,   57,
  222,  626,  739,  398,   63,
  211,  614,  743,  411,   69,
  200,  602,  746,  424,   76,
  189,  590,  749,  437,   83,
  178,  578,  751,  451,   90,
  168,  566,  753,  464,   97,
  158,  553,  755,  477,  105,
  148,  541,  756,  490,  113,
  139,  528,  757,  503,  121};
```

10.3.4.4.2.3 ppfcoef_scale_eq_4_32_phases_flip.dat

```
5,   32,   11,
  116,  515,  786,  515,  116,
  107,  502,  785,  530,  124,
   99,  488,  784,  544,  133,
   92,  473,  783,  557,  143,
   85,  459,  781,  571,  152,
   78,  445,  778,  585,  162,
   71,  431,  775,  598,  173,
   65,  417,  772,  611,  183,
   59,  403,  767,  624,  195,
   53,  389,  763,  637,  206,
   48,  375,  758,  649,  218,
   43,  362,  752,  661,  230,
   38,  348,  747,  673,  242,
   34,  334,  740,  685,  255,
   30,  321,  733,  696,  268,
   26,  308,  726,  707,  281,
  298,  726,  726,  298,    0,
  281,  707,  726,  308,   26,
  268,  696,  733,  321,   30,
  255,  685,  740,  334,   34,
  242,  673,  747,  348,   38,
  230,  661,  752,  362,   43,
  218,  649,  758,  375,   48,
  206,  637,  763,  389,   53,
  195,  624,  767,  403,   59,
  183,  611,  772,  417,   65,
  173,  598,  775,  431,   71,
  162,  585,  778,  445,   78,
  152,  571,  781,  459,   85,
  143,  557,  783,  473,   92,
  133,  544,  784,  488,   99,
  124,  530,  785,  502,  107};
```

10.3.4.4.2.4 ppfcoef_scale_eq_5_32_phases_flip.dat

```
5,   32,   11,
   98,  515,  822,  515,   98,
   90,  500,  821,  531,  106,
   83,  484,  820,  547,  114,
   75,  469,  819,  562,  123,
   69,  453,  816,  577,  133,
   63,  438,  813,  592,  142,
   57,  422,  809,  607,  153,
   51,  407,  805,  622,  163,
   46,  391,  801,  636,  174,
   41,  376,  795,  650,  186,
   37,  361,  789,  664,  197,
```

```
 32,  347,  782,  678,  209,
 28,  332,  775,  691,  222,
 25,  317,  767,  704,  235,
 22,  303,  759,  716,  248,
 18,  289,  750,  729,  262,
278,  746,  746,  278,    0,
262,  729,  750,  289,   18,
248,  716,  759,  303,   22,
235,  704,  767,  317,   25,
222,  691,  775,  332,   28,
209,  678,  782,  347,   32,
197,  664,  789,  361,   37,
186,  650,  795,  376,   41,
174,  636,  801,  391,   46,
163,  622,  805,  407,   51,
153,  607,  809,  422,   57,
142,  592,  813,  438,   63,
133,  577,  816,  453,   69,
123,  562,  819,  469,   75,
114,  547,  820,  484,   83,
106,  531,  821,  500,   90};
```

10.3.4.4.2.5 ppfcoef_scale_eq_6_32_phases_flip.dat

```
5,   32,   11,
   77,  513,  868,  513,   77,
   70,  496,  867,  531,   84,
   63,  479,  866,  548,   92,
   57,  461,  864,  566,  100,
   51,  444,  861,  583,  109,
   46,  427,  857,  600,  118,
   41,  409,  853,  617,  128,
   36,  393,  847,  633,  139,
   32,  376,  841,  650,  149,
   28,  359,  835,  666,  160,
   24,  343,  827,  682,  172,
   21,  327,  819,  697,  184,
   18,  311,  810,  712,  197,
   15,  296,  800,  727,  210,
   13,  281,  790,  741,  223,
   11,  266,  779,  755,  237,
  253,  771,  771,  253,    0,
  237,  755,  779,  266,   11,
  223,  741,  790,  281,   13,
  210,  727,  800,  296,   15,
  197,  712,  810,  311,   18,
  184,  697,  819,  327,   21,
  172,  682,  827,  343,   24,
  160,  666,  835,  359,   28,
  149,  650,  841,  376,   32,
  139,  633,  847,  393,   36,
  128,  617,  853,  409,   41,
  118,  600,  857,  427,   46,
  109,  583,  861,  444,   51,
  100,  566,  864,  461,   57,
   92,  548,  866,  479,   63,
   84,  531,  867,  496,   70};
```

10.3.4.4.2.6 ppfcoef_scale_eq_7_32_phases_flip.dat

```
5,   32,   11,
   53,  510,  922,  510,   53,
   47,  490,  922,  529,   60,
   41,  470,  921,  549,   67,
   36,  451,  918,  569,   74,
   32,  431,  915,  588,   82,
```

```
  27,  412,  910,  608,   91,
  23,  393,  905,  627,  100,
  20,  374,  898,  646,  110,
  17,  356,  890,  665,  120,
  14,  337,  882,  684,  131,
  11,  320,  873,  702,  142,
   9,  302,  863,  720,  154,
   7,  285,  852,  737,  167,
   6,  269,  840,  753,  180,
   4,  253,  827,  770,  194,
   3,  237,  815,  785,  208,
 223,  801,  801,  223,    0,
 208,  785,  815,  237,    3,
 194,  770,  827,  253,    4,
 180,  753,  840,  269,    6,
 167,  737,  852,  285,    7,
 154,  720,  863,  302,    9,
 142,  702,  873,  320,   11,
 131,  684,  882,  337,   14,
 120,  665,  890,  356,   17,
 110,  646,  898,  374,   20,
 100,  627,  905,  393,   23,
  91,  608,  910,  412,   27,
  82,  588,  915,  431,   32,
  74,  569,  918,  451,   36,
  67,  549,  921,  470,   41,
  60,  529,  922,  490,   47};
```

10.3.4.4.2.6.1 ppfcoef_scale_eq_8div16_32_phases_ver_5tap_flip.dat

```
5  32 11
   28  502  988  502   28
   24  479  987  524   34
   19  457  985  547   40
   15  435  982  570   46
   12  413  978  592   53
    9  392  972  614   61
    6  371  965  637   69
    4  350  957  659   78
    2  330  948  680   88
    0  310  938  702   98
   -1  291  926  723  109
   -2  272  914  744  120
   -3  254  900  764  133
   -3  237  886  783  145
   -4  220  871  802  159
   -4  204  855  820  173
  188  836  836  188    0
  173  820  855  204   -4
  159  802  871  220   -4
  145  783  886  237   -3
  133  764  900  254   -3
  120  744  914  272   -2
  109  723  926  291   -1
   98  702  938  310    0
   88  680  948  330    2
   78  659  957  350    4
   69  637  965  371    6
   61  614  972  392    9
   53  592  978  413   12
   46  570  982  435   15
   40  547  985  457   19
   34  524  987  479   24
```

10.3.4.4.2.6.2 ppfcoef_scale_eq_9div16_32_phases_ver_5tap_flip.dat

```
5  32 11
    3  489 1064  489    3
    0  464 1062  515    7
   -3  439 1060  540   12
   -5  414 1056  566   17
   -7  390 1050  592   23
   -9  366 1044  618   29
  -10  343 1035  644   36
  -11  320 1025  670   44
  -12  298 1014  695   53
  -12  277 1001  720   62
  -12  256  987  745   72
  -12  236  972  769   83
  -12  217  956  792   95
  -11  199  938  815  107
  -10  181  920  837  120
  -10  165  900  859  134
  148  876  876  148    0
  134  859  900  165  -10
  120  837  920  181  -10
  107  815  938  199  -11
   95  792  956  217  -12
   83  769  972  236  -12
   72  745  987  256  -12
   62  720 1001  277  -12
   53  695 1014  298  -12
   44  670 1025  320  -11
   36  644 1035  343  -10
   29  618 1044  366   -9
   23  592 1050  390   -7
   17  566 1056  414   -5
   12  540 1060  439   -3
    7  515 1062  464    0
```

10.3.4.4.2.6.3 ppfcoef_scale_eq_10div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -20  470 1148  470  -20
  -22  442 1147  499  -18
  -23  413 1144  529  -15
  -24  386 1139  558  -11
  -24  359 1132  588   -7
  -24  333 1124  618   -3
  -24  308 1113  648    3
  -23  283 1101  678    9
  -23  260 1088  707   16
  -22  237 1072  737   24
  -21  215 1056  765   33
  -19  194 1037  793   43
  -18  174 1017  822   53
  -16  156  995  848   65
  -15  138  973  875   77
  -13  121  949  900   91
  105  919  919  105    0
   91  900  949  121  -13
   77  875  973  138  -15
   65  848  995  156  -16
   53  822 1017  174  -18
   43  793 1037  194  -19
   33  765 1056  215  -21
   24  737 1072  237  -22
   16  707 1088  260  -23
    9  678 1101  283  -23
    3  648 1113  308  -24
   -3  618 1124  333  -24
```

```
 -7  588 1132  359  -24
-11  558 1139  386  -24
-15  529 1144  413  -23
-18  499 1147  442  -22
```

10.3.4.4.2.6.4 ppfcoef_scale_eq_11div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -40  444 1240  444  -40
  -40  412 1240  476  -40
  -40  381 1236  510  -39
  -39  350 1231  544  -38
  -37  321 1223  577  -36
  -36  293 1212  612  -33
  -34  265 1200  646  -29
  -32  239 1185  681  -25
  -30  214 1169  715  -20
  -28  190 1150  750  -14
  -26  167 1130  783   -6
  -23  146 1107  816    2
  -21  126 1083  849   11
  -19  107 1057  882   21
  -17   90 1030  913   32
  -15   73 1002  943   45
   58  966  966   58    0
   45  943 1002   73  -15
   32  913 1030   90  -17
   21  882 1057  107  -19
   11  849 1083  126  -21
    2  816 1107  146  -23
   -6  783 1130  167  -26
  -14  750 1150  190  -28
  -20  715 1169  214  -30
  -25  681 1185  239  -32
  -29  646 1200  265  -34
  -33  612 1212  293  -36
  -36  577 1223  321  -37
  -38  544 1231  350  -39
  -39  510 1236  381  -40
  -40  476 1240  412  -40
```

10.3.4.4.2.6.5 ppfcoef_scale_eq_12div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -56  409 1342  409  -56
  -54  373 1342  445  -58
  -51  339 1337  482  -59
  -49  306 1330  521  -60
  -46  274 1321  559  -60
  -43  244 1308  598  -59
  -40  215 1293  638  -58
  -36  187 1275  678  -56
  -33  161 1255  718  -53
  -30  137 1233  757  -49
  -27  114 1208  797  -44
  -24   93 1182  836  -39
  -21   73 1152  875  -31
  -18   55 1122  912  -23
  -16   38 1090  950  -14
  -14   23 1056  986   -3
    9 1015 1015    9    0
   -3  986 1056   23  -14
  -14  950 1090   38  -16
  -23  912 1122   55  -18
  -31  875 1152   73  -21
  -39  836 1182   93  -24
```

```
  -44  797 1208  114  -27
  -49  757 1233  137  -30
  -53  718 1255  161  -33
  -56  678 1275  187  -36
  -58  638 1293  215  -40
  -59  598 1308  244  -43
  -60  559 1321  274  -46
  -60  521 1330  306  -49
  -59  482 1337  339  -51
  -58  445 1342  373  -54
```

10.3.4.4.2.6.6 ppfcoef_scale_eq_13div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -65  364 1450  364  -65
  -61  326 1448  404  -69
  -57  289 1443  445  -72
  -53  253 1435  488  -75
  -48  220 1423  531  -78
  -44  188 1408  576  -80
  -40  158 1390  621  -81
  -36  130 1370  666  -82
  -32  103 1346  713  -82
  -28   79 1320  758  -81
  -24   56 1290  805  -79
  -21   36 1259  850  -76
  -18   17 1224  896  -71
  -15    0 1188  940  -65
  -12  -15 1149  984  -58
  -10  -28 1109 1027  -50
  -40 1064 1064  -40    0
  -50 1027 1109  -28  -10
  -58  984 1149  -15  -12
  -65  940 1188    0  -15
  -71  896 1224   17  -18
  -76  850 1259   36  -21
  -79  805 1290   56  -24
  -81  758 1320   79  -28
  -82  713 1346  103  -32
  -82  666 1370  130  -36
  -81  621 1390  158  -40
  -80  576 1408  188  -44
  -78  531 1423  220  -48
  -75  488 1435  253  -53
  -72  445 1443  289  -57
  -69  404 1448  326  -61
```

10.3.4.4.2.6.7 ppfcoef_scale_eq_14div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -67  310 1562  310  -67
  -61  269 1559  353  -72
  -55  230 1553  398  -78
  -50  193 1543  445  -83
  -44  158 1529  493  -88
  -39  125 1512  543  -93
  -34   94 1491  594  -97
  -30   66 1468  645 -101
  -25   41 1439  697 -104
  -22   17 1408  751 -106
  -18   -4 1373  804 -107
  -15  -23 1336  857 -107
  -12  -40 1296  910 -106
   -9  -55 1253  962 -103
   -7  -67 1208 1013  -99
   -5  -78 1161 1064  -94
```

```
  -86 1110 1110  -86    0
  -94 1064 1161  -78   -5
  -99 1013 1208  -67   -7
 -103  962 1253  -55   -9
 -106  910 1296  -40  -12
 -107  857 1336  -23  -15
 -107  804 1373   -4  -18
 -106  751 1408   17  -22
 -104  697 1439   41  -25
 -101  645 1468   66  -30
  -97  594 1491   94  -34
  -93  543 1512  125  -39
  -88  493 1529  158  -44
  -83  445 1543  193  -50
  -78  398 1553  230  -55
  -72  353 1559  269  -61
```

10.3.4.4.2.6.8 ppfcoef_scale_eq_15div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -61  248 1674  248  -61
  -54  204 1673  293  -68
  -47  163 1665  342  -75
  -41  125 1654  392  -82
  -35   90 1638  445  -90
  -29   57 1618  499  -97
  -24   27 1593  556 -104
  -20    0 1565  613 -110
  -16  -24 1532  672 -116
  -12  -46 1495  732 -121
   -9  -65 1455  793 -126
   -6  -81 1411  854 -130
   -4  -95 1364  915 -132
   -2 -107 1315  975 -133
    0 -116 1262 1035 -133
    1 -123 1208 1094 -132
 -128 1152 1152 -128    0
 -132 1094 1208 -123    1
 -133 1035 1262 -116    0
 -133  975 1315 -107   -2
 -132  915 1364  -95   -4
 -130  854 1411  -81   -6
 -126  793 1455  -65   -9
 -121  732 1495  -46  -12
 -116  672 1532  -24  -16
 -110  613 1565    0  -20
 -104  556 1593   27  -24
  -97  499 1618   57  -29
  -90  445 1638   90  -35
  -82  392 1654  125  -41
  -75  342 1665  163  -47
  -68  293 1673  204  -54
```

10.3.4.4.2.6.9 ppcoef_scale_1x_ver_5tap.dat

```
5 32 11
0 0 2048 0 0
-40 133 1785 225 -55
-33 91 1778 276 -64
-27 53 1765 330 -73
-21 18 1747 386 -82
-15 -13 1722 445 -91
-11 -41 1693 507 -100
-7 -66 1660 570 -109
-3 -88 1622 635 -118
0 -107 1579 703 -127
```

```
2 -122 1532 771 -135
4 -135 1482 839 -142
5 -145 1428 909 -149
6 -153 1371 978 -154
7 -158 1310 1047 -158
7 -161 1247 1116 -161
162 1186 1186 -162 0
161 1116 1247 -161 7
158 1047 1310 -158 7
154 978 1371 -153 6
149 909 1428 -145 5
142 839 1482 -135 4
135 771 1532 -122 2
127 703 1579 -107 0
118 635 1622 -88 -3
109 570 1660 -66 -7
100 507 1693 -41 -11
-91 445 1722 -13 -15
-82 386 1747 18 -21
-73 330 1765 53 -27
-64 276 1778 91 -33
-55 225 1785 133 -40
```

10.3.4.4.3 VS (Bilinear Filter Coefficients)

10.3.4.4.3.1 ppfcoef_scale_eq_1_32_phases_flip_PPF3_peak5_gain_eq_1_25.dat

This is not applicable for this device

```
7  32 11
  -11 -106  190 1902  190 -106  -11
   -9 -105  153 1897  230 -105  -13
   -8 -105  121 1887  274 -105  -16
   -6 -104   91 1869  320 -104  -18
   -5 -102   65 1843  370 -102  -21
   -4 -101   42 1812  424 -101  -24
   -3  -99   20 1776  480  -99  -27
   -2  -96    3 1730  539  -96  -30
   -1  -93  -12 1679  602  -93  -34
   -1  -90  -26 1627  665  -90  -37
    0  -87  -37 1568  732  -87  -41
    0  -84  -46 1506  801  -84  -45
    0  -80  -54 1439  871  -80  -48
    0  -76  -60 1371  941  -76  -52
    1  -72  -65 1299 1013  -72  -56
    1  -68  -69 1227 1085  -68  -60
  -64  -64 1152 1152  -64  -64    0
  -60  -68 1085 1227  -69  -68    1
  -56  -72 1013 1299  -65  -72    1
  -52  -76  941 1371  -60  -76    0
  -48  -80  871 1439  -54  -80    0
  -45  -84  801 1506  -46  -84    0
  -41  -87  732 1568  -37  -87    0
  -37  -90  665 1627  -26  -90   -1
  -34  -93  602 1679  -12  -93   -1
  -30  -96  539 1730    3  -96   -2
  -27  -99  480 1776   20  -99   -3
  -24 -101  424 1812   42 -101   -4
  -21 -102  370 1843   65 -102   -5
  -18 -104  320 1869   91 -104   -6
  -16 -105  274 1887  121 -105   -8
  -13 -105  230 1897  153 -105   -9
```

10.3.5 VPE Color Space Converter (CSC)

The color space converter (CSC) module is used to convert video data from one color space to another with nine programmable integer multipliers.

10.3.5.1 CSC Features

- All parameters are programmable
- Each parameter is configurable in signed 13-bits
- Support for Bypass Mode

10.3.5.2 CSC Functional Description

The conversion between the different color spaces requires addition and multiplication operations on color and intensity components. The mathematical expression of the conversion can be written as:

```
Y   = A0*R + B0*G + C0*B + D0
Cb  = A1*R + B1*G + C1*B + D1
Cr  = A2*R + B2*G + C2*B + D2
```

Color space coeficients are set through the following registers:

- For luma component:
  - VPE_CSC00[12:0] A0
  - VPE_CSC00[28:16] B0
  - VPE_CSC01[28:16] C0
  - VPE_CSC04[27:16] D0
- For Cb component:
  - VPE_CSC01[28:16] A1
  - VPE_CSC02[12:0] B1
  - VPE_CSC02[27:16] C1
  - VPE_CSC05[11:0] D1
- For Cr component :
  - VPE_CSC03[12:0] A2
  - VPE_CSC03[27:16] B2
  - VPE_CSC04[12:0] C2
  - VPE_CSC05[27:16] D2

Using YUV to RGB conversion as an example: YUV represents one color space and RGB represents another color space. The conversion can be written in the matrix format shown in Figure 10-34.

Figure 10-34. Matrix Format

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} A0 & B0 & C0 \\ A1 & B1 & C1 \\ A2 & B2 & C2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} D0 \\ D1 \\ D2 \end{bmatrix}$$

Since HDTV and SDTV have different conversion requirements, both conversions of RGB-to-YCbCr and YCbCr-to-RGB are described. The details of derivations of these matrixes will be given in the following subsections.

10.3.5.2.1 HDTV Application

10.3.5.2.1.1 HDTV Application with Video Data Range

The two equations presented in this section are for the HDTV application. The chromaticity parameters are defined by ITU-R709 standard.

The input video data for these equations should be within the range that is defined for video application.

In an 8-bit system:

- Rd, Gd, Bd, and Yd will be in the range [16-235]
- Cb and Cr will be the range [16-240]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, and Yd will be in the range [64-940]
- Cb and Cr will be in the range [64-960]
- D = 4

Conversion from RGB to YCbCr:

Figure 10-35. Conversion from RGB to YCbCr

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4646 & -0.0468 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

Figure 10-36. Conversion from YCbCr to RGB

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5396 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -197 \\ 82 \\ -232 \end{bmatrix} D$$

10.3.5.2.1.2 HDTV Application with Graphics Data Range

The two equations presented in this section are for the HDTV application with graphics range data input. The main application is for computer graphics display. The chromaticity parameters are defined by ITU-R709 standard.

The input data ranges for these equations are as follows.

In an 8-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be the range [0-255]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be in the range [0-1023]
- D = 4

Conversion from RGB to YCbCr:

Figure 10-37. Conversion from RGB to YCbCr

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.1826 & 0.6142 & 0.0620 \\ -0.1006 & -0.3385 & 0.4392 \\ 0.4392 & -0.3990 & -0.0402 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

Figure 10-38. Conversion from YCbCr to RGB

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1.1644 & -0.0003 & 1.7927 \\ 1.1644 & -0.2132 & -0.5329 \\ 1.1642 & 2.1125 & -0.0001 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -248 \\ 77 \\ -289 \end{bmatrix} D$$

10.3.5.2.1.3 Quantized Coefficients for Color Space Converter in HDTV

This section quantizes all the coefficients of color space conversion based on a 10-bit system for HDTV application. These coefficients can be input to the registers of programmable color space converter. Refer to register section for the corresponding register setting.

Table 10-12. Quantized Coefficients of HDTV Application with Video Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VPE_CSC00[12:0] A0	0.2126	218	0x00DA	1	1024	0x0400
B0(13-bit)	VPE_CSC00[28:16] B0	0.7152	732	0x02DC	0	0	0x0000
C0(13-bit)	VPE_CSC01[28:16] C0	0.0722	74	0x004A	1.5396	1577	0x0629
A1(13-bit)	VPE_CSC01[28:16] A1	-0.1172	-120	0x1F88	1	1024	0x0400
B1(13-bit)	VPE_CSC02[12:0] B1	-0.3942	-404	0x1E6C	-0.1831	-187	0x1F45
C1(13-bit)	VPE_CSC02[27:16] C1	0.5114	524	0x020C	-0.4577	-469	0x1E2B
A2(13-bit)	VPE_CSC03[12:0] A2	0.5114	524	0x020C	1	1024	0x0400
B2(13-bit)	VPE_CSC03[27:16] B2	-0.4646	-476	0x1E24	1.8142	1858	0x0742
C2(13-bit)	VPE_CSC04[12:0] C2	-0.0468	-48	0x1FD0	0	0	0x0000
D0(12-bit)	VPE_CSC04[27:16] D0	0	0	0x000	-197	-788	0xCEC
D1(12-bit)	VPE_CSC05[11:0] D1	128	512	0x200	82	328	0x148
D2(12-bit)	VPE_CSC05[27:16] D2	128	512	0x200	-232	-928	0xC60

Table 10-13. Quantized Coefficients of HDTV Application with Graphics Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VPE_CSC00[12:0] A0	0.1826	187	0x00BB	1.1644	1192	0x04A8
B0(13-bit)	VPE_CSC00[28:16] B0	0.6142	629	0x0275	-0.0003	0	0x0000

Table 10-13. Quantized Coefficients of HDTV Application with Graphics Data Range (continued)

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Real Number Format**	**Quantized Format**	**Hex Format**
C0(13-bit)	VPE_CSC01[28:16] C0	0.062	63	0x003F	1.7927	1836	0x072C
A1(13-bit)	VPE_CSC01[28:16] A1	-0.1006	-103	0x1F99	1.1644	1192	0x04A8
B1(13-bit)	VPE_CSC02[12:0] B1	-0.3385	-347	0x1EA5	-0.2132	-218	0x1F26
C1(13-bit)	VPE_CSC02[27:16] C1	0.4392	450	0x01C2	-0.5329	-546	0x1DDE
A2(13-bit)	VPE_CSC03[12:0] A2	0.4392	450	0x01C2	1.1642	1192	0x04A8
B2(13-bit)	VPE_CSC03[27:16] B2	-0.399	-409	0x1E67	2.1125	2163	0x0873
C2(13-bit)	VPE_CSC04[12:0] C2	-0.0402	-41	0x1FD7	-0.0001	0	0x0000
D0(12-bit)	VPE_CSC04[27:16] D0	16	64	0x040	-248	-992	0xC20
D1(12-bit)	VPE_CSC05[11:0] D1	128	512	0x200	77	308	0x134
D2(12-bit)	VPE_CSC05[27:16] D2	128	512	0x200	-289	-1156	0xB7C

10.3.5.2.2 SDTV Application

10.3.5.2.2.1 SDTV Application with Video Data Range

The two equations presented in this section are for the SDTV application. The chromaticity parameters are defined by ITU-R601 standard.

The input video data for these equations should be within the range that is defined for video application.

In an 8-bit system:

- Rd, Gd, Bd, and Yd will be in the range [16-235]
- Cb and Cr will be the range [16-240]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, and Yd will be in the range [64-940]
- Cb and Cr will be in the range [64-960]
- D = 4

Conversion from RGB to YCbCr:

Figure 10-39. Conversion from RGB to YCbCr

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.172 & -0.339 & 0.511 \\ 0.511 & -0.428 & -0.083 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

Figure 10-40. Conversion from YCbCr to RGB

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1 & -0.0003 & 1.3717 \\ 1 & -0.3365 & -0.6984 \\ 1 & 1.7336 & -0.0016 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -176 \\ 132 \\ -222 \end{bmatrix} D$$

10.3.5.2.2.2 SDTV Application with Graphics Data Range

The two equations presented in this section are for the SDTV application with graphics range data input. The main application is for computer graphics display. The chromaticity parameters are defined by ITU-R601 standard.

The input data ranges for these equations are as following.

In an 8-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be the range [0-255]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be in the range [0-1023]
- D = 4

Conversion from RGB to YCbCr:

Figure 10-41. Conversion from RGB to YCbCr

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

Figure 10-42. Conversion from YCbCr to RGB

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1.1641 & -0.0018 & 1.5958 \\ 1.1641 & -0.3914 & -0.8135 \\ 1.1641 & 2.0178 & -0.0012 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -223 \\ 136 \\ -277 \end{bmatrix} D$$

10.3.5.2.2.3 Quantized Coefficients for Color Space Converter in SDTV

This section quantizes all the coefficients of color space conversion based on a 10-bit system for SDTV application. These coefficients can be input to the registers of programmable color space converter. Refer to register section for the corresponding register setting.

Table 10-14. Quantized Coefficients of SDTV Application with Video Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	Registers	Real Number Format	Quantized Format	Hex Format	Real Number Format	Quantized Format	Hex Format
A0(13-bit)	VPE_CSC00[12:0] A0	0.299	306	0x0132	1	1024	0x0400

Table 10-14. Quantized Coefficients of SDTV Application with Video Data Range (continued)

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Real Number Format**	**Quantized Format**	**Hex Format**
B0(13-bit)	VPE_CSC00[28:16] B0	0.587	601	0x0259	-0.0003	0	0x0000
C0(13-bit)	VPE_CSC01[28:16] C0	0.114	117	0x0075	1.3717	1405	0x057D
A1(13-bit)	VPE_CSC01[28:16] A1	-0.172	-176	0x1F50	1	1024	0x0400
B1(13-bit)	VPE_CSC02[12:0] B1	-0.339	-347	0x1EA5	-0.3365	-345	0x1EA7
C1(13-bit)	VPE_CSC02[27:16] C1	0.511	523	0x020B	-0.6984	-715	0x1D35
A2(13-bit)	VPE_CSC03[12:0] A2	0.511	523	0x020B	1	1024	0x0400
B2(13-bit)	VPE_CSC03[27:16] B2	-0.428	-438	0x1E4A	1.7336	1775	0x06EF
C2(13-bit)	VPE_CSC04[12:0] C2	-0.083	-85	0x1FAB	-0.0016	-2	0x1FFE
D0(12-bit)	VPE_CSC04[27:16] D0	0	0	0x000	-176	-704	0xD40
D1(12-bit)	VPE_CSC05[11:0] D1	128	512	0x200	132	528	0x210
D2(12-bit)	VPE_CSC05[27:16] D2	128	512	0x200	-222	-888	0xC88

Table 10-15. Quantized Coefficients of SDTV Application with Graphics Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB		
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VPE_CSC00[12:0] A0	0.257	263	0x0107	1.1641	1192	0x04A8
B0(13-bit)	VPE_CSC00[28:16] B0	0.504	516	0x0204	-0.0018	-2	0x1FFE
C0(13-bit)	VPE_CSC01[28:16] C0	0.098	100	0x0064	1.5958	1634	0x0662
A1(13-bit)	VPE_CSC01[28:16] A1	-0.148	-152	0x1F68	1.1641	1192	0x04A8
B1(13-bit)	VPE_CSC02[12:0] B1	-0.291	-298	0x1ED6	-0.3914	-401	0x1E6F
C1(13-bit)	VPE_CSC02[27:16] C1	0.439	450	0x01C2	-0.8135	-833	0x1CBF
A2(13-bit)	VPE_CSC03[12:0] A2	0.439	450	0x01C2	1.1641	1192	0x04A8
B2(13-bit)	VPE_CSC03[27:16] B2	-0.368	-377	0x1E87	2.0178	2066	0x0812
C2(13-bit)	VPE_CSC04[12:0] C2	-0.071	-73	0x1FB7	-0.0012	-1	0x1FFF
D0(12-bit)	VPE_CSC04[27:16] D0	16	64	0x040	-223	-892	0xC84
D1(12-bit)	VPE_CSC05[11:0] D1	128	512	0x200	136	544	0x220
D2(12-bit)	VPE_CSC05[27:16] D2	128	512	0x200	-277	-1108	0xBAC

10.3.5.3 CSC Bypass Mode

CSC module can be bypassed by setting VPE_CSC05[28] BYPASS bit-field to 1.

10.3.6 VPE Chroma Up-Sampler (CHR_US)

The chroma up-sampler (CHR_US) module is used to convert from YCbCr 4:2:0 data format input to YCbCr 4:2:2 format output.

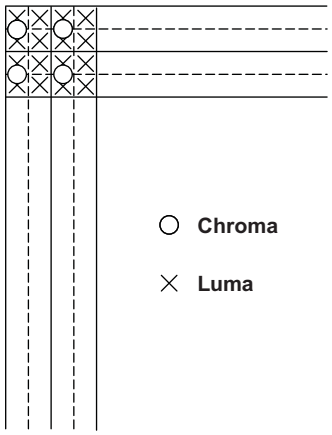
10.3.6.1 Features

- Supports both interlaced and progressive inputs
- 4-tap interpolation filtering
- Filter coefficients are all programmable
- Four sets of coefficients (each set has four coefficients) corresponding to anchor pixels and interpolated pixels of top field and bottom field
- For progressive inputs, the coefficients corresponding to top and bottom field must be identical
- Each coefficient is 14 bit (4.10 format)
- Default filter coefficients are based on Catmull-Rom algorithm
- Capable of removing the half pel vertical offset so all chroma samples are on-grid. This step ensures that the output does not suffer from any kind of rainbow effect due to chroma-upsampling.
- Provides a 10-bit interface in both directions: 10-bit input and 10-bit output
- Support bypass mode for 4:2:2 input

10.3.6.2 Functional Description

The YUV420 input to Chroma Upsampler module must be in the format shown in Figure 10-43, in which the chroma sample lies in the left column of a 2x2 pixel block with half pel vertical shift.

Figure 10-43. 4:2:0 YCrCb Color Space with Chroma Left-aligned



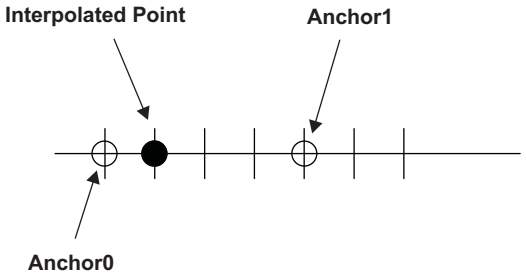
The upsampling is performed by an interpolation filter which uses Catmull-Rom algorithm. The Catmull-Rom Filter is based on four anchor pixels representing a four tap filter. The general 4-tap filter Catmull-Rom filter is defined in Figure 10-44.

Figure 10-44. 4:2:0 YCrCb Color Space with Chroma Left-aligned

$$[\,1 \quad x \quad x^2 \quad x^3\,] * \begin{bmatrix} 0 & 1 & 0 & 0 \\ -a & 0 & a & 0 \\ 2a & -3+a & 3-2a & -a \\ -a & 2-a & -(2-a) & a \end{bmatrix}$$

In the previous figure, x is the distance to the interpolated point between the two anchor points. In Figure 10-45, the example shows the desired interpolated point to be ¼ of the distance between Anchor0 and Anchor1. Thus, x = ¼.

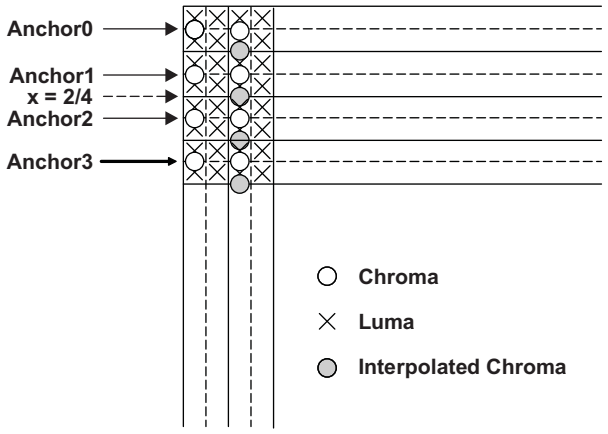
Figure 10-45. 4:2:0 YCrCb Color Space with Chroma Left-aligned



The variable 'a' determines the characteristics of the filter. a = ½ is generally used because the filter will produce an interpolated output that is an exact match to a linear input curve. In the literature, some people have noted that a=0.75 or a=1.0 may be more pleasing to the eye. In the implementation, the filter coefficients are programmable through MMR.

For the interpolated pixel, the variable x defines the positional offset relative to anchor pixel1. Figure 10-46 shows four anchor pixels with Anchor0 being near the top of the image and Anchor3 near the bottom. x is relative to Anchor1. Positive values of x goes down towards Anchor3. Negative x values imply a direction towards Anchor0. For example, if we want to interpolate a pixel midway between Anchor1 and Anchor2, x would be 2/4=1/2. There are four half pels between Anchor1 and Anchor2. Midway is two pels, so x=2/4.

Figure 10-46. Anchor Pixels



In the implementation, we need to interpolate the anchor pixel to get it on-grid. Then, we also need to interpolate a completely new pixel for YPrPb 4:2:2.

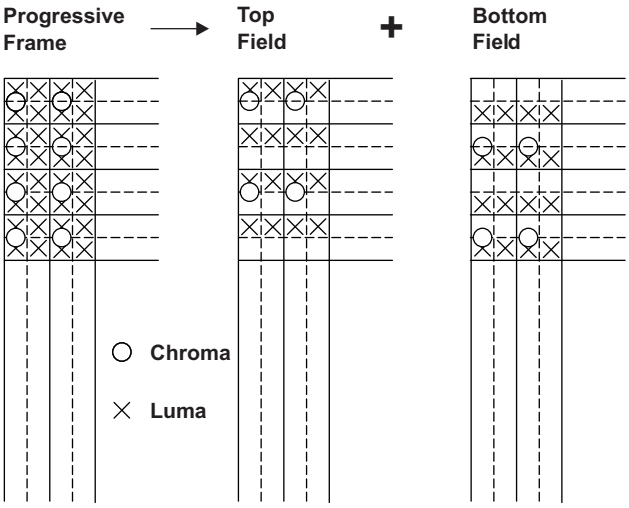
For a progressive input, x = -1/4 for getting the anchor pixel on-grid. x = 1/4 for generating the new pixel.

Lines are scanned from the top of the picture to the bottom.

10.3.6.3 For Interlaced YUV420 Input Data

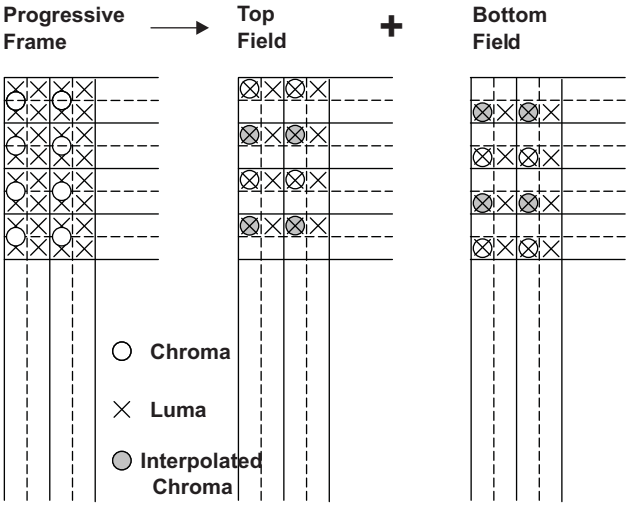
Figure 10-47 shows how 4:2:0 video is split into top and bottom fields in interlaced format. Chroma is attached to alternating Luma lines in the top and bottom fields such that color is equally spread out between the top field and the bottom field. This 4:2:0 interlaced chroma representation is shown in the following figure.

Figure 10-47. 4:2:0 Interlaced Scan



In each interlaced field, the chroma anchor points are separated by 4 pixel lines, or 8 half pels. The 4:2:2 chroma interpolated color space representation of interlaced pictures is shown in Figure 10-48.

Figure 10-48. Ideal 4:2:2 Chroma Upsampling for Interlaced Scan



Following interpolation, the chroma samples lie on a 4:2:2 grid.

Anchor pixels for the top field have been sited up by half a pel (x = -1/8). The new interpolated pixel is sited 3 half pels (x=3/8) down. Samples from the top field are distinct from samples in the bottom field.

For the bottom field, anchor pixels are sited 3 haf pels up (x = -3/8). The new interpolated pixel is sited 1 half pel down (x=1/8).

The chroma upsampling filter accepts different coefficients for the top field and the bottom field. In the case of progressive input, the coefficients for the top field and bottom field must be the same.

It should be noted that a different implementation could have been chosen to use the same coefficients for the top and bottom fields. Instead of pushing pixels from the top to bottom of a picture, it can be shown that pushing the bottom field through the upsampling filter from the bottom to the top of the picture permits using the same values for x as in the top field case.

10.3.6.4 Edge Effects

Several methods with increasing levels of difficulty resulting in increasing quality can be employed to deal with chroma pixels near the edges. In this module, the edge pixels can be mathematically approximated using the same filter as the rest of the picture. Edge pixels are duplicated going into the filter.

10.3.6.5 Modes of Operation (VPDMA)

In both primary (PRI) and auxiliary (AUX) paths, the mode in which the VPDMA needs to be operated depends whether the chroma upsampler and de-interlacers are enabled or not. Table 1-12 shows the modes of operation.

Table 10-16. VPDMA Modes of Operation

Mode A	Mode B
Input data is 4:2:0	Input data is 4:2:2

These modes need to be set in the following register bit-fields of particular instances being used:

VPE_REG0[17:16] CFG_MODE

LINE_MODE bitfield in VPDMA registers of format RD_LB_CLIENT_CTL_STATUS need to configured as follows:

Mode A corresponds to MMR value 0

Mode B corresponds to MMR value 1

The following are the VPDMA client related bit-fields that need to be configured:

VPE_PRI_CHROMA_CSTAT[9:8] LINE_MODE

VPE_PRI_FLD1_CHROMA_CSTAT[9:8] LINE_MODE

VPE_PRI_FLD2_CHROMA_CSTAT[9:8] LINE_MODE

NOTE: 1. For VPDMA luma clients, Mode B should always be used.

2. For VPDMA chroma clients, Mode A is used for 420 data and Mode B is used for 422 data.

10.3.6.6 Coefficient Configuration

The filter coefficients are left-aligned 14-bit binary values in signed Q4.10 format. The decimal point is between bits 9 and 10 using the convention of the least significant bit being at position zero. The most significant bit, 13, is the sign bit.

In the register map, the most significant nibble of the coefficient is the sign and the integer portion of the value. The next 10 bits represent the fractional portion of the coefficient value.

Chroma upsampling requires two sets of coefficients. Each coefficient set is comprised of four 14-bit Q4.10 values. One set is used for the top field of an interlaced picture, and the other set is used for the bottom field of an interlaced picture. For a progressive picture, both sets must be identical.

The coefficients and settings should be used for the follwoing video source types:

4:2:2 input (progressive or interlaced input)

VPDMA line mode = 1

VPE_REG0[17:16] CFG_MODE = 0x1 (mode B)

CHR_US coefficients are not used in this mode, so values are "don't care"

4:2:0 input (interlaced input):

VPDMA line mode = 0

VPE_REG0[17:16] CFG_MODE = 0x0 (mode A)

VPE_REG0[31:18] ANCHOR_FID0_C0 = 0x51

VPE_REG0[15:2] ANCHOR_FID0_C1 = 0x3d5

VPE_REG1[31:18] ANCHOR_FID0_C2 = 0x3fe3

VPE_REG1[15:2] ANCHOR_FID0_C3 = 0x3ff7

VPE_REG2[31:18] INTERP_FID0_C0 = 0x3fb5

VPE_REG2[15:2] INTERP_FID0_C1 = 0x2e9

VPE_REG3[31:18] INTERP_FID0_C2 = 0x18f

VPE_REG3[15:2] INTERP_FID0_C3 = 0x3fd3

VPE_REG4[31:18] ANCHOR_FID1_C0 = 0x16b

VPE_REG4[15:2] ANCHOR_FID1_C1 = 0x247

VPE_REG5[31:18] ANCHOR_FID1_C2 = 0xb1

VPE_REG5[15:2] ANCHOR_FID1_C3 = 0x3f9d

VPE_REG6[31:18] INTERP_FID1_C0 = 0x3fcf

VPE_REG6[15:2] INTERP_FID1_C1 = 0x3db

VPE_REG7[31:18] INTERP_FID1_C2 = 0x5d

VPE_REG7[15:2] INTERP_FID1_C3 = 0x3ff9

4:2:0 input (progressive input):

VPDMA line mode = 0

VPE_REG0[17:16] CFG_MODE = 0x0 (mode A)

VPE_REG0[31:18] ANCHOR_FID0_C0 = 0x00C8

VPE_REG0[15:2] ANCHOR_FID0_C1 = 0x0348

VPE_REG1[31:18] ANCHOR_FID0_C2 = 0x0018

VPE_REG1[15:2] ANCHOR_FID0_C3 = 0x3fd8

VPE_REG2[31:18] INTERP_FID0_C0 = 0x3fb8

VPE_REG2[15:2] INTERP_FID0_C1 = 0x0378

VPE_REG3[31:18] INTERP_FID0_C2 = 0x00e8

VPE_REG3[15:2] INTERP_FID0_C3 = 0x3fe8

VPE_REG4 to VPE_REG7 are not used so their values are "don't care".

10.3.7 VPE Chroma Down-Sampler (CHR_DS)

When the picture input is 4:2:2, the chroma must be downsampled to 4:2:0 before it is stored into DRAM for later compression by the imaging subsystem. The downsampling is performed by an averaging filter.

An array with chroma samples must be used- C_{IN}[] with range from 0 to N-1. (C_{IN}[0] is the topmost chroma sample and C_{IN}[N-1] is bottom chroma sample) .

The output array C_{OUT}(ranging from 0 to N/2 - 1) is calculated by the following formula:

$C_{OUT}[i] = CLIP[(C_{IN}[2i] + C_{IN}[2i+1]) / 2]CLIP[x]$ (x= 0 to 255).

This filter performs simple averaging of two input lines into one output line.

10.3.8 VPE YUV422 to YUV444 Conversion

As shown on Figure 10-2, prior the CSC the video format needs to be converted from YUV422 to YUV444 format. This conversion only applies to the Chroma samples in the color space, and is implemented using a 4-tap Catmull-Rom algorithm using the following equations:

$C_{OUT}[2*i[= C_{IN}[i]$ $C_{OUT}[2*i+1] = CLIP\ (9/16*C_{IN}[i] + 9/16*C_{IN}[i+1] – 1/16*C_{IN}[i-1] – 1/16C_{IN}[i+2] + 1)$

NOTE: Edge effects are treated by repeating the first and last pixel per line.

10.3.9 VPE Video Port Direct Memory Access (VPDMA)

10.3.9.1 VPDMA Introduction

The VPDMA primary function is to move data between external memory and internal processing modules that source or sink data. VPDMA is capable buffering this data and then delivering the data as demanded to the modules as programmed. The modules that source or sink data are referred to as clients. A channel is setup inside the VPDMA to connect a specific memory buffer to a specific client. The VPDMA centralizes the DMA control functions and buffering required to allow all the clients to minimize the effect of long latency times. The VPDMA also supports a descriptor based mode where lists of descriptors can be setup to configure all the channels as they become available.

Additionally, in a third-party configuration, the VPDMA is capable of performing DMA transfers as requested by the pulsing of an event strobe. The VPDMA is capable of generation of an address which may reach any location in the range for which it is configured. It is capable of moving this data either to or from a Shared Buffer and ultimately to or from a Client Buffer. For the two type of Client Buffers that are used to drive data into a subsystem (Streaming Buffer and Random Access Buffer) data is either pushed into the buffer or pulled out of the buffer dependent upon the direction of the data transfer. For third-party DMA operations the data is transferred into a Client buffer, and then transferred out of the Client Buffer to the transfer destination. These transfers are triggered by an Event pulse to each of the Client Buffers which are Routing Buffer types.

10.3.9.2 VPDMA Basic Definitions

10.3.9.2.1 Client

The modules that source or sink data are referred to as clients. The clients of the VPDMA are the physical between the processing modules (VPE) and external memory. A channel is the mechanism inside the VPDMA that connects a specific memory buffer or transfer to a specific client.

10.3.9.2.2 Channel

The VPDMA requires a channel to be setup for each group of transfers. All the channels are described through a Data Transfer Descriptor that has a common format. The client that the channel is mapped to interprets the information in the descriptor to perform the requested data transfer.

Each of the channels has a type of data that it can support based upon the client that it services. The VPDMA supports four types of channels:

- **YUV Channel** - Clients taking data YUV data
- **RGB Channel** - Clients taking RGB data
- **Miscellaneous Channel** - The Miscellaneous channel type is for any data type that is not a normal video type. The Miscellaneous channel type makes no assumptions on data type and just passes the data to the client and supports a single buffer for the client.
- **Free Channel** - Used in video compositions. The Free channel data type is always ignored as it uses the same data type of the descriptor that first calls the free channel.

10.3.9.2.3 List

A list is a group of descriptors that makes up a set of DMA transfers that need to be completed. The VPDMA supports two types of lists: a Regular List and a Self-Modifying List.

- The **Regular List** is a single list that the VPDMA will execute each descriptor once and initiate an interrupt when the list has completed. A regular list can contain any kind of descriptor without limitation and be of any size.

The VPDMA Controller works on lists of descriptors. In this mode the processor writes the lists of descriptors in the order it wants them executed. It then writes the location of the list to the VPE_LIST_ADDR register, followed by writing the size (bit LIST_SIZE) and type (bit LIST_TYPE) of the list, and list number (bit LIST_NUM) to the LIST_ATTR register. The List Manager module then schedules a DMA transfer to pull in the portion of the list that it can store in internal VPDMA memory. The List Manager will sequentially process the active list of descriptors until either the descriptor requires the use of a client that is currently active, or if it is waiting for the next portion of the list to be transferred from a DMA request. If there is no active list to process the list manager will go into an IDLE mode waiting for any client that is blocking a list or a list DMA transfer to complete.

All the DMA transfers are controlled by List Manager module inside VPDMA. List Manager needs to be loaded with FIRMWARE, before any DMA transfer from memory, after the VPDMA reset. The first MMR write to VPE_LIST_ADDR register after VPDMA reset should be the address of the memory buffer(128-bit aligned, that is, last four bits of the buffer address should be zero) where the firmware is stored. List Manager then schedules a DMA transaction to fetch the firmware and sets the list_attr.rdy bit after the firmware loading is complete.

10.3.9.2.4 Data Formats Supported

Following list summarizes the data formats supported in the VPDMA. For more inforamtion see Section 10.3.9.8, *VPDMA Data Formats*.

- RGB Data Types:
  - RGB16-565
  - ARGB-1555
  - ARGB-4444
  - RGBA-5551
  - RGBA-4444
  - ARGB24-6666
  - RGB24-888
  - ARGB32-8888
  - RGBA24-6666
  - RGBA32-8888
- YUV Data Types:
  - Y 4:4:4
  - Y 4:2:2
  - Y 4:2:0
  - C 4:4:4
  - C 4:2:2
  - C 4:2:0
  - CY 4:2:2
  - YCbC 4:4:4
  - YC 4:2:2

NOTE: VPDMA supports swapping formats (RGB/BGR and Cb/Cr)

10.3.9.3 VPDMA Client Buffering and Functionality

Table 10-17 lists for each client:

- The channels used, amount of buffering allocated for it, and the shared buffer used for its memory
- The line sizes it handles for tiled and non-tiled memory spaces, as well as any additional features it supports

Table 10-17. VPDMA Client Buffering and Functionality

Client	Channel(s)	Tiled Memory Max Line Size	Non-Tiled Memory Max Line Size	Additional Features
dei_hq_1_chroma	hq_vid1_chroma	1920 (color seperate) 960 (interleaved)	1920 (color seperate) 960 (interleaved)	Virtual Video Buffer with line buffer limitations TILED
dei_hq_1_luma	hq_vid1_luma	1920 (color seperate) 960 (interleaved)	4096	Virtual Video Buffer TILED
dei_hq_2_luma	hq_vid2_luma	1920 (color seperate) 960 (interleaved)	4096	Virtual Video Buffer TILED
dei_hq_2_chroma	hq_vid2_chroma	1920 (color seperate) 960 (interleaved)	1920 (color seperate) 960 (interleaved)	Virtual Video Buffer with line buffer limitations TILED
dei_hq_3_luma	hq_3_luma	1920 (color seperate) 960 (interleaved)	4096	Virtual Video Buffer TILED
dei_hq_3_chroma	hq_3_chroma	1920 (color seperate) 960 (interleaved)	1920 (color seperate) 960 (interleaved)	Virtual Video Buffer with line buffer limitations TILED
dei_hq_mv_in	hq_mv_in	Tiled Data Not Supported	4096	
dei_hq_mv_out	hq_mv_out	Tiled Data Not Supported	4096	
vip1_up_y	vip1_porta_luma vip1_porta_rgb	1920 (color seperate) 960 (interleaved)	4096	TILED
vip1_up_uv	vip1_porta_chroma	1920 (color seperate) 960 (interleaved)	4096	TILED

10.3.9.4 VPDMA Channels Assignment

Table 10-18 lists all of the channels in VPDMA and its base attributes. The Data Type column states what type of data YUV, RGB or OTHER the channel handles and in parentheses are the legal data type values that can be entered into a data transfer decriptor. The Client field states the name of the Client and in parentheses it states the reference number in Figure 10-2, *VPE Block Diagram*.

Table 10-18. VPDMA Channels Assignment

Channel	Description	Channel Number	Data Type	Client
hq_vid1_luma	High Quality DEI Video 420 Luma Data/ 422 Interleaved Data	0	YUV (0x1,0x2,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_1_luma (7)
hq_vid1_chroma	High Quality DEI Video 420 Chroma Data	1	YUV (0x5,0x6,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_1_chroma (7)
hq_vid2_luma	Low Cost DEI Field Minus 1420 Luma Data	2	YUV (0x1,0x2,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_2_luma (8)
hq_vid2_chroma	Low Cost DEI Field Minus 1 420 Chroma Data	3	YUV (0x5,0x6,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_2_chroma (8)

Table 10-18. VPDMA Channels Assignment (continued)

Channel	Description	Channel Number	Data Type	Client
hq_vid3_luma	Low Cost DEI Field Minus 2 420 Luma Data	4	YUV (0x1,0x2,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_3_luma (9)
hq_vid3_chroma	Low Cost DEI Field Minus 2 420 Chroma Data	5	YUV (0x5,0x6,0x7 If data type 0x7 is used half the data fetched will be thrown out.)	dei_hq_3_chroma (9)
hq_mv	Low Cost DEI Motion Vector	12	OTHER (4)	dei_hq_mv_in (6)
hq_mv_out	Low Cost DEI Motion Vector Write	15	OTHER (4)	dei_hq_mv_out (6)
vip1_porta_luma	Video Input 1 Port A 420 Data Luma	102	YUV (0x1, 0x2,0x7)	vip1_up_y (1)
vip1_porta_chroma	Video Input 1 Port A 420 Data Chroma	103	YUV (0x5,0x6,0x7)	vip1_up_uv (1)
vip1_porta_rgb	Video Input 1 Port A RGB Data	106	RGB (0x0-0x8)	vip1_up_y (1)

10.3.9.5 VPDMA Interrupts

The VPDMA has 4 interrupt group(s). Each group has an interrupt for all the client interrupts, an interrupt for every 32 channels, a interrupt for each list complete, an interrupt for each list notify and an interrupt for for all of the descriptor interrupts. Each of these groups can be individually masked so that only the interrupts specified will trigger the higher level interrupt.

Each interrupt source can be individually masked independently for each seperate interrupt group. A status register bit exists for each interrupt source for for each interrupt group, that is set whenever the interrupt event occurs even when if the interrupt is masked. The status register bit will remain set until cleared by software by writing a one to the status bit.

Table 10-19 shows all interrupt events from the VPDMA that go to VPE top level. The interrupt events are mapped to one interrupt line, INT0, that go to VPE top level.

Table 10-19. VPDMA Interrupt Events

Interrupt	Event Flag Registers	Event Mask Registers	Description
vpdma_int_channel_group0	VPE_INT0_CHANNEL0_INT_STAT	VPE_INT0_CHANNEL0_INT_MASK	An unmasked channel interrupt for interrupt group 0 in channel register 0 has fired.
vpdma_int_channel_group1	VPE_INT0_CHANNEL1_INT_STAT	VPE_INT0_CHANNEL1_INT_MASK	An unmasked channel interrupt for interrupt group 1 in channel register 0 has fired.
vpdma_int_channel_group2	VPE_INT0_CHANNEL2_INT_STAT	VPE_INT0_CHANNEL2_INT_MASK	An unmasked channel interrupt for interrupt group 2 in channel register 0 has fired.
vpdma_int_channel_group3	VPE_INT0_CHANNEL3_INT_STAT	VPE_INT0_CHANNEL3_INT_MASK	An unmasked channel interrupt for interrupt group 3 in channel register 0 has fired.
vpdma_int_channel_group4	VPE_INT0_CHANNEL4_INT_STAT	VPE_INT0_CHANNEL4_INT_MASK	An unmasked channel interrupt for interrupt group 4 in channel register 0 has fired.

Table 10-19. VPDMA Interrupt Events (continued)

Interrupt	Event Flag Registers	Event Mask Registers	Description
vpdma_int_channel_group5	VPE_INT0_CHANNEL5_INT_STAT	VPE_INT0_CHANNEL5_INT_MASK	An unmasked channel interrupt for interrupt group 5 in channel register 0 has fired.
vpdma_int_list0_complete			List 0 has completed
vpdma_int_list0_notify			The data transfer in list 0 with the Notify Field set in the descriptor has completed
vpdma_int_list1_complete			List 1 has completed
vpdma_int_list1_notify			The data transfer in list 1 with the Notify Field set in the descriptor has completed
vpdma_int_list2_complete			List 2 has completed
vpdma_int_list2_notify			The data transfer in list 2 with the Notify Field set in the descriptor has completed
vpdma_int_list3_complete			List 3 has completed
vpdma_int_list3_notify			The data transfer in list 3 with the Notify Field set in the descriptor has completed
vpdma_int_list4_complete	VPE_INT0_LIST0_INT_STAT	VPE_INT0_LIST0_INT_MASK	List 4 has completed
vpdma_int_list4_notify			The data transfer in list 4 with the Notify Field set in the descriptor has completed
vpdma_int_list5_complete			List 5 has completed
vpdma_int_list5_notify			The data transfer in list 5 with the Notify Field set in the descriptor has completed
vpdma_int_list6_complete			List 6 has completed
vpdma_int_list6_notify			The data transfer in list 6 with the Notify Field set in the descriptor has completed
vpdma_int_list7_complete			List 7 has completed
vpdma_int_list7_notify			The data transfer in list 7 with the Notify Field set in the descriptor has completed
vpdma_int_client	VPE_INT0_CLIENT0_INT_STAT VPE_INT0_CLIENT1_INT_STAT	VPE_INT0_CLIENT0_INT_MASK VPE_INT0_CLIENT1_INT_MASK	Client Interrupt
vpdma_int_descriptor	VPE_INT0_LIST0_INT_STAT	VPE_INT0_LIST0_INT_STAT	Descriptor Interrupt

In Table 10-19 above, the “channel_group”, “client” and “descriptor” interrupts are actually a set of additional interrupts. When software receives an interrupt from a “channel_group,” “client,” or “descriptor” it must read the appropriate register within the VPDMA (refer to Table 10-20 to determine what the actual interrupt was).

Table 10-20. VPE Interrupt Sources

Interrupt	Interrupt Group	Description
channel_hq_mv	channel_group0	The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager.
channel_hq_mv_out	channel_group0	The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager.
channel_hq_vid1_chroma	channel_group0	The last write DMA transaction has completed for channel hq_vid1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_hq_vid1_luma	channel_group0	The last write DMA transaction has completed for channel hq_vid1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_hq_vid2_chroma	channel_group0	The last write DMA transaction has completed for channel hq_vid2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_hq_vid2_luma	channel_group0	The last write DMA transaction has completed for channel hq_vid2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_hq_vid3_chroma	channel_group0	The last write DMA transaction has completed for channel hq_vid3_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_hq_vid3_luma	channel_group0	The last write DMA transaction has completed for channel hq_vid3_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_porta_chroma	channel_group3	The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_porta_luma	channel_group3	The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_porta_rgb	channel_group3	The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point.

Table 10-20. VPE Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
client_dei_hq_1_chroma	client	The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_1_luma	client	The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_2_chroma	client	The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_2_luma	client	The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_3_chroma	client	The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_3_luma	client	The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_mv_in	client	The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
client_dei_hq_mv_out	client	The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_up_uv	client	The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_up_y	client	The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vpi_ctl	client	The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
control_descriptor_int0	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0.
control_descriptor_int1	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1.
control_descriptor_int10	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10.
control_descriptor_int11	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11.

Table 10-20. VPE Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
control_descriptor_int12	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12.
control_descriptor_int13	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13.
control_descriptor_int14	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14.
control_descriptor_int15	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15.
control_descriptor_int2	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2.
control_descriptor_int3	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3.
control_descriptor_int4	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4.
control_descriptor_int5	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5.
control_descriptor_int6	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6.
control_descriptor_int7	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7.
control_descriptor_int8	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8.
control_descriptor_int9	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9.
list0_complete	list0_complete	List 0 has completed and a new list can be loaded.
list0_notify	list0_notify	A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel.
list1_complete	list1_complete	List 1 has completed and a new list can be loaded.
list1_notify	list1_notify	A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel.
list2_complete	list2_complete	List 2 has completed and a new list can be loaded.
list2_notify	list2_notify	A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel.
list3_complete	list3_complete	List 3 has completed and a new list can be loaded.
list3_notify	list3_notify	A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel.
list4_complete	list4_complete	List 4 has completed and a new list can be loaded.
list4_notify	list4_notify	A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel.
list5_complete	list5_complete	List 5 has completed and a new list can be loaded.
list5_notify	list5_notify	A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel.
list6_complete	list6_complete	List 6 has completed and a new list can be loaded.
list6_notify	list6_notify	A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel.
list7_complete	list7_complete	List 7 has completed and a new list can be loaded.
list7_notify	list7_notify	A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel.
other		Any channel that is not assigned to a specific client has completed.

10.3.9.6 VPDMA Descriptors

The VPDMA needs to be programmed through descriptors (a pre-defined structure of eight or four 32-bit words depending on type of descriptors) other than VPDMA Memory Mapped Registers (MMR). Descriptors are of three types:

(i) **Data Transfer Descriptors** - A memory structure used to describe a desired memory transaction to or from a client.

(ii) **Control Descriptors** - A memory structure used to perform a control operation inside the DMA controller

(iii) **Configuration Descriptors** - A memory structure used to described a setup that should be applied to an processing modules like MMR write, scalar coefficient write etc.

10.3.9.6.1 Data Transfer Descriptors

In order to set up data transfers from the VPDMA, a data transfer descriptor is added into a list. The fields used for an Inbound and Outbound descriptor vary slightly. An outbound transfer can have two different outbound transfers. The two transfers are the main data transfer and a write of an Inbound Descriptor. The created inbound descriptor is formatted so it can be read back in directly to the next channel specified in the original descriptor.

Figure 10-49. Inbound Data Transfer Descriptor Format

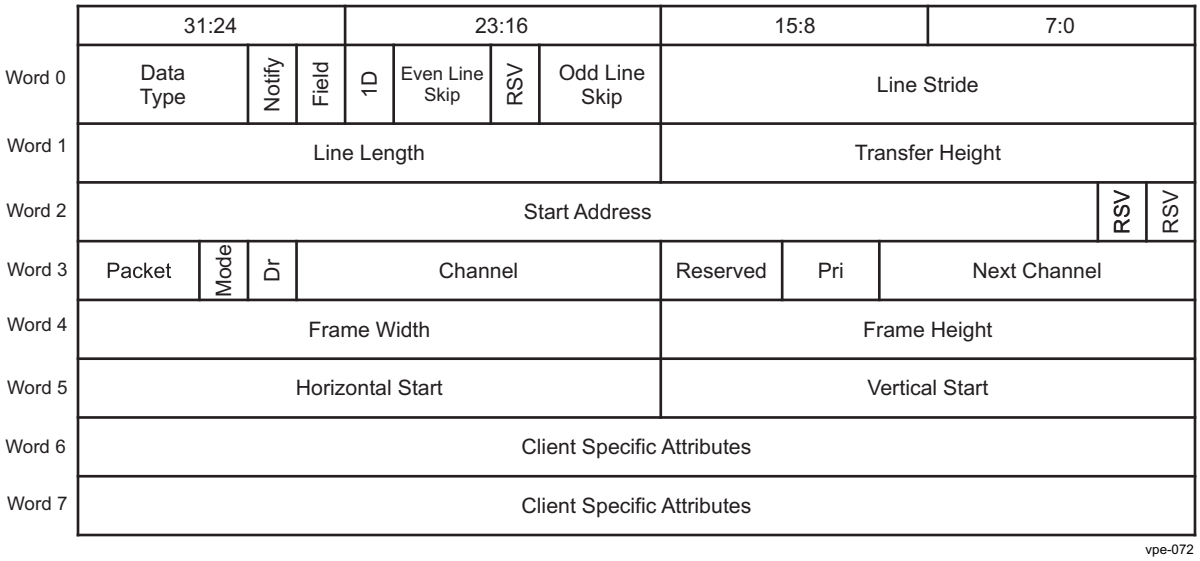
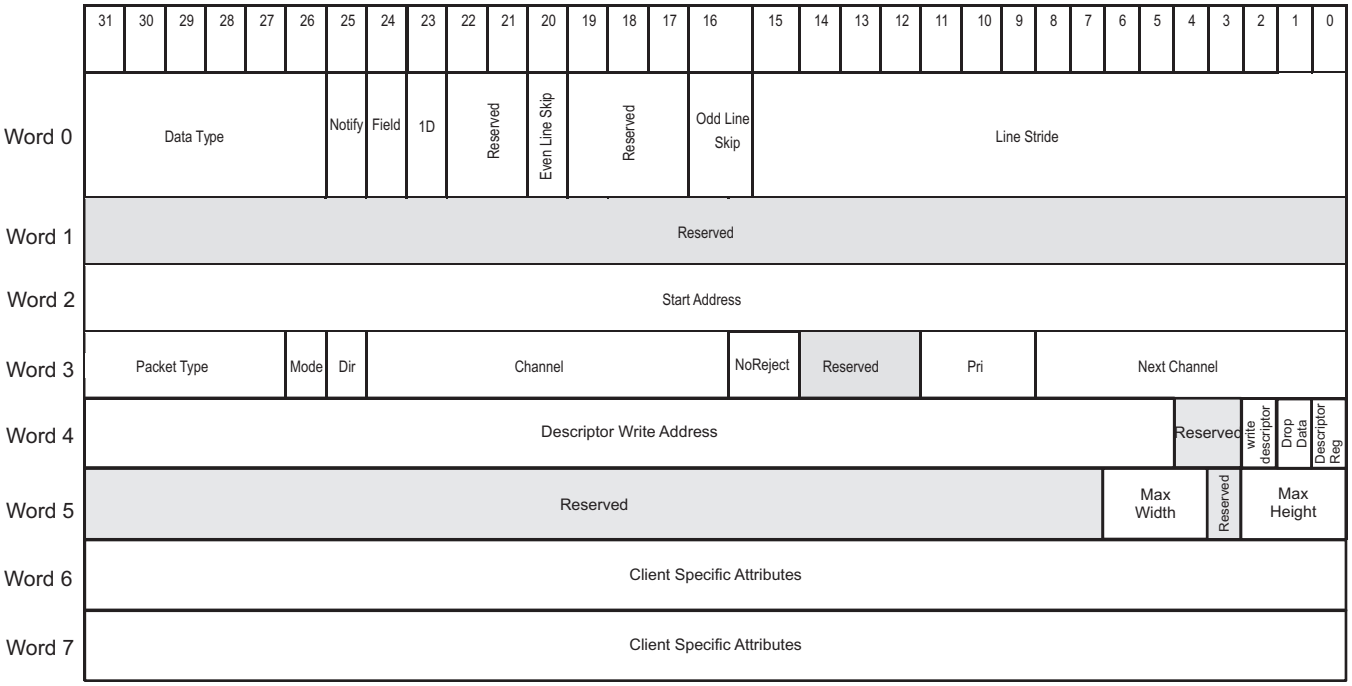


Figure 10-50. Outbound Data Transfer Descriptor Format



The general data transfer Descriptor formats can be seen in the above figures. The descriptor consists of 8 × 32bit words. A Data Transfer Descriptor will be removed from the list when the resource specified by the Channel field is free. If the Channel is not free when the list reaches a data transfer descriptor then the list will stall until the current transfer on the channel has completed.

10.3.9.6.1.1 Data Packet Descriptor Word 0 (Data)

Table 10-21. Data Packet Descriptor Word 0 Field Descriptions

Bit	Field	Value	Description
31:26	Data Type		**Miscellaneous Channel**
			Sets the pixel size in bits plus 1
			RGB Channel
		0	RGB16-565
		1h	ARGB-1555
		2h	ARGB-4444
		3h	RGBA-5551
		4h	RGBA-4444
		5h	ARGB24-6666
		6h	RGB24-888
		7h	ARGB32-8888
		8h	RGBA24-6666
		9h	RGBA32-8888
		10h	BGR16-565
		11h	ABGR-1555
		12h	ABGR-4444
		13h	BGRA-5551
		14h	BGRA-4444
		15h	ABGR24-6666
		16h	BGR24-888
		17h	ABGR32-8888
		18h	BGRA24-6666
		19h	BGRA32-8888
			YUV Channel
		0	Y 4:4:4
		1	Y 4:2:2
		2	Y 4:2:0
		4	C 4:4:4
		5	C 4:2:2
		6	C 4:2:0
		7	CY 4:2:2
		8	YCbC 4:4:4
		14h	Cb 4:4:4
		15h	Cb 4:2:2
		16h	Cb 4:2:0
		17h	CbY 4:2:2
		27h	YC 4:2:2
		37h	YCb 4:2:2
25	Notify	0-1	Send List Notification Interrupt upon last transfer of this channel
24	Field	0-1	Field Value
23	1D	0-1	The transfer is one dimensional. The transfer and frame sizes are combined to make a single 32 bit transfer size. For writes this value is passed to the generated descriptor. This feature is not supported by all clients. Only clients that support the feature will recognize this bit.

Table 10-21. Data Packet Descriptor Word 0 Field Descriptions (continued)

Bit	Field	Value	Description
22:20	Even Line Skip		Field Value
		0	+1 line
		1h	+2 lines
		2h-7h	Reserved
19	Reserved		Reserved for future use
18:16	Odd Line Skip		Field Value
		0	+1 line
		1h	+2 lines
		2h-7h	Reserved
15:0	Line Stride	0-FFFFh	Address stride between lines in bytes

10.3.9.6.1.1.1 Data Type

Bits 31-26 indicate the type of data that is to be transferred. This value is used to compute the number of bytes per pixel so that transactions may be made for the appropriate amount of data. The types of data are dependent on the type of channel. The VPDMA descriptor data types RGB or YUV are defined associated with the VPDMA channel assignment. This helps the engine to distinguish between the overlapping values of the descriptor data types for RGB and YUV data.

- For the Miscellaneous channel, the Data Type selects the size in bits of the data. The range is from 0 to 63 to represent the sizes from 1 to 64 bits.
- For a YUV channel, the Data Type determines, if the data channel is interleaved or color space separate. If color spaced separate, it is still assumed that the two chroma pixels are interleaved.

CAUTION

VPDMA defines the component ordering for its RGB data types in the opposite direction of what commonly used image identifiers expect. To avoid color component swapping in the display and/or in the video/image data written out to the memory, the proper Data Type settings for both RGB and YUV data types must be made. The following paragraphs provide more details on how to set Data Type correctly, in order to match the data stored or expected in the memory.

Setting RGB Data Types

The commonly used RGB format identifiers require the color components to be stored in a little-endian style, where the left most component is the LSB component.

- For an ARGB data type, the A component is the LSB location, as shown in Table 10-22 and Table 10-23;
- For a BGRA data type, the B component would be in the LSB location;

Table 10-22. Common ARGB in Memory (Byte Order)

Addr (LSB)	Addr + 1	Addr + 2	Addr + 3 (MSB)
A	R	G	B

Table 10-23. Common ARGB in 32-bit Memory/CPU Register

Bit 31	Bit 23	Bit 15	Bit 7
B	G	R	A

VPDMA specifies its component ordering in the big-endian style, which requires the data to be stored in the reversed order. Example with ARGB data type is shown in Table 10-24 and Table 10-25. The VPDMA ordering for ARGB data type matches the common BGRA data format.

Table 10-24. VPDMA ARGB in Memory (Byte Order)

Addr (LSB)	Addr + 1	Addr + 2	Addr + 3 (MSB)
B	G	R	A

Table 10-25. VPDMA ARGB in 32-bit Memory/CPU Register

Bit 31	Bit 23	Bit 15	Bit 7
A	R	G	B

In order color components to be mapped correctly and to avoid swapping, the reversal must be taken into consideration when configuring the Data Type in the VPDMA transfer descriptor.

Table 10-26 shows the proper settings required for RGB data types for both storage schemes.

Table 10-26. VPDMA Descriptor RGB Data Type Mapping

Destination Image		VPDMA Data Type Mapping Value	
RGB Component order	Common Image Format Names	**Column A** Data stored in the VPDMA defined order	**Column B** Data stored in the opposite of VPDMA defined order
RGB	RGB16-565	0x0	0x10
	ARGB-1555	0x1	0x13
	ARGB-4444	0x2	0x14
	RGBA-5551	0x3	0x11
	RGBA-4444	0x4	0x12
	ARGB24-6666	0x5	0x18
	RGB24-888	0x6	0x16
	ARGB32-8888	0x7	0x19
	RGBA24-6666	0x8	0x15
	RGBA32-8888	0x9	0x17
BGR	BGR16-565	0x10	0x0
	ABGR-1555	0x11	0x3
	ABGR-4444	0x12	0x4
	BGRA-5551	0x13	0x1
	BGRA-4444	0x14	0x2
	ABGR24-6666	0x15	0x8
	BGR24-888	0x16	0x6
	ABGR32-8888	0x17	0x7
	BGRA24-6666	0x18	0x5
	BGRA32-8888	0x19	0x9

In Table 10-26, if the application uses the same data type definition as the VPDMA (that is, RGB24 refers to the B in the LSB), the data types in Column A should be used. But, if the application expects the common data type component order for RGB data type names, the VPDMA data types in Column B should be used.

For example:

- To display an ARGB32-8888 source image data with A in the LSB, the data type in the descriptor should be set to 0x19. But, to display an ARGB32-888 source image data with B in the LSB, the data type in the descriptor should be set to 0x7.

Setting YUV Data Types

There is no component order reversal for YUV data types. The VPDMA uses generic data type names to specify the memory storage format and the application simply needs to follow the VPDMA defined ordering.

Table 10-27 shows how common YUV data types map to the VPDMA YUV data types in order to clarify the YUV data type configuration.

Table 10-27. VPDMA Descriptor YUV Data Type Mapping

Source YUV Image Types			VPDMA Data Type Mapping (Value)		
Chroma Sub-sample	Common YUV Image Format Type Names	Memory Packed Order [MSB - LSB]	Luma/Chroma Interleaved Channel	Luma-only Channel	Chroma-only Channel
444	YUV	V U Y	YC 4:4:4 (0x8)		
	UVY	Y V U	Cb 4:4:4 (0x14)		
422	NV16 (YUV422SP_UV)	V U		Y 4:2:2 (0x1)	C 4:2:2 (0x5)
	NV16 (YUV422SP_VU)	U V		Y 4:2:2 (0x1)	Cb 4:2:2 (0x15)
	YUV2/YUYV/V422 (YUV422I_YUYV)	V Y U Y	YC 4:2:2 (0x7)		
	YUV422I_YVYU	U Y V Y	CbY 4:2:2 (0x17)		
	Y422/UYVY (YUV422I_UYVY)	Y V Y U	YC 4:2:2 (0x27)		
	YUV422I_VYUY	Y U Y V	YCb 4:2:2 (0x37)		
420	NV12 (YUV420SP_UV)	V U		Y 4:2:0 (0x2) YC 4:2:2 (0x7) (see (1))	C 4:2:0 (0x6) YC 4:2:2 (0x7) (see (1))
	NV21 (YUV420SP_VU)	U V		Y 4:2:0 (0x2) YC 4:2:2 (0x7) (see (1))	Cb 4:2:0 (0x16) YC 4:2:2 (0x7) (see (1))

(1) If 422 source data is used, unused component data fetched (either Luma or Chroma) will be discarded.

For further details on the data formats, refer to Section 10.3.9.8, *VPDMA Data Formats*.

10.3.9.6.1.1.2 Notify

The Notify bit is used in conjunction with the Notify interrupts and when the channel has completed the DMA transfer the Notify Interrupt for the list that contains the descriptor will fire. The last Notify bit set for a specific list will be used so only a single Notify should be set in a list.

10.3.9.6.1.1.3 Field

The Field bit is the field value of the data that will be passed down to the clients. If the data is interlaced, then this value should be cleared to 0.

10.3.9.6.1.1.4 1D

This bit is set if a large one dimensional frame needs to be send to the client. In this case the stride is ignored and for the write the stride the generated descriptor will always be 0. If this bit is set then the transfer length and transfer height and frame width and frame height fields are combined to form one 32 bit field with the upper 8 bits reserved and the lower 24 bits being the size of the frame in pixels.

10.3.9.6.1.1.5 Even Line Skip

The Even Line skip is used with Line Stride to generate the next line address on an even line. All frames start on line 0. This value allows for the DMA controller to skip lines for interlaced data in a progressive frame buffer.

10.3.9.6.1.1.6 Odd Line Skip

The Odd Line skip is used with Line Stride to generate the next line address on an odd line. All frames start on line 0, so this will apply starting with the second line. This value allows for the DMA controller to skip lines for interlaced data in a progressive frame buffer.

10.3.9.6.1.1.7 Line Stride

Bits 15:0 are the stride between lines in bytes at the external address. This value is added or subtracted based upon an adjustment using the current skip value. Operation of the external address pointer shall load the Source Address upon start of the transfer, then at the end of each line increment or decrement by the value computed using the Line Stride and Skip value for the line. The line stride must be aligned to an L3 data bus width. The lower bits of the stride will always be treated as zero to force the alignment.

10.3.9.6.1.2 Data Packet Descriptor Word 1

Table 10-28. Data Packet Descriptor Word 1 Field Description

Bits	Name	Description
31:16	Line Length	Line Length in Pixels
15:0	Transfer Height	Number of rows in transfer.

10.3.9.6.1.2.1 Line Length

Bits 31-16 are the line length in pixels of the current channel. This is ignored for the outbound transfer as the client will provide the line length with the end of line signal on the client interface. The maximum supported line length is currently 4096.

10.3.9.6.1.2.2 Transfer Height

Bits 15-0 are the number of lines to be transferred in the current channel. This is ignored for outbound transfers as the client will provide the line length with the end of frame signal on the client interface. The maximum supported transfer size is currently 2048.

10.3.9.6.1.3 Data Packet Descriptor Word 2

Table 10-29. Data Packet Descriptor Word 2 Field Descriptions

Bit	Field	Value	Description
31:0	Start Address		32-bit data source address [31:0]
			If Mode is TILED, then TILER specific ADDRESS Map is used:
			Bits 31-29:
		0	0-degree view
		1h	180-degree view + mirroring
		2h	0-degree view + mirroring
		3h	180-degree view
		4h	270-degree view + mirroring
		5h	270-degree view
		6h	90-degree view
		7h	90-degree view + mirroring
			Bits 28-27:
		0	8-bit container
		1h	16-bit container
		2h	32-bit container
		3h	Page Mode
			If Mode is NORMAL, then bits 31-26 are the upper bits of the address.

10.3.9.6.1.3.1 Start Address

This is the byte aligned address for the first data transfer. The address on the OCP bus will always be word aligned.

10.3.9.6.1.4 Data Packet Descriptor Word 3

Table 10-30. Data Packet Descriptor Word 3 Field Descriptions

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xa
26	Mode	0= Normal, 1=TILED
25	Direction	Inbound = 0, Outbound = 1
24:16	Channel	Channel for which this descriptor describes
15	Reserved	Reserved for future use
11:9	Priority	Only Bit 9 and Bit 11 are used to set the priority. Bit 10 is ignored. Highest = 0, Lowest = 3 By default, hardware assigns priority = 3. This priority level is used in the arbitration between the masters (for DDR access). See Section 10.3.9.6.1.4.5, *Priority*, for more details.
8:0	Next Channel	Next Channel to execute on a line or the channel to use in the generated write descriptor.

10.3.9.6.1.4.1 Packet Type

Bits 31:27 are a unique code which indicates that this Descriptor is a VPDMA descriptor.

10.3.9.6.1.4.2 Mode

Bit 26 is used to indicate if the transfer is to regular memory space or to Tiled memory space. The VPDMA will use this to determine if it does standard raster based addressing or if it assumes that the data is stored in TILER format. If data is stored in TILER format then the buffer is turned into 2 or 4 line buffers depending on the container type. For shared clients that use the memory, such as the ancillary data and the VIP port, only one can be active, if the mode field is set. Only clients that support Tiling will properly pack the data for tiling on the output interface. This must only be set for channels going to clients that support the TILING feature in the client configuration.

10.3.9.6.1.4.3 Direction

Bit 25 is used to indicate the direction of transfer. This bit indicates that the data flow is from an external source to an internal buffer (inbound) or data transfers form an internal buffer to an external location (outbound).

10.3.9.6.1.4.4 Channel

Bits 24:16 are the Channels which is supported by the descriptor. This is the identification of the specific Channel which is controlled by the contents of the descriptor. The channel assignments can be found in the channel table.

10.3.9.6.1.4.5 Priority

Bits 11:9 are set to indicate priority of the transfer, these are directly mapped to the OCP reqinfo bits.

10.3.9.6.1.4.6 Next Channel

Bits 8:0 give the next channel to use to create a composite frame. The next channel must be to a free channel. The last channel of a row should point back to the initial channel which must be a channel tied directly to a client. The Descriptor for the Next Channel must be of the same type as the current descriptor.

10.3.9.6.1.5 Data Packet Descriptor Word 4

10.3.9.6.1.5.1 Inbound data

Table 10-31. Data Packet Descriptor Word 4 Inbound Data Field Descriptions

Bits	Name	Description
31:16	Frame Width	Width of the client frame.
15:0	Frame Height	Height of the client frame

10.3.9.6.1.5.1.1 Frame Width

Bits 31:16 indicate the width in pixels of the frame. This is the width of the entire frame and not just the width of the data represented by the current channel unless this channel fills the entire frame. This allows for the VPDMA to present a larger frame of data to the client while only fetching the required data. The currently supported maximum width is 4096.

10.3.9.6.1.5.1.2 Frame Height

Bits 15:0 indicate the height in pixels of the entire frame. This is the height of the entire frame and not just the height of the data represented by the current channel unless this channel fills the entire frame. This allows for the VPDMA to present a larger frame of data to the client while only fetching the required data. The currently supported maximum height is 2048.

10.3.9.6.1.5.2 Outbound data

Table 10-32. Data Packet Descriptor Word 4 Outbound Data Field Descriptions

Bits	Name	Description
31:5	Descriptor Write Address	The 32 byte aligned location to write an inbound descriptor
2	Write Descriptor	If set to 1, a descriptor will be generated when the client completes a frame.
1	Drop Data	If set to 1, the data will not be written out. Also, if this is set, the write descriptor bit must be set. This allows for descriptors to only be written out so that software can determine the size of the required buffer before data is written out.
0	Use Descriptor Register	If set to 1, the CURRENT_DESCRIPTOR register will be used for the write address location. If set to 0, the Descriptor Write Address field in this word will be used for the Descriptor Write Address.

10.3.9.6.1.5.2.1 Descriptor Write Address

Bits 31:5 set the 32 byte address to write the generated descriptor. This address is only used if the Write Descriptor bit is set to 1 and the Use Descriptor Register bit is set to 0. If this case is met when the channel is complete a descriptor that meets the inbound descriptor format will be written to the location specified by this field. This allows for software to have a specific channel write its descriptor to a specific address no matter what order the channel completes compared to other outbound channels.

10.3.9.6.1.5.2.2 Write Descriptor

Bit 2 determines if a descriptor should be written out when the client is completed. If this bit is set the descriptor will be written. The format of the descriptor will be of an Inbound Data Transfer descriptor with the LINE LENGTH and TRANSFER HEIGHT fields determined by the counters in the clients. This means that the direction bit will be the opposite of the inbound descriptor. The FRAME WIDTH and FRAME HEIGHT values will match the LINE LENGTH and TRANSFER HEIGHT fields. The CHANNEL and NEXT CHANNEL fields will match the NEXT CHANNEL field of the original descriptor. The FIELD bit will match the source field captured by the client. The NOTIFY bit will always be 0. All other fields will match the original outbound descriptor. If the Outbound descriptor MODE is set to TILED data then the descriptor address used will be in TILED data space and software must ensure that the address is in page mode for the address of the descriptor.

10.3.9.6.1.5.2.3 Drop Data

Bit 1 determines if the data should be written out or not. In some cases software might only want to write the descriptor out to determine the size of the buffer that will be required to store incoming data. By setting this bit,no data will be written out. If this bit is set then the Write Descriptor bit MUST be set. Bit 0 determines where the descriptor should be written. If it is desired that all the descriptors are written in order to a set queue then this bit should be set and the CURRENT_DESCRIPTOR, DESCRIPTOR_TOP and DESCRIPTOR_BOTTOM registers are used to define the location of the descriptor queue that will be written.

10.3.9.6.1.5.2.4 Use Descriptor Register

Bit 0 determines where the descriptor should be written. If it is desired that all the descriptors are written in order to a set queue then this bit should be set and the CURRENT_DESCRIPTOR, DESCRIPTOR_TOP and DESCRIPTOR_BOTTOM registers are used to define the location of the descriptor queue that will be written.

10.3.9.6.1.6 Data Packet Descriptor Word 5

10.3.9.6.1.6.1 Outbound data

Table 10-33. Data Packet Descriptor Word 5 Outbound Data Field Descriptions

Bits	Name	Description
6:4	Max Width	The maximum allowable pixels per line. 0: Unlimited Line Size 1: Use VPE_MAX_SIZE1 Max Width field 2: Use VPE_MAX_SIZE2 Max Width field 3: Use VPE_MAX_SIZE3 Max Width field 4: 352 pixels 5: 768 pixels 6: 1280 pixels 7: 1920 pixels Others: Reserved
2:0	Max Height	The maximum allowable lines per frame. 0: Unlimited Frame Size 1: Use VPE_MAX_SIZE1 Max Height field 2: Use VPE_MAX_SIZE2 Max Height field 3: Use VPE_MAX_SIZE3 Max Height field 4: 288 lines 5: 576 lines 6: 720 lines 7: 1080 lines Others: Reserved

NOTE: Width and Height are set in the following register bit-fields:

- For VPE_MAX_SIZE1: VPE_MAX_SIZE1[31:16] MAX_WIDTH and VPE_MAX_SIZE1[15:0] MAX_HEIGHT
- For VPE_MAX_SIZE2: VPE_MAX_SIZE2[31:16] MAX_WIDTH and VPE_MAX_SIZE2[15:0] MAX_HEIGHT registers
- For VPE_MAX_SIZE3: VPE_MAX_SIZE3[31:16] MAX_WIDTH and VPE_MAX_SIZE3[15:0] MAX_HEIGHT

10.3.9.6.1.6.1.1 Max Width

Bits 6:4 are encoded to set the maximum transferred line size. If the field is not set to unlimited if the client sending data into the VPDMA exceeds the maximum allowed pixels the data will continue to be received from the client but will not be sent to external memory until an end of line is received from the client sending data. The outbound descriptor if created will still have the transmitted size of the last line of the frame which will match the max width. If the frame does not exceed the max width then the actual transmitted size will be placed in the descriptor. Tiled clients such as the noise filter should always set this to 0. If Max Width is 0 and the line size is larger then 4096 pixels then the address counters will overflow and the data at the start of the line will be overwritten.

10.3.9.6.1.6.1.2 Max Height

Bits 2:0 are encoded to set the maximum transferred frame size. If the field is not set to unlimited if the client sending data into the VPDMA exceeds the maximum allowed lines the data will continue to be received from the client but will not be sent to external memory until an end of frame is received from the client sending data. The outbound descriptor if created will still have the transmitted number of lines received which would match the max height. If the frame does not exceed the max height then the actual transmitted size will be placed in the descriptor. Tiled clients such as the noise filter should always set this to 0.

10.3.9.6.1.7 Data Packet Descriptor Word 6/7 (Data)

The words 4/5 give a 64 bit of configuration that can be passed specifically to the module that supports it. This is passed directly down to the module through a VPI Control port. Please see the section on the specific clients for the format of this data.

10.3.9.6.2 Configuration Descriptor

The Configuration Descriptor is used in a List to setup a client configuration. The configuration descriptor consists of a header and a payload portion. The payload can either be part of the list that the descriptor is contained in or it can be at a separate location. The VPDMA will pass the payload of the configuration descriptor down to the client over the VPI Control port interface or through the MMR configuration port depending on the destination field. A Configuration Descriptor to any single destination except Destination 0 must be on a single list. Configuration Descriptors to different destinations may be on different lists..

The Configuration Descriptor Header is 4 × 32 bit words. A configuration descriptor is not consumed until the destination specified has received the entire configuration payload. Therefore the list is expected to stall while the configuration takes place. This ensures that all descriptors after a configuration descriptor in the list will occur after the desired configuration.

10.3.9.6.2.1 Configuration Descriptor Header Word0

Table 10-34. Configuration Descriptor Header Word0 Field Descriptions

Bits	Name	Description
31:0	Address Offset of the Destination	This is the address offset location in the destination that the block of data in the payload should be written. This field is only used if the descriptor Class is a block type. Otherwise this field is reserved.

10.3.9.6.2.2 Configuration Descriptor Header Word1

Table 10-35. Configuration Descriptor Header Word1 Field Descriptions

Bits	Name	Description
15:0	Number of Data Words	Length of First Data Packet for Class 1(block).

10.3.9.6.2.2.1 Number of Data Words

Bits 15:0 indicate the length of the first data block if the class is 1 as specified in Configuration Descriptor Header Word3. If the class is not 1 then this field should be set to 0.

10.3.9.6.2.3 Configuration Descriptor Header Word2

Table 10-36. Configuration Descriptor Header Word2 Field Descriptions

Bits	Name	Description
31:0	Payload Location	Pointer to the data payload

10.3.9.6.2.3.1 Payload Location

Bits 31:0 contain the pointer to the data payload if the command packet is an indirect type. This value along with the Payload Length will be combined to issue a DMA transaction that must complete before the List Manager can finish processing this descriptor. The list will be made inactive pending this DMA completion. This address should be on a 16 byte boundary so the lower 4 bits should always be 0.

10.3.9.6.2.4 Configuration Descriptor Header Word3

Table 10-37. Configuration Descriptor Header Word3 Field Descriptions

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xb
26	Direct	Direct Command = 1 Indirect Command = 0
25:24	Class	0 = Address, Data Set 1 = Block 2,3 Reserved for Future Use
23:16	Destination	Destination of the configuration payload
15:0	Payload Length	Length of Payload in Words.

10.3.9.6.2.4.1 Packet Type

Bits 31:27, this field indicates the type of descriptor and should be set 0xB for configuration descriptor.

10.3.9.6.2.4.2 Direct

Bit 26, this field indicates that the descriptor is a direct command or indirect command. A direct command means that the payload is contiguous with the descriptor and will be pulled in by the list manager descriptor control as it processes the list. A direct payload must end on or before a 256 byte boundary in the list. An indirect command is when the Address is located in Word2 is a pointer to the data payload. A DMA transaction will be scheduled to bring the payload into the List Manager to allow for the payload to be sent to the destination.

10.3.9.6.2.4.3 Class

Bits 25:24, this field indicates the type of payload is associated with command descriptor, and thus how to handle the payload.

A 0 indicates that the payload consists of blocks of data with each block having an address and length. The first word after a sub-block contains the next address and length in bytes. This allows for writing to multiple configuration regions in a client. The Configuration Descriptor word1 is the first address and length pair. All addresses used must be larger then the previous address for all destinations except for the MMR destination. If a sub block length is not evenly divisible into 16 then zeroes should be padded in the payload so that the Next Client Address and Length field start on a 16 byte boundary.

10.3.9.6.2.4.3.1 Address Data Block Format

Table 10-38. Address Data Block Format Field Descriptions

Bits	Name	Description
31-0		Next Client Address
31-0		Configuration for Next Client Address
31-0		Configuration for Next Client Address + 4
31-0		Configuration for Next Client Address + 8
31-0		Configuration for Next Client Address + 12
31-0		Configuration for Next Client Address + 16
31-0		Next Client Address 2
15-0		Sub Block Length

A 1 indicates the payload is simply block data. The data is contiguous and starts at the offset of the destination as specified in word1.

10.3.9.6.2.4.4 Destination

The Destination field is used to determine where the configuration payload should be sent. The values for the destination field can be seen in the table below.

Table 10-39. Destination Field Description

Destination Value	Actual Destination	Description
0	mmr_client	Write to MMR registers to setup other modules
4	VPE Scaler	VPE Scaler Coefficient Tables

10.3.9.6.2.4.5 Descriptor Length

Bits 15:0, this field indicates the size of the payload in words for the command. This is 128 bit words. The maximum number of words is 1023 words in a single payload

10.3.9.6.3 Control Descriptor

10.3.9.6.3.1 Generic Control Descriptor Format

A control descriptor is the mechanism that is used in a list to give commands to the List Manager. These descriptors are not considered consumed until the List Manager has done the instruction required by the descriptor. All control descriptors have a common Header located at Word 3 but the remaining words are based on the specific control descriptor.

10.3.9.6.3.2 Control Descriptor Header Description

Table 10-40. Control Descriptor Header Description

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xc
26:25	Reserved	Reserved
24:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	The type of control descriptor that should be run by the List Manager

10.3.9.6.3.2.1 Packet Type

This field indicates a VPDMA control descriptor.

10.3.9.6.3.2.2 Source

The source is combined with the control field to determine what the control descriptor should use as the source for its synchronization event.

10.3.9.6.3.2.3 Control

The Control field defines the specific function of the descriptor. Table 10-41 lists the different control descriptors.

10.3.9.6.3.3 Control Descriptor Types

Table 10-41. Control Descriptor Types Summary

Control Descriptor	Control Field Value	Description
Sync on Client	0	Wait for the client attached to the channel specified to reach a certain event before proceeding.
Sync on List	1h	Wait for another list(s) to reach its Sync on List Descriptor
Sync on External Event	2h	Wait for a Register write to the VPE_LIST_STAT_SYNC bit specified or for an external event.
Sync on Channel	4h	Wait for the channel specified to complete
Change Client Interrupt	5h	Change the interrupt event for a client interrupt but do not wait for the event to occur.
Send Interrupt	6h	Generate an interrupt event on one of the Control Descriptor Interrupts
Reload List	7h	Reload the list from the location and size specified.
Abort Channel	8h	Abort the transfer in the channel specified.

10.3.9.6.3.3.1 Sync on Client

A Sync on Client Control descriptor uses the source field to select a channel to modify the attached clients interrupt generation event. For a client that supports multiple channels then only an event on the portion of the client that supports that client will cause the interrupt to be generation. After configuring the interrupt generation event the list will then stall until that event has occurred.

Table 10-42. Sync on Client Field Descriptions (Word - 1)

Bits	Name	Description
31:16	PIXEL_COUNT	Specify the pixel position on a line specified by LINE_COUNT where a pixel based event would occur.
15:0	LINE_COUNT	Specify the line where a line based event would trigger

Table 10-43. Sync on Client Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 0

10.3.9.6.3.3.2 Sync on List

The Sync on List Control descriptor is a mechanism to ensure that multiple lists have all reached a common point. The Sync on List descriptor uses the Source field as a mask for each list that should be synchronized which must include the list where the control descriptor resides. Once all lists specified in the source field have reached their Sync on List descriptor all lists will resume with the lowest list number resuming first. All Sync on List Control descriptors in each of the lists must have the same source field so that all lists will synchronize upon reaching that point.

If it is desired to synchronize list 0 and list 1 then the source field would be 0x3. If it is desired to synchronize list 1, list 3, and list 4 then the source field would be 0x1a. Word 0, Word 1 and Word 2 are reserved.

Table 10-44. Sync on List Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 1h

10.3.9.6.3.3.3 Sync on External Event

A Sync on External Event descriptor ensures an external event has occurred before proceeding. The external event can be a write to a bit of the VPE_LIST_STAT_SYNC register or other external events that are determined at VPDMA elaboration to have occurred. This descriptor can be used to allow for external software to control progression of the list. The descriptor can also be used to select external signals that are brought into the VPDMA. The current implementation just synchronize on the VPE_LIST_STAT_SYNC bit for the list number that called the descriptor.

Table 10-45. Sync on External Event Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:4	Reserved	Reserved
3:0	Control	Control type = 2h

10.3.9.6.3.3.4 Sync on Channel

A Sync on Channel descriptor stalls the list until the channel specified is free. If the channel is already free then the descriptor will not cause the list to stall. Word 0, Word 1 and Word 2 are reserved.

Table 10-46. Sync on Channel Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 4h

10.3.9.6.3.3.5 Sync on LM Timer

A Sync on LM Timer descriptor sets a value from the current timer position to wait. The LM timer is a free running counter at the LM processing clock. The Timer Value in the descriptor is added to the value of the timer at the time the descriptor is received and the list will stall for this many cycles before it becomes active again.

10.3.9.6.3.3.6 Change Client Interrupt

A Change Channel Interrupt Source descriptor uses the source field to select a channel to modify the attached clients interrupt generation event. The list will not stall on this descriptor. The format of the fields is identical to the Sync on Client Descriptor. Word 0 is reserved.

Table 10-47. Change Client Interrupt Field Descriptions (Word - 1)

Bits	Name	Description
31:16	PIXEL_COUNT	Specify the pixel position on a line specified by LINE_COUNT where a pixel based event would occur.
15:0	LINE_COUNT	Specify the line where a line based event would trigger.

Table 10-48. Change Client Interrupt Field Descriptions (Word - 2)

Bits	Name	Description
31:4	Reserved	Reserved
3:0	Event	Specify the event which should trigger the client interrupt.

Table 10-49. Change Client Interrupt Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 5h

10.3.9.6.3.3.7 Send Interrupt

A Send Interrupt descriptor will cause the VPDMA to generate an interrupt on the list manager controlled interrupts as specified by the Source Field. The list will not stall on this descriptor. For example, if source is 0 then control_descriptor_int0 will fire. If source is 12, then control_descriptor_int12 will fire. For more information ot VPDMA interrupt events, see Section 10.3.9.5, *VPDMA Interrupts*.

Table 10-50. Send Interrupt Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 6h

10.3.9.6.3.3.8 Reload List

A Reload List descriptor causes ending descriptors after this descriptor in the original list to be dropped and a new list at the location and of the size specified in the descriptor. This descriptor can be used to allow for linked lists in the VPDMA as the list will continue from this point and fetch the list specified and it does not have to be continuous with the current list.

Table 10-51. Reload List Field Descriptions (Word - 0)

Bits	Name	Description
31:0	LIST_ADDRESS	31 most-significant Bits of the memory address where the descriptors to be loaded are stored. Address must be 16 byte aligned.

Table 10-52. Reload List Field Descriptions (Word - 1)

Bits	Name	Description
31:16	Reserved	Reserved
15:0	LIST_SIZE	Size of the list to load

Table 10-53. Reload List Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:4	Reserved	Reserved
3:0	Control	Control type = 7h

10.3.9.6.3.3.9 Abort Channel

An Abort Channel descriptor is used to clear a channel. This clears the channel from issuing any more requests. Any outstanding requests for that channel will complete as originally scheduled. All data inside the client will be flushed. For Tiled Clients such as the noise filter it is required to issue two consecutive abort channel descriptors to ensure the channel is aborted for both the current tile and next tile. Word 0, Word 1 and Word 2 are reserved.

Table 10-54. Abort Channel Field Descriptions (Word - 3)

Bit	Name	Description
31:27	Packet Type	Host Packet Descriptor type = 0xC
26:24	Reserved	Reserved
23:16	Source	VPDMA Channel Number whose transfers are to be aborted
15:4	Reserved	Reserved
3:0	Control	Control type = 9h

10.3.9.7 VPDMA Configuration

The following section describes the different ways of configuring VPDMA for data transfers.

10.3.9.7.1 Regular List

A regular list executes each descriptor in order until the end of the list is reached. When the end of the list is reached an interrupt is sent and the list can be reused by software. A regular list can contain any descriptor types. Software creates the list at some location in external memory. After completing writing the list software then writes the location of the list to the VPE_LIST_ADDR[31:0] VPE_LIST_ADDR register and then writes the LIST_ATTR register. If the NUMBER in the VPE_LIST_ATTR[26:24] LIST_NUM is not an active list then the List will be loaded and begin to execute the next time the List Manager gets to IDLE after processing previous loaded lists. If the NUMBER in the VPE_LIST_ATTR[26:24] LIST_NUM is busy then the VPE_LIST_ADDR[31:0] LIST_ADDR and VPE_LIST_ATTR registers will be locked until the active list specified by NUMBER completes.

The different ports inside VPDMA requires different list setup, as explained in the following sections.

10.3.9.7.2 Video Input Ports

The Video Input Ports can be used in multiple ways. Depending on how the input ports are configured will determine how the lists to manage them need to be setup. The ways in which the ports can work are multiplexed data stream, single YUV color separate stream, dual YUV interleaved or single RGB stream. Each port also has an ancillary data port that can run either multiplexed or single data stream and shares the buffering with the main data stream. For all cases the descriptor must be loaded into the client before the vertical sync is received on the VIP port or the entire frame will be dropped. As with all write clients the VIP channels can be shadowed so the next frame/field descriptor can be loaded while the previous frame is running without stalling the list.

10.3.9.7.2.1 Single YUV Color Separate

If the port is configured to be used as sending out color separated YUV data then the channels that must be used are VIP1_PORTA_LUMA and VIP1_PORTA_CHROMA for the luma and chroma components respectively. The data type can be either 420 or 422 color separate in this case.

10.3.9.7.2.2 Dual YUV Interleaved

If the port is configured to be used to send out 422 interleaved data from a non-multiplexed source then the channels that must be used are VIP1_PORTA_LUMA or VIP1_PORTA_CHROMA depending on if the data stream is connected to the luma or chroma port. The data type must be 422 interleaved in this case.

10.3.9.7.2.3 Single RGB Stream

If the port is configured to be used to send out RGB stream then the channel VIP1_PORTA_RGB must be used. The incoming data is then assumed to be RGB 888 data and this is combined with the Background Color Alpha register field to make ARGB8888 data that will then be sent out based on the data type. If the data type uses less then the full 8 bits the lower bits are dropped and just the upper bits are sent to get the correct data format.

10.3.9.8 VPDMA Data Formats

Each of the channels has a type of data that it can support based upon the client that it services. Also for each channel, the data type will assume that data is packed in memory a certain way and will be prevented to the client in the same manner to the client no matter what the format of the data in memory.

10.3.9.8.1 YUV Data Formats

The YUV formatted channels expect video data that is in YUV color space. The YUV data can be type is for YUV data and it can support both interleaved data where Luma and Chroma are in the same data buffer or it can support co-planar data where the Luma and Chroma are in separate data buffers. The YUV channel also can be given a position to give a different start position for the client instead of the default upper left hand corner of the frame. The storage format for YUV data depends on the data type field. The data type must be set to a data type that the channel can support. All the clients that data are provided too will either accept Luma Only, Chroma Only or Luma and Chroma in a parallel data bus.

10.3.9.8.1.1 Y 4:4:4 (Data Type 0)

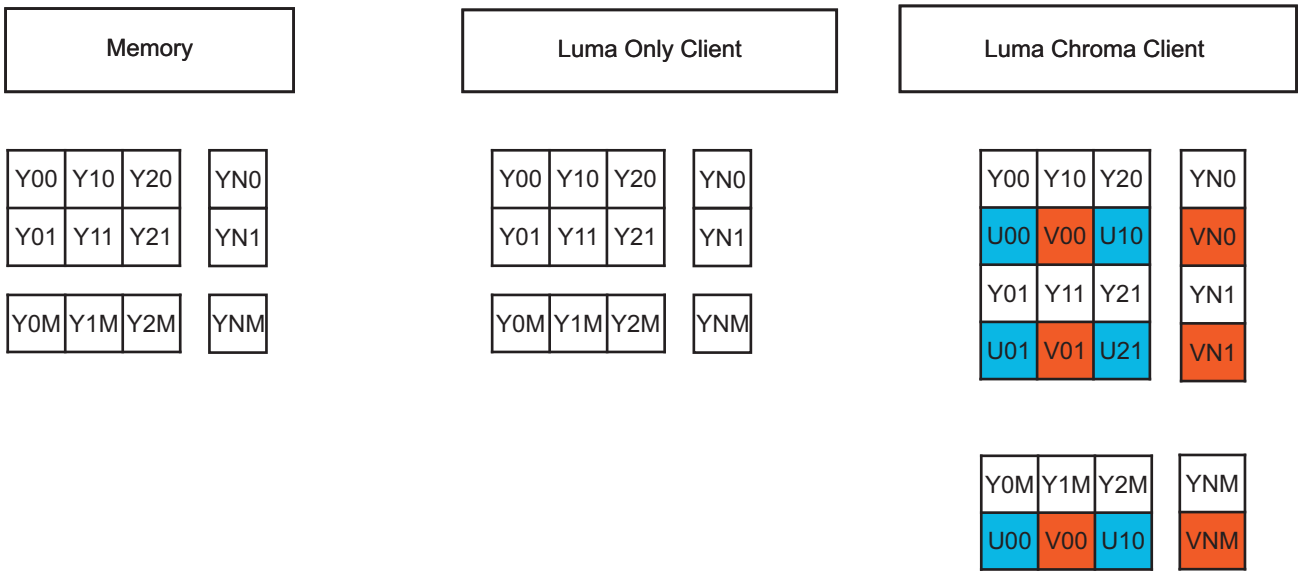
The Y 4:4:4 data type is used for a co-planar 4:4:4 data. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and height set to the desired frame size expected by the receiving client.

Figure 10-51. Y 4:4:4 (Data Type 0)

10.3.9.8.1.2 Y 4:2:2 (Data Type 1)

The Y 4:2:2 data type is used for a co-planar 4:2:2 data. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and the height of the desired frame sent by the VPDMA to the receive client.

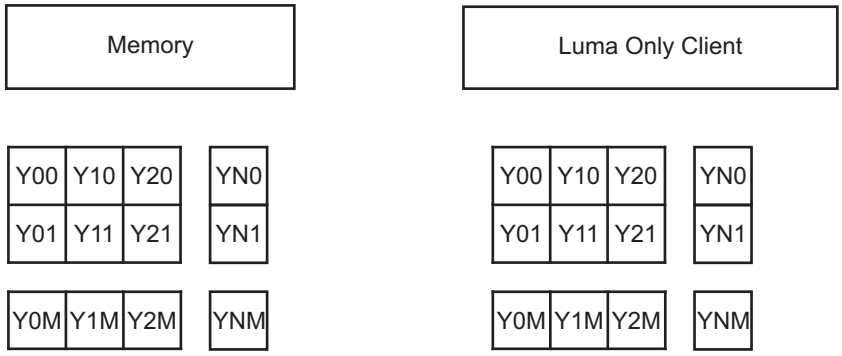
Figure 10-52. Y 4:2:2 (Data Type 1)



10.3.9.8.1.3 Y 4:2:0 (Data Type 2)

The Y 4:2:0 data type is used for a co-planar 4:2:0 data type. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and the height of the expected frame for the client.

Figure 10-53. Y 4:2:0 (Data Type 2)



10.3.9.8.1.4 C 4:4:4 (Data Type 4)

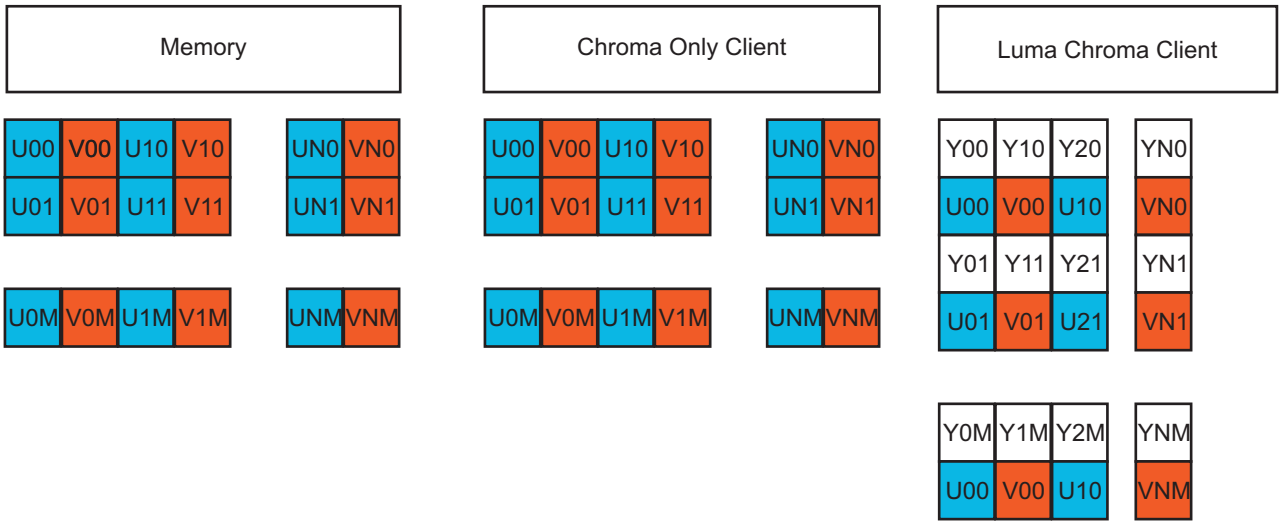
The C 4:4:4 data type is used for a co-planar 4:4:4 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container. This data block should have the width and height of the expected client frame.

Figure 10-54. C 4:4:4 (Data Type 4)

10.3.9.8.1.5 C 4:2:2 (Data Type 5)

The C 4:2:2 data type is used for co-planar 4:2:2 data.It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container. This data block should have the width and the height of the expected client frame.

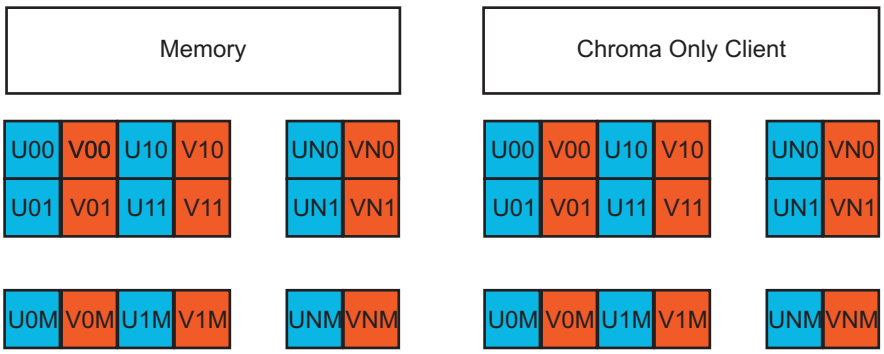
Figure 10-55. C 4:2:2 (Data Type 5)



10.3.9.8.1.6 C 4:2:0 (Data Type 6)

The C 4:2:0 data type is used for a co-planar 4:2:0 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in a 16 bit container. This data block should have the width and half the height of the expected clients frame.

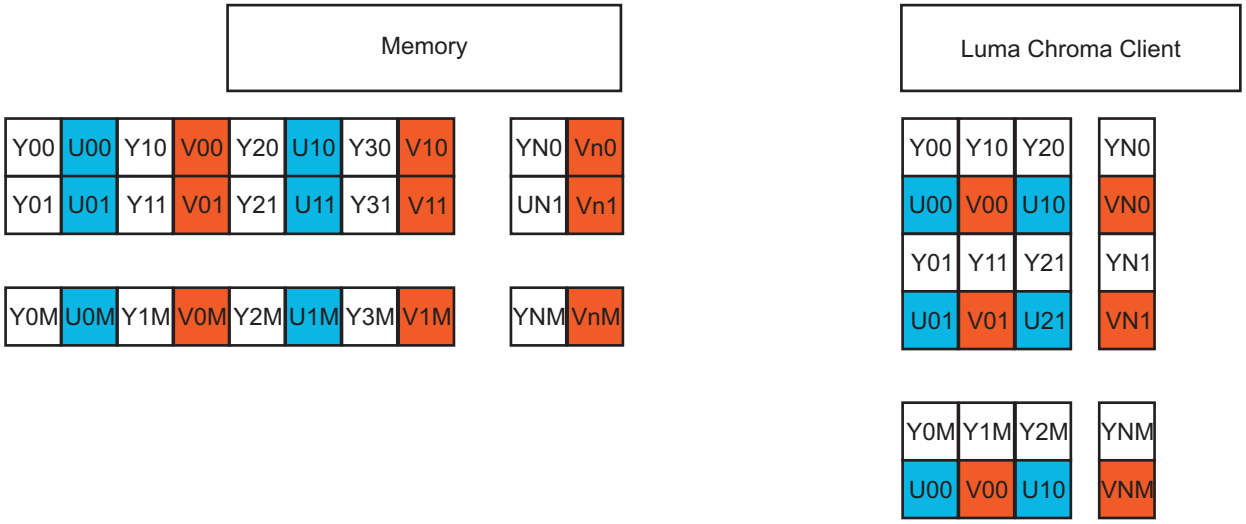
Figure 10-56. C 4:2:0 (Data Type 6)



10.3.9.8.1.7 YC 4:2:2 (Data Type 7)

The YC 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Y0 in the lowest byte followed by Cb followed by Y1 finally Cr in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container. The data block width and height should be set the same as the expected client.

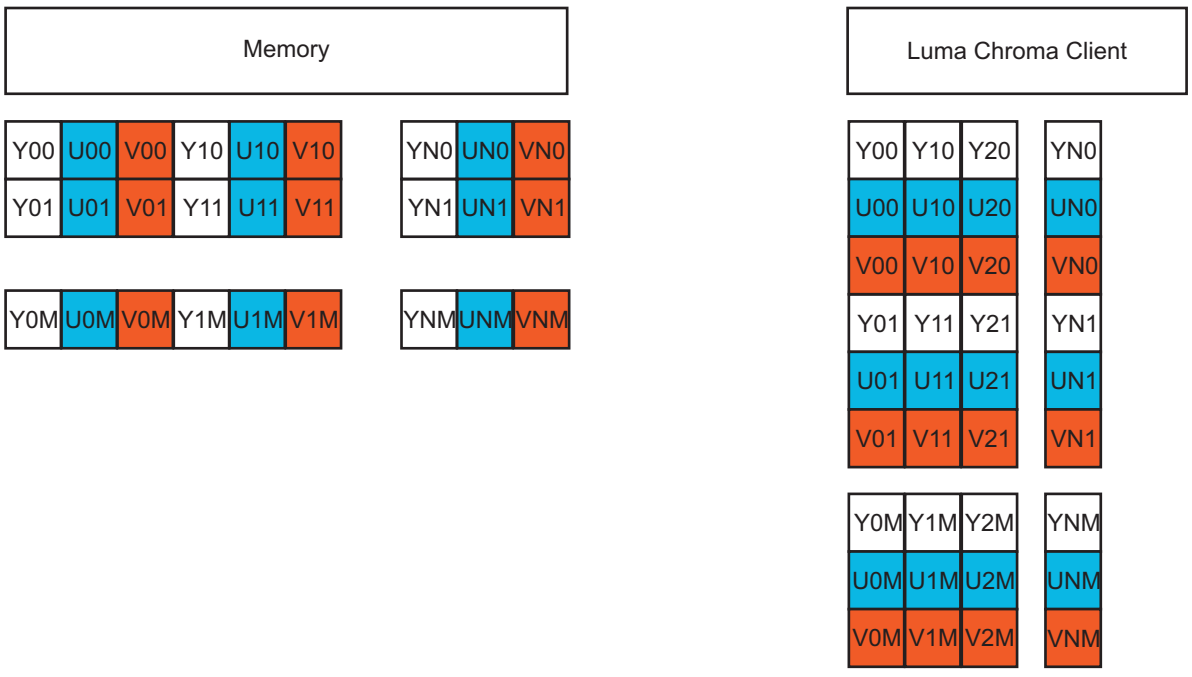
Figure 10-57. YC 4:2:2 (Data Type 7)



10.3.9.8.1.8 YC 4:4:4 (Data Type 8)

The YC 4:4:4 data type is used for interleaved 4:4:4 data. It is expected to be packed with Y0 in the lowest byte followed by Cb followed by Cr. The transfer counts each YCbCr triplet in a 24 bit word as a pixel. A 2D transfer of the data is NOT supported in the VPDMA. The data block width and height should be set to the number of pixels and lines that are expected by the client.

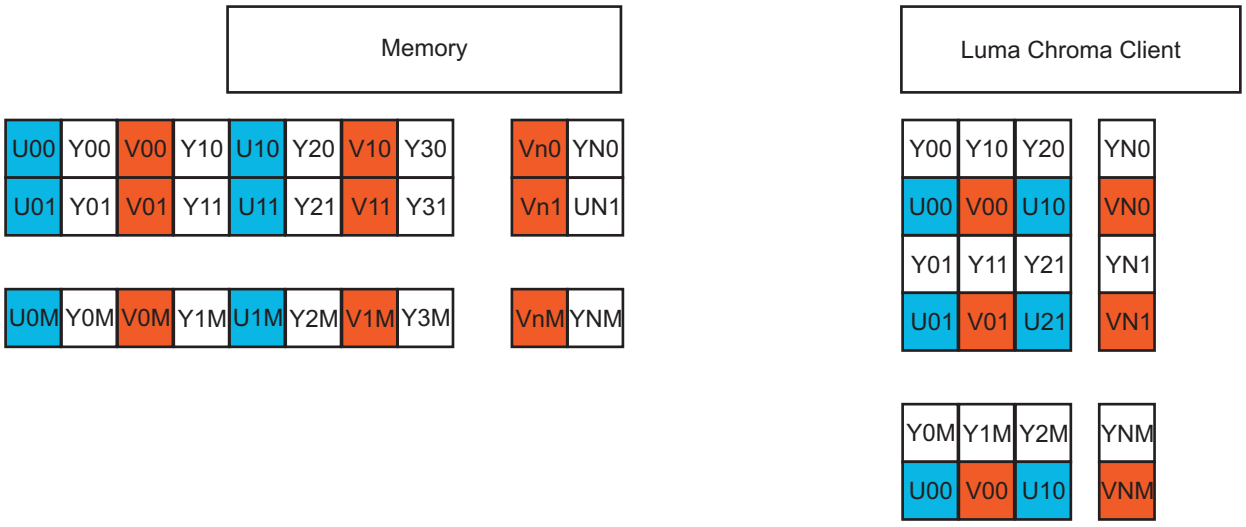
Figure 10-58. YC 4:4:4 (Data Type 8)



10.3.9.8.1.9 CY 4:2:2 (Data Type 23)

The CY 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Cb in the lowest byte followed by Y0 followed by Cr finally Y1 in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container. The data block width and height should be set the same as the expected client.

Figure 10-59. CY 4:2:2 (Data Type 23h)



10.3.9.8.2 RGB Data Formats

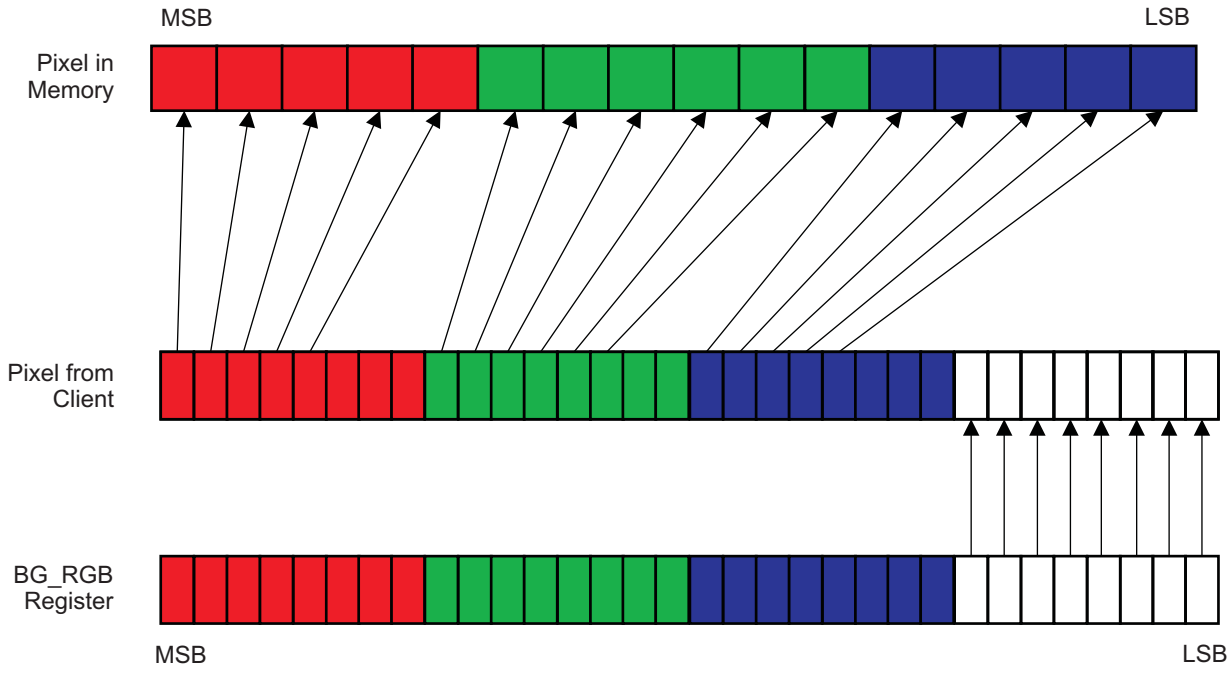
The RGB channel type is used to provide data for a client that expects to transmit RGB data. In all modes the client is always RGBA 8888 data. The lower bits, if not provided by the data stream, are a replication of the lower bit of the data. For outbound data the input is assumed to be RGB 888 and the Alpha value is taken from the Background Color register to make a full ARGB 8888. The lower bits are dropped from this data, if a data type specifies less then the full 8 bits per color. The client has individual data buses for each component so they have no order dependency in the data bus.

10.3.9.8.2.1 Input Data Formats

10.3.9.8.2.1.1 RGB16-565 (Data Type 0)

In RGB16-565 mode, each pixel is a single RGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the middle 6 bits for green data and the upper 5 bits for red data.

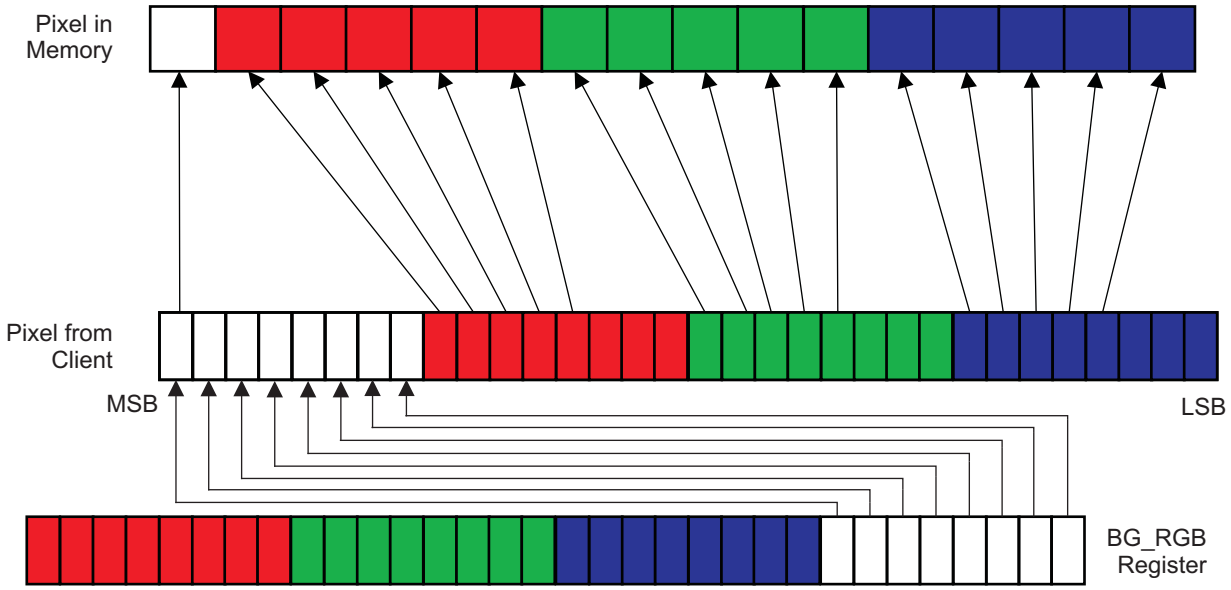
Figure 10-60. RGB16-565 (Data Type 0)



10.3.9.8.2.1.2 ARGB-1555 (Data Type 1)

In ARGB-1555 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data and the upper most bit for the blend value to use 0 or 0xFF.

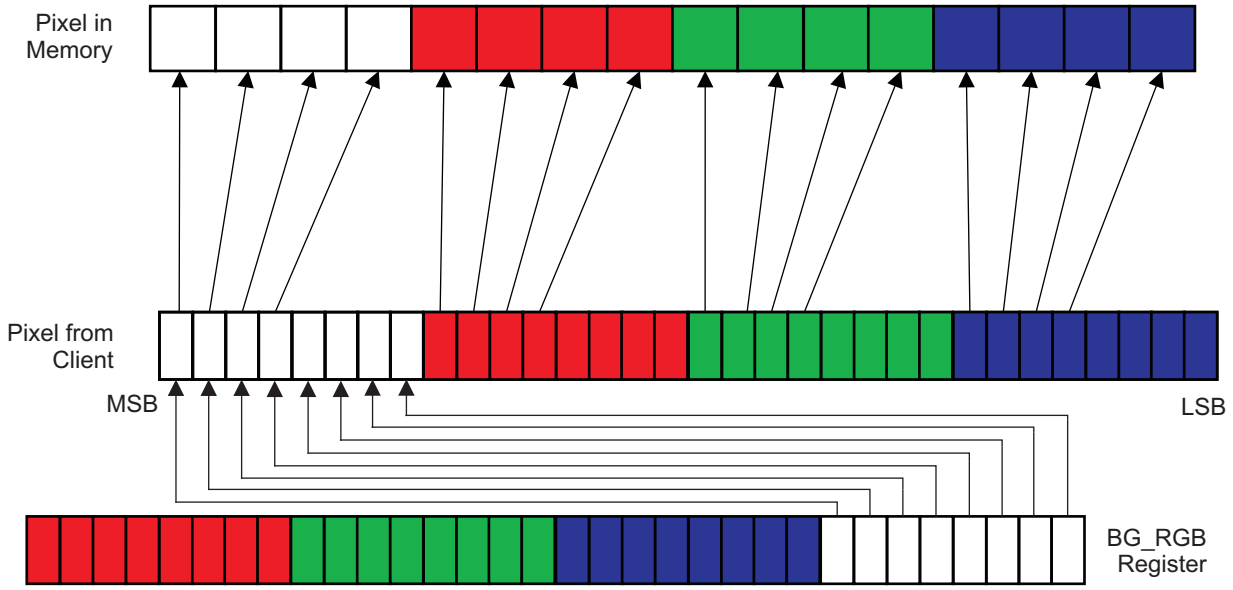
Figure 10-61. ARGB-1555 (Data Type 1)



10.3.9.8.2.1.3 ARGB-4444 (Data Type 2)

In ARGB16-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for blue data, the next 4 bits for green data the next 4 bits for red data and the upper most 4 bits for the blend value.

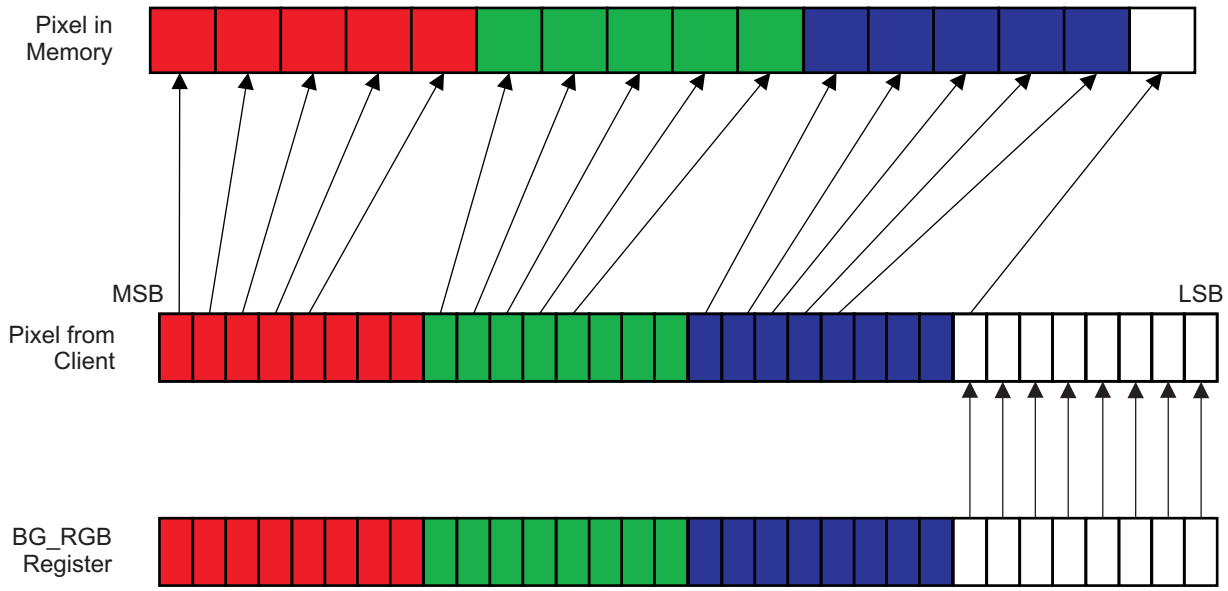
Figure 10-62. ARGB-4444 (Data Type 2)



10.3.9.8.2.1.4 RGBA-5551 (Data Type 3)

In RGBA-5551 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest most bit for the blend value to use 0 or 0xff the next 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data.

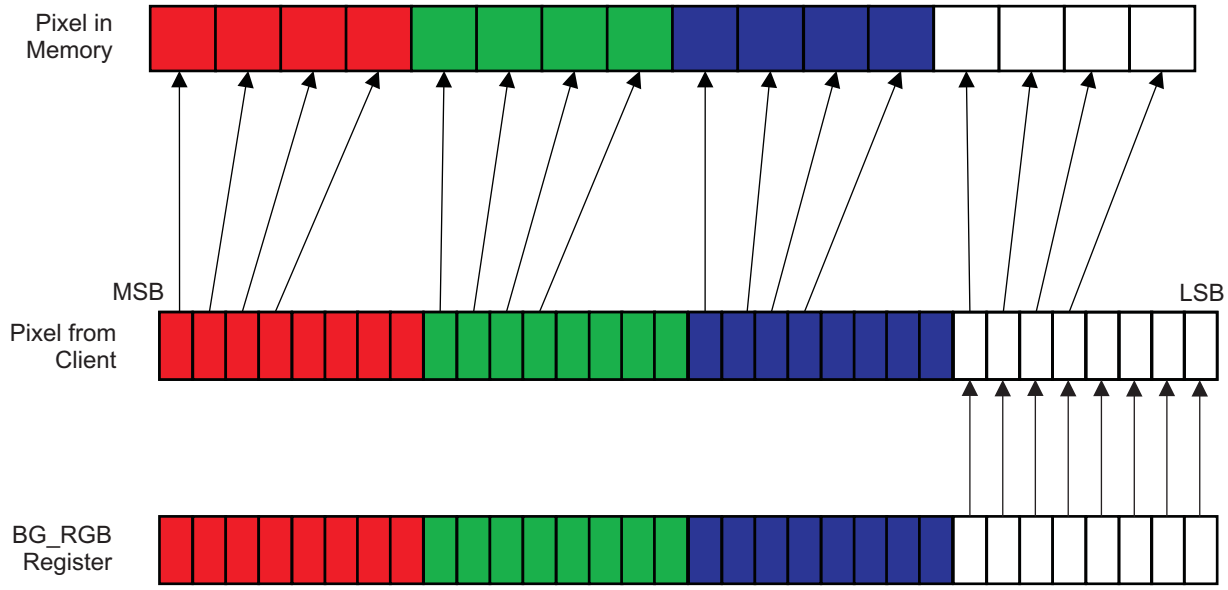
Figure 10-63. RGBA-5551 (Data Type 3)



10.3.9.8.2.1.5 RGBA-4444 (Data Type 4)

In RGBA-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for the blend value, the next 4 bits for blue data, the next 4 bits for green data and the upper most 4 bits for red data.

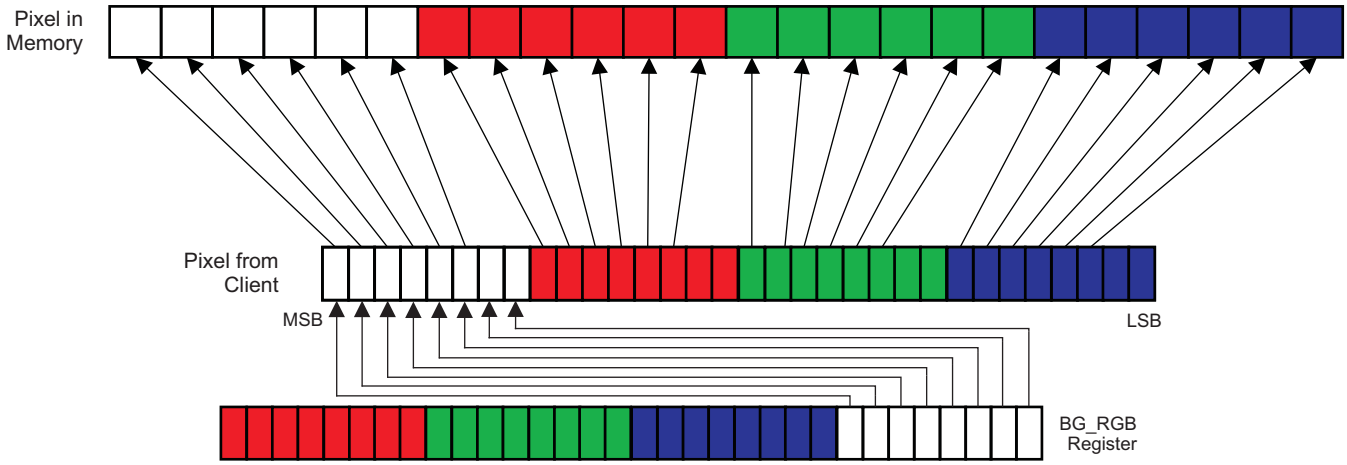
Figure 10-64. RGBA-4444 (Data Type 4)



10.3.9.8.2.1.6 ARGB24-6666 (Data Type 5)

In ARGB24-6666 mode, each pixel is a single ARGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper most 6 bits for red data.

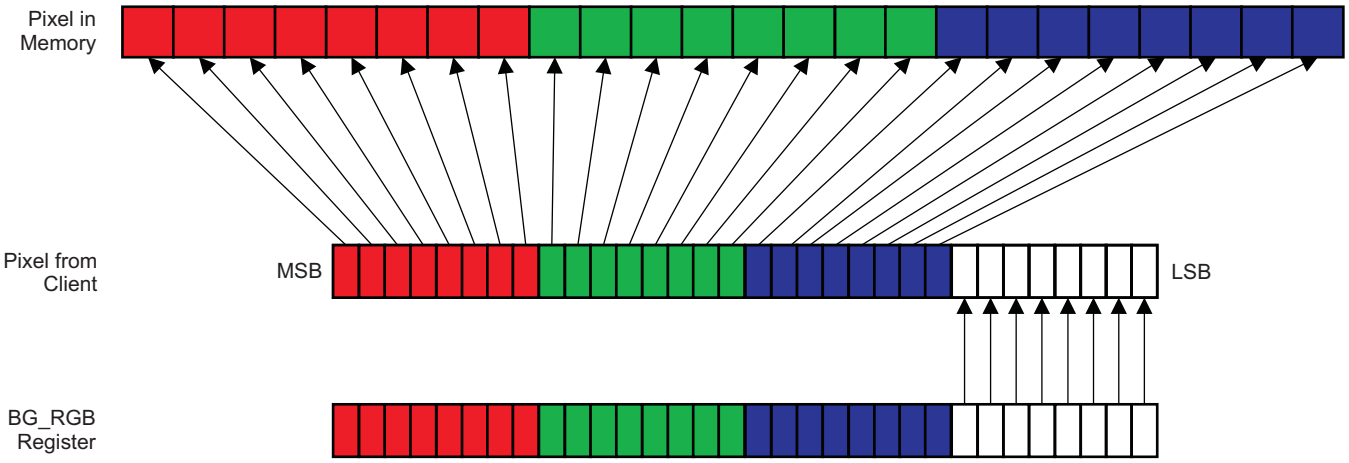
Figure 10-65. ARGB24-6666 (Data Type 5)



10.3.9.8.2.1.7 RGB24-888 (Data Type 6)

In RGB24-888 mode, each pixel is a single RGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data it uses the VPE_BG_RGB Blend value for the Blend value.

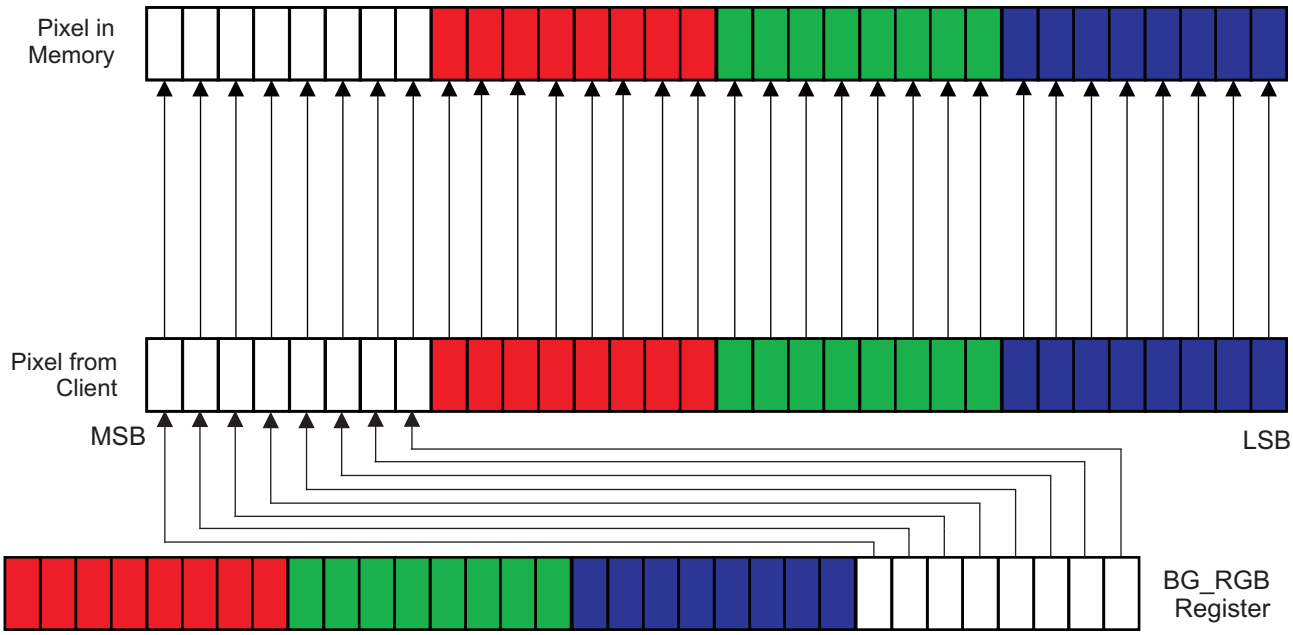
Figure 10-66. RGB24-888 (Data Type 6)



10.3.9.8.2.1.8 ARGB32-8888 (Data Type 7)

In ARGB32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bitsof data uses the lowest 8 bits for blue data, the next 8 bits for green data and the next 8 bits for red data it uses the upper most bits for the blend value.

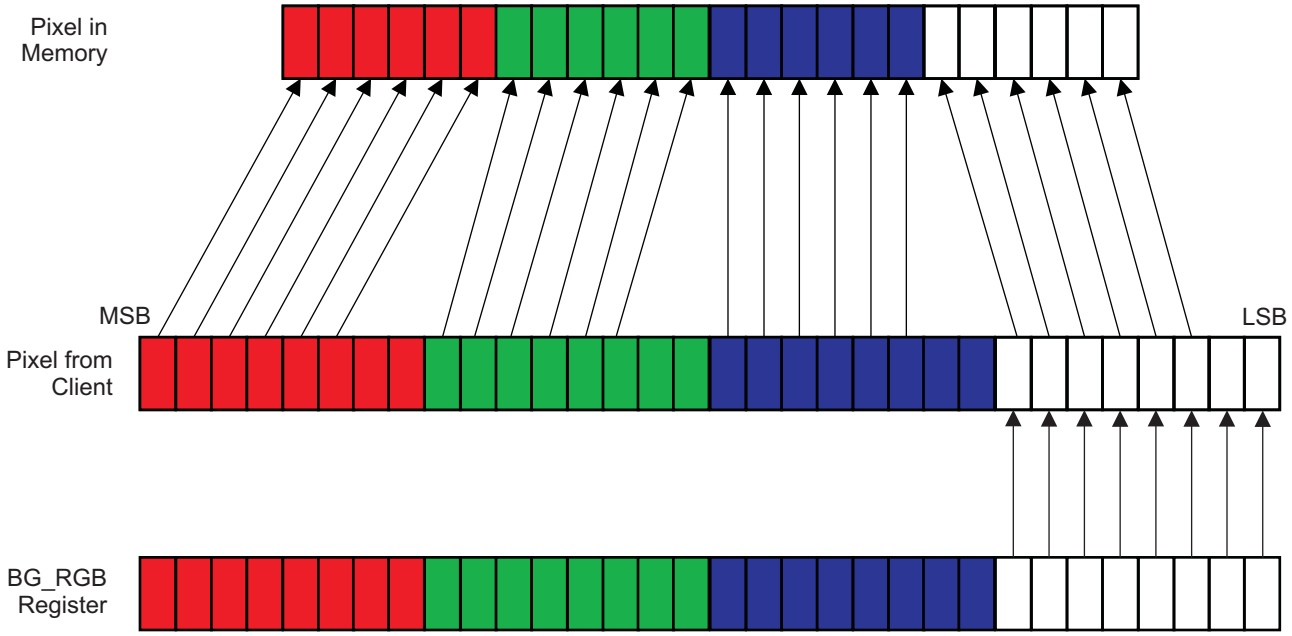
Figure 10-67. ARGB32-8888 (Data Type 7)



10.3.9.8.2.1.9 RGBA24-6666 (Data Type 8)

In RGBA24-6666 mode, each pixel is a single RGBA pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper 6 bits for red data.

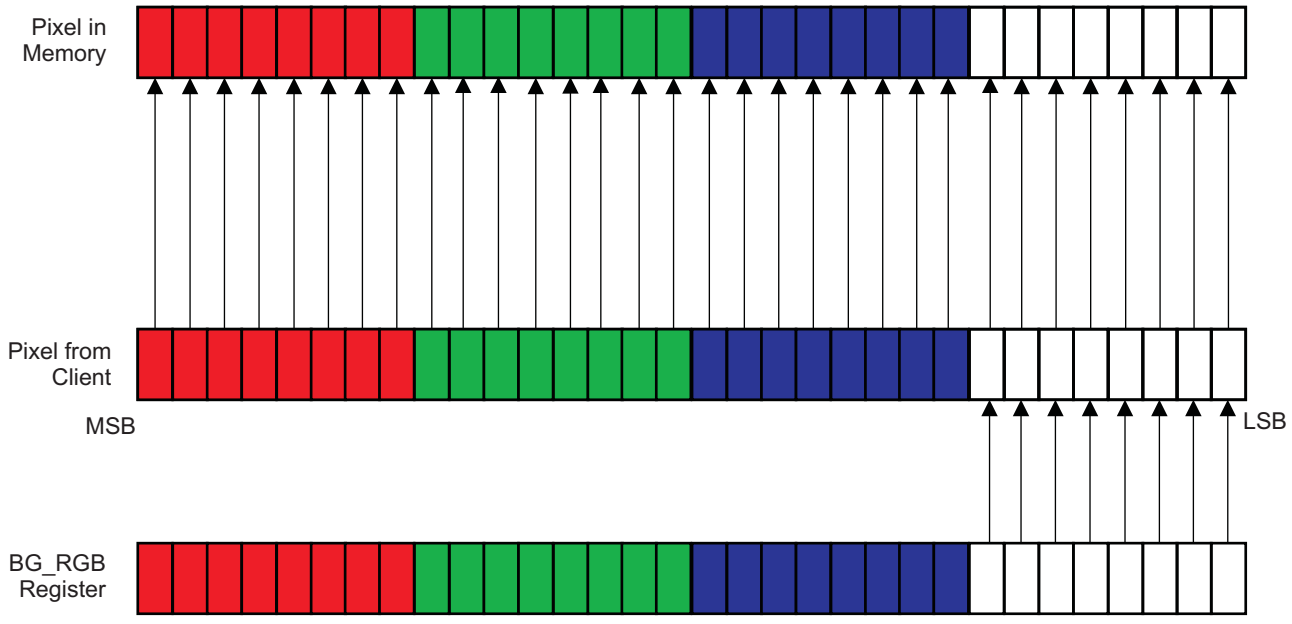
Figure 10-68. RGBA24-6666 (Data Type 8)



10.3.9.8.2.1.10 RGBA32-8888 (Data Type 9)

In RGBA32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bits of data uses the lowest 8 bits for the blend value the next 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data.

Figure 10-69. RGBA32-8888 (Data Type 9)

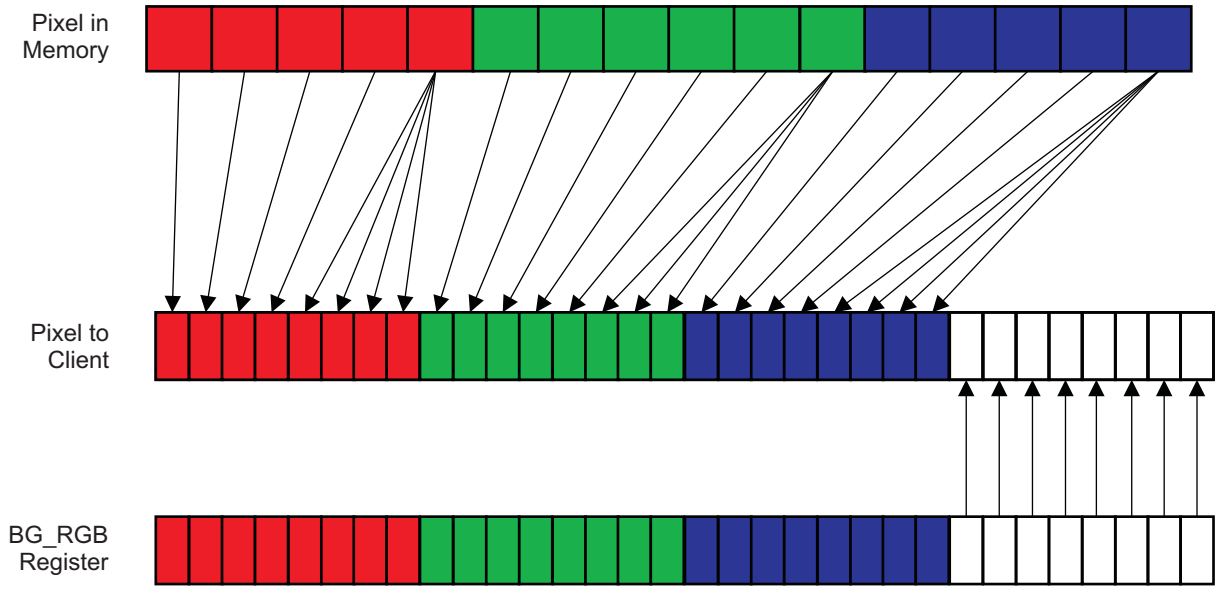


10.3.9.8.2.2 Output Data Formats

10.3.9.8.2.2.1 RGB16-565 (Data Type 0)

In RGB16-565 mode, each pixel is a single RGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the middle 6 bits for green data and the upper 5 bits for red data.

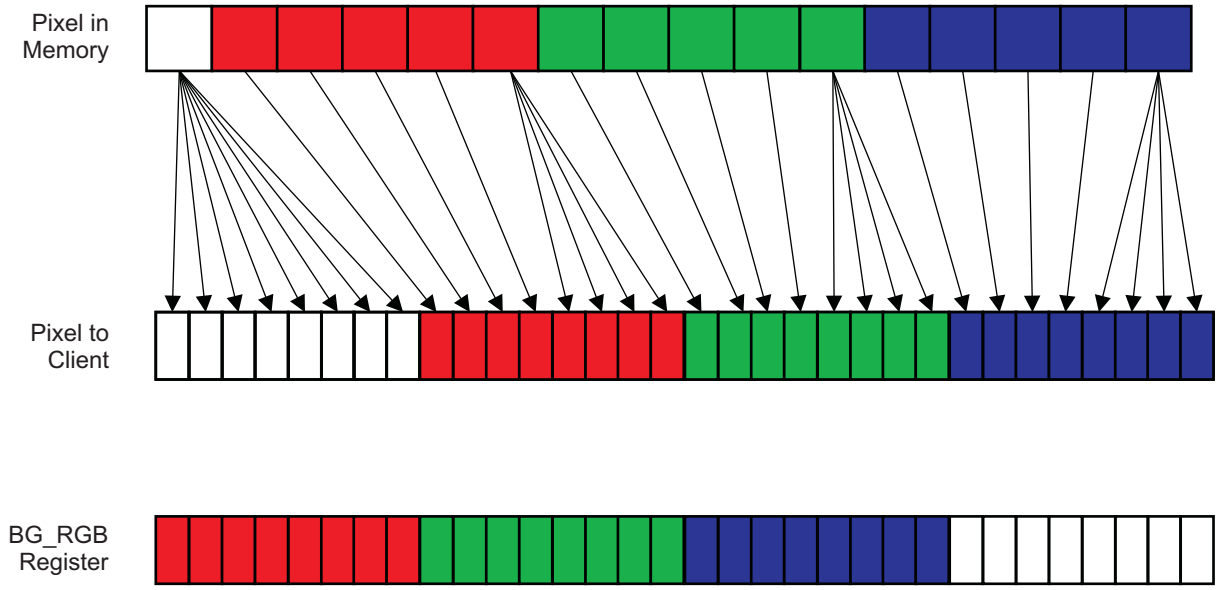
Figure 10-70. RGB16-565 (Data Type 0)



10.3.9.8.2.2.2 ARGB-1555 (Data Type 1)

In ARGB-1555 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data and the upper most bit for the blend value to use 0 or 0xFF.

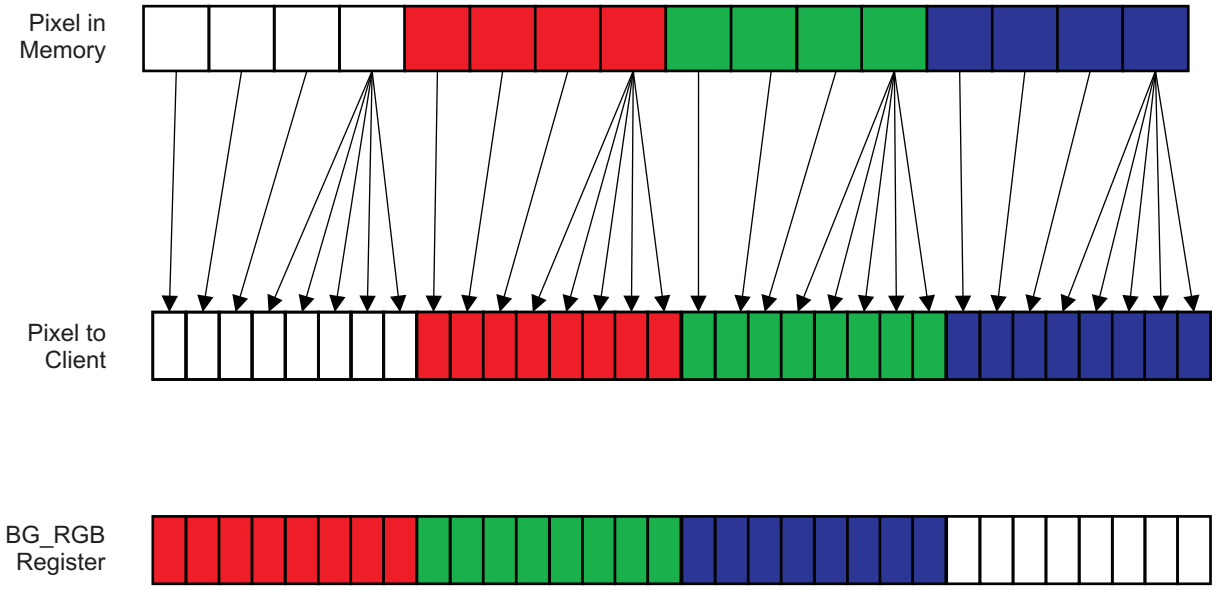
Figure 10-71. ARGB-1555 (Data Type 1)



10.3.9.8.2.2.3 ARGB-4444 (Data Type 2)

In ARGB16-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for blue data, the next 4 bits for green data the next 4 bits for red data and the upper most 4 bits for the blend value.

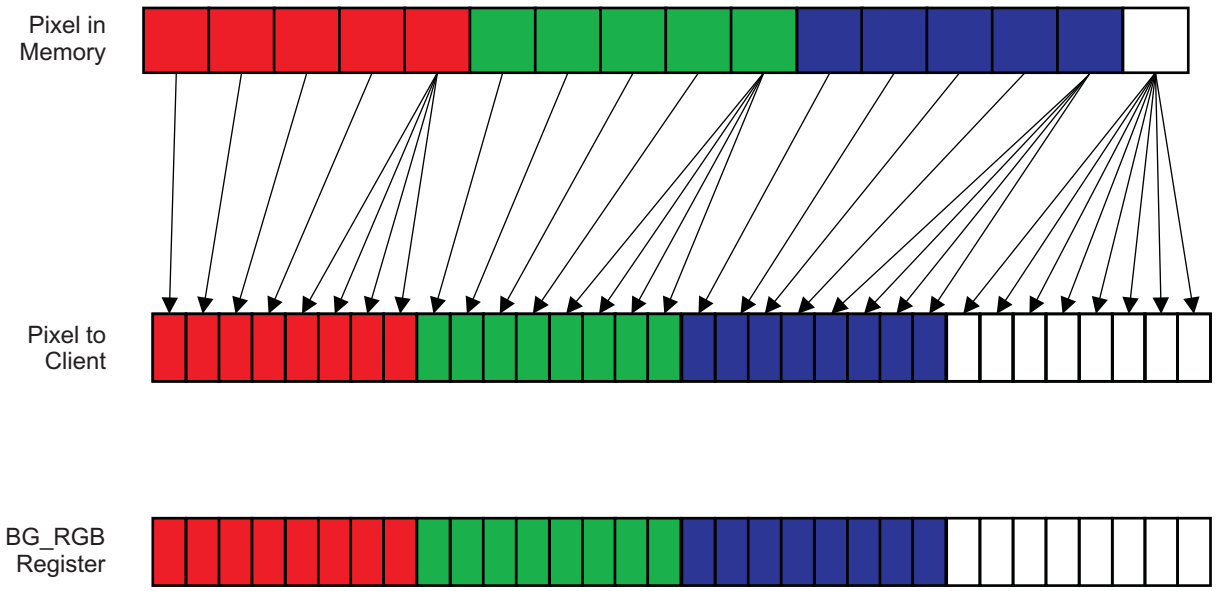
Figure 10-72. ARGB-4444 (Data Type 2)



10.3.9.8.2.2.4 RGBA-5551 (Data Type 3)

In RGBA-5551 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest most bit for the blend value to use 0 or 0xff the next 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data.

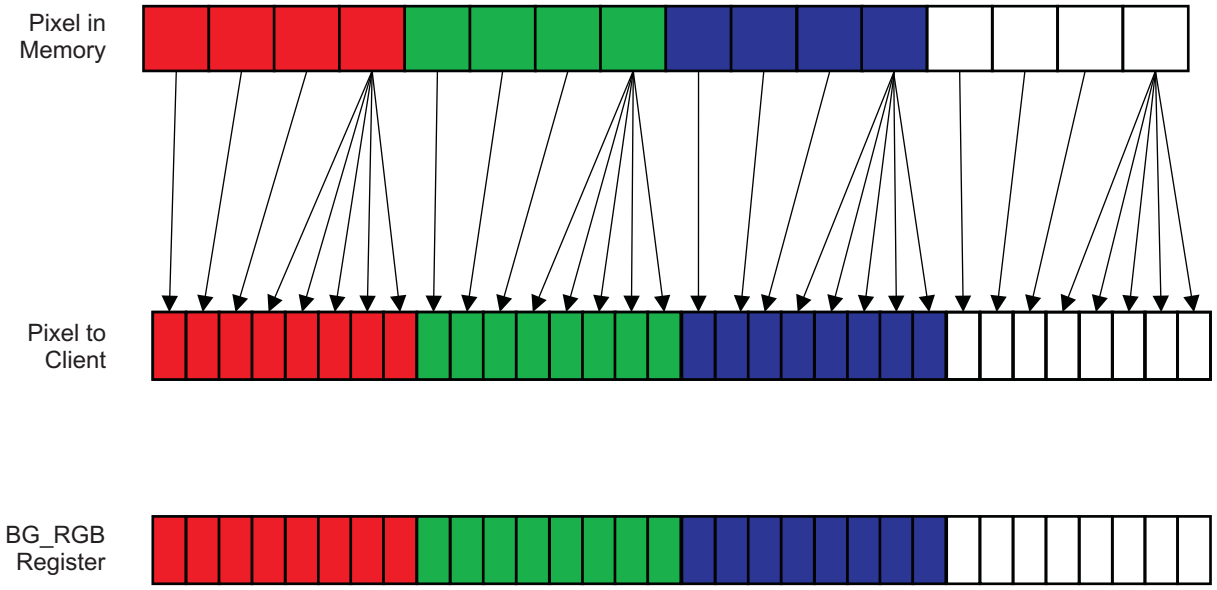
Figure 10-73. RGBA-5551 (Data Type 3)



10.3.9.8.2.2.5 RGBA-4444 (Data Type 4)

In RGBA-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for the blend value, the next 4 bits for blue data, the next 4 bits for green data and the upper most 4 bits for red data.

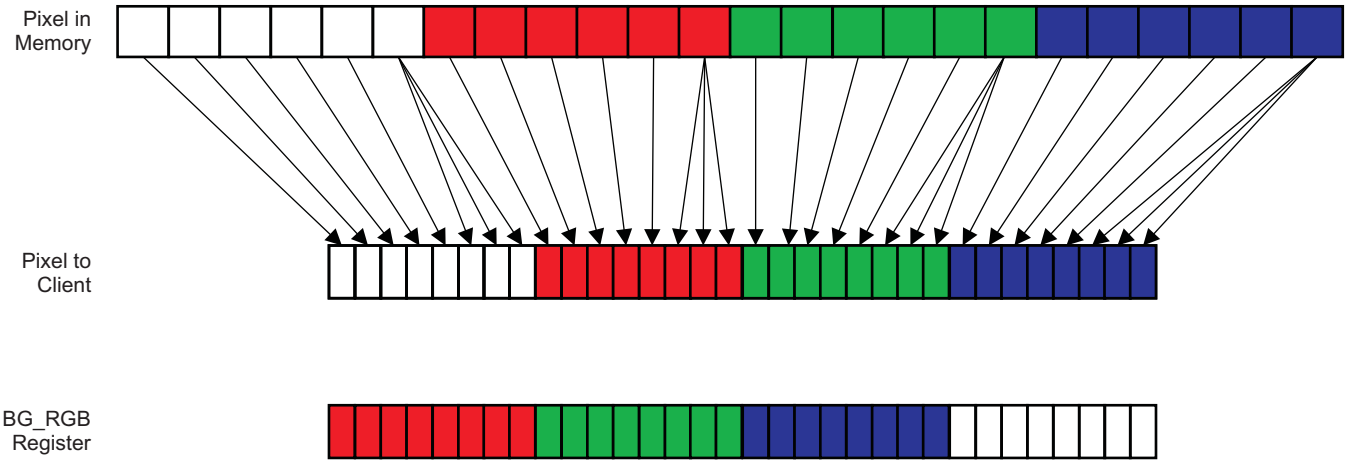
Figure 10-74. RGBA-4444 (Data Type 4)



10.3.9.8.2.2.6 ARGB24-6666 (Data Type 5)

In ARGB24-6666 mode, each pixel is a single ARGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper most 6 bits for red data.

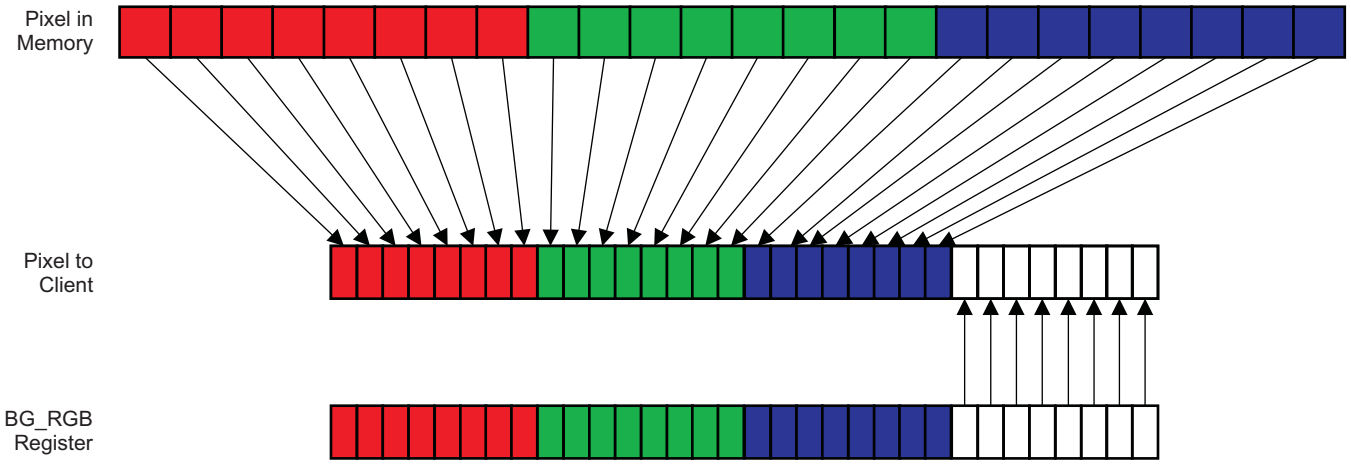
Figure 10-75. ARGB24-6666 (Data Type 5)



10.3.9.8.2.2.7 RGB24-888 (Data Type 6)

In RGB24-888 mode, each pixel is a single RGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data it uses the VPE_BG_RGB Blend value for the Blend value.

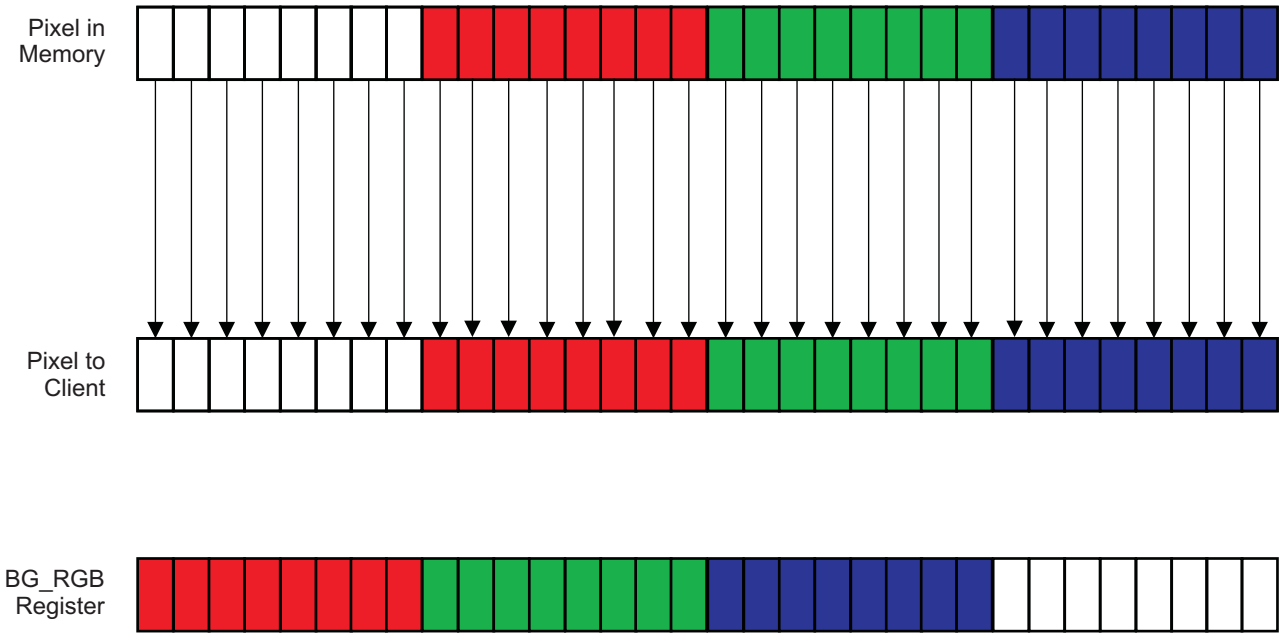
Figure 10-76. RGB24-888 (Data Type 6)



10.3.9.8.2.2.8 ARGB32-8888 (Data Type 7)

In ARGB32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bitsof data uses the lowest 8 bits for blue data, the next 8 bits for green data and the next 8 bits for red data it uses the upper most bits for the blend value.

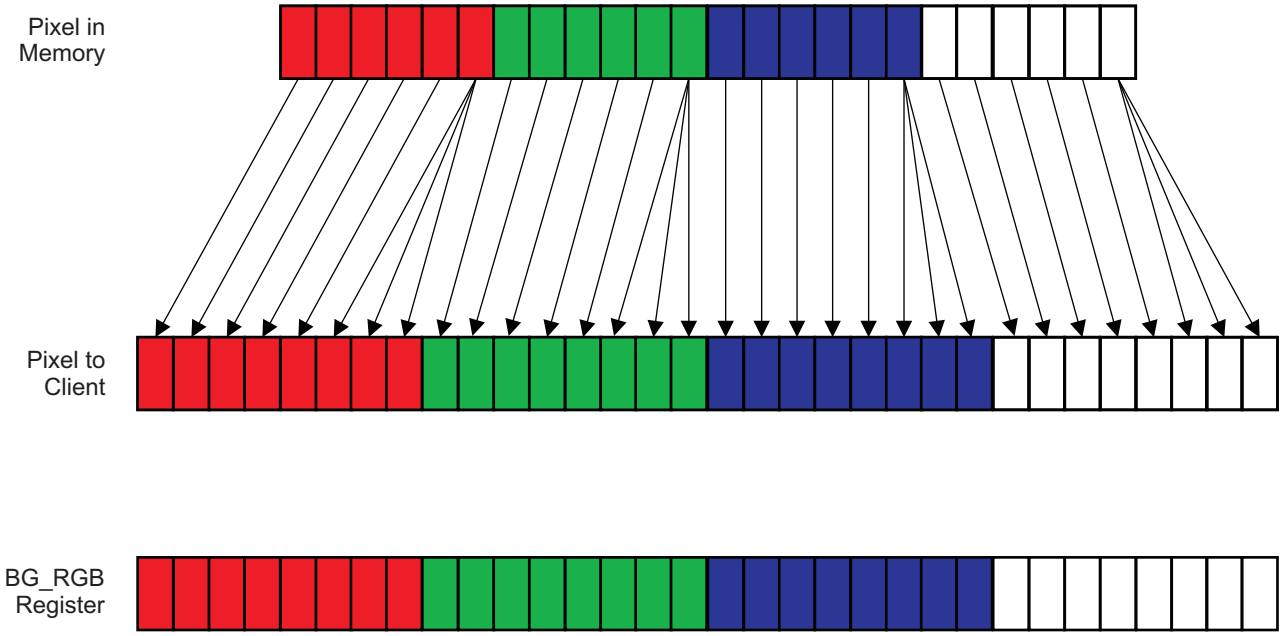
Figure 10-77. ARGB32-8888 (Data Type 7)



10.3.9.8.2.2.9 RGBA24-6666 (Data Type 8)

In RGBA24-6666 mode, each pixel is a single RGBA pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper 6 bits for red data.

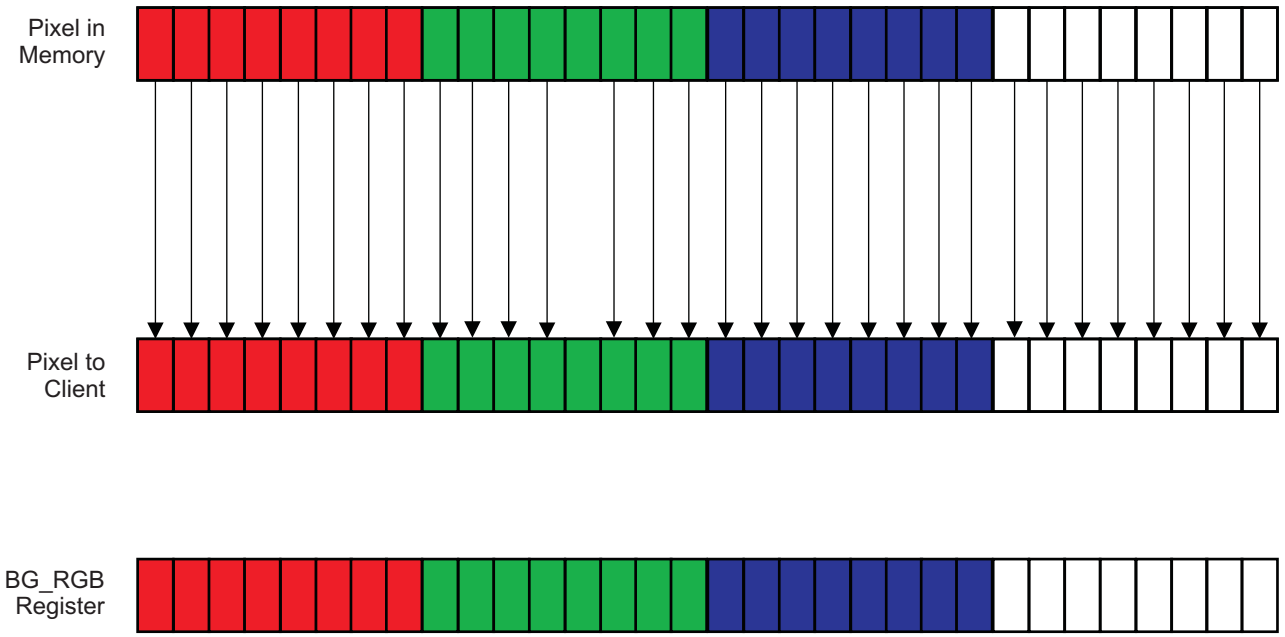
Figure 10-78. RGBA24-6666 (Data Type 8)



10.3.9.8.2.2.10 RGBA32-8888 (Data Type 9)

In RGBA32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bits of data uses the lowest 8 bits for the blend value the next 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data.

Figure 10-79. RGBA32-8888 (Data Type 9)



10.3.9.8.3 Miscellaneous Data Type

The Miscellaneous channel type is for any data type that is not a normal video type. The Miscellaneous channel type makes no assumptions on data type and just passes the data to the client and supports a single buffer for the client. The size of the miscellaneous data type is always determined by the Data Type field of the descriptor which specifies the number of bits in the descriptor.

A memory structure used to describe a desired memory transaction to or from a client. The descriptor at a minimum gives an address location for the memory portion of the transfer, the channel to use for this transaction and the size of the transaction. The data descriptor can also contain attributes to be passed down to the client or be linked to another data descriptor to form a larger frame from many smaller frames.

10.3.10 VPE Software Reset

Software reset in the VPE module can be done by setting the VPE_CLKC_RST[1] PRIM_DP_RST and VPE_CLKC_RST[0] VPDMA_RST for VPE VPDMA to 0x1. Software must ensure that the software reset completes before performing operations within the VPE module.

10.3.11 VPE Power and Clocks Management

The VPE modules support the MStandby/Wait and IdleReq/SidleAck protocols as defined in Chapter 3, *Power, Reset, and Clock Management.*

Power Management within the VPE module can be accomplished in several ways:

- L4 MConnect/SConnect can disable the internal L4 clock network
- L3 MConnect/Sconnect can disable the internal L3 clock network

These items are accomplished using the standard slave idle (for L4) and master standby (for L3) protocols. When these modules are instructed to disable clocks for the internal L3 or L4 (MMR) clock domains, the internal clock networks will be shut down. This shut down applies to the clock signals - L3_CLK and L4_CLK.

10.3.11.1 VPE Clocks

The VPE internal clock domains can only be shut down by writing the appropriate register bit within the Clock Enable register - VPE_CLKC_CLKEN[1] PRIM_DP_EN and VPE_CLKC_CLKEN[0] VPDMA_EN for the VPDMA engine

10.3.11.2 VPE Idle Mode

The VPE supports no-idle mode, force-idle mode, and smart-idle mode. The mode can be selected byprogramming the appropriate value in the VPE_SYSCONFIG[3:2] IDLEMODE bit field.

Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.

- Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements.
- No-idle mode: local target never enters idle state.
- Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA related requests) wakeup events

10.3.11.3 VPE StandBy Mode

The VPE supports no-standby mode, force-standby mode, and a single smart-standby mode. The mode is set in the VPE_SYSCONFIG[5:4] STANDBYMODE bit field.

Configuration of the local initiator state management mode:

- Force-standby mode: local initiator is unconditionally placed in standby state.
- No-standby mode: local initiator is unconditionally placed out of standby state.

10.4 VPE Register Manual

10.4.1 VPE Instance Summary

Table 10-55. VPE Instance Summary

Module Name	Module Base Address	Size
VPE_TOP_LEVEL	0x489D 0000	288 Bytes
VPE_CHR_US_INST_0	0x489D 0300	36 Bytes
VPE_CHR_US_INST_1	0x489D 0400	36 Bytes
VPE_CHR_US_INST_2	0x489D 0500	36 Bytes
VPE_DEI	0x489D 0600	60 Bytes
VPE_SC	0x489D 0700	128 Bytes
VPE_CSC	0x489D 5700	24 Bytes
VPE_VPDMA	0x489D D000	980 Bytes

10.4.2 VPE_CSC Registers

10.4.2.1 VPE_CSC Register Summary

Table 10-56. VPE_CSC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_CSC Base Address
VPE_CSC00	RW	32	0x0000 0000	0x489D 5700
VPE_CSC01	RW	32	0x0000 0004	0x489D 5704
VPE_CSC02	RW	32	0x0000 0008	0x489D 5708
VPE_CSC03	RW	32	0x0000 000C	0x489D 570C
VPE_CSC04	RW	32	0x0000 0010	0x489D 5710
VPE_CSC05	RW	32	0x0000 0014	0x489D 5714

10.4.2.2 VPE_CSC Register Description

Table 10-57. VPE_CSC00

Address Offset	0x0000 0000		
Physical Address	0x489D 5700	**Instance**	VPE_CSC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								B0								RESERVED								A0							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	B0	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0
15:13	RESERVED		R	0x0
12:0	A0	Its is represented as Q3.10 number. So the value ranges from -4 to +4. To convert a decimal number, multiply the number by 1024 and write it in the register in hex format. For example, to program 0.673, 0x2B1 should be written in the register. (int)(0.673 X 1024) = (int)689.152 = 689 = 0x2B1. If the real number is negative, then multiply it by 1024, and convert it to 2's compliment format in 12-bit. For example, if a coefficient is - 1.893, 0x186E needs to be written in the register. (int)(-1.893*1024)= -1938 = 0x186E (2'S compliment format of -1938 in 13-bit width)	RW	0x0

Table 10-58. Register Call Summary for Register VPE_CSC00

VPE Functional Description
• CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
VPE Register Manual
• VPE_CSC Register Summary: [10]
• VPE_CSC Register Description: [11] [12] [13] [14] [15] [16] [17] [18]

Table 10-59. VPE_CSC01

Address Offset	0x0000 0004		
Physical Address	0x489D 5704	**Instance**	VPE_CSC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								A1								RESERVED								C0							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	A1	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0
15:13	RESERVED		R	0x0
12:0	C0	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0

Table 10-60. Register Call Summary for Register VPE_CSC01

VPE Functional Description
• CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
VPE Register Manual
• VPE_CSC Register Summary: [10]

Table 10-61. VPE_CSC02

Address Offset	0x0000 0008		
Physical Address	0x489D 5708	**Instance**	VPE_CSC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								C1								RESERVED								B1							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	C1	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0
15:13	RESERVED		R	0x0
12:0	B1	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0

Table 10-62. Register Call Summary for Register VPE_CSC02

VPE Functional Description

- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VPE Register Manual

- VPE_CSC Register Summary: [10]

Table 10-63. VPE_CSC03

Address Offset	0x0000 000C		
Physical Address	0x489D 570C	**Instance**	VPE_CSC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								B2								RESERVED								A2							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:16	B2	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0
15:13	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
12:0	A2	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0

Table 10-64. Register Call Summary for Register VPE_CSC03

VPE Functional Description
- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VPE Register Manual
- VPE_CSC Register Summary: [10]

Table 10-65. VPE_CSC04

Address Offset	0x0000 0010		
Physical Address	0x489D 5710	**Instance**	VPE_CSC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								D0								RESERVED								C2							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	D0	Coefficients of color space converter. This coefficient is an integer number in the range of 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. For example, if this coefficient is 749, then 0x2ED (hex format) should be assigned to this register. Another example, if this coefficient is -1021, then 0xC03 should be assigned to this register.	RW	0x0
15:13	RESERVED		R	0x0
12:0	C2	Coefficients of color space converter. This coefficient is a real number in the range of -4. to +4 represent in Q3.10 format. The MSB is sign bit. (Same format conversion as A0 in VPE_CSC00)	RW	0x0

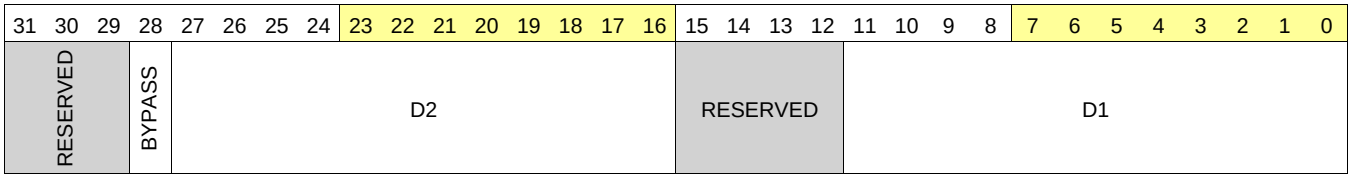
Table 10-66. Register Call Summary for Register VPE_CSC04

VPE Functional Description
- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]

VPE Register Manual
- VPE_CSC Register Summary: [10]
- VPE_CSC Register Description: [11] [12]

Table 10-67. VPE_CSC05

Address Offset	0x0000 0014		
Physical Address	0x489D 5714	**Instance**	VPE_CSC
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	BYPASS	Full CSC bypass mode	RW	0x0
27:16	D2	Coefficients of color space converter. This coefficient is an integer number in the range of -2048 to 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0 in VPE_CSC04)	RW	0x0
15:12	RESERVED		R	0x0
11:0	D1	Coefficients of color space converter. This coefficient is an integer number in the range of -2048 to 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0 in VPE_CSC04)	RW	0x0

Table 10-68. Register Call Summary for Register VPE_CSC05

VPE Functional Description
- CSC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9]
- CSC Bypass Mode: [10]

VPE Register Manual
- VPE_CSC Register Summary: [11]

10.4.3 VPE_SC Registers

10.4.3.1 VPE_SC Register Summary

Table 10-69. VPE_SC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_SC Base Address
VPE_CFG_SC0	RW	32	0x0000 0000	0x489D 0700
VPE_CFG_SC1	RW	32	0x0000 0004	0x489D 0704
VPE_CFG_SC2	RW	32	0x0000 0008	0x489D 0708
VPE_CFG_SC3	RW	32	0x0000 000C	0x489D 070C
VPE_CFG_SC4	RW	32	0x0000 0010	0x489D 0710
VPE_CFG_SC5	RW	32	0x0000 0014	0x489D 0714
VPE_CFG_SC6	RW	32	0x0000 0018	0x489D 0718
RESERVED	R	32	0x0000 001C	0x489D 071C
VPE_CFG_SC8	RW	32	0x0000 0020	0x489D 0720
VPE_CFG_SC9	RW	32	0x0000 0024	0x489D 0724
VPE_CFG_SC10	RW	32	0x0000 0028	0x489D 0728
VPE_CFG_SC11	RW	32	0x0000 002C	0x489D 072C
VPE_CFG_SC12	RW	32	0x0000 0030	0x489D 0730
VPE_CFG_SC13	RW	32	0x0000 0034	0x489D 0734
RESERVED	R	32	0x0000 0038	0x489D 0738
RESERVED	R	32	0x0000 003C	0x489D 073C
RESERVED	R	32	0x0000 0040	0x489D 0740
RESERVED	R	32	0x0000 0044	0x489D 0744
VPE_CFG_SC18	RW	32	0x0000 0048	0x489D 0748

Table 10-69. VPE_SC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VPE_SC Base Address
VPE_CFG_SC19	RW	32	0x0000 004C	0x489D 074C
VPE_CFG_SC20	RW	32	0x0000 0050	0x489D 0750
VPE_CFG_SC21	RW	32	0x0000 0054	0x489D 0754
VPE_CFG_SC22	RW	32	0x0000 0058	0x489D 0758
RESERVED	R	32	0x0000 005C	0x489D 075C
VPE_CFG_SC24	RW	32	0x0000 0060	0x489D 0760
VPE_CFG_SC25	RW	32	0x0000 0064	0x489D 0764
RESERVED	R	32	0x0000 0068	0x489D 0768
RESERVED	R	32	0x0000 006C	0x489D 076C
RESERVED	R	32	0x0000 0070	0x489D 0770
RESERVED	R	32	0x0000 0074	0x489D 0774
RESERVED	R	32	0x0000 0078	0x489D 0778
RESERVED	R	32	0x0000 007C	0x489D 077C

10.4.3.2 VPE_SC Register Description

Table 10-70. VPE_CFG_SC0

Address Offset	0x0000 0000		
Physical Address	0x489D 0700	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															CFG_FID_SELFGEN	CFG_TRIM	CFG_Y_PK_EN	RESERVED		RESERVED	CFG_INTERLACE_I	CFG_HP_BYPASS	CFG_DCM_4X	CFG_DCM_2X	CFG_AUTO_HS	RESERVED	CFG_USE_RAV	CFG_INVT_FID	CFG_SC_BYPASS	CFG_LINEAR	CFG_INTERLACE_O

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	CFG_FID_SELFGEN	FID self generate enable. When input is progressive and this bit is set, the SC generates self-toggling (top/bottom) output FID when performing interlacing.	RW	0x0
15	CFG_TRIM	Trimming enable. When 1, the input image whose size is specified by orgW and orgH registers is trimmed to the size with srcW and srcH from the offset specified by offW and offH. 0x0 : Disable trimming 0x1 : Enable trimming	RW	0x0
14	CFG_Y_PK_EN	This parameter is used by peaking block. 0: disable luma peaking 1: enable luma peaking	RW	0x0
13:12	RESERVED		R	0x0
11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10	CFG_INTERLACE_I	This parameter is used by horizontal and vertical scaling. 0x0 : The input video format is progressive 0x1 : The input video format is interlace	RW	0x0
9	CFG_HP_BYPASS	This parameter is used by horizontal scaling. If cfg_auto_hs is 0, horizontal polyphase filter is always enabled. In this case, this register is DON'T CARE. If cfg_auto_hs is 1, then: 0x0: The polyphase scaler is always used regardless of the scaling ratio. 0x1: The polyphase scaler is bypassed only when (tar_w == src_w) or (tar_w == src_w/2) or (tar_w == src_w/4)	RW	0x0
8	CFG_DCM_4X	This parameter is used by horizontal scaling. 0: the 4X decimation filter is disabled 1: the 4X decimation filter is enabled **Note:** (1) Either 2X or 4X can be enabled, but they cannot be enabled simultaneously. (2) This register is only set to 1 when it makes sense to do so. Typically, it is used when (horizontal scale ratio < 0.25). (3) This register is DON'T CARE when cfg_auto_hs = 1	RW	0x0
7	CFG_DCM_2X	This parameter is used by horizontal scaling. 0: the 2X decimation filter is disabled 1: the 2X decimation filter is enabled **Note:** (1) Either 2X or 4X can be enabled, but they cannot be enabled simultaneously. (2) This register is only set to 1 when it makes sense to do so. Typically, it is used when (0.25 < horizontal scale ratio < 0.5). (3) This register is DON'T CARE when cfg_auto_hs = 1.	RW	0x0
6	CFG_AUTO_HS	This parameter is used by horizontal scaling. 0x0 : the cfg_dcm_2x and cfg_dcm_4x bits will enable appropriate decimation filters 0x1 : HW will decide whether up-scaling or down-scaling is required based on horizontal scaling ratio (SR). SR > 0.5 : horizontal polyphase filter is enabled, all decimation filters are disabled SR = 0.5 : dcm_2x is enabled, horizontal polyphase filter is enabled or disabled based on cfg_hp_bypass 0.5 > SR > 0.25 : dcm_2x and horizontal polyphase filter both are enabled SR = 0.25 : dcm_4x is enabled, horizontal polyphase filter is enabled or disabled based on cfg_hp_bypass 0.25 > SR > 0.125 : dcm_4x and horizontal polyphase filter are both enabled SR <= 0.125 : Functionally supported, but not recommended in auto mode for image quality concerns	RW	0x0
5	RESERVED		R	0x0
4	CFG_USE_RAV	This parameter is used by vertical scaling. 0x0 : Poly-phase filter will be used for the vertical scaling 0x1 : Running average filter will be used for the vertical scaling (down scaling only)	RW	0x0

Bits	Field Name	Description	Type	Reset
3	CFG_INVT_FID	This parameter is used by vertical scaling. 0x0 : Progressive input 0x1 : Interlaced input Must be set to 1 when CFG_INTERFACE_I = 1.	RW	0x0
2	CFG_SC_BYPASS	This parameter is general purpose. 0x0 : Scaling module will be engaged 0x1 : Scaling module will be bypassed	RW	0x0
1	CFG_LINEAR	This parameter is used by horizontal scaling. 0x0 : Anamorphic scaling 0x1 : Linear scaling	RW	0x0
0	CFG_INTERLACE_O	This parameter is used by vertical scaling. 0x0 : The output format of SC is progressive (default); 0x1 : The output format of SC is interlace	RW	0x0

Table 10-71. Register Call Summary for Register VPE_CFG_SC0

VPE Functional Description
- SC Functional Description: [0] [1] [2] [3] [4] [5] [6] [7] [8] [9] [10] [11] [12] [13]
- SC Code: [14] [15] [16] [17] [18] [19] [20] [21] [22]

VPE Register Manual
- VPE_SC Register Summary: [23]

Table 10-72. VPE_CFG_SC1

Address Offset	0x0000 0004		
Physical Address	0x489D 0704	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_ROW_ACC_INC																							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:0	CFG_ROW_ACC_INC	This parameter is used by vertical scaling. It defines the increment of the row accumulator in vertical poly-phase filter. It can be calculated by following formulas: For progressive in/progressive out row_acc_inc = round(2^16*(src_h-1)/(tar_h - 1)) For progressive_in/interlace_out row_acc_inc = round(2^16*2*(src_h-1)/(2*tar_h - 1)) For interlace_in/progressive_out row_acc_inc = round(2^16*(2*src_h-1)/(2*(tar_h - 1))) For interlace_in/interlace_out row_acc_inc = round(2^16*(2*src_h - 1)/(2*tar_h - 1)) In case of interlaced input, srcH is input field height (number of field lines), as specified in VPE_CFG_SC5. In case of interlaced output, tarH is output field height (number of field lines), as specified in VPE_CFG_SC4.	RW	0x0

Table 10-73. Register Call Summary for Register VPE_CFG_SC1

VPE Functional Description • SC Functional Description: [0]
VPE Register Manual • VPE_SC Register Summary: [1]

Table 10-74. VPE_CFG_SC2

Address Offset	0x0000 0008		
Physical Address	0x489D 0708	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_ROW_ACC_OFFSET																									

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	CFG_ROW_ACC_OFFSET	This parameter is used by vertical scaling. It defines the vertical offset during vertical scaling. In progressive mode: this offset will be applied to a frame. In interlace mode: this offset will be applied to the top field.	RW	0x0

Table 10-75. Register Call Summary for Register VPE_CFG_SC2

VPE Functional Description • SC Functional Description: [0]
VPE Register Manual • VPE_SC Register Summary: [1]

Table 10-76. VPE_CFG_SC3

Address Offset	0x0000 000C		
Physical Address	0x489D 070C	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_ROW_ACC_OFFSET_B																									

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	CFG_ROW_ACC_OFFSET_B	This parameter is used by vertical scaling. It defines the vertical offset during vertical scaling. In progressive mode: this parameter will not be used. In interlace mode: this offset will be applied to the bottom field.	RW	0x0

Table 10-77. Register Call Summary for Register VPE_CFG_SC3

VPE Functional Description • SC Functional Description: [0]
VPE Register Manual • VPE_SC Register Summary: [1]

Table 10-78. VPE_CFG_SC4

Address Offset	0x0000 0010		
Physical Address	0x489D 0710	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CFG_NLIN_ACC_INIT_U			RESERVED	CFG_LIN_ACC_INC_U			RESERVED	CFG_TAR_W											RESERVED	CFG_TAR_H										

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	CFG_NLIN_ACC_INIT_U	This parameter is used by horizontal scaling. The 3 MSBbits of 'nlin_acc_init' that is defined in CFG_SC10	RW	0x0
27	RESERVED		RW	0x0
26:24	CFG_LIN_ACC_INC_U	This parameter is used by horizontal scaling. The 3 MSBbits of 'lin_acc_inc' that is defined in CFG_SC9	RW	0x0
23	RESERVED		RW	0x0
22:12	CFG_TAR_W	This parameter is a general purpose. Scaled target picture width. unit is pixel. This parameter defines the final output picture size	RW	0x0
11	RESERVED		RW	0x0
10:0	CFG_TAR_H	This parameter is a general purpose. Scaled target picture height (unit is line). This parameter defines the final output picture size. For the interlace output, it should be the number of lines per field.	RW	0x0

Table 10-79. Register Call Summary for Register VPE_CFG_SC4

VPE Functional Description

- SC Functional Description: [0] [1] [2] [3] [4] [5] [6]

VPE Register Manual

- VPE_SC Register Summary: [7]
- VPE_SC Register Description: [8]

Table 10-80. VPE_CFG_SC5

Address Offset	0x0000 0014		
Physical Address	0x489D 0714	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CFG_NLIN_ACC_INC_U			RESERVED								CFG_SRC_W				RESERVED				CFG_SRC_H							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:24	CFG_NLIN_ACC_INC_U	This parameter is used by horizontal scaling. The 3 MSBbits of 'nlin_acc_inc' that is defined in CFG_SC11	RW	0x0
23	RESERVED		RW	0x0
22:12	CFG_SRC_W	This parameter is a general purpose. This parameter defines the width of the source image	RW	0x0
11	RESERVED		RW	0x0
10:0	CFG_SRC_H	This parameter is a general purpose. This parameter defines the height of the source image. For the interlace input, it should be the number of lines per field.	RW	0x0

Table 10-81. Register Call Summary for Register VPE_CFG_SC5

VPE Functional Description
- SC Functional Description: [0] [1] [2] [3] [4]

VPE Register Manual
- VPE_SC Register Summary: [5]
- VPE_SC Register Description: [6]

Table 10-82. VPE_CFG_SC6

Address Offset	0x0000 0018		
Physical Address	0x489D 0718	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CFG_ROW_ACC_INIT_RAV_B											CFG_ROW_ACC_INIT_RAV								

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:10	CFG_ROW_ACC_INIT_RAV_B	This parameter is used by vertical scaling. it is used only when the input is interlace format. In vertical down scaling.. the running average filter is applied. This parameter sets the initialization value of the row accumulator in running average filter (for bottom field of interlace format)	RW	0x0
9:0	CFG_ROW_ACC_INIT_RAV	This parameter is used by vertical scaling. In vertical down scaling.. the running average filter is applied. This parameter sets the initialization value of the row accumulator in running average filter (for progressive format or top field of interlace format)	RW	0x0

Table 10-83. Register Call Summary for Register VPE_CFG_SC6

VPE Functional Description
• SC Functional Description: [0] [1] [2] [3]
VPE Register Manual
• VPE_SC Register Summary: [4]

Table 10-84. VPE_CFG_SC8

Address Offset	0x0000 0020		
Physical Address	0x489D 0720	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_NLIN_RIGHT												RESERVED	CFG_NLIN_LEFT										

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22:12	CFG_NLIN_RIGHT	This parameter is used by horizontal scaling. In anamorphic mode. this parameter defines the width of the strip on right-hand side. In other words. it defines the location of the last pixel where the linear scaling is ended. The unit is the 'pixel location' in an active video line. This parameter will not be used in linear scaling	RW	0x0
11	RESERVED		RW	0x0
10:0	CFG_NLIN_LEFT	This parameter is used by horizontal scaling. In anamorphic mode. this parameter defines the width of the strip on left-hand side. In other words. it defines the location of the last pixel in the left-sidenonlinear strip. The unit is the 'pixel location' in an active video line. This parameter will not be used in linear scaling	RW	0x0

Table 10-85. Register Call Summary for Register VPE_CFG_SC8

VPE Functional Description
• SC Functional Description: [0] [1]
VPE Register Manual
• VPE_SC Register Summary: [2]

Table 10-86. VPE_CFG_SC9

Address Offset	0x0000 0024		
Physical Address	0x489D 0724	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_LIN_ACC_INC																															

Bits	Field Name	Description	Type	Reset
31:0	CFG_LIN_ACC_INC	This parameter is used by horizontal scaling. It defines the increment of the linear accumulator. if SR > 0.5, then • lin_acc_inc = round(2^{24}*(srcWi -1) /(tarWi -1)) else if 0.25 < SR <= 0.5 • lin_acc_inc = round(2^24*(srcWi/2 -1) /(tarWi - 1)) else if SR <= 0.25 • lin_acc_inc = round(2^{24}*(srcWi/4 -1) /(tarWi - 1)) where srcWi and tarWi are the inner source width and the inner target width respectively.	RW	0x0

Table 10-87. Register Call Summary for Register VPE_CFG_SC9

VPE Functional Description
- SC Functional Description: [0] [1]

VPE Register Manual
- VPE_SC Register Summary: [2]

Table 10-88. VPE_CFG_SC10

Address Offset	0x0000 0028		
Physical Address	0x489D 0728	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_NLIN_ACC_INIT																															

Bits	Field Name	Description	Type	Reset
31:0	CFG_NLIN_ACC_INIT	This parameter is used by horizontal scaling. It is used by nonlinear scaling only. It defines the initialization value of the nonlinear accumulator. nlin_acc_init = K*(1-2*d) Here the definitions of K and d are the same as in CFG_SC11	RW	0x0

Table 10-89. Register Call Summary for Register VPE_CFG_SC10

VPE Register Manual
- VPE_SC Register Summary: [0]

Table 10-90. VPE_CFG_SC11

Address Offset	0x0000 002C		
Physical Address	0x489D 072C	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_NLIN_ACC_INC																															

Bits	Field Name	Description	Type	Reset

31:0	CFG_NLIN_ACC_INC	This parameter is used by horizontal scaling. It is used by nonlinear scaling only. It defines the increment of the nonlinear accumulator.
		if upscaling then
		d = 0
		if Ltar !=0 then
		K =round[2^{24}*Lsrc/(Ltar*Ltar)]
		where Lsrc= (srcW-srcWi)/2
		else K = 0
		elseif downscaling
		d = (tarW-1)/2
		if Ltar!=0 then
		K = round[2^{24} * Lsrc / (Ltar*(Ltar-2d))]
		where Lsrc= (srcW-srcWi)/(2n) and n=1..2 or 4
		else K = 0
		nlin_acc_inc = 2*K (negative for downscaling)
			RW	0x0

Table 10-91. Register Call Summary for Register VPE_CFG_SC11

VPE Register Manual
- VPE_SC Register Summary: [0]

Table 10-92. VPE_CFG_SC12

Address Offset	0x0000 0030		
Physical Address	0x489D 0730	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_COL_ACC_OFFSET																									

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:0	CFG_COL_ACC_OFFSET	This parameter is used in horizontal scaling. It defines the luma accumulator's offset. Normally this parameter can be set as 0 if no horizontal offset is involved. In some applications, such as Pan and Scan. A corresponding offset value should be set. The format is 1.24.	RW	0x0

Table 10-93. Register Call Summary for Register VPE_CFG_SC12

VPE Functional Description
- SC Code: [0]

VPE Register Manual
- VPE_SC Register Summary: [1]

Table 10-94. VPE_CFG_SC13

Address Offset	0x0000 0034		
Physical Address	0x489D 0734	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CFG_SC_FACTOR_RAV									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	CFG_SC_FACTOR_RAV	This parameter is used by vertical scaling. Vertical scaling factor: It is defined as following: 1024*tarH/srcH. It is used for downscaling by the running average filter	RW	0x0

Table 10-95. Register Call Summary for Register VPE_CFG_SC13

VPE Functional Description
- SC Functional Description: [0] [1] [2]
- SC Code: [3] [4]

VPE Register Manual
- VPE_SC Register Summary: [5]

Table 10-96. VPE_CFG_SC18

Address Offset	0x0000 0048		
Physical Address	0x489D 0748	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CFG_HS_FACTOR							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	CFG_HS_FACTOR	This parameter is used by horizontal scaling. Horizontal-scaling-factor = tarWi/srcWi. Numerical format: 6.4 (6 bit integer and 4 bit fraction)	RW	0x0

Table 10-97. Register Call Summary for Register VPE_CFG_SC18

VPE Functional Description
- SC Code: [0]

VPE Register Manual
- VPE_SC Register Summary: [1]

Table 10-98. VPE_CFG_SC19

Address Offset	0x0000 004C		
Physical Address	0x489D 074C	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_HPF_COEF3								CFG_HPF_COEF2								CFG_HPF_COEF1								CFG_HPF_COEF0							

Bits	Field Name	Description	Type	Reset
31:24	CFG_HPF_COEF3	This parameter is used by the peaking block. Defines the coefficient 3 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
23:16	CFG_HPF_COEF2	This parameter is used by the peaking block. Defines the coefficient 2 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0

Bits	Field Name	Description	Type	Reset
15:8	CFG_HPF_COEF1	This parameter is used by the peaking block. Defines the coefficient 1 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
7:0	CFG_HPF_COEF0	This parameter is used by the peaking block. Defines the coefficient 0 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0

Table 10-99. Register Call Summary for Register VPE_CFG_SC19

VPE Functional Description
- SC Functional Description: [0] [1]
- SC Code: [2]

VPE Register Manual
- VPE_SC Register Summary: [3]

Table 10-100. VPE_CFG_SC20

Address Offset	0x0000 0050		
Physical Address	0x489D 0750	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			CFG_NL_LIMIT									RESERVED	CFG_HPF_NORM_SHIFT			CFG_HPF_COEF5								CFG_HPF_COEF4							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:20	CFG_NL_LIMIT	This parameter is used by the peaking block. The maximum of clipping.	RW	0x0
19	RESERVED		R	0x0
18:16	CFG_HPF_NORM_SHIFT	This parameter is used by the peaking block. Defines the decimal point of the hpf coefficient.	RW	0x0
15:8	CFG_HPF_COEF5	This parameter is used by the peaking block. Defines the coefficient 5 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
7:0	CFG_HPF_COEF4	This parameter is used by the peaking block. Defines the coefficient 4 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0

Table 10-101. Register Call Summary for Register VPE_CFG_SC20

VPE Functional Description
- SC Functional Description: [0] [1] [2]
- SC Code: [3]

VPE Register Manual
- VPE_SC Register Summary: [4]

Table 10-102. VPE_CFG_SC21

Address Offset	0x0000 0054		
Physical Address	0x489D 0754	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_NL_LO_SLOPE								RESERVED								CFG_NL_LO_THR							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	CFG_NL_LO_SLOPE	This parameter is used by the peaking block. Slope of the nonlinear peaking function. The format is fixed point 4.4.	RW	0x0
15:9	RESERVED		R	0x0
8:0	CFG_NL_LO_THR	This parameter is used by the peaking block. Threshold for the nonlinear peaking function. Must be 0	RW	0x0

Table 10-103. Register Call Summary for Register VPE_CFG_SC21

VPE Functional Description

- SC Functional Description: [0] [1]
- SC Code: [2]

VPE Register Manual

- VPE_SC Register Summary: [3]

Table 10-104. VPE_CFG_SC22

Address Offset	0x0000 0058		
Physical Address	0x489D 0758	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													CFG_NL_HI_SLOPE_SHIFT			RESERVED								CFG_NL_HI_THR							

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:16	CFG_NL_HI_SLOPE_SHIFT	This parameter is used by the peaking block. Slope of the nonlinear peaking function. The gain is $2^{(nl\_hi\_slope\_shift-3)}$.	RW	0x0
15:9	RESERVED		R	0x0
8:0	CFG_NL_HI_THR	This parameter is used by the peaking block. Threshold for the nonlinear peaking function. Must be nl_hi_thr.	RW	0x0

Table 10-105. Register Call Summary for Register VPE_CFG_SC22

VPE Functional Description • SC Functional Description: [0] [1] [2] • SC Code: [3]
VPE Register Manual • VPE_SC Register Summary: [4]

Table 10-106. VPE_CFG_SC24

Address Offset	0x0000 0060		
Physical Address	0x489D 0760	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_ORG_W								RESERVED								CFG_ORG_H							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	CFG_ORG_W	This parameter is used by the trimmer. Horizontal offset from the left of the original input image.	RW	0x0
15:11	RESERVED		R	0x0
10:0	CFG_ORG_H	This parameter is used by the trimmer. Vertical offset from the top of the original input image.	RW	0x0

Table 10-107. Register Call Summary for Register VPE_CFG_SC24

VPE Functional Description • SC Functional Description: [0] [1]
VPE Register Manual • VPE_SC Register Summary: [2]

Table 10-108. VPE_CFG_SC25

Address Offset	0x0000 0064		
Physical Address	0x489D 0764	**Instance**	VPE_SC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_OFF_W								RESERVED								CFG_OFF_H							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	CFG_OFF_W	This parameter is used by the trimmer. Horizontal offset from the left of the original input image.	RW	0x0
15:11	RESERVED		R	0x0
10:0	CFG_OFF_H	This parameter is used by the trimmer. Vertical offset from the top of the original input image.	RW	0x0

Table 10-109. Register Call Summary for Register VPE_CFG_SC25

VPE Functional Description • SC Functional Description: [0] [1]
VPE Register Manual • VPE_SC Register Summary: [2]

10.4.4 VPE_CHR_US Registers

10.4.4.1 VPE_CHR_US Register Summary

Table 10-110. VPE_CHR_US Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_CHR_US_INST_0 Base Address	VPE_CHR_US_INST_1 Base Address	VPE_CHR_US_INST_2 Base Address
VPE_PID	RW	32	0x0000 0000	0x489D 0300	0x489D 0400	0x489D 0500
VPE_REG0	RW	32	0x0000 0004	0x489D 0304	0x489D 0404	0x489D 0504
VPE_REG1	RW	32	0x0000 0008	0x489D 0308	0x489D 0408	0x489D 0508
VPE_REG2	RW	32	0x0000 000C	0x489D 030C	0x489D 040C	0x489D 050C
VPE_REG3	RW	32	0x0000 0010	0x489D 0310	0x489D 0410	0x489D 0510
VPE_REG4	RW	32	0x0000 0014	0x489D 0314	0x489D 0414	0x489D 0514
VPE_REG5	RW	32	0x0000 0018	0x489D 0318	0x489D 0418	0x489D 0518
VPE_REG6	RW	32	0x0000 001C	0x489D 031C	0x489D 041C	0x489D 051C
VPE_REG7	RW	32	0x0000 0020	0x489D 0320	0x489D 0420	0x489D 0520

10.4.4.2 VPE_CHR_US Register Description

Table 10-111. VPE_PID

Address Offset	0x0000 0000		
Physical Address	0x489D 0300 0x489D 0400 0x489D 0500	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID																															

Bits	Field Name	Description	Type	Reset
31:0	PID		R	0x0

Table 10-112. Register Call Summary for Register VPE_PID

VPE Register Manual • VPE_CHR_US Register Summary: [0]

Table 10-113. VPE_REG0

Address Offset	0x0000 0004		
Physical Address	0x489D 0304 0x489D 0404 0x489D 0504	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANCHOR_FID0_C0														CFG_MODE		ANCHOR_FID0_C1														RESERVED	

Bits	Field Name	Description	Type	Reset
31:18	ANCHOR_FID0_C0	C0 coefficient for Anchor Pixel. Use when field_id = 0	RW	0x0
17:16	CFG_MODE	0x0 : Mode A 0x1 : Mode B	RW	0x0
15:2	ANCHOR_FID0_C1	C1 coefficient for Anchor Pixel. Use when field_id = 0	RW	0x0
1:0	RESERVED		RW	0x0

Table 10-114. Register Call Summary for Register VPE_REG0

VPE Functional Description

- Modes of Operation (VPDMA): [0]
- Coefficient Configuration: [1] [2] [3] [4] [5] [6] [7]

VPE Register Manual

- VPE_CHR_US Register Summary: [8]

Table 10-115. VPE_REG1

Address Offset	0x0000 0008		
Physical Address	0x489D 0308 0x489D 0408 0x489D 0508	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANCHOR_FID0_C2														RESERVED		ANCHOR_FID0_C3														RESERVED	

Bits	Field Name	Description	Type	Reset
31:18	ANCHOR_FID0_C2	C2 coefficient for Anchor Pixel. Use when field_id = 0	RW	0x0
17:16	RESERVED		RW	0x0
15:2	ANCHOR_FID0_C3	C3 coefficient for Anchor Pixel. Use when field_id = 0	RW	0x0
1:0	RESERVED		RW	0x0

Table 10-116. Register Call Summary for Register VPE_REG1

VPE Functional Description • Coefficient Configuration: [0] [1] [2] [3]
VPE Register Manual • VPE_CHR_US Register Summary: [4]

Table 10-117. VPE_REG2

Address Offset	0x0000 000C		
Physical Address	0x489D 030C 0x489D 040C 0x489D 050C	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERP_FID0_C0														RESERVED		INTERP_FID0_C1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:18	INTERP_FID0_C0	C0 coefficient for Interpolated Pixel. Use when field_id = 0	RW	0x0
17:16	RESERVED		RW	0x0
15:2	INTERP_FID0_C1	C1 coefficient for Interpolated Pixel. Use when field_id = 0	RW	0x0
1:0	RESERVED		RW	0x0

Table 10-118. Register Call Summary for Register VPE_REG2

VPE Functional Description • Coefficient Configuration: [0] [1] [2] [3]
VPE Register Manual • VPE_CHR_US Register Summary: [4]

Table 10-119. VPE_REG3

Address Offset	0x0000 0010		
Physical Address	0x489D 0310 0x489D 0410 0x489D 0510	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERP_FID0_C2														RESERVED		INTERP_FID0_C3								RESERVED							

Bits	Field Name	Description	Type	Reset
31:18	INTERP_FID0_C2	C2 coefficient for Interpolated Pixel. Use when field_id = 0	RW	0x0
17:16	RESERVED		RW	0x0
15:2	INTERP_FID0_C3	C3 coefficient for Interpolated Pixel. Use when field_id = 0	RW	0x0
1:0	RESERVED		RW	0x0

Table 10-120. Register Call Summary for Register VPE_REG3

VPE Functional Description
• Coefficient Configuration: [0] [1] [2] [3]
VPE Register Manual
• VPE_CHR_US Register Summary: [4]

Table 10-121. VPE_REG4

Address Offset	0x0000 0014		
Physical Address	0x489D 0314 0x489D 0414 0x489D 0514	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANCHOR_FID1_C0														RESERVED		ANCHOR_FID1_C1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:18	ANCHOR_FID1_C0	C0 coefficient for Anchor Pixel. Use when field_id = 1	RW	0x0
17:16	RESERVED		R	0x0
15:2	ANCHOR_FID1_C1	C1 coefficient for Anchor Pixel. Use when field_id = 1	RW	0x0
1:0	RESERVED		R	0x0

Table 10-122. Register Call Summary for Register VPE_REG4

VPE Functional Description
• Coefficient Configuration: [0] [1] [2]
VPE Register Manual
• VPE_CHR_US Register Summary: [3]

Table 10-123. VPE_REG5

Address Offset	0x0000 0018		
Physical Address	0x489D 0318 0x489D 0418 0x489D 0518	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANCHOR_FID1_C2														RESERVED		ANCHOR_FID1_C3														RESERVED	

Bits	Field Name	Description	Type	Reset
31:18	ANCHOR_FID1_C2	C2 coefficient for Anchor Pixel. Use when field_id = 1	RW	0x0
17:16	RESERVED		R	0x0
15:2	ANCHOR_FID1_C3	C3 coefficient for Anchor Pixel. Use when field_id = 1	RW	0x0
1:0	RESERVED		R	0x0

Table 10-124. Register Call Summary for Register VPE_REG5

VPE Functional Description
• Coefficient Configuration: [0] [1]
VPE Register Manual
• VPE_CHR_US Register Summary: [2]

Table 10-125. VPE_REG6

Address Offset	0x0000 001C		
Physical Address	0x489D 031C 0x489D 041C 0x489D 051C	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERP_FID1_C0														RESERVED		INTERP_FID1_C1														RESERVED	

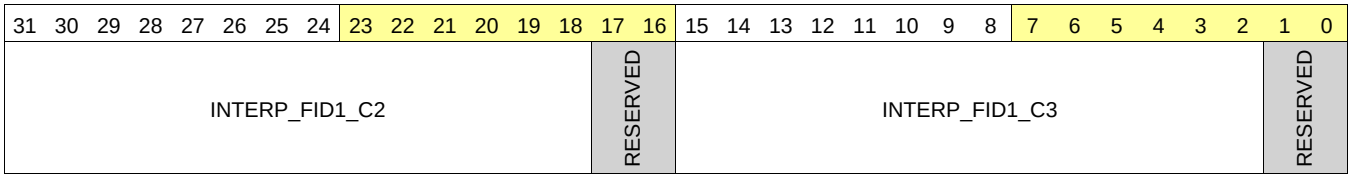
Bits	Field Name	Description	Type	Reset
31:18	INTERP_FID1_C0	C0 coefficient for Interpolated Pixel. Use when field_id = 1	RW	0x0
17:16	RESERVED		R	0x0
15:2	INTERP_FID1_C1	C1 coefficient for Interpolated Pixel. Use when field_id = 1	RW	0x0
1:0	RESERVED		R	0x0

Table 10-126. Register Call Summary for Register VPE_REG6

VPE Functional Description
• Coefficient Configuration: [0] [1]
VPE Register Manual
• VPE_CHR_US Register Summary: [2]

Table 10-127. VPE_REG7

Address Offset	0x0000 0020		
Physical Address	0x489D 0320 0x489D 0420 0x489D 0520	**Instance**	VPE_CHR_US_INST_0 VPE_CHR_US_INST_1 VPE_CHR_US_INST_2
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:18	INTERP_FID1_C2	C2 coefficient for Interpolated Pixel. Use when field_id = 1	RW	0x0
17:16	RESERVED		R	0x0
15:2	INTERP_FID1_C3	C3 coefficient for Interpolated Pixel. Use when field_id = 1	RW	0x0
1:0	RESERVED		R	0x0

Table 10-128. Register Call Summary for Register VPE_REG7

VPE Functional Description
- Coefficient Configuration: [0] [1] [2]

VPE Register Manual
- VPE_CHR_US Register Summary: [3]

10.4.5 VPE_DEI Registers

10.4.5.1 VPE_DEI Register Summary

Table 10-129. VPE_DEI Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_DEI Base Address
VPE_DEI_REG0	RW	32	0x0000 0000	0x489D 0600
VPE_DEI_REG1	RW	32	0x0000 0004	0x489D 0604
VPE_DEI_REG2	RW	32	0x0000 0008	0x489D 0608
VPE_DEI_REG3	RW	32	0x0000 000C	0x489D 060C
VPE_DEI_REG4	RW	32	0x0000 0010	0x489D 0610
VPE_DEI_REG5	RW	32	0x0000 0014	0x489D 0614
VPE_DEI_REG6	RW	32	0x0000 0018	0x489D 0618
VPE_DEI_REG7	RW	32	0x0000 001C	0x489D 061C
VPE_DEI_REG8	RW	32	0x0000 0020	0x489D 0620
VPE_DEI_REG9	RW	32	0x0000 0024	0x489D 0624
VPE_DEI_REG10	RW	32	0x0000 0028	0x489D 0628
VPE_DEI_REG11	RW	32	0x0000 002C	0x489D 062C
VPE_DEI_REG12	R	32	0x0000 0030	0x489D 0630
VPE_DEI_REG13	R	32	0x0000 0034	0x489D 0634
VPE_DEI_REG14	R	32	0x0000 0038	0x489D 0638

10.4.5.2 VPE_DEI Register Description

Table 10-130. VPE_DEI_REG0

Address Offset	0x0000 0000		
Physical Address	0x489D 0600	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROGRESSIVE_BYPASS	FIELD_FLUSH	INTERLACE_BYPASS	RESERVED		HEIGHT											RESERVED					WIDTH										

Bits	Field Name	Description	Type	Reset
31	PROGRESSIVE_BYPASS	Progressive Mode 0x0 : Normal Deinterlace Mode 0x1 : Progressive source	RW	0x0
30	FIELD_FLUSH	Field Flush Mode 0x0 : Normal Operation 0x1 : Flush Internal Pipe for Current output Frame	RW	0x0
29	INTERLACE_BYPASS	Interlace Bypass Mode 0x0 : Normal Deinterlace Mode 0x1 : Pass Interlace Content directly to output	RW	0x0
28:27	RESERVED	Always read as 0	R	0x0
26:16	HEIGHT	Frame Height	RW	0x0
15:11	RESERVED	Always read as 0	R	0x0
10:0	WIDTH	Frame Width	RW	0x0

Table 10-131. Register Call Summary for Register VPE_DEI_REG0

VPE Functional Description • Bypass Mode: [0] [1]
VPE Register Manual • VPE_DEI Register Summary: [2]

Table 10-132. VPE_DEI_REG1

Address Offset	0x0000 0004		
Physical Address	0x489D 0604	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														MDT_SPATMAX_BYPASS	MDT_TEMPMAX_BYPASS

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	MDT_SPATMAX_BYPASS	Spatial Maximum Filtering Bypass for motion values used in EDI 0x0 : Enable 0x1 : Bypass	RW	0x0
0	MDT_TEMPMAX_BYPASS	Spatio-temporal Maximum Filtering Bypass for motion valued used in EDI 0x0 : Enable 0x1 : Bypass	RW	0x0

Table 10-133. Register Call Summary for Register VPE_DEI_REG1

VPE Register Manual
- VPE_DEI Register Summary: [0]

Table 10-134. VPE_DEI_REG2

Address Offset	0x0000 0008		
Physical Address	0x489D 0608	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MDT_MVSTMAX_COR_THR				MDT_MV_COR_THR				MDT_SF_SC_THR3								MDT_SF_SC_THR2								MDT_SF_SC_THR1							

Bits	Field Name	Description	Type	Reset
31:28	MDT_MVSTMAX_COR_THR	This is used for increasing noise robustness. Increasing this threshold leads to more robustness to noise, but with the potential of introducing ghosting effect. Note that this threshold is used for motion values for EDI only, and it is in addition mdt_mv_cor_thr.	RW	0x0

Bits	Field Name	Description	Type	Reset
27:24	MDT_MV_COR_THR	This threshold is for the coring for motion value, mv. MDT will become more noise robust if this value increases. But the picture may be washed out if this value is set to high. This threshold can be interpreted as the noise threshold for calculating motion values for all blocks.	RW	0x0
23:16	MDT_SF_SC_THR3	Spatial frequency threshold 3	RW	0x0
15:8	MDT_SF_SC_THR2	Spatial frequency threshold 2	RW	0x0
7:0	MDT_SF_SC_THR1	Spatial frequency threshold It is used for adaptive scaling of motion values according to how busy the texture is. If the texture is flat, motion values need to be scaled up to reflect the sensitivity of motion values with respect to the detection error. Increasing the thresholds will make the motion value scaling more sensitive to the frequency of the texture. Note: 0 = mdt_sf_sc_thr1 = mdt_sf_sc_thr2 = mdt_sf_sc_thr3	RW	0x0

Table 10-135. Register Call Summary for Register VPE_DEI_REG2

VPE Functional Description
- Bypass Mode: [0] [1] [2]

VPE Register Manual
- VPE_DEI Register Summary: [3]

Table 10-136. VPE_DEI_REG3

Address Offset	0x0000 000C		
Physical Address	0x489D 060C	**Instance**	VPE_DEI
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3	2	1 0
EDI_COR_SCALE_FACTOR	EDI_DIR_COR_LOWER_THR	EDI_CHROMA3D_COR_THR	RESERVED	EDI_CHROMA_3D_ENABLE	EDI_ENABLE_3D	EDI_INP_MODE

Bits	Field Name	Description	Type	Reset
31:24	EDI_COR_SCALE_FACTOR	Scaling factor for correlation along detected edge	RW	0x0
23:16	EDI_DIR_COR_LOWER_THR	Lower threshold used for correlation along detected edge	RW	0x0
15:8	EDI_CHROMA3D_COR_THR	Correlation threshold used in 3D processing for chroma. Because the motion values used for chroma 3D processing are based on luma only. Extra protection is needed. Temporal interpolation is only performed for chroma, when there is strong spatial or temporal correlation for the chroma pixel being processed. When the pixel difference is less than this threshold, it is assumed that there exists strong correlation between these two pixels. Thus, increasing this value leads to more chroma pixels being processed in 3D	RW	0x0
7:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3	EDI_CHROMA_3D_ENABLE	3D Chroma Enable 0x0 : Disable 3D processing for chroma 0x1 : Enable 3D processing (temporal interpolation)	RW	0x0
2	EDI_ENABLE_3D	3D Enable 0x0 : Disable 3D processing 0x1 : Enable 3D processing (temporal interpolation)	RW	0x0
1:0	EDI_INP_MODE	Interpolation mode. Note that mode 00 and 01 are used for debug purpose 0x0 : line average 0x1 : field average 0x2 : edge-directed interpolation for luma only 0x3 : edge-directed interpolation for both luma and chroma	RW	0x0

Table 10-137. Register Call Summary for Register VPE_DEI_REG3

VPE Functional Description • Bypass Mode: [0] [1] [2]
VPE Register Manual • VPE_DEI Register Summary: [3]

Table 10-138. VPE_DEI_REG4

Address Offset	0x0000 0010		
Physical Address	0x489D 0610	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EDI_LUT3					RESERVED			EDI_LUT2					RESERVED			EDI_LUT1					RESERVED			EDI_LUT0			

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	EDI_LUT3	EDI Lookup Table 3	RW	0x0
23:21	RESERVED	Always read as 0	R	0x0
20:16	EDI_LUT2	EDI Lookup Table 2	RW	0x0
15:13	RESERVED	Always read as 0	R	0x0
12:8	EDI_LUT1	EDI Lookup Table 1	RW	0x0
7:5	RESERVED	Always read as 0	R	0x0
4:0	EDI_LUT0	EDI Lookup Table 0	RW	0x0

Table 10-139. Register Call Summary for Register VPE_DEI_REG4

VPE Register Manual • VPE_DEI Register Summary: [0]

Table 10-140. VPE_DEI_REG5

Address Offset	0x0000 0014		
Physical Address	0x489D 0614	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EDI_LUT7					RESERVED			EDI_LUT6					RESERVED			EDI_LUT5					RESERVED			EDI_LUT4			

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	EDI_LUT7	EDI Lookup Table 7	RW	0x0
23:21	RESERVED	Always read as 0	R	0x0
20:16	EDI_LUT6	EDI Lookup Table 6	RW	0x0
15:13	RESERVED	Always read as 0	R	0x0
12:8	EDI_LUT5	EDI Lookup Table 5	RW	0x0
7:5	RESERVED	Always read as 0	R	0x0
4:0	EDI_LUT4	EDI Lookup Table 4	RW	0x0

Table 10-141. Register Call Summary for Register VPE_DEI_REG5

VPE Register Manual

- VPE_DEI Register Summary: [0]

Table 10-142. VPE_DEI_REG6

Address Offset	0x0000 0018		
Physical Address	0x489D 0618	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EDI_LUT11					RESERVED			EDI_LUT10					RESERVED			EDI_LUT9					RESERVED			EDI_LUT8			

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	EDI_LUT11	EDI Lookup Table 11	RW	0x0
23:21	RESERVED	Always read as 0	R	0x0
20:16	EDI_LUT10	EDI Lookup Table 10	RW	0x0
15:13	RESERVED	Always read as 0	R	0x0
12:8	EDI_LUT9	EDI Lookup Table 9	RW	0x0
7:5	RESERVED	Always read as 0	R	0x0
4:0	EDI_LUT8	EDI Lookup Table 8	RW	0x0

Table 10-143. Register Call Summary for Register VPE_DEI_REG6

VPE Register Manual

- VPE_DEI Register Summary: [0]

Table 10-144. VPE_DEI_REG7

Address Offset	0x0000 001C		
Physical Address	0x489D 061C	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EDI_LUT15					RESERVED			EDI_LUT14					RESERVED			EDI_LUT13					RESERVED			EDI_LUT12			

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	EDI_LUT15	EDI Lookup Table 15	RW	0x0
23:21	RESERVED	Always read as 0	R	0x0
20:16	EDI_LUT14	EDI Lookup Table 14	RW	0x0
15:13	RESERVED	Always read as 0	R	0x0
12:8	EDI_LUT13	EDI Lookup Table 13	RW	0x0
7:5	RESERVED	Always read as 0	R	0x0
4:0	EDI_LUT12	EDI Lookup Table 12	RW	0x0

Table 10-145. Register Call Summary for Register VPE_DEI_REG7

VPE Register Manual

- VPE_DEI Register Summary: [0]

Table 10-146. VPE_DEI_REG8

Address Offset	0x0000 0020		
Physical Address	0x489D 0620	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FMD_WINDOW_ENABLE	RESERVED							FMD_WINDOW_MAXX								RESERVED								FMD_WINDOW_MINX							

Bits	Field Name	Description	Type	Reset
31	FMD_WINDOW_ENABLE	Enable FMD operation window	RW	0x0
30:27	RESERVED		R	0x0
26:16	FMD_WINDOW_MAXX	Right boundary of FMD operation window Must be less than width	RW	0x0
15:11	RESERVED		R	0x0
10:0	FMD_WINDOW_MINX	Left boundary of FMD operation window	RW	0x0

Table 10-147. Register Call Summary for Register VPE_DEI_REG8

VPE Functional Description • Bypass Mode: [0] [1]
VPE Register Manual • VPE_DEI Register Summary: [2]

Table 10-148. VPE_DEI_REG9

Address Offset	0x0000 0024		
Physical Address	0x489D 0624	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FMD_WINDOW_MAXY								RESERVED								FMD_WINDOW_MINY							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	FMD_WINDOW_MAXY	Bottom boundary of FMD operation window Must be less than height/2	RW	0x0
15:11	RESERVED		R	0x0
10:0	FMD_WINDOW_MINY	Top boundary of FMD operation window	RW	0x0

Table 10-149. Register Call Summary for Register VPE_DEI_REG9

VPE Functional Description • Bypass Mode: [0] [1]
VPE Register Manual • VPE_DEI Register Summary: [2]

Table 10-150. VPE_DEI_REG10

Address Offset	0x0000 0028		
Physical Address	0x489D 0628	**Instance**	VPE_DEI
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
FMD_CAF_LINE_THR	FMD_CAF_FIELD_THR	RESERVED	FMD_BED_ENABLE, FMD_JAM_DIR, FMD_LOCK, FMD_ENABLE

Bits	Field Name	Description	Type	Reset
31:24	FMD_CAF_LINE_THR	CAF threshold used for the pixels from two lines in one field This is the threshold used for combing artifacts detection. The difference of two consecutive lines from the same field (so there is one line in between if two fields are merged into one progressive frame) is compared with this threshold. Decreasing this threshold leads to be more conservative in detecting CAF. Both fmd_caf_field_thr and fmd_caf_line_thr are close the values that two pixels differed by this value is observable.	RW	0x0
23:16	FMD_CAF_FIELD_THR	CAF threshold used for the pixels from two fields This is the threshold used for combing artifacts detection. The difference of two consecutive lines (when merging two fields into one progressive frame) is used to compare with this threshold. Increasing this threshold leads to be more conservative in detecting CAF.	RW	0x0
15:4	RESERVED		R	0x0
3	FMD_BED_ENABLE	Film Mode Bad Edit Detection 0x0 : Disable 0x1 : Enable	RW	0x0
2	FMD_JAM_DIR	Film Mode Field Jamming Direction 0x0 : Current field jammed with previous field 0x1 : Current field jammed with next field	RW	0x0
1	FMD_LOCK	Film Mode Field Jamming Direction 0x0 : Current field jammed with previous field 0x1 : Current field jammed with next field	RW	0x0
0	FMD_ENABLE	Enable film mode processing 0x0 : Disable 0x1 : Enable	RW	0x0

Table 10-151. Register Call Summary for Register VPE_DEI_REG10

VPE Functional Description • Bypass Mode: [0] [1] [2]
VPE Register Manual • VPE_DEI Register Summary: [3]

Table 10-152. VPE_DEI_REG11

Address Offset	0x0000 002C		
Physical Address	0x489D 062C	**Instance**	VPE_DEI
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED		FMD_CAF_THR	

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	FMD_CAF_THR	CAF threshold used for leaving film mode: If the combing artifacts is greater than this threshold, CAF is detected and thus the state machine will be forced to leave the film mode. If the user prefers to be more conservative in using film mode, decrease this threshold.	RW	0x0

Table 10-153. Register Call Summary for Register VPE_DEI_REG11

VPE Functional Description
- Bypass Mode: [0]

VPE Register Manual
- VPE_DEI Register Summary: [1]

Table 10-154. VPE_DEI_REG12

Address Offset	0x0000 0030		
Physical Address	0x489D 0630	**Instance**	VPE_DEI
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FMD_RESET	RESERVED							FMD_CAF															

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	FMD_RESET	When “1”, the film mode detection module needs to be reset by the software. This bit needs to be checked at each occurrence of the film mode detection interrupt	R	0x0
23:21	RESERVED		R	0x0
20:0	FMD_CAF	Detected combing artifacts	R	0x0

Table 10-155. Register Call Summary for Register VPE_DEI_REG12

VPE Functional Description
- Bypass Mode: [0] [1]

VPE Register Manual
- VPE_DEI Register Summary: [2]

Table 10-156. VPE_DEI_REG13

Address Offset	0x0000 0034		
Physical Address	0x489D 0634	**Instance**	VPE_DEI
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FMD_FIELD_DIFF																							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	FMD_FIELD_DIFF	Field difference (difference between two neighboring fields, one top and one bottom)	R	0x0

Table 10-157. Register Call Summary for Register VPE_DEI_REG13

VPE Functional Description
- Bypass Mode: [0]

VPE Register Manual
- VPE_DEI Register Summary: [1]

Table 10-158. VPE_DEI_REG14

Address Offset	0x0000 0038		
Physical Address	0x489D 0638	**Instance**	VPE_DEI
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FMD_FRAME_DIFF																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	FMD_FRAME_DIFF	Frame difference (difference between two top or two bottom fields)	R	0x0

Table 10-159. Register Call Summary for Register VPE_DEI_REG14

VPE Functional Description
- Bypass Mode: [0]

VPE Register Manual
- VPE_DEI Register Summary: [1]

10.4.6 VPE_VPDMA Registers

NOTE: The functionality of the following sets of registers is not supported by VPE VPDMA in this family of devices:

- All VPE_INT1_* registers
- All VPE_INT2_* registers
- All VPE_INT3_* registers

The following channels are not used by VPE VPDMA in this family of devices. All register bit-fields corresponding to these channels should be kept at their reset value.

- VIP1_*, except for the following:
  - VIP1_PORTA_RGB
  - VIP1_PORTA_LUMA
  - VIP1_PORTA_CHROMA
- VIP2_*
- GRPX_*
- SCALER_OUT
- SCALER_LUMA
- SCALER_CHROMA
- NF_*
- TRANSCODE1_*
- TRANSCODE2_*
- AUX_IN
- PIP_FRAME
- POST_COMP_WR
- VBI_SD_VENC

NOTE: The following clients are not used by VPE VPDMA in this family of devices. All register bit-fields corresponding to these clients should be kept at their reset value.

- VIP1_*, except for the following:
  - VIP1_UP_UV
  - VIP1_UP_Y
- VIP2_*
- TRANS1_LUMA
- TRANS1_CHROMA
- TRANS2_LUMA
- TRANS2_CHROMA
- HDMI_WRBK
- VBI_SDVENC
- NF_420_UV_OUT
- NF_420_Y_OUT
- NF_420_UV_IN
- NF_420_Y_IN
- NF_422_IN
- GRPX1_ST
- GRPX2_ST
- GRPX3_ST
- GRPX1_DATA
- GRPX2_DATA
- GRPX3_DATA
- PIP_WRBK
- SC_IN_*
- SC_OUT
- COMP_WRBK

10.4.6.1 VPE_VPDMA Register Summary

Table 10-160. VPE_VPDMA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_VPDMA Base Address
VPE_VPDMA_PID	R	32	0x0000 0000	0x489D D000
VPE_LIST_ADDR	RW	32	0x0000 0004	0x489D D004
VPE_LIST_ATTR	RW	32	0x0000 0008	0x489D D008
VPE_LIST_STAT_SYNC	RW	32	0x0000 000C	0x489D D00C
VPE_BG_RGB	RW	32	0x0000 0018	0x489D D018
VPE_BG_YUV	RW	32	0x0000 001C	0x489D D01C
VPE_VPDMA_SETUP	RW	32	0x0000 0030	0x489D D030
VPE_MAX_SIZE1	RW	32	0x0000 0034	0x489D D034
VPE_MAX_SIZE2	RW	32	0x0000 0038	0x489D D038
VPE_MAX_SIZE3	RW	32	0x0000 003C	0x489D D03C
VPE_INT0_CHANNEL0_INT_STAT	RW	32	0x0000 0040	0x489D D040
VPE_INT0_CHANNEL0_INT_MASK	RW	32	0x0000 0044	0x489D D044
VPE_INT0_CHANNEL1_INT_STAT	RW	32	0x0000 0048	0x489D D048

Table 10-160. VPE_VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VPE_VPDMA Base Address
VPE_INT0_CHANNEL1_INT_MASK	RW	32	0x0000 004C	0x489D D04C
VPE_INT0_CHANNEL2_INT_STAT	RW	32	0x0000 0050	0x489D D050
VPE_INT0_CHANNEL2_INT_MASK	RW	32	0x0000 0054	0x489D D054
VPE_INT0_CHANNEL3_INT_STAT	RW	32	0x0000 0058	0x489D D058
VPE_INT0_CHANNEL3_INT_MASK	RW	32	0x0000 005C	0x489D D05C
VPE_INT0_CHANNEL4_INT_STAT	RW	32	0x0000 0060	0x489D D060
VPE_INT0_CHANNEL4_INT_MASK	RW	32	0x0000 0064	0x489D D064
VPE_INT0_CHANNEL5_INT_STAT	RW	32	0x0000 0068	0x489D D068
VPE_INT0_CHANNEL5_INT_MASK	RW	32	0x0000 006C	0x489D D06C
VPE_INT0_CLIENT0_INT_STAT	RW	32	0x0000 0078	0x489D D078
VPE_INT0_CLIENT0_INT_MASK	RW	32	0x0000 007C	0x489D D07C
VPE_INT0_CLIENT1_INT_STAT	RW	32	0x0000 0080	0x489D D080
VPE_INT0_CLIENT1_INT_MASK	RW	32	0x0000 0084	0x489D D084
VPE_INT0_LIST0_INT_STAT	RW	32	0x0000 0088	0x489D D088
VPE_INT0_LIST0_INT_MASK	RW	32	0x0000 008C	0x489D D08C
VPE_INT1_CHANNEL0_INT_STAT	RW	32	0x0000 0090	0x489D D090
VPE_INT1_CHANNEL0_INT_MASK	RW	32	0x0000 0094	0x489D D094
VPE_INT1_CHANNEL1_INT_STAT	RW	32	0x0000 0098	0x489D D098
VPE_INT1_CHANNEL1_INT_MASK	RW	32	0x0000 009C	0x489D D09C
VPE_INT1_CHANNEL2_INT_STAT	RW	32	0x0000 00A0	0x489D D0A0
VPE_INT1_CHANNEL2_INT_MASK	RW	32	0x0000 00A4	0x489D D0A4
VPE_INT1_CHANNEL3_INT_STAT	RW	32	0x0000 00A8	0x489D D0A8
VPE_INT1_CHANNEL3_INT_MASK	RW	32	0x0000 00AC	0x489D D0AC
VPE_INT1_CHANNEL4_INT_STAT	RW	32	0x0000 00B0	0x489D D0B0
VPE_INT1_CHANNEL4_INT_MASK	RW	32	0x0000 00B4	0x489D D0B4
VPE_INT1_CHANNEL5_INT_STAT	RW	32	0x0000 00B8	0x489D D0B8
VPE_INT1_CHANNEL5_INT_MASK	RW	32	0x0000 00BC	0x489D D0BC
VPE_INT1_CLIENT0_INT_STAT	RW	32	0x0000 00C8	0x489D D0C8
VPE_INT1_CLIENT0_INT_MASK	RW	32	0x0000 00CC	0x489D D0CC
VPE_INT1_LIST0_INT_STAT	RW	32	0x0000 00D8	0x489D D0D8
VPE_INT1_LIST0_INT_MASK	RW	32	0x0000 00DC	0x489D D0DC
VPE_INT2_CHANNEL0_INT_STAT	RW	32	0x0000 00E0	0x489D D0E0
VPE_INT2_CHANNEL0_INT_MASK	RW	32	0x0000 00E4	0x489D D0E4
VPE_INT2_CHANNEL1_INT_STAT	RW	32	0x0000 00E8	0x489D D0E8
VPE_INT2_CHANNEL1_INT_MASK	RW	32	0x0000 00EC	0x489D D0EC
VPE_INT2_CHANNEL2_INT_STAT	RW	32	0x0000 00F0	0x489D D0F0
VPE_INT2_CHANNEL2_INT_MASK	RW	32	0x0000 00F4	0x489D D0F4
VPE_INT2_CHANNEL3_INT_STAT	RW	32	0x0000 00F8	0x489D D0F8
VPE_INT2_CHANNEL3_INT_MASK	RW	32	0x0000 00FC	0x489D D0FC
VPE_INT2_CHANNEL4_INT_STAT	RW	32	0x0000 0100	0x489D D100
VPE_INT2_CHANNEL4_INT_MASK	RW	32	0x0000 0104	0x489D D104
VPE_INT2_CHANNEL5_INT_STAT	RW	32	0x0000 0108	0x489D D108
VPE_INT2_CHANNEL5_INT_MASK	RW	32	0x0000 010C	0x489D D10C
VPE_INT2_CLIENT0_INT_STAT	RW	32	0x0000 0118	0x489D D118
VPE_INT2_CLIENT0_INT_MASK	RW	32	0x0000 011C	0x489D D11C
VPE_INT2_LIST0_INT_STAT	RW	32	0x0000 0128	0x489D D128

Table 10-160. VPE_VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VPE_VPDMA Base Address
VPE_INT2_LIST0_INT_MASK	RW	32	0x0000 012C	0x489D D12C
VPE_INT3_CHANNEL0_INT_STAT	RW	32	0x0000 0130	0x489D D130
VPE_INT3_CHANNEL0_INT_MASK	RW	32	0x0000 0134	0x489D D134
VPE_INT3_CHANNEL1_INT_STAT	RW	32	0x0000 0138	0x489D D138
VPE_INT3_CHANNEL1_INT_MASK	RW	32	0x0000 013C	0x489D D13C
VPE_INT3_CHANNEL2_INT_STAT	RW	32	0x0000 0140	0x489D D140
VPE_INT3_CHANNEL2_INT_MASK	RW	32	0x0000 0144	0x489D D144
VPE_INT3_CHANNEL3_INT_STAT	RW	32	0x0000 0148	0x489D D148
VPE_INT3_CHANNEL3_INT_MASK	RW	32	0x0000 014C	0x489D D14C
VPE_INT3_CHANNEL4_INT_STAT	RW	32	0x0000 0150	0x489D D150
VPE_INT3_CHANNEL4_INT_MASK	RW	32	0x0000 0154	0x489D D154
VPE_INT3_CHANNEL5_INT_STAT	RW	32	0x0000 0158	0x489D D158
VPE_INT3_CHANNEL5_INT_MASK	RW	32	0x0000 015C	0x489D D15C
VPE_INT3_CLIENT0_INT_STAT	RW	32	0x0000 0168	0x489D D168
VPE_INT3_CLIENT0_INT_MASK	RW	32	0x0000 016C	0x489D D16C
VPE_INT3_LIST0_INT_STAT	RW	32	0x0000 0178	0x489D D178
VPE_INT3_LIST0_INT_MASK	RW	32	0x0000 017C	0x489D D17C
VPE_PERF_MON0	RW	32	0x0000 0200	0x489D D200
VPE_PERF_MON1	RW	32	0x0000 0204	0x489D D204
VPE_PERF_MON2	RW	32	0x0000 0208	0x489D D208
VPE_PERF_MON3	RW	32	0x0000 020C	0x489D D20C
VPE_PERF_MON4	RW	32	0x0000 0210	0x489D D210
VPE_PERF_MON5	RW	32	0x0000 0214	0x489D D214
VPE_PERF_MON6	RW	32	0x0000 0218	0x489D D218
VPE_PERF_MON7	RW	32	0x0000 021C	0x489D D21C
VPE_PERF_MON8	RW	32	0x0000 0220	0x489D D220
VPE_PERF_MON9	RW	32	0x0000 0224	0x489D D224
VPE_PERF_MON10	RW	32	0x0000 0228	0x489D D228
VPE_PERF_MON11	RW	32	0x0000 022C	0x489D D22C
VPE_PERF_MON12	RW	32	0x0000 0230	0x489D D230
VPE_PERF_MON13	RW	32	0x0000 0234	0x489D D234
VPE_PERF_MON14	RW	32	0x0000 0238	0x489D D238
VPE_PERF_MON15	RW	32	0x0000 023C	0x489D D23C
VPE_PERF_MON16	RW	32	0x0000 0240	0x489D D240
VPE_PERF_MON17	RW	32	0x0000 0244	0x489D D244
VPE_PERF_MON18	RW	32	0x0000 0248	0x489D D248
VPE_PERF_MON19	RW	32	0x0000 024C	0x489D D24C
VPE_PERF_MON20	RW	32	0x0000 0250	0x489D D250
VPE_PERF_MON21	RW	32	0x0000 0254	0x489D D254
VPE_PERF_MON22	RW	32	0x0000 0258	0x489D D258
VPE_PERF_MON23	RW	32	0x0000 025C	0x489D D25C
VPE_PERF_MON24	RW	32	0x0000 0260	0x489D D260
VPE_PERF_MON25	RW	32	0x0000 0264	0x489D D264
VPE_PERF_MON26	RW	32	0x0000 0268	0x489D D268
VPE_PERF_MON27	RW	32	0x0000 026C	0x489D D26C
VPE_PERF_MON28	RW	32	0x0000 0270	0x489D D270

Table 10-160. VPE_VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VPE_VPDMA Base Address
VPE_PERF_MON29	RW	32	0x0000 0274	0x489D D274
VPE_PERF_MON30	RW	32	0x0000 0278	0x489D D278
VPE_PERF_MON31	RW	32	0x0000 027C	0x489D D27C
VPE_PERF_MON32	RW	32	0x0000 0280	0x489D D280
VPE_PERF_MON33	RW	32	0x0000 0284	0x489D D284
VPE_PERF_MON34	RW	32	0x0000 0288	0x489D D288
VPE_PERF_MON35	RW	32	0x0000 028C	0x489D D28C
VPE_PERF_MON36	RW	32	0x0000 0290	0x489D D290
VPE_PERF_MON37	RW	32	0x0000 0294	0x489D D294
VPE_PERF_MON38	RW	32	0x0000 0298	0x489D D298
VPE_PERF_MON39	RW	32	0x0000 029C	0x489D D29C
VPE_PERF_MON40	RW	32	0x0000 02A0	0x489D D2A0
VPE_PERF_MON41	RW	32	0x0000 02A4	0x489D D2A4
VPE_PERF_MON42	RW	32	0x0000 02A8	0x489D D2A8
VPE_PERF_MON43	RW	32	0x0000 02AC	0x489D D2AC
VPE_PERF_MON44	RW	32	0x0000 02B0	0x489D D2B0
VPE_PERF_MON45	RW	32	0x0000 02B4	0x489D D2B4
VPE_PERF_MON46	RW	32	0x0000 02B8	0x489D D2B8
VPE_PERF_MON47	RW	32	0x0000 02BC	0x489D D2BC
VPE_PERF_MON48	RW	32	0x0000 02C0	0x489D D2C0
VPE_PERF_MON49	RW	32	0x0000 02C4	0x489D D2C4
VPE_PERF_MON50	RW	32	0x0000 02C8	0x489D D2C8
VPE_PERF_MON51	RW	32	0x0000 02CC	0x489D D2CC
VPE_PERF_MON52	RW	32	0x0000 02D0	0x489D D2D0
VPE_PRI_CHROMA_CSTAT	RW	32	0x0000 0300	0x489D D300
VPE_PRI_LUMA_CSTAT	RW	32	0x0000 0304	0x489D D304
VPE_PRI_FLD1_LUMA_CSTAT	RW	32	0x0000 0308	0x489D D308
VPE_PRI_FLD1_CHROMA_CSTAT	RW	32	0x0000 030C	0x489D D30C
VPE_PRI_FLD2_LUMA_CSTAT	RW	32	0x0000 0310	0x489D D310
VPE_PRI_FLD2_CHROMA_CSTAT	RW	32	0x0000 0314	0x489D D314
VPE_PRI_MV0_CSTAT	RW	32	0x0000 0330	0x489D D330
VPE_PRI_MV_OUT_CSTAT	RW	32	0x0000 033C	0x489D D33C
VPE_VIP0_UP_Y_CSTAT	RW	32	0x0000 0390	0x489D D390
VPE_VIP0_UP_UV_CSTAT	RW	32	0x0000 0394	0x489D D394
VPE_VPI_CTL_CSTAT	RW	32	0x0000 03D0	0x489D D3D0

10.4.6.2 VPE_VPDMA Register Description

Table 10-161. VPE_VPDMA_PID

Address Offset	0x0000 0000		
Physical Address	0x489D D000	**Instance**	VPE_VPDMA
Description	This register follows the format described in PDR3.5		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME								FUNC								RTL					MAJOR			VPDMA_LOAD_COMPLETE	VPDMA_ACCESS_TYPE	MINOR							

Bits	Field Name	Description	Type	Reset
31:30	SCHEME	The scheme of the register used. Currently this is PDR 3.5 Scheme	R	0x0
29:16	FUNC	The funcition of the module being used. The value is for vpe0_vayu_vpdma.	R	0x0
15:11	RTL	RTL Release Version The PDR release number of this IP. After Bootup this value becomes the firmware Revision ID	R	0x0
10:8	MAJOR	Major Release Number	R	0x0
7	VPDMA_LOAD_COMPLETE	This bit will be 1 when the VPDMA state machines image and data image have successfuly been fetched and loaded.	R	0x0
6	VPDMA_ACCESS_TYPE	After bootup this bit states how DMA transaction are setup by lists or through register access. 0x0 : Lists 0x1 : Register Access	R	0x0
5:0	MINOR	Minor Release Number	R	0x0

Table 10-162. Register Call Summary for Register VPE_VPDMA_PID

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-163. VPE_LIST_ADDR

Address Offset	0x0000 0004		
Physical Address	0x489D D004	**Instance**	VPE_VPDMA
Description	The location of a new list to begin processing.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LIST_ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	LIST_ADDR	Location of a new list of descriptors. This register must be written with the VPDMA Configuration Location after reset.	RW	0x0

Table 10-164. Register Call Summary for Register VPE_LIST_ADDR

VPE Functional Description
- VPDMA Introduction:
- VPDMA Basic Definitions: [1] [2]
- VPDMA Configuration: [3] [4] [5]

Table 10-164. Register Call Summary for Register VPE_LIST_ADDR (continued)

VPE Register Manual

- VPE_VPDMA Register Summary: [6]
- VPE_VPDMA Register Description: [7]

Table 10-165. VPE_LIST_ATTR

Address Offset	0x0000 0008		
Physical Address	0x489D D008	**Instance**	VPE_VPDMA
Description	The attributes of a new list. This register should always be written after list_addr.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					LIST_NUM			RESERVED			STOP	RDY	LIST_TYPE			LIST_SIZE																

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:24	LIST_NUM	The list number that should be assigned to the list located at LIST_ADDR. If the list is still active this will block all future list writes until the list is available.	RW	0x0
23:21	RESERVED		R	0x0
20	STOP	This bit is written with the LIST_NUMBER field to stop a self-modifying list. When this bit is written a one the list specified by the LIST_NUMBER is sent a stop signal and will finish the current frame of transfers and then free the list resources.	RW	0x0
19	RDY	This bit is low when a new list cannot be written to the VPE_LIST_ADDR register. The reasons this bit would be low are at initial startup if the LIST_MANAGER State Machine image has not completed loading. It also would be low if the last write to the LIST_ATTR attempted to start a list that is currently active. When this bit is low any writes to the list address register will cause access to not be accepted until this bit has set by the previous list having completed.	R	0x0
18:16	LIST_TYPE	The type of list that has been generated. 0x0 : Normal List 0x1 : Self-Modifying List 0x2 : List Doorbell Others Reserved for future use	RW	0x0
15:0	LIST_SIZE	Number of 128 bit word in the new list of descriptors. Writes to this register will activate the list in the list stack of the list manager and begin transfer of the list into VPDMA. This size can not be 0.	RW	0x0

Table 10-166. Register Call Summary for Register VPE_LIST_ATTR

VPE Functional Description

- VPDMA Configuration: [0] [1] [2]

VPE Register Manual

- VPE_VPDMA Register Summary: [3]

Table 10-167. VPE_LIST_STAT_SYNC

Address Offset	0x0000 000C		
Physical Address	0x489D D00C	**Instance**	VPE_VPDMA
Description	The register is used for processor to List Manager syncronization and status registers for the list.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LIST7_BUSY	LIST6_BUSY	LIST5_BUSY	LIST4_BUSY	LIST3_BUSY	LIST2_BUSY	LIST1_BUSY	LIST0_BUSY	RESERVED								SYNC_LISTS7	SYNC_LISTS6	SYNC_LISTS5	SYNC_LISTS4	SYNC_LISTS3	SYNC_LISTS2	SYNC_LISTS1	SYNC_LISTS0

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	LIST7_BUSY	The list 7 is currently running. Any attempt to load a new list to list 7 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
22	LIST6_BUSY	The list 6 is currently running. Any attempt to load a new list to list 6 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
21	LIST5_BUSY	The list 5 is currently running. Any attempt to load a new list to list 5 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
20	LIST4_BUSY	The list 4 is currently running. Any attempt to load a new list to list 4 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
19	LIST3_BUSY	The list 3 is currently running. Any attempt to load a new list to list 3 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
18	LIST2_BUSY	The list 2 is currently running. Any attempt to load a new list to list 2 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
17	LIST1_BUSY	The list 1 is currently running. Any attempt to load a new list to list 1 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
16	LIST0_BUSY	The list 0 is currently running. Any attempt to load a new list to list 0 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
15:8	RESERVED		R	0x0
7	SYNC_LISTS7	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 7 waiting on it.	RW	0x0
6	SYNC_LISTS6	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 6 waiting on it.	RW	0x0
5	SYNC_LISTS5	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 5 waiting on it.	RW	0x0
4	SYNC_LISTS4	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 4 waiting on it.	RW	0x0
3	SYNC_LISTS3	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 3 waiting on it.	RW	0x0

Bits	Field Name	Description	Type	Reset
2	SYNC_LISTS2	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 2 waiting on it.	RW	0x0
1	SYNC_LISTS1	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 1 waiting on it.	RW	0x0
0	SYNC_LISTS0	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 0 waiting on it.	RW	0x0

Table 10-168. Register Call Summary for Register VPE_LIST_STAT_SYNC

VPE Functional Description
- VPDMA Descriptors: [0] [1] [2]

VPE Register Manual
- VPE_VPDMA Register Summary: [3]

Table 10-169. VPE_BG_RGB

Address Offset	0x0000 0018		
Physical Address	0x489D D018	**Instance**	VPE_VPDMA
Description	The registers used to set the background color for RGB		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RED								GREEN								BLUE								BLEND							

Bits	Field Name	Description	Type	Reset
31:24	RED	The red value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
23:16	GREEN	The green value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
15:8	BLUE	The blue value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
7:0	BLEND	The blend value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0

Table 10-170. Register Call Summary for Register VPE_BG_RGB

VPE Functional Description
- VPDMA Configuration:
- VPDMA Data Formats: [3] [4]

VPE Register Manual
- VPE_VPDMA Register Summary: [5]

Table 10-171. VPE_BG_YUV

Address Offset	0x0000 001C		
Physical Address	0x489D D01C	**Instance**	VPE_VPDMA
Description	The registers used to set the background color for YUV		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								Y								CR								CB							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	Y	The Y value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0
15:8	CR	The Cr value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0
7:0	CB	The Cb value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0

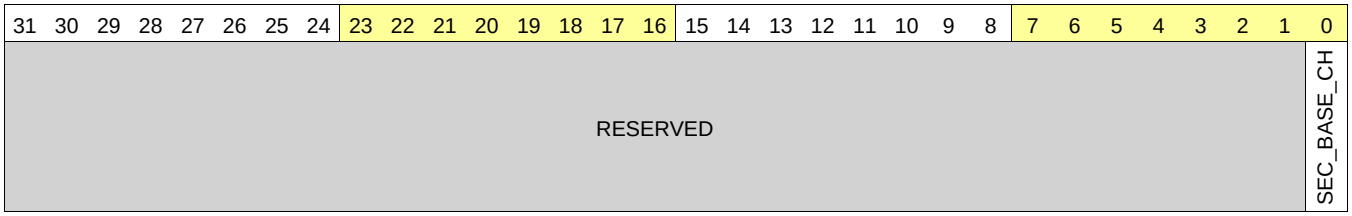
Table 10-172. Register Call Summary for Register VPE_BG_YUV

VPE Functional Description
- VPDMA Configuration:

VPE Register Manual
- VPE_VPDMA Register Summary: [5]

Table 10-173. VPE_VPDMA_SETUP

Address Offset	0x0000 0030		
Physical Address	0x489D D030	**Instance**	VPE_VPDMA
Description	Configures global parameters that are shared by all clients.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SEC_BASE_CH	Use Secondary Channels for Mosaic mode	RW	0x0

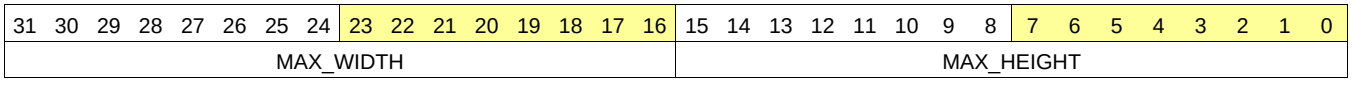
Table 10-174. Register Call Summary for Register VPE_VPDMA_SETUP

VPE Functional Description
- VPDMA Configuration:

VPE Register Manual
- VPE_VPDMA Register Summary: [2]

Table 10-175. VPE_MAX_SIZE1

Address Offset	0x0000 0034		
Physical Address	0x489D D034	**Instance**	VPE_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 1 in write descriptor.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 1 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 1 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 10-176. Register Call Summary for Register VPE_MAX_SIZE1

VPE Functional Description
- VPDMA Descriptors: [0] [1] [2] [3] [4]

VPE Register Manual
- VPE_VPDMA Register Summary: [5]

Table 10-177. VPE_MAX_SIZE2

Address Offset	0x0000 0038		
Physical Address	0x489D D038	**Instance**	VPE_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 2 in write descriptor.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_WIDTH																MAX_HEIGHT															

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 2 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 2 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 10-178. Register Call Summary for Register VPE_MAX_SIZE2

VPE Functional Description
- VPDMA Descriptors: [0] [1] [2] [3] [4]

VPE Register Manual
- VPE_VPDMA Register Summary: [5]

Table 10-179. VPE_MAX_SIZE3

Address Offset	0x0000 003C		
Physical Address	0x489D D03C	**Instance**	VPE_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 3 in write descriptor.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_WIDTH																MAX_HEIGHT															

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 3 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 3 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 10-180. Register Call Summary for Register VPE_MAX_SIZE3

VPE Functional Description
- VPDMA Descriptors: [0] [1] [2] [3] [4]

VPE Register Manual
- VPE_VPDMA Register Summary: [5]

Table 10-181. VPE_INT0_CHANNEL0_INT_STAT

Address Offset	0x0000 0040		
Physical Address	0x489D D040	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_GRPX3	INT_STAT_GRPX2	INT_STAT_GRPX1	INT_STAT_SCALER_OUT	RESERVED								INT_STAT_SCALER_CHROMA	INT_STAT_SCALER_LUMA	INT_STAT_HQ_SCALER	RESERVED	INT_STAT_HQ_MV_OUT	RESERVED		INT_STAT_HQ_MV	RESERVED				INT_STAT_HQ_VID3_CHROMA	INT_STAT_HQ_VID3_LUMA	INT_STAT_HQ_VID2_CHROMA	INT_STAT_HQ_VID2_LUMA	INT_STAT_HQ_VID1_CHROMA	INT_STAT_HQ_VID1_LUMA		

Bits	Field Name	Description	Type	Reset
31	INT_STAT_GRPX3	The last read DMA transaction has occurred for channel grpx3 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_GRPX2	The last read DMA transaction has occurred for channel grpx2 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_GRPX1	The last read DMA transaction has occurred for channel grpx1 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
28	INT_STAT_SCALER_OUT	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19	INT_STAT_SCALER_CHROMA	The last write DMA transaction has completed for channel scaler_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_SCALER_LUMA	The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_HQ_SCALER	The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_STAT_HQ_MV_OUT	The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_STAT_HQ_MV	The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_HQ_VID3_CHROMA	The last write DMA transaction has completed for channel hq_vid3_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_HQ_VID3_LUMA	The last write DMA transaction has completed for channel hq_vid3_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	INT_STAT_HQ_VID2_CHROMA	The last write DMA transaction has completed for channel hq_vid2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_HQ_VID2_LUMA	The last write DMA transaction has completed for channel hq_vid2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_HQ_VID1_CHROMA	The last write DMA transaction has completed for channel hq_vid1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_HQ_VID1_LUMA	The last write DMA transaction has completed for channel hq_vid1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-182. Register Call Summary for Register VPE_INT0_CHANNEL0_INT_STAT

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-183. VPE_INT0_CHANNEL0_INT_MASK

Address Offset	0x0000 0044		
Physical Address	0x489D D044	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX3	INT_MASK_GRPX2	INT_MASK_GRPX1	INT_MASK_SCALER_OUT	RESERVED								INT_MASK_SCALER_CHROMA	INT_MASK_SCALER_LUMA	INT_MASK_HQ_SCALER	RESERVED	INT_MASK_HQ_MV_OUT	RESERVED		INT_MASK_HQ_MV	RESERVED				INT_MASK_HQ_VID3_CHROMA	INT_MASK_HQ_VID3_LUMA	INT_MASK_HQ_VID2_CHROMA	INT_MASK_HQ_VID2_LUMA	INT_MASK_HQ_VID1_CHROMA	INT_MASK_HQ_VID1_LUMA	

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX3	The interrupt for Graphcis 2 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_GRPX2	The interrupt for Graphics 1 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_GRPX1	The interrupt for Graphics 0 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_SCALER_OUT	The interrupt for Low Cost DEI Scalar Write to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19	INT_MASK_SCALER_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_SCALER_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_HQ_SCALER	The interrupt for High Quality DEI Scaler Write to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_HQ_MV_OUT	The interrupt for Low Cost DEI Motion Vector Write should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_MASK_HQ_MV	The interrupt for Low Cost DEI Motion Vector should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_HQ_VID3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_HQ_VID3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_HQ_VID2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_HQ_VID2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_HQ_VID1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_HQ_VID1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-184. Register Call Summary for Register VPE_INT0_CHANNEL0_INT_MASK

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-185. VPE_INT0_CHANNEL1_INT_STAT

Address Offset	0x0000 0048		
Physical Address	0x489D D048	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP1_MULT_PORTB_SRC9	INT_STAT_VIP1_MULT_PORTB_SRC8	INT_STAT_VIP1_MULT_PORTB_SRC7	INT_STAT_VIP1_MULT_PORTB_SRC6	INT_STAT_VIP1_MULT_PORTB_SRC5	INT_STAT_VIP1_MULT_PORTB_SRC4	INT_STAT_VIP1_MULT_PORTB_SRC3	INT_STAT_VIP1_MULT_PORTB_SRC2	INT_STAT_VIP1_MULT_PORTB_SRC1	INT_STAT_VIP1_MULT_PORTB_SRC0	INT_STAT_VIP1_MULT_PORTA_SRC15	INT_STAT_VIP1_MULT_PORTA_SRC14	INT_STAT_VIP1_MULT_PORTA_SRC13	INT_STAT_VIP1_MULT_PORTA_SRC12	INT_STAT_VIP1_MULT_PORTA_SRC11	INT_STAT_VIP1_MULT_PORTA_SRC10	INT_STAT_VIP1_MULT_PORTA_SRC9	INT_STAT_VIP1_MULT_PORTA_SRC8	INT_STAT_VIP1_MULT_PORTA_SRC7	INT_STAT_VIP1_MULT_PORTA_SRC6	INT_STAT_VIP1_MULT_PORTA_SRC5	INT_STAT_VIP1_MULT_PORTA_SRC4	INT_STAT_VIP1_MULT_PORTA_SRC3	INT_STAT_VIP1_MULT_PORTA_SRC2	INT_STAT_VIP1_MULT_PORTA_SRC1	INT_STAT_VIP1_MULT_PORTA_SRC0	RESERVED						

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
27	INT_STAT_VIP1_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP1_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP1_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP1_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_VIP1_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP1_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP1_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP1_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
11	INT_STAT_VIP1_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 10-186. Register Call Summary for Register VPE_INT0_CHANNEL1_INT_STAT

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-187. VPE_INT0_CHANNEL1_INT_MASK

Address Offset	0x0000 004C		
Physical Address	0x489D D04C	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP1_MULT_PORTB_SRC9	INT_MASK_VIP1_MULT_PORTB_SRC8	INT_MASK_VIP1_MULT_PORTB_SRC7	INT_MASK_VIP1_MULT_PORTB_SRC6	INT_MASK_VIP1_MULT_PORTB_SRC5	INT_MASK_VIP1_MULT_PORTB_SRC4	INT_MASK_VIP1_MULT_PORTB_SRC3	INT_MASK_VIP1_MULT_PORTB_SRC2	INT_MASK_VIP1_MULT_PORTB_SRC1	INT_MASK_VIP1_MULT_PORTB_SRC0	INT_MASK_VIP1_MULT_PORTA_SRC15	INT_MASK_VIP1_MULT_PORTA_SRC14	INT_MASK_VIP1_MULT_PORTA_SRC13	INT_MASK_VIP1_MULT_PORTA_SRC12	INT_MASK_VIP1_MULT_PORTA_SRC11	INT_MASK_VIP1_MULT_PORTA_SRC10	INT_MASK_VIP1_MULT_PORTA_SRC9	INT_MASK_VIP1_MULT_PORTA_SRC8	INT_MASK_VIP1_MULT_PORTA_SRC7	INT_MASK_VIP1_MULT_PORTA_SRC6	INT_MASK_VIP1_MULT_PORTA_SRC5	INT_MASK_VIP1_MULT_PORTA_SRC4	INT_MASK_VIP1_MULT_PORTA_SRC3	INT_MASK_VIP1_MULT_PORTA_SRC2	INT_MASK_VIP1_MULT_PORTA_SRC1	INT_MASK_VIP1_MULT_PORTA_SRC0	RESERVED					

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_PORTB_SRC9	The interrupt for Video Input 1 Port B Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_PORTB_SRC8	The interrupt for Video Input 1 Port B Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_PORTB_SRC7	The interrupt for Video Input 1 Port B Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_PORTB_SRC6	The interrupt for Video Input 1 Port B Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_PORTB_SRC5	The interrupt for Video Input 1 Port B Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_PORTB_SRC4	The interrupt for Video Input 1 Port B Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_PORTB_SRC3	The interrupt for Video Input 1 Port B Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_PORTB_SRC2	The interrupt for Video Input 1 Port B Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_PORTB_SRC1	The interrupt for Video Input 1 Port B Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP1_MULT_PORTB_SRC0	The interrupt for Video Input 1 Port B Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_PORTA_SRC15	The interrupt for Video Input 1 Port A Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_MASK_VIP1_MULT_PORTA_SRC14	The interrupt for Video Input 1 Port A Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_PORTA_SRC13	The interrupt for Video Input 1 Port A Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_PORTA_SRC12	The interrupt for Video Input 1 Port A Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_PORTA_SRC11	The interrupt for Video Input 1 Port A Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_PORTA_SRC10	The interrupt for Video Input 1 Port A Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_PORTA_SRC9	The interrupt for Video Input 1 Port A Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_PORTA_SRC8	The interrupt for Video Input 1 Port A Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_PORTA_SRC7	The interrupt for Video Input 1 Port A Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_PORTA_SRC6	The interrupt for Video Input 1 Port A Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_MULT_PORTA_SRC5	The interrupt for Video Input 1 Port A Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_PORTA_SRC4	The interrupt for Video Input 1 Port A Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_PORTA_SRC3	The interrupt for Video Input 1 Port A Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_PORTA_SRC2	The interrupt for Video Input 1 Port A Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_MULT_PORTA_SRC1	The interrupt for Video Input 1 Port A Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_PORTA_SRC0	The interrupt for Video Input 1 Port A Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 10-188. Register Call Summary for Register VPE_INT0_CHANNEL1_INT_MASK

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-189. VPE_INT0_CHANNEL2_INT_STAT

Address Offset	0x0000 0050		
Physical Address	0x489D D050	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP1_MULT_ANCB_SRC9	INT_STAT_VIP1_MULT_ANCB_SRC8	INT_STAT_VIP1_MULT_ANCB_SRC7	INT_STAT_VIP1_MULT_ANCB_SRC6	INT_STAT_VIP1_MULT_ANCB_SRC5	INT_STAT_VIP1_MULT_ANCB_SRC4	INT_STAT_VIP1_MULT_ANCB_SRC3	INT_STAT_VIP1_MULT_ANCB_SRC2	INT_STAT_VIP1_MULT_ANCB_SRC1	INT_STAT_VIP1_MULT_ANCB_SRC0	INT_STAT_VIP1_MULT_ANCA_SRC15	INT_STAT_VIP1_MULT_ANCA_SRC14	INT_STAT_VIP1_MULT_ANCA_SRC13	INT_STAT_VIP1_MULT_ANCA_SRC12	INT_STAT_VIP1_MULT_ANCA_SRC11	INT_STAT_VIP1_MULT_ANCA_SRC10	INT_STAT_VIP1_MULT_ANCA_SRC9	INT_STAT_VIP1_MULT_ANCA_SRC8	INT_STAT_VIP1_MULT_ANCA_SRC7	INT_STAT_VIP1_MULT_ANCA_SRC6	INT_STAT_VIP1_MULT_ANCA_SRC5	INT_STAT_VIP1_MULT_ANCA_SRC4	INT_STAT_VIP1_MULT_ANCA_SRC3	INT_STAT_VIP1_MULT_ANCA_SRC2	INT_STAT_VIP1_MULT_ANCA_SRC1	INT_STAT_VIP1_MULT_ANCA_SRC0	INT_STAT_VIP1_MULT_PORTB_SRC15	INT_STAT_VIP1_MULT_PORTB_SRC14	INT_STAT_VIP1_MULT_PORTB_SRC13	INT_STAT_VIP1_MULT_PORTB_SRC12	INT_STAT_VIP1_MULT_PORTB_SRC11	INT_STAT_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
28	INT_STAT_VIP1_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP1_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP1_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP1_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_STAT_VIP1_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP1_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP1_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP1_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_VIP1_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP1_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	INT_STAT_VIP1_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP1_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP1_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-190. Register Call Summary for Register VPE_INT0_CHANNEL2_INT_STAT

VPE Functional Description • VPDMA Interrupts: [0]
VPE Register Manual • VPE_VPDMA Register Summary: [1]

Table 10-191. VPE_INT0_CHANNEL2_INT_MASK

Address Offset	0x0000 0054		
Physical Address	0x489D D054	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP1_MULT_ANCB_SRC9	INT_MASK_VIP1_MULT_ANCB_SRC8	INT_MASK_VIP1_MULT_ANCB_SRC7	INT_MASK_VIP1_MULT_ANCB_SRC6	INT_MASK_VIP1_MULT_ANCB_SRC5	INT_MASK_VIP1_MULT_ANCB_SRC4	INT_MASK_VIP1_MULT_ANCB_SRC3	INT_MASK_VIP1_MULT_ANCB_SRC2	INT_MASK_VIP1_MULT_ANCB_SRC1	INT_MASK_VIP1_MULT_ANCB_SRC0	INT_MASK_VIP1_MULT_ANCA_SRC15	INT_MASK_VIP1_MULT_ANCA_SRC14	INT_MASK_VIP1_MULT_ANCA_SRC13	INT_MASK_VIP1_MULT_ANCA_SRC12	INT_MASK_VIP1_MULT_ANCA_SRC11	INT_MASK_VIP1_MULT_ANCA_SRC10	INT_MASK_VIP1_MULT_ANCA_SRC9	INT_MASK_VIP1_MULT_ANCA_SRC8	INT_MASK_VIP1_MULT_ANCA_SRC7	INT_MASK_VIP1_MULT_ANCA_SRC6	INT_MASK_VIP1_MULT_ANCA_SRC5	INT_MASK_VIP1_MULT_ANCA_SRC4	INT_MASK_VIP1_MULT_ANCA_SRC3	INT_MASK_VIP1_MULT_ANCA_SRC2	INT_MASK_VIP1_MULT_ANCA_SRC1	INT_MASK_VIP1_MULT_ANCA_SRC0	INT_MASK_VIP1_MULT_PORTB_SRC15	INT_MASK_VIP1_MULT_PORTB_SRC14	INT_MASK_VIP1_MULT_PORTB_SRC13	INT_MASK_VIP1_MULT_PORTB_SRC12	INT_MASK_VIP1_MULT_PORTB_SRC11	INT_MASK_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_ANCB_SRC9	The interrupt for Video Input 1 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_ANCB_SRC8	The interrupt for Video Input 1 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_ANCB_SRC7	The interrupt for Video Input 1 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_ANCB_SRC6	The interrupt for Video Input 1 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_ANCB_SRC5	The interrupt for Video Input 1 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_ANCB_SRC4	The interrupt for Video Input 1 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_ANCB_SRC3	The interrupt for Video Input 1 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_ANCB_SRC2	The interrupt for Video Input 1 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_ANCB_SRC1	The interrupt for Video Input 1 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP1_MULT_ANCB_SRC0	The interrupt for Video Input 1 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_ANCA_SRC15	The interrupt for Video Input 1 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_MASK_VIP1_MULT_ANCA_SRC14	The interrupt for Video Input 1 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_ANCA_SRC13	The interrupt for Video Input 1 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_ANCA_SRC12	The interrupt for Video Input 1 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_ANCA_SRC11	The interrupt for Video Input 1 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_ANCA_SRC10	The interrupt for Video Input 1 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_ANCA_SRC9	The interrupt for Video Input 1 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_ANCA_SRC8	The interrupt for Video Input 1 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_ANCA_SRC7	The interrupt for Video Input 1 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_ANCA_SRC6	The interrupt for Video Input 1 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_MULT_ANCA_SRC5	The interrupt for Video Input 1 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_ANCA_SRC4	The interrupt for Video Input 1 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_ANCA_SRC3	The interrupt for Video Input 1 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_ANCA_SRC2	The interrupt for Video Input 1 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_MULT_ANCA_SRC1	The interrupt for Video Input 1 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_ANCA_SRC0	The interrupt for Video Input 1 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_PORTB_SRC15	The interrupt for Video Input 1 Port B Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	INT_MASK_VIP1_MULT_PORTB_SRC14	The interrupt for Video Input 1 Port B Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_PORTB_SRC13	The interrupt for Video Input 1 Port B Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_PORTB_SRC12	The interrupt for Video Input 1 Port B Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_PORTB_SRC11	The interrupt for Video Input 1 Port B Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_PORTB_SRC10	The interrupt for Video Input 1 Port B Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-192. Register Call Summary for Register VPE_INT0_CHANNEL2_INT_MASK

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-193. VPE_INT0_CHANNEL3_INT_STAT

Address Offset	0x0000 0058		
Physical Address	0x489D D058	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP2_MULT_PORTB_SRC3	INT_STAT_VIP2_MULT_PORTB_SRC2	INT_STAT_VIP2_MULT_PORTB_SRC1	INT_STAT_VIP2_MULT_PORTB_SRC0	INT_STAT_VIP2_MULT_PORTA_SRC15	INT_STAT_VIP2_MULT_PORTA_SRC14	INT_STAT_VIP2_MULT_PORTA_SRC13	INT_STAT_VIP2_MULT_PORTA_SRC12	INT_STAT_VIP2_MULT_PORTA_SRC11	INT_STAT_VIP2_MULT_PORTA_SRC10	INT_STAT_VIP2_MULT_PORTA_SRC9	INT_STAT_VIP2_MULT_PORTA_SRC8	INT_STAT_VIP2_MULT_PORTA_SRC7	INT_STAT_VIP2_MULT_PORTA_SRC6	INT_STAT_VIP2_MULT_PORTA_SRC5	INT_STAT_VIP2_MULT_PORTA_SRC4	INT_STAT_VIP2_MULT_PORTA_SRC3	INT_STAT_VIP2_MULT_PORTA_SRC2	INT_STAT_VIP2_MULT_PORTA_SRC1	INT_STAT_VIP2_MULT_PORTA_SRC0	INT_STAT_VIP1_PORTB_RGB	INT_STAT_VIP1_PORTA_RGB	INT_STAT_VIP1_PORTB_CHROMA	INT_STAT_VIP1_PORTB_LUMA	INT_STAT_VIP1_PORTA_CHROMA	INT_STAT_VIP1_PORTA_LUMA	INT_STAT_VIP1_MULT_ANCB_SRC15	INT_STAT_VIP1_MULT_ANCB_SRC14	INT_STAT_VIP1_MULT_ANCB_SRC13	INT_STAT_VIP1_MULT_ANCB_SRC12	INT_STAT_VIP1_MULT_ANCB_SRC11	INT_STAT_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP2_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP2_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP2_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP2_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
23	INT_STAT_VIP2_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP2_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP2_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
15	INT_STAT_VIP2_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_PORTB_RGB	The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_PORTA_RGB	The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_PORTB_CHROMA	The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_PORTB_LUMA	The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	INT_STAT_VIP1_PORTA_CHROMA	The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_PORTA_LUMA	The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP1_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP1_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP1_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP1_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-194. Register Call Summary for Register VPE_INT0_CHANNEL3_INT_STAT

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-195. VPE_INT0_CHANNEL3_INT_MASK

Address Offset	0x0000 005C		
Physical Address	0x489D D05C	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_MULT_PORTB_SRC3	INT_MASK_VIP2_MULT_PORTB_SRC2	INT_MASK_VIP2_MULT_PORTB_SRC1	INT_MASK_VIP2_MULT_PORTB_SRC0	INT_MASK_VIP2_MULT_PORTA_SRC15	INT_MASK_VIP2_MULT_PORTA_SRC14	INT_MASK_VIP2_MULT_PORTA_SRC13	INT_MASK_VIP2_MULT_PORTA_SRC12	INT_MASK_VIP2_MULT_PORTA_SRC11	INT_MASK_VIP2_MULT_PORTA_SRC10	INT_MASK_VIP2_MULT_PORTA_SRC9	INT_MASK_VIP2_MULT_PORTA_SRC8	INT_MASK_VIP2_MULT_PORTA_SRC7	INT_MASK_VIP2_MULT_PORTA_SRC6	INT_MASK_VIP2_MULT_PORTA_SRC5	INT_MASK_VIP2_MULT_PORTA_SRC4	INT_MASK_VIP2_MULT_PORTA_SRC3	INT_MASK_VIP2_MULT_PORTA_SRC2	INT_MASK_VIP2_MULT_PORTA_SRC1	INT_MASK_VIP2_MULT_PORTA_SRC0	INT_MASK_VIP1_PORTB_RGB	INT_MASK_VIP1_PORTA_RGB	INT_MASK_VIP1_PORTB_CHROMA	INT_MASK_VIP1_PORTB_LUMA	INT_MASK_VIP1_PORTA_CHROMA	INT_MASK_VIP1_PORTA_LUMA	INT_MASK_VIP1_MULT_ANCB_SRC15	INT_MASK_VIP1_MULT_ANCB_SRC14	INT_MASK_VIP1_MULT_ANCB_SRC13	INT_MASK_VIP1_MULT_ANCB_SRC12	INT_MASK_VIP1_MULT_ANCB_SRC11	INT_MASK_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_PORTB_SRC3	The interrupt for Video Input 2 Port B Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_PORTB_SRC2	The interrupt for Video Input 2 Port B Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_PORTB_SRC1	The interrupt for Video Input 2 Port B Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_PORTB_SRC0	The interrupt for Video Input 2 Port B Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP2_MULT_PORTA_SRC15	The interrupt for Video Input 2 Port A Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_PORTA_SRC14	The interrupt for Video Input 2 Port A Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_PORTA_SRC13	The interrupt for Video Input 2 Port A Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
24	INT_MASK_VIP2_MULT_PORTA_SRC12	The interrupt for Video Input 2 Port A Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_PORTA_SRC11	The interrupt for Video Input 2 Port A Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP2_MULT_PORTA_SRC10	The interrupt for Video Input 2 Port A Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP2_MULT_PORTA_SRC9	The interrupt for Video Input 2 Port A Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP2_MULT_PORTA_SRC8	The interrupt for Video Input 2 Port A Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_PORTA_SRC7	The interrupt for Video Input 2 Port A Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_PORTA_SRC6	The interrupt for Video Input 2 Port A Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_PORTA_SRC5	The interrupt for Video Input 2 Port A Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_PORTA_SRC4	The interrupt for Video Input 2 Port A Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_PORTA_SRC3	The interrupt for Video Input 2 Port A Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_PORTA_SRC2	The interrupt for Video Input 2 Port A Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_PORTA_SRC1	The interrupt for Video Input 2 Port A Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_PORTA_SRC0	The interrupt for Video Input 2 Port A Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_PORTB_RGB	The interrupt for Video Input 1 Port B RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_PORTA_RGB	The interrupt for Video Input 1 Port A RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
8	INT_MASK_VIP1_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_ANCB_SRC15	The interrupt for Video Input 1 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP1_MULT_ANCB_SRC14	The interrupt for Video Input 1 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_ANCB_SRC13	The interrupt for Video Input 1 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_ANCB_SRC12	The interrupt for Video Input 1 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_ANCB_SRC11	The interrupt for Video Input 1 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_ANCB_SRC10	The interrupt for Video Input 1 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-196. Register Call Summary for Register VPE_INT0_CHANNEL3_INT_MASK

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-197. VPE_INT0_CHANNEL4_INT_STAT

Address Offset	0x0000 0060		
Physical Address	0x489D D060	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP2_MULT_ANCB_SRC3	INT_STAT_VIP2_MULT_ANCB_SRC2	INT_STAT_VIP2_MULT_ANCB_SRC1	INT_STAT_VIP2_MULT_ANCB_SRC0	INT_STAT_VIP2_MULT_ANCA_SRC15	INT_STAT_VIP2_MULT_ANCA_SRC14	INT_STAT_VIP2_MULT_ANCA_SRC13	INT_STAT_VIP2_MULT_ANCA_SRC12	INT_STAT_VIP2_MULT_ANCA_SRC11	INT_STAT_VIP2_MULT_ANCA_SRC10	INT_STAT_VIP2_MULT_ANCA_SRC9	INT_STAT_VIP2_MULT_ANCA_SRC8	INT_STAT_VIP2_MULT_ANCA_SRC7	INT_STAT_VIP2_MULT_ANCA_SRC6	INT_STAT_VIP2_MULT_ANCA_SRC5	INT_STAT_VIP2_MULT_ANCA_SRC4	INT_STAT_VIP2_MULT_ANCA_SRC3	INT_STAT_VIP2_MULT_ANCA_SRC2	INT_STAT_VIP2_MULT_ANCA_SRC1	INT_STAT_VIP2_MULT_ANCA_SRC0	INT_STAT_VIP2_MULT_PORTB_SRC15	INT_STAT_VIP2_MULT_PORTB_SRC14	INT_STAT_VIP2_MULT_PORTB_SRC13	INT_STAT_VIP2_MULT_PORTB_SRC12	INT_STAT_VIP2_MULT_PORTB_SRC11	INT_STAT_VIP2_MULT_PORTB_SRC10	INT_STAT_VIP2_MULT_PORTB_SRC9	INT_STAT_VIP2_MULT_PORTB_SRC8	INT_STAT_VIP2_MULT_PORTB_SRC7	INT_STAT_VIP2_MULT_PORTB_SRC6	INT_STAT_VIP2_MULT_PORTB_SRC5	INT_STAT_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP2_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP2_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
25	INT_STAT_VIP2_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP2_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP2_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP2_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_STAT_VIP2_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
9	INT_STAT_VIP2_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP2_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP2_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP2_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
1	INT_STAT_VIP2_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP2_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-198. Register Call Summary for Register VPE_INT0_CHANNEL4_INT_STAT

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-199. VPE_INT0_CHANNEL4_INT_MASK

Address Offset	0x0000 0064		
Physical Address	0x489D D064	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_MULT_ANCB_SRC3	INT_MASK_VIP2_MULT_ANCB_SRC2	INT_MASK_VIP2_MULT_ANCB_SRC1	INT_MASK_VIP2_MULT_ANCB_SRC0	INT_MASK_VIP2_MULT_ANCA_SRC15	INT_MASK_VIP2_MULT_ANCA_SRC14	INT_MASK_VIP2_MULT_ANCA_SRC13	INT_MASK_VIP2_MULT_ANCA_SRC12	INT_MASK_VIP2_MULT_ANCA_SRC11	INT_MASK_VIP2_MULT_ANCA_SRC10	INT_MASK_VIP2_MULT_ANCA_SRC9	INT_MASK_VIP2_MULT_ANCA_SRC8	INT_MASK_VIP2_MULT_ANCA_SRC7	INT_MASK_VIP2_MULT_ANCA_SRC6	INT_MASK_VIP2_MULT_ANCA_SRC5	INT_MASK_VIP2_MULT_ANCA_SRC4	INT_MASK_VIP2_MULT_ANCA_SRC3	INT_MASK_VIP2_MULT_ANCA_SRC2	INT_MASK_VIP2_MULT_ANCA_SRC1	INT_MASK_VIP2_MULT_ANCA_SRC0	INT_MASK_VIP2_MULT_PORTB_SRC15	INT_MASK_VIP2_MULT_PORTB_SRC14	INT_MASK_VIP2_MULT_PORTB_SRC13	INT_MASK_VIP2_MULT_PORTB_SRC12	INT_MASK_VIP2_MULT_PORTB_SRC11	INT_MASK_VIP2_MULT_PORTB_SRC10	INT_MASK_VIP2_MULT_PORTB_SRC9	INT_MASK_VIP2_MULT_PORTB_SRC8	INT_MASK_VIP2_MULT_PORTB_SRC7	INT_MASK_VIP2_MULT_PORTB_SRC6	INT_MASK_VIP2_MULT_PORTB_SRC5	INT_MASK_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_ANCB_SRC3	The interrupt for Video Input 2 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_ANCB_SRC2	The interrupt for Video Input 2 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
29	INT_MASK_VIP2_MULT_ANCB_SRC1	The interrupt for Video Input 2 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_ANCB_SRC0	The interrupt for Video Input 2 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP2_MULT_ANCA_SRC15	The interrupt for Video Input 2 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_ANCA_SRC14	The interrupt for Video Input 2 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_ANCA_SRC13	The interrupt for Video Input 2 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_ANCA_SRC12	The interrupt for Video Input 2 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_ANCA_SRC11	The interrupt for Video Input 2 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP2_MULT_ANCA_SRC10	The interrupt for Video Input 2 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP2_MULT_ANCA_SRC9	The interrupt for Video Input 2 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP2_MULT_ANCA_SRC8	The interrupt for Video Input 2 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_ANCA_SRC7	The interrupt for Video Input 2 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_ANCA_SRC6	The interrupt for Video Input 2 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_ANCA_SRC5	The interrupt for Video Input 2 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_ANCA_SRC4	The interrupt for Video Input 2 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_ANCA_SRC3	The interrupt for Video Input 2 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_ANCA_SRC2	The interrupt for Video Input 2 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
13	INT_MASK_VIP2_MULT_ANCA_SRC1	The interrupt for Video Input 2 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_ANCA_SRC0	The interrupt for Video Input 2 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_PORTB_SRC15	The interrupt for Video Input 2 Port B Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_PORTB_SRC14	The interrupt for Video Input 2 Port B Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_PORTB_SRC13	The interrupt for Video Input 2 Port B Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_PORTB_SRC12	The interrupt for Video Input 2 Port B Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP2_MULT_PORTB_SRC11	The interrupt for Video Input 2 Port B Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_PORTB_SRC10	The interrupt for Video Input 2 Port B Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP2_MULT_PORTB_SRC9	The interrupt for Video Input 2 Port B Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_PORTB_SRC8	The interrupt for Video Input 2 Port B Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP2_MULT_PORTB_SRC7	The interrupt for Video Input 2 Port B Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP2_MULT_PORTB_SRC6	The interrupt for Video Input 2 Port B Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_PORTB_SRC5	The interrupt for Video Input 2 Port B Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_PORTB_SRC4	The interrupt for Video Input 2 Port B Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-200. Register Call Summary for Register VPE_INT0_CHANNEL4_INT_MASK

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-201. VPE_INT0_CHANNEL5_INT_STAT

Address Offset	0x0000 0068		
Physical Address	0x489D D068	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_TRANSCODE2_CHROMA	INT_STAT_TRANSCODE2_LUMA	INT_STAT_TRANSCODE1_CHROMA	INT_STAT_TRANSCODE1_LUMA	INT_STAT_AUX_IN	INT_STAT_PIP_FRAME	INT_STAT_POST_COMP_WR	INT_STAT_VBI_SD_VENC	RESERVED	INT_STAT_NF_LAST_CHROMA	INT_STAT_NF_LAST_LUMA	INT_STAT_NF_WRITE_CHROMA	INT_STAT_NF_WRITE_LUMA	INT_STAT_OTHER	INT_STAT_VIP2_PORTB_RGB	INT_STAT_VIP2_PORTA_RGB	INT_STAT_VIP2_PORTB_CHROMA	INT_STAT_VIP2_PORTB_LUMA	INT_STAT_VIP2_PORTA_CHROMA	INT_STAT_VIP2_PORTA_LUMA	INT_STAT_VIP2_MULT_ANCB_SRC15	INT_STAT_VIP2_MULT_ANCB_SRC14	INT_STAT_VIP2_MULT_ANCB_SRC13	INT_STAT_VIP2_MULT_ANCB_SRC12	INT_STAT_VIP2_MULT_ANCB_SRC11	INT_STAT_VIP2_MULT_ANCB_SRC10	INT_STAT_VIP2_MULT_ANCB_SRC9	INT_STAT_VIP2_MULT_ANCB_SRC8	INT_STAT_VIP2_MULT_ANCB_SRC7	INT_STAT_VIP2_MULT_ANCB_SRC6	INT_STAT_VIP2_MULT_ANCB_SRC5	INT_STAT_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_TRANSCODE2_CHROMA	The last write DMA transaction has completed for channel transcode2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_TRANSCODE2_LUMA	The last write DMA transaction has completed for channel transcode2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_TRANSCODE1_CHROMA	The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_TRANSCODE1_LUMA	The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
27	INT_STAT_AUX_IN	The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_PIP_FRAME	The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_POST_COMP_WR	The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VBI_SD_VENC	The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_STAT_NF_LAST_CHROMA	The last write DMA transaction has completed for channel nf_last_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_NF_LAST_LUMA	The last write DMA transaction has completed for channel nf_last_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_NF_WRITE_CHROMA	The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_NF_WRITE_LUMA	The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
18	INT_STAT_OTHER	This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_PORTB_RGB	The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_PORTA_RGB	The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_PORTB_CHROMA	The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_PORTB_LUMA	The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_PORTA_CHROMA	The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_PORTA_LUMA	The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
9	INT_STAT_VIP2_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP2_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP2_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP2_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
1	INT_STAT_VIP2_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP2_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-202. Register Call Summary for Register VPE_INT0_CHANNEL5_INT_STAT

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-203. VPE_INT0_CHANNEL5_INT_MASK

Address Offset	0x0000 006C		
Physical Address	0x489D D06C	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_TRANSCODE2_CHROMA	INT_MASK_TRANSCODE2_LUMA	INT_MASK_TRANSCODE1_CHROMA	INT_MASK_TRANSCODE1_LUMA	INT_MASK_AUX_IN	INT_MASK_PIP_FRAME	INT_MASK_POST_COMP_WR	INT_MASK_VBI_SD_VENC	RESERVED	INT_MASK_NF_LAST_CHROMA	INT_MASK_NF_LAST_LUMA	INT_MASK_NF_WRITE_CHROMA	INT_MASK_NF_WRITE_LUMA	INT_MASK_OTHER	INT_MASK_VIP2_PORTB_RGB	INT_MASK_VIP2_PORTA_RGB	INT_MASK_VIP2_PORTB_CHROMA	INT_MASK_VIP2_PORTB_LUMA	INT_MASK_VIP2_PORTA_CHROMA	INT_MASK_VIP2_PORTA_LUMA	INT_MASK_VIP2_MULT_ANCB_SRC15	INT_MASK_VIP2_MULT_ANCB_SRC14	INT_MASK_VIP2_MULT_ANCB_SRC13	INT_MASK_VIP2_MULT_ANCB_SRC12	INT_MASK_VIP2_MULT_ANCB_SRC11	INT_MASK_VIP2_MULT_ANCB_SRC10	INT_MASK_VIP2_MULT_ANCB_SRC9	INT_MASK_VIP2_MULT_ANCB_SRC8	INT_MASK_VIP2_MULT_ANCB_SRC7	INT_MASK_VIP2_MULT_ANCB_SRC6	INT_MASK_VIP2_MULT_ANCB_SRC5	INT_MASK_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_TRANSCODE2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_TRANSCODE2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_TRANSCODE1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
28	INT_MASK_TRANSCODE1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_AUX_IN	The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_PIP_FRAME	The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_POST_COMP_WR	The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VBI_SD_VENC	The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_MASK_NF_LAST_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_NF_LAST_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_NF_WRITE_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_NF_WRITE_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_OTHER	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_PORTB_RGB	The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_PORTA_RGB	The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_ANCB_SRC15	The interrupt for Video Input 2 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_ANCB_SRC14	The interrupt for Video Input 2 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
9	INT_MASK_VIP2_MULT_ANCB_SRC13	The interrupt for Video Input 2 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_ANCB_SRC12	The interrupt for Video Input 2 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP2_MULT_ANCB_SRC11	The interrupt for Video Input 2 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_ANCB_SRC10	The interrupt for Video Input 2 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP2_MULT_ANCB_SRC9	The interrupt for Video Input 2 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_ANCB_SRC8	The interrupt for Video Input 2 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP2_MULT_ANCB_SRC7	The interrupt for Video Input 2 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP2_MULT_ANCB_SRC6	The interrupt for Video Input 2 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_ANCB_SRC5	The interrupt for Video Input 2 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_ANCB_SRC4	The interrupt for Video Input 2 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-204. Register Call Summary for Register VPE_INT0_CHANNEL5_INT_MASK

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-205. VPE_INT0_CLIENT0_INT_STAT

Address Offset	0x0000 0078		
Physical Address	0x489D D078	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_GRPX1_DATA	INT_STAT_COMP_WRBK	INT_STAT_SC_OUT	RESERVED								INT_STAT_SC_IN_LUMA	INT_STAT_SC_IN_CHROMA	INT_STAT_PIP_WRBK	INT_STAT_DEI_SC_OUT	RESERVED	INT_STAT_DEI_HQ_MV_OUT	RESERVED		INT_STAT_DEI_HQ_MV_IN	RESERVED					INT_STAT_DEI_HQ_3_CHROMA	INT_STAT_DEI_HQ_3_LUMA	INT_STAT_DEI_HQ_2_CHROMA	INT_STAT_DEI_HQ_2_LUMA	INT_STAT_DEI_HQ_1_LUMA	INT_STAT_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_STAT_GRPX1_DATA	The client interface grpx1_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_COMP_WRBK	The client interface comp_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_SC_OUT	The client interface sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28:21	RESERVED	Reserved	R	0x00
20	INT_STAT_SC_IN_LUMA	The client interface sc_in_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_SC_IN_CHROMA	The client interface sc_in_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_PIP_WRBK	The client interface pip_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_STAT_DEI_SC_OUT	The client interface dei_sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_STAT_DEI_HQ_MV_OUT	The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_STAT_DEI_HQ_MV_IN	The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_DEI_HQ_3_CHROMA	The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_DEI_HQ_3_LUMA	The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_DEI_HQ_2_CHROMA	The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_DEI_HQ_2_LUMA	The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_DEI_HQ_1_LUMA	The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INT_STAT_DEI_HQ_1_CHROMA	The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-206. Register Call Summary for Register VPE_INT0_CLIENT0_INT_STAT

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-207. VPE_INT0_CLIENT0_INT_MASK

Address Offset	0x0000 007C		
Physical Address	0x489D D07C	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX1_DATA	INT_MASK_COMP_WRBK	INT_MASK_SC_OUT	RESERVED								INT_MASK_SC_IN_LUMA	INT_MASK_SC_IN_CHROMA	INT_MASK_PIP_WRBK	INT_MASK_DEI_SC_OUT	RESERVED	INT_MASK_DEI_HQ_MV_OUT	RESERVED		INT_MASK_DEI_HQ_MV_IN	RESERVED					INT_MASK_DEI_HQ_3_CHROMA	INT_MASK_DEI_HQ_3_LUMA	INT_MASK_DEI_HQ_2_CHROMA	INT_MASK_DEI_HQ_2_LUMA	INT_MASK_DEI_HQ_1_LUMA	INT_MASK_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX1_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_COMP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28:21	RESERVED	Reserved	R	0x00
20	INT_MASK_SC_IN_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_SC_IN_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_PIP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_MASK_DEI_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_DEI_HQ_MV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_MASK_DEI_HQ_MV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_DEI_HQ_3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_DEI_HQ_3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_DEI_HQ_2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_DEI_HQ_2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_DEI_HQ_1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_DEI_HQ_1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-208. Register Call Summary for Register VPE_INT0_CLIENT0_INT_MASK

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-209. VPE_INT0_CLIENT1_INT_STAT

Address Offset	0x0000 0080		
Physical Address	0x489D D080	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	INT_STAT_VIP2_ANC_B	INT_STAT_VIP2_ANC_A	INT_STAT_VIP1_ANC_B	INT_STAT_VIP1_ANC_A	INT_STAT_TRANS2_LUMA	INT_STAT_TRANS2_CHROMA	INT_STAT_TRANS1_LUMA	INT_STAT_TRANS1_CHROMA	INT_STAT_HDMI_WRBK_OUT	INT_STAT_VPI_CTL	INT_STAT_VBI_SDVENC	RESERVED	INT_STAT_NF_420_UV_OUT	INT_STAT_NF_420_Y_OUT	INT_STAT_NF_420_UV_IN	INT_STAT_NF_420_Y_IN	INT_STAT_NF_422_IN	INT_STAT_GRPX3_ST	INT_STAT_GRPX2_ST	INT_STAT_GRPX1_ST	INT_STAT_VIP2_UP_UV	INT_STAT_VIP2_UP_Y	INT_STAT_VIP2_LO_UV	INT_STAT_VIP2_LO_Y	INT_STAT_VIP1_UP_UV	INT_STAT_VIP1_UP_Y	INT_STAT_VIP1_LO_UV	INT_STAT_VIP1_LO_Y	INT_STAT_GRPX3_DATA	INT_STAT_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESERVED	Reserved	R	0
29	INT_STAT_VIP2_ANC_B	The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
28	INT_STAT_VIP2_ANC_A	The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
27	INT_STAT_VIP1_ANC_B	The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
26	INT_STAT_VIP1_ANC_A	The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
25	INT_STAT_TRANS2_LUMA	The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
24	INT_STAT_TRANS2_CHROMA	The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
23	INT_STAT_TRANS1_LUMA	The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
22	INT_STAT_TRANS1_CHROMA	The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
21	INT_STAT_HDMI_WRBK_OUT	The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
20	INT_STAT_VPI_CTL	The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
19	INT_STAT_VBI_SDVENC	The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
18	RESERVED	Reserved	R	0
17	INT_STAT_NF_420_UV_OUT	The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
16	INT_STAT_NF_420_Y_OUT	The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
15	INT_STAT_NF_420_UV_IN	The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
14	INT_STAT_NF_420_Y_IN	The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
13	INT_STAT_NF_422_IN	The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_GRPX3_ST	The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
11	INT_STAT_GRPX2_ST	The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
10	INT_STAT_GRPX1_ST	The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
9	INT_STAT_VIP2_UP_UV	The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
8	INT_STAT_VIP2_UP_Y	The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
7	INT_STAT_VIP2_LO_UV	The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
6	INT_STAT_VIP2_LO_Y	The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
5	INT_STAT_VIP1_UP_UV	The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
4	INT_STAT_VIP1_UP_Y	The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
3	INT_STAT_VIP1_LO_UV	The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
2	INT_STAT_VIP1_LO_Y	The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
1	INT_STAT_GRPX3_DATA	The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
0	INT_STAT_GRPX2_DATA	The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Table 10-210. Register Call Summary for Register VPE_INT0_CLIENT1_INT_STAT

VPE Functional Description • VPDMA Interrupts: [0]
VPE Register Manual • VPE_VPDMA Register Summary: [1]

Table 10-211. VPE_INT0_CLIENT1_INT_MASK

Address Offset	0x0000 0084		
Physical Address	0x489D D084	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	INT_MASK_VIP2_ANC_B	INT_MASK_VIP2_ANC_A	INT_MASK_VIP1_ANC_B	INT_MASK_VIP1_ANC_A	INT_MASK_TRANS2_LUMA	INT_MASK_TRANS2_CHROMA	INT_MASK_TRANS1_LUMA	INT_MASK_TRANS1_CHROMA	INT_MASK_HDMI_WRBK_OUT	INT_MASK_VPI_CTL	INT_MASK_VBI_SDVENC	RESERVED	INT_MASK_NF_420_UV_OUT	INT_MASK_NF_420_Y_OUT	INT_MASK_NF_420_UV_IN	INT_MASK_NF_420_Y_IN	INT_MASK_NF_422_IN	INT_MASK_GRPX3_ST	INT_MASK_GRPX2_ST	INT_MASK_GRPX1_ST	INT_MASK_VIP2_UP_UV	INT_MASK_VIP2_UP_Y	INT_MASK_VIP2_LO_UV	INT_MASK_VIP2_LO_Y	INT_MASK_VIP1_UP_UV	INT_MASK_VIP1_UP_Y	INT_MASK_VIP1_LO_UV	INT_MASK_VIP1_LO_Y	INT_MASK_GRPX3_DATA	INT_MASK_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESERVED	Reserved	R	0
29	INT_MASK_VIP2_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
28	INT_MASK_VIP2_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
27	INT_MASK_VIP1_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
26	INT_MASK_VIP1_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
25	INT_MASK_TRANS2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
24	INT_MASK_TRANS2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
23	INT_MASK_TRANS1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
22	INT_MASK_TRANS1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
21	INT_MASK_HDMI_WRBK_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
20	INT_MASK_VPI_CTL	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
19	INT_MASK_VBI_SDVENC	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
18	RESERVED	Reserved	R	0
17	INT_MASK_NF_420_UV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
16	INT_MASK_NF_420_Y_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
15	INT_MASK_NF_420_UV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Bits	Field Name	Description	Type	Reset
14	INT_MASK_NF_420_Y_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
13	INT_MASK_NF_422_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
12	INT_MASK_GRPX3_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
11	INT_MASK_GRPX2_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
10	INT_MASK_GRPX1_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
9	INT_MASK_VIP2_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
8	INT_MASK_VIP2_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
7	INT_MASK_VIP2_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
6	INT_MASK_VIP2_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
5	INT_MASK_VIP1_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
4	INT_MASK_VIP1_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
3	INT_MASK_VIP1_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
2	INT_MASK_VIP1_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
1	INT_MASK_GRPX3_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
0	INT_MASK_GRPX2_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Table 10-212. Register Call Summary for Register VPE_INT0_CLIENT1_INT_MASK

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-213. VPE_INT0_LIST0_INT_STAT

Address Offset	0x0000 0088		
Physical Address	0x489D D088	**Instance**	VPE_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_CONTROL_DESCRIPTOR_INT15	INT_STAT_CONTROL_DESCRIPTOR_INT14	INT_STAT_CONTROL_DESCRIPTOR_INT13	INT_STAT_CONTROL_DESCRIPTOR_INT12	INT_STAT_CONTROL_DESCRIPTOR_INT11	INT_STAT_CONTROL_DESCRIPTOR_INT10	INT_STAT_CONTROL_DESCRIPTOR_INT9	INT_STAT_CONTROL_DESCRIPTOR_INT8	INT_STAT_CONTROL_DESCRIPTOR_INT7	INT_STAT_CONTROL_DESCRIPTOR_INT6	INT_STAT_CONTROL_DESCRIPTOR_INT5	INT_STAT_CONTROL_DESCRIPTOR_INT4	INT_STAT_CONTROL_DESCRIPTOR_INT3	INT_STAT_CONTROL_DESCRIPTOR_INT2	INT_STAT_CONTROL_DESCRIPTOR_INT1	INT_STAT_CONTROL_DESCRIPTOR_INT0	INT_STAT_LIST7_NOTIFY	INT_STAT_LIST7_COMPLETE	INT_STAT_LIST6_NOTIFY	INT_STAT_LIST6_COMPLETE	INT_STAT_LIST5_NOTIFY	INT_STAT_LIST5_COMPLETE	INT_STAT_LIST4_NOTIFY	INT_STAT_LIST4_COMPLETE	INT_STAT_LIST3_NOTIFY	INT_STAT_LIST3_COMPLETE	INT_STAT_LIST2_NOTIFY	INT_STAT_LIST2_COMPLETE	INT_STAT_LIST1_NOTIFY	INT_STAT_LIST1_COMPLETE	INT_STAT_LIST0_NOTIFY	INT_STAT_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_STAT_CONTROL_DESCRIPTOR_INT15	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_CONTROL_DESCRIPTOR_INT14	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_CONTROL_DESCRIPTOR_INT13	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_CONTROL_DESCRIPTOR_INT12	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_CONTROL_DESCRIPTOR_INT11	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_CONTROL_DESCRIPTOR_INT10	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_CONTROL_DESCRIPTOR_INT9	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_CONTROL_DESCRIPTOR_INT8	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_CONTROL_DESCRIPTOR_INT7	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	INT_STAT_CONTROL_DESCRIPTOR_INT6	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_CONTROL_DESCRIPTOR_INT5	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_CONTROL_DESCRIPTOR_INT4	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_CONTROL_DESCRIPTOR_INT3	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_CONTROL_DESCRIPTOR_INT2	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_CONTROL_DESCRIPTOR_INT1	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_CONTROL_DESCRIPTOR_INT0	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_LIST7_NOTIFY	A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_LIST7_COMPLETE	List 7 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_LIST6_NOTIFY	A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_LIST6_COMPLETE	List 6 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_LIST5_NOTIFY	A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_LIST5_COMPLETE	List 5 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
9	INT_STAT_LIST4_NOTIFY	A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_LIST4_COMPLETE	List 4 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_LIST3_NOTIFY	A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_LIST3_COMPLETE	List 3 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_LIST2_NOTIFY	A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_LIST2_COMPLETE	List 2 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_LIST1_NOTIFY	A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_LIST1_COMPLETE	List 1 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_LIST0_NOTIFY	A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_LIST0_COMPLETE	List 0 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 10-214. Register Call Summary for Register VPE_INT0_LIST0_INT_STAT

VPE Functional Description
- VPDMA Interrupts: [0] [1] [2]

VPE Register Manual
- VPE_VPDMA Register Summary: [3]

Table 10-215. VPE_INT0_LIST0_INT_MASK

Address Offset	0x0000 008C		
Physical Address	0x489D D08C	**Instance**	VPE_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_CONTROL_DESCRIPTOR_INT15	INT_MASK_CONTROL_DESCRIPTOR_INT14	INT_MASK_CONTROL_DESCRIPTOR_INT13	INT_MASK_CONTROL_DESCRIPTOR_INT12	INT_MASK_CONTROL_DESCRIPTOR_INT11	INT_MASK_CONTROL_DESCRIPTOR_INT10	INT_MASK_CONTROL_DESCRIPTOR_INT9	INT_MASK_CONTROL_DESCRIPTOR_INT8	INT_MASK_CONTROL_DESCRIPTOR_INT7	INT_MASK_CONTROL_DESCRIPTOR_INT6	INT_MASK_CONTROL_DESCRIPTOR_INT5	INT_MASK_CONTROL_DESCRIPTOR_INT4	INT_MASK_CONTROL_DESCRIPTOR_INT3	INT_MASK_CONTROL_DESCRIPTOR_INT2	INT_MASK_CONTROL_DESCRIPTOR_INT1	INT_MASK_CONTROL_DESCRIPTOR_INT0	INT_MASK_LIST7_NOTIFY	INT_MASK_LIST7_COMPLETE	INT_MASK_LIST6_NOTIFY	INT_MASK_LIST6_COMPLETE	INT_MASK_LIST5_NOTIFY	INT_MASK_LIST5_COMPLETE	INT_MASK_LIST4_NOTIFY	INT_MASK_LIST4_COMPLETE	INT_MASK_LIST3_NOTIFY	INT_MASK_LIST3_COMPLETE	INT_MASK_LIST2_NOTIFY	INT_MASK_LIST2_COMPLETE	INT_MASK_LIST1_NOTIFY	INT_MASK_LIST1_COMPLETE	INT_MASK_LIST0_NOTIFY	INT_MASK_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_MASK_CONTROL_DESCRIPTOR_INT15	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_CONTROL_DESCRIPTOR_INT14	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_CONTROL_DESCRIPTOR_INT13	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_CONTROL_DESCRIPTOR_INT12	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_CONTROL_DESCRIPTOR_INT11	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_CONTROL_DESCRIPTOR_INT10	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_CONTROL_DESCRIPTOR_INT9	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_CONTROL_DESCRIPTOR_INT8	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_CONTROL_DESCRIPTOR_INT7	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_CONTROL_DESCRIPTOR_INT6	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_CONTROL_DESCRIPTOR_INT5	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_CONTROL_DESCRIPTOR_INT4	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_CONTROL_DESCRIPTOR_INT3	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_CONTROL_DESCRIPTOR_INT2	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_MASK_CONTROL_DESCRIPTOR_INT1	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_CONTROL_DESCRIPTOR_INT0	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_LIST7_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_LIST7_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_LIST6_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_LIST6_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_LIST5_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_LIST5_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_LIST4_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_LIST4_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_LIST3_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_LIST3_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_LIST2_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_LIST2_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_LIST1_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_LIST1_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_LIST0_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_LIST0_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 10-216. Register Call Summary for Register VPE_INT0_LIST0_INT_MASK

VPE Functional Description
- VPDMA Interrupts: [0]

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-217. VPE_PERF_MON0

Address Offset	0x0000 0200		
Physical Address	0x489D D200	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_1_chroma 0x1 : vpi_ctl 0x2: dei_hq_1_luma 0x3: dei_hq_2_luma	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_1_chroma 0x1 : vpi_ctl 0x2: dei_hq_1_luma 0x3: dei_hq_2_luma	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-218. Register Call Summary for Register VPE_PERF_MON0

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-219. VPE_PERF_MON1

Address Offset	0x0000 0204		
Physical Address	0x489D D204	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_1_luma 0x1 : dei_hq_1_chroma 0x2: dei_hq_2_luma 0x3: dei_hq_2_chroma	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_1_luma 0x1 : dei_hq_1_chroma 0x2: dei_hq_2_luma 0x3: dei_hq_2_chroma	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-220. Register Call Summary for Register VPE_PERF_MON1

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-221. VPE_PERF_MON2

Address Offset	0x0000 0208		
Physical Address	0x489D D208	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_2_luma 0x1 : dei_hq_1_luma 0x2: dei_hq_2_chroma 0x3: dei_hq_3_luma	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_2_luma 0x1 : dei_hq_1_luma 0x2: dei_hq_2_chroma 0x3: dei_hq_3_luma	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-222. Register Call Summary for Register VPE_PERF_MON2

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-223. VPE_PERF_MON3

Address Offset	0x0000 020C		
Physical Address	0x489D D20C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_2_luma 0x1 : dei_hq_1_luma 0x2: dei_hq_2_chroma 0x3: dei_hq_3_luma	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_2_luma 0x1 : dei_hq_1_luma 0x2: dei_hq_2_chroma 0x3: dei_hq_3_luma	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-224. Register Call Summary for Register VPE_PERF_MON3

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-225. VPE_PERF_MON4

Address Offset	0x0000 0210		
Physical Address	0x489D D210	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_3_luma 0x1 : dei_hq_2_chroma 0x 2: dei_hq_3_chroma 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_3_luma 0x1 : dei_hq_2_chroma 0x 2: dei_hq_3_chroma 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-226. Register Call Summary for Register VPE_PERF_MON4

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-227. VPE_PERF_MON5

Address Offset	0x0000 0214		
Physical Address	0x489D D214	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_3_luma 0x1 : dei_hq_2_chroma 0x 2: dei_hq_3_chroma 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_3_luma 0x1 : dei_hq_2_chroma 0x 2: dei_hq_3_chroma 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-228. Register Call Summary for Register VPE_PERF_MON5

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-229. VPE_PERF_MON6

Address Offset	0x0000 0218		
Physical Address	0x489D D218	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : dei_hq_3_chroma 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_3_chroma 0x1 : dei_hq_3_chroma 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-230. Register Call Summary for Register VPE_PERF_MON6

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-231. VPE_PERF_MON7

Address Offset	0x0000 021C		
Physical Address	0x489D D21C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-232. Register Call Summary for Register VPE_PERF_MON7

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-233. VPE_PERF_MON8

Address Offset	0x0000 0220		
Physical Address	0x489D D220	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-234. Register Call Summary for Register VPE_PERF_MON8

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-235. VPE_PERF_MON9

Address Offset	0x0000 0224		
Physical Address	0x489D D224	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-236. Register Call Summary for Register VPE_PERF_MON9

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-237. VPE_PERF_MON10

Address Offset	0x0000 0228		
Physical Address	0x489D D228	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3: dei_hq_mv_in	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3: dei_hq_mv_in	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-238. Register Call Summary for Register VPE_PERF_MON10

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-239. VPE_PERF_MON11

Address Offset	0x0000 022C		
Physical Address	0x489D D22C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: dei_hq_mv_in 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: dei_hq_mv_in 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-240. Register Call Summary for Register VPE_PERF_MON11

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-241. VPE_PERF_MON12

Address Offset	0x0000 0230		
Physical Address	0x489D D230	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_mv_in 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_mv_in 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-242. Register Call Summary for Register VPE_PERF_MON12

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-243. VPE_PERF_MON13

Address Offset	0x0000 0234		
Physical Address	0x489D D234	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : dei_hq_mv_in 0x2: 0x3:dei_hq_mv_out	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : dei_hq_mv_in 0x2: 0x3:dei_hq_mv_out	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-244. Register Call Summary for Register VPE_PERF_MON13

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-245. VPE_PERF_MON14

Address Offset	0x0000 0238		
Physical Address	0x489D D238	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: dei_hq_mv_out 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: dei_hq_mv_out 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-246. Register Call Summary for Register VPE_PERF_MON14

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-247. VPE_PERF_MON15

Address Offset	0x0000 023C		
Physical Address	0x489D D23C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : dei_hq_mv_out 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : dei_hq_mv_out 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-248. Register Call Summary for Register VPE_PERF_MON15

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-249. VPE_PERF_MON16

Address Offset	0x0000 0240		
Physical Address	0x489D D240	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : dei_hq_mv_out 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : dei_hq_mv_out 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-250. Register Call Summary for Register VPE_PERF_MON16

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-251. VPE_PERF_MON17

Address Offset	0x0000 0244		
Physical Address	0x489D D244	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-252. Register Call Summary for Register VPE_PERF_MON17

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-253. VPE_PERF_MON18

Address Offset	0x0000 0248		
Physical Address	0x489D D248	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-254. Register Call Summary for Register VPE_PERF_MON18

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-255. VPE_PERF_MON19

Address Offset	0x0000 024C		
Physical Address	0x489D D24C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-256. Register Call Summary for Register VPE_PERF_MON19

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-257. VPE_PERF_MON20

Address Offset	0x0000 0250		
Physical Address	0x489D D250	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-258. Register Call Summary for Register VPE_PERF_MON20

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-259. VPE_PERF_MON21

Address Offset	0x0000 0254		
Physical Address	0x489D D254	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-260. Register Call Summary for Register VPE_PERF_MON21

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-261. VPE_PERF_MON22

Address Offset	0x0000 0258		
Physical Address	0x489D D258	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-262. Register Call Summary for Register VPE_PERF_MON22

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-263. VPE_PERF_MON23

Address Offset	0x0000 025C		
Physical Address	0x489D D25C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-264. Register Call Summary for Register VPE_PERF_MON23

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-265. VPE_PERF_MON24

Address Offset	0x0000 0260		
Physical Address	0x489D D260	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-266. Register Call Summary for Register VPE_PERF_MON24

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-267. VPE_PERF_MON25

Address Offset	0x0000 0264		
Physical Address	0x489D D264	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-268. Register Call Summary for Register VPE_PERF_MON25

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-269. VPE_PERF_MON26

Address Offset	0x0000 0268		
Physical Address	0x489D D268	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-270. Register Call Summary for Register VPE_PERF_MON26

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-271. VPE_PERF_MON27

Address Offset	0x0000 026C		
Physical Address	0x489D D26C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-272. Register Call Summary for Register VPE_PERF_MON27

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-273. VPE_PERF_MON28

Address Offset	0x0000 0270		
Physical Address	0x489D D270	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-274. Register Call Summary for Register VPE_PERF_MON28

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-275. VPE_PERF_MON29

Address Offset	0x0000 0274		
Physical Address	0x489D D274	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-276. Register Call Summary for Register VPE_PERF_MON29

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-277. VPE_PERF_MON30

Address Offset	0x0000 0278		
Physical Address	0x489D D278	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-278. Register Call Summary for Register VPE_PERF_MON30

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-279. VPE_PERF_MON31

Address Offset	0x0000 027C		
Physical Address	0x489D D27C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-280. Register Call Summary for Register VPE_PERF_MON31

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-281. VPE_PERF_MON32

Address Offset	0x0000 0280		
Physical Address	0x489D D280	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-282. Register Call Summary for Register VPE_PERF_MON32

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-283. VPE_PERF_MON33

Address Offset	0x0000 0284		
Physical Address	0x489D D284	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-284. Register Call Summary for Register VPE_PERF_MON33

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-285. VPE_PERF_MON34

Address Offset	0x0000 0288		
Physical Address	0x489D D288	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3: vip1_up_y	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: 0x3: vip1_up_y	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0x0 : command request 0x1 : command accept 0x2: data request 0x3: data rcvd 0x4: data empty 0x5: data full 0x6: frame start 0x7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-286. Register Call Summary for Register VPE_PERF_MON34

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-287. VPE_PERF_MON35

Address Offset	0x0000 028C		
Physical Address	0x489D D28C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : 0x1 : 0x2: vip1_up_y 0x3: vip1_up_uv	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : 0x1 : 0x2: vip1_up_y 0x3: vip1_up_uv	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-288. Register Call Summary for Register VPE_PERF_MON35

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-289. VPE_PERF_MON36

Address Offset	0x0000 0290		
Physical Address	0x489D D290	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-290. Register Call Summary for Register VPE_PERF_MON36

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-291. VPE_PERF_MON37

Address Offset	0x0000 0294		
Physical Address	0x489D D294	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-292. Register Call Summary for Register VPE_PERF_MON37

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-293. VPE_PERF_MON38

Address Offset	0x0000 0298		
Physical Address	0x489D D298	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0x0 : Running Average 0x1 : Minimum Value 0x2: Maximum Value 0x3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0x0 : vip1_up_uv 0x1 : vip1_up_y 0x2: 0x3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-294. Register Call Summary for Register VPE_PERF_MON38

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-295. VPE_PERF_MON39

Address Offset	0x0000 029C		
Physical Address	0x489D D29C	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-296. Register Call Summary for Register VPE_PERF_MON39

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-297. VPE_PERF_MON40

Address Offset	0x0000 02A0		
Physical Address	0x489D D2A0	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-298. Register Call Summary for Register VPE_PERF_MON40

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-299. VPE_PERF_MON41

Address Offset	0x0000 02A4		
Physical Address	0x489D D2A4	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-300. Register Call Summary for Register VPE_PERF_MON41

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-301. VPE_PERF_MON42

Address Offset	0x0000 02A8		
Physical Address	0x489D D2A8	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-302. Register Call Summary for Register VPE_PERF_MON42

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-303. VPE_PERF_MON43

Address Offset	0x0000 02AC		
Physical Address	0x489D D2AC	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-304. Register Call Summary for Register VPE_PERF_MON43

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-305. VPE_PERF_MON44

Address Offset	0x0000 02B0		
Physical Address	0x489D D2B0	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-306. Register Call Summary for Register VPE_PERF_MON44

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-307. VPE_PERF_MON45

Address Offset	0x0000 02B4		
Physical Address	0x489D D2B4	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-308. Register Call Summary for Register VPE_PERF_MON45

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-309. VPE_PERF_MON46

Address Offset	0x0000 02B8		
Physical Address	0x489D D2B8	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-310. Register Call Summary for Register VPE_PERF_MON46

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-311. VPE_PERF_MON47

Address Offset	0x0000 02BC		
Physical Address	0x489D D2BC	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-312. Register Call Summary for Register VPE_PERF_MON47

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-313. VPE_PERF_MON48

Address Offset	0x0000 02C0		
Physical Address	0x489D D2C0	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-314. Register Call Summary for Register VPE_PERF_MON48

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-315. VPE_PERF_MON49

Address Offset	0x0000 02C4		
Physical Address	0x489D D2C4	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter.	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter.	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-316. Register Call Summary for Register VPE_PERF_MON49

VPE Register Manual

- VPE_VPDMA Register Summary: [0]

Table 10-317. VPE_PERF_MON50

Address Offset	0x0000 02C8		
Physical Address	0x489D D2C8	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0:vpi_ctl 1:vpi_ctl 2:vpi_ctl 3:vpi_ctl	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0:vpi_ctl 1:vpi_ctl 2:vpi_ctl 3:vpi_ctl	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-318. Register Call Summary for Register VPE_PERF_MON50

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-319. VPE_PERF_MON51

Address Offset	0x0000 02CC		
Physical Address	0x489D D2CC	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vpi_ctl 1: dei_hq_1_chroma 2: dei_hq_1_chroma 3: dei_hq_1_luma	RW	0x0
27	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vpi_ctl 1: dei_hq_1_chroma 2: dei_hq_1_chroma 3: dei_hq_1_luma	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-320. Register Call Summary for Register VPE_PERF_MON51

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-321. VPE_PERF_MON52

Address Offset	0x0000 02D0		
Physical Address	0x489D D2D0	**Instance**	VPE_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vpi_ctl 1: dei_hq_1_chroma 2: dei_hq_1_chroma 3: dei_hq_1_luma	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vpi_ctl 1: dei_hq_1_chroma 2: dei_hq_1_chroma 3: dei_hq_1_luma	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 10-322. Register Call Summary for Register VPE_PERF_MON52

VPE Register Manual
- VPE_VPDMA Register Summary: [0]

Table 10-323. VPE_PRI_CHROMA_CSTAT

Address Offset	0x0000 0300		
Physical Address	0x489D D300	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START				LINE_MODE		RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:8	LINE_MODE	Selects the output mode of the line buffer. 0: repeat lines twice each output data line gets 2 times the number of frame lines. 1: each line once with Line Buffer Disabled, so no mirroring. Each line gets frame lines with identical data. 2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines. 3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines.	RW	0x0
7:0	RESERVED		R	0x0

Table 10-324. Register Call Summary for Register VPE_PRI_CHROMA_CSTAT

VPE Functional Description

- Modes of Operation (VPDMA): [0]
- VPDMA Basic Definitions:

VPE Register Manual

- VPE_VPDMA Register Summary: [2]

Table 10-325. VPE_PRI_LUMA_CSTAT

Address Offset	0x0000 0304		
Physical Address	0x489D D304	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-326. Register Call Summary for Register VPE_PRI_LUMA_CSTAT

VPE Functional Description • VPDMA Basic Definitions:
VPE Register Manual • VPE_VPDMA Register Summary: [1]

Table 10-327. VPE_PRI_FLD1_LUMA_CSTAT

Address Offset	0x0000 0308		
Physical Address	0x489D D308	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-328. Register Call Summary for Register VPE_PRI_FLD1_LUMA_CSTAT

VPE Functional Description • VPDMA Basic Definitions:
VPE Register Manual • VPE_VPDMA Register Summary: [1]

Table 10-329. VPE_PRI_FLD1_CHROMA_CSTAT

Address Offset	0x0000 030C		
Physical Address	0x489D D30C	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START				LINE_MODE		3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines.							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0

Bits	Field Name	Description	Type	Reset
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:8	LINE_MODE	Selects the output mode of the line buffer. 0: repeat lines twice each output data line gets 2 times the number of frame lines. 1: each line once with Line Buffer Disabled, so no mirroring. Each line gets frame lines with identical data. 2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines. 3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines.	RW	0x0
7:0	3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines.		R	0x0

Table 10-330. Register Call Summary for Register VPE_PRI_FLD1_CHROMA_CSTAT

VPE Functional Description
- Modes of Operation (VPDMA): [0]
- VPDMA Basic Definitions:

VPE Register Manual
- VPE_VPDMA Register Summary: [2]

Table 10-331. VPE_PRI_FLD2_LUMA_CSTAT

Address Offset	0x0000 0310		
Physical Address	0x489D D310	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13 12 11 10	9 8	7 6 5 4 3 2 1 0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED	

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-332. Register Call Summary for Register VPE_PRI_FLD2_LUMA_CSTAT

VPE Functional Description
- VPDMA Basic Definitions:

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-333. VPE_PRI_FLD2_CHROMA_CSTAT

Address Offset	0x0000 0314		
Physical Address	0x489D D314	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13 12 11 10	9 8	7 6 5 4 3 2 1 0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	LINE_MODE	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:8	LINE_MODE	Selects the output mode of the line buffer. 0: repeat lines twice each output data line gets 2 times the number of frame lines. 1: each line once with Line Buffer Disabled, so no mirroring. Each line gets frame lines with identical data. 2: Each line seen once Mirroring is enabled so the top lines get the top lines repeated at the top of the frame and the bottom lines have the bottom lines repeated. Each line of data gets frame lines + number of buffered lines. 3: each line once only on one line. Each data line gets number of frame lines divided by number of buffered lines.	RW	0x0
7:0	RESERVED		R	0x0

Table 10-334. Register Call Summary for Register VPE_PRI_FLD2_CHROMA_CSTAT

VPE Functional Description
- Modes of Operation (VPDMA): [0]
- VPDMA Basic Definitions:

VPE Register Manual
- VPE_VPDMA Register Summary: [2]

Table 10-335. VPE_PRI_MV0_CSTAT

Address Offset	0x0000 0330		
Physical Address	0x489D D330	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-336. Register Call Summary for Register VPE_PRI_MV0_CSTAT

VPE Functional Description
- VPDMA Basic Definitions:

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-337. VPE_PRI_MV_OUT_CSTAT

Address Offset	0x0000 033C		
Physical Address	0x489D D33C	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-338. Register Call Summary for Register VPE_PRI_MV_OUT_CSTAT

VPE Functional Description • VPDMA Basic Definitions:
VPE Register Manual • VPE_VPDMA Register Summary: [1]

Table 10-339. VPE_VIP0_UP_Y_CSTAT

Address Offset	0x0000 0390		
Physical Address	0x489D D390	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13 12 11 10	9 8	7 6 5 4 3 2 1 0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED	

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-340. Register Call Summary for Register VPE_VIP0_UP_Y_CSTAT

VPE Functional Description • VPDMA Basic Definitions:
VPE Register Manual • VPE_VPDMA Register Summary: [1]

Table 10-341. VPE_VIP0_UP_UV_CSTAT

Address Offset	0x0000 0394		
Physical Address	0x489D D394	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-342. Register Call Summary for Register VPE_VIP0_UP_UV_CSTAT

VPE Functional Description
- VPDMA Basic Definitions:

VPE Register Manual
- VPE_VPDMA Register Summary: [1]

Table 10-343. VPE_VPI_CTL_CSTAT

Address Offset	0x0000 03D0		
Physical Address	0x489D D3D0	**Instance**	VPE_VPDMA
Description	The register holds status information and control for the client.		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13 12 11 10	9 8	7 6 5 4 3 2 1 0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED	

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client. 0 : Change in value of hdmi_field_id 1 : Change in value of dvo2_field_id 2 : Change in value of hdcomp_field_id 3 : Change in value of sd_field_id 4 : Use List Manager Internal Field0 5 : Use List Manager Internal Field1 6 : Use List Manager Internal Field2 7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 10-344. Register Call Summary for Register VPE_VPI_CTL_CSTAT

VPE Functional Description • VPDMA Basic Definitions:
VPE Register Manual • VPE_VPDMA Register Summary: [2]

10.4.7 VPE_TOP_LEVEL Registers

10.4.7.1 VPE_TOP_LEVEL Register Summary

Table 10-345. VPE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VPE_TOP_LEVEL Base Address
VPE_CLKC_PID	RW	32	0x0000 0000	0x489D 0000
VPE_SYSCONFIG	RW	32	0x0000 0010	0x489D 0010
VPE_INTC_INTR0_STATUS_RAW0	RW	32	0x0000 0020	0x489D 0020
VPE_INTC_INTR0_STATUS_RAW1	RW	32	0x0000 0024	0x489D 0024

Table 10-345. VPE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VPE_TOP_LEVEL Base Address
VPE_INTC_INTR0_STATUS_ENA0	RW	32	0x0000 0028	0x489D 0028
VPE_INTC_INTR0_STATUS_ENA1	RW	32	0x0000 002C	0x489D 002C
VPE_INTC_INTR0_ENA_SET0	RW	32	0x0000 0030	0x489D 0030
VPE_INTC_INTR0_ENA_SET1	RW	32	0x0000 0034	0x489D 0034
VPE_INTC_INTR0_ENA_CLR0	RW	32	0x0000 0038	0x489D 0038
VPE_INTC_INTR0_ENA_CLR1	RW	32	0x0000 003C	0x489D 003C
VPE_INTC_EOI	RW	32	0x0000 00A0	0x489D 00A0
VPE_CLKC_CLKEN	RW	32	0x0000 0100	0x489D 0100
VPE_CLKC_RST	RW	32	0x0000 0104	0x489D 0104
VPE_CLKC_DPS	RW	32	0x0000 010C	0x489D 010C
VPE_RANGE_MAP	RW	32	0x0000 011C	0x489D 011C

10.4.7.2 VPE_TOP_LEVEL Register Description

Table 10-346. VPE_CLKC_PID

Address Offset	0x0000 0000		
Physical Address	0x489D 0000	**Instance**	VPE_TOP_LEVEL
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID																															

Bits	Field Name	Description	Type	Reset
31:0	PID		R	0x0

Table 10-347. Register Call Summary for Register VPE_CLKC_PID

VPE Register Manual
- VPE_TOP_LEVEL Register Summary: [0]

Table 10-348. VPE_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x489D 0010	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										STANDBYMODE		IDLEMODE		RESERVED	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:4	STANDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state 0x0: Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only 0x1: No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only 0x2: Same behavior as bit-field value of 0x1. 0x3: Reserved	RW	0x2
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state 0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only 0x1: No-idle mode: local target never enters idle state. Backup mode, for debug only 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented	RW	0x2
1:0	RESERVED		R	0x0

Table 10-349. Register Call Summary for Register VPE_SYSCONFIG

VPE Functional Description
- VPE Idle Mode: [0]
- VPE StandBy Mode: [1]

VPE Register Manual
- VPE_TOP_LEVEL Register Summary: [2]

Table 10-350. VPE_INTC_INTR0_STATUS_RAW0

Address Offset	0x0000 0020		
Physical Address	0x489D 0020	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													DEI_FMD_INT_RAW	RESERVED	VPDMA_INT0_DESCRIPTOR_RAW	VPDMA_INT0_LIST7_NOTIFY_RAW	VPDMA_INT0_LIST7_COMPLETE_RAW	VPDMA_INT0_LIST6_NOTIFY_RAW	VPDMA_INT0_LIST6_COMPLETE_RAW	VPDMA_INT0_LIST5_NOTIFY_RAW	VPDMA_INT0_LIST5_COMPLETE_RAW	VPDMA_INT0_LIST4_NOTIFY_RAW	VPDMA_INT0_LIST4_COMPLETE_RAW	VPDMA_INT0_LIST3_NOTIFY_RAW	VPDMA_INT0_LIST3_COMPLETE_RAW	VPDMA_INT0_LIST2_NOTIFY_RAW	VPDMA_INT0_LIST2_COMPLETE_RAW	VPDMA_INT0_LIST1_NOTIFY_RAW	VPDMA_INT0_LIST1_COMPLETE_RAW	VPDMA_INT0_LIST0_NOTIFY_RAW	VPDMA_INT0_LIST0_COMPLETE_RAW

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	DEI_FMD_INT_RAW	DEI Film Mode Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
17	RESERVED		RW	0x0
16	VPDMA_INT0_DESCRIPTOR_RAW	VPDMA INT0 Descriptor Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_RAW	VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_RAW	VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_RAW	VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_RAW	VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_RAW	VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_RAW	VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_RAW	VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_RAW	VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_RAW	VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_RAW	VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_RAW	VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_RAW	VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
3	VPDMA_INT0_LIST1_NOTIFY_RAW	VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_RAW	VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_RAW	VPDMA INT0 List0 Notify Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_RAW	VPDMA INT0 List0 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 10-351. Register Call Summary for Register VPE_INTC_INTR0_STATUS_RAW0

VPE Register Manual
- VPE_TOP_LEVEL Register Summary: [0]

Table 10-352. VPE_INTC_INTR0_STATUS_RAW1

Address Offset	0x0000 0024		
Physical Address	0x489D 0024	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VIP1_CHR_DS_1_UV_ERR_INT_RAW	RESERVED						DEI_ERROR_INT_RAW	RESERVED								VPDMA_INT0_CLIENT_RAW	RESERVED	VPDMA_INT0_CHANNEL_GROUP5_RAW	VPDMA_INT0_CHANNEL_GROUP4_RAW	VPDMA_INT0_CHANNEL_GROUP3_RAW	VPDMA_INT0_CHANNEL_GROUP2_RAW	VPDMA_INT0_CHANNEL_GROUP1_RAW	VPDMA_INT0_CHANNEL_GROUP0_RAW		

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_RAW	VIP1 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
21:17	RESERVED		RW	0x0
16	DEI_ERROR_INT_RAW	DEI Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
15:8	RESERVED		R	0x0
7	VPDMA_INT0_CLIENT_RAW	VPDMA INT0 Client Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		RW	0x0

Bits	Field Name	Description	Type	Reset
5	VPDMA_INT0_CHANNEL_GROUP5_RAW	VPDMA INT0 Channel Group5 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_RAW	VPDMA INT0 Channel Group4 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_RAW	VPDMA INT0 Channel Group3 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_RAW	VPDMA INT0 Channel Group2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_RAW	VPDMA INT0 Channel Group1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_RAW	VPDMA INT0 Channel Group0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 10-353. Register Call Summary for Register VPE_INTC_INTR0_STATUS_RAW1

VPE Register Manual
- VPE_TOP_LEVEL Register Summary: [0]

Table 10-354. VPE_INTC_INTR0_STATUS_ENA0

Address Offset	0x0000 0028		
Physical Address	0x489D 0028	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													DEI_FMD_INT_ENA	RESERVED	VPDMA_INT0_DESCRIPTOR_ENA	VPDMA_INT0_LIST7_NOTIFY_ENA	VPDMA_INT0_LIST7_COMPLETE_ENA	VPDMA_INT0_LIST6_NOTIFY_ENA	VPDMA_INT0_LIST6_COMPLETE_ENA	VPDMA_INT0_LIST5_NOTIFY_ENA	VPDMA_INT0_LIST5_COMPLETE_ENA	VPDMA_INT0_LIST4_NOTIFY_ENA	VPDMA_INT0_LIST4_COMPLETE_ENA	VPDMA_INT0_LIST3_NOTIFY_ENA	VPDMA_INT0_LIST3_COMPLETE_ENA	VPDMA_INT0_LIST2_NOTIFY_ENA	VPDMA_INT0_LIST2_COMPLETE_ENA	VPDMA_INT0_LIST1_NOTIFY_ENA	VPDMA_INT0_LIST1_COMPLETE_ENA	VPDMA_INT0_LIST0_NOTIFY_ENA	VPDMA_INT0_LIST0_COMPLETE_ENA

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	DEI_FMD_INT_ENA	DEI Film Mode Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA	VPDMA INT0 Descriptor Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
15	VPDMA_INT0_LIST7_NOTIFY_ENA	VPDMA INT0 List7 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA	VPDMA INT0 List7 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA	VPDMA INT0 List6 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_ENA	VPDMA INT0 List6 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA	VPDMA INT0 List5 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA	VPDMA INT0 List5 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA	VPDMA INT0 List4 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA	VPDMA INT0 List4 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_ENA	VPDMA INT0 List3 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA	VPDMA INT0 List3 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA	VPDMA INT0 List2 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA	VPDMA INT0 List2 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA	VPDMA INT0 List1 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA	VPDMA INT0 List1 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA	VPDMA INT0 List0 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA	VPDMA INT0 List0 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 10-355. Register Call Summary for Register VPE_INTC_INTR0_STATUS_ENA0

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-356. VPE_INTC_INTR0_STATUS_ENA1

Address Offset	0x0000 002C		
Physical Address	0x489D 002C	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									VIP1_CHR_DS_1_UV_ERR_INT_ENA	RESERVED					DEI_ERROR_INT_ENA	RESERVED								VPDMA_INT0_CLIENT_ENA	RESERVED	VPDMA_INT0_CHANNEL_GROUP5_ENA	VPDMA_INT0_CHANNEL_GROUP4_ENA	VPDMA_INT0_CHANNEL_GROUP3_ENA	VPDMA_INT0_CHANNEL_GROUP2_ENA	VPDMA_INT0_CHANNEL_GROUP1_ENA	VPDMA_INT0_CHANNEL_GROUP0_ENA	

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA	VIP1 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
21:17	RESERVED		RW	0x0
16	DEI_ERROR_INT_ENA	DEI Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
15:8	RESERVED		RW	0x0
7	VPDMA_INT0_CLIENT_ENA	VPDMA INT0 Client Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		RW	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA	VPDMA INT0 Channel Group5 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA	VPDMA INT0 Channel Group4 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA	VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA	VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA	VPDMA INT0 Channel Group1 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
0	VPDMA_INT0_CHANNEL_GROUP0_ENA	VPDMA INT0 Channel Group0 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 10-357. Register Call Summary for Register VPE_INTC_INTR0_STATUS_ENA1

VPE Register Manual
- VPE_TOP_LEVEL Register Summary: [0]

Table 10-358. VPE_INTC_INTR0_ENA_SET0

Address Offset	0x0000 0030		
Physical Address	0x489D 0030	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													DEI_FMD_INT_ENA_SET	RESERVED	VPDMA_INT0_DESCRIPTOR_ENA_SET	VPDMA_INT0_LIST7_NOTIFY_ENA_SET	VPDMA_INT0_LIST7_COMPLETE_ENA_SET	VPDMA_INT0_LIST6_NOTIFY_ENA_SET	VPDMA_INT0_LIST6_COMPLETE_ENA_SET	VPDMA_INT0_LIST5_NOTIFY_ENA_SET	VPDMA_INT0_LIST5_COMPLETE_ENA_SET	VPDMA_INT0_LIST4_NOTIFY_ENA_SET	VPDMA_INT0_LIST4_COMPLETE_ENA_SET	VPDMA_INT0_LIST3_NOTIFY_ENA_SET	VPDMA_INT0_LIST3_COMPLETE_ENA_SET	VPDMA_INT0_LIST2_NOTIFY_ENA_SET	VPDMA_INT0_LIST2_COMPLETE_ENA_SET	VPDMA_INT0_LIST1_NOTIFY_ENA_SET	VPDMA_INT0_LIST1_COMPLETE_ENA_SET	VPDMA_INT0_LIST0_NOTIFY_ENA_SET	VPDMA_INT0_LIST0_COMPLETE_ENA_SET
R-0													RW-0	R-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0	RW-0

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	DEI_FMD_INT_ENA_SET	DEI Film Mode Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA_SET	VPDMA INT0 Descriptor Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA_SET	VPDMA INT0 List7 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA_SET	VPDMA INT0 List7 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA_SET	VPDMA INT0 List6 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
12	VPDMA_INT0_LIST6_COMPLETE_ENA_SET	VPDMA INT0 List6 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA_SET	VPDMA INT0 List5 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA_SET	VPDMA INT0 List5 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA_SET	VPDMA INT0 List4 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA_SET	VPDMA INT0 List4 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_ENA_SET	VPDMA INT0 List3 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA_SET	VPDMA INT0 List3 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA_SET	VPDMA INT0 List2 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA_SET	VPDMA INT0 List2 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA_SET	VPDMA INT0 List1 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA_SET	VPDMA INT0 List1 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA_SET	VPDMA INT0 List0 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA_SET	VPDMA INT0 List0 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 10-359. Register Call Summary for Register VPE_INTC_INTR0_ENA_SET0

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-360. VPE_INTC_INTR0_ENA_SET1

Address Offset	0x0000 0034		
Physical Address	0x489D 0034	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	RESERVED						DEI_ERROR_INT_ENA_SET	RESERVED								VPDMA_INT0_CLIENT_ENA_SET	RESERVED	VPDMA_INT0_CHANNEL_GROUP5_ENA_SET	VPDMA_INT0_CHANNEL_GROUP4_ENA_SET	VPDMA_INT0_CHANNEL_GROUP3_ENA_SET	VPDMA_INT0_CHANNEL_GROUP2_ENA_SET	VPDMA_INT0_CHANNEL_GROUP1_ENA_SET	VPDMA_INT0_CHANNEL_GROUP0_ENA_SET

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP1 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
21:17	RESERVED		R	0x0
16	DEI_ERROR_INT_ENA_SET	DEI Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15:8	RESERVED		RW	0x0
7	VPDMA_INT0_CLIENT_ENA_SET	VPDMA INT0 Client Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA_SET	VPDMA INT0 Channel Group5 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA_SET	VPDMA INT0 Channel Group4 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA_SET	VPDMA INT0 Channel Group3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA_SET	VPDMA INT0 Channel Group2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA_SET	VPDMA INT0 Channel Group1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
0	VPDMA_INT0_CHANNEL_GROUP0_ENA_SET	VPDMA INT0 Channel Group0 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 10-361. Register Call Summary for Register VPE_INTC_INTR0_ENA_SET1

VPE Register Manual
- VPE_TOP_LEVEL Register Summary: [0]

Table 10-362. VPE_INTC_INTR0_ENA_CLR0

Address Offset	0x0000 0038		
Physical Address	0x489D 0038	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													DEI_FMD_INT_ENA_CLR	RESERVED	VPDMA_INT0_DESCRIPTOR_ENA_CLR	VPDMA_INT0_LIST7_NOTIFY_ENA_CLR	VPDMA_INT0_LIST7_COMPLETE_ENA_CLR	VPDMA_INT0_LIST6_NOTIFY_ENA_CLR	VPDMA_INT0_LIST6_COMPLETE_ENA_CLR	VPDMA_INT0_LIST5_NOTIFY_ENA_CLR	VPDMA_INT0_LIST5_COMPLETE_ENA_CLR	VPDMA_INT0_LIST4_NOTIFY_ENA_CLR	VPDMA_INT0_LIST4_COMPLETE_ENA_CLR	VPDMA_INT0_LIST3_NOTIFY_ENA_CLR	VPDMA_INT0_LIST3_COMPLETE_ENA_CLR	VPDMA_INT0_LIST2_NOTIFY_ENA_CLR	VPDMA_INT0_LIST2_COMPLETE_ENA_CLR	VPDMA_INT0_LIST1_NOTIFY_ENA_CLR	VPDMA_INT0_LIST1_COMPLETE_ENA_CLR	VPDMA_INT0_LIST0_NOTIFY_ENA_CLR	VPDMA_INT0_LIST0_COMPLETE_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	DEI_FMD_INT_ENA_CLR	DEI Film Mode Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA_CLR	VPDMA INT0 Descriptor Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA_CLR	VPDMA INT0 List7 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA_CLR	VPDMA INT0 List7 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA_CLR	VPDMA INT0 List6 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
12	VPDMA_INT0_LIST6_COMPLETE_ENA_CLR	VPDMA INT0 List6 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA_CLR	VPDMA INT0 List5 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA_CLR	VPDMA INT0 List5 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA_CLR	VPDMA INT0 List4 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA_CLR	VPDMA INT0 List4 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_ENA_CLR	VPDMA INT0 List3 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA_CLR	VPDMA INT0 List3 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA_CLR	VPDMA INT0 List2 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA_CLR	VPDMA INT0 List2 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA_CLR	VPDMA INT0 List1 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA_CLR	VPDMA INT0 List1 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA_CLR	VPDMA INT0 List0 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA_CLR	VPDMA INT0 List0 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 10-363. Register Call Summary for Register VPE_INTC_INTR0_ENA_CLR0

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-364. VPE_INTC_INTR0_ENA_CLR1

Address Offset	0x0000 003C		
Physical Address	0x489D 003C	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	RESERVED					DEI_ERROR_INT_ENA_CLR	RESERVED								VPDMA_INT0_CLIENT_ENA_CLR	RESERVED	VPDMA_INT0_CHANNEL_GROUP5_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP4_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP3_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP2_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP1_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP0_ENA_CLR	

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP1 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
21:17	RESERVED		R	0x0
16	DEI_ERROR_INT_ENA_CLR	DEI Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
15:8	RESERVED		RW	0x0
7	VPDMA_INT0_CLIENT_ENA_CLR	VPDMA INT0 Client Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA_CLR	VPDMA INT0 Channel Group5 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA_CLR	VPDMA INT0 Channel Group4 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA_CLR	VPDMA INT0 Channel Group3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA_CLR	VPDMA INT0 Channel Group2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA_CLR	VPDMA INT0 Channel Group1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
0	VPDMA_INT0_CHANNEL_GROUP0_ENA_CLR	VPDMA INT0 Channel Group0 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 10-365. Register Call Summary for Register VPE_INTC_INTR0_ENA_CLR1

VPE Register Manual
- VPE_TOP_LEVEL Register Summary: [0]

Table 10-366. VPE_INTC_EOI

Address Offset	0x0000 00A0		
Physical Address	0x489D 00A0	**Instance**	VPE_TOP_LEVEL
Description	INTC EOI Register. This register contains the EOI vector register contents as defined by HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOI_VECTOR																															

Bits	Field Name	Description	Type	Reset
31:0	EOI_VECTOR	Number associated with the ipgenericirq for intr output. There are 4 interrupt outputs: 0x0 : Write to intr0 IP Generic 0x1 : Write to intr1 IP Generic 0x2 : Write to intr2 IP Generic 0x3 : Write to intr3 IP Generic Any other write value is ignored.	RW	0x0

Table 10-367. Register Call Summary for Register VPE_INTC_EOI

VPE Register Manual
- VPE_TOP_LEVEL Register Summary: [0]

Table 10-368. VPE_CLKC_CLKEN

Address Offset	0x0000 0100		
Physical Address	0x489D 0100	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														PRIM_DP_EN	VPDMA_EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	PRIM_DP_EN	Primary Video Data Path Clock Enable 0x0 : Clock Disabled 0x1 : Clock Enabled	RW	0x0
0	VPDMA_EN	VPDMA Clock Enable 0x0 : Clock Disabled 0x1 : Clock Enabled	RW	0x0

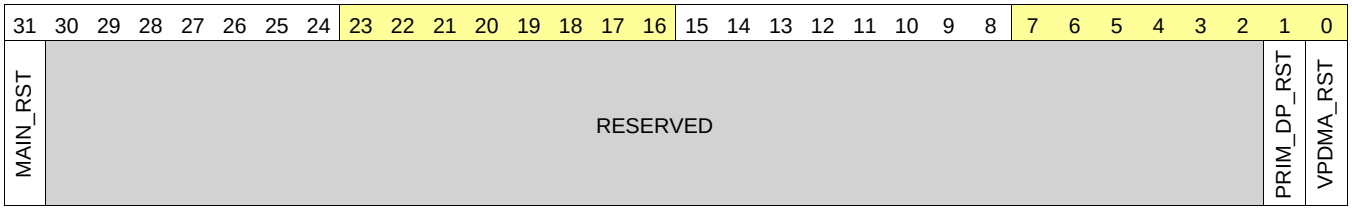
Table 10-369. Register Call Summary for Register VPE_CLKC_CLKEN

VPE Functional Description
- VPE Clocks: [0] [1]

VPE Register Manual
- VPE_TOP_LEVEL Register Summary: [2]

Table 10-370. VPE_CLKC_RST

Address Offset	0x0000 0104		
Physical Address	0x489D 0104	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31	MAIN_RST	Reset for entire data path in VPE0	RW	0x0
30:2	RESERVED		R	0x0
1	PRIM_DP_RST	Primary Video Data Path Reset	RW	0x0
0	VPDMA_RST	VPDMA Reset	RW	0x0

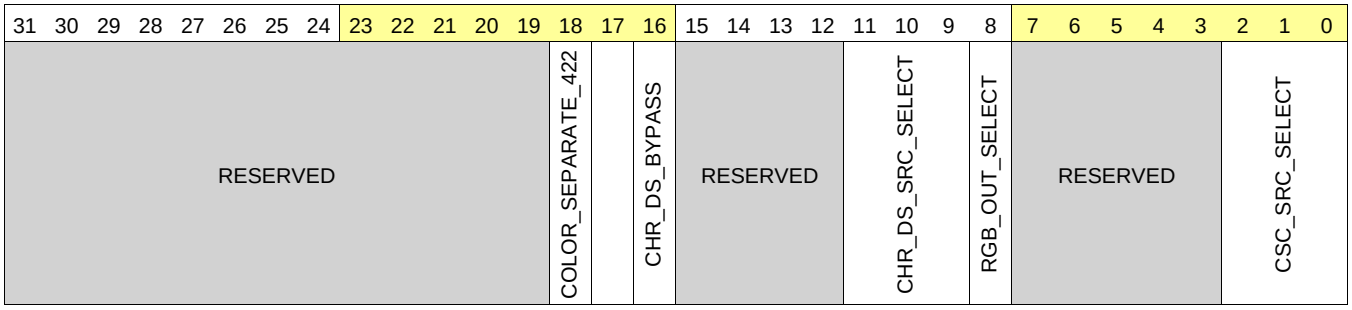
Table 10-371. Register Call Summary for Register VPE_CLKC_RST

VPE Functional Description
- VPE Software Reset: [0] [1]

VPE Register Manual
- VPE_TOP_LEVEL Register Summary: [2]

Table 10-372. VPE_CLKC_DPS

Address Offset	0x0000 010C		
Physical Address	0x489D 010C	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	COLOR_SEPARATE_422	422 Color Separate Select 0x0 : 422 output will be 16 bit interleaved (YCbCr) 0x1 : 422 output will be 8 bit coplanar (Y, CbCr) This bit controls whether 422 output will be color separate or interleaved. This bit only applies IF chr_ds_bypass is 1 (means 422 output, not 420) and rgb_out_select is 0 (means 422 output, not RGB or 444). 420 is always coplanar, so this only applies if the output type is 422.	RW	0x0
17	RESERVED		R	0x0
16	CHR_DS_BYPASS	Chroma Downsampler Bypass 0x0 : Chroma Downsampler selected 0x1 : Chroma Downsampler Bypassed Chroma Downsampler Bypassed means the output format from VPE0 will be 422 data. Selected means the output format will be 420. This bit is only applicable if rgb_out_select is 0. It is a don't care if rgb_out_select is 1.	RW	0x0
15:12	RESERVED		R	0x0
11:9	CHR_DS_SRC_SELECT	Chroma Downsampler Source Select 000 : Path Disabled (no input to CHR_DS) 001 : Reserved (Path Disabled) 010 : Reserved (Path Disabled) 011 : Reserved (Path Disabled) 100 : Reserved (Path Disabled) 101 : Source from DEI Scaler (422) 110 : Reserved (Path Disabled) 111 : Reserved (Path Disabled)	RW	0x0
8	RGB_OUT_SELECT	RGB Output Select 0x0 : Output Type is 420/422 0x1 : Output Type is RGB/444	RW	0x0
7:3	RESERVED		R	0x0
2:0	CSC_SRC_SELECT	CSC Source Select: 000 : Path Disabled 001 : Reserved (Path Disabled) 010 : Reserved (Path Disabled) 011 : Source from DEI Scaler (422) 100 : Reserved (Path Disabled) 101 : Reserved (Path Disabled) 110 : Reserved (Path Disabled) 111 : Reserved (Path Disabled)	RW	0x0

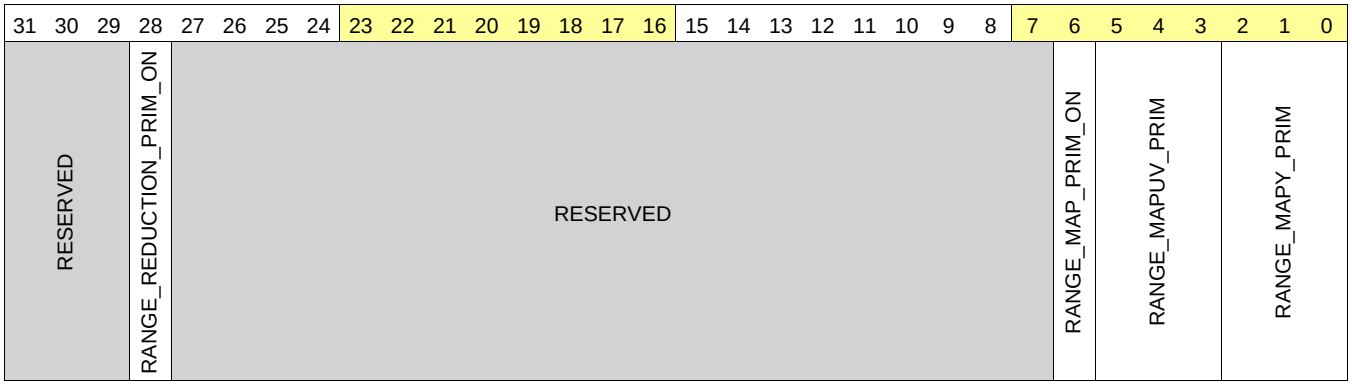
Table 10-373. Register Call Summary for Register VPE_CLKC_DPS

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]

Table 10-374. VPE_RANGE_MAP

Address Offset	0x0000 011C		
Physical Address	0x489D 011C	**Instance**	VPE_TOP_LEVEL
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	RANGE_REDUCTION_PRIM_ON	Range Reduction ON for Primary input	RW	0x0
27:7	RESERVED		R	0x0
6	RANGE_MAP_PRIM_ON	Range Mapping ON for Primary input	RW	0x0
5:3	RANGE_MAPUV_PRIM	Range Map UV for Primary input	RW	0x0
2:0	RANGE_MAPY_PRIM	Range Map Y for Primary input	RW	0x0

Table 10-375. Register Call Summary for Register VPE_RANGE_MAP

VPE Register Manual

- VPE_TOP_LEVEL Register Summary: [0]