MB86292<ORCHID>
Graphics Controller Specifications

Revision 0.10

Oct2003

FUJITSU

CONTENTS

1. Overview

1.1 Introduction

The MB86292 <ORCHID> graphics controller develops the FUJITSU MB86290A graphics controller; it enables geometry processing and digital video capture.

Building SDRAM into this controller also achieves high-band graphics data transfer, providing high-speed graphics processing.

Target applications

- Car navigation systems
- Mobile equipment
- Digital information home electric appliance
- Amusement equipment

1.2 System Configuration

The following figure shows an example of the application of the MB86292 to a car navigation system.

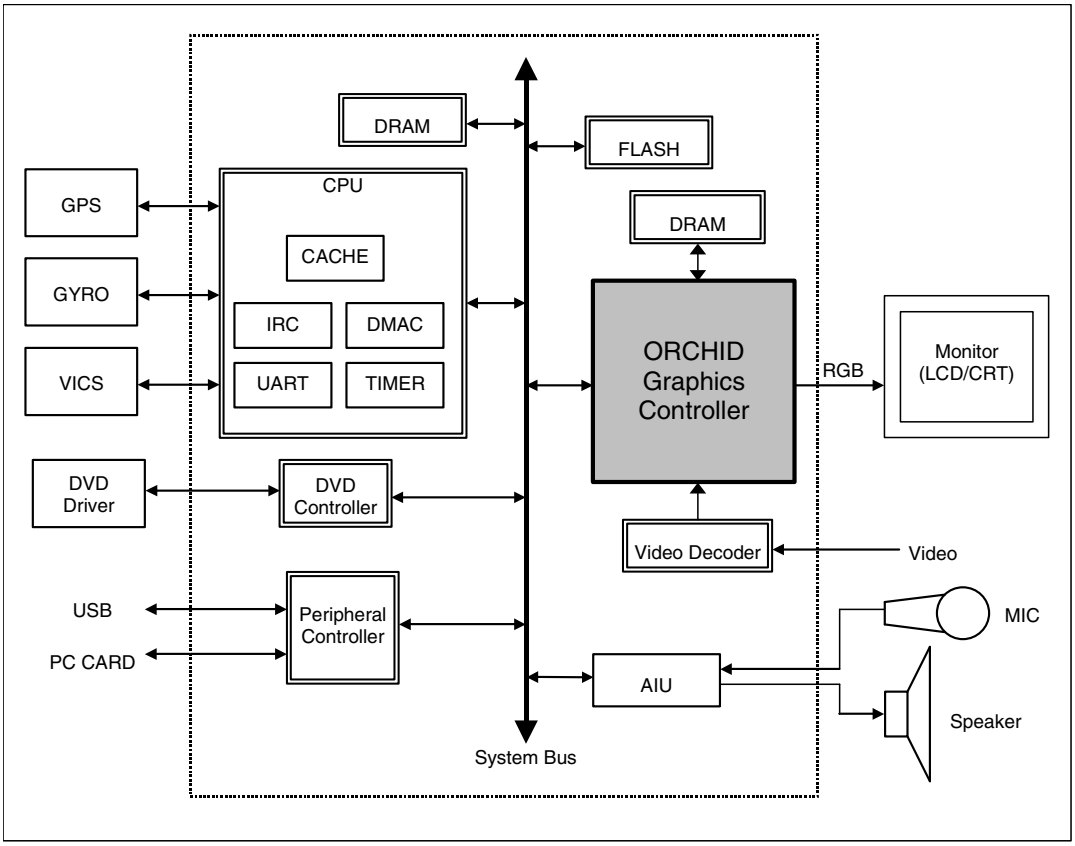


Fig.1.1 System Configuration (TBD)

1.3 Outline

- Built-in geometry engine

  The MB86292 has a built-in hardware engine that performs geometry processing such as coordinate transformation. This enables a large reduction in numerical computation for graphics processing with high CPU loads in the embedded system.

- Digital video capture

  The digital video capture function can store digital video data such as TV in graphics memory; it can display rendered graphics and video graphics on the same screen.

- Display controller

  The MB86292 has a built-in display controller that is compatible with the MB86290A.

  This controller provides functions such as XGA display (1024 x 768 pixels), 4-layer overlay, left/right split display, wrap-around scrolling, double buffers, and translucent display.

  In addition to analog RGB output, this controller supports digital RBG output and picture-in-picture video data.

- 2D and 3D Rendering

  The MB86292 has a rendering function that is compatible with the MB86290A. It can render data with the display list created for the MB86290A.

  The MB96291 also supports 3D rendering, such as perspective texture mapping with perspective collection and Gouraud shading, alpha bending, and anti-aliasing for rendering smooth lines.

- Others

  CMOS technology with 0.25-µm DRAM

  HQFP256 Package (lead pitch 0.4 mm)

  Supply voltage : 2.5 V (internal operation)/3.3 V (I/O)

1.4 Block Diagram

ORCHID block diagram is shown below:

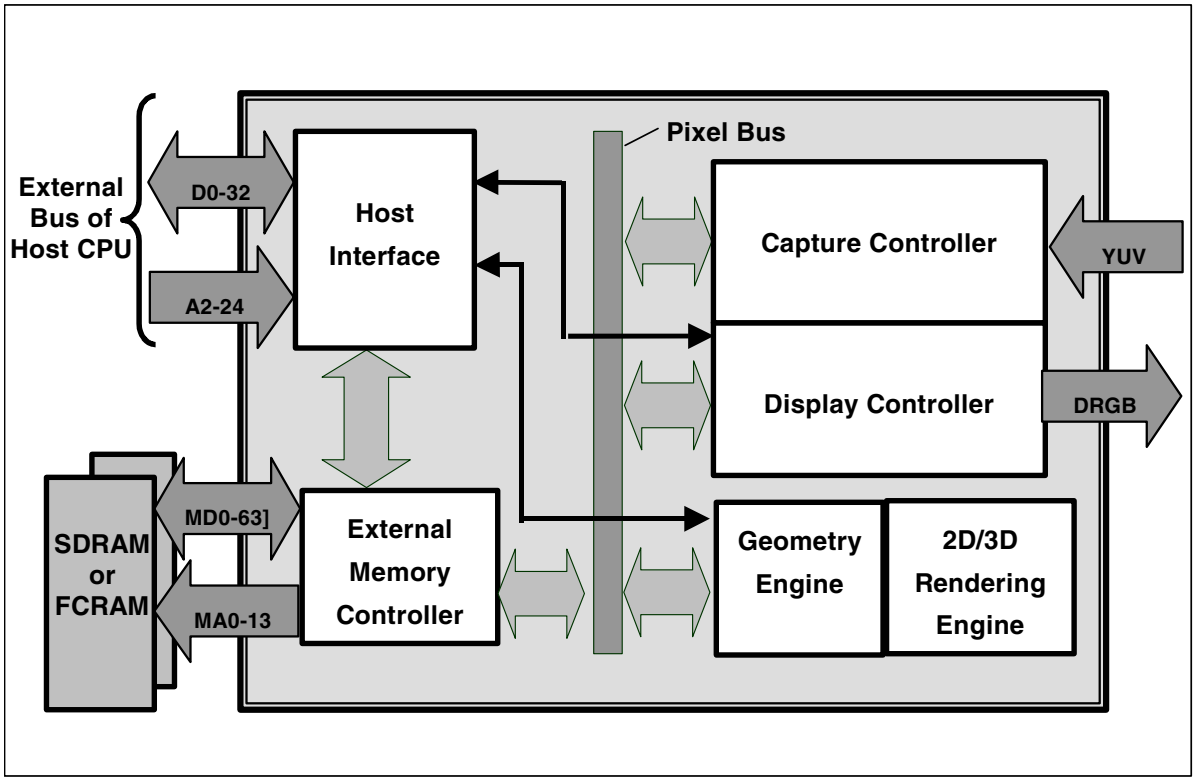


Fig.1.2 ORCHID Block Diagram (TBD)

1.5 Functional Overview

1.5.1 System Configuration

Host CPU interface

The MB86292can be connected to Hitachi's SH3 or SH4 CPUs, NEC's V832 microprocessor and Fujitsu SPARClite (MB86833) without an external circuit. The host MB86292 CPU interface transfer display list, texture pattern data from the main memory to the ORCHID graphic memory or internal registers by using an external DMA controller.

Graphics memory

Synchronous DRAM is attached externally. Either the 32-bit or 64-bit mode is supported as the interface with these external SDRAM devices. The external SDRAM operation frequency is the same as MB86292 (up to 100MHz). Applicable memory devise configurations are as follows.

External memory configuration

Type	Width of data bus	Number	Capacity	MMR set value
FCRAM 16MBit (x16Bit)	32Bit	2	4MByte	0x009a9842
FCRAM 16MBit (x16Bit)	64Bit	4	8MByte	0x009a984a
SDRAM 64MBit (x32Bit)	32Bit	1	8MByte	0x01477853
SDRAM 64MBit (x32Bit)	64Bit	2	16MByte	0x0147785b
SDRAM 64MBit (x16Bit)	32Bit	2	16MByte	0x014fba63
SDRAM 64MBit (x16Bit)	64Bit	4	32MByte	0x014fba6b
SDRAM 128MBit(x32Bit)	32Bit	1	16MByte	0x01477863
SDRAM 128MBit(x32Bit)	64Bit	2	32MByte	0x0147786b
SDRAM 128MBit(x16Bit)	32Bit	2	32MByte	0x01cfb9e3

Graphics output

The MB86292 has normally 5-bit RGB digital video graphics output pins each of which can be connected to an external digital video encoder, etc.. The MB86292 also has 8-bit RGB digital video graphics output pins when it was selected 32-bit mode as the interface with the external SDRAM devices.

Video input pin

The MB86292 has an 8-bit video input pin that can be connected to an external digital video decoder thereby supporting input of digital video in accordance with ITU RBT-656.

1.5.2 Display Controller

Screen resolution

Various resolutions are achieved by using a programmable timing generator as follows:

Screen Resolutions

Resolution
1024 × 768
1024 × 600
800 × 600
854 × 480
640 × 480
480 ×234
400 × 234
320 × 234

Display colors

There are two pixel color modes (indirect and direct). In the indirect mode, each pixel is expressed in 8-bit code. The actual display color is referenced using a color look-up table (color pallet). In this mode, each color of the look-up table is represented as 17 bits (RGB 6 bits each and independent alpha-blend bit), and 256 colors are selected from 262,144 colors. In the direct mode, each pixel is expressed as 16-bit code (RGB 5 bits each and reserved intensity bit). In this mode, 32,768 colors can be displayed.

Overlay

Up to three extra layers can be overlaid on the base window. When multiple layers are overlaid, the lower layer image can be displayed according to the setting of the transparency option. Any codes in the color pallet can be assigned a transparent color. Code 0 in the indirect mode or color value 0 in the direct mode sets this transparent option.

Hardware cursor

MB86292 supports two separate hardware cursor functions. Each of these hardware cursors is specified as a 64 × 64-pixel area. Each pixel of these hardware cursors is 8 bits and uses the indirect mode look-up table.

1.5.3 Frame Control

Double buffer scheme

This mode provides smooth animation. The display frame and drawing frame are switched back and forth at each scan frame. A program in the vertical blanking period controls flipping.

Scroll scheme

Wrap around scrolling can be done by setting the drawing area, display area, display size and start address independently.

Windows display

The whole screen can be split into two vertically separate windows. Both windows can be controlled independently.

1.5.4 Video Capture

The video capture function captures ITU RBT-656 format videos. Video data is stored in graphics memory once and then displayed on the screen in synchronization with the display scan.

Both NTSC and PAL video formats are supported.

1.5.5 Geometry Processing

The MB86292 has a geometry engine for performing the numerical operations required for graphics processing. The geometry engine uses the floating-point format to perform high-precision numerical operations. It selects the required geometry processing according to the set drawing mode and primitive type and executes processing to the final drawing.

MVP Transformation

MVP Transformation

Setting a 4 x 4 conversion matrix enables transformation of a 3D model view projection. Two-dimensional affine transformation is also possible.

Clipping

Clipping stops drawing of figures outside the screen (field of view). Polygons (including concave shapes) can also be clipped.

3D-2D Transformation

This functions transforms 3D coordinates (normalization) into 2D coordinates in orthogonal or perspective projections.

View port transformation

This function transforms normalized 2D coordinates into drawing (device) coordinates.

Primitive setup

This function automatically performs a variety of slope computations, etc., based on transforming vertex data into coordinates and sets up (preprocesses) rendering.

1.5.6 2D Drawing

2D Primitives

MB86292 provides automatic drawing of various primitives and patterns (drawing surfaces) to frame memory in either indirect color (8 bits/pixel referencing appropriate palette) or direct color (16 bits/pixel) mode. Alpha blending and anti-aliasing features are useful when the direct color mode is selected.

A triangle is drawn in a single color, mapped with a style image formed by a single color or 2D pattern (tiling), or mapped with a texture pattern by designating coordinates of the 2D pattern at each vertex (texture mapping). Alpha blending can be applied either per entire shape in single color mode or per pixel in tiling/texture mapping mode. When an object is drawn in single color or

filled with a 2D pattern (without using Gouraud shading or texture mapping), dedicated primitives, such as Fast2DLine and Fast2DTriangle, are used. Only vertex coordinates are set for these primitives. Fast2Dtriangle is also used to draw polygons.

2D Primitives

Primitive type	Description
Point	Plots point
Line	Draws line
Triangle	Draws triangle
Fast2DLine	Draws lines The number of parameters set for this primitive is less than that for Line. The CPU load to use this primitive is lighter than using Line.
Fast2DTriangle	Draws triangles. When a triangle is drawn in one color or filled with a 2D pattern, the CPU load to apply this primitive is lighter than using Triangle.

Polygon draw

This function draws various random shapes formed using multiple vertices. There is no restriction on the number of vertices number, however, if any sides forming the random shape cross each other, the shape is unsupported. The Polygon draw flag buffer must be defined in graphics memory as a work field to draw random shapes.

BLT/Rectangle fill

This function draws a rectangle using logical calculations. It is used to clear the frame memory and Z buffer. At scrolling, the rolled over part can be cleared by using this function in the blanking time period.

BLT Attributes

Attribute	Description
Raster operation	Selects two source logical operation mode
Transparent processing	This functions does not draw the pixel matching the transparent color.

Pattern (Text) drawing

This function draws a binary pattern (text) in a designated color.

Pattern (Text) Drawing Attributes

Attribute	Description
Enlarge	2×2 Horizontally $\times 2$
Shrink	Horizontally 1/2 $1/2 \times 1/2$

Clipping

This function sets a rectangular window in a frame memory drawing surface and disables drawing of anything outside that window.

1.5.7 3D Drawing

3D Primitives

This function draws 3D objects in frame memory in the direct color mode.

3D Primitives

Primitive	Description
Point	Plots 3D point
Line	Draws 3D line
Triangle	Draws 3D triangle

3D Drawing attributes

MB86292 has various professional 3D graphics features, including Gouraud shading and texture mapping with bi-linear filtering/automatic perspective correction, and provides high- quality realistic 3D drawing. A built-in sophisticated texture mapping unit delivers fast pixel calculations. This unit also delivers color blending between the shading color and texture color as well as alpha blending per pixel.

Hidden surface management

MB86292 supports the Z buffer for hidden surface management.

1.5.8 Special Effects

Anti-aliasing

Anti-aliasing manipulates lines and borders of polygons in sub-pixel units to eliminate jaggies on bias lines. It is used as a functional option for 2D drawing (in direct color mode only).

Line drawing

This function draws lines of a specific width. Detecting a line pattern can also draw a broken line. The anti-aliasing feature is also useful to draw smooth lines.

Line Draw Attributes

Attribute	Description
Width	Selectable from 1 to 32 pixels
Broken line	Set by 32 bit or 24 bit of broken line pattern

Alpha blending

Alpha blending blends two separate colors to provide a transparency effect. ORCHID supports two types of alpha blending; blending two different colors at drawing, and blending overlay planes at display. Transparent color is not used for these blending options.

Alpha Blending

Type	Description
Drawing	Transparent ratio set in particular register
	While one primitive (polygon, pattern, etc.), being drawn, registered transparent ratio applied
Overlay display	Blends top layer pixel color and lower layer pixel at same position
	Transparent ratio set in particular register
	Registered transparent ratio applied during one frame scan

Shading

Gouraud shading is supported in the direct color mode to provide realistic 3D objects and color gradation.

Texture mapping

MB86292 supports texture mapping to map a style pattern onto the surface of 3D polygons. Perspective correction is calculated automatically. For 2D pattern texture mapping, MB86292 has a built-in buffer memory for a field of up to 64 × 64 pixels. Texture mapping is performed at high speeds while texture patterns are stored in this buffer. The texture pattern can also be stored in the graphics memory. In this case, a large pattern of up to 256 × 256 pixels can be used.

Texture Mapping

Function	Description
Texture filtering	Point sample Bi-linear filter
Texture coordinate correction	Linear Perspective
Texture blending	Decal Modulate Stencil
Texture alpha blending	Normal Stencil Stencil alpha
Texture wrap	Repeat Cramp Border

1.5.9 Display List

ORCHID is operated by feeding display lists which consists of a set of display commands, arguments and pattern data for them. Normally, these display lists are stored either in off- screen frame memory (part of ORCHID's local buffer) or host (main) memory that the DMAC of the host CPU can access directly. ORCHID reads these display lists, decodes the commands, and executes them after reading all the necessary arguments. By executing this operation set until the end of the display list, all graphics operations, including image/object drawing and display control, are separated from the CPU. Of course, the CPU program can also feed the display list information directly to MB86292's designated registers.

2. Signal Pins

2.1 Signals

2.1.1 Signals

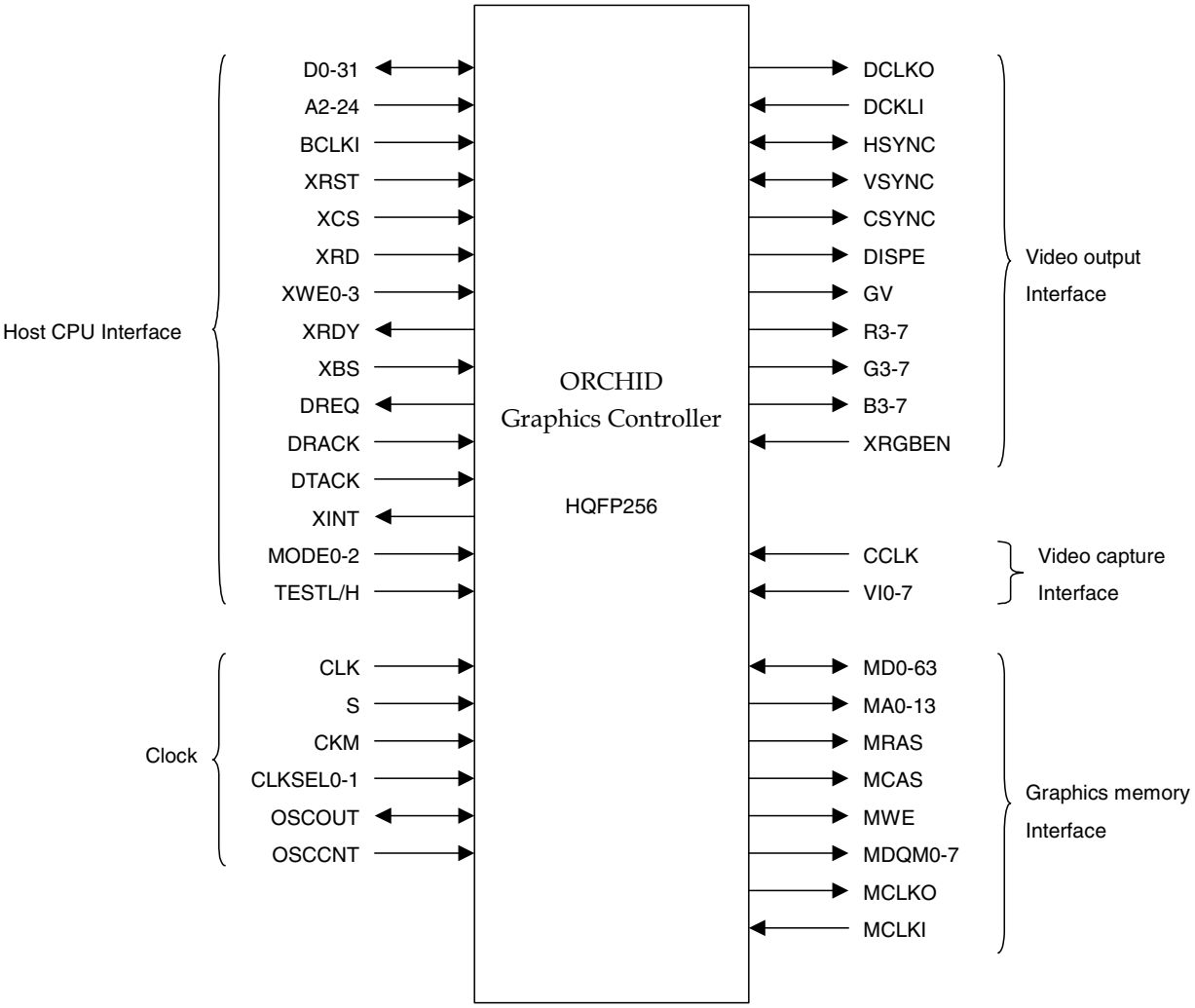


Fig. 2.1 ORCHID Signals

2.2 Pin Assignment

2.2.1 Pin Assignment Diagram

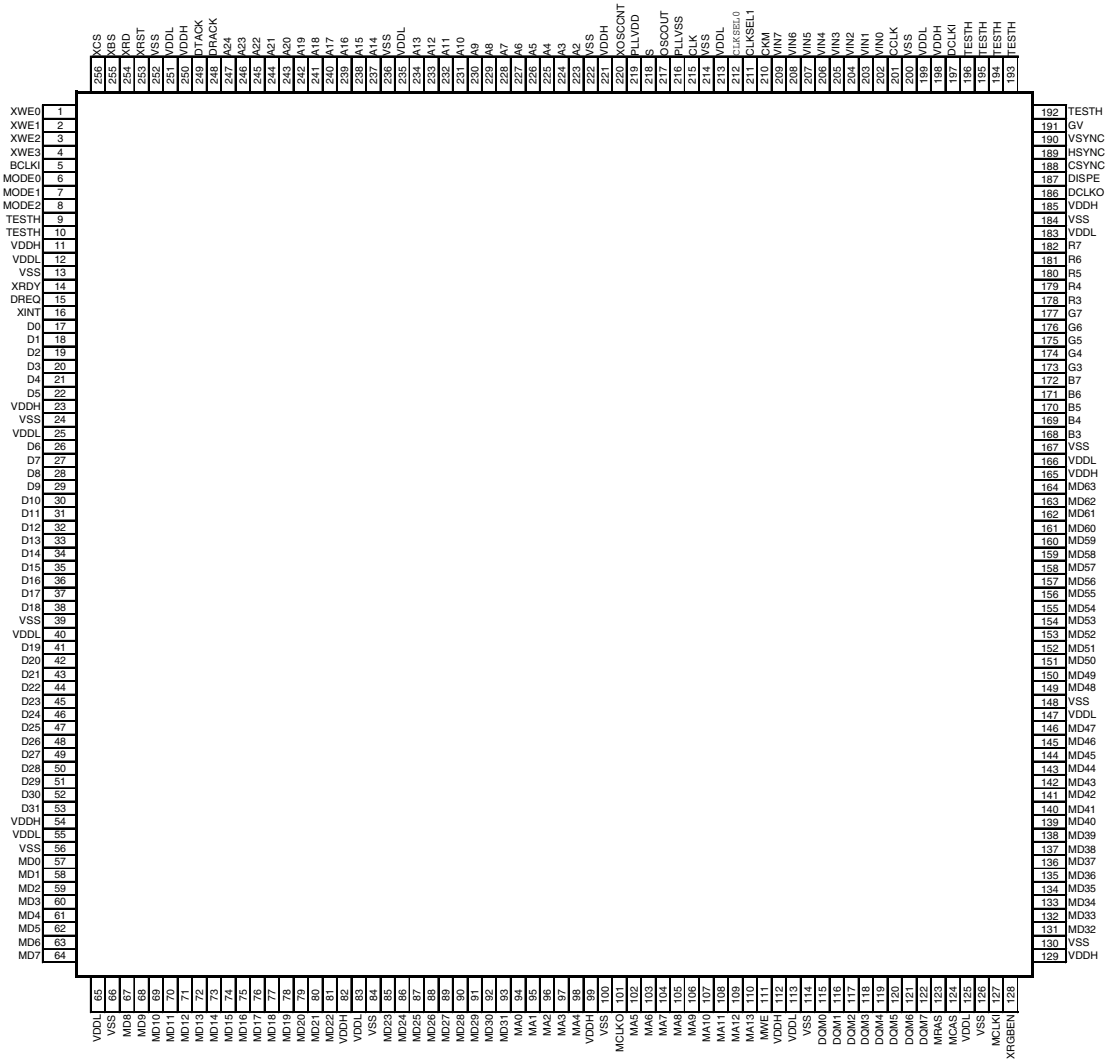


Fig. 2.2 ORCHID Pin Assignment

2.2.2 Pin Assignment Table

No.	Name	No.	Name	No.	Name	No.	Name
1	XWE0	65	VDDL	129	VDDH	193	TESTH
2	XWE1	66	VSS	130	VSS	194	TESTH
3	XWE2	67	MD8	131	MD32	195	TESTH
4	XWE3	68	MD9	132	MD33	196	TESTH
5	BCLKI	69	MD10	133	MD34	197	DCLKI
6	MODE0	70	MD11	134	MD35	198	VDDH
7	MODE1	71	MD12	135	MD36	199	VDDL
8	MODE2	72	MD13	136	MD37	200	VSS
9	TESTH	73	MD14	137	MD38	201	CCLK
10	TESTH	74	MD15	138	MD39	202	VIN0
11	VDDH	75	MD16	139	MD40	203	VIN1
12	VDDL	76	MD17	140	MD41	204	VIN2
13	VSS	77	MD18	141	MD42	205	VIN3
14	XRDY	78	MD19	142	MD43	206	VIN4
15	DREQ	79	MD20	143	MD44	207	VIN5
16	XINT	80	MD21	144	MD45	208	VIN6
17	D0	81	MD22	145	MD46	209	VIN7
18	D1	82	VDDH	146	MD47	210	CKM
19	D2	83	VDDL	147	VDDL	211	CLKSEL1
20	D3	84	VSS	148	VSS	212	CLKSEL0
21	D4	85	MD23	149	MD48	213	VDDI
22	D5	86	MD24	150	MD49	214	VSS
23	VDDH	87	MD25	151	MD50	215	CLK
24	VSS	88	MD26	152	MD51	216	PLLVSS
25	VDDI	89	MD27	153	MD52	217	OSCOUT
26	D6	90	MD28	154	MD53	218	S
27	D7	91	MD29	155	MD54	219	PLLVDD
28	D8	92	MD30	156	MD55	220	XOSCCNT
29	D9	93	MD31	157	MD56	221	VDDH
30	D10	94	MA0	158	MD57	222	VSS
31	D11	95	MA1	159	MD58	223	A2
32	D12	96	MA2	160	MD59	224	A3
33	D13	97	MA3	161	MD60	225	A4
34	D14	98	MA4	162	MD61	226	A5
35	D15	99	VDDH	163	MD62	227	A6
36	D16	100	VSS	164	MD63	228	A7
37	D17	101	MCLKO	165	VDDH	229	A8
38	D18	102	MA5	166	VDDL	230	A9
39	VSS	103	MA6	167	VSS	231	A10
40	VDDI	104	MA7	168	B3	232	A11
41	D19	105	MA8	169	B4	233	A12
42	D20	106	MA9	170	B5	234	A13
43	D21	107	MA10	171	B6	235	VDDL
44	D22	108	MA11	172	B7	236	VSS
45	D23	109	MA12	173	G3	237	A14
46	D24	110	MA13	174	G4	238	A15
47	D25	111	MWE	175	G5	239	A16
48	D26	112	VDDH	176	G6	240	A17
49	D27	113	VDDI	177	G7	241	A18
50	D28	114	VSS	178	R3	242	A19
51	D29	115	DQM0	179	R4	243	A20
52	D30	116	DQM1	180	R5	244	A21
53	D31	117	DQM2	181	R6	245	A22
54	VDDH	118	DQM3	182	R7	246	A23
55	VDDL	119	DQM4	183	VDDL	247	A24
56	VSS	120	DQM5	184	VSS	248	DRACK
57	MD0	121	DQM6	185	VDDH	249	DTACK
58	MD1	122	DQM7	186	DCLKO	250	VDDH
59	MD2	123	MRAS	187	DISPE	251	VDDI
60	MD3	124	MCAS	188	CSYNC	252	VSS
61	MD4	125	VDDI	189	HSYNC	253	XRST
62	MD5	126	VSS	190	VSYNC	254	XRD
63	MD6	127	MCLKI	191	GV	255	XRS
64	MD7	128	XRGBEN	192	TESTH	256	XCS

Table 2.1 Pin Assignment

Notes

V_{SS}/$PLLV_{SS}$	: Ground
V_{DDE}	: 3.3-V power supply
V_{DDI}/$PLLV_{DD}$	: 2.5-V power supply
$PLLV_{DD}$	: PLL Power supply
OPEN	: NC
TESTH	: Input a high-level TESTH signal.

- $PLLV_{DD}$ should be isolated on the PCB.
- Insert a bypass capacitor with good high frequency characteristics between the power supply and ground.

  Position the capacitor as near as possible to the pin.

2.3 Signal Descriptions

2.3.1 Host CPU Interface

Table 2.2 Host CPU Interface Signals

Signal Name	I/O	Description
MODE0-2	Input	Host CPU mode, ready mode selection,
XRESET	Input	Hardware reset
D0-31	In/Out	Host CPU bus data
A2-A24	Input	Host CPU bus address (In the V832 mode, A[24] is connected to XMWR.)
BCLKI	Input	Host CPU bus clock
XBS	Input	Bus cycle start
XCS	Input	Chip select
XRD	Input	Read strobe
XWE0	Input	Write strobe for D0-D7
XWE1	Input	Write strobe for D8-D15
XWE2	Input	Write strobe for D16-D23
XWE3	Input	Write strobe for D24-D31
XRDY	Output Tri-state	Wait request signal (In the SH3 mode, when this signal is 0, it indicates the wait state; in the SH4, V832 and SPARClite modes, when this signal is 1, it indicates the wait state.)
DREQ	Output	DMA request signal (This signal is low-active in both the SH mode and V832 mode.)
DRACK/DMAAK	Input	Acknowledge signal issued in response to DMA request (DMAAK is used in the V832 mode; this signal is high-active in both the SH mode and V832 mode.)
DTACK/XTC	Input	DMA transfer strobe signal (XTC is used in the V832 mode. In the SH mode, this signal is high-active; in the V832 mode, it is low-active.)
XINT	Output	Interrupt signal issued to host CPU (In the SH mode, this signal is low-active; in the V832 mode, it is high-active)
TEST	Input	Test signals

- B86291 can be connected to the Hitachi SH4 (SH7750), SH3 (SH7709/09A) NEC V832 and Fujitsu SPARClite (MB86833). In the SRAM interface mode, MB86292 can be used with any other CPU as well. The host CPU is specified by the MODE0 and 1 pins.

MODE 1	MODE 0	CPU
L	L	SH3
L	H	SH4
H	L	V832
H	H	SPARClite

- When the bus cycle terminates, a ready signal level can be set by the MODE2 pin. When using the high-level MODE2 signal, set two cycles as the CPU software wait of the CPU.

MODE 2	Ready signal mode
L	When the bus cycle terminate, sets the XRDY signal to the 'not ready' level.
H	When the bus cycle terminates, sets the XRDY signal to the 'ready' level.

- The host interface data bus is 32-bits wide (fixed).
- The address bus is 24-bits wide (per double word), and has a 32-Mbyte address field. MB86290A uses a 32-Mbyte address field.
- The external bus frequency is up to 100 MHz.
- In the SH4 mode and V832 mode, when the XRDY signal is low, it is in the ready state. In the SH3 mode, when the XRDY signal is low, it is in the wait state.
- DMA data transfer is supported using an external DMAC.
- An interrupt request signal is generated to the host CPU.
- The XRESET input must be kept low (active) for at least 300 µs after setting the S (PLL reset) signal to high.
- TEST signals must be clamped to high level.
- In the V832 mode, MB86292 signals are connected to the V832 CPU as follows:

ORCHID Signal Pins	V832 Signal Pins
A24	XMWR
DTACK	XTC
DRACK	DMAAK

2.3.2 Video output Interface

Table 2.3 Video Output Interface Signals

Signal Name	I/O	Description
DCLKO	Output	Dot clock signal for display
DCLKI	Input	Dot clock signal input
HSYNC	I/O	Horizontal sync signal output Horizontal sync input in external sync mode
VSYNC	I/O	Vertical sync signal output Vertical sync input in external sync mode
CSYNC	Output	Composite sync signal output
DISPE	Output	Display valid period signal
GV	Output	Graphics/video switch
R3-7	Output	Degital signal (R) output
G3-7	Output	Degital signal (G) output
B3-7	Output	Degital signal (B) output
XRGBEN	Input	RGB2-0output / Memory bus (MD63-55) switch

- Contains 8-bit precision D/A converters and outputs analog RGB signals. It is also possible to connect each 8-bit RGB digital output pin to an external digital video encoder.
- External circuits to generate composite video signal
- Can display synchronously to external video signal
- Can synchronize to either DCLKI signal input or internal dot clock
- HSYNC and VSYNC reset to output mode. These signals must be pulled up externally.
- AOUTR, AOUTG and AOUTB must be terminated at 75 Ω.
- 1.1 V is input to VREF. A bypass capacitor (with good high-frequency characteristics) must be inserted between VREF and AVS.
- ACOMPR, ACOMPG and ACOMPB are tied to analog VDD via 0.1-μF ceramic capacitors.
- VRO must be pulled down to analog ground by a 2.7-kΩ resistor.
- When producing a non-interlaced display in the external synchronous mode, input 0 to the EO pin by using a pull-down resistor, etc.
- The GV signal switches graphics and video at chroma key operation. When video I is selected, the L level is output.

2.3.3 Video capture Interface

Table 2.4 Video capture Interface Signals

Signal Name	I/O	Description
CCLK	Input	Digital video input clock signal input
VI0-7	Input	Digital video data input

- Inputs ITU-RBT-656 format digital video signal
- Inputs ITU-RBT-656 format digital video signal

2.3.4 Graphic memory Interface

Graphic memory Interface

Name	I/O	Description
MD0-54	**I/O**	Graphics memory bus data
MD55-63/RGB	**I/O**	Graphics memory bus data or RGB0-2 output
MA0-13	**O**	Graphics memory bus data
MRAS	**O**	Low address strobe
MCAS	**O**	Column address strobe
MWE	**O**	Write enable
MDQM0-7	**O**	Data Mask
MCLKO	**O**	Graphics memory clock output
MCLKI	**I**	Graphics memory clock input

♦ These connect external memory used graphic data memory. It is possible to connect 128M bit SDRAM and 64M bit SDRAM data bus 16 bit and 2 bit without external circuit.

- Memory bus data is chosen between 64 bit and 32 bit. In case of 32 bit, if RGB 8 bit output(XRGBEN interface=0)MD32-MD63 and MDQM4-7, RGB 8 bit output(XRGBEN interface=0), MD32-MD54 and MDQM4-7 are opened.
- Connect between MCLKI and MCLKO.
- When XRGBEN is fixed 1, it is possible to use MD55-MD63 for graphic memory bus data. When XRGBEN is fixed 0, RGB0-2 is outputted.

2.3.5 Clock Input

Table 2.5 Clock Input Signals

	I/O	Description
CLK	Input	Clock input signal
S	Input	PLL reset signal
CKM	Input	Clock mode signal
CLKSEL [1:0]	Input	Clock rate selection signal
OSCOUT*1	I/O	Crystal oscillator connection pin (reserved)
OSCCNT*2	Input	Crystal oscillator selection pin (reserved)

*1 Do not connect anything.

*2 Input a high-level OSCCNT signal.

- Inputs source clock for generating internal operation clock and display dot clock. Normally, 4 Fsc(= 14.31818 MHz:NTSC) is input. An internal PLL generates the internal operation clock of 100 MHz and the display base clock of 200 MHz.
- For the internal operation clock, use either the output clock of the internal PLL (100 MH_Z) or the bus clock input (BCLK1) from the host CPU. When the host CPU bus speed is 100 MHz, the BCLK1 input should be selected.

CKM	Clock mode
L	Output from internal PLL selected
H	Host CPU bus clock (BCLK1) selected

- When CKM = L, selects input clock frequency when built-in PLL used according to setting of CLKSEL pins

CLKSEL1	CLKSEL1	Clock frequency
L	L	Inputs 13.5-MHz clock frequency
L	H	Inputs 14.32-MHz clock frequency
H	L	Inputs 17.73-MHz clock frequency
H	H	Reserved

- At power-on, a low-level signal must be input to the S-signal pin for more than 500 ns and then set to high. After the S-signal input is set to high, a low-level signal must be input to XRESET for another 300 µs.

3. Host Interface

3.1 Operation Mode

3.1.1 Host CPU Mode

Select the host CPU by setting the MODE0 to MODE1 signals as follows:

Table 3.1 CPU Type Setting

MODE1	MODE0	CPU Type
L	L	SH3
L	H	SH4
H	L	V832
H	H	SPARClite

3.1.2 Ready Signal Mode

The MODE2 pin can be used to set the ready signal level when the bus cycle of the host CPU terminates. When using this device in the normal ready mode, set two cycles as the CPU software wait. When using this device in the 'normal not ready' mode, set one cycle as the software wait.

The 'normal not ready mode' is the mode in which the ORCHID XRDY signal is always in the wait status and Ready is returned only when read/write is made ready.

The 'normal ready mode' is the mode in which the ORCHID XRDY signal is always in the Ready status and it is put into the wait status only when read/write cannot be performed immediately.

Table 3.2 Ready Signal Mode

MODE 2	Ready signal operation
L	Recognizes XRDY signal as 'not ready level' and terminates bus cycle (normal not ready mode)
H	Recognizes XRDY signal as 'ready level' and terminates bus cycle (normal ready mode)

3.1.3 Endian

ORCHID operates in little-endian mode. All the register address descriptions in these specifications are byte address in little endian. When using a big-endian CPU, note that the byte or word addresses are different from these descriptions.

3.2 Access Mode

3.2.1 SRAM Interface

Data can be transferred to/from ORCHID using a typical SRAM access protocol. ORCHID internal registers, internal memory and external memory are all mapped to the physical address field of the host CPU.

ORCHID uses hardware wait based on the XRDY signal; enable the hardware wait setting of the host CPU. When using the 'normal not ready mode,' set the software wait to 1. When using the 'normal ready mode,' set the software wait to 2. Switch the ready mode using the MODE[2] signal.

CPU Read

The host CPU reads data from internal registers and memory of ORCHID in double-word (32 bit) units. Valid data is output continuously while XRD and XCS are being asserted at a low level after XRDY has been asserted.

CPU Write

The host CPU writes data to internal registers and memory of ORCHID in byte units.

3.2.2 FIFO Interface

This interface transfers display lists in host memory. Display list information is transferred efficiently by using a single address mode DMA operation. This FIFO is mapped to the physical address field of the host CPU so that the same data transfer can be performed in either the SRAM mode or dual address DMA mode by specifying the FIFO in the destination address.

3.3 DMA Transfer

3.3.1 Data Transfer Unit

DMA transfer is performed in double-word (32 bit) units or 8 double-word (32 Byte) units. Byte and word access is not supported.

Note: 8 double-word transfer is supported only in the SH4 mode.

3.3.2 Address Mode

Dual address mode

DMA is performed at memory-to-memory transfer between host memory (source) and MB86292 internal registers, memory, or external memory (destination). Both the host memory address and ORCHID is used. In the SH4 mode, the 1 double-word transfer (32 bits) and 8 double-word transfer (32 bytes) can be used.

When the CPU transfer destination address is fixed, data can also be transferred to the FIFO interface. However, in this case, even the SH4 mode supports only the 1 double-word transfer.

DREQ and DRACK pins and SRAM interface signals are used. In V832, the DREQ, DMAAK, and XTC pins and SRAM interface signals are used.

Note: The SH3 mode supports the direct address mode; it does not support the indirect address mode.

Single address mode (FIFO interface)

DMA is performed between host memory (source) and FIFO (destination). Address output from the host CPU is only applied to designate the source, and the data output from the host memory is transferred to the FIFO using the DACK signal. In this mode, data read from the host memory and data write to the FIFO occur in the same bus cycle. This mode does not support data write to the host memory. When the FIFO is full, the DREQ signal is tentatively negated and the DMA transfer is suspended until the FIFO has room for more data.

The 1 double-word transfer (32 bits) and the 8 double-word transfer (32 Bytes) can be used.

DREQ, DTACK, and DRACK signal pins are used.

Note: The single-address mode is supported only in the SH4 mode.

3.3.3 Bus Mode

MB86290A supports the DMA transfer cycle steal mode and burst mode. Either mode is selected by setting to the external DMA mode.

Cycle steal mode (In the V832 mode, the burst mode is called the single transfer mode.)

In the cycle steal mode, the bus right is transferred back to the host CPU at every DMA transaction unit. The DMA transaction unit is either 1 double-word (32 bits) or 8 double-words (32 B).

Burst mode (In the V832 mode, the burst mode is called the demand transfer mode.)

When DMA transfer is started, the right to use the bus is acquired and the transfer begins. The data transfer unit can be selected from between the 1 double word (32 bits) and 8 double words (32 B).

Note: When performing DMA transfer in the dual-address mode, a function for automatically negating DREQ is provided based on the setting of the DBM register.

3.3.4 DMA Transfer Request

- Single-address mode

  DMA is started when the ORCHID issues an external request to DMAC of the host processor.

  Set the transfer count in the transfer count register of the ORCHID and then issue DREQ.

  Fix the CPU destination address to the FIFO address.

- Dual-address mode

  DMA is started by two procedures: ORCHID issues an external request to DMAC of the host processor, or the CPU itself is started (auto request mode, etc.). Set the transfer count in the transfer count register of ORCHID and then issue DREQ.

  Note: The V832 mode requires no setting of the transfer count register.

3.3.5 Ending DMA Transfer

- SH3/SH4

  When the ORCHID transfer count register is set to 0, DMA transfer ends and DREQ is negated.

- V832

  When the XTC signal from the CPU is low-asserted while the DMAAK signal to ORCHID is high-asserted, the end of DMA transfer is recognized and DREQ is negated.

- The end of DMA transfer is detected in two ways: the DMA status register (DST) is polled, and an interrupt to end the drawing command (FD000000H) is added to the display list and the interrupt is detected.

3.4 Interrupt Request

MB86292 issues interrupt requests to the host CPU. The following events issue interrupt requests. An interrupt request caused by each of these events is enabled/disabled independently by IMR (Interrupt Mask Register).

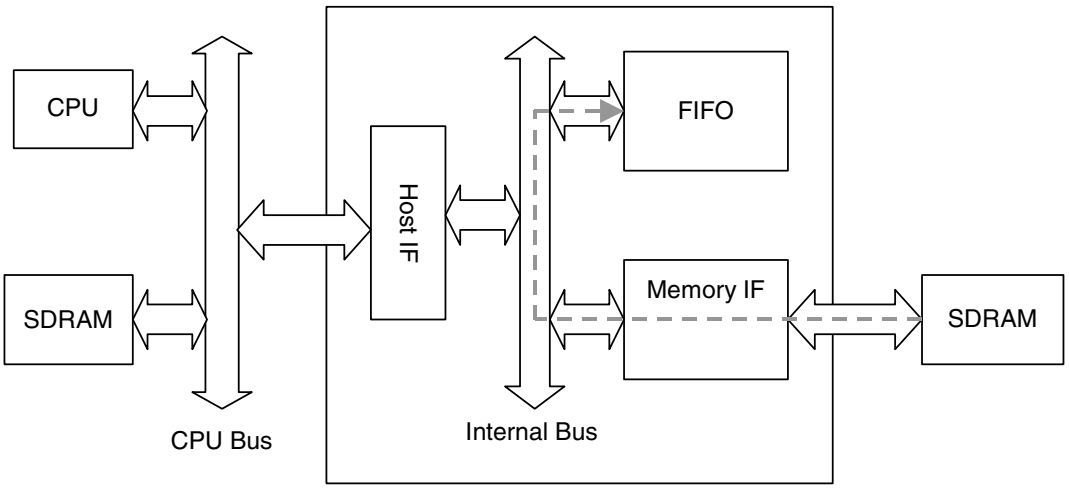
- Vertical synchronization timing detect
- Field synchronization timing detect
- External synchronization error detect
- Command error
- Command complete

3.5 Transfer of Local Display List

This is the mode in which the ORCHID internal bus is used to transfer the display list stored in the graphics memory to the FIFO interface.

During transfer of the local display list, the host bus can be used to perform read/write for the CPU.

How to transfer list: Store the display list in the local memory of ORCHID, set the transfer source local address (LSA) and the transfer count (LCO), and then issue a request (LREQ). Whether or not the local display list is currently being transferred is checked using the local transfer status register (LSTA).



Transfer Path for Local Display List

3.6 Memory Map

The following table shows the memory map of ORCHID to the host CPU address field. The physical address is mapped differently in each CPU type (SH3, SH4 or V832).

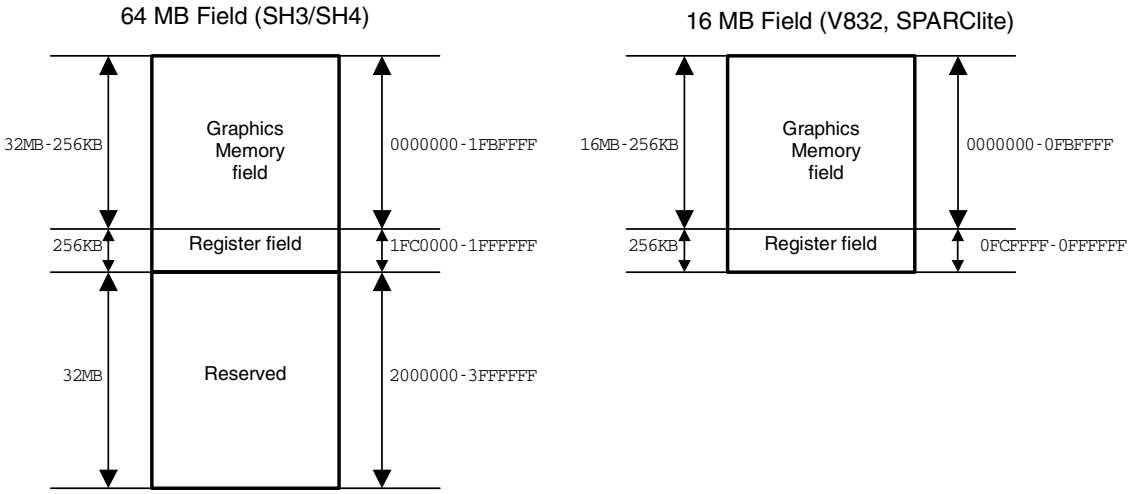


Fig. 3.1 Memory Map

Table 3-3 Address Mapping in SH3/SH4 Mode

Size	Resource	Base address	(Name)
2 MB	Graphics memory	00000000	
64 KB	Host interface registers	01FC0000	(HostBase)
32 KB	Display engine registers	01FD0000	(DisplayBase)
32 KB	Video capture registers	01FD8000	(CaptureBase)
64 KB	Internal texture memory	01FE0000	(TextureBase)
32 KB	Drawing engine registers	01FF0000	(DrawBase)
32 KB	Geometry engine registers	01FF8000	(GeometryBase)
32 MB	Reserved *	02000000	

The memory contents of 00000000-01FFFFFF are duplicated in this reserved field.

Table 3-4 Address Mapping in V832, SPARClite Mode

Size	Resource	Base address	(Name)
2 MB	Graphics memory	00000000	
64 KB	Host interface registers	00FC0000	(HostBase)
32 KB	Display engine registers	00FD0000	(DisplayBase)
32 KB	Video capture registers	00FD8000	(CaptureBase)
64 KB	Internal texture memory	00FE0000	(TextureBase)
32 KB	Drawing engine registers	00FF0000	(DrawBase)
32 KB	Geometry engine registers	00FF8000	(GeometryBase)

4. Graphics Memory

4.1 Configuration

ORCHID uses local external memory (Graphics Memory) for drawing and display management. The configuration of this Graphics Memory is described as follows:

4.1.1 Data Type

ORCHID handles the following types of data. Display list can be stored in the host (main) memory as well. Texture-tiling pattern and text pattern can be defined by a display list as well.

Drawing frame

This is a rectangular image data field for 2D/3D drawing. Two or more drawing frames can be used at once. The frame size can be bigger than the display frame size and display part of it. The drawing frame can be applied in 32-pixel units (both horizontally and vertically), and the maximum size is 4096 × 4096. Both direct and indirect color modes can be used.

Display frame

This is a rectangular image data field for display. Up to four layers (three of graphics and one of video/graphics) can be overlaid and displayed at once. From bottom to the top, these are called the B (Base), M (Middle), W (Window), and C (Console) layers.

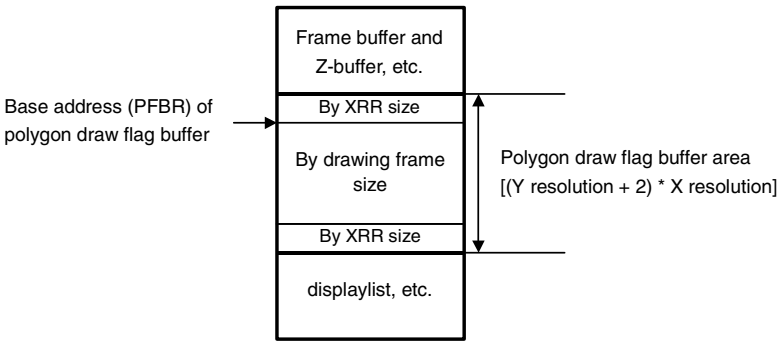
Z buffer

The Z buffer eliminates hidden surfaces in 3D drawing. The configuration is the same as drawing frame (defined for 3D drawing). 2 bytes/pixel of memory resources must be assigned. The Z buffer must be cleared prior to 3D drawing.

Polygon draw flag buffer

This is a work field for random shape drawing of multiple vertices. Allocate a 1-bit memory area with the same shape as the drawing frame per pixel and then an area of X resolution before and after the memory area.

This flag buffer must be cleared prior to drawing.



Display list

This is a set of commands and parameters executed by ORCHID.

Texture pattern

This is pattern data for texture mapping. The 16-bit direct color mode must be used for texture pattern. The maximum size of this pattern is 256 × 256 pixels. The texture pattern is referenced from either graphics memory or internal texture buffer.

Cursor pattern

This is the pattern data for hardware cursors. Each pixel is described in 8-bit indirect color mode. Two sets of 64 × 64-pixel patterns can be used.

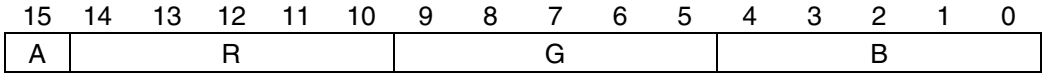
4.1.2 Memory mapping

Graphics memory is mapped linearly to host CPU address field. Each of these data can be allocated anywhere in the Graphics Memory according to the respective register setting (However, depending on the type of data, data boundaries may be restricted).

4.1.3 Memory Data Format

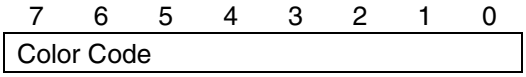
Direct color

Color data is described in 15-bit RGB (RGB 5 bits, respectively). Bit 15 is used as the alpha bit when producing a semi-transparent display for the C layer. For other layers, set bit 15 to 0.



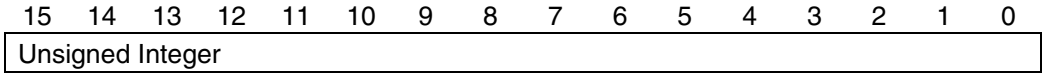
Indirect color

The color index code is in 8 bits.



Z value

This unsigned integer data describes the Zvalue in a 3D coordinate.



Polygon draw flag

This is binary data describing each pixel in 1 bit.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P31	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16

Texture/tiling pattern (direct color)

This is color data described in the direct color mode (RGB 5 bits, respectively). The MSB is an alpha bit used for the transparency effect of alpha blending.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
A	R					G					B				

Tiling pattern (indirect color)

This is a color index code in 8 bits.

7	6	5	4	3	2	1	0
Color Code							

Cursor pattern

This is a color index code in 8 bits.

7	6	5	4	3	2	1	0
Color Code							

Video graphics data

16-bit video graphics data per pixel is stored in memory in 4:2:2 YCbCr format.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y0								Cb							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Y1								Cr							

4.2 Frame Management

4.2.1 Single Buffer

The entire or partial area of the drawing frame is assigned as a display frame. The display field is scrolled by relocating the position of the display frame. When the display frame crosses the border of the drawing frame, the other side of the drawing frame is displayed, assuming that the drawing frame is rolled over (top and left edges assumed logically connected to bottom and right edges, respectively). To avoid the affect of drawing on display, the drawing data can be transferred to the Graphics Memory in the blanking time period.

4.2.2 Double Buffer

Two drawing frames are set. While one frame is displayed, drawing is done at the other frame. Flicker-less animation can be performed by flipping these two frames back and forth. Flipping is done in the blanking time period. There are two flipping modes: automatically at every scan frame period, and by user control. The double buffer is assigned independently for the Base and Middle layers. When the screen partition mode is selected (so that both Base and Middle layers split into separate left and right windows), the double buffer can be assigned independently for left and right windows.

4.3 Memory Access

4.3.1 Memory Access by Host CPU

The Graphics Memory is mapped to the host CPU physical address field. The host CPU can access the Graphics Memory of ORCHID like a typical memory device.

4.3.2 Priority of Memory Access

The Graphics Memory accesses priority is as follows:

1. Refresh
2. Display
3. Video Capture
4. Host CPU Access
5. Drawing

5. Display Controller

5.1 Overview

Display control

Overlay of four display layers, screen partition, scroll, etc., is applicable.

Video timing generator

The video display timing is generated according to the display resolution (from 320 × 240 to 1024 × 768).

Color look-up

There are two sets of color look-up tables (pallet RAM) for the indirect color mode (8 bits/pixel).

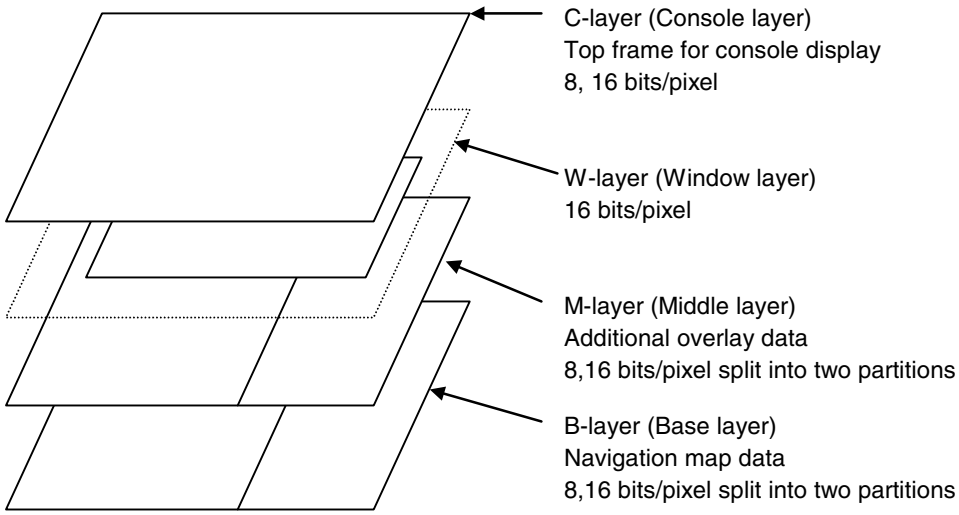
Cursor

Two sets of hardware cursor patterns (8 bits/pixel, 64 × 64 pixels each) can be used.

5.2 Display Function

5.2.1 Layer Configuration

MB86292 supports four layers of display frames (C, W, M and B). Furthermore, the M and B layers can be split into two separate windows at any position (L frame and R frame). All these six frames are assigned as logically separated fields in the Graphics Memory.



Configuration of Display Layers

However, if high resolutions are displayed, the number of layers that can be displayed simultaneously and pixel data may be restricted according to the graphics memory ability to supply data.

5.2.2 Overlay

Simple priority mode

The top layer has the higher priority. Each pixel color is determined according to the following rules:

1. If the C layer is not transparent, the C layer color is displayed.
2. If the C layer is transparent and W layer image is at that position, the W layer color is displayed.
3. If the C layer is transparent and there is no W layer image at that position, and if the M layer color is not transparent, the M layer color is displayed.
4. If the C and M layers are transparent and there is no W layer image at that position, the B layer color is displayed.

Transparent color is set by putting a specific transparent color code in the register.

Blend mode

The W, M and B layers are managed in the same way as the simple priority mode described above. The result of the W/M/B layer priority color is blended with the C layer color according to the blending ratio specified in the register. This mode is applied when the alpha bit of that pixel in the C layer is 1. If this alpha bit is set to 0, the result is the same as the simple priority mode.

When the C layer display priority is cursor display, the cursor color and C layer color are alpha blended at the pixel position with alpha bit = 1. The alpha blend ratio is calculated as follows:

- When BRS bit of BRATIO register = 0
  Display color = ((C layer color x blend coefficient) +
  (Mixed color of W/M/B layers x (1-blend coefficient))
- When BRS bit of BRATIO register = 1
  Display color = (C layer color x (1-blend coefficient)) +
  (Mixed color of W/M/B layers x blend coefficient)

5.2.3 Display Parameters

The display field is specified according to the following parameters. Each parameter is set independently at the respective register.

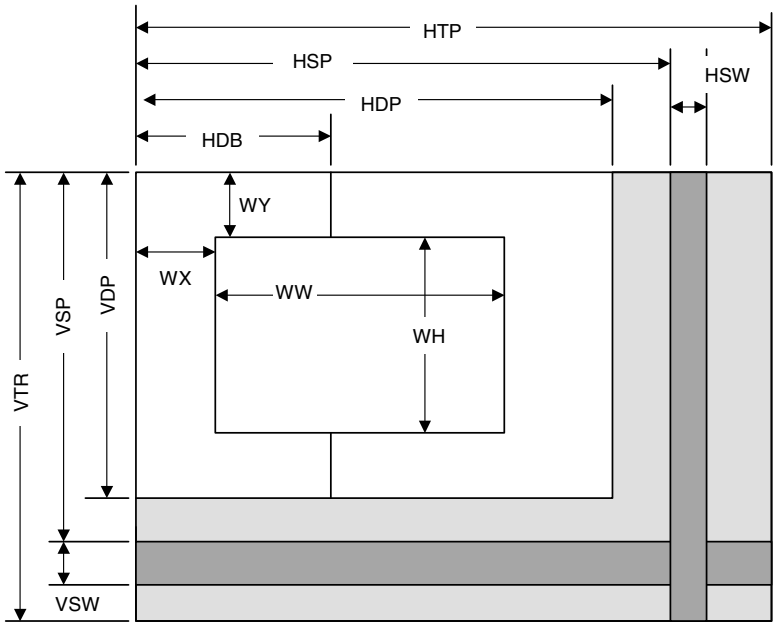


Fig. 5.1 Display Parameters

HTP	Horizontal Total Pixels
HSP	Horizontal Synchronize pulse Position
HSW	Horizontal Synchronize pulse Width
HDP	Horizontal Display Period
HDB	Horizontal Display Boundary
VTR	Vertical Total Raster
VSP	Vertical Synchronize pulse Position
VSW	Vertical Synchronize pulse Width
VDP	Vertical Display Period
WX	Window position X
WY	Window position Y
WW	Window Width
WH	Window Height

When not splitting the screen, set HDP to HDB and display only the left side of the screen. The settings must meet the following size relationship:

$0 < HDB \leq HDP < HSP < HSP + HSW + 1 < HTP$

$0 < VDP < VSP < VSP \quad VSW + 1 < VTR$

5.2.4 Display Position Control

The graphic image data to be displayed is located in the logical 2D coordinate area (logical graphics field) in the Graphics Memory. There are six logical graphics fields as follows:

- C layer
- W layer
- ML layer (left field of M layer)
- MR layer (right field of M layer)
- BL layer (left field of B layer)
- MR layer (right field of M layer)

  The correlation between the logical graphics field and physical display position is defined as follows:

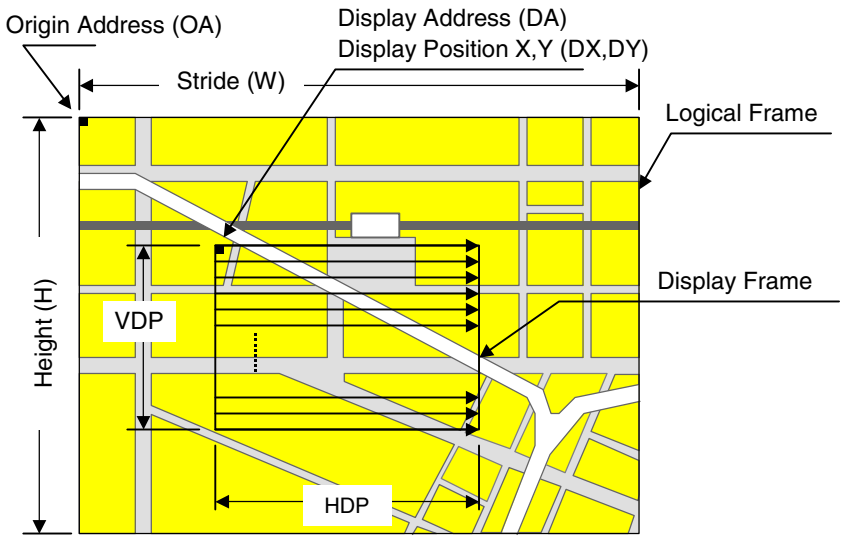


Fig. 5.2 Display Position Parameters

OA	Origin Address	Origin address of logical graphics field. Memory address of top left edge pixel in logical frame origin
W	Width	Width of logical graphics field. Defined in 64-byte unit
H	Height	Height of logical graphics field. Total raster (pixel) count of field
DA	Display Address	Display origin address. Top left position address of display frame origin
DX DY	Display Position	Display origin coordinate. Coordinate in logical frame field of display frame origin

MB86292 scans the logical graphics field as if the entire field is rolled over in both the horizontal and vertical directions. By using this function, if the display frame crosses the border of the logical graphics field, the part outside the border is covered with the other side of the logical graphics field, which is assumed to be connected cyclically as shown below:

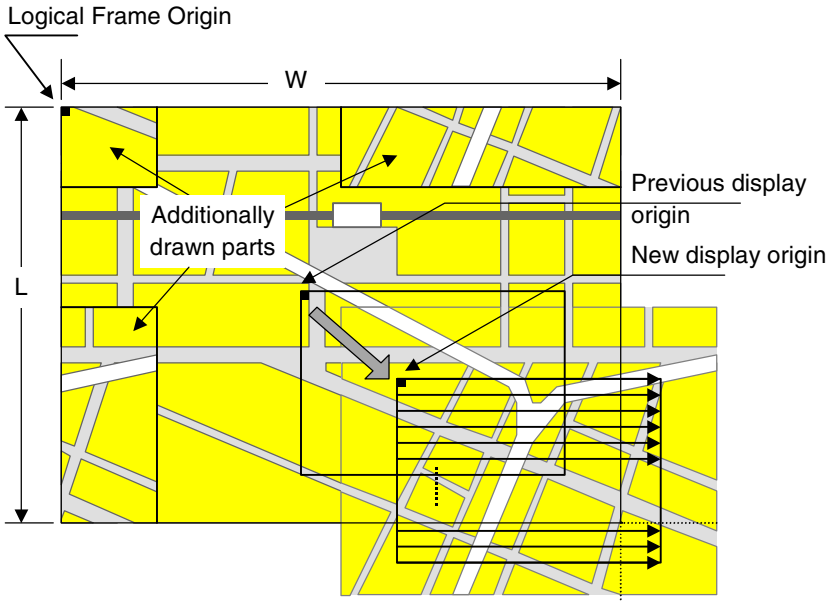


Fig. 5.3 Wrap Around Management of Display Frame

The relational expression of the X- and Y-coordinates in the frame and their corresponding linear addresses (in bytes) is shown below.

$A(x,y) = x \times bpp/8 + 64wy$ (bpp = 8 or 16)

The origin of the displayed coordinates must be within the frame. To be more specific, the parameters are subject to the following constraints:

$0 \leq DX < w \times 64 \times 8/bpp$ (bpp = 8 or 16)

$0 \leq DY < H$

DX, DY, and DA must indicate the same point within the frame. In other words, the following relationship must be established.

$DA = OA + DX \times bpp/8 + 64w \times DY$ (bpp = 8 or 16)

5.3 Display Color

Either direct color mode (16 bits/pixel) or indirect color mode (8 bits/pixel) can be used for the C, M, and B layers. Only the direct color mode can be used for the W layer.

5.3.1 Color Look-up Table

MB86292 has two color look-up tables (pallets) for the indirect color mode. Each pallet has 256 entries. A color data item contains 18 bits of data (RGB 6 bit, respectively), which is correlated to each color code specified in 8-bit data. Therefore, each pallet can show 256 colors at one time out of 262,144 color selections.

C layer palette

This pallet is dedicated to the C layer and hardware cursors. If the overlay blend mode is used, an alpha bit must be set at each color data. When this alpha bit is set to 1, color blending between the C layer pixel and W/M/B layer pixels is performed according to the priority order specified in the overlay section. This blending option cannot be used for the hardware cursor.

M/B-layer palette

This pallet is shared by the M and B layers. If both the M and B layers are set to the indirect color mode, they share this same color pallet.

5.4 Cursor

5.4.1 Cursor Display Function

ORCHID can display two hardware cursors simultaneously. Each cursor is specified as 64 x 64 pixels, and the style pattern is set in the Graphics Memory. Only the indirect color mode (8 bits/pixel) can be used and the C layer pallet is used for the color look-up. However, transparent color management (transparent color code setting and management of code 0) is different from ordinary C layer pixels— alpha blending cannot be used for the cursor color and the alpha bit in the color data registered to the color palette is ignored.

5.4.2 Cursor Management

The display priority for hardware cursors is programmable. The cursor can be displayed either on top or underneath the C layer using this feature. A separate setting can be made for each hardware cursor. If part of a hardware cursor crosses the display frame border, the part outside the border is not shown.

However, with cursor 1 displayed over the C layer and cursor 0 displayed under the C layer, the cursor 1 display has priority over the cursor 0 display.

5.5 Processing Flow for Display Data

Processing such as layer overlapping (superimposing) key is performed as follows:

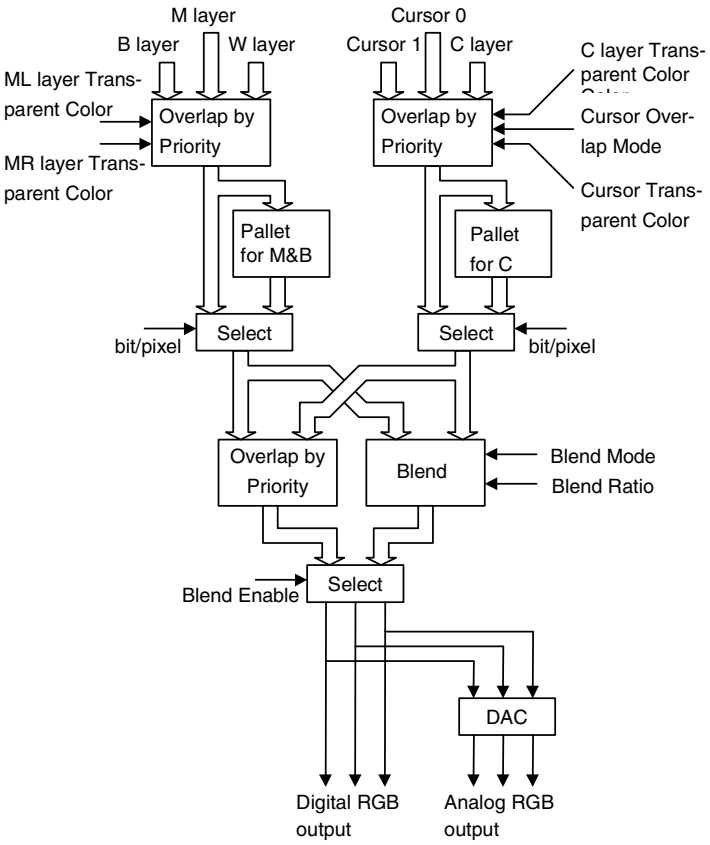


Fig. 5.4 Display data processing flow

ML layer Transparent Color

Specifies transparent color code for left side of M layer

The color code corresponding to the transparent color is used to output transparent image data for the lower layer.

ML layer Transparent Color

Specifies transparent color code for right side of M layer

The color code corresponding to the transparent color is used to output transparent image data for the lower layer.

C layer Transparent Color

Specifies transparent color code for C layer

The color code corresponding to the transparent color is used to output transparent image data for the lower layer.

Cursor Transparent Color

Specifies transparent color code for cursor

Cursor Priority Mode

Specifies whether or not to display cursor above C layer

Blend Mode

Defines correspondence between blend coefficients and variables used when applying blend coefficients

Blend ratio

Specifies blend ratio with accuracy of 1/16

Blend Enable

Specifies whether or not to use Blend

5.6 Synchronization Control

5.6.1 Applicable Display Resolution

The following table shows typical display resolutions and their sync signal frequencies. The pixel clock frequency is determined by setting the division rate of the display reference clock. The display reference clock is either the internal PLL (200.45452 MHz at input frequency of 14.31818 MHz), or the clock supplied to the DCLKI input pin. The following table gives the clock division rate used when the internal PLL is the display reference clock:

Table 5-1 Resolution and Display frequency

Resolution	Division rate of reference clock	Pixel frequency	Horizontal total pixel count	Horizontal frequency	Vertical total raster count	Vertical frequency
320 × 240	1/30	6.7 MHz	424	15.76 kHz	263	59.9 Hz
400 × 240	1/24	8.4 MHz	530	15.76 kHz	263	59.9 Hz
480 × 240	1/20	10.0 MHz	636	15.76 kHz	263	59.9 Hz
640 × 480	1/8	25.1 MHz	800	31.5 kHz	525	59.7 Hz
854 × 480	1/6	33.4 MHz	1062	31.3 kHz	525	59.9 Hz
800 × 600	1/5	40.1 MHz	1056	38.0 kHz	633	60.0 Hz
1024 × 768	1/3	66.8 MHz	1389	48.1 kHz	806	59.9 Hz

Pixel frequency =clock x ratio x reference clock division rate (when internal PLL selected)

= DCLKI input frequency × reference clock division rate (when DCLKI selected)

Horizontal frequency = Pixel frequency/Horizontal total pixel count

Vertical frequency = Horizontal frequency/Vertical total raster count

5.6.2 Interlace Display

ORCHID can generate both a non-interlace display and an interlace display. For the interlace display, the 1st, 3rd, … (2n+1)th rasters of the display screen are output to odd fields, and 2nd, 4th, … 2n-th rasters of the display screen are output to even fields.

5.7 Video Interface, NTSC/PAL Output

If an NTSC signal is required, an NTSC / PAL encoder device should be connected externally as shown below:

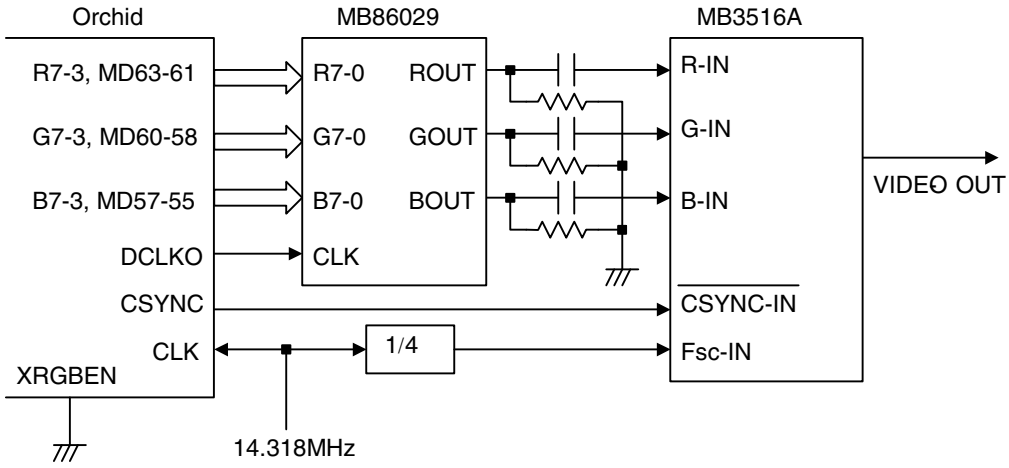


Fig. 5.5 Example of NTSC Encoder Connection

Besides, to use digital NTSC encoder is possible, but typically usable pixel frequency and resolution are limited. Refer to digital NTSC encoder specification.

6. Video Capture

6.1 Format

6.1.1 Input Data Format

Input a digital video stream in the ITU RBT-656 format. NTSC and PAL signals are both supported.

6.1.2 Video Signal Capture

When the VIE bit of the video capture mode register (VCM) is 1, ORCHID is enabled to capture video stream data from the 8-bit VI pin in synchronization with the CCLK clock. Only a digital video stream conforming to ITU-RBT656 can be processed. For this reason, a Y,Cb,Cr 4:2:2 format to which timing reference codes are added is used. The video stream is captured according to the timing reference codes; ORCHID automatically supports both NTSC and PAL. However, to detect error codes, set NTSC/PAL in the VS bit of VCM. If NTSC is not set, reference the number of data in the capture data count register (CDCN). If PAL is not set, reference the number of data in the capture data counter register (CDCP). If the reference data does not match the stream data, bit 4 to bit 0 of the video capture status register (VCS) will be values other than 0000.

6.1.3 Non-interlace Transformation

Captured video graphics can be displayed in non-interlaced format. Two modes (BOB and WEAVE) can be selected at non-interlace transformation.

- BOB Mode

In odd fields, the even-field rasters generated by average interpolation are added to produce one frame. In even fields, the odd-field rasters generated by average interpolation are added to produce one frame.

The BOB mode is selected by enabling vertical interpolation with the VI bit of the video capture mode register (VCM) and setting the WIM bit of the W layer mode register (WM) to 0.

- WEAVE Mode

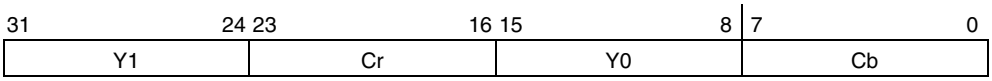
Odd and even fields are merged in the video capture buffer to produce one frame. Vertical resolutions in the WEAVE mode are higher than those in the BOB mode but raster dislocation appears at moving places.

The WEAVE mode is selected by disabling vertical interpolation with the VI bit of VCM and setting the WIM bit of WM to 1.

6.2 Video Buffer

6.2.1 Data Format

Captured graphics are stored in memory in the 16-bit/pixel YcbCr format. Video data is transformed to the RGB format when it is displayed in the W-layer.



6.2.2 Synchronous Control

Video graphics data is written to scan-independent memory for display. Memory for video capture is controlled by the ring buffer method. When graphics data for one frame is ready in memory, the frame is displayed.

If the video capture frame rate is different from the display frame rate, a frame is omitted or the same frame is displayed continuously.

6.2.3 Area Allocation

Allocate an area of about 2.2 frames to the video capture buffer. The size of this area is equivalent to the size that considers the margin equivalent to the double buffer of the frame. Set the starting address and upper-limit address of the area in the CBOA/CBLA registers. Here, specify the raster start position as the upper-limit address.

To allocate n rasters as the video capture buffer, set the upper-limit value as follows:

CBLA = CBOA + 64n X CBS

If CBLA does not match the head of a raster, video capture data is written beyond the upper limit by only 1 raster (max.). Note that if other meaningful data is held in the area, the user-intended operation is hindered by overwriting.

For reduced display, allocate the buffer area of the reduced frame size.

6.2.4 Window Display

The W layer is used to display the captured video graphics. A part or the whole of the captured graphics can be displayed as the full screen or as a window.

To capture and display video graphics, set the W layer to the capture synchronous mode (WCS = 1). In the capture synchronous mode, the W layer displays the latest frame in the video capture buffer. The display addresses used in the normal mode are ignored.

The stride of the W layer must match that of the video capture buffer. If they do not match, the displayed graphics have oblique distortion.

Match the display size of the W layer with the reducted graphics size of the video capture. Setting the display size of the W layer larger than the capture image size causes display of invalid data.

The W layer supports selection of the RGB display format and YcbCr display format. To capture video graphics, select the YcbCr display format (WYC = 1).

6.2.5 Interlace Display

The graphics captured in the video capture buffer in the WEAVE mode can be displayed in interlace. Interlace display setting is the same as WEAVE mode setting. Select 'Interlace & video display' for display scan.

Flicker appears in moving video graphics. To prevent flicker, set the OO (Odd Only) bit of the capture buffer mode register (CBM) to "1".

6.3 Scaling

6.3.1 Video Reduction Function

When the CM bits of the video capture mode register (VCM) are 11, ORCHID reduces the video screen size. The reduction can be set independently in the vertical and horizontal scales. The reduction is set per line in the vertical direction and in 2-pixel units in the horizontal direction. The scale setting value is defined by an input/output value. It is a 16-bit fixed fraction where the integer is represented by 5 bits and the fraction is represented by 11 bits. Valid setting values are from 0800_H to $FFFF_H$. Set the vertical direction at bit 31 to bit 16 of the capture scale register (CSC) and the horizontal direction at bits 15 to bit 00. The initial value for this register is 08000800_H (once). An example of the expressions for setting a reduction in the vertical and horizontal directions is shown below.

Reduction in vertical direction	576 → 490 lines	576/490 = 1.176
	1.176×2048=2408	→ 0968_H
Reduction in horizontal direction	720 → 648 pixels	720/648 = 1.111
	1.111×2048=2275	→ $08E3_H$

Therefore, $096808E3_H$ is set in CSC.

The capture horizontal pixel register (CHP) and capture vertical pixel register (CVP) are used to limit the number of pixels processed during scaling. They are not used to set scaling values. Clamp processing is performed on the video streaming data outside the values set in CHP and CVP. Usually, the defaults for these registers are used.

6.3.2 Vertical Interpolation

When the VI bit of the video capture mode register (VCM) is “0”, data in the same field is used to interpolate the interlace screen vertically. The interlace screen is doubled in the vertical direction. When the VI bit is “1”, the interlace screen is not interpolated vertically.

6.4 Error Handling

6.4.1 Error Detection Function

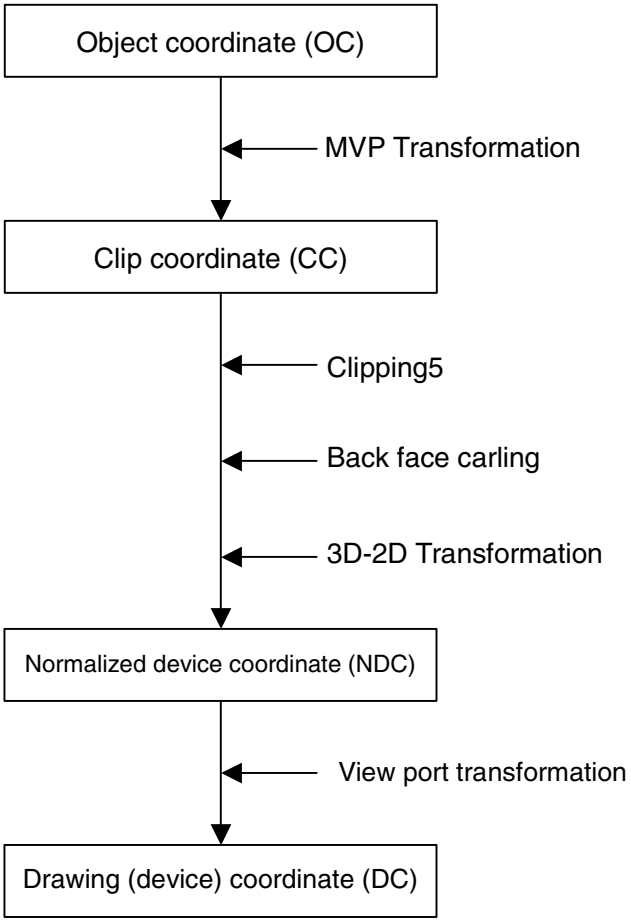
If an expected control code is not detected in the input video stream, an error occurs. If an error occurs, the status is returned to the register.

7. Geometry Engine

7.1 Geometry Pipeline

7.1.1 Processing Flow

The flow of geometry processing is shown below.



7.1.2 Model-View-Projection (MVP) Transformation (OC→CC Coordinate Transformation)

The geometry engine transforms the vertex of the “OC” coordinate system specified by the G_Vertex packet to the “CC” coordinate system according to the coordinate transformation matrix (OC→CC Matrix) specified by the G_LoadMatrix packet. The “OC→CC Matrix” is a “4 x 4” matrix consisting of a ModelView matrix and a Projection matrix.

If “Zoc” is not included in the input parameter of the G_Vertex packet (Z-bit of GMDR0 is off), (OC→CC) coordinate transformation is processed as “Zoc = 0”.

When GMDR0[0] is 0 (orthogonal projection transformation), OC→CC coordinate transformation is processed as “Wcc = 1.0”.

OC:Object Coordinates

CC:Clip Coordinates

$$\begin{pmatrix} Xcc \\ Ycc \\ Zcc \\ Wcc \end{pmatrix} \quad \begin{pmatrix} Ma0 & Ma1 & Ma2 & Ma3 \\ Mb0 & Mb1 & Mb2 & Mb3 \\ Mc0 & Mc1 & Mc2 & Mc3 \\ Md0 & Md1 & Md2 & Md3 \end{pmatrix} \quad \begin{pmatrix} Xoc \\ Yoc \\ Zoc \\ 1 \end{pmatrix}$$

Ma0 to Md3: OC → CC Matrix

Xoc to Zoc: X, Y, and Z of OC coordinate system

Xcc to Woc: X, Y, Z, and W of CC coordinate system

7.1.3 3D-2D Transformation (CC→NDC Coordinate Transformation)

The geometry engine divides “XYZ” of the “CC” coordinate system by “Wcc” (Perspective Division).

NDC: Normalized Device Coordinates

$$\begin{pmatrix} Xndc \\ Yndc \\ Zndc \end{pmatrix} \quad 1/Wcc \quad \begin{pmatrix} Xcc \\ Ycc \\ Zcc \end{pmatrix}$$

Xndc to Zndc: X, Y, and Z of “NDC” coordinate system

7.1.4 View Port Transformation (NDC→DC Coordinate Transformation)

The geometry engine transforms “XYZ” of the “NDC” coordinate system to the “DC” coordinate system according to the transformation coefficient specified by G_ViewPort and G_DepthRange.

“X_Scaling,X_Offset” and “Y_Scaling,Y_Offset” are coefficients to be mapped finally to Frame Buffer. Xdc and Ydc must be included within the drawing input range (-4096 to 4095). “Z_Scaling” and “Z_Offset” are coefficients to be mapped finally to “Z Buffer”. “Zdc” must be included within the “Z Buffer” range (0 to 65535).

DC: Device Coordinates

Xdc = X_Scaling*Xndc + X_Offset

Ydc = Y_Scaling*Yndc + Y_Offset

Zdc = Z_Scaling*Zndc + Z_Offset

7.1.5 View Volume Clipping

Expression for determination

The expression for determining the ORCHID view volume clipping is shown below. W clipping is intended to prevent the overflow caused by 1/W.

$Xmin*Wcc \leq Xcc \leq Xmax*Wcc$

$Ymin*Wcc \leq Ycc \leq Ymax*Wcc$

$Zmin*Wcc \leq Zcc \leq Zmax*Wcc$

$Wmin \leq Wcc$

Note: Xmin, Xmax, Ymin, Ymax, Zmin, Zmax, and Wmin are the clip boundary values set by the G_ViewVolumeXYClip/ZClip/WClip packet.

Clipping-on/-off

View volume clipping-on/-off can be switched by using the clip boundary values set by the G_ViewVolumeXYClip/Zclip/WClip packet. To switch view volume clipping to off, set the maximum and minimum values of the geometry data format (IEEE single-precision floating point(*1)) in the “Clip.max” value(*2) and “Clip.min” value(*3), respectively. In this case, ‘All coordinate transformation results within view volume range’ can be evaluated, making it possible to obtain the effect of view volume clipping-off. This method is available when W clipping does not generate. When clip boundary (Wmin) generating W clipping is set, set adequate clip boundary to Clip.max, Clip.min because of clipping process in terms of each clip surface.

If other values are set in “Clip.max” and Clip.min, view volume clipping-on operates. The coordinate transformation result is always compared with the values set in “Clip.max” and “Clip.min”.

*1: Maximum value = 0x7ff7ffff, minimum value = 0xfff7ffff

*2: Xmin,Ymin, Zmin, Wmin

*3: Xmax, Ymax, Zmax

An example of the G_ViewVolumeZclip packet is shown below.

0xf1012010 //Setting of GMDR0

0x00000000 //Data format: Floating point data format

0x45000000 //G_ViewVolumeZclip packet

0xff7fffff //Zmin.float setting value (minimum value of IEEE single-precision floating point)

0x7f7fffff //Zmax.float setting value (maximum value of IEEE single-precision floating point)

Example of G_ViewVolumeZclip Packet when Z Clipping Off

"W" clipping at orthogonal projection transformation

"W" at orthogonal projection transformation (GMDR0[0] = 0) is treated as "Wcc=1.0". For this reason, to suppress "W" clipping, the set "Wmin" value must be greater than 0 and 1.0 or less.

Relationship with drawing clip frame

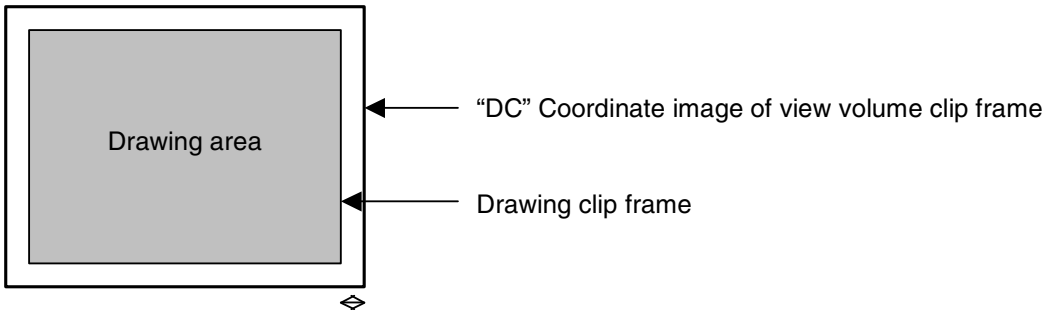
For the following reasons, the clip boundary values of the view volume should be set so that the values after DC coordinate transformation will be greater than the drawing clip frame (2 pixels or more).

(1) "XY" on the view volume clip frame of the "CC" coordinate system may be drawn one pixel outside or inside the frame due to an operation error when it is finally mapped to the "DC" coordinate system.

(2) When the end point of a line overlaps the view volume frame mapped to the "DC" coordinate system, there are two cases, where the dots on the frame are drawn, and not drawn depending on the specification of the line drawing attribute (end point drawing/non-drawing).

(3) When the starting point of a line overlaps the view volume frame mapped to the "DC" coordinate system, the dots on the frame are always drawn. When the line drawing attribute is 'end point non-drawing,' the dots on the frame are drawn at the starting point, but they may not be drawn at the end point.

(4) When applying to triangle and polygon drawing the rasterizing rule 'dots containing center of pixel drawn. Dots on right side and base of triangle not drawn.' depending on the value of the fraction, a gap may be produced between the right side and base of the frame.



7.1.6 Back face carling

In ORCHID, a triangle direction can be defined and a mode in which drawing in the backward direction is inhibited (back face carling) is supported. The on/off operation is controlled by the GMDR2[0] setting. GMDR2[0] must be set to 1 (on)only when back face carling is required. When back face carling is not required as in 'line,' 'point,' and 'polygon primitive,' GMDR2[0] must be set to 0 (off).

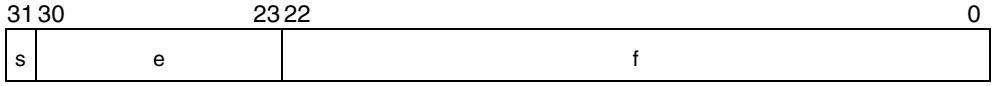
7.2 Data Format

7.2.1 Data Format

The supported data formats are 32-bit single-precision floating-point format, 32-bit fixed-point format, integer packed format, and RGB packed format. All internal processing is performed in the floating-point format. For this reason, the integer packed format, fixed-point format, and RGB packed format must be converted to the floating-point format. The processing speeds in these formats are slightly lower than in the 32-bit single-precision floating-point format.

The data format to use is selected by setting the GMDR0 register.

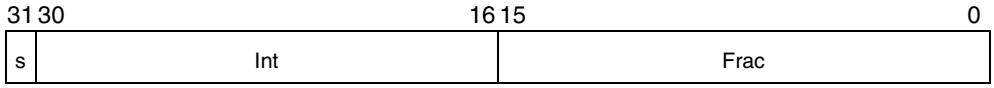
(1) 32-bit single-precision floating-point format



s: Sign bit (1 bit)
e: Exponent part (8 bits)
f: Mantissa (23 bits) '1.f' indicates the fraction. '1' is a hidden bit.

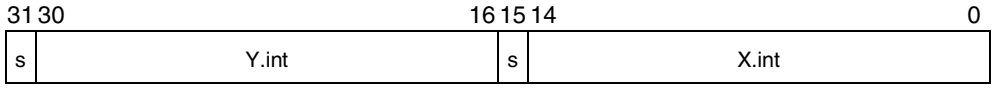
The numerical value of the floating-point format becomes $(-1)^{s}(1.f)2^{(e-127)}$ $(0 < e < 255)$.

(2) Signed fixed-point format (SFIX16.16)



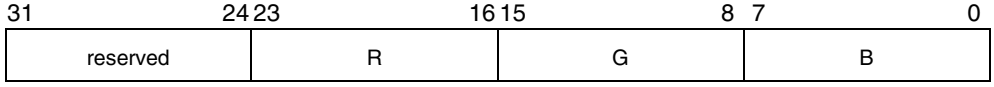
s: Sign bit (1 bit)
int: Integer (15 bits)
frac: Fraction (16 bits)

(3) Signed integer packed format (SINT16.SINT16)



s: Sign bit (1 bit)
int: Integer (15 bits)

(4) RGB packed format



R, G, B: Color bits (8 bits)

7.3 Setup Engine

7.3.1 Setup Processing

The vertex data transformed by the geometry engine is transferred to the setup engine. ORCHID has a drawing interface that is compatible with the MB86290A. It operates parameters for various slope calculations, etc., with the setup engine. When the obtained parameters are set in the drawing engine, the final drawing processing starts.

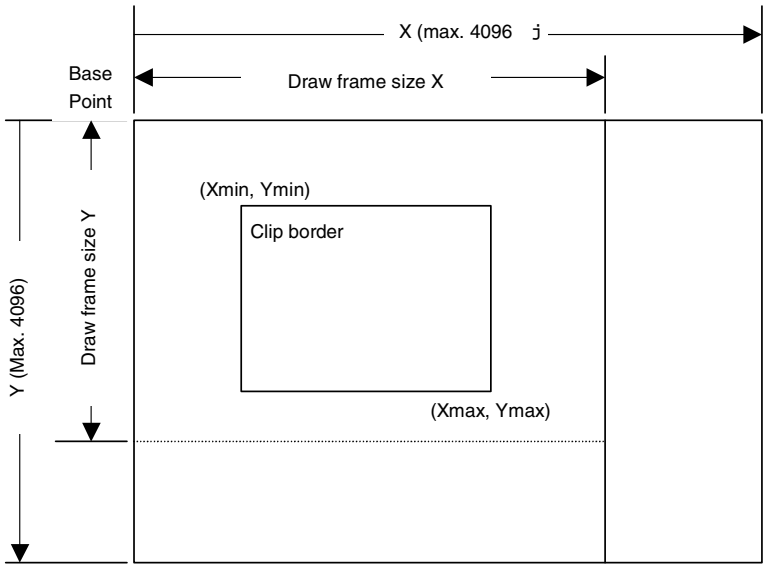
8. Drawing Processing

8.1 Coordinate System

8.1.1 Drawing Coordinate

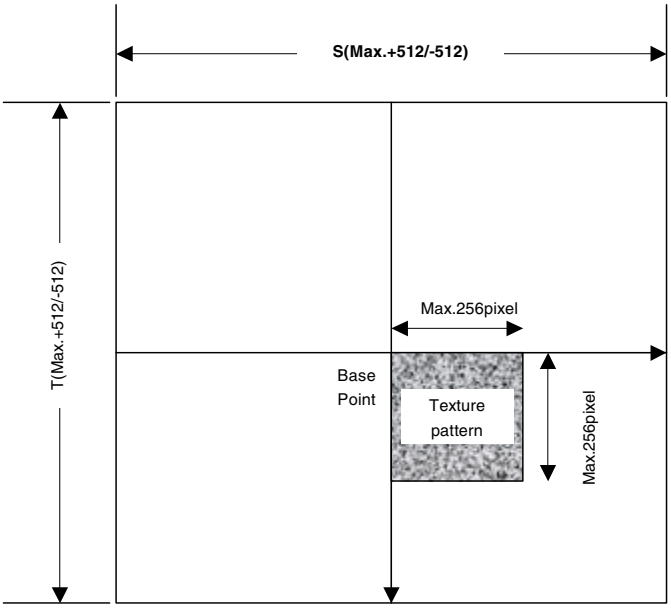
After coordinates have been calculated by the geometry engine, ORCHID draws data in the drawing frame in the graphics memory that finally uses the drawing coordinates (device coordinates).

Drawing frame as a 2D coordinate with the origin at the top left edge. The maximum coordinate is 4096 x 4096. Each drawing frame is located in the Graphics Memory by setting the address of the origin and width (pixel size of X span). Although the maximum size of Y span does not need to be specified, take care about the memory size allocation so as not to overlap any other frames. Also, setting the clip field (top left and bottom right coordinates in registers) prevents drawing of all images outside the border of the clip window.



8.1.2 Texture Coordinate

This is another 2D coordinate specified as S and T (S: horizontal, T: vertical). Any integer in a range of –512 to +511 can be used as the S and T coordinates. The texture coordinate is correlated to the 2D coordinate of a vertex. All vertices forming a polygon have correlated texture coordinates. One texture style pattern can be applied to up to 256 × 256 pixels. The applied texture size is set in the register. When the S and T coordinate exceeds the maximum size of the texture style pattern, the repeat, cramp or border color option is selected.



8.1.3 Frame Buffer

For drawing, the following area must be assigned to the Graphics Memory. The frame size (number of pixels on X span) is common for these areas.

Drawing frame

The results of drawing are contained in the graphical image data area. Both the direct and indirect color mode are applicable.

Z buffer

nts area dr used to eliminate hidden surfaces in drawinga3D graphics. 2i bytes/pixel of area is required.

Polygon draw flag buffer

This area is used to perform polygon drawing hidden surfaces in 3D graphics drawing. 1bit/pixel of area is required. 1 line is aligned by byte to byte.

8.2 Figure Drawing

8.2.1 Drawing Primitives

ORCHID has a drawing interface that is compatible with the MB86290A graphics controller, which does not perform geometry processing. The following types of figure drawing primitives are compatible with the MB86290A.

- Point
- Line
- Triangle
- Fast2DLine
- Fast2Dtriangle
- Polygon

8.2.2 Polygon Drawing

An irregular polygon (including concave shape) is drawn by dedicated hardware as follows:

1.Execute PolygonBegin command

Initialize polygon draw enginew

2.Draw vertices.

Draw outline of polygon and plot all vertices to polygon draw flag buffer utilizing Fast2Dtriangle primitive.

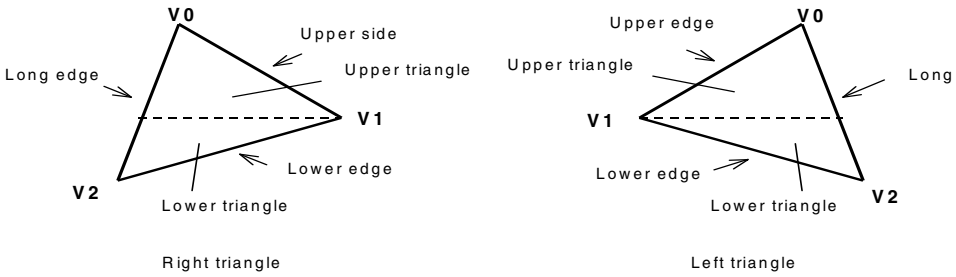
3.Execute PolygonEnd command.

Copy shape in polygon draw flag buffer to drawing frame and fill shape with color or specified tiling pattern.

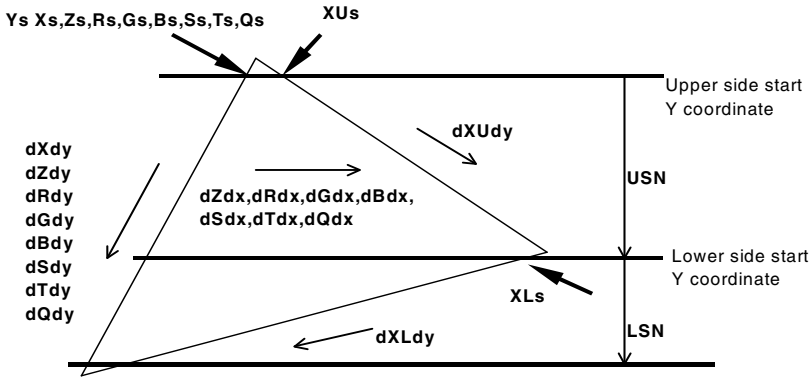
8.2.3 Drawing Parameters

The MB86290A-compatible interface uses the following parameters to draw data:

The triangles (Right triangle and Left triangle) are distinguished according to the locations of three vertices as follows (not used for Fast2Dtriangle):



The following parameters are required for drawing triangles (For Fast2Dtriangle, X and Y coordinates of each vertex are specified).



Note: Be careful about the positional relationship between coordinates Xs, XUs, and XLs.

For example, in the above diagram, when a right-hand triangle is drawn using the parameter that shows the coordinates positional relationship Xs (upper edge start Y coordinate) > XUs or Xs (lower edge start Y coordinate) > XLs, the expected picture may not be drawn.

Ys	Y coordinate start position of long side
Xs	X coordinate start position of long side
XUs	X coordinate start position of upper side
XLs	X coordinate start position of lower side
Zs	Z coordinate start position of long side
Rs	R value at (Xs, Ys, Zs) of long side
Gs	G value at (Xs, Ys, Zs) of long side
Bs	B value at (Xs, Ys, Zs) of long side
Ss	S coordinate of texture at (Xs, Ys, Zs) of long side
Ts	T coordinate of texture at (Xs, Ys, Zs) of long side
Qs	Q (Perspective correction value) of texture at (Xs, Ys, Zs) of long side
dXdy	X DDA value of long side
dXUdy	X DDA value of upper side
dXLdy	X DDA value of lower side
dZdy	Z DDA value of long side
dRdy	R DDA value of long side
dGdy	G DDA value of long side
dBdy	B DDA value of long side
dSdy	S DDA value of long side
dTdy	T DDA value of long side
dQdy	Q DDA value of long side
USN	Number of spans (rasters) of top triangle
LSN	Number of spans (rasters) of bottom triangle
dZdx	Z DDA value of horizontal way
dRdx	R DDA value of horizontal way
dGdx	G DDA value of horizontal way
dBdx	B DDA value of horizontal way
dSdx	S DDA value of horizontal way
dTdx	T DDA value of horizontal way
dQdx	Q DDA value of horizontal way

8.2.4 Anti-aliasing Function

ORCHID performs anti-aliasing to eliminate jaggies on line edges and make lines appear smooth. To use this function at the edges of primitives, redraw the primitive edges with anti-alias lines.

8.3 Bit Map Processing

8.3.1 BLT

A rectangular shape in pixel units can be transferred between two separate physical memory areas as follows:

1. From host CPU to Drawing frame memory
2. From Graphics Memory (other than Drawing frame memory area) to drawing memory
3. From host CPU to internal texture memory
4. From Graphics Memory to internal texture memory

When Drawing frame memory is designated as the destination, the result of logical calculation between the source and current value in the designated destination can be stored as well.

Setting a transparent color enables permiable drawing of a specific pixel.

If part of the source and destination of the BLT field are physically overlapped in the display frame, the start address (from which vertex the BLT field to be transferred) must be set carefully.

8.3.2 Pattern Data Format

ORCHID can handle three bit map data formats: indirect color mode (8 bits/pixel), direct color mode (16 bits/pixel), and binary bit map (1 bit/pixel). The direct color mode is used for texture patterns. Either the indirect or direct color mode is used for tiling patterns. The binary bit map is used for character/font patterns, where foreground color is used for bitmap = 1 pixel, and background color is applied for bitmap = 0 pixels.

8.4 Texture Mapping

Texture mapping is supported when the direct color mode (16 bits/pixel) drawing frame is used.

8.4.1 Texture Size

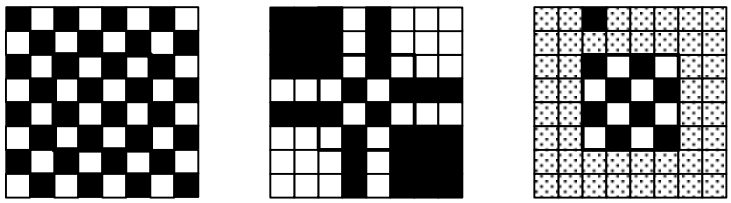
ORCHID reads texcel data from the specified texture coordinate (S, T) position, and pastes that data at the correlated pixel position of the polygon. The applicable texture data size is 16, 32, 64, 128 or 256 pixels per S and T, respectively. Texture mapping is used only when the direct color mode (16bit/pixel) is used.

8.4.2 Texture Memory

Texture pattern data is stored in either ORCHID internal texture buffer or external Graphics Memory. The internal texture buffer size is 8 Kbyte and can hold up to 64 $\times$ 64 pixels of texture. If the texture pattern size is smaller than 64 $\times$ 64pixels, it is best to store it in the internal texture buffer because the texture mapping speed is faster. Please note that it cannot be guaranteed to draw correctly by access to the internal texture buffer under drawing except for displaylist access, for instance CPU read or write.

8.4.3 Texture Lapping

If a negative or larger than applicable value is specified as the texture coordinate (S, T), according to the setting, one of these options (repeat, cramp or border) is selected for the 'out-of-range' texture mapping. The mapping image for each case is shown below:



Repeat Cramp Border

Repeat

This just masks the upper bits of the applied (S, T) coordinate and enables the lower bits of the coordinate within the specified texture pattern size. When the texture pattern size is 64 $\times$ 64pixels, it masks the upper bits of the integer part of (S, T) the coordinate and enables the lower 6 bits.

Cramp

When the applied (S, T) coordinate is either negative or larger than the specified texture pattern size, cramp the (S, T) coordinate as follows:

S < 0	S = 0
S > Texture X size – 1	S = Texture X size – 1

Border

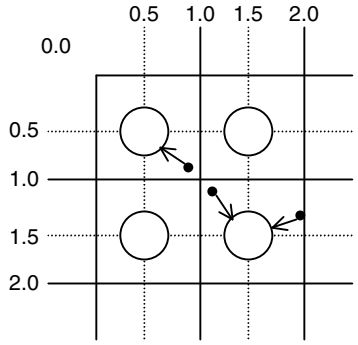
When the applied (S, T) coordinate is either negative or larger than the specified texture pattern size, the outside of the specified texture pattern is rendered in the 'border' color.

8.4.4 Filtering

ORCHID supports two texture filtering modes: point filtering, and bi-linear filtering.

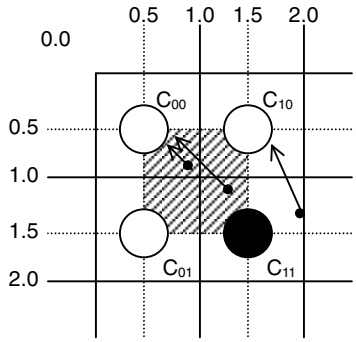
Point filtering

This mode uses the texcel data specified by the (S, T) coordinate. The nearest texcel in the texture pattern is chosen according to the calculated (S, T) coordinate.



Bi linear filtering

This mode picks the four nearest texcels from the calculated (S, T) coordinate. The color is blended and the texcel image is defined according to the distance between each of these texcels and the calculated (S, T) coordinate.



8.4.5 Perspective Collection

This function adjusts the depth distortion of the 3D projection in the texture mapping process. For this adjustment, the 'Q' element of the texture coordinate (Q = 1/W) is defined from the 3D coordinate of the correlated vertex.

8.4.6 Texture Blending

ORCHID supports the following three texture blending modes:

Decal

This mode displays the mapped texcel color regardless the native polygon color.

Modulate

This mode multiplies the native polygon color (C_P) and sampled texcel color (C_R) and display the result (C_O).

$$C_0 = C_R \times C_P$$

Stencil

This mode uses the MSB to select the display color from the sampled texcel color.

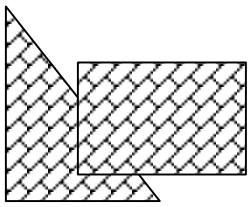
MSB = 1: Texcel color

MSB = 0: Polygon color

8.5 Rendering

8.5.1 Tiling

Tiling reads the pixel color from the correlated tiling pattern and maps it onto the polygon. The tiling pixel is determined by the coordinate of the correlated pixel irrespective of the primitive position and size. Since the tiling pattern is stored in the internal texture buffer, this function and texture mapping cannot be used at the same time. Also, the tiling pattern size is limited to within 64 x 64 pixels.



Example of Tiling

8.5.2 Alpha Blending

Alpha blending blends the pixel's native color and current color of that pixel position according to the blending ratio parameter set in the alpha register. This function cannot be used simultaneously with logical calculation. It can be used only when the direct color mode (16 bits/pixel) is used. The blended color C is calculated as shown below when the native color of the pixel to be rendered is C_P, the current pixel color of that position is C_F, and the alpha value set in the alpha register is A:

$C = C_P \times A + (1\text{-}A) \times C_F$

The alpha value is specified as 8-bit data. 00h means alpha value 0% and FFh means alpha value 100%. When the texture mapping function is enabled, the following blending modes are applicable:

Normal

Blends post texture mapping color with current frame buffer color

Stencil

Uses MSB of texcel color to select display color:

MSB = 1: Texcel color

MSB = 0: Current frame buffer color

Stencil alpha

Uses MSB of texcel color to select and activate alpha-blend function:

MSB = 1: Alpha blend texcel color and current frame buffer color

MSB = 0: Current frame buffer color

8.5.3 Logical Calculation

This mode executes a logical calculation between the new pixel color to be rendered and the current frame memory color and displays the result. Alpha blending cannot be used when this function is used.

Type	ID	Operation	Type	ID	Operation
CLEAR	0000	0	AND	0001	S & D
COPY	0011	S	OR	0111	S | D
NOP	0101	D	NAND	1110	!(S & D)
SET	1111	1	NOR	1000	!(S | D)
COPY INVERTED	1100	!S	XOR	0110	S xor D
INVERT	1010	!D	EQUIV	1001	!(S xor D)
AND REVERSE	0010	S & !D	AND INVERTED	0100	!S & D
OR REVERSE	1011	S | !D	OR INVERTED	1101	!S | D

8.5.4 Hidden Surface Management

ORCHID supports the Z buffer for hidden surface management.

This function compares the Z value of a new pixel to be rendered and the existing Z value in the Z buffer. Display/not display is switched according to the Z-compare mode setting. Define the Z-buffer access options in the ZWRITEMASK mode. The Z-comparison type is determined by the Z compare mode.

ZWRITEMASK	1	Compare Z values, no Z buffer overwrite
	0	Compare Z values and overwrite result to Z buffer

Z Compare mode	ID	Condition
NEVER	000	Never draw
ALWAYS	001	Always draw
LESS	010	Draw if pixel Z value < current Z buffer value
LEQUAL	011	Draw if pixel Z value $\leq$ current Z buffer value
EQUAL	100	Draw if pixel Z value = current Z buffer value
GEQUAL	101	Draw if pixel Z value $\leq$ current Z buffer value
GREATER	110	Draw if pixel Z value > current Z buffer value
NOTEQUAL	111	Draw if pixel Z value != current Z buffer value

8.6 Drawing Attributes

8.6.1 Line Draw Attributes

When line draw operations are performed, the following attributes apply:

Line Draw Attributes

Drawing Attribute	Description
Line Width	Line width selectable in range of 1-32 pixels
Broken Line Draw	Specify broken line pattern in 32-bit data
Anti-alias	Line edge smoothed when anti-aliasing enabled

8.6.2 Triangle Draw Attributes

When triangle draw operations are performed, the following attributes apply. Texture mapping and tiling have separated texture attributes:

Triangle Draw Attributes

Drawing Attribute	Description
Shading	Gouraud shading or flat shading selectable
Alpha blending	Set alpha blend enable per polygon
Blending parameter	Set color blend ratio of alpha blend

8.6.3 Texture Attributes

The following attributes apply for texture mapping:

Texture Attributes

Drawing Attribute	Description
Texture mode	Select either texture mapping or tiling
Texture memory mode	Select either internal texture buffer or external Graphics Memory to use in texture mapping
Texture filter	Select either point sampling or bi-linear filtering
Texture coordinate correction	Select either linear or perspective correction
Texture wrap	Select either repeat or cramp of texture pattern
Texture blend mode	Select either decal or modulate

8.6.4 BLT Attributes

When BLT draw are performed, the following attributes apply:

BLT Attributes

Drawing Attribute	Description
Logical calculation mode	Specify two source logical calculation mode
Transparency processing	Set transparent copy mode and transparent color

8.6.5 Character Pattern Drawing Attributes

Character Pattern Drawing

Drawing Attribute	Description
Character pattern enlarge/shrink	2×2, $\times 2$ horizontal, $1/2 \times 1/2$, $\times 1/2$ horizontal
Character pattern color	Set character color and background color
Transparency/non-transparency	Set background color to transparency/non-transparency

9 Display List

9.1 Overview

Display list is a set of display list commands, parameters and pattern data. All display list commands in a display list are executed consequently (Note that display list command does not mean draw command).

The display list is transferred to the display list FIFO by one of the following methods:

- CPU write to display FIFO
- DMA transfer from main memory to display FIFO
- Register set to transfer from graphics memory to display FIFO

Display list Command-1
Data 1-1
Data 1-2
Data 1-3
Display list Command-2
Data 2-1
Data 2-2
Data 2-3

Display List

9.1.1 Header Format

The format of the display list header is shown below.

Format Overview

Format	31 ... 24	23 ... 16	15 ... 0		
Format 1	Type	Reserved	Reserved		
Format 2	Type	Count	Address		
Format 3	Type	Reserved	Reserved		Vertex
Format 4	Type	Reserved	Reserved	Flag	Vertex
Format 5	Type	Command	Reserved		
Format 6	Type	Command	Count		
Format 7	Type	Command	Reserved		Vertex
Format 8	Type	Command	Reserved	Flag	Vertex
Format 9	Type	Reserved	Reserved	Flag	
Format 10	Type	Reserved	Count		

Description of Each Field

Type	Display list type
Command	Command
Count	Number of parameters excluding header
Address	Address value used at data transfer
Vertex	Vertex number
Flag	Dedicated attribute flag of display list command

Vertex Number Specified in Vertex Code

Vertex	Vertex number (Line)	Vertex number (Triangle)
00	V0	V0
01	V1	V1
10	Inhibited	V2
11	Inhibited	Inhibited

9.1.2 Parameter format

The parameter format of the geometry command depends on the value set in the D field of GMDR0. When the D field is 00, all parameters are handled in the floating-point format. When the D field is 01, colors are handled as the packed RGB format, and others are handled as the fixed-point format. When the D field is 11, XY is handled as the packed integer format, colors are handled as the packed RGB format, and others are handled as the fixed-point format.

In the following text, the floating-point format is suffixed by **.float**, the fixed point format is suffixed by **.fixed**, and the integer format is suffixed by **.int**. Set GMDR0 properly to match parameter suffixes.

Rendering command parameters conform to the MB86290A data format.

9.2 Geometry Commands

9.2.1 Geometry command list

ORCHID geometry commands and each command code are shown in the table below.

Type	Command	Description
G_Nop	No operation	No operation
G_Begin	Geometry Command See the code table.	Primitive type specification and preprocessing
G_BeginCont	No operation	Primitive type specification (vertex processing in same mode as previous mode)
G_End	No operation	End of primitive
G_Vertex	No operation	Set and draw vertex parameters
G_Init	No operation	Initialize geometry engine
G_Viewport	No operation	Scale to screen coordinates (X, Y) and set origin offset
G_DepthRange	No operation	Scale to screen coordinate (Z) and set origin offset
G_LoadMatirix	No operation	Load geometrical transformation matrix
G_ViewVolumeXYClip	No operation	Set boundary value (X,Y) of view volume clip
G_ViewVolumeZClip	No operation	Set boundary value (Z) of view volume clip
G_ViewVolumeWClip	No operation	Set boundary value (W) of view volume clip
SetLVertex2i	No operation	Pass through high-speed 2D line drawing register
SetLVertex2iP	No operation	Pass through high-speed 2D line drawing register

Type code tables

Type	Code
G_Nop	0010_0000
G_Begin	0010_0001
G_BeginCont	0010_0010
G_End	0010_0011
G_Vertex	0011_0000
G_Init	0100_0000
G_Viewport	0100_0001
G_DepthRange	0100_0010
G_LoadMatirix	0100_0011
G_ViewVolumeXYClip	0100_0100
G_ViewVolumeZClip	0100_0101
G_ViewVolumeWClip	0100_0110
SetLVertex2i	0111_0010
SetLVertex2iP	0111_0011

Geometry command code tables

(1) Floating point setup type

When setup processing is performed, all the parameters including “XY“ are calculated in the floating-point format.

Command	Code
Points	0000_0000
Lines	0000_0001
Polygon	0000_0010
Triangles	0000_0011
Line_Strip	0000_0101
Triangle_Strip	0000_0111
Triangle_Fan	0000_1000

(2) Integer setup type

When setup processing is performed, “XY“ is calculated in the integer format and other parameters are calculated in the floating-point format.

Command	Code
Points.int	0001_0000
Lines.int	0001_0001
Polygon.int	0001_0010
Triangles.int	0001_0011
Line_Strip.int	0001_0101
Triangle_Strip.int	0001_0111
Triangle_Fan.int	0001_1000

(3) “Unclipped“ integer setup type

This command does not clip the view volume.

Only “XY“ is permissible as the input parameter.

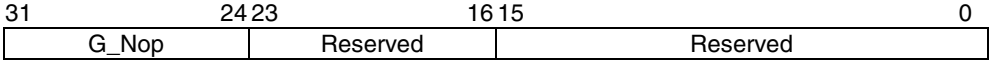
When setup processing is performed, “XY“ is calculated in the integer format.

This command is not guaranteed perspective projection transformation(GMDR0[0]=1)

Command	Code
nclip_Points.int	0011_0000
nclip_Lines.int	0011_0001
nclip_Polygon.int	0011_0010
nclip_Triangles.int	0011_0011
nclip_Line_Strip.int	0011_0101
nclip_Triangle_Strip.int	0011_0111
nclip_Triangle_Fan.int	0011_1000

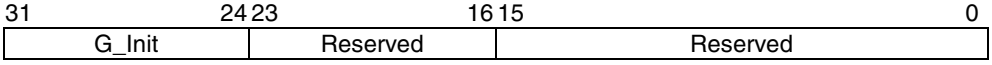
9.2.2 Explanation of Geometry Commands

G_Nop (Format 1)



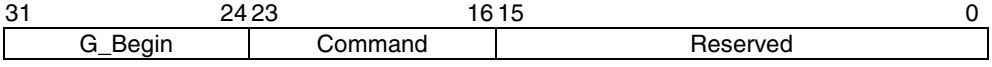
No operation

G_Init (Format 1)



Initializes geometry engine. Execute this command before processing.

G_Begin (Format 5)



Sets types of primitive for geometry processing and drawing. A vertex is set and drawn by the ***G_Vertex*** command. The ***G_Vertex*** command must be specified between the ***G_Begin*** or ***G_BeginCont*** command and ***G_End*** command.

Command:

Points*	Handles primitive as point
Lines*	Handles primitive as independent line
Polygon*	Handles primitive as polygon
Triangles*	Handles primitive as independent triangle
Line_Strip*	Handles primitive as line strip
Triangle_Strip*	Handles primitive as triangle strip
Triangle_Fan*	Handles primitive as triangle fan

G_BeginCont (Format 1)

31 24	23 16	15 0
G_BeginCont	Reserved	Reserved

When the primitive type set by the ***G_Begin*** command the last time and drawing mode are not modified, the ***G_BeginCont*** command is used instead of the ***G_Begin*** command. The ***G_BeginCont*** command is processed faster than the ***G_Begin*** command.

The packet that can be set between the ***G_End*** packet set just before and the ***G_BeginCont*** packet is only 'foreground color setting by the SetRegister packet.' The ***G_Vertex*** command must be specified between the ***G_Begin*** or ***G_BeginCont*** command and ***G_End*** command. No primitive type need be specified in the ***G_BeginCont*** command.

G_Vertex (Format 1)

When data format is floating-point format

31 24	23 16	15 0
G_Vertex	Reserved	Reserved
X.float		
Y.float		
Z.float		
R.float		
G.float		
B.float		
S.float		
T.float		

When data format is fixed-point format

31 24	23 16	15	0
G_Vertex	Reserved	Reserved	
X.fixed			
Y.fixed			
Z.fixed			
	R.int	G.int	B.int
S.fixed			
T.fixed			

When data format is packed integer format

31 24	23 16	15 0	
G_Vertex	Reserved	Reserved	
Y.int		X.int	
Z.fixed			
	R.ing	G.int	B.int
S.fixed			
T.fixed			

The ***G_Vertex*** command sets vertex parameters and processes and draws the geometry of the primitive specified by the ***G_Begin*** command. Note the following when using this command:

- Required parameters depend on the setting of the ***GMDR0*** register. Proper values must be set as the mode values of the ***MDR0*** to ***MDR4*** registers to be finally reflected at drawing. That is, when "Z" comparison is made (ZC bit of MDR1 or MDR2 = 1), the Z bit of the GMDR0 register must be set to 1. When Gouraud shading is performed (SM bit of MDR2 = 1), the C bit of the GMDR0 register must be set to 1. When texture mapping is performed (TT bits of MDR2 = 10), the ST bit of the GMDR0 register must be set to 1.
- When the Z bit of the GMDR0 register is 0, input "Z" (Zoc) is treated as "0".
- Use values normalized to 0 and 1 as texture coordinates (S, T).
- When the color RGB is floating-point format, use values normalized to 0 and 1 as the 8-bit color value. For the packed RGB, use the 8-bit color value directly.
- The GMDR1 register is valid only for line drawing; it is ignored in primitives other than line.
- The GMDR2 register is meaningful only when a triangle (not including a polygon) is drawn. At primitives other than triangle, set "0".

Usable combinations of GMDR0 mode setting and primitives are as follows:

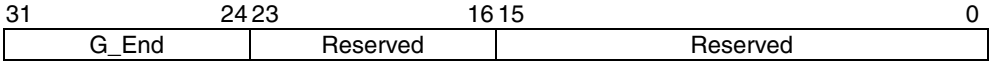
Unclipped primitives (nclip*)

(ST,Z,C)	Point	Line	Triangle	Polygon
(0,0,0)	○	○	○	○
Others	×	×	×	×

Primitives other than unclipped primitives

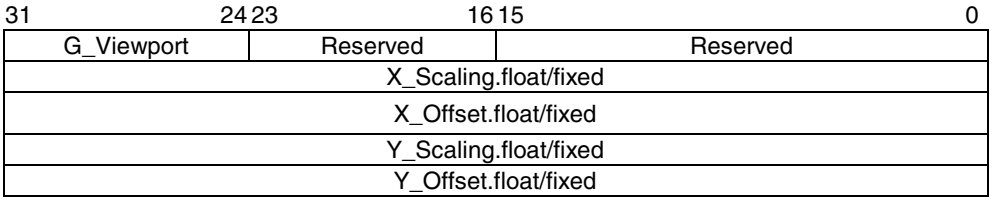
(ST,Z,C)	Point	Line	Triangle	Polygon
(0,0,0)	○	○	○	○
(0,0,1)	×	×	○	×
(0,1,0)	○	○	○	×
(0,1,1)	×	×	○	×
(1,x,x)	×	×	○	×

G_End (Format 1)



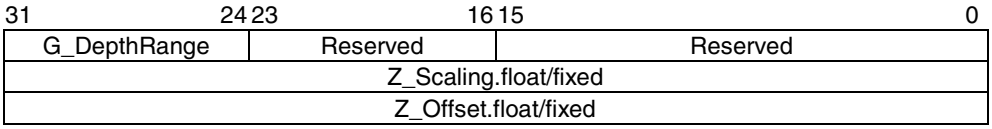
The ***G_End*** command ends one primitive. The ***G_Vertex*** command must be specified between the ***G_Begin*** or ***G_BeginCont*** command and ***G_End*** command.

G_Viewport (Format 1)



The ***G_Viewport*** command sets the “X,Y” scale/offset value used when a normalized device coordinate (NDC) is transformed into a device coordinate (DC).

G_DepthRange (Format 1)



The ***G_DepthRange*** command sets the “Z” scale/offset value used when an NDC is transformed into a DC.

G_LoadMatrix (Format 1)

31 24	23 16	15 0
G_LoadMatrix	Reserved	Reserved
Matrix_a0.float/fixed		
Matrix_a1.float/fixed		
Matrix_a2.float/fixed		
Matrix_a3.float/fixed		
Matrix_b0.float/fixed		
Matrix_b1.float/fixed		
Matrix_b2.float/fixed		
Matrix_b3.float/fixed		
Matrix_c0.float/fixed		
Matrix_c1.float/fixed		
Matrix_c2.float/fixed		
Matrix_c3.float/fixed		
Matrix_d0.float/fixed		
Matrix_d1.float/fixed		
Matrix_d2.float/fixed		
Matrix_d3.float/fixed		

The ***G_LoadMatrix*** command sets the transformation matrix used when an object coordinate (OC) is transformed into a clip coordinate (CC).

G_ViewVolumeXYClip (Format 1)

31 24	23 16	15 0
G_ViewVolumeXYClip	Reserved	Reserved
XMIN.float/fixed		
XMAX.float/fixed		
YMIN.float/fixed		
YMAX.float/fixed		

The ***G_ViewVolumeXYClip*** command sets the X,Y coordinates of the clip boundary value in view volume clipping.

G_ViewVolumeZClip (Format 1)

31 24	23 16	15 0
G_ViewVolumeZClip	Reserved	Reserved
ZMIN.float/fixed		
ZMAX.float/fixed		

The ***G_ViewVolumeZClip*** command sets the Z coordinate of the clip boundary value in view volume clipping.

G_ViewVolumeWClip (Format 1)

31 24	23 16	15 0
G_ViewVolumeWClip	Reserved	Reserved
WMIN.float/fixed		

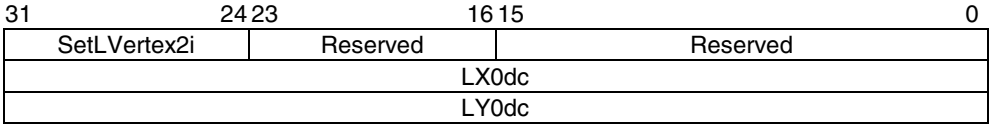
The ***G_ViewVolumeWClip*** command sets the W coordinate of the clip boundary value in view volume clipping (Minimum value only).

SetRegister (Format 2)

31 24	23 16	15 0
SetRegister	Count	Address
(Val 0)		
(Val 1)		
…		
(Val n)		

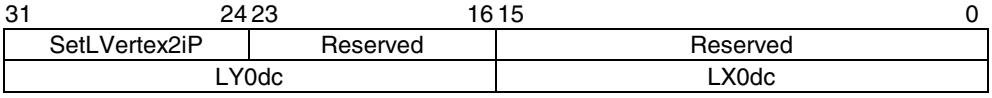
The ***SetRegister*** command is upwards-compatible with the Cremson SetRegister command. Register addresses in the geometry engine can be specified.

SetLVertex2i (Format 1)



In the geometry FIFO interface, the ***SetLVertex2i*** command issues ***SetRegister_LX0dc/LY0dc*** (vertex setting command for starting MB86290A line drawing). This command is processed faster than the ***SetRegister_LX0dc/LY0dc*** command to be input directly to the geometry FIFO interface.

SetLVertex2iP (Format 1)



The ***SetLVertex2iP*** command corresponds to the packed “XY” of the ***SetLVertex2i*** command.

9.3 Rendering Command

9.3.1 Command Overview

The following table lists ORCHID rendering commands and their command codes.

Type	Command	Description
Nop	-	No operation
Interrupt	-	Interrupt request to host CPU
Sync	-	Synchronization with events
SetRegister	-	Data set to register
SetVertex2i	Normal	Data set to Fast2DTriangle VRTX register
	PolygonBegin	Initialization of border rectangle calculation of multiple vertices random shape
Draw	PolygonEnd	Polygon flag clear (post random shape drawing operation)
	Flush_FB/Z	Flushes drawing pipelines
DrawPixel	Pixel	Plot Point
DrawPixelZ	PixelZ	Plot Point with Z value
DrawLine	Xvector	Draw Line (X)
	Yvector	Draw Line (Y)
	AntiXvector	Draw Line with anti-alias option (X)
	AntiYvector	Draw Line with anti-alias option (Y)
DrawLine2i DrawLine2iP	ZeroVector	Draw Fast2DLine (start from vertex 0)
	OneVector	Draw Fast2DLine (start from vertex1)
DrawTrap	TrapRight	Draw Right Triangle
	TrapLeft	Draw Left Triangle
DrawVertex2i DrawVertex2iP	TriangleFan	Draw Fast2DTriangle
	FlagTriangleFan	Draw Fast2DTriangle for multiple vertices random shape
DrawRectP	BltFill	Draw rectangle with one color
	ClearPolyFlag	Clear Polygon flag buffer
DrawBitmapP	BltDraw	Draw Blt
	Bitmap	Draw binary bit map pattern (character)
BltCopyP BltCopy- AlternateP	TopLeft	Blt transfer from left upper vertex
	TopRight	Blt transfer from right upper vertex
	BottomLeft	Blt transfer from left lower vertex
	BottomRight	Blt transfer from right lower vertex
LoadTextureP	LoadTexture	Load texture pattern
	LoadTILE	Load tile pattern
BltTextureP	LoadTexture	Load texture pattern from Local Memory
	LoadTILE	Load tile pattern from Local Memory

Type Field Code Table

Type	Code
DrawPixel	0000_0000
DrawPixelZ	0000_0001
DrawLine	0000_0010
DrawLine2i	0000_0011
DrawLine2iP	0000_0100
DrawTrap	0000_0101
DrawVertex2i	0000_0110
DrawVertex2iP	0000_0111
DrawRectP	0000_1001
DrawBitmapP	0000_1011
BitCopyP	0000_1101
BitCopyAlternateP	0000_1111
LoadTextureP	0001_0001
BltTextureP	0001_0011
SetVertex2i	0111_0000
SetVertex2iP	0111_0001
Draw	1111_0000
SetRegister	1111_0001
Sync	1111_1100
Interrupt	1111_1101
Nop	1111_1111

Command Code Table (1)

Command	Code
Pixel	000_00000
PixelZ	000_00001
Xvector	001_00000
Yvector	001_00001
XvectorNoEnd	001_00010
YvectorNoEnd	001_00011
XvectorBlpClear	001_00100
YvectorBlpClear	001_00101
XvectorNoEndBlpClear	001_00110
YvectorNoEndBlpClear	001_00111
AntiXvector	001_01000
AntiYvector	001_01001
AntiXvectorNoEnd	001_01010
AntiYvectorNoEnd	001_01011
AntiXvectorBlpClear	001_01100
AntiYvectorBlpClear	001_01101
AntiXvectorNoEndBlpClear	001_01110
AntiYvectorNoEndBlpClear	001_01111
ZeroVector	001_10000
Onevector	001_10001
ZeroVectorNoEnd	001_10010
OnevectorNoEnd	001_10011
ZeroVectorBlpClear	001_10100
OnevectorBlpClear	001_10101
ZeroVectorNoEndBlpClear	001_10110
OnevectorNoEndBlpClear	001_10111
AntiZeroVector	001_11000
AntiOnevector	001_11001
AntiZeroVectorNoEnd	001_11010
AntiOnevectorNoEnd	001_11011
AntiZeroVectorBlpClear	001_11100
AntiOnevectorBlpClear	001_11101
AntiZeroVectorNoEndBlpClear	001_11110
AntiOnevectorNoEndBlpClear	001_11111

Command Code Table (2)

Command	Code
BltFill	010_00001
BltDraw	010_00010
Bitmap	010_00011
TopLeft	010_00100
TopRight	010_00101
BottomLeft	010_00110
BottomRight	010_00111
LoadTexture	010_01000
LoadTILE	010_01001
TrapRight	011_00000
TrapLeft	011_00001
TriangleFan	011_00010
FlagTriangleFan	011_00011
Flush_FB	110_00001
Flush_Z	110_00010
PolygonBegin	111_00000
PolygonEnd	111_00001
ClearPolyFlag	111_00010
Normal	111_11111

9.3.2 Details of Rendering Commands

All parameters belonging to their command are set in correlated registers. The definition of each parameter is figured out in the section of each command description.

Nop (Format1)

31 – 24	23 – 16	15 – 0
Nop	Reserved	Reserved

No operation

Interrupt (Format1)

31 – 24	23 – 16	15 – 0
Interrupt	Reserved	Reserved

Generates interrupt request to host CPU

Sync (Format9)

31 – 24	23 – 16	15 – 4	3 – 0
Sleep	Reserved	Reserved	flag

Suspends all subsequent display list operations until event specified in Flag field detected

Flag:

Bit number	4	3	2	1	0
Bit field name	Reserved	Reserved	Reserved	Reserved	VBLANK

Bit0 VBLANK
VBLANK Synchronization
0 No operation
1 Wait for VSYNC detection

SetRegister (Format2)

31 – 24	23 – 16	15 – 0
SetRegister	Count	Address
(Val 0)		
(Val 1)		
(Val n)		

Sets data at consecutive registers

Count: Data word count (in double-word unit)

Address: Register address

SetVertex2i (Format8)

31 – 24	23 – 16	15 – 4	3 – 2	1 – 0
SetVertex2i	Command	Reserved	flag	vertex
Xdc				
Ydc				

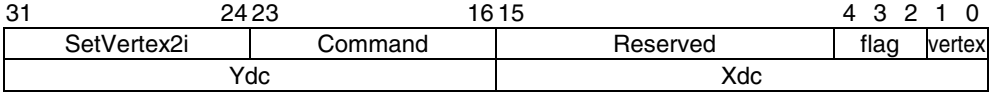
Sets vertices data for Fast2DLine or Fast2DTriangle command at registers

Commands:

Normal	Set vertex data (X, Y).
PolygonBegin	Start calculation of circumscribed rectangle for random shape to be drawn. Calculate vertices of rectangle including all vertices of random shape defined between PolygonBegin and PolygonEnd.

Flag: Not used

SetVertex2iP (Format8)



Sets vertices data for Fast2DLine or Fast2DTriangle command to registers

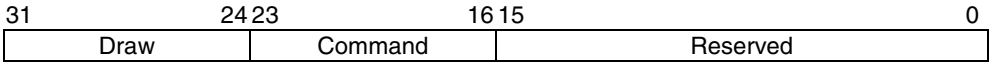
Only the packed integer format can be used specify these vertices.

Command:

Normal	Set vertices data.
PolygonBegin	Start calculation of circumscribed rectangle of random shape to be drawn. Calculate vertices of rectangle including all vertices of random shape defined between PolygonBegin and PolygonEnd.

Flag: Not used

Draw (Format5)



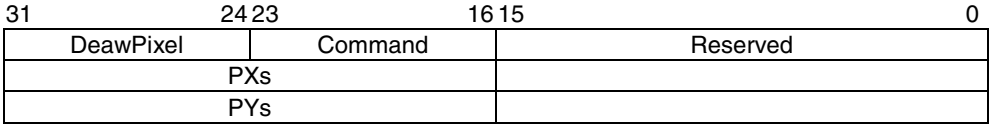
Executes draw command

All parameters required at execution of a draw command must be set at their appropriate registers.

Commands:

PolygonEnd	Draw random shape of multiple vertices. Fill random shape with color according to flags generated by FlagTriangleFan command and information of circumscribed rectangle generated by PolygonBegin command.
Flush_FB	This command flushes drawing data in the drawing pipeline into the graphics memory. Place this command at the end of the display list.
Flush_Z	This command flushes Z-value data in the drawing pipeline into the graphics memory. When using the Z buffer, place this command together with the Flush_FB command at the end of the display list.

DrawPixel (Format5)

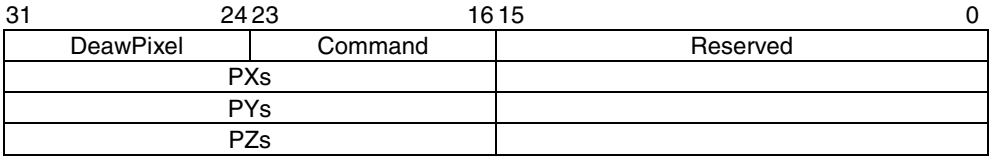


Plots pixel

Command:

Pixel	Plot pixel (without Z value).

DrawPixelZ (Format5)



Plots 3D pixel

Command:

PixelZ	Plot pixel (with Z value).

DrawLine (Format5)

31 24	23 16	15 0
DrawLine	Command	Reserved
LPN		
LXs		
LXde		
LYs		
LYde		

Draws line

Start drawing after setting all parameters at line draw registers.

Commands:

Xvector	Draw line (principal axis X).
Yvector	Draw line (principal axis Y).
XvectorNoEnd	Draw line without end point (principal axis Y).
YvectorNoEnd	Draw line without end point (principal axis Y).
XvectorBlpClear	Draw line (principal axis X). Prior to drawing, clear reference position of broken line pattern.
YvectorBlpClear	Draw a line (principal axis Y) Prior to drawing, clear reference position of broken line pattern.
XvectorNoEndBlpClear	Draw line without end point (principal axis X). Prior to drawing, clear reference position of broken line pattern.
YvectorNoEndBlpClear	Draw line without end point (principal axis Y). Prior to drawing, clear reference position of broken line pattern.
AntiXvector	Draw anti-alias line (principal axis X).
AntiYvector	Draw anti-alias line (principal axis Y).
AntiXvectorNoEnd	Draw anti-alias line without end point (principal axis Y).
AntiYvectorNoEnd	Draw anti-alias line without end point (principal axis Y).
AntiXvectorBlpClear	Draw anti-alias line (principal axis X). Prior to drawing, clear reference position of broken line pattern.
AntiYvectorBlpClear	Draw anti-alias line (principal axis Y). Prior to drawing, clear reference position of broken line pattern.
AntiXvectorNoEndBlpClear	Draw anti-alias line without end point (principal axis X). Prior to drawing, clear reference position of broken line pattern.
AntiYvectorNoEndBlpClear	Draw anti-alias line without end point (principal axis Y). Prior to drawing, clear reference position of broken line pattern

DrawLine2i (Format7)

31 24	23 16	15	0
DrawLine2i	Command	Reserved	vertex
LFXs		0	
LFYs		0	

Draws Fast2Dline

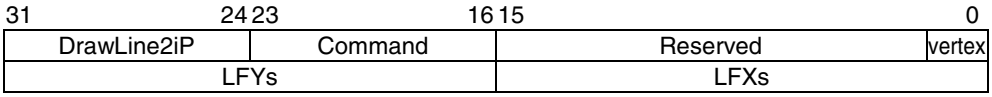
Start drawing after setting parameters at the Fast2DLIne draw registers. Integer data can only be used for vertices.

Commands:

ZeroVector	Draw line from vertex 0 to vertex 1.
OneVector	Draw line from vertex 1 to vertex 0.
ZeroVectorNoEnd	Draw line from vertex 0 to vertex 1 without end point.

OneVectorNoEnd	Draw line from vertex 1 to vertex 0 without end point.
ZeroVectorBlpClear	Draw line from vertex 0 to vertex 1. (Prior drawing, clear reference position of broken line pattern.)
OneVectorBlpClear	Draw line from vertex 1 to vertex 0. (Prior to drawing, clear reference position of broken line pattern.)
ZeroVectorNoEndBlpClear	Draw line from vertex 0 to vertex 1 without end point. (Prior to draw, clear reference position of broken line pattern.
OneVectorNoEndBlpClear	Draw line from vertex 1 to vertex 0 without end point. (Prior to drawing, clear reference position of broken line pattern.)
AntiZeroVector	Draw anti-alias line from vertex 0 to vertex 1.
AntiOneVector	Draw anti-alias line from vertex 1 to vertex 0.
AntiZeroVectorNoEnd	Draw anti-alias line from vertex 0 to vertex 1 without end point.
AntiOneVectorNoEnd	Draw anti-alias line from vertex 1 to vertex 0 without end point.
AntiZeroVectorBlpClear	Draw anti-alias line from vertex 0 to vertex 1. (Prior to drawing, clear reference position of broken line pattern.)
AntiOneVectorBlpClear	Draw anti-alias line from vertex 1 to vertex 0. (Prior to drawing, clear reference position of broken line pattern.)
AntiZeroVectorNoEndBlpClear	Draw anti-alias line from vertex 0 to vertex 1 without end point. (Prior to drawing, clear reference position of broken line pattern.)
AntiOneVectorNoEndBlpClear	Draw anti-alias line from vertex 1 to vertex 0 without end point. (Prior to drawing, clear reference position of broken line pattern.)

DrawLine2iP (Format7)



Draws Fast2Dline

Start drawing after setting parameters at Fast2DLIne draw registers. Only packed integer data can be used for vertices.

Commands:

ZeroVector	Draw line from vertex 0 to vertex 1.
OneVector	Draw line from vertex 1 to vertex 0.
ZeroVectorNoEnd	Draw line from vertex 0 to vertex 1 without end point.
OneVectorNoEnd	Draw line from vertex 1 to vertex 0 without end point.
ZeroVectorBlpClear	Draw line from vertex 0 to vertex 1. (Prior to drawing, clear the reference position of the broken line pattern.)
OneVectorBlpClear	Draw line from vertex 1 to vertex 0. (Prior to drawing, clear reference position of broken line pattern.)
ZeroVectorNoEndBlpClear	Draw line from vertex 0 to vertex 1 without end point.

	(Prior to drawing, clear reference position of broken line pattern.)
OneVectorNoEndBlpClear	Draw line from vertex 1 to vertex 0 without end point. (Prior to drawing, clear reference position of broken line pattern.)
AntiZeroVector	Draw anti-alias line from vertex 0 to vertex 1.
AntiOneVector	Draw anti-alias line from vertex 1 to vertex 0.
AntiZeroVectorNoEnd	Draw anti-alias line from vertex 0 to vertex 1 without end point.
AntiOneVectorNoEnd	Draw anti-alias line from vertex 1 to vertex 0 without end point.
AntiZeroVectorBlpClear	Draw anti-alias line from vertex 0 to vertex 1. (Prior to drawing, clear reference position of broken line pattern.)
AntiOneVectorBlpClear	Draw anti-alias line from vertex 1 to vertex 0. (Prior to drawing, clear reference position of broken line pattern.)
AntiZeroVectorNoEndBlpClear	Draw anti-alias line from vertex 0 to vertex 1 without end point. (Prior to drawing, clear reference position of broken line pattern.)
AntiOneVectorNoEndBlpClear	Draw anti-alias line from vertex 1 to vertex 0 without end point. (Prior to drawing, clear reference position of broken line pattern.)

DrawTrap (Format5)

31 24	23 16	15 0
DrawTrap	Command	Reserved
Ys		0
Xs		
DXdy		
XUs		
DXUdy		
XLs		
DXLdy		
USN		0
LSN		0

Draws Triangle

Operation is started after setting all the related parameters at the Plane Draw registers.

Commands:

TrapRight	Draw Right Triangle.
TrapLeft	Draw Left Triangle.

DrawVertex2i (Format7)

31 24	23 16	15	0
DrawVertex2i	Command	Reserved	vertex
Xdc		0	
Ydc		0	

Draws Fast2Dtriangle

Operation is started after setting all the related parameters at the Plane Draw registers.

Commands:

TriangleFan	Draw Fast2Dtriangle.

FlagTriangleFan Draw Fast2DTriangle for random shape with multiple vertices.

DrawVertex2iP (Format7)

31 24	23 16	15 0
DrawVertex2iP	Command	Reserved vertex
Ydc		Xdc

Draws Fast2Dtriangle

Operation is started after setting all the related parameters at Plane Draw registers

Only the packed integer format can be used for vertex coordinates.

Commands:

TriangleFan Draw Fast2Dtriangle.

FlagTriangleFan Draw Fast2DTriangle for random shape with multiple vertices.

DrawRectP (Format5)

31 24	23 16	15 0
DrawRectP	Command	Reserved
RYs		RXs
RsizeY		RsizeX

Fills rectangle

The designated rectangle is filled with the current color after setting all the related parameters at the rectangle registers.

Commands:

BltFill Fill rectangle with current color (single).

ClearPolyFlag Fill polygon flag field with 0. The size is defined in RsizeX,Y.

DrawBitmapP (Format6)

31 24	23 16	15 0
DrawBitmapP	Command	Count
RYs		RXs
RsizeY		RsizeX
(Pattern 0)		
(Pattern 1)		
…		
(Pattern n)		

Draws rectangle

Commands:

BltDraw Draw rectangle of 8 bits/pixel or 16 bits/pixel.

DrawBitmap Draw binary bitmap character pattern. Bit0 is drawn in transparent or background color, and bit1 is drawn in foreground olor. Background color is specified in the BC register, and foreground color is specified in the FC register.

BltCopyP (Format5)

31 24	23 16	15 0
BltCopyP	Command	Reserved
SRYs		SRXs
DRYs		DRXs
BRsizeY		BRsizeX

Copies rectangle pattern within one drawing frame

Commands:

TopLeft Start BitBlt transfer from top left vertex.

TopRight	Start BitBlt transfer from top right vertex.
BottomLeft	Start BitBlt transfer from bottom left vertex.
BottomRight	Start BitBlt transfer from bottom right vertex.

BltCopyAlternateP (Format5)

31 24	23 16	15 0
BltCopyAlternateP	Command	Reserved
SADDR		
SStride		
SRYs		SRXs
DADDR		
DStride		
DRYs		DRXs
BRsizeY		BRsizeX

Copies rectangle between two separate drawing frames

Commands:

TopLeft	Start BitBlt transfer from top left vertex.

LoadTextureP (Format6)

31 24	23 16	15 0
LoadTextureP	Command	Count
(Pattern 0)		
(Pattern 1)		
…		
(Pattern n)		

Loads texture or tile pattern into internal texture buffer memory

Supply a texture pattern to the internal texture buffer according to the current pattern size (TXS/TIS) and offset address (XBO).

Commands:

LoadTexture	Load texture pattern to internal texture buffer.
LoadTile	Load tile pattern to internal texture buffer.

BltTextureP (Format5)

31 24	23 16	15 0
BltTextureP	Command	Reserved
SrcADDR		
SrcStride		
SrcRectYs		SrcRectXs
BRsizeY		BRsizeX
DestOffset		

Loads texture or tile pattern into internal texture buffer memory from Graphics Memory

Supply a texture pattern to the internal texture buffer according to current pattern size (TXS/TIS) and offset address (XBO).

For DestOffset, specify the word-aligned byte address (16 bits) (bit 0 is always 0).

Commands:

LoadTexture	Load texture pattern into internal texture buffer.
LoadTile	Load tile pattern into internal texture buffer.

10. Registers

10.1 Description

All the terms in this chapter are explained below:

1. Register address
   Indicates address of register
2. Bit number
   #Indicates bit number
3. Bit field name
   Indicates name of each bit field in register
4. R/W
   Indicates access attribute (Read/Write) of each field
   Each sign shown in this section means the following:

   R0 0 always read at read. Write access is Don't care.

   W0 Only 0 can be written

   R Enable read

   RX Enable read (read values undefined)

   RW Enable read and write any data

   RW0 Enable read and write 0

5. Default
   This section shows the reset defaults for each bit field.

10.1.1 Host Interface Registers

DTC (DMA Transfer Count)

Register address	HostBaseAddress + 00_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								DTC																							
R/W	R0								RW																							
Default	0								Don't care																							

DTCR is a 32-bit wide register to set the DMA data transfer count to either one long-word (32 bits) or eight long-word (32 bytes) units. This register is read/write enabled. When 1h is set, one data unit is transferred by DMA. However, when 0h is set, it indicates the maximum transfer data count and 16M (16,777,216) data units are transferred. After DMA transfer is started, the register value cannot be overwritten until DMA transfer is completed.

Note: In the V832 mode, no setting is required for this register.

DSU (DMA Set Up)

Register address	HostBaseAddress + 04H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved					DAM	DBM	DW
R/W	R0					RW	RW	RW
Default	0					0	0	0

Bit0 DW(DMA Word)

Sets DMA transfer unit

0: 1 long words (32 bytes) per DMA transaction

1: 8 long word (32 bits) per DMA transaction (only SH4)

Bit1 DBM (DMA Bus request Mode)

Selects DREQ mode used when performing DMA transfer in dual-address mode

0: DREQ is irrelevant to the cycle steal mode or burst mode, and is not negated during DMA transfer.

1: DREQ is irrelevant to the cycle steal mode or burst mode, and is negated when ORCHID cannot receive data (that is, when Ready cannot be returned immediately). When ORCHID is ready to receive data, DREQ is reasserted (When DMA transfer is performed in the single-address mode, DREQ is controlled automatically).

Bit2 DAM(DMA Address Mode)

Sets DMA addressing mode

0: Dual address mode

1: Single address mode (SH4 only)

DRM (DMA Request Mask)

Register address	HostBaseAddress + 05H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							DRM
R/W	R0							RW
Default	0							0

This register controls the DMA request to the host CPU. Setting 1 at this register tentatively masks the DMA request from the ORCHID. The DMA request is restarted when 0 is set at this register.

DST (DMA STatus)

Register address	HostBaseAddress + 06H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							DST
R/W	R0							R
Default	0							0

This register indicates the DMA status. DST is set to 1 during DMA transfer. This state is cleared to 0 when the DMA transfer is completed.

DTS (DMA Transfer Stop)

Register address	HostBaseAddress + 08H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							DTS
R/W	R0							RW
Default	0							0

This register suspends DMA transfer. An ongoing DMA transfer is suspended by setting DTS to 1.

LTS (display Transfer Stop)

Register address	HostBaseAddress + 09H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							LTS
R/W	R0							RW
Default	0							0

This register suspends DisplayList transfer.

Ongoing DisplayList transfer is suspended by setting LTS to 1.

LSTA (displayList transfer STAtus)

Register address	HostBaseAddress + 10H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							LSTA
R/W	R0							R
Default	0							0

This register indicates the DisplayList transfer status from Graphics Memory. LSTA is set to 1 while DisplayList transfer is in progress. This status is cleared to 0 when DisplayList transfer is completed

DRQ (DMA ReQquest)

Register address	HostBaseAddress + 18H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							DRQ
R/W	R0							RW1
Default	0							0

Starts sending external DMA request signal

DMA transfer using the external DMA request handshake is triggered by setting DRQ to “1”. The external DREQ signal is not asserted when DMA is masked by the DRM register. This register cannot be set to “0”. When DMA transfer is completed, this status is cleared automatically to 0.

IST (Interrupt STatus)

Register address	HostBaseAddress + 20H																										
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved														ISR		Reserved											ISR			Resv	
R/W	R0														RW0		R0											RW0			R0	
Default	0														0		0											0			0	

This register indicates the current interrupt status. When an interrupt request to the host CPU is asserted, this register displays 1. The interrupt status is cleared by setting 0 at this register.

Bit 0	CERR (Command Error Flag)
	Indicates drawing command execution error interrupt
Bit 1	CEND (Command END)
	Indicates drawing command end interrupt
Bit 2	VSYNC (Vertical Sync.)
	VSYNC detection interrupt
Bit 3	FSYNC (Frame Sync.)
	Indicates frame synchronization interrupt

Bit 4 SYNCERR (Sync. Error)

Indicates external synchronization error interrupt

Bit 16 Reserved

Bit 17 Reserved

MASK (Interrupt MASK)

Register address	HostBaseAddress + 24H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved														IMASK		Reserved											IMASK			Resv	
R/W	R0														RW		R0											RW			R0	
Default	0														0		0											0			0	

This register masks interrupt requests. When the flag is set to 1, the respective event is masked so that no interrupt request is asserted to the host CPU when an event occurs.

Bit 0 CERR (Command Error Flag)

Indicates drawing command execution error interrupt

Bit 1 CEND (Command END)

Indicates drawing command end interrupt

Bit 2 VSYNCM (VerticalSync.Interrupt Mask)

Masks vertical synchronous interrupt

Bit 3 FSYNCH (Frame Sync. Interrupt Mask)

Makes frame synchronous interrupt

Bit 4 SYNCERRM Sync Error Mask

Masks external synchronous error interrupt

Bit 16 Reserved

Bit 17 Reserved

SRST (Software ReSeT)

Register address	HostBaseAddress + 2CH							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							SRST
R/W	R0							W1
Default	0							0

This register controls software reset. When 1 is set at this register, a software reset is issued.

LSA (displayList Source Address)

Register address	HostBaseAddress + 40H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								LSA																							
R/W	R0								RW																							R0
Default	0								Don't care																							0

This register sets the DisplayList transfer source address. When DisplayList is transferred from Graphics Memory, set the List start address. Since the lowest two bits of this register are always set to 0, DisplayList must be 4-byte aligned. The contents set at this register do not change until another value is set.

LCO (displayList Count)

Register address	HostBaseAddress + 44H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								LCO																							
R/W	R0								RW																							
Default	0								Don't care																							

This register sets the DisplayList. transfer word count. When 1 is set, 1-word data is transferred. When 0 is set, it is considered to be the maximum number and 16M (16,777,216) words of data are transferred. The contents set at this register do not change until another value is set.

LREQ (displayList transfer REQuest)

Register address	HostBaseAddress + 48H							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							LREQ
R/W	R0							RW1
Default	0							0

This register triggers DisplayList transfer from the Graphics Memory. Transfer is started by setting LREQ to 1. DisplayList. The DisplayList is transferred from the Graphics Memory to the internal display list FIFO. Access to the display list FIFO by the CPU or DMA is prohibited while this transfer is in progress.

10.1.2 Graphics Memory Interface Registers

MMR (Memory I/F Mode Register)

Register address	HostBaseAddress + FFFCh																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field	*1	tWR	Reserve		ID	*1	TRRD		TRC			TRP		TRAS			TRCD		LOWD		RTS				SAW		ASW	CL			
/	RW	RW	R		W1	R	RW		RW			RW		RW			RW		RW		RW				RW		RW	RW			
Initial value	0	0	invalid		0	0	11		1001			11		110			11		10		0111				10		0	011			

This register controls the graphics memory interface mode setting. An appropriate value must be set at this register after reset (even if the default value is used). This register is not initialized by a software reset.

Bit2-0 CL (CAS Latency)

Set CAS latency. Write same value to mode register of SDRAM.

011 CL3

010 CL2

Other than those above is prohibited to set.

Bit3 ASW (Attached SDRAM bit Width)

Set bit width of data bus (memory bus width mode).

1 64bit

0 32bit

Bit6-4 SAW (SDRAM Address Width)

Set bit width of SDRAM address.

111 14bit BANK 2bit ROW 12bit COL 9bit SDRAM

110 14bit BANK 2bit ROW 12bit COL 8bit SDRAM

101 13bit BANK 2bit ROW 12bit COL 8bit SDRAM

100 12bit BANK 1bit ROW 11bit COL 8bit FCRAM

Other than those above is prohibited to set.

Bit9-7 RTS (Refresh Timing Setting)

Set interval of refresh.

000 384 per internal clock

111 1552 per internal clock

001 110 Set interval of 64 384 per 64 n internal clock.

Bit11-10 LOWD

Set clock from last data output to write command.

10 2clock

Other than those above is prohibited to set.

Bit13-12 TRCD

Set waiting time from bank active to CAS in clock.

00 4clock

11 3clock

10 2clock

01 1clock

Bit16-14 TRAS

Set least time of bank active in clock.

111 7clock

110 6clock

101 5clock

100 4clock

011 3clock

010 2clock

Other than those above is prohibited to set.

Bit18-17 TRP

Set waiting time from free charge to bank active in clock.

00 4clock

11 3clock

10 2clock

01 1clock

Bit22-19 TRC

Set waiting time from refresh to bank active in clock.

1010 10clock

1001 9clock

1000 8clock

0111 7clock

0110 6clock

0101 5clock

0100 4clock

0011 3clock

Other than those above is prohibited to set.

Bit24-23 TRRD

Set waiting time from bank active to next in clock.

11 3clock

10 2clock

01 1clock

Bit26 ORCHID ID

Bit30 TWR

Set WRITE recovery time (interval time from WRITE command to READ or Precharge command).

1 1clock

0 0clock

10.1.3 Display Control Register

DCM (Display Control Mode)

Register address	Display Base Address + 00H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	CKS	DCS	Reserve	SC					EEQ	ODE	EO	Reserve	SF	ESY	SYNC	
R/W	RW	RW	RX	RW					RW	RW	RW	RX	RW	RW	RW	
Default	0	0	X	11110					0	0	0	X	0	0	00	

This register controls the display mode. It is not initialized by software reset.

Bits 1-0 SYNC (Synchronize)

Set synchronization mode

X0 Non-interlace mode

10 Interlace mode

11 Interlace video mode

Bit 2 ESY (External Synchronize)

Sets external synchronization mode

0: Disable

1: Enable

Bit 3 SF (Synchronize signal output format)

Sets active level of synchronization (VSYNC, HSYNC) signals

0: Low active

1: High active

Bit 5 EO (Even/Odd signal mode)

Defines EO signal output format

0: Low level output at even frame, High level output at odd frame

1: High level output at even frame, Low level output at odd frame

Bit 6 ODE (Output Display Enable)

Sets operating mode of CCYNC pin

0: Outputs normal CCYNC signal

1: Outputs Display Enable signal

When the Display Enable signal is output, this bit goes high. When the Display Enable signal is not output, this bit goes low.

Bit 7 EEQ (Enable Equalizing pulse)

Sets CCYNC signal mode

0: Does not insert equivalent pulse into CCYNC signal

1: Inserts equivalent pulse into CCYNC signal

Bits 12-8 SC (Scaling)

Define pre-scaling ratio to generate dot clock

00000 No pre-scaling

00001 1/2

00010 1/3

11110 1/31 (default)

11111 1/32

Bit 14 DCS (Display Clock Select)

Selects clock signal output from DCLKO pin

0: Outputs dot clock set in SC field

1: Divides clock signal input from CLK pin by 1/4 and outputs it

Bit 15 CKS (Clock Source)

Selects source clock

0: Internal PLL output clock

1: DCLKI input

DCE (Display Controller Enable)

Register address	Display Base Address + 02_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DEN	Reserved											BE	ME	WE	CE
R/W	RW	R0											RW	RW	RW	RW

Default	0	0	0	0	0	0

This register controls the video signal output and enables display of each layer.

Bit 0 CE (C layer Enable)

Enables C layer display

0: Does not display C layer

1: Displays C layer

Bit 1 WE (W layer Enable)

Enables W layer display

0: Does not display W layer

1: Displays W layer

Bit 2 ME (M layer Enable)

Enables M layer display

0: Does not display M layer

1: Displays M layer

Bit 3 BE (B layer Enable)

Enables B layer display

0: Does not display B layer

1: Displays B layer

Bit 15 DEN (Display Enable)

Enables display

0: Does not output display signal

1: Outputs display signal

HTP (Horizontal Total Pixels)

Register address	Display Base Address + 06_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				HTP											
R/W	R0				RW											
Default	0				Don't care											

This register controls the total pixel count. Setting + 1 is the total pixel count.

HDP (Horizontal Display Period)

Register address	Display Base Address + 08_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				HDP											
R/W	R0				RW											
Default	0				Don't care											

This register controls the total horizontal display period in pixel clock units. Setting + 1 is the pixel count for the display period.

HDB (Horizontal Display Boundary)

Register address	Display Base Address + $0A_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				HDB											
R/W	R0				RW											
Default	0				Don't care											

This register controls the display period of the left partition in pixel raster units Setting + 1 is the pixel count for the display period of the left partition. When the screen is not partitioned into right and left before display, set the same value as HDP.

HSP (Horizontal Synchronize pulse Position)

Register address	Display Base Address + $0C_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				HSP											
R/W	R0				RW											
Default	0				Don't care											

This register controls the HSYNC pulse position in pixel clock unit. When the clock count since the start of the display period reaches Setting + 1, the horizontal synchronization signal is asserted.

HSW (Horizontal Synchronize pulse Width)

Register address	Display Base Address + $0E_H$							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved		HSW					
R/W	R0		RW					
Default	0		Don't care					

This register controls the HSYNC pulse width in pixel-clock units. Setting + 1 is the pulse width clock count.

VSW (Vertical Synchronize pulse Width)

Register address	Display Base Address + $0F_H$							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved		VSW					
R/W	R0		RW					
Default	0		Don't care					

This register controls the VSYNC pulse width in raster units. Setting + 1 is the pulse width raster count.

VTR (Vertical Total Rasters)

Register address	DisplayBaseAddress + 12_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				VTR											
R/W	R0				RW											
Default	0				Don't care											

This register controls the total raster count. Setting + 1 is the total raster count. For the interlace display, Setting + 1.5 is the total raster count for 1 field; 2 × setting + 3 is the total raster count for 1 frame (see Section 8.3.2).

VSP (Vertical Synchronize pulse Position)

Register address	DisplayBaseAddress + 14_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				VSP											
R/W	R0				RW											
Default	0				Don't care											

This register controls the VSYNC pulse position in raster units. The vertical synchronization pulse is asserted starting at the Setting + 1-th raster relative to the display start raster.

VDP (Vertical Display Period)

Register address	DisplayBaseAddress + 16_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				VDP											
R/W	R0				RW											
Default	0				Don't care											

This register controls the vertical display period in raster unit. Setting + 1 is the count of rasters to be displayed.

WX (Window position X)

Register address	DisplayBaseAddress + 18_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				WX											
R/W	R0				RW											
Default	0				Don't care											

This register controls the horizontal position of the left edge of the Window layer.

WY (Window position Y)

Register address	DisplayBaseAddress + $1A_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				WY											
R/W	R0				RW											
Default	0				Don't care											

This register controls the vertical position of the top edge of the Window layer.

WW (Window Width)

Register address	DisplayBaseAddress + $1C_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				WW											
R/W	R0				RW											
Default	0				Don't care											

This register controls the horizontal size (pixel count) of the Window layer. Do not specify 0.

WH (Window Height)

Register address	DisplayBaseAddress + $1E_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				WH											
R/W	R0				RW											
Default	0				Don't care											

This register controls the vertical height (raster count) of the Window layer. Setting + 1 is the height.

CM (C layer Mode)

Register address	DisplayBaseAddress + 20_H																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	CC	Reserve		Reserve							CW						Reserve				CH											
R/W	RW	R0		R0							RW						R0				RW											
Default	0	0		0							Don't care						0				Don't care											

Bits 11-0 CH (C layer Height)

Set height of Console layer logical frame size in raster units. Setting + 1 is the height.

Bits 23-16 CW (C layer memory Width)

Set width of Console layer logical frame size in 64-byte units

Bit 31 CC (C layer Color mode)

Sets color mode used for Console layer

0: Indirect color mode (8 bits/pixel)

1: Direct color mode (16 bits/pixel)

COA(C layer Origin Address)

Register address	DisplayBaseAddress + 24_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						COA																									
R/W	R0						RW																								R0	
Default	0						Don't care																								0000	

This register controls the base address of the logical frame of the Console layer. Since the lowest 4 bits are fixed to 0, this address is 16-byte aligned.

CDA (C layer Display Address)

Register address	DisplayBaseAddress + 28_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						CDA																									
R/W	R0						RW																									
Default	0						Don't care																									

This register controls the base address of the display field of the Console layer. When the direct color mode is used, the LSB is fixed to 0 and this address is 2-byte aligned.

CDX (C layer Display position X)

Register address	DisplayBaseAddress + $2C_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CDX											
R/W	R0				RW											
Default	0				Don't care											

Set the display start position (X coordinate) for the C layer in pixel units relative to the origin of the logical frame.

CDY (C layer Display position Y)

Register address	DisplayBaseAddress + $2E_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CDY											
R/W	R0				RW											
Default	0				Don't care											

Set the display start position (Y coordinate) for the C layer in pixel units relative to the origin of the logical frame.

WM (W layer Mode)

Register address	DisplayBaseAddress + 30_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	WC	WYC	WCS	WIM	Reserve				WW								Reserve															
R/W	RW	RW	RW	RW	R0				RW								R0															
Default	0	0	0	0	0				Don't care								0															

Bit 23-16 WW (W layer memory Width)

Set width of Window layer logical frame size in 64-byte units.

Bit 28 WIM (W layer Interlace Mode)

Sets video capture operation mode when WCS in capture mode

0: Normal mode

1: For non-interlace display, displays captured video graphics in WEAVE mode

For interlace and video display, buffers are managed in frame units (pair of odd field and even field).

Bit 29 WCS (W layer Capture Synchronize)

0: Normal mode

1: Capture mode

Bit 30 WYC (W layer YC mode)

Sets color mode of W layer

The YC mode must be set for capturing videos.

0: RGB mode

1: YC mode

Bit 31 WC (W layer Color mode)

Sets color mode for W layer

0: Indirect color (8 bits/pixel) mode

1: Direct color (16 bits/pixel) mode

WOA (W layer Origin Address)

Register address	DisplayBaseAddress + 34_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						WOA																									
R/W	R0						RW																							R0		
Default	0						Don't care																							0000		

This register controls the base address of the logical frame of the Window layer. Since the lowest 4-bits are fixed to 0, this address is 16-byte aligned.

WDA (W layer Display Address)

Register address	DisplayBaseAddress + 38_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						WDA																									
R/W	R0						RW																									
Default	0						Don't care																									

This register controls the base address of the display field of the Window layer. Since only the direct color mode is applicable to the Window layer, the LSB is fixed to 0 and this address is 2-byte aligned.

MLM (ML layer Mode)

Register address	DisplayBaseAddress + 40_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MLC	MLFLP	Reserve								MLW						Reserve				MLH											
R/W	RW	R0	R0								RW						R0				RW											
Default	0	0	0								Don't care						0				Don't care											

Bits 11-0 MLH (ML layer Height)

Set height of Middle Left (ML) layer logical frame size in raster units. Setting + 1 is the height.

Bits 23-16 MLW (ML layer memory Width)

Set width of Middle Left (ML) layer logical frame size in 64-byte units

Bits 30-29 MLFLP (ML layer Flip mode)

Set flipping mode for Middle Left (ML) layer

00 Display frame 0

01 Display frame 1

10 Switch frame 0 and 1 back and forth

11 Reserved

Bit 31 MLC (ML layer Color mode)

Sets color mode for Middle Left (ML) layer

0: Indirect color mode (8 bits/pixel)

1: Direct color mode (16 bits/pixel)

MLOA0 (ML layer Origin Address 0)

Register address	DisplayBaseAddress + 44_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						MLOA0																									
R/W	R0						RW																								R0	
Default	0						Don't care																								0000	

This register controls the base address of the logical frame (frame0) of the Middle Left (ML) layer. Since the lowest 4 bits are fixed to 0, this address is 16-byte aligned.

MLDA0 (ML layer Display Address 0)

Register address	DisplayBaseAddress + 48_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						MLDA0																									
R/W	R0						RW																									
Default	0						Don't care																									

This register controls the base address of the Middle Left (ML) layer display field in frame0. When the direct color mode is used, the LSB is fixed to 0 and this address is 2-byte aligned.

MLOA1 (ML layer Origin Address 1)

Register address	DisplayBaseAddress + $4C_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						MLOA1																									
R/W	R0						RW																								R0	
Default	0						Don't care																								0000	

This register controls the base address of the logical frame (frame1) of the Middle Left (ML) layer. Since the lowest 4-bits are fixed to 0, this address is 16-byte aligned.

MLDA1 (ML layer Display Address 1)

Register address	DisplayBaseAddress + 50_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						MLDA1																									
R/W	R0						RW																									
Default	0						Don't care																									

This register controls the base address of the Middle Left (ML) layer display field in frame1. When the direct color mode is used, the LSB is fixed to 0 and this address is 2-byte aligned.

MLDX (ML layer Display position X)

Register address	DisplayBaseAddress + 54_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				MLDX											
R/W	R0				RW											
Default	0				Don't care											

Set the display start position (X coordinate) for the ML layer in pixel units relative to the origin of the logical frame.

MLDY (ML layer Display position Y)

Register address	DisplayBaseAddress + 56_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				MLDY											
R/W	R0				RW											
Default	0				Don't care											

Set the display start position (Y coordinate) for the ML layer in pixel units relative to the origin of the logical frame.

MRM (MR layer Mode)

Register address	DisplayBaseAddress + 58_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MRC	MRFLP	Reserve				MRW										Reserve				MRH											
R/W	RW	R0	R0				RW										R0				RW											
Default	0	0	0				Don't care										0				Don't care											

Bits 11-0 MRH (MR layer Height)

Set height of Middle Right (MR) layer logical frame size in raster units. Setting + 1 is the height.

Bits 23-16 MRW (MR layer memory Width)

Set width of Middle Right (MR) layer logical frame size in 64-byte units

Bits 30-29 MRFLP (MR layer Flip mode)

Set flipping mode for Middle Right (MR) layer

00 Display frame 0

01 Display frame 1

10 Switch frame 0 and 1 back and forth

11 Reserved

Bit 31 MRC (MR layer Color mode)

Sets color mode for Middle Right (MR) layer

0: Indirect color mode (8 bits/pixel)

1: Direct color mode (16 bits/pixel)

MROA0 (MR layer Origin Address 0)

Register address	DisplayBaseAddress + $5C_H$																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						MROA0																									
R/W	R0						RW																							R0		
Default	0						Don't care																							0000		

This register controls the base address of the logical frame (frame0) of the Middle Right (MR) layer. Since the lowest 4 bits are fixed to 0, this address is 16-byte aligned.

MRDA0 (MR layer Display Address 0)

Register address	DisplayBaseAddress + 60_H																													
Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						MRDA0																									
R/W	R0						RW																									
Default	0						Don't care																									

This register controls the base address of the Middle Left (ML) layer display field in frame0. When the direct color mode is used, the LSB is fixed to 0 and this address is 2-byte aligned.

MROA1 (MR layer Origin Address 1)

Register address	DisplayBaseAddress + 64_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						MROA1																									
R/W	R0						RW																							R0		
Default	0						Don't care																							0000		

This register controls the base address of the logical frame (frame1) of the Middle Right (MR) layer. Since the lowest 4 bits are fixed to 0, this address is 16-byte aligned.

MRDA1 (MR layer Display Address 1)

Register address	DisplayBaseAddress + 68_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						MRDA1																									
R/W	R0						RW																									
Default	0						Don't care																									

This register controls the base address of the Middle Right (MR) layer display field in frame1. When the direct color mode is used, the LSB is fixed to 0 and this address is 2-byte aligned.

MRDX (MR layer Display position X)

Register address	DisplayBaseAddress + $6C_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				MRDX											
R/W	R0				RW											
Default	0				Don't care											

Set the display start position (X coordinate) for the MR layer in pixel units relative to the origin of the logical frame.

MRDY (MR layer Display position Y)

Register address	DisplayBaseAddress + $6E_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				MRDY											
R/W	R0				RW											
Default	0				Don't care											

Set the display start position (Y coordinate) for the MR layer in pixel units relative to the origin of the logical frame.

BLM (BL layer Mode)

Register address	DisplayBaseAddress + 70_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BLC	BLFLP		Reserve					BLW								Reserve				BLH											
R/W	RW	R0		R0					RW								R0				RW											
Default	0	0		0					Don't care								0				Don't care											

Bits 11-0 BLH (BL layer Height)

Set height of Base Left (BL) layer logical frame size in raster units. Setting + 1 is the height.

Bits 23-16 BLW (BL layer memory Width)

Set width of Base Left (BL) layer logical frame size in 64-byte units

Bits 30-29 BLFLP (BL layer Flip mode)

Set flipping mode for Base Left (BL) layer

00 Display frame 0

01 Display frame 1

10 Switch frame 0 and 1 back and forth

11 Reserved

Bit 31 BLC (BL layer Color mode)

Sets color mode for Base Left (BL) layer

0: Indirect color mode (8 bits/pixel)

1: Direct color mode (16 bits/pixel)

BLOA0 (BL layer Origin Address 0)

Register address	DisplayBaseAddress + 74_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						BLOA0																									
R/W	R0						RW																								R0	
Default	0						Don't care																								0000	

This register controls the base address of the logical frame (frame0) of the Base Left (BL) layer. Since the lowest 4 bits are fixed to 0, this address is 16-byte aligned.

BLDA0 (BL layer Display Address 0)

Register address	DisplayBaseAddress + 78_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						BLDA0																									
R/W	R0						RW																									
Default	0						Don't care																									

This register controls the base address of the Base Left (BL) layer display field in frame0. When the direct color mode is used, the LSB is fixed to 0 and this address is 2-byte aligned.

BLOA1 (BL layer Origin Address 1)

Register address	DisplayBaseAddress + $7C_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						BLOA1																									
R/W	R0						RW																								R0	
Default	0						Don't care																								0000	

This register controls the base address of the logical frame (frame1) of the Base Left (BL) layer. Since the lowest 4 bits are fixed to 0, this address is 16-byte aligned.

BLDA1 (BL layer Display Address 1)

Register address	DisplayBaseAddress + 80_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						BLDA1																									
R/W	R0						RW																									
Default	0						Don't care																									

This register controls the base address of the Base Left (BL) layer display field in frame1. When the direct color mode is used, the LSB is fixed to 0 and this address is 2-byte aligned.

BLDX (BL layer Display position X)

Register address	DisplayBaseAddress + 84_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				BLDX											
R/W	R0				RW											
Default	0				Don't care											

Set the display start position (X coordinate) for the BL layer in pixel units relative to the origin of the logical frame.

BLDY (BL layer Display position Y)

Register address	DisplayBaseAddress + 86_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				BLDY											
R/W	R0				RW											
Default	0				Don't care											

Set the display start position (Y coordinate) for the BL layer in pixel units relative to the origin of the logical frame.

BRM (BR layer Mode)

Register address	DisplayBaseAddress + 88_H																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BRC	BRFLP		Reserve							BRW						Reserve				BRH											
R/W	RW	R0		R0							RW						R0				RW											
Default	0	0		0							Don't care						0				Don't care											

Bits 11-0 BRH (BR layer Height)

Set height of Base Right (BR) layer logical frame size in raster units. Setting + 1 is the height.

Bits 23-16 BRW (BR layer memory Width)

Set width of Base Right (BR) layer logical frame size in 64-byte units

Bits 30-29 BRFLP (BR layer Flip mode)

Set flipping mode for Base Right (BR) layer

00 Display frame 0

01 Display frame 1

10 Switch frame 0 and 1 back and forth

11 Reserved

Bit 31 BRC (BR layer Color mode)

Sets color mode for Base Right (BR) layer

0: Indirect color mode (8 bits/pixel)

1: Direct color mode (16 bits/pixel)

BROA0 (BR layer Origin Address 0)

Register address	DisplayBaseAddress + $8C_H$																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						BROA0																									
R/W	R0						RW																								R0	
Default	0						Don't care																								0000	

This register controls the base address of the logical frame (frame0) of the Base Right (BR) layer. Since the lowest 4 bits are fixed to 0, this address is 16-byte aligned.

BRDA0 (BR layer Display Address 0)

Register address	DisplayBaseAddress + 90_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						BRDA0																									
R/W	R0						RW																									
Default	0						Don't care																									

This register controls the base address of the Base Right (BR) layer display field in frame0. When the direct color mode is used, the LSB is fixed to 0 and this address is 2-byte aligned.

BROA1 (BR layer Origin Address 1)

Register address	DisplayBaseAddress + 94_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						BROA1																									
R/W	R0						RW																							R0		
Default	0						Don't care																							0000		

This register controls the base address of the logical frame (frame1) of the Base Right (BR) layer. Since the lowest 4 bits are fixed to 0, this address is 16-byte aligned.

BRDA1 (BR layer Display Address 1)

Register address	DisplayBaseAddress + 98_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						BRDA1																									
R/W	R0						RW																									
Default	0						Don't care																									

This register controls the base address of Base Right (BR) layer display field in frame1. When the direct color mode is used, the LSB is fixed to 0 and this address is 2-byte aligned.

Register address	DisplayBaseAddress + $9C_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				BRDX											
R/W	R0				RW											
Default	0				Don't care											

Set the display start position (X coordinate) for the BR layer in pixel units relative to the origin of the logical frame.

BRDY (BR layer Display position Y)

Register address	DisplayBaseAddress + $9E_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				BRDY											
R/W	R0				RW											
Default	0				Don't care											

T Set the display start position (Y coordinate) for the BR layer in pixel units relative to the origin of the logical frame.

CUTC (Cursor Transparent Control)

Register address	DisplayBaseAddress + $A0_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved							CUZT	CUTC							
R/W	R0							RW	RW							
Default	0							Don't care	Don't care							

Bits 7-0 CUTC (Cursor Transparent Code)

Set transparency color code

Bit 8 CUZT (Cursor Zero Transparency)

Defines treatment of color code 0

0: Code 0 transparency color

1: Code 0 not transparency color

CPM (Cursor Priority Mode)

Register address	DisplayBaseAddress + $A2_H$							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved		CEN1	CEN0	Reserved		CUO1	CUO0
R/W	R0		RW	RW	R0		RW	RW
Default	0		0	0	0		0	0

This register controls the display priority of cursors. Cursor 0 is always prioritized to cursor 1.

Bit 0 CUO0 (Cursor Overlap 0)

Sets display priority between cursor 0 and pixels of Console layer

0: Put cursor 0 at bottom of Console layer.

1: Put cursor 0 at top of Console layer.

Bit 1 CUO1 (Cursor Overlap 1)

Sets display priority between cursor 1 and pixels of Console layer

0: Put cursor 1 at bottom of Console layer.

1: Put cursor 1 at top of Console layer.

Bit 4 CEN0 (Cursor Enable 0)

Sets display enable of cursor 0

0: Disable

1: Enable

Bit 5 CEN1 (Cursor Enable 1)

Sets display enable of cursor 1

0: Disable

1: Enable

CUOA0 (Cursor-0 Origin Address)

Register address	DisplayBaseAddress + $A4_H$																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						CUOA0																									
R/W	R0						RW																							R0		
Default	0						Don't care																							0000		

This register controls the start address of the cursor-0 pattern. Since the lowest 4 bits are fixed to 0, this address is 16-byte aligned.

CUX0 (Cursor-0 X position)

Register address	DisplayBaseAddress + $A8_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUX0											
R/W	R0				RW											
Default	0				Don't care											

This register controls the horizontal position of the cursor-0 pattern left edge. The reference position of the coordinate is the top left of the cursor pattern.

CUY0 (Cursor-0 Y position)

Register address	DisplayBaseAddress + Aa_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUY0											
R/W	R0				RW											
Default	0				Don't care											

This register controls the vertical position of the cursor-0 pattern top edge. The reference position of the coordinate is the top left of the cursor pattern.

CUOA1 (Cursor-1 Origin Address)

Register address	DisplayBaseAddress + AC_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						CUOA1																									
R/W	R0						RW																							R0		
Default	0						Don't care																							0000		

This register controls the start address of the cursor-1 pattern. Since the lowest 4 bits are fixed to "0", this address is 16-byte aligned.

CUX1 (Cursor-1 X position)

Register address	DisplayBaseAddress + $B0_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUX1											
R/W	R0				RW											
Default	0				Don't care											

This register controls the horizontal position of the cursor-1 pattern left edge. The reference position of the coordinate is the top left of the cursor pattern.

CUY1 (Cursor-1 Y position)

Register address	DisplayBaseAddress + $B2_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUY1											
R/W	R0				RW											
Default	0				Don't care											

The reference position of the coordinate is the top left of the cursor pattern.

This register sets the display position of 'cursor 1' (Y coordinate) in pixels.

BRATIO (Blend Ratio)

Register address	DisplayBaseAddress + $B4_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BRS	Reserved							BRATIO				Reserved			
R/W	RW	R0							RW				R0			
Default	0	0							0				0000			

This register controls the blending ratio for Console layer pixels when using the blending mode.

Bits 7-4 BRATIO (Blend Ratio)

Set blending ratio

0000 Coefficient = 0

0001 Coefficient = 1/16

: :

1111 Coefficient = 15/16

Bit 15 BRS (Blend Ratio Select)

Selects formula for alpha blending

0 (C layer color x Coefficient) + (Combination color of W/M/B layers x (1 - Coefficient))

1 (C layer color x (1 - Coefficient)) + (Combination color of W/M/B layers x Coefficient)

BMODE (Blend MODE)

Register address	DisplayBaseAddress + $B6_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved														Reserve	BLEND
R/W	R0														R0	RW
Default	0														0	0

This register controls the Console layer overlay options. The color set as a transparent color is irrelevant to the alpha bit and blend processing is not performed.

Bit 0 BLEND

Overlays mode between C and B/M/W

0: Simple priority mode (C layer given priority at all times)

1: Blending mode

When performing blend processing, specify the blend mode for this bit; alpha must be enabled previously in C layer display data. In the direct color mode, specify alpha for the most significant bit. In the indirect color mode, specify alpha for the most significant bit of pallet data.

CTC (C layer Transparent Control)

Register address	DisplayBaseAddress + BC_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	CZT	CTC														
R/W	RW	RW														
Default	0	Don't care														

This register controls the transparent color setting for the C layer. The color defined as a transparent color by this register is treated as a transparent color even in the blending mode. When both CTC and CZT are set to 0, color 0 is displayed in black (not transparent).

Bits 14-0 CTC (C layer Transparent Color)

Set color code of transparent color used in Console layer. Bits 7-0 used in indirect color mode.

Bit 15 CZT (C layer Zero Transparency)

Sets treatment for code 0 in Console layer

0: Code 0 not transparent color

1: Code 0 transparent color

MRTC (MR layer Transparent Control)

Register address	DisplayBaseAddress + $C0_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MRZT	MRTC														
R/W	RW	RW														
Default	0	Don't care														

This register controls the transparent color setting for the MR layer. When both MRTC and MRZT are set to 0, color 0 is displayed in black (not transparent).

Bits 14-0 MRTC (MR layer Transparent Color)

Set color code of transparent color used in MR layer. Bits 7-0 used in indirect color mode.

Bit 15 MRZT (MR layer Zero Transparency)

Sets treatment for code 0 in MR layer

0: Code 0 not transparent color

1: Code 0 transparent color

MLTC (ML layer Transparent Control)

Register address	DisplayBaseAddress + $C2_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MLZT	MLTC														
R/W	RW	RW														
Default	0	Don't care														

This register controls the transparent color setting for the ML layer. When both MLTC and MLZT are set to 0, color 0 is displayed in black (not transparent).

Bits 14-0 MLTC (ML layer Transparent Color)

Set color code of transparent color used in ML layer. Bits 7-0 used in indirect color mode.

Bit 15 MLZT (ML layer Zero Transparency)

Sets treatment for code 0 in ML layer

0: Code 0 not transparent color

1: Code 0 transparent color

CPAL0-255 (C layer Pallet 0-255)

Register address	DisplayBaseAddress + 400_H -- DisplayBaseAddress + $7FF_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	A								R								G								B							
R/W	RW	R0							RW							R0	RW							R0	RW							R0
Default	Don't care	0000000							Don't care							00	Don't care							00	Don't care							00

These are color pallet registers for Console layer and cursors. In the indirect color mode, a color code in the display field indicates the pallet register number (pallet entry number), and the color information set in that entry is applied as the display color of that pixel.

Bits 7-2 B (Blue)

Set blue color element

Bit 15-10 G (Green)

Set green color element

Bits 23-18 R (Red)

Set red color element

Bit 31 A (Alpha)

When blending mode used, color blended with B/M/W layer pixel color according to blending ratio for pixel of C layer with bit = 1. Alpha blending mode ignored when used as cursor color.

MBPAL0-255 (M layer and B layer Pallet 0-255)

Register address	DisplayBaseAddress + 800_H -- DisplayBaseAddress + BFF_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve								R								G								B							
R/W	R0								RW							R0	RW							R0	RW							R0
Default	0								Don't care							00	Don't care							00	Don't care							00

These are color pallet registers for Middle and Base layers. In the indirect color mode, a color code in the display field indicates the pallet register number (pallet entry number), and the color information set in that entry is applied as the display color of that pixel.

Bits 7-2 B (Blue)

Set blue color element

Bits 15-10 G (Green)

Set green color element

Bits 23-18 R (Red)

Set red color element

10.1.4 Video Capture Registers

VCM (Video Capture Mode)

Register address	CaputureBaseAddress + 00_H																												
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VIE	Reserve					CM		Reserve			VI	Reserve																		VS	Rsv
R/W	RW	RX					RW		RX			RW	RX																		RW	RX
Default	0	X					00		X			0	X																		0	X

This register sets the video capture mode.

Bit 31 VIE (Video Input Enable)

Enables video capture function

0: Does not capture video

1: Captures video

Bits 25-24 CM (Capture Mode)

Set video capture mode

To capture vides, set these bits to 11.

00: Default

01: Reserved

10: Reserved

11: Capture

Bit 20 VI (Vertical Interpolation)

Sets whether to perform vertical interpolation

0: Performs vertical interpolation

The graphics are enlarged vertically by two times

1: Does not perform vertical interpolation

Bit 1 VS (Video Select)

Selects NTSC or PAL

0: NTSC

1: PAL

CSC (Capture SCale)

Register address	CaputureBaseAddress + 04_H																												
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	VSCI					VSCF											HSCI					HSCF										
R/W	RW					RW											RW					RW										
Default	00001					00000000000											00001					00000000000										

This register sets the video capture enlargement/reduction ratio.

Bits 31-27 VSCI (Vertical SCale Integer)

Set integer part of vertical enlargement/reduction ratio

Bits 26-16 VSCF (Vertical SCale Fraction)

Set fraction part of vertical enlargement/reduction ratio

Bits 15-11 HSCI (Horizontal Scale Integer)

Set integer part of horizontal enlargement/reduction ratio

Bits 10-0 HSCF (Horizontal SCale Fraction)

Set fraction part of horizontal enlargement/reduction ratio

VCS (Video Capture Status)

Register address	CaputureBaseAddress + 08_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve																														CE	
R/W	RX																														RW	
Default	Don't care																														00000	

This register indicates the video capture status.

Bits 4-0 CE (Capture Error)

Indicate whether error occurred during video capture

00000: No error

Others: Error

CBM (vide Capture Buffer Mode)

Register address	CaputureBaseAddress + 10_H																															
Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	00	Reserved							CBW								Reserved															
R/W	RW	RX							RW								Rx															
Default	Don't care								Don't care								Don't care															

Bit 23-16 CBW (Capture Buffer memory Width)

Set memory width (stride) of capture buffer in 64-byte units

Bit 31 OO (Odd Only mode)

Specifies whether to capture odd fields only

0: Normal mode

1: Odd only mode

CBOA (video Capture Buffer Origin Address)

Register address	CaputureBaseAddress + 14_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																CBOA															
R/W	RX																RW														R0	
Default	Don't care																Don't care														0	

This register specifies the starting (origin) address of the video capture buffer.

CBLA (video Capture Buffer Limit Address)

Register address	CaputureBaseAddress + 18_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																CBLA															
R/W	RX																RW														R0	
Default	Don't care																Don't care														0	

This register specifies the end (limit) address of the video capture buffer.

"CBLA must be larger than CBOA".

CIHSTR (Capture Image Horizontal STaRt)

Register address	CaputureBaseAddress + $1C_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						CIHSTR									
R/W	RX						RW									
Default	Don't care						Don't care									

This register sets the range of the images to be written (captured) to the video capture buffer. Specify the X coordinate located in the top left of the image range as the number of pixels from the top left of the image. For reduction, apply this setting to the post-reduction image coordinate.

CIVSTR (Capture Image Vertical STaRt)

Register address	CaputureBaseAddress + $1E_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						CIVSTR									
R/W	RX						RW									
Default	Don't care						Don't care									

This register sets the range of the images to be written (captured) to the video capture buffer. Specify the Y coordinate located in the top left of the image range as the number of pixels from the top left of the image. For reduction, apply this setting to the post-reduction image coordinate.

CIHEND (Capture Image Horizontal END)

Register address	CaputureBaseAddress + 20_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						CIHEND									
R/W	RX						RW									
Default	Don't care						Don't care									

This register sets the range of the images to be written (captured) to the video capture buffer. Specify the X coordinate located in the bottom right of the image range as the number of pixels from the top left of the image. For reduction, apply this setting to the post-reduction image coordinate.

If the pixel at the right end of the image is not aligned on 64 bits/word boundary, extra data is written before 64 bits/word boundary.

If the width of the input image is less than the range set by this command, data is written only at the size of input image.

CIVEND (Capture Image Vertical END)

Register address	CaputureBaseAddress + 22_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						CIVEND									
R/W	RX						RW									
Default	Don't care						Don't care									

This register sets the range of the images to be written (captured) to the video capture buffer. Specify the Y coordinate located in the bottom right of the image range as the count of pixels from the top left of the original image to be input. For reduction, apply this setting to the post-reduction image coordinate.

If the count of rasters of the input image is less than the range set by this command, data is written only at the size of the input image.

CHP (Capture Horizontal Pixel)

Register address	CaputureBaseAddress + 28h																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																						CHP									
R/W	RX																						RW									
Default	X																						168_H (360_D)									

This register sets the number of horizontal pixels of the image output after scaling. Specify the count of horizontal pixels in 2-pixel units.

CVP (Capture Vertical Pixel)

Register address	CaputureBaseAddress + $2c_H$																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved										CVPP						Reserved						CVPN									
R/W	RX										RW						RX						RW									
Default	X										271_H (625_D)						X						$20D_H$ (525_D)									

This register sets the count of vertical pixels of the image output after scaling. The fields used depend on the video format used.

Bit 25-16 CVPP (Capture Vertical Pixel for PAL)

Set count of vertical pixels of output image when PAL format used

Bit 9-0 CVPN (Capture Vertical Pixel for NTSC)

Set count of vertical pixels of output image when NTSC format used

CDCN (Capture Data Count for NTSC)

Register address	CaputureBaseAddress + 4000_H																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved										BDCN						Reserved						VDCN									
R/W	RX										RW						RX						RW									
Default	X										$10f_H$ (271_D)						X						$5A3_H$ (1443)									

This register sets the count of data of the input video stream when the NTSC format is used.

Bit 25-16 BDCN (Blanking Data Count for NTSC)

Set count of data processed during blanking period when NTSC format used

Bit 10-0 VDCN (Valid Data Count for NTSC)

Set count of data processed during valid period when NTSC format used

CDCP (Capture Data Count for PAL)

Register address	CaputureBaseAddress + 4004_H			
Bit #	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10	9 8 7 6 5 4 3 2	1 0
Bit field name	Reserved	BDCP	Reserved	VDCP
R/W	RX	RW	RX	RW
Default	X	$11B_H$ (283_D)	X	$5A3_H$(1443)

This register sets the count of data of the input video stream when the PAL format is used.

Bit 25-16 BDCP (Blanking Data Count for PAL)

Set count of data processed during blanking period when PAL format used

Bit 10-0 VDCP (Valid Data Count for PAL)

Set number of data processed during valid period when PAL format used

10.1.5 Draw Control Registers

CTR (Control Register)

Register address	DrawBaseAddress + 400_H																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name								FO	PE	CE		FCNT						NF	FF	FE			SS				DS				PS	
R/W								RW	RW	RW		R						R	R	R			R				R				R	
Default								0	0	0		100000						0	0	1			00				00				00	

This register indicates draw flags and status. Bits 24-22 are not cleared until 0 is set.

Bits 1-0 PS (Pixel engine Status)

Indicate status of pixel engine unit

00 Idle

01 Busy

10 Reserved

11 Reserved

Bits 5-4 DS (DDA Status)

Indicate status of DDA

00 Idle

01 Busy

10 Busy

11 Reserved

Bits 9-8 SS (Setup Status)

Indicate status of Setup unit

00 Idle

01 Busy

10 Reserved

11 Reserved

Bit 12 FE (FIFO Empty)

Indicates status of display list FIFO

0 Valid data

1 No valid data

Bit 13 FF (FIFO Full)

Indicates fullness of display list FIFO

0 Not full

1 Full

Bit 14 NF (FIFO Near Full)

Indicates entries of display list FIFO

0 Empty entries equal to or more than half

1 Empty entries less than half

Bits 20-15 FCNT(FIFO Counter)

Indicate number of empty entries (0: Full - 32: Empty)

Bit 22 CE (Display List Command Error)

Indicates command error detection

0 Normal

1 Command error detected

Bit 23 PE (Display List Packet code Error)

Indicates packet code error detection

0 Normal

1 Packet code error detected

Bit 24 FO (FIFO Overflow)

Indicates FIFO overflow status

0 Normal

1 FIFO overflow detected

IFSR (Input FIFO Status Register)

Register address	DrawBaseAddress + 404_H																														
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																														NF	FF	FE
R/W																														R	R	R
Default																														0	0	1

This is a miller register for bits 14-12 of the CTR register.

IFCNT (Input FIFO Counter)

Register address	DrawBaseAddress + 408_H																														
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																											FCNT					
R/W																											R					
Default																											100000					

This is a miller register for bits 19-15 of the CTR register.

SST (Setup engine Status)

Register address	DrawBaseAddress + $40C_H$																														
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																															SS	
R/W																															R	
Default																															00	

This is a miller register for bits 9-8 of the CTR register.

DST (DDA Status)

Register address	DrawBaseAddress + 410_H																														
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																															DS	
R/W																															RW	
Default																															00	

This is a miller register for bits 5-4 of the CTR register.

PST (Pixel engine Status)

Register address	DrawBaseAddress + 414_H																														
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																															PS	
R/W																															R	
Default																															00	

This is a miller register for bits 1-0 of the CTR register.

EST (Error Status)

Register address	DrawBaseAddress + 418_H																														
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																														FO	PE	CE
R/W																														RW	RW	RW
Default																														0	0	0

This is a miller register for bits 24-22 of the CTR register.

10.1.6 Draw mode Registers

When write to the registers, use the SetRegister command. The registers cannot be accessed from the CPU.

MDR0 (Mode Register for miscellaneous)

Register address	DrawBaseAddress + 420_H																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																	CF						CY	CX					BSV		BSH	
R/W																	RW						RW	RW					RW		RW	
Default																	0						0	0					00		00	

Bits 1-0 BSH (Bitmap Scale Horizontal)

Set horizontal zoom ratio of bitmap draw

00 x1

01 x2

10 x1/2

01 Reserved

Bits 3-2 BSV (Bitmap Scale Vertical)

Set vertical zoom ratio of bitmap draw

00 x1

01 x2

10 x1/2

01 Reserved

Bit 8 CX (Clip X enable)

Sets X coordinate clipping mode

0 Disable

1 Enable

Bit 9 CY (Clip Y enable)

Sets Y coordinate clipping mode

0 Disable

1 Enable

Bit 15 CF (Color Format)

Sets drawing color format of current draw frame

0 Indirect color mode (8 bits/pixel)

1 Direct color mode (16 bits/pixel)

MDR1 (Mode Register for LINE)

Register address	DrawBaseAddress + 420_H																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																	CF						CY	CX					BSV		BSH	
R/W																	RW						RW	RW					RW		RW	
Default																	0						0	0					00		00	

This register controls the mode of line draw and pixel plot.

Bit 2 ZC (Z Compare mode)

Sets Z comparison mode

0 Disable

1 Enable

Bits 5-3 ZCL (Z Compare Logic)

Select type of Z comparison

000 NEVER

001 ALWAYS

010 LESS

011 LEQUAL

100 EQUAL

101 GEQUAL

110 GREATER

111 NOTEQUAL

Bit 6 ZW (Z Write mask)

Sets ZWRITEMASK

0 Compare Z values and overwrite result to Z buffer.

1 Compare Z values and do not overwrite to Z buffer.

Bits 8-7 BM (Blend Mode)

Set blend mode

00 Normal (source copy)

01 Alpha blending

10 Logical operation enable

11 Reserved

Bits 12-9 LOG (Logical operation)

Set type of logical operation

0000 CLEAR

0001 AND

0010 AND REVERSE

0011 COPY

0100 AND INVERTED

0101 NOP

0110 XOR

0111 OR

1000 NOR

1001 EQUIV

1010 INVERT

1011 OR REVERSE

1100 COPY INVERTED

1101 OR INVERTED

1110 NAND

1111 SET

Bit 19 BL (Broken Line)

Selects line type

0 Solid line

1 Broken line

Bit 20 BP (Broken line Period)

Selects broken line period

0: 32 bits

1: 24 bits

Bits 28-24 LW (Line Width)

Set line width

0000 1 pixel
0

0000 2 pixels
1

: :

1111 32 pixels
1

MDR2 (Mode Register for Polygon)

Register address	DrawBaseAddress + 428_H																												
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name			TT																	LOG				BM		ZW	ZCL			ZC		SM
R/W			RW																	RW				RW		RW	RW			RW		RW
Default			00																	0011				0		0	0000			0		0

This register controls the polygon draw mode.

Bit 0 SM (Shading Mode)

Sets shading mode

0 Flat shading

1 Gouraud shading

Bit 2 ZC (Z Compare mode)

Sets Z comparison mode

0 Disable

1 Enable

Bits 5-3 ZCL (Z Compare Logic)

Select type of Z comparison

000 NEVER

001 ALWAYS

010 LESS

011 LEQUAL

100 EQUAL

101 GEQUAL

110 GREATER

111 NOTEQUAL

Bit 6 ZW (Z Write mask)

Sets ZWRITEMASK

0 Compare Z values and overwrite result to Z buffer

1 Compare Z values and do not overwrite result to Z buffer

Bits 8-7 BM (Blend Mode)

Set blend mode

00 Normal (source copy)

01 Alpha blending

10 Logical calculation enable

11 Reserved

Bits 12-9 LOG (Logical operation)

Set type of logical operation

0000 CLEAR

0001 AND

0010 AND REVERSE

0011 COPY

0100 AND INVERTED

0101 NOP

0110 XOR

0111 OR

1000 NOR

1001 EQUIV

1010 INVERT

1011 OR REVERSE

1100 COPY INVERTED

1101 OR INVERTED

1110 NAND

1111 SET

Bits 29-28 TT (Texture-Tile Select)

Select texture or tile pattern

00 Not used

01 Enable tiling operation

10 Enable texture mapping

11 Reserved

MDR3 (Mode Register for Texture)

Register address	DrawBaseAddress + $42C_H$																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name											TAB				TBL						TWS		TWT				TF		TC			TBU
R/W											RW				RW						RW		RW				RW		RW			RW
Default											00				00						00		00				0		0			0

This register controls the texture mapping mode.

Bit 0 TBU (Texture Buffer)

Selects texture memory (internal buffer always used in tiling)

0 External Graphics Memory

1 Internal texture buffer

Bit 3 TC (Texture coordinates Correct)

Controls perspective correction mode

0 Disable

1 Enable

Bit 5 TF (Texture Filtering)

Sets texture filtering mode

0 Point sampling

1 Bi-linear filtering

Bits 9-8 TWT (Texture Wrap T)

Set texture T coordinate wrapping mode

00 Repeat

01 Cramp

10 Border

11 Reserved

Bits 11-10 TWS (Texture Wrap S)

Set texture S coordinate wrapping mode

00 Repeat

01 Cramp

10 Border

11 Reserved

Bits 17-16 TBL (Texture Blend mode)

Set texture blending mode

00 Decal

01 Modulate

10 Stencil

11 Reserved

Bits 21-20 TAB (Texture Alpha Blend mode)

Set texture alpha blending mode. The stencil and the stencil alpha mode are used only when the BM bits in the MDR2 register are set to 01 (alpha blending). If any other mode is set at the BM bit field, the stencil and the stencil alpha mode are treated as the stencil mode.

00 Normal

01 Stencil

10 Stencil alpha

11 Reserved

MDR4 (Mode Register for BLT)

Register address	DrawBaseAddress + 430_H																																
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit field name																				LOG				BM							TE		
R/W																				RW				RW							RW		
Default																				0011				00							0		

This register controls the BitBLT. Mode.

Bits 1 TE (Transparent Enable)

Sets transparent mode

0: Does not perform transparent processing

1: Does not draw pixels that do not match set transparent color when BLT performed (permeation copy)

Note: Set the blend mode (BM) to normal.

Bits 8-7 BM (Blend Mode)

Set blend mode

00 Normal (source copy)

01 Reserved

10 Logical calculation enable

11 Reserved

Bits 12-9 LOG (Logical operation)

Set logical calculation type

0000 CLEAR

0001 AND

0010 AND REVERSE

0011 COPY

0100 AND INVERTED

0101 NOP

0110 XOR

0111 OR

1000 NOR

1001 EQUIV

1010 INVERT

1011 OR REVERSE

1100 COPY INVERTED

1101 OR INVERTED

1110 NAND

1111 SET

FBR (Frame buffer Base)

Register address	DrawBaseAddress + 440_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name							FBASE																									
R/W							RW																								R0	
Default							Don’t care																								0	

This register controls the base address of the drawing frame memory.

XRES (X Resolution)

Register address	DrawBaseAddress + 444_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																					XRES											
R/W																					RW											
Default																					Don’t care											

This register controls the drawing frame horizontal resolution.

ZBR (Z-buffer Base)

Register address	DrawBaseAddress + 448_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name							ZBASE																									
R/W							RW																								R0	
Default							Don’t care																								0	

This register controls the Z buffer base address.

TBR (Texture memory Base)

Register address	DrawBaseAddress + $44C_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name							TBASE																									
R/W							RW																								R0	
Default							Don’t care																								0	

This register controls the texture memory base address.

PFBR (2D Polygon Flag-Buffer Base)

Register address	DrawBaseAddress + 450_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name							PFBASE																									
R/W							RW																								R0	
Default							Don't care																								0	

This register controls the polygon flag buffer base address.

CXMIN (Clip X minimum)

Register address	DrawBaseAddress + 454_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																					CLIPXMIN											
R/W																					RW											
Default																					Don't care											

This register controls the clip frame minimum X position.

CXMAX (Clip X maximum)

Register address	DrawBaseAddress + 458_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																					CLIPXMAX											
R/W																					RW											
Default																					Don't care											

This register controls the clip frame maximum X position.

CYMIN (Clip Y minimum)

Register address	DrawBaseAddress + $45C_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																					CLIPYMIN											
R/W																					RW											
Default																					Don't care											

This register controls the clip frame minimum Y position.

CYMAX (Clip Y maximum)

Register address	DrawBaseAddress + 460_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																					CLIPYMAX											
R/W																					RW											
Default																					Don't care											

This register controls the clip frame maximum Y position.

TXS (Texture Size)

Register address	DrawBaseAddress + 464_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name								TXSN																TXSM								
R/W								RW																RW								
Default								100000000																100000000								

This register controls the texture size (m, n).

Bits 8-0 TXSM (Texture Size M)

Set horizontal texture size. Any power of 2 between 4 and 256 can be used. Values that are not a power of 2 cannot be used.

000000100 M=4

000001000 M=8

000010000 M=16

000100000 M=32

001000000 M=64

010000000 M=128

100000000 M=256

Others Prohibited

Bits 24-16 TXSN (Texture Size N)

Set vertical texture size. Any power of 2 between 4 and 256 can be used. Values that are not a power of 2 cannot be used.

000000100 N=4

000001000 N=8

000010000 N=16

000100000 N=32

001000000 N=64

010000000 N=128

100000000 N=256

Others Prohibited

TIS (Tile Size)

Register address	DrawBaseAddress + 468_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name										TISN																TISM						
R/W										RW																RW						
Default										1000000																1000000						

This register controls the tile size (m, n).

Bits 6-0 TISM (Title Size M)

Set horizontal tile pattern size. Any power of 2 between 4 and 64 can be used. Values that are not a power of 2 cannot be used.

0.000100 M=4

0001000 M=8

0010000 M=16

0100000 M=32

1000000 M=64

Others Prohibited

Bits 22-16 TISN (Title Size N)

Set vertical tile pattern size. Any power of 2 between 4 and 643 can be used. Values that are not a power of 2 cannot be used.

0000100 N=4

0001000 N=8

0010000 N=16

0100000 N=32

1000000 N=64

Others Prohibited

TOA (Texture Buffer Offset address)

Register address	DrawBaseAddress + $46C_H$																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																				XBO												
R/W																				RW												
Default																				Don't care												

This registers controls the texture buffer offset address of. By using this offset value, multiple texture patterns can be used and referred to the texture buffer memory.

Specify the word-aligned byte address (16 bits). (Bit 0 is always 0.)

FC (Foreground Color)

Register address	DrawBaseAddress + 480_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																	FGC															
R/W																	RW															
Default																	0															

This register controls the drawing frame foreground color. This color is used for the object color of flat shading and foreground color of bitmap draw and broken line draw. At bitmap drawing, all bits set to 1 are drawn in the color set at this register.

Bits 14-0 FGC (Foreground Color)

Set foreground color value. In the indirect color mode, the lower 8 bits (bits 7-0) are used.

Bits 15 This bit is valid only when a bitmap or rectangle is drawn.

When others are drawn, this bit is always treated as "0".

BC (Background Color)

Register address	DrawBaseAddress + 484_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																	BT	BGC														
R/W																	RW	RW														
Default																	0	0														

This register controls the drawing frame background color. This color is used for the background color of bitmap draw and broken line draw. At bitmap drawing, all bits set to 1 are drawn in the color set at this register.

Bits 14-0 BGC (Background Color)

Set background color value. In the indirect color mode, the lower 8 bits (bit 7-0) are used.

Bit 15 BT (Background Transparency)

Sets transparent mode of background color

0 Draw background in color used in BGC field.

1 Don't draw background (use current color).

ALF (Alpha Factor)

Register address	DrawBaseAddress + 488_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																									A							
R/W																									RW							
Default																									0							

This register controls the alpha blending ratio.

BLP (Broken Line Pattern)

Register address	DrawBaseAddress + $48C_H$																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BLP																															
R/W	RW																															
Default	0																															

This register controls the broken-line pattern. The bit 1 set in the broken-line pattern is drawn in the foreground color and bit 0 is drawn in the background color. The actual line pattern is pasted from MSB to LSB to the line to be drawn. The BLPO register is used to manage the bit numbers of the broken-line pattern. 32 or 24 bits can be selected as the repetition of the broken-line pattern by setting the BP bit of the MDR1 register. When 24 bits are selected, bits 23 to 0 of the BLP register are used.

TBC (Texture Border Color)

Register address	DrawBaseAddress + 494_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																	-	BC														
R/W																	R0	RW														
Default																	0	0														

This register controls the texture mapping border color.

Bits 14-0 BC (Border Color)

Set border color of texture mapping. Only the direct color mode is used.

BLPO (Broken Line Pattern Offset)

Register address	DrawBaseAddress + $3E0_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																												BCR				
R/W																												RW				
Default																												11111				

This register controls the start bit position of the broken line pattern set to BLP registers, for broken line drawing. The lowest 5 bits contain the bit number of the broken line pattern. This value is decremented at each pixel draw. Broken line drawing can be started from any position of the specified broken line pattern by setting any number at this register.

10.1.7 Triangle Draw Registers

Each register is used by the drawing commands. The registers cannot be accessed from the CPU or by using the SetRegister command.

(XY coordinate register)

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ys	0000_H	S	S	S	S	Int																		0									
Xs	0004_H	S	S	S	S	Int																		Frac									
dXdy	0008_H	S	S	S	S	Int																		Frac									
XUs	$000c_H$	S	S	S	S	Int																		Frac									
dXUdy	0010_H	S	S	S	S	Int																		Frac									
XLs	0014_H	S	S	S	S	Int																		Frac									
dXLdy	0018_H	S	S	S	S	Int																		Frac									
USN	$001b_H$	0	0	0	0	Int																		0									
LSN	0020_H	0	0	0	0	Int																		0									

Address	Offset value from DrawBaseAddress
S	Sign bit or sign extension
0	Not used or 0 extension
Int	Integer or integer part of fixed point data
Frac	Fraction part of fixed point data

Sets (X, Y) coordinate for triangle drawing

Ys	Y coordinate start position of long side
Xs	X coordinate start position of long side
dXdy	X DDA value of long side
XUs	X coordinate start position of top side
dXUdy	X DDA value of top side
XLs	X coordinate start position of bottom side
dXLdy	X DDA value of lower side
USN	Number of spans (rasters) of top triangle. If this value is 0, the top triangle is not drawn.
LSN	Number of spans (rasters) of bottom triangle. If this value is 0, the bottom triangle is not drawn.

(Color register)

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Rs	0040_H	0	0	0	0	0	0	0	0	Int								Frac															
dRdx	0044_H	S	S	S	S	S	S	S	S	Int								Frac															
dRdy	0048_H	S	S	S	S	S	S	S	S	Int								Frac															
Gs	$004C_H$	0	0	0	0	0	0	0	0	Int								Frac															
dGdx	0050_H	S	S	S	S	S	S	S	S	Int								Frac															
dGdy	0054_H	S	S	S	S	S	S	S	S	Int								Frac															
Bs	0058_H	0	0	0	0	0	0	0	0	Int								Frac															
dBdx	$005c_H$	S	S	S	S	S	S	S	S	Int								Frac															
dBdy	0060_H	S	S	S	S	S	S	S	S	Int								Frac															

Address	Offset from DrawBaseAddress
S	Sign bit or sign extension
0	Not used or 0 extension
Int	Integer or integer part of fixed point data
Frac	Fraction part of fixed point data

Sets color parameters for triangle drawing. These parameters are used in the Gouraud shading mode.

Rs	R value at (Xs, Ys, Zs) of long side
dRdx	R DDA value of horizontal way
dRdy	R DDA value of long side
Gs	G value at (Xs, Ys, Zs) of long side
dGdx	G DDA value of horizontal way
dGdy	G DDA value of long side
Bs	B value at (Xs, Ys, Zs) of long side
dBdx	B DDA value of horizontal way
dBdy	B DDA value of long side

(Z coordinates register)

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Zs	0080h	0	Int															Frac															
dZdx	0084h	S	Int															Frac															
dZdy	008ch	S	Int															Frac															

Address	Offset from DrawBaseAddress
S	Sign bit or sign extension
0	Not used or 0 extension
Int	Integer or integer part of fixed point data
Frac	Fraction part of fixed point data

Sets Z coordinate for 3D triangle drawing

Zs	Z coordinate start position of long side
dZdx	Z DDA value of horizontal way
dZdy	Z DDA value of long side

(Texture coordinate register)

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ss	$00c0_H$	S	S	S	S	S	S	S	Int									Frac															
dSdx	$00c4_H$	S	S	S	S	S	S	S	Int									Frac															
dSdy	$00c8_H$	S	S	S	S	S	S	S	Int									Frac															
Ts	$00cc_H$	S	S	S	S	S	S	S	Int									Frac															
dTdx	$00d0_H$	S	S	S	S	S	S	S	Int									Frac															
dTdy	$00d4_H$	S	S	S	S	S	S	S	Int									Frac															
Qs	$00d8_H$	0	0	0	0	0	0	0	Int	Frac																							
dQdx	$00dc_H$	S	S	S	S	S	S	S	Int	Frac																							
dQdy	$00e0_H$	S	S	S	S	S	S	S	Int	Frac																							

Address	Offset from DrawBaseAddress
S	Sign bit or sign extension
0	Not used or 0 extension
Int	Integer or integer part of fixed point data
Frac	Fraction part of fixed point data

Sets texture coordinate parameters for triangle drawing

Ss	S coordinate of texture at (Xs, Ys, Zs) of long side
dSdx	S DDA value of horizontal way
dSdy	S DDA value of long side
Ts	T coordinate of texture at (Xs, Ys, Zs) of long side
dTdx	T DDA value of horizontal way
dTdy	T DDA value of long side
Qs	Q (Perspective correction value) of texture at (Xs, Ys, Zs) of long side
dQdx	Q DDA value of horizontal way
dQdy	Q DDA value of long side

10.1.8 Line Draw Registers

Each register is used by the drawing commands. The registers cannot be accessed from the CPU or by using the SetRegister command.

(Coordinate register)

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LPN	0140_H	0	0	0	0	Int												0															
LXs	0144_H	S	S	S	S	Int												Frac															
LXde	0148_H	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	Int	Frac															
LYs	$014c_H$	S	S	S	S	Int												Frac															
LYde	0150_H	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	Int	Frac															
LZs	0154_H	S	Int															Frac															
LZde	0158_H	S	Int															Frac															

Address	Offset from DrawBaseAddress
S	Sign bit or sign extension
0	Not used or 0 extension
Int	Integer or integer part of fixed point data
Frac	Fraction part of fixed point data

Sets coordinate parameters for line drawing

LPN	Pixel length of line
LXs	X coordinate position of line draw start vertex (In principal axis Y) Integer value of X coordinate rounded. (In principal axis Y) Set X coordinate of fixed point data.
LXde	Line angle data for X coordinate (In principal axis X) Increment or decrement according to drawing direction. (In principal axis Y) Set fraction part of DX/DY.
LYs	Y coordinate position of line draw start vertex (In principal axis X) Set current integer part of fixed point Y coordinate data. (In principal axis Y) Set integer value of Y coordinate rounded.
LYde	Line angle data for Y axis (In principal axis X) Set fraction part of dY/dX. (In principal axis Y) Increment or decrement according to drawing direction.
LZs	Z coordinate position of line draw start vertex
LZde	Z angle

10.1.9 Pixel Plot Registers

Each register is used by the drawing commands. The registers cannot be accessed from the CPU or by using the SetRegister command.

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PXdc	0180_H	0	0	0	0	Int												0															
PYdc	0184_H	0	0	0	0	Int												0															
PZdc	0188_H	0	0	0	0	Int												0															

Address	Offset from DrawBaseAddress
S	Sign bit or sign extension
0	Not used or 0 extension
Int	Integer or integer part of fixed point data
Frac	Fraction part of fixed point data

Sets coordinate parameter for pixel plot. The foreground color is used.

PXdc	Set X coordinate position
PYdc	Set Y coordinate position
PZdc	Set Z coordinate position

10.1.10 Rectangle Draw Registers

Each register is used by the drawing commands. The registers cannot be accessed from the CPU or by using the SetRegister command.

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXs	0200_H	0	0	0	0	Int												0															
RYs	0204_H	0	0	0	0	Int												0															
RsizeX	0208_H	0	0	0	0	Int												0															
RsizeY	$020c_H$	0	0	0	0	Int												0															

Address	Offset from DrawBaseAddress
S	Sign bit or sign extension
0	Not used or 0 extension
Int	Integer or integer part of fixed point data
Frac	Fraction part of fixed point data

Sets coordinate parameters for rectangle drawing. The foreground color is used.

RXs	Set the X coordinate of top left vertex
RYs	Set the Y coordinate of top left vertex
RsizeX	Set horizontal size
RsizeY	Set vertical size

10.1.11 Blt Registers

Set the parameters of each register as follows:

- Set the Tcolor register with the ***SetRegister*** command.

  Note that the Tcolor register cannot be set at access from the CPU and by drawing commands.

- Each register other than the Tcolor register is set by executing a drawing command.

  Note that access from the CPU and the ***SetRegister*** command cannot be used.

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR	0240_H	0	0	0	0	0	0	0	Address																								
SStride	0244_H	0	0	0	0	Int												0															
SRXs	0248_H	0	0	0	0	Int												0															
SRYs	$024c_H$	0	0	0	0	Int												0															
DADDR	0250_H	0	0	0	0	0	0	0	Address																								
DStride	0254_H	0	0	0	0	Int												0															
DRXs	0258_H	0	0	0	0	Int												0															
DRYs	$025c_H$	0	0	0	0	Int												0															
BRsizeX	0260_H	0	0	0	0	Int												0															
BRsizeY	0264_H	0	0	0	0	Int												0															
TColor	0280_H	0																Color															

Address	Offset from DrawBaseAddress
S	Sign bit or sign extension
0	Not used or 0 extension
Int	Integer or integer part of fixed point data
Frac	Fraction part of fixed point data

Sets parameters for Blt operations

SADDR	Sets start address of source field in byte boundary.
SStride	Sets horizontal size of source field
SRXs	Sets start X coordinate position of source rectangle
SRYs	Sets start Y coordinate position of source rectangle
DADDR	Sets start address of destination rectangle in byte boundary
DStride	Sets horizontal size of destination field
DRXs	Sets start X coordinate position of destination rectangle
DRYs	Sets start Y coordinate position of destination rectangle
BRsizeX	Sets horizontal size of rectangle
BRsizeY	Sets vertical size of rectangle
Tcolor	Sets transparent color For indirect color, set a palette code in the 8 low-order bits.

10.1.12 Fast2DLine Draw Registers

Each register is used by the drawing commands. The registers cannot be accessed from the CPU.

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LX0dc	0540_H	0	0	0	0	Int												0															
LY0dc	0544_H	0	0	0	0	Int												0															
LX1dc	0548_H	0	0	0	0	Int												0															
LY1dc	$054c_H$	0	0	0	0	Int												0															

Address	Offset from DrawBaseAddress
S	Sign bit or sign extension
0	Not used or 0 extension
Int	Integer or integer part of fixed point data
Frac	Fraction part of fixed point data

Sets coordinate parameters of both end points for Fast2DLine drawing

LX0dc	Sets X coordinate of vertex V0
LY0dc	Sets Y coordinate of vertex V0
LX1dc	Sets X coordinate of vertex V1
LY1dc	Sets Y coordinate of vertex V1

10.1.13 Fast2DTriangle Draw Registers

Each register is used by the drawing commands. The registers cannot be accessed from the CPU or by using the SetRegister command.

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X0dc	0580_H	0	0	0	0	Int																		0									
Y0dc	0584_H	0	0	0	0	Int																		0									
X1dc	0588_H	0	0	0	0	Int																		0									
Y1dc	$058c_H$	0	0	0	0	Int																		0									
X2dc	0590_H	0	0	0	0	Int																		0									
Y2dc	0594_H	0	0	0	0	Int																		0									

Address	Offset from DrawBaseAddress
S	Sign bit or sign extension
0	Not used or 0 extension
Int	Integer or integer part of fixed point data
Frac	Fraction part of fixed point data

Sets coordinate parameters of three vertices for Fast2DTriangle drawing

X0dc	Sets X coordinate of vertex V0
Y0dc	Sets Y coordinate of vertex V0
X1dc	Sets X coordinate of vertex V1
Y1dc	Sets Y coordinate of vertex V1
X2dc	Sets X coordinate of vertex V2
Y2dc	Sets Y coordinate of vertex V2

10.1.14 Geometry Control Register

GCTR (Geometry Control Register)

Register address	GeometryBaseAddress + 00_H																																
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve							FO	Rsv			FCNT						NF	FF	FE	Rsv		GS	Rsv		SS	Rsv		PS				
R/W	RX							RX	RX			RX						RX	RX	RX	RX		R	RX		R	RX		R				
Default	X							0	X			100000						0	0	1	X		00	X		00	X		00				

The flags and status information of the geometry section are reflected in this register.

Note that the flags and status information of the drawing section are reflected in CTR.

Bits 1-0 PS (Pixel engine Status)
Indicate status of pixel engine unit
00: Idle
01: Processing
10: Reserved
11: Reserved

Bits 5-4 SS (geometry Setup engine Status)
Indicate status of geometry setup engine unit
00: Idle
01: Processing
10: Processing
11: Reserved

Bits 9-8 GS (Geometry engine Status)
Indicate status of geometry engine unit
00: Idle
01: Processing
10: Reserved
11: Reserved

Bit 12 FE (FIFO Empty)
Indicates absence of data in display list FIFO (DFIFOD)
0: Data in DFIFOD
1: No data in DFIFOD

Bit 13 FF (FIFO Full)
Indicates whether DFIFOD full
0: DFIFOD not full
1: DFIFOD full

Bit 14 NF (FIFO Near Full)
Indicates amount of free space in DFIFOD
0: More than half of DFIFOD free
1: Less than half of DFIFOD free

Bits 20-15 FCNT (FIFO Counter)
Indicate number of free stages (0 to 100000_H) of DFIFOD

Bit 24 FO (FIFO Overflow)
Indicates whether FIFO overflow occurred
0: Normal
1: FIFO overflows

10.1.15 Geometry Mode Registers

The ***SetRegister*** command is used to write values to geometry mode registers. The geometry mode registers cannot be accessed from the CPU.

GMDR0 (Geometry Mode Register for Vertex)

Register address	GeometryBaseAddress + 40_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																									CF	DF			ST	Z	C	F
R/W																									RW	RW			RW	RW	RW	RW
Default																									0	00			0	0	0	0

This register sets the types of parameters input as vertex data and the type of projective transformation.

Bit 7 CF (Color Format)
Specifies color data format
0: Independent RGB format
1: Packed RGB format

Bits 6-5 DF (Data Format)
Specify vertex coordinate data format
00: Specifies floating-point format (Only independent RGB format can be used as color data format.)
01: Specifies fixed-point format (Only packed RGB format can be used as color data format.)
10: Reserved
11: Specifies packed integer format (Only packed RGB format can be used as color data format.)

CF	DF	Input data format
0	00	Floating-point format + independent RGB format
	01	Reserved
	10	Reserved
	11	Reserved
1	00	Reserved
	01	Fixed-point format + packed RGB format
	10	Reserved
	11	Packed integer format + packed RGB format

Bit 3 ST (texture S and T data enable)
Sets whether to use texture ST coordinate
0: Does not use texture ST coordinate
1: Uses texture ST coordinate

Bit 2 Z (Z data enable)
Sets whether to use Z coordinate
0: Does not use Z coordinate
1: Uses Z coordinate

Bit 1 C (Color data enable)
Sets whether to set vertex color.
0: Does not set vertex color.
1: Sets vertex color.

Bit 0 F (Frustum mode)
Sets projective transformation mode
0: Orthogonal projection transformation mode
1: Perspective projection transformation mode

GMDR1 (Geometry Mode Register for Line)

Register address	GeometryBaseAddress + 44_H																														
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																												BO		EP		AA
R/W																												RW		RW		RW
Default																												0		0		0

This register sets the line drawing mode.

Bit 4 BO (Broken line Offset)
Sets whether to clear the broken line reference position
0: Does not clear broken line reference position
1: Clears broken line reference position

Bit 2 EP (End Point mode)
Sets end point drawing mode (whether to draw end point)
Note that the end point is not drawn in line strip.
0: Does not draw end point
1: Draws the end point

Bit 0 AA (Anti-alias mode)
Sets anti-alias mode (whether to set anti-alias)
0: Does not set anti alias
1: Sets anti alias

GMDR2 (Geometry Mode Register for Triangle)

Register address	GeometryBaseAddress + 48_H																														
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																														FD		CF
R/W																														RW		RW
Default																														0		0

This register sets the geometry processing mode when a triangle is drawn.

Bit 2 FD (Face Definition)
Sets face definition
0: Sets face definition when vertex list counterclockwise
1: Sets face definition when vertex list clockwise

Bit 0 CF (Cull Face)
Sets whether to draw rear (cull face)
0: Draws rear
1: Does not draw rear (disabled when polygon drawn)

10.1.16 DisplayList FIFO Registers

DFIFOG (Geometry Displaylist FIFO with Geometry)

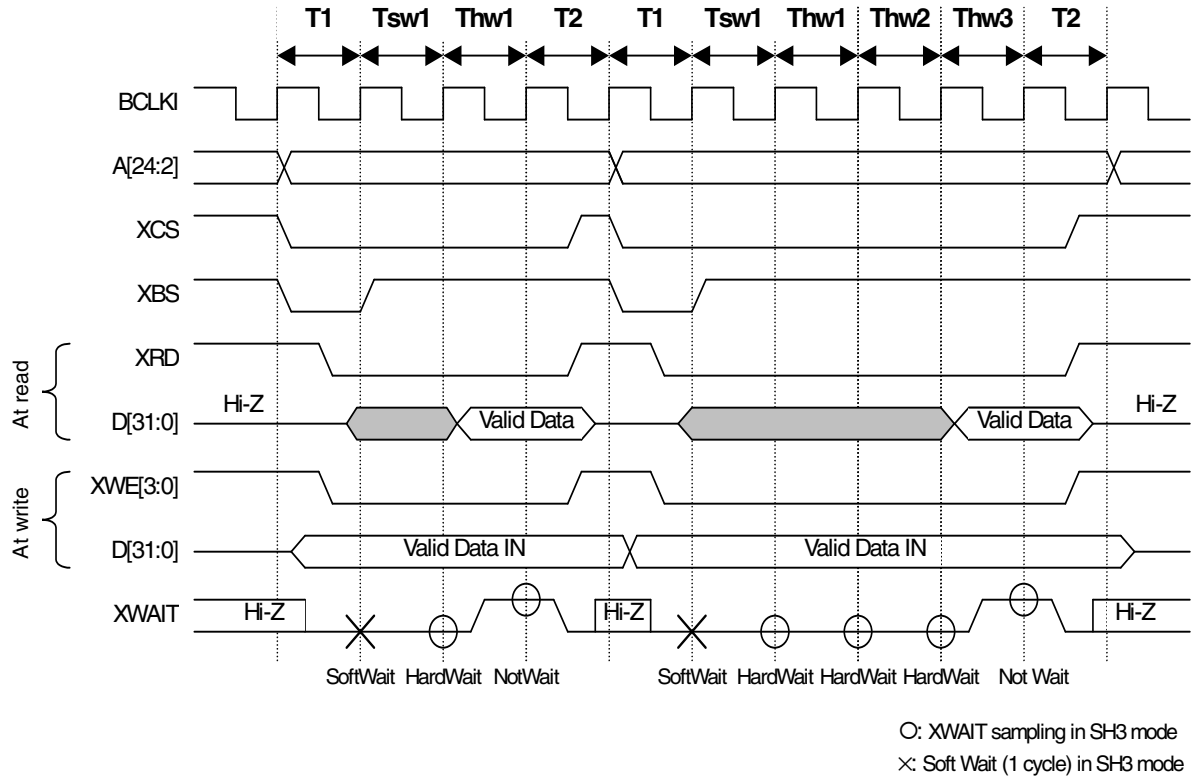
Register address	Geometry BaseAddress + 400_H																													
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DFIFOG																															
R/W	W																															
Default	Don't care																															

FIFO registers for DisplayList transfer

11. Timing Diagram

11.1 Host Interface

11.1.1 CPU Read/Write Timing Diagram for SH3 Mode (Normally Not Ready Mode)



T1: Read/write start cycle (XRDY in wait state)

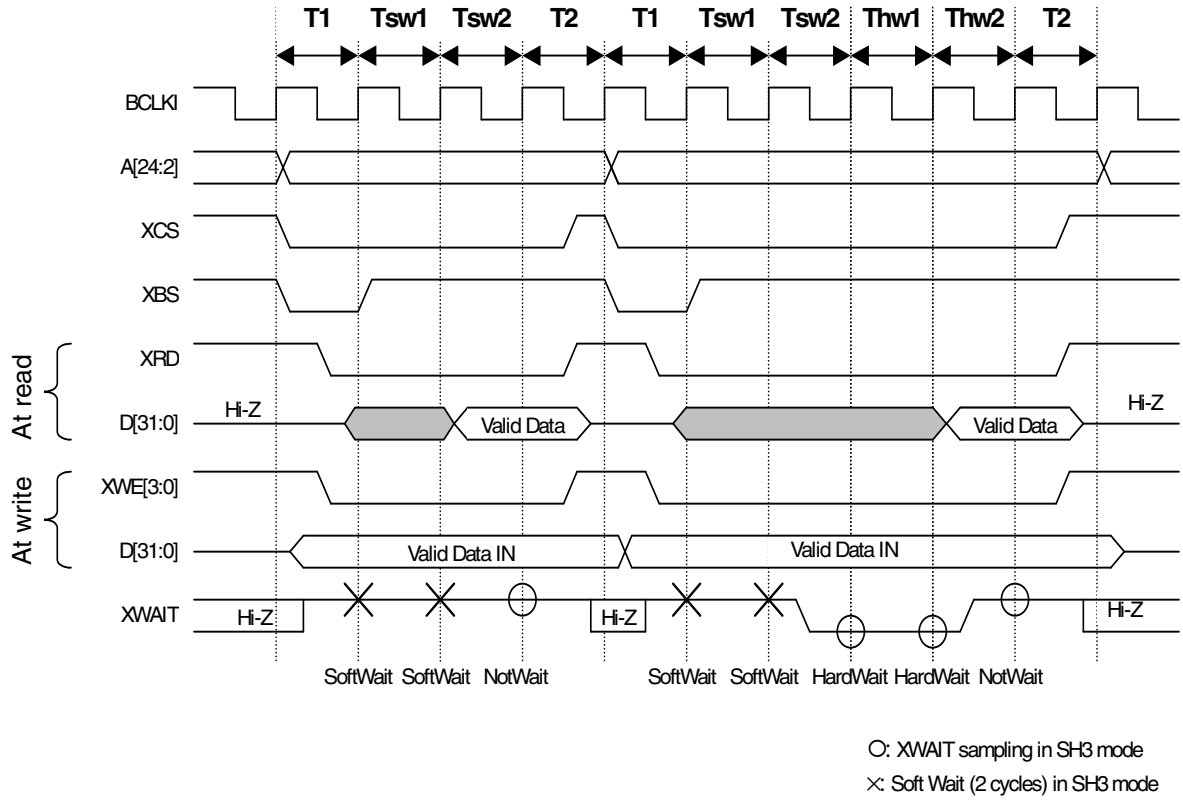
Tsw*: Software wait insertion cycle (1 cycle setting)

Thw*: Cycles inserted by hardware wait (XRDY cancels the wait state as soon as the preparations are made.)

T2: Read/write end cycle (XRDY ends in the wait state.)

Fig. 11.1 Read/Write Timing Diagram for SH3 (Normally Not Ready Mode)

11.1.2 CPU Read/Write Timing Diagram for SH3 Mode (Normally Ready Mode)



T1: Read/write start cycle (XRDY in not wait state)

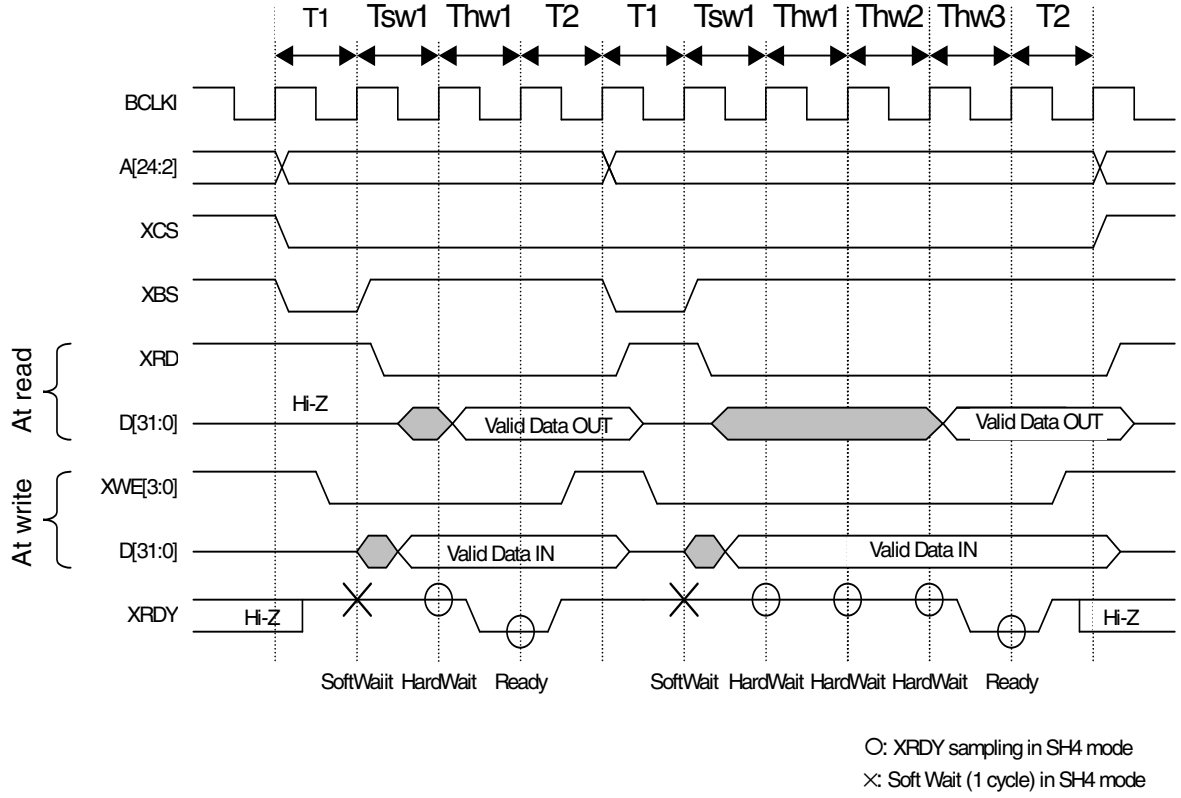
Tsw*: Software wait insertion cycle (2-cycle setting required)

Thw*: Cycles inserted by hardware wait (In hardware state when the immediate accessing is disabled)

T2: Read/write end cycle (XRDY ends in the not wait state.)

Fig. 11.2 Read/Write Timing Diagram for SH3 (Normally Ready Mode)

11.1.3 CPU Read/Write Timing Diagram for SH4 Mode (Normally Not Ready Mode)



T1: Read/write start cycle (XRDY is in the not-ready state.)

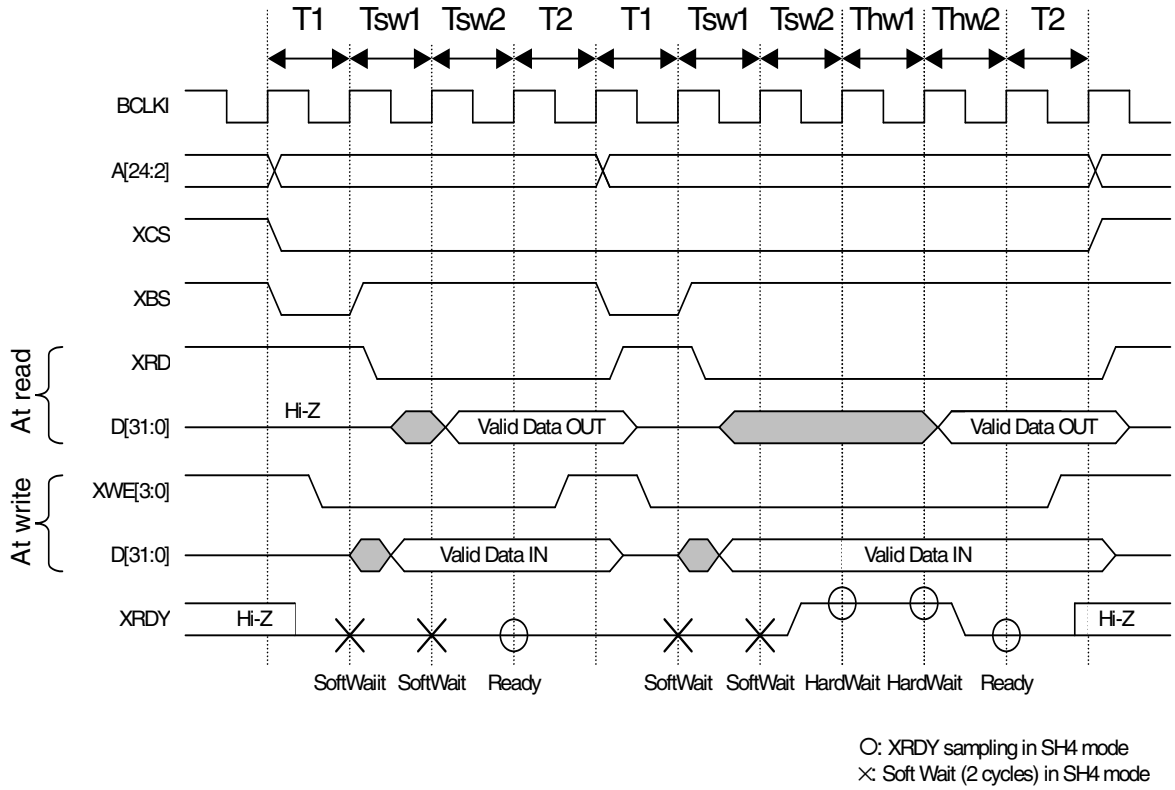
Tsw*: Software wait insertion cycle (1 cycle)

Twh*: Cycles inserted by hardware wait (XRDY asserts Ready as soon as the preparations are made.)

T2: Read/write end cycle (XRDY ends in the not-ready state.)

Fig. 11.3 Read/Write Timing Diagram for SH4 Mode (Normally Not Ready Mode)

11.1.4 CPU Read/Write Timing Diagram for SH4 Mode (Normally Ready Mode)



T1: Read/write start cycle (XRDY is in the ready state.)

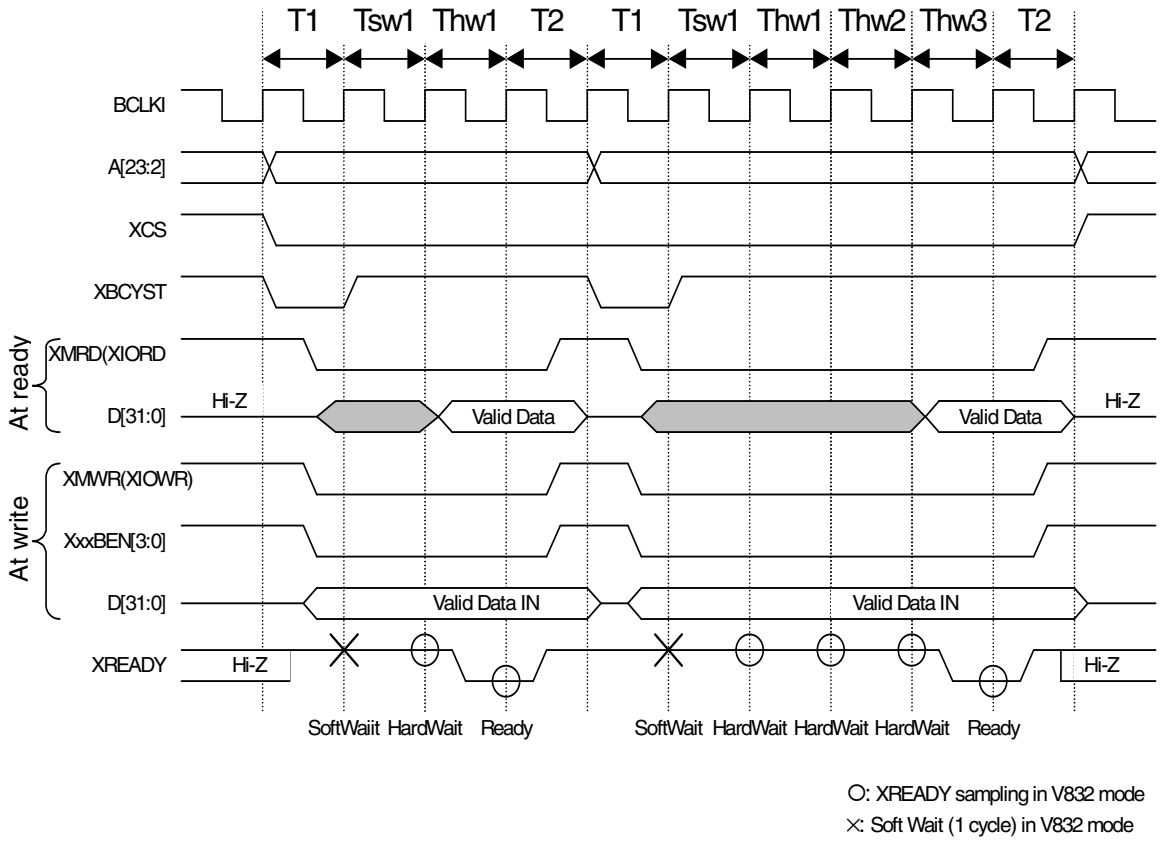
Tsw*: Software wait insertion cycle (2-cycle setting required)

Twh*: Cycles inserted by hardware (XRDY asserts Ready as soon as the preparations are made.)

T2: Read/write end cycle (XRDY ends in the ready state.)

Fig. 11.4 CPU Read/Write Timing Diagram for SH4 Mode (Normally Ready Mode)

11.1.5 CPU Read/Write Timing Diagram in V832 Mode (Normally Not Ready Mode)



T1: Read/write start cycle (XREADY is in the not-ready state.)

Tsw*: Software wait insertion cycle

Twh*: Cycles inserted by hardware wait (XREADY asserts Ready as soon as the preparations are made.)

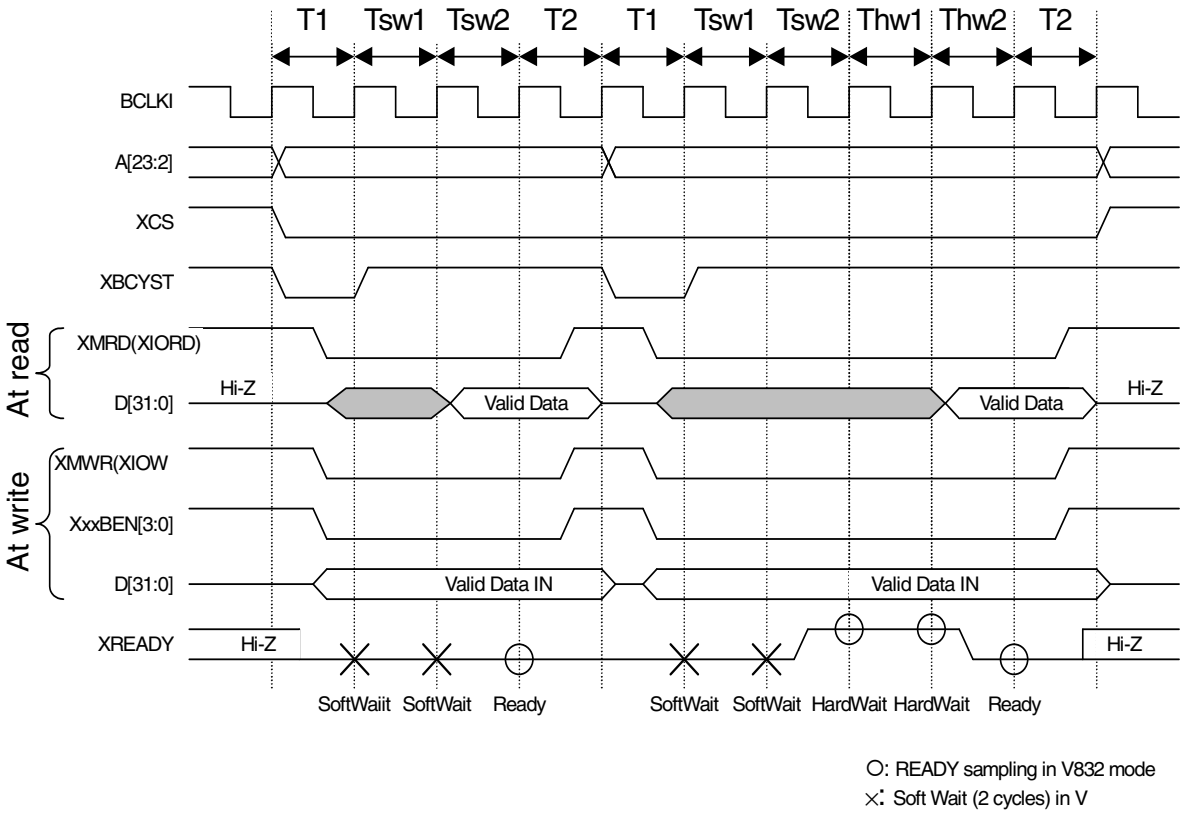
T2: Read/write end cycle (XREADY end in the not-ready state.)

Notes:1.The XxxBEN signal is used only when performing a write from the CPU; it is not used when performing a read from the CPU.

2.The CPU always inserts one cycle wait after read access.

Fig. 11.5 Read/Write Timing Diagram in V832 Mode (Normally Not Ready Mode)

11.1.6 CPU Read/Write Timing Diagram in V832 Mode (Normally Ready Mode)



T1: Read/write start cycle (XREADY is in the ready state.)

Tsw*: Software wait insertion cycle(2-cycle setting required)

Twh*: Cycles inserted by hardware wait (XREADY asserts Ready as soon as the preparations are made.)

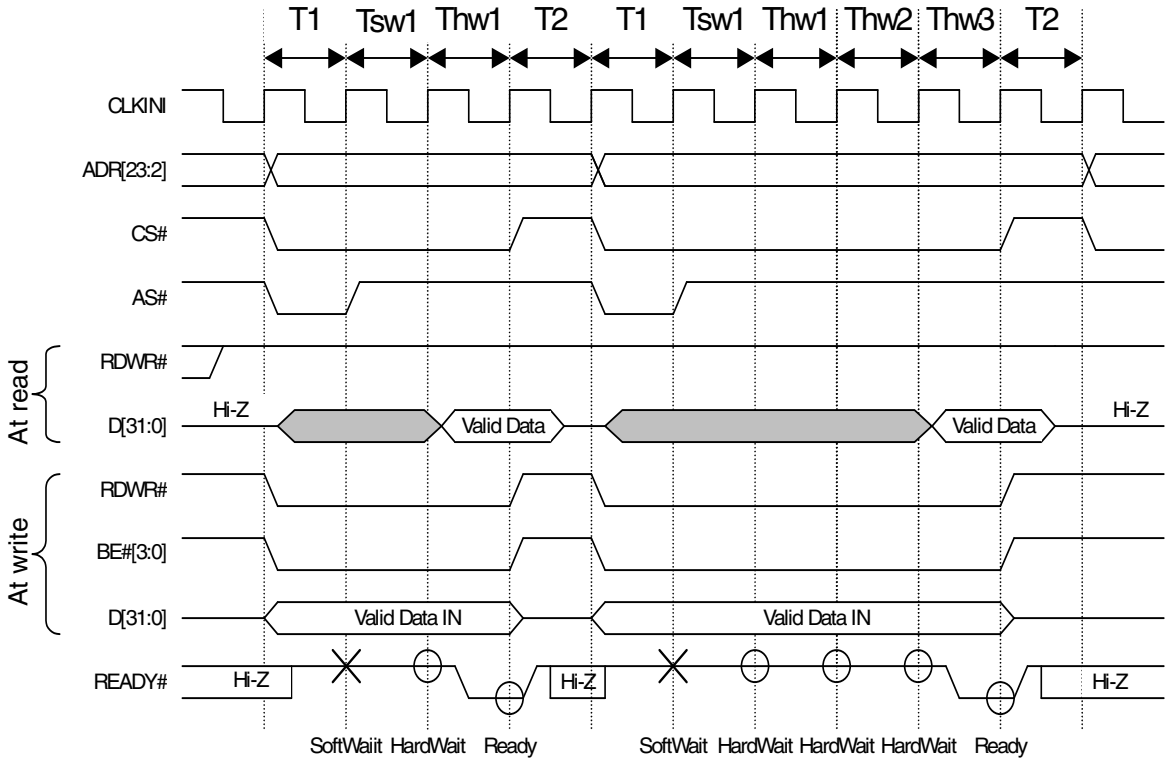
T2: Read/write end cycle (XREADY end in the ready state.)

Notes:1.The XxxBEN signal is used only when performing a write from the CPU; it is not used when performing a read from the CPU.

2.The CPU always inserts one cycle wait after read access.

Fig. 11.6 Read/Write Timing Diagram in V832 Mode (Normally Ready Mode)

11.1.7 CPU Read/Write Timing Diagram in SPARClite (Normally Not Ready Mode)



T1: Read/write start cycle (READY# is in the not-ready state.)

Tsw*: Software wait insertion cycle

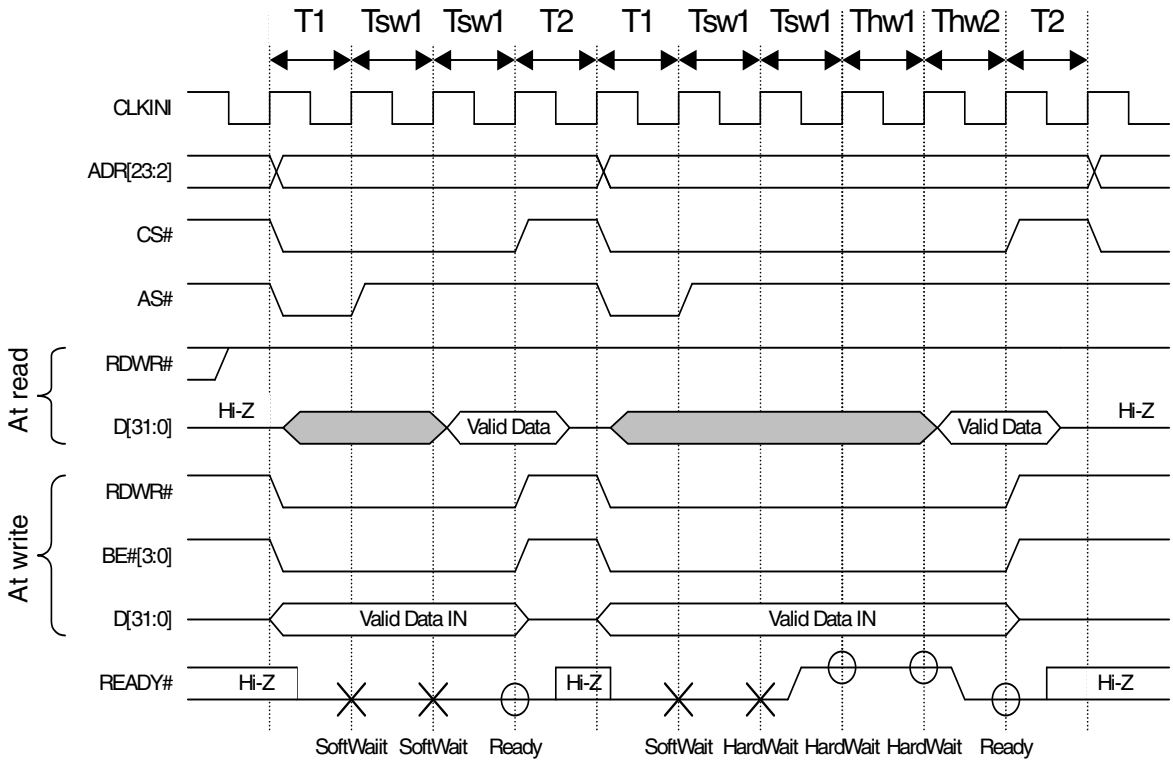
Twh*: Cycles inserted by hardware wait (READY# asserts Ready as soon as the preparations are made.)

T2: Read/write end cycle (READY# end in the not-ready state.)

Note: BE# signal is used only when performing a write from the CPU; it is not used when performing a read from the CPU.

Fig. 11.7 Read/Write Timing Diagram in SPARClite (Normally Not Ready Mode)

11.1.8 CPU Read/Write Timing Diagram in SPARClite (Normally Ready Mode)



T1: Read/write start cycle (READY# is in the ready state.)

Tsw*: Software wait insertion cycle (2-cycle setting required)

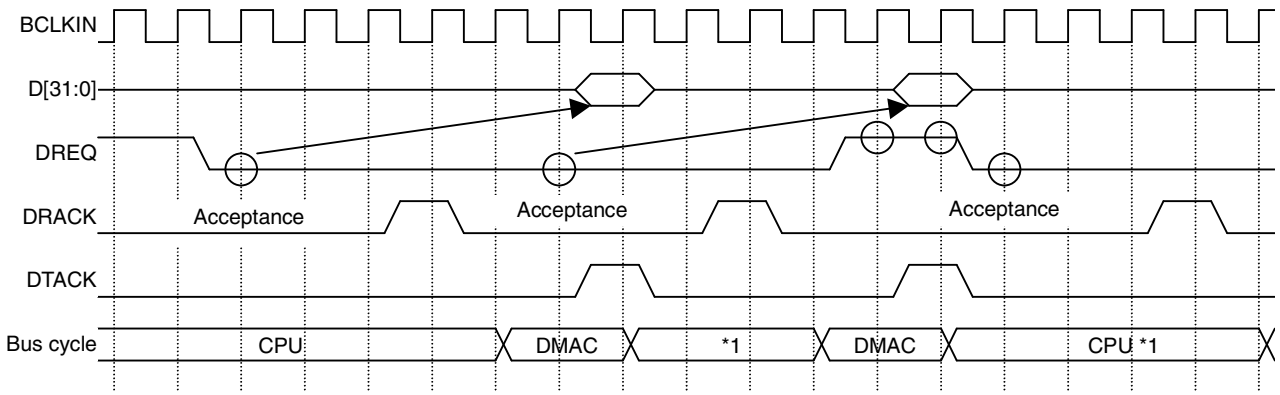
Twh*: Cycles inserted by hardware wait (READY# asserts Ready as soon as the preparations are made.)

T2: Read/write end cycle (READY# end in the ready state.)

Note: BE# signal is used only when performing a write from the CPU; it is not used when performing a read from the CPU.

Fig. 11.8 Read/Write Timing Diagram in SPARClite (Normally Ready Mode)

11.1.9 SH4 Single-address DMA Write (Transfer of 1 Long Word)



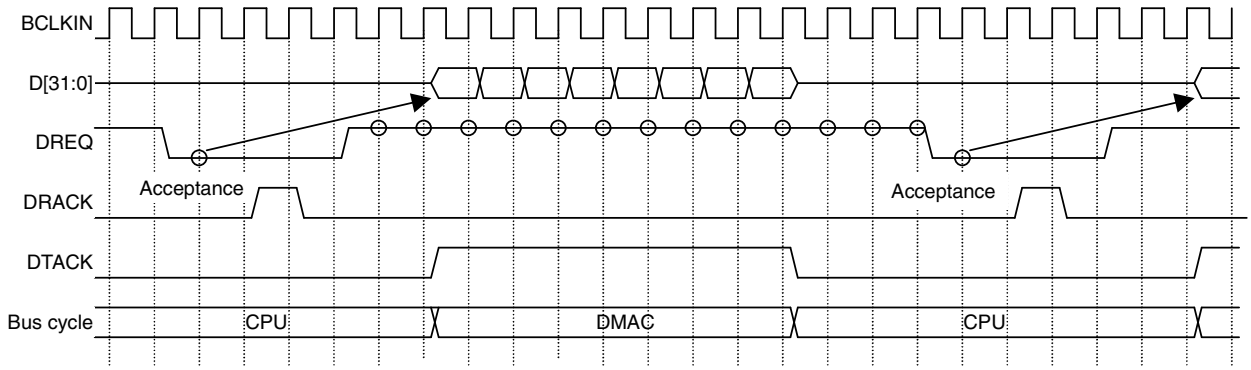
○: DREQ sampling and channel priority determination for SH mode (DREQ = level detection)

*1: In the cycle steal mode, even when DREQ is already asserted at the 2nd DREQ sampling, the right to use the bus is returned to the CPU once. In the burst mode, DMAC secures the right to use the bus unless DREQ is negated.

Fig. 11.9 SH4 Single-address DMA Write (Transfer of 1 Long Word)

ORCHID writes data according to the DTACK assert timing. When data cannot be received, the DREQ signal is automatically negated. And then the DREQ signal is reasserted as soon as data becomes ready to be received.

11.1.10 SH4 Single-address DMA Write (Transfer of 8 Long Words)

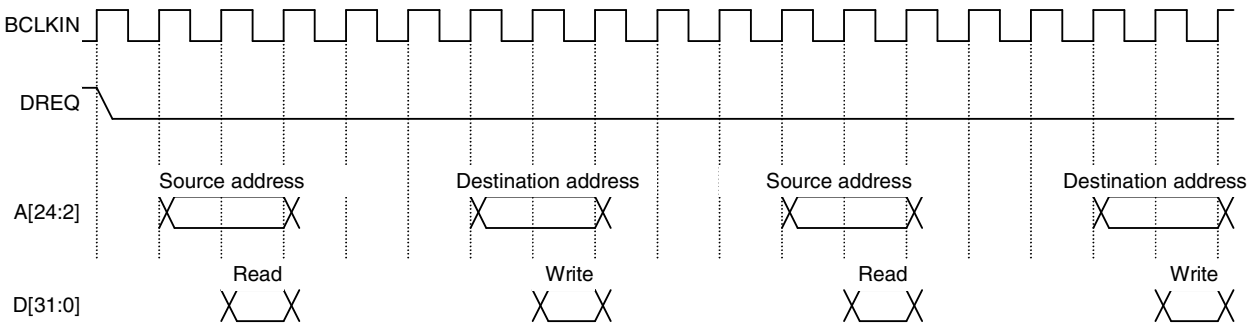


○: DREQ sampling and channel priority determination for SH mode (DREQ = level detection)

Fig. 11.10 SH4 Single-address DMA Write (Transfer of 8 Long Words)

After the CPU has asserted DRACK, ORCHID negates DREQ and receives 32-byte data according to the DTACK assertion timing. As soon as the next data is ready to be received, ORCHID reasserts DREQ but the reassertion timing depends on the internal status.

11.1.11 SH3/4 Dual-address DMA (Transfer of 1 Long Word)

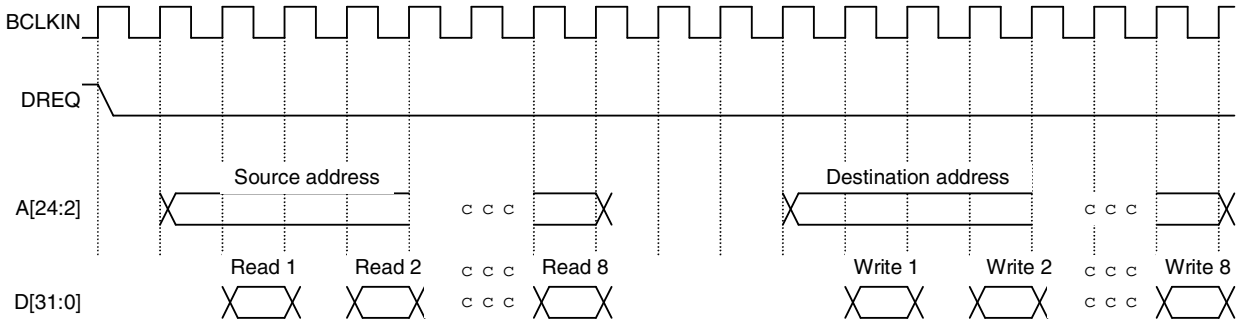


For the MB86290A, the read/write operation is performed according to the SRAM protocol.

Fig. 11.11 SH3/4 Dual-address DMA (Transfer of 1 Long Word)

In the dual-address mode, the DREQ signal is kept asserted until the transfer ends by default. Consequently, to negate the DREQ signal when ORCHID cannot return the Ready signal immediately, set the DBM register.

11.1.12 SH3/4 Dual-Address DMA (Transfer of 8 Long Words)

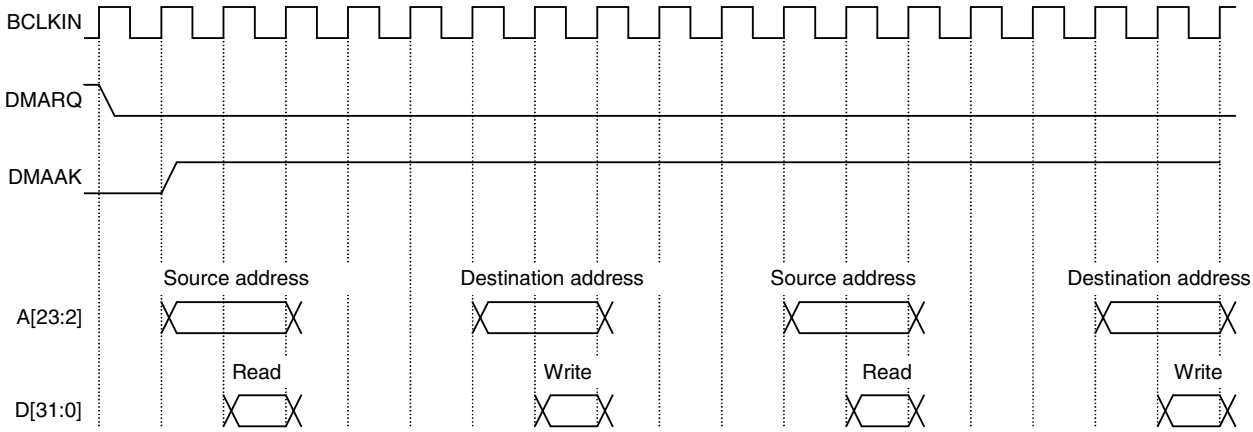


For the MB86290A, the read/write operation is performed according to the SRAM protocol.

Fig. 11.12 SH3/4 Dual-address DMA (Transfer of 8 Long Words)

In the dual-address mode, the DREQ signal is kept asserted until the transfer ends by default. Consequently, to negate the DREQ signal when ORCHID cannot return the Ready signal immediately, set the DBM register.

11.1.13 V832 DMA Transfer

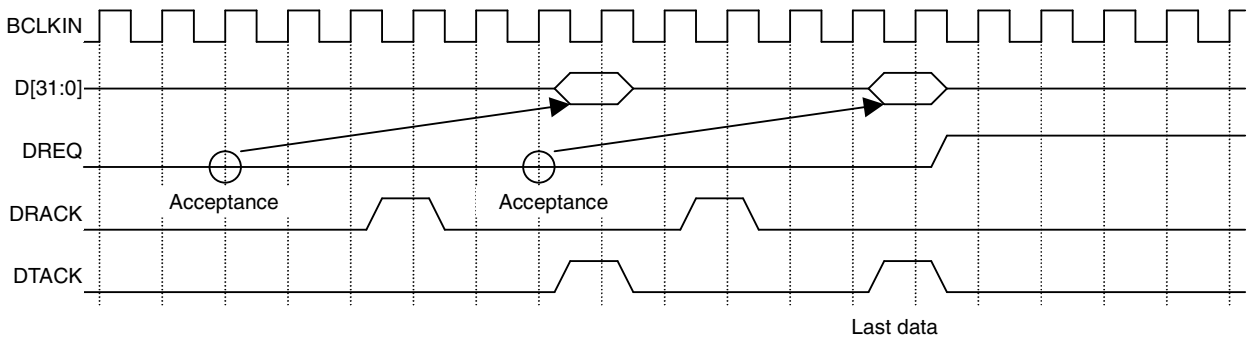


For the MB86290A, the read/write operation is performed according to the SRAM protocol.

Fig. 11.13 V832 DMA Transfer

During DMA transfer, the DREQ signal is kept asserted until the transfer ends by default. Consequently, to negate the DREQ signal when ORCHID cannot return the Ready signal immediately, set the DBM register.

11.1.14 SH4 Single-address DMA Transfer End Timing

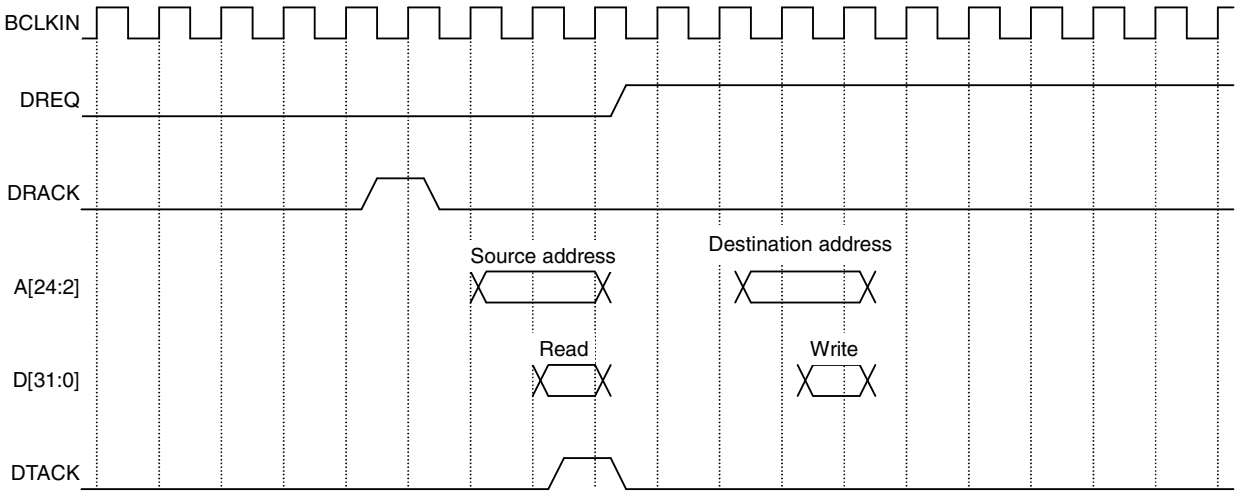


○: DREQ sampling and channel priority determination for SH mode (DREQ = level detection)

Fig. 11.14 SH4 Single-address DMA Transfer End Timing

DREQ is negated three cycles after DRACK is written as the last data.

11.1.15 SH3/4 Dual-address DMA Transfer End Timing



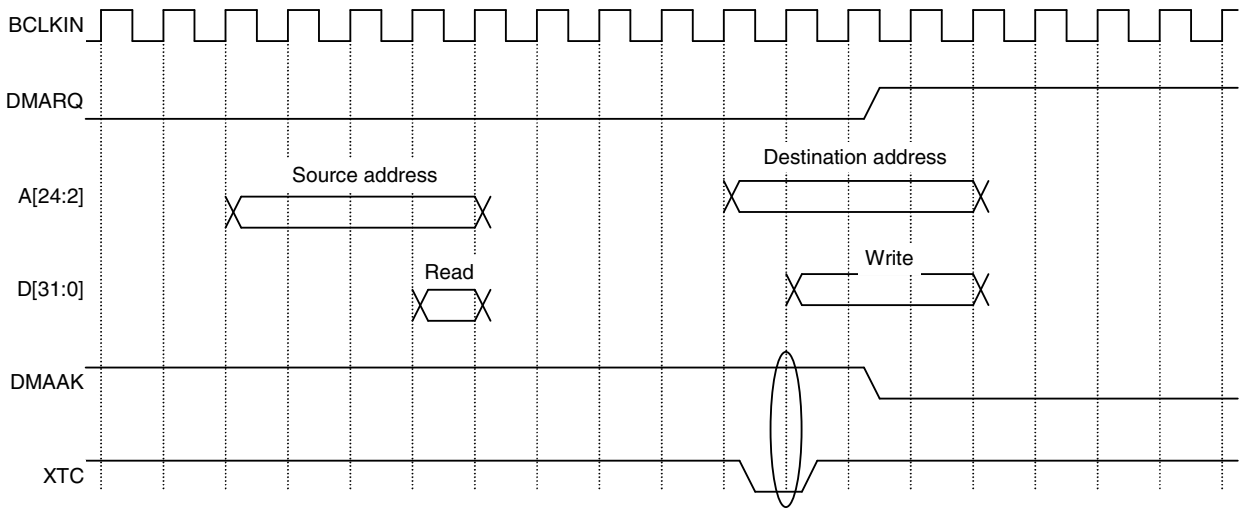
For the MB86290A, the read/write operation is performed according to the SRAM protocol.

Fig. 11.15 SH3/4 Dual-address DMA Transfer End Timing

DREQ is negated three cycles after DRACK is written as the last data.

Note: When the dual address mode (DMA) is used, the DTACK signal is not used.

11.1.16 V832 DMA Transfer End Timing



For the MB86290A, the read/write operation is performed according to the SRAM protocol.

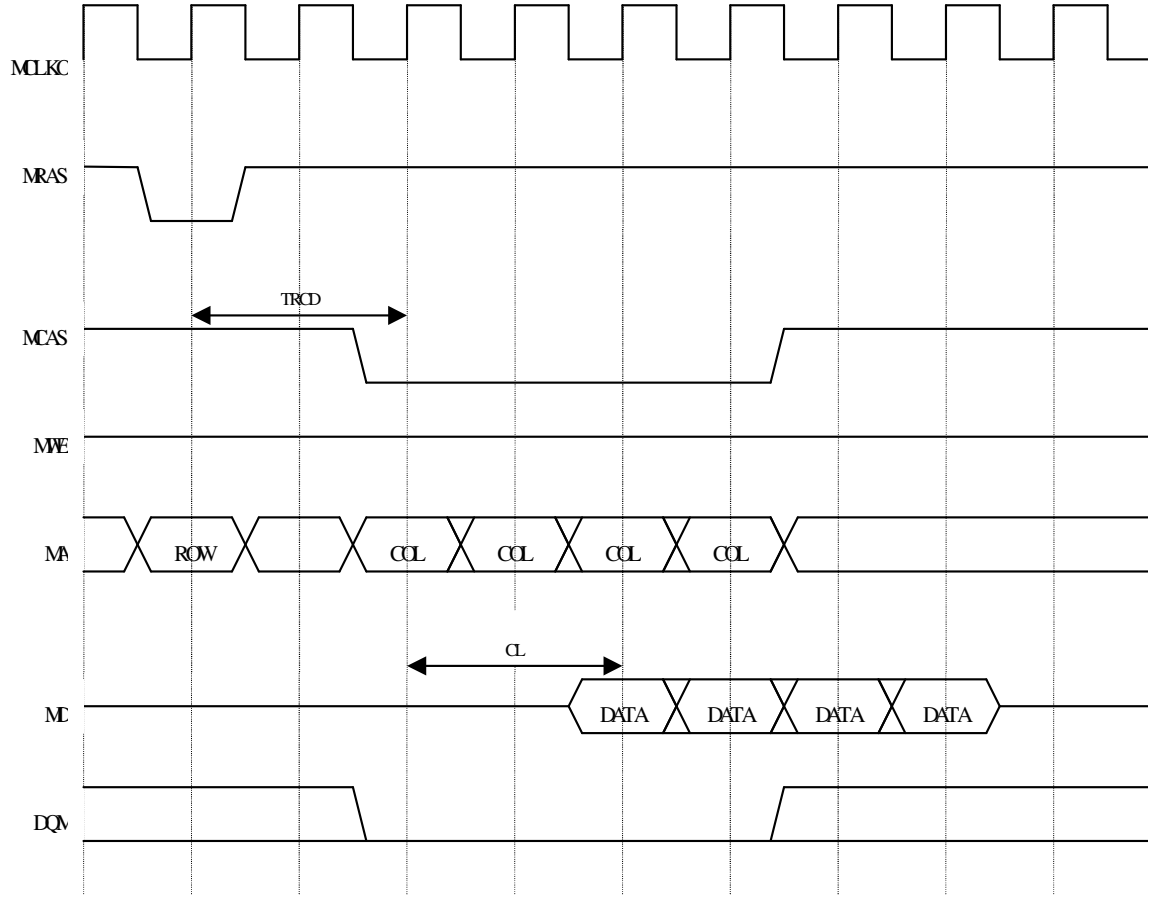
Fig. 11.16 V832 DMA Transfer End Timing

DMMAK and XTC are ANDed inside ORCHID to end DMA.

11.2 Graphic Memory Interface

Access timing between ORCHID and graphic memory is described.

11.2.1 Read access timing to same ROW address



ROW Row Address TRCD RAS to CAS Delay Time

COL Column Address CL CAS Latency

DATA READ DATA

Fig.11.17 Read access timing to same ROW address

This diagram is Timing chart in case of reading address in same ROW address of SDRAM from ORCHID four times. ACTV command is issued, and READ command is issued after TRCD. READ command is issued, and DATA is outputted after CL that is fetched in.

11.2.2 Read access timing to different ROW address

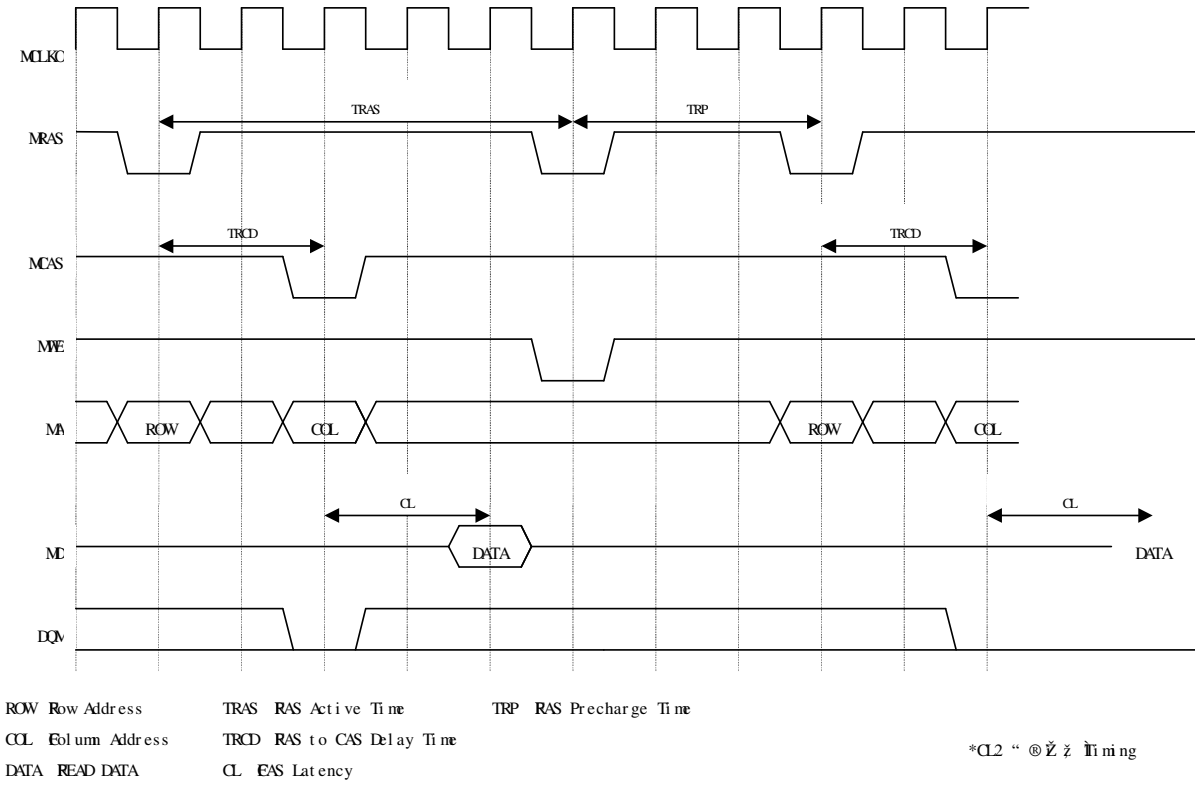


Fig.11.18 Read access timing to different ROW address

This diagram is Timing chart in case of reading address in different ROW address of SDRAM from ORCHID four times. ACTV command is issued, and READ command is issued after TRCD. READ command is issued, and DATA is outputted after CL that is fetched in. Because address demanding first READ and address demanding next READ step over page boundary of DRAM, When the timing of TRAS, Precharge command is issued, after TRP ACTV command is reissued and READ command is issued.

11.2.3 Write access timing to same ROW address

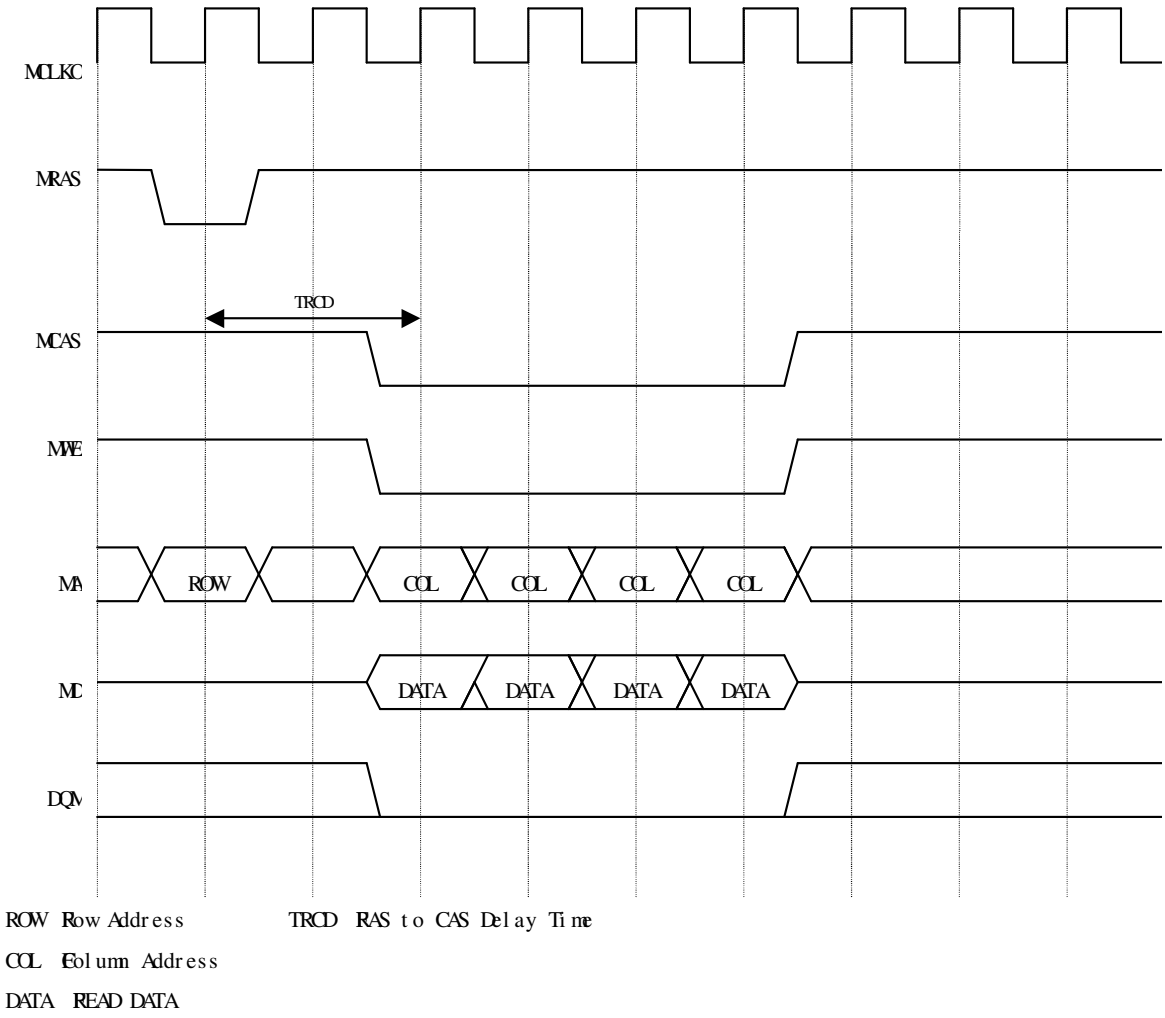
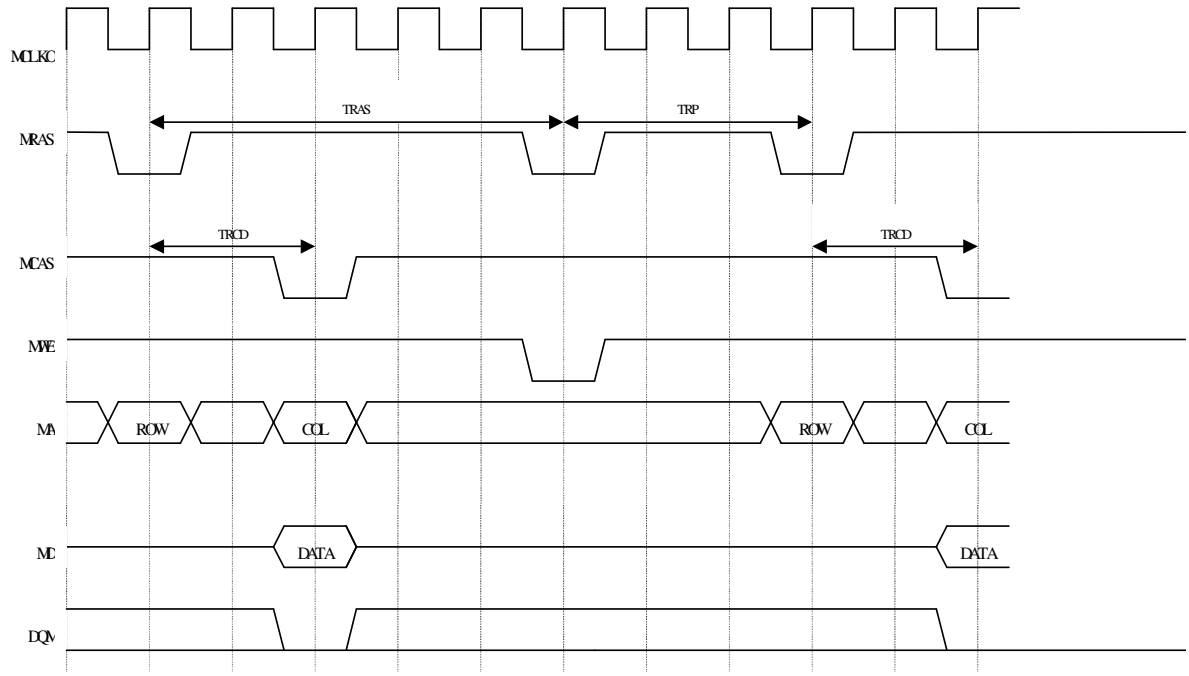


Fig.11.19 Write access timing to same ROW address

This diagram is Timing chart in case of writing address in same ROW address of SDRAM from ORCHID four times. ACTV command is issued, and WRITE command is issued after TRCD. Then SDRAM is written in.

11.2.4 Write access timing to different ROW address



ROW Row Address
COL Column Address
DATA READ DATA
TRAS RAS Active Time
TRCD RAS to CAS Delay Time
TRP RAS Precharge Time

Fig.11.20 Write access timing to different ROW address

This diagram is Timing chart in case of writing address in different ROW address of SDRAM from ORCHID. ACTV command is issued, and READ command is issued after TRCD. Because address demanding first WRITE and address demanding next WRITE step over page boundary of DRAM, When the timing of TRAS, Precharge command is issued, after TRP ACTV command is reissued and WITE command is issued.

11.2.5 READ and Write access timing to same ROW address

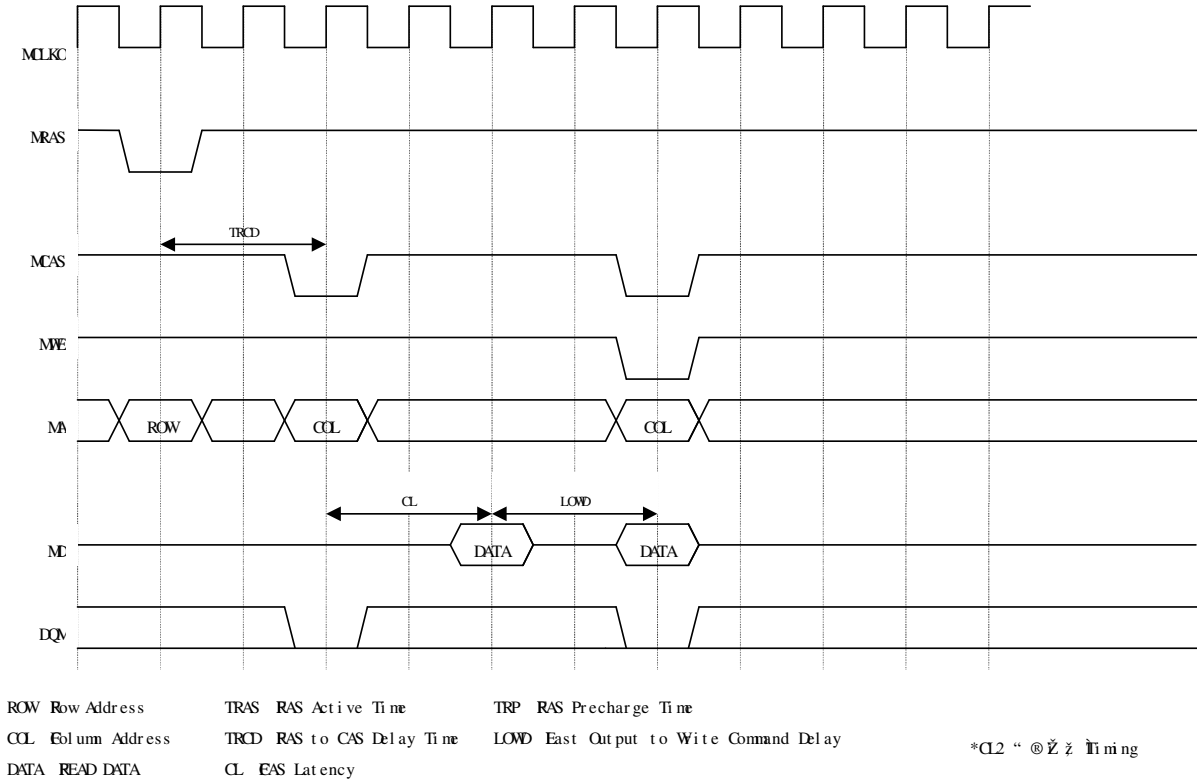
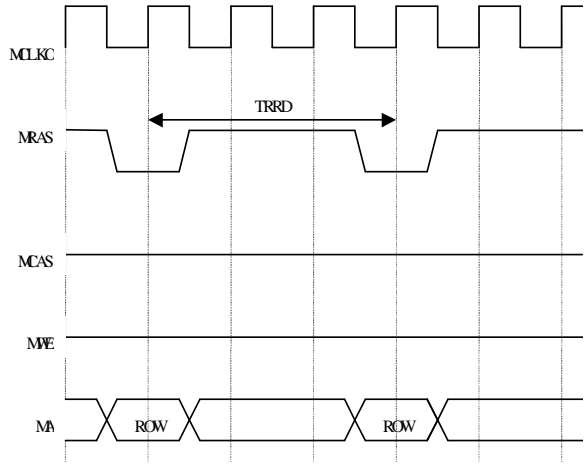


Fig.11.21 READ and Write access timing to same ROW address

This diagram is Timing chart when reading address in same ROW address of SDRAM from ORCHID is done, as soon as write is done. DATA is outputted, after LOWD WRITE command is issued.

11.2.6 DELAY of ACTV command

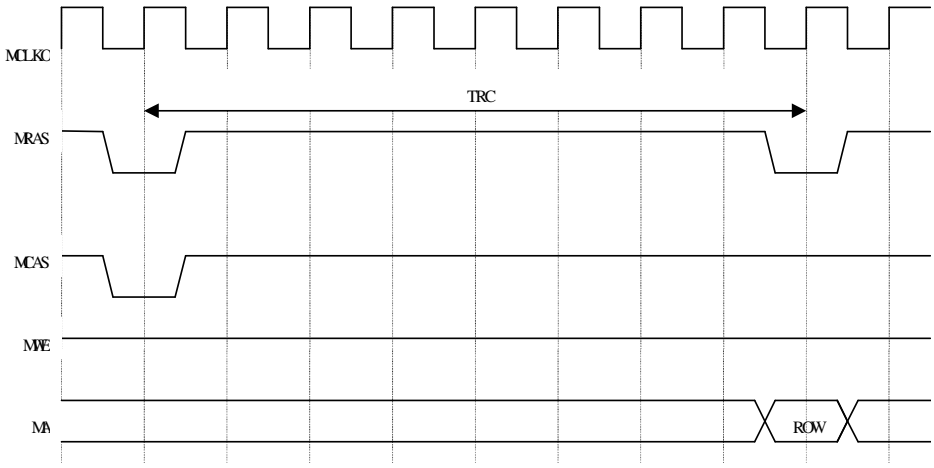


ROW Row Address
TRRD RAS to RAS Bank Active Delay Time

Fig.11.22 DELAY of ACTV command

ACTV command for ROW address of SDRAM from ORCHID is issued after TRRD from previous ACTV command.

11.2.7 DELAY of next ACTV command from Refresh command

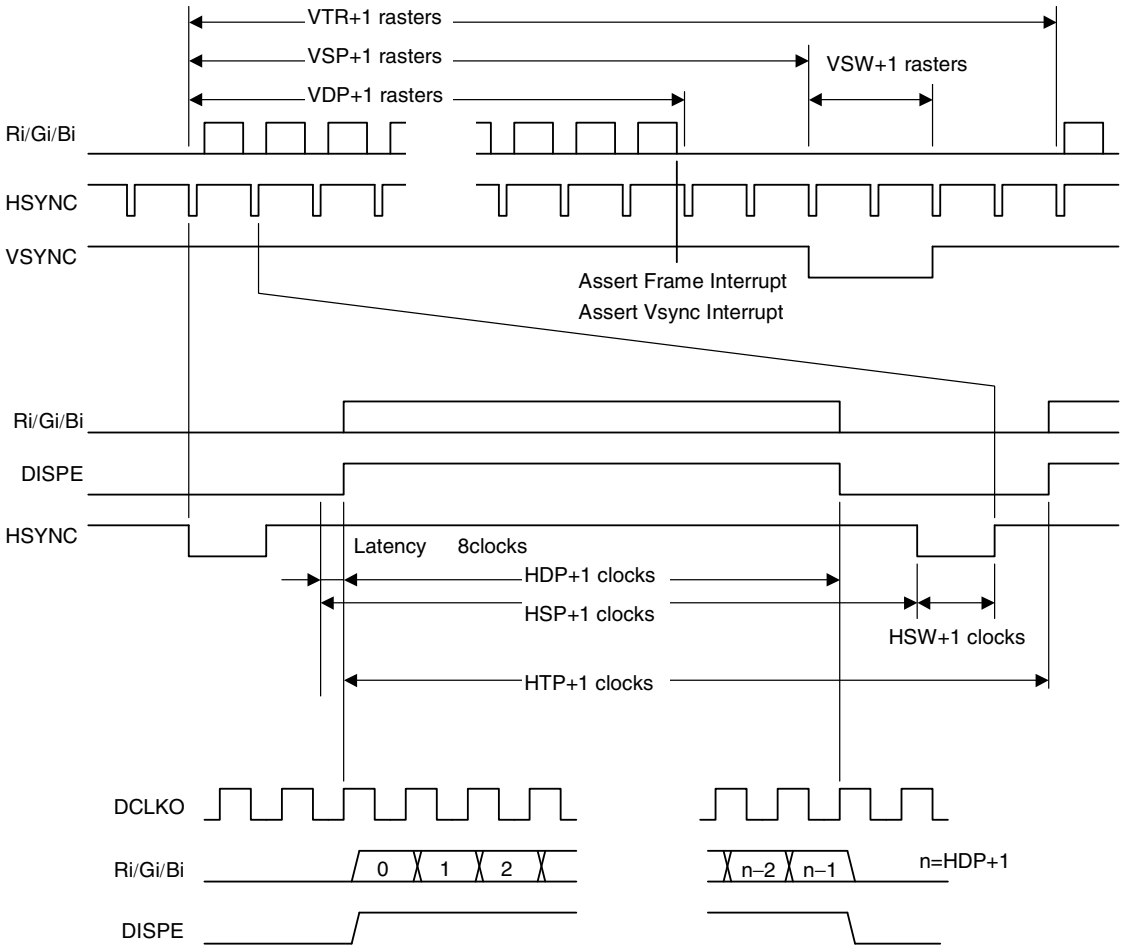


ROW Row Address
TRC RAS Cycle Time

Fig.11.23 DELAY of next ACTV command from Refresh command

After Refresh command is issued, ACTV command is issued after TRC.

11.3 Display Timing

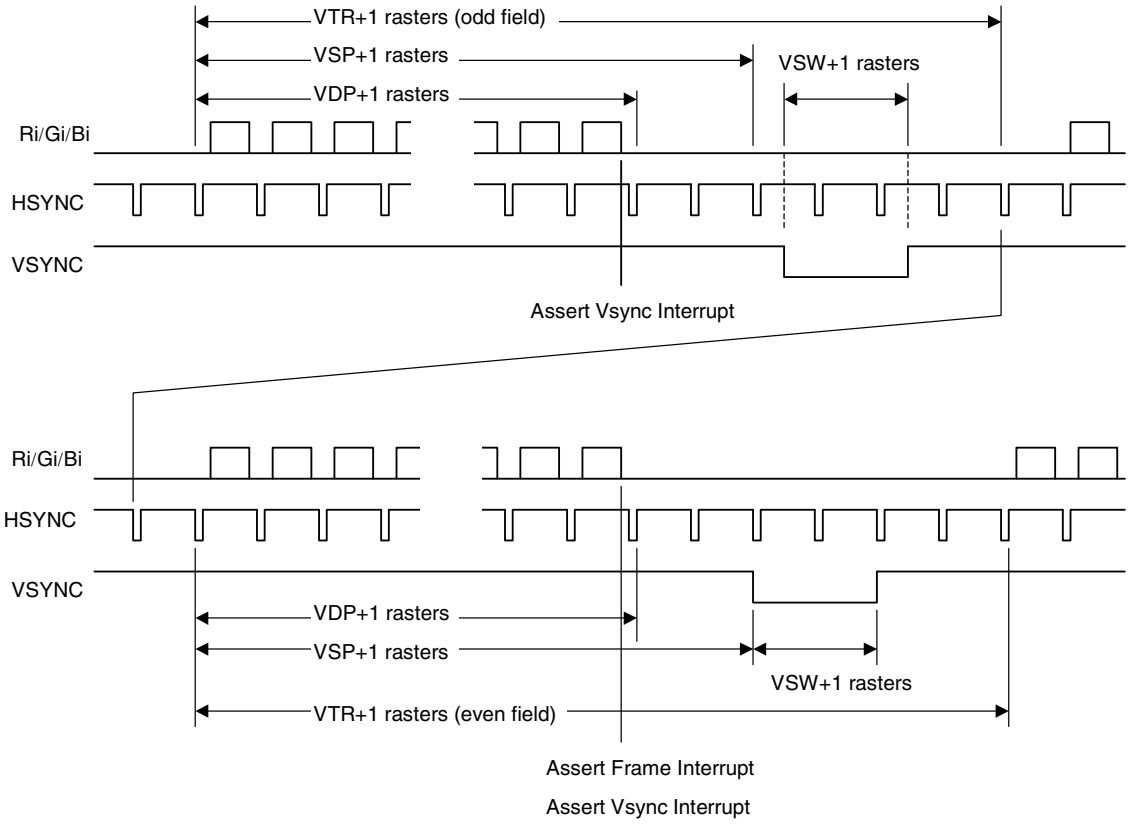


11.3.1 Non-interlaced Video Mode

In the above diagram, VTR, HDP, etc., are the settings of their associated registers.

The VSYNC/frame interrupt is asserted when display of the last raster ends. When updating display parameters, synchronize with the frame interrupt so no display disturbance occurs. Calculation for the next frame is started immediately after the vertical synchronization pulse is asserted, so the parameters must be updated by the time that calculation is started.

11.3.2 Interlaced Video Mode



In the above diagram, VTR, HDP, etc., are the settings of their associated registers.

Interlaced mode is also operated in same timing. But output graphic data is different.

11.3.3 Composite Sync Signal

When EEQ bit of DCM resistor is 0, CSYNC output signal is shown bellow.

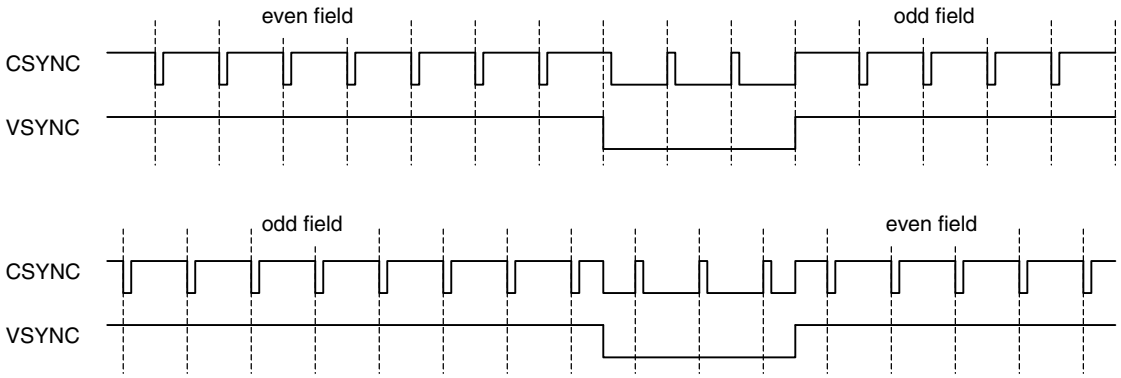


Fig. 11.24 Composite sync signal / without equalization pulse

When EEQ bit of DCM resistor is 1, CSYNC output signal is shown bellow.

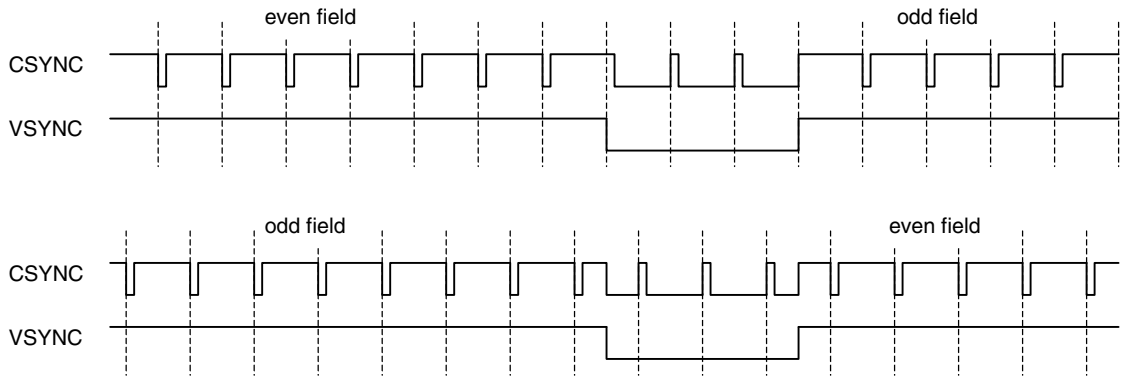


Fig. 11.17 Composite sync signal / with equalization pulse

Equalization pulse start to be inserted when vertical retrace line period start. It is also inserted three times after vertical sync period.

Cautions

11.4 CPU Cautions

1) Enable the hardware wait for the areas to which ORCHID is linked. When the normal not read mode (MODE[2] = 0) is used, set the software wait count to 1. When the normal ready mode (MODE[2] = 1) is used, set the count to 2.

2) When starting DMA by issuing an external request, do so after setting the transfer count register (DTCR) and mode setting register (DSUR) of ORCHID to the same value as the CPU setting. In the V832 mode, there is no need to set DTCR.

3) When ORCHID is read-/write-accessed from the CPU during DMA transfer, do not access the registers and memories related to DMA transfer. If these registers and memories are accessed, reading and writing of the correct value is not assured.

4) In the SH mode, only the lowers 32 MB are used (A[25] is not used), so do not access the uppers 32 MB. When linking other devices to the uppers 32 MB, create Chip Select for ORCHID by using glue logic.

5) Set DREQ (DMARQ) to detection.

6) Set the SH-mode DACK/DRACK to high active output, V832-mode DMAAK to high active, and V832-mode TC to low active.

11.5 SH3 Mode

1) When the XRDY pin is low, it is in the wait state.

2) DMA transfer in the single-address mode is not supported.

3) DMA transfer in the dual-address mode supports the direct address transfer mode, but does not support the indirect address transfer mode.

4) 16-byte DMA transfer in the dual-address mode is not supported.

5) The XINT signal is low active.

11.6 SH4 Mode

1) When the XRDY pin is low, it is in the ready state.

2) At DMA transfer in the single-address mode, transfer from the main memory (SH-mode memory) to FIFO of ORCHID can be performed, but transfer from ORCHID to the main memory cannot be performed.

3) DMA transfer in the single-address mode is performed in units of 32 bits or 32 bytes.

4) SH4-mode 32-byte DMA transfer in the dual-address mode supports inter-memory transfer, but does not support transfer from memory to FIFO.

5) The XINT signal is low active.

11.7 V832 Mode

1) When the XRDY pin is low, it is in the ready state.

2) Set the active level of DMAAK to high-active in V832 mode.

3) DMA transfer supports the single transfer mode and demand transfer mode.

4) The XINT signal is high-active. Set the V832-mode registers to high-level trigger.

11.8 SPARClite

1) When the XRDY pin is low, it is in the ready state.

2) The SPARClite does not support the DMA transfer that issues the DREQ.

3) The XINT signal asserts the low active signal

11.9 DMA Transfer Modes Supported by SH3, SH4, and V832

	Single-address mode	Dual-address mode
SH3	SH 3 does not support the single-address mode.	SH3 supports the direct address transfer mode; it does not support the indirect address transfer mode. Transfer is performed in 32-bit units. SH3 supports the cycle steal mode and burst mode.
SH4	Transfer is performed in units of 32 bits or 32 bytes. SH4 supports the cycle steal mode and burst mode.	Transfer is performed in 32-bit units. Transfer to memory is performed in 32-byte units. SH4 supports transfer to FIFO. SH4 supports the cycle steal mode and burst mode.
V832		Transfer is performed in 32-bit units. V832 supports the single transfer mode and demand transfer mode.
SPARClite		

Fig.0-1 Table of DMA Transfer Modes

12. Electrical Characteristics

12.1 Absolute Maximum Ratings

Maximum Ratings

Parameter	Symbol	Maximum Ratings	Unit
Supply voltage	VDDL[*1] VDDH	-0.5 < VDDL < 3.0 -0.5 < VDDH < 4.0	V
Input voltage	VI	-0.5 < VI < VDDH+0.5 (<4.0)	V
Output current	IO	±13	MA
Power current	IPOW	60	MA
Storage temperature	TST	-55 < TST < +125	°C

*1 Includes analog power supply and PLL power supply

12.2 Recommended Operating Conditions

12.2.1 Recommended Operating Conditions

Recommended Operating Conditions

Parameter	Symbol	Rating			Unit
		Min.	Typ.	Max.	
Supply voltage	VDDL *1 VDDH	2.3 3.0	2.5 3.3	2.7 3.6	V
Input voltage (High level)	VIH	2.0		VDDH+0.3	V
Input voltage (low level)	VIL	-0.3		0.8	V
Ambient temperature	TA	-40		85	°C

*1 Includes PLL power supply

12.2.2 Power-on Precautions

- There is no restriction on the order of power-on/power-off between VDDL and VDDH. However, do not supply only VDDH for more than a few seconds.
- After power-on, hold the S input at the 'L' level for at least 500 ns. Then, after setting the S-input to the 'H' level, hold the XRESET input at the 'L' level for at least 300 µs.

12.3 DC Characteristics

Condition: VDDL = 2.5 ± 0.2 V, VDDH = 3.3 0.3 V, VSS = 0.0 V, Ta = 0-70°C

Parameter	Symbol	Ratings			Unit
		Min.	Typ.	Max.	
Output voltage[*1] (High level)	VOH	VDDH-0.2		VDDH	V
Output voltage[*2] (Low level)	VOL	0.0		0.2	V
Output current (High level)	IOH1[*3] IOH2[*4] IOH3[*5]	-2.0 -4.0 -8.0			mA
Output current (Low level)	IOL1[*3] IOL2[*4] IOL3[*5]	2.0 4.0 8.0			mA
Input leakage current	IL			+-5	µA
Load capacitance	C			16	pF

*1 IOH = -100 µA

*2 IOL = 100 µA

*3 Output characteristic of XINT, DREQ, XRDY

*4 Output characteristic of all signals except *3.

*5 Output characteristic of MCLKO

12.4 AC Characteristics

12.4.1 Host Interface

Clock

Parameter	Symbol	Condition	Limit Value			Unit
			Min.	Typ.	Max.	
BCLKI Frequency	f_{BCLKI}				100	MHz
BCLKI H-width	t_{HBCLKI}		1			ns
BCLKI L-width	t_{LBCLKI}		1			ns

Host interface signals

(External load: 20 pF)

Parameter	Symbol	Condition	Limit Value			Unit
			Min.	Typ.	Max.	
Address set up time	t_{ADS}		3.0			ns
Address hold time	t_{ADH}		0.0			ns
XBS Set up time	t_{BSS}		3.5			ns
XBS Hold time	t_{BSH}		0.0			ns
XCS Set up time	t_{CSS}		3.5			ns
XCS Hold time	t_{CSH}		0.0			ns
XRD Set up time	t_{RDS}		3.0			ns
XRD Hold time	t_{RDH}		0.0			ns
XWE Set up time	t_{WES}		5.5			ns
XWE Hold time	t_{WEH}		0.0			ns
Write data set up time	t_{WDS}		3.5			ns
Write data hold time	t_{WDH}		0.0			ns
DTACK Set up time	t_{DAKS}		3.5			ns
DTACK Hold time	t_{DAKH}		0.0			ns
DRACK Set up time	t_{DRKS}		4.0			ns
DRACK Hold time	t_{DRKH}		0.0			ns
Read data delay time (for XRD)	t_{RDDZ}		2.5		8.5	ns
Read data delay time	t_{RDD}		4.0		10.5	ns
XRDY Delay time (for XCS)	t_{RDYDZ}		2.0		6.0	ns
XRDY Delay time	t_{RDYD}		2.5		6.5	ns
XINT Delay time	t_{INTD}		2.5		7.0	ns
DREQ Delay time	t_{DQRD}		2.5		6.5	ns
MODE Hold time	t_{MODH}	*1			20.0	ns

*1 Hold time requirement for RESET release

12.4.2 Video Interface

Clock

Parameter	Symbol	Condition	Limit Value			Unit
			Min.	Typ.	Max.	
CLK Frequency	f_{CLK}			14.318		MHz
CLK H-width	t_{HCLK}		25			ns
CLK L-width	t_{LCLK}		25			ns
DCLKI Frequency	f_{DCLKI}				67	MHz
DCLKI H-width	t_{HDCLKI}		5			ns
DCLKI L-width	t_{LDCLKI}		5			ns
DCLKO frequency	f_{DCLKO}				67	MHz

Input signals

Parameter	Symbol	Condition	Limit Value			Unit
			Min.	Typ.	Max.	
HSYNC Input pulse width	$t_{WHSYNC0}$	*1	3			clock
	$t_{WHSYNC1}$	*2	3			clock
HSYNC Input set up time	t_{SHSYNC}	*2	10			ns
HSYNC Input hold time	t_{HHSYNC}	*2	10			ns
VSYNC Input pulse width	$t_{WHSYNC1}$		1			HSYNC period

*1 In PLL synchronization mode (CKS = 0), base clock output from internal PLL (period = 1/14*fCLK)

*2 In DCLKI synchronization mode (CKS = 1), base clock = DCLKI

*3 For VSYNC negation edge

Output signals

Parameter	Symbol	Condition	Limit Value			Unit
			Min.	Typ.	Max.	
RGB Output delay time	T_{RGD}		2		10	ns
DISPE Output delay time	t_{DEO}		2		10	ns
HSYNC Output delay time	t_{DHSYNC}		2		10	ns
VSYNC Output delay time	t_{DVSYNC}		2		10	ns
CSYNC Output delay time	t_{DCSYNC}		2		10	ns
GV Output delay time	t_{DGV}		2		10	ns

12.4.3 Video Capture Interface

Clock

Parameter	Symbol	Condition	Limit Value			Unit
			Min.	Typ.	Max.	
MCLKO frequency	f_{MCLK}				*1	MHz
MCLKO H period	t_{HMCLKO}		1.0			ns
MCLKO L period	t_{LMCLKO}		1.0			ns
MCLKI frequency	f_{MCLKI}				*1	MHz
MCLKI H period	t_{HMCLKI}		1.0			ns
MCLKI L period	t_{LMCLKI}		1.0			ns
delay of MCLKI for MCLKO	t_{OID}		0.0		3.5	ns

Input/output signals

Parameter	Symbol	Condition	Limit Value			Unit
			Min.	Typ.	Max.	
MA,MRAS,MCAS,MWE Setup time	t_{MADS}	*1	3.2			ns
MA,MRAS,MCAS,MWE Hold time	t_{MADH}	*1	1.3			ns
MDQM data setup time	t_{MDQMDS}	*1	3.2			ns
MDQM data hold time	t_{MDQMDH}	*1	1.3			ns
MD output data setup time	t_{MDODS}	*1	3.2			ns
MD output data hold time	t_{MDODH}	*1	1.3			ns
MD input data setup time	t_{MDIDS}	*2	3.0			ns
MD output data hold time	t_{MDIDH}	*2	1.0			ns

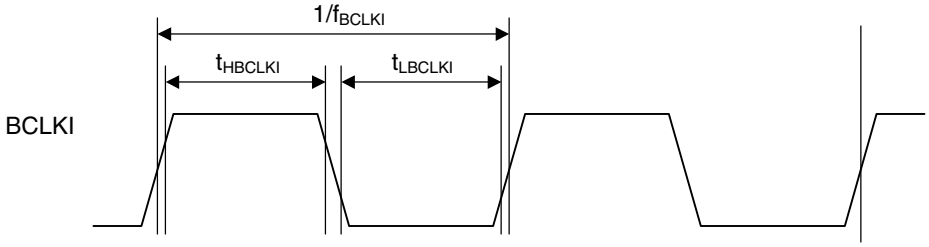
12.4.4 PLL Specifications

Parameter	Limit Value	Description
Input frequency (typ.)	14.31818 MHz	
Output frequency	200.45452 MHz	x 14
Duty ratio	101.3~93.1%	H/L Pulse width ratio of PLL output
Jitter	180~-150ps	Frequency tolerant of two consecutive clock cycles

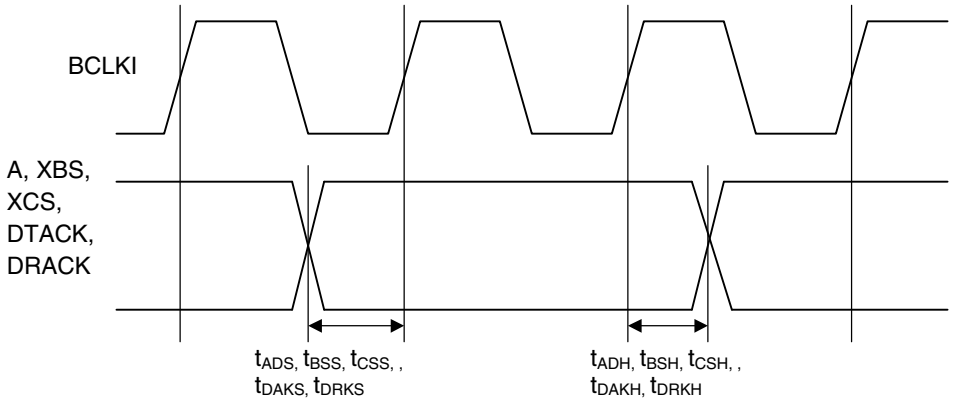
12.5 Timing Diagram

12.5.1 Host Interface

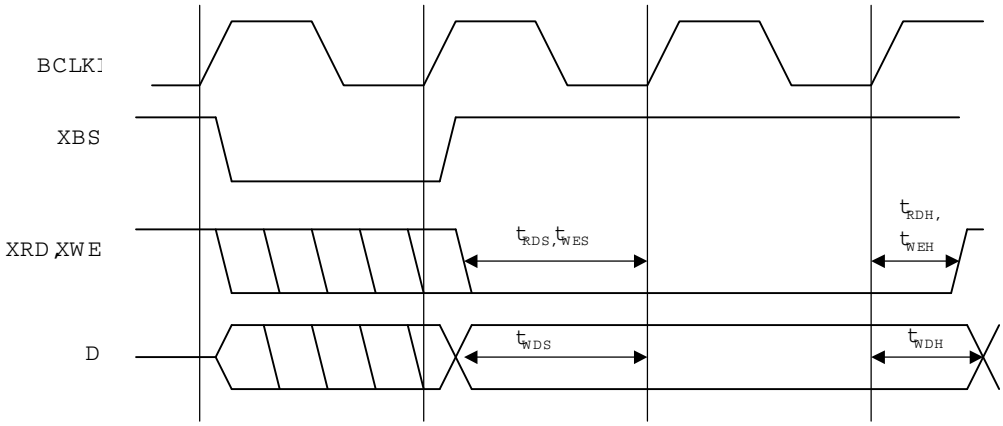
Clock



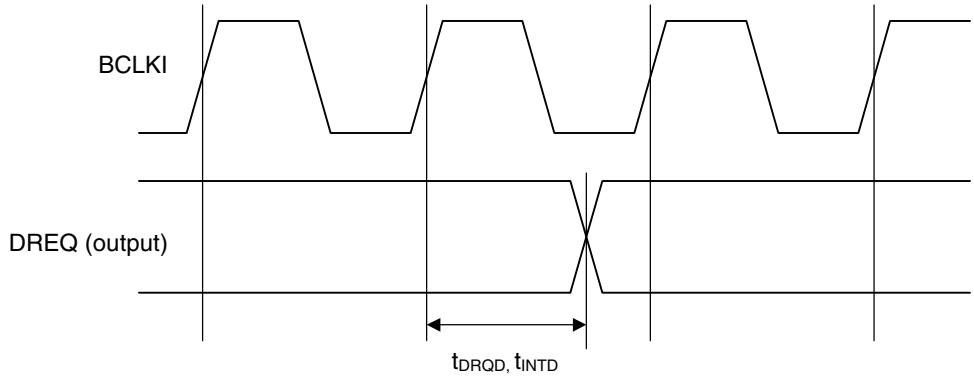
Input signal setup/hold times



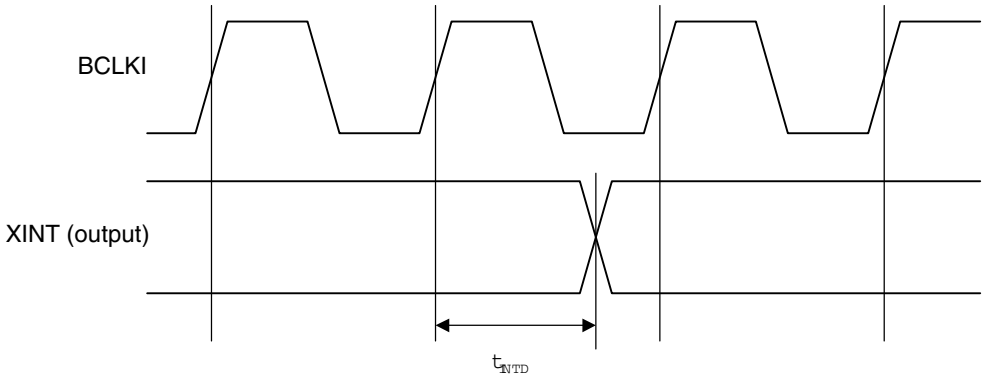
Read/write enable (XRD,XWE)/input data (D) setup time



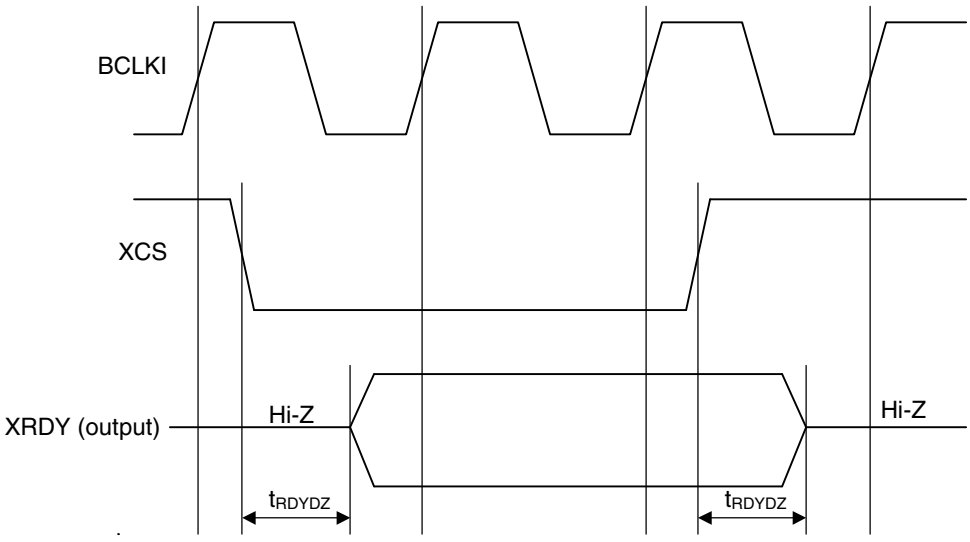
DREQ output delay times



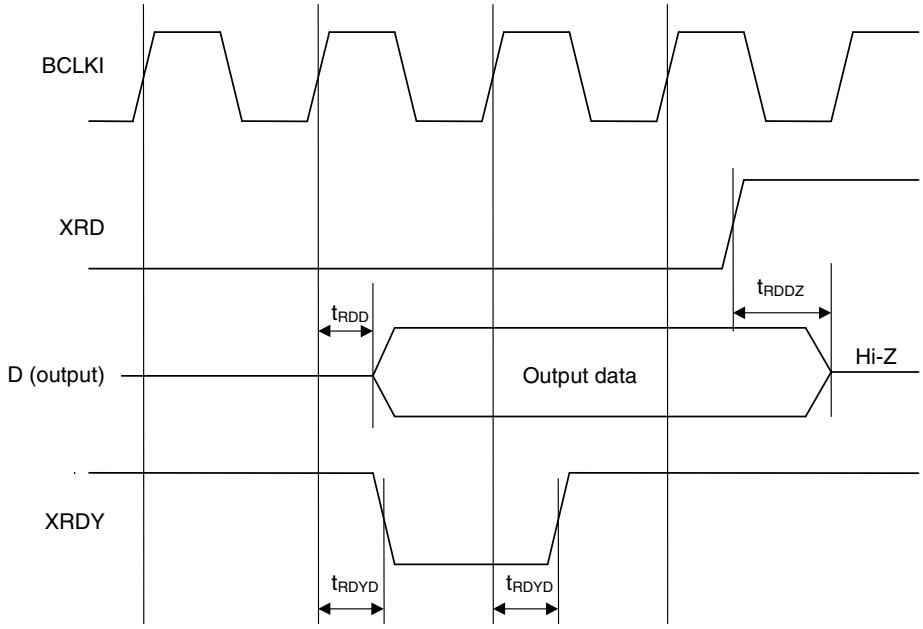
XINT output delay times



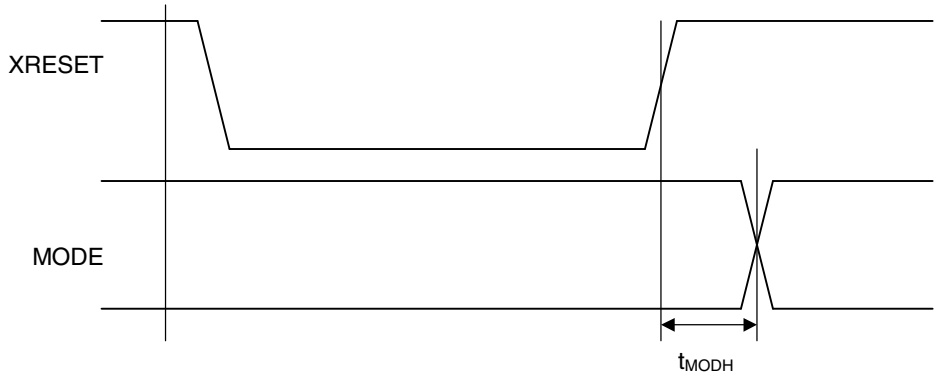
XRDY delay (for XCS)



XRDY, D output delay

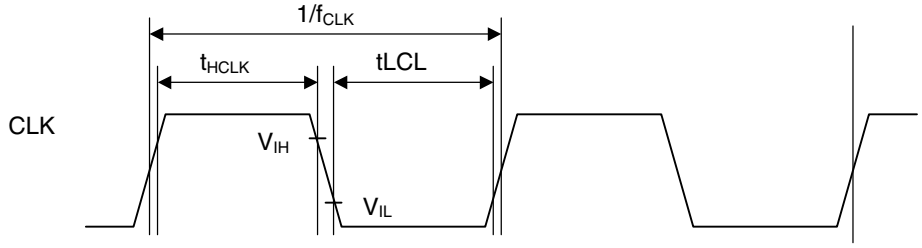


MODE hold time

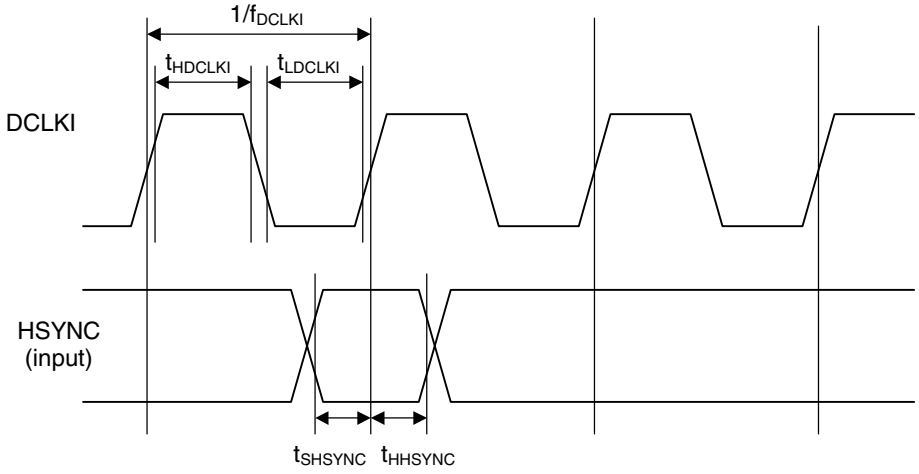


12.5.2 Video Interface

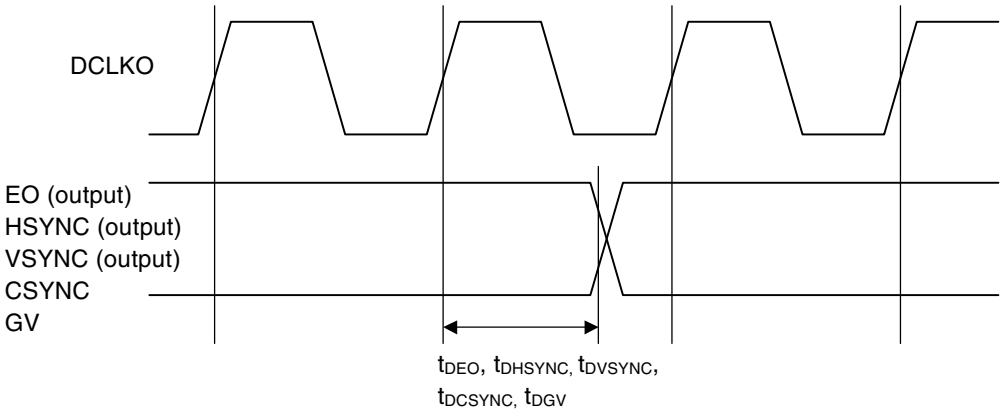
Clock



HSYNC signal setup/hold

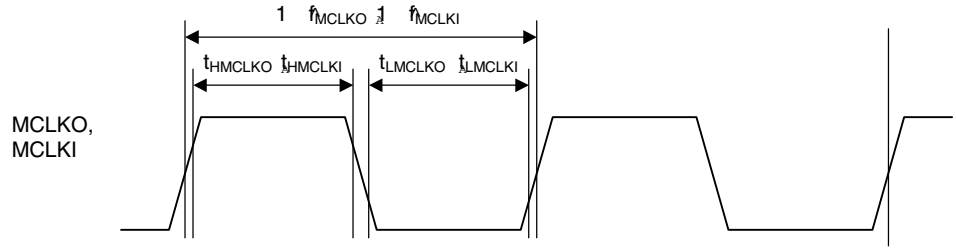


Output signal delay

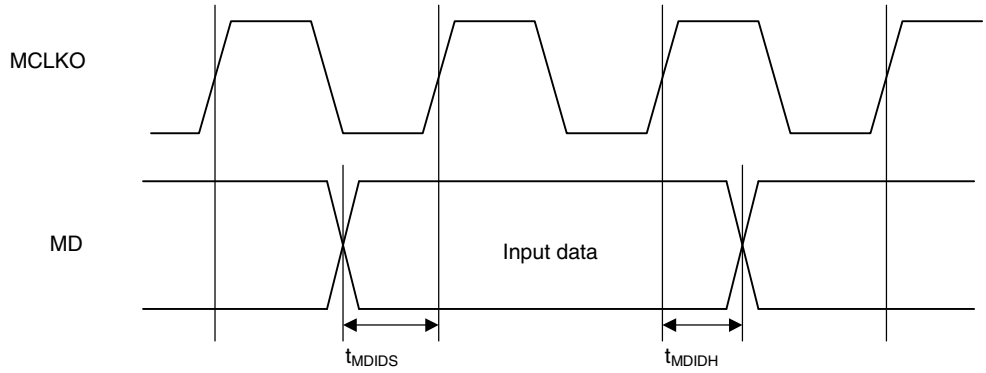


12.5.3 Graphic memory Interface

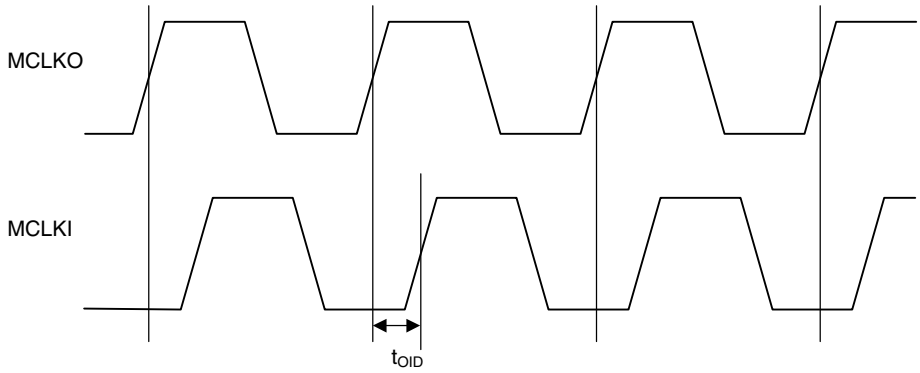
Clock



HSYNC signal setup/hold



MCLKI signal delay



Output signal delay