Open Graphics Programming Manual

Chrome9 HC3 Graphics Processor

VX800 / VX820 Series

Part II: 3D / Video

Preliminary Revision 1.0
November 19, 2008

VIA TECHNOLOGIES, INC.

Copyright Notice:

Trademark Notices:

Disclaimer Notice:

Offices:

VIA Technologies Incorporated
Taiwan Office:
1st Floor, No. 531
Chung-Cheng Road, Hsin-Tien
Taipei, Taiwan ROC
Tel: 886-2-2218-5452
FAX: 886-2-2218-5453
Home page: http://www.via.com.tw

VIA Technologies Incorporated
USA Office:
940 Mission Court
Fremont, CA 94539
USA
Tel: 510-683-3300
FAX: 510-683-3301 or 510-687-4654
Home Page: http://www.viatech.com

REVISION HISTORY

Document Release	Date	Revision	Initials
1.0	11/19/08	Initial public release	LW

TABLE OF CONTENTS

LIST OF TABLES

INTRODUCTION

This document contains detailed graphics registers descriptions and other general information for the Chrome9 HC3 graphic engine. The graphics registers for the Chrome9 HC3 main features and its underlying subsystems are described explicitly in the following chapters.

About This Programming Guide

The programming manual is organized into 2 volumes (Part I & Part II). A brief description of each chapter is given below:

Part I:

Introduction
An overview of the Chrome9 HC3 design features is given in this chapter, along with block diagram and reference list.

Register Overview
Register specifications for register addressing and I/O space division are shown in this chapter.

PCI Interface Register Descriptions
PCI interface summary table and detailed register descriptions are presented in this chapter.

VGA I/O Register Descriptions
This chapter provides detailed VGA-related register summary and descriptions. The various video modes support by the Chrome9 HC3 controller are also included in the configuration section.

2D Engine Register Descriptions
In this chapter provides detailed 2D Engine register summary and descriptions.

DMA Register Descriptions
In this chapter provides detailed DMA register summary and descriptions.

CBU Rotation Register Descriptions
In this chapter provides detailed CBU rotation register summary and descriptions.

LVDS Register Descriptions
In this chapter provides detailed LVDS register summary and descriptions.

Part II:

Video Register Descriptions
This chapter provides detailed video register summary and descriptions.

3D Engine Register Descriptions
In this chapter provides detailed 3D Engine register summary and descriptions.

VIDEO REGISTERS

This chapter provides detailed video register summary table. Register descriptions on video play back, blending, engine capture and high quality video registers are provided in the sequent sections.

Video Registers

The Chrome9 HC3 Graphic Engine has an integrated video playback and blending engine. The video playback engine can simultaneously support up to three live video windows, and each video window can be independently scaled. Two blending engines support the graphics-to-video blending, followed by video-to-video blending effect. The blending factor is from constant, alpha stream or is combined with graphics data. Table 1 summarizes the video playback and blending engine registers. Detail register description follows.

These video register tables document the I/O port, I/O index and attribute ("Attribute") for each register. Attribute definitions being used are RW (Read/Write), RO (Read/Only), and RWC (or just WC) (Read / Write 1's to Clear individual bits). Registers indicated as RW may have some read/only bits that always read back a fixed value (usually 0 if unused); registers designated as RWC or WC may have some read-only or read write bits (see individual register descriptions for details). All offset and default values are shown in hexadecimal unless otherwise indicated.

Table 1. Video Display Engine Registers

Offset	Register Name	Attribute
Video Related Engines Register Space 1 (0x00000200 ~ 0x000003FF)		
203-200	Interrupt Flags & Masks Control	RW
207-204	Address Flip Status	RO
20B-208	Alpha Window / HI (For Second Display) Horizontal and Vertical Location Start	RW
20F-20C	Alpha Window Horizontal and Vertical End & HI (For Second Display) Center Offset	RW
213-210	Alpha Window Control	RW
217-214	CRT Starting Address	RW
21B-218	The Second Display Starting Address	RW
21F-21C	Alpha Stream Frame Buffer Stride	RW
223-220	Primary Display Color Key	RW
227-224	Alpha Window & HI (For Second Display) Frame Buffer Starting Address	RW
23B-228	Chroma Key Lower Bound	RW
22F-22C	Chroma Key Upper Bound	RW
233-230	Video Stream 1 Control	RW
237-234	Video Window 1 Fetch Count	RW
23B-238	Video Window 1 Frame Buffer Y Starting Address 1	RW
23F-23C	Video Window 1 Frame Buffer Stride	RW
243-240	Video Window 1 Horizontal and Vertical Start Location	RW
247-244	Video Window 1 Horizontal and Vertical Ending Location	RW
24B-248	Video Window 1 Frame Buffer Y Starting Address 2	RW
24F-24C	Video Window 1 Display Zoom Control	RW
253-250	Video Window 1 Minify and Interpolation Control	RW
257-254	Video Window 1 Frame Buffer Y Starting Address 0	RW
25B-258	Video 1 FIFO Depth and Threshold Control	RW
25F-25C	Video Window 1 Horizontal and Vertical Starting Location Offset	RW
263-260	HI Control For Second Display	RW
267-264	The Second Display Color Key	RW
26B-268	V3 and Alpha Window FIFO Pre-threshold Control	RW
26F-26C	Video Window 1 Display Count On Screen Control	RW
273-270	HI Transparent Color For Second Display	RW

277-274	HI Inverse Color For Second Display	RW
27B-278	V3 and Alpha Window FIFO Depth and Threshold Control	RW
27F-27C	V3 Display Count On Screen Control	RW
283-280	Primary Display Second Color Key	RW
287-284	V1 Color Space Conversion & Enhancement Control 1	RW
28B-288	V1 Color Space Conversion & Enhancement Control 2	RW
28F-28C	P Signature Adder Result 1	RW
293-290	Alpha Window/Color Cursor Ending Position (For Primary Display)	RW
297-294	3D AGP Pause Address MMIO Port	WO
29B-298	Compose Output Modes Select	RW
29F-29C	V3 Frame Buffer Starting Address 2	RW
2A3-2A0	V3 Control	RW
2A7-2A4	V3 Frame Buffer Starting Address 0	RW
2AB-2A8	V3 Frame Buffer Starting Address 1	RW
2AF-2AC	V3 Frame Buffer Stride	RW
2B3-2B0	V3 Horizontal and Vertical Start	RW
2B7-2B4	V3 Horizontal and Vertical End	RW
2BB-2B8	V3 and Alpha Window Fetch Count	RW
2BF-2BC	V3 Display Zoom Control	RW
2C3-2C0	V3 Minify & Interpolation Control	RW
2C7-2C4	V3 Color Space Conversion & Enhancement Control 1	RW
2CB-2C8	V3 Color Space Conversion & Enhancement Control 2	RW
2CF-2CC	T Signature Adder Result 1	RW
2D3-2D0	Graphics Hardware Cursor Mode Control	RW
2D7-2D4	Graphics Hardware Cursor Position	RW
2DB-2D8	Graphics Hardware Cursor Origin	RW
2DF-2DC	Graphics Hardware Cursor Background Color	RW
2E3-2E0	Graphics Hardware Cursor Foreground Color	RW
2E7-2E4	T Signature Data Result 1	RW
2EB-2E8	HI for Primary Display FIFO Control Signal	RW
2EF-2EC	HI for Primary Display Transparent color	RW
2F3-2F0	HI for Primary Display Control Signal	RW
2F7-2F4	HI for Primary Display Frame Buffer Starting Address	RW
2FB-2F8	HI for Primary Display Horizontal and Vertical Start	RW
2FF-2FC	HI for Primary Display Center Offset	RW
Video Related Engines Register Space 2 (0x00001200 ~ 0x000013FF)		
1203-1200	Video 1Gamma R Correction Control	RW
1207-1204	Video 1Gamma G Correction Control	RW
120B-1208	Video 1Gamma B Correction Control	RW
120F-120C	HI for Primary Display Inverse Color	RW
1213-1210	PCIe Shadow Register 1	RW
1217-1214	PCIe Shadow Register 2	RW
121B-1218	PCIe Shadow Register 3	RW
121F-121C	PCIe Shadow Register 4	RW
1223-1220	Video 3 Gamma R Correction Control	RW
1227-1224	Video 3 Gamma G Correction Control	RW
122B-1228	Video 3 Gamma B Correction Control	RW
122F-122C	Video 3 Position Offset	RW
127C-1230	Reserved	RO
1283-1280	Interrupt Flags and Masks Control	RW
1287-1284	Logic Signature Setting	RW
128B-1288	P Logic Signature Address Result 0	RW

128F-128C	T Logic Signature Address Result 0	RW
1293-1290	IGA1 Display Position Counter 0	RO
1297-1294	IGA1 Display Position Counter 1	RO
129B-1298	IGA1 Display Position Counter 2	RW
129F-129C	T Logic Signature Data Result 0	RW
12A3-12A0	IGA2 Display Position Counter 0	RO
12A7-12A4	IGA2 Display Position Counter 1	RO
12AB-12A8	IGA2 Display Position Counter 2	RW
12B3-12B0	Primary Display Data Color Space Conversion and Enhancement Control 1	RW
12B7-12B4	Primary Display Data Color Space Conversion and Enhancement Control 2	RW
12BB-12B8	Primary Display Data Color Space Conversion and Enhancement Control 3	RW
12BF-12BC	Primary Display Data Color Space Conversion and Enhancement Control 4	RW
12F0-12C0	Reserved	RO
Extended Video Engines Register Space 2 (0x00003200 ~ 0x000033FF)		
3260	Video ID Control	
326C	Video Wait Control Register	

Note: 1) Port Address: MB1 + Offset Address

MB1 is declared in the register with offset address 18h~1Fh in the PCI configuration space.

2) There is additional register space to match the above register definition. When write a register to this space, it will be sent to "command regulator" first and then pass to video display engine. However, register read will be same as the original action. The relationship between the additional register space and original register space is:

(The additional register address) = (The original register address) + 16'h2000.

Table 2. Video Capture Engine and High Quality Video Registers

Offset	Register Name	Attribute
Video Capture Engine Register		
303-300	Capture Interrupt Control and Flags	RW
307-304	Reserved	RO
30B-308	Transport Stream Control	RW
30F-30C	Reserved	RO
313-310	Capture Interface Control	RW
317-314	Active Video Horizontal Range (CCIR601 only)	RW
31B-318	Active Video Vertical Range (CCIR601 only)	RW
31F-31C	Active Video Scaling Control	RW
323-320	VBI Data Horizontal Range	RW
327-324	VBI Data Vertical Range	RW
32B-328	First VBI Buffer Starting Address	RW
32F-32C	VBI Buffer Stride	RW
333-330	Ancillary Data Count Setting	RW
337-334	Maximum Data Count of Active Video	RW
33B-338	Maximum Data Count of VBI or ANC	RW
33F-33C	Capture Data Count	RO
343-340	First Active Video Frame Buffer Starting Address	RW
347-344	Second Active Video Frame Buffer Starting Address	RW
34B-348	Third Active Video Frame Buffer Starting Address	RW
34F-34C	Second VBI Buffer Starting Address	RW
353-350	Stride of Active Video Buffer and Coring Function Control	RW
357-354	TS Buffer 0 Error Packet Indicator	RO
35B-358	TS Buffer 1 Error Packet Indicator	RO
35F-35C	TS Buffer 2 Error Packet Indicator	RO
360-37C	Reserved	RO

Offset	Register Name	Attribute
HQV (High Quality Video) Engine Registers		
383-380	HQV Source Data Offset Control 1	RW
387-384	HQV Source Data Offset Control 2	RW
38B-388	HQV Source Data Offset Control 3	RW
38F-38C	HQV Source Data Offset Control 4	RW
393-390	HQV Parameters of Hardware Tuning Performance/Quality	RW
397-394	HQV Extended Control	RW
39B-398	HQV Static Record Frame Buffer Starting Address	RW
39F-39C	HQV Static Record Frame Buffer Stride	RW
3A3-3A0	HQV Color Adjustment Control 1	RW
3A7-3A4	HQV Color Adjustment Control 2	RW
3AB-3A8	HQV Color Adjustment Control 3	RW
3AF-3AC	HQV Color Adjustment Control 4	RW
3B3-3B0	HQV Horizontal Scale Control	RW
3B7-3B4	HQV Vertical Scale Control	RW
3BB-3B8	HQV Default Video Color	RW
3BF-3BC	HQV De-blocking Factor	RW
3C3-3C0	HQV Sub-picture Frame Buffer Stride and Control	RW
3C7-3C4	HQV Sub-picture Frame Buffer Starting Address	RW
3CB-3C8	HQV Sub-picture 4 x 16 RAM Table Write Control	RW
3D3-3D0	HQV Stream Control and Status	RW
3D7-3D4	HQV SW Source Data –Luma or Packed Starting Address	RW
3DB-3D8	HQV SW Source Data – Chroma Starting Address	RW
3DF-3DC	HQV Linear/Tile Address Mode, Color Space Conversion, Gamma and De-blocking Control	RW
3E3-3E0	HQV Source Data Line Count and Fetch Count Per Line	RW
3E7-3E4	HQV Motion Adaptive De-interlace Control & Threshold	RW
3EF-3EC	HQV Destination Frame Buffer Starting Address 0	RW
3F3-3F0	HQV Destination Frame Buffer Starting Address 1	RW
3F7-3F4	HQV Destination Frame Buffer Stride	RW
3FB-3F8	HQV Source Frame Buffer Stride	RW
3FF-3FC	HQV Destination Data Starting Address 2	RW

Offset	Register Name	Attribute
Second Capture Engine Registers (Refer to Rx300-37C register descriptions for detail.)		
1303-1300	Capture Interrupt Control and Flags	RW
1307-1304	Reserved	RO
130B-1308	Transport Stream Control	RW
130F-130C	Reserved	RO
1313-1310	Capture Interface Control	RW
1317-1314	Active Video Horizontal Range (CCIR601 only)	RW
131B-1318	Active Video Vertical Range (CCIR601 only)	RW
131F-131C	Active Video Scaling Control	RW
1323-1320	VBI Data Horizontal Range	RW
1327-1324	VBI Data Vertical Range	RW
132B-1328	First VBI Buffer Starting Address	RW
132F-132C	VBI Buffer Stride	RW
1333-1330	Ancillary Data Count Setting	RW
1337-1334	Maximum Data Count of Active Video	RW
133B-1338	Maximum Data Count of VBI or ANC	RW
133F-133C	Capture Data Count	RO
1343-1340	First Active Video Frame Buffer Starting Address	RW
1347-1344	Second Active Video Frame Buffer Starting Address	RW
134B-1348	Third Active Video Frame Buffer Starting Address	RW
134F-134C	Second VBI Buffer Starting Address	RW
1353-1350	Stride of Active Video Buffer and Coring Function Control	RW
1357-1354	TS Buffer0 Error Packet Indicator	RO
135B-1358	TS Buffer1 Error Packet Indicator	RO
135F-135C	TS Buffer2 Error Packet Indicator	RO
1360-137C	Reserved	RO

Offset	Register Name	Attribute
Second HQV Engine Registers (Refer to Rx380-3FF register descriptions for detail.)		
1380-13B8	Reserved	RO
13BF-13BC	De-blocking Factor	RW
13C3-13C0	Subpicture Frame Buffer Stride and Control	RW
13C7-13C4	Subpicture Frame Buffer Starting Address	RW
13CB-13C8	Subpicture 4 X 16 RAM Table Write Control	RW
13CF-13CC	HQV Source Data Offset Control	RW
13D3-13D0	HQV Stream Control and Status	RW
13D7-13D4	HQV SW Source Data –Luma or Packed Starting Address	RW
13DB-13D8	HQV SW Source Data – Chroma Starting Address	RW
13DF-13DC	HQV Linear/Tile Address Mode and Color Space Conversion First Control	RW
13E3-13E0	HQV Source Data Line Count and Fetch Count Per Line	RW
13E7-13E4	HQV Motion Adaptive De-interlace Control and Threshold	RW
13EB-13E8	HQV Scale Control	RW
13EF-13EC	HQV Destination Data Starting Address 0	RW
13F3-13F0	HQV Destination Data Starting Address 1	RW
13F7-13F4	HQV Destination Frame Buffer Stride	RW
13FB-13F8	HQV Source Frame Buffer Stride	RW
13FF-13FC	HQV Destination Data Starting Address 2	RW

Note:1) Port Address: MB1 + Offset Address

MB1 is declared in the register with offset address 18h~1Fh in the PCI configuration space.

2) There is additional register space to match the above register definition. When write a register to this space, it will be sent to “command regulator” first and then pass to video display engine. However, register read will be same as the original action. The relationship between the additional register space and original register space is

(The additional register address) = (The original register address) + 16’h2000

Video Display Engine Register Descriptions (200-12F0h)

Offset Address: 203-200h

Interrupt Flags and Masks Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**Interrupt Enable** 0: Disable 1: Enable
30	RW	0	**LVDS Sense Interrupt Enable** 0: Disable 1: Enable
29	RW	0	**Capture 0 VBI Capture End Interrupt Enable** 0: Disable 1: Enable
28	RW	0	**Capture 0 Active Video Data Capture End Enable** 0: Disable 1: Enable
27	RW1C	0	**LVDS Sense Interrupt Status**
26	RW	0	**Capture 1 VBI Capture End Interrupt Enable** 0: Disable 1: Enable
25	RW	0	**First HQV Engine Interrupt Enable** 0: Disable 1: Enable (Refer to 03D0h)
24	RW	0	**Capture 1 Active Video Data Capture End Enable** 0: Disable 1: Enable
23	RW	0	**DMA1 Transfer Done Interrupt Enable** 0: Disable 1: Enable
22	RW	0	**DMA1 Descriptor Done Interrupt Enable** 0: Disable 1: Enable
21	RW	0	**DMA0 Transfer Done Interrupt Enable** 0: Disable 1: Enable
20	RW	0	**DMA0 Descriptor Done Interrupt Enable** 0: Disable 1: Enable
19	RW	0	**VGA VSYNC Interrupt Mask Enable** 0: Disable 1: Enable
18	RW	0	**MC Complete Frame Interrupt Mask Enable** 0: Disable 1: Enable
17	RW	0	**Secondary Display VSYNC Interrupt Enable** 0: Disable 1: Enable
16	RW	0	**Reserved**
15	RW1C	0	**Secondary Display VSYNC Interrupts Status**
14	RW1C	0	**Capture 1 VBI Capture End Interrupt Status**
13	RW1C	0	**Capture 0 VBI Capture End Interrupt Status**
12	RW1C	0	**Capture 0 Active Video Data Capture End Interrupt Status**
11	RW	0	**Second HQV Engine Interrupt Enable** 0: Disable 1: Enable (Refer to 13D0h for more detail.)
10	RW	0	**Second HQV Engine Interrupt Status** (Refer to 13D0h for more detail.)
9	RW1C	0	**First HQV Engine Interrupt Status** (Refer to 03D0h for more detail.)
8	RW1C	0	**Capture 1 Active Video Data Capture End Interrupt Status**
7	RW1C	0	**DMA 1 Transfer Done Interrupt Status**
6	RW1C	0	**DMA 1 Descriptor Done Interrupt Status**
5	RW1C	0	**DMA 0 Transfer Done Interrupt Status**
4	RW1C	0	**DMA 0 Descriptor Done Interrupt Status**
3	RW1C	0	**VGA VSYNC Interrupt Status**
2	RW1C	0	**MC Complete Frame Interrupt Status**
1	RO	0	**Vertical Blanking Status**
0	RW1C	0	**Reserved**

Offset Address: 207-204h

Address Flip Status **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:12	RO	0	**Reserved**
11	RO	0	**Video Window 1 SW Flip Status (R)** Write B0+254'h port to clear this bit.
10	RO	0	**Video Window 3 SW Flip Status (R)** Write B0+2A4'h port to clear this bit.
9:6	RO	0	**Reserved**
5	RW1C	0	**CR Interrupt Status**
4	RO	0	**Alpha Window Starting Address Update Status** 0: Updated 1: Not yet updated It will be set as writes Rx224, and be clear as starting address is updated.
3	RO	0	**Video Window 3 Starting Address Update Status** 0: Updated 1: Not yet updated It will be set as writes Rx2A4, and be clear as starting address is updated.
2	RO	0	**IGA2 Vertical Blanking Status**
1	RO	0	**Graphics Starting Address Update Status** 0: Updated 1: Not yet updated It will be set as writes Rx214, and be clear as starting address is updated.
0	RO	0	**Video Window 1 Starting Address Update Status** 0: Updated 1: Not yet updated It will be set as writes Rx254, and be clear as starting address is updated.

Offset Address: 20B-208h

Alpha Window / Hardware Icon (HI) Horizontal and Vertical Location Start **Default Value: 0000 0000h**

Bit	Attribute	Default	Description	
31:27	RO	0	**Reserved**	
26:16	WO	0	**Depend on Hardware Icon Enable (Rx260[0])** 0: Alpha window horizontal (X) starting location 1: Hardware icon horizontal (X) starting location Unit: Pixel	
15:11	RO	0	**Reserved**	
10:0	WO	0	**Write** **Depend on Hardware Icon Enable (Rx260[0])** 0: Alpha window vertical (Y) starting location 1: Hardware icon vertical (Y) starting location Unit: Line	**Read** **Graphic Display Vertical Line Number** Unit: Line

Offset Address: 20F-20Ch

Alpha Window Horizontal and Vertical Location End / Hardware Icon (HI) Center Offset **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RW	0	**Reserved**
26:16	RW	0	**Depend on Hardware Icon Enable (Rx260[0])** 0: [26:16] Alpha window horizontal (X) ending location 1: [22:16] Hardware icon horizontal (X) center offset Unit: Pixel
15:11	RO	0	**Reserved**
10:0	RW	0	**Depend on Hardware Icon Enable (Rx260[0])** 0: [10:0] Alpha window vertical (Y) ending location 1: [6:0] Hardware icon vertical (Y) enter offset Unit: Line

Offset Address: 213-210h

Alpha Window Control **Default Value: 0000 FF00h**

Bit	Attribute	Default	Description
31:21	RO	0	**Reserved**
20:16	RW	0	**Alpha Stream Request Expire Number** Unit: 4 Requests
15:8	RW	FFh	**Constant Alpha Factor Setting For Graphics Blending**
7:2	RO	0	**Reserved**
1:0	RW	00b	**Graphics Blending Alpha Select (Alpha*VID+(1-Alpha)*GRA)** 00: Blending using constant alpha factor [15:8] 01: Alpha is from alpha stream 10: Alpha is from graphics stream 11: Reserved

Offset Address: 217-214h

CRT Starting Address Shadow **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**IGA1 Down Scaling Flip** Address equivalent to 3X5.EC[1]
30	RW	0	**IGA1 Down Scaling Line Flip Enable**
29	RO	0	**Reserved**
28:0	RW	0	**Primary Display Starting Address or** **IGA1 Down Scaling Source Starting Address (Valid when 3X5.EC[0] = 1)** Address equivalent to: 3X5.48 [4:0] 3X5.34 [7:0] 3X5.0C [7:0] 3X5.0D [7:0]

Note: In monochrome mode, the "X" in the above table stands for **"B".** In color mode, the "X" in the above table stands for **"D".**

Offset Address: 21B-218h

The Second Display Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Display Address Selection** 00: S.L Others are not supported
28:3	RW	0	**The Second Display Starting Address or** **IGA2 Down Scaling Source Starting Address (Valid when 3X5.E8[4] = 1)** Unit: 8 bytes
2	RO	0	**Reserved**
1	RW	0	**IGA2 Down Scaling Line Flip Enable**
0	RW	0	**IGA2 Down Scaling Flip** Address equivalent to 3X5.E8[5].

Note: This register should be the same with 3X5.5E, 5D and 5C registers.

Offset Address: 21F-21Ch

Alpha Stream Frame Buffer Stride **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:13	RO	0	**Reserved**
12:4	RW	0	**Alpha Stream Frame Buffer Stride** Unit: 16 bytes
3:0	RO	0	**Reserved**

Offset Address: 223-220h

Primary Display Color Key **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**CRT Color Key** For RGB10 color mode [29].
30	RW	0	**CRT Color Key Enable** 0: Disable 1: Enable
29	RW	0	**CRT Color Key Inverse Control** 0: Display video if color key match 1: Display video if not color key match
28:0	RW	0	**CRT Color Key** Bits [28:0]: For RGB10 color mode [28:0] Bits [23:0]: For 32-bit true color mode Bits [15:0]: For 565 Hi color mode Bits [14:0]: For 555 Hi color mode Bits [7:0]: For 256 color mode

Offset Address: 227-224h

Alpha Window / Hardware Icon Frame Buffer Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30:29	RW	00b	**Target of Frame Buffer Starting Address** 00: S.L 01: S.F 1x: Reserved
28:4	RW	0	**Depend On Hardware Icon Enable (Rx260[0])** 0: Frame buffer starting address for alpha window 1: Frame buffer starting address for hardware icon Unit: 16 bytes
3:0	RO	0	**Reserved**

Offset Address: 22B-228h

Chroma Key Lower Bound **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**Chroma Key Lower for Y/G Bit [1]**
30	RW	0	**Chroma Key Enable** 0: Disable 1: Enable
29	RW	0	**Chroma Key Inverse Control** 0: Display video if chroma key not match 1: Display video if chroma key match
28:0	RW	0	**Chroma Key Lower** {[23:16], [31], [28]}: Y/G {[15:8], [27:26]}: U/R {[7:0], [25:24]}: V/B

Offset Address: 22F-22Ch

Chroma Key Upper Bound **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30	RW	0	**Chroma Key Select** 1 : Select Video 3 0 : Select Video 1
29:0	RO	0	**Chroma Key Upper** {[23:16], [29:28]} Y/G {[15:8], [27:26]}: U/R {[7:0], [25:24]}: V/B

Offset Address: 233-230h

Video Stream 1 Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**V1 Display to CRT or the Second Display** 0: CRT 1: Second Display
30	RW	0	**V1 Window Pre-fetch Enable**
29	RW	0	**V1 Window Gamma Function Enable**
28	RW	0	**V1 Window De-Gamma Function Enable**
27	RW	0	**V1 Window Adder Tile Mode Enable**
26:25	RW	00b	**V1 Flip Control** 00: SW Flip 01: HW Flip and triggle by the HQV engine. 10: HW Flip and triggle by the Capture Port 0. 11: HW Flip and triggle by the Capture Port 1.
24	RW	0	**V1 Frame to Field Enable** 0: Disable 1: Enable If enabled, the stride will 2 times of original values. This bit is valid at 1. Software flip. 2. HQV flip. 3. Capture frame mode flip. 4. MC frame mode flip.
23	RO	0	**Reserved**
22	RW	0	**V1 De-interlace Mode** If enabled hardware will add one line to the top of odd (bottom) field. 0: Disable 1: Enable
21	RW	0	**V1 Line Flip Only in Non Video Active Period Enable** 0: Disable 1: Enable
20:16	RW	0	**V1 Request Expire Number** (Unit: 4 requests)
15:10	RO	0	**Reserved**
9	RW	0	**Divided V1 Flip for HQV Engine VSYNC Number to Half Enable** 0: Disable 1: Enable
8	RW	0	**V1 Color Space Conversion Disable** 0: Enable 1: Disable
7	RW	0	**V1 Color Space Conversion Chroma Sign Bits Conversion**
6:5	RO	0	**Reserved**
4:2	RW	000b	**V1 Stream Data Format** 000: YUV422 001: RGB32 010: RGB15 011: RGB16 100: YUV411 101: RGB10 Other : reserved
1	RO	0	**Reserved**
0	RW	0	**V1 Enable** 0: Disable 1: Enable

Offset Address: 237-234h

Video Window 1 Fetch Count **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:20	RW	0	**V1 Per Line Fetch Count** It is equal, no-sizing line fetch count / minify times (Unit: 16 bytes)
19:0	RO	0	**Reserved**

Offset Address: 23B-238h

Video Window 1 Fetch Buffer Y Starting Address 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:26	RO	0	**Reserved**
30:29	RW	00b	**Target of The Second Frame Buffer Starting Address** 00: S.L. 01: S.F 1x: Reserved
28:3	RW	0	**V1 Packed Mode** The second frame buffer starting address.
2:0	RO	0	**Reserved**

Offset Address: 23F-23Ch

Video Window 1 Frame Buffer Stride **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:14	RO	0	**Reserved**
13:4	RW	0	**V1 Packed Mode** Frame buffer stride (Unit: 16 bytes)
3:0	RO	0	**Reserved**

Offset Address: 243-240h

Video Window 1 Horizontal and Vertical Starting Location **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Video Window 1 Horizontal (X) Starting Location (-1)**. (Unit: pixel)
15:11	RO	0	**Reserved**
10:0	RW	0	**Video Window 1 Vertical (Y) Starting Location (-1)**. (Unit: Line)

Offset Address: 247-244h

Video Window 1 Horizontal and Vertical Ending Location **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Video Window 1 Horizontal (X) Ending Location (-1)**. (Unit: pixel)
15:11	RO	0	**Reserved**
10:0	RW	0	**Video Window 1 Vertical (Y) Ending Location (-1)**. (Unit: Line)

Offset Address: 24B-248h

Video Window 1 Frame Buffer Y Starting Address 2 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30:29	RW	00b	**Target of The Third Frame Buffer Starting Address** 00: S.L. 01: S.F 1x: Reserved
28:3	RW	0	**V1 Packed Mode** The third frame buffer starting address. (Unit: 16 bytes)
2:0	RO	0	**Reserved**

Offset Address: 24F-24Ch

Video Window 1 Display Zoom Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**Video Window 1 Horizontal (X) Zoom Enable** 0: Disable 1: Enable
30:27	RO	0	**Reserved**
26:16	RW	0	**Video Window 1 Horizontal (X) Zoom Factor**
15	RW	0	**Video Window 1 Vertical (Y) Zoom Enable** 0: Disable 1: Enable
14:10	RO	0	**Reserved**
9:0	RW	0	**Video Window 1 Vertical (Y) Zoom Factor**

Offset Address: 253-250h

Video Window 1 Minify & Interpolation Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:24	RW	000b	**V1 Horizontal (X) Minify Control** 000: No minify; 001: Minify by a factor of 2 010: Reserved 011: Minify by a factor of 4 101: Minify by a factor of 8 111: Minify by a factor of 16 Other cases: reserved
23:19	RO	0	**Reserved**
18:16	RW	000b	**V1 Vertical (Y) Minify Control** 000: No minify 001: Minify by a factor of 2 010: Reserved 011: Minify by a factor of 4 101: Minify by a factor of 8 111: Minify by a factor of 16 Other cases: reserved
15:3	RO	0	**Reserved**
2	RW	0	**V1 Luma-only Interpolation When the Vertical Interpolation Is Enabled** 0: Only luma values interpolated. 1: All YUV/YcbCr values interpolated.
1	RW	0	**V1 Horizontal (X) Interpolation Mode Select** 0: Pixel is replicated 1: Enable interpolation
0	RW	0	**V1 Vertical (Y) Interpolation Mode Select** 0: Pixel is replicated 1: Enable interpolation

Offset Address: 257-254h

Video Window 1 Frame Buffer Y Starting Address 0 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**V1 Play Odd (1) / Even (0) Field When SW Playback And Field Base Picture Are Selected** 1: Odd field
30:29	RO	00b	**Target of The First Frame Buffer Starting Address** 00: S.L. 01: S.F 1x: Reserved
28:2	RO	0	**V1 Packed Mode** The first frame buffer starting address. (Unit : 4 bytes)
1:0	RO	0	**Reserved**

Note: In packed mode, we could use Rx254[3:2] to get
1.No minify: 4 bytes alignment. Rx254[3:2] are valid
2. (Minify = 2): 8 bytes alignment. Rx254[3] is valid, and Rx254[2] is omitted.
3. (Minify > 2): 16 bytes alignment. a Rx254[3:2] are omitted.

Offset Address: 25B-258h

Video 1 FIFO Depth and Threshold Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:24	RW	0	**V1 FIFO Pre-threshold** Let V1 to issue request early. Normally, this value is more or equal than V1 FIFO threshold (Rx258[15:8]). (Unit : level)
23:26	RO	0	**Reserved**
15:8	RW	0	**V1 FIFO Threshold** Let V1 request priority from low to high. (Unit: level)
7:0	RW	0	**V1 FIFO Depth (- 1)** (Unit: level)

Note: One level is equal to 16 bytes.

Offset Address: 25F-25Ch

Video Window 1 Horizontal and Vertical Starting Location Offset **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Video Window 1 Horizontal (X) Starting Location Offset** (Unit: 16 bytes)
15:11	RO	0	**Reserved**
10:0	RW	0	**Video Window 1 Vertical (Y) Starting Location Offset** (Unit: Line)

Offset Address: 263-260h

Hardware Icon (HI) Control (Only for Second Display) **Default Value: 000F 00F0h**

Bit	Attribute	Default	Description
31	RW	0	**V4 Displays to CRT (0) or Secondary Display (1)** 0: CRT 1: Secondary Display
30	RW	0	**V4 Window Pre-fetch Enable**
29	RW	0	**HWI + (1-aplha)*Graphics Mode** 0 Disable 1 Enable
28	RW	0	**Alpha Value Come From Where** (Only for the true color Hardware icon) 0: From bit [23:16] 1: From the bit [31:24] of Hardware icon
27:26	RW	00b	**HI Window Size** 00: 32x32 01: 64x64 1x: 128x128 Unit: Pixel X line
25:24	RW	00b	**HI Data Stream Format** 00: RGB555 01: RGB565 10: RGB32 11: RGB10
23:20	RO	0	**Reserved**
19:16	RW	Fh	**HI Constant Alpha [3:0] (HIAPA)**
15:12	RW	0	**Alpha Changed Value (HICV) Per Frame As HI Fan In/Out Turn on Rx260[8]** ALPHAn[7:0] = ALPHAn-1[7:0] +/- HICV[3:0], where just ALPHAn[7:4] is valid alpha value and ALPHA0 = {HIAPA}.
11:10	RW	0	**DMA1 Descriptor Done Interrupt Enable** 0: Disable 1: Enable
9	RW	0	**HI Fan In / Out Selector** 0: Default 1: Fan in (+)
8	RW	0	**HI Fan In / Out Enable** 0: Disable 1: Enable
7:4	RW	Fh	**HI Constant Alpha[7:4] (HIAPA)** The reset bits are put on [19:16].
3	RO	0	**Reserved**
2	RW	0	**HI Blending Enable** 0: Disable 1: Enable
1	RO	0	**Reserved**
0	RW	0	**HI Enable** 0: Disable 1: Enable

Offset Address: 267-264h

The Second Display Color Key **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**The Second Display Color Key Bit [29] For RGB10 True Color Mode** See also bits [28:0] for detail.
30	RW	0	**Second Display Color Key Enable** 0: Disable 1: Enable
29	RW	0	**Second Display Color Key Inverse Control** 0: Display video if color key match 1: Display video if not color key match
28:0	RW	0	**The Second Display Color Key** Bits [31], [28:0]: For RGB10 true color mode Bits [23:0]: For 32-bit true color mode Btis [15:0]: For 565 Hi color mode Bits [14:0]: For 555 Hi color mode Bits [7:0]: For 256 color mode

Offset Address: 26B-268h

V3 and Alpha Window FIFO Pre-Threshold Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:23	RO	0	**Reserved**
22:16	RW	0	**Alpha Window FIFO Pre-threshold** Let alpha engine issues request early. This value is normally more than or equal to the alpha engine FIFO threshold (Rx278[30:24]). Unit : Level
15:8	RO	0	**Reserved**
7:0	RW	0	**V3 FIFO Pre-threshold** Let V3 issues request early. This value is normally more than or equal to V3 FIFO threshold (Rx278[15:8]). Unit: Level

Offset Address: 26F-26Ch

Video Window 1 Display Count On Screen Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:28	RO	0	**Reserved**
27:26	RW	0	**V1 Vertical Line Count That Shows On Screen** (Unit: Line)
25:12	RO	0	**Reserved**
11:0	RW	0	**V1 Horizontal Pixel Count That Shows On Screen (-1)** (unit: pixel)

Offset Address: 273-270h

Hardware Icon (HI) Transparent Color (Only For Second Display) **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:0	RW	0	**HI Transparent Color** Bits [29:0]: For RGB10 Bits [23:0]: For RGB32 Bits [15:0]: For RGB565 Bits [14:0]: For RGB555

Offset Address: 277-274h

Hardware Icon (HI) Inverse Color (Only For Second Display) **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:0	RW	0	**HI Inverse Color** Bits [29:0]: For RGB10 Bits [23:0]: For RGB32 Bits [15:0]: For RGB565 Bits [14:0]: For RGB555

Offset Address: 27B-278h

V3 and Alpha Window FIFO Depth and Threshold Contorl **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30:24	RW	0	**Alpha Window FIFO Threshold** Unit: Level
23	RO	0	**Reserved**
22:16	RW	0	**Alpha Window FIFO Depth (-1)** Unit: Level
15:8	RW	0	**Video Window 3 FIFO Threshold** Unit: Level
7:0	RW	0	**Video Window 3 FIFO Depth (-1)** Unit: Level

Note: One level is equal to 16 bytes.

Offset Address: 27F-27Ch

Video Window 3 Display Count On Screen Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:28	RO	0	**Reserved**
27:26	RW	0	**V3 Vertical Line Count That Shows On Screen** (Unit: Line)
15:12	RO	0	**Reserved**
11:0	RW	0	**V1 Horizontal Pixel Count That Shows On Screen (-1)** (Unit: pixel)

Offset Address: 283-280h

Primary Display Second Color Key **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30	RW	0	**CRT Color Key Enable** 0: Disable 1: Enable
29	RW	0	**CRT Color Key Inverse Control** 0: Display video if color key match 1: Display video if not color key match
28:24	RO	0	**Reserved**
23:0	RW	0	**Primary Display Color Key** Bits [23:0]: For 32-bit true color mode Bits [15:0]: For 565 Hi color mode Bits [14:0]: For 555 Hi color mode Bits [7:0]: For 256 color mode

Offset Address: 287-284h

Video Window 1 Color Space Conversion and Enhancement Control 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**SDTV (BT601) Coefficient Enable** 0: Disable 1: Enable
30	RW	0	**HDTV (BT709) Coefficient Enable** 0: Disable 1: Enable
29	RO	0	**Reserved**
28:24	RW	0	**Coefficient A** X.XXXX From 0 to 1.9375
23:22	RO	0	**Reserved**
21:16	RW	0	**Coefficient B1** SXX.XXX S=1 negative S=0 positive, from –2.125 to 2.125
15:14	RO	0	**Reserved**
13:8	RW	0	**Coefficient C1** SXX.XXX S=1 negative S=0 positive, from –2.125 to 2.125
7:0	RW	0	**Coefficient D** 2's complement integer from -128 to 127

Note: $R = AY+B_1Cb+C_1C_r+D$

Offset Address: 28B-288h

Video Window 1 Color Space Conversion and Enhancement Control 2 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:24	RW	0	**Coefficient B2** SX.XXX S=1 negative S=0 positive, from –1.875 to 1.875
23:21	RO	0	**Reserved**
20:16	RW	0	**Coefficient C2** SX.XXX S=1 negative S=0 positive, from –1.875 to 1.875
15:14	RO	0	**Reserved**
13:8	RW	0	**Coefficient B3** SXX.XXX S=1 negative S=0 positive, from –3.875 to 3.875
7:6	RO	0	**Reserved**
5:0	RW	0	**Coefficient C3** SXX.XXX S=1 negative S=0 positive, from –3.875 to 3.875

Note: $G = AY+B_2Cb+C_2C_r+D$
$B = AY+B_3Cb+C_3C_r+D$

Offset Address: 28F-28Ch

P Logic Adder Result 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:0	RW	0	**P Logic Adder Result 1**

Offset Address: 293-290h

Alpha Window / Color Cursor Ending (Only For Primary Display) **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:28	RO	0	**Reserved**
27:16	RW	0	**Hardware Icon Horizontal (X) Ending Position**
15:11	RO	0	**Reserved**
10:0	RW	0	**Hardware Icon Vertical (Y) Ending Position** Unit: Line

Offset Address: 297-294h

3D AGP Pause Address MMIO Port **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:0	RW	0	**3D AGP Pause Address MMIO Port**

Offset Address: 29B-298h

Compose Output Mode Select **Default Value: 0300 0000h**

Bit	Attribute	Default	Description
31	RW	0	**Video 1 Command End, V1 Load New Register Setting** 1: Fire If registers are updated to engine, this bit will be cleared to 0 and the deafault is set at 0.
30	RW	0	**Video 3 Command End, V3 Load New Register Setting** 1: Fire If registers are updated to engine, this bit will be cleared to 0 and the deafault is set at 0.
29	RW	0	**Video Register Always Loaded** For hardware simulation and the default is set at 0.
28	RW	0	**Video Register Loaded at Vertical Blanking Without Waiting Source Flip** Need to write [31] or [30] and default is set at 0.
27	RW	0	**Video 3 Register Always Loaded** For hardware simulation and the default is set at 0.
26	RW	0	**Video 3 Register Loaded at Vertical Blanking Without Waiting Source Flip** Need to write [31] or [30] and default is set at 0.
25:24	RW	11b	**Interpolation FIFO Clock Select** 00: Not in use 01: V1 HDTV 10: V3 HDTV 11: V1 SDTV and V3 SDTV
23:21	RO	0	**Reserved**
20	RW	0	**Video Output Overlap Control** 0: V1 is on top 1: V3 is on top
19:8	RO	0	**Reserved**
7	RW	0	**MCK Bypass Enable** 0: Disable 1: Enable
6	RO	0	**Reserved**
5	RW	0	**Bypass LCD Horizontal Magnify Function**
4	RW	0	**Bypass LCD Vertical Magnify Function**
3	RW	0	**Video 3 Line Flip Enable** 0: Disable 1: Enable
2	RW	0	**Video 1 Line Flip Enable** 0: Disable 1: Enable
1	RO	0	**Reserved**
0	RW	0	**Video 1 Round Control Enable** 0: Disable 1: Enable

Offset Address: 29F-29Ch

Video Window 3 Frame Buffer Starting Address 2 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30:29	RW	00b	**Target of The Third Frame Buffer Starting Address** 00: S.L. 01: S.F. 1x: Reserved
28:3	RW	0	**The Third Frame Buffer Starting Address of V3** Unit: 16 bytes
2:0	RO	0	**Reserved**

Offset Address: 2A3-2A0h

Video Stream 3 Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30	RW	0	**V3 Window Pre-fetch Enable** 0: Disable 1: Enable
29	RW	0	**V3 Window Gamma Function Enable** 0: Disable 1: Enable
28:27	RO	0	**Reserved**
26:25	RW	00b	**V3 Flip Control** 00: SW flip 01: HW flip and triggle by the HQV engine 10: Reserved 10: HW flip and triggle by the Capture Port 0 11: Reserved 11: HW flip and triggle by the Capture Port 1
24	RW	0	**V3 Frame to Field Enable** 0: Disable 1: Enable If enabled, the stride will be 2 times of original values. This bit is valid at: Software flip, HQV flip.
23	RO	0	**Reserved**
22	RW	0	**V3 De-interlace Mode** 0: Disable 1: Enable If enabled, hardware will add one line to the top of odd (bottom) field
21	RW	0	**V3 Line Flip Only in Non Video Active Period Enable** 0: Disable 1: Enable
20:16	RW	0	**V3 Request Expire Number** Unit: 4 requests
15:10	RO	0	**Reserved**
9	RW	0	**Divided V3 Flip for HQV Engine VSYNC Number to Half Enable** 0: Disable 1: Enable
8	RW	0	**V3 Color Space Conversion Disable** 0: Enable 1: Disable
7	RW	0	**V3 Color Space Conversion Chroma Sign Bits Conversion** 1: Inverse
6:5	RO	0	**Reserved**
4:2	RW	000b	**V3 Stream Data Format** x00: YUV422 001: RGB32 x10: RGB15 011: RGB16
1	RO	0	**Reserved**
0	RW	0	**V3 Enable** 0: Disable 1: Enable

Offset Address: 2A7-2A4h

Video Window 3 Frame Buffer Starting Address 0 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**Play Odd (1) / Even (0) Field When SW Playback and Field Base Picture Are Selected**
30:29	RO	00b	**Target of The First Frame Buffer Starting Address** 00: S.L. 01: S.F. 1x: Reserved
28:2	RW	0	**The First Frame Buffer Starting Address of V3** Unit: 16 bytes
1:0	RO	0	**Reserved**

Offset Address: 2AB-2A8h

Video Window 3 Frame Buffer Starting Address 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30:29	RW	00b	**Taraget of The Second Frame Buffer Starting Address** 00: S.L. 01: S.F. 1x: Reserved
28:3	RW	0	**The Second Frame Buffer Starting Address of V3** Unit: 16 bytes
2:0	RO	0	**Reserved**

Offset Address: 2AF-2ACh

Video Window 3 Frame Buffer Stride **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:14	RO	0	**Reserved**
13:4	RW	0	**V3 Frame Buffer Stride** Unit: 16 bytes
3:0	RO	0	**Reserved**

Offset Address: 2B3-2B0h

Video Window 3 Horizontal and Vertical Start **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**V3 Horizontal (X) Starting Location** Unit: Pixel
15:11	RO	0	**Reserved**
10:0	RW	0	**V3 Vertical (Y) Starting Location** Unit: Line

Offset Address: 2B7-2B4h

Video Window 3 Horizontal and Vertical End **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**V3 Horizontal (X) Ending Location** Unit: Pixel
15:11	RO	0	**Reserved**
10:0	RW	0	**V3 Vertical (Y) Ending Location** Unit: Line

Offset Address: 2BB-2B8h

Video Window 3 and Alpha Window Fetch Count **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:20	RW	0	**V3 Per Line Fetch Count** Unit: Pixel
19:10	RO	0	**Reserved**
9:0	RW	0	**Alpha Window Per Line Fetch Count** Unit: Line

Offset Address: 2BF-2BCh

Video Window 3 Display Zoom Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**V3 Horizontal (X) Zoom Enable** 0: Disable 1: Enable
30:27	RO	0	**Reserved**
26:16	RW	0	**V3 Horizontal (X) Zoom Factor**
15	RW	0	**V3 Vertical (Y) Zoom Enable** 0: Disable 1: Enable
14:10	RO	0	**Reserved**
9:0	RW	0	**V3 Vertical (Y) Zoom Factor**

Offset Address: 2C3-2C0h

Video Window 3 Minify and Interpolation Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RW	0	**Reserved**
26:24	RW	000b	**V3 Horizontal (X) Minify Control** 000: No minify 001: Minify by a factor of 2 010: Reserved 011: Minify by a factor of 4 101: Minify by a factor of 8 111: Minify by a factor of 16 Other cases are reserved.
23:19	RO	0	**Reserved**
18:16	RW	000b	**V3 Vertical (Y) Minify Control** 000: No minify 001: Minify by a factor of 2 010: Reserved 011: Minify by a factor of 4 101: Minify by a factor of 8 111: Minify by a factor of 16 Other cases are reserved.
15:3	RO	0	**Reserved**
2	RW	0	**V3 Luma-only Interpolation When The Vertical Interpolation Is Enabled** 0: Only luma values interpolated 1: All YUV/YcbCr values interpolated
1	RW	0	**V3 Horizontal (X) Interpolation Mode Select** 0: Pixel is replicated 1: Enable interpolation
0	RW	0	**V3 Vertical (Y) Interpolation Mode Select** 0 Pixel is replicated 1 Enable interpolation Note: V1 and V3 can support interpolation simultaneously when both video source resolutions are lower than 800x600. If anyone exceeds the 800x600 resolution, only one of them can support interpolation. The control bit is defined at 0x298[25:24].

Offset Address: 2C7-2C4h

Video Window 3 Color Space Conversion and Enhancement Control 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**SDTV (BT601) Coefficient Enable** 0: Disable 1: Enable
30	RW	0	**HDTV (BT709) Coefficient Enable** 0: Disable 1: Enable
29	RO	0	**Reserved**
28:24	RW	0	**Coefficient A** X.XXXX from 0 to 1.9375
23:22	RO	0	**Reserved**
21:16	RW	0	**Coefficient B1** SXX.XXX S=1 negative S=0 positive From – 2.125 to 2.125
15:14	RO	0	**Reserved**
13:8	RW	0	**Coefficient C1** SXX.XXX S=1 negative S=0 positive From – 2.125 to 2.125
7:0	RW	0	**Coefficient D[10:3]** 2's complement integer from -128 to 127

Note: R=AY+B1Cb+C1Cr+D

Offset Address: 2CB-2C8h

Video Window 3 Color Space Conversion and Enhancement Control 2 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RW	0	**Coefficient D[2:0]** 2's complement integer from -128 to 127
28:24	RW	0	**Coefficient B2** SX.XXX S=1 negative S=0 positive, from –1.875 to 1.875
23:21	RO	0	**Reserved**
20:16	RW	0	**Coefficient C2** SX.XXX S=1 negative S=0 positive, from –1.875 to 1.875
15:14	RO	0	**Reserved**
13:8	RW	0	**Coefficient B3** SXX.XX S=1 negative S=0 positive, from 0 to 3.75
7:6	RO	0	**Reserved**
5:0	RW	0	**Coefficient C3** SX.XX S=1 negative S=0 positive, from –3.875 to 3.875

Note: G = AY+B2Cb+C2Cr+D, B = AY+B3Cb+C3Cr+D

Offset Address: 2CF-2CCh

T Logic Adder Result 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:0	RW	0	**T Logic Adder Result 1**

Offset Address: 2D3-2D0h

Graphic Hardware Cursor Mode Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**Mono Cursor Display Path** 0: Primary 1: Secondary
30:29	RW	00b	**Target of The Hardware Cursor Buffer Starting Address** 00: S.L. 01: S.F. 1x: Reserved
28:26	RO	0	**Reserved**
25:8	RW	0	**Hardware Cursor Base Address** Up to 64M bytes For 32x32x2 pattern: Bits [25:8] define the base address For 64x64x2 pattern: Bits [25:10] define the base address
7:2	RO	0	**Reserved**
1	RW	0	**Hardware Cursor Size** 0: 64x64x2 1: 32x32x2
0	RW	0	**Hardware Cursor Enable** 0: Disable 1: Enable

Offset Address: 2D7-2D4h

Graphic Hardware Cursor Position **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Hardware Cursor Position in the X-coordinate**
15:11	RO	0	**Reserved**
10:0	RW	0	**Hardware Cursor Position in the Y-coordinate**

Offset Address: 2DB-2D8h

Graphic Hardware Cursor Origin **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Hardware Cursor Origin in the X-coordinate**
15:11	RO	0	**Reserved**
10:0	RW	0	**Hardware Cursor Origin in the Y-coordinate**

Offset Address: 2DF-2DCh

Graphic Hardware Cursor Background **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:24	RO	0	**Reserved**
23:0	RW	0	**For 256 Color Mode** Bits [7:0] specify hardware cursor background color **For 555 Hi Color Mode** Bits [14:0] specify hardware cursor background color **For 565 Hi Color Mode** Bits [15:0] specify hardware cursor background color **For 32-bits True Color Mode** Bits [23:0] specify hardware cursor background color

Offset Address: 2E3-2E0h

Graphic Hardware Cursor Foreground **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:24	RO	0	**Reserved**
23:0	RW	0	**For 256 Color Mode** Bits [7:0] specify hardware cursor foreground color. **For 555 Hi Color Mode** Bits [14:0] specify hardware cursor foreground color. **For 565 Hi Color Mode** Bits [15:0] specify hardware cursor foreground color. **For 32-bits True Color Mode** Bits [23:0] specify hardware cursor foreground color.

Table 3 below shows the hardware color cursor operation. Please be noted this table is only applicable to the graphics modes. For the text modes, the VGA registers control the hardware color cursor.

Table 3. Graphics Hardware Color Cursor Operation

Pixel Operation	AND Plane	XOR Plane
Choose graphics hardware color cursor background color	0	0
Choose graphics hardware color cursor foreground color	0	1
Transparent	1	0
VGA data is inverted	1	1

Offset Address: 2E7-2E4h

T Logic Data Result 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:0	RW	0	**T Logic Data Result 1**

Offset Address: 2EB-2E8h

HI FIFO Depth and Threshold Control (Only For Primary Display) **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:24	RW	0	**HI FIFO Pre-threshold** Let HI to issue request early. Normally, this value is more than or equal to HI FIFO threshold (Rx2E8[14:8]). (Unit: level)
23:16	RO	0	**Reserved**
15:8	RW	0	**HI FIFO Threshold** Let HI request priority from low to high. (Unit: level)
7:0	RO	0	**HI FIFO Depth (- 1)** Unit: level

Offset Address: 2EF-2ECh

Hardware Icon (HI) Transparent Color (Only For Primary Display) **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:0	RW	0	**HI Transparent Color** Bits [29:0]: For RGB10 Bits [23:0]: For RGB32 Bits [15:0]: For RGB565 Bits [14:0]: For RGB555

Offset Address: 2F3-2F0h

Hardware Icon (HI) Control (Only for Primary Display) **Default Value: 000F 00F0h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30	RW	0	**HI Window Pre-fetch Enable**
29	RW	0	**HWI + (1-aplha)*Graphics Mode** 0 Disable 1 Enable
28	RW	0	**Alpha Value Come From Where** Only for the true color hardware icon. 0: From bit [23:16] 1: From the bit [31:24] of hardware icon
27:26	RW	00b	**HI Window Size** 00: 32 x 32 01: 64 x 64 1x: 128 x 128 Unit: Pixel x line
25:24	RW	00b	**HI Data Stream Format** 00: RGB555 01: RGB565 10: RGB32 11: RGB10
23:20	RO	0	**Reserved**
19:16	RW	Fh	**HI Constant Alpha [3:0] (HIAPA)**
15:12	RW	0	**Alpha Changed Value (HICV) Per Frame as HI Fan In / Out Turn On Rx260[8]** ALPHAn[7:0] = ALPHAn-1[7:0] +/- HICV[3:0], where just ALPHAn[7:4] is valid alpha value and ALPHA0 = {HIAPA}.
11:10	RO	0	**Reserved**
9	RW	0	**HI Fan In / Out Selector** 0: Default 1: Fan in (+)
8	RW	0	**HI Fan In / Out Enable** 0: Disable 1: Enable
7:4	RW	Fh	**HI Constant Alpha[7:4] (HIAPA)** The Rest Bits are put on bits [19:16]
3	RO	0	**Reserved**
2	RW	0	**HI Blending Enable** 0: Default 1: Enable
1	RO	0	**Reserved**
0	RW	0	**HI Enable** 0: Default 1: Enable

Offset Address: 2F7-2F4h

Hardware Icon Frame Buffer Starting Address (Only For Primary Display) **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30:29	RW	00b	**Target of The Frame Buffer Starting Address** 00: S.L. 01: S.F. 1x: Reserved
28:4	RW	0	**Frame Buffer Starting Address for Hardware Icon** Unit: 16 bytes
3:0	RO	0	**Reserved**

Offset Address: 2FB-2F8h

Hardware Icon (HI) Horizontal and Vertical Location Start (Only For Primary Display) **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Hardware Icon Horizontal (X) Starting Location** Unit: Pixel
15:11	RO	0	**Reserved**
10:0	RW	0	**Hardware Icon Vertical (Y) Starting Location** Unit: Line

Offset Address: 2FF-2FCh

Hardware Icon (HI) Center Offset (Only For Primary Display) **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:23	RO	0	**Reserved**
22:16	RW	0	**Hardware Icon Horizontal (X) Center Offset** Unit: Pixel
15:7	RO	0	**Reserved**
6:0	RW	0	**Hardware Icon Horizontal (Y) Center Offset** Unit: Line

Offset Address: 1203-1200h

Video Gamma Color R Register for Video 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:26	RO	0	**Reserved**
25:16	RW	0	**Color Value After Gamma Function**
15:5	RO	0	**Reserved**
4:0	RW	0	**Color Index Number for Gamma Function Division**

Offset Address: 1207-1204h

Video Gamma Color G Register for Video 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:26	RO	0	**Reserved**
25:16	RW	0	**Color Value After Gamma Function**
15:5	RO	0	**Reserved**
4:0	RW	0	**Color Index Number for Gamma Function Division**

Offset Address: 120B-1208h

Video Gamma Color B Register for Video 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:26	RO	0	**Reserved**
25:16	RW	0	**Color Value After Gamma Function**
15:5	RO	0	**Reserved**
4:0	RW	0	**Color Index Number for Gamma Function Division**

Offset Address: 120F-120Ch

Hardware Icon (HI) Inverse Color (Only For Primary Display) **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:0	RW	0	**HI Inverse Color** Bits [29:0]: For RGB10 Bits [23:0]: For RGB32 Btis [15:0]: For RGB565 Bits [14:0]: For RGB555

Offset Address: 1223-1220h

Video Gamma Color R Register for Video 3 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:26	RO	0	**Reserved**
25:16	RW	0	**Color Value After Gamma Function**
15:5	RO	0	**Reserved**
4:0	RW	0	**Color Index Number for Gamma Function Division**

Offset Address: 1227-1224h

Video Gamma Color G Register for Video 3 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:26	RO	0	**Reserved**
25:16	RW	0	**Color Value After Gamma Function**
15:5	RO	0	**Reserved**
4:0	RW	0	**Color Index Number for Gamma Function Division**

Offset Address: 122B-1228h

Video Gamma Color B Register for Video 3 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:26	RO	0	**Reserved**
25:16	RW	0	**Color Value After Gamma Function**
15:5	RO	0	**Reserved**
4:0	RW	0	**Color Index Number for Gamma Function Division**

Offset Address: 122F-122Ch

Video Window 3 Horizontal and Vertical Starting Location Offset **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Video Window 3 Horizontal (X) Starting Location Offset** (Unit: 16 bytes)
15:11	RO	0	**Reserved**
10:0	RW	0	**Video Window 3 Vertical (Y) Starting Location Offset** (Unit: Line)

Offset Address: 1283-1280h

Interrupt Flags and Masks Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW1C	0	**MSI Pending Interrupt Re-trigger Bit** When SW wants to exit interrupt service, please clear the bit. HW may send out interrupt again if pending interrupt exists. The function is enabled when MSI Enable = 1'b1.
30:20	RO	0	**Reserved**
19	RW	0	**DMA3 Transfer Done Interrupt Enable** 0: Disable 1: Enable
18	RW	0	**DMA3 Descriptor Done Interrupt Enable** 0: Disable 1: Enable
17	RW	0	**DMA2 Transfer Done Interrupt Enable** 0: Disable 1: Enable
16	RW	0	**DMA2 Descriptor Done Interrupt Enable** 0: Disable 1: Enable
15:5	RO	0	**Reserved**
4	RO	0	**CRT Sense Interrupt Status**
3	RO	0	**DMA3 Transfer Done Interrupt Status**
2	RO	0	**DMA3 Descriptor Done Interrupt Status**
1	RO	0	**DMA2 Transfer Done Interrupt Status**
0	RO	0	**DMA2 Descriptor Done Interrupt Status**

Note: Write 1 to bit [4:0] to clear bits

Offset Address: 1287-1284h

Logic Signature Setting **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:8	RO	0	**Reserved**
7	RW	0	**T Signature RD Disable 1**
6	RW	0	**P & T Select Signal 1**
5	RW	0	**T Signature Enable 1**
4	RW	0	**P Signature Enbale 1**
3	RW	0	**T Signature RD Disable 0**
2	RW	0	**P & T Select Signal 0**
1	RW	0	**T Signature Enable 0**
0	RW	0	**P Signature Enbale 0**

Offset Address: 128B-1288h

P Logic Adder Result 0 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:0	RW	0	**P Logic Adder Result 0**

Offset Address: 128F-128Ch

T Logic Adder Result 0 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:0	RW	0	**T Logic Adder Result 0**

Offset Address: 1293-1290h

IGA1 Display Position Counter 0 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:16	RO	0	**IGA1 Display Line Counter**
15:0	RO	0	**IGA1 Display Frame Counter**

Offset Address: 1297-1294h

IGA1 Display Position Counter 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:16	RO	0	**Reserved**
15:0	RO	0	**IGA1 Display Frame Counter**

Offset Address: 129B-1298h

IGA1 Display Position Counter 2 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:16	RO	0	**Reserved**
15:0	RW	0	**IGA1 Display Frame Counter Enable** 0: Disable 1: Enable When this bit is disabled, frame counter always gets 16'h0.

Offset Address: 129F-129Ch

T Logic Data Result 0 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:0	RW	0	**T Logic Data Result 0**

Offset Address: 12A3-12A0h

IGA2 Display Position Counter 0 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:16	RO	0	**IGA2 Display Line Counter**
15:0	RO	0	**IGA2 Display Frame Counter**

Offset Address: 12A7-12A4h

IGA2 Display Position Counter 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:16	RO	0	**Reserved**
15:0	RO	0	**IGA2 Display Frame Counter**

Offset Address: 12AB-12A8h

IGA2 Display Position Counter 2 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:16	RO	0	**Reserved**
15:0	RW	0	**IGA2 Display Frame Counter Enable** 0: Disable 1: Enable When this bit is disabled, frame counter always gets 16'h0.

Offset Address: 12B3-12B0h

Primary Display Data Color Space Conversion and Enhancement Control 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:20	RW	0	**A1 <= XXXXXXXXXX**
19:10	RW	0	**B1 <= XXXXXXXXXX**
9:0	RW	0	**C1 <= XXXXXXXXXX**

Note: Y = A1R+B1G+C1B+D

Coefficient A1, B1, C1: 10 bits, 0.XXXXXXXX from 0 to 0.99903 Coefficient D: 8 bit positive integer from 16 to 255

Offset Address: 12B7-12B4h

Primary Display Data Color Space Conversion and Enhancement Control 2 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:20	RW	0	**A2[9:0] <= XXXXXXXXXX**
19:10	RW	0	**B2[9:0] <= XXXXXXXXXX**
9:0	RW	0	**C2[9:0] <= XXXXXXXXXX**

Note: Cr = A2R+B2G+C2B +128

Coefficient A2, B2, C2: 11 bits S.XXXXXXXX 2's complement from -0.99903 to 0.99903

Offset Address: 12BB-12B8h

Primary Display Data Color Space Conversion and Enhancement Control 3 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:20	RW	0	**A3[9:0] <= XXXXXXXXXX**
19:10	RW	0	**B3[9:0] <= XXXXXXXXXX**
9:0	RW	0	**C3[9:0] <= XXXXXXXXXX**

Note: Cr = A3R+B3G+C3B +128

Coefficient A3, B3, C3: 11 bits. S.XXXXXXXX 2's complement from -0.99903 to 0.99903

Offset Address: 12BF-12BCh

Primary Display Data Color Space Conversion and Enhancement Control 4 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:14	RO	0	**Reserved**
13	RW	0	**A2[10] <= S**
12	RW	0	**B2[10] <= S**
11	RW	0	**C2[10] <= S**
10	RW	0	**A3[10] <= S**
9	RW	0	**B3[10] <= S**
8	RW	0	**C3[10] <= S**
7:0	RW	0	**D**

Video Capture Engine Register Descriptions (300-37Ch)

Offset Address: 303-300h

Capture Interrupt Control and Flags **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:11	RO	0	**Reserved**
10	RW	0	**Current Writing VBI Buffer ID**
9	RW	0	**End of VBI Interrupt Enable**
8	RW	0	**End of Active Video Interrupt Enable** If TS (Transport Stream) is enabled, it defines as TS data over a buffer interrupt enable.
7	RO	0	**Video Capture Port Internal FIFO Full Status**
6	RO	0	**Current Active Video Input Field Status** 0: Top field 1: Bottom field
5	RO	0	**Current Input Vsync Status** 0: Vertical blanking 1: Active video
4:3	RO	0	**Current Writing Active Video (or TS data) Frame Buffer ID**
2	RO	0	**Flipping Active Video Field Status** 0: Top field 1: Bottom field
1	RO	0	**Video Capture End-of-VBI Status**
0	RO	0	**Video Capture End-of-Active Video Status** If TS is enabled, it defines as TS data over a buffer status

Note: Write 1 to clear bits [1:0] and [7]

Offset Address: 30B-308h

Transport Stream Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:23	RO	0	**Reserved**
22	RW	0	**Serial Input Enable** 0: Enable parallel TS input 1: Enable serial TS input
21	RW	0	**Bit Alignment** Serial input mode only 0: LSB first 1: MSB first
20	RW	0	**Packet Starting Signal Disable** 0:Enable 1:Disable
19	RW	0	**Change Buffer Mode** 0: According to count packet number 1: According to byte count
18:4	RW	0	**When bit 19 = 0** **Packet_Number_Minus_One** There are (Packet_Number_Minus_One + 1) packets per buffer. **When bit 19 = 1** **KBytes_Count** There are KBytes_Count Kbytes per buffer.
3:2	RW	00b	**Method to Move Received TS Data** 0x: Capture engine write data to FB. After fill a buffer, trigger an interrupt to driver. 10: Capture engine write data to FB. After fill a buffer, trigger an interrupt to DMA. 11: Capture engine control DMA to move data. (not via frame buffer) (In mode = 2'b11, set FIFO Threshold Rx310[27:24] to 1)
1	RW	0	**Drop Error Packet** This bit is only valid when TS_DERR pin is available 0: Write all received data out. 1: Drop the data of error packet
0	RW	0	**Transport Stream Input Enable** 0: Disable 1: Enable Turn on this bit before enabling capture engine Rx310[0].

Offset Address: 313-310h

Capture Interface Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**Capture CLK Enable** 0: Disable 1: Enable This bit should be turned on before Rx310[0].
30	RW	0	**Capture FIELD Signal Output Inverted Select** 1: Inverted
29	RW	0	**Vertical Count Starting Reference** 0: Negative edge of VREF 1: Positive edge of VREF
28	RW	0	**Horizontal Count Starting Reference** 0: Negative edge of HREF 1: Positive edge of HREF
27:24	RW	0	**Capture FIFO Threshold** (Unit: level) Once the queuing captured data is more than the threshold, it starts to write data out. Unit of the 1st capture engine is 4-levels, the FIFO size is 64 level x 64 bit. Unit of the 2nd capture engine is 2-levels, the FIFO size is 32 level x 64 bit.
23	RW	0	**Switch Capture Clock Source** Since there are two capture clock sources, use this bit to switch it. In VIP2.0 while using task bit to differentiate video stream, it needs to switch the clock source as the same one. For 1st Capture Engine: 0: Clock from 1st capture CLK pin 1: Clock from 2nd capture CLK pin For 2nd Capture Engine: 0: Clock from 2nd capture CLK pin 1: Clock from 1st capture CLK pin
22	RW	0	**Capture FIELD Input Inverted Select** 1: Inverted
21	RW	0	**Capture HREF Input Inverted Select** 1: Inverted
20	RW	0	**Capture VREF Input Inverted Select** 1: Inverted
19	RW	0	**Capture CLK Input Inverted Select** 1: Inverted
18:16	RW	000b	**Capture Horizontal Filter Mode Select (2P)** 000: No filtering 001: 2 tap (1,1)/2 010: 3 tap (1,2,1)/4 011: 4 tap (1,3,3,1)/8 100: 5 tap (1,2,2,2,1)/8 101~111:Reserved
15	RW	0	**Capture Flipping Control When Rx310[13:12] Is Set to 2'b11** 0: Capture engine flips to HQV or video engine after captured a frame. 1: Capture engine flips to HQV or video engine after captured a field (HQV or Video should set to frame_to_field).
14	RW	0	**4:2:2 to 4:4:4 Cb, Cr Type Select** 0: Duplication 1: Interpolation
13:12	RW	00b	**Capture De-interlace Mode Select** 00: Capture odd field only, 30fps 01: Capture even field only, 30fps 10: Capture odd / even field, 60fps; place on the same location 11: Capture odd / even field, 30fps; place in interlace fashion doubling the storage space
11	RW	0	**Input FIELD Signal Enable** If TS is enabled, it defines as TS_DERR signal enable. 0: Disable 1: Enable
10	RW	0	**VIP Type** 0: VIP1.1, VBI data region specify by task bit. 1: VIP2, VBI data region specify by SAV / EAV during vertical blanking period.

(Continued for Rx310h)

Bit	Attribute	Default	Description
9:8	RW	00b	**Byte Swapping Control** 00: 0123 (no swap: YUYV) 01: 1032 (C, Y swap: UYVY) 10: 0321 (Cr, Cb swap: YVYU) 11: 3012 (Cr, Cb swap and Y swap: VYUY)
7	RW	0	**16 Bit Input Low/High Swap** 1: Low/high byte inverted
6	RW	0	**CCIR656-16 Bit Header Decode Mode** 0: Low 8bit 1: 16bit all
5:4	RW	00b	**Input Stream Type** 00: CCIR601-8bit 01: CCIR656-8bit 10: CCIR601-16bit 11: CCIR656-16bit
3	RW	0	**VIP Enable** 0: Disable 1: Enable
2	RW	0	**Buffer Mode** 0: Double buffers, use starting address 1 and 2 1: Triple buffers, use starting address 1, 2 and 3
1	RW	0	**Bit Stream Selection of VIP2.0** In VIP2.0, task bit to differentiate video stream. For the 1st capture engine: 0: Capture the data of task bit is 0 1: Capture the data of task bit is 1 For the 2nd capture engine: 0: Capture the data of task bit is 1 1: Capture the data of task bit is 0
0	RW	0	**Capture Enable** 0: Disable 1: Enable

Offset Address: 317-314h

Active Video Horizontal Range **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
27:16	RW	0	**Horizontal Ending Line** (CCIR601 only). (Unit: Line)
15:12	RO	0	**Reserved**
11:0	RW	0	**Horizontal Starting Line** (CCIR601 only). (Unit: Line)

Offset Address: 31B-318h

Active Video Vertical Range **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
27:16	RW	0	**Vertical Ending Line (CCIR601 only)**. (Unit: Line)
15:11	RO	0	**Reserved**
10:0	RW	0	**Vertical Starting Line (CCIR601 only)**. (Unit: Line)

Offset Address: 31F-31Ch

Active Video Scaling Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:28	RO	0	**Reserved**
26	RW	0	**Vertical Minify Enable** 0: Disable 1: Enable
25:16	RW	0	**Vertical Minify Factor**
15:12	RO	0	**Reserved**
11	RW	0	**Horizontal Minify Enable** 0: Disable 1: Enable
10:0	RW	0	**Horizontal Minify Factor**

Offset Address: 323-320h

VBI Data Horizontal Range **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:28	RO	0	**Reserved**
27:16	RW	0	**Horizontal Ending Line** (Unit: Line)
15:12	RO	0	**Reserved**
11:0	RW	0	**Horizontal Starting Line** (Unit: Line)

Offset Address: 327-324h

VBI Data Vertical Range **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Vertical Ending Line** (Unit: Line)
15:11	RO	0	**Reserved**
10:0	RW	0	**Vertical Starting Line** (Unit: Line)

Offset Address: 32B-328h

First VBI Buffer Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**VBI Data Enable** 0: Disable 1: Enable
30	RW	0	**VBI Mode Select** 0: Range depend on SAV/EAV 1: Capture by specify range (define by register 320h, 324h) If VIP enable (Rx310[3]=1), this bit setting would be ignored, capture VBI data defined as VIP spec.
29	RO	0	**Reserved**
28:4	RW	0	**VBI or ANC Buffer 0 Starting Address** (Unit: 16 bytes)
3:2	RO	0	**Reserved**
1:0	RW	00b	**Buffer Selection** 00: S.L 01: S.F 10: S.M 11: Reserved

Offset Address: 32F-32Ch
VBI Buffer Stride **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:14	RO	0	**Reserved**
13	RW	0	**VBI Data Placement Method** 0: Linear, no stride needed 1: With stride
12:4	RW	0	**VBI Buffer Stride** (Unit: 16 bytes)
3:0	RO	0	**Reserved**

Offset Address: 333-330h
Ancillary Data Count Setting **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:15	RO	0	**Reserved**
14	RW	0	**Ancillary Data Type** 0: Type2 – 00-FF-FF-DID-SDID-NN-….-….-CheckSumByte-FillBytes. Total captured data are (8Bytes + NN*4Bytes). 1: Type1 – 00-FF-FF-DID-DBN-NN-….-….-CheckSumByte-FillBytes. Total captured data are (8Bytes + (DBN+NN)*4Bytes).
13	RW	0	**Ancillary Data Enable** 0: Disable 1: Enable
12	RW	0	**Ancillary Data Count Reference Select** 0: By header decoder 1: By register (define by Rx330[11:0]).
11:0	RW	0	**Ancillary Data Should Be Capture Length** (Unit: Double-word)

Offset Address: 337-334h
Maximum Data Count of Active Video **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:26	RO	0	**Reserved**
26:16	RW	0	**Maximum Active Video Line Count In A Field** (Unit: Line) If TS enable, it defines as the maximum TS data count of a packet (Unit: Byte)
15:9	RO	0	**Reserved**
8:0	RW	0	**Maximum Active Video QW Count In A Line** (Unit: 8 bytes)

Offset Address: 33B-338h
Maximum Data Count of VBI or ANC **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:26	RO	0	**Reserved**
26:16	RW	0	**Maximum VBI or ANC Line Count In A Field** (Unit: Line)
15:9	RO	0	**Reserved**
8:0	RW	0	**Maximum VBI or ANC QW Count In A Line** (Unit: 8 bytes)

Offset Address: 33F-33Ch
Capture Data Count **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Current Active Video Line Counter** (Unit: Line)
15:13	RO	0	**Reserved**
12:0	RW	0	**VBI or ANC Data Length That Has Been Captured** (Unit: 8 bytes)

Offset Address: 343-340h

First Active Video Frame Buffer Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:4	RW	0	**Active Video Frame Buffer 0 Starting Address** (Unit: 16 bytes) If TS is enabled, it defines as frame buffer 0 starting address for TS data.
3:2	RO	0	**Reserved**
1:0	RW	00b	**Buffer Selection** 00: S.L 01: S.F 10: S.M 11: Reserved

Offset Address: 347-344h

Second Active Video Frame Buffer Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:4	RW	0	**Active Video Frame Buffer 1 Starting Address** (Unit: 16 bytes) If TS is enabled, it defines as frame buffer 1 starting address for TS data.
3:2	RO	0	**Reserved**
1:0	RW	00b	**Buffer Selection** 00: S.L 01: S.F 10: S.M 11: Reserved

Offset Address: 34B-348h

Third Active Video Frame Buffer Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:4	RW	0	**Active Video Frame Buffer 2 Starting Address** (Unit: 16 bytes) If TS is enabled, it defines as frame buffer 2 starting address for TS data.
3:2	RO	0	**Reserved**
1:0	RW	00b	**Buffer Selection** 00: S.L 01: S.F 10: S.M 11: Reserved

Offset Address: 34F-34Ch

Second VBI Buffer Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:4	RW	0	**VBI or ANC Buffer 1 Starting Address** (Unit: 16 bytes)
3:2	RO	0	**Reserved**
1:0	RW	00b	**Buffer Selection** 00: S.L 01: S.F 10: S.M 11: Reserved

Offset Address: 353-350h

Stride of Active Video Buffer & Coring Function Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:24	RO	0	**Reserved**
23	RW	0	**Coring Function Enable**
22:16	RW	0	**Coring Function Compare Data (CCD)** If coring function enable (Rx350[23]) and (-(CCD+1) <= U, V <= CCD), then all of these U and V will be truncated to zero)
15:13	RO	0	**Reserved**
12:4	RW	0	**Stride of Active Video Buffer** (Unit: 8 bytes)
3:0	RO	0	**Reserved**

Offset Address: 357-354h

TS Buffer 0 Error Packet Indicator **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Last Error Packet Indicator** 0: Less than one error packet in this buffer, defined at bits [15:0] 1: More than two error packets, the last error packet ID defined at bits [30:16]
30:16	RO	0	**Last Error Packet ID**
15	RO	0	**First Error Packet Indicator** 0: No error packet in this buffer 1: More than one error packet, the first error packet ID defined at bits [14:0]
14:0	RO	0	**First Error Packet ID**

Offset Address: 35B-358h

TS Buffer 1 Error Packet Indicator **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Last Error Packet Indicator** 0: Less than one error packet in this buffer, defined at bits [15:0] 1: More than two error packets, the last error packet ID defined at bits [30:16]
30:16	RO	0	**Last Error Packet ID**
15	RO	0	**First Error Packet Indicator** 0: No error packet in this buffer 1: More than one error packet, the first error packet ID defined at bits [14:0]
14:0	RO	0	**First Error Packet ID**

Offset Address: 35F-35Ch

TS Buffer 2 Error Packet Indicator **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RO	0	**Last Error Packet Indicator** 0: Less than one error packet in this buffer, defined at bits [15:0] 1: More than two error packets, the last error packet ID defined at bits [30:16]
30:16	RO	0	**Last Error Packet ID**
15	RO	0	**First Error Packet Indicator** 0: No error packet in this buffer 1: More than one error packet, the first error packet ID defined at bits [14:0]
14:0	RO	0	**First Error Packet ID**

Note: Capture supports 2 input interface; therefore, an additional register space is provided to match the above registers definition.
Writing a register to this space, it will write to the second Capture Engine.
The relationship between the additional register space and original register space is
(The additional register address) = (The original register address) + 16'h1000.

HQV Engine Register Descriptions (380-3FFh)

Offset Address: 383-380h

HQV Source Data Offset Control 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Horizontal Offset of Start Point for Video Location In Destination Picture** Unit: pixel (2P) Either video or sub-picture destination data horizontal offset of start point should be set to zero.
15:11	RO	0	**Reserved**
10:0	RW	0	**Vertical Offset of Start Point for Video Location In Destination Picture** Unit: line (2P) Either video or sub-picture destination data vertical offset of start point should be set to zero.

Offset Address: 387-384h

HQV Source Data Offset Control 2 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Horizontal Offset of Start Point for Sub-picture Location In Destination Picture** Unit: pixel (2P) Either video or sub-picture destination data horizontal offset of start point should be set to zero.
15:11	RO	0	**Reserved**
10:0	RW	0	**Vertical Offset of Start Point for Sub-picture Location In Destination Picture** Unit: line (2P) Either video or sub-picture destination data vertical offset of start point should be set to zero.

Offset Address: 38B-388h

HQV Source Data Offset Control 3 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Horizontal Offset of End Point for Video Location In Destination Picture** Unit: pixel (2P)
15:11	RO	0	**Reserved**
10:0	RW	0	**Vertical Offset of End Point for Video Location In Destination Picture** Unit: line (2P)

Offset Address: 38F-38Ch

HQV Source Data Offset Control 4 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:27	RO	0	**Reserved**
26:16	RW	0	**Horizontal Offset of End Point for Sub-picture Location In Destination Picture** Unit: byte (2P)
15:11	RO	0	**Reserved**
10:0	RW	0	**Vertical Offset of End Point for Sub-picture Location In Destination Picture** Unit: line (2P)

Offset Address: 393-390h

HQV Parameters of Hardware Tuning Performance / Quality **Default Value: 4664 8688h**

Bit	Attribute	Default	Description
31	RO	0	**Reserved**
30:28	RW	100b	**Threshold of Inter-Field Complexity for Pull Down Detection (x4)** Calculate the difference between current & previous field.
27	RO	0	**Reserved**
26:25	RW	11b	**Factor of Calculating Threshold of Intra-Field Complexity** 00: Threshold = 390[30:28]*4 + 390[30:28]*1 01: Threshold = 390[30:28]*4 + 390[30:28]*2 10: Threshold = 390[30:28]*4 + 390[30:28]*3 11: Threshold = 390[30:28]*4 + 390[30:28]*4
24:23	RO	0	**Reserved**
22:21	RW	11b	**Static Judgment Number (SJN)** As static record number is equal to SJN, then static flag is asserted
20	RW	0	**The Field Number for The Increment of Static Record** 0: 1 field / (static record) 1: 2 fields / (static record)
19:18	RW	01b	**Pull-down Factor 2** Apply this factor while pull-down detected 00: 3/8 01: 4/8 10: 5/8 11: 6/8
17:16	RW	00b	**Pull-down Factor 1** Apply this factor while pull-down not yet detected. 00: 2/8 01: 3/8 10: 4/8 11: 5/8
15:14	RW	10b	**Threshold for Pull Down Detection** (Rx3DC.[10:0]<<Rx390.[15:14]) as the minimum threshold for valid pull down detection
13:12	RO	0	**Reserved**
11:8	RW	6h	**Threshold for Motion Detection (x2)**
7:4	RW	8h	**Edge Detection Threshold**: **for Degree 90 (x2)**
3:0	RW	8h	**Maximum Difference between Block Boundary (x4)** Apply de-blocking with sin(x) function while the difference between block boundary is greater than this setting.

Offset Address: 397-394h
HQV Extended Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:8	RO	0	**Reserved**
7	RW	0	**Color Adjustment Enable** 0: Disable 1: Enable
6	RW	0	**Bob de-interlacing method** 0: Line average 1: Line duplication (for TV output)
5	RW	0	**YUV Output Format Control** (2p) 0: YUV422 out 1: YUV444 out This bit is effective when (H or V scaling size) < (1/2 H or V original) size. For the other cases, it just uses YUV422 out
4	RW	0	**Color Space Conversion Method** 0: BT601 1: BT709
3	RW	0	**Color Format Convert Method** (from YUV420 ➔ YUV422) 0: 4-tap interpolation 1: Method 1 (-1 9 9 –1).
2:1	RW	00b	**Color Format Convert Method** (from YUV422 ➔ YUV444) 00: Method 1 (WMV9 and H.264) 01: Reserved 1x: Method 3 (-1 9 9 –1).
0	RW	0	**Color Format Convert Method** (from YUV444 ➔ YUV422) 0: Method 1 (1 1) 1: Method 2 (Drop) This bit is effective as (no scaling) or (scaling size) > (1/2 original size). For the other cases, it just uses method 2.

Offset Address: 39B-398h
HQV Static Record Frame Buffer Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:4	RW	0	**Static Record Frame Buffer Starting Address** Unit: 16 bytes
3:2	RO	0	**Reserved**
1:0	RW	0	**Memory Location**

Offset Address: 39F-39Ch
HQV Static Record Frame Buffer Stride **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:14	RO	0	**Reserved**
9:4	RW	0	**Static Record Frame Buffer Stride** Unit: 16 bytes
3:0	RO	0	**Reserved**

Offset Address: 3A3-3A0h
HQV Color Adjustment Control 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:20	RW	0	**Coefficient C1(s[28:27].[26:20])** (2p)
19:10	RW	0	**Coefficient B1(s[18:17].[16:10])** (2p)
9:0	RW	0	**Coefficient A1(s[8:7].[6:0])** (2p)

Note: Y'(R') = A1*Y(R) + B1*Cb(G) + C1*Cr(B) + D1

Offset Address: 3A7-3A4h

HQV Color Adjustment Control 2 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:20	RW	0	**Coefficient C2(s[28:27].[26:20])** (2p)
19:10	RW	0	**Coefficient B2(s[18:17].[16:10])** (2p)
9:0	RW	0	**Coefficient A2(s[8:7].[6:0])** (2p)

Note: Cb'(G') = A2*Y(R) + B2*Cb(G) + C2*Cr(B) + D2

Offset Address: 3AB-3A8h

HQV Color Adjustment Control 3 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:20	RW	0	**Coefficient C3(s[28:27].[26:20])** (2p)
19:10	RW	0	**Coefficient B3(s[18:17].[16:10])** (2p)
9:0	RW	0	**Coefficient A3(s[8:7].[6:0])** (2p)

Note: Cr'(B') = A3*Y(R) + B3*Cb(G) + C3*Cr(B) + D3

Offset Address: 3AF-3ACh

HQV Color Adjustment Control 4 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:20	RW	0	**Coefficient D3(s[28:21].[20])** (2p)
19:10	RW	0	**Coefficient D2(s[18:11].[10])** (2p)
9:0	RW	0	**Coefficient D1(s[8:1].[0])** (2p)

Offset Address: 3B3-3B0h

HQV Horizontal Scale Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**Horizontal Scale Enable** 1: Enable (2p)
30	RO	0	**Reserved**
29:28	RW	00b	**Horizontal Scale Function** (2p) 00: Scale up. 01: 1~1/4 10: $1/4^{-}$~$1/8^{+}$ 11: <1/8
27:15	RO	0	**Reserved**
14:0	RW	0	**Horizontal Scale Factor [14:12].[11:0]** (2p)

Note: Scale factor:

1. Scale up: source/destination
2. 1~1/4: source/(destination+0.5)
3. $1/4^{-}$~$1/8^{+}$: source/destination.
4. <1/8: destination/source

Offset Address: 3B7-3B4h

HQV Vertical Scale Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**Vertical Scale Enable** 1: Enable (2p)
30:29	RO	0	**Reserved**
28	RW	0	**Vertical Scale Function** (2p) 0: Scale up. 1: Scale down
27:17	RO	0	**Reserved**
16:0	RW	0	**Vertical Scale Factor [16:12].[11:0]** (2p)

Note: Scale factor

1. Scale up: source/destination;
2. Scale down: source/(destination+0.5)

PS:
For all scaling calculation, the final results should be rounded to the nearest integer (四捨五入).
For bi-linear factor, please use 6 binary fraction of factor (need be rounded to the nearest 6th fraction) to do the calculation.

Offset Address: 3BB-3B8h

HQV Default Video Color **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RO	0	**Reserved**
29:20	RW	0	**Luma(Y) or Red Color Value**
19:10	RW	0	**Chroma(Cb) or Green Color Value**
9:0	RW	0	**Chroma(Cr) or Blue Color Value**

Offset Address: 3BF-3BCh

HQV De-blocking Factor **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:20	RO	0	**Reserved**
19:18	RO	0	**HQV Current Process Destination Buffer ID**
17	RW	0	**HQV Output Field** 1: Bottom field
16	RW	0	**HQV Current Process Field** 1: Bottom field
15:0	RO	0	**Reserved**

Offset Address: 3C3-3C0h

HQV Sub-picture Frame Buffer Stride and Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:24	RW	0	**MC Flipping Count** For MC flip to HQV path: Read only For SW flip path: R/W HQV update read out register data at the beginning of HQV processing a frame.
23:20	RO	0	**Reserved**
19	RW	0	**Sub-picture Format** 0: AI44 or IA44 1: AYUV (MSB A(8)-Y(8)-U(8)-V(8) LSB)
18	RW	0	**Inverse Alpha Value in AI44 Mode** 1: Inverse (One's Complement)
17	RW	0	**Alpha, Index Exchange in AI44 Mode** 0: AI44 1: IA44
16	RW	0	**HQV Sub-picture Enable** 1: Enable 0: Disable Only active at HQV source format is YUV.
15:14	RO	0	**Reserved**
13:4	RW	0	**Subpicture Frame Buffer Stride** Unit: 16 bytes
3:0	RO	0	**Reserved**

Offset Address: 3C7-3C4h

HQV Sub-picture Frame Buffer Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:4	RW	0	**Sub-picture Frame Buffer Starting Address** (Unit: 16 bytes)
3:2	RO	0	**Reserved**
1:0	RW	0	**Memory Location**

Offset Address: 3CB-3C8h

HQV Sub-picture 4x16 RAM Table Write Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:8	RW	0	**RAM Table Write Data** V: Bits [31:24] U: Bits [23:16] Y: Bits [15:8]
7:4	RW	0	**RAM Table Read / Write Address** Indicate which entry of palette table will be written or read. Palette table contains 16 entries of palette data. (HQV3C8[7:4] = 0x0000 ~ HQV3C8[7:4] = 0x1111) Need to program HQV3C8[31:8] and HQV3C8[7:4] 16 times to fill the subpicture palette table.
3	RO	0	**Reserved**
2	RW	0	**V Write Enable** 0: Disable 1: Enable
1	RW	0	**U Write Enable** 0: Disable 1: Enable
0	RW	0	**Y Write Enable** 0: Disable 1: Enable

Offset Address: 3D3-3D0h

HQV Stream Control and Status **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:28	RW	0	**Video Data Stream Format [3:0]** 0000: RGB32 – (X8R8G8B8) 0001: RGB32 – (X2R10G10B10) 0010: RGB16 – (R5G6B5) 0011: RGB15 – (X1R5G5B5) 0100: YUV444 – (X8Y8U8V8) 0101: V410 – (V10Y10U10X2) 1000: YUV422 – ($V8Y_1 8U8Y_0 8$) 1100: YUV420 – (NV12; planar mode) Others:reserved
27	RW	0	**High Quality Video Enable** 0: Disable 1: Enable
26	RW	0	**Buffer Mode** 0: Double destination buffers 1: Triple destination buffers
25:24	RW	00b	**Video Stream Source [1:0]** 00: SW 01: Reserved 10: Capture 0 11: Capture 1
23	RW	0	**Advanced De-interlace Mode Enable** (reference more than one field)
22	RW	0	**Vertical Low Pass Filter Enable** 0: Disable 1: Enable
21	RO	0	**Reserved**
20	RW	0	**Planar Mode Chrominance Source Data Format** 0: Source Chrominance is saved by frame picture 1: Source Chrominance is saved by field picture
19	RW	0	**Inverse Input Field** 1: Inversed
18	RW	0	**Frame To Field** 1: Frame base source, extract a field from a frame.
17	RW	0	**Field To Frame** 1: Field source, de-interlace to progressive frame.
16	RO	0	**Reserved**
15	RW	0	**Sub-Picture Flip** Software writes 1 to this bit indicates a new sub-picture need to blend. After blending completes, hardware clears it to 0. Software can read this bit to check the status if hardware completes blending.
14:13	RO	0	**Reserved**
12	RO	0	**HQV Flip FIFO Full Status** The FIFO depth defined in Rx3F8 [17:16]
11	RO	0	**Reserved**
10:8	RO	0	**State Machine Status of HQV Flip Module**
7	RW	0	**HQV Interrupt Enable** 0: Disable HQV interrupt. 1: HQV send interrupt signal after done a frame. Relative setting: Rx3D0[0].
6	RW	0	**Single Destination Buffer** 1: Single buffer used Rx3D0[26] would be ignored.
5	RW	0	**Field of Software Source Input** 0: Top 1: Bottom
4	RW	0	**Software Source Flip** Software writes 1 to flip a image to HQV. (Rx3D0[25:24] should be 00b) After processing completes, hardware clears it to 0. Software reads this bit to check flip status.
3	RO	0	**HQV Engine Idle State** 1: Idle
2:1	RO	0	**HQV Output Buffer ID** HQV destination buffer ID
0	RW	0	**HQV End of Frame Status** 1:HQV has been output an image. (Software writes 1 to clear this bit) After HQV done an image, this bit will be pulled high, and it will be pulled down only whent software writes 1b to it.

Offset Address: 3D7-3D4h

HQV SW Source Data – Luma or Packed Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:4	RW	0	**SW Source Data Y or Packed Mode Starting Address** (Unit: 16 bytes)
3:2	RO	0	**Reserved**
1:0	RW	0	**Memory Location**

Offset Address: 3DB-3D8h

HQV SW Source Data – Chroma Starting Address **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:4	RW	0	**SW Source Buffer U, V Starting Addresses** (Unit: 16 bytes)
3:2	RO	0	**Reserved**
1:0	RW	0	**Memory Location**

Offset Address: 3DF-3DCh

HQV Linear / Tile Address Mode, Color Space Conversion, Gamma, De-blocking Control **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:30	RW	00b	**Linear / Tile Address Mode Control** (Tile address only enable at source data from MC, 3D0[25:24]=01) 00: Linear 01: Reserved 10: 256 bits tile mode (Addr = SA + (Y[10:4]*PTH*16) + {X[6:1],Y[3:0],X[0]}) 11: 512 bits tile mode (Addr = SA + (Y[10:4]*PTH*16) + {X[6:2], Y[3:0],X[1:0]}). Where unit of X is 128-bit; unit of Y is line.
29	RW	0	**HQV Output Data Pack In 32-bits Mode.** 0:16 bits (RGB565). 1:32 bits (RGB888) Only valid when the source is YUV and color space conversion is enabled.
28	RW	0	**Color Space Conversion Enable** 0: Disable 1: Enable
27	RW	0	**De-blocking Enable**
26:25	RO	0	**Reserved**
24:20	RW	0	**HQV Output FIFO Threshold for Write Request Control** (Unit: level) HQV output FIFO has 64 levels, once the data in output FIFO touch the threshold (32+3DC.[24:20]), the write request would be triggered.
19:16	RO	0	**Reserved**
15	RW	0	**Constant Alpha of RGB32 Format** 0: Alpha = 00 1: Alpha = FF
14	RW	0	**Enable Synchronization Flipping Field with Interlaced IGA** 1: Enable
13	RW	0	**IGA Field Inverse** 1: Inverse
12:11	RO	0	**Reserved**
10:0	RW	0	**Image Size / 1024** Pull-down detection use.

Offset Address: 3E3-3E0h

HQV Source Data Line Count and Fetch Count Per Line **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:16	RW	0	**Video Source Data Fetch Count Per Line (-1)** Unit: Bytes
15:11	RO	0	**Reserved**
10:0	RW	0	**Video Source Data Line Number (-1)** Unit: Line

Offset Address: 3E7-3E4h

HQV Motion Adaptive De-interlace Control & Threshold **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**2:2 Pull Down Sequence Detection Enable** 0: Disable 1: Enable Relative setting: Rx3DC[10:0]
30:28	RO	0	**Reerved**
27	RW	0	**3:2 Pull Down Sequence Detection Enable** 0: Disable 1: Enable Relative setting: Rx3DC[10:0]
26:25	RO	0	**Reerved**
24	RW	0	**2:3:3:2 Pull Down Sequence Detection Enable** 0: Disable 1: Enable Relative setting: Rx3DC[10:0]
23:13	RO	0	**Reserved**
12:8	RW	0	**Motion Detection Enable**
7	RO	0	**2:2 Pull Down Detection Status**
6	RO	0	**3:2 Pull Down Detection Status**
5	RO	0	**2:3:3:2 Pull Down Detection Status**
4:1	RO	0	**Reserved**
0	RW	0	**Edge Detection Enable** 0: Disable 1: Enable

Offset Address: 3EF-3ECh

HQV Destination Frame Buffer Starting Address 0 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**HQV Output Data Pack In 32 Bits xRGB2-10-10-10 Format After HQV's Color Space Conversion** 1: Enable Note: Only one of {Rx3EC[31], Rx3DC[29]} can be set to 1.
30	RW	0	**Enable Output In Tile Mode** 1:Enable. Addr = ST_ADDR[28:4] + Y[10:3]*{PITCH[10:0], 3'b0} + {X[10:1], Y[2:0], X[0]}
29	RO	0	**Reserved**
28:4	RW	0	**Destination Frame Buffer Starting Address 0** Unit: 16 bytes
3:2	RO	0	**Reserved**
1:0	RW	0	**Memory Location**

Offset Address: 3F3-3F0h

HQV Destination Frame Buffer Starting Address 1 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:4	RW	0	**Destination Frane Buffer Starting Address 1** Unit: 16 bytes
3:2	RO	0	**Reserved**
1:0	RW	0	**Memory Location**

Offset Address: 3F7-3F4h

HQV Destination Frame Buffer Stride **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:14	RO	0	**Reserved**
13:4	RW	0	**Destination Frame Buffer Stride** $_{(2p)}$ Unit: 16 bytes
3:0	RO	0	**Reserved**

Offset Address: 3FB-3F8h

HQV Source Frame Buffer Stride **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31	RW	0	**Load Starting Address Rx3D4[25:4], Rx3D8[25:4] for Advanced De-interlacing** 0: Not used. Hardware keep starting address 1: Load starting address to current field **Note**: Command sequence Step1: Write Rx3D4, Rx3D8 Step2: Write Rx3F8; new address to PN Step3: Write Rx3D4, Rx3D8 Step4: Write Rx3F8; PN to PC; new address to PN Step5: Write Rx3D4, Rx3D8 Step6: Write Rx3F8; PC to PP; PN to PC; new address to PN
30	RW	0	**SJN Reset** 1: Reset static judgment number
29	RW	0	**Pull Down Detection Low-Threshold Value** For fixing bug: Spare register**.**
28	RW	0	**Pull Down Detection Error Sequence Check One Time**
27:26	RW	0	**For Fixing Bug: Spare Register**
25	RW	0	**Not Check Size** 0: Check size 1: Not check size
24:21	RW	0	**For Fixing Bug. Spare Register**
20	RW	0	**Software Flip Queue Enable** 0: Pull Rx3D0[4] low at frame done. 1: Pull Rx3D0[4] low at beginning of processing frame
19	RW	0	**Read Debugging Register**
18	RO	0	**Reserved**
17:16	RW	00b	**FIFO Depth of HQV Flip Control Engine** For hardware flip only. Rx3D0[25:24] ! = 00b. Only supports 2 stages FIFO queuing hardware flipping. 00: Pull "FIFO full status Rx3D0[12]" high, while both two stage are queuing. Drop current processing frame while both two stage are queuing. 01: Pull "FIFO full status Rx3D0[12]" high, while one stage is queuing. Drop current processing frame while both two stage are queuing. 10: Pull "FIFO full status Rx3D0[12]" high, while both two stage are queuing. Never drop current processing frame. 11: Pull "FIFO full status Rx3D0[12]" high, while one stage is queuing. Never drop current processing frame.
15:14	RO	0	**Reserved**
13:4	RW	0	**Source Frame Buffer Stride** (Unit: 16 bytes)
3:0	RO	0	**Reserved**

Offset Address: 3FF-3FCh

HQV Destination Data Starting Address 2 **Default Value: 0000 0000h**

Bit	Attribute	Default	Description
31:29	RO	0	**Reserved**
28:4	RW	0	**Destination Data Starting Address 2** Unit: 16 bytes
3:2	RO	0	**Reserved**
1:0	RW	0	**Memory Location**

Note: HQV supports 2 Video Streams; therefore, an additional register space is provided to match the above registers definition. Writing a register to this space, it will write to the second HQV, which output to V3 and source is from the second MC engine.

The relationship between the additional register space and the original register space is
(The additional register address) = (The original register address) + 16'h1000.

3D REGISTERS

This chapter provides detailed 3D register descriptions are followed in the sequent sections. Offsets Rx1Ch to Rx3Bh can both be used for setting the Command Regulator registers with the same HParaType 10h and HparaType 11h. Settings through Rx1Ch would not enable the 3D Engine clock, while settings through Rx3Ch would enable the 3D Engine clock.

Definition of I/O Register

The I/O Register Base Address for 3D is 400h.

For Write Mode

Setting of Command Regulator

<table>
<tr><th>Scope</th><th>Offset</th><th>Description</th><th>Mnemonic</th></tr>
<tr><td rowspan="4">Transmission Setting</td><td>1Ch</td><td>The Beginning of Internal Address for Parameter Programming</td><td>HParaAdr</td></tr>
<tr><td>1Dh</td><td>Offset Setting for Some Special Parameter Types</td><td>HParaOS</td></tr>
<tr><td>1Eh</td><td>Parameter Type

<table><tr><th>HParaType</th><th>Description</th></tr><tr><td>0000 0000 ~ 0000 1111</td><td>Reserved</td></tr><tr><td>0001 0000</td><td>Command Decoded in front of Command Regulator</td></tr><tr><td>0001 0001 ~ 1111 1101</td><td>Reserved</td></tr><tr><td>1111 1110</td><td>Frame Swapping</td></tr><tr><td>1111 1111</td><td>Reserved</td></tr></table>
For more details for the parameters, please refer to Definition of Parameter section.</td><td>HParaType</td></tr>
<tr><td>1Fh</td><td>Parameter Type Sub-code</td><td>HParaSubType</td></tr>
<tr><td rowspan="5">Transmission Space</td><td>23h-20h</td><td>Parameter 0</td><td>Hpara0</td></tr>
<tr><td>27h-24h</td><td>Parameter 1</td><td>Hpara1</td></tr>
<tr><td>2Bh-28h</td><td>Parameter 2</td><td>Hpara2</td></tr>
<tr><td>…….</td><td>…….</td><td></td></tr>
<tr><td>3Bh-38h</td><td>Parameter 7</td><td>Hpara7</td></tr>
</table>

Setting of 3D Engine

Scope	Offset	Description	Mnemonic
Transmission Setting	3Ch	The Beginning of Internal Address for Parameter Programming	HParaAdr
	3Dh	Offset Setting for Some Special Parameter Types	HParaOS

	3Eh	**Parameter Type** **HParaType**	**HParaType**

HParaType	Description
0000 0000	Primitive Vertex Data or Vertex Index
0000 0001	Attribute Other Than Texture
0000 0010	Attribute of Texture n The setting for texture is divided into 2 kinds of stages: One is defined as Texture Stage with HParaSubType 0nH. Another type is Texture Sample Stage with HParaSubType 02nH. n is the number stored in HParaSubType (Rx3F): 0000 0000 = Texture 0 0000 0001 = Texture 1 0000 0010 = Texture 2 0000 0011 = Texture 3 0000 0100 = Texture 4 0000 0101 = Texture 5 0000 0110 = Texture 6 0000 0111 = Texture 7 0000 1000 = Texture 8 0000 1001 = Texture 9 0000 1010 = Texture A 0000 1011 = Texture B 0000 1100 = Texture C 0000 1101 = Texture D 0000 1110 = Texture E 0000 1111 = Texture F n is the number stored in HParaSubType(Rx3F): 0010 0000 = Texture 0 0010 0001 = Texture 1 0010 0010 = Texture 2 0010 0011 = Texture 3 0010 0100 = Texture 4 0010 0101 = Texture 5 0010 0110 = Texture 6 0010 0111 = Texture 7 0010 1000 = Texture 8 0010 1001 = Texture 9 0010 1010 = Texture A 0010 1011 = Texture B 0010 1100 = Texture C 0010 1101 = Texture D 0010 1110 = Texture E 0010 1111 = Texture F 1111 1110 = General Texture Setting The use of 1111 1111 is prohibited. Others = Reserved
0000 0011	Palette HParaSubType (Rx3F) shows the palette type: 0000 0000 = Texture Palette 0 (Only for AI44 and IA44 Video Texture) 0000 0001 ~ 0000 0111 = Reserved 0000 1xxx = Reserved 0001 0000 = Base Address Offset of Each Texture 0001 0001 = Texture 4X4 Filter Coefficient Table 0001 0010 = Optimized Merge Rules for Coarse Z Test 0001 0011 = Resreved 0001 0100 = Stipple Palette 0001 0101 = De-Gamma Table for Reading Texture 0001 0110 = Reserved 0001 0111 = Gamma-de-Gamma Table for Writing Destination Color 0010 0000 = Pixel Shader ALU Instruction 0010 0001 = Pixel Shader TAU Instruction 0010 0010 = Pixel Shader Constant Register
0000 0100	Vertex and Primitive Setting
0000 1111 ~ 0000 0101	Reserved
0001 0000	Command Decoded in front of Command Regulator
0001 0001	Command for Frame Buffer AutoSwapping and CR's Miscellaneous Setting
1111 1101 ~ 0001 0010 ~ 1111 1111	Reserved

For more details for the parameters, please refer to Definition of Parameter section.

	3Fh	Parameter Type Sub-code	HParaSubType
Transmission Space	43h-40h	Parameter 0	Hpara0
	47h-44h	Parameter 1	Hpara1
	4Bh-48h	Parameter 2	Hpara2
	…….	…….	
	1F7h-1F4h	Parameter 109	Hpara6D
	1FBh-1F8h	Parameter 110	Hpara6E
	1FFh-1FCh	Parameter 111	Hpara6F
	……	…….	
	2FFh-2FCh	Parameter 175	HparaAF

For Read Mode

IO Address 400h to 41Ch are used for general purpose Reading-Back registers.
IO Address 420h to 42Ch are used for Reading-Back registers of E32CR_WBREG subset which is addressed by HSetRBGAdr.
IO Address 430h to 5FCh are used for Reading-Back registers of specific sub-engine set by HSetRGBID which could be read-back as "HRRGBID".
Definition of HSetRGBID

Bit [7:0]	HSetRBGID	ID for Reading-Back Register
	0000 0000:	Reading the RB registers from CR
	0000 0001:	Reading the RB registers from FE(including VP, CL and SE)
	0000 0010:	Reading the RB registers from PE(including RZ and CZ)
	0000 0011:	Reading the RB registers from RC
	0000 0100:	Reading the RB registers from PS
	0000 0101:	Reading the RB registers from XE
	0000 0110:	Reading the RB registers from BE(including GEMI)
	Others:	Reserved

General Purpose Reading_Back Registers (00h-2Fh)

Offset Address: 00-03h

Engine Status

Bit	Description	Mnemonic
Engine Status		
31:24	**ID for Reading-Back Register** 0000 0000: Reading the RB registers from CR 0000 0001: Reading the RB registers from FE(including VP, CL and SE) 0000 0010: Reading the RB registers from PE(including RZ) 0000 0011: Reading the RB registers from RC 0000 0100: Reading the RB registers from PS 0000 0101: Reading the RB registers from XE 0000 0110: Reading the RB registers from BE(including GEMI) Others: Reserved	**HRRBGID**
23	**Reserved**	
22	**CR AutoFBSW Status** 0: Idle 1: Busy	**HRCRFBSWST**
21	**LCD DN Status** 0: Idle 1: Busy	**HRLCDDNST**
20	**DMA(4channel) Status** 0: Idle 1: Busy	**HRDMAST**
19	**Reserved**	
18	**HQV-V1 Engine Status** 0: Idle 1: Busy	**HRHQV1St**
17	**HQV-V3 Engine Status** 0: Idle 1: Busy	**HRHQV3St**
16	**Sequencer Status (T_Arbiter 0)** 0: Idle 1: Busy	**HRSEQ0St**
15	**Sequencer Status (T_Arbiter 1)** 0: Idle 1: Busy	**HRSEQ1St**
14	**Reserved**	
3D Status		
13	**GEMI(including GEMI0 and GEMI1) Status** 0: Idle 1: Busy	**HRGEMIst**
12	**Burst Engine Status** 0: Idle 1: Busy	**HRBESt**
11	**Pixel Engine Status** 0: Idle 1: Busy	**HRXESt**
10	**Pixel Shader Status** 0: Idle 1: Busy	**HRPSSt**
9	**Reading Color Engine Status** 0: Idle 1: Busy	**HRRCSt**
8	**Primitive Engine(including RZ engine) Status** 0: Idle 1: Busy	**HRPESt**
7	**Setup Engine Status** 0: Idle 1: Busy	**HRSESt**

6	**Clipping Engine Status** 0: Idle 1: Busy	**HRCLSt**
5	**Vertex Processor Status** 0: Idle 1: Busy	**HRVPSt**
4	**Command Regulator Status** 0: Idle 1: Busy	**HRCRSt**
3	**Reserved**	**HRMPGSt**
2	**CMDQ Status** 0: Idle 1: Busy	**HRVQSt**
1	**2D Engine Status** 0: Idle 1: Busy	**HRE2St**
0	**3D Engine Status** 0: Idle 1: Busy	**HRE3St**

Offset Address: 04-07h
Current Command SW Flag

Bit	Description	Mnemonic
31:0	**SW Inspect Flag of Current Command Header**	**HRCRFlag**

Offset Address: 08-0Bh
3D Command SW Flag

Bit	Description	Mnemonic
31:0	**SW Inspect Flag of 3D Command**	**HRE3Flag**

Offset Address: 0C-0Fh
2D Command SW Flag

Bit	Description	Mnemonic
31:0	**SW Inspect Flag of 2D Command**	**HRE2Flag**

Offset Address: 10-13h
MC Command SW Flag

Bit	Description	Mnemonic
31:0	**SW Inspect Flag of MC Command**	**HRMCFlag**

Offset Address: 14-17h
Read Global Register Address

Bit	Description	Mnemonic
31:24	**Reserved**	
23:0	**Read Global Register Address**	**HRRBGAdr**

Offset Address: 18-1Bh
Status and Result of Latched FIFO Auto Testing

Bit	Description	Mnemonic
31	**Error for Latched FIFO Auto Testing** 0: Pass 1: Fail	**HRFIFOATError**
30:24	**Reserved**	
23	**GEMI's Auto Testing Result** 0: Pass 1: Fail	**HRATFGEMI**
22	**BE's Auto Testing Result** 0: Pass 1: Fail	**HRATFBE**
21	**XE's Auto Testing Result** 0: Pass 1: Fail	**HRATFXE**
20	**PS's Auto Testing Result** 0: Pass 1: Fail	**HRATFPS**
19	**RC's Auto Testing Result** 0: Pass 1: Fail	**HRATFRC**
18	**PE's Auto Testing Result (Including RZ&HZ)** 0: Pass 1: Fail	**HRATFPE**
17	**FE's Auto Testing Result** 0: Pass 1: Fail	**HRATFFE**
16	**Reserved**	
15	**Busy for Latched FIFO Auto Testing** 0: Finished 1: Busy, be testing	**HRFIFOATBusy**
14:8	**Reserved**	
7	**Busy for GEMI's Auto Testing** 0: Finished 1: Busy, be testing	**HRATFGEMIBusy**
6	**Busy for BE's Auto Testing** 0: Finished 1: Busy, be testing	**HRATFBEBusy**
5	**Busy for XE's Auto Testing** 0: Finished 1: Busy, be testing	**HRATFXEBusy**
4	**Busy for PS's Auto Testing** 0: Finished 1: Busy, be testing	**HRATFPSBusy**
3	**Busy for RC's Auto Testing** 0: Finished 1: Busy, be testing	**HRATFRCBusy**
2	**Busy for PE's Auto Testing** 0: Finished 1: Busy, be testing	**HRATFPE**
1	**Busy for FE's Auto Testing** 0: Finished 1: Busy, be testing	**HRATFFEBUSY**
0	**Reserved**	

Offset Address: 1C-1Fh
Command Head SW Flag from E3's BE

Bit	Description	Mnemonic
31:0	SW Inspect Flag of Command Header from E3's BE	HRCMDHFlagBE

Offset Address: 20-23h
3D Global Register 0

Bit	Description	Mnemonic
31:0	E32CR_WBREG[31:0]	HE3WBReg0

Offset Address: 24-27h
3D Global Register 1

Bit	Description	Mnemonic
31:0	E32CR_WBREG[63:32]	HE3WBReg1

Offset Address: 28-2Bh
3D Global Register 2

Bit	Description	Mnemonic
31:0	E32CR_WBREG[95:64]	HE3WBReg2

Offset Address: 2C-2Fh
3D Global Register 3

Bit	Description	Mnemonic
31:0	E32CR_WBREG[127:96]	HE3WBReg3

CR Reading Back Registers (30h-9Fh)

Offset Address: 30-33h

CR Miscellaneous Status 0

Bit	Description	Mnemonic
31	**CR Clock Enable** 0: Gating clock 1: Enable	**CR_CLK_EN**
30	**E2 Clock Enable** 0: Gating clock 1: Enable	**E2_CLK_EN**
29	**E3 Clock Enable** 0: Gating clock 1: Enable	**E3_CLK_EN**
28	**VIDEO Clock Enable** 0: Gating clock 1: Enable	**MC_CLK_EN**
27	**Command Acceptable**	**CR_ACTIVE**
26	**CR4 ACK**	**CR4_ACK**
25	**CR4 E-pip ACK**	**CR4E_ACK**
24	**CR4 V-pip ACK**	**CR4V_ACK**
23	**CR3 FMTDEC ACK**	**FMTDEC_ACK**
22	**CR3 FMTDEC Ready**	**FMTDEC_RDY**
21	**Interrupt State On**	**INT_CMD**
20	**Fence State On**	**FENCE_CMD**
19	**Lock E2 Request**	**E2CMD_LOCK**
18	**Lock E3 Request**	**E3CMD_LOCK**
17	**F2 Empty**	**HRCRF2Empty**
16	**F1 Empty**	**HRCRF1Empty**
15	**F2 Full**	**HRCRF2Full**
14	**F1 Full**	**HRCRF1Full**
13:7	**F2 Ready Counter Output**	**F2_CNT_RDY[6:0]**
6:0	**F1 Ready Counter Output**	**F1_CNT_RDY[6:0]**

Offset Address: 34-37h

CR Miscellaneous Status 1

Bit	Description	Mnemonic
31	**Fence Command Interrupt Flag**	**CR_INT**
30	**3D to CR Interrupt Enable**	**E32CR_INTEN**
29	**CR to E3 Interrupt Request**	**CR2E3_INTR**
28	**MC to CR ACK**	**MC2CR_GOTREG**
27	**E2 to CR ACK**	**E22CR_RDY**
26	**E3 to CR ACK**	**E32CR_RDY**
25	**CR to Video Ready**	**CR2V_W**
24	**CR to MC Ready**	**CR2MC_W**
23	**CR to E2 Ready**	**CR2E2_W**
22	**CR to E3 Ready**	**CR2E3_RDY**
21	**CR to E3 AGP Flag**	**CR2E3_AGP**
20:10	**CR to Video Address**	**CR2V_WADR[12:2]**
9:0	**CR to E3 Address**	**CR2E3_WADR[11:2]**

Offset Address: 38-3Bh
CR Miscellaneous Status 2

Bit	Description	Mnemonic
31	Branch Request Cycle	BRANCH_REQ_CYC
30	Branch Get Data Cycle	BRANCH_DAT_CYC
29	Branch Request for Channel 0	BRANCH_CRRCMREQ0
28	Branch Request for Channel 1	BRANCH_CRRCMREQ1
27:21	Number of Branch Data Have Requested But None Back	BRANCH_AGP_RB_CNT[6:0]
20:19	Fence Command Status	FENCE_STATE[1:0]
18:16	Reserved	
15	Branch for Re-store Enable	HRSTBRANCH
14	Branch for Re-store Lock	BRANCH_LOCK
13	Lock V-pip with 326C	LOCK6C_V
12	Lock E-pip with 6C	LOCK6C_E
11:6	V3 Lock Register	WAIT_BUS_V3[5:0]
5:0	V1 Lock Register	WAIT_BUS_V1[5:0]

Offset Address: 3C-3Fh
AGP Status

Bit	Description	Mnemonic
31:4	Current Executed AGP Command	HRAGPBCUR
3:0	Reserved	

Offset Address: 40-43Fh
AGP Status

Bit	Description	Mnemonic
31:4	Current AGP Command Request Address	HRAGPBCUR
3:2	Reserved	
1	AGP Test Mode (Header 6) Command Error on CR0	CR_ERR1
0	AGP Test Mode (Header 6) Command Error on CR3	CR_ERR0

Offset Address: 44-47h
AGP Status & VQ Status

Bit	Description	Mnemonic
31	AGP_PAUSE &(HAGPBPID == 2'b00)	HRAGPCYCPause
30	AGP Command Cycle Flag	AGP_CYC
29	AGP Command Cycle Active	AGP_SEL
28	AGP Command Same Pause Address	SAME_BP
27	AGP Command Same End Address	SAME_BEND
26	AGP Command Same Jump Address	SAME_BJUMP
25	Pause AGP Cycle	AGP_PAUSE
24	AGP Command Request	AGP_CRRCMREQ
23	AGP Command ACK	CRRCMACK
22	AGP Request Hold	AGP_HOLD
21	AGP Request VQ Stop	VQ_AGP_STOP
20	AGP Cycle Last Request Command	LAST_REQ
19	Stop to Read AGP Command	AGP_REQ_STOP
18:10	Number of AGP Data Have Requested But None Back	AGP_RB_CNT[8:0]
9:0	Virtual Queue V cnt in Frame Buffer	VQ_CNT[9:0]

Offset Address: 48-4Bh
VQ Status

Bit	Description	Mnemonic
31	**Virtual Queue V Cycle**	**VQ_CYC**
30	**Virtual Queue V Full**	**VQ_FULL**
29	**Virtual Queue V has Valid Data**	**VQ_RDY**
28:4	**Virtual Queue V Address**	**CRCMDQ_ADR[28:4]**
3	**Virtual Queue V Enable**	**HENCMDQ**
2	**Virtual Queue V for AGP**	**HCMDQ4AGP**
1	**Virtual Queue V MI to CR Acknowledge**	**MI2CRTCM_ACK**
0	**HENVQ_V & MI2CR_ACK_V**	**MI2CR_ACK**

Offset Address: 4C-4Fh
Interrupt Status

Bit	Description	Mnemonic
31:29	**Interrupt Status**	**INT_STATE[2:0]**
28	**Stop 3D Vertex**	**STOP_VERTEX**
27:24	**Number of Restore Data have Requested but None Back**	**RST_REG_CNT[3:0]**
23:0	**Number of CR3 Data have Requested but None Back** The Left Vertex number within one DIP.	**AGPFD_CNT[23:0]**

Offset Address: 50-53h
Flip Count

Bit	Description	Mnemonic
31:16	**Frame Buffer Switching Count R Register for IGA2**	**HFLIPCNT2[15:0]**
15:0	**Frame Buffer Switching Count R Register for IGA1**	**HFLIPCNT1[15:0]**

Offset Address: 54-57h
FBSW Control 1 Status

Bit	Description	Mnemonic
31:0	**FBSW Control 1 Status** { CRFBSW2IGA, FBSW_IDLE, VLD2IGA, DP_BAS_LCD[28:3], DP2FBSWACK, DP_LOC[1:0] }	**HRDISPLAYBST1[31:0]**

Offset Address: 58-5Bh
FBSW Control 2 Status

Bit	Description	Mnemonic
31:0	**FBSW Control 1 Status** { CRFBSW2IGA, FBSW_IDLE, VLD2IGA, DP_BAS_LCD[28:3], DP2FBSWACK, DP_LOC[1:0] }	**HRDISPLAYBST2[31:0]**

Offset Address: 5C-5Fh
AGP Pause Address

Bit	Description	Mnemonic
31:4	**AGP Pause Address**	**HAGPBP[31:4]**

Offset Address: 60-63h
AGP Jump Address

Bit	Description	Mnemonic
31:4	**AGP Jump Address**	**HAGPBJUMP[31:4]**

Offset Address: 64-67h
AGP Start Address

Bit	Description	Mnemonic
31:4	AGP Buffer Start Address	HAGPBST[31:4]
3:2	Reserved	
1:0	AGP Buffer Start Address Location	HAGPBLOC

Offset Address: 68-6Bh
AGP End Address

Bit	Description	Mnemonic
31:4	AGP End Address	HAGPBEnd[31:4]
3:0	Reserved	

Offset Address: 6C-6Fh
SW Event Tag AA

Bit	Description	Mnemonic
31:24	Reserved	
23:0	SW Event Tag AA	HRSWFLAGAA

Offset Address: 70-73h
SW Event Tag AB

Bit	Description	Mnemonic
31:24	Reserved	
23:0	SW Event Tag AB	HRSWFLAGAB

Offset Address: 74-77h
Fence Command ID Valid

Bit	Description	Mnemonic
31:4	Reserved	
3:0	Fence Command ID Valide	HRFC_VLD

Offset Address: 78-7Bh
CR's Miscellaneous Status 3

Bit	Description	Mnemonic
31	F3 Full	HRCRF3Full
30	F3 Empty	HRCRF3Empty
29:23	F3 Ready Counter Output	F3_CNT_RDY[6:0]
22:13	Virtual Queue E cnt in Frame Buffer	VQ_CNT_E[9:0]
12	Virtual Queue E Cycle	VQ_CYC_E
11	Virtual Queue E Full	VQ_FULL_E
10	Virtual Queue E has Valid Data	VQ_RDY_E
9	Virtual Queue E Enable	HENCMDQ_E
8	Virtual Queue E for AGP	HCMDQ4AGP_E
7	CR6 FMTDEC ACK	FMTDEC_ACK_E
6	CR6 FMTDEC Ready	FMTDEC_RDY_E
5	CR7ACK	CR7_ACK
4	Lock V-pip with 326C & VMR Lock	LOCK_V
3	Lock E-pip with 6C & VMR Lock	LOCK_E
2	CR_VQ_REQFIFO EMPTY	RF_EMPTY
1	CR_VQ_REQFIFO Full	RF_FULL
0	Virtual Queue E MI to CR Acknowledge	MI2CR_ACK_E

Offset Address: 7C-7Fh
Lock Control 1

Bit	Description	Mnemonic
31:8	Read Command Stream0 Buf Status	HRCS1FSM
7	For VMR Path HQV Busy Status	HQV_BUSY
6	VMR Lock V-pip	VMR_LOCK_V
5	Lock V-pip with 326C	LOCK6C_V
4	VMR Lock E-pip	VMR_LOCK_E
3	Lock E-pip with 6C	LOCK6C_E
2	Lock, Wait V-Blank	LOCK_VBLK
1	Video Command Start	V_START
0	2D / 3D Command Start	E_START

Offset Address: 80-83h
Lock Control 2

Bit	Description	Mnemonic
31:8	Read Command Stream 1 Buf Status	HRCS2FSM
7:0	Start_ID of Output Command	START_ID

Offset Address: 84-87h
Lock Control 3

Bit	Description	Mnemonic
31:16	For VMR Path, V Buffer Lock Status	HRLOCK_V
15:0	For VMR Path, E Buffer Lock Status	HRLOCK_E

Offset Address: 88-8Bh
Lock Control 4

Bit	Description	Mnemonic
31:16	For VMR Path, V Buffer Working ID	V_WORKINGID
15:0	For VMR Path, E Buffer Working ID	E_WORKINGID

Offset Address: 90-93h
Fence Command ID A

Bit	Description	Mnemonic
31:0	Fence Command ID A	HRFC_PCIADD_A

Offset Address: 94-97h
Fence Command ID A

Bit	Description	Mnemonic
31:0	Fence Command ID A	HRFC_ID_A

Offset Address: 98-9Bh
Fence Command ID B

Bit	Description	Mnemonic
31:0	Fence Command ID B	HRFC_PCIADD_B

Offset Address: 9C-9Fh
Fence Command ID B

Bit	Description	Mnemonic
31:0	**Fence Command ID B** Fence Queue read by mmio read address **by mmioRead_490 and mmioRead_498 to add read point.**	HRFC_ID_B

HParaType 00h: Primitive Vertex Data or Vertex Index

HParaTye 00h is used for Primitive Vertex Data (for Vertex Command Mode) or Vertex Index (for Vertex Buffer Index Mode). There is no sub-address in this ParaType, and the steps of how to fire 3D Engine are followed:

Step 1: Set the correct value to all the related 3D globe registers through ParaType 01h, 02h, 03h, 04h, 10h, 11h and FEh.

Step 2: Sent Vertex Data or Vertex Index (according to HVertexMode) through ParaType 00h. CR would configure the vertex (according HVFVLEN and HnFVF) and generate fire signal automatically whenever all the vertices of a primitive are ready (according to HVCycle, H2and1VT and HPMType). CR would also generate the PLEND (Primitive List End) signal whenever the entire listed vertexes are finished (according to HVTXnum), or receive a "Stop Command".

For next primitive list, repeat the above two steps.

HParaType 01h: Attribute Other Than Texture

HParaType = 01h

Sub-Address 00h-0Fh: Enable Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
00h	23	**Reserved**	
	22	**Inverse Enable (disable) BE's 32-byte (adiacent 128-bit) Packing** 0: Enable 1: Disable	**Hen32BytePack_N**
	21	**Inverse Enable (disable) BE's Smart Packing** 0: Enable 1: Disable	**HenSMRTPack_N**
	20	**Enable Alpha Test Result of RT0 for all Render Targets** 0: If Alpha Test pass or fail of RTn depends on its own alpha test result and HenATMRTn. n = 0, 1, 2 and 3 1: Do the alpha test of RT0 and all render targets depend on the result to be killed or not. HenATMRT0, HenATMRT1, HenATMRT3 & HenATMRT3 are ignored	**HenAT4allRT**
	19	**Enable Alpha Test for Render Target 3** 0: Disable 1: Enable	**HenATMRT3**
	18	**Enable Alpha Test for Render Target 2** 0: Disable 1: Enable	**HenATMRT2**
	17	**Enable Alpha Test for Render Target 1** 0: Disable 1: Enable	**HenATMRT1**
	16	**Enable Alpha Test for Render Target 0** 0: Disable 1: Enable	**HenATMRT0**
	15	**Enable Specula Color for Render Target 3** 0: Disable 1: Enable	**HenSCMRT3**
	14	**Enable Specula Color for Render Target 2** 0: Disable 1: Enable	**HenSCMRT2**
	13	**Enable Specula Color for Render Target 1** 0: Disable 1: Enable	**HenSCMRT1**
	12	**Enable Specula Color for Render Target 0** 0: Disable 1: Enable	**HenSCMRT0**
	11	**Enable Fog for Render Target 3** 0: Disable 1: Enable	**HenFOGMRT3**
	10	**Enable Fog for Render Target 2** 0: Disable 1: Enable	**HenFOGMRT2**
	9	**Enable Fog for Render Target 1** 0: Disable 1: Enable	**HenFOGMRT1**
	8	**Enable Fog for Render Target 0** 0: Disable 1: Enable	**HenFOGMRT0**
	7	**Enable Alpha Blending for Render Target 3** 0: Disable 1: Enable	**HenABLMRT3**
	6	**Enable Alpha Blending for Render Target 2** 0: Disable 1: Enable	**HenABLMRT2**
	5	**Enable Alpha Blending for Render Target 1** 0: Disable 1: Enable	**HenABLMRT1**
	4	**Enable Alpha Blending for Render Target 0** 0: Disable 1: Enable	**HenABLMRT0**
	3	**Enable Dither for Render Target 3** 0: Disable 1: Enable	**HenDTMRT3**
	2	**Enable Dither for Render Target 2** 0: Disable 1: Enable	**HenDTMRT2**
	1	**Enable Dither for Render Target 1** 0: Disable 1: Enable	**HenDTMRT1**
	0	**Enable Dither for Render Target 0** 0: Disable 1: Enable	**HenDTMRT0**

01h	23	**Enable Line drawing from Up(Left) to Down(Right)** 0: Draw line from vertex a to vertex b. 1: For vertical line, draw from up vertex to down vertex. For horizontal line, draw from left vertex to right vertex.	**HenLUL2DR**
	22	**Enable Diamond Rule for Line Drawing** 0: Disable 1: Enable	**HenLDiamond**
	21	**Enable Point Sprite** 0: Disable 1: Enable This setting is only available for PMType "point" and "triangle point". All Points have sizes (default is 1.0), and Point Sprite only affects the texture coordinate. *If non-PointSprite points:* All the texture coordinates of point 4 vertexes are all those from FVF. *If PointSprite points:* Texture coordinate of the upper-left corner is (0, 0, 0, 1) Texture coordinate of the upper-right corner is (1, 0, 0, 1) Texture coordinate of the lower-left corner is (0, 1, 0, 1) Texture coordinate of the lower-right corner is (1, 1, 0, 1)	**HenPSprite**
	20	**Enable Clipping Coordinate to Screen Coordinate Transformation** 0: Disable 1: Enable	**HenC2S**
	19	**Enable Fog Perspective Correction** 0: Disable 1: Enable	**HenFOGPP**
	18	**Enable Spectra Color Perspective Correction** 0: Disable 1: Enable	**HenSCPP**
	17	**Enable Diffuse Color Perspective Correction** 0: Disable 1: Enable	**HenCPP**
	16:14	**Number of Render Target** 000: Only RT0 001: RT0 & RT1 010: RT0, RT1 and RT2 011: RT0, RT1, RT2 and RT3 1xx: Reserved	**HMRTnum**
	13	**Enable of Coarse Z Test** 0: Disable 1: Enable Note that Coarse Z Test is only available for triangles, not the lines.	**HenCZ**
	12	**Enable Vertex Cache** 0: Disable 1: Enable	**HenVC**
	11	**Enable Clipping Engine** 0: Disable 1: Enable	**HenCL**
	10	**Enable Pixel Shader** 0: Disable (Texture map is also disabled) 1: Enable	**HenPS**
	9	**Enable Writing Coarse Z Buffer** 0: Don't update the Coarse Z Buffer 1: Enable	**HenWCZ**
	8	**Enable Texture Cache** 0: Disable Texture Cache and Clear Texture Cache 1: Enable	**HenTXCH**
	7	**Enable Back Face Culling** 0: Disable 1: Enable	**HenBFCull**
	6	**Enable Color Write** 0: Disable 1: Enable	**HenCW**
	5	**Enable Anti-Aliasing** 0: Disable 1: Enable	**HenAA**
	4	**Enable Stencil Test** 0: Disable (always pass) 1: Enable (Depth buffer must contain stencil bits)	**HenST**
	3	**Enable Z Test** 0: Disable (always pass) 1: Enable	**HenZT**
	2	**Enable Z Write** 0: Disable 1: Enable	**HenZW**
	1	**Enable Stipple Pattern** 0: Disable 1: Enable	**HenSP**
	0	**Enable Line Pattern** 0: Disable 1: Enable	**HenLP**
02-0Fh	23:0	**Reserved**	

HParaType = 01h

Sub-Address 10h-22h: Z Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
10h	23:0	**ZW Buffer Base Address** In unit of 256 bytes.	**HZWBBas**
11h	23	**Reserved**	
	22:13	**ZW Buffer Pitch** In unit of 32 bytes for linear mode. In unit of tile (256 bytes) for tile mode.	**HZWBPit**
	12	**Enable Reading-Z Cache** 0: Disable 1: Enable	**HZenRZCache**
	11	**Clear Reading-Z Cache**	**HZRZCClr**
	10	**Mode of Reading-Z Cache** 0: 128-bit Mode 1: 256-bit Mode	**HZRZCMode**
	9	**Z and Stencil Value are written through to BE directly**	**HZSTWTH2BE**
	8:2	**Reserved**	
	1:0	**Location Setting of Z Buffer** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HZWBLoc**
12h	23	**ZW Buffer Type** 0: ZW buffer stores Z value **1: ZW buffer stores W value (PP replaces Z by W)**	**HZWBType**
	22	**Reserved**	
	21	**Force the Z Value from 1.0 to 1.0- (from 1.00000000h to 0.FFFFFFFFh)** 0: Keep the original 1.0 1: Force to 1.0- whenever Z equal to 1.0 This register is only useful for fixed-point format.	**HZONEasFF**
	20	**Clamp the Z Value is over 1.0 to 1.0- (from 1.xxxxxxxxh to 0.FFFFFFFFh)** 0: Keep the original value over 1.0 1: Clamp to 1.0- whenever Z is over 1.0 This register is only useful for fixed-point format.	**HZOONEasFF**
	19	**Clamp the Z Value is Negative to Zero- (from –x.xxxxxxxxh to 0.00000000 h)** 0: Keep the original negative value 1: Clamp to 0.0- whenever Z is negative Note: These clamping registers do not influence the test of nearby or distant plane. The tests of the nearby or distant plane is by the original biased and un-clamped Z value with floating format.	**HZNEGasZERO**
	18:16	**ZW Buffer Format** For Z Buffer 000: 16-bit fix point format, 0.0 ≤ Z < 1.0 001: 16-bit floating format s[5].10 from +2^31*1.FFFF to –2^31*1.FFFF 010: Reserved 011: Reserved 100: 32-bit fix point format, 0.0 ≤ Z < 1.0 101: 32-bit fix point format s[8].23 110: 24-bit fix point format Z, 0.0 ≤ Z < 1.0, and Stencil Z is located in bit [31:8], Stencil is located in bit [7:0]	**HZWBFM**
	15:7	**Reserved**	
	6:5	**Location Setting of Separated Stencil Buffer** 00: Syntem Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HZWBLoc**
	4	**Write Stencil Value of the Separated Stencil Buffer to the Z Buffer** 0: Normal 1: Force “STVALID” as true and stencil operation as “Keep”, so the stencil value in the separated buffer can be filled into Z buffer.	**HSTB2ZB**

	3	**Synchronized with the Separated Stencil Value in Z Buffer** If the stencil value in the separated stencil buffer is synchronized with the stencil value in Z buffer 0: No synchronization. HW just update the stencil value in the separated stencil buffer, which may not be synchronized to the stencil value in Z buffer. 1: The stencil value in the separated stencil buffer is synchronized to the stencil value in Z buffer. HW would update the separated stencil buffer and Z buffer.	**HSTBSync2ZB**
	2	**If there is a Separated Stencil Buffer** 0: No separated stencil buffer 1: There is a separated stencil buffer	**HSTBSeperated**
	1	**Extend for Z Format Transformation** 0: Do nothing Consider Z with 32-bit floating s[8].23, its mantissa 1.Z[22:0]. We extend it to 1.{Z[22:0], 8'h00} and the transformed to fix format. 1: Extend mantissa to 32 bits before format transformation Consider Z with 32-bit floating s[8].23, its mantissa 1.Z[22:0]. We extend it to 1.{Z[22:0], 1'b1, Z[22:16]} and the transformed to fix format.	**HZWExtend**
	0	**Memory Mode of ZW Buffer** 0: Linear mode 1: Tile mode	**HZWMMode**
13h	23	**Source Z is generated by Pixel Shader instead of Shading** 0: Source Z is generated by normal shading 1: Source Z is generated in PS, RZ module should pipe the Zdst to PE. The Zdst and Zsrc for depth test come from BE.	**HZSrcPS**
	22:19	**Reserved**	
	18:16	**ZW Test Mode** 000: Z or W Test Never Pass 001: Z or W Test Pass if Znew < Zdst 010: Z or W Test Pass if Znew = Zdst 011: Z or W Test Pass if Znew ≤ Zdst 100: Z or W Test Pass if Znew > Zdst 101: Z or W Test Pass if Znew ≠ Zdst 110: Z or W Test Pass if Znew ≥ Zdst 111: Z or W Test Always Pass Where Znew is the calculated Z value and Zdst is the Z stored in the Z buffer.	**HZWTMD**
	15:8	**Reserved**	
	7:0	**Z Normalization Factor** The range is from 0 to 255. We define that Z can be divided by a value of power of 2. Thus, it allows the input Z free from the constrain of Z < 1. Z Normalization is $Z = Zin / 2^{HZNF}$	**HZNF**
14h	23:0	**Lower 3 Bytes of ZW Clear Data**	**HZWCDL**
15h	23:8	**Negative of Mask to the Zsrc's last 16-bit Mantissa** Note that this mask is just implemented to the rendered Z value for format transformation, then a Z test is conducted and wrote-back to Z buffer Consider the generated Z value "Zsrc[31:0]" with floating s[8].23 Step1:Zsrc[15:0] = Zsrc[15:0] & ~HZWMMSK_N[15:0] Step2: Format transform Zsrc according to HZWFM Step3: Z test Step4: Updated Z buffer with the format transformed Zsrc if Z test is passed	**HZWMMSK_N**
	7:0	**Highest Byte of ZW Clear Data**	**HZWCDH**
16h	23:0	**Lower 3 Bytes of Z Bias Offset** With 32-bit floating format, since Z format transformation is between fix and floating, and HZBiasOffset description is suggested to be modified as below: If (HZWBFM == 32-bit fix) HZBiasOffset = HZBiasOffset * (2^32 – 1) / 2^32 Else if (HZWBFM == 24-bit fix) HZBiasOffset = HZBiasOffset * (2^24 – 1) / 2^24 Else if (HZWBFM == 16-bit fix) HZBiasOffset = HZBiasOffset * (2^16 – 1) / 2^16	**HZBiasOffsetL**
17h	23:16	**Bias Scale with 32-bit Floating Format**	**HZBiasScaleHZ**
	15:8	**Reserved**	
	7:0	**Highest Byte of Z Bias Offset** SEZbias = max(Zdx, Zdy) * HZBiasScale + HZBias Offset	**HZBiasOffsetH**
18h	23:0	**Z Bias Scale With 32-bit Floating Format**	**HZBiasScaleL**
19h	23:0	**Low 23 Bits of Low Boundary to Clamp the Z Bias** With format of 32-bit floating.	**HZBiasLClampL**
1Ah	23:0	**Low 23 Bits of High Boundary to Clamp the Z Bias** With format of 32-bit floating.	**HZBiasHClampL**

1Bh	23	**Enhance Z's Precision during PE Rendering** 0: Disable 1: Enable	**HZPrecisionEnhance**
	22:16	**Reserved**	
	15:8	**High 8 Bits of High Boundary to Clamp the Z Bias** With format of 32-bit floating.	**HZBiasHClampH**
	7:0	**High 8 Bits of Low Boundary to Clamp the Z Bias** With format of 32-bit floating.	**HZBiasLClampH**
1Ch	23:0	**Low 24 Bits Occlusion Count of both Z Test and Stencil Test Result**	**HZOcclusionCNTL**
1Dh	23:8	**Reserved**	
	7:0	**High 8 Bits Occlusion Count of both Z Test and Stencil Test Result**	**HZOcclusionCNTH**
1Eh	23:0	**Lower 24 Bits of Clip Plane's Far Value**	**HZClipFarL**
1Fh	23:0	**Lower 24 Bits of Clip Plane's Near Value**	**HZClipNearL**
20h	23:16	**Reserved**	
	15:8	**Higher 8 Bits of Clip Plane's Far Value** With format of 32-bit floating.	**HZClipFarH**

	7:0	**Higher 8Bits of Clip Plane's Near Value**
		With format of 32-bit floating.
		Check each pixel's Z value before depth testing, and remove this pixel if it's out of the range.
		If (Z > HZClipFar | Z < HZClipNear)
		Drop this pixel
		Else
		Do Depth Testing
		Note that this check is in higher priority than "Z Window".
			HZClipNearH

21h	23:0	**Low 24 Bits of Video Memory Address to Address to write the "HZOcclusionCNT"** In a unit of 4 bytes. Note for Driver: Whenever a non-zero value is set to this register , the "HZOcclusionCNT" would be written back to the video memory with the address of "HZOcclusionAdr". Since the address is separated into 2 sub-addresses, please set driver "HZOcclusionAdrL" and then "HZOcclusionAdrH" in order. Whenever decoding the "HZOcclusionAdrH", HW would check if the address (HZOcclusionAdrH and HZOcclusionAdrL cascaded) is zero or not	**HZOcclusionAdrL**
22h	23:22	**Location Setting of Z Occlusion Counter (HZOcclusionCNT)** 00: Syntem Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HZOcclusionLoc**
	21:8	**Reserved**	
	7:0	**High 8 Bits of Video Memory Address to Address** For writing the "HZOcclusionCNT", in the unit of 4 bytes.	**HZOcclusionAdrH**

HParaType = 01h

Sub-Address 23h-26h: Stencil Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
23h	23:16	**Stencil Test Reference Value for Clock-Wise Face** A positive 8-bit fix point number with range from 0 to 255. We will use this value for Stencil Test: A comparison between Stencil and HSTCWREF.	**HSTCWREF**
	15:8	**Stencil Test Operation Mask for Clock-Wise Face** The usage is a comparison between (Stencil & HSTCWOPMSK) and (HSTCWREF & HSTCWBMSK)	**HSTCWOPMSK**
	7:0	**Stencil Buffer Bit Mask for Clock-Wise Face** If a bit = 0, the relative bit in the stencil buffer cannot be changed. Otherwise, it can be changed.	**HSTCWBMSK**
24h	23:19	**Reserved**	
	18:16	**Stencil Test Mode for Clock-Wise Face** 000: Stencil Test Never Pass 001: Stencil Test Pass if (HSTCWREF & HSTCWOPMSK) < (Stencil & HSTCWOPMSK) 010: Stencil Test Pass if (HSTCWREF & HSTCWOPMSK) = (Stencil & HSTCWOPMSK) 011: Stencil Test Pass if (HSTCWREF & HSTCWOPMSK) ≤ (Stencil & HSTCWOPMSK) 100: Stencil Test Pass if (HSTCWREF & HSTCWOPMSK) > (Stencil & HSTCWOPMSK) 101: Stencil Test Pass if (HSTCWREF & HSTCWOPMSK) ≠ (Stencil & HSTCWOPMSK) 110: Stencil Test Pass if (HSTCWREF & HSTCWOPMSK) ≥ (Stencil & HSTCWOPMSK) 111: Stencil Test Always Pass	**HSTCWMD**
	15:9	**Reserved**	

	8:6	**Stencil Operation for Stencil Test Fail for Clock-Wise Face** 000: KEEP 001: ZERO 010: REPLACE 011: INCRSAT 100: DECRSAT 101: INVERT 110: INCR 111: DECR	**HSTCWOPSF**
	5:3	**Stencil Operation for Stencil Test Pass and Z Test Fail for Clock-Wise Face** 000: KEEP 001: ZERO 010: REPLACE 011: INCRSAT 100: DECRSAT 101: INVERT 110: INCR 111: DECR	**HSTCWOPSPZF**
	2:0	**Stencil Operation for Stencil Test Pass and Z Test Pass for Clock-Wise Face** 000: KEEP 001: ZERO 010: REPLACE 011: INCRSAT 100: DECRSAT 101: INVERT 110: INCR 111: DECR	**HSTCWOPSPZP**
25h	23:16	**Stencil Test Reference Value for Counter-Clock-Wise Face** A positive 8-bit fix point number with range from 0 to 255. We will use this value for Stencil Test: A comparison between Stencil and HSTCCWREF.	**HSTCCWREF**
	15:8	**Stencil Test Operation Mask for Counter-Clock-Wise Face** The usage is a comparison between (Stencil & HSTCCWOPMSK) and (HSTCCWREF & HSTCCWBMSK)	**HSTCCWOPMSK**
	7:0	**Stencil Buffer Bit Mask for Counter-Clock-Wise Face** If a bit = 0, the relative bit in the stencil buffer cannot be changed. Otherwise, it can be changed.	**HSTCCWBMSK**
26h	23:19	**Reserved**	
	18:16	**Stencil Test Mode for Counter-Clock-Wise Face** 000: Stencil Test Never Pass 001: Stencil Test Pass if (HSTCCWREF & HSTCCWOPMSK) < (Stencil & HSTCCWOPMSK) 010: Stencil Test Pass if (HSTCCWREF & HSTCCWOPMSK) = (Stencil & HSTCCWOPMSK) 011: Stencil Test Pass if (HSTCCWREF & HSTCCWOPMSK) ≤ (Stencil & HSTCCWOPMSK) 100: Stencil Test Pass if (HSTCCWREF & HSTCCWOPMSK) > (Stencil & HSTCCWOPMSK) 101: Stencil Test Pass if (HSTCCWREF & HSTCCWOPMSK) ≠ (Stencil & HSTCCWOPMSK) 110: Stencil Test Pass if (HSTCCWREF & HSTCCWOPMSK) ≥ (Stencil & HSTCCWOPMSK) 111: Stencil Test Always Pass	**HSTCCWMD**
	15:9	**Reserved**	
	8:6	**Stencil Operation for Stencil Test Fail for Counter-Clock-Wise Face** 000: KEEP 001: ZERO 010: REPLACE 011: INCRSAT 100: DECRSAT 101: INVERT 110: INCR 111: DECR	**HSTCCWOPSF**
	5:3	**Stencil Operation for Stencil Test Pass and Z Test Fail for Counter-Clock-Wise Face** 000: KEEP 001: ZERO 010: REPLACE 011: INCRSAT 100: DECRSAT 101: INVERT 110: INCR 111: DECR	**HSTCCWOPSPZF**

	2:0	**Stencil Operation for Stencil Test Pass and Z Test Pass for Counter-Clock-Wise Face** 000: KEEP 001: ZERO 010: REPLACE 011: INCRSAT 100: DECRSAT 101: INVERT 110: INCR 111: DECR	**HSTCCWOPSPZP**

HParaType = 01h

Sub-Address 29h-2Ah: Setting for Coarse Z Test Function

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
29h	23	**Force CZ Retest if Original is "Reject" and the Related Primitive is Clock-Wise** Force the CZ result as "ReTEST" if original is "REJECT" and the related primitive is Clock-Wise. 0: Normal CZ test 1: Never reject	**HCZMskREJECT_CW**
	22	**Force CZ Retest if Original is "Pass" and the Related Primitive is Clock-Wise** Force the CZ result as "ReTEST" if original is "Pass" and the related primitive is Clock-Wise. 0: Normal CZ test 1: Never pass	**HCZMskPASS_CW**
	21:20	**Coarse Z Buffer Location Setting** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HCZLoc**
	19:18	**Reserved**	
	17	**Inverse Mode for CZ Test** 0: Smaller Z value as nearer and larger Z value as farer, for HZWTMD is LESS or LESSEQUAL 1: Larger Z value as nearer and smaller Z value as farer, for HZWTMD is GREATER or GREATEREQUAL	**HCZTInvMD**
	16	**Coarse Z Write Back Mode** 0: Whenever the CZTAG_CNT is "0" and CZTAG_UPDATED, then write back the CZ in cache back to video memory 1: Whenever the cell is selected for new CZ value and CZTAG_UPDATED, then write back the old CZ in cache back to video memory, and then read the new CZ value	**HCZWBMD**
	15	**Reset of CZ Cache's TAG** Instead of SW to clear this setting, it would be auto-cleared by HW itself. 0: Normal 1: Rest	**HCZRST**
	14	**Reserved**	
	13:12	**Coarse Z Test by Using Conservative Mode** 00: Use optimized merging rule (HCZTMDRrAa) 01: Use Conservative Merging Rule and only update both "MIN" value and "MAX" value 10: Use Conservative Merging Rule and only update "MIN" value 11: Reserved	**HCZTCsrvR**
	11	**Force CZ Retest if Original is "Reject" and the Related Primitive is Counter-Clock-Wise** Force the CZ result as "ReTEST" if original is "REJECT" and the related primitive is Counter-Clock-Wise. 0: Normal CZ test 1: Never reject	**HCZMskREJECT_CCW**
	10	**Force CZ Retest if Original is "Pass" and the Related Primitive is Counter-Clock-Wise** Force the CZ result as "ReTEST" if original is "PASS" and the related primitive is Counter-Clock-Wise. 0: Normal CZ test 1: Never reject	**HCZMskPASS_CCW**
	9	**Force ReTest as the result of HZ Test** 0: Normal 1: Force	**HCZForceRT**
	8:0	**Coarse Z Buffer's Pitch** In unit of 32 bytes.	**HCZPit**
2Ah	23:0	**Coarse Z Buffer's Base Address** In unit of 256 bytes. E3R(W)CZADR = HCZBas*256 + Y*HCZPit*32 + (2*X)*16	**HCZBas**

HParaType = 01h

Sub-Address 33h-4Fh: Alpha Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
33h	23:15	**Reserved**	
	14:12	**Alpha Test Mode** 000: Alpha Test Never Pass 001: Alpha Test Pass if Anew < HATREF 010: Alpha Test Pass if Anew = HATREF 011:A lpha Test Pass if Anew ≤ HATREF 100: Alpha Test Pass if Anew > HATREF 101: Alpha Test Pass if Anew ≠ HATREF 110: Alpha Test Pass if Anew ≥ HATREF 111: Alpha Test Always Pass	**HATMD**
	11	**Reserved**	
	10:0	**Alpha Test Reference Value** Positive fix-point from 0.0 to 1.0.	**HATREF**
34h		**Alpha Blending Equation of RGB: Equation of RGB:** Equation of RGB: Cout = ((AB_FCa * AB_Ca) AB_Cop (AB_FCb * AB_Cb)) If (HABLCsat = false) Clamp Cout to 1.0 to 0.0	
	23:17	**Reserved**	
	16	**RGB Saturation Control of Alpha Blending Calculation** 0: Cout will be clamp to 0.0 ~ 1.0 1: Cout will not be clamp to 0.0 ~ 1.0	**HABLCsat**
	15:10	**Ca of Alpha Blending Equation** **HABLCa[4:3]** / **R of AB_Ca** / **G of AB_Ca** / **B of AB_Ca** **00** Reserved Reserved Reserved **01** Reserved Reserved Reserved **10** Reserved Reserved Reserved **11** Reserved Reserved Reserved **HABLCa[3:0]** / **R_of_OPCa** / **G_of_OPCa** / **B_of_OPCa** **0001** Rsrc Gsrc Bsrc **0001** Rdst Gdst Bdst **0010** Reserved Reserved Reserved **0011** Reserved Reserved Reserved **0100** Reserved Reserved Reserved **0101** R of HABLRca G of HABLRCa B of HABLRCa **0110** min (Rsrc, Rdst) min (Gsrc, Gdst) min (Bsrc, Bdst) **0111** max (Rsrc, Rdst) max (Gsrc, Gdst) max (Bsrc, Bdst) **1xxx** Reserved Reserved Reserved	**HABLCa**
	9:4	**FCa of Alpha Blending Equation** **HABLFCa[5:4]** / **R of AB_Fca** / **G of AB_FCa** / **B of AB_FCa** **00** R_of_OPFCa G_of_OPFCa B_of_OPFCa **01** 1.0 - R_of_OPFCa 1.0 - G_of_OPFCa 1.0 - B_of_OPFCa **10** Reserved Reserved Reserved **11** Reserved Reserved Reserved **HABLFCa[3:0]** / **R_of_OPFCa** / **G_of_OPFCa** / **B_of_OPFCa** **0000** Rsrc Gsrc Bsrc **0001** Rdst Gdst Bdst **0010** Asrc Asrc Asrc **0011** Adst Adst Adst **0100** Reserved Reserved Reserved **0101** R of HABLRFCa G of HABLRFCa B of HABLRFCa **0110** Reserved Reserved Reserved **0111** Reserved Reserved Reserved **1000** min (Asrc, 1-Adst) min (Asrc, 1-Adst) min (Asrc, 1-Adst) **1111-1001** Reserved Reserved Reserved	**HABLFCa**
	3:0	**Reserved**	
35h	23:16	**Reserved**	
	15:14	**Cop of Alpha Blending Equation** HABLCop / Cop 00 + 01 - 10 Max 11 Min	**HABLCop**

	13:8	**Cb of Alpha Blending Equation** **HABLCb[5:4] / R of AB_Cb / G of AB_Cb / B of AB_Cb** **00** Reserved / Reserved / Reserved **01** Reserved / Reserved / Reserved **10** Reserved / Reserved / Reserved **11** Reserved / Reserved / Reserved **HABLCb[3:0] / R_of_OPCb / G_of_OPCb / B_of_OPCb** **0000** Rsrc / Gsrc / Bsrc **0001** Rdst / Gdst / Bdst **0010** Reserved / Reserved / Reserved **0011** Reserved / Reserved / Reserved **0100** Reserved / Reserved / Reserved **0101** R of HABLRCb / G of HABLRCb / B of HABLRCb **0110** Reserved / Reserved / Reserved **0111** Reserved / Reserved / Reserved **1xxx** Reserved / Reserved / Reserved	**HABLCb**
	7:2	**FCb of Alpha Blending Equation** **HAB_FCb[5:4] / R of AB_FCb / G of AB_FCb / B of AB_FCb** **00** R_of_OPFCb / G_of_OPFCb / B_of_OPFCb **01** 1.0 - (R_of_OPFCb) / 1.0 - (G_of_OPFCb) / 1.0 - (B_of_OPFCb) **10** Reserved / Reserved / Reserved **11** Reserved / Reserved / Reserved **HAB_FCb[3:0] / R_of_OPFCb / G_of_OPFCb / B_of_OPFCb** **0000** Rsrc / Gsrc / Bsrc **0001** Rdst / Gdst / Bdst **0010** Asrc / Asrc / Asrc **0011** Adst / Adst / Adst **0100** Reserved / Reserved / Reserved **0101** R of HABLRFCb / G of HABLRFCb / B of HABLRFCb **0110** Reserved / Reserved / Reserved **0111** Reserved / Reserved / Reserved **1000** min (Asrc, 1-Adst) / min (Asrc, 1-Adst) / min (Asrc, 1-Adst) **1111-1001** Reserved / Reserved / Reserved	**HABLFCb**
	1:0	**Reserved**	
36h		**Equation of A:** Aout = ((AB_FAa * AB_Aa) AB_Aop (AB_FAb * AB_Ab)) If (HABLAsat = false) Clamp Aout to 1.0 to 0.0	
	23:17	**Reserved**	
	16	**Alpha Saturation Control of Alpha Blending Calculation** 0: Aout will be clamped to 0.0 ~ 1.0 1: Aout will not be clamped to 0.0 ~ 1.0	**HABLAsat**
	15:10	**Aa of Alpha Blending Equation** **HABLAa[5:4] / AB_Aa** **00** Reserved **01** Reserved **10** Reserved **11** Reserved **HABLAa[3:0] / OPAa** **0000** 0 **0001** Asrc **0010** Adst **0011** Reserved **0100** Reserved **0101** min (Asrc, Adst) **0110** Reserved **0111** max (Asrc, Adst) **1000** Reserved **1001** HABLRAa **1111-1010** Reserved	**HABLAa**

	9:4	**FAa of Alpha Blending Equation** **HABLFAa[5:4] AB_FAa** **00** OPFAa **01** 1 - OPFAa **10** Reserved **11** Reserved **HABLFAa[3:0] OPFAa** **0000** 0 **0001** Asrc **0010** Adst **0011** Reserved **0100** Reserved **0101** Reserved **0110** Reserved **0111** Reserved **1000** min (Asrc, 1-Adst) **1001** HABLRFAa **1111-1101** Reserved	**HABLFAa**
	3:0	**Reserved**	
37h	23:16	**Reserved**	
	15:14	**Aop of Alpha Blending Equation** **HAB_Aop AB_Aop** 00 + 01 - 10 Max 11 Min	**HABLAop**
	13:8	**Ab of Alpha Blending Equation** **HAB_Ab[5:4] AB_Ab** **00** Reserved **01** Reserved **10** Reserved **11** Reserved **HABLAb[3:0] OPAb** **0000** 0 **0001** Asrc **0010** Adst **0011** Reserved **0100** Reserved **0101** Reserved **0110** Reserved **0111** Reserved **1000** Reserved **1001** HABLRAb **1111-1001** Reserved	**HABLAb**
	7:2	**FAb of Alpha Blending Equation** **HABLFAb[5:4] AB_FAb** **00** OPFAb **01** 1 – OPFAb **10** Reserved **11** Reserved **HABLFAb[3:0] OPFAb** **0000** 0 **0001** Asrc **0010** Adst **0011** Reserved **0100** Reserved **0101** Reserved **0110** Reserved **0111** Reserved **1000** min (Asrc, 1-Adst) **1001** HABLRFAb **1111-1010** Reserved	**HABLFAb**
	1:0	**Reserved**	
38h	23	**Reserved**	
	22:12	**G of HABLRCa** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
	11	**Reserved**	
	10:0	**B of HABLRCa** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
39h	23	**Reserved**	

	22:12	**R of HABLRFCa** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
	11	**Reserved**	
	10:0	**R of HABLRCa** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
3Ah	23	**Reserved**	
	22:12	**G of HABLRFCa** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
	11	**Reserved**	
	10:0	**B of HABLRFCa** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
3Bh	23	**Reserved**	
	22:12	**G of HABLRCb** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
	11	**Reserved**	
	10:0	**B of HABLRCb** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
3Ch	23	**Reserved**	
	22:12	**R of HABLRFCb** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
	11	**Reserved**	
	10:0	**R of HABLRCb** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
3Dh	23	**Reserved**	
	22:12	**G of HABLRFCb** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
	11	**Reserved**	
	10:0	**B of HABLRFCb** This is a 11-bit positive fix-point number from 0.0 to 1.0.	
3Eh	23	**Reserved**	
	22:12	**Constant Register of Aa** This is a 11-bit positive fix-point number from 0.0 to 1.0.	**HABLRAa**
	11	**Reserved**	
	10:0	**Constant Register of FAa** This is a 11-bit positive fix-point number from 0.0 to 1.0.	**HABLRFAa**
3Fh	23	**Reserved**	
	22:12	**Constant Register of Ab** This is a 11-bit positive fix-point number from 0.0 to 1.0.	**HABLRAb**
	11	**Reserved**	
	10:0	**Constant Register of FAb** This is a 11-bit positive fix-point number from 0.0 to 1.0.	**HABLRFAb**
40-4Fh	23:0	**Reserved**	

HParaType = 01h

Sub-Address 50h-57h: Destination Setting – Render Target 0

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
50h	23:0	**Render Target0's Base Address** In unit of 256 bytes.	**HMRT0Bas**
51h	23	**Memory Mode of Render Target 0** 0: Linear mode 1: Tile Mode	**HMRT0MMode**
	22	**Render Target0's Tile is 16-texel high** 0: Normal 8-pixel high 1: 16-pixel high	**HMRT0TileH16**
	21:20	**Location Setting of Render Target 0** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HMRT0Loc**
	19:0	**Reserved**	

52h	23:16	**Render Target0's Format** **Bit [23:19]** 00000: Reserved 00001: Reserved 00010: Luminance Format 00011: Reserved 00100: Reserved 00101: Reserved 00110: YUV(Video Texture) Format 00111-10000: Reserved 10001: ARGB_16bpp Format 10010: Reserved 10011: ARGB_32bpp Format 10100: Reserved 10101: ABGR_16bpp Format 10110: Reserved 10111: ABGR_32bpp Format 11000: Reserved 11001: RGBA_16bpp Format 11010: Reserved 11011: RGBA_32bpp Format 11100: BGRA_16bpp Format 11101: BGRA_32bpp Format 11110: Floating Color Format 11111: Reserved **If there is NO Alpha channel in the defined format, fill 1.0 as Alpha value.** **Bit [18:16]** ***For Luminance Format*** 000-100: Reserved 101: AL88 (Bit[15:8] = A, Bit[7:0] = L) 110: L16 (Bit[15:0] = L) For L16 format, post-rendering must be disabled. 111: Reserved ***For YUV format (Video Texture)*** 000: Package mode (Bit[31:24] = V or Cr, Bit[23:16] = Y1, Bit[15:8] = U or Cb, Bit[7:0] = Y0) For reading color, consider the 8-bit Y as positive 8-bit G, 8-bit U as positive 8-bit R, and 8-bit V as positive 8-bit B. Then extend them to s1.10 according to setting of HRT0CExtend. For writing color, the adjacent 2 pixel are combinen into 32-bit and share same 8-bit U and V. Consider an even pixel with R0, G0 & B0(dithered 8-bit color), and the next odd pixel with R1, G1& B1(dithered 8-bit color), the packed result is {(B0 + B1)/2, G1, (R0+R1)/2, G0}. Note the (B0+B1)/2 and (R0+R1)/2 are the result of rounding. ***For ARGB_16bpp Format*** 000: RGB555 (Bit[14:10] = R, Bit[9:5] = G, Bit[4:0] = B) 001: RGB565 (Bit[15:11] = R, Bit[10:5] = G, Bit[4:0] = B) 010: ARGB1555 (Bit[15] = A, Bit[14:10] = R, Bit[9:5] = G, Bit[4:0] = B) 011: ARGB4444 (Bit[15:12] = A, Bit[11:8] = R, Bit[7:4] = G, Bit[3:0] = B) 100: Reserved 101: RGB565 for write color, Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R But for read color, Bit[15:11] = B, Bit[10:6] = G, Bit[4:0] = R 11x: Reserved ***For ARGB_32bpp Format*** 000: ARGB0888 (Bit[23:16] = R, Bit[15:8] = G, Bit[7:0] = B) 001: ARGB8888 (Bit[31:24] = A, Bit[23:16] = R, Bit[15:8] = G, Bit[7:0] = B) 010: ARGB2_10_10_10 (Bit[31:30] = A, Bit[29:20] = R, Bit[19:10] = G, Bit[9:0] = B) 011-1xx: Reserved ***For ABGR_16bpp Format*** 000: BGR555 (Bit[14:10] = B, Bit[9:5] = G, Bit[4:0] = R) 001: BGR565 (Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R) 010: ABGR1555 (Bit[15] = A, Bit[14:10] = B, Bit[9:5] = G, Bit[4:0] = R) 011: ABGR4444 (Bit[15:12] = A, Bit[11:8] = B, Bit[7:4] = G, Bit[3:0] = R) 100: Reserved 101: BGR565 for write color, Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R But for read color, Bit[15:11] = B, Bit[10:6] = G, Bit[4:0] = R 11x: Reserved ***For ABGR_32bpp Format*** 000: ABGR0888 (Bit[23:16] = B, Bit[15:8] = G, Bit[7:0] = R) 001: ABGR8888 (Bit[31:24] = A, Bit[23:16] = B, Bit[15:8] = G, Bit[7:0] = R) 010: ABGR2_10_10_10 (Bit[31:30] = A, Bit[29:20] = B, Bit[19:10] = G, Bit[9:0] = R) 011: G16R16 (Bit[31:16] = G, Bit[15:0] = R) For G16R16 format, post-rendering must be disabled. 1xx: Reserved ***For RGBA_16bpp Format***	**HMRT0FM**

	15	**Color Extending Mode (Excluding Alpha)** 0: Extending with high color bit Translate to 1.10 format **10 => 11** $C_{11} = C_{10} * (1024/1023) = C_{10} + (1/1023) * C_{10}$ $C_{10} < 1023$, $C_{11} = 0.xxxxxxxxxx(C_{10})$ $C_{10} == 1023$, $C_{11} = 1.0000000000$ **8 => 11** $C_{11} = C_8 * (1024/255) = 4*C_8 + (4/255)*C_8$ $C_8 < 64$, $C_{11} = 0.C_800$ $64 \leqq C_8 < 128$, $C_{11} = 0.C_801$ $128 \leqq C_8 < 192$, $C_{11} = 0.C_810$ $192 \leqq C_8 < 255$, $C_{11} = 0.C_811$ $C_8 == 255$, $C_{11} = 1.0000000000$ **6 => 11** $C_{11} = C_6 * (1024/63) = 16*C_6 + (16/63)*C_6$ $C_6 < 4$, $C_{11} = 0.C_60000$ $4 \leqq C_6 < 8$, $C_{11} = 0.C_60001$ $8 \leqq C_6 < 12$, $C_{11} = 0.C_60010$ $12 \leqq C_6 < 16$, $C_{11} = 0.C_60011$ $16 \leqq C_6 < 20$, $C_{11} = 0.C_60100$ $20 \leqq C_6 < 24$, $C_{11} = 0.C_60101$ $24 \leqq C_6 < 28$, $C_{11} = 0.C_60110$ $28 \leqq C_6 < 32$, $C_{11} = 0.C_60111$ $32 \leqq C_6 < 36$, $C_{11} = 0.C_61000$ $36 \leqq C_6 < 40$, $C_{11} = 0.C_61001$ $40 \leqq C_6 < 44$, $C_{11} = 0.C_61010$ $44 \leqq C_6 < 48$, $C_{11} = 0.C_61011$ $48 \leqq C_6 < 52$, $C_{11} = 0.C_61100$ $52 \leqq C_6 < 56$, $C_{11} = 0.C_61101$ $56 \leqq C_6 < 60$, $C_{11} = 0.C_61110$ $60 \leqq C_6 < 63$, $C_{11} = 0.C_61111$ $C_6 == 63$, $C_{11} = 1.0000000000$ **5 => 11** $C_{11} = C_5 * (1024/31) = 33C_5 + (1/31)*C_5 = 32*C_5 + C_5 + (1/31)*C_5$ $C_5 < 31$, $C_{11}=0.C_5C_5$ $C_5 == 31$, $C_{11}=1.0000000000$ **4=>11** $C_{11} = C_4 * (1024/15) = 68*C_4 + (4/15)*C_4 = 64*C_4 + 4*C_4 + (4/15)*C_4$ $C_4 < 4$, $C_{11}=C_4C_400$ $4 \leqq C_4 < 8$, $C_{11}=C_4C_401$ $8 \leqq C_4 < 12$, $C_{11}=C_4C_410$ $12 \leqq C_4 < 15$, $C_{11}=C_4C_411$ $C_4 == 15$, $C_{11}=1.0000000000$ 1: Extending with zero Considering 6=>11 as example: C11 = 0. $C_6$0000	**HRT0CExtend**
	14:13	**Reserved**	
	12	**Saturate of PS's Output for Render Target "M"** 0: Clamp PS's output color oCm to related render tager format's range For format ARGB and Luminance: MINVALUE = 0.0, MAXVALUE = 1.0 For 16-bit floating color format: MINVALUE = 16'hFBFF, MAXVALUE = 16'h7BFF For 32-bit floating color format: MINVALUE = 32'h7F7FFFFF, MAXVALUE =32'hFF7FFFFF Clamped Value = min(MAXVALUE, max(oCm, MINVALUE)) Note if oCm is "NAN", clamped value is MINVALUE 1: oCm doesn't clamp	**HMRT0PSOsat**
	11:10	**Reserved**	
	9:0	**Render Target0's Pitch** In unit of 32 bytes for linear mode In unit of tile(256 bytes or 512 byte depend on HMRT0TileH16) for tile mode	**HMRT0Pit**
53h	23	**Render Target0's Y Inverse for Dither** 0: Not inverse 1: Inverse	**HMRT0DTYInverse**
	22:21	**Render Target0's Y Bias for Dither**	**HMRT0DTYBias**

	20	**Render Target0's X Inverse for Dither** 0: Not inverse 1: Inverse	**HMRT0DTXInverse**
	19:18	**Render Target0's X Bias for Dither**	**HMRT0DTXBias**
	17	**Reserved**	
	16:15	**Render Target0's Dither Mode** 00: Dither Table with multiplied by (2^n – 1) 10: Rounding with multiplied by (2^n – 1) 01: Dither Table without multiplied by (2^n – 1) 11: Rounding without multiplied by (2^n – 1)	**HMRT0DTMode**
	14:12	**Reserved**	
	11:8	**Render Target0's Raster Operation** 0000: BLACK 0 0001: NOT_MERGE_PEN DPon 0010: MASK_NOT_PEN DPna 0011: NOT_COPY_PEN Pn 0100: MASK_PEN_NOT PDna 0101: NOT Dn 0110: XOR_PEN DPx 0111: NOT_MASK_PEN DPan 1000: MASK_PEN DPa 1001: NOT_XOR_PEN DPxn 1010: NOP D 1011: MERGE_NOT_PEN DPno 1100: COPY_PEN P 1101: MERGE_PEN_NOT PDno 1110: MERGE_PEN DPo 1111: WHITE 1	**HMRT0ROP**
	7	**Reserved**	
	6	**DeGamma for Render Target0's Reading Color** 0: Disable 1: Enable. Use one-by-one mapping to deGamma the readen color from 2.2 to 1.0 color field before alpha blending operation.	**HMRT0RDG**
	5:4	**Render Target0 is SRGB** 00: Gamma 1.0 field. Disable Gamma correction of writing back color 01: Gamma correction enable. Use one-by-one mapping to transform the color to gamma 2.2 field before wroten back to color buffer 10: Gamme correction enable. Use Gamma Table "HGTWC" to transform each color channel to a specific color field before wroten back to color buffer 11: Reserved	**HMRT0SRGB**
	3	**Mask of Render Target0's Alpha Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT0AMSK**
	2	**Mask of Render Target0's Red Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT0RMSK**
	1	**Mask of Render Target0's Green Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT0GMSK**
	0	**Mask of Render Target0's Blue Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT0BMSK**
54-57h	23:0	**Reserved**	

HParaType = 01h

Sub-Address 58h-5Fh: Destination Setting – Render Target 1

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
58h	23:0	**Render Target1's Base Address** In unit of 256 bytes.	**HMRT1Bas**
59h	23	**Memory Mode of Render Target 1** 0: Linear mode 1: Tile Mode	**HMRT1MMode**
	22	**Render Target1's Tile is 16-texel high** 0: Normal 8-pixel high 1: 16-pixel high	**HMRT1TileH16**
	21:20	**Location Setting of Render Target 1** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HMRT1Loc**
	19:0	**Reserved**	

5Ah	23:16	**Render Target1's Format** **Bit [23:19]** 00000: Reserved 00001: Reserved 00010: Luminance Format 00011: Reserved 00100: Reserved 00101: Reserved 00110: YUV(Video Texture) Format 00111-10000: Reserved 10001: ARGB_16bpp Format 10010: Reserved 10011: ARGB_32bpp Format 10100: Reserved 10101: ABGR_16bpp Format 10110: Reserved 10111: ABGR_32bpp Format 11000: Reserved 11001: RGBA_16bpp Format 11010: Reserved 11011: RGBA_32bpp Format 11100: BGRA_16bpp Format 11101: BGRA_32bpp Format 11110: Floating Color Format 11111: Reserved **If there is NO Alpha channel in the defined format, fill 1.0 as Alpha value.** **Bit [18:16]** ***For Luminance Format*** 000-100: Reserved 101: AL88 (Bit[15:8] = A, Bit[7:0] = L) 110: L16 (Bit[15:0] = L) For L16 format, post-rendering must be disabled. 111: Reserved ***For YUV format (Video Texture)*** 000: Package mode (Bit[31:24] = V or Cr, Bit[23:16] = Y1, Bit[15:8] = U or Cb, Bit[7:0] = Y0) For reading color, consider the 8-bit Y as positive 8-bit G, 8-bit U as positive 8-bit R, and 8-bit V as positive 8-bit B. Then extend them to s1.10 according to setting of HRT1CExtend. For writing color, the adjacent 2 pixel are combinen into 32-bit and share same 8-bit U and V. Consider an even pixel with R0, G0 & B0(dithered 8-bit color), and the next odd pixel with R1, G1& B1(dithered 8-bit color), the packed result is {(B0 + B1)/2, G1, (R0+R1)/2, G0}. Note the (B0+B1)/2 and (R0+R1)/2 are the result of rounding. ***For ARGB_16bpp Format*** 000: RGB555 (Bit[14:10] = R, Bit[9:5] = G, Bit[4:0] = B) 001: RGB565 (Bit[15:11] = R, Bit[10:5] = G, Bit[4:0] = B) 010: ARGB1555 (Bit[15] = A, Bit[14:10] = R, Bit[9:5] = G, Bit[4:0] = B) 011: ARGB4444 (Bit[15:12] = A, Bit[11:8] = R, Bit[7:4] = G, Bit[3:0] = B) 100: Reserved 101: RGB565 for write color, Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R But for read color, Bit[15:11] = B, Bit[10:6] = G, Bit[4:0] = R 11x: Reserved ***For ARGB_32bpp Format*** 000: ARGB0888 (Bit[23:16] = R, Bit[15:8] = G, Bit[7:0] = B) 001: ARGB8888 (Bit[31:24] = A, Bit[23:16] = R, Bit[15:8] = G, Bit[7:0] = B) 010: ARGB2_10_10_10 (Bit[31:30] = A, Bit[29:20] = R, Bit[19:10] = G, Bit[9:0] = B) 011-1xx: Reserved ***For ABGR_16bpp Format*** 000: BGR555 (Bit[14:10] = B, Bit[9:5] = G, Bit[4:0] = R) 001: BGR565 (Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R) 010: ABGR1555 (Bit[15] = A, Bit[14:10] = B, Bit[9:5] = G, Bit[4:0] = R) 011: ABGR4444 (Bit[15:12] = A, Bit[11:8] = B, Bit[7:4] = G, Bit[3:0] = R) 100: Reserved 101: BGR565 for write color, Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R But for read color, Bit[15:11] = B, Bit[10:6] = G, Bit[4:0] = R 11x: Reserved ***For ABGR_32bpp Format*** 000: ABGR0888 (Bit[23:16] = B, Bit[15:8] = G, Bit[7:0] = R) 001: ABGR8888 (Bit[31:24] = A, Bit[23:16] = B, Bit[15:8] = G, Bit[7:0] = R) 010: ABGR2_10_10_10 (Bit[31:30] = A, Bit[29:20] = B, Bit[19:10] = G, Bit[9:0] = R) 011: G16R16 (Bit[31:16] = G, Bit[15:0] = R) For G16R16 format, post-rendering must be disabled. 1xx: Reserved ***For RGBA_16bpp Format***	**HMRT1FM**

	15	**Color Extending Mode (Excluding Alpha)** 0: Extending with high color bit Translate to 1.10 format **10 => 11** $C_{11} = C_{10} * (1024/1023) = C_{10} + (1/1023) * C_{10}$ $C_{10} < 1023$, $C_{11} = 0.xxxxxxxxxx(C_{10})$ $C_{10} == 1023$, $C_{11} = 1.0000000000$ **8 => 11** $C_{11} = C_8 * (1024/255) = 4*C_8 + (4/255)*C_8$ $C_8 < 64$, $C_{11} = 0.C_800$ $64 \leqq C_8 < 128$, $C_{11} = 0.C_801$ $128 \leqq C_8 < 192$, $C_{11} = 0.C_810$ $192 \leqq C_8 < 255$, $C_{11} = 0.C_811$ $C_8 == 255$, $C_{11} = 1.0000000000$ **6 => 11** $C_{11} = C_6 * (1024/63) = 16*C_6 + (16/63)*C_6$ $C_6 < 4$, $C_{11} = 0.C_60000$ $4 \leqq C_6 < 8$, $C_{11} = 0.C_60001$ $8 \leqq C_6 < 12$, $C_{11} = 0.C_60010$ $12 \leqq C_6 < 16$, $C_{11} = 0.C_60011$ $16 \leqq C_6 < 20$, $C_{11} = 0.C_60100$ $20 \leqq C_6 < 24$, $C_{11} = 0.C_60101$ $24 \leqq C_6 < 28$, $C_{11} = 0.C_60110$ $28 \leqq C_6 < 32$, $C_{11} = 0.C_60111$ $32 \leqq C_6 < 36$, $C_{11} = 0.C_61000$ $36 \leqq C_6 < 40$, $C_{11} = 0.C_61001$ $40 \leqq C_6 < 44$, $C_{11} = 0.C_61010$ $44 \leqq C_6 < 48$, $C_{11} = 0.C_61011$ $48 \leqq C_6 < 52$, $C_{11} = 0.C_61100$ $52 \leqq C_6 < 56$, $C_{11} = 0.C_61101$ $56 \leqq C_6 < 60$, $C_{11} = 0.C_61110$ $60 \leqq C_6 < 63$, $C_{11} = 0.C_61111$ $C_6 == 63$, $C_{11} = 1.0000000000$ **5 => 11** $C_{11} = C_5 * (1024/31) = 33C_5 + (1/31)*C_5 = 32*C_5 + C_5 + (1/31)*C_5$ $C_5 < 31$, $C_{11}=0.C_5C_5$ $C_5 == 31$, $C_{11}=1.0000000000$ **4=>11** $C_{11} = C_4 * (1024/15) = 68*C_4 + (4/15)*C_4 = 64*C_4 + 4*C_4 + (4/15)*C_4$ $C_4 < 4$, $C_{11}=C_4C_400$ $4 \leqq C_4 < 8$, $C_{11}=C_4C_401$ $8 \leqq C_4 < 12$, $C_{11}=C_4C_410$ $12 \leqq C_4 < 15$, $C_{11}=C_4C_411$ $C_4 == 15$, $C_{11}=1.0000000000$ 1: Extending with zero Considering 6=>11 as example: C11 = 0. $C_6$0000	**HRT1CExtend**
	14:13	**Reserved**	
	12	**Saturate of PS's Output for Render Target "M"** 0: Clamp PS's output color oCm to related render tager format's range For format ARGB and Luminance: MINVALUE = 0.0, MAXVALUE = 1.0 For 16-bit floating color format: MINVALUE = 16'hFBFF, MAXVALUE = 16'h7BFF For 32-bit floating color format: MINVALUE = 32'h7F7FFFFF, MAXVALUE =32'hFF7FFFFF Clamped Value = min(MAXVALUE, max(oCm, MINVALUE)) Note if oCm is "NAN", clamped value is MINVALUE 1: oCm doesn't clamp	**HMRT1PSOsat**
	11:10	**Reserved**	
	9:0	**Render Target1's Pitch** In unit of 32 bytes for linear mode In unit of tile(256 bytes or 512 byte depend on HMRT1TileH16) for tile mode	**HMRT1Pit**
5Bh	23	**Render Target1's Y Inverse for Dither** 0: Not inverse 1: Inverse	**HMRT1DTYInverse**
	22:21	**Render Target1's Y Bias for Dither**	**HMRT1DTYBias**

	20	**Render Target1's X Inverse for Dither** 0: Not inverse 1: Inverse	**HMRT1DTXInverse**
	19:18	**Render Target1's X Bias for Dither**	**HMRT1DTXBias**
	17	**Reserved**	
	16:15	**Render Target1's Dither Mode** 00: Dither Table with multiplied by (2^n – 1) 10: Rounding with multiplied by (2^n – 1) 01: Dither Table without multiplied by (2^n – 1) 11: Rounding without multiplied by (2^n – 1)	**HMRT1DTMode**
	14:12	**Reserved**	
	11:8	**Render Target1's Raster Operation** 0000: BLACK 0 0001: NOT_MERGE_PEN DPon 0010: MASK_NOT_PEN DPna 0011: NOT_COPY_PEN Pn 0100: MASK_PEN_NOT PDna 0101: NOT Dn 0110: XOR_PEN DPx 0111: NOT_MASK_PEN DPan 1000: MASK_PEN DPa 1001: NOT_XOR_PEN DPxn 1010: NOP D 1011: MERGE_NOT_PEN DPno 1100: COPY_PEN P 1101: MERGE_PEN_NOT PDno 1110: MERGE_PEN DPo 1111: WHITE 1	**HMRT1ROP**
	7	**Reserved**	
	6	**DeGamma for Render Target1's Reading Color** 0: Disable 1: Enable. Use one-by-one mapping to deGamma the readen color from 2.2 to 1.0 color field before alpha blending operation.	**HMRT1RDG**
	5:4	**Render Target1 is SRGB** 00: Gamma 1.0 field. Disable Gamma correction of writing back color 01: Gamma correction enable. Use one-by-one mapping to transform the color to gamma 2.2 field before wroten back to color buffer 10: Gamme correction enable. Use Gamma Table "HGTWC" to transform each color channel to a specific color field before wroten back to color buffer 11: Reserved	**HMRT1SRGB**
	3	**Mask of Render Target1's Alpha Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT1AMSK**
	2	**Mask of Render Target1's Red Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT1RMSK**
	1	**Mask of Render Target1's Green Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT1GMSK**
	0	**Mask of Render Target1's Blue Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT1BMSK**
5C-5Fh	23:0	**Reserved**	

HParaType = 01h

Sub-Address 60h-67h: Destination Setting – Render Target 2

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
60h	23:0	**Render Target2's Base Address** In unit of 256 bytes.	**HMRT2BasL**
61h	23	**Memory Mode of Render Target 2** 0: Linear mode 1: Tile Mode	**HMRT2MMode**
	22	**Render Target2's Tile is 16-texel high** 0: Normal 8-pixel high 1: 16-pixel high	**HMRT2TileH16**
	21:20	**Location Setting of Render Target 2** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HMRT2Loc**
	19:0	**Reserved**	

62h	23:16	**Render Target2's Format** **Bit [23:19]** 00000: Reserved 00001: Reserved 00010: Luminance Format 00011: Reserved 00100: Reserved 00101: Reserved 00110: YUV(Video Texture) Format 00111-10000: Reserved 10001: ARGB_16bpp Format 10010: Reserved 10011: ARGB_32bpp Format 10100: Reserved 10101: ABGR_16bpp Format 10110: Reserved 10111: ABGR_32bpp Format 11000: Reserved 11001: RGBA_16bpp Format 11010: Reserved 11011: RGBA_32bpp Format 11100: BGRA_16bpp Format 11101: BGRA_32bpp Format 11110: Floating Color Format 11111: Reserved **If there is NO Alpha channel in the defined format, fill 1.0 as Alpha value.** **Bit [18:16]** ***For Luminance Format*** 000-100: Reserved 101: AL88 (Bit[15:8] = A, Bit[7:0] = L) 110: L16 (Bit[15:0] = L) For L16 format, post-rendering must be disabled. 111: Reserved ***For YUV format (Video Texture)*** 000: Package mode (Bit[31:24] = V or Cr, Bit[23:16] = Y1, Bit[15:8] = U or Cb, Bit[7:0] = Y0) For reading color, consider the 8-bit Y as positive 8-bit G, 8-bit U as positive 8-bit R, and 8-bit V as positive 8-bit B. Then extend them to s1.10 according to setting of HRT2CExtend. For writing color, the adjacent 2 pixel are combinen into 32-bit and share same 8-bit U and V. Consider an even pixel with R0, G0 & B0(dithered 8-bit color), and the next odd pixel with R1, G1& B1(dithered 8-bit color), the packed result is {(B0 + B1)/2, G1, (R0+R1)/2, G0}. Note the (B0+B1)/2 and (R0+R1)/2 are the result of rounding. Others: Reserved ***For ARGB_16bpp Format*** 000: RGB555 (Bit[14:10] = R, Bit[9:5] = G, Bit[4:0] = B) 001: RGB565 (Bit[15:11] = R, Bit[10:5] = G, Bit[4:0] = B) 010: ARGB1555 (Bit[15] = A, Bit[14:10] = R, Bit[9:5] = G, Bit[4:0] = B) 011: ARGB4444 (Bit[15:12] = A, Bit[11:8] = R, Bit[7:4] = G, Bit[3:0] = B) 100: Reserved 101: RGB565 for write color, Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R But for read color, Bit[15:11] = B, Bit[10:6] = G, Bit[4:0] = R 11x: Reserved ***For ARGB_32bpp Format*** 000: ARGB0888 (Bit[23:16] = R, Bit[15:8] = G, Bit[7:0] = B) 001: ARGB8888 (Bit[31:24] = A, Bit[23:16] = R, Bit[15:8] = G, Bit[7:0] = B) 010: ARGB2_10_10_10 (Bit[31:30] = A, Bit[29:20] = R, Bit[19:10] = G, Bit[9:0] = B) 011-1xx: Reserved ***For ABGR_16bpp Format*** 000: BGR555 (Bit[14:10] = B, Bit[9:5] = G, Bit[4:0] = R) 001: BGR565 (Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R) 010: ABGR1555 (Bit[15] = A, Bit[14:10] = B, Bit[9:5] = G, Bit[4:0] = R) 011: ABGR4444 (Bit[15:12] = A, Bit[11:8] = B, Bit[7:4] = G, Bit[3:0] = R) 100: Reserved 101: BGR565 for write color, Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R But for read color, Bit[15:11] = B, Bit[10:6] = G, Bit[4:0] = R 11x: Reserved ***For ABGR_32bpp Format*** 000: ABGR0888 (Bit[23:16] = B, Bit[15:8] = G, Bit[7:0] = R) 001: ABGR8888 (Bit[31:24] = A, Bit[23:16] = B, Bit[15:8] = G, Bit[7:0] = R) 010: ABGR2_10_10_10 (Bit[31:30] = A, Bit[29:20] = B, Bit[19:10] = G, Bit[9:0] = R) 011: G16R16 (Bit[31:16] = G, Bit[15:0] = R) For G16R16 format, post-rendering must be disabled. 1xx: Reserved	**HMRT2FM**

	15	**Color Extending Mode (Excluding Alpha)** 0: Extending with high color bit Translate to 1.10 format **10 => 11** $C_{11} = C_{10} * (1024/1023) = C_{10} + (1/1023) * C_{10}$ $C_{10} < 1023$, $C_{11} = 0.xxxxxxxxxx(C_{10})$ $C_{10} == 1023$, $C_{11} = 1.0000000000$ **8 => 11** $C_{11} = C_8 * (1024/255) = 4*C_8 + (4/255)*C_8$ $C_8 < 64$, $C_{11} = 0.C_800$ $64 \leqq C_8 < 128$, $C_{11} = 0.C_801$ $128 \leqq C_8 < 192$, $C_{11} = 0.C_810$ $192 \leqq C_8 < 255$, $C_{11} = 0.C_811$ $C_8 == 255$, $C_{11} = 1.0000000000$ **6 => 11** $C_{11} = C_6 * (1024/63) = 16*C_6 + (16/63)*C_6$ $C_6 < 4$, $C_{11} = 0.C_60000$ $4 \leqq C_6 < 8$, $C_{11} = 0.C_60001$ $8 \leqq C_6 < 12$, $C_{11} = 0.C_60010$ $12 \leqq C_6 < 16$, $C_{11} = 0.C_60011$ $16 \leqq C_6 < 20$, $C_{11} = 0.C_60100$ $20 \leqq C_6 < 24$, $C_{11} = 0.C_60101$ $24 \leqq C_6 < 28$, $C_{11} = 0.C_60110$ $28 \leqq C_6 < 32$, $C_{11} = 0.C_60111$ $32 \leqq C_6 < 36$, $C_{11} = 0.C_61000$ $36 \leqq C_6 < 40$, $C_{11} = 0.C_61001$ $40 \leqq C_6 < 44$, $C_{11} = 0.C_61010$ $44 \leqq C_6 < 48$, $C_{11} = 0.C_61011$ $48 \leqq C_6 < 52$, $C_{11} = 0.C_61100$ $52 \leqq C_6 < 56$, $C_{11} = 0.C_61101$ $56 \leqq C_6 < 60$, $C_{11} = 0.C_61110$ $60 \leqq C_6 < 63$, $C_{11} = 0.C_61111$ $C_6 == 63$, $C_{11} = 1.0000000000$ **5 => 11** $C_{11} = C_5 * (1024/31) = 33C_5 + (1/31)*C_5 = 32*C_5 + C_5 + (1/31)*C_5$ $C_5 < 31$, $C_{11}=0.C_5C_5$ $C_5 == 31$, $C_{11}=1.0000000000$ **4=>11** $C_{11} = C_4 * (1024/15) = 68*C_4 + (4/15)*C_4 = 64*C_4 + 4*C_4 + (4/15)*C_4$ $C_4 < 4$, $C_{11}=C_4C_400$ $4 \leqq C_4 < 8$, $C_{11}=C_4C_401$ $8 \leqq C_4 < 12$, $C_{11}=C_4C_410$ $12 \leqq C_4 < 15$, $C_{11}=C_4C_411$ $C_4 == 15$, $C_{11}=1.0000000000$ 1: Extending with zero Considering 6=>11 as example: C11 = 0. $C_6$0000	**HRT2CExtend**
	14:13	**Reserved**	
	12	**Saturate of PS's Output for Render Target "M"** 0: Clamp PS's output color oCm to related render tager format's range For format ARGB and Luminance: MINVALUE = 0.0, MAXVALUE = 1.0 For 16-bit floating color format: MINVALUE = 16'hFBFF, MAXVALUE = 16'h7BFF For 32-bit floating color format: MINVALUE = 32'h7F7FFFFF, MAXVALUE =32'hFF7FFFFF Clamped Value = min(MAXVALUE, max(oCm, MINVALUE)) Note if oCm is "NAN", clamped value is MINVALUE 1: oCm doesn't clamp	**HMRT2PSOsat**
	11:10	**Reserved**	
	9:0	**Render Target2's Pitch** In unit of 32 bytes for linear mode In unit of tile(256 bytes or 512 byte depend on HMRT2TileH16) for tile mode	**HMRT2Pit**
63h	23	**Render Target2's Y Inverse for Dither** 0: Not inverse 1: Inverse	**HMRT2DTYInverse**
	22:21	**Render Target2's Y Bias for Dither**	**HMRT2DTYBias**

	20	**Render Target2's X Inverse for Dither** 0: Not inverse 1: Inverse	**HMRT2DTXInverse**
	19:18	**Render Target2's X Bias for Dither**	**HMRT2DTXBias**
	17	**Reserved**	
	16:15	**Render Target2's Dither Mode** 00: Dither Table with multiplied by (2^n – 1) 10: Rounding with multiplied by (2^n – 1) 01: Dither Table without multiplied by (2^n – 1) 11: Rounding without multiplied by (2^n – 1)	**HMRT2DTMode**
	14:12	**Reserved**	
	11:8	**Render Target2's Raster Operation** 0000: BLACK 0 0001: NOT_MERGE_PEN DPon 0010: MASK_NOT_PEN DPna 0011: NOT_COPY_PEN Pn 0100: MASK_PEN_NOT PDna 0101: NOT Dn 0110: XOR_PEN DPx 0111: NOT_MASK_PEN DPan 1000: MASK_PEN DPa 1001: NOT_XOR_PEN DPxn 1010: NOP D 1011: MERGE_NOT_PEN DPno 1100: COPY_PEN P 1101: MERGE_PEN_NOT PDno 1110: MERGE_PEN DPo 1111: WHITE 1	**HMRT2ROP**
	7	**Reserved**	
	6	**DeGamma for Render Target2's Reading Color** 0: Disable 1: Enable. Use one-by-one mapping to deGamma the readen color from 2.2 to 1.0 color field before alpha blending operation.	**HMRT2RDG**
	5:4	**Render Target2 is SRGB** 00: Gamma 1.0 field. Disable Gamma correction of writing back color 01: Gamma correction enable. Use one-by-one mapping to transform the color to gamma 2.2 field before wroten back to color buffer 10: Gamme correction enable. Use Gamma Table "HGTWC" to transform each color channel to a specific color field before wroten back to color buffer 11: Reserved	**HMRT2SRGB**
	3	**Mask of Render Target2's Alpha Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT2AMSK**
	2	**Mask of Render Target2's Red Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT2RMSK**
	1	**Mask of Render Target2's Green Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT2GMSK**
	0	**Mask of Render Target2's Blue Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT2BMSK**
64-67h	23:0	**Reserved**	

HParaType = 01h

Sub-Address 68h-6Fh: Destination Setting – Render Target 3

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
68h	23:0	**Render Target3's Base Address** In unit of 256 bytes.	**HMRT3BasL**
69h	23	**Memory Mode of Render Target 3** 0: Linear mode 1: Tile Mode	**HMRT3MMode**
	22	**Render Target3's Tile is 16-texel high** 0: Normal 8-pixel high 1: 16-pixel high	**HMRT3TileH16**
	21:20	**Location Setting of Render Target 3** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserve	**HMRT3Loc**
	19:0	**Reserved**	

6Ah	23:16	**Render Target3's Format** **Bit [23:19]** 00000: Reserved 00001: Reserved 00010: Luminance Format 00011: Reserved 00100: Reserved 00101: Reserved 00110: YUV(Video Texture) Format 00111-10000: Reserved 10001: ARGB_16bpp Format 10010: Reserved 10011: ARGB_32bpp Format 10100: Reserved 10101: ABGR_16bpp Format 10110: Reserved 10111: ABGR_32bpp Format 11000: Reserved 11001: RGBA_16bpp Format 11010: Reserved 11011: RGBA_32bpp Format 11100: BGRA_16bpp Format 11101: BGRA_32bpp Format 11110: Floating Color Format 11111: Reserved **If there is NO Alpha channel in the defined format, fill 1.0 as Alpha value.** **Bit [18:16]** ***For Luminance Format*** 000-100: Reserved 101: AL88 (Bit[15:8] = A, Bit[7:0] = L) 110: L16 (Bit[15:0] = L) For L16 format, post-rendering must be disabled. 111: Reserved ***For YUV format (Video Texture)*** 000: Package mode (Bit[31:24] = V or Cr, Bit[23:16] = Y1, Bit[15:8] = U or Cb, Bit[7:0] = Y0) For reading color, consider the 8-bit Y as positive 8-bit G, 8-bit U as positive 8-bit R, and 8-bit V as positive 8-bit B. Then extend them to s1.10 according to setting of HRT3CExtend. For writing color, the adjacent 2 pixel are combinen into 32-bit and share same 8-bit U and V. Consider an even pixel with R0, G0 & B0(dithered 8-bit color), and the next odd pixel with R1, G1& B1(dithered 8-bit color), the packed result is {(B0 + B1)/2, G1, (R0+R1)/2, G0}. Note the (B0+B1)/2 and (R0+R1)/2 are the result of rounding. ***For ARGB_16bpp Format*** 000: RGB555 (Bit[14:10] = R, Bit[9:5] = G, Bit[4:0] = B) 001: RGB565 (Bit[15:11] = R, Bit[10:5] = G, Bit[4:0] = B) 010: ARGB1555 (Bit[15] = A, Bit[14:10] = R, Bit[9:5] = G, Bit[4:0] = B) 011: ARGB4444 (Bit[15:12] = A, Bit[11:8] = R, Bit[7:4] = G, Bit[3:0] = B) 100: Reserved 101: RGB565 for write color, Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R But for read color, Bit[15:11] = B, Bit[10:6] = G, Bit[4:0] = R 11x: Reserved ***For ARGB_32bpp Format*** 000: ARGB0888 (Bit[23:16] = R, Bit[15:8] = G, Bit[7:0] = B) 001: ARGB8888 (Bit[31:24] = A, Bit[23:16] = R, Bit[15:8] = G, Bit[7:0] = B) 010: ARGB2_10_10_10 (Bit[31:30] = A, Bit[29:20] = R, Bit[19:10] = G, Bit[9:0] = B) 011-1xx: Reserved ***For ABGR_16bpp Format*** 000: BGR555 (Bit[14:10] = B, Bit[9:5] = G, Bit[4:0] = R) 001: BGR565 (Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R) 010: ABGR1555 (Bit[15] = A, Bit[14:10] = B, Bit[9:5] = G, Bit[4:0] = R) 011: ABGR4444 (Bit[15:12] = A, Bit[11:8] = B, Bit[7:4] = G, Bit[3:0] = R) 100: Reserved 101: BGR565 for write color, Bit[15:11] = B, Bit[10:5] = G, Bit[4:0] = R But for read color, Bit[15:11] = B, Bit[10:6] = G, Bit[4:0] = R 11x: Reserved ***For ABGR_32bpp Format*** 000: ABGR0888 (Bit[23:16] = B, Bit[15:8] = G, Bit[7:0] = R) 001: ABGR8888 (Bit[31:24] = A, Bit[23:16] = B, Bit[15:8] = G, Bit[7:0] = R) 010: ABGR2_10_10_10 (Bit[31:30] = A, Bit[29:20] = B, Bit[19:10] = G, Bit[9:0] = R) 011: G16R16 (Bit[31:16] = G, Bit[15:0] = R) For G16R16 format, post-rendering must be disabled. 1xx: Reserved ***For RGBA_16bpp Format***	**HMRT3FM**

	15	**Color Extending Mode (Excluding Alpha)** 0: Extending with high color bit Translate to 1.10 format **10 => 11** $C_{11} = C_{10} * (1024/1023) = C_{10} + (1/1023) * C_{10}$ $C_{10} < 1023$, $C_{11} = 0.xxxxxxxxxx(C_{10})$ $C_{10} == 1023$, $C_{11} = 1.0000000000$ **8 => 11** $C_{11} = C_8 * (1024/255) = 4*C_8 + (4/255)*C_8$ $C_8 < 64$, $C_{11} = 0.C_800$ $64 \leqq C_8 < 128$, $C_{11} = 0.C_801$ $128 \leqq C_8 < 192$, $C_{11} = 0.C_810$ $192 \leqq C_8 < 255$, $C_{11} = 0.C_811$ $C_8 == 255$, $C_{11} = 1.0000000000$ **6 => 11** $C_{11} = C_6 * (1024/63) = 16*C_6 + (16/63)*C_6$ $C_6 < 4$, $C_{11} = 0.C_60000$ $4 \leqq C_6 < 8$, $C_{11} = 0.C_60001$ $8 \leqq C_6 < 12$, $C_{11} = 0.C_60010$ $12 \leqq C_6 < 16$, $C_{11} = 0.C_60011$ $16 \leqq C_6 < 20$, $C_{11} = 0.C_60100$ $20 \leqq C_6 < 24$, $C_{11} = 0.C_60101$ $24 \leqq C_6 < 28$, $C_{11} = 0.C_60110$ $28 \leqq C_6 < 32$, $C_{11} = 0.C_60111$ $32 \leqq C_6 < 36$, $C_{11} = 0.C_61000$ $36 \leqq C_6 < 40$, $C_{11} = 0.C_61001$ $40 \leqq C_6 < 44$, $C_{11} = 0.C_61010$ $44 \leqq C_6 < 48$, $C_{11} = 0.C_61011$ $48 \leqq C_6 < 52$, $C_{11} = 0.C_61100$ $52 \leqq C_6 < 56$, $C_{11} = 0.C_61101$ $56 \leqq C_6 < 60$, $C_{11} = 0.C_61110$ $60 \leqq C_6 < 63$, $C_{11} = 0.C_61111$ $C_6 == 63$, $C_{11} = 1.0000000000$ **5 => 11** $C_{11} = C_5 * (1024/31) = 33C_5 + (1/31)*C_5 = 32*C_5 + C_5 + (1/31)*C_5$ $C_5 < 31$, $C_{11}=0.C_5C_5$ $C_5 == 31$, $C_{11}=1.0000000000$ **4=>11** $C_{11} = C_4 * (1024/15) = 68*C_4 + (4/15)*C_4 = 64*C_4 + 4*C_4 + (4/15)*C_4$ $C_4 < 4$, $C_{11}=C_4C_400$ $4 \leqq C_4 < 8$, $C_{11}=C_4C_401$ $8 \leqq C_4 < 12$, $C_{11}=C_4C_410$ $12 \leqq C_4 < 15$, $C_{11}=C_4C_411$ $C_4 == 15$, $C_{11}=1.0000000000$ 1: Extending with zero Considering 6=>11 as example: C11 = 0. $C_6$0000	**HRT3CExtend**
	14:13	**Reserved**	
	12	**Saturate of PS's Output for Render Target "M"** 0: Clamp PS's output color oCm to related render tager format's range For format ARGB and Luminance: MINVALUE = 0.0, MAXVALUE = 1.0 For 16-bit floating color format: MINVALUE = 16'hFBFF, MAXVALUE = 16'h7BFF For 32-bit floating color format: MINVALUE = 32'h7F7FFFFF, MAXVALUE =32'hFF7FFFFF Clamped Value = min(MAXVALUE, max(oCm, MINVALUE)) Note if oCm is "NAN", clamped value is MINVALUE 1: oCm doesn't clamp	**HMRT3PSOsat**
	11:10	**Reserved**	
	9:0	**Render Target3's Pitch** In unit of 32 bytes for linear mode In unit of tile(256 bytes or 512 byte depend on HMRT3TileH16) for tile mode	**HMRT3Pit**
6Bh	23	**Render Target3's Y Inverse for Dither** 0: Not inverse 1: Inverse	**HMRT3DTYInverse**
	22:21	**Render Target3's Y Bias for Dither**	**HMRT3DTYBias**

	20	**Render Target3's X Inverse for Dither** 0: Not inverse 1: Inverse	**HMRT3DTXInverse**
	19:18	**Render Target3's X Bias for Dither**	**HMRT3DTXBias**
	17	**Reserved**	
	16:15	**Render Target3's Dither Mode** 00: Dither Table with multiplied by (2^n – 1) 10: Rounding with multiplied by (2^n – 1) 01: Dither Table without multiplied by (2^n – 1) 11: Rounding without multiplied by (2^n – 1)	**HMRT3DTMode**
	14:12	**Reserved**	
	11:8	**Render Target3's Raster Operation** 0000: BLACK 0 0001: NOT_MERGE_PEN DPon 0010: MASK_NOT_PEN DPna 0011: NOT_COPY_PEN Pn 0100: MASK_PEN_NOT PDna 0101: NOT Dn 0110: XOR_PEN DPx 0111: NOT_MASK_PEN DPan 1000: MASK_PEN DPa 1001: NOT_XOR_PEN DPxn 1010: NOP D 1011: MERGE_NOT_PEN DPno 1100: COPY_PEN P 1101: MERGE_PEN_NOT PDno 1110: MERGE_PEN DPo 1111: WHITE 1	**HMRT3ROP**
	7	**Reserved**	
	6	**DeGamma for Render Target3's Reading Color** 0: Disable 1: Enable. Use one-by-one mapping to deGamma the readen color from 2.2 to 1.0 color field before alpha blending operation.	**HMRT3RDG**
	5:4	**Render Target3 is SRGB** 00: Gamma 1.0 field. Disable Gamma correction of writing back color 01: Gamma correction enable. Use one-by-one mapping to transform the color to gamma 2.2 field before wroten back to color buffer 10: Gamme correction enable. Use Gamma Table "HGTWC" to transform each color channel to a specific color field before wroten back to color buffer 11: Reserved	**HMRT3SRGB**
	3	**Mask of Render Target3's Alpha Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT3AMSK**
	2	**Mask of Render Target3's Red Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT3RMSK**
	1	**Mask of Render Target3's Green Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT3GMSK**
	0	**Mask of Render Target3's Blue Channel** If the bit value is 0, the relative data bit will remain the same value in Frame Buffer. If the bit value is 1, the relative data bit will be updated by a new calculated number.	**HMRT3BMSK**
6C-6Fh	23:0	**Reserved**	

HParaType = 01h

Sub-Address 70h-7Fh: Fog Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
70h	23:8	**Reserved**	
	7	**Fog Factor Sourec From PS** 0: Fog factor is generated in PE by no matter vertex, linear or exponential Fog 1: Fog factor and Fog operation is dealt by the PS Instead of 8-bit fixed Fog factor, PE send "Fog coordinate" with format floating s[7].10 to PS. **Note that is this bit is "true", HenFOGRT0, HenFOGRT1, HenFOGMRT2 and HenFOGMRT3 should be disabled.**	**HFogSrcPS**
	6	**Linear Fog Calculation Factor Setting 2** 0: Use W or Z to calculate Linear Fog by setting of bit 4 1: Use attributr "Fog" as fog coordinate to calculate Linear Fog If (HFogLF2 == 1) Use Fog attribute to calculate linear fog or exponential fog(fog per pixel) Else if (HFogLF == 0) Use Z attribute to calculate linear fog or exponential fog(fog per pixel) Else Use W attribute to calculate linear fog or exponential fog(fog per pixel)	**HFogLF2**
	5	**Fog Factor from Spectra Color's Alpha** 0: Individual Fog Attribute 1: Use Spectral (Color 2) Alpha as Fog factor	**HFogSCA**
	4	**Linear Fog Calculation Factor Setting** 0: Use W to calculate Linear Fog 1: Use Z to calculate Linear Fog	**HFogLF**

	3	**Fog Equation**
		//Fog Equation 0: Cout = f * (Cin + Csepc) + (1-f) * HCFog
		//Fog Equation 1: Cout = (1-f) * (Cin + Cspec) + f * HCFog
		0: Use Fog Equation 0
		1: Use Fog Equation 1
			HFogEq

	2:0	**Fog Mode** If instead of vertex Z, Z is calculated in PS (HZSrcPS = 1), linear or non-linear fog from Z is not allowable 000: Local Fog 001: Reserved Global Fog 010: Linear Fog 011: Reserved 1xx: Non-linear Fog (Using Fog Table) 100: Exponential Fog 101: Exponential_2 Fog 11x: Reserved	**HFogMD**
71h	23	**Reserved**	
	22:12	**G of HCFogCL. Positive fix-point from 0 to 1.0**	**HCFogG**
	11	**Reserved**	
	10:0	**B of HCFogCL. Positive fix-point from 0 to 1.0**	**HCFogB**
72h	23:11	**Reserved**	
	10:0	**R of Fog Color. Positive fix-point from 0 to 1.0**	**HCFogR**
73h	23:15	**Reserved**	
	14:0	**Fog Start** Floating-point is used to calculate fog factor, the format of HFogSt is floating-point [8].7.	**HFogSt**
74h	23:0	**Reserved**	
75h	23:4	**Reserved**	
	3:0	**Mantissa part of the One Over (Fog End - Fog Start)** Note that not contain leading one.	**HFogOOdMF**
76h	23:8	**Reserved**	
	7:0	**Exponential part of the One Over (Fog End - Fog Start) and as IEEE's floating presentation.** The value of 1/(Fog End – Fog Start) is (1.HfogOOdMF[1:0] * 2^(HfogOOdEF – 127))	**HFogOOdEF**
77h	23:15	**Reserved**	
	14:0	**Lower 3 Bytes of Fog End** The format of HFogEnd is floating-point [8].7.	**HFogEnd**
78h	23:21	**Reserved**	
	20:8	**Fog Density with positive floating format [8].5.**	**HFogDenst**
	7:0	**Reserved**	

79-7Fh	23:0	**Reserved**	

HParaType = 01h

Sub-Address 80h-8Fh: Miscellaneous Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
80h	23:12	**Color Window Top Clipping Value in the range of 0 to 2048**	**HCWClipT**
	11:0	**Color Window Bottom Clipping Value in the range of 0 to 2048**	**HCWClipB**
81h	23:12	**Color Window Left Clipping Value in the range of 0 to 2048**	**HCWClipL**
	11:0	**Color Window RightClipping Value in the range of 0 to 2048**	**HCWClipR**
82h	23:12	**Scissor Window Top Clipping Value in the range of 0 to 2048**	**HScissorWClipT**
	11:0	**Scissor Window Bottom Clipping Value in the range of 0 to 2048**	**HScissorWClipB**
83h	23:12	**Scissor Window Left Clipping Value in the range of 0 to 2048**	**HScissorWClipL**
	11:0	**Scissor Window Right Clipping Value in the range of 0 to 2048**	**HScissorWClipR**
84h	23:16	**Reserved**	
	15:0	**Line Pattern** The Line Pattern bit is start from the LSB.	**HLP**
85h	23	**Line Pattern Reset** HW would reset related repeat counter automatically whenever this register is set to "1". It is NOT necessary for driver to clear this bit. HW would clear it after the counter reset.	**HLPrst**
	22:16	**Reserved**	
	15:0	**Line Pattern Repeat Factor** This number denotes how many times that a line pattern bit will be used for several line pixels.	**HLPRF**
86h	**Lower 3 Bytes of Solid Shading Color**		**HSolidCL**
	23:16	**R of Solid Shading Color**	**HsolidR**
	15:8	**G of Solid Shading Color**	**HsolidG**
	7:0	**B of Solid Shading Color**	**HsolidB**
87h	23:8	**Reserved**	
	Highest Byte of Solid Shading Color		**HSolidCH**
	7:0	**Alpha of Solid Shading Color**	**HSolidA**
88h	23:13	**Reserved**	
	12:0	**Guard Band Window Left Clipping Value** Format as s12 2'scomplement	**HClipGL**
89h	23:13	**Reserved**	
	12:0	**Guard Band Window Right Clipping Value** Format as s12 2'scomplement	**HClipGR**
8Ah	23	**Enhance TX's Precision during PE's rendering** 0: Disable 1: Enable	**HTXPrecisionEnhance**
	22:17	**Reserved**	
	16	**Zero Round Mode for the Texture Coordinate from PE to PS** 0: Round to zero 1: Round to 0	**HPETXZeroRND**
	15:4	**Bottom Y value for PS's Location Register** As 12 positive integer.	**HYB4LocationReg**

	3	**Enable HYB4LocationReg**
		0: Disable
		1: Enable
		If (HPSLocationReg == true)
		If (HenYB4LocationReg == true)
		LocationReg.Y = floating(HYB4LocationReg – PEY)
		ElseLocationReg.Y = floating(PEY)
			HenYB4LocationReg

	2	**Indicate if Color 0(diffuse color) has been PreModulated with Ws** 0: Not pre-modulated, multiply C0 and Ws in SE 1: Pre-modulated, **don't** multiply C0 and Ws in WS	**HpreMWsC0**
	1	**Indicate if Color 1 (specula color)has been PreModulated with Ws** 0: Not pre-modulated, multiply C1 and Ws in SE 1: Pre-modulated, **don't** multiply C1 and Ws in WS	**HpreMWsC1**
	0	**Indicate if Fog has been PreModulated with Ws** 0: Not pre-modulated, multiply Fog and Ws in SE 1: Pre-modulated, **don't** multiply Fog and Ws in WS	**HpreMWsFog**
8B-8Fh	23:0	**Reserved**	

HParaType = 01h

Sub-Address 90h-9Ah: Pixel Shader Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
90h	23	**Disable the funtion of detecting "ALUOut" by HW PS** 0: PS would detect the "ALUOut" automaticly 1: "ALUOut" set by SW's PS code	**HPSDctALUOut_N**
	22	**SoftWare Sets the HPSDPNTAU_N** 0: HW judge dependant TAU instruction by (HPSINAtoT[127:0] != 128'h0 | HPSFIREALU) 1: SoftWare set HPSDPNTAU_N AND hw FOLLOWED	**HPSSWsetDPNTAU**
	21	**Indicate if TAU Execute Dependant Instruction** 0: There is dependant TAU instruction 1: No dependant TAU instruction	**HPSDPNTAU_N**
	20	**Indicate if TAU Excutes Dependant TXKILL Instruction** 0: TAU would excute dependant TXKILL instruction 1: TAU would NOT excute dependant TXKILL instruction Example of "dependant TXKILL" *Example 1* txld r0, t0, s0 ← load texture for CVALID | STVALID mad t1, r0, xx, xx txkill t1 ← effect both CVALID and STVALID Note to SW Driver: Whenever there is any "dependant TXKILL", please keep this register "false". Only WITHOUT any "depandant TXKILL", set "HPSDPNTXkill_N" to "true". For "independand" PS code, "dependant TXKILL" must not exist thus HPSDPNTXkill_N" has better to be set for performance issue.	**HPSDPNTXkill_N**
	19	**Pixel Shader gets Location Register (X, Y, Z, 1/W) from PE** 0: No Location Register used in PS 1: Location Register used in PS	**HPSLocationReg**
	18	**Pixel Shader is fired by ALU instruction initially. Default as "false"** 0: Pixel Shader is fired by TAU instruction initially 1: Pixel Shader is fired by ALU instruction initially	**HPSFireALU**
	17	**The content of PS's constant registers with index 32 to 54 is the same as the index from 0 to 22** 0: Filled independent 1: Fill the same value to n and (n+32)	**HPSDbCnstR**
	16:15	**Pixel Shader Configure** 00: Normal Configure. There are 16 texture registers and 12 temporary registers used for each pixel. 01: Double Configure. There are 8 texture registers and 6 temporary registers used for each pixel. 10: Tripple Configure. There are 5 texture registers and 4 temporary registers used for each pixel. 11: 4-Time Configure. There are 4 texture registers and 3 temporary registers used for each pixel.	**HPSConFig**
	14	**Texture Register 14(t14) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPSTEtoALU**
	13	**Texture Register 13(t13) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPSTDtoALU**
	12	**Texture Register 12(t12) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPSTCtoALU**
	11	**Texture Register 11(t11) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPSTBtoALU**
	10	**Texture Register 10(t10) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPSTAtoALU**
	9	**Texture Register 9(t9) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPST9toALU**
	8	**Texture Register 8(t8) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPST8toALU**

	7	**Texture Register 7(t7) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPST7toALU**
	6	**Texture Register 6(t6) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPST6toALU**
	5	**Texture Register 5(t5) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPST5toALU**
	4	**Texture Register 4(t4) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPST4toALU**
	3	**Texture Register 3(t3) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPST3toALU**
	2	**Texture Register 2(t2) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPST2toALU**
	1	**Texture Register 1(t1) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPST1toALU**
	0	**Texture Register 0(t0) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPST0toALU**

91h	23	**Texture Register 15(t15) is as the input to ALU** 0: None 1: For ALU(arithmetic unit)	**HPSTFtoALU**
	22:16	**Reserved**	
	15	**Texture Register 15(t15) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPETXZeroRND**
	14	**Texture Register 14(t14) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPSTEtoTAU**
	13	**Texture Register 13(t13) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPSTDtoTAU**
	12	**Texture Register 12(t12) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPSTCtoTAU**
	11	**Texture Register 11(t11) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPSTBtoTAU**
	10	**Texture Register 10(t10) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPSTAtoTAU**
	9	**Texture Register 9(t9) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPST9toTAU**
	8	**Texture Register 8(t8) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPST8toTAU**
	7	**Texture Register 7(t7) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPST7toTAU**
	6	**Texture Register 6(t6) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPST6toTAU**
	5	**Texture Register 5(t5) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPST5toTAU**
	4	**Texture Register 4(t4) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPST4toTAU**
	3	**Texture Register 3(t3) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPST3toTAU**
	2	**Texture Register 2(t2) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPST2toTAU**

	1	**Texture Register 1(t1) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPST1toTAU**
	0	**Texture Register 0(t0) is as the input to TAU** 0: None 1: For TAU(texture address unit)	**HPST0toTAU**

92h	23:22	**Reserved**	
	21:16	**Length of TAU Instruction**	**PSTAUILen**
	15	**Enable Texture Register 12-15 Swapping to Internal Register 0-3 whenever TAU-to-ALU Switches** 0: Disable 1: Enable	**HPSSWPT2A**
	14	**Enable Internal Register 0-3 Swapping to Texture Register 12-15 whenever ALU-to-TAU Switches** 0: Disable 1: Enable	**HPSSWPA2T**
	13:9	**Reserved**	
	8:0	**Length of ALU Instruction**	**PSALUILen**

93h	23	**Instruction Switch from TAU to ALU after the 23rd TAU Instruction** 0: Operate next TAU instruction 1: Switch to operate ALU instruction	**HPSI23TtoA**
	22	**Instruction Switch from TAU to ALU after the 22ndTAU instruction**	**HPSI22TtoA**
	21	**Instruction Switch from TAU to ALU after the 21st TAU instruction**	**HPSI21TtoA**
	20	**Instruction Switch from TAU to ALU after the 20th TAU instruction**	**HPSI20TtoA**
	19	**Instruction Switch from TAU to ALU after the 19thTAU instruction**	**HPSI19TtoA**
	18	**Instruction Switch from TAU to ALU after the 18thTAU instruction**	**HPSI18TtoA**
	17	**Instruction Switch from TAU to ALU after the 17thTAU instruction**	**HPSI17TtoA**
	16	**Instruction Switch from TAU to ALU after the 16thTAU instruction**	**HPSI16TtoA**
	15	**Instruction Switch from TAU to ALU after the 15thTAU instruction**	**HPSI15TtoA**
	14	**Instruction Switch from TAU to ALU after the 14thTAU instruction**	**HPSI14TtoA**
	13	**Instruction Switch from TAU to ALU after the 13thTAU instruction**	**HPSI13TtoA**
	12	**Instruction Switch from TAU to ALU after the 12thTAU instruction**	**HPSI12TtoA**
	11	**Instruction Switch from TAU to ALU after the 11thTAU instruction**	**HPSI11TtoA**
	10	**Instruction Switch from TAU to ALU after the 10thTAU instruction**	**HPSI10TtoA**
	9	**Instruction Switch from TAU to ALU after the 9thTAU instruction**	**HPSI9TtoA**
	8	**Instruction Switch from TAU to ALU after the 8thTAU instruction**	**HPSI8TtoA**
	7	**Instruction Switch from TAU to ALU after the 7thTAU instruction**	**HPSI7TtoA**
	6	**Instruction Switch from TAU to ALU after the 6thTAU instruction**	**HPSI6TtoA**
	5	**Instruction Switch from TAU to ALU after the 5thTAU instruction**	**HPSI5TtoA**
	4	**Instruction Switch from TAU to ALU after the 4thTAU instruction**	**HPSI4toA**
	3	**Instruction Switch from TAU to ALU after the 3rd TAU instruction**	**HPSI3TtoA**
	2	**Instruction Switch from TAU to ALU after the 2nd TAU instruction**	**HPSI2TtoA**
	1	**Instruction Switch from TAU to ALU after the 1st TAU instruction**	**HPSI1TtoA**
	0	**Instruction Switch from TAU to ALU after the 0th TAU instruction**	**HPSI0TtoA**

94h	23:8	**Reserved**	
	7	**Instruction Switch from TAU to ALU after the 31st TAU instruction**	**HPS31TtoA**
	6	**Instruction Switch from TAU to ALU after the 30th TAU instruction**	**HPS30TtoA**
	5	**Instruction Switch from TAU to ALU after the 29th TAU instruction**	**HPS29TtoA**
	4	**Instruction Switch from TAU to ALU after the 28th TAU instruction**	**HPS28TtoA**
	3	**Instruction Switch from TAU to ALU after the 27th TAU instruction**	**HPS27TtoA**
	2	**Instruction Switch from TAU to ALU after the 26th TAU instruction**	**HPS26TtoA**
	1	**Instruction Switch from TAU to ALU after the 25th TAU instruction**	**HPS25TtoA**
	0	**Instruction Switch from TAU to ALU after the 24th TAU instruction**	**HPS24TtoA**

95h	23	**Instruction Switch from ALU to TAU after the 23rd ALU Instruction** 0: Operate next ALU instruction 1: Switch to operate TAU instruction	**HPSI23AtoT**
	22	**Instruction Switch from ALU to TAU after the 22nd ALU Instruction**	**HPSI22AtoT**
	21	**Instruction Switch from ALU to TAU after the 21st ALU Instruction**	**HPSI21AtoT**
	20	**Instruction Switch from ALU to TAU after the 20th ALU Instruction**	**HPSI20AtoT**
	19	**Instruction Switch from ALU to TAU after the 19th ALU Instruction**	**HPSI19AtoT**
	18	**Instruction Switch from ALU to TAU after the 18th ALU Instruction**	**HPSI18AtoT**
	17	**Instruction Switch from ALU to TAU after the 17th ALU Instruction**	**HPSI17AtoT**
	16	**Instruction Switch from ALU to TAU after the 16th ALU Instruction**	**HPSI16AtoT**
	15	**Instruction Switch from ALU to TAU after the 15th ALU Instruction**	**HPSI15AtoT**
	14	**Instruction Switch from ALU to TAU after the 14th ALU Instruction**	**HPSI14AtoT**
	13	**Instruction Switch from ALU to TAU after the 13th ALU Instruction**	**HPSI13AtoT**
	12	**Instruction Switch from ALU to TAU after the 12th ALU Instruction**	**HPSI12AtoT**
	11	**Instruction Switch from ALU to TAU after the 11th ALU Instruction**	**HPSI11AtoT**
	10	**Instruction Switch from ALU to TAU after the 10th ALU Instruction**	**HPSI10AtoT**
	9	**Instruction Switch from ALU to TAU after the 9th ALU Instruction**	**HPSI9AtoT**
	8	**Instruction Switch from ALU to TAU after the 8th ALU Instruction**	**HPSI8AtoT**
	7	**Instruction Switch from ALU to TAU after the 7th ALU Instruction**	**HPSI7AtoT**
	6	**Instruction Switch from ALU to TAU after the 6th ALU Instruction**	**HPSI6AtoT**
	5	**Instruction Switch from ALU to TAU after the 5th ALU Instruction**	**HPSI5AtoT**
	4	**Instruction Switch from ALU to TAU after the 4th ALU Instruction**	**HPSI4AtoT**
	3	**Instruction Switch from ALU to TAU after the 3rd ALU Instruction**	**HPSI3AtoT**
	2	**Instruction Switch from ALU to TAU after the 2nd ALU Instruction**	**HPSI2AtoT**
	1	**Instruction Switch from ALU to TAU after the 1st ALU Instruction**	**HPSI1AtoT**
	0	**Instruction Switch from ALU to TAU after the 0th ALU Instruction**	**HPSI0AtoT**

96h	23	**Instruction Switch from ALU to TAU after the 47th ALU Instruction** 0: Operate next ALU instruction 1: Switch to operate TAU instruction	**HPSI47AtoT**
	22	**Instruction Switch from ALU to TAU after the 46th ALU Instruction**	**HPSI46AtoT**
	21	**Instruction Switch from ALU to TAU after the 45th ALU Instruction**	**HPSI45AtoT**
	20	**Instruction Switch from ALU to TAU after the 44th ALU Instruction**	**HPSI44AtoT**
	19	**Instruction Switch from ALU to TAU after the 43rd ALU Instruction**	**HPSI43AtoT**
	18	**Instruction Switch from ALU to TAU after the 42nd ALU Instruction**	**HPSI42AtoT**
	17	**Instruction Switch from ALU to TAU after the 41st ALU Instruction**	**HPSI41AtoT**
	16	**Instruction Switch from ALU to TAU after the 40th ALU Instruction**	**HPSI40AtoT**
	15	**Instruction Switch from ALU to TAU after the 39th ALU Instruction**	**HPSI39AtoT**
	14	**Instruction Switch from ALU to TAU after the 38th ALU Instruction**	**HPSI38AtoT**
	13	**Instruction Switch from ALU to TAU after the 37th ALU Instruction**	**HPSI37AtoT**
	12	**Instruction Switch from ALU to TAU after the 36th ALU Instruction**	**HPSI36AtoT**
	11	**Instruction Switch from ALU to TAU after the 35th ALU Instruction**	**HPSI35AtoT**
	10	**Instruction Switch from ALU to TAU after the 34th ALU Instruction**	**HPSI34AtoT**
	9	**Instruction Switch from ALU to TAU after the 33th ALU Instruction**	**HPSI33AtoT**
	8	**Instruction Switch from ALU to TAU after the 32th ALU Instruction**	**HPSI32AtoT**
	7	**Instruction Switch from ALU to TAU after the 31st ALU Instruction**	**HPSI31AtoT**
	6	**Instruction Switch from ALU to TAU after the 30th ALU Instruction**	**HPSI30AtoT**
	5	**Instruction Switch from ALU to TAU after the 29th ALU Instruction**	**HPSI29AtoT**
	4	**Instruction Switch from ALU to TAU after the 28th ALU Instruction**	**HPSI28AtoT**
	3	**Instruction Switch from ALU to TAU after the 27th ALU Instruction**	**HPSI27AtoT**
	2	**Instruction Switch from ALU to TAU after the 26th ALU Instruction**	**HPSI26AtoT**
	1	**Instruction Switch from ALU to TAU after the 25th ALU Instruction**	**HPSI25AtoT**
	0	**Instruction Switch from ALU to TAU after the 24th ALU Instruction**	**HPSI24AtoT**

97h	23	**Instruction Switch from ALU to TAU after the 71st ALU Instruction** 0: Operate next ALU instruction 1: Switch to operate TAU instruction	**HPSI71AtoT**
	22	**Instruction Switch from ALU to TAU after the 70th ALU Instruction**	**HPSI70AtoT**
	21	**Instruction Switch from ALU to TAU after the 69th ALU Instruction**	**HPSI69AtoT**
	20	**Instruction Switch from ALU to TAU after the 68th ALU Instruction**	**HPSI68AtoT**
	19	**Instruction Switch from ALU to TAU after the 67th ALU Instruction**	**HPSI67AtoT**
	18	**Instruction Switch from ALU to TAU after the 66th ALU Instruction**	**HPSI66AtoT**
	17	**Instruction Switch from ALU to TAU after the 65th ALU Instruction**	**HPSI65AtoT**
	16	**Instruction Switch from ALU to TAU after the 64th ALU Instruction**	**HPSI64AtoT**
	15	**Instruction Switch from ALU to TAU after the 63rd ALU Instruction**	**HPSI63AtoT**
	14	**Instruction Switch from ALU to TAU after the 62nd ALU Instruction**	**HPSI62AtoT**
	13	**Instruction Switch from ALU to TAU after the 61st ALU Instruction**	**HPSI61AtoT**

	12	**Instruction Switch from ALU to TAU after the 60th ALU Instruction**	**HPSI60AtoT**
	11	**Instruction Switch from ALU to TAU after the 59th ALU Instruction**	**HPSI59AtoT**
	10	**Instruction Switch from ALU to TAU after the 58th ALU Instruction**	**HPSI58AtoT**
	9	**Instruction Switch from ALU to TAU after the 57th ALU Instruction**	**HPSI57AtoT**
	8	**Instruction Switch from ALU to TAU after the 56th ALU Instruction**	**HPSI56AtoT**
	7	**Instruction Switch from ALU to TAU after the 55th ALU Instruction**	**HPSI55AtoT**
	6	**Instruction Switch from ALU to TAU after the 54th ALU Instruction**	**HPSI54AtoT**
	5	**Instruction Switch from ALU to TAU after the 53rd ALU Instruction**	**HPSI53AtoT**
	4	**Instruction Switch from ALU to TAU after the 52nd ALU Instruction**	**HPSI52AtoT**
	3	**Instruction Switch from ALU to TAU after the 51st ALU Instruction**	**HPSI51AtoT**
	2	**Instruction Switch from ALU to TAU after the 50th ALU Instruction**	**HPSI50AtoT**
	1	**Instruction Switch from ALU to TAU after the 49th ALU Instruction**	**HPSI49AtoT**
	0	**Instruction Switch from ALU to TAU after the 48th ALU Instruction**	**HPSI48AtoT**

98h	23	**Instruction Switch from ALU to TAU after the 95th ALU Instruction** 0: Operate next ALU instruction 1: Switch to operate TAU instruction	**HPSI95AtoT**
	22	**Instruction Switch from ALU to TAU after the 94th ALU Instruction**	**HPSI94AtoT**
	21	**Instruction Switch from ALU to TAU after the 93rd ALU Instruction**	**HPSI93AtoT**
	20	**Instruction Switch from ALU to TAU after the 92nd ALU Instruction**	**HPSI92AtoT**
	19	**Instruction Switch from ALU to TAU after the 91st ALU Instruction**	**HPSI91AtoT**
	18	**Instruction Switch from ALU to TAU after the 90th ALU Instruction**	**HPSI90AtoT**
	17	**Instruction Switch from ALU to TAU after the 89th ALU Instruction**	**HPSI89AtoT**
	16	**Instruction Switch from ALU to TAU after the 88th ALU Instruction**	**HPSI88AtoT**
	15	**Instruction Switch from ALU to TAU after the 87th ALU Instruction**	**HPSI87AtoT**
	14	**Instruction Switch from ALU to TAU after the 86th ALU Instruction**	**HPSI86AtoT**
	13	**Instruction Switch from ALU to TAU after the 85th ALU Instruction**	**HPSI85AtoT**
	12	**Instruction Switch from ALU to TAU after the 84th ALU Instruction**	**HPSI84AtoT**
	11	**Instruction Switch from ALU to TAU after the 83rd ALU Instruction**	**HPSI83AtoT**
	10	**Instruction Switch from ALU to TAU after the 82nd ALU Instruction**	**HPSI82AtoT**
	9	**Instruction Switch from ALU to TAU after the 81st ALU Instruction**	**HPSI81AtoT**
	8	**Instruction Switch from ALU to TAU after the 80th ALU Instruction**	**HPSI80AtoT**
	7	**Instruction Switch from ALU to TAU after the 79th ALU Instruction**	**HPSI79AtoT**
	6	**Instruction Switch from ALU to TAU after the 78th ALU Instruction**	**HPSI78AtoT**
	5	**Instruction Switch from ALU to TAU after the 77th ALU Instruction**	**HPSI77AtoT**
	4	**Instruction Switch from ALU to TAU after the 76th ALU Instruction**	**HPSI76AtoT**
	3	**Instruction Switch from ALU to TAU after the 75th ALU Instruction**	**HPSI75AtoT**
	2	**Instruction Switch from ALU to TAU after the 74th ALU Instruction**	**HPSI74AtoT**
	1	**Instruction Switch from ALU to TAU after the 73rd ALU Instruction**	**HPSI73AtoT**
	0	**Instruction Switch from ALU to TAU after the 72nd ALU Instruction**	**HPSI72AtoT**

99h	23	**Instruction Switch from ALU to TAU after the 119th ALU Instruction** 0: Operate next ALU instruction 1: Switch to operate TAU instruction	**HPSI119AtoT**
	22	**Instruction Switch from ALU to TAU after the 118th ALU Instruction**	**HPSI118AtoT**
	21	**Instruction Switch from ALU to TAU after the 117th ALU Instruction**	**HPSI117AtoT**
	20	**Instruction Switch from ALU to TAU after the 116th ALU Instruction**	**HPSI116AtoT**
	19	**Instruction Switch from ALU to TAU after the 115th ALU Instruction**	**HPSI115AtoT**
	18	**Instruction Switch from ALU to TAU after the 114th ALU Instruction**	**HPSI114AtoT**
	17	**Instruction Switch from ALU to TAU after the 113rd ALU Instruction**	**HPSI113AtoT**
	16	**Instruction Switch from ALU to TAU after the 112nd ALU Instruction**	**HPSI112AtoT**
	15	**Instruction Switch from ALU to TAU after the 111st ALU Instruction**	**HPSI111AtoT**
	14	**Instruction Switch from ALU to TAU after the 110th ALU Instruction**	**HPSI110AtoT**
	13	**Instruction Switch from ALU to TAU after the 109th ALU Instruction**	**HPSI109AtoT**
	12	**Instruction Switch from ALU to TAU after the 108th ALU Instruction**	**HPSI108AtoT**
	11	**Instruction Switch from ALU to TAU after the 107th ALU Instruction**	**HPSI107AtoT**
	10	**Instruction Switch from ALU to TAU after the 106th ALU Instruction**	**HPSI106AtoT**
	9	**Instruction Switch from ALU to TAU after the 105th ALU Instruction**	**HPSI105AtoT**
	8	**Instruction Switch from ALU to TAU after the 104th ALU Instruction**	**HPSI104AtoT**
	7	**Instruction Switch from ALU to TAU after the 103rd ALU Instruction**	**HPSI103AtoT**
	6	**Instruction Switch from ALU to TAU after the 1012nd ALU Instruction**	**HPSI102AtoT**
	5	**Instruction Switch from ALU to TAU after the 101st ALU Instruction**	**HPSI101AtoT**
	4	**Instruction Switch from ALU to TAU after the 100th ALU Instruction**	**HPSI100AtoT**
	3	**Instruction Switch from ALU to TAU after the 99th ALU Instruction**	**HPSI99AtoT**
	2	**Instruction Switch from ALU to TAU after the 98th ALU Instruction**	**HPSI98AtoT**
	1	**Instruction Switch from ALU to TAU after the 97th ALU Instruction**	**HPSI97AtoT**
	0	**Instruction Switch from ALU to TAU after the 96th ALU Instruction**	**HPSI96AtoT**

9Ah	23	**Temporary Register 15(r15)'s initial setting for Dstination Check bit**	**HPSR15DstChk**
	22	**Temporary Register 14(r14)'s initial setting for Dstination Check bit**	**HPSR14DstChk**
	21	**Temporary Register 13(r13)'s initial setting for Dstination Check bit**	**HPSR13DstChk**
	20	**Temporary Register 12(r12)'s initial setting for Dstination Check bit**	**HPSR12DstChk**
	19	**Temporary Register 11(r11)'s initial setting for Dstination Check bit**	**HPSR11DstChk**
	18	**Temporary Register 10(r10)'s initial setting for Dstination Check bit**	**HPSR10DstChk**
	17	**Temporary Register 9(r9)'s initial setting for Dstination Check bit**	**HPSR9DstChk**
	16	**Temporary Register 8(r8)'s initial setting for Dstination Check bit**	**HPSR8DstChk**
	15	**Temporary Register 7(r7)'s initial setting for Dstination Check bit**	**HPSR7DstChk**
	14	**Temporary Register 6(r6)'s initial setting for Dstination Check bit**	**HPSR6DstChk**
	13	**Temporary Register 5(r5)'s initial setting for Dstination Check bit**	**HPSR5DstChk**
	12	**Temporary Register 4(r4)'s initial setting for Dstination Check bit**	**HPSR4DstChk**
	11	**Temporary Register 3(r3)'s initial setting for Dstination Check bit**	**HPSR3DstChk**
	10	**Temporary Register 2(r2)'s initial setting for Dstination Check bit**	**HPSR2DstChk**
	9	**Temporary Register 1(r1)'s initial setting for Dstination Check bit**	**HPSR1DstChk**

	8	**Temporary Register 0(r0)'s initial setting for Dstination Check bit**
		0: It is **not** necessary to check r0's valid bit whenever used as ALU's destination at the 1st ime.
		1: It is necessary to check r0's valid bit whenever used as ALU's destination at the 1st time.
		
		Note to Driver*:* Define HPSRnDstChk for PS's temporary register n. The default value is "0". If some temporary register is as both TAU and ALU's destination when 1st used, the HPSRnDstChk must be set as "1". HW would check r#'s valid bit to see if data from "txld" is valid or not before the ALU instruction which uses r# as "partial" destination. Here are some examples
		txld r0, t0;
		mov r0.a c0.a;
		……
		or
		txld r0, t0;
		txld r1, t1;
		mov r2, r1
		mov r0.a, c0.a:
		……
		The HPSR0DstChk must be set as "1" by driver. It is very, very important, wrong setting would resulted to wrong result.
			HPSR0DstChk

	7	**Instruction Switch from ALU to TAU after the 127th ALU Instruction** 0: Operate next ALU instruction 1: Switch to operate TAU instruction	**HPSI127AtoT**
	6	**Instruction Switch from ALU to TAU after the 126th ALU Instruction**	**HPSI126AtoT**
	5	**Instruction Switch from ALU to TAU after the 125th ALU Instruction**	**HPSI125AtoT**
	4	**Instruction Switch from ALU to TAU after the 124th ALU Instruction**	**HPSI124AtoT**
	3	**Instruction Switch from ALU to TAU after the 123rd ALU Instruction**	**HPSI123AtoT**
	2	**Instruction Switch from ALU to TAU after the 122nd ALU Instruction**	**HPSI122AtoT**
	1	**Instruction Switch from ALU to TAU after the 121st ALU Instruction**	**HPSI121AtoT**
	0	**Instruction Switch from ALU to TAU after the 120th ALU Instruction**	**HPSI120AtoT**

HParaType = 01h

Sub-Address AAh: SW Inspection

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
AAh	23:16	**Reserved**	
	15:0	**Flag Number for SW Inspection**	**HCRFlagNum**

HParaType 02h: Attribute of Texture Stage n (HParaSubType 00h to 0Fh)

The register table in this section is used for the following listed HParaSubTypes (from 00h to 0Fh).

HParaSubType = 0000 0000 (00h) -- For Texture 0
HParaSubType = 0000 0001 (01h) -- For Texture 1
HParaSubType = 0000 0010 (02h) -- For Texture 2
HParaSubType = 0000 0011 (03h) -- For Texture 3
HParaSubType = 0000 0100 (04h) -- For Texture 4
HParaSubType = 0000 0101 (05h) -- For Texture 5
HParaSubType = 0000 0110 (06h) -- For Texture 6
HParaSubType = 0000 0111 (07h) -- For Texture 7
HParaSubType = 0000 1000 (08h) -- For Texture 8
HParaSubType = 0000 1001 (09h) -- For Texture 9
HParaSubType = 0000 1010 (0Ah) -- For Texture A
HParaSubType = 0000 1011 (0Bh) -- For Texture B
HParaSubType = 0000 1100 (0Ch) -- For Texture C
HParaSubType = 0000 1101 (0Dh) -- For Texture D
HParaSubType = 0000 1110 (0Eh) -- For Texture E
HParaSubType = 0000 1111 (0Fh) -- For Texture F

HParaType = 02h (HParaSubType = 00h-0Fh)
Sub-Address 00h-30h

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
00h	23:0	**Face 0's Level 0 Base Address in unit of 256 bytes** This is A31 to A8.	**HTXnF0L0Bas**
01h	23:0	**Face 1's Level 0 Base Address in unit of 256 bytes** This is A31 to A8.	**HTXnF1L0Bas**
02h	23:0	**Face 2's Level 0 Base Address in unit of 256 bytes** This is A31 to A8. program it back to 0.	**HTXnF2L0Bas**
03h	23:0	**Face 3's Level 0 Base Address in unit of 256 bytes** This is A31 to A8.	**HTXnF3L0Bas**
04h	23:0	**Face 4's Level 0 Base Address in unit of 256 bytes** This is A31 to A8.	**HTXnF4L0Bas**
05h	23:0	**Face 5's Level 0 Base Address in unit of 256 bytes** This is A31 to A8.	**HTXnF5L0Bas**
06-17h	23:0	**Reserved**	
18h	23	**Force Miss for Texture n's Texture Cache Hit Detection** 0: Detect hit or miss normally 1: Always force to miss	**HTXnCHMiss**
	22:18	**Reserved**	
	17:16	**Texture Location of Face0** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HTXnF5LOC**
	15:14	**Texture Location of Face0** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HTXnF4LOC**
	13	**Reserved**	
	12	**Base Address Mode for Texture Sample n** 0: HTXSnLmOffset is not Offset related to Level 0, but an independent Base Address of Level m. 1: This mode is for cubic texture and planner mode texture Base Address of Level m = HTXSnFfL0Bas + HTXSnLmOffset	**HTXnBaseMode**
	11:8	**Mode of Texture n** 0000: Reserved 0001: 2 Dimension, both S and T coordinates. 0010: 3 Dimension volume texture. S,T,R coordinates. 0011: Cube Texture 1xxx: Projection Texture Others: Reserved	**HTXnMode**

	7:6	**Texture Location of Face0** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HTXnF3LOC**
	5:4	**Texture Location of Face2 or Cr Buffer for Y-Cb-Cr Format** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HTXnF2LOC**
	3:2	**Texture Location of Face1, Cb Buffer for Y-Cb-Cr Format, or Crb Buffer for Y-Crb Format** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HTXnF1LOC**

	1:0	**Texture Location of Face0 or Y Buffer for Y-Cb-Cr or Y-Crb Format**
		00: System Local Frame Buffer (S.L.)
		01: System Dynamic Frame Buffer (S.F.)
		10: Reserved (System Memory)
		11: Reserved
		
		The procedure to determine the location of accessed texture:
		Consider fetch the texture with level "l" of texture stage "n"'s face "f"
		If (HTXnBaseMode == 1 | HTXnPower2 == 1) {//with offset mode for the base address
		LOC = HTXnFfLOC
		} else {//with base mode for the base address
		If ("l" == 0) {
		LOC = HTXnF0LOC
		} else if ("l" <= 7) {
		LOC = HTXnLlLOC
		} else {// "l" == 8,9,10 or 11
		LOC = HTXnL8LOC
		}
		}
			HTXnF0LOC

19-1Fh	23:0	**Reserved**	
20h	23:12	**Height of texture Level 0, maximum to 2048** The Heights of each level are all calculated from HTXSnL0H Height at Level "l" = HTXSnL0H >> l	**HTXnL0H**
	11:0	**Width of Texture Level 0, maximum to 2048** The Widths of each level are all calculated from HTXSnL0W Height at Level "l" = HTXSnL0W >> l	**HTXnL0W**
21h	23:12	**Reserved**	
	11:0	**Length of texture Level 0, maximum to 2048 (for 3D volume texture's R axis)** The Lengths of each level are all calculated from HTXSnL0L Length at Level "l" = HTXSnL0L >> l, but NO MIP for 3D volume texture.	**HTXnL0L**
22h	23:18	**Reserved**	
	17	**Memory Mode of texture n** 0: Linear mode 1: Tile mode	**HTXnMMode**
	16	**Texture n 's Tile is 16-texel High** 0: Normal 8-texel high 1: 16-texel high	**HTXnTileH16**
	15	**Texture n 's Width and High are Both Power of 2** 0: Non-power of 2 texture 1: Power of 2 texture	**HTXnPower2**
	14:12	**Reserved**	
	11:8	**Exponential of Length of Texture n Level 0, maximum to 11(2^11)**	**HTXnL0EL**
	7:4	**Exponential of High of Texture n Level 0, maximum to 11(2^11)**	**HTXnL0EH**
	3:0	**Exponential of Width of Texture n Level 0, maximum to 11(2^11)**	**HTXnL0EW**
23-2Fh	23:0	**Reserved**	

30h	23:16	**Texture Format**
		Bit [23:19]
		00000: Reserved
		00001: Intensity Format
		R, G, B <= I
		A <= 1.0
		00010: Luminance Format
		R, G, B <= L
		A <= 1.0 or A
		00011: Alpha Format
		R, G, B <= 0.0
		A <= A
		00100: Reserved
		00101: Compressed Texture
		00110: YUV (Video Texture) Format
		00111: Format for BumpMaping
		11111-01000: Reserved
		10000: Reserved
		10001: ARGB_16bpp Format
		10010: Reserved
		10011: ARGB_32bpp Format
		10100: Reserved
		10101: ABGR_16bpp Format
		10110: Reserved
		10111: ABGR_32bpp Format
		11000: Reserved
		11001: RGBA_16bpp Format
		11010: Reserved
		11011: RGBA_32bpp Format
		11100: BGRA_16bpp Format
		11101: BGRA_32bpp Format
		11110: Floating Color Format
		11111: Scale (said Z format or only one component)
		If there is NO Alpha channel in the defined format, fill 1.0 as Alpha value.
		
		Bit [18:16]
		For Intensity Format
		000: Reserved
		001: Reserved
		010: T4 (Bit[3:0] = T)
		011: T8 (Bit[7:0] = T)
		1xx: Reserved
		
		For Luminance Format
		000: Reserved
		001: Reserved
		010: L4 (Bit[3:0] = L)
		011: L8 (Bit[7:0] = L)
		100: AL44 (Bit[7:4] = A, Bit[3:0] = L)
		101: AL88 (Bit[15:8] = A, Bit[7:0] = L)
		110: L16 (Bit[15:0] = L)
		For L16 format
		A = 1.0f
		R = G = B = float(L/65536)
		// if (L = FFFFh) R = G = B = 1.0
		// else fill “0” into mantissa
		111: Reserved
		
		For Alpha Format
		000: Reserved
		001: Reserved
		010: A4 (Bit[3:0] = A)
		011: A8 (Bit[7:0] = A)
		R = G = B = 0.0
		1xx: Reserved
		
		For Compressed Format
		000: Reserved
		001: DXT1, 16-bpp format
		010: DXT2, DXT3, 4-bit Alpha format
		011: DXT4, DXT5, 3-bit Alpha format
		100-101: Reserved
		The G8R8_G8B8 can be defined by set as YUV format’s package mode and set HTXSnYUV2RGBmode to RGB mode
		
		For YUV format (Video Texture)
		000: Package mode (Bit[31:24] = V or Cr, Bit[23:16] = Y1, Bit[15:8] = U or Cb, Bit[7:0] = Y0) A = 1.0
		001: AYUV (Bit[31:24] = A, Bit[23:16] = Y, Bit[15:8] = U, Bit[7:0] = V)
			HTXnFM

	15	**Texture Color Extending mode(excluding Alpha)** For Alpha channel, always extended with high color bit For YUV format(video texture) and HTXnYUV2RGBmode setten as 0, always extended with zero 0: Extending with high color bit 1: Extending with zero	**HTXnCExtend**
	14	**Inverse the Texel Order in One Byte for those texture with 1bpp, 2bpp or 3bpp** 0: Normal Consider Index 1: Bit[7] for tex[8n+7], bit[6] for tex[8n+6]……..bit[0] for tex[8n] Consider Index 2: Bit[7:6] for tex[4n+3], bit[5:4] for tex[4n+2]……..bit[1:0] for tex[4n] Consider Index 4: Bit[7:4] for tex[2n+1], bit[3:0] for tex[2n] 1: Inverse Consider Index 1: Bit[7] for tex[8n], bit[6] for tex[8n+1]……..bit[0] for tex[8n+7] Consider Index 2: Bit[7:6] for tex[4n], bit[5:4] for tex[4n+1]……..bit[1:0] for tex[4n+3] Consider Index 4: Bit[7:4] for tex[2n], bit[3:0] for tex[2n+1]	**HTXnInv124bpp**

	13:11	**Mode for Z(Depth) Format Texture** 000: D3D Mode R = 0.0 G = Z B = 0.0 A = 1.0 001: Shadow Map with Percentage Closer Filter as Greater or Equal Note that only for 24-bit fix Z format R = (accessed Z ≧ texture coordinate's R) ? 1.0 : 0.0 G = (accessed Z ≧ texture coordinate's R) ? 1.0 : 0.0 B = (accessed Z ≧ texture coordinate's R) ? 1.0 : 0.0 A = 1.0 010: Luminnance mode R = Z G = Z B = Z A = 1.0 011: Shadow Map with Percentage Closer Filter as Greater Note that only for 24-bit fix Z format R = (accessed Z > texture coordinate's R) ? 1.0 : 0.0 G = (accessed Z > texture coordinate's R) ? 1.0 : 0.0 B = (accessed Z > texture coordinate's R) ? 1.0 : 0.0 A = 1.0 100: Intensity mode R = Z G = Z B = Z A = Z 101: Shadow Map with Percentage Closer Filter as Smaller or Equal Note that only for 24-bit fix Z format R = (accessed Z ≦ texture coordinate's R) ? 1.0 : 0.0 G = (accessed Z ≦ texture coordinate's R) ? 1.0 : 0.0 B = (accessed Z ≦ texture coordinate's R) ? 1.0 : 0.0 A = 1.0 110: Alpha mode R = 0.0 G = 0.0 B = 0.0 A = Z 111: Shadow Map with Percentage Closer Filter as Smaller Note that only for 24-bit fix Z format R = (accessed Z < texture coordinate's R) ? 1.0 : 0.0 G = (accessed Z < texture coordinate's R) ? 1.0 : 0.0 B = (accessed Z < texture coordinate's R) ? 1.0 : 0.0 A = 1.0 Note1: For Z format texture(but shadow map with PCF excluded), the value should be transformed to s[7].16 floating-point before sent to Pixel Shader. Note2: For OpenGL Driver and Depth texture's border color, please setting the R, G, B, A component to HTXSnTBR, HTXSnTBG, HTXSnTBB and HTXSnTBA separately. Although ICD define the 1st component of texture border color as the "Depth" value, HW would just adopt HTXSnTBR, HTXSnTBG, HTXSnTBB and HTXSnTBA, not matter what is the HTXnMode's setting.	**HTXnZMode**
	10:9	**Reserved**	
	8	**Texture is as sRGB(non-gamma 1.0).** The deGamma correction is necessary by deGamma table "HDGTRTX". DeGamma correction is only used to R, G and B component. Any color format cauld be deGammaed just after the filtering (& color space conversation). 0: Disable DeGamma 1: Enable DeGamma	**TXnSRGB**
	7:2	**Reserved**	

	1:0	**Video Texture is as BT601(SDTV), BT709(HDTV) or just RGB** 00: RGB For this format, consider the 8-bit Y as positive 8-bit G, 8-bit U as positive 8-bit R, and 8-bit V as positive 8-bit B. Then extend them to s1.10 according to setting of HTXnCExtend. Then filter the texels. For this setting, YUV2RGB transform is not done in format decoder module or YUV2RGB transformed implemented by PS. 01: BT601(SDTV) R = clip(round(((Y – 16)*1.164383 + (V – 128)*1.596027)*256/255)) G = clip(round(((Y – 16)*1.164383 – (U – 128)*0.391762) – (V – 128)*0.812968))*256/255)) B = clip(round(((Y – 16)*1.164838 + (U – 128)*2.017232)*256/255)) 10: BT709(HDTV) R = clip(round(((Y – 16)*1.164383 – (U – 128)*0.0002) + (V – 128)*1.7927)*256/255)) G = clip(round(((Y – 16)*1.164383 – (U – 128)*0.2132) – (V – 128)*0.5329))*256/255)) B = clip(round(((Y – 16)*1.164838 + (U – 128)*2.2114) – (V – 128)*0.0001))*256/255)) 11: Table For 8-bit YUV R = clip(round((Y*A + U*B1 + V *C1 + D)*256/255)) G = clip(round((Y*A + U*B2 + V *C2 + D)*256/255)) B = clip(round((Y*A + U*B3 + V *C3 + D)*256/255)) For 10-bit YUV R = clip(round((Y*A + U*B1 + V *C1 + D)*1024/1023)) G = clip(round((Y*A + U*B2 + V *C2 + D)*1024/1023)) B = clip(round((Y*A + U*B3 + V *C3 + D)*1024/1023))	**HTXnYUV2RGBmode**
31-4Fh	23:0	**Reserved**	
50h	23:16	**Reserved**	
	15:0	**Texture Sample n's 1st Flag number for SW inspetion**	**HTXSnFlag1**
51h	23:16	**Reserved**	
	15:0	**Texture Sample n's 2nd Flag number for SW inspetion**	**HTXSnFlag2**

HParaType 02h: Attribute of Texture Sample Stage n (HParaSubType 20h to 2Fh)

The register table in this section is used for the following listed HParaSubTypes (from 20h to 2Fh).

HParaSubType = 0010 0000 (20h) For Texture Sample 0
HParaSubType = 0010 0001 (21h) For Texture Sample 1
HParaSubType = 0010 0010 (22h) For Texture Sample 2
HParaSubType = 0010 0011 (23h) For Texture Sample 3
HParaSubType = 0010 0100 (24h) For Texture Sample 4
HParaSubType = 0010 0101 (25h) For Texture Sample 5
HParaSubType = 0010 0110 (26h) For Texture Sample 6
HParaSubType = 0010 0111 (27h) For Texture Sample 7
HParaSubType = 0010 1000 (28h) For Texture Sample 8
HParaSubType = 0010 1001 (29h) For Texture Sample 9
HParaSubType = 0010 1010 (2Ah) For Texture Sample A
HParaSubType = 0010 1011 (2Bh) For Texture Sample B
HParaSubType = 0010 1100 (2Ch) For Texture Sample C
HParaSubType = 0010 1101 (2Dh) For Texture Sample D
HParaSubType = 0010 1110 (2Eh) For Texture Sample E
HParaSubType = 0010 1111 (2Fh) For Texture Sample F

HParaType = 02h (HParaSubType = 20h-2Fh)
Sub-Address 00h-40h

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
00-2Fh	23:0	**Reserved**	
30h	23:22	**Reserved**	
	21:12	**Texture Level 0 Offset** Format: 2's Complement Fix Point Number with 5 bits integer and 5 bits fraction The real Level 0 is (Texture Minimum Level + HTXnL0OS). **Bit [21:17]** 5 bits integer of Texture n Level 0 **Bit [16:12]** 5 bits fraction of Texture n Level 0	**HTXSnL0OS**
	11:6	**Maximum Texture Level** 000000: Texture n Maximum Level = 0 000001: Texture n Maximum Level = 1 000010: Texture n Maximum Level = 2 000011: Texture n Maximum Level = 3 000100: Texture n Maximum Level = 4 000101: Texture n Maximum Level = 5 000110: Texture n Maximum Level = 6 000111: Texture n Maximum Level = 7 001000: Texture n Maximum Level = 8 001001: Texture n Maximum Level = 9 001010: Texture n Maximum Level = A 001011: Texture n Maximum Level = B Others: Reserved	**HTXSnLVmax**
	5:0	**Minimum Texture Level** 000000: Texture n Minimum Level = 0 000001: Texture n Minimum Level = 1 000010: Texture n Minimum Level = 2 000011: Texture n Minimum Level = 3 000100: Texture n Minimum Level = 4 000101: Texture n Minimum Level = 5 000110: Texture n Minimum Level = 6 000111: Texture n Minimum Level = 7 001000: Texture n Minimum Level = 8 001001: Texture n Minimum Level = 9 001010: Texture n Minimum Level = A 001011: Texture n Minimum Level = B Others: Reserved	**HTXSnLVmin**
31h	23:21	**Reserved**	

	20:16	**Maximum Ratio for Anisotropy (max up to 16)** 00000: Max ratio to 0(no anisotropy) 00001: Max ratio to 1(no anisotropy) 00010: Max ratio to 2 00011: Max ratio to 3 00100: Max ratio to 4 …… 01111: Max ratio to 15 10000: Max ratio to 16 Others: Reserved	**HTXSnRatiomax**
	15:13	**Texture Filter Setting in S Direction for Texture Enlargement** 000: Nearest 001: Linear 010: Linear Anisotropy 011: 4x4 filter Others: Reserved	**HTXSnFLSe**
	12:10	**Texture Filter Setting in S Direction for Texture Shrinking** 000: Nearest 001: Linear 010: Linear Anisotropy 011: 4x4 filter Others: Reserved	**HTXSnFLSs**
	9:7	**Texture Filter Setting in T Direction for Texture Enlargement** 000: Nearest 001: Linear 010: Linear Anisotropy 011: 4x4 filter Others: Reserved ***HTXSnFLTe & HTXSnFLSe must be the same when Linear Anisotropy is set.**	**HTXSnFLTe**
	6:4	**Texture Filter Setting in T Direction For Texture Shrinking** 000: Nearest 001: Linear 010: Linear Anisotropy 011: 4x4 filter Others: Reserved *** HTXSnFLTs & HTXSnFLSs must be the same when Linear Anisotropy is set.**	**HTXSnFLT**
	3:0	**Texture Filter Setting in D Direction For Texture Shrinking** 0000: Always uses Texture Level 0 0001: Nearest 0010: Linear 0011: Reserved 0100: Dither Others: Reserved	**HTXSnFLDs**
32h	23:22	**Reserved**	
	21:19	**Texture Filter Setting in R Direction for Texture Enlargement (Only for 3D volume texture)** 0: Nearest 1: Linear	**HTXSnFLRe**
	18:16	**Texture Filter Setting in R Direction For Texture Shrinking (Only for 3D volume texture)** 0: Nearest 1: Linear	**HTXSnFLRs**
	15:12	**Reserved**	

	11:0	Texture Mapping Mode **Bit [11:9]** Reserved **Bit [8:6]** R Axis Setting (for 3D Volume texture) 000: Border Color 001: Clamp 010: Repeat 011: Mirror 100-101: Reserved 111: Mirror Once **Bit [5:3]** T Axis Setting 000: Border Color 001: Clamp 010: Repeat 011: Mirror 100-101: Reserved 111: Mirror Once **Bit [2:0]** S Axis Setting 000: Border Color 001: Clamp 010: Repeat 011: Mirror 100-101: Reserved 111: Mirror Once	**HTXSnMPMD**
33h	23:0	**Reserved**	
35h	23:12	**Red Color or U Component of Texture Border** As s1.10 from –1.0 to 1.0	**HTXSnTBR**
	11:0	**Green Color or Y Component of Texture Border** As s1.10 from –1.0 to 1.0	**HTXSnTBG**
36h	23:12	**Blue Color or V Component of Texture Border** As s1.10 from –1.0 to 1.0	**HTXSnTBB**

	11:0	**Texture Border Alpha**
		As s1.10 from –1.0 to 1.0
		
		Note for texture border color's usage:
		If (((HTXnFM == L16 or VU16 or G16R16 or G16FR16F or G32F))
		& (any filtered texel is border color)) {
		A channel of filter output = HTXSnTBA
		R channel of filter output = HTXSnTBR
		G channel of filter output = HTXSnTBG
		B channel of filter output = HTXSnTBB
		} else {
		If (the texel is border color) {
		A channel of filter input of the texel = HTXSnTBA
		R channel of filter input of the texel = HTXSnTBR
		G channel of filter input of the texel = HTXSnTBG
		B channel of filter input of the texel = HTXSnTBB
		} else { //the texel is just from the texture
		A channel of filter input of the texel = A after "color extending"
		R channel of filter input of the texel = R after "color extending"
		G channel of filter input of the texel = G after "color extending"
		B channel of filter input of the texel = B after "color extending"
		}
		}
		Note for YUV format and HTXnYUV2RGBmode is SDTV or HDTV, the border color is not in RGB space any more. It must be inverse transformed to YUV space as HTXSnTBG <= Y, HTXSnTBR <= U & HTXSnTBB <= V. And the HTXSnTBG[1:0], HTXSnTBG[1:0] & HTXSnTBR[1:0] are all zero.
			HTXSnTBA

37-40h	23:0	**Reserved**	

HParaType = 02h (HParaSubType = 20h-2Fh)

Sub-Address 50h-51h: SW Inspection for Texture Sample n

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic

50h	23:16	**Reserved**	
	15:0	**Texture Sample n's 1st Flag number for SW inspetion**	**HTXSnFlag1**
51h	23:16	**Reserved**	
	15:0	**Texture Sample n's 2nd Flag number for SW inspetion**	**HTXSnFlag2**

HParaType 02h: Attribute of Texture Stage n (HParaSubType FEh)

The register tables in this section are used for HParaSubType (FEh).

HParaType = 02h (HParaSubType = FEh)

Sub-Address 00h: For General Texture Attribute

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
00h	23:9	**Reserved**	
	8	**Check Divisor and Mantissa** Check if the divisor and dividend's mantissa in the instruction texldp are the same or not. If so, assign 1.0 to the mantissa of quotient. 0: Disable 1: Enable	**HTXtexldpchpM**
	7:4	**Number of Texture** n: n Texture, max up to 8	**HTXNum**
	3:2	**Configuration of Data FIFO for Reading Texture** 00: DFIFO1 is assigned to System memory's 1st T-Arbitrator(SL) DFIFO2 is assigned to System memory's 2nd T-Arbitrator(SF) DFIFO3 is assigned to Local Memory's T-Arbitrator 01: DFIFO1 is assigned to System memory's 1st T-Arbitrator(SL) DFIFO2 is assigned to Local Memory's T-Arbitrator DFIFO3 is assigned to System memory's 2nd T-Arbitrator(SF) 10: DFIFO1 is assigned to Local Memory's T-Arbitrator DFIFO2 is assigned to System memory's 2nd T-Arbitrator(SF) DFIFO3 is assigned to System memory's 1st T-Arbitrator(SL) 11: Reserved	**HTXDFIFOConfig**
	1	**Fetch Texture 2 QWs (256 bits) or 4QWs (512 bits) for Each Request** 0: Fetch 4 QWs(512 bits) for tiled-mode texture 1: Fetch 2QWs(256 bits)	**HTX2or4QWFetch**
	0	**Clear Texture Cache** 0: Don't care 1: Clear Texture Cache	**HTXCHCLR**

HParaType = 02h (HParaSubType = FEh)

Sub-Address 01h-07h: Texture 0 to Texture 7 is defined for Primitive Engine

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
01h	23	**Reserved**	
	22:21	**Perspective Mode of Texture 2** 00: Disable 01: Enable Perspective Correction 10: Enable Projection and be implemented to UVD module 11: Reserved	**HTX2PPmode**
	20:18	**Source of Texture 2** 000: Texture come from Texture A 001: Texture come from Texture B 010: Texture come from Texture C 011: Texture come from Texture D 100: Texture come from Texture E 101: Texture come from Texture F 110: Texture come from Texture G 111: Texture come from Texture H	**HTX2Src**
	17:16	**Dimension of Texture 2** 00: 1 Dimension, only S coordinate 01: 2 Dimension, Both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTX2Dim**
	15	**Reserved**	**HPST1toALU**
	14:13	**Perspective Mode of Texture 1** 00: Disable 01: Enable Perspective Correction 10: Enable Projection and be implemented to UVD module 11: Reserved	**HTX1PPmode**

	12:10	**Source of Texture 1** 000: Texture come from Texture A 001: Texture come from Texture B 010: Texture come from Texture C 011: Texture come from Texture D 100: Texture come from Texture E 101: Texture come from Texture F 110: Texture come from Texture G 111: Texture come from Texture H	**HTX1Src**
	9:8	**Dimension of Texture 1** 00: 1 Dimension, only S coordinate 01: 2 Dimension, Both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTX1Dim**
	7	**Reserved**	
	6:5	**Perspective Mode of Texture 0** 00: Disable 01: Enable Perspective Correction 10: Enable Projection and be implemented to UVD module 11: Reserved	**HTX0PPmode**
	4:2	**Source of Texture 0** 000: Texture come from Texture A 001: Texture come from Texture B 010: Texture come from Texture C 011: Texture come from Texture D 100: Texture come from Texture E 101: Texture come from Texture F 110: Texture come from Texture G 111: Texture come from Texture H	**HTX0Src**
	1:0	**Dimension of Texture 0** 00: 1 Dimension, only S coordinate 01: 2 Dimension, Both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTX0Dim**
02h	23	**Reserved**	
	22:21	**Perspective Mode of Texture 5** 00: Disable 01: Enable Perspective Correction 10: Enable Projection and be implemented to UVD module 11: Reserved	**HTX5PPmode**
	20:18	**Source of Texture 5** 000: Texture come from Texture A 001: Texture come from Texture B 010: Texture come from Texture C 011: Texture come from Texture D 100: Texture come from Texture E 101: Texture come from Texture F 110: Texture come from Texture G 111: Texture come from Texture H	**HTX5Src**
	17:16	**Dimension of Texture 5** 00: 1 Dimension, only S coordinate 01: 2 Dimension, Both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTX5Dim**
	15	**Reserved**	
	14:13	**Perspective Mode of Texture 4** 00: Disable 01: Enable Perspective Correction 10: Enable Projection and be implemented to UVD module 11: Reserved	**HTX4PPmode**
	12:10	**Source of Texture 4** 000: Texture come from Texture A 001: Texture come from Texture B 010: Texture come from Texture C 011: Texture come from Texture D 100: Texture come from Texture E 101: Texture come from Texture F 110: Texture come from Texture G 111: Texture come from Texture H	**HTX4Src**

	9:8	**Dimension of Texture 4** 00: 1 Dimension, only S coordinate 01: 2 Dimension, Both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTX4Dim**
	7	**Reserved**	
	6:5	**Perspective Mode of Texture 3** 00: Disable 01: Enable Perspective Correction 10: Enable Projection and be implemented to UVD module 11: Reserved	**HTX3PPmode**
	4:2	**Source of Texture 3** 000: Texture come from Texture A 001: Texture come from Texture B 010: Texture come from Texture C 011: Texture come from Texture D 100: Texture come from Texture E 101: Texture come from Texture F 110: Texture come from Texture G 111: Texture come from Texture H	**HTX3Src**
	1:0	**Dimension of Texture 3** 00: 1 Dimension, only S coordinate 01: 2 Dimension, Both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTX3Dim**
03h	23:15	**Reserved**	
	14:13	**Perspective Mode of Texture 7** 00: Disable 01: Enable Perspective Correction 10: Enable Projection and be implemented to UVD module 11: Reserved	**HTX7PPmode**
	12:10	**Source of Texture 7** 000: Texture come from Texture A 001: Texture come from Texture B 010: Texture come from Texture C 011: Texture come from Texture D 100: Texture come from Texture E 101: Texture come from Texture F 110: Texture come from Texture G 111: Texture come from Texture H	**HTX7Src**
	9:8	**Dimension of Texture 7** 00: 1 Dimension, only S coordinate 01: 2 Dimension, Both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTX7Dim**
	7	**Reserved**	
	6:5	**Perspective Mode of Texture 6** 00: Disable 01: Enable Perspective Correction 10: Enable Projection and be implemented to UVD module 11: Reserved	**HTX6PPmode**
	4:2	**Source of Texture 6** 000: Texture come from Texture A 001: Texture come from Texture B 010: Texture come from Texture C 011: Texture come from Texture D 100: Texture come from Texture E 101: Texture come from Texture F 110: Texture come from Texture G 111: Texture come from Texture H	**HTX6Src**
	1:0	**Dimension of Texture 6** 00: 1 Dimension, only S coordinate 01: 2 Dimension, Both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTX6Dim**
04h	23:20	**Exponential of Width for the Texture Coordinate Replaced by (x, y)** s = x / 2^ HTXXYrpSTHE	**HTXXYrpSTWE**
	19:16	**Exponential of High for the Texture Coordinate Replaced by (x, y)** t = y / 2^ HTXXYrpSTHE	**HTXXYrpSTHE**
	15:8	**Reserved**	

	7:0	**Use Screen Coordinate (x,y) to Replace (s, t)** **Bit [7]** 0: Normal (s,t) for Texture 7 1: Use (x,y) to replace (s, t) for Texture 7 **Bit [6]** 0: Normal (s,t) for Texture 6 1: Use (x,y) to replace (s, t) for Texture 6 **Bit [5]** 0: Normal (s,t) for Texture 5 1: Use (x,y) to replace (s, t) for Texture 5 **Bit [4]** 0: Normal (s,t) for Texture 4 1: Use (x,y) to replace (s, t) for Texture 4 **Bit [3]** 0: Normal (s,t) for Texture3 1: Use (x,y) to replace (s, t) for Texture 3 **Bit [2]** 0: Normal (s,t) for Texture 2 1: Use (x,y) to replace (s, t) for Texture 2 **Bit [1]** 0: Normal (s,t) for Texture 1 1: Use (x,y) to replace (s, t) for Texture 1 **Bit [0]** 0: Normal (s,t) for Texture 0 1: Use (x,y) to replace (s, t) for Texture 0	**HTXXYrpST**
05h	23:16	**Reserved**	
	15	**Enable of 2nd group of User Defined Clipping Planes** 0: Disable, No user defined clipping plane or only 1 texture is used for "User defined Clipping Plane" 1: Enable, a 2nd texture is used for "User defined Clipping Plane"	**HTXUCP1Enable**
	14	**Perspective Mode of the Texture for 2nd group of User Defined Clipping Planes** 0: Disable 1: Enable Perspective Correction	**HTXUCP1PPmode**
	13:10	**Source of the Texture for 2nd group of User Defined Clipping Planes** 0000: Texture come from Texture A 0001: Texture come from Texture B 0010: Texture come from Texture C 0011: Texture come from Texture D 0100: Texture come from Texture E 0101: Texture come from Texture F 0110: Texture come from Texture G 0111: Texture come from Texture H 1000: Texture come from Texture I 1001: Texture come from Texture J Others: Reserved	**HTXUCP1Src**
	9:8	**Dimension of the Texture for 2nd group of User Defined Clipping Planes** 00: 1 Dimension, only S coordinate 01: 2 Dimension, Both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXUCP1Dim**
	7	**Enable of 1st group of User Defined Clipping Planes** 0: Disable, No user defined clipping plane. 1: Enable, a texture is used for "User defined Clipping Plane"	**HTXUCP0Enable**
	6	**Perspective Mode of the Texture for 1st group of User Defined Clipping Planes** 0: Disable 1: Enable Perspective Correction	**HTXUCP0PPmode**
	5:2	**Source of the Texture for 1st group of User Defined Clipping Planes** 0000: Texture come from Texture A 0001: Texture come from Texture B 0010: Texture come from Texture C 0011: Texture come from Texture D 0100: Texture come from Texture E 0101: Texture come from Texture F 0110: Texture come from Texture G 0111: Texture come from Texture H 1000: Texture come from Texture I 1001: Texture come from Texture J Others: Reserved	**HTXUCP0Src**

	1:0	**Dimension of the Texture for 1st group of User Defined Clipping Planes** 00: 1 Dimension, only S coordinate 01: 2 Dimension, Both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate Note to Driver: It is forbidden to enable HTXUCP1Enable but disable HTXUCP0Enable.	**HTXUCP0Dim**
06-07h	23:0	**Reserved**	

HParaType = 02h (HParaSubType = FEh)

Sub-Address 08h-13h: Texture A to Texture H is defined in Vertex Buffer

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
08h	23:22	**Reserved**	
	21	**Texture S Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXBSMD**
	20	**Texture T Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXBTMD**
	19	**Texture R Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXBRMD**
	18	**Texture Q Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXBQMD**
	17:16	**Dimension of Texture B** 00: 1 Dimension, only S coordinate 01: 2 Dimension, both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXBDim**
	15:14	**Reserved**	**HTX1Src**
	13	**Texture S Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXASMD**
	12	**Texture T Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXATMD**
	11	**Texture R Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXARMD**
	10	**Texture Q Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXAQMD**
	9:8	**Dimension of Texture A** 00: 1 Dimension, only S coordinate 01: 2 Dimension, both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXADim**
	7:4	**Reserved**	
	3:0	**Number of Texture in Vertex Buffer** n: n Texture	**HTXNum4VP**
09h	23:22	**Reserved**	
	21	**Texture S Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXESMD**
	20	**Texture T Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXETMD**
	19	**Texture R Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXERMD**
	18	**Texture Q Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXEQMD**

	17:16	**Dimension of Texture E** 00: 1 Dimension, only S coordinate 01: 2 Dimension, both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXEDim**
	15:14	**Reserved**	**HTX1Src**
	13	**Texture S Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXDSMD**
	12	**Texture T Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXDTMD**
	11	**Texture R Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXDRMD**
	10	**Texture Q Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXDQMD**
	9:8	**Dimension of Texture D** 00: 1 Dimension, only S coordinate 01: 2 Dimension, both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXDDim**
	7:6	**Reserved**	
	5	**Texture S Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXCSMD**
	4	**Texture T Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXCTMD**
	3	**Texture R Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXCRMD**
	2	**Texture Q Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXCQMD**
	1:0	**Dimension of Texture C** 00: 1 Dimension, only S coordinate 01: 2 Dimension, both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXCDim**
0Ah	23:22	**Reserved**	
	21	**Texture S Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXHMD**
	20	**Texture T Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXHTMD**
	19	**Texture R Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXHRMD**
	18	**Texture Q Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXHQMD**
	17:16	**Dimension of Texture H** 00: 1 Dimension, only S coordinate 01: 2 Dimension, both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXHDim**
	15:14	**Reserved**	
	13	**Texture S Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXGSMD**
	12	**Texture T Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXGTMD**
	11	**Texture R Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXGRMD**

	10	**Texture Q Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXGQMD**
	9:8	**Dimension of Texture G** 00: 1 Dimension, only S coordinate 01: 2 Dimension, both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXGDim**
	7:6	**Reserved**	
	5	**Texture S Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXFSMD**
	4	**Texture T Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXFTMD**
	3	**Texture R Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXFRMD**
	2	**Texture Q Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXFQMD**
	1:0	**Dimension of Texture F** 00: 1 Dimension, only S coordinate 01: 2 Dimension, both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXFDim**
0Bh	23:16	**Reserved**	
	15	**Texture D Wrap Correction along S Coordinate** 0: No Wrap 1: Wrap	**HTXDSWrapC**
	14	**Texture D Wrap Correction along T Coordinate** 0: No Wrap 1: Wrap	**HTXDTWrapC**
	13:12	**Reserved**	
	11	**Texture C Wrap Correction along S Coordinate** 0: No Wrap 1: Wrap	**HTXCSWrapC**
	10	**Texture C Wrap Correction along T Coordinate** 0: No Wrap 1: Wrap	**HTXCTWrapC**
	9:8	**Reserved**	
	7	**Texture B Wrap Correction along S Coordinate** 0: No Wrap 1: Wrap	**HTXBSWrapC**
	6	**Texture B Wrap Correction along T Coordinate** 0: No Wrap 1: Wrap	**HTXBTWrapC**
	5:4	**Reserved**	
	3	**Texture A Wrap Correction along S Coordinate** 0: No Wrap 1: Wrap	**HTXASWrapC**
	2	**Texture A Wrap Correction along T Coordinate** 0: No Wrap 1: Wrap	**HTXATWrapC**
	1:0	**Rserved**	
0Ch	23:16	**Reserved**	
	15	**Texture H Wrap Correction along S Coordinate** 0: No Wrap 1: Wrap	**HTXHSWrapC**
	14	**Texture H Wrap Correction along T Coordinate** 0: No Wrap 1: Wrap	**HTXHTWrapC**
	13:12	**Reserved**	
	11	**Texture G Wrap Correction along S Coordinate** 0: No Wrap 1: Wrap	**HTXGSWrapC**
	10	**Texture G Wrap Correction along T Coordinate** 0: No Wrap 1: Wrap	**HTXGTWrapC**
	9:8	**Reserved**	

	7	**Texture F Wrap Correction along S Coordinate** 0: No Wrap 1: Wrap	**HTXFSWrapC**
	6	**Texture F Wrap Correction along T Coordinate** 0: No Wrap 1: Wrap	**HTXFTWrapC**
	5:4	**Reserved**	
	3	**Texture E Wrap Correction along S Coordinate** 0: No Wrap 1: Wrap	**HTXESWrapC**
	2	**Texture E Wrap Correction along T Coordinate** 0: No Wrap 1: Wrap	**HTXETWrapC**
	1:0	**Rserved**	
0Dh	23:16	**Exponential of Modulus**	**HTXModE**
	15:8	**Reserved**	
	7	**Do "Modulus of 2^HTXMODE" with texture H's coordinate**	**HTXHMod**
	6	**Do "Modulus of 2^HTXMODE" with texture G's coordinate**	**HTXGMod**
	5	**Do "Modulus of 2^HTXMODE" with texture F's coordinate**	**HTXFMod**
	4	**Do "Modulus of 2^HTXMODE" with texture E's coordinate**	**HTXEMod**
	3	**Do "Modulus of 2^HTXMODE" with texture D's coordinate**	**HTXDMod**
	2	**Do "Modulus of 2^HTXMODE" with texture C's coordinate**	**HTXCMod**
	1	**Do "Modulus of 2^HTXMODE" with texture B's coordinate**	**HTXBMod**
	0	**Do "Modulus of 2^HTXMODE" with texture A's coordinate**	**HTXAMod**
0Eh	23:14	**Reserved**	
	13	**Texture S Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXJSMD**
	12	**Texture T Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXJTMD**
	11	**Texture R Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXJRMD**
	10	**Texture Q Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXJQMD**
	9:8	**Dimension of Texture J** 00: 1 Dimension, only S coordinate 01: 2 Dimension, both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXJDim**
	7:6	**Reserved**	
	5	**Texture S Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXISMD**
	4	**TextureT Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXITMD**
	3	**Texture R Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXIRMD**
	2	**Texture Q Coordinate Input Mode** 0: Un-normalized 1: Normalized	**HTXIQMD**
	1	**Dimension of Texture I** 00: 1 Dimension, only S coordinate 01: 2 Dimension, both S and T coordinate 10: 3 Dimension volume texture. S, T, R coordinate 11: 4 Dimension. S, T, R, Q coordinate	**HTXIDim**
0Fh	23:0	**Reserved**	

10h	23	**YUV422 (Packet Mode) Texture Decode Mode**
		0: Even texel in S direction (Ye, U_n, V_n)
		odd texel in S direction (Yo, U_n, V_n)
		1: Even texel in S direction (Ye, U_n, V_n)
		odd texel in S direction (Yo, $(U_n + U_{n+1})/2$, $(V_n + V_{n+1})/2$)
		
		Note:
		If (HTXYUV422DM == true & HTXnFM == 00110 000b
		& (HTXSnFLSe == nearest for enlarge | HTXSnFLSs == nearest for shrink)
		& (odd texel in S direction) & (not the rightest texel of texture)) {
		Y = Yo
		U = $(U_n + U_{n+1})/2$
		V = $(V_n + V_{n+1})/2$
		} else {
		Y = Y
		U = U_n
		V = V_n
		}
			HTXYUV422DM

	22:12	**Coefficient D of YUV to RGB Conversion** Format as 2's complement s2.8	**HTXYUV2RGBD**
	11	**Reserved**	
	10:0	**Coefficient A of YUV to RGB Conversion** Format as 2's complement s2.8	**HTXYUV2RGBA**
11h	23	**Reserved**	
	22:12	**Coefficient C1 of YUV to RGB Conversion** Format as 2's complement s2.8	**HTXYUV2RGBC1**
	11	**Reserved**	
	10:0	**Coefficient B1 of YUV to RGB Conversion** Format as 2's complement s2.8	**HTXYUV2RGBB1**
12h	23	**Reserved**	
	22:12	**Coefficient C2 of YUV to RGB Conversion** Format as 2's complement s2.8	**HTXYUV2RGBC2**
	11	**Reserved**	
	10:0	**Coefficient B2 of YUV to RGB Conversion** Format as 2's complement s2.8	**HTXYUV2RGBB2**
13h	23	**Reserved**	
	22:12	**Coefficient C3 of YUV to RGB Conversion** Format as 2's complement s2.8	**HTXYUV2RGBC3**
	11	**Reserved**	
	10:0	**Coefficient B3 of YUV to RGB Conversion** Format as 2's complement s2.8	**HTXYUV2RGBB3**

HParaType 03h: Palette (HParaSubType 00h-22h)

HParaType = 03h (HParaSubType = 00h)
Sub-Address 00h: Texture Palette 0

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
00h	31:24	Reserved	
	23:16	Y of Texture Palette n Data	HTPnY
	15:8	U of Texture Palette n Data	HTPnU
	7:0	V of Texture Palette n Data	HTPnV

HParaType = 03h (HParaSubType = 01h)
Sub-Address 01h: Texture Palette1 (Reserved)

HParaType = 03h (HParaSubType = 02h)
Sub-Address 02h: Texture Palette1 (Reserved)

HParaType = 03h (HParaSubType = 03h)
Sub-Address 03h: Texture Palette1 (Reserved)

HParaType = 03h (HParaSubType = 04h)
Sub-Address 04h: Texture Palette1 (Reserved)

HParaType = 03h (HParaSubType = 05h)
Sub-Address 05h: Texture Palette1 (Reserved)

HParaType = 03h (HParaSubType = 06h)
Sub-Address 06h: Texture Palette1 (Reserved)

HParaType = 03h (HParaSubType = 07h)
Sub-Address 07h: Texture Palette1 (Reserved)

HParaType = 03h (HParaSubType = 08-0Fh)
Sub-Address 08-0Fh: Reserved

HParaType = 03h (HParaSubType = 10h)

Sub-Address 10h: Offset or Base Address of Texture from Level 1 to Level 11 for the 16 Texture Samples

Ther are 16 texture samples and each sample can be maximum to 8 levels, so there should be 16x8 = 128 entries
HaaraAdr 0 -> HTXS0L1Offset
HaaraAdr 1 -> HTXS0L2Offset
…….
HaaraAdr 7 -> HTXS0L8Offset
HaaraAdr 8 -> HTXS1L1Offset
HaaraAdr 9 -> HTXS1L2Offset
…….
HaaraAdr 15 -> HTXS1L8Offset
HaaraAdr 16 -> HTXS2L1Offset
HaaraAdr 17 -> HTXS2L2Offset
…….
HaaraAdr 23 -> HTXS2L8Offset
……..
……..
HaaraAdr 118 -> HTXSFL1Offset
HaaraAdr 119 -> HTXSFL2Offset
…….
HaaraAdr 127 -> HTXSFL8Offset

To sum up, consider HTXS**n**L**m**BasOffset, its entry is (**n***8 + **m**), where **n** is from 0 to Fh, and **m** is from 1 to 8h

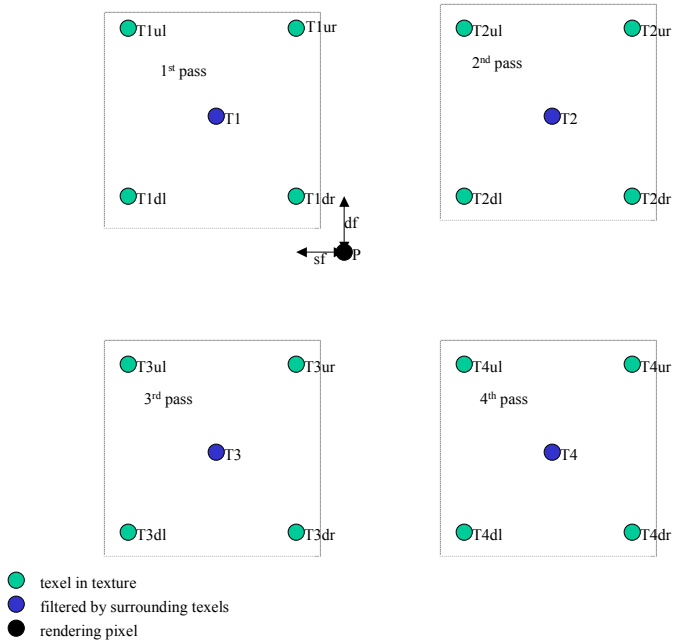
Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
10h	31:30	**Location of Texture Sample n's Level m** 00: Syntem Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved	**HTXnLmLOC**
	23:16	**Reserved**	
	7:0	**Offset Related to Texture Sample n's Level 0 for Level m Base Address** In Unit Byte,t his must be 256-byte boundary (in unit of 256 bytes or [31:8]).	**HTXnLmBasOffset**

HParaType = 03h (HParaSubType = 11h)

Sub-Address 11h: Texture 4x4 Filter Coeffiecient Table

There are 2^5 = 32 entries. The 5-bit fraction of "sf" or "tf" is as the index.

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
11h	31:22	**Reserved**	
	21:16	**Coefficient Cm for 4x4 Filter** Format as 1.5 positive fix-point maximum to 1.0	**HTX4X4FltCm**
	15	**Double the Coefficient Cr** 0: Crd = HTX4X4FltCrd 1: Crd = HTX4X4FltCrd << 1	**HTX4X4FltCrdE**
	14:8	**Coefficient C2 for 4x4 Filter** Format as s1.5 2's complement fix-point maximum to 1.0 and minimum to –0.5	**HTX4X4FltCrd**
	7	**Double the Coefficient Clu** 0: Clu = HTX4X4FltClu 1: Clu = HTX4X4FltClu << 1	**HTX4X4FltCluE**
	6:0	**Coefficient Clu for 4x4 Filter** Format as s1.5 2's complement fix-point maximum to 1.0 and minimum to –0.5	**HTX4X4FltClu**



Cl = HTX4X4Clu(sf) * 2^HTX4X4CluE(sf)
Cr = HTX4X4Crd(sf) * 2^HTX4X4CrdE(sf)
Csm = HTX4X4Cm(sf)
Cu = HTX4X4Clu(tf) * 2^HTX4X4CluE(tf)
Cd = HTX4X4Crd(tf) * 2^HTX4X4CrdE(tf)
Ctm = HTX4X4Cm(tf)

Cp1 = (1 – Csm)*(1 – Ctm)
Cp2 = Csm*(1 – Ctm)
Cp3 = (1 – Csm)*Ctm
Cp4 = Csm*Ctm

T1 = (1 – Cl)*(1 – Cu)*T1ul + Cl*(1 – Cu)*T1ur + (1 – Cl)*Cu*T1dl + Cl*Cu*T1dr
T2 = (1 – Cr)*(1 – Cu)*T2ul + Cr*(1 – Cu)*T2ur + (1 – Cr)*Cu*T2dl + Cr*Cu*T2dr
T3 = (1 – Cl)*(1 – Cd)*T3ul + Cl*(1 – Cd)*T3ur + (1 – Cl)*Cd*T3dl + Cl*Cd*T3dr
T4 = (1 – Cr)*(1 – Cd)*T4ul + Cr*(1 – Cd)*T4ur + (1 – Cr)*Cd*T4dl + Cr*Cd*T4dr
P = T1*Cp1 + T2*Cp2 + T3*Cp3 + T4*Cp4

HParaType = 03h (HParaSubType = 14h)
Sub-Address 14h: Stipple Palette

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
14h	31:0	32-Bit Stipple Palette Data	HSP

HParaType = 03h (HParaSubType = 15h, HParaAdr = 00h)
Sub-Address 15h: de-Gamma Table for Reading Texture

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
15h	31:30	Rounding Mode 00: Truncate 01: Rounding 1x: Reserved	HDGTRTXRnd
	29:20	De-Gamma Table Value for Reading Texture at C = 10'h0C0	HDGTRTX3
	19:10	De-Gamma Table Value for Reading Texture at C = 10'h080	HDGTRTX2
	9:0	De-Gamma Table Value for Reading Texture at C = 10'h040	HDGTRTX1

HParaType = 03h (HParaSubType = 15h, HParaAdr = 01h)
Sub-Address 15h: de-Gamma Table for Reading Texture

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
15h	31:30	Reserved	
	29:20	De-Gamma Table Value for Reading Texture at C = 10'h180	HDGTRTX6
	19:10	De-Gamma Table Value for Reading Texture at C = 10'h140	HDGTRTX5
	9:0	De-Gamma Table Value for Reading Texture at C = 10'h100	HDGTRTX4

HParaType = 03h (HParaSubType = 15h, HParaAdr = 02h)
Sub-Address 15h: de-Gamma Table for Reading Texture

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
15h	31:30	Reserved	
	29:20	De-Gamma Table Value for Reading Texture at C = 10'h240	HDGTRTX9
	19:10	De-Gamma Table Value for Reading Texture at C = 10'h200	HDGTRTX8
	9:0	De-Gamma Table Value for Reading Texture at C = 10'h1C0	HDGTRTX7

HParaType = 03h (HParaSubType = 15h, HParaAdr = 03h)
Sub-Address 15h: de-Gamma Table for Reading Texture

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
15h	31:30	Reserved	
	29:20	De-Gamma Table Value for Reading Texture at C = 10'h300	HDGTRTXC
	19:10	De-Gamma Table Value for Reading Texture at C = 10'h2C0	HDGTRTXB
	9:0	De-Gamma Table Value for Reading Texture at C = 10'h280	HDGTRTXA

HParaType = 03h (HParaSubType = 15h, HParaAdr = 04h)
Sub-Address 15h: de-Gamma Table for Reading Texture

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
15h	31:30	Reserved	
	29:20	De-Gamma Table Value for Reading Texture at C = 10'h3C0	HDGTRTXF
	19:10	De-Gamma Table Value for Reading Texture at C = 10'h380	HDGTRTXE
	9:0	De-Gamma Table Value for Reading Texture at C = 10'h340	HDGTRTXD

HParaType = 03h (HParaSubType = 17h, HParaAdr = 00h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	**Rounding Mode for R Channel** 00: Truncate 01: Rounding 1x: Reserved	HGTWRRnd
	29:20	**De-Gamma Table Value for Writing R Channel at R = 10'h001**	HGTWR1
	19:10	**De-Gamma Table Value for Writing G Channel at G = 10'h001**	HGTWG1
	9:0	**De-Gamma Table Value for Writing B Channel at B = 10'h001**	HGTWB1

HParaType = 03h (HParaSubType = 17h, HParaAdr = 01h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	**Rounding Mode for G Channel** 00: Truncate 01: Rounding 1x: Reserved	HGTWGRnd
	29:20	**De-Gamma Table Value for Writing R Channel at R = 10'h002**	HGTWR2
	19:10	**De-Gamma Table Value for Writing G Channel at G = 10'h002**	HGTWG2
	9:0	**De-Gamma Table Value for Writing B Channel at B = 10'h002**	HGTWB2

ParaType = 03h (HParaSubType = 17h, HParaAdr = 02h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	**Rounding Mode for B Channel** 00: Truncate 01: Rounding 1x: Reserved	HGTWBRnd
	29:20	**De-Gamma Table Value for Writing R Channel at R = 10'h003**	HGTWR3
	19:10	**De-Gamma Table Value for Writing G Channel at G = 10'h003**	HGTWG3
	9:0	**De-Gamma Table Value for Writing B Channel at B = 10'h003**	HGTWB3

HParaType = 03h (HParaSubType = 17h, HParaAdr = 03h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31	**Instead of Gamma Correction, but deGamma correction for R Channel** 0: Gamma correction 1: de-Gamma correction	HGTWRDeGamma
	30	**Reserved**	
	29:20	**De-Gamma Table Value for Writing R Channel at R = 10'h004**	HGTWR4
	19:10	**De-Gamma Table Value for Writing G Channel at G = 10'h004**	HGTWG4
	9:0	**De-Gamma Table Value for Writing B Channel at B = 10'h004**	HGTWB4

HParaType = 03h (HParaSubType = 17h, HParaAdr = 04h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31	**Instead of Gamma Correction, but deGamma correction for G Channel** 0: Gamma correction 1: de-Gamma correction	HGTWGDeGamma
	30	**Reserved**	
	29:20	**De-Gamma Table Value for Writing R Channel at R = 10'h006**	HGTWR5
	19:10	**De-Gamma Table Value for Writing G Channel at G = 10'h006**	HGTWG5
	9:0	**De-Gamma Table Value for Writing B Channel at B = 10'h006**	HGTWB5

HParaType = 03h (HParaSubType = 17h, HParaAdr = 05h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31	**Instead of Gamma Correction, but deGamma correction for B Channel** 0: Gamma correction 1: de-Gamma correction	HGTWBDeGamma
	30	**Reserved**	
	29:20	**De-Gamma Table Value for Writing R Channel at R = 10'h008**	HGTWR6
	19:10	**De-Gamma Table Value for Writing G Channel at G = 10'h008**	HGTWG6
	9:0	**De-Gamma Table Value for Writing B Channel at B = 10'h008**	HGTWB6

HParaType = 03h (HParaSubType = 17h, HParaAdr = 06h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	**Reserved**	
	29:20	**De-Gamma Table Value for Writing R Channel at R = 10'h00C**	HGTWR7
	19:10	**De-Gamma Table Value for Writing G Channel at G = 10'h00C**	HGTWG7
	9:0	**De-Gamma Table Value for Writing B Channel at B = 10'h00C**	HGTWB7

HParaType = 03h (HParaSubType = 17h, HParaAdr = 07h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	**Reserved**	
	29:20	**De-Gamma Table Value for Writing R Channel at R = 10'h010**	HGTWR8
	19:10	**De-Gamma Table Value for Writing G Channel at G = 10'h010**	HGTWG8
	9:0	**De-Gamma Table Value for Writing B Channel at B = 10'h010**	HGTWB8

HParaType = 03h (HParaSubType = 17h, HParaAdr = 08h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	**Reserved**	
	29:20	**De-Gamma Table Value for Writing R Channel at R = 10'h018**	HGTWR9
	19:10	**De-Gamma Table Value for Writing G Channel at G = 10'h018**	HGTWG9
	9:0	**De-Gamma Table Value for Writing B Channel at B = 10'h018**	HGTWB9

HParaType = 03h (HParaSubType = 17h, HParaAdr = 09h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h020	HGTWRA
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h020	HGTWGA
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h020	HGTWBA

HParaType = 03h (HParaSubType = 17h, HParaAdr = Ah)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h030	HGTWRB
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h030	HGTWGB
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h030	HGTWBB

HParaType = 03h (HParaSubType = 17h, HParaAdr = Bh)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h040	HGTWRC
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h040	HGTWGC
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h040	HGTWBC

HParaType = 03h (HParaSubType = 17h, HParaAdr = Ch)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h050	HGTWRD
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h050	HGTWGD
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h050	HGTWBD

HParaType = 03h (HParaSubType = 17h, HParaAdr = Dh)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h060	HGTWRE
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h060	HGTWGE
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h060	HGTWBE

HParaType = 03h (HParaSubType = 17h, HParaAdr = Eh)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h070	HGTWRF
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h070	HGTWGF
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h070	HGTWBF

HParaType = 03h (HParaSubType = 17h, HParaAdr = Fh)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h080	HGTWR10
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h080	HGTWG10
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h090	HTWB10

HParaType = 03h (HParaSubType = 17h, HParaAdr = 10h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h0A0	HGTWR11
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h0A0	HDGTWG11
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h0A0	HGTWB11

HParaType = 03h (HParaSubType = 17h, HParaAdr = 11h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h0C0	HGTWR12
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h0C0	HGTWG12
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h0C0	HGTWB12

HParaType = 03h (HParaSubType = 17h, HParaAdr = 12h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h0E0	HGTWR13
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h0E0	HGTWG13
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h0E0	HGTWB13

HParaType = 03h (HParaSubType = 17h, HParaAdr = 13h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h100	HGTWR14
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h100	HGTWG14
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h100	HGTWB14

HParaType = 03h (HParaSubType = 17h, HParaAdr = 14h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h140	HGTWR15
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h140	HGTWG15
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h140	HGTWB15

HParaType = 03h (HParaSubType = 17h, HParaAdr = 15h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h180	HGTWR16
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h180	HGTWG16
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h180	HGTWB16

HParaType = 03h (HParaSubType = 17h, HParaAdr = 16h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h1C0	HGTWR17
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h1C0	HGTWG17
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h1C0	HGTWB17

HParaType = 03h (HParaSubType = 17h, HParaAdr = 17h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h200	HGTWR18
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h200	HGTWG18
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h200	HGTWB18

HParaType = 03h (HParaSubType = 17h, HParaAdr = 18h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h240	HGTWR19
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h240	HGTWG19
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h240	HGTWB19

HParaType = 03h (HParaSubType = 17h, HParaAdr = 19h)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h280	HGTWR1A
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h280	HGTWG1A
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h280	HGTWB1A

HParaType = 03h (HParaSubType = 17h, HParaAdr = 1Ah)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h2C0	HGTWR1B
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h2C0	HGTWG1B
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h2C0	HGTWB1B

HParaType = 03h (HParaSubType = 17h, HParaAdr = 1Bh)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h300	HGTWR1C
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h300	HGTWG1C
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h300	HGTWB1C

HParaType = 03h (HParaSubType = 17h, HParaAdr = 1Ch)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h340	HGTWR1D
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h340	HGTWG1D
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h340	HGTWB1D

HParaType = 03h (HParaSubType = 17h, HParaAdr = 1Dh)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h380	HGTWR1E
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h380	HGTWG1E
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h380	HGTWB1E

HParaType = 03h (HParaSubType = 17h, HParaAdr = 1Eh)
Sub-Address 17h: de-Gamma Table for Writing Color

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
17h	31:30	Reserved	
	29:20	De-Gamma Table Value for Writing R Channel at R = 10'h3C0	HGTWR1F
	19:10	De-Gamma Table Value for Writing G Channel at G = 10'h3C0	HGTWG1F
	9:0	De-Gamma Table Value for Writing B Channel at B = 10'h3C0	HGTWB1F

HParaType = 03h (HParaSubType = 20h)

Sub-Address 20h: Pixel Shader ALU Instruction

Each instruction contains 4 double words. HParaAdr (4n+3) to (4n) are as the n-th instruction. Because of total 96 instructions, there are up to 96*4(=384) entries.

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
20h	31:0	**Pixel Shader ALU instruction**	**HPSALUINST**

HParaType = 03h (HParaSubType = 21h)

Sub-Address 21h: Pixel Shader TAU Instruction

There are totally 32 32-bit TAU instructions.

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
21h	31:0	**Pixel Shader TAU instruction**	**HPSTAUINST**

HParaType = 03h (HParaSubType = 22h)

Sub-Address 22h: Pixel Shader Constant Registers

Each constant register contains 4 32-bit components.
HParaAdr (4n) is as the 1st component of the n-th constant register.
HParaAdr (4n+1) is as the 2nd component of the n-th constant register.
HParaAdr (4n+2) is as the 3rd component of the n-th constant register.
HParaAdr (4n+3) is as the 4th component of the n-th constant register.
Where n is from 0 to 22, and 32 to 54
If HPSDbCnstR is set, the n from 32 to 54 is the same as the n from 0 to 22. And HW would fill n from 32 to 54 automatly.

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
22h	31:0	**Pixel Shader's Constant Register as Floating s[8].23** HW would automately transform the 32-bit floating to 24-bit floating format.	**HPSCnstReg**

HParaType 04h: Vertex and Primitive Setting

HParaType = 04h

Sub-Address 00h-1Fh: Flexible Vertex Format

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
00h	23:22	**Flexibility Vertex Format's (X, Y) Test Mode** 00: Disable 01: If (X or Y is "Not-a-Number) then ignore corresponded **primitive** 1x: If (X or Y is "Not-a-Number) then ignore corresponded **primitive list**	**HFVFXYTMode**
	21:20	**Flexibility Vertex Format's Z Test Mode** 00: Disable 01: If (Z is "Not-a-Number) then ignore corresponded **primitive** 1x: If (Z is "Not-a-Number) then ignore corresponded **primitive list**	**HFVFZTMode**
	19:18	**Flexibility Vertex Format's W Test Mode** 00: Disable 01: If (W is "Not-a-Number) then ignore corresponded **primitive** 1x: If (W is "Not-a-Number) then ignore corresponded **primitive list**	**HFVFWTMode**
	17	**Switch the Flexibility Vertex Format's X to Y, and Y to X** 0: Keep 1: Switch	**HFVFXYSwitch**
	16:15	**Location of Vertex Buffer** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: Reserved (System Memory) 11: Reserved Note to Driver and HW: All vertex buffers are located in SF.	**HVBLoc**
	14:6	**Reserved**	
	5:0	**Length of FVF Vertex Length (in unit of 32 bits)**	**HFVFLEN**
01h	23:22	**Reserved**	
	21:16	**The 3rd FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H03FVF**
	15:14	**Reserved**	
	13:8	**The 2nd FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H02FVF**
	7:6	**Reserved**	

	5:0	**The 1st FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)** 00h: X 01h: Y 02h: Z 03h: W 04h: Point Size 05h: Color 0 : Diffuse 06h: Color 1 : Specula 07h: Fog Factor 08h: S of TextureA 09h: T of TextureA 0Ah: R of TextureA 0Bh: Q of TextureA 0Ch: S of TextureB 0Dh: T of TextureB 0Eh: R of TextureB 0Fh: Q of TextureB 10h: S of TextureC 11h: T of TextureC 12h: R of TextureC 13h: Q of TextureC 14h: S of TextureD 15h: T of TextureD 16h: R of TextureD 17h: Q of TextureD 18h: S of TextureE 19h: T of TextureE 1Ah: R of TextureE 1Bh: Q of TextureE 1Ch: S of TextureF 1Dh: T of TextureF 1Eh: R of TextureF 1Fh: Q of TextureF 20h: S of TextureG 21h: T of TextureG 22h: R of TextureG 23h: Q of TextureG 24h: S of TextureH 25h: T of TextureH 26h: R of TextureH 27h: Q of TextureH 28h: S of TextureI 29h: T of TextureI 2Ah: R of TextureI 2Bh: Q of TextureI 2Ch: S of TextureJ 2Dh: T of TextureJ 2Eh: R of TextureJ 2Fh: Q of TextureJ 30h: Reserved 31h: Back Face Color0(BFCdiff) 32h: Back Face Color1(BFCspec)	**H01FVF**
02h	23:22	**Reserved**	
	21:16	**The 6th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H06FVF**
	15:14	**Reserved**	
	13:8	**The 5th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H05FVF**
	7:6	**Reserved**	
	5:0	**The 4th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H04FVF**
03h	23:22	**Reserved**	
	21:16	**The 9th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H09FVF**
	15:14	**Reserved**	
	13:8	**The 8th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H08FVF**
	7:6	**Reserved**	
	5:0	**The 7th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H07FVF**
04h	23:22	**Reserved**	
	21:16	**The 12th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H12FVF**
	15:14	**Reserved**	
	13:8	**The 11th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H11FVF**
	7:6	**Reserved**	
	5:0	**The 10th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H10FVF**
05h	23:22	**Reserved**	

	21:16	**The 15th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H15FVF**
	15:14	**Reserved**	
	13:8	**The 14th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H14FVF**
	7:6	**Reserved**	
	5:0	**The 13th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H13FVF**
06h	23:22	**Reserved**	
	21:16	**The 18th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H18FVF**
	15:14	**Reserved**	
	13:8	**The 17th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H17FVF**
	7:6	**Reserved**	
	5:0	**The 16th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H16FVF**
07h	23:22	**Reserved**	
	21:16	**The 21st FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H21FVF**
	15:14	**Reserved**	
	13:8	**The 20th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H20FVF**
	7:6	**Reserved**	
	5:0	**The 19nd FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H19FVF**
08h	23:22	**Reserved**	
	21:16	**The 24th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H24FVF**
	15:14	**Reserved**	
	13:8	**The 23rd FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H23FVF**
	7:6	**Reserved**	
	5:0	**The 22nd FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H22FVF**
09h	23:22	**Reserved**	
	21:16	**The 27th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H27FVF**
	15:14	**Reserved**	
	13:8	**The 26th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H26FVF**
	7:6	**Reserved**	
	5:0	**The 25th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H25FVF**
0Ah	23:22	**Reserved**	
	21:16	**The 30th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H30FVF**
	15:14	**Reserved**	
	13:8	**The 29th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H29FVF**
	7:6	**Reserved**	
	5:0	**The 28th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H28FVF**
0Bh	23:22	**Reserved**	
	21:16	**The 33th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H33FVF**
	15:14	**Reserved**	
	13:8	**The 32nd FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H32FVF**
	7:6	**Reserved**	
	5:0	**The 31st FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H31FVF**
0Ch	23:22	**Reserved**	
	21:16	**The36th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H36FVF**
	15:14	**Reserved**	
	13:8	**The 35th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H35FVF**
	7:6	**Reserved**	
	5:0	**The 34th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H34FVF**
0Dh	23:22	**Reserved**	
	21:16	**The 39th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H39FVF**
	15:14	**Reserved**	
	13:8	**The 38th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H38FVF**
	7:6	**Reserved**	
	5:0	**The 37th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H37FVF**
0Eh	23:18	**The 43rd FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H43FVF**
	17:12	**The 42nd FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H42FVF**
	11:6	**The 41st FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H41FVF**
	5:0	**The 40th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H40FVF**
0Fh	23:18	**The 47th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H47FVF**
	17:12	**The 46th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H46FVF**
	11:6	**The 45th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H45FVF**
	5:0	**The 44th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H44FVF**
10h	23:22	**Flexible Vertex Format Mask** Bit 23 - **BFCdiff Parameter Mask** 0: Primitive Vertex Parameter does not have Back Face Color0 1: Primitive Vertex Parameter has Back Face Color0 Bit 22 - **BFCspec Parameter Mask** 0: Primitive Vertex Parameter does not have Back Face Color1 1: Primitive Vertex Parameter has Back Face Color1	**HFVFMSK4**
	21:18	**Reserved**	

	17:12	**The 50th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H50FVF**
	11:6	**The 49th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H49FVF**
	5:0	**The 48th FVF Attribute Number. (X, Y, …..QJ, BFCdiff, BFCspec)**	**H48FVF**
11-1Fh	23:0	**Reserved**	

HParaType = 04h

Sub-Address 20h-3Fh: Vertex Buffer & Primitive Setting

There are totally 32 32-bit TAU instructions.

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
20h	23	**Disable Clipping for Triangle Point (Just near plane)** 0: Enable 1: Disable Note to Driver: This register is only for triangle with fill mode "point". When this register enabled, there might be 2 new points resulted fromnear plane clipping, and their attribute such as point size, color, texsture…. Are interpolated from the original 3 vertices. When disabled(HenCL4TriPoint_N = 1), HW just renders the vertices inside the view volumn.	**HenCL4TriPoint_N**
	22:21	**Reserved**	
	21:16	**The Setting of Second or Even One-vertex Triangle** **Bit [21:20] Setting of Vertex a** This setting is used for both triangle and line rendering. 00: Vertex a is a new input 01: The Vertex a will be replaced by previous Vertex a. 10: The Vertex a will be replaced by previous Vertex b. 11: The Vertex a will be replaced by previous Vertex c. **Bit [19:18] Setting of Vertex b** This setting is used for both triangle and line rendering. 00: Vertex b is a new input 01: The Vertex b will be replaced by previous Vertex a. 10: The Vertex b will be replaced by previous Vertex b. 11: The Vertex b will be replaced by previous Vertex c. **Bit [17:16] Setting of Vertex c** This setting is used for both triangle and line rendering. 00: Vertex c is a new input 01: The Vertex c will be replaced by previous Vertex a. 10: The Vertex c will be replaced by previous Vertex b. 11: The Vertex c will be replaced by previous Vertex c.	**H2nd1VT**
	15:8	**Primitive Render Mode** 00000000: Full Vertex Cycle For Triangle Rendering, this is the 3 vertexes cycle For Line Rendering, this is the 2 vertexes cycle 10xxxxxx: Reserved x1xxxxxx: Automatic Fast Primitive Vertex Cycle For Triangle Rendering, this is a fast way to render 3111 Mode For Line Rendering, this is a fast way to render 2111 Mode The first or odd single vertex cycle will use the setting of bit [5:0]. The second or even single vertex cycle will use the setting of H2nd1VT. **Bit [13:12] Setting of Vertex a** This setting is used for both triangle and line rendering. 00: Vertex a is a new input 01: The Vertex a will be replaced by previous Vertex a. 10: The Vertex a will be replaced by previous Vertex b. 11: The Vertex a will be replaced by previous Vertex c. **Bit [11:10] Setting of Vertex b** This setting is used for both triangle and line rendering. 00: Vertex b is a new input 01: The Vertex b will be replaced by previous Vertex a. 10: The Vertex b will be replaced by previous Vertex b. 11: The Vertex b will be replaced by previous Vertex c. **Bit [9:8] Setting of Vertex c** This setting is used for both triangle and line rendering. 00: Vertex c is a new input 01: The Vertex c will be replaced by previous Vertex a. 10: The Vertex c will be replaced by previous Vertex b. 11: The Vertex c will be replaced by previous Vertex c.	**HVCycle**
	7:2	**Reserved**	
	1	**Vertex Buffer Index Mode** 0: 16-bits index 1: 32-bits index	**HVBIndexMode**
	0	**Vertex Mode** 0: Command Mode Vertex 1: Index Mode Vertex	**HVertexMode**
21h	23:0	**Lower 3 Bytes of Vertex Buffer Base Address**	**HVBBaseL**

22h	23:16	**Reserved**	
	15:8	**Pitch of Each Vertex in Vertex Buffer**	**HVBPitch**
	7:0	**Higher Byte of Vertex Buffer Base Address**	**HVBBaseH**
23h	23:16	**Reserved**	
	15	**Last Pixel Control for Drawing Line** 0: Discard the last pixel of each line 1: Draw the last pixel of each line	**HLLastP**
	14:12	**Shading Setting** 000: Solid Shading 001: Flat Shading Via Vertex a 010: Flat Shading Via Vertex b 011: Flat Shading Via Vertex c 100: Gouraud Shading	**HShading**
	11:9	**Edge Flag** We assume the vertex transmission sequence of a triangle is a, b, then c. 000: Render NO Edge for triangle wire-frame or antialiasing 1xx: Render Edge(a, b) for triangle wire-frame or antialiasing x1x: Render Edge(b, c) for triangle wire-frame or antialiasing xx1: Render Edge(c, a) for triangle wire-frame or antialiasing	**HEFlag**
	8	**Back Face Mode for "Culling"** 0: If the vertex input is in the order of clockwise, it would be "culled" 1: If the vertex input is in the order of counterclockwise, it would be "culled"	**HBFace4Cull**

	7	**Back Face Mode for VS's output "oBFD#"**
		0: If the vertex input is in the order of clockwise, "oBFD#" would be selected as the vertex color
		1: If the vertex input is in the order of counterclockwise, "oBFD#" would be selected as the vertex color
		Note to Driver: Whenever the configure of VS's Output Registers contains oBD#, and HenBFCull is false, HW would select oD# or oBFD# as Color 0 and Color1. And the algorithm is as:
		If (HenBFCull & triangle meets HBFcae4Cull)
		Drop the triangle
		Else if ((FVFMask has Back Face Color0 | FVFMask has Back Face Color1)& triangle meet HBFcae4BFD)
		C0 = Back Face Color 0
		C1 = Back Face Color 1
		Else
		C0 = Color 0
		C1 = Color 1
			HBFace4BFD

	6:5	**Primitive Type for Clock-Wise Triangle** 00: Triangle Rendering for Hen2FRender enabled and clock-wise primitive 01: Reserved 10: Triangle Wire-frame Rendering for Hen2FRender enabled and clock-wise primitive 11: Triangle Point Rendering for Hen2FRender enabled and clock-wise primitive	**HPMTypeCW**
	4	**Render mode(PMType) is Different for Front-Face and Back-Face Primitive** The related PMType is "Triangle", "Triangle Wire-Frame" and "Triangle Point". 0: The PMType for both front-face and back-face is the same, or only one kind of face is rendered. 1: The PMType is different for front-face and back-face primitive.	**Hen2FRender**
	3:0	**Primitive Type** 0000: Point Rendering 0001: Line Rendering 0010: Triangle Rendering for "Hen2Frender" is false, or Hen2Frender enabled and counter-clock-wise primitive 0011: Reserved 0100: Rectangle 0101: Reserved 0110: Triangle Wire-frame Rendering for "Hen2Frender" is false, or Hen2Frender enabled and counter-clock-wise primitive 0111: Triangle Point Rendering for "Hen2Frender" is false, or Hen2Frender enabled and counter-clock-wise primitive 1xxx: Reserved	**HPMType**
24h	23:0	**Vertexes Number** n: There are n vertexes in current primitive list	**HVTXNum**
25h	23:0	**Vertexes Number** **Bit [23]: X Parameter Mask** 0: Primitive Vertex Parameter does not have X 1: Primitive Vertex Parameter has X **Bit [22]: Y Parameter Mask** 0: Primitive Vertex Parameter does not have Y	**HFVFMSK1**

		1: Primitive Vertex Parameter has Y **Bit [21]: Z Parameter Mask** 0: Primitive Vertex Parameter does not have Z 1: Primitive Vertex Parameter has Z **Bit [20]: W Parameter Mask** 0: Primitive Vertex Parameter does not have W 1: Primitive Vertex Parameter has W **Bit [19]: Point-Size Parameter Mask** 0: Primitive Vertex Parameter does not have Point-Size If the primitive type is point, use HVPointSize 1: Primitive Vertex Parameter has Point-Size If the primitive type is point, use the value from Vertex Data **Bit [18]: Cd(C0) Parameter Mask** 0: Primitive Vertex Parameter does not have Diffuse Color, ARGB 1: Primitive Vertex Parameter has Diffuse Color, ARGB **Bit [17]: Cs(C1) Parameter Mask** 0: Primitive Vertex Parameter does not have Specula Color, SR SG SB, and SA 1: Primitive Vertex Parameter has Specula Color, SR SG SB, and SA **Bit [16]: Fog-Factor Parameter Mask** 0: Primitive Vertex Parameter does not have Fog Factor 1: Primitive Vertex Parameter has Fog Factor **Bit [15]: Texture A's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [14]: Texture A's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [13]: Texture A's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [12]: Texture A's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [11]: Texture B's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [10]: Texture B's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [9]: Texture B's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [8]: Texture B's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [7]: Texture C's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [6]: Texture C's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [5]:Texture C's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [4]: Texture C's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [3]: Texture D's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [2]: Texture D's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [1]: Texture D's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [0]: Texture D's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q 1: Primitive Vertex Parameter has Texture Coordinate Q	

26h	23:16	**Flexible Vertex Format Mask** **Bit [23]: Texture I's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [22]: Texture I's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [21]: Texture I's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [20]: Texture I's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [19]: Texture J's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [18]: Texture J's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [17]: Texture J's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [16]: Texture J's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q 1: Primitive Vertex Parameter has Texture Coordinate Q	**HFVFMSK3**

	15:0	**Flexible Vertex Format Mask** **Bit [15]: TextureE's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [14]: Texture E's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [13]: Texture E's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [12]: Texture E's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [11]: Texture F's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [10]: Texture F's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [9]: Texture F's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [8]: Texture F's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [7]: Texture G's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [6]: Texture G's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [5]: Texture G's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [4]: Texture G's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [3]: Texture H's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [2]: Texture H's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [1]: Texture H's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [0]: Texture H's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q 1: Primitive Vertex Parameter has Texture Coordinate Q	**HFVFMSK2**
27h	23:0	**Starting Primitive Count**	**HVPPMCNTst**
28h	23:0	**Vertex Parameter Mask//Note that the "??" is by HVPDV_XX's Setting** **Bit [23]: X Parameter Mask** 0: Primitive Vertex Parameter does not have X thus use default Value 0.0 1: Primitive Vertex Parameter has X **Bit [22]: Y Parameter Mask** 0: Primitive Vertex Parameter does not have Y thus use default Value 0.0 1: Primitive Vertex Parameter has Y **Bit [21]: Z Parameter Mask** 0: Primitive Vertex Parameter does not have Z thus use default Value ?? 1: Primitive Vertex Parameter has Z **Bit [20]: W Parameter Mask** 0: Primitive Vertex Parameter does not have W thus use default Value ?? 1: Primitive Vertex Parameter has W **Bit [19]: Reserved** **Bit [18]: Cd(C0) Parameter Mask** 0: Primitive Vertex Parameter does not have Diffuse Color, ARGB thus use default Value ?? 1: Primitive Vertex Parameter has Diffuse Color, ARGB **Bit [17]: Cs(C1) Parameter Mask** 0: Primitive Vertex Parameter does not have Specula Color, SR SG SB, and SA thus use default Value ??	**HVPMask1**

		1: Primitive Vertex Parameter has Specula Color, SR SG SB, and SA **Bit [16]: Fog-Factor Parameter Mask** 0: Primitive Vertex Parameter does not have Fog Factor thus use default Value ?? 1: Primitive Vertex Parameter has Fog Factor **Bit [15]: Texture 0's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [14]: Texture 0's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [13]: Texture 0's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [12]: Texture 0's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [11]: Texture 1's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [10]: Texture 1's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [9]: Texture 1's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [8]: Texture 1's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [7]: Texture 2's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [6]: Texture 2's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [5]:Texture 2's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [4]: Texture 2's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [3]: Texture 3's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [2]: Texture 3's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [1]: Texture 3's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [0]: Texture 3's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate Q	
29h	23:16	**Reserved**	

	15:0	**Vertex Parameter Mask** **Bit [15]: Texture 4's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [14]: Texture 4's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [13]: Texture 4's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [12]: Texture 4's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [11]: Texture 5's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [10]: Texture 5's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [9]: Texture 5's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [8]: Texture 5's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [7]: Texture 6's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [6]: Texture 6's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [5]:Texture 6's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [4]: Texture 6's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate Q **Bit [3]: Texture 7's S Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate S thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate S **Bit [2]: Texture 7's T Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate T thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate T **Bit [1]: Texture 7's R Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate R thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate R **Bit [0]: Texture 7's Q Parameter Mask** 0: Primitive Vertex Parameter does not have Texture Coordinate Q thus use default Value ?? 1: Primitive Vertex Parameter has Texture Coordinate Q	**HVPMSK2**

2Ah	23:0	**Point's Default Width as Floating s[8].15**
		When point size comes from FVF, this setting is as the maximum value of point size
		Point Sprite is not only useful for 3D AP, but also 2D and vedio AP. This can be considered as just filled a rectangle and make use of all those 3D feature at the same time.
		The maximum value is 2048.0
		The minimum value is 1.0
		If (HFVFMask of "point size" is "0", and primitive type is point)
		Point_width = HVPointW
		Else
		Point_width = max(HVPointH, min(HVPointW, value from Primitive Vertex data's "point_size")
			HVPointW

2Bh	23:0	**Point's Default Hight as Floating s[8].15** When point size comes from FVF, this setting is as the minimum value of point size The maximum value is 2048.0 The minimum value is 1.0	**HVPointH**
2Ch	23:0	**Lower 24-bit for Maximum Value of Vertex Buffer's Index**	**HVBIndexMaxL**
2Dh	23:0	**Lower 24-bit for Minimum Value of Vertex Buffer's Index**	**HVBIndexMinL**
2Eh	23:16	**Reserved**	
	15:8	**Higher 8-bit for Maximum Value of Vertex Buffer's Index**	**HVBIndexMaxH**

	7:0	**Higher 8-bit for Minimum Value of Vertex Buffer's Index** If (index of vertex > HVBIndexMax | index of vertex < HVBIndexMin) Ignore this and the followed indices	**HVBIndexMinH**
2Fh	23	**Default Value for Vertex Parameter Z** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_Z**
	22	**Default Value for Vertex Parameter W** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_W**
	21	**Default Value for Vertex Parameter Fog** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_F**
	20	**Default Value for Vertex Parameter Color 0** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_C0**
	19	**Default Value for Vertex Parameter Alpha 0** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_A0**
	18	**Default Value for Vertex Parameter Color 1** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_C1**
	17	**Default Value for Vertex Parameter Alpha 1** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_A1**
	16	**Default Value for Vertex Parameter Texture Coordinate S** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_S**
	15	**Default Value for Vertex Parameter Texture Coordinate T** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_T**
	14	**Default Value for Vertex Parameter Texture Coordinate R** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_R**
	13	**Default Value for Vertex Parameter Texture Coordinate Q** 0: Default value is 0.0 1: Default value is 1.0	**HVPDV_Q**
	12:10	**Reserved**	
	9	**Point Sprite Enable for Texture J** 0: Disable 1: Enable	**HVPenPSpriteTXJ**
	8	**Point Sprite Enable for Texture I** 0: Disable 1: Enable	**HVPenPSpriteTXI**
	7	**Point Sprite Enable for Texture H** 0: Disable 1: Enable	**HVPenPSpriteTXH**
	6	**Point Sprite Enable for Texture G** 0: Disable 1: Enable	**HVPenPSpriteTXG**
	5	**Point Sprite Enable for Texture F** 0: Disable 1: Enable	**HVPenPSpriteTXF**
	4	**Point Sprite Enable for Texture E** 0: Disable 1: Enable	**HVPenPSpriteTXE**
	3	**Point Sprite Enable for Texture D** 0: Disable 1: Enable	**HVPenPSpriteTXD**
	2	**Point Sprite Enable for Texture C** 0: Disable 1: Enable	**HVPenPSpriteTXC**
	1	**Point Sprite Enable for Texture B** 0: Disable 1: Enable	**HVPenPSpriteTXB**

	0	**Point Sprite Enable for Texture A**
		0: Disable
		1: Enable
		
		Note to Driver: There are 2 enable setting for Point Sprite, "HenPSprite" and "HVPenPSpriteTXn", where n is from A to J. For OpenGL, the "HVPenPSpriteTXn" are used to set point sprite for **each** texture. But for D3D, "HenPSprite" is used to set point sprite for **ALL** textures. HW would do point sprite if any one of the 2 registers is true. That is
		For (n = A to J)
		If ((PMType is "point or "triangle point") & (HenPSprite | HVPenPSpriteTXn))
		Texture n is setup as "Point Sprite"
			HVPenPSpriteTXA

30-3Fh	23:0	**Reserved**	

HParaType = 04h

Sub-Address 40h-52h: Clipping Window to Screen Window Transformation Setting

All the transforming coefficients are 32-bit floating-point.

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
40h	23:0	**Lower 3 Bytes of Scaling for X Transform**	**HC2SXScaleL**
41h	23:0	**Lower 3 Bytes of Offset for X Transform**	**HC2SXOffsetL**
42h	23:16	**Reserved**	
	15:8	**Higher Byte of Offset for X Transform**	**HC2SXOffsetH**
	7:0	**Higher Byte of Scaling for X Transform**	**HC2SXScaleH**
43h	23:0	**Lower 3 Bytes of Scaling for Y Transform**	**HC2SYScaleL**
44h	23:0	**Lower 3 Bytes of Offset for Y Transform**	**HC2SYOffsetL**
45h	23:16	**Reserved**	
	15:8	**Higher Byte of Offset for Y Transform**	**HC2SYOffsetH**
	7:0	**Higher Byte of Scaling for Y Transform**	**HC2SYScaleH**
46h	23:0	**Lower 3 Bytes of Scaling for Z Transform**	**HC2SZScaleL**
47h	23:0	**Lower 3 Bytes of Offset for Z Transform**	**HC2SZOffsetL**
48h	23:16	**A Threshold Value For More Accurate Area Calculating** A threshold value of screen coordinate's Exponential part for more accurate area calculation. If any one screen coordinate's expomomtial is over "HC2SXYEmax4Area", the area calculating equation would be XbYc –XbYa – XaYc – XcYb +XcYa + XaYb. And add each term in the order from absolute largest to the absolute smallest. Note to HW: The minimum value of HC2SXYEmax4Area is 19+127(that is 2^19). HW has to have a check and adjustment as Used HC2SXYEmax4Area = max(146, decoded HC2SXYEmax4Area). Thus the original patterns not to be re-generated.	**HC2SXYEmax4Area**
	15:8	**Higher Byte of Offset for Z Transform**	**HC2SZOffsetH**
	7:0	**Higher Byte of Scaling for Z Transform**	**HC2SZScaleH**
49-4Fh	23:0	**Reserved**	
50h	23:0	**Lower 3 Bytes of Upper Clamping Value for Screen Coordinate**	**HC2SXYUClampL**
51h	23:0	**Lower 3 Bytes of Down Clamping Value for Screen Coordinate**	**HC2SXYDClampL**
52h	23:16	**Maximum Exponential Value Clipped Screen Coordinate** If the clipped new vertex's screen coordinate is over ±2^ HC2SXYEmax4CL, re-generate this clipped vertex to a smaller screen coordinate.	**HC2SXYEmax4CL**
	15:8	**Higher Byte of Upper Clamping Value for Screen Coordinate**	**HC2SXYUClampH**

	7:0	**Higher Byte of Down Clamping Value for Screen Coordinate**
		If (Xs >= HC2SXYUClamp)
		Xs = HC2SXYUClamp
		Else if (Xs <= HC2SXYDClamp)
		Xs = *HC2SXYDClamp
		If (Ys >= HC2SXYUClamp)
		Ys = HC2SXYUClamp
		Else if (Ys <= HC2SXYDClamp)
		Ys = HC2SXYDClamp
			HC2SXYDClampH

53-A9h	23:0	**Reserved**	
AAh	23:16	**Reserved**	
	15:0	**Flag Number for SW Inspection**	**HCRFlagNum**

HParaType 10h: Commands for Command Regulator

HParaType = 10h

Sub-Address 00h: PCI Command Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
00h	23:9	**Reserved**	
	8	**Enable of Package the PCI Command in front of CR**	**HenPCIP**
	7:0	**Number of ECLK Cycle for PCI Command Packaging Time out**	**HPCIPTimeOut**

HParaType = 10h

Sub-Address 02-03h: Read Register Back Command Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
02h	23:21	**Reserved**	
	20:12	**Reading-Back Register to Debug Port** Address [10:2] for Reading-Back Register to Debug port; will be muxed with HI2CR_RADR[8:2].	**HSetDBAdr**
	11	**Enable Reading-Back Register to Debug Port**	**HenRB2DB**
	10:8	**Reserved**	
	7:0	**ID for Reading-Back Register** 0000 0000: Reading the RB registers from CR 0000 0001: Reading the RB registers from FE(including VP, CL and SE) 0000 0010: Reading the RB registers from PE(including RZ and CZ) 0000 0011: Reading the RB registers from RC 0000 0100: Reading the RB registers from PS 0000 0101: Reading the RB registers from XE 0000 0110: Reading the RB registers from BE(including GEMI) Others: Reserved	**HSetRBGID**
03h	23:0	**Address for Reading-Back Register**	**HSetRBGAdr**

HParaType = 10h

Sub-Address 04-07h: Interrupt Command

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
04h	23:4	**Lower 24-bit of Interrupt State Buffer Base Address** It is A[23:0] which A[3:0] is useless, 128-bit alignment.	**HIRSBBasL**
	3:2	**Reserved**	
	1:0	**Interrupt State Buffer Location** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: System Memory (S.M.) 11: Reserved	**HIRSBLoc**
05h	23:9	**Reserved**	
	8	**Store 3D Engine's Register Setting in State Buffer** 0: Disable. It is not necessary to store 3D's register setting (default) 1: Enable. Store 3D's register setting into the State Buffer	**HIRSB4E3**
	7:0	**Higher 8-bit of Interrupt State Buffer Base Address** It is A[31:24].	**HIRSBBasH**
06h	23:0	**Threshold Value of the Left Vertex Data within One DIP in CR** In unit of Dword. When the left vertex data in CR are more than HIRFthStop and HIRStop is asserted, CR would stop 3D immediately. Otherwise, CR would stop 3D at the end of DIP	**HIRFthStop**
07h	23:2	**Reserved**	
	1	**Interrupt Request for Pause** Force CR into "Pause State" until this bit is cleared. When in "Pause State", CR just holds and not sends any command or data to the Video engine, 2D engine or 3D engine	**HIRPause**
	0	**Interrupt Request for Stop** When this register set, CR would interrupt the GPU and wait new Command triggered	**HIRStop**

Note: HIRPause and HIRStop can not be set at the same time.

HParaType = 10h

Sub-Address 10h: VMR Control

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
10h	23:1	**Reserved**	
	0	**VMR ID Buffer Status Reset**	**HIDRST**

HParaType = 10h

Sub-Address 60-68h: AGP Command Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
60h	23:4	**Lower 3 bytes of AGP Buffer Start Address** It is A[23:0] which A[3:0] is useless. 128-bit alignment.	**HAGPBstL**
	3:2	**Reserved**	
	1:0	**AGP Buffer Location** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: System Memory (S.M.) 11: Reserved	**HAGPBLoc**
61h	23:8	**Reserved**	
	7:0	**Higher Byte of AGP Buffer Start Address** It is A[31:24].	**HAGPBstH**
62h	23:0	**Lower 3 bytes of AGP Buffer End Address** It is A[23:0] which A[3:0] is useless. 128-bit alignment.	**HAGPBendL**
63h	23:8	**Reserved**	
	7:0	**Higher Byte of AGP Buffer End Address** It is A[31:24].	**HAGPBendH**
64h	23:0	**Lower 3 bytes of AGP Buffer Pause Address** It is A[31:0] which A[3:0] is useless. 128-bit alignment.	**HABPBpL**
65h	23:10	**Reserved**	
	9:8	**AGP Buffer Pause Address ID** 00: HAGPBp is the Pause Address of AGP Command Fetch. If AGP Command Fetcher wants to continuously fetch AGP Command again, he will start on the next address after HAGPBp. 01: HAGPBp is the End Address of a portion of AGP Command Block. When AGP Command Fetcher reaches this address, he will start to fetch next AGP Command addressed by HAGPBst. No waiting or pause at this time. 10: HAGPBp is the AGP Command Stop Address. Whenever, AGP Command Fetcher reach this address, the AGP Command Fetching is finished. If we want to do another AGP Command Fetching, we have to set HAGPBTrig as 1. 11: Reserved	**HAGPBpID**
	7:0	**Higher byte of AGP Buffer Pause Address** It is A[31:24]	**HAGPBpH**
66h	23:0	**Lower 3 bytes of AGP Buffer Jump Address** It is A[23:0] which A[3:0] is useless. 128-bit alignment.	**HAGPBjumpL**
67h	23:8	**Reserved**	
	7:0	**Higher Byte of AGP Buffer Jump Address** It is A[31:24].	**HAGPBjumpH**
68h	23:22	**Reserved**	
	21:16	**Threshold value of Read AGP Command** Default value is 8.	**HFthRCM**
	15:5	**Reserved**	
	4	**Clear Fence Queue**	**HFCQClear**
	3	**Clear AGP Cycle**	**HAGPBClear**
	2	**Trigger Restore AGP Command Cycle** HW will only restore AGP command.	**HRSTAGPTrig**
	1	**Trigger Restore 3D Register Cycle**	**HRSTTrig**
	0	**Trigger AGP Cycle** The Trig signal of AGP command.	**HAGPBTrig**

HParaType = 10h

Sub-Address 69h: Branch Command Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
69h	23:1	**Reserved**	
	0	**Default is On** 0: Disable AGP Branch. Branch AGP Header (FE8x) is forbidden. 1: Enable AGP Branch	**HenBranch**

HParaType = 10h

Sub-Address 6C-6Fh: Restore Command Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
6Ch	23:4	**Lower 3 bytes of Interrupt State Buffer Restore Start Address** It is A[23:4]. 128-bit alignment.	**HIRSBrstL**
	3:2	**Reserved**	
	1:0	**Interrupt State Buffer Restore Location** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: System Memory (S.M.) 11: Reserved	**HIRSBrstLoc**
6Dh	23:8	**Reserved**	
	7:0	**Higher Byte of Interrupt State Buffer Restore Start Address** It is A[31:24]. 128-bit alignment.	**HIRSBrstH**
6Eh	23:4	**Lower 3 bytes of AGP Buffer Restore Start Address** It is A[23:0] which A[3:0] is useless. 128-bit alignment.	**HAGPBrstL**
	3:0	**Reserved**	
6Fh	23:8	**Reserved**	
	7:0	**Higher Byte of AGP Buffer Restore Start Address** It is A[31:24]. 128-bit alignment.	**HAGPBrstH**

HParaType = 10h

Sub-Address 70-7Ch: CMDQ Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
70h	23:4	**Lower 3 bytes of Command Queue Start Address** It is A[23:0] which A[3:0] is useless. 128-bit alignment.	**HCMDQstL**
	3:2	**Reserved**	
	1:0	**Command Queue Location** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: System Memory (S.M.) 11: Reserved	**HCMDQLoc**
71h	23:0	**Lower 3 bytes of Command Queue End Address** It is A[23:0] which A[3:0] is useless. 128-bit alignment.	**HCMDQendL**
72h	23:16	**Reserved**	
	15:8	**Higher byte of Command Queue End Address** 128-bit alignment.	**HCMDQendH**
	7:0	**Higher byte of Command Queue Start Address** 128-bit alignment.	**HCMDQstH**
73h	23:0	**Length of Command Queue** In unit of 128 bits. The minimum value is 24'h800.	**HCMDQLen**
74h	23:22	**Reserved**	
	21:16	**Threshold Value of Write Command Queue FIFO**	**HFthCMDQW**
	15:14	**Reserved**	
	13:8	**Threshold Value of Read Command Queue FIFO**	**HFthCMDQR**

	7:4	**Control Setting for CMDQ Read Request Interrupting Write Request** 0000: Never interrupt. CMDQ Read request only after CMDQ write finished 0001: Interrupt CMDQ Write request whenever CMDQ Read request 1000: Interrupt whenever 16 CMDQ Write requests accepted 1001: Interrupt whenever 32 CMDQ Write requests accepted 1100: Interrupt whenever 32 CMDQ Write request cycle 1101: Interrupt whenever 64 CMDQ Write request cycle Others: Reserved	**HCMDQWReqMask**
	3	**Reserved**	
	2	**CMDQ is used for Command from AGP or PCI** 0: Store Command from PCI 1: Store Command from AGP	**HCMDQ4AGP**
	1	**Enable Request Length for CMDQ of Command Regulator** 0: Disable, always 1 128-bit command per request. 1: Enable it, 1, 2, or 4 128-bit commands per request is possible	**HenCMDQRLen**
	0	**Enable Command Queue** 0: Disable 1: Enable	**HenCMDQ**
78h	23:4	**Lower 3 bytes of Command Queue Start Address for 2D/3D Command** It is A[23:0] which A[3:0] is useless.	**HCMDQstL_E**
	3:2	**Reserved**	
	1:0	**Command Queue location for 2D/3D Command** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: System Memory (S.M.) 11: Reserved	**HCMDQLoc_E**
79h	23:0	**Lower 3 bytes of Command Queue End Address for 2D/3D Command** It is A[23:0] which A[3:0] is useless.	**HCMDQendL_E**
7Ah	23:16	**Reserved**	
	15:8	**Higher byte of Command Queue End Address for 2D/3D Command**	**HCMDQendH_E**
	7:0	**Higher byte of Command Queue Start Address for 2D/3D Command.**	**HCMDQstH_E**
7Bh	23:0	**Length of Command Queue for 2D/3D Command** In unit of 128 bits. The minimum value is 24'h800.	**HCMDQLen_E**
7Ch	23:14	**Reserved**	
	13:8	**Threshold value of Read Command Queue FIFO for 2D/3D Command**	**HFthCMDQR_E**
	7:4	**Control Setting for CMDQ Read Request interrupting Write Request for 2D/3D Command** 0000: Never interrupt. CMDQ Read request only after CMDQ write finished 0001: Interrupt CMDQ Write request whenever CMDQ Read request 1000: Interrupt whenever 16 CMDQ Write requests accepted 1001: Interrupt whenever 32 CMDQ Write requests accepted 1100: Interrupt whenever 32 CMDQ Write request cycle 1101: Interrupt whenever 64 CMDQ Write request cycle Others: Reserved	**HCMDQWReqMask_E**
	3	**Reserved**	**HFCQClear**
	2	**CMDQ is used for Command from AGP or PCI for 2D/3D Command** 0: Store Command from PCI 1: Store Command from AGP	**HCMDQ4AGP_E**
	1	**Enable Request Length for CMDQ of Command Regulator for 2D/3D Command** 0: Disable, always 1 128-bit command per request. 1: Enable, it1, 2, or 4 128-bit commands per request is possible	**HenCMDQRLen_E**
	0	**Enable Command Queue for 2D/3D Command** 0: Disable 1: Enable	**HenCMDQ_E**

HParaType 11h: Commands for Frame Buffer Swapping and CR's Miscellaneous Setting

HParaType = 11h

Sub-Address 00h: CR's Miscellaneous Setting

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
00h	23:17	**Reserved**	
	16	**Enable 2D/3D Request Lock Control** 0: Disable. When one of 3D/2D engine is busy, do not sent commands to another engine. 1: Enable. When one of 3D/2D engine is busy, CR sent commands to another engine but another engine do not access memory. Only one of the 2D and 3D engine is working in a time. However, E3FIRE can be sent to 3D engine when 2D engine is busy. Also, E2FIRE can send to 2D engine when 3D engine is busy. In the first case, Command Regulator will set CRLock3D before issuing E3FIRE and reset CRLock3D after 3D engine idle. Similarly, CRLock2D is working in the same way.	**HenGEMILock**
	15:11	**Reserved**	
	10:8	**The Depth n of FIFO 1 in CR** 000: n = 16 001: n = 24 010: n = 32 (default) 011: n = 48 100: n = 64 Others: Reserved	**HF1DEEPTYPE**
	7:5	**The Depth n of FIFO 2 in CR** 000: n = 16 001: n = 24 010: n = 32 (default) 011: n = 48 100: n = 64 Others: Reserved	**HF2DEEPTYPE**
	4:2	**The Depth n of FIFO 3 in CR** 000: n = 16 001: n = 24 010: n = 32 (default) 011: n = 48 100: n = 64 Others: Reserved	**HF3DEEPTYPE**
	1	**Reserved**	
	0	**Enable to Treat followed CMDs as "Critical"(not breakable)** 0: The Command Queue can be broken by an "Interrupt command" 1: The Command Queue can not be broken. The "STOP" procedure only being executed after "HenCriticalCMD" is cleared.	**HenCriticalCMD**

HParaType = 11h

Sub-Address 04-07h: Fence Command

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
04h	23:20	**Reserved**	
	19:0	**Higher Bits of Fence Command Base Address** If HFCMode[2]=1, higher 20-bit Fence Command Base address for PCI Master Write. It is A[43:24]. If HFCMode[2]=0, Higher 12-bit Fence Type. It is FenceType[31:20].	**HFCWBasH**
05h	23:4	**Lower bits of Fence Command base address** If HFCMode[2]=1, Lower 24-bit Fence Command Base address for PCI Master Write. It is A[23:4].	**HFCWBasL**
	3:0	**Reserved**	
06h	23:0	**Lower 24-bit Fence Command ID**	**HFCIDL**
07h	23	**Reserved**	
	22	**Fence Command Queue Full Skip – Active at First CR Command** (for HFCMode with Writes Fence Queue &generate interrupt signal) 0: CR command (41c/420) pause, until Fence Command Queue has space. **(default)** 1: CR keep running, overwrite Fence_ID to **current** write point in Fence queue.	**HFCQSkip**

	21	**Fence Command Interrupt Wait** (for HFCMode with Writes Fence Queue & generate interrupt signal) 0: CR keep runninge, when CR_INT active. And write Fence_ID to Fence queue **(default)** 1: CR pause, until CR_INT is cleared by Fence queue empty (not 204[5]).	**HFCIntWait**
	20	**Fence Command Trigger**	**HFCTrig**
	19:16	**Fence Command Mode** 0000: Just record the HFCID 0001: Record the HFCID when all engines idle 0010: Just writes Fence Queue & generate interrupt signal and record the HFCID 0011: Writes Fence Queue & generate interrupt signal and record the HFCID when all engines idle 0100: Just Write out HFCID to System Memory and record the HFCID 0101: Write out HFCID to System Memory and record the HFCID when all engines idle 0110: Just Write out HFCID to System Memory, writes Fence Queue & generate interrupt signal and record the HFCID 0111: Write out HFCID to System Memory, writes Fence Queue & generate interrupt signal and record the HFCID when all engines idle Others: Reserved	**HFCMode**
	15:8	**Higher 8-bit Fence Command ID**	**HFCIDH**
	7:0	**Reserved**	

HParaType = 11h

Sub-Address 08-0Bh: Save Command

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
08h	23:4	**Lower 24-bit of Save Buffer Start Base Address** It is A[23:0] which A[3:0] is useless. 128-bit alignment.	**HSvBStL**
	3:2	**Reserved**	
	1:0	**Save Buffer Location** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: System Memory (S.M.) 11: Reserved	**HSvBLoc**
09h	23:9	**Reserved**	
	8	**Store 3D Engine's Register Write Enable** 0: Disable. It is not necessary to store 3D's register. **(default)** 1: Enable. Store 3D's register.	**HSvWTEn**
	7:0	**Higher 8-bit of Save Buffer Start Base Address** It is A[31:24].	**HSvBStH**
0Bh	23:1	**Reserved**	
	0	**Save Trig for Save 3D Write Back Registers** When this register set, CR save 3D write back registers and then keep run.	**HSvTrig**

Note: Priority HIRSB4E3 > HSvWTEn, if HIRSB4E3=1, all 3D save register will save to HIRSBBas not HSvBSt.

HParaType = 11h

Sub-Address 10-34h: Frame Buffer Automatic Swapping

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
10h	23:3	**Lower 3 bytes of Display Frame Buffer Base Address of IGA1**	**HFB1BasL**
	2	**Reserved**	
	1:0	**Display Frame Buffer Location** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: System Memory (S.M.) 11: Reserved	**HFB1Loc**
11h	23:8	**Frame Flip Count of IGA1**	**HFlipCNT1**
	7:0	**Higher byte of Display Frame Buffer Base Address of IGA1**	**HFB1BasH**
12h	23:4	**Reserved**	
	3	**Enable Frame Buffer Automatic Swapping for IGA1**	**HenFB1ASwap**
	2	**Skip to Wait Blank when Automatic Swapping for IGA1 (default=0)**	**HskipFlipFB1**
	1:0	**Reserved**	
13-17h	23:0	**Reserved**	
18h	23:3	**Lower 3 bytes of Display Frame Buffer Base Address of IGA2**	**HFB2BasL**
	2	**Reserved**	
	1:0	**Display Frame Buffer Location of IGA2** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: System Memory (S.M.) 11: Reserved	**HFB2Loc**
19h	23:8	**Frame Flip Count of IGA2**	**HFlipCNT2**
	7:0	**Higher byte of Display Frame Buffer Base Address of IGA2**	**HFB2BasH**
1Ah	23:4	**Reserved**	
	3	**Enable Frame Buffer Automatic Swapping for IGA2**	**HenFB2ASwap**
	2	**Skip to Wait Blank when Automatic Swapping for IGA2. (default=0)**	**HskipFlipFB2**
	1:0	**Reserved**	
1B-2Fh	23:0	**Reserved**	
30h	23:3	**Lower 3 bytes of Display Frame Buffer Base Address of IGA Duo-view Control**	**HFBBasL**
	2	**Reserved**	
	1:0	**Display Frame Buffer location of IGA** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: System Memory (S.M.) 11: Reserved	**HFBLoc**
31h	23:8	**Frame Flip Count of IGA**	**HFlipCNT**
	7:0	**Higher byte of Display Frame Buffer Base Address of IGA**	**HFBBasH**
32h	23:4	**Reserved**	
	3	**Enable Frame Buffer Automatic Swapping for IGA**	**HenFBASwap**
	2	**Skip to wait blank when Automatic Swapping for IGA (default=0)**	**HskipFlipFB**
	1:0	**Reserved**	
33-34h	23:0	**Reserved**	

HParaType = 11h

Sub-Address 68-6Bh: Branch Command Setting

T11A68-6B, only active in "AGP format" command.

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
68h	23:1	**Reserved**	
	0	**Branch Type** 0: Branch for Normal (default) Branch commands independent on previous command, Insert Branch commands in CR command queue. 1: Branch for Restore Wait all previous command finish before Branch header (FE8x), and then active Branch request.	**HRstBranch**
69h	23:1	**Lower 3 bytes of Branch Buffer Start Address** It is A[23:1] which A[0] is useless. In unit of word(16-bit alignment) Note to Driver and HW: 1. The branch buffer address is modified to be alignment with 16 bits because of this address points to the vertex index in vertex buffer. 2. Branch for 3D vertex index in vertex buffer (16bits align.) is only from Header3 (Do not set in Header 2). 3. DWcount of header = valid vertex data DWcount + invalid vertex data DWcount before valid vertex data (ex: HAGPBranchL[3:1]=2 , i.e. invalid vertex data(16-bit alignment) DWcount before valid vertex data = 1)	**HAGPBranchL**
	0	**Reserved**	
6Ah	23:22	**Branch Buffer Location** 00: System Local Frame Buffer (S.L.) 01: System Dynamic Frame Buffer (S.F.) 10: System Memory (S.M.) 11: Reserved	**HAGPBranchLoc**
	21:8	**Reserved**	
	7:0	**Higher byte of Branch Buffer Start Address** It is A[31:24].	**HAGPBranchH**
6Bh	23:0	**Size of Branch Buffer** In unit of 16 bytes(128 bits).	**HAGPBranchSize**

Note: For Sub-Address 6Ah, branch command trigger is hidden in AGP header (FE8x). "NESTED" branch buffer is forbidden.

HParaType = 11h

Sub-Address AA-ABh: SW Inspection (R/W)

Bit [31:24] Sub-Address	Bit [23:0]	Description	Mnemonic
AAh	23:0	**SW Event TAG for SW Inspection**	**HSWFLAGAA**
ABh	23:0	**SW Event TAG for SW Inspection**	**HSWFLAGAB**

CR Registers in 2D Register Space (60-6Ch)

For detailed 2D register descriptions, please refer to 2D chapter.

Offset Address: 60h
3D / 2D ID Control

Bit	Attribute	Description	Mnemonic	Chip Rev	Sug
31	WO	**Reserved**			
30	WO	**3D / 2D Command Force Start (Software Must Fill Zero)**			
29:28	WO	**3D / 2D Command Status** 00: 3D / 2D command start 01: 3D / 2D command end 10: 3D / 2D command end and wait 3D idle 11: 3D / 2D command end and wait 2D idle			
27	WO	**3D / 2D Command Stream Kinds**			
26:24	WO	**3D / 2D Wokring Buffer Number**			
23:16	WO	**Reserved for Hardware Use**			
15:0	WO	**3D / 2D Working ID**			

Offset Address: 6Ch
3D / 2D Wait Control

Bit	Attribute	Description	Mnemonic	Chip Rev	Sug
31	WO	**Wait HQV0 IDLE**			
30	WO	**Wait HQV1 IDLE**			
29	WO	**Wait MCIDLE**			
28:24	WO	**Wait Idle Count**			
23	WO	**Wait 3D IDLE – for 3D / 2D Command Path**			
22	WO	**Wait 2D IDLE – for 3D / 2D Command Path**			
21	WO	**Wait DMA Channel 3 Idle**			
20	WO	**Wait DMA Channel 2 Idle**			
19	WO	**Wait DMA Channel 1 Idle**			
18	WO	**Wait DMA Channel 0 Idle**			
17	WO	**Wait LCD DN Idle**			
16:14	WO	**Reserved**			
13	WO	**Command Wait Later IGA VBLK Interval to Start to Work**			
12	WO	**Command Wait Former IGA VBLK Interval to Start to Work**			
11	WO	**Command Wait Later IGA VBLK End Pulse to Start to Work**			
10	WO	**Command Wait Former IGA VBLK End Pulse to Start to Work**			
9	WO	**Command Wait Later IGA VBLK Start Pulse to Start to Work**			
8	WO	**Command Wait Former IGA VBLK Start Pulse to Start to Work**			
7:6	WO	**Reserved**			
5	WO	**Command Wait IGA2 VBLK Interval to Start to Work**			
4	WO	**Command Wait IGA1 VBLK Interval to Start to Work**			
3	WO	**Command Wait IGA2 VBLK End Pulse to Start to Work**			
2	WO	**Command Wait IGA1 VBLK End Pulse to Start to Work**			
1	WO	**Command Wait IGA2 VBLK Start Pulse to Start to Work**			
0	WO	**Command Wait IGA1 VBLK Start Pulse to Start to Work**			

CR Registers in Video Control Register Space (3260-326Ch)

For detailed video register descriptions, please refer to Video chapter.

Offset Address: 3260h

Video ID Control

Bit	Attribute	Description	Mnemonic	Chip Rev	Sug
31	WO	**Reserved**			
30	WO	**Video Command Force Start (Software Must Fill Zero)**			
29:28	WO	**Video Command Status** 00: Video command start 01: Video command end 10: Video command end and wait Video idle 11: Video command end and wait Video idle			
27	WO	**Video Command Stream Kinds**			
26:24	WO	**Video Wokring Buffer Number**			
23:16	WO	**Reserved for Hardware Use**			
15:0	WO	**Video Working ID**			

Offset Address: 326Ch

Video Wait Control

Bit	Attribute	Description	Mnemonic	Chip Rev	Sug
31	WO	**Wait 3D IDLE – for Video Command Path**			
30	WO	**Wait 2D IDLE – for Video Command Path**			
29:25	WO	**Wait Idle Count**			
24:22	WO	**Reserved**			
21	WO	**Wait HQV1 Starting Address Load**			
20	WO	**Wait HQV1 Fire Bit**			
19	WO	**Wait HQV1 HW Flip FIFO Full**			
18	WO	**Wait HQV1 Subpicture / Updateoverlay Flip**			
17	WO	**Wait HQV1 SW Flip**			
16	WO	**Wait HQV1 Finish a Frame**			
15	WO	**Wait DMA Channel 3 Idle**			
14	WO	**Wait DMA Channel 2 Idle**			
13	WO	**Wait DMA Channel 1 Idle**			
12	WO	**Wait DMA Channel 0 Idle**			
11:6	WO	**Reserved**			
5	WO	**Wait HQV0 Starting Address Load**			
4	WO	**Wait HQV0 Fire Bit**			
3	WO	**Wait HQV0 HW Flip FIFO Full**			
2	WO	**Wait HQV0 Subpicture / Updateoverlay Flip**			
1	WO	**Wait HQV0 SW Flip**			
0	WO	**Wait HQV0 Finish a Frame**			