

# MB86293 <CORAL\_LQ>

## Graphics Controller Specifications

Revision 1.1

14th Jan, 2003



Copyright © FUJITSU LIMITED 2001

ALL RIGHTS RESERVED

- The specifications in this manual are subject to change without notice. Contact our Sales Department before purchasing the product described in this manual.
- Information and circuit diagrams in this manual are only examples of device applications, they are not intended to be used in actual equipment. Also, Fujitsu accepts no responsibility for infringement of patents or other rights owned by third parties caused by use of the information and circuit diagrams.
- The contents of this manual must not be reprinted or duplicated without permission of Fujitsu.
- Fujitsu's semiconductor devices are intended for standard uses (such as office equipment (computers and OA equipment), industrial/communications/measuring equipment, and personal/home equipment). Customers using semiconductor devices for special applications (including aerospace, nuclear, military and medical applications) in which a failure or malfunction might endanger life or limb and which require extremely high reliability must contact our Sales Department first. If damage is caused by such use of our semiconductor devices without first consulting our Sales Department, Fujitsu will not assume any responsibility for the loss.
- Semiconductor devices fail with a known probability. Customers must use safety design (such as redundant design, fireproof design, over-current prevention design, and malfunction prevention design) so that failures will not cause accidents, injury or death).
- If the products described in this manual fall within the goods or technologies regulated by the Foreign Exchange and Foreign Trade Law, permission must be obtained before exporting the goods or technologies.

## Update history

Date	Version	Page count	Change
2001.2.22	0.1	31	First edition
2001.7.23	0.2	228	See separate paper (Page count difference between Orchid and Coral)
2001.8.23	0.3	238	See separate paper (Page count difference between Rev0.2 and Rev0.3)
2001.11.12	0.4	262	See separate paper (Page count difference between Rev0.3 and Rev0.4)
2001.12.8	0.5	266	See separate paper (Page count difference between Rev0.4 and Rev0.5)
2002.1.9	0.6	266	See separate paper (Page count difference between Rev0.5 and Rev0.6)
2002.4.15	0.7	274	See separate paper (Page count difference between Rev0.6 and Rev0.7)
2002.5.21	0.8	274	See separate paper (Page count difference between Rev0.7 and Rev0.8)
2003.1.14	1.1	284	See separate paper (Page count difference between Rev0.8 and Rev1.1)

# CONTENTS

<b>1</b>	<b>GENERAL</b> .....	<b>10</b>
1.1	PREFACE .....	10
1.2	FEATURES .....	11
1.3	BLOCK DIAGRAM .....	12
1.4	FUNCTIONAL OVERVIEW .....	13
1.4.1	<i>Host CPU interface</i> .....	13
1.4.2	<i>External memory interface</i> .....	15
1.4.3	<i>Display controller</i> .....	16
1.4.4	<i>Geometry processing</i> .....	18
1.4.5	<i>2D Drawing</i> .....	19
1.4.6	<i>3D Drawing</i> .....	21
1.4.7	<i>Special effects</i> .....	22
1.4.8	<i>Others</i> .....	25
<b>2</b>	<b>PINS</b> .....	<b>26</b>
2.1	SIGNALS .....	26
2.1.1	<i>Signal lines</i> .....	26
2.2	PIN ASSIGNMENT .....	27
2.2.1	<i>Pin assignment diagram</i> .....	27
2.2.2	<i>Pin assignment table</i> .....	28
2.3	PIN FUNCTION .....	30
2.3.1	<i>Host CPU interface</i> .....	30
2.3.2	<i>Video output interface</i> .....	32
2.3.3	<i>Graphics memory interface</i> .....	33
2.3.4	<i>Clock input</i> .....	34
2.3.5	<i>Test pins</i> .....	35
2.3.6	<i>Reset sequence</i> .....	35
<b>3</b>	<b>PROCEDURE OF THE HARDWARE INITIALIZATION</b> .....	<b>36</b>
3.1	HARDWARE RESET .....	36
3.2	RE-RESET .....	36
3.3	SOFTWARE RESET .....	36
<b>4</b>	<b>HOST INTERFACE</b> .....	<b>37</b>
4.1	OPERATION MODE .....	37
4.1.1	<i>Host CPU mode</i> .....	37
4.1.2	<i>Ready signal mode</i> .....	37
4.1.3	<i>BS signal mode</i> .....	38

4.1.4	<i>Endian</i> .....	38
4.2	ACCESS MODE.....	39
4.2.1	<i>SRAM interface</i> .....	39
4.2.2	<i>FIFO interface (fixed transfer destination address)</i> .....	39
4.3	DMA TRANSFER.....	40
4.3.1	<i>Data transfer unit</i> .....	40
4.3.2	<i>Address mode</i> .....	40
4.3.3	<i>Bus mode</i> .....	41
4.3.4	<i>DMA transfer request</i> .....	41
4.3.5	<i>Ending DMA transfer</i> .....	41
4.4	TRANSFER OF LOCAL DISPLAY LIST.....	42
4.5	INTERRUPT.....	43
4.6	SH3 MODE.....	43
4.7	WAIT.....	43
4.8	MEMORY MAP.....	44
<b>5</b>	<b>GRAPHICS MEMORY</b> .....	<b>46</b>
5.1	CONFIGURATION.....	46
5.1.1	<i>Data type</i> .....	46
5.1.2	<i>Memory Mapping</i> .....	47
5.1.3	<i>Data Format</i> .....	47
5.2	FRAME MANAGEMENT.....	49
5.2.1	<i>Single Buffer</i> .....	49
5.2.2	<i>Double Buffer</i> .....	49
5.3	MEMORY ACCESS.....	49
5.3.1	<i>Memory Access by host CPU</i> .....	49
5.3.2	<i>Priority of memory accessing</i> .....	49
5.4	CONNECTION WITH MEMORY.....	50
5.4.1	<i>Connection with memory</i> .....	50
<b>6</b>	<b>DISPLAY CONTROLLER</b> .....	<b>52</b>
6.1	OVERVIEW.....	52
6.2	DISPLAY FUNCTION.....	52
6.2.1	<i>Layer configuration</i> .....	52
6.2.2	<i>Overlay</i> .....	54
6.2.3	<i>Display parameters</i> .....	56
6.2.4	<i>Display position control</i> .....	57
6.3	DISPLAY COLOR.....	59
6.4	CURSOR.....	60
6.4.1	<i>Cursor display function</i> .....	60
6.4.2	<i>Cursor control</i> .....	60

6.5	DISPLAY SCAN CONTROL .....	61
6.5.1	<i>Applicable display</i> .....	61
6.5.2	<i>Interlace display</i> .....	62
6.6	THE EXTERNAL SYNCHRONOUS SIGNAL.....	63
6.7	VIDEO INTERFACE, NTSC/PAL OUTPUT.....	66
<b>7</b>	<b>GEOMETRY ENGINE .....</b>	<b>67</b>
7.1	GEOMETRY PIPELINE.....	67
7.1.1	<i>Processing flow</i> .....	67
7.1.2	<i>Model-view-projection (MVP) transformation (OC@CC coordinate transformation)</i> 68	
7.1.3	<i>3D-2D transformation (CC@NDC coordinate transformation)</i> .....	68
7.1.4	<i>View port transformation (NDC@DC coordinate transformation)</i> .....	69
7.1.5	<i>View volume clipping</i> .....	69
7.1.6	<i>Back face curling</i> .....	71
7.2	DATA FORMAT .....	72
7.2.1	<i>Data format</i> .....	72
7.3	SETUP ENGINE.....	73
7.3.1	<i>Setup processing</i> .....	73
7.4	LOG OUTPUT OF DEVICE COORDINATES .....	73
7.4.1	<i>Log output mode</i> .....	73
7.4.2	<i>Log output destination address</i> .....	73
<b>8</b>	<b>DRAWING PROCESSING .....</b>	<b>74</b>
8.1	COORDINATE SYSTEM .....	74
8.1.1	<i>Drawing coordinates</i> .....	74
8.1.2	<i>Texture coordinates</i> .....	75
8.1.3	<i>Frame buffer</i> .....	76
8.2	FIGURE DRAWING .....	77
8.2.1	<i>Drawing primitives</i> .....	77
8.2.2	<i>Polygon drawing function</i> .....	77
8.2.3	<i>Drawing parameters</i> .....	78
8.2.4	<i>Anti-aliasing function</i> .....	79
8.3	BIT MAP PROCESSING.....	80
8.3.1	<i>BLT</i> .....	80
8.3.2	<i>Pattern data format</i> .....	80
8.4	TEXTURE MAPPING .....	81
8.4.1	<i>Texture size</i> .....	81
8.4.2	<i>Texture color</i> .....	81
8.4.3	<i>Texture lapping</i> .....	82
8.4.4	<i>Filtering</i> .....	83
8.4.5	<i>Perspective correction</i> .....	83

8.4.6	<i>Texture blending</i> .....	84
8.4.7	<i>Bi-linear high-speed mode</i> .....	84
8.5	RENDERING .....	86
8.5.1	<i>Tiling</i> .....	86
8.5.2	<i>Alpha blending</i> .....	86
8.5.3	<i>Logic operation</i> .....	87
8.5.4	<i>Hidden plane management</i> .....	87
8.6	DRAWING ATTRIBUTES .....	88
8.6.1	<i>Line drawing attributes</i> .....	88
8.6.2	<i>Triangle drawing attributes</i> .....	88
8.6.3	<i>Texture attributes</i> .....	89
8.6.4	<i>BLT attributes</i> .....	90
8.6.5	<i>Character pattern drawing attributes</i> .....	90
8.7	BOLD LINE.....	91
8.7.1	<i>Starting and ending points</i> .....	91
8.7.2	<i>Broken line pattern</i> .....	92
8.7.3	<i>Edging</i> .....	93
8.7.4	<i>Interpolation of bold line joint</i> .....	94
<b>9</b>	<b>DISPLAY LIST</b> .....	<b>95</b>
9.1	OVERVIEW .....	95
9.1.1	<i>Header format</i> .....	96
9.1.2	<i>Parameter format</i> .....	96
9.2	GEOMETRY COMMANDS .....	97
9.2.1	<i>Geometry command list</i> .....	97
9.2.2	<i>Explanation of geometry commands</i> .....	100
9.3	RENDERING COMMAND .....	110
9.3.1	<i>Command list</i> .....	110
9.3.2	<i>Details of rendering commands</i> .....	115
<b>10</b>	<b>REGISTER</b> .....	<b>125</b>
10.1	REGISTER LIST.....	125
10.1.1	<i>Host interface register list</i> .....	125
10.1.2	<i>Graphics memory interface register list</i> .....	127
10.1.3	<i>Display controller register list</i> .....	128
10.1.4	<i>Drawing engine register list</i> .....	133
10.1.5	<i>Geometry engine register list</i> .....	139
10.2	EXPLANATION OF REGISTER .....	140
10.2.1	<i>Host interface registers</i> .....	141
10.2.2	<i>Graphics memory interface registers</i> .....	148
10.2.3	<i>Display control register</i> .....	151

10.2.4	Drawing control registers .....	199
10.2.5	Drawing mode registers.....	202
10.2.6	Triangle drawing registers.....	220
10.2.7	Line drawing registers.....	223
10.2.8	Pixel drawing registers.....	224
10.2.9	Rectangle drawing registers.....	224
10.2.10	Blt registers .....	225
10.2.11	High-speed 2D line drawing registers.....	226
10.2.12	High-speed 2D triangle drawing registers.....	227
10.2.13	Geometry control register.....	228
10.2.14	Geometry mode registers.....	230
10.2.15	Display list FIFO registers .....	237
<b>11</b>	<b>TIMING DIAGRAM.....</b>	<b>238</b>
11.1	HOST INTERFACE.....	238
11.1.1	CPU read/write timing diagram in SH3 mode (Normally Not Ready Mode).....	238
11.1.2	CPU read/write timing diagram in SH3 mode (Normally Ready Mode) .....	239
11.1.3	CPU read/write timing diagram in SH4 mode (Normally Not Ready Mode).....	240
11.1.4	CPU read/write timing diagram in SH4 mode (Normally Ready Mode) .....	241
11.1.5	CPU read/write timing diagram in V832 mode (Normally Not Ready Mode) .....	242
11.1.6	CPU read/write timing diagram in V832 mode (Normally Ready Mode).....	243
11.1.7	CPU read/write timing diagram in SPARClite (Normally Not Ready Mode).....	244
11.1.8	CPU read/write timing diagram in SPARClite (Normally Ready Mode).....	245
11.1.9	SH4 single-address DMA write (transfer of 1 long word).....	246
11.1.10	SH4 single-address DMA write (transfer of 8 long words).....	247
11.1.11	SH3/4 dual-address DMA (transfer of 1 long word).....	248
11.1.12	SH3/4 dual-address DMA (transfer of 8 long words).....	248
11.1.13	V832 DMA transfer .....	249
11.1.14	SH4 single-address DMA transfer end timing.....	250
11.1.15	SH3/4 dual-address DMA transfer end timing .....	250
11.1.16	V832 DMA transfer end timing.....	251
11.1.17	SH4 dual DMA write without ACK .....	252
11.1.18	Dual-address DMA (without ACK) end timing.....	253
11.2	GRAPHICS MEMORY INTERFACE .....	254
11.2.1	Timing of read access to same row address .....	254
11.2.2	Timing of read access to different row addresses.....	255
11.2.3	Timing of write access to same row address .....	256
11.2.4	Timing of write access to different row addresses.....	257
11.2.5	Timing of read/write access to same row address .....	258
11.2.6	Delay between ACTV commands.....	259



11.2.7	<i>Delay between Refresh command and next ACTV command</i> .....	259
11.3	DISPLAY TIMING.....	260
11.3.1	<i>Non-interlace mode</i> .....	260
11.3.2	<i>Interlace video mode</i> .....	261
11.3.3	<i>Composite synchronous signal</i> .....	262
11.4	CPU CAUTIONS .....	262
11.5	SH3 MODE.....	263
11.6	SH4 MODE.....	263
11.7	V832 MODE .....	264
11.8	SPARCLITE.....	264
11.9	SUPPORTED DMA TRANSFER MODES.....	264
<b>12</b>	<b>ELECTRICAL CHARACTERISTICS .....</b>	<b>265</b>
12.1	INTRODUCTION.....	265
12.2	MAXIMUM RATING .....	265
12.3	RECOMMENDED OPERATING CONDITIONS .....	266
12.3.1	<i>Recommended operating conditions</i> .....	266
12.3.2	<i>Note at power-on</i> .....	266
12.4	DC CHARACTERISTICS .....	268
12.4.1	<i>DC Characteristics</i> .....	268
12.4.2	<i>V-I characteristics diagram</i> .....	269
12.5	AC CHARACTERISTICS .....	270
12.5.1	<i>Host interface</i> .....	270
12.5.2	<i>Video interface</i> .....	271
12.5.3	<i>Graphics memory interface</i> .....	272
12.5.4	<i>PLL specifications</i> .....	276
12.6	AC CHARACTERISTICS MEASURING CONDITIONS .....	277
12.7	TIMING DIAGRAM.....	278
12.7.1	<i>Host interface</i> .....	278
12.7.2	<i>Video interface</i> .....	282
12.7.3	<i>Graphics memory interface</i> .....	283

# 1 GENERAL

## 1.1 Preface

Coral graphics controller has some functions and optional efficiency and is planned to be serial-manufactured according to purposes.

For ES version, the specifications in which common descriptions to CORAL series are written are planned to provide and for mass-production version, the specifications which are unique to each series are planned to be provided. Therefore, please bear in mind that the contents for ES and mass-production versions may be different from each other.

## 1.2 Features

- Geometry engine

Geometry engine supports the geometry processing that is compatible with ORCHID (MB86292). Using the display list created by ORCHID enables drawing. \*\*(But Floating point setup command is deleted.)

Heavy processing of geometric operations such as coordinates conversions or clipping performed by this device can reduce the CPU loads dramatically.

- 2D and 3D Drawing

Coral has a drawing function that is compatible with the CREMSON (MB86290A). It can draw data using the display list created for CREMSON. \*\*(But Internal texture RAM is deleted.)

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

- Display controller

Coral has a display controller that is compatible with ORCHID.

In addition to the traditional XGA (1024 × 768 pixels) display, 4-layer overlay, left/right split display, wrap-around scrolling, double buffers, and translucent display, function of 6layer overlay, 4-siding for palette are expanded.

- Host CPU interface

Can be connected to SH3 and SH4 manufactured by Hitachi, to V832 microprocessor by NEC and to SPARClike (MB86833) by Fujitsu without external circuits.

- External memory interface

SDRAM and FCRAM can be connected.

- Optional function

Final device can be selected either geometry high-/low-speed version.

- Others

CMOS technology with 0.18- $\mu$ m

HQFP256 Package (lead pitch 0.4 mm)

Supply voltage: 1.8 V (internal operation) /3.3 V (I/O)

Current consumption ( TYPICAL )

1.8 V power supply : 500mA

3.3V power supply : 100mA

### 1.3 Block Diagram

CORAL general block diagram is shown below:

Please note that the capture controller is deleted from this figure in Coral-LQ.

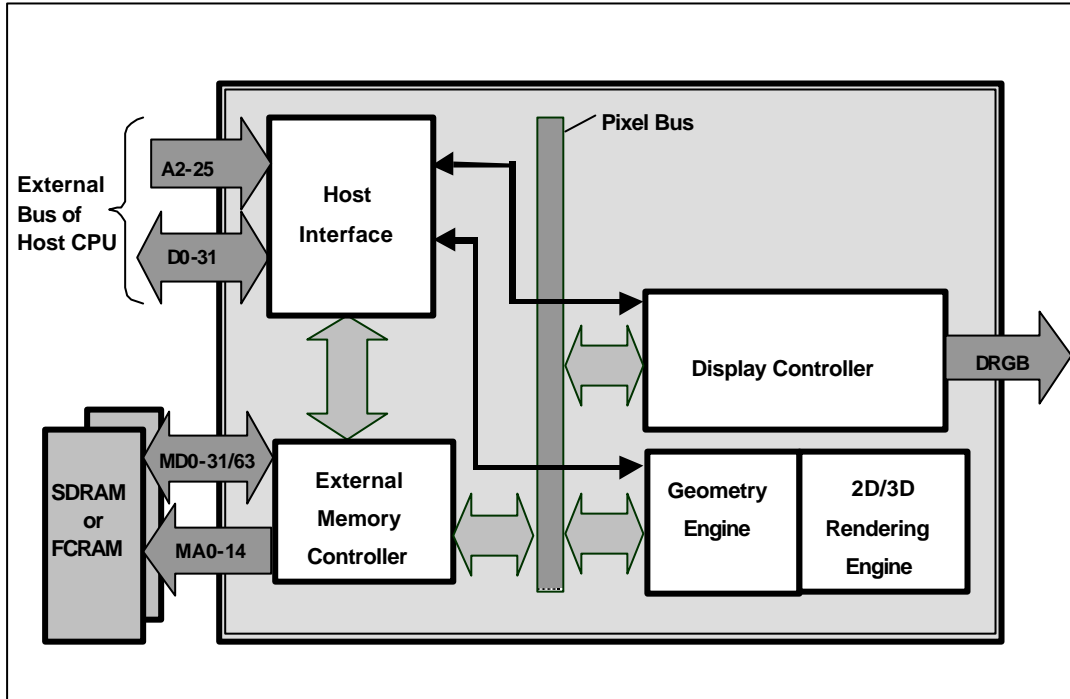


Fig.1.1 CORAL Block Diagram

## 1.4 Functional Overview

### 1.4.1 Host CPU interface

#### Supported CPU

Coral can be connected to SH3 and SH4 manufactured by HITACHI, V832 by NEC, SPARClite (MB86833) by Fujitsu.

#### External Bus Clock

Can be connected at max. 100 MHz (when using SH4 interface)

#### Ready Mode

Supports normal ready/not ready.

#### Endian

Supports little endian.

#### Access Mode

SRAM interface

FIFO interface (transfer destination address fixed)

#### DMA transfer

Supports 1-double word (32 bits) /8-double word (32 bytes) (only SH4) for transfer unit.

ACK used/unused mode can be selected as protocol (only for DAM in dual address mode)

Supports dual address/mode single address mode (only SH4).

Supports cycle steal/burst.

Supports local display list transfer.

#### Interrupt

Vertical (frame) synchronous detection

Field synchronous detection

External synchronous error detection

Drawing command error

Drawing command execution end

### Switching internal operating frequency

Switch the operating frequency immediately after a reset (before rewriting MMR mode register of external memory interface).

Any operating frequency can be selected from the five combinations shown in **Table 2-6**.

**Table 1-1 Frequency Setting Combinations**

Clock for geometry engine	Clock for other than geometry engine
166 MHz	133 MHz
166 MHz	100 MHz
133 MHz	133 MHz
133 MHz	100 MHz
100 MHz	100 MHz

The following relationship is disabled: Clock for geometry engine < Clock for other than geometry engine

## 1.4.2 External memory interface

SDRAM or FCRAM can be connected.

64 bits or 32 bits can be selected for data bus.

Max. 133 MHz is available for operating frequency.

Connectable memory configuration is as shown below.

**External Memory Configuration**

Type	Data bus width	Use count	Total capacity
FCRAM 16 Mbits (x16 Bits)	32 Bits	2	4 Mbytes
FCRAM 16 Mbits (x16 Bits)	64 Bits	4	8 Mbytes
SDRAM 64 Mbits (x32 Bits)	32 Bits	1	8 Mbytes
SDRAM 64 Mbits (x32 Bits)	64 Bits	2	16 Mbytes
SDRAM 64 Mbits (x16 Bits)	32 Bits	2	16 Mbytes
SDRAM 64 Mbits (x16 Bits)	64 Bits	4	32 Mbytes
SDRAM 128 Mbits (x32 Bits)	32 Bits	1	16 Mbytes
SDRAM 128 Mbits (x32 Bits)	64 Bits	2	32 Mbytes
SDRAM 128 Mbits (x16 Bits)	32 Bits	2	32 Mbytes
SDRAM 128 Mbits (x16 Bits)	64 Bits	4	64 Mbytes
SDRAM 256 Mbits (x16 Bits)	32 Bits	2	64 Mbytes

### 1.4.3 Display controller

#### Video data output

Each 6-bit digital video output is provided.

#### Screen resolution

LCD panels with wide range of resolutions are supported by using a programmable timing generator as follows:

Screen Resolutions
Resolutions
1024 × 768
1024 × 600
800 × 600
854 × 480
640 × 480
480 × 234
400 × 234
320 × 234

#### Hardware cursor

Coral supports two 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 same look-up table as indirect color mode.

#### Double buffer method

Double buffer method in which drawing window and display window is switched in units of 1 frame enables the smooth animation.

Flipping (switching of display window area) is performed in synchronization with the vertical blanking period using program.

#### Scroll method

Independent setting of drawing and display windows and their starting position enables the smooth scrolling.

#### Display colors

- Supports indirect color mode which uses the look-up table (color palette) in 8 bits/pixels.
- Entry for look-up table (color palette) corresponds to color code for 8 bits, in other words, 256. Color data is each 6 bits of RGB. Consequently, 256 colors can be displayed out of 260,000 colors.
- Supports direct color mode which specifies RGB with 16 bits/pixels.



## Overlay

### Compatibility mode

Up to four extra layers (C, W, M and B) can be displayed overlaid.

The overlay position for the hardware cursors is above/below the top layer (C).

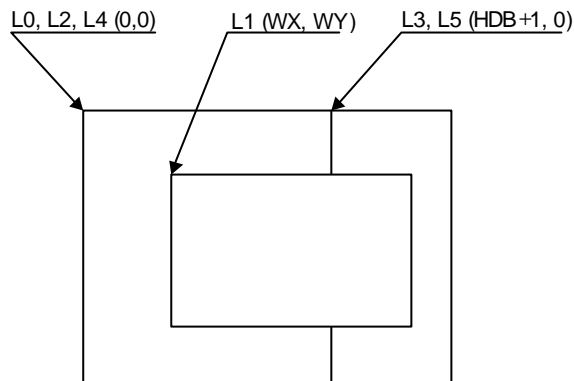
The transparent mode or the blend mode can be selected for overlay.

The M- and B-layers can be split into separate windows.

Window display can be performed for the W-layer.

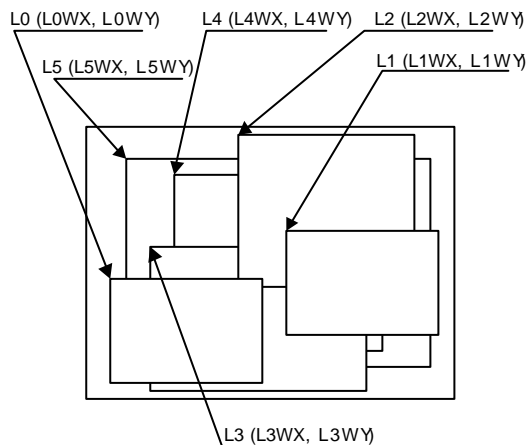
Two palettes are provided: Glayer and M-/B-layer.

The W-layer is used as the video input layer.



### Window mode

- Up to six screens (L0 to 5) can be displayed overlaid.
- The overlay sequence of the L0- to L5-layers can be changed arbitrarily.
- The overlay position for the hardware cursors is above/below the L0-layer.
- The transparent mode or the blend mode can be selected for overlay.
- The L5-layer can be used as the blend coefficient plane (8 bits/pixel).
- Window display can be performed for all layers.
- Four palettes corresponded to L0 to 3 are provided.
- The L1-layer is used as the video input layer.
- Background color display is supported in window display for all layers.



## 1.4.4 Geometry processing

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

### Primitives

Point, line, line strip, independent triangle, triangle strip, triangle fan, and arbitrary polygon are supported.

### MVP Transformation

MVP Transformation

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

### Clipping

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

### Culling

Triangles on the back are not drawn.

### 3D-2D Transformation

This function 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 prepares for rendering (setup).

### Log output of device coordinates

The view port conversion results are output to the local memory.

## 1.4.5 2D Drawing

### 2D Primitives

Coral can perform 2D drawing for graphics memory (drawing plane) in direct color mode or indirect color mode.

Bold lines with width and broken lines can be drawn. With anti-aliasing smooth diagonal lines also can be drawn.

A triangle can be tiled in a single color or 2D pattern (tiling), or mapped with a texture pattern by specifying coordinates of the 2D pattern at each vertex (texture mapping). At texture mapping, drawing/non-drawing can be set in pixel units. Moreover, transparent processing can be performed using alpha blending. When drawing in single color or tiling without Gouraud shading or texture mapping, high-speed 2DLine and high-speed 2DTriangle can be used. Only vertex coordinates are set for these primitives. High-speed 2DTriangle is also used to draw polygons.

#### 2D Primitives

Primitive type	Description
Point	Plots point
Line	Draws line
Bold line strip (provisional name)	Draws continuous bold line This primitive is used when interpolating the bold line joint.
Triangle	Draws triangle
High-speed 2DLine	Draws lines Compared to line, this reduces the host CPU processing load.
Arbitrary polygon	Draws arbitrary closed polygon containing concave shapes consisting of vertices

### Arbitrary polygon drawing

Using this function, arbitrary closed polygon containing concave shapes consisting of vertices can be drawn. (There is no restriction on the count of vertices, however, the polygon with its sides crossed are not supported.) In this case, as a work area for drawing, polygon drawing flag buffer is used on the graphics memory. In drawing polygon, draw triangle for polygon drawing flag buffer using high-speed 2DTriangle. Decide any vertex as a starting point to draw triangle along the periphery. It enables you to draw final polygon form in single color or with tiling/texture mapping in a drawing frame.

### BLT/Rectangle drawing

This function draws a rectangle using logic operations. It is used to draw pattern and copy the image pattern within the drawing frame. It is also used for clearing drawing frame and Z buffer.

#### BLT Attributes

Attribute	Description
Raster operation	Selects two source logical operation mode
Transparent processing	Performs BLT without drawing pixel consistent with the transparent color.
Alpha blending	The alpha map and source in the memory is subjected to alpha blending and then copied to the destination.

### Pattern (Text) drawing

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

#### Pattern (Text) Drawing Attributes

Attribute	Description
Enlarge	Vertically $2 \times 2$ Horizontally $\times 2$ Vertically and Horizontally $\times 2$
Shrink	Vertically $1/2 \times 1/2$ Horizontally $1/2$ Vertically and Horizontally $1/2$

### Drawing clipping

This function sets a rectangle frame in drawing frame to prohibit the drawing of the outside the frame.

## 1.4.6 3D Drawing

### 3D Primitives

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

#### 3D Primitives

Primitive	Description
Point	Plots 3D point
Line	Draws 3D line
Triangle	Draws 3D triangle
Arbitrary polygon	Draws arbitrary closed polygon containing concave shapes consisting of vertexes

### 3D Drawing attributes

Texture mapping with bi-linear filtering/automatic perspective correction and Gouraud shading provides high-quality realistic 3D drawing. A built-in texture mapping unit performs fast pixel calculations. This unit also delivers color blending between the shading color and texture color.

### Hidden plane management

Coral supports the Z buffer for hidden plane management.

## 1.4.7 Special effects

### Anti-aliasing

Anti-aliasing manipulates line borders of polygons in sub-pixel units and blend the pre-drawing pixel color with color to make the jaggies be seen smooth. It is used as a functional option for 2D drawing (in direct color mode only).

### Bold line and broken line drawing

This function draws lines of a specific width and a broken line.

**Line Drawing Attributes**

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

- Supports the verticality of starting and ending points.
- Supports the verticality of broken line pattern.
- Interpolation of bold line joint supports the following modes:
  - (1) Broken line pattern reference address fix mode
    - The same broken line pattern is kept referencing for the period of some pixels starting from the joint and the starting point for the next line.
  - (2) No interpolation
- Supports the equalization of the width of bold lines.
- Supports the bold line edging.
- Not support the Anti-aliasing of dashed line patterns.
- For a part overlaid due to connection of bold lines, natural overlay can be represented by providing depth information. (Z value).

### Shading

Supports the shading primitive.

Drawing is performed to the body primitive coordinates (X, Y) with an offset as a shade. At this drawing, the Z buffer is used in order to differentiate between the body and shade.

## Alpha blending

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

There are two ways of specifying alpha blending for drawing:

- (1) Set a transparent coefficient to the register; the transparent coefficient is applied for transparency processing of one plane.
- (2) Set a transparent coefficient for each vertex of the plane; as with Gouraud shading, the transparent coefficient is linear-interpolated to perform transparent processing in pixel units.

In addition to the above, the following settings can be performed at texture mapping. When the most significant bit of each texture cell is 1, drawing or transparency can be set. When the most significant bit of each texture cell is 0, non-drawing can be set.

### Alpha Blending

Type	Description
Drawing	Transparent ratio set in particular register  While one primitive (polygon, pattern, etc.), being drawn, registered transparent ratio applied  A transparent coefficient set for each vertex. A linear-interpolated transparent coefficient applied.  This is possible only in direct color mode.
Overlay display	Blends top layer pixel color with lower layer pixel color  Transparent coefficient set in particular register  Registered transparent coefficient applied during one frame scan

## Gouraud Shading

Gouraud shading can be used in the direct color mode to provide 3D object real shading and color gradation.

## Gray Scale Gouraud Shading

Gray scale gouraud shading can be used in the in-direct color(8bit/pixel) mode to draw a blend coefficient layer.

## Texture mapping

Coral supports texture mapping to map an image pattern onto the surface of plane. The texture pattern can be laid out in the graphics memory. In this case, max. 4096 × 4096 pixels can be used.

For drawing 8-bit color, only point sampling can be specified for texture interpolation; only de-curl can be specified for the blend mode.

### Texture Mapping

Function	Description
Filtering	Point sample Bi-linear filter
Coordinates correction	Linear Perspective
Blend	De-curl Modulate Stencil
Alpha blend	Normal Stencil Stencil alpha
Wrap	Repeat Cramp Border



## 1.4.8 Others

### Drawing color

8-bit indirect color and 16-bit direct color are supported as a drawing input data.

### Top-left rule non-applicable mode

In addition to the top-left rule applicable mode in which the triangle borders are compatible with CREMSON, the top-left rule non-applicable mode can be used. (In case of non-top-left polygon drawing, an object has to be in a geometry clipping area.)

Caution: Use perspective correct mode when use texture at the top-left rule non-applicable mode.

Top-left rule non-applicable primitives cannot use Geometry clip function.

Non-top-left-part's pixel quality is less than body. (using approximate calculation)

# 2 PINS

## 2.1 Signals

### 2.1.1 Signal lines

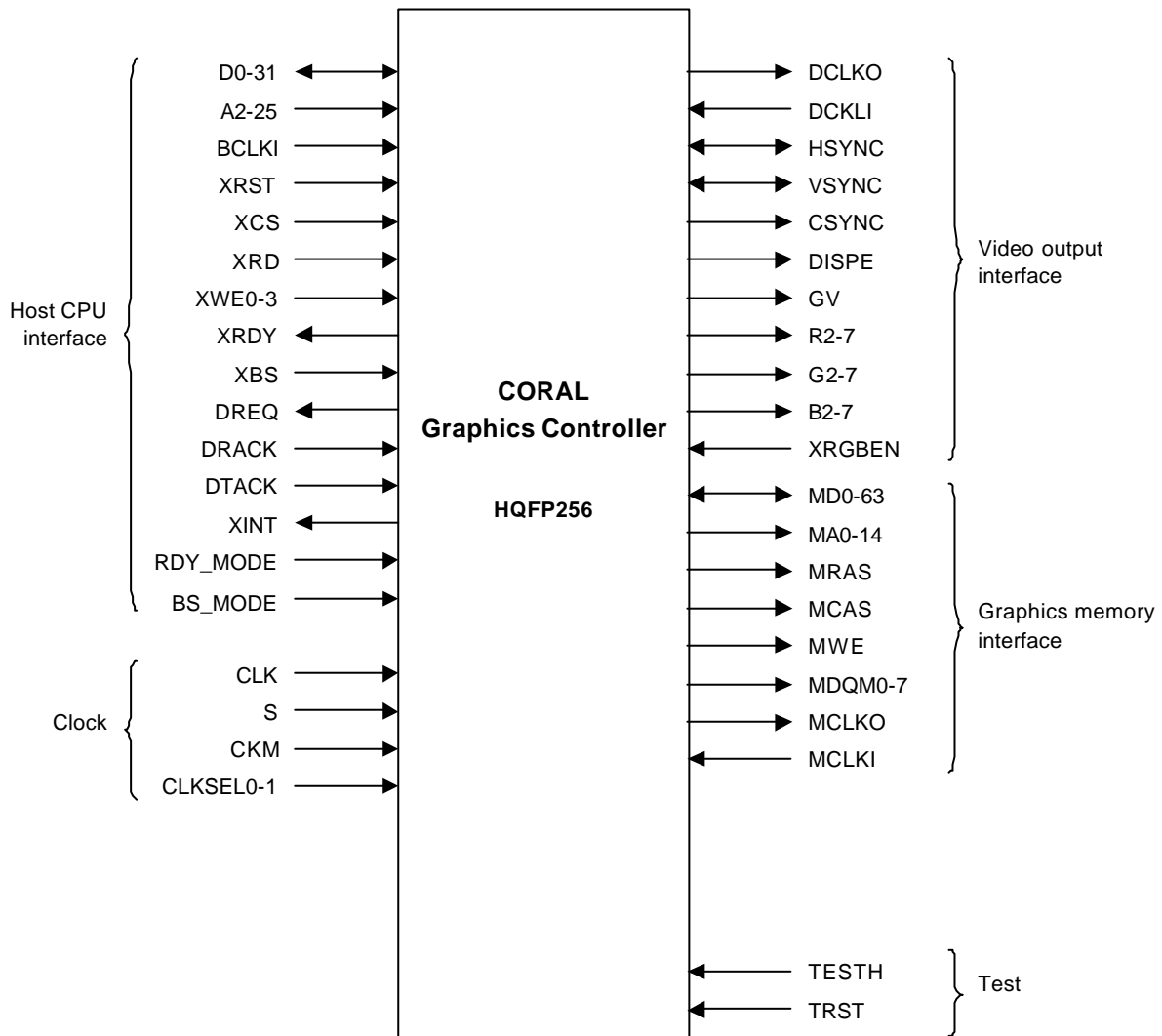


Fig. 2.1 CORAL Signal Lines



## 2.2.2 Pin assignment table

No.	Name	No.	Name	No.	Name	No.	Name
1	DRACK	65	MD5	129	VSS	193	DR3
2	BS_MODE	66	VDDL	130	VSS	194	VDDL
3	RDY_MODE	67	MD6	131	VDDH	195	DR4
4	XINT	68	MD7	132	MCLKI	196	DR5
5	DREQ	69	MD8	133	VSS	197	DR6
6	XRDY	70	MD9	134	VSS	198	DR7
7	VDDH	71	MD10	135	VDDL	199	VDDH
8	VDDL	72	MD11	136	TESTH	200	DCLKO
9	BCLKI	73	MD12	137	XRGBEN	201	DE
10	XCS	74	MD13	138	MD32	202	GV
11	XRD	75	MD14	139	MD33	203	CSYNC
12	XBS	76	MD15	140	MD34	204	VSS
13	CKM	77	VDDH	141	MD35	205	VSS
14	MODE0	78	MD16	142	MD36	206	HSYNC
15	MODE1	79	MD17	143	MD37	207	VSYSN
16	MODE2	80	MD18	144	MD38	208	TESTH
17	XRST	81	VDDL	145	MD39	209	VDDL
18	VSS	82	MD19	146	VSS	210	TESTH
19	VSS	83	MD20	147	VSS	211	VSS
20	D0	84	MD21	148	MD40	212	VSS
21	D1	85	MD22	149	MD41	213	DCLKI
22	D2	86	MD23	150	MD42	214	TESTH
23	D3	87	MD24	151	MD43	215	TESTH
24	VDDL	88	MD25	152	VDDL	216	TRST
25	D4	89	MD26	153	MD44	217	S
26	D5	90	MD27	154	MD45	218	PLLSS
27	VDDH	91	MD28	155	MD46	219	PLLVD
28	D6	92	MD29	156	MD47	220	CLK
29	D7	93	MD30	157	TESTH	221	VSS
30	D8	94	MD31	158	TESTH	222	VSS
31	D9	95	VDDL	159	MD48	223	CLKSEL1
32	D10	96	VSS	160	MD49	224	VDDL
33	D11	97	VSS	161	MD50	225	CLKSEL0
34	D12	98	VDDH	162	MD51	226	A2
35	D13	99	DQM0	163	MD52	227	A3
36	D14	100	DQM1	164	MD53	228	A4
37	D15	101	DQM2	165	MD54	229	A5
38	D16	102	DQM3	166	MD55	230	A6
39	D17	103	MA0	167	MD56	231	A7
40	VDDL	104	MA1	168	VDDL	232	A8
41	D18	105	MA2	169	MD57	233	A9
42	VDDH	106	MA3	170	MD58/B0	234	A10
43	D19	107	MA4	171	MD59/B1	235	A11
44	D20	108	MA5	172	MD60/G0	236	A12
45	D21	109	MA6	173	MD61/G1	237	A13
46	D22	110	MA7	174	MD62/R0	238	A14
47	D23	111	VDDL	175	MD63/R1	239	VDDL
48	D24	112	MA8	176	VSS	240	A15
49	D25	113	MA9	177	VSS	241	A16
50	D26	114	MA10	178	VDDH	242	A17
51	D27	115	MA11	179	DB2	243	A18
52	D28	116	MA12	180	DB3	244	A19
53	D29	117	MA13	181	DB4	245	A20
54	D30	118	MA14	182	DB5	246	A21
55	D31	119	VDDH	183	DB6	247	A22
56	VDDL	120	MRAS	184	VDDL	248	A23
57	VDDH	121	MCAS	185	DB7	249	A24
58	VSS	122	MWE	186	DG2	250	A25
59	VSS	123	DQM4	187	DG3	251	XWE3
60	MD0	124	DQM5	188	DG4	252	XWE2
61	MD1	125	DQM6	189	DG5	253	XWE1
62	MD2	126	DQM7	190	DG6	254	VDDL
63	MD3	127	VDDL	191	DG7	255	XWE0
64	MD4	128	MCLKO	192	DR2	256	DTACK

Notes

$V_{SS}/PLL_{V_{SS}}$	:	Ground
$V_{DDH}$	:	3.3-V power supply
$V_{DDL}/PLL_{V_{DD}}$	:	1.8-V power supply
$PLL_{V_{DD}}$	:	PLL power supply
OPEN	:	Do not connect anything.
TESTH	:	Input a 3.3 V-power supply.

- It is recommended that  $PLL_{V_{DD}}$  should be isolated on the PCB.
- Insert a bypass capacitor with good high frequency characteristics between the power supply and ground.

Place the capacitor as near as possible to the pin.

## 2.3 Pin Function

### 2.3.1 Host CPU interface

Table 2-1 Host CPU Interface Pins

Pin name	I/O	Description
MODE0-2	Input	Host CPU mode select
RDY_MODE	Input	Normally ready, Not ready select
BS_MODE	Input	BS signal with/without select
XRST	Input	Hardware reset
D0-31	In/Out	Host CPU bus data
A2-A25	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 signal
XCS	Input	Chip select signal
XRD	Input	Read strobe signal
XWE0	Input	Write strobe for D0 to D7 signal
XWE1	Input	Write strobe for D8 to D15 signal
XWE2	Input	Write strobe for D16 to D23 signal
XWE3	Input	Write strobe for D24 to D31 signal
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 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, and SPARClite this signal is low-active; in the V832 mode, it is high-active)

- Coral can be connected to the Hitachi SH4 (SH7750), SH3 (SH7709) NEC V832 and Fujitsu SPARClite (MB86833) without external circuit. In the SRAM interface mode, Coral can be used with any other CPU as well. The host CPU is specified by the MODE0 to 2 pins.

MODE 2	MODE 1	MODE 0	CPU
L	L	L	SH3
L	L	H	SH4
L	H	L	V832
L	H	H	SPARClite
H	X	X	Reserved

When the bus cycle terminates, a ready signal level can be set. When using the RDY\_MODE signal at “High” level, set two cycles as the CPU software wait of the CPU. (When BS\_MODE = “High” level, set the CPU software wait to three cycles.)

RDY_MODE	Ready signal mode
L	When the bus cycle terminates, sets the XRDY signal to the ‘not ready’ level.
H	When the bus cycle terminates, sets the XRDY signal to the ‘ready’ level.

A CPU with no BS (Bus Start) pin can be used. Setting can be performed in all CPU modes. Connection can be made to a CPU with no BS signal by setting the BS\_MODE signal to “High” level.

When not using the BS signal, fix the BS pin of CORAL at “High” level.

When using the BS\_MODE signal as “High” level in the normally ready mode, set the CPU software wait to three cycles.

BS_MODE	BS signal mode
L	Connect to a CPU with the BS signal
H	Connect to a CPU without the BS signal

The data signal is 32 bits (fixed).

The address signal is 32 bits (per one double-word) × 24, and has a 64-Mbyte address field. (16-MByte address space is provided for V832 and SPARClite.)

The external bus operating frequency is up to 100 MHz.

In the SH4, V832, and SPARClite modes, when the XRDY signal is low, it is in the ready state. However, in the SH3 mode, when the XRDY signal is low, it is in the wait state. This signal is a tri-state output that is synchronized with the rising edge of BCLKI.

DMA data transfer is supported using an external DMA controller.

An interrupt signal is generated to the host CPU.

The XRST input must be kept low for at least 300 μs after setting the S (PLL reset) signal to high.

In the V832 mode, Coral signals are connected to the V832 CPU as follows:

CORAL Pins	V832 Signals
A24	XMWR
DTACK	XTC
DRACK	DMAAK

## 2.3.2 Video output interface

Table 2-2 Video Output Interface Pins

Pin 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 enable period signal
GV	Output	Graphics/video switch
R2-7	Output	Digital picture (R) output
G2-7	Output	Digital picture (G) output
B2-7	Output	Digital picture (B) output
XRGBEN	Input	Signal to switch between RGB1 and 0 output/memory bus (MD 63 to 58)

6-bit display data is output as standard for R, G, and B. Depending on the condition, 8-bit display data can also be output for R, G, and B. Fixing XRGBEN at 0, R0, 1, G0, 1, and B0, 1 can be output to MD62, 63, MD60, 61, and MD58, 59 respectively. When 8-bit output is selected for R, G, and B, only the 32-bit mode can be used for the memory bus width mode.

Additional setting of external circuits can generate composite video signal.

Synchronous to external video signal display can be performed.

Either mode which is synchronous to DCLKI signal or one which is synchronous to dot clock, as for normal display can be selected.

Since HSYNC and VSYNC signals are set to input state after reset, these signals must be pulled up LSI externally.

The GV signal switches graphics and video at chroma key operation. When video is selected, the "Low" level is output.



## 2.3.3 Graphics memory interface

Table 2-3 Graphics memory interface pins

Pin name	I/O	Description
MD0 to 57	I/O	Graphics memory bus data
MD58 to 63/RGB	I/O	Graphics memory bus data or RGB0 to 1 output
MA0 to 14	Output	Graphics memory bus data
MRAS	Output	Row address strobe
MCAS	Output	Column address strobe
MWE	Output	Write enable
MDQM0 to 7	Output	Data mask
MCLK0	Output	Graphics memory clock output
MCLK1	Input	Graphics memory clock input

Connect the interface to the external memory used as memory for image data. The interface can be connected to 64-/128-/256-Mbit SDRAM (16- or 32-bit length data bus) without using any external circuit.

64 bits or 32 bits can be selected for the memory bus data. When 32-bit memory bus data is used and 6-bit output is used for R, G, and B (XRGBEN pin = 1), set MD32 to 63 and MDQM4 to 7 to the open state. When 32-bit memory bus data is used and 8-bit output is used for R, G, and B (XRGBEN pin = 0), set MD32 to 39 to "High" level input and set MD40 to 57 and MDQM4 to 7 to the open state.

Connect MCLKI to MCLK0.

When XRGBEN is fixed at "1", MD58 to 63 can be used as graphics memory bus data.

When XRGBEN is fixed at "0", MD58 to 63 can be used as digital RGB0 to 1 outputs.

## 2.3.4 Clock input

Table 2-4 Clock Input Pins

Pin name	I/O	Description
CLK	Input	Clock input signal
S	Input	PLL reset signal
CKM	Input	Clock mode signal
CLKSEL [1:0]	Input	Clock rate select signal

Inputs source clock for 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 166 MHz/133 MHz and the display base clock of 400 MHz.

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	CLKSEL0	Input clock frequency	Multiplication rate	Display reference clock
L	L	Inputs 13.5-MHz clock frequency	×29	391.5 MHz
L	H	Inputs 14.32-MHz clock frequency	×28	400.96 MHz
H	L	Inputs 17.73-MHz clock frequency	×22	390.06 MHz
H	H	Reserved		

### 2.3.5 Test pins

Table 2-5 Test Pins

Pin name	I/O	Description
TESTH	Input	Input 3.3-V power.
TRST	Input	This is the test reset signal. Before performing reset via S/XRST, perform reset via this signal (TRST).

### 2.3.6 Reset sequence

See **Section 10.3.2**.

## 3 PROCEDURE OF THE HARDWARE INITIALIZATION

### 3.1 Hardware reset

1. Do the hardware reset. (see section 11.3.2)
2. After the hardware reset, set the CCF(Change of Frequency) register (section 9.2.1).  
In being unstable cycle after the hardware reset, keep 32 bus cycles open.
3. Set the graphics memory interface register, MMR (Memory I/F Mode Register).  
After setting the CCF register, take 200 us to set the MMR register.  
In being unstable memory access cycle, keep 32 bus cycles open.
4. Other registers, except for the CCF register and the MMR register, should be set after setting the CCF register.  
In case of not using memory access, the MMR register could be set in any order after the CCF register is set.

### 3.2 Re-reset

1. Reset XRST signal.
2. See section 3.1 for registers setting after the procedure of re-reset.

### 3.3 Software reset

1. Set the value of the SRST register (see section 9.2.1) for re-reset.
2. It is not necessary to reset the CCF register and the MMR register again.

# 4 HOST INTERFACE

## 4.1 Operation Mode

### 4.1.1 Host CPU mode

Select the host CPU by setting the MODE0 to MODE2 signals as follows:

Table 4-1 CPU Type Setting

MODE 2	MODE 1	MODE 0	CPU
L	L	L	SH3
L	L	H	SH4
L	H	L	V832
L	H	H	SPARClike
H	X	X	Reserved

### 4.1.2 Ready signal mode

The MODE2 pin can be used to set the ready signal level when the bus cycle of the host CPU terminates. For the normally not ready mode, set the software wait to 0 or 1 cycles. When using this device in the normally ready mode, set the software wait to 2 cycles. When using this device in the normally not ready mode, set the software wait to one cycle. (When **BS\_MODE = H** three cycles are needed for the software wait.)

The 'normally not ready mode' is the mode in which the CORAL XRDY signal is always in the wait state and Ready is returned only when read/write is ready.

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

Table 4-2 Ready Signal Mode

RDY_MODE	Ready signal operation
L	Recognizes XRDY signal as 'not ready level' and terminates bus cycle (normally not ready mode)
H	Recognizes XRDY signal as 'ready level' and terminates bus cycle (normally ready mode)

### 4.1.3 BS signal mode

Connection to a CPU without the BS signal can be made via the **BS\_MODE** signal. This setting can be performed for all CPU modes. To connect to a CPU without the BS signal, set the **BS\_MODE** signal to “High” level.

When not using the BS signal, fix the BS pin of CORAL at “High” level.

When using the **BS\_MODE** signal as “High” level, with the normally ready mode established, set the CPU software wait to three cycles.

**Table 4-3 BS Signal Mode**

<b>BS_MODE</b>	<b>Operation of BS signal</b>
L	Connects to CPU with BS signal
H	Connects to CPU without BS signal

### 4.1.4 Endian

CORAL operates in little-endian mode. All the register address descriptions in the 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.

## 4.2 Access Mode

### 4.2.1 SRAM interface

Data can be transferred to/from CORAL using SRAM access protocol. CORAL internal registers and graphics memory are all mapped to the physical address area of the host processor.

CORAL uses hardware wait based on the XRDY signal, enabling the hardware wait setting of the host CPU. When using the normally not ready mode, set the software wait to "1". When using the normally ready mode, set the software wait to "2". (When using the **BS\_MODE** signal as 'High' level, with the normally ready mode established, set the CPU software wait to three cycles.) Switch the ready mode using the **RDY\_MODE** signal.

#### CPU Read

The host processor reads data from internal registers and memory of CORAL 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 CORAL in byte, word(16 bit) and double-word( 32 bit) units.

### 4.2.2 FIFO interface (fixed transfer destination address)

This interface transfers display lists stored in host memory. Display list information is transferred efficiently using a single address mode DMA transfer. Data can be transferred to FIFO in relation to FIFO buffer area mapped in memory area using SRAM interface or dual address mode.

## 4.3 DMA Transfer

### 4.3.1 Data transfer unit

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

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

### 4.3.2 Address mode

#### Dual address mode (mode using ACK)

DMA is performed at memory-to-memory transfer between host memory and registers mapped in memory space or graphics memory (destination). Both the host memory address and CORAL 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.

#### Dual address mode (mode not using ACK)

When not using the ACK signal with the dual address mode established, set bit3 at HostBase+0004h (DNA: Dual address No Ack mode) to 1.

When the ACK is not used, the DREQ signal is in the edge mode and the DREQ signal is negated per transfer and then reasserted it in the next cycle. If processing cannot be performed immediately inside CORAL, the DREQ signal remains negated.

The transfer count register (DTC) of CORAL is not used, so in order to end DMA transfer, write "1" to the DMA transfer stop register (DTS) from the CPU.

Note 1: In the dual DMA mode (mode without ACK), the destination address can be used only for the FIFO.

In DMA transfer to the graphics memory, etc., use the dual DMA mode.

Note 2: DMA read is not supported.

#### Single address mode (FIFO interface)

Data is transferred between host memory (source) and FIFO (destination). Only the address output from the host memory is used, and the data is transferred to the FIFO. This mode does not support data write to the host memory. When the FIFO is full, the DMA transfer is suspended.

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

DREQ, DTACK, and DRACK signal are used.

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



### 4.3.3 Bus mode

Coral supports the DMA transfer cycle steal mode and burst mode according to setting of 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 right to use the bus is obtained or released at every data transfer of 1 unit. The DMA transfer unit can be selected from between the 1 double-word (32 bits) and 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.

### 4.3.4 DMA transfer request

#### **Single-address mode**

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

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

Fix the CPU destination address to the FIFO address.

#### **Dual-address mode**

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

Note: In the Ack unused mode and the V832 mode requires no setting of the transfer count register.

### 4.3.5 Ending DMA transfer

- SH3/SH4

When the CORAL 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 S CORAL is high-asserted, the end of DMA transfer is recognized and DREQ is negated.

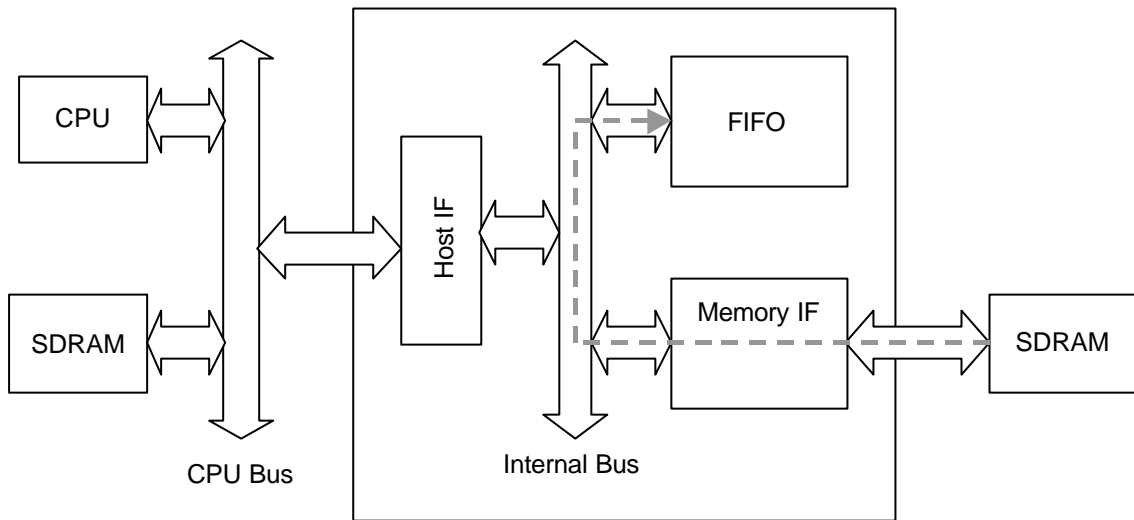
- The end of DMA transfer is detected in two ways: the DMA status register (DST) is polled, and an interrupt to end the drawing command (FD000000<sub>H</sub>) is added to the display list and the interrupt is detected.
- In the dual address mode (mode not using ACK), the DMA transfer count register (DTC) is not used, so the DMA ending cannot be determined. The DREQ signal can be negated to end DMA by writing 1 from the CPU to the DMA transfer stop register (DTS) of CORAL at DMA transfer end.

## 4.4 Transfer of Local Display List

This is the mode in which the CORAL 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 for CPU read/write.

How to transfer list: Store the display list in the local memory of CORAL, 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

## 4.5 Interrupt

Coral issues interrupt requests to the host CPU. Following shows the types of interrupt factor and they can be enabled/disabled by IMASK (Interrupt Mask Register).

- Vertical synchronization detect
- Field synchronization detect
- External synchronization error detect
- Drawing command error
- Drawing command execution end

## 4.6 SH3 Mode

In the SH3 mode, operation is assured under the following conditions:

### Normally not ready mode

- BCLK (CPU bus clock) is 50 MHz or less.
- The XWAIT setup time is 9.0 ns or less.

### Normally ready mode

- Three cycles or more are set for the software wait.

## 4.7 Wait

### Software wait

The software wait is a wait performed on the CPU side; this wait specifies how many cycles of the ready signal (XRDY) sampling timing is ignored.

### Hardware wait

The hardware wait is a wait on the CORAL side that occurs when CORAL itself cannot read/write data immediately.

## 4.8 Memory Map

The following shows the memory map of CORAL to the host CPU memory space. The address is mapped differently in SH3, SH4 and V832.

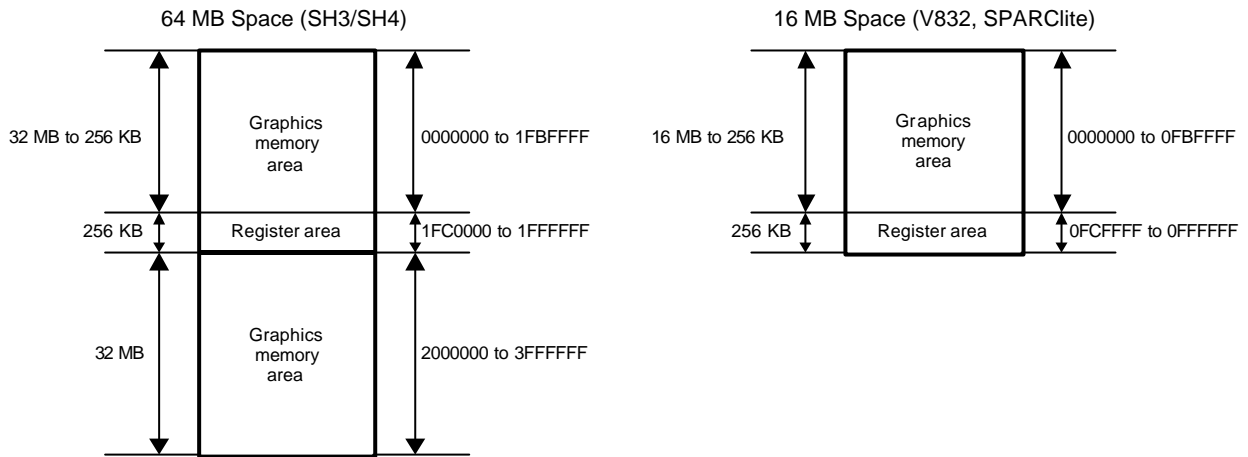


Fig. 4.1 Memory Map

Table 4-4 Address Space in SH3/SH4 Mode

Size	Resource	Base address	(Name)
32 MB to 256 KB		00000000	
64 KB	Host interface registers	01FC0000	(HostBase)
32 KB	Display registers	01FD0000	(DisplayBase)
32 KB	Drawing registers	01FF0000	(DrawBase)
32 KB	Geometry engine registers	01FF8000	(GeometryBase)
32 MB	Graphics memory	02000000	

Table 4-5 Address Space in V832, SPARClike Mode

Size	Resource	Base address	(Name)
16 MB to 256 KB	Graphics memory	00000000	
64 KB	Host interface registers	00FC0000	(HostBase)
32 KB	Display registers	00FD0000	(DisplayBase)
32 KB	Drawing registers	00FF0000	(DrawBase)
32 KB	Geometry engine registers	00FF8000	(GeometryBase)

When the SH3 or SH4 mode is used, the register area can be moved by writing 1 to bit 0 at HostBase + 005Ch (RSW: Register location Switch). In the initial state, the register space is at the center (1FC0000) of the 64 MB space; access CORAL after about 20 bus clocks after writing 1 to RSW.

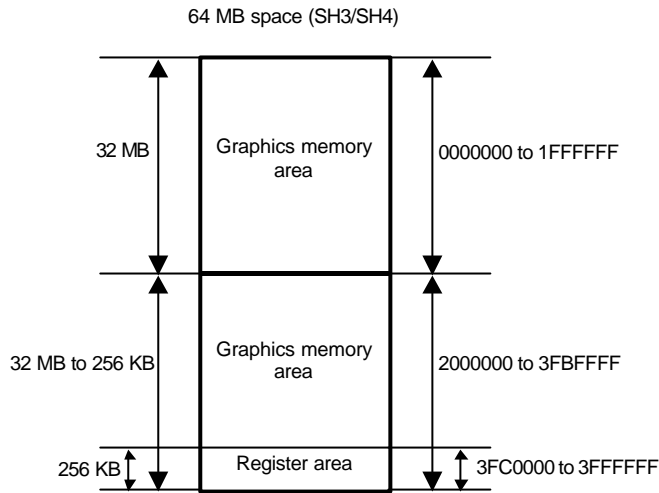


Fig. 4.2 Memory Map

Table 4-6 Address Mapping in SH3/SH4 Mode

Size	Resource	Base address	(Name)
64 MB to 256 KB	Graphics memory	00000000	
64 KB	Host interface registers	03FC0000	(HostBase)
32 KB	Display registers	03FD0000	(DisplayBase)
32 KB	Drawing registers	03FF0000	(DrawBase)
32 KB	Geometry engine registers	03FF8000	(GeometryBase)

# 5 Graphics Memory

## 5.1 Configuration

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

### 5.1.1 Data type

The Coral handles the following types of data. Display list can be stored in the host (main) memory as well. Texture/tile 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. The Coral is able to have plural drawing frames and display a part of these area if it is set to be bigger than display size. The maximum size is 4096x4096 pixel in 32 pixel units. And both indirect color ( 8 bits / pixel) and direct color ( 16 bits / pixel) mode are applicable.

#### Display Frame

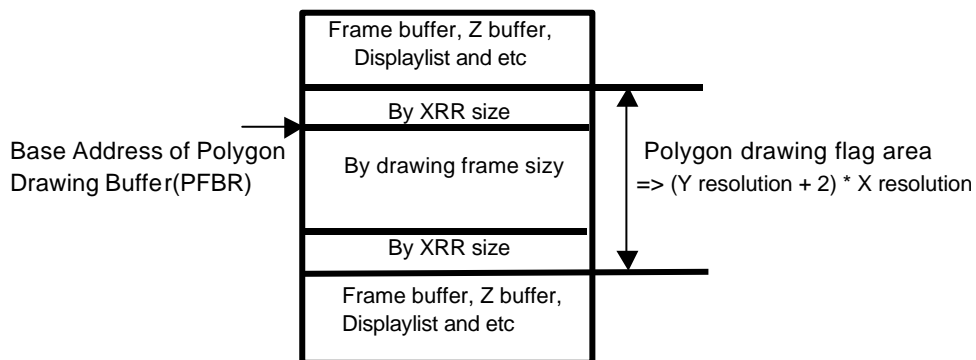
This is a rectangle picture area for display. The Coral is able to set display layer up to 6 layers.

#### Z Buffer

Z buffer is required for eliminating hidden surfaces. In 16 bits modes, 2 bytes and in 8 bits mode, 1 byte are required per 1 pixel. This area has to be cleared before drawing.

#### Polygon Drawing Flag Buffer

This area is used for polygon drawing. It is required 1 bit memory area per 1 pixel and 1 x-axis line area both backward and forward of it. This area has to be cleared before drawing.



#### Displaylist Buffer

The displaylist is a list of drawing commands and parameters.

#### Texture Pattern

This pattern is used for texture mapping. The maximum size is up to 4096 x 4096 pixels.

#### Cursor Pattern

This is used for hardware cursor. The data format is indirect color ( 8 bits / pixel) mode. And the Coral is able to display two cursor of 64 x 64 pixel size.

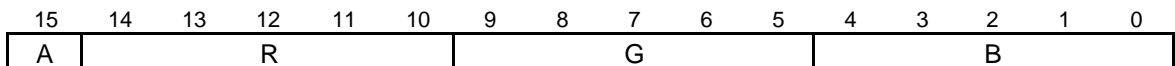
### 5.1.2 Memory Mapping

A graphics memory is mapped linearly to host CPU address field. Each of these above data is able to be allocated anywhere in the Graphics memory according to the respective register setting. ( However there is some restrictions of an addressing boundary depending on a data type.)

### 5.1.3 Data Format

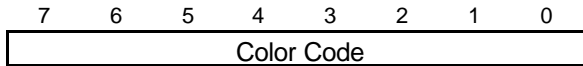
#### Direct Color ( 16 bits / pixel )

This data format is described RGB as each 5 bit. Bit15 is used for alpha bit of layer blending.



#### Indirect Color ( 8 bits / pixel )

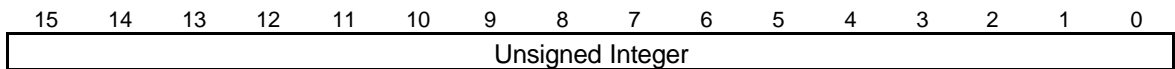
This data format is a color index code for looking up table (palette).



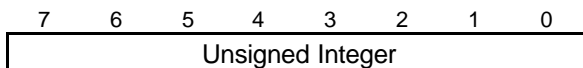
#### Z Value

It is possible to use Z value as 8 bits or 16 bits. These data format are unsigned integer.

##### 1 ) 16 bits mode

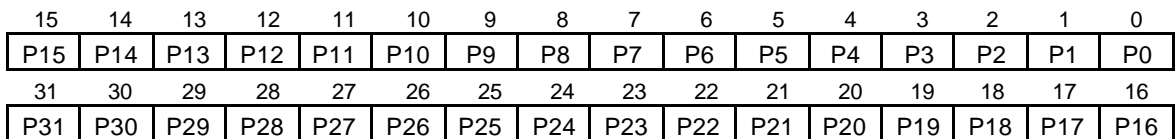


##### 2 ) 8 bits mode



#### Polygon Drawing Flag

This data format is 1 bit per 1 pixel.

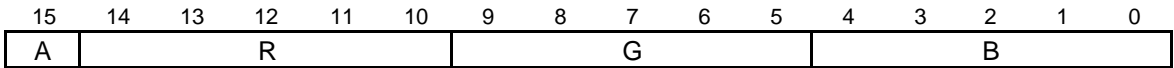


**Texture / Tile Pattern**

It is possible to use a pattern as direct color mode ( 16 bits / pixel) or indirect color mode ( 8 bits / pixel).

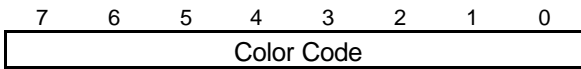
1 ) Direct color mode ( 16 bits / pixel)

This data format is described RGB as each 5 bit. Bit15 is used for alpha bit of stencil or stencil blending. ( Only texture mapping)



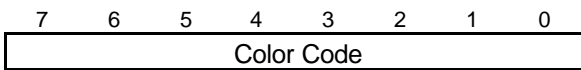
2) Indirect color mode ( 8 bits / pixel)

This data format is a color index code for looking up table (palette).



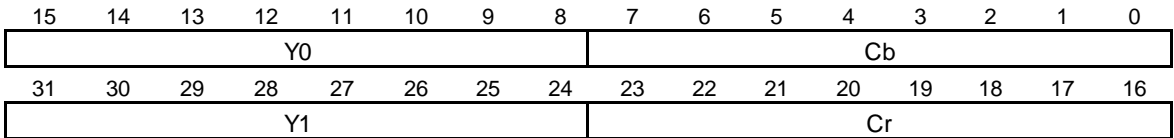
**Cursor Pattern**

This data format is a color index code for looking up table (palette).



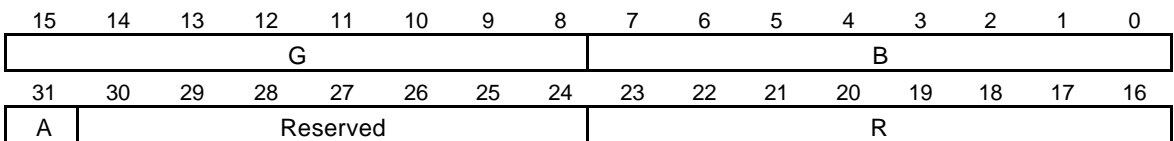
**Video Capture data**

This data format is Y:Cb:Cr=4:2:2 and 32 bits per 2 pixel.



**Direct Color ( 32 bits / pixel )**

This data format is described RGB as each 8 bit. Bit31 is used for alpha bit of layer blending. But the Coral does not support this color mode drawing. Therefore please draw this layer by CPU writing.





## **5.2 Frame Management**

### **5.2.1 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.

### **5.2.2 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 L2, L3, L4, L5 layers.

## **5.3 Memory Access**

### **5.3.1 Memory Access by host CPU**

Graphics memory is mapped linearly to host CPU address field. The host CPU can access the Graphics memory like a SRAM.

### **5.3.2 Priority of memory accessing**

The priority of Graphics memory accessing is the follows:

1. Refresh
2. Display processing
3. Host CPU accessing
4. Drawing accessing

## 5.4 Connection with memory

### 5.4.1 Connection with memory

The memory controller of Coral supports simple connection with SD/FCRAM by setting MMR(Memory Mode Register).

If there is N(=11 to 13) address pins in SD/FCRAM, please connect the SD/FCRAM address(A[n]) pin to the Coral's memory address(MA[n]) pin and SD/FCRAM bank pin to the Coral's next address(MA[N]) pin. Then please set MMR by a number and type of memory.

The follows are the connection table between Coral pin and SD/FCRAM pin.

**64M bit SDRAM(x16 bit)**

Coral pins	SDRAM pins
MA[11:0]	A[11:0]
MA12	BA0
MA13	BA1

**64M bit SDRAM(x32 bit)**

Coral pins	SDRAM pins
MA[10:0]	A[10:0]
MA11	BA0
MA12	BA1

**128M bit SDRAM(x16 bit)**

Coral pins	SDRAM pins
MA[11:0]	A[11:0]
MA12	BA0
MA13	BA1

**128M bit SDRAM(x32 bit)**

Coral pins	SDRAM pins
MA[11:0]	A[11:0]
MA12	BA0
MA13	BA1

**256M bit SDRAM(x16 bit)**

Coral pins	SDRAM pins
MA[12:0]	A[12:0]
MA13	BA0
MA14	BA1

**16M bit FCRAM(x16 bit)**

Coral pins	FCRAM pins
MA[10:0]	A[10:0]
MA11	BA



# 6 DISPLAY CONTROLLER

## 6.1 Overview

### Display control

Window display can be performed for six layers. Window scrolling, etc., can also be performed.

### Backward compatibility

Backward compatibility with previous products is supported in the four-layer display mode or in the left/right split display mode.

### Video timing generator

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

### Color look-up

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

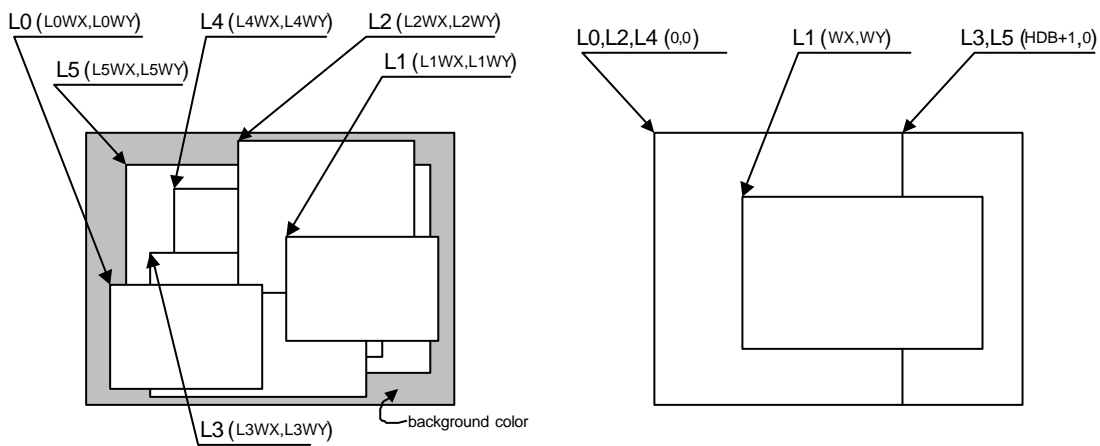
### Cursor

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

## 6.2 Display Function

### 6.2.1 Layer configuration

Six-layer window display is performed. Layer overlay sequence can be set in any order. A four-layer display mode and left/right split display mode are also provided, supporting backward compatibility with previous products.



(a) Six layered window display

(b) Four layered display for downward compatibility

### Configuration of Display Layers

The correspondence between the display layers for this product and for previous products is shown below.

Layer correspondence		Coordinates of starting point		Width/height	
		Window mode	Compatibility mode	Window mode	Compatibility mode
L0	C	(LOWX, LOWY)	(0, 0)	(LOWW, LOWH + 1)	(HDP + 1, VDP + 1)
L1	W	(L1WX, L1WY)	(WX, WY)	(L1WW, L1WH + 1)	(WW, WH + 1)
L2	ML	(L2WX, L2WY)	(0, 0)	(L2WW, L2WH + 1)	(HDB + 1, VDP + 1)
L3	MR	(L3WX, L3WY)	(HDB, 0)	(L3WW, L3WH + 1)	(HDP - HDB, VDP + 1)
L4	BL	(L4WX, L4WY)	(0, 0)	(L4WW, L4WH + 1)	(HDB + 1, VDP + 1)
L5	BR	(L5WX, L5WY)	(HDB, 0)	(L5WW, L5WH + 1)	(HDP - HDB, VDP + 1)

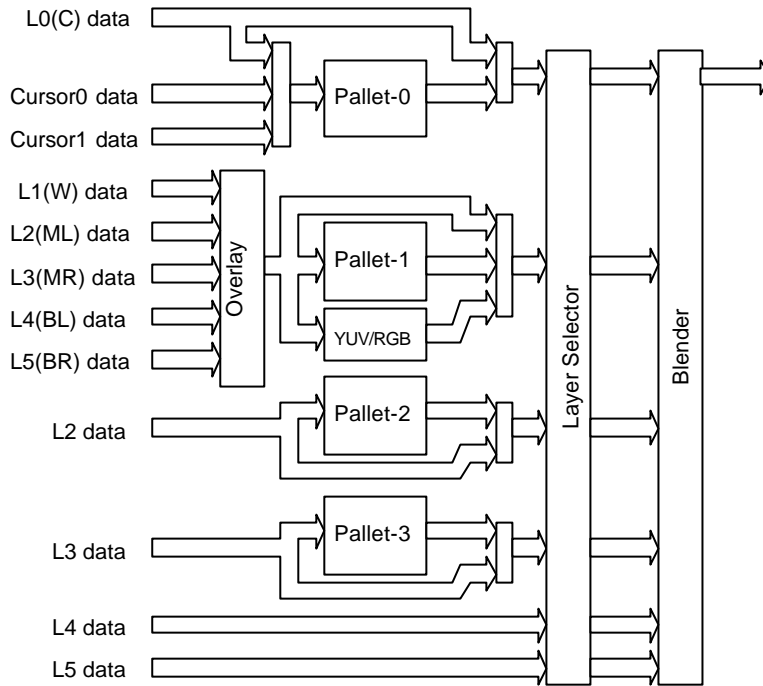
C, W, ML, MR, BL, and BR above mean layers for previous products. The window mode or the compatibility mode can be selected for each layer. It is possible to use new functions through minor program changes by allowing the coexistence of display modes instead of separating them completely.

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

## 6.2.2 Overlay

### (1) Overview

Image data for the six layers (L0 to L5) is processed as shown below.



The fundamental flow is: Palette → Layer selection → Blending. The palettes convert 8-bit color codes to the RGB format. The layer selector exchanges the layer overlay sequence arbitrarily. The blender performs blending using the blend coefficient defined for each layer or overlays in accordance with the transparent-color definition.

The L0 layer corresponds to the C layer for previous products and shares the palettes with the cursor. As a result, the L0 layer and cursor are overlaid before blend operation.

The L1 layer corresponds to the W layer for previous products. To implement backward compatibility with previous products, the L1 layer and lower layers are overlaid before blend operation.

The L2 to L5 layers have two paths; in one path, these layers are input to the blender separately and in the other, these layers and the L1 layer are overlaid and then are input to the blender. When performing processing using the extended mode, select the former; when performing the same processing as previous products, select the latter. It is possible to specify which one to select for each layer.

## (2) Overlay mode

Image layer overlay is performed in two modes: simple priority mode, and blend mode.

In the simple priority mode, processing is performed according to the transparent color defined for each layer. When the color is a transparent color, the value of the lower layer is used as the image value for the next stage; when the color is not a transparent color, the value of the layer is used as the image value for the next stage.

$$\begin{aligned} D_{\text{view}} &= D_{\text{new}} \text{ (when } D_{\text{new}} \text{ does not match transparent color)} \\ &= D_{\text{lower}} \text{ (when } D_{\text{new}} \text{ matches transparent color)} \end{aligned}$$

When the L1 layer is in the YCbCr mode, transparent color checking is not performed for the L1 layer; processing is always performed assuming that transparent color is not used.

In the blend mode, the blend ratio “r” defined for each layer is specified using 8-bit tolerance, and the following operation is performed:

$$D_{\text{view}} = D_{\text{new}} * r + D_{\text{lower}} * (1 - r)$$

Blending is enabled for each layer by mode setting and a specific bit of the pixel is set to “1”. For 8 bits/pixel, the MSB of RAM data enables blending; for 16 bits/pixel, the MSB of data of the relevant layer enables blending; for 24 bits/pixel, the MSB of the word enables blending.

## (3) Blend coefficient layer

In the normal blend mode, the blend coefficient is fixed for each layer. However, in the blend coefficient layer mode, the L5 layer can be used as the blend coefficient layer. In this mode, the blend coefficient can be specified for each pixel, providing gradation, for example. When using this mode, set the L5 layer(L5M and L5EM register) to 8 bits/pixel, window display mode and extend overlay mode.

### 6.2.3 Display parameters

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

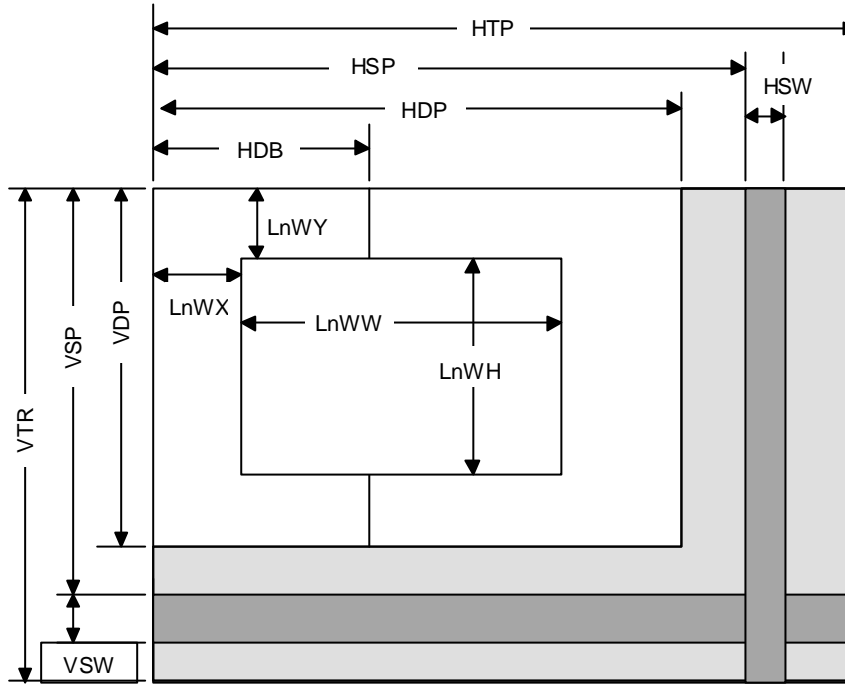


Fig. 5.1 Display Parameters

Note: The actual parameter settings are little different from the above. The details, please refer “11.3.1 Interlaced mode”.

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
LnWX	Layer n Window position X
LnWY	Layer n Window position Y
LnWW	Layer n Window Width
LnWH	Layer n Window Height

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

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

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

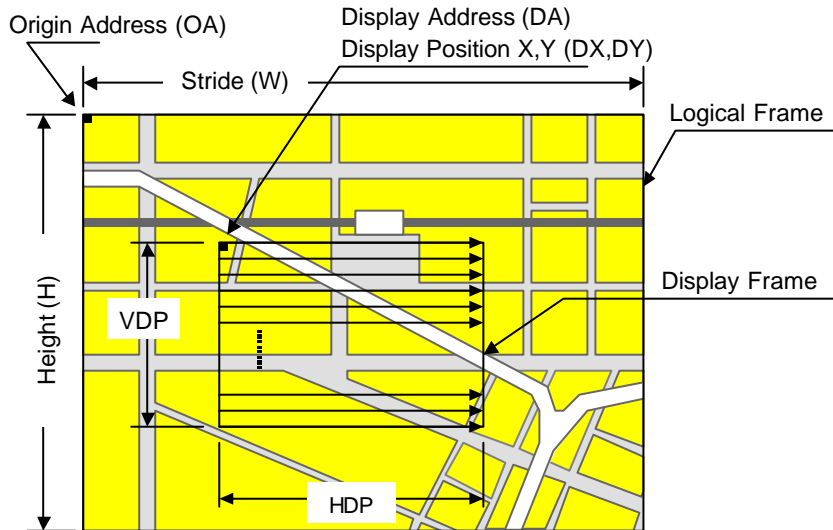


## 6.2.4 Display position control

The graphic image data to be displayed is located in the logical 2D coordinates space (logical graphics space) in the Graphics Memory. There are six logical graphics spaces as follows:

- L0 layer
- L1 layer
- L2 layer
- L3 layer
- L4 layer
- L5 layer

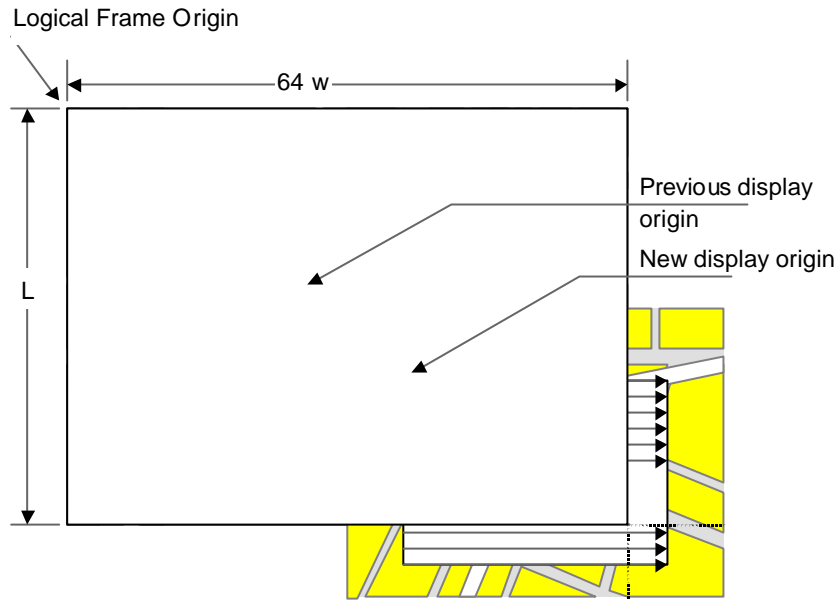
The relation between the logical graphics space and display position is defined as follows:



**Fig. 5.2 Display Position Parameters**

OA	Origin Address	Origin address of logical graphics space. Memory address of top left edge pixel in logical frame origin
W	Stride	Width of logical graphics space. Defined in 64-byte unit
H	Height	Height of logical graphics space. 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 coordinates. Coordinates in logical frame space of display frame origin

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



**Fig. 5.3 Wrap Around of Display Frame**

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

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

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

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

$$0 \leq DY < H$$

DX, DY, and DA have to indicate the same point within the frame. In short, the following relationship must be satisfied.

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

## 6.3 Display Color

Color data is displayed in the following modes:

### Indirect color (8 bits/pixel)

In this mode, the index of the palette RAM is displayed. Data is converted to image data consisting of 6 bits for R, G, and B via the palette RAM and is then displayed.

### Direct color (16 bits/pixel)

Each level of R, G, and B is represented using 5 bits.

### Direct color (24 bits/pixel)

Each level of R, G, and B is represented using 8 bits.

### YCbCr color (16 bits/pixel)

In this mode, image data is displayed with YCbCr = 4:2:2. Data is converted to image data consisting of 8 bits for R, G, and B using the operation circuit and is then displayed.

The display colors for each layer are shown below.

Layer	Compatibility mode	Extended mode
L0	Direct color (16, 24), Indirect color (P0)	Direct color (16, 24), Indirect color (P0)
L1	Direct color (16, 24), Indirect color (P1), YCbCr	Direct color (16, 24), Indirect color (P1), YCbCr
L2	Direct color (16, 24), Indirect color (P1)	Direct color (16, 24), Indirect color (P2)
L3	Direct color (16, 24), Indirect color (P1)	Direct color (16, 24), Indirect color (P3)
L4	Direct color (16, 24), Indirect color (P1)	Direct color (16, 24)
L5	Direct color (16, 24), Indirect color (P1)	Direct color (16, 24)

“Pn” stands for the corresponding palette RAM. Four palettes are used as follows:

### Palette 0 (P0)

This palette corresponds to the C-layer palette for previous products. This palette is used for the L0 layer. This palette can also be used for the cursor.

### Palette 1 (P1)

This palette corresponds to the M/B layer palette for previous products. In the compatibility mode, this palette is common to layers L1 to 5. In the extended mode, this palette is dedicated to the L1 layer.

### Palette 2 (P2)

This palette is dedicated to the L2 layer. This palette can be used only for the extended mode.

### Palette 3 (P3)

This palette is dedicated to the L2 layer. This palette can be used only for the extended mode.

## **6.4 Cursor**

### **6.4.1 Cursor display function**

CORAL can display two hardware cursors. Each cursor is specified as 64 × 64 pixels, and the cursor pattern is set in the Graphics Memory. The indirect color mode (8 bits/pixel) is used and the L0 layer palette is used. However, transparent color control (handling of transparent color code and code 0) is independent of L0 layer. Blending with lower layer is not performed.

### **6.4.2 Cursor control**

The display priority for hardware cursors is programmable. The cursor can be displayed either on upper or lower the L0 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.

Usually, cursor 0 is preferred to cursor 1. However, with cursor 1 displayed upper the L0 layer and cursor 0 displayed lower the L0 layer, the cursor 1 display is preferred to the cursor 0.

## 6.5 Display Scan Control

### 6.5.1 Applicable display

The following table shows typical display resolutions and their synchronous 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 (400.9 MHz at input frequency of 14.318 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 4-1 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	1/60	6.7 MHz	424	15.76 kHz	263	59.9 Hz
400 × 240	1/48	8.4 MHz	530	15.76 kHz	263	59.9 Hz
480 × 240	1/40	10.0 MHz	636	15.76 kHz	263	59.9 Hz
640 × 480	1/16	25.1 MHz	800	31.5 kHz	525	59.7 Hz
854 × 480	1/12	33.4 MHz	1062	31.3 kHz	525	59.9 Hz
800 × 600	1/10	40.1 MHz	1056	38.0 kHz	633	60.0 Hz
1024 × 768	1/6	66.8 MHz	1389	48.1 kHz	806	59.9 Hz

Pixel frequency = 14.318 MHz × 28 × 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

## 6.5.2 Interlace display

CORAL can perform both a non-interlace display and an interlace display.

When the DCM register synchronization mode is set to interlace video (11), images in memory are output in odd and even rasters alternately to each field, and one frame (odd + even fields) forms one screen.

When the DCM register synchronization mode is set to interlace (10), images in memory are output in raster order. The same image data is output to odd fields and even fields. Consequently, the count of rasters on the screen is half of that of interlace video. However, unlike the non-interlace mode, there is a distinction between odd and even fields depending on the phase relationship between the horizontal and vertical synchronous signals.

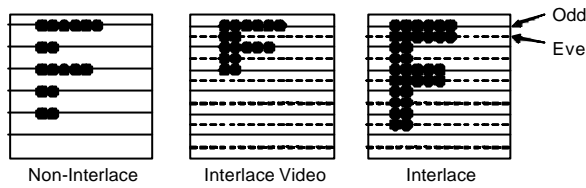
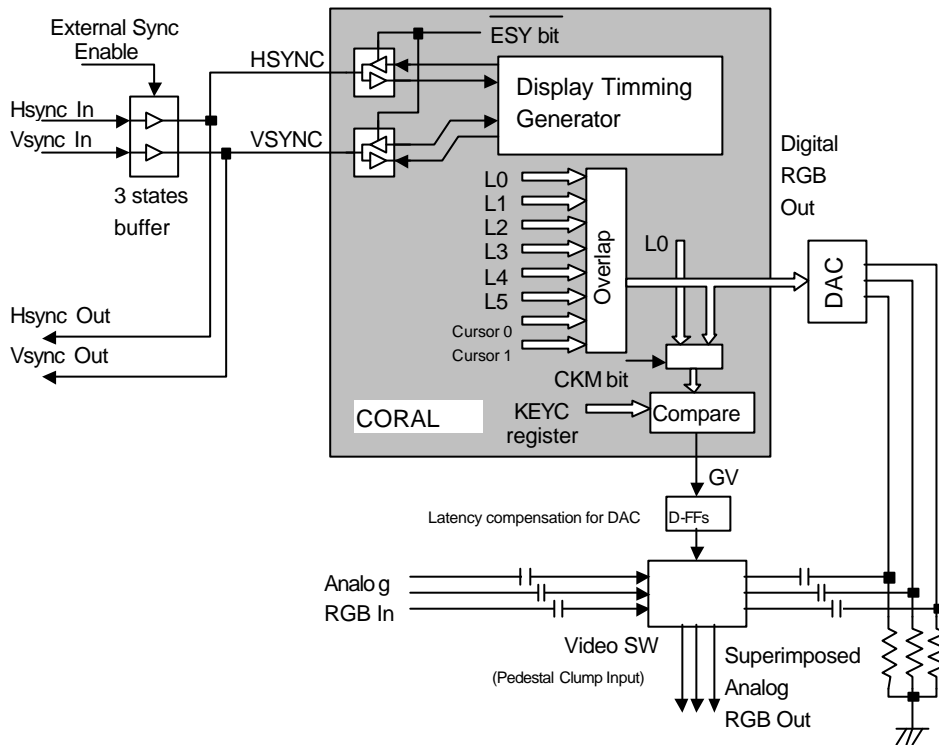


Fig. 5.4 Display Difference between Synchronization Modes

## 6.6 The external synchronous signal

The display scan can be performed by synchronizing horizontal/vertical synchronous signal from the external.

In selecting the external synchronization mode, Coral is sampling the HSYNC signal and displays the synchronizing the external video signal. Either the internal PLL clock or the DCLKI input signal could be selected for the sampling clock. Also, the superimposed analog output is performed by the chroma key process. The following diagram shows an example of the external synchronization circuit.



**An example of the external synchronization circuit**

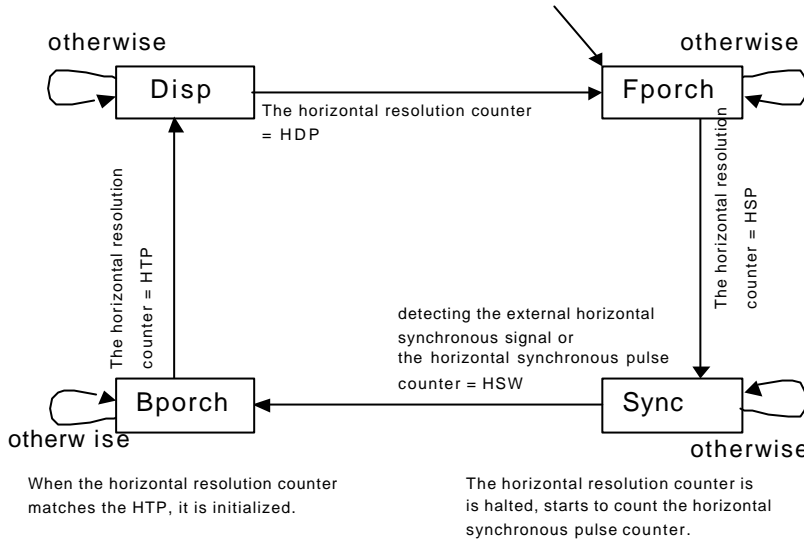
The external synchronization mode is performed by setting the ESY bit of the DCM register. In setting the external synchronization mode, HSYNC, VSYNC, and EO pin of Coral is changed to the input mode. After that it needs to be provided the synchronous signal by using the 3 state buffer from the external. When turning off the external synchronization mode, Coral internal ESY bit needs to be switched OFF after disconnecting the synchronous input signal from the external.

The buffer of the external synchronization signal must not be switched ON when the synchronous output signal of Coral is ON. Follow the previous instruction to prevent simultaneous ON from occurring.

In using the external synchronous signal with the display clock based on the internal PLL, Coral extends the clock period and fits the clock phase with the horizontal synchronous signal phase after inputting the horizontal synchronous pulse. The following caution is necessary. In case of connecting the high speed transmit signal, such as LVDS, with the digital RGB output, PLL with a built-in the high speed serial transmission is temporally unstable due to this connection. Therefore,

the external synchronous signal based on the internal PLL must not be used with high speed synchronous transmit signal.

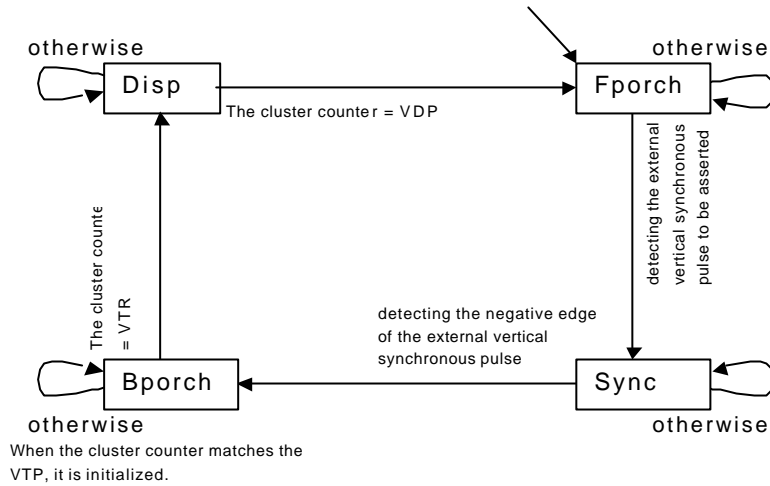
The synchronization of the horizontal direction is controlled by the following state diagram.



The finite state diagram is controlled by the horizontal resolution counter. The period of outputting the signal is assigned the Disp state. When the value of the horizontal resolution counter matches that of the HDP register, it ends to output the signal and the current state is transmitted from Disp state to Fporch state (front porch). In the Fporch state, when the value of the vertical resolution register matches that of the HSP register, the current state is transmitted to the Sync state. In this state, it waits for the horizontal synchronous signal from the external. Coral detects the negative edge of the horizontal synchronous pulse from the external and synchronizes it. In detecting the horizontal synchronous signal from the external, the current state is transmitted to the Bporch state (back porch). The horizontal resolution register does not count in the Sync state, instead the horizontal synchronous counter is incremented from zero. When the value of this counter matches the setting value of the HSW register, the current state is transmitted to the Bporch state without detecting the horizontal synchronous signal from the external. When the value of the horizontal resolution counter matches that of the HTP register in the Bporch state, the horizontal resolution counter is reset, and also the current state is transmitted to the Disp state and it begins to display the next cluster.



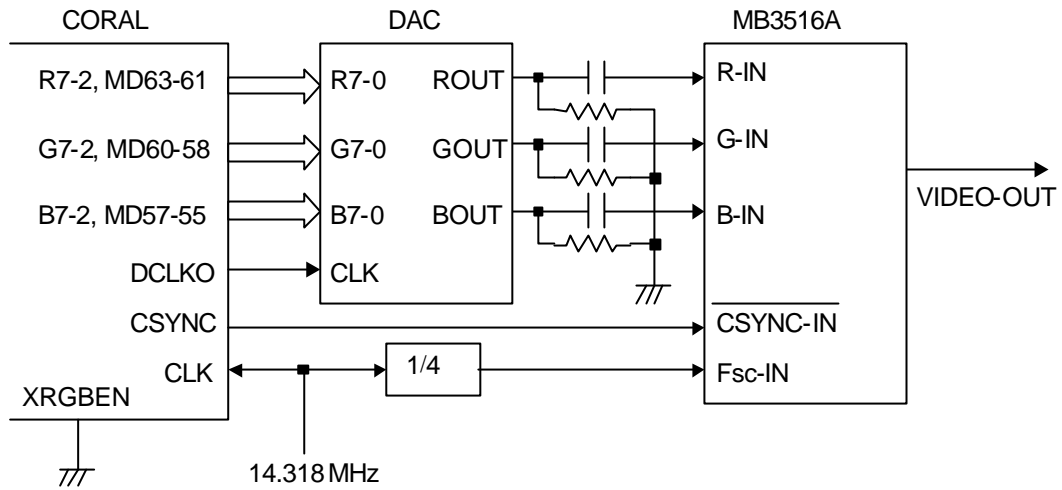
The synchronization of vertical direction is controlled by the following state diagram.



The state diagram of the vertical direction is controlled by the value of the cluster counter. The period of outputting the signal is assigned the Disp state. When the value of the cluster counter matches the value of the VDP register, it ends to output the signal and the current state is transmitted from the Disp state to the Fporch state. In the Fporch state, it waits the external synchronous pulse to be asserted. In detecting the external synchronous pulse to be asserted, the current state is transmitted to the Sync state. In the Sync state, it waits for the negative edge of the external synchronous signal. In detecting the negative edge, the current state is transmitted to the Bporch state. When the value of the cluster counter matches the values of the VTR register, the cluster counter is reset, and also the current state is transmitted to the Disp state and it starts to display the next field.

## 6.7 Video Interface, NTSC/PAL Output

In outputting NTSC signal, NTSC/PAL encoder must be connected externally as shown below:



**Fig. 5.4 Example of NTSC Encoder Connection**

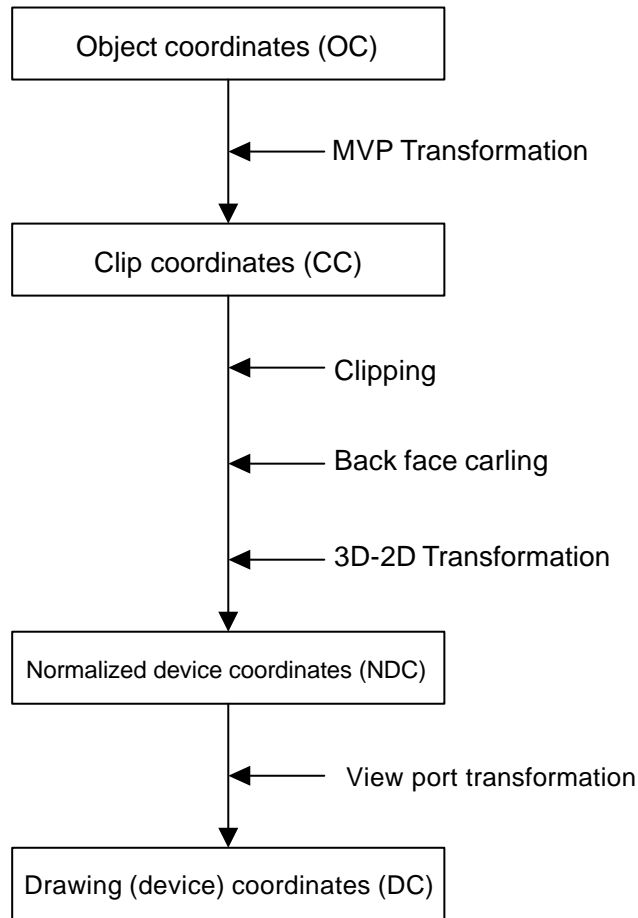
The digital NTSC encoder can also be used, but in general, the usable pixel frequency/resolution are limited. For details, refer to the specifications for each company's digital NTSC encoder.

# 7 GEOMETRY ENGINE

## 7.1 Geometry Pipeline

### 7.1.1 Processing flow

The flow of geometry 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 × 4” matrix consisting of a ModelView matrix and a Projection matrix.

If “Zoc” is not contained 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

$$\begin{pmatrix} X_{cc} \\ Y_{cc} \\ Z_{cc} \\ W_{cc} \end{pmatrix} = \begin{pmatrix} Ma0 & Ma1 & Ma2 & Ma3 \\ Mb0 & Mb1 & Mb2 & Mb3 \\ Mc0 & Mc1 & Mc2 & Mc3 \\ Md0 & Md1 & Md2 & Md3 \end{pmatrix} \begin{pmatrix} X_{oc} \\ Y_{oc} \\ Z_{oc} \\ 1 \end{pmatrix}$$

Ma0 to Md3: OC → CC Matrix

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

Xcc to Wcc: 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

$$\begin{pmatrix} X_{ndc} \\ Y_{ndc} \\ Z_{ndc} \end{pmatrix} = \frac{1}{W_{cc}} \begin{pmatrix} X_{cc} \\ Y_{cc} \\ Z_{cc} \end{pmatrix}$$

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 CORAL 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’ can be evaluated as within view volume range, making it possible to obtain the effect of view volume clipping-off.

This method is valid only when W clipping does not occur. When a clip boundary value (Wmin) that causes W clipping to occur is set, clipping is also performed for each clip area. Consequently, set an appropriate clip boundary value for Clip. Max value. and Clip. Min value., respectively.

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 = 0x7f7ffff, minimum value = 0xff7ffff

\*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

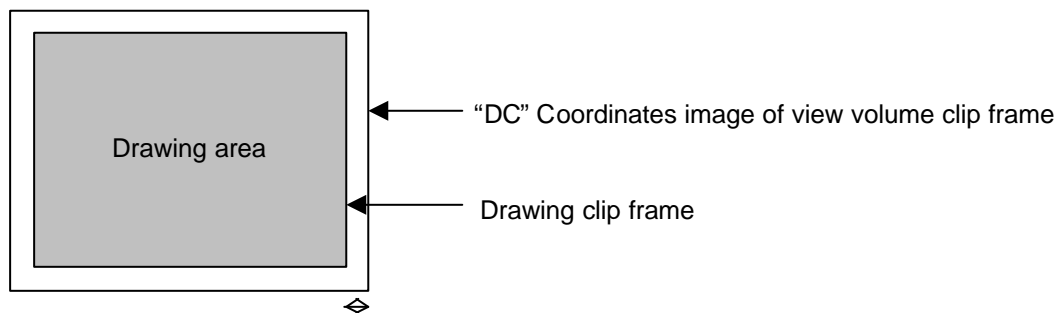
#### “W” clipping at orthogonal projection transformation

“W” at orthogonal projection transformation ( $GMDR0[0] = 0$ ) is treated as “ $W_{cc}=1.0$ ”. For this reason, to suppress “W” clipping, the set “ $W_{min}$ ” value must be larger 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 larger 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 specifying of the line drawing attribute (end point drawing/non-drawing).
- (3) When the start 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.



A space of two pixels or more is required.

### **7.1.6 Back face curling**

In CORAL, a triangle direction can be defined and a mode in which drawing for the back face 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 only when back face carling is required. When back face carling is not required such as in 'line,' 'point,' and 'polygon primitive,' GMDR2[0] must be set to 0.

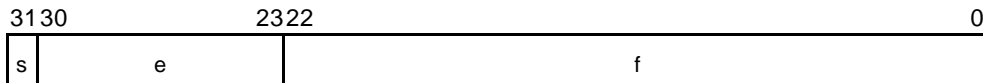
## 7.2 Data Format

### 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.

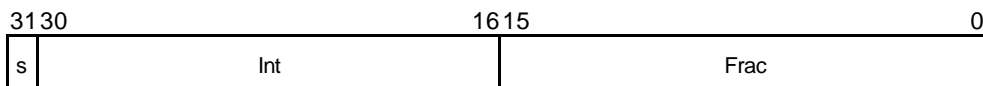
(1) 32-bit single-precision floating-point format



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

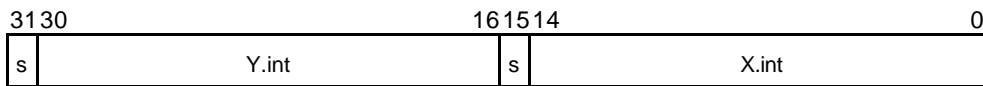
The numerical value of the floating-point format becomes  $(-1)^s(1.f)2^{(e-127)}$  ( $0 < e < 255$ ).

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



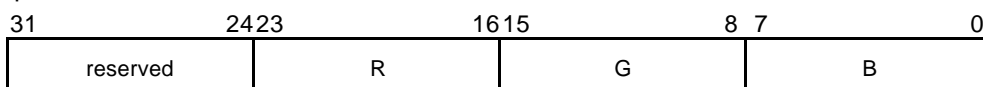
- s: Sign bit (1 bit)
- int: Integer (15 bits)
- frac: Fraction (16 bits)

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



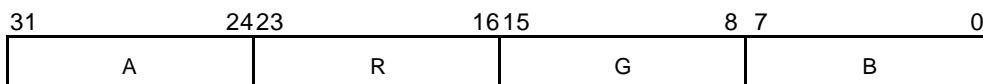
- s: Sign bit (1 bit)
- int: Integer (15 bits)

(4) RGB packed format



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

(5) ARGB packed format



A: Alpha bits (8 bits)  
 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. CORAL 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.

## **7.4 Log Output of Device Coordinates**

A function is provided to output device coordinates (DC) data obtained by view port conversion to local memory (graphics memory).

### **7.4.1 Log output mode**

#### **Drawing & log output command**

Log output of drawing coordinates (device coordinates) can be performed concurrently with primitive drawing.

Log output can be controlled using the command with log output on/off attribute; log output is performed only when the log output on attribute is specified.

#### **Log output dedicated command**

When the log output dedicated command is used, log output of the device coordinates can be performed.

### **7.4.2 Log output destination address**

The log output destination address is controlled by the device coordinates log pointer. Once set an address, this pointer automatically increment an output address.

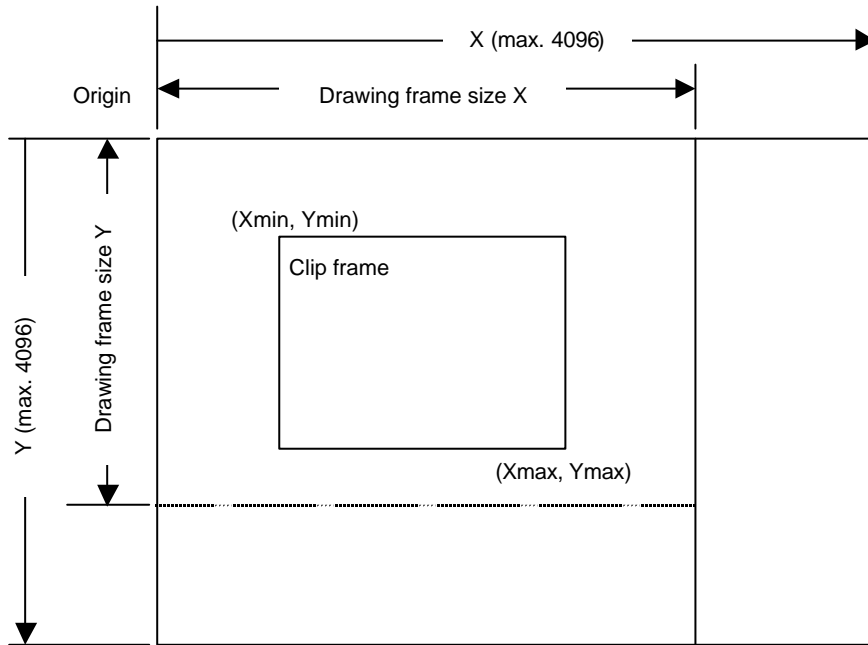
# 8 DRAWING PROCESSING

## 8.1 Coordinate System

### 8.1.1 Drawing coordinates

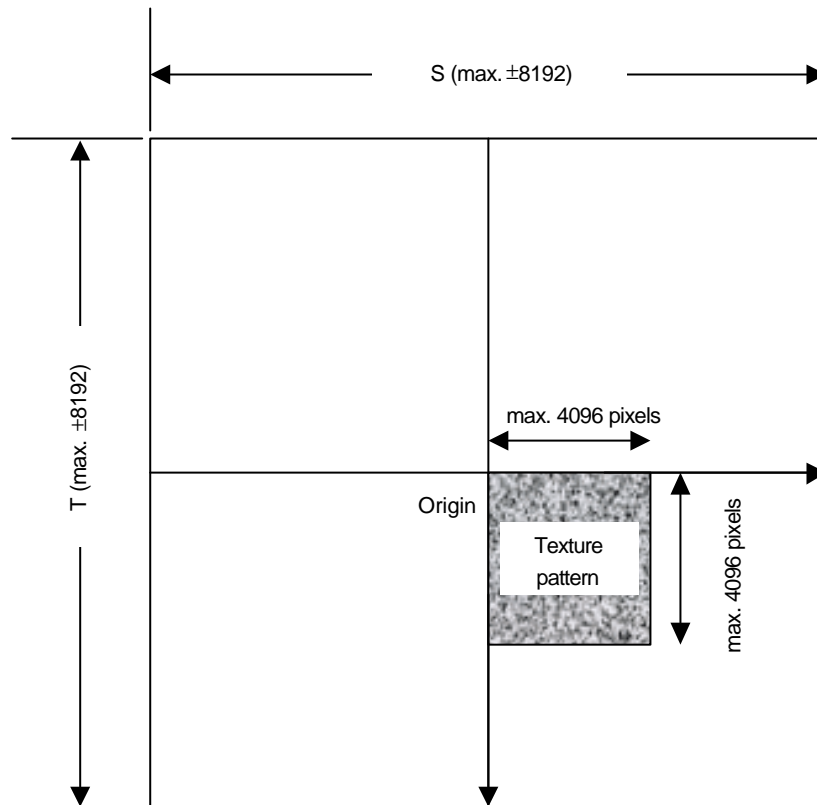
After the calculation of coordinates by the geometry engine, CORAL draws data in the drawing frame in the graphics memory that finally uses the drawing coordinates (device coordinates).

Drawing frame is treated as 2D coordinates with the origin at the top left as shown in the figure below. The maximum coordinates is  $4096 \times 4096$ . Each drawing frame is located in the Graphics Memory by setting the address of the origin and resolution of X direction (size). Although the size of Y direction does not need to be set, Y coordinates which are max. at drawing must not be overlapped with other area. In addition, at drawing, specifying the clip frame (top left and bottom right coordinates) can prevent the drawing of images outside the clip frame.



## 8.1.2 Texture coordinates

Texture coordinate is a 2D coordinate system represented as S and T (S: horizontal, T: vertical). Any integer in a range of  $-8192$  to  $+8191$  can be used as the S and T coordinates. The texture coordinates is correlated to the 2D coordinates of a vertex. One texture pattern can be applied to up to  $4096 \times 4096$  pixels. The pattern size is set in the register. When the S and T coordinates exceed the maximum pattern size, 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 (count of pixels on X direction) is common for these areas.

#### **Drawing frame**

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

#### **Z buffer**

Z buffer is required for eliminating hidden surfaces. In 16 bits mode, 2 bytes and in 8 bits mode, 1 byte are required per 1 pixel.

#### **Polygon drawing flag buffer**

This area is used for polygon drawing. 1bit is required per 1 pixel.

## **8.2 Figure Drawing**

### **8.2.1 Drawing primitives**

CORAL 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
- High-speed 2DLine
- High-speed 2DTriangle
- Polygon

### **8.2.2 Polygon drawing function**

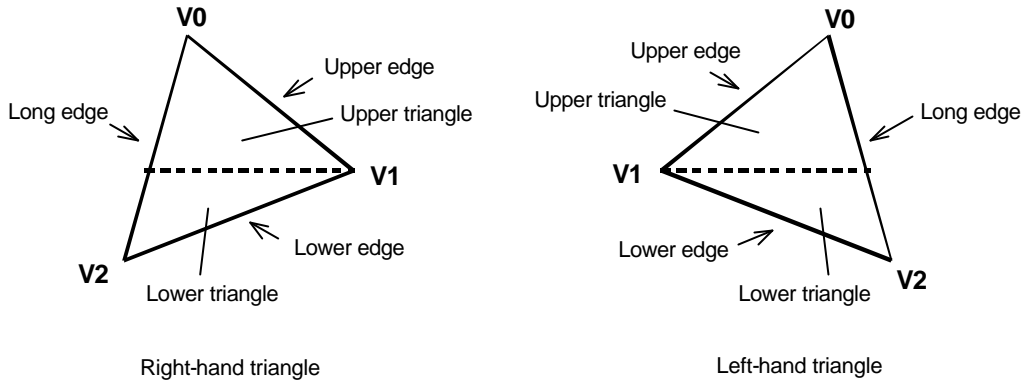
An irregular polygon (including concave shape) is drawn by hardware in the following manner:

1. Execute PolygonBegin command.  
Initialize polygon drawing hardware.
2. Draw vertices.  
Draw outline of polygon and plot all vertices to polygon draw flag buffer using high-speed 2DTriangle primitive.
3. Execute PolygonEnd command.  
Copy shape in polygon draw flag buffer to drawing frame and fill shape with color or specified tiling pattern.

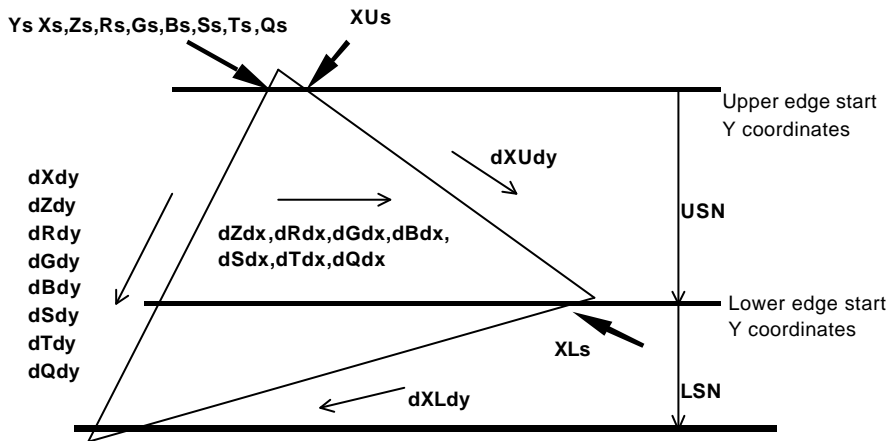
### 8.2.3 Drawing parameters

The MB86290A-compatible interface uses the following parameters for drawing:

The triangles (Right triangle and Left triangle) are distinguished according to the locations of three vertices as follows (not used for high-speed 2DTriangle):



The following parameters are required for drawing triangles (for high-speed 2DTriangle, 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 coordinates)  $>$   $XUs$  or  $Xs$  (lower edge start Y coordinates)  $>$   $XLs$ , the appropriate picture may not be drawn.

Ys	Y coordinates start position of long edge in drawing triangle
Xs	X coordinates start position of long edge corresponding to Ys
XUs	X coordinates start position of upper edge
XLs	X coordinates start position of lower edge
Zs	Z coordinates start position of long edge corresponding to Ys
Rs	R color value of long edge corresponding to Ys
Gs	G color value of long edge corresponding to Ys
Bs	B color value of long edge corresponding to Ys
Ss	S coordinate of textures of long edge corresponding to Ys
Ts	T coordinate of textures of long edge corresponding to Ys
Qs	Q perspective correction value of texture of long edge corresponding to Ys
dXdY	X DDA value of long edge direction
dXUdy	X DDA value of upper edge direction
dXLdy	X DDA value of lower edge direction
dZdy	Z DDA value of long edge direction
dRdy	R DDA value of long edge direction
dGdy	G DDA value of long edge direction
dBdy	B DDA value of long edge direction
dSdy	S DDA value of long edge direction
dTdy	T DDA value of long edge direction
dQdy	Q DDA value of long edge direction
USN	Count of spans of upper triangle
LSN	Count of spans of lower triangle
dZdx	Z DDA value of horizontal direction
dRdx	R DDA value of horizontal direction
dGdx	G DDA value of horizontal direction
dBdx	B DDA value of horizontal direction
dSdx	S DDA value of horizontal direction
dTdx	T DDA value of horizontal direction
dQdx	Q DDA value of horizontal direction

### 8.2.4 Anti-aliasing function

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

( The edge of line is blended with a frame buffer color at that time. Ideally please draw sequentially from father object.)

## **8.3 Bit Map Processing**

### **8.3.1 BLT**

A rectangular shape in pixel units can be transferred. There are following types of transfer:

1. Transfer from host CPU to Drawing frame memory
2. Transfer between Graphics Memories including Drawing frame

Concerning 1 and 2 above, 2-term logic operation is performed between source and destination data and its result can be stored.

Setting a transparent color enables a drawing of a specific pixel with transmission.

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 correctly.

### **8.3.2 Pattern data format**

CORAL 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 binary bit map is used for character/font patterns, where foreground color is used for bitmap = 1 pixel, and background color (background color can be set to be transparent by setting) is applied for bitmap = 0 pixels.



## **8.4 Texture Mapping**

### **8.4.1 Texture size**

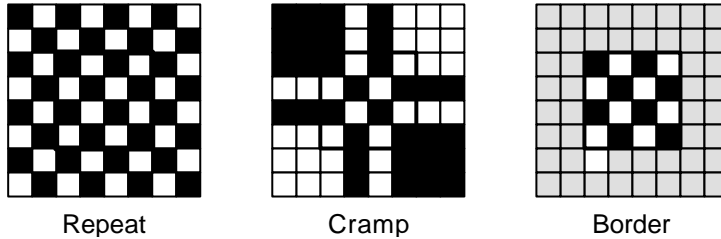
CORAL reads texcel corresponding to the specified texture coordinates (S, T), and draws that data at the correlated pixel position of the polygon. For the S and T coordinates, the selectable texture data size is any value in the range from 4 to 4096 pixels represented as an exponent of 2.

### **8.4.2 Texture color**

Drawing of 8-/16-bit direct color is supported for the texture pattern. For drawing 8-bit direct color, only point sampling can be specified for texture interpolation; only de-curl can be specified for the blend mode.

### 8.4.3 Texture lapping

If a negative or larger than the specified texture pattern size is specified as the texture coordinates (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

This just simply masks the upper bits of the applied (S, T) coordinates. When the texture pattern size is  $64 \times 64$  pixels, the lower 6 bits of the integer part of (S, T) coordinates are used for S and T coordinates.

#### Cramp

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

$S < 0$	$S = 0$
$S > \text{Texture X size} - 1$	$S = \text{Texture X size} - 1$

#### Border

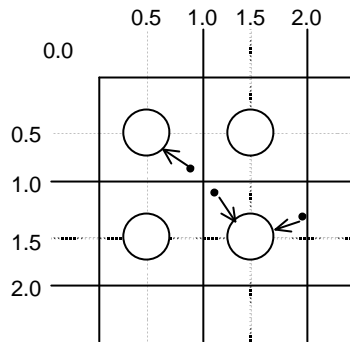
When the applied (S, T) coordinate is either negative or larger than the specified texture pattern size, the outside of the specified texture pattern is rendered in the 'border' color.

## 8.4.4 Filtering

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

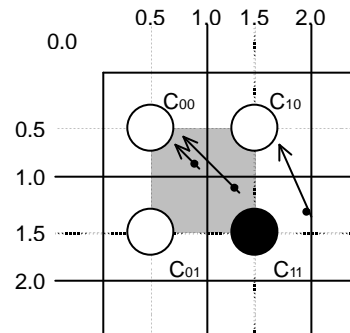
### Point filtering

This mode uses the texture pixel specified by the (S, T) coordinates as they are for drawing. The nearest pixel in the texture pattern is chosen according to the calculated (S, T) coordinates.



### Bi-linear filtering

The four nearest pixels specified with (S, T) coordinate are blended according to the distance from specified point and used in drawing.



## 8.4.5 Perspective correction

This function corrects the distortion of the 3D perspective in the texture mapping. For this correction, the 'Q' component of the texture coordinates ( $Q = 1/W$ ) is set based on the W component of 3D coordinates of the vertex.

When the texture coordinates are large values, the texture may not be drawn correctly when perspective correction is performed. This phenomenon occurs due to the precision limitation of the arithmetical unit for perspective correction. The coordinates for the texture that cannot be drawn normally vary with the value of the Q component; as a guide, when this value, texture coordinates (S, T) is smaller than -2048 or larger than 2048, normal drawing results are less likely to be obtained.

## 8.4.6 Texture blending

CORAL supports the following three blend modes for texture mapping:

### De-curl

This mode displays the selected texture pixel color regardless of the polygon color.

### Modulate

This mode multiplies the native polygon color ( $C_P$ ) and selected texture pixel color ( $C_T$ ) and the result is used for drawing. Rendering color is calculated as follows ( $C_O$ ):

$$C_O = C_T \times C_P$$

### Stencil

This mode selects the display color from the texture color with MSB as a flag.

MSB = 1: Texture color

MSB = 0: Polygon color

## 8.4.7 Bi-linear high-speed mode

Bi-linear filtering is performed at high speed by creating normal texture data in advance with four-pixel redundancy for one pixel.

One pixel requires information of about four pixels, so an area of four times the normal area is used. This data format can only be used only for the bi-linear filtering mode; it cannot be used for the point sampling mode.

The color mode is limited to 16-bit color.

	0	1	2	3	4	5	6	7
0	00	01	02	03	04	05	06	07
1	08	09	10	11	12	13	14	15
2	16	17	18	19	20	21	22	23
3	24	25	26	27	28	29	30	31
4	32	33	34	35	36	37	38	39
5	40	41	42	43	44	45	46	47
6	48	49	50	51	52	53	54	55
7	56	57	58	59	60	61	62	63

**Normal texture layout (8 × 8 pixels)**

	0		1			6		7									
0	00	01	08	09	01	02	09	10	to	06	07	14	15	07	00	15	08
1	08	09	16	17	09	10	17	18	to	14	15	12	13	15	08	23	16
2	16	17	24	25	17	18	25	26	to	22	23	30	31	23	16	31	24
3	24	25	32	33	25	26	33	34	to	30	31	38	39	31	24	39	32
4	32	33	40	41	33	34	41	42	to	38	39	46	47	39	32	47	40
5	40	41	48	49	41	42	49	50	to	46	47	54	55	47	40	55	48
6	48	49	56	57	49	50	57	58	to	54	55	62	63	55	48	63	56
7	56	57	00	01	57	58	01	02	to	62	63	06	07	63	56	07	00

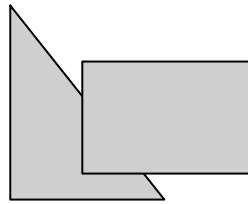
**Texture layout in bi-linear mode (8 × 8 pixels)**

## 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 determines the pixel on the pattern read by pixel coordinates to be drawn, irrespective of position and size of primitive.

The tiling pattern size is limited to within  $64 \times 64$  pixels. (at 16-bit color)



Example of Tiling

### 8.5.2 Alpha blending

Alpha blending blends the drawn in frame buffer to-be-drawn pixel or pixel already according to the alpha value set in the alpha register. This function cannot be used simultaneously with logic operation drawing. 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 color of the pixel to be drawn is  $C_P$ , the color of frame buffer is  $C_F$ , and the alpha value is  $A$ :

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

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 can be selected:

#### Normal

Blends post texture mapping color with frame buffer color

#### Stencil

Uses MSB of texcel color for ON/OFF control:

MSB = 1: Texcel color

MSB = 0: Frame buffer color

#### Stencil alpha

Uses MSB of texcel color for  $\alpha$ /OFF control:

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

MSB = 0: Frame buffer color

Note: MSB of frame buffer is drawn MSB of texcel in both stencil and stencil alpha mode.

Therefore in case MSB of texcel is MSB=0, a color of frame buffer is frame buffer, but MSB of frame buffer is set to 0.

### 8.5.3 Logic operation

This mode executes a logic operation between the pixel to be drawn and the one already drawn in frame buffer and its result is drawn. Alpha blending cannot be used when this function is specified.

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 plane management

CORAL supports the Z buffer for hidden plane management.

This function compares the Z value of a new pixel to be drawn and the existing Z value in the Z buffer. Display/not display is switched according to the Zcompare mode setting. Define the Zbuffer access options in the ZWRITEMASK mode.

The Z compare operation type is determined by the Z compare mode.

Either 16 or 8 bits can be selected for the Z-value.

ZWRITEMASK		
	1	Compare Z values, no Z value write overwrite
	0	Compare Z values, Z value write

Z Compare mode	Code	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 ≤ current Z buffer value
EQUAL	100	Draw if pixel Z value = current Z buffer value
GEQUAL	101	Draw if pixel Z value ≥ 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 drawing attributes

In drawing lines, the following attributes apply:

Line Drawing Attributes

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

### 8.6.2 Triangle drawing attributes

In drawing triangles, the following attributes apply (these attributes are disabled in high-speed 2DTriangle). Texture mapping and tiling have separated texture attributes:

Triangle Drawing Attributes

Drawing Attribute	Description
Shading	Gouraud shading or flat shading selectable. In case of indirect color mode, gray scale gouraud shading is possible.
Alpha blending	Set alpha blending enable/disable per polygon
Alpha blending coefficient	Set color blending ratio of alpha blending

#### How to set gray scale gouraud shading

1. Set Frustum bit of GMDR0 register to 0.
2. Set identity matrix.
3. Set MDR2 register to the below.  
SM bit = 1, ZC bit = 0, ZW bit = 0, BM bit = 00, TT bit = 00
4. Set GG bit of MDR7 register to 1.
5. Execute drawing by same method as a direct color gouraud shading object.  
Note: - Please don't use G\_BeginE command.  
- Please don't use floating data format in G\_Vertex command.  
- R (red) parameter is used as a color parameter.
6. Set GG bit of MDR7 register to 0 after rendering.



### 8.6.3 Texture attributes

In texture mapping, the following attributes apply:

**Texture Attributes**

<b>Drawing Attribute</b>	<b>Description</b>
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 coordinates correction	Select either linear or perspective correction
Texture wrap	Select either repeat or clamp of texture pattern
Texture blend mode	Select either decal or modulate
Bi-linear high-speed mode	Texture data is created in a dedicated format to perform high-speed bi-linear filtering.

### 8.6.4 BLT attributes

In BLT drawing, the following attributes apply:

#### BLT Attributes

Drawing Attribute	Description
Logic operation mode	Specify two source logic operation mode
Transparency mode	Set transparent copy mode and transparent color
Alpha map mode	Blend a color according to alpha map

### 8.6.5 Character pattern drawing attributes

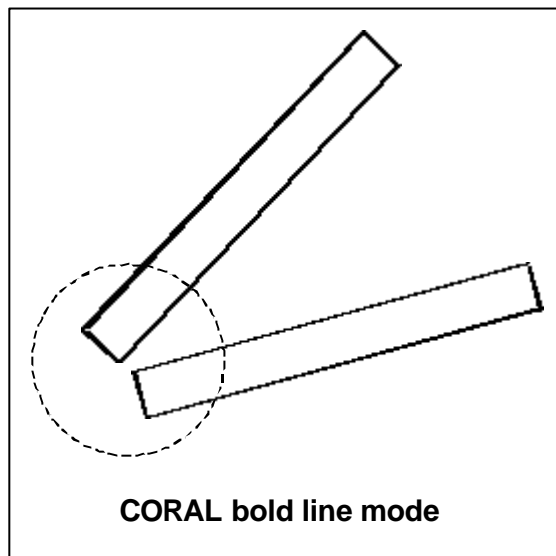
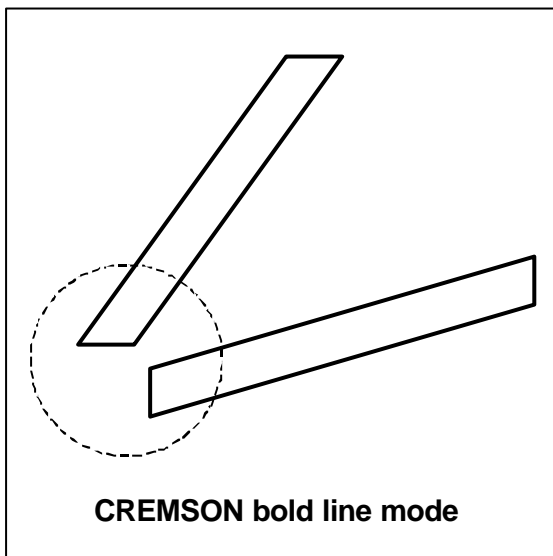
#### Character Pattern Drawing

Drawing Attribute	Description
Character pattern enlarge/shrink	$2 \times 2$ , $\times 2$ 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

## 8.7 Bold Line

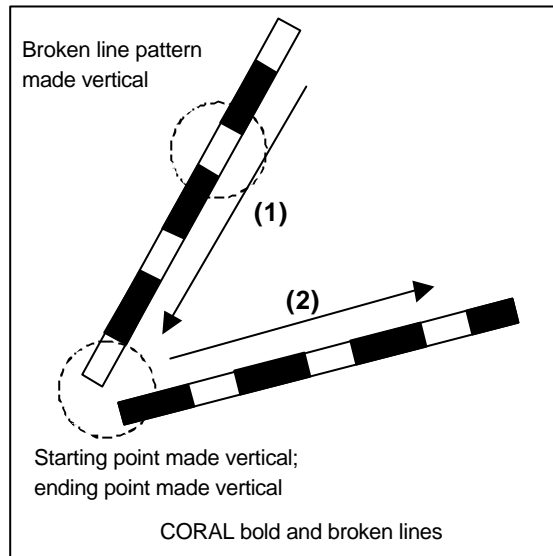
### 8.7.1 Starting and ending points

- In the CREMSON bold line mode, the starting and ending points are vertical to the principal axis.
- In the CORAL bold line mode, the starting and ending points are vertical to the theoretical line.
- Caution: CORAL bold line is generated by different algorithm. Thus drawing position is little bit different from other primitive.



### 8.7.2 Broken line pattern

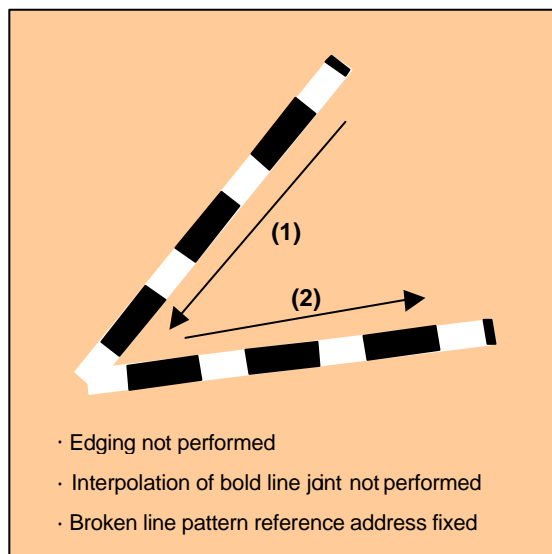
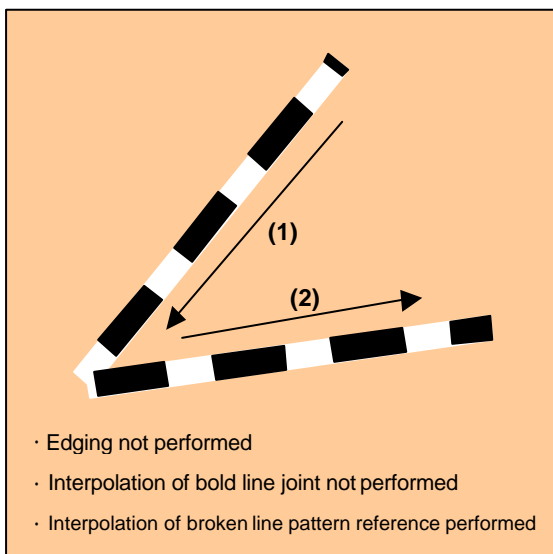
- The broken line pattern vertical to the theoretical line (the CORAL broken line pattern) is supported.
- In the CREMSON bold line mode, lines can be drawn using the broken line pattern vertical to the CREMSON-compatible principal axis (the CREMSON broken line pattern), and can also be drawn using the CORAL broken line pattern.
- In the CORAL bold line mode, only the CORAL broken line pattern is supported.



#### Interpolation of broken line pattern

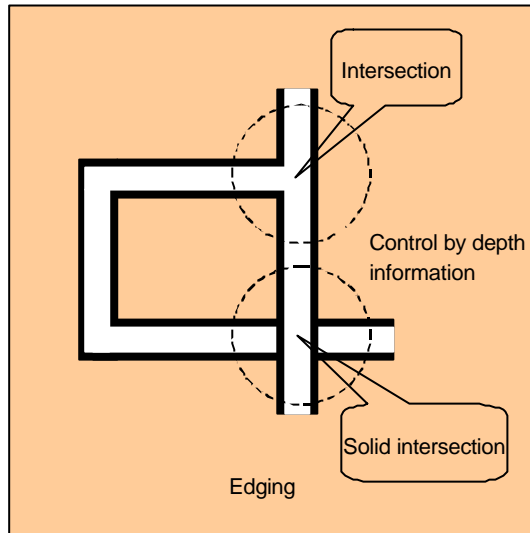
Two types of interpolation modes are supported:

- No interpolation mode: Interpolation is not performed.
- Broken line pattern reference address fix mode: The same broken line pattern is referenced for several pixels before and after the joint of the bold line. Any pixel count can be set by the user.



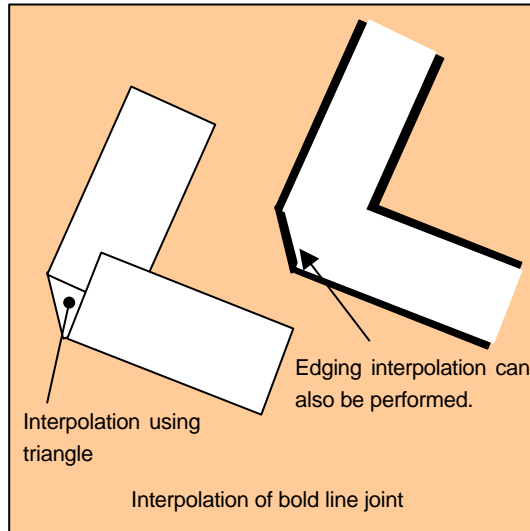
### 8.7.3 Edging

- The edging line is supported.
- The line body and edging section can have depth information (Z offset). This mechanics makes it possible to easily represent a good connection of the overlaid part of the edging line. For example, when the line body depth information and edging section depth information are the same, the drawing result of the edging line is like the intersection shown in the figure below. Also, when the line body depth information and edging section depth information are different, the drawing result of the edging line is like the solid intersection shown in the figure below.



### 8.7.4 Interpolation of bold line joint

- In the bold line joint interpolation mode, the bold line joint is interpolated using a triangle as shown in the figure below.
- The edging line joint is also interpolated using a triangle, but the said depth information makes it possible to represent a good connection as shown in the figure below.
- Caution: Sometime joint shape looks not perfect. ( using approximate calculation)



# 9 DISPLAY LIST

## 9.1 Overview

Display list is a set of display list commands, parameters and pattern data. All display list commands stored in a display list are executed consequently.

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

- Write to display FIFO by CPU
- Transfer from main memory to display FIFO by external DMA
- Transfer from graphics memory to display FIFO by register setting

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 List**

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													
Format 11	Type				Reserved				Reserved							Count						
															Count							

**Description of Each Field**

Type	Display list type
Command	Command
Count	Count of data excluding header
Address	Address value used at data transfer
Vertex	Vertex number
Flag	Attribute flag peculiar to display list command

**Vertex Number Specified in Vertex Code**

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

### 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

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

Type	Command	Description
G_Nop	—	No operation
G_Begin	See <b>Geometry command code table</b> .	Specifies primitive type and pre-processes
G_BeginCont	—	Specifies primitive type (vertex processing in same mode as previous mode)
G_BeginE	See <b>Geometry command code table</b> .	Specifies primitive type and pre-processes This command is used at execution of the CORAL extended function.
G_BeginECont	—	Specifies primitive type (vertex processing in same mode as previous mode) This command is used at execution of the CORAL extended function.
G_End	—	Ends primitive This command is used at execution of G_Begin or G_BeginCont
G_EndE	—	Ends primitive This command is used at execution of G_BeginE or G_BeginECont.
G_Vertex	—	Sets vertex parameter and draws
G_VertexLOG	—	Sets vertex parameter and draws Outputs device coordinates
G_VertexNopLOG	—	Only outputs device coordinates
G_Init	—	Initialize geometry engine
G_Viewport	—	Scale to screen coordinates (X, Y) and set origin offset
G_DepthRange	—	Scale to screen coordinates (Z) and set origin offset
G_LoadMatirix	—	Load geometric transformation matrix
G_ViewVolumeXYClip	—	Set boundary value (X, Y) of view volume clip
G_ViewVolumeZClip	—	Set boundary value (Z) of view volume clip
G_ViewVolumeWClip	—	Set boundary value (W) of view volume clip
OverlapXYOfft	See <b>Command table</b> .	Sets XY offset at shading
OverlapZOfft	See <b>Command table</b> .	Sets Z offset of shade primitive; sets Z offset of edge primitive; sets Z offset of interpolation primitive at 2D drawing with top-left non-applicable
DC_LogOutAddr	—	Sets starting address of device coordinates output
SetModeRegister	See <b>Command table</b> .	Sets drawing extended mode register
SetGModeRegister	See <b>Command table</b> .	Sets geometry extended mode register
SetColorRegister	See <b>Command table</b> .	Sets body color, shade color, and edge color
SetLVertex2i	—	Pass through high-speed 2DLine drawing register
SetLVertex2iP	—	Pass through high-speed 2DLine drawing register

**Type code table**

Type	Code
G_Nop	0010_0000
G_Begin	0010_0001
G_BeginCont	0010_0010
G_End	0010_0011
G_Vertex	0011_0000
G_VertexLOG	0011_0010
G_VertexNopLOG	0011_0011
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
SetModeRegister	1100_0000
SetGModeRegister	1100_0001
OverlapXY0fft	1100_1000
OverlapZ0fft	1100_1001
DC_LogOutAddr	1100_1100
SetColorRegister	1100_1110
G_BeginE	1110_0001
G_BeginContE	1110_0010
G_EndE	1110_0011

### Geometry command code table

(1) Integer setup type

In setup processing, “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

(2) “Unclipped” integer setup type

This command does not clip the view volume.

Only “XY” is enabled as the input parameter.

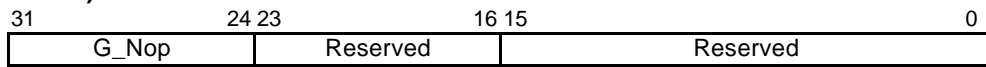
In setup processing, “XY” is calculated in the integer format.

The screen projection (GMDR0[0]=1) performed using this command is not assured.

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

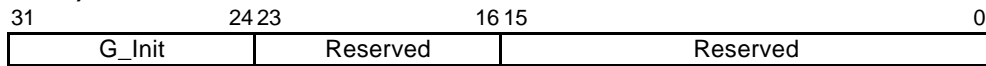
## 9.2.2 Explanation of geometry commands

### G\_Nop (Format 1)



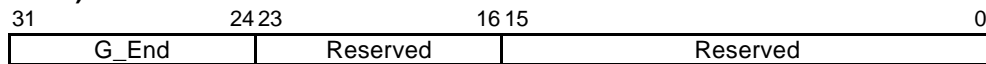
No operation

### G\_Init (Format 1)



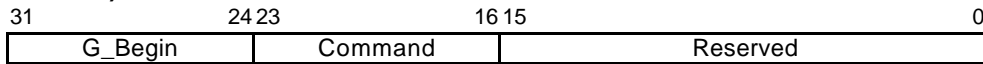
The **G\_Init** command initializes geometry engine. Execute this command before processing.

### 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\_Begin (Format 5)**



The **G\_Begin** command 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

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

Unclipped primitives (nclip\*)

(ST,Z,C)	Point	Line	Triangle	Polygon
(0,0,0)	○	○	○	○
Other than above	∕	∕	∕	∕

Primitives other than unclipped primitives

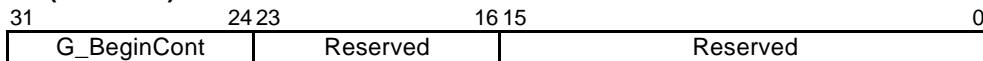
(ST,Z,C)	Point	Line	Triangle	Polygon(*2)
(0,0,0)	○	○	○	○
(0,0,1)	∕	∕	○	∕
(0,1,0)	∕ (*3)	○	○	○
(0,1,1)	∕	∕	○	∕
(1,x,x)	∕	∕	○	○ (*1)

\*1: Shading is not assured.

\*2: In case of drawing polygon with Z,ST=1, the algorithm is approximate calculation. The triangle algorithm is more accurate.

\*3: Please use a geometry lines which coordinates set to same value. And set GMDR1/GMDR1E to "End point drawn" and set MDR1 to "Z compare enable", "solid", "1 pixel line width".

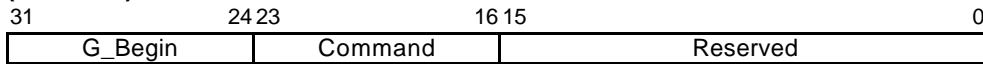
**G\_BeginCont (Format 1)**



When the primitive type set by the **G\_Begin** command the last time and drawing mode are not changed, 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\_BeginE (Format 5)**



This is the extended **G\_Begin** command.

When using the following functions, this command must be executed instead of **G\_Begin**.

- Mode register  
MDR1S/MDR1B/MDR1TL/MDR2S/MDR2TL/GMDR1E/GMDR2E
- Log output of device coordinates  
G\_VertexLOG/G\_VertexNopLOG

The **G\_BeginE** command sets types of primitive for geometry processing and drawing. Vertex setting/drawing using the above extended function is performed using the **G\_Vertex\*** command. The **G\_Vertex\*** command must be set between the **G\_BeginE** command (or the **G\_BeginECont** command) and the **G\_EndE** command.

Command:

- Points\*                      Handles primitive as point
- Lines\*                        Handles primitive as independent line  
Interpolation of the joint and broken line pattern is not supported.
- 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

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

Unclipped primitives (nclip\*)

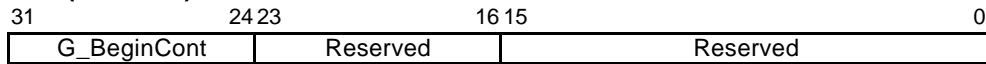
(ST,Z,C)	Point	Line	Triangle	Polygon
(0,0,0)	○	○	○	○
Other than above	∕	∕	∕	∕

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)	∕	∕	○	○ (*1)

\*1: Shading is not assured.

### G\_BeginECont (Format 1)



When the primitive type set by the **G\_BeginE** command the last time and drawing mode are not changed, the **G\_BeginECont** command is used instead of the **G\_BeginE** command. The **G\_BeginECont** command is processed faster than the **G\_BeginE** 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/G\_VertexLOG/G\_VertexNopLOG (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.int		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 matters only when a triangle (excluding a polygon) is drawn. At primitives other than triangle, set “0”.
- The use of both G\_BeginE(G\_BeginEcont) to G\_EndE, and G\_VertexLOG/NopLOG is not assured.
- G\_VertexNopLOG, except for the primitive as point is not assured.



- A vertex data is processed at every time. For example, the Coral draws interpolation of bold line joint, edging line, shadows at every vertices.

**G\_Viewport (Format 1)**

31	24 23	16 15	0
G_Viewport	Reserved	Reserved	
X_Scaling.float/fixe			
X_Offset.float/fixe			
Y_Scaling.float/fixe			
Y_Offset.float/fixe			

The **G\_Viewport** command sets the “X,Y” scale/offset value used when normalized device coordinates (NDC) is transformed into device coordinates (DC).

**G\_DepthRange (Format 1)**

31	24 23	16 15	0
G_DepthRange	Reserved	Reserved	
Z_Scaling.float/fixe			
Z_Offset.float/fixe			

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/fixe			
Matrix_a1.float/fixe			
Matrix_a2.float/fixe			
Matrix_a3.float/fixe			
Matrix_b0.float/fixe			
Matrix_b1.float/fixe			
Matrix_b2.float/fixe			
Matrix_b3.float/fixe			
Matrix_c0.float/fixe			
Matrix_c1.float/fixe			
Matrix_c2.float/fixe			
Matrix_c3.float/fixe			
Matrix_d0.float/fixe			
Matrix_d1.float/fixe			
Matrix_d2.float/fixe			
Matrix_d3.float/fixe			

The **G\_LoadMatrix** command sets the transformation matrix used when object coordinates (OC) is transformed into clip coordinates (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 coordinates 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 coordinates of the clip boundary value in view volume clipping (minimum value only).

**OverlapXYOfft (Format5)**

31	24 23	16 15	0
OverlapXYOfft		Command	Reserved
Y Offset		X Offset	

The **OverlapXYOfft** command sets the XY offset of the shade primitive relative to the body primitive at shading drawing. Shadow shape is same as body.

Command:

Command	Code	Explanation
ShadowXY	0000_0000	ShadowXY command sets the XY offset of the shade primitive relative to the body primitive.
ShadowXYcomposition	0000_0001	ShadowXYcomposition command sets the XY offset of the shade synthetic primitive relative to the body primitive. It command synthesizes a shade from the relationship between the XY offset set using ShadowXY and this XY offset. This command is enabled for only lines.

**OverlapZOfft (Format5)**

31	24 23	16 15	0
OverlapZOfft		Command	Reserved
don't care		Z Offset	

Note: When MDR0 ZP = 1, only lower 8 bits are enabled.

31	24 23	16 15	0
OverlapZOfft		Packed_ONBS	Reserved
S_Z Offset		B_Z Offset	N_Z Offset
		O_Z Offset	

The **OverlapZOfft** command sets the Z offset of the shade primitive relative to the body primitive, sets the Z-offset of the edge primitive relative to the body primitive, and sets the Z offset of the interpolation primitive relative to the body primitive, with the top-left rule non-applicable in effect.

At this time, the following relationship must be satisfied when, for example, GREATER is specified for the Z value comparison mode:

- Body primitive > Top-left rule non-applicable interpolation primitive
- > Edge primitive > Shade primitive

Command:

Command	Code	Explanation
Origin	0000_0000	Origin command sets the Z offset of the body primitive. When drawing one primitive below the other primitive (for example, when drawing a solid intersection), this Z offset is changed. When drawing an ordinary intersection, set the same Z offset as other primitives.
NonTopLeft	0000_0001	NonTopLeft command sets the Z offset of the interpolation primitive, with the top-left non-applicable.
Border	0000_0010	Border command sets the Z offset of the edge primitive.
Shadow	0000_0011	Shadow command sets the Z offset of the shade primitive.
Packed_ONBS	0000_0111	Packed_ONBS command sets the above four types of Z offsets.

**DC\_LogOutAddr (Format5)**

31	24 23	16 15	0
OverlapXYOfft	Command	Reserved	
000000	LogOutAddr		

The **DC\_LogOutAddr** command sets the starting address of the log output destination of the device coordinates.

**SetModeRegister (Format5)**

31	24 23	16 15	0
SetModeRegister	Command	Reserved	
MDR1*/MDR2*			

The **SetModeRegister** command sets the mode register for shade primitive, for edge primitive, and for top-left non-applicable primitive. At drawing of these primitives, also set the mode register (MDR1/MDR2) for the body primitive, using this packet.

Command:

Command	Code	Explanation
MDR1	0000_0000	MDR1 command sets MDR1 for the body primitive.
MDR1S	0000_0010	MDR1S command sets MDR1 for the shade primitive.
MDR1B	0000_0100	MDR1B command sets MDR1 for the edge primitive.
MDR2	0000_0001	MDR2 command sets MDR2 for the body primitive.
MDR2S	0000_0011	MDR2S command sets MDR2 for the shade primitive.
MDR2LT	0000_0111	MDR2LT command sets MDR2 for the top-left non-applicable primitive.

**SetGModeRegister (Format5)**

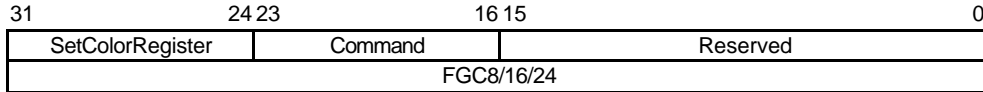
31	24 23	16 15	0
SetGModeRegister	Command	Reserved	
GMDR1E/GMDR2E			

The **SetGModeRegister** command sets the geometry extended mode register.

Command:

Command	Code	Explanation
GMDR1E	0001_0000	GMDR1E command sets GMDR1E and at the same time, updates GMDR1.
GMDR2E	0010_0000	GMDR2E command sets GMDR2E and at the same time, updates GMDR2.

**SetColorRegister (Format5)**

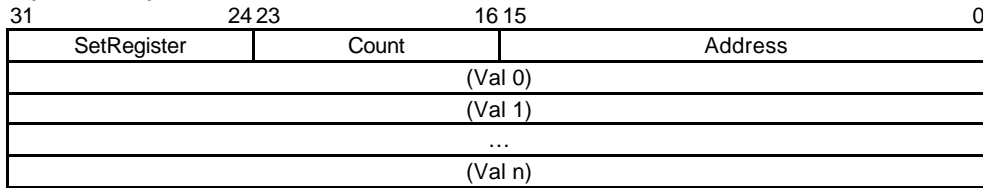


The **SetColorRegister** command sets the foreground color and background color of the body primitive, shade primitive, and edge primitive.

Commands:

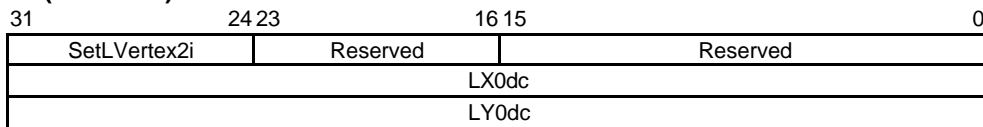
Command	Code	Explanation
ForeColor	0000_0000	ForeColor command sets the foreground color for the body primitive.
BackColor	0000_0001	BackColor command sets the background color for the body primitive.
ForeColorShadow	0000_0010	ForeColorShadow command sets the foreground color for the shade primitive.
BackColorShadow	0000_0011	BackColorShadow command sets the background color for the shade primitive.
ForeColorBorder	0000_0100	ForeColorBorder command sets the foreground color for the edge primitive.
BackColorBorder	0000_0101	BackColorBorder command sets the background color for the edge primitive.

**SetRegister (Format 2)**



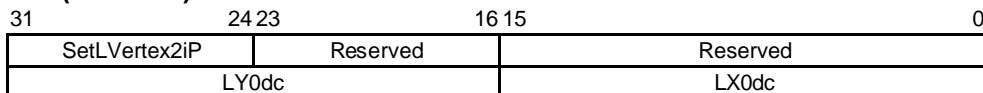
The **SetRegister** command is upper compatible with CREMSON **SetRegister**. It can specify the address of a register in the geometry engine.

**SetLVertex2i (Format 1)**



The SetLVertex2i command issues the **SetRegister\_LX0dc/LY0dc** command (MB86290A command to set starting vertex at line drawing) in the geometry FIFO interface. This performs processing faster than when the **SetRegister\_LX0dc/LY0dc** command is input directly to the geometry FIFO.

**SetLVertex2iP (Format 1)**



The **SetLVertex2iP** command supports packed XY of SetLVertex21.

## 9.3 Rendering Command

### 9.3.1 Command list

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

Type	Command	Description
Nop	—	No operation
Interrupt	—	Interrupt request to host CPU
Sync	—	Synchronization with events
SetRegister	—	Sets data to register
SetVertex2i	Normal	Sets data to high-speed 2DTriangle vertex register
	PolygonBegin	Initializes border rectangle calculation of multiple vertices random shape
Draw	PolygonEnd	Clears polygon flag after drawing polygon
	Flush_FB/Z	Flushes drawing pipelines
DrawPixel	Pixel	Draws point
DrawPixelZ	PixelZ	Draws point with Z
DrawLine	Xvector	Draws line (principal axis X)
	Yvector	Draws line (principal axis Y)
	AntiXvector	Draws line with anti-alias option (principal axis X)
	AntiYvector	Draws line with anti-alias option (principal axis Y)
DrawLine2i	ZeroVector	Draws high-speed 2DLine (with vertex 0 as starting point)
DrawLine2iP	OneVector	Draws high-speed 2DLine (with vertex 1 as starting point)
DrawTrap	TrapRight	Draws right triangle
	TrapLeft	Draws left triangle
DrawVertex2i	TriangleFan	Draws high-speed 2DTriangle
DrawVertex2iP	FlagTriangleFan	Draws high-speed 2DTriangle for multiple vertices random shape
DrawRectP	BlitFill	Draws rectangle with single color
	ClearPolyFlag	Clears polygon flag buffer
DrawBitmapP	BlitDraw	Draws Blt (16-bit)
	Bitmap	Draws binary bit map (character)
DrawBitmapLargeP	BlitDraw	Draws Blt (32-bit)
BltCopyP BltCopy-AlternateP	TopLeft	Blt transfer from top left coordinates
	TopRight	Blt transfer from top right coordinates
	BottomLeft	Blt transfer from bottom left coordinates
	BottomRight	Blt transfer from bottom right coordinates
LoadTextureP	LoadTexture	Loads texture pattern
	LoadTILE	Loads tile pattern
BltTextureP	LoadTexture	Loads texture pattern from local memory
	LoadTILE	Loads tile pattern from local memory

BltCopyAlt- AlphaBlendP	—	Alpha blending is supported (see the alpha map). BltCopyAlternateP
----------------------------	---	---

**Type Code Table**

<b>Type</b>	<b>Code</b>
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
BltCopyAltAlphaBlendP	0001_1111
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)**

<b>Command</b>	<b>Code</b>
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

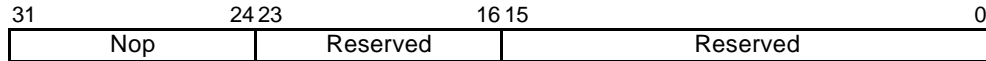
**Command Code Table (2)**

<b>Command</b>	<b>Code</b>
BitFill	010_00001
BitDraw	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

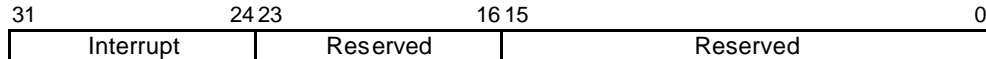
All parameters belonging to their command are stored in relevant registers. The definition of each parameter is explained in the section of each command.

#### Nop (Format1)



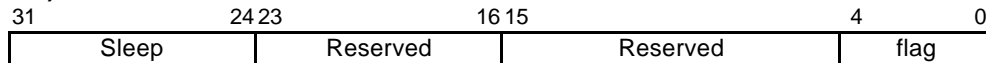
No operation

#### Interrupt (Format1)



The **Interrupt** command generates interrupt request to host CPU.

#### Sync (Format9)



The **Sync** command suspends all subsequent display list processing until event set in flag detected.

Flag:

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

- Bit 0      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)			

The **SetRegister** command sets data to sequential registers.

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

Address: Register address

Set the value of the address for **SetRegister** given in the register list.  
 When transferring two or more data, set the starting register address.

**SetVertex2i (Format8)**

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

The **SetVertex2i** command sets vertices data for high-speed 2DLine or high-speed 2DTriangle to registers.

Commands:

Normal Sets vertex data (X, Y).

PolygonBegin Starts 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)**

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

The **SetVertex2iP** command sets vertices data for high-speed 2DLine or high-speed 2DTriangle to registers.

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

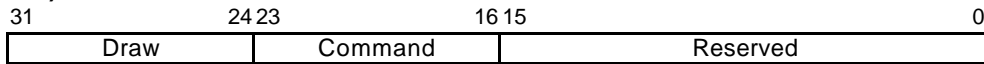
Commands:

Normal Sets vertices data.

PolygonBegin Starts 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)**

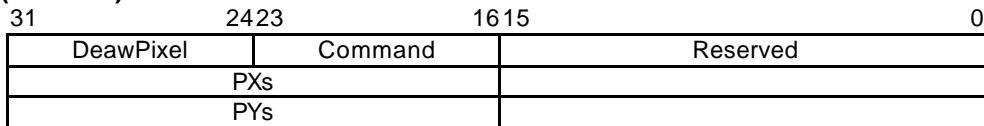


The **Draw** command executes drawing command. All parameters required for drawing command execution must be set at their appropriate registers.

Commands:

- PolygonEnd     Draws polygon end.  
 Fills random shape with color according to flags generated by **FlagTriangleFan** command and information of circumscribed rectangle generated by **PolygonBegin** command.
- Flush\_FB       Flashes drawing data in the drawing pipeline into the graphics memory.  
 Place this command at the end of the display list.
- Flush\_Z        Flashes 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)**

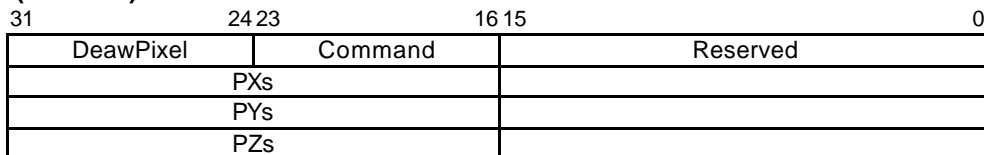


The **DrawPixel** command draws pixel.

Command:

- Pixel            Draws pixel without Z value.

**DrawPixelZ (Format5)**



The **DrawPixelZ** command draws pixel with Z value.

Command:

- PixelZ          Draws pixel with Z value.

**DrawLine (Format5)**

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

The **DrawLine** command draws line. It starts drawing after setting all parameters at line draw registers.

Commands:

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

**DrawLine2i (Format7)**

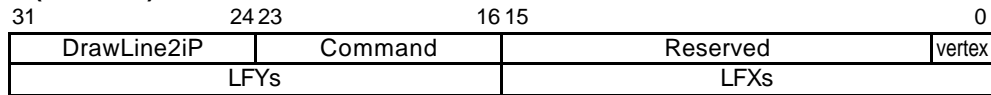
31	24 23	16 15	0
DrawLine2i	Command	Reserved	vertex
LFXs		0	
LFYs		0	

The **DrawLine2i** command draws high-speed 2DLine. It starts drawing after setting parameters at the high-speed 2DLine drawing registers. Integer data can only be used for coordinates.

Commands:

ZeroVector	Draws line from vertex 0 to vertex 1.
OneVector	Draws line from vertex 1 to vertex 0.
ZeroVectorNoEnd	Draws line from vertex 0 to vertex 1 (without drawing end point).
OneVectorNoEnd	Draws line from vertex 1 to vertex 0 (without drawing end point).
ZeroVectorBlpClear	Draws line from vertex 0 to vertex 1 (principal axis X, and prior to drawing, broken line pattern reference position cleared).
OneVectorBlpClear	Draws line from vertex 1 to vertex 0 (principal axis Y, and prior to drawing, broken line pattern reference position cleared).
ZeroVectorNoEndBlpClear	Draws line from vertex 0 to vertex 1 (principal axis X, without end point drawing and prior to drawing, broken line pattern reference position cleared).
OneVectorNoEndBlpClear	Draws line from vertex 1 to vertex 0 (principal axis Y, without end point drawing and prior to drawing, broken line pattern reference position cleared).
AntiZeroVector	Draws anti-alias line from vertex 0 to vertex 1.
AntiOneVector	Draws anti-alias line from vertex 1 to vertex 0.
AntiZeroVectorNoEnd	Draws anti-alias line from vertex 0 to vertex 1 (without end point).
AntiOneVectorNoEnd	Draws anti-alias line from vertex 1 to vertex 0 (without end point).
AntiZeroVectorBlpClear	Draws anti-alias line from vertex 0 to vertex 1 (principal axis X and prior to drawing, broken line pattern reference position cleared).
AntiOneVectorBlpClear	Draws anti-alias line from vertex 1 to vertex 0 (principal axis Y and prior to drawing, broken line pattern reference position cleared).
AntiZeroVectorNoEndBlpClear	Draws anti-alias line from vertex 0 to vertex 1 (principal axis X, without end point drawing and prior to drawing, broken line pattern reference position cleared).
AntiOneVectorNoEndBlpClear	Draws anti-alias line from vertex 1 to vertex 0 (principal axis Y, without end point drawing and prior to drawing, broken line pattern reference position cleared).

**DrawLine2iP (Format7)**



The **DrawLine2iP** command draws high-speed 2DLine. It starts drawing after setting parameters at high-speed 2DLine drawing registers. Only packed integer data can be used for coordinates.

Commands:

ZeroVector	Draws line from vertex 0 to vertex 1.
OneVector	Draws line from vertex 1 to vertex 0.
ZeroVectorNoEnd	Draws line from vertex 0 to vertex 1 (without drawing end point).
OneVectorNoEnd	Draws line from vertex 1 to vertex 0 (without drawing end point).
ZeroVectorBlpClear	Draws line from vertex 0 to vertex 1 (principal axis X, and prior to drawing, broken line pattern reference position cleared).
OneVectorBlpClear	Draws line from vertex 1 to vertex 0 (principal axis Y, and prior to drawing, broken line pattern reference position cleared).
ZeroVectorNoEndBlpClear	Draws line from vertex 0 to vertex 1 (principal axis X, without end point drawing and prior to drawing, broken line pattern reference position cleared).
OneVectorNoEndBlpClear	Draws line from vertex 1 to vertex 0 (principal axis Y, without end point drawing and prior to drawing, broken line pattern reference position cleared).
AntiZeroVector	Draws anti-alias line from vertex 0 to vertex 1.
AntiOneVector	Draws anti-alias line from vertex 1 to vertex 0.
AntiZeroVectorNoEnd	Draws anti-alias line from vertex 0 to vertex 1 (without end point).
AntiOneVectorNoEnd	Draws anti-alias line from vertex 1 to vertex 0 (without end point).
AntiZeroVectorBlpClear	Draws anti-alias line from vertex 0 to vertex 1 (principal axis X and prior to drawing, broken line pattern reference position cleared).
AntiOneVectorBlpClear	Draws anti-alias line from vertex 1 to vertex 0 (principal axis Y and prior to drawing, broken line pattern reference position cleared).
AntiZeroVectorNoEndBlpClear	Draws anti-alias line from vertex 0 to vertex 1 (principal axis X, without end point drawing and prior to drawing, broken line pattern reference position cleared).
AntiOneVectorNoEndBlpClear	Draws anti-alias line from vertex 1 to vertex 0 (principal axis Y, without end point drawing and prior to drawing, broken line pattern reference position cleared).



**DrawTrap (Format5)**

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

The **DrawTrap** command draws Triangle. It starts drawing after setting parameters at the Triangle Drawing registers (coordinates).

Commands:

- TrapRight                 Draws right triangle.
- TrapLeft                 Draws left triangle.

**DrawVertex2i (Format7)**

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

The **DrawVertex2i** command draws high-speed 2DTriangle

It starts triangle drawing after setting parameters at 2DTriangle Drawing registers.

Commands:

- TriangleFan                 Draws high-speed 2DTriangle.
- FlagTriangleFan             Draws high-speed 2DTriangle for polygon drawing in the flag buffer.

**DrawVertex2iP (Format7)**

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

The **DrawVertex2iP** command draws high-speed 2DTriangle

It starts drawing after setting parameters at 2DTriangle Drawing registers

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

Commands:

- TriangleFan                 Draw high-speed 2DTriangle.
- FlagTriangleFan             Draws high-speed 2DTriangle for polygon drawing in the flag buffer.

**DrawRectP (Format5)**

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

The **DrawRectP** command fills rectangle. The rectangle is filled with the current color after setting parameters at the rectangle registers.

Commands:

- BlitFill                      Fills rectangle with current color (single).
- ClearPolyFlag              Fills **polygon drawing** flag buffer area with 0. The size of drawing frame 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)			

The **DrawBitmapP** command draws rectangle patterns.

Commands:

- BlitDraw                      Draws rectangle of 8 bits/pixel or 16 bits/pixel.
- DrawBitmap                  Draws binary bitmap character pattern. Bit 0 is drawn in transparent or background color, and bit 1 is drawn in foreground color.

**DrawBitmapLargeP (Format11)**

31	24 23	16 15	0
DrawBitmapLargeP	Command	Reserved	
Count			
Rys		Rxs	
RsizeY		RsizeX	
(Pattern 0)			
(Pattern 1)			
...			
(Pattern n)			

The **DrawBitmapP** command draws rectangle patterns.

The parameter(count field) could be used up to 32-bit(\*1) unlike DrawBitmapP.

(\*1: The data format of counter field is signed long. Thus actually it is possible to use up to 31-bit.)

Commands:

- BlitDraw                      Draws rectangle of 8 bits/pixel or 16 bits/pixel.

**BltCopyP (Format5)**

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

The **BltCopyP** command copies rectangle pattern within drawing frame.

Commands:

- TopLeft                      Starts BitBlt transfer from top left coordinates.
- TopRight                     Starts BitBlt transfer from top right coordinates.
- BottomLeft                  Starts BitBlt transfer from bottom left coordinates.
- BottomRight                 Starts BitBlt transfer from bottom right coordinates.

**BltCopyAlternateP (Format5)**

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

The **BltCopyAlternateP** command copies rectangle between two separate drawing frames.

Command:

- TopLeft                      Starts BitBlt transfer from top left coordinates.

**BltCopyAltAlphaBlendP (Format5)**

31	24 23	16 15	0
BltCopyAlternateP	Command	Reserved	
SADDR			
SStride			
SRYs		SRXs	
BlendStride			
BlendRYs		BlendRXs	
DRYs		DRXs	
BRsizeY		BRsizeX	

The **BltCopyAltAlphaBlendP** command performs alpha blending for the source (specified using SADDR, SStride, SRXs, SRXy) and the alpha map (specified using ABR (alpha base address), BlendStride, BlendRXs, BlendRYs) and then copies the result of the alpha blending to the destination (specified using FBR (frame buffer base address), XRES (X resolution), DRXs, and DRYs).

Command:

reserved                      Set 0000\_0000 to maintain future compatibility.

# 10 REGISTER

## 10.1 Register List

### 10.1.1 Host interface register list

Base = HostBase

Offset	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													
000	DTC																																												
																	DTC																												
004	DST																DRM								DSU																				
																	DST								DRM								DNA								DAM		DBM		DW
008																	LTS								DTS																				
																	LTS																DTS												
010	LSTA																																												
																																LSTA													
018	DRQ																																												
																																DRQ													
020	IST																																												
																																IST													
024	IMASK																																												
																																IMASK													
02C																	SRST																												
																	SRST																												
038	CCF																CGE		COT																										
040	LSA																																												
																	LSA																												
044	LCO																																												
																	LCO																												
048																	LREQ																												
																	LREQ																												
05C																	RSW																												
																	RSW																												
0f0	CID																																												



### 10.1.2 Graphics memory interface register list

Base = HostBase

Offset	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
FFFC	MMR																															
		TWR			ID			TRRD		TRC		TRP		TRAS		TRCD		LOWD		RTS		SAW		ASW							CL	

### 10.1.3 Display controller register list

Base = DisplayBase

Offset	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			
000	DCE (Display Controller Enable)																DCM (Display Control Mode)																		
	DEN													L46E	L23E	L1E	LOE	CKS	DCS	SC				EEQ				SF	ESY	SYNC					
100	DCEE (Display Controller Extend Enable)																DCEM (Display Control Extend Mode)																		
	DEN											L5E	L4E	L3E	L2E	L1E	LOE	CKS	DCS	SC				EEQ	EDE	EOF	EOD	SF	ESY	SYNC					
004	HTP (H Total Pixels)																																		
008	HDB (H Display Boundary)																HDP (H Display Period)																		
00C	VSW								HSW								HSP (H Sync pulse Position)																		
010	VTR (V Total Rasters)																																		
014	VDP (V Display Period)																VSP (V Sync pulse Position)																		
018	WY (Window Y)																WX (Window X)																		
01C	WH (Window Height)																WW (Window Width)																		
020	L0M (L0 Mode)																																		
	L0C	L0S (L0 Width)																L0H (L0 Height)																	
024	L0OA (L0 Origin Address)																																		
028	L0DA (L0 Display Address)																																		
02C	L0DY (L0 Display Y)																L0DX (L0 Display X)																		
110	L0EM (L0 Extend Mode)																																		
	L0EC	L0PB																L0WP																	
114	LOWY (L0 Window Y)																LOWX (L0 Window X)																		
118	LOWH (L0 Window Height)																LOWW (L0 Window Width)																		
030	L1M (L1 Mode)																																		
	L1C	L1YC	L1CS	L1IM	L1S (L1 Width)																														
034	L1DA (L1 Display Address)																																		
120	L1EM (L1 Extend Mode)																																		
	L1EC	L1PB																																	
040	L2M (L2 Mode)																																		
	L2C	L2FLP	L2S (L2 Width)																L2H (L2 Height)																
044	L2OA0 (L2 Origin Address 0)																																		
048	L2DA0 (L2 Display Address 0)																																		
04C	L2OA1 (L2 Origin Address 1)																																		
050	L2DA1 (L2 Display Address 1)																																		
054	L2DY (L2 Display Y)																L2DX (L2 Display X)																		
130	L2EM (L2 Extend Mode)																																		
	L2EC	L2PB																L2WM																	
134	L2WY (L2 Window Y)																L2WX (L2 Window X)																		
138	L2WH (L2 Window Height)																L2WW (L2 Window Width)																		



Offset	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			
058	L3M (L3 Mode)																																		
	L3C	L3FLP																																	
05C																																			
060																																			
064																																			
068																																			
06C																																			
140	L3EM (L3 Extend Mode)																																		
	L3EC																																L3OM	L3WP	
144																																			
148																																			
070	L4M (L4 Mode)																																		
	L4C	L4FLP																																	
074																																			
078																																			
07C																																			
080																																			
084																																			
150	L4EM (L4 Extend Mode)																																		
	L4EC																																	L4OM	L4WP
154																																			
158																																			
088	L5M (L5 Mode)																																		
	L5C	L5FLP																																	
08C																																			
090																																			
094																																			
098																																			
09C																																			
160	L5EM (L5 Extend Mode)																																		
	L5EC																																		L5OM
164																																			
168																																			

Offset	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					
0A0											CPM				CUTC (Cursor Transparent Control)																						
											CUE1	CUE0					CUO1	CUO0														CUTZ	CUTC				
0A4	CUOA0 (CUrsor0 Origin Address)																																				
0A8	CUY0 (CUrsor0 Position Y)																CUX0 (CUrsor0 Position X)																				
0AC	CUOA1 (CUrsor1 Origin Address)																																				
0B0	CUY1 (CUrsor1 Position Y)																CUX1 (CUrsor1 Position X)																				
180	DLS (Display Layer Select)																																				
																	DLS5	DLS4	DLS3	DLS2	DLS1	DLS0															
184	DBGC (Display Back Ground Color)																																				
0B4	L0BLD (L0 Blend)																																				
																	LOBE	LOB	LOBI	LOBP																	L0BR
188	L1BLD (L1 Blend)																																				
																	L1BE	L1BS	L1BI	L1BP																	L1BR
18C	L2BLD (L2 Blend)																																				
																	L2BE	L2BS	L2BI	L2BP																	L2BR
190	L3BLD (L3 Blend)																																				
																	L3BE	L3BS	L3BI	L3BP																	L3BR
194	L4BLD (L4 Blend)																																				
																	L4BE	L4BS	L4BI	L4BP																	L4BR
198	L5BLD (L5 Blend)																																				
																	L5BE	L5BS	L5BI																	L5BR	

Offset	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
0BC																	L0TC (L0 Transparent Control)															
																	L0T	L0TC (L0 Transparent Color)														
0C0	L2TR (L2 Transparent Control)																L3TR (L3 Transparent Control)															
	L2Z																L3T	L3TR (L3 Transparent Color)														
1A0	L0TEC (L0 Extend Transparency Control)																															
	L0EZT																L0ETC (L0 Extend Transparent Color)															
1A4	L1TEC (L1 Transparent Extend Control)																															
	L1EZT																L1ETC (L1 Extend Transparent Color)															
1A8	L2TEC (L2 Transparent Extend Control)																															
	L2EZT																L2ETC (L2 Extend Transparent Color)															
1AC	L3TEC (L3 Transparent Extend Control)																															
	L3EZT																L3ETC (L3 Extend Transparent Color)															
1B0	L4ETC (L4 Extend Transparent Control)																															
	L4EZT																L4ETC (L4 Extend Transparent Color)															
1B4	L5ETC (L5 Extend Transparent Control)																															
	L5EZT																L5ETC (L5 Extend Transparent Color)															

Offset	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			
400	L0PAL0																																		
	A																																		
404	L0PAL1																																		
:	:																																		
7FC	L0PAL255																																		
800	L1PAL0																																		
	A																																		
804	L1PAL1																																		
:	:																																		
BFC	L1PAL255																																		
1000	L2PAL0																																		
	A																																		
1004	L2PAL1																																		
:	:																																		
13FC	L2PAL255																																		
1400	L3PAL0																																		
	A																																		
1404	L3PAL1																																		
:	:																																		
17FC	L3PAL255																																		

### 10.1.4 Drawing engine register list

The parenthesized value in the Offset field denotes the absolute address used by the **SetRegister** command.

Base = DrawBase

Offset	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
000 (000)	Ys																															
	s	s	s	s	Int								Frac																			
004 (001)	Xs																															
	s	s	s	s	Int								Frac																			
008 (002)	dXdY																															
	s	s	s	s	Int								Frac																			
00C (003)	XUs																															
	s	s	s	s	Int								Frac																			
010 (004)	dXUdy																															
	s	s	s	s	Int								Frac																			
014 (005)	XLs																															
	s	s	s	s	Int								Frac																			
018 (006)	dXLdy																															
	s	s	s	s	Int								Frac																			
01C (007)	USN																															
	0	0	0	0	Int								0																			
020 (008)	LSN																															
	0	0	0	0	Int								0																			
040 (010)	Rs																															
	0	0	0	0	0	0	0	0	Int								Frac															
044 (011)	dRdx																															
	s	s	s	s	s	s	s	s	Int								Frac															
048 (012)	dRdy																															
	s	s	s	s	s	s	s	s	Int								Frac															
04C (013)	Gs																															
	0	0	0	0	0	0	0	0	Int								Frac															
050 (014)	dGdx																															
	s	s	s	s	s	s	s	s	Int								Frac															
054 (015)	dGdy																															
	s	s	s	s	s	s	s	s	Int								Frac															
058 (016)	Bs																															
	0	0	0	0	0	0	0	0	Int								Frac															
05C (017)	dBdx																															
	s	s	s	s	s	s	s	s	Int								Frac															
060 (018)	dBdy																															
	s	s	s	s	s	s	s	s	Int								Frac															





Offset	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				
3E0	BLPO																																			
(0F8)																									BCR											
400	CTR																																			
(100)													ED	EE	CE	FCNT						NF	FF	FE	SS				DS				PS			
404	IFSR																																			
(-)																													NF	FF	FE					
408	IFCNT																																			
(-)																									FCNT											
40C	SST																																			
(-)																													SS							
410	DS																																			
(-)																									DS											
414	PST																																			
(-)																									PS											
418	EST																																			
(-)																													FD	CE						
420	MDR0																																			
(108)													ZP	CF						CY	CX	BSV				BSH										
424	MDR1/MDR1S/MDR1B																																			
(109)	LW						BP		BL	LOG						BM		ZW	ZCL	ZC																
428	MDR2/MDR2S/MDR2TL																																			
(10a)	TT														LOG						BM		ZW	ZCL	ZC	AS	SM									
42C	MDR3																																			
(10b)							BA	TAB						TBL		TWS				TWT		TF		TC												
430	MDR4																																			
(10c)													LOG						BM						TE											
43C	MDR7																																			
(10f)																									LTH		EZ	OG					PGH		PTH	PZH



Offset	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
440	FBR																															
(110)	FBASE																															
444	XRES																															
(111)																								XRES								
448	ZBR																															
(112)	ZBASE																															
44C	TBR																															
(113)	TBASE																															
450	PFBR																															
(114)	PFBASE																															
454	CXMIN																															
(115)																								CLIPXMIN								
458	CXMAX																															
(116)																								CLIPXMAX								
45C	CYMIN																															
(117)																								CLIPYMIN								
460	CYMAX																															
(118)																								CLIPYMAX								
464	TXS																															
(119)	TXSN																					TXSM										
468	TIS																															
(11a)	TISN																					TISM										
46C	TOA																															
(11b)																								XBO								
470	SHO																															
(11C)	SHOFFS																															
474	ABR																															
(11D)	ABASE																															
480	FC																															
(120)																	FGC8/16															
484	BC																															
(121)																	BGC8/16															
488	ALF																															
(122)																								A								
48C	BLP																															
(123)																																
494	TBC																															
(125)																	BC8/16															

Offset	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
540	LX0dc																															
(150)	0	0	0	0	Int												0															
544	LY0dc																															
(151)	0	0	0	0	Int												0															
548	LX1dc																															
(152)	0	0	0	0	Int												0															
54C	LY1dc																															
(153)	0	0	0	0	Int												0															
580	X0dc																															
(160)	0	0	0	0	Int												0															
584	Y0dc																															
(161)	0	0	0	0	Int												0															
588	X1dc																															
(162)	0	0	0	0	Int												0															
58C	Y1dc																															
(163)	0	0	0	0	Int												0															
590	X2dc																															
(164)	0	0	0	0	Int												0															
594	Y2dc																															
(165)	0	0	0	0	Int												0															



## 10.2 Explanation of Register

Terms appeared 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 included in register
4. R/W  
Indicates access attribute (read/write) of each field  
Each symbol shown in this section denotes the following:  
  
R0    "0" always read at read. Write access is Don't care.  
W0    Only "0" can be written.  
R     Read enabled  
W     Write enabled  
RX    Read enabled (read values undefined)  
RW    Read and write enabled  
RW0   Read and write 0 enabled
5. Initial value  
Indicates initial value of immediately before the reset of each bit field.
6. Handling of reserved bits  
"0" is recommended for the write value so that compatibility can be maintained with future products.

## 10.2.1 Host interface registers

### DTC (DMA Transfer Count)

Register address	HostBaseAddress + 00H																															
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															
Initial value	0																Don't care															

DTC is a readable/writable 32-bit register which sets the transfer count in either one long-word (32 bits) or 32 bytes units. When “1h” is set transfer is performed once. However, when “0h” is set, it indicates the maximum transfer count and 16M (16,777,216) data are transferred. During DMA transfer, the remaining transfer count is shown, therefore, the register value cannot be overwritten until DMA transfer is completed.

Note: This register need not be set in a mode in which Dual DMA ACK is not used, or the V832 mode.

### 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
Initial value	0					0	0	0

**Bit 0 DW (DMA Word)**

Specifies DMA transfer count

- 0: 1-double word (32 bits) per DMA transfer
- 1: 8-double words (32 bytes) per DMA transfer (only SH4)

**Bit 1 DBM (DMA Bus request Mode)**

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

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

**Bit 2 DAM (DMA Address Mode)**

Selects DMA address mode in issuing external request

- 0: Dual address mode
- 1: Single address mode (SH4 only)

**Bit 3 DNA (Dual address No Ack mode)**

This bit is selected when using the dual-address-mode DMA that does not use the ACK signal.

- 0: Uses dual-address-mode DMA that uses ordinary ACK signal
- 1: Uses dual-address-mode DMA that does not use ACK signal  
 Detection of the DREQ edge is supported; DREQ is negated per transfer. When data cannot be received irrespective of the Bit1 setting, DREQ continues being negated.

**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
Initial value	0							0

This register enables the DMA request. Setting “1” to this register to temporarily stop the DMA request from the CORAL. The external request is enabled by setting “0” to 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
Initial value	0							0

This register indicates the DMA transfer 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
Initial value	0							0

This register suspends DMA transfer.

An ongoing DMA transfer is suspended by setting DTS to “1”.

In the dual-address without ACK mode, to end the DMA transfer, write “1” to this register after CPU DMA transfer.

**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
Initial value	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
Initial value	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
Initial value	0							0

This register starts sending external DMA request.

DMA transfer using the external request handshake is triggered by setting DRQ to “1”. The external DREQ signal cannot be issued when DMA is masked by the DRM register. This register cannot be written “0”. When DMA transfer is completed, this status is cleared 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																Resv	Reserved										IST	IST			
R/W	R0																R0W0	R0										RW0	RW0			
Initial value	0																0	0										0	0			

This register indicates the current interrupt status. It shows that an interrupt request is issued when “1” is set to this register. The interrupt status is cleared by writing “0” to 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.)  
Indicates vertical interrupt synchronization
- Bit 3            FSYNC (Frame Sync.)  
Indicates frame synchronization interrupt
- Bit 4            SYNCERR (Sync. Error)  
Indicates external synchronization error interrupt
- Bit 17 and 16   Reserved  
This field is provided for testing.  
Normally, the read value is “0”, but note that it may be “1” when a drawing command error (Bit 0) has occurred.

### IMASK (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																Resv	Reserved																IMASK	IMASK
R/W	R0																ROW0	R0																RW	RW
Initial value	0																0	0																0	0

This register masks interrupt requests. Even when the interrupt request is issued for the bit to which "0" is written, interrupt signal is not asserted for CPU.

- Bit 0      CERRM (Command Error Interrupt Mask)  
 Masks drawing command execution error interrupt
- Bit 1      CENDM (Command Interrupt Mask)  
 Masks drawing command end interrupt
- Bit 2      VSYNCM (Vertical Sync. Interrupt Mask)  
 Masks vertical synchronization interrupt
- Bit 3      FSYNCH (Frame Sync. Interrupt Mask)  
 Masks frame synchronization interrupt
- Bit 4      SYNCERRM (Sync Error Mask)  
 Masks external synchronization error interrupt

### 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
Initial value	0							0

This register controls software reset. When "1" is set to this register, a software reset is performed.



**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
Initial value	0																Don't care																0

This register sets the DisplayList transfer source address. When DisplayList is transferred from Graphics Memory, set the transfer start address of DisplayList stored in Graphics Memory. Since the lower two bits of this register are always treated as “0”, DisplayList must be 4byte aligned. The values set at this register do not change during or after transfer.

**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															
Initial value	0																Don't care															

This register sets the DisplayList transfer count. Set the display list transfer count by the long word. When “1h” is set, 1-word data is transferred. When “0” is set, it is considered to be the maximum count and 16M (16,777,216) words of data are transferred. The values set at this register do not change during or after transfer.

**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
Initial value	0							0

This register triggers DisplayList transfer from the Graphics Memory. Transfer is started by setting LREQ to “1”. 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 disabled during transfer.

**RSW (Register location Switch)**

Register address	HostBaseAddress + 5C <sub>H</sub>							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved							RSW
R/W	R0							RW
Initial value	0							0

In SH3 or SH4 mode, set this register when moving the register area from the center (1FC0000) to the end of the CORAL area (3FC0000). This move can be performed when “1” is written to this register.

Set this register at the first access after reset. Access CORAL after about 20 bus clocks after setting the register.

**CID (Chip ID register)**

Register address	HostBaseAddress + f0 <sub>H</sub>																															
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																CN						VER									
R/W	R0																R						R									
Initial value	0																0000_0011						0									

This is the chip identification register.

Bit 7 to 0 VER (VERsion)

This field indicates the chip’s unique version number. Note that the unique version number for the ES version and that of the mass-produced version are different.

- 0000\_0000 ES
- 0000\_0001 Reserved
- 0000\_0010 Reserved for LQ
- others Reserved

Bit 15 to 8 CN (Chip Name)

This field indicates the chip name.

- 0000\_0000 Reserved
- 0000\_0001 Reserved
- 0000\_0010 Reserved
- 0000\_0011 CORAL
- others Reserved

**CCF (Change of Clock Frequency)**

Register address	HostBaseAddress + 38 <sub>H</sub>																															
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												CGE	COT	Reserved																	
R/W	RW0												RW	RW	RW0																	
Initial value	0												00	00	0																	

This register changes the operating frequency.

Bit 19 and 18 CGE (Clock select for Geometry Engine)  
 Selects the clock for the geometry engine

11	Reserved
10	166 MHz
01	133 MHz
00	100 MHz

Bit 17 and 16 COT (Clock select for the others except-geometry engine)  
 Selects the clock for other than the geometry engine

11	Reserved
10	Reserved
01	133 MHz
00	100 MHz

Notes:

1. Write "0" to the bit field other than the above ([31:20], [15:00]).
2. Operation is not assured when the clock setting relationship is CGE < COT.

## 10.2.2 Graphics memory interface registers

### MMR (Memory I/F Mode Register)

Register address	HostBaseAddress + FFFC <sub>H</sub>																															
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	*1	RW	Reserved	*1	*1	TRRD	TRC	TRP	TRAS	TRCD	LOWD	RTS	RAW	ASW	CL																	
R/W	RW	RW	R	R1 W0	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW																	
Initial value	0	0	Don't care	1	0	00	0000	00	000	00	00	000	000	0	000																	

\*1: Reserved

This register sets the mode of the graphics memory interface. A value must be written to this register after a reset. (When default setting is performed, a value must also be written to this register.) Only write once to this register; do not change the written value during operation.

This register is not initialized at a software reset.

Bit 2 to 0 CL (CAS Latency)  
 Sets the CAS latency. Write the same value as this field, to the mode register for SDRAM

011	CL3
010	CL2
Other than the above	Setting disabled

Bit 3 ASW (Attached SDRAM bit Width)  
 Sets the bit width of the data bus (memory bus width mode)

1	64 bit
0	32 bit

Bit 6 to 4 SAW (SDRAM Address Width)  
 Sets the bit width of the SDRAM address

001	15 bit BANK 2 bit ROW 13 bit COL 9 bit SDRAM
111	14 bit BANK 2 bit ROW 12 bit COL 9 bit SDRAM
110	14 bit BANK 2 bit ROW 12 bit COL 8 bit SDRAM
101	13 bit BANK 2 bit ROW 11 bit COL 8 bit SDRAM
100	12 bit BANK 1 bit ROW 11 bit COL 8 bit FCRAM
000	14 bit BANK 2 bit ROW 12 bit COL 8 bit SDRAM
Other than the above	Setting disabled

Bit 9 to 7 RTS (Refresh Timing Setting)  
 Sets the refresh interval

000	Refresh is performed every 384 internal clocks.
111	Refresh is performed every 1552 internal clocks.
001 to 110	Refresh is performed every '64 × n' internal clocks in the 64 to 384 range.

Bit 11 and 10	LOWD	
		Sets the count of clocks secured for the period from the instant the ending data is output to the instant the write command is issued.
	10	2 clocks
	00	2 clocks
	Other than the above	Setting disabled
Bit 13 and 12	TRCD	
		Sets the wait time secured from the bank active to CAS. The clock count is used to express the wait time.
	11	3 clocks
	10	2 clocks
	01	1 clock
	00	0 clock
Bit 16 to 14	TRAS	
		Sets the minimum time for 1 bank active. The clock count is used to express the minimum time.
	111	7 clocks
	110	6 clocks
	101	5 clocks
	100	4 clocks
	011	3 clocks
	010	2 clocks
	Other than the above	Setting disabled
Bit 18 and 17	TRP	
		Sets the wait time secured from the pre-charge to the bank active. The clock count is used to express the wait time.
	11	3 clocks
	10	2 clocks
	01	1 clock
Bit 22 to 19	TRC	
		This field sets the wait time secured from the refresh to the bank active. The clock count is used to express the wait time.
	1010	10 clocks
	1001	9 clocks
	1000	8 clocks
	0111	7 clocks
	0110	6 clocks
	0101	5 clocks
	0100	4 clocks

	0011	3 clocks
	Other than the above	Setting disabled
Bit 24 and 23	TRRD	
		Sets the wait time secured from the bank active to the next bank active. The clock count is used to express the wait time.
	11	3 clocks
	10	2 clocks
Bit 26	Reserved	
		Always write "0" at write. "1" is always read at read.
Bit 30	TWR	
		Sets the write recovery time (the time from the write command to the read or to the pre-charge command).
	1	2 clocks
	0	1 clock

### 10.2.3 Display control register

#### DCM (Display Control Mode) / DCEM (Display Control Extend Mode)

Register address	DisplayBaseAddress + 00 <sub>H</sub> (DisplayBaseAddress + 100 <sub>H</sub> )															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	CKS	Reserved	SC						EEQ	ODE	Reserved	Reserved	SF	ESY	SYNC	
R/W	RW	RW0	RW						RW	RW	RX	RX	RW	RW	RW	
Initial value	0	0	01110 (DCM)						0	X			0	1	00	
			11101 (DCEM)													

This register controls the display count mode. It is not initialized by a software reset. This register is mapped to two addresses. The difference between the two registers is the format of the frequency division rate setting (SC).

Bit 1 to 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:    External synchronization disabled  
 1:    External synchronization enabled

Bit 3            SF (Synchronize signal format)  
 Sets format of synchronization (VSYNC, HSYNC) signals  
 0:    Negative logic  
 1:    Positive logic

Bit 7            EEQ (Enable Equalizing pulse)  
 Sets CCYNC signal mode  
 0:    Does not insert equalizing pulse into CCYNC signal  
 1:    Inserts equalizing pulse into CCYNC signal

Bit 13 to 8    SC (Scaling)  
 Divides display reference clock by the preset ratio to generate dot clock

Offset = 0		Offset = 100 <sub>H</sub>	
x00000	Frequency not divided	000000	Frequency not divided
x00001	Frequency division rate = 1/4	000001	Frequency division rate = 1/2
x00010	Frequency division rate = 1/6	000010	Frequency division rate = 1/3
X00011	Frequency division rate = 1/8	000011	Frequency division rate = 1/4
:		:	
x11111	Frequency division rate = 1/64	111111	Frequency division rate = 1/64

When n is set, with Offset = 0, the frequency division rate is  $1/(2n + 2)$ .

When m is set, with Offset = 100h, the frequency division rate is  $1/(m + 1)$ .

Basically, these are setting parameters with the same function ( $2n + 2 = m + 1$ ).  
Because of this,  $m = 2n + 1$  is established. When n is set to the SC field with Offset = 0,  
 $2n + 1$  is reflected with Offset = 100h.

Also, when PLL is selected as the reference clock, frequency division rates 1/1 to 1/5 are  
non-functional even when set; other frequency division rates are assigned.

- Bit 15      CKS (Clock Source)  
Selects reference clock  
0:    Internal PLL output clock  
1:    DCLKI input



**DCE (Display Controller Enable)**

Register address	DisplayBaseAddress + 02 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DEN	Reserved											L45E	L23E	L1E	L0E
R/W	RW	R0											RW	RW	RW	RW
Initial value	0	0											0	0	0	0

This register controls enabling the video signal output and display of each layer. Layer enabling is specified in four-layer units to maintain backward compatibility with previous products.

- Bit 0      L0E (L0 layer Enable)

Enables display of the L0 layer. The L0 layer corresponds to the C layer for previous products.

0:      Does not display L0 layer

1:      Displays L0 layer
  
- Bit 1      L1E (L1 layer Enable)

Enables display of the L1 layer. The L1 layer corresponds to the W layer for previous products.

0:      Does not display L1 layer

1:      Displays L1 layer
  
- Bit 2      L23E (L2 & L3 layer Enable)

Enables simultaneous display of the L2 and L3 layers. These layers correspond to the M layer for previous products.

0:      Does not display L2 and L3 layer

1:      Displays L2 and L3 layer
  
- Bit 3      L45E (L4 & L5 layer Enable)

Enables simultaneous display of the L4 and L5 layers. These layers correspond to the B layer for previous products.

0:      Does not display L4 and L5 layer

1:      Displays L4 and L5 layer
  
- Bit 15     DEN (Display Enable)

Enables display

0:      Does not output display signal

1:      Outputs display signal

**DCEE (Display Controller Extend Enable)**

Register address	DisplayBaseAddress + 102 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DEN	Reserved									L5E	L4E	L3E	L2E	L1E	L0E
R/W	RW	R0									RW	RW	RW	RW	RW	RW
Initial value	0	0									0	0	0	0	0	0

This register controls enabling the video signal output and display of each layer. This register has the same function as DCE.

- Bit 0      L0E (L0 layer Enable)  
 Enables L0 layer display  
 0:      Does not display L0 layer  
 1:      Displays L0 layer
  
- Bit 1      L1E (L1 layer Enable)  
 Enables L1 layer display  
 0:      Does not display L1 layer  
 1:      Displays L1 layer
  
- Bit 2      L2E (L2 layer Enable)  
 Enables L2 layer display  
 0:      Does not display L2 layer  
 1:      Displays L2 layer
  
- Bit 3      L3E (L3 layer Enable)  
 Enables L3 layer display  
 0:      Does not display L3 layer  
 1:      Displays L3 layer
  
- Bit 4      L4E (L4 layer Enable)  
 Enables L4 layer display  
 0:      Does not display L4 layer  
 1:      Displays L4 layer
  
- Bit 5      L5E (L5 layer Enable)  
 Enables L5 layer display  
 0:      Does not display L5 layer  
 1:      Displays L5 layer
  
- Bit 15     DEN (Display Enable)  
 Enables display  
 0:      Does not output display signal  
 1:      Outputs display signal

### HTP (Horizontal Total Pixels)

Register address	DisplayBaseAddress + 06 <sub>H</sub>															
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							
Initial value	0								Don't care							

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

### HDP (Horizontal Display Period)

Register address	DisplayBaseAddress + 08 <sub>H</sub>															
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							
Initial value	0								Don't care							

This register controls the total horizontal display period in unit of pixel clocks. Setting value + 1 is the pixel count for the display period.

### HDB (Horizontal Display Boundary)

Register address	DisplayBaseAddress + 0A <sub>H</sub>															
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							
Initial value	0								Don't care							

This register controls the display period of the left part of the window in unit of pixel clocks. Setting value + 1 is the pixel count for the display period of the left part of the window. When the window is not divided into right and left before display, set the same value as HDP.

### HSP (Horizontal Synchronize pulse Position)

Register address	DisplayBaseAddress + 0C <sub>H</sub>															
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							
Initial value	0								Don't care							

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

### HSW (Horizontal Synchronize pulse Width)

Register address	DisplayBaseAddress + 0E <sub>H</sub>							
Bit number	7	6	5	4	3	2	1	0
Bit field name	HSW							
R/W	RW							
Initial value	Don't care							

This register controls the pulse width of the horizontal synchronization signal in unit of pixel clocks. Setting value + 1 is the pulse width clock count.

**VSW (Vertical Synchronize pulse Width)**

Register address	DisplayBaseAddress + 0F <sub>H</sub>															
Bit number	7	6	5	4	3	2	1	0								
Bit field name	Reserved								VSW							
R/W	R0								RW							
Initial value	0								Don't care							

This register controls the pulse width of vertical synchronization signal in unit of raster. Setting value + 1 is the pulse width raster count.

**VTR (Vertical Total Rasters)**

Register address	DisplayBaseAddress + 12 <sub>H</sub>															
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							
Initial value	0								Don't care							

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

**VSP (Vertical Synchronize pulse Position)**

Register address	DisplayBaseAddress + 14 <sub>H</sub>															
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							
Initial value	0								Don't care							

This register controls the pulse position of vertical synchronization signal in unit of raster. The vertical synchronization pulse is asserted starting at the setting value + 1st raster relative to the display start raster.

**VDP (Vertical Display Period)**

Register address	DisplayBaseAddress + 16 <sub>H</sub>															
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							
Initial value	0								Don't care							

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

**LOM (L0 layer Mode)**

Register address	DisplayBaseAddress + 20 <sub>H</sub>																															
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	L0C	Reserved		Reserved				LOW				Reserved				CH																
R/W	RW	R0		R0				RW				R0				RW																
Initial value	0	0		0				Don't care				0				Don't care																

Bit 11 to 0      L0H (L0 layer Height)  
 Specifies the height of the logic frame of the L0 layer in pixel units. Setting value + 1 is the height

Bit 23 to 16    LOW (L0 layer memory Width)  
 Sets the memory width (stride) of the logic frame of the L0 layer in 64-byte units

Bit 31          L0C (L0 layer Color mode)  
 Sets the color mode for L0 layer  
 0      Indirect color (8 bits/pixel) mode  
 1      Direct color (16 bits/pixel) mode

**LOEM (L0-layer Extended Mode)**

Register address	DisplayBaseAddress + 110 <sub>H</sub>																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	-----	4	3	2	1	0
Bit field name	L0EC		Reserved				L0PB				Reserved										L0WP							
R/W	RW		R0				RW				R0										RW							
Initial value			0								0										0							

Bit 0            L0 WP (L0 layer Window Position enable)  
 Selects the display position of L0 layer  
 0      Compatibility mode display (C layer supported)  
 1      Window display

Bit 23 to 20    L0PB (L0 layer Palette Base)  
 Shows the value added to the index when subtracting palette of L0 layer. 16 times of setting value is added.

Bit 31 and 30   L0EC (L0 layer Extended Color mode)  
 Sets extended color mode for L0 layer  
 00    Mode determined by L0C  
 01    Direct color (24 bits/pixel) mode  
 1x    Reserved

**LOOA (L0 layer Origin Address)**

Register address	DisplayBaseAddress + 24 <sub>H</sub>																															
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				LOOA																											
R/W	R0				RW																								R0			
Initial value	0				Don't care																								0000			

This register sets the origin address of the logic frame of the L0 layer. Since lower 4 bits are fixed at "0", address 16-byte-aligned.

**LODA (L0-layer Display Address)**

Register address	DisplayBaseAddress + 28 <sub>H</sub>																															
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				LODA																											
R/W	R0				RW																											
Initial value	0				Don't care																											

This register sets the display origin address of the L0 layer. For the direct color mode (16 bits/pixel), the lower 1 bit is "0", and this address is treated as being aligned in 2 bytes.

**LODX (L0-layer Display position X)**

Register address	DisplayBaseAddress + 2C <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LODX											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the display starting position (X coordinates) of the L0 layer on the basis of the origin of the logic frame in pixels.

**LODY (L0-layer Display position Y)**

Register address	DisplayBaseAddress + 2E <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LODY											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the display starting position (Y coordinates) of the L0 layer on the basis of the origin of the logic frame in pixels.

**LOWX (L0 layer Window position X)**

Register address	DisplayBaseAddress + 114 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LOWX											
R/W	R0				RW											
Initial value	0															

This register sets the X coordinates of the display position of the L0 layer window.

**LOWY (L0 layer Window position Y)**

Register address	DisplayBaseAddress + 116 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LOWY											
R/W	R0				RW											
Initial value	0															

This register sets the Y coordinates of the display position of the L0 layer window.

**LOWW (L0 layer Window Width)**

Register address	DisplayBaseAddress + 118 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LOWW											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the horizontal direction display size (width) of the L0 layer window. Do not specify "0".

**LOWH (L0 layer Window Height)**

Register address	DisplayBaseAddress + 11A <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				LOWH											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the vertical direction display size (height) of the L0 layer window. Setting value + 1 is the height.

**L1M (L1-layer Mode)**

Register address	DisplayBaseAddress + 30 <sub>H</sub>																												
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	---	5	4	3	2	1	0
Bit field name	L1C	L1YC	L1CS	L1IM	Reserved				L1W								Reserved												
R/W	RW	RW	RW	RW	R0				RW								R0												
Initial value	0	0	0	0	0				Don't Care								0												

- Bit 23 to 16    L1W (L1 layer memory Width)  
 Sets the memory width (stride) of the logic frame of the L layer in unit of 64 bytes
- Bit 28        L1IM (L1 layer Interface Mode)  
 Sets video capture mode when L1CS 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        L1CS (L1 layer Capture Synchronize)  
 Sets whether the layer is used as normal display layer or as video capture  
 0:    Normal mode  
 1:    Capture mode
- Bit 30        L1YC (L1 layer YC mode)  
 Sets color format of L1 layer  
 The YC mode must be set for video capture.  
 0:    RGB mode  
 1:    YC mode
- Bit 31        L1C (L1 layer Color mode)  
 Sets color mode for L1 layer  
 0:    Indirect color (8 bits/pixel) mode  
 1:    Direct color (16 bits/pixel) mode



**L1EM (L1 layer Extended Mode)**

Register address	DisplayBaseAddress + 120 <sub>H</sub>																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	---	4	3	2	1	0
Bit field name	L1EC			Reserved				L1PB			Reserved																	
R/W	RW			R0				RW			R0																	
Initial value	0			0				0			0																	

Bit 23 to 20 L1PB (L1 layer Palette Base)  
 Shows the value added to the index when subtracting palette of L1 layer. 16 times of setting value is added.

Bit 31 to 30 L1EC (L1 layer Extended Color mode)  
 Sets extended color mode for L1 layer  
 00 Mode determined by L0C  
 01 Direct color (24 bits/pixel) mode  
 1x Reserved

**L1DA (L1 layer Display Address)**

Register address	DisplayBaseAddress + 34 <sub>H</sub>																															
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								LODA																							
R/W	R0								RW																							
Initial value	0								Don't care																							

This register sets the display origin address of the L1 layer. For the direct color mode (16 bits/pixel), the lower 1 bit is "0", and this register is treated as being aligned in 2 bytes. Wraparound processing is not performed for the L1 layer, so the frame origin linear address and display position (X coordinates, and Y coordinates) are not specified.

**L1WX (L1 layer Window position X)**

Register address	DisplayBaseAddress + 124 <sub>H</sub> (DisplayBaseAddress + 18 <sub>H</sub> )															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								L1WX							
R/W	R0								RW							
Initial value	0								Don't care							

This register sets the X coordinates of the display position of the L1 layer window. This register is placed in two address spaces. The parenthesized address is the register address to maintain compatibility with previous products. The same applies to L1WY, L1WW, and L1WH.

**L1WY (L1 layer Window position Y)**

Register address	DisplayBaseAddress + 126 <sub>H</sub> (DisplayBaseAddress + 1A <sub>H</sub> )															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved								L1WY							
R/W	R0								RW							
Initial value	0								Don't care							

This register sets the Y coordinates of the display position of the L1 layer window.

**L1WW (L1 layer Window Width)**

Register address	DisplayBaseAddress + 128 <sub>H</sub> (DisplayBaseAddress + 1C <sub>H</sub> )															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L1WW											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the horizontal direction display size (width) of the L1 layer window. Do not specify "0".

**L1WH (L1 layer Window Height)**

Register address	DisplayBaseAddress + 12A <sub>H</sub> ((DisplayBaseAddress + 1E <sub>H</sub> ))															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L1WH											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the vertical direction display size (height) of the L1 layer window. Setting value + 1 is the height.

**L2M (L2 layer Mode)**

Register address	DisplayBaseAddress + 40 <sub>H</sub>																															
Bit number	31	30	29	28	27	--	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	L2C	L2FLP	Reserved				L2W					Reserved			L2H																	
R/W	RW	RW	R0				RW					R0			RW																	
Initial value	0	00	0				Don't care					0			Don't care																	

Bit 11 to 0      L2H (L2 layer Height)  
 Specifies the height of the logic frame of the L2 layer in pixel units. Setting value + 1 is the height

Bit 23 to 16      L2W (L2 layer memory Width)  
 Sets the memory width (stride) of the logic frame of the L2 layer in 64-byte units

Bit 30 and 29      L2FLP (L2 layer Flip mode)  
 Sets flipping mode for L2 layer

- 00    Displays frame 0
- 01    Displays frame 1
- 10    Switches frame 0 and 1 alternately for display
- 11    Reserved

Bit 31              L2C (L2 layer Color mode)  
 Sets the color mode for L2 layer

- 0      Indirect color (8 bits/pixel) mode
- 1      Direct color (16 bits/pixel) mode

**L2EM (L2 layer Extended Mode)**

Register address	DisplayBaseAddress + 130H																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	-----	4	3	2	1	0
Bit field name	L2EC		Reserved						L2PB				Reserved										L2OM	L0WP				
R/W	RW		R0						RW				R0										RW	RW				
Initial value	00		0						0				0											0				

- Bit 0            L2 WP (L2 layer Window Position enable)  
 Selects the display position of L2 layer  
 0    Compatibility mode display (ML layer supported)  
 1    Window display
  
- Bit 1            L2OM (L2 layer Overlay Mode)  
 Selects the overlay mode for L2 layer  
 0    Compatibility mode  
 1    Extended mode
  
- Bit 23 to 20    L2PB (L2 layer Palette Base)  
 Shows the value added to the index when subtracting palette of L2 layer. 16 times of setting value is added.
  
- Bit 31 and 30   L2EC (L2 layer Extended Color mode)  
 Sets extended color mode for L2 layer  
 00   Mode determined by L2C  
 01   Direct color (24 bits/pixel) mode  
 1x   Reserved

**L2OA0 (L2 layer Origin Address 0)**

Register address	DisplayBaseAddress + 44 <sub>H</sub>																															
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				L2OA0																											
R/W	R0				RW													R0														
Initial value	0				Don't care													0000														

This register sets the origin address of the logic frame of the L2 layer in frame 0. Since lower 4 bits are fixed to “0”, this address is 16-byte aligned.

**L2DA0 (L2 layer Display Address 0)**

Register address	DisplayBaseAddress + 48 <sub>H</sub>																															
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				L2DA0																											
R/W	R0				RW																											
Initial value	0				Don't care																											

This register sets the origin address of the L2 layer in frame 0. For the direct color mode (16 bits/pixel), the lower 1 bit is “0” and this address is 2-byte aligned.

**L2OA1 (L2 layer Origin Address 1)**

Register address	DisplayBaseAddress + 4C <sub>H</sub>																															
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				L2OA1																											
R/W	R0				RW													R0														
Initial value	0				Don't care													0000														

This register sets the origin address of the logic frame of the L2 layer in frame 1. Since lower 4-bits are fixed to “0”, this address is 16-byte aligned.

**L2DA1 (L2 layer Display Address 1)**

Register address	DisplayBaseAddress + 50 <sub>H</sub>																															
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				L2DA1																											
R/W	R0				RW																											
Initial value	0				Don't care																											

This register sets the origin address of the L2 layer in frame 1. For the direct color mode (16 bits/pixel), the lower 1 bit is “0” and this address is 2-byte aligned.

**L2DX (L2 layer Display position X)**

Register address	DisplayBaseAddress + 54 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2DX											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the display starting position (X coordinates) of the L2 layer on the basis of the origin of the logic frame in pixels.

**L2DY (L2 layer Display position Y)**

Register address	DisplayBaseAddress + 56 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2DY											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the display starting position (Y coordinates) of the L2 layer on the basis of the origin of the logic frame in pixels.

**L2WX (L2 layer Window position X)**

Register address	DisplayBaseAddress + 134 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2WX											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the X coordinates of the display position of the L2 layer window.

**L2WY (L2 layer Window position Y)**

Register address	DisplayBaseAddress + 136 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2WY											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the Y coordinates of the display position of the L2 layer window.

**L2WW (L2 layer Window Width)**

Register address	DisplayBaseAddress + 138 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2WW											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the horizontal direction display size (width) of the L2 layer window. Do not specify "0".

**L2WH (L2 layer Window Height)**

Register address	DisplayBaseAddress + 13A <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L2WH											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the vertical direction display size (height) of the L2 layer window. Setting value + 1 is the height.

**L3M (L3 layer Mode)**

Register address	DisplayBaseAddress + 58 <sub>H</sub>																															
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	L3C	L3FLP		Reserved								L3W				Reserved				L3H												
R/W	RW	R0		R0								RW				R0				RW												
Initial value	0	0		0								Don't care				0				Don't care												

Bit 11 to 0      L3H (L3 layer Height)  
 Specifies the height of the logic frame of the L3 layer in pixel units. Setting value + 1 is the height

Bit 23 to 16      L3W (L3 layer memory Width)  
 Sets the memory width (stride) of the logic frame of the L3 layer in 64-byte units

Bit 30 and 29      L3FLP (L3 layer Flip mode)  
 Sets flipping mode for L3 layer

- 00    Displays frame 0
- 01    Displays frame 1
- 10    Switches frame 0 and 1 alternately for display
- 11    Reserved

Bit 31              L3C (L3 layer Color mode)  
 Sets the color mode for L3 layer

- 0    Indirect color (8 bits/pixel) mode
- 1    Direct color (16 bits/pixel) mode

**L3EM (L3 layer Extended Mode)**

Register address	DisplayBaseAddress + 140 <sub>H</sub>																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	---	4	3	2	1	0
Bit field name	L3EC		Reserved						L3PB			Reserved										L3OM	L3WP					
R/W	RW		R0						RW			R0										RW	RW					
Initial value	00		0						0			0										0	0					

- Bit 0            L3 WP (L3 layer Window Position enable)  
 Selects the display position of L3 layer  
 0    Compatibility mode display (MR layer supported)  
 1    Window display
- Bit 1            L3OM (L3 layer Overlay Mode)  
 Selects the overlay mode for L3 layer  
 0    Compatibility mode  
 1    Extended mode
- Bit 23 to 20    L3PB (L3 layer Palette Base)  
 Shows the value added to the index when subtracting palette of L3 layer. 16 times of setting value is added.
- Bit 31 and 30   L3EC (L3 layer Extended Color mode)  
 Sets extended color mode for L3 layer  
 00   Mode determined by L3C  
 01   Direct color (24 bits/pixel) mode  
 1x   Reserved



**L3OA0 (L3 layer Origin Address 0)**

Register address	DisplayBaseAddress + 5C <sub>H</sub>																															
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				L3OA0																											
R/W	R0				RW												R0															
Initial value	0				Don't care												0000															

This register sets the origin address of the logic frame of the L3 layer in frame 0. Since lower 4 bits are fixed to “0”, this address is 16-byte aligned.

**L3DA0 (L3 layer Display Address 0)**

Register address	DisplayBaseAddress + 60 <sub>H</sub>																															
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				L3DA0																											
R/W	R0				RW												R0															
Initial value	0				Don't care												0000															

This register sets the origin address of the L3 layer in frame 0. For the direct color mode (16 bits/pixel), the lower 1 bit is “0” and this address is 2-byte aligned.

**L3OA1 (L3 layer Origin Address 1)**

Register address	DisplayBaseAddress + 64 <sub>H</sub>																															
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				L3OA1																											
R/W	R0				RW												R0															
Initial value	0				Don't care												0000															

This register sets the origin address of the logic frame of the L3 layer in frame 1. Since lower 4-bits are fixed to “0”, this address is 16-byte aligned.

**L3DA1 (L3 layer Display Address 1)**

Register address	DisplayBaseAddress + 68 <sub>H</sub>																															
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				L3DA1																											
R/W	R0				RW												R0															
Initial value	0				Don't care												0000															

This register sets the origin address of the L3 layer in frame 1. For the direct color mode (16 bits/pixel), the lower 1 bit is “0” and this address is 2-byte aligned.

**L3DX (L3 layer Display position X)**

Register address	DisplayBaseAddress + 6C <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3DX											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the display starting position (X coordinates) of the L3 layer on the basis of the origin of the logic frame in pixels.

**L3DY (L3 layer Display position Y)**

Register address	DisplayBaseAddress + 6E <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3DY											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the display starting position (Y coordinates) of the L3 layer on the basis of the origin of the logic frame in pixels.

**L3WX (L3 layer Window position X)**

Register address	DisplayBaseAddress + 144 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3WX											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the X coordinates of the display position of the L3 layer window.

**L3WY (L3 layer Window position Y)**

Register address	DisplayBaseAddress + 146 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3WY											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the Y coordinates of the display position of the L3 layer window.

**L3WW (L3 layer Window Width)**

Register address	DisplayBaseAddress + 148 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3WW											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the horizontal direction display size (width) of the L3 layer window. Do not specify "0".

**L3WH (L3-layer Window Height)**

Register address	DisplayBaseAddress + 14A <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3WH											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the vertical direction display size (height) of the L3 layer window. Setting value + 1 is the height.

**L4M (L4 layer Mode)**

Register address	DisplayBaseAddress + 70 <sub>H</sub>																															
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	L4C	L4FLP		Reserved				L4W				Reserved				L4H																
R/W	RW	RW		R0				RW				R0				RW																
Initial value				0				Don't care				0				Don't care																

Bit 11 to 0      L4H (L4 layer Height)  
 Specifies the height of the logic frame of the L4 layer in pixel units. Setting value + 1 is the height

Bit 23 to 16      L4W (L4 layer memory Width)  
 Sets the memory width (stride) logic frame of the L4 layer in 64-byte units

Bit 30 and 29      L4FLP (L4 layer Flip mode)  
 Sets flipping mode for L4 layer

- 00    Displays frame 0
- 01    Displays frame 1
- 10    Switches frame 0 and 1 alternately for display
- 11    Reserved

Bit 31              L4C (L4 layer Color mode)  
 Sets the color mode for L4 layer

- 0      Indirect color (8 bits/pixel) mode
- 1      Direct color (16 bits/pixel) mode

**L4EM (L4 layer Extended Mode)**

Register address	DisplayBaseAddress + 150 <sub>H</sub>																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	---	4	3	2	1	0
Bit field name	L4EC		Reserved														L4OM		L4WP									
R/W	RW		R0				RW				R0				RW		RW											
Initial value	00		0				0				0				0		0											

- Bit 0            L4 WP (L4 layer Window Position enable)  
 Selects the display position of L4 layer

  - 0    Compatibility mode display (BL layer supported)
  - 1    Window display
  
- Bit 1            L4OM (L4 layer Overlay Mode)  
 Selects the overlay mode for L4 layer

  - 0    Compatibility mode
  - 1    Extended mode
  
- Bit 23 to 20    L4PB (L4 layer Palette Base)  
 Shows the value added to the index when subtracting palette of L4 layer. 16 times of setting value is added.
  
- Bit 31 and 30   L4EC (L4 layer Extended Color mode)  
 Sets extended color mode for L4 layer

  - 00   Mode determined by L4C
  - 01   Direct color (24 bits/pixel) mode
  - 1x   Reserved

**L4OA0 (L4 layer Origin Address 0)**

Register address	DisplayBaseAddress + 74 <sub>H</sub>																															
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				L4OA0																											
R/W	R0				RW												R0															
Initial value	0				Don't care												0000															

This register sets the origin address of the logic frame of the L4 layer in frame 0. Since lower 4 bits are fixed to “0”, this address is 16-byte aligned.

**L4DA0 (L4 layer Display Address 0)**

Register address	DisplayBaseAddress + 78 <sub>H</sub>																															
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				L4DA0																											
R/W	R0				RW																											
Initial value	0				Don't care																											

This register sets the origin address of the L4 layer in frame 0. For the direct color mode (16 bits/pixel), the lower 1 bit is “0” and this address is 2-byte aligned.

**L4OA1 (L4 layer Origin Address 1)**

Register address	DisplayBaseAddress + 7C <sub>H</sub>																															
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				L4OA1																											
R/W	R0				RW												R0															
Initial value	0				Don't care												0000															

This register sets the origin address of the logic frame of the L4 layer in frame 1. Since lower 4-bits are fixed to “0”, this address is 16-byte aligned.

**L4DA1 (L4 layer Display Address 1)**

Register address	DisplayBaseAddress + 80 <sub>H</sub>																															
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				L4DA1																											
R/W	R0				RW																											
Initial value	0				Don't care																											

This register sets the origin address of the L4 layer in frame 1. For the direct color mode (16 bits/pixel), the lower 1 bit is “0” and this address is 2-byte aligned.

**L4DX (L4 layer Display position X)**

Register address	DisplayBaseAddress + 84 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4DX											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the display starting position (X coordinates) of the L4 layer on the basis of the origin of the logic frame in pixels.

**L4DY (L4 layer Display position Y)**

Register address	DisplayBaseAddress + 86 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4DY											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the display starting position (Y coordinates) of the L4 layer on the basis of the origin of the logic frame in pixels.

**L4WX (L4 layer Window position X)**

Register address	DisplayBaseAddress + 154 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4WX											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the X coordinates of the display position of the L4 layer window.

**L4WY (L4 layer Window position Y)**

Register address	DisplayBaseAddress + 156 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4WY											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the Y coordinates of the display position of the L4 layer window.

**L4WW (L4 layer Window Width)**

Register address	DisplayBaseAddress + 158 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4WW											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the horizontal direction display size (width) of the L4 layer window. Do not specify "0".

**L4WH (L4 layer Window Height)**

Register address	DisplayBaseAddress + 15A <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L4WH											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the vertical direction display size (height) of the L4 layer window. Setting value + 1 is the height.

**L5M (L5 layer Mode)**

Register address	DisplayBaseAddress + 88 <sub>H</sub>																															
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	L5C	L5FLP		Reserved				L5W				Reserved				L5H																
R/W	RW	RW		R0				RW				R0				RW																
Initial value				0				Don't care				0				Don't care																

Bit 11 to 0      L5H (L5 layer Height)  
 Specifies the height of the logic frame of the L5 layer in pixel units. Setting value + 1 is the height

Bit 23 to 16      L5W (L5 layer memory Width)  
 Sets the memory width (stride) logic frame of the L5 layer in 64-byte units

Bit 30 and 29      L5FLP (L5 layer Flip mode)  
 Sets flipping mode for L5 layer

- 00    Displays frame 0
- 01    Displays frame 1
- 10    Switches frame 0 and 1 alternately for display
- 11    Reserved

Bit 31              L5C (L5 layer Color mode)  
 Sets the color mode for L5 layer

- 0    Indirect color (8 bits/pixel) mode
- 1    Direct color (16 bits/pixel) mode

**L5EM (L5 layer Extended Mode)**

Register address	DisplayBaseAddress + 160 <sub>H</sub>																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	---	4	3	2	1	0
Bit field name	L5EC		Reserved														L5OM		L5WP									
R/W	RW		R0														RW		RW									
Initial value	00		0																0									

Bit 0            L5 WP (L5 layer Window Position enable)  
 Selects the display position of L5 layer  
 0    Compatibility mode display (BR layer supported)  
 1    Window display

Bit 1            L5OM (L5 layer Overlay Mode)  
 Selects the overlay mode for L5 layer  
 0    Compatibility mode  
 1    Extended mode

Bit 31 to 30    L5EC (L5 layer Extended Color mode)  
 Sets extended color mode for L5 layer  
 00   Mode determined by L5C  
 01   Direct color (24 bits/pixel) mode  
 1x   Reserved



**L5OA0 (L5 layer Origin Address 0)**

Register address	DisplayBaseAddress + 8C <sub>H</sub>																															
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				L5OA0																											
R/W	R0				RW													R0														
Initial value	0				Don't care													0000														

This register sets the origin address of the logic frame of the L5 layer in frame 0. Since lower 4 bits are fixed to “0”, this address is 16-byte aligned.

**L5DA0 (L5 layer Display Address 0)**

Register address	DisplayBaseAddress + 90 <sub>H</sub>																															
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				L5DA0																											
R/W	R0				RW																											
Initial value	0				Don't care																											

This register sets the origin address of the L5 layer in frame 0. For the direct color mode (16 bits/pixel), the lower 1 bit is “0” and this address is 2-byte aligned.

**L5OA1 (L5 layer Origin Address 1)**

Register address	DisplayBaseAddress + 94 <sub>H</sub>																															
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				L5OA1																											
R/W	R0				RW													R0														
Initial value	0				Don't care													0000														

This register sets the origin address of the logic frame of the L5 layer in frame 1. Since lower 4-bits are fixed to “0”, this address is 16-byte aligned.

**L5DA1 (L5 layer Display Address 1)**

Register address	DisplayBaseAddress + 98 <sub>H</sub>																															
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				L5DA1																											
R/W	R0				RW																											
Initial value	0				Don't care																											

This register sets the origin address of the L5 layer in frame 1. For the direct color mode (16 bits/pixel), the lower 1 bit is “0” and this address is 2-byte aligned.

**L5DX (L5 layer Display position X)**

Register address	DisplayBaseAddress + 9C <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5DX											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the display starting position (X coordinates) of the L5 layer on the basis of the origin of the logic frame in pixels.

**L5DY (L5 layer Display position Y)**

Register address	DisplayBaseAddress + 9E <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5DY											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the display starting position (Y coordinates) of the L5 layer on the basis of the origin of the logic frame in pixels.

**L5WX (L5 layer Window position X)**

Register address	DisplayBaseAddress + 164 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5WX											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the X coordinates of the display position of the L5 layer window.

**L5WY (L5 layer Window position Y)**

Register address	DisplayBaseAddress + 166 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5WY											
R/W	R0				RW											
Initial value	0				Don't care											

This register sets the Y coordinates of the display position of the L5 layer window.

**L5WW (L5 layer Window Width)**

Register address	DisplayBaseAddress + 168 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5WW											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the horizontal direction display size (width) of the L5 layer window. Do not specify "0".

**L5WH (L5 layer Window Height)**

Register address	DisplayBaseAddress + 16A <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L5WH											
R/W	R0				RW											
Initial value	0				Don't care											

This register controls the vertical direction display size (height) of the L5 layer window. Setting value + 1 is the height.

**CUTC (Cursor Transparent Control)**

Register address	DisplayBaseAddress + A0 <sub>H</sub>															
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							
Initial value	0							Don't care	Don't care							

Bit 7 to 0      CUTC (Cursor Transparent Code)  
 Sets color code handled as transparent code

Bit 8            CUZT (Cursor Zero Transparency)  
 Defines handling of color code 0  
 0      Code 0 as transparency color  
 1      Code 0 as non-transparency color

**CPM (Cursor Priority Mode)**

Register address	DisplayBaseAddress + A2 <sub>H</sub>							
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
Initial value	0		0	0	0		0	0

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

Bit 0            CUO0 (Cursor Overlap 0)  
 Sets display priority between cursor 0 and pixels of Console layer  
 0      Puts cursor 0 at lower than L0 layer.  
 1      Puts cursor 0 at higher than L0 layer.

Bit 1            CUO1 (Cursor Overlap 1)  
 Sets display priority between cursor 1 and C layer  
 0      Puts cursor 1 at lower than L0 layer.  
 1      Puts cursor 1 at lower than L0 layer.

Bit 4            CEN0 (Cursor Enable 0)  
 Sets enabling display of cursor 0  
 0      Disabled  
 1      Enabled

Bit 5            CEN1 (Cursor Enable 1)  
 Sets enabling display of cursor 1  
 0      Disabled  
 1      Enabled

**CUOA0 (Cursor-0 Origin Address)**

Register address	DisplayBaseAddress + A4 <sub>H</sub>																															
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				CUOA0																											
R/W	R0				RW													R0														
Initial value	0				Don't care													0000														

This register sets the start address of the cursor 0 pattern. Since lower 4 bits are fixed to “0”, this address is 16-byte aligned.

**CUX0 (Cursor-0 X position)**

Register address	DisplayBaseAddress + A8 <sub>H</sub>															
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											
Initial value	0				Don't care											

This register sets the display position (X coordinates) of the cursor 0 in pixels. The reference position of the coordinates is the top left of the cursor pattern.

**CUY0 (Cursor-0 Y position)**

Register address	DisplayBaseAddress + Aa <sub>H</sub>															
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											
Initial value	0				Don't care											

This register sets the display position (Y coordinates) of the cursor 0 in pixels. The reference position of the coordinates is the top left of the cursor pattern.

**CUOA1 (Cursor-1 Origin Address)**

Register address	DisplayBaseAddress + AC <sub>H</sub>																															
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				CUOA1																											
R/W	R0				RW													R0														
Initial value	0				Don't care													0000														

This register sets the start address of the cursor 1 pattern. Since lower 4 bits are fixed to “0”, this address is 16-byte aligned.

**CUX1 (Cursor-1 X position)**

Register address	DisplayBaseAddress + B0 <sub>H</sub>															
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											
Initial value	0				Don't care											

This register sets the display position (X coordinates) of the cursor 1 in pixels. The reference position of the coordinates is the top left of the cursor pattern.

**CUY1 (Cursor-1 Y position)**

Register address	DisplayBaseAddress + B2 <sub>H</sub>															
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											
Initial value	0				Don't care											

This register sets the display position (Y coordinates) of the cursor 1 in pixels. The reference position of the coordinates is the top left of the cursor pattern.

**DLS (Display Layer Select)**

Register address	DisplayBaseAddress + 180 <sub>H</sub>																													
Bit number	31	30	29	-----	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				DLS5			DLS4			DLS3			DLS2			DLS1			DSL0										
R/W	R0				R0	RW	R0	RW	R0	RW	R0	RW	R0	RW	R0	RW	R0	RW	R0	RW	R0	RW	R0	RW	R0	RW	R0	RW	R0	RW
Initial value							101			100			011			010			001			000								

This register defines the blending sequence.

- Bit 3 to 0      DSL0 (Display Layer Select 0)  
 Selects the top layer subjected to blending.

0000	L0 layer
0001	L1 layer
:	:
0101	L5 layer
0110	Reserved
:	:
0110	Reserved
0111	Not selected
  
- Bit 7 to 4      DSL1 (Display Layer Select 1)  
 Selects the second layer subjected to blending. The bit values are the same as DSL0.
  
- Bit 11 to 8     DSL2 (Display Layer Select 2)  
 Selects the third layer subjected to blending. The bit values are the same as DSL0.
  
- Bit 15 to 12   DSL3 (Display Layer Select 3)  
 Selects the fourth layer subjected to blending. The bit values are the same as DSL0.
  
- Bit 19 to 16   DSL4 (Display Layer Select 4)  
 Selects the fifth layer subjected to blending. The bit values are the same as DSL0.
  
- Bit 23 to 20   DSL5 (Display Layer Select 5)  
 Selects the bottom layer subjected to blending. The bit values are the same as DSL0.

**DBGC (Display Background Color)**

Register address	DisplayBaseAddress + 184 <sub>H</sub>																															
Bit number	31	30	29	-----	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				DBGR								DBGG								DBGB											
R/W	R0																															
Initial value																																

This register specifies the color to be displayed in areas outside the display area of each layer on the window.

Bit 7 to 0      DBGB (Display Background Blue)  
 Specifies the blue level of the background color.

Bit 15 to 8    DBGG (Display Background Green)  
 Specifies the green level of the background color.

Bit 23 to 16   DBGR (Display Background Red)  
 Specifies the red level of the background color.

**L0BLD (L0 Blend)**

Register address	DisplayBaseAddress + B4 <sub>H</sub>																									
Bit number	31	30	29	28	-----	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								LOBE	LOBS	LOBI	LOBP	Reserved				LOBR									
R/W	R0								RW	RW	RW	RW	R0				RW									
Initial value									0	0	0	0					0									

This register specifies the blend parameters for the L0 layer. This register corresponds to BRATIO or BMODE for previous products.

- Bit 7 to 0      L0BR (L0 layer Blend Ratio)  
 Sets the blend ratio. Basically, the blend ratio is setting value/256.
  
- Bit 13          L0BP (L0 layer Blend Plane)  
 Specifies that the L5 layer is the blend plane.  
 0      Value of L0BR used as blend ratio  
 1      Pixel of L5 layer used as blend ratio
  
- Bit 14          L0BI (L0 layer Blend Increment)  
 Selects whether or not 1/256 is added when the blend ratio is not "0".  
 0      Blend ratio calculated as is  
 1      1/256 added when blend ratio ≠ 0
  
- Bit 15          L0BS (L0 layer Blend Select)  
 Selects the blend calculation expression.  
 0      Upper image × Blend ratio + Lower image × (1 – Blend ratio)  
 1      Upper image × (1 – Blend ratio) + Lower image × Blend ratio
  
- Bit 16          L0BE (L0 layer Blend Enable)  
 This bit enables blending.  
 0      Overlay via transparent color  
 1      Overlay via blending

Before blending, the blend mode must be specified using L0BE, and alpha must also be enabled for L0 layer display data. For direct color, alpha is specified using the MSB of data; for indirect color, alpha is specified using the MSB of palette data.



**L1BLD (L1 Blend)**

Register address	DisplayBaseAddress + 188 <sub>H</sub>																									
Bit number	31	30	29	28	-----	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								L1BE	L1BS	L1BI	L1BP	Reserved					L1BR								
R/W	R0								RW	RW	RW	RW	R0					RW								
Initial value									0	0	0	0						0								

This register specifies the blend parameters for the L1 layer.

- Bit 7 to 0    L1BR (L1 layer Blend Ratio)  
 Sets the blend ratio. Basically, the blend ratio is setting value/256.
  
- Bit 13       L1BP (L1 layer Blend Plane)  
 Specifies that the L5 layer is the blend plane.  
 0       Value of L1BR used as blend ratio  
 1       Pixel of L5 layer used as blend ratio
  
- Bit 14       L1BI (L1 layer Blend Increment)  
 Selects whether or not 1/256 is added when the blend ratio is not "0".  
 0       Blend ratio calculated as is  
 1       1/256 added when blend ratio ≠ 0
  
- Bit 15       L1BS (L1 layer Blend Select)  
 Selects the blend calculation expression.  
 0       Upper image × Blend ratio + Lower image × (1 – Blend ratio)  
 1       Upper image × (1 – Blend ratio) + Lower image × Blend ratio
  
- Bit 16       L1BE (L1 layer Blend Enable)  
 This bit enables blending.  
 0       Overlay via transparent color  
 1       Overlay via blending

Before blending, the blend mode must be specified using L1BE, and alpha must also be enabled for L1 layer display data. For direct color, alpha is specified using the MSB of data; for indirect color, alpha is specified using the MSB of palette data.

**L2BLD (L2 Blend)**

Register address	DisplayBaseAddress + 18C <sub>H</sub>																									
Bit number	31	30	29	28	-----	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								L2BE	L2BS	L2BI	L2BP	Reserved				L2BR									
R/W	R0								RW	RW	RW	RW	R0				RW									
Initial value									0	0	0	0					0									

This register specifies the blend parameters for the L2 layer.

- Bit 7 to 0    L2BR (L2 layer Blend Ratio)  
 Sets the blend ratio. Basically, the blend ratio is setting value/256.
  
- Bit 13       L2BP (L2 layer Blend Plane)  
 Specifies that the L5 layer is the blend plane.  
 0       Value of L2BR used as blend ratio  
 1       Pixel of L5 layer used as blend ratio
  
- Bit 14       L2BI (L2 layer Blend Increment)  
 Selects whether or not 1/256 is added when the blend ratio is not "0".  
 0       Blend ratio calculated as is  
 1       1/256 added when blend ratio ≠ 0
  
- Bit 15       L2BS (L2 layer Blend Select)  
 Selects the blend calculation expression.  
 0       Upper image × Blend ratio + Lower image × (1 – Blend ratio)  
 1       Upper image × (1 – Blend ratio) + Lower image × Blend ratio
  
- Bit 16       L2BE (L2 layer Blend Enable)  
 This bit enables blending.  
 0       Overlay via transparent color  
 1       Overlay via blending

Before blending, the blend mode must be specified using L2BE, and alpha must also be enabled for L2 layer display data. For direct color, alpha is specified using the MSB of data; for indirect color, alpha is specified using the MSB of palette data.

**L3BLD (L3 Blend)**

Register address	DisplayBaseAddress + 190 <sub>H</sub>																									
Bit number	31	30	29	28	-----	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								L3BE	L3BS	L3BI	L3BP	Reserved					L3BR								
R/W									RW	Rw	RW	RW						RW								
Initial value									0	0	0	0						0								

This register specifies the blend parameters for the L3 layer.

- Bit 7 to 0    L3BR (L3 layer Blend Ratio)  
 Sets the blend ratio. Basically, the blend ratio is setting value/256.
  
- Bit 13        L3BP (L3 layer Blend Plane)  
 Specifies that the L5 layer is the blend plane.  
 0        Value of L3BR used as blend ratio  
 1        Pixel of L5 layer used as blend ratio
  
- Bit 14        L3BI (L3 layer Blend Increment)  
 Selects whether or not 1/256 is added when the blend ratio is not "0".  
 0        Blend ratio calculated as is  
 1        1/256 added when blend ratio ≠ 0
  
- Bit 15        L3BS (L3 layer Blend Select)  
 Selects the blend calculation expression.  
 0        Upper image × Blend ratio + Lower image × (1 – Blend ratio)  
 1        Upper image × (1 – Blend ratio) + Lower image × Blend ratio
  
- Bit 16        L3BE (L3 layer Blend Enable)  
 This bit enables blending.  
 0        Overlay via transparent color  
 1        Overlay via blending

Before blending, the blend mode must be specified using L3BE, and alpha must also be enabled for L3 layer display data. For direct color, alpha is specified using the MSB of data; for indirect color, alpha is specified using the MSB of palette data.

**L4BLD (L4 Blend)**

Register address	DisplayBaseAddress + 194 <sub>H</sub>																									
Bit number	31	30	29	28	-----	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								L4BE	L4BS	L4BI	L4BP	Reserved				L4BR									
R/W	R0								RW	RW	RW	RW	R0				RW									
Initial value									0	0	0	0					0									

This register specifies the blend parameters for the L4 layer.

- Bit 7 to 0    L4BR (L4 layer Blend Ratio)  
 Sets the blend ratio. Basically, the blend ratio is setting value/256.
  
- Bit 13        L4BP (L4 layer Blend Plane)  
 Specifies that the L5 layer is the blend plane.  
 0        Value of L4BR used as blend ratio  
 1        Pixel of L5 layer used as blend ratio
  
- Bit 14        L4BI (L4 layer Blend Increment)  
 Selects whether or not 1/256 is added when the blend ratio is not "0".  
 0        Blend ratio calculated as is  
 1        1/256 added when blend ratio ≠ 0
  
- Bit 15        L4BS (L4 layer Blend Select)  
 Selects the blend calculation expression.  
 0        Upper image × Blend ratio + Lower image × (1 – Blend ratio)  
 1        Upper image × (1 – Blend ratio) + Lower image × Blend ratio
  
- Bit 16        L4BE (L4 layer Blend Enable)  
 This bit enables blending.  
 0        Overlay via transparent color  
 1        Overlay via blending

Before blending, the blend mode must be specified using L4BE, and alpha must also be enabled for L4 layer display data. For direct color, alpha is specified using the MSB of data; for indirect color, alpha is specified using the MSB of palette data.

**L5BLD (L5 Blend)**

Register address	DisplayBaseAddress + 198h																															
Bit number	31	30	29	28	-----	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										L5BE	L5BS	L5BI	Reserved							L5BR											
R/W	R0										RW	RW	RW	R0							RW											
Initial value											0	0	0																			

This register specifies the blend parameters for the L5 layer.

- Bit 7 to 0    L5BR (L5 layer Blend Ratio)  
 Sets the blend ratio. Basically, the blend ratio is setting value/256.
  
- Bit 14       L5BI (L5 layer Blend Increment)  
 Selects whether or not 1/256 is added when the blend ratio is not "0".  
 0       Blend ratio calculated as is  
 1       1/256 added when blend ratio ≠ 0
  
- Bit 15       L5BS (L5 layer Blend Select)  
 Selects the blend calculation expression.  
 0       Upper image × Blend ratio + Lower image × (1 – Blend ratio)  
 1       Upper image × (1 – Blend ratio) + Lower image × Blend ratio
  
- Bit 16       L5BE (L5 layer Blend Enable)  
 This bit enables blending.  
 0       Overlay via transparent color  
 1       Overlay via blending

Before blending, the blend mode must be specified using L5BE, and alpha must also be enabled for L5 layer display data. For direct color, alpha is specified using the MSB of data; for indirect color, alpha is specified using the MSB of palette data.

**L0TC (L0 layer Transparency Control)**

Register address	DisplayBaseAddress + BC <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L0ZT	L0TC														
R/W	RW	RW														
Initial value	0	0														

This register sets the transparent color for the L0 layer. Color set by this register is transparent in blend mode. When L0TC = 0 and L0ZT = 0, color 0 is displayed in black (transparent).

This register corresponds to the CTC register for previous products.

- Bit 14 to 0    L0TC (L0 layer Transparent Color)  
 Sets transparent color code for the L0 layer. In indirect color mode (8 bits/pixel) bits 7 to 0 are used.
  
- Bit 15        L0ZT (L0 layer Zero Transparency)  
 Sets handling of color code 0 in L0 layer  
 0: Code 0 as transparency color  
 1: Code 0 as non-transparency color

**L2TC (L2 layer Transparency Control)**

Register address	DisplayBaseAddress + C2 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L2ZT	L2TC														
R/W	RW	RW														
Initial value	0	0														

This register sets the transparent color for the L2 layer.

When L2TC = 0 and L2ZT = 0, color 0 is displayed in black (transparent).

This register corresponds to the MLTC register for previous products.

- Bit 14 to 0    L2TC (L2 layer Transparent Color)  
 Sets transparent color code for the L2 layer. In indirect color mode (8 bits/pixel) bits 7 to 0 are used.
  
- Bit 15        L2ZT (L2 layer Zero Transparency)  
 Sets handling of color code 0 in L2 layer  
 0    Code 0 as transparency color  
 1    Code 0 as non-transparency color

**L3TC (L3 layer Transparency Control)**

Register address	DisplayBaseAddress + C0 <sub>H</sub>															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	L3ZT	L3TC														
R/W	RW	RW														
Initial value	0	0														

This register sets the transparent color for the L3 layer. When L3TC = 0 and L3ZT = 0, color 0 is displayed in black (transparent).

This register corresponds to the MLTC register for previous products.

- Bit 14 to 0    L3TC (L3 layer Transparent Color)  
 Sets transparent color code for the L3 layer. In indirect color mode (8 bits/pixel) bits 7 to 0 are used.
  
- Bit 15        L3ZT (L3 layer Zero Transparency)  
 Sets handling of color code 0 in L3 layer  
 0    Code 0 as transparency color  
 1    Code 0 as non-transparency color

**L0ETC (L0 layer Extend Transparency Control)**

Register address	DisplayBaseAddress + 1A0 <sub>H</sub>																															
Bit number	31	30	29	28	---	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	L0ETZ	Reserved					L0TEC																									
R/W	RW	RO					RW																									
Initial value	0						0																									

This register sets the transparent color for the L0 layer. The 24 bits/pixel transparent color is set using this register. The lower 15 bits of this register are physically the same as L0TC. Also, L0ETZ is physically the same as L0TZ.

When L0ETC = 0 and L0EZT = 0, color 0 is displayed in black (transparent).

- Bit 23 to 0    L0ETC (L0 layer Extend Transparent Color)  
 Sets transparent color code for the L0 layer. In indirect color mode (8 bits/pixel) bits 7 to 0 are used.
  
- Bit 31        L0EZT (L0 layer Extend Zero Transparency)  
 Sets handling of color code 0 in L0 layer  
 0    Code 0 as transparency color  
 1    Code 0 as non-transparency color

**L1ETC (L1 layer Extend Transparency Control)**

Register address	DisplayBaseAddress + 1A4 <sub>H</sub>																															
Bit number	31	30	29	28	---	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	L1ETZ		Reserved										L1TEC																			
R/W	RW		R0										RW																			
Initial value	0												0																			

This register sets the transparent color for the L1 layer. When L1ETC = 0 and L1EZT = 0, color 0 is displayed in black (transparent).

For YCbCr display, transparent color checking is not performed; processing is always performed assuming that transparent color is not used.

- Bit 23 to 0    L1ETC (L1 layer Extend Transparent Color)  
 Sets transparent color code for the L1 layer. In indirect color mode (8 bits/pixel) bits 7 to 0 are used.
  
- Bit 31        L1EZT (L1 layer Extend Zero Transparency)  
 Sets handling of color code 0 in L1 layer  
 0    Code 0 as transparency color  
 1    Code 0 as non-transparency color

**L2ETC (L2 layer Extend Transparency Control)**

Register address	DisplayBaseAddress + 1A8 <sub>H</sub>																															
Bit number	31	30	29	28	---	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	L2ETZ		Reserved										L2TEC																			
R/W	RW		R0										RW																			
Initial value	0												0																			

This register sets the transparent color for the L2 layer. The 24 bits/pixel transparent color is set using this register. The lower 15 bits of this register are physically the same as L2TC. Also, L2ETZ is physically the same as L2TZ.

When L2ETC = 0 and L2EZT = 0, color 0 is displayed in black (transparent).

- Bit 23 to 0    L2ETC (L2 layer Extend Transparent Color)  
 Sets transparent color code for the L2 layer. In indirect color mode (8 bits/pixel) bits 7 to 0 are used.
  
- Bit 31        L2EZT (L2 layer Extend Zero Transparency)  
 Sets handling of color code 0 in L2 layer  
 0    Code 0 as transparency color  
 1    Code 0 as non-transparency color



**L3ETC (L3 layer Extend Transparency Control)**

Register address	DisplayBaseAddress + 1AC <sub>H</sub>																															
Bit number	31	30	29	28	---	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	L3ETZ		Reserved										L3TEC																			
R/W	RW		R0										RW																			
Initial value	0												0																			

This register sets the transparent color for the L3 layer. The 24 bits/pixel transparent color is set using this register. The lower 15 bits of this register are physically the same as L3TC. Also, L3ETZ is physically the same as L3TZ.

When L3ETC = 0 and L3EET = 0, color 0 is displayed in black (transparent).

- Bit 23 to 0    L3ETC (L3 layer Extend Transparent Color)  
 Sets transparent color code for the L3 layer. In indirect color mode (8 bits/pixel) bits 7 to 0 are used.
  
- Bit 31        L3EET (L3 layer Extend Zero Transparency)  
 Sets handling of color code 0 in L3 layer  
 0    Code 0 as transparency color  
 1    Code 0 as non-transparency color

**L4ETC (L4 layer Extend Transparency Control)**

Register address	DisplayBaseAddress + 1B0 <sub>H</sub>																															
Bit number	31	30	29	28	---	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	L4ETZ		Reserved										L4TEC																			
R/W	RW		R0										RW																			
Initial value	0												0																			

This register sets the transparent color for the L4 layer. This register sets the transparent color for the L4 layer. When L4ETC = 0 and L4EET = 0, color 0 is displayed in black (transparent).

- Bit 23 to 0    L4ETC (L4 layer Extend Transparent Color)  
 Sets transparent color code for the L4 layer. In indirect color mode (8 bits/pixel) bits 7 to 0 are used.
  
- Bit 31        L4EET (L4 layer Extend Zero Transparency)  
 Sets handling of color code 0 in L4 layer  
 0    Code 0 as transparency color  
 1    Code 0 as non-transparency color

**L5ETC (L5 layer Extend Transparency Control)**

Register address	DisplayBaseAddress + 1B4 <sub>H</sub>																															
Bit number	31	30	29	28	---	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	L5ETZ		Reserved										L5TEC																			
R/W	RW		R0										RW																			
Initial value	0												0																			

This register sets the transparent color for the L5 layer. This register sets the transparent color for the L5 layer. When L5ETC = 0 and L5EZT = 0, color 0 is displayed in black (transparent).

- Bit 23 to 0    L5ETC (L5 layer Extend Transparent Color)  
 Sets transparent color code for the L5 layer. In indirect color mode (8 bits/pixel) bits 7 to 0 are used.
  
- Bit 31        L5EZT (L5 layer Extend Zero Transparency)  
 Sets handling of color code 0 in L5 layer
  - 0    Code 0 as transparency color
  - 1    Code 0 as non-transparency color

**L0PAL0-255 (L0 layer Palette 0-255)**

Register address	DisplayBaseAddress + 400 <sub>H</sub> -- DisplayBaseAddress + 7FF <sub>H</sub>																																			
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
Initial value	Don't care	0000000								Don't care								00	Don't care								00	Don't care								00

These are color palette registers for L0 layer and cursors. In the indirect color mode, a color code in the display frame indicates the palette register number, and the color information set in that register is applied as the display color of that pixel. This register corresponds to the CPAL<sub>n</sub> register for previous products.

- Bit 7 to 2      B (Blue)  
 Sets blue color component
  
- Bit 15 to 10    G (Green)  
 Sets green color component
  
- Bit 23 to 18    R (Red)  
 Sets red color component
  
- Bit 31          A (Alpha)  
 Specifies whether or not to perform blending with lower layers when the blending mode is enabled.
  - 0      Blending not performed even when blending mode enabled  
 Overlay is performed via transparent color.
  - 1      Blending performed

**L1PAL0-255 (L1 layer Palette 0-255)**

Register address	DisplayBaseAddress + 800 <sub>H</sub> -- DisplayBaseAddress + BFF <sub>H</sub>																																			
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
Initial value	Don't care	0000000								Don't care								00	Don't care								00	Don't care								00

These are color palette registers for L1 layer and cursors. In the indirect color mode, a color code in the display frame indicates the palette register number, and the color information set in that register is applied as the display color of that pixel. This register corresponds to the MBPALn register for previous products.

- Bit 7 to 2      B (Blue)  
Sets blue color component
  
- Bit 15 to 10    G (Green)  
Sets green color component
  
- Bit 23 to 18    R (Red)  
Sets red color component
  
- Bit 31          A (Alpha)  
Specifies whether or not to perform blending with lower layers when the blending mode is enabled.
  - 0      Blending not performed even when blending mode enabled  
         Overlay is performed via transparent color.
  - 1      Blending performed

**L2PAL0-255 (L2 layer Palette 0-255)**

Register address	DisplayBaseAddress + 1000 <sub>H</sub> -- DisplayBaseAddress + 13FF <sub>H</sub>																															
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								
Initial value	Don't care	0000000								Don't care				00	Don't care				00	Don't care				00								

These are color palette registers for L2 layer and cursors. In the indirect color mode, a color code in the display frame indicates the palette register number, and the color information set in that register is applied as the display color of that pixel.

- Bit 7 to 2      B (Blue)  
 Sets blue color component
  
- Bit 15 to 10    G (Green)  
 Sets green color component
  
- Bit 23 to 18    R (Red)  
 Sets red color component
  
- Bit 31          A (Alpha)  
 Specifies whether or not to perform blending with lower layers when the blending mode is enabled.
  - 0      Blending not performed even when blending mode enabled  
 Overlay is performed via transparent color.
  - 1      Blending performed

**L3PAL0-255 (L3 layer Palette 0-255)**

Register address	DisplayBaseAddress + 1400 <sub>H</sub> -- DisplayBaseAddress + 17FF <sub>H</sub>																															
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												
Initial value	Don't care	0000000				Don't care				00	Don't care				00	Don't care				00												

These are color palette registers for L3 layer and cursors. In the indirect color mode, a color code in the display frame indicates the palette register number, and the color information set in that register is applied as the display color of that pixel.

- Bit 7 to 2      B (Blue)  
 Sets blue color component
  
- Bit 15 to 10    G (Green)  
 Sets green color component
  
- Bit 23 to 18    R (Red)  
 Sets red color component
  
- Bit 31          A (Alpha)  
 Specifies whether or not to perform blending with lower layers when the blending mode is enabled.
  - 0      Blending not performed even when blending mode enabled  
 Overlay is performed via transparent color.
  - 1      Blending performed

## 10.2.4 Drawing control registers

### CTR (Control Register)

Register address	DrawBaseAddress + 400 <sub>H</sub>																																		
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	CE	FCNT							NF	FF	FE	SS							DS	PS							
R/W								R	R	R	R							R	R	R	R							R	R						
Initial value								0	0	0	011101							0	0	1	00							00	00						

This register indicates drawing flags and status information. Bits 24 to 22 are not cleared until 0 is set.

Bit 1 and 0 PS (Pixel engine Status)

Indicate status of pixel engine unit

00 Idle

01 Busy

10 Reserved

11 Reserved

Bit 5 and 4 DS (DDA Status)

Indicate status of DDA

00 Idle

01 Busy

10 Busy

11 Reserved

Bit 9 and 8 SS (Setup Status)

Indicate status of Setup unit

00 Idle

01 Busy

10 Reserved

11 Reserved

Bit 12 FE (FIFO Empty)

Indicates whether data contained or not in display list FIFO

0 Valid data

1 No valid data

Bit 13 FF (FIFO Full)

Indicates whether display list FIFO is full or not

0 Not full

1 Full

Bit 14 NF (FIFO Near Full)

Indicates how empty the display list FIFO is

0 Empty entries equal to or more than half

	1	Empty entries less than half
Bit 20 to 15		FCNT (FIFO Counter) Indicates count of empty entries of display list FIFO (0 to 100000 <sub>b</sub> )
Bit 23-22		CE (Display List Command Error) Indicates command error occurrence
	00	Normal
	11	Command error detected
Bit 24		FO (FIFO Overflow) Indicates FIFO overflow occurrence
	0	Normal
	1	FIFO overflow detected



**IFSR (Input FIFO Status Register)**

Register address	DrawBaseAddress + 404 <sub>H</sub>																															
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		
Initial value																												0	0	1		

This is a mirror register for bits 14 to 12 of the CTR register.

**IFCNT (Input FIFO Counter)**

Register address	DrawBaseAddress + 408 <sub>H</sub>																															
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				
Initial value																												011101				

This is a mirror register for bits 19 to 15 of the CTR register.

**SST (Setup engine Status)**

Register address	DrawBaseAddress + 40C <sub>H</sub>																															
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				
Initial value																												00				

This is a mirror register for bits 9 to 8 of the CTR register.

**DST (DDA Status)**

Register address	DrawBaseAddress + 410 <sub>H</sub>																															
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				
Initial value																												00				

This is a mirror register for bits 5 to 4 of the CTR register.

**PST (Pixel engine Status)**

Register address	DrawBaseAddress + 414 <sub>H</sub>																															
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				
Initial value																												00				

This is a mirror register for bits 1 to 0 of the CTR register.

**EST (Error Status)**

Register address	DrawBaseAddress + 418 <sub>H</sub>																															
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		
Initial value																												0	0	0		

This is a mirror register for bits 24 to 22 of the CTR register.

## 10.2.5 Drawing 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 <sub>H</sub>																															
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													ZP					CF					CY	CX					BSV	BSH		
R/W													RW					RW					RW	RW					RW	RW		
Initial value													0					00					0	0					00	00		

Bit 1 to 0      BSH (Bitmap Scale Horizontal)  
 Sets horizontal zoom ratio of bitmap draw  
 00    x1  
 01    x2  
 10    x1/2  
 01    Reserved

Bit 3 to 2      BSV (Bitmap Scale Vertical)  
 Sets vertical zoom ratio of bitmap draw  
 00    x1  
 01    x2  
 10    x1/2  
 01    Reserved

Bit 8            CX (Clip X enable)  
 Sets X coordinates clipping mode  
 0    Disabled  
 1    Enabled

Bit 9            CY (Clip Y enable)  
 Sets Y coordinates clipping mode  
 0    Disabled  
 1    Enabled

Bit 16 and 15    CF (Color Format)  
 Sets drawing color format  
 00    Indirect color mode (8 bits/pixel)  
 01    Direct color mode (16 bits/pixel)  
 10    Direct color mode (24 bits/pixel)

Bit 20            ZP (Z Precision)  
 Sets the precision of the Z value used for erasing hidden planes.  
                   16 bits/pixel  
                   8 bits/pixel

**MDR1/MDR1S/MDR1B (Mode Register for LINE/for Shadow/for Border)**

Register address	DrawBaseAddress + 424 <sub>H</sub>																															
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					LW								BP	BL									LOG	BM	ZW	ZCL	ZC					
R/W					RW								RW	RW									RW	RW	RW	RW	RW					
Initial value					00000								0	0									0011	0	0	0000	0					

This register sets the mode of line and pixel drawing.

This register is used for the body primitive, for the shade primitive, for the edge primitive.

The value after a drawing that involves the shade primitive, the edge primitive is the value set for MDR1.

Please set ZC bit ( bit 2) to 0 when draw BitCopyAltAlphaBlendP command.

Bit 1 AS (Alpha Shading mode)  
 Sets the shading mode for alpha.  
 0 Alpha flat shading  
 1 Alpha Gouraud shading

Bit 2 ZC (Z Compare mode)  
 Sets Z comparison mode  
 0 Disabled  
 1 Enabled

Bit 5 to 3 ZCL (Z Compare Logic)  
 Selects 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 mode)  
 Sets Z write mode  
 0 Writes Z values.  
 1 Not write Z values.

Bit 8 to 7 BM (Blend Mode)  
 Sets blend mode  
 00 Normal (source copy)  
 01 Alpha blending  
 10 Drawing with logic operation  
 11 Reserved

Bit 12 to 9	LOG (Logical operation) Sets type of logic 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 cycle
0:	32 bits
1:	24 bits
Bit 28 to 24	LW (Line Width) Sets line width for drawing line
00000	1 pixel
00001	2 pixels
:	:
11111	32 pixels

**MDR2/MDR2S/MDR2TL (Mode Register for Polygon/for Shadow/for TopLeft)**

Register address	DrawBaseAddress + 428 <sub>H</sub>																															
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	AS	SM			
R/W			RW																		RW		RW	RW	RW		RW	RW	RW			
Initial value			00																		0011		0	0	0000		0	0	0			

This register sets the polygon drawing mode.

This register is used for the body primitive, for the shade primitive, and for the top-left non-applicable primitive.

The value after a drawing that involves the shade primitive or the top-left non-applicable primitive is the value set for MDR2.

MDR2S register is able to use only SM=0, AS=0 and TT=00 settings.

Bit 0 SM (Shading Mode)  
 Sets shading mode  
 0 Flat shading  
 1 Gouraud shading

Bit 1 AS (Alpha Shading mode)  
 Sets alpha shading mode. This mode is enabled for only alpha.  
 0 Alpha flat shading  
 1 Alpha gouraud shading

Bit 2 ZC (Z Compare mode)  
 Sets Z comparison mode  
 0 Disabled  
 1 Enabled

Bit 5 to 3 ZCL (Z Compare Logic)  
 Selects 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 Z write mode  
 0 Writes Z values  
 1 Not write Z values

Bit 8 to 7      BM (Blend Mode)  
Sets blend mode  
00    Normal (source copy)  
01    Alpha blending  
10    Drawing with logic operation  
11    Reserved

Bit 12 to 9    LOG (Logical operation)  
Sets type of logic 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 29 to 28   TT (Texture-Tile Select)  
Selects texture or tile pattern  
00    Neither used  
01    Enabled tiling  
10    Enabled texture  
11    Reserved

**MDR3 (Mode Register for Texture)**

Register address	DrawBaseAddress + 42C <sub>H</sub>																															
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									BA	TAB				TBL				TWS		TWT		TF		TC								
R/W									RW	RW				RW				RW		RW		RW		RW								
Initial value									0	00				00				00		00		0		0								

This register sets the texture mapping mode.

- Bit 3                    TC (Texture coordinates Correct)  
 Sets texture coordinates correction mode

  - 0    Disabled
  - 1    Enabled
  
- Bit 5                    TF (Texture Filtering)  
 Sets type of texture interpolation (filtering)

  - 0    Point sampling
  - 1    Bi-linear filtering
  
- Bit 9 and 8            TWT (Texture Wrap T)  
 Sets type of texture coordinates T direction wrapping

  - 00   Repeat
  - 01   Cramp
  - 10   Border
  - 11   Reserved
  
- Bit 11 and 10        TWS (Texture Wrap S)  
 Sets type of texture coordinates S direction wrapping

  - 00   Repeat
  - 01   Cramp
  - 10   Border
  - 11   Reserved
  
- Bit 17 and 16        TBL (Texture Blend mode)  
 Sets texture blending mode

  - 00   De-curl
  - 01   Modulate
  - 10   Stencil
  - 11   Reserved
  
- Bit 21 and 20        TAB (Texture Alpha Blend mode)  
 Sets texture blending mode

The stencil mode and the stencil alpha mode are enabled only when the MDR2 register blend mode (BM) is set to the alpha blending mode. If it is not set to the alpha blending mode, the stencil mode and stencil alpha mode perform the same function as the normal mode.

- 00 Normal
- 01 Stencil
- 10 Stencil alpha
- 11 Reserved

Bit 24

BA (Bilinear Accelerate Mode)

Improves the performance of bi-linear filtering, although a texture area of four times the default texture area is used.

- 0 Default texture area used
- 1 Texture area four times default texture area used



**MDR4 (Mode Register for BLT)**

Register address	DrawBaseAddress + 430 <sub>H</sub>																															
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			
Initial value												0011				00													0			

This register controls the BLT mode.

Bit 1            TE (Transparent Enable)  
 Sets transparent mode  
 0:            Not perform transparent processing  
 1:            Not draw pixels that corresponds to set transparent color in BLT (transparency copy)  
 Note: Set the blend mode (BM) to normal.

Bit 8 to 7      BM (Blend Mode)  
 Sets blend mode  
 00          Normal (source copy)  
 01          Reserved  
 10          Drawing with logic operation  
 11          Reserved

Bit 12 to 9    LOG (Logical operation)  
 Sets logic 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

**MDR7 (Mode Register for Extension)**

Register address	DrawBaseAddress + 43C <sub>H</sub>																															
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																											LTH	EZ	GG	PGH	PTH	PZH
R/W																											W	W	W	W	W	W
Initial value																											1	0	0	0	0	0

This register controls the BLT mode.

- Bit 0            PZH (Polygon Z Hard mask)  
 Sets polygon-fill Z reference mode  
 0:            Hard mask on ( compatible Orchid)  
 1:            Hard mask off ( extension mode)
  
- Bit 1            PTH (Polygon Texture Hard mask)  
 Sets polygon-texture mode  
 0:            Hard mask on ( compatible Orchid)  
 1:            Hard mask off ( extension mode)
  
- Bit 2            PGH (Polygon Gouraud shading Hard mask)  
 Sets polygon-gouraud shading mode  
 0:            Hard mask on ( compatible Orchid)  
 1:            Hard mask off ( extension mode)
  
- Bit 4            GG (Gray scale Gouraud Shading)  
 Sets gray scale gouraud shading mode  
 0:            Hard mask on ( compatible Orchid)  
 1:            Hard mask off ( extension mode)
  
- Bit 5            EZ (Extend Z)  
 Sets new Z mode  
 0:            Z 1 bit extend off ( compatible Orchid)  
 1:            Z 1 bit extend on ( extension mode)
  
- Bit 6            LTH (Line Texture Hard mask)  
 Sets line texture mode  
 0:            Hard mask on ( compatible Orchid)  
 1:            Hard mask off ( extension mode)

Note: This register is used for gray scale gouraud shading. This register is changed by internal processing. Please don't set these bits except GG bit.

In case of gray scale gouraud shading drawing, please set this register to the follows.

1. Set this register to 0x00000050( GG bit and LTH bit equal to 1) before drawing.
2. Set this register to 0x00000040( LTH bit equal to 1) after drawing.

**FBR (Frame buffer Base)**

Register address	DrawBaseAddress + 440 <sub>H</sub>																															
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							
Initial value	Don't care																								0							

This register stores the base address of the drawing frame.

**XRES (X Resolution)**

Register address	DrawBaseAddress + 444 <sub>H</sub>																															
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															
Initial value																	Don't care															

This register sets the drawing frame horizontal resolution.

**ZBR (Z buffer Base)**

Register address	DrawBaseAddress + 448 <sub>H</sub>																															
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							
Initial value	Don't care																								0							

This register sets the Z buffer base address.

**TBR (Texture memory Base)**

Register address	DrawBaseAddress + 44C <sub>H</sub>																															
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							
Initial value	Don't care																								0							

This register sets the texture memory base address.

**PFBR (2D Polygon Flag-Buffer Base)**

Register address	DrawBaseAddress + 450 <sub>H</sub>																															
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							
Initial value	Don't care																								0							

This register sets the polygon flag buffer base address.

**CXMIN (Clip X minimum)**

Register address	DrawBaseAddress + 454 <sub>H</sub>																															
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															
Initial value																	Don't care															

This register sets the clip frame minimum X position.

**CXMAX (Clip X maximum)**

Register address	DrawBaseAddress + 458 <sub>H</sub>																															
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															
Initial value																	Don't care															

This register sets the clip frame maximum X position.

**CYMIN (Clip Y minimum)**

Register address	DrawBaseAddress + 45C <sub>H</sub>																															
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															
Initial value																	Don't care															

This register sets the clip frame minimum Y position.

**CYMAX (Clip Y maximum)**

Register address	DrawBaseAddress + 460 <sub>H</sub>																															
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															
Initial value																	Don't care															

This register sets the clip frame maximum Y position.

**TXS (Texture Size)**

Register address	DrawBaseAddress + 464 <sub>H</sub>																															
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															
Initial value	000010000000																000010000000															

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

Bit 12 to 0 TXSM (Texture Size M)

Sets horizontal texture size. Any power of 2 between 4 and 4096 can be used. Values that are not a power of 2 cannot be used.

0_0000_0000_0100	M=4	0_0010_0000_0000	M=512
0_0000_0000_1000	M=8	0_0100_0000_0000	M=1024
0_0000_0001_0000	M=16	0_1000_0000_0000	M=2048
0_0000_0010_0000	M=32	1_0000_0000_0000	M=4096
0_0000_0100_0000	M=64		
0_0000_1000_0000	M=128		
0_0001_0000_0000	M=256	Other than the above	Setting disabled

Bit 28 to 16 TXSN (Texture Size N)

Sets vertical texture size. Any power of 2 between 4 and 4096 can be used. Values that are not a power of 2 cannot be used.

0_0000_0000_0100	N=4	0_0010_0000_0000	N=512
0_0000_0000_1000	N=8	0_0100_0000_0000	N=1024
0_0000_0001_0000	N=16	0_1000_0000_0000	N=2048
0_0000_0010_0000	N=32	1_0000_0000_0000	N=4096
0_0000_0100_0000	N=64		
0_0000_1000_0000	N=128		
0_0001_0000_0000	N=256	Other than the above	Setting disabled

**TIS (Tile Size)**

Register address	DrawBaseAddress + 468 <sub>H</sub>																															
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							
Initial value																	1000000								1000000							

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

Bit 6 to 0      TISM (Title Size M)  
 Sets horizontal tile 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
Other than the above	Setting disabled

Bit 22 to 16    TISN (Title Size N)  
 Sets vertical tile size. Any power of 2 between 4 and 64 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
Other than the above	Setting disabled

**TOA (Tiling Offset address)**

Register address	DrawBaseAddress + 46C <sub>H</sub>																															
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															
Initial value																	Don't care															

This register sets the texture buffer offset address. Using this offset value, texture patterns can be referred to the texture buffer memory. TOA is used for only the tiling drawing, and is not used for referring the texture pattern.

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

**SHO (SHadow Offset)**

Register address	DrawBaseAddress + 470 <sub>H</sub>																															
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	SHOFFS																															
R/W	RW																															
Initial value	Don't care																															

This register sets the offset address of the shadow relative to the body primitive at drawing with shadow.

At body drawing, this offset address is set to "0"; at shadow drawing, the offset address calculated from each offset value of the X coordinates and of the Y coordinates is set. This register is hardware controlled.

**ABR (Alpha map Base)**

Register address	DrawBaseAddress + 474 <sub>H</sub>																															
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	ABASE																															
R/W	RW																								R0							
Initial value	Don't care																								0							

This register sets the base address of the alpha map.

**FC (Foreground Color)**

Register address	DrawBaseAddress + 480 <sub>H</sub>																															
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																	FGC8/16															
R/W																	RW															
Initial value																	0															

This register sets the drawing foreground color. This color is for the object color for flat shading and foreground color for bitmap drawing and broken line drawing. All bits set to “1” are drawn in the color set at this register.

8 bit color mode:

- Bit 7 to 0      FGC8 (Foreground 8 bit Color)  
                      Sets the indirect color for the foreground (color index code).
- Bit 31 to 8     These bits are not used.

16 bit color mode:

- Bit 15 to 0     FGC16 (Foreground 16 bit Color)  
                      This field sets the 16-bit direct color for the foreground.  
                      Note that the handling of bit 15 is different from that in ORCHID.  
                      Up to ORCHID, bit 15 is “0” for other than bit map and rectangular drawing, but starting with CORAL, the setting value is reflected in memory as is. This bit is also reflected in bit 15 of the 16-bit color at Gouraud shading.
- Bit 31 to 16    These bits are not used.



**BC (Background Color)**

Register address	DrawBaseAddress + 484 <sub>H</sub>																															
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																	BGC8/16															
R/W																	RW															
Initial value																	0															

This register sets the drawing frame background color. This color is used for the background color of bitmap drawing and broken line drawing. At bitmap drawing, all bits set to “0” are drawn in the color set at this register. BT bit of this register allows the background color of be transparent (no drawing).

8 bit color mode:

- Bit 7 to 0      BGC8 (Background 8 bit Color)  
                   Sets the indirect color for the background (color index code)
- Bit 14 to 8    Not used
- Bit 15         BT (Background Transparency)  
                   Sets the transparent mode for the background color  
                   0      Background drawn using color set for BGC field  
                   1      Background not drawn (transparent)
- Bit 31 to 16   Not used

16 bit color mode:

- Bit 14 to 0    BGC16 (Background 16 bit Color)  
                   Sets 16-bit direct color (RGB) for the background
- Bit 15         BT (Background Transparency)  
                   Sets the transparent mode for the background color  
                   0      Background drawn using color set for BGC field  
                   1      Background not drawn (transparent)
- Bit 31 to 16   Not used

**ALF (Alpha Factor)**

Register address	DrawBaseAddress + 488 <sub>H</sub>																															
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			
Initial value																													0			

This register sets the alpha blending coefficient.

**BLP (Broken Line Pattern)**

Register address	DrawBaseAddress + 48C <sub>H</sub>																															
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																															
Initial value	0																															

This register sets 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 line pattern for 1 pixel line is laid out in the direction of MSB to LSB and when it reaches LSB, it goes back to MSB. The BLPO register manages the bit numbers of the broken-line pattern. 32 or 24 bits can be selected as the repetition of the broken-line pattern by 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 <sub>H</sub>																															
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	BC8/16																															
R/W	RW																															
Initial value	0																															

This register sets the border color for texture mapping.

8 bit color mode:

- Bit 7 to 0 BC8 (Border Color)  
Sets the 8-bit direct color for the texture border color

16 bit color mode:

- Bit 15 to 0 BC16 (Border Color)  
Sets the 16-bit direct color for the texture border color

**BLPO (Broken Line Pattern Offset)**

Register address	DrawBaseAddress + 3E0 <sub>H</sub>																															
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			
Initial value																													11111			

This register stores the bit number of the broken-line pattern set to BLP registers, for broken line drawing. This value is decremented at each pixel drawing. Broken line can be drawn starting from any starting position of the specified broken-line pattern by setting any value at this register.

When no write is performed, the position of broken-line pattern is sustained.

**PNBPI (Pixel Number of Broken line pattern Pointer Inter lock)**

Register address	DrawBaseAddress + 28C <sub>H</sub>																															
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																													PN			
R/W																													W			
Initial value																													00000			

This register is valid when BC(16bit)=1 of the GMDR1E register, and determines how many pixels should be fixed before and behind reference address of broken-line pattern(broken-line pointer). The recommended value is same as the line width.



**(Color setting 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 <sub>H</sub>	0	0	0	0	0	0	0	0	Int								Frac															
dRdx	0044 <sub>H</sub>	S	S	S	S	S	S	S	S	Int								Frac															
dRdy	0048 <sub>H</sub>	S	S	S	S	S	S	S	S	Int								Frac															
Gs	004C <sub>H</sub>	0	0	0	0	0	0	0	0	Int								Frac															
dGdx	0050 <sub>H</sub>	S	S	S	S	S	S	S	S	Int								Frac															
dGdy	0054 <sub>H</sub>	S	S	S	S	S	S	S	S	Int								Frac															
Bs	0058 <sub>H</sub>	0	0	0	0	0	0	0	0	Int								Frac															
dBdx	005C <sub>H</sub>	S	S	S	S	S	S	S	S	Int								Frac															
dBdy	0060 <sub>H</sub>	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 enabled in the Gouraud shading mode.

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

**(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	0080 <sub>H</sub>	0	Int								Frac																						
dZdx	0084 <sub>H</sub>	S	Int								Frac																						
dZdy	0088 <sub>H</sub>	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 coordinates for 3D triangle drawing

Zs	Z coordinate start position of long edge
dZdx	Z DDA value of horizontal direction
dZdy	Z DDA value of long edge

**(Texture coordinates-setting 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 <sub>H</sub>	S	S	S																															
dSdx	00c4 <sub>H</sub>	S	S	S																															
dSdy	00c8 <sub>H</sub>	S	S	S																															
Ts	00cc <sub>H</sub>	S	S	S																															
dTdx	00d0 <sub>H</sub>	S	S	S																															
dTdy	00d4 <sub>H</sub>	S	S	S																															
Qs	00d8 <sub>H</sub>	0	0	0	0	0	0	0	0	Int																									
dQdx	00dc <sub>H</sub>	S	S	S	S	S	S	S	S	Int																									
dQdy	00e0 <sub>H</sub>	S	S	S	S	S	S	S	S	Int																									

- 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 coordinates parameters for triangle drawing

Ss	S texture coordinates (Xs, Ys, Zs) of long edge corresponding to Ys
dSdx	S DDA value of horizontal direction
dSdy	S DDA value of long edge direction
Ts	T texture coordinates (Xs, Ys, Zs) of long edge corresponding to Ys
dTdx	T DDA value of horizontal direction
dTdy	T DDA value of long edge direction
Qs	Q (Perspective correction value) of texture at (Xs, Ys, Zs) of long edge corresponding to Ys
dQdx	Q DDA value of horizontal direction
dQdy	Q DDA value of long edge direction



### 10.2.8 Pixel drawing registers

Each register is used by the drawing commands. The registers cannot be accessed from the CPU or 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 <sub>H</sub>	0	0	0	0	Int												0															
PYdc	0184 <sub>H</sub>	0	0	0	0	Int												0															
PZdc	0188 <sub>H</sub>	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 coordinates parameter for drawing pixel. The foreground color is used.

PXdc	Sets X coordinates position
PYdc	Sets Y coordinates position
PZdc	Sets Z coordinates position

### 10.2.9 Rectangle drawing registers

Each register is used by the drawing commands. The registers cannot be accessed from the CPU or 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 <sub>H</sub>	0	0	0	0	Int												0															
RYs	0204 <sub>H</sub>	0	0	0	0	Int												0															
RsizeX	0208 <sub>H</sub>	0	0	0	0	Int												0															
RsizeY	020C <sub>H</sub>	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 coordinates parameters for rectangle drawing. The foreground color is used.

RXs	Sets the X coordinates of top left vertex
RYs	Sets the Y coordinates of top left vertex
RsizeX	Sets horizontal size
RsizeY	Sets vertical size



## 10.2.10 Blt registers

Sets the parameters of each register as described below:

- 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 except 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 <sub>H</sub>	0	0	0	0	0	0	0	Address																								
SStride	0244 <sub>H</sub>	0	0	0	0	Int										0																	
SRXs	0248 <sub>H</sub>	0	0	0	0	Int										0																	
SRYs	024C <sub>H</sub>	0	0	0	0	Int										0																	
DADDR	0250 <sub>H</sub>	0	0	0	0	0	0	0	Address																								
DStride	0254 <sub>H</sub>	0	0	0	0	Int										0																	
DRXs	0258 <sub>H</sub>	0	0	0	0	Int										0																	
DRYs	025C <sub>H</sub>	0	0	0	0	Int										0																	
BRsizeX	0260 <sub>H</sub>	0	0	0	0	Int										0																	
BRsizeY	0264 <sub>H</sub>	0	0	0	0	Int										0																	
TColor	0280 <sub>H</sub>	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 rectangle area in byte address
SStride	Sets stride of source
SRXs	Sets X coordinates start position of source rectangle area
SRYs	Sets Y coordinates start position of source rectangle area
DADDR	Sets start address of destination rectangle area in byte address
DStride	Sets stride of destination
DRXs	Sets X coordinates start position of destination rectangle area
DRYs	Sets Y coordinates start position of destination rectangle area
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 lower 8 bits.

### 10.2.11 High-speed 2D line drawing 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 <sub>H</sub>	0	0	0	0	Int												0															
LY0dc	0544 <sub>H</sub>	0	0	0	0	Int												0															
LX1dc	0548 <sub>H</sub>	0	0	0	0	Int												0															
LY1dc	054C <sub>H</sub>	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 coordinates of line end points for High-speed 2DLine drawing

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

### 10.2.12 High-speed 2D triangle drawing registers

Each register is used by the drawing commands. The registers cannot be accessed from the CPU or 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	0580h	0	0	0	0	Int												0															
Y0dc	0584h	0	0	0	0	Int												0															
X1dc	0588h	0	0	0	0	Int												0															
Y1dc	058ch	0	0	0	0	Int												0															
X2dc	0590h	0	0	0	0	Int												0															
Y2dc	0594h	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 coordinates of three vertices for High-speed 2D Triangle drawing

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

### 10.2.13 Geometry control register

#### GCTR (Geometry Control Register)

Register address	GeometryBaseAddress + 00H																															
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				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									
Initial value	X				0	X		011111				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.

Bit 1 and 0 PS (Pixel engine Status)

Indicates status of pixel engine unit

- 00 Idle
- 01 Processing
- 10 Reserved
- 11 Reserved

Bit 5 and 4 SS (geometry Setup engine Status)

Indicates status of geometry setup engine unit

- 00 Idle
- 01 Processing
- 10 Processing
- 11 Reserved

Bit 9 and 8 GS (Geometry engine Status)

Indicates status of geometry engine unit

- 00 Idle
- 01 Processing
- 10 Reserved
- 11 Reserved

Bit 12 FE (FIFO Empty)

Indicates whether the data is contained in display list FIFO (DFIFOD)

- 0 Data in DFIFOD
- 1 No data in DFIFOD

Bit 13 FF (FIFO Full)

Indicates whether display list FIFO (DFIFOD) is full or not

- 0 DFIFOD not full
- 1 DFIFOD full

- Bit 14      NF (FIFO Near Full)  
Indicates free space in display list FIFO (DFIFOD)  
0      More than half of DFIFOD free  
1      Less than half of DFIFOD free
- Bit 20 to 15    FCNT (FIFO Counter)  
Indicates count of free stages (0 to 100000<sub>B</sub>) of display list FIFO (DFIFOD)
- Bit 24      FO (FIFO Overflow)  
Indicates whether FIFO overflow occurred  
0      Normal  
1      FIFO overflow

### 10.2.14 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 <sub>H</sub>																															
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	RWRWRWRW		RWRWRWRW				
Initial value																								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 / Packed RGB format  
 1    Reserved

- Bit 6 and 5    DF (Data Format)  
 Specifies vertex coordinates 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	Fixed-point format + packed RGB format
	10	Reserved
	11	Packed integer format + packed RGB format
1	00	Reserved
	01	Reserved
	10	Reserved
	11	Reserved

- Bit 3      ST (texture S and T data enable)  
Sets whether to use texture ST coordinates  
0      Not use texture ST coordinates  
1      Uses texture ST coordinates
- Bit 2      Z (Z data enable)  
Sets whether to use Z coordinates  
0      Not use Z coordinates  
1      Uses Z coordinates
- Bit 1      C (Color data enable)  
Sets whether to use vertex color  
0      Not use vertex color  
1      Uses 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 <sub>H</sub>																															
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																												W	W	W		
Initial value																												0	0	0		

This register sets the geometry mode at line drawing. This register is sharing hardware with GMDR1E, so that if GMDR1 is changed, the same bit of GMDR1E is also changed.

Bit 4      BO (Broken line Offset)  
 Sets broken line reference position  
 0      Broken line reference position not cleared  
 1      Broken line reference position cleared

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

Bit 0      AA (Anti-alias mode)  
 Sets anti-alias mode  
 0      Anti-alias not performed  
 1      Anti-alias performed



**GMDR1E (Geometry Mode Register for Line Extension)**

Register address	(SetGModeRegister)																															
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	P0											TC		BC		UW			BM		TM		BP		SP		BO		EP		AA	
R/W	W											W		W		W			W		W		W		W		W		W			
Initial value	0											0		0		0			0		0		0		0		0		0			

This register sets the geometry processing extended mode at line drawing.

The CORAL extended function can be used only when the C, Z, and ST fields of GMDR0 are "0".

This register is sharing hardware with GMDR1, so that if GMDR1E is changed, the same bit of GMDR1 is also changed.

- Bit31      P0 (Primitive Order control)

Sets the drawing control mode for the main, the border, and the shadow primitive.

Recommend to set main bit=1 in anti-aliasing and blending.

0      Draws the order of, main->border->shadow(performance is regarded as important)

1      Draws the order of , shadow->border->main(blending affect is regarded as important)
  
- Bit30      LV (Line Version control)

Specify the Coral line's algorithm version.

V2.0 is improvement version from V1.0. Recommend V2.0.

0      Version 1.0 ( for backward compatibility )

1      Version 2.0 (Recommended)
  
- Bit 20      TC (Thick line Correct)

Sets the interpolation mode for the bold line joint

0      Interpolation of bold lien joint not performed

1      Interpolation of bold line joint performed (valid for only CORAL line)
  
- Bit 16      BC (Broken line Correct)

Sets the interpolation mode for the dashed-line pattern

0      Interpolation not performed

1      Interpolation performed using dashed-line pattern reference address fixed mode (valid for only CORAL line)
  
- Bit 14      UW (Uniform line Width)

Sets the line width equalization mode

0      Equalization of line width not performed

1      Equalization of lien width performed (valid for only CORAL line)
  
- Bit 13      BM (Broken line Mode)

Sets the dashed-line pattern mode

0      Dashed-line pattern pasted vertical to principal axis of line (compatible with CREMSON) (valid for only CREMSON line)

1      Dashed-line pattern pasted vertical to theoretical line
  
- Bit 12      TM (Thick line Mode)

- Sets the bold line mode
- 0 Bold line drawn vertical to principal axis of line (compatible with CREMSON) (CREMSON line)  
 Operation is not assured when TM=0 is used together with TC=-1, SP=1, or BP=1.
  - 1 Bold line drawn vertical to theoretical line. (CORAL line)  
 Operation is not assured when TM=1 is used together with BM=0.
- Bit 9 BP (Border Primitive)  
 Sets the drawing mode for the border primitive
- 0 Border primitive not drawn
  - 1 Border primitive drawn (valid for only CORAL line)
- Bit 8 SP (Shadow Primitive)  
 Sets the drawing mode for the shadow primitive
- 0 Shadow primitive not drawn
  - 1 Shadow primitive drawn (valid for only CORAL line)
- Bit 4 BO (Broken line Offset)  
 Sets the reference position of the dashed-line pattern
- 0 Reference position of dashed-line pattern cleared
  - 1 Reference position of dashed-line pattern not cleared
- Bit 2 EP (End Point mode)  
 Sets the drawing mode for the end point  
 Note that the end point is always not drawn in line strip(CREMSON line(TN=0))
- 0 End point not drawn
  - 1 End point drawn
- Bit 0 AA (Anti-alias mode)  
 Sets anti-alias mode
- 0 Anti-alias not performed
  - 1 Anti-alias performed

**GMDR2 (Geometry Mode Register for Triangle)**

Register address	GeometryBaseAddress + 48 <sub>H</sub>																															
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																											W	W				
Initial value																											0	0				

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

Drawing performed using commands in range from G\_Begin/G\_BeginCont to G\_End

- Bit 2 FD (Face Definition)  
 Sets the face definition
- 0 Face defined as state with vertexes arranged clockwise
  - 1 Face defined as state with vertexes arranged counterclockwise

Bit 0	CF (Cull Face)
	Sets the drawing mode of the back
0	Back drawn
1	Back not drawn (value disabled for polygons)

**GMDR2E (Geometry Mode Register for Triangle Extension)**

Register address	(SetGModeRegister)																															
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																						TL	SP						FD	CF		
R/W																						W	W						W	W		
Initial value																						0	0						0	0		

This register sets the geometry processing extended mode at triangle drawing.

In case of TL=1 with texture mapping, please set perspective correction.

Non-top-left-part's pixel quality is less than body. (using approximate calculation)

- Bit 10 TL (Top-Left rule mode)  
 Sets the drawing algorithm

  - 0 Top-left rule applied (compatible with CREMSON)
  - 1 Top-left rule not applied
  
- Bit 8 SP (Shadow Primitive)  
 Sets the drawing mode for the shadow primitive

  - 0 Shadow primitive not drawn
  - 1 Shadow primitive drawn
  
- Bit 2 FD (Face Definition)  
 Sets the face definition

  - 0 Face defined as state with vertexes arranged clockwise
  - 1 Face defined as state with vertexes arranged counterclockwise
  
- Bit 0 CF (Cull Face)  
 Sets the drawing mode of the back

  - 0 Back drawn
  - 1 Back not drawn (value disabled for polygons)

### 10.2.15 Display list FIFO registers

#### DFIFOG (Geometry Displaylist FIFO with Geometry)

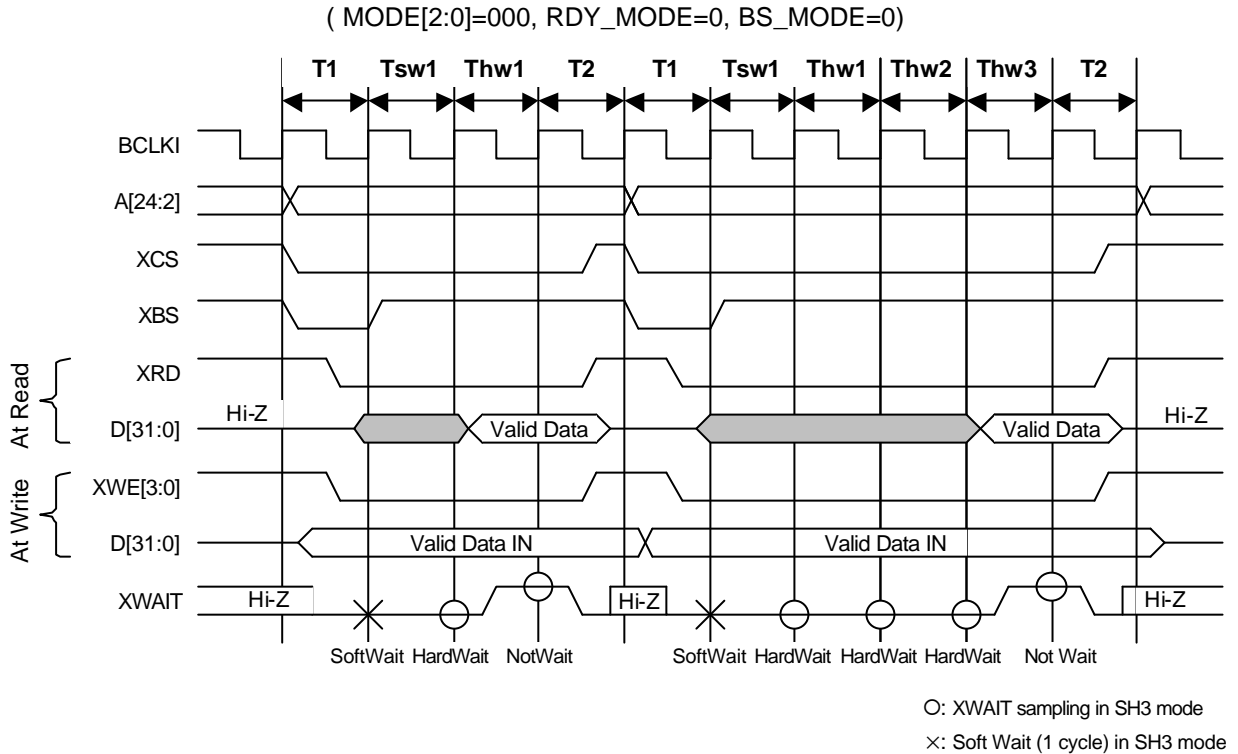
Register address	Geometry BaseAddress + 400 <sub>H</sub>																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DFIFOG																															
R/W	W																															
Initial value	Don't care																															

FIFO registers for Display List transfer

## 11 TIMING DIAGRAM

### 11.1 Host Interface

#### 11.1.1 CPU read/write timing diagram in SH3 mode (Normally Not Ready Mode)

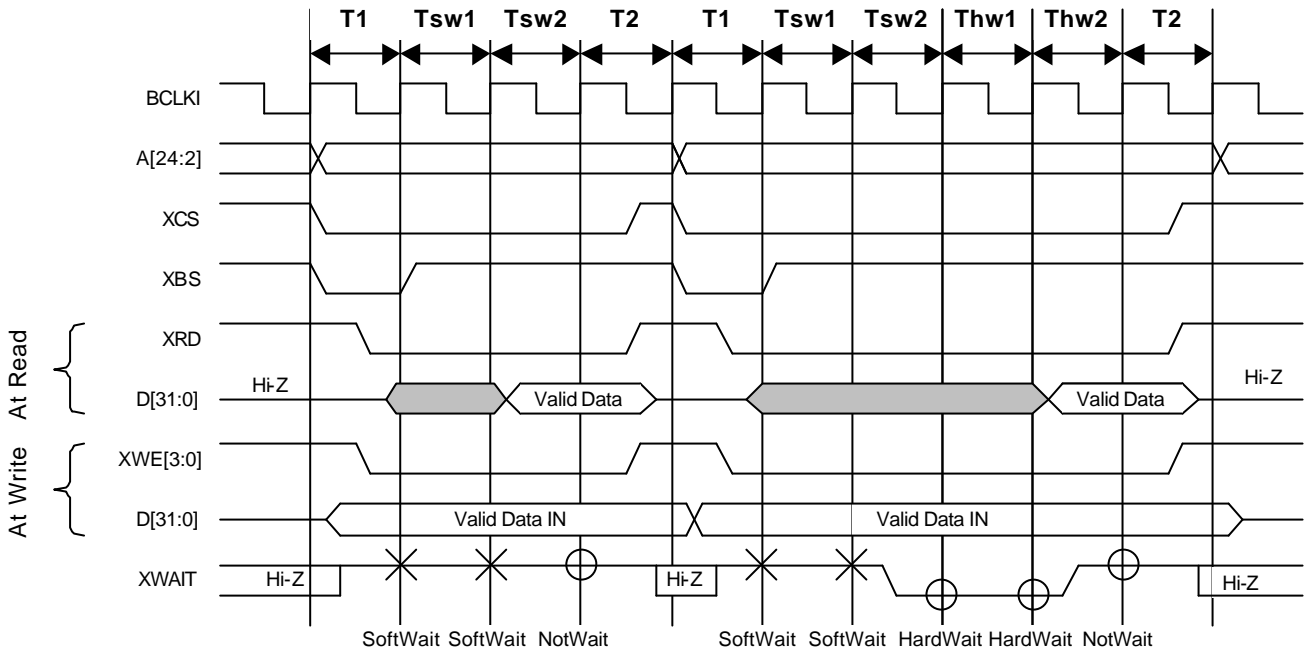


- T1: Read/write start cycle (XRDY in wait state)
- Tsw\*: Software wait insertion cycle (1 cycle setting)
- Thw\*: Hardware wait insertion cycle (XRDY cancels the wait state after the preparations)
- T2: Read/write end cycle (XRDY ends in wait state)

Fig. 10.1 Read/Write Timing Diagram for SH3 (Normally Not Ready Mode)

### 11.1.2 CPU read/write timing diagram in SH3 mode (Normally Ready Mode)

( MODE[2:0]=000, RDY\_MODE=1, BS\_MODE=0)



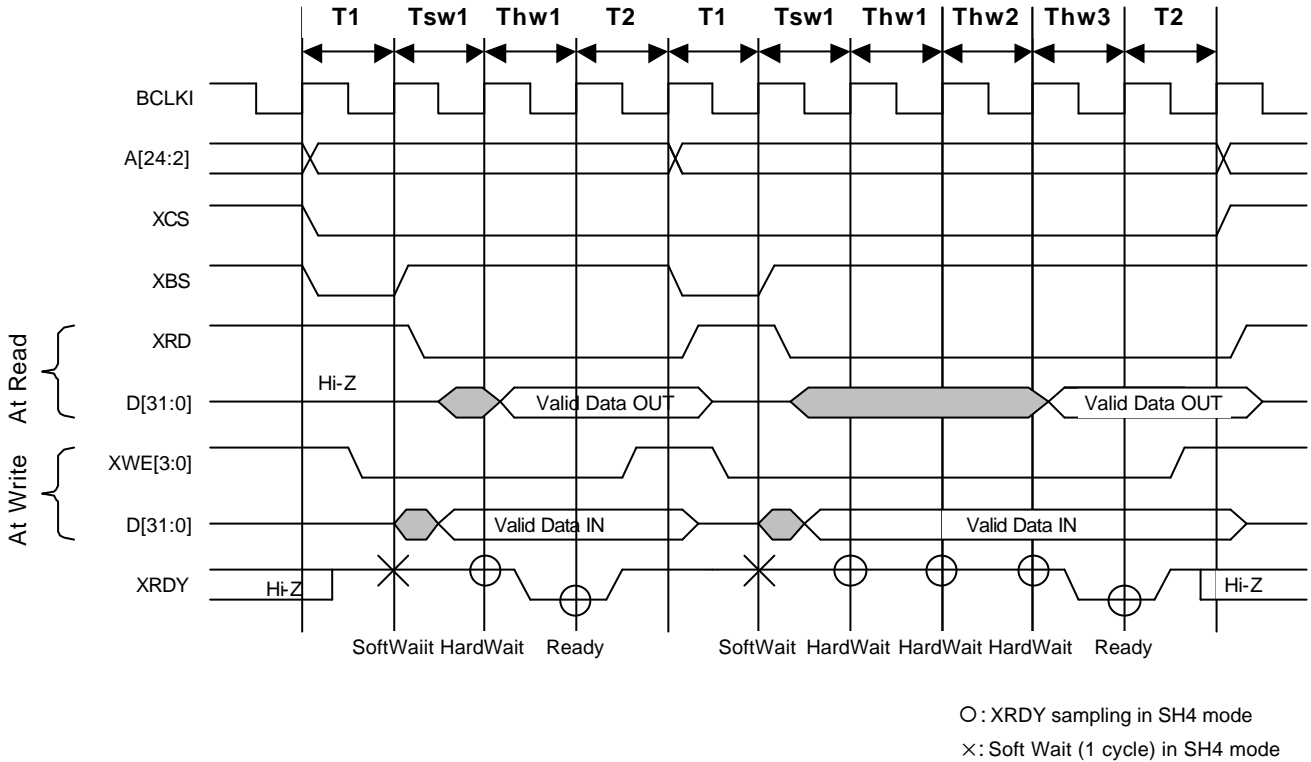
O : XWAIT sampling in SH3 mode  
 X : Soft Wait (2 cycles) in SH3 mode

- T1: Read/write start cycle (XRDY in not wait state)
- Tsw\*: Software wait insertion cycle (2-cycle setting required)
- Thw\*: Hardware wait insertion cycle (In hardware state when the immediate accessing is disabled)
- T2: Read/write end cycle (XRDY ends in not wait state)

Fig. 10.2 Read/Write Timing Diagram for SH3 (Normally Ready Mode)

### 11.1.3 CPU read/write timing diagram in SH4 mode (Normally Not Ready Mode)

( MODE[2:0]=001, RDY\_MODE=0, BS\_MODE=0)



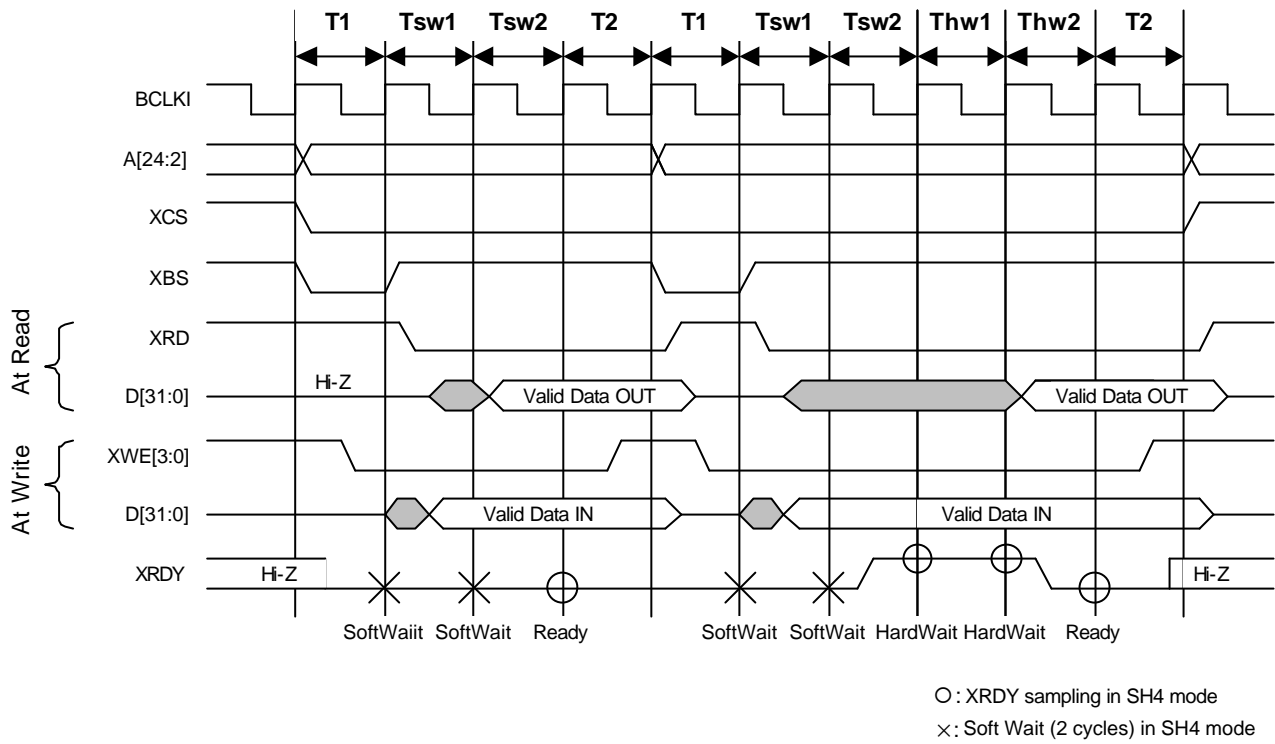
- T1: Read/write start cycle (XRDY in the not ready state)
- Tsw\*: Software wait insertion cycle (1 cycle)
- Thw\*: Hardware wait insertion cycle (XRDY asserts Ready after the preparations)
- T2: Read/write end cycle (XRDY ends in not ready state)

**Fig. 10.3 Read/Write Timing Diagram for SH4 Mode (Normally Not Ready Mode)**



### 11.1.4 CPU read/write timing diagram in SH4 mode (Normally Ready Mode)

( MODE[2:0]=001, RDY\_MODE=1, BS\_MODE=0)

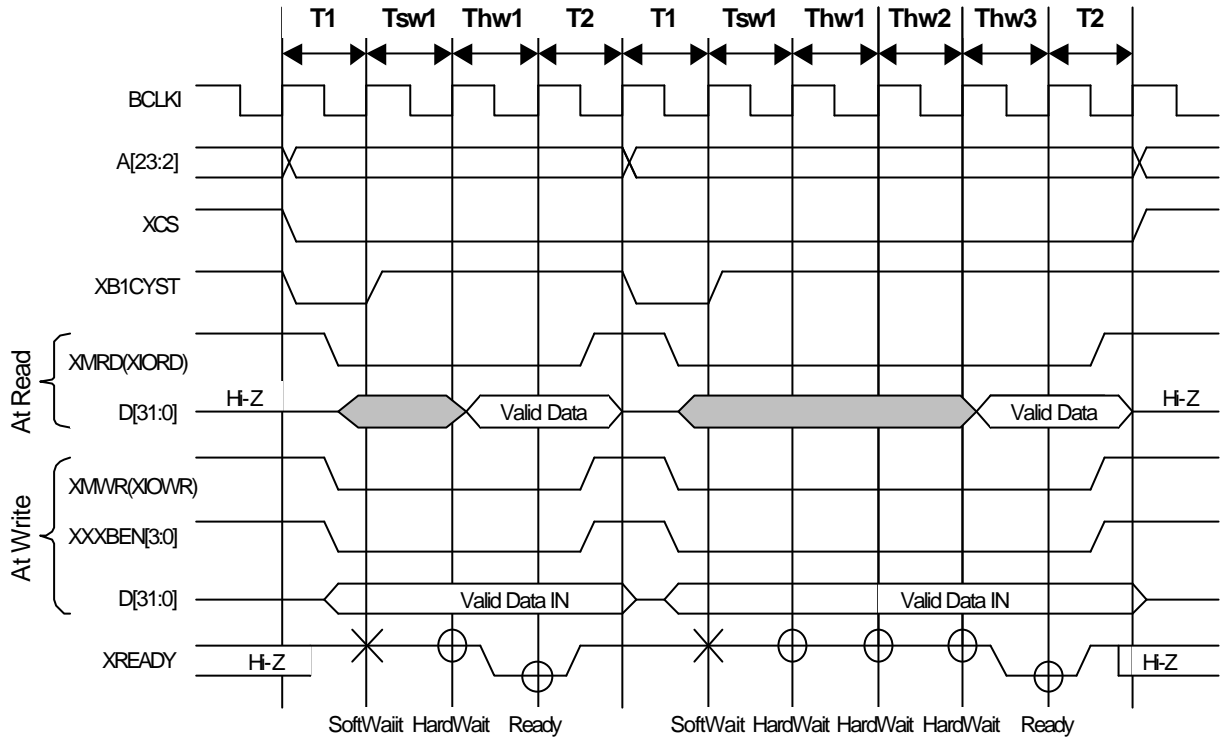


- T1: Read/write start cycle (XRDY in ready state)
- Tsw\*: Software wait insertion cycle (2-cycle setting required)
- Thw\*: Hardware wait insertion cycle (XRDY asserts Ready after the preparations)
- T2: Read/write end cycle (XRDY ends in ready state.)

**Fig. 10.4 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)

( MODE[2:0]=010, RDY\_MODE=0, BS\_MODE=0)



○: XREADY sampling in V832 mode

×: Soft Wait (1 cycle) in V832 mode

T1: Read/write start cycle (XREADY in not ready state)

Tsw\*: Software wait insertion cycle

Twh\*: Hardware wait insertion cycle (XREADY asserts Ready after the preparations)

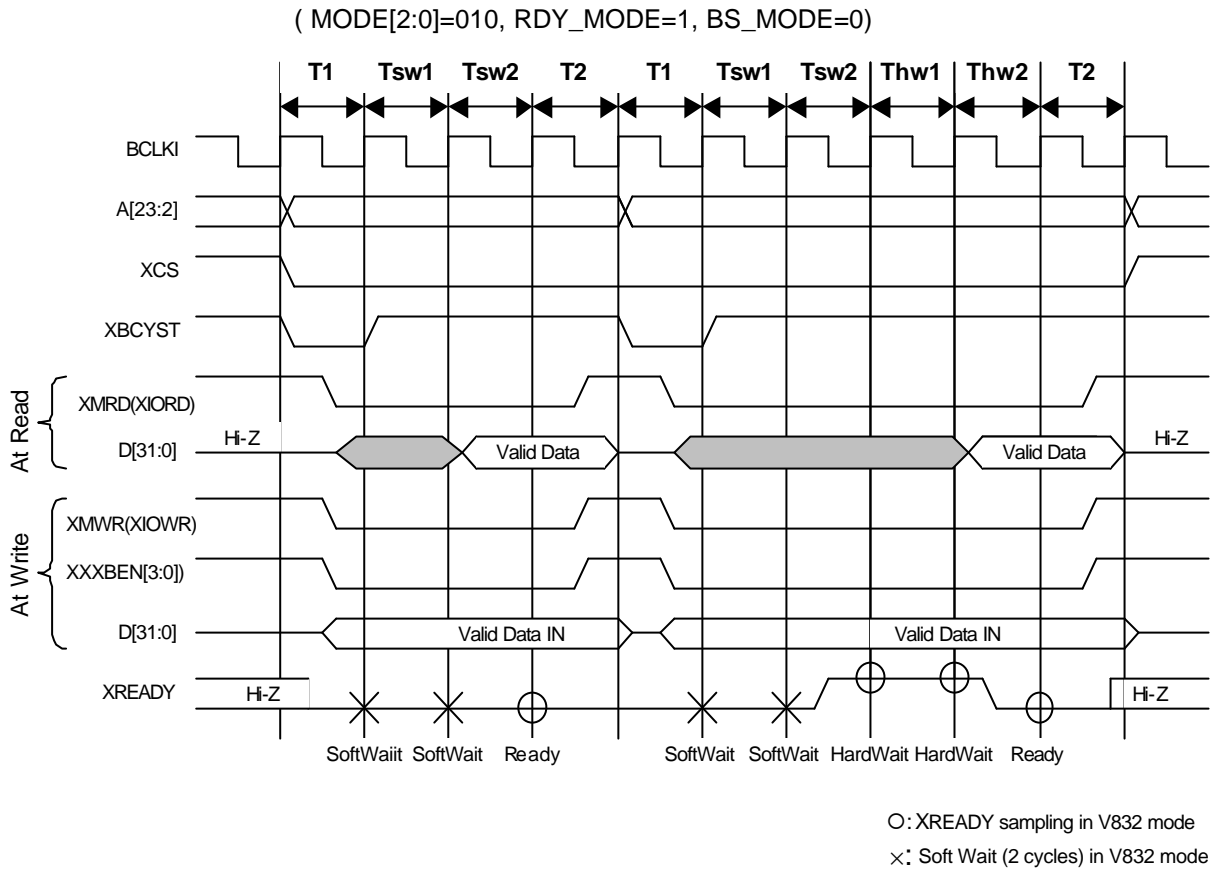
T2: Read/write end cycle (XREADY ends in not ready state)

Notes: 1.The XxxBEN signal is used only for a write from the CPU; it is not used for a read from the CPU.

2.The CPU always inserts one cycle wait after read access.

Fig. 10.5 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 in ready state)

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

Twh\*: Hardware wait insertion cycle (XREADY asserts Ready after the preparations)

T2: Read/write end cycle (XREADY ends in ready state)

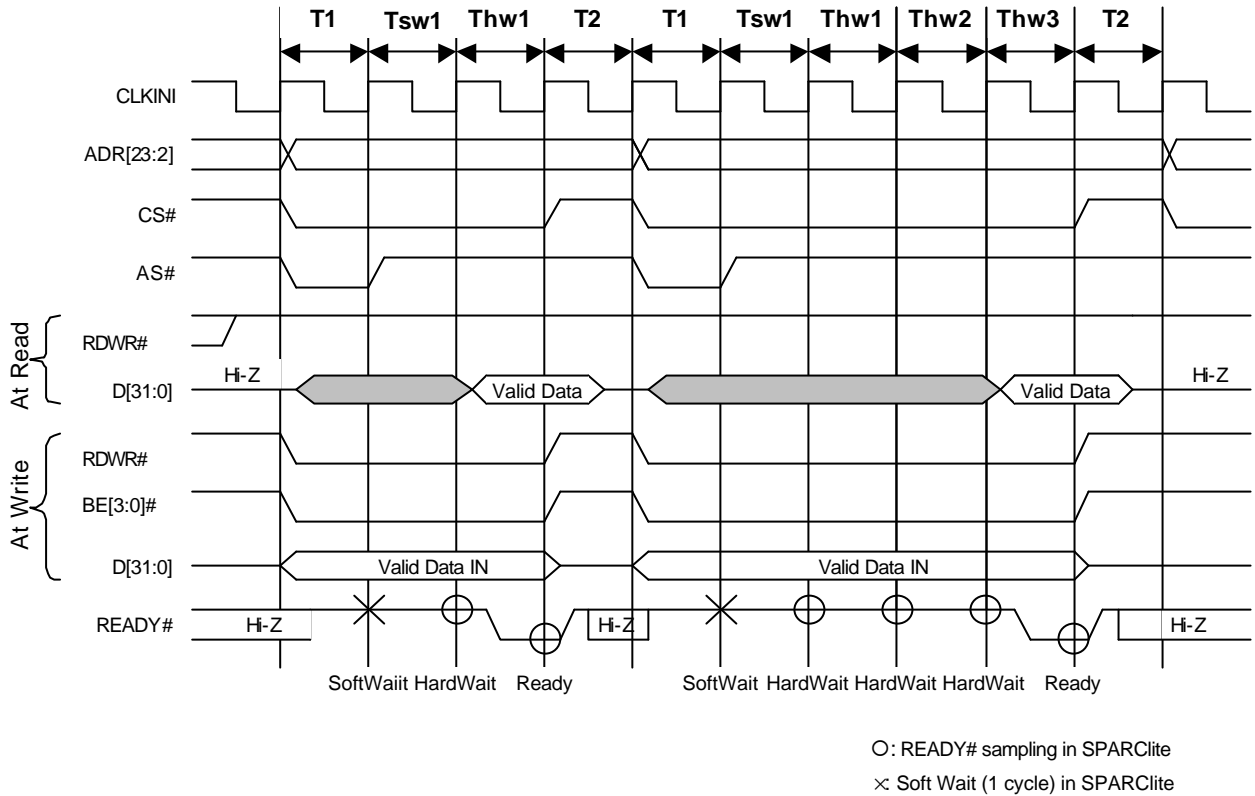
Notes: 1.The XxxBEN signal is used only for a write from the CPU; it is not used for a read from the CPU.

2.The CPU always inserts one cycle wait after read access.

**Fig. 10.6 Read/Write Timing Diagram in V832 Mode (Normally Ready Mode)**

### 11.1.7 CPU read/write timing diagram in SPARClike (Normally Not Ready Mode)

( MODE[2:0]=011, RDY\_MODE=0, BS\_MODE=0)



T1: Read/write start cycle (READY# in not ready state)

Tsw\*: Software wait insertion cycle

Thw\*: Hardware wait insertion cycle (READY# asserts Ready after the preparations)

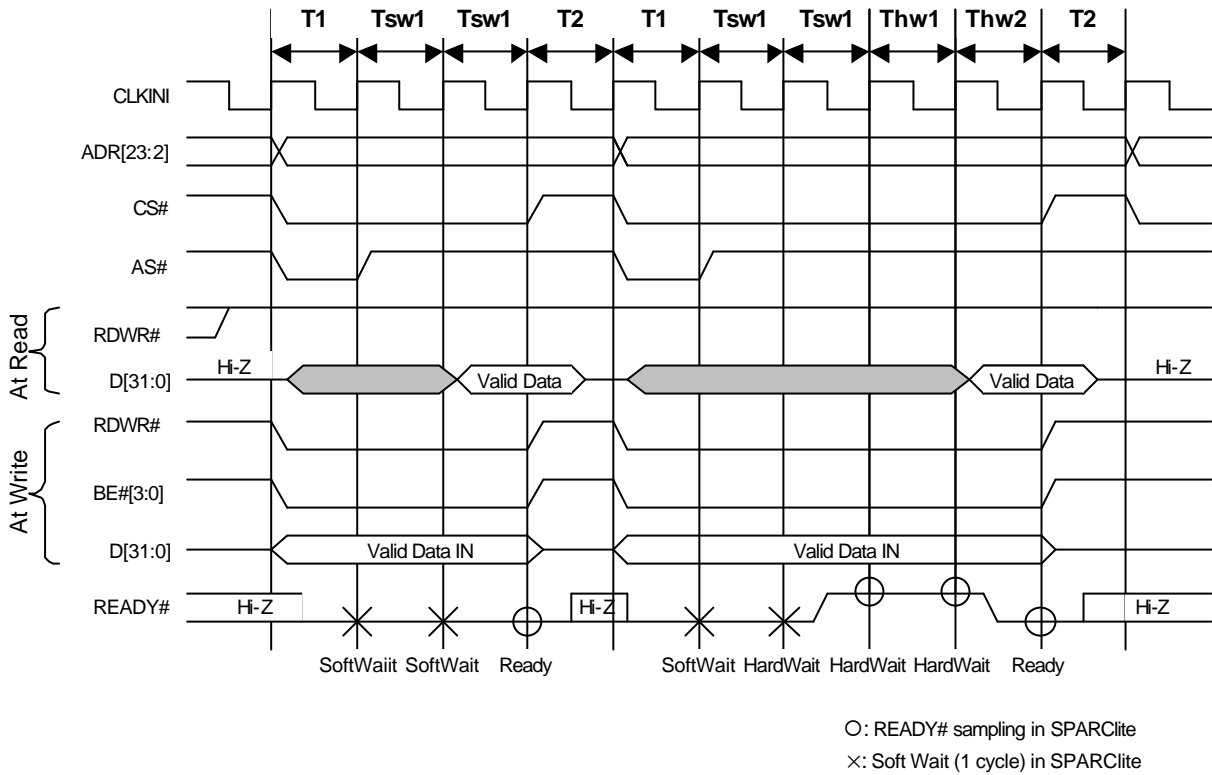
T2: Read/write end cycle (READY# ends in not ready state)

Note: BE# signal is used only for a write from the CPU; it is not used for a read from the CPU.

**Fig. 10.7 Read/Write Timing Diagram in SPARClike (Normally Not Ready Mode)**

### 11.1.8 CPU read/write timing diagram in SPARClike (Normally Ready Mode)

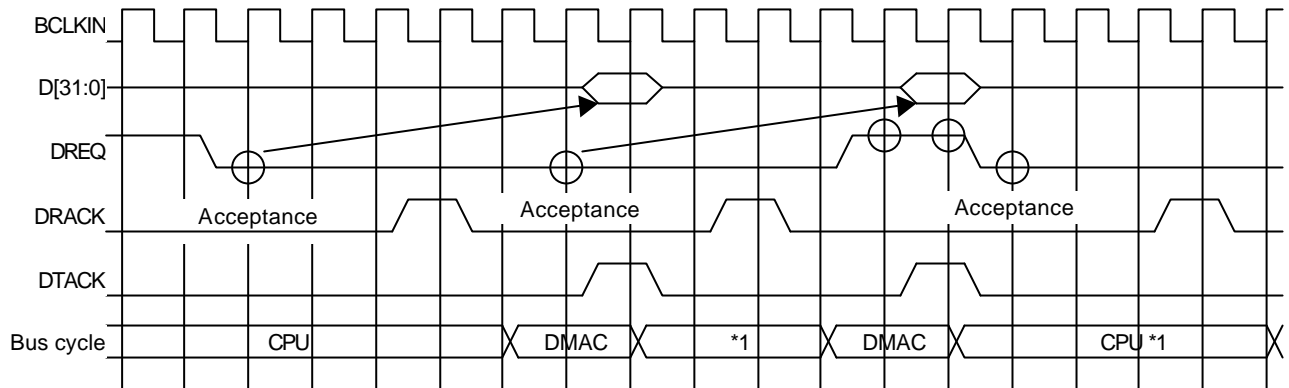
( MODE[2:0]=011, RDY\_MODE=1, BS\_MODE=0)



- T1: Read/write start cycle (READY# in ready state)
  - Tsw\*: Software wait insertion cycle (2-cycle setting required)
  - Thw\*: Hardware wait insertion cycle (READY# asserts Ready after the preparations)
  - T2: Read/write end cycle (READY# ends in ready state)
- Note: BE# signal is used only for a write from the CPU; it is not used for a read from the CPU.

**Fig. 10.8 Read/Write Timing Diagram in SPARClike (Normally Ready Mode)**

### 11.1.9 SH4 single-address DMA write (transfer of 1 long word)

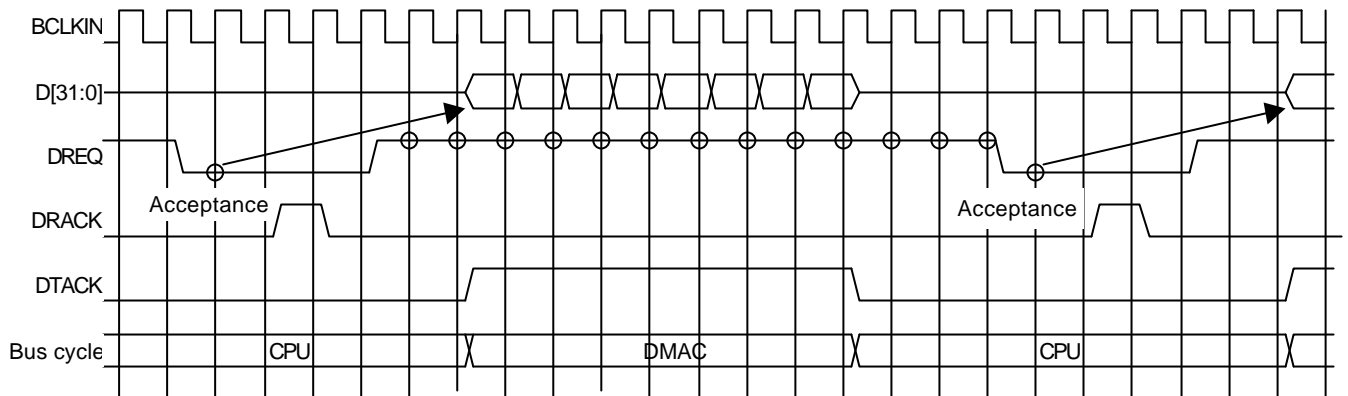


- : DREQ sampling and channel priority determination for SH mode (DREQ = level detection)
- \*1: In the cycle steal mode, even when DREQ is already asserted at the 2nd DREQ sampling, the right to use the bus is returned to the CPU temporarily. In the burst mode, DMAC secures the right to use the bus unless DREQ is negated.

**Fig. 10.9 SH4 Single-address DMA Write (Transfer of 1 Long Word)**

CORAL 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 reception is ready.

### 11.1.10 SH4 single-address DMA write (transfer of 8 long words)

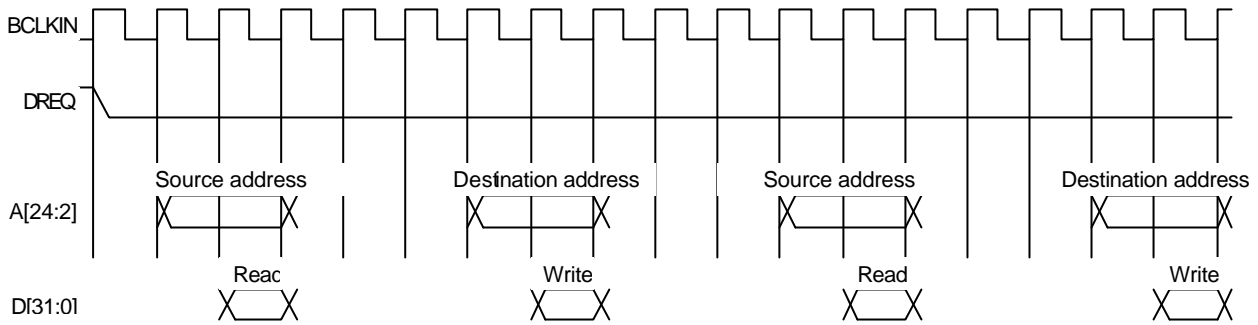


○: DREQ sampling and channel priority determination for SH mode (DREQ = level detection)

**Fig. 10.10 SH4 Single-address DMA Write (Transfer of 8 Long Words)**

After the CPU has asserted DRACK, CORAL negates DREQ and receives 32-byte data in line with the DTACK assertion timing. As soon as the next data is ready to be received, CORAL reasserts DREQ but the reassertion timing depends on the internal status.

### 11.1.11 SH3/4 dual-address DMA (transfer of 1 long word)

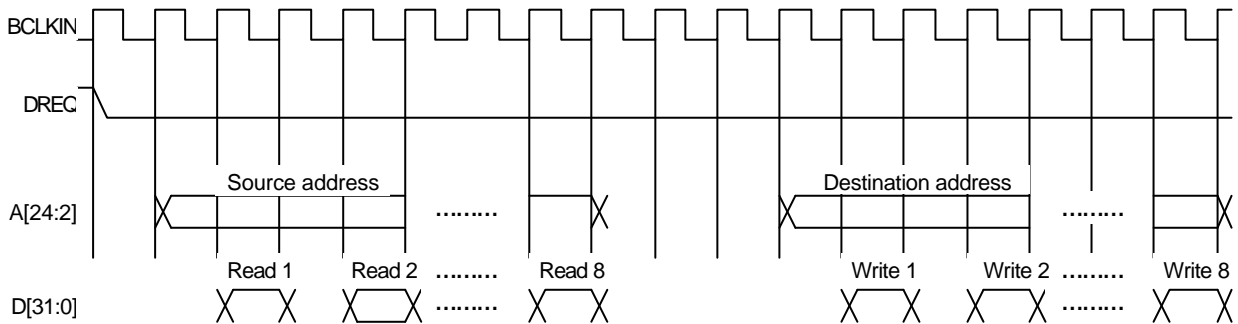


For the CORAL, the read/write operation is performed according to the SRAM protocol.

**Fig. 10.11 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, when CORAL cannot return the ready signal immediately, in order to negate the DREQ signal set the DBM register.

### 11.1.12 SH3/4 dual-address DMA (transfer of 8 long words)



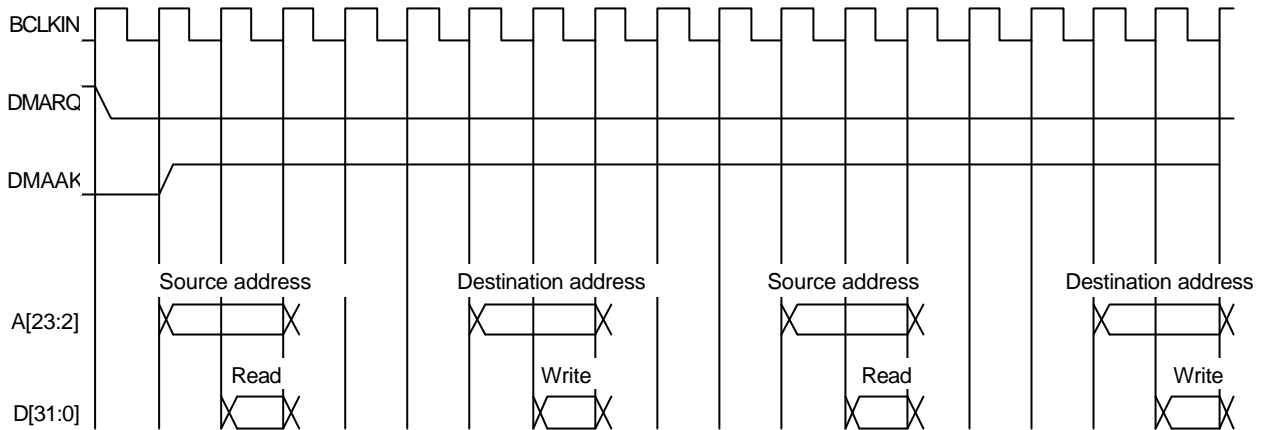
For the CORAL, the read/write operation is performed according to the SRAM protocol.

**Fig. 10.12 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, when CORAL cannot return the ready signal immediately, in order to negate the DREQ signal set the DBM register.



### 11.1.13 V832 DMA transfer

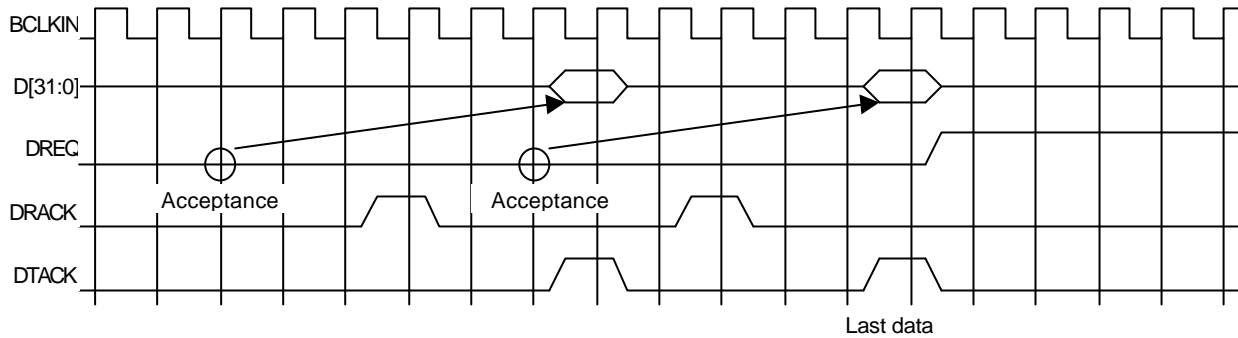


For the CORAL, the read/write operation is performed according to the SRAM protocol.

**Fig. 10.13 V832 DMA Transfer**

In the dual-address mode, the DREQ signal is kept asserted until the transfer ends by default. Consequently, when CORAL cannot return the ready signal immediately, in order to negate the DREQ signal set the DBM register.

### 11.1.14 SH4 single-address DMA transfer end timing

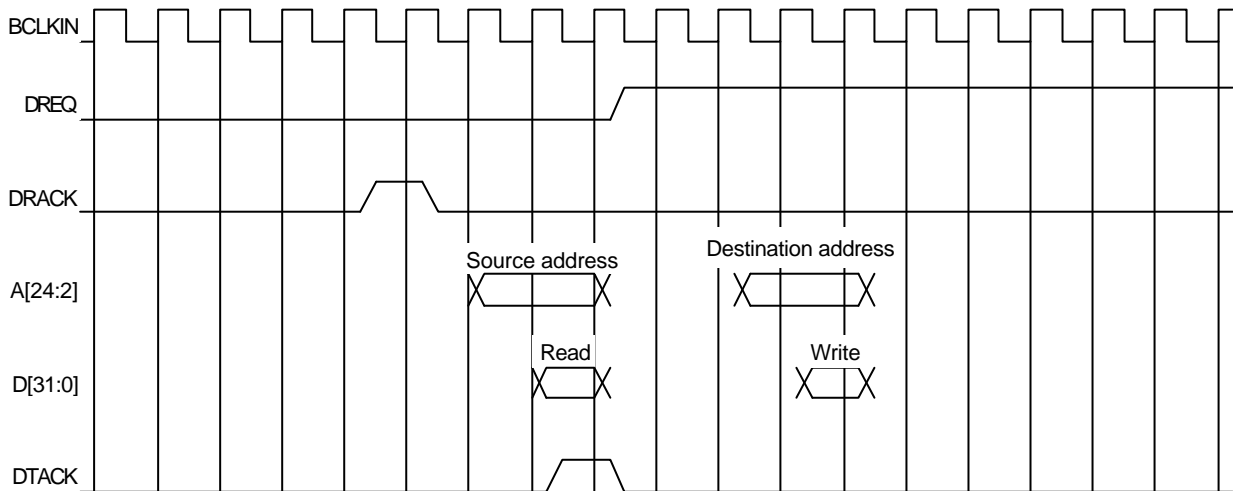


O: DREQ sampling and channel priority determination for SH mode (DREQ = level detection)

**Fig. 10.14 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



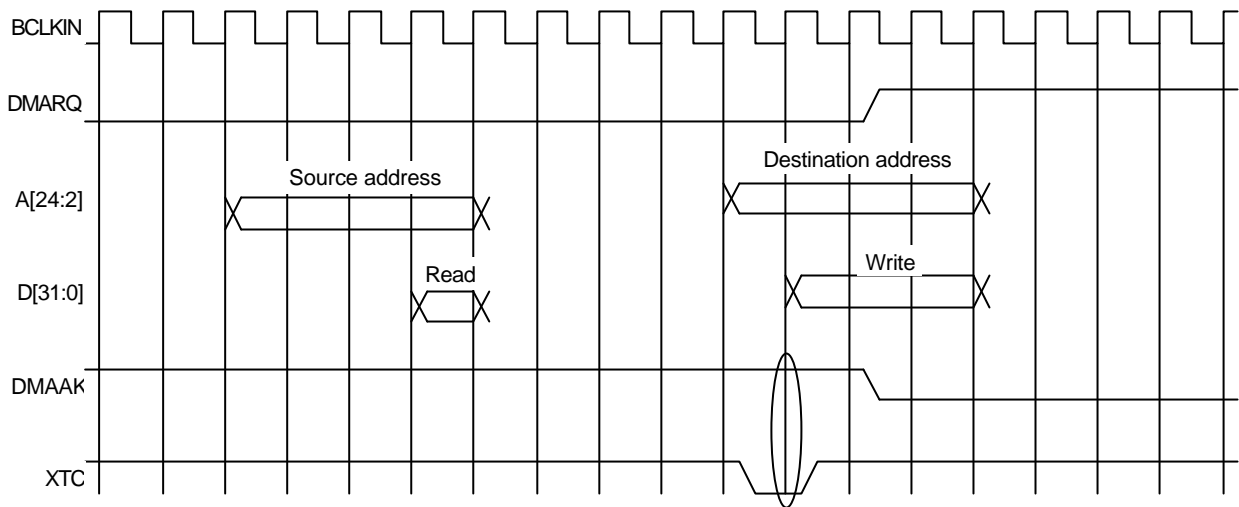
For the CORAL, the read/write operation is performed according to the SRAM protocol.

**Fig. 10.15 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

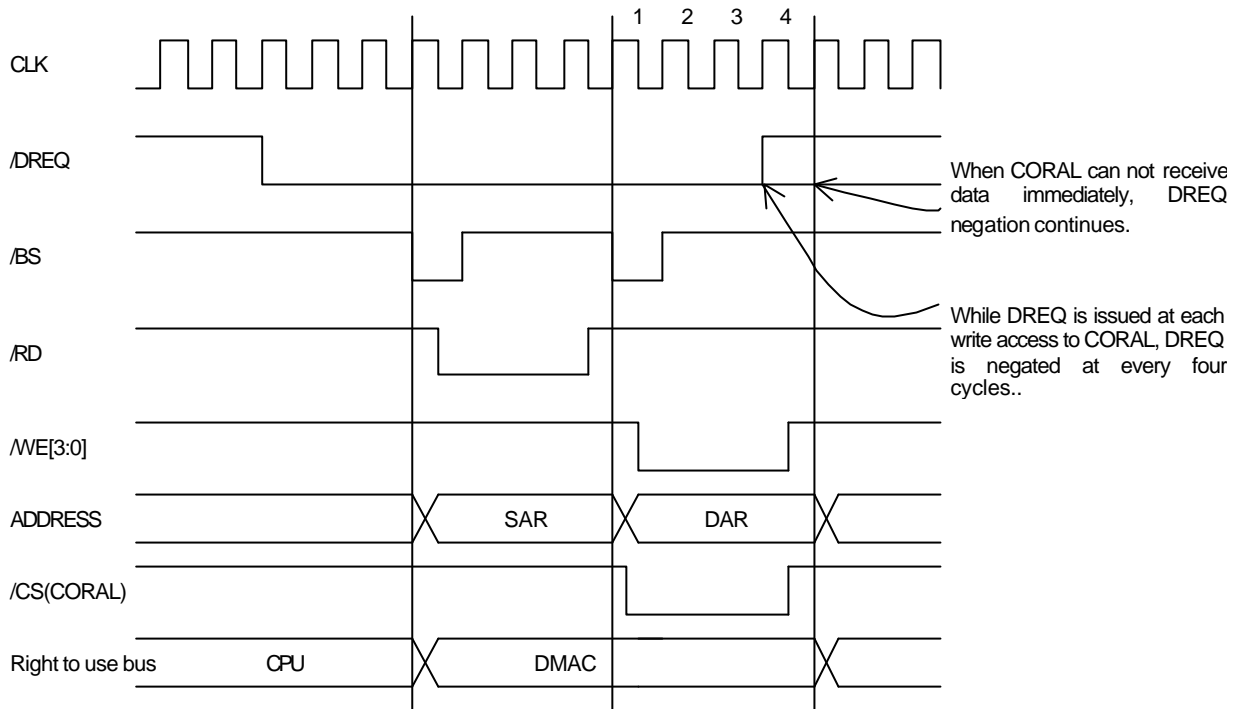


For the CORAL, the read/write operation is performed according to the SRAM protocol.

**Fig. 10.16 V832 DMA Transfer End Timing**

DMAAK and XTC are logic ANDed inside CORAL to end DMA.

### 11.1.17 SH4 dual DMA write without ACK



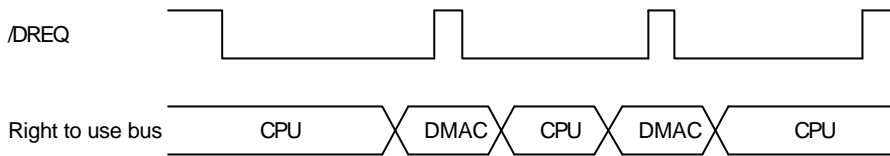
**Fig. 10.17 DREQ Negate Timing for Each Transfer**

At each DMA transfer, DREQ is negated and then reasserted at the next cycle.

Only the FIFO address can be used as the destination address.

When CORAL cannot receive data immediately, DREQ negation continues. At that time, the negate timing is not only above diagram.

### 11.1.18 Dual-address DMA (without ACK) end timing



**Fig. 10.18 Dual-address DMA (without ACK) End Timing**

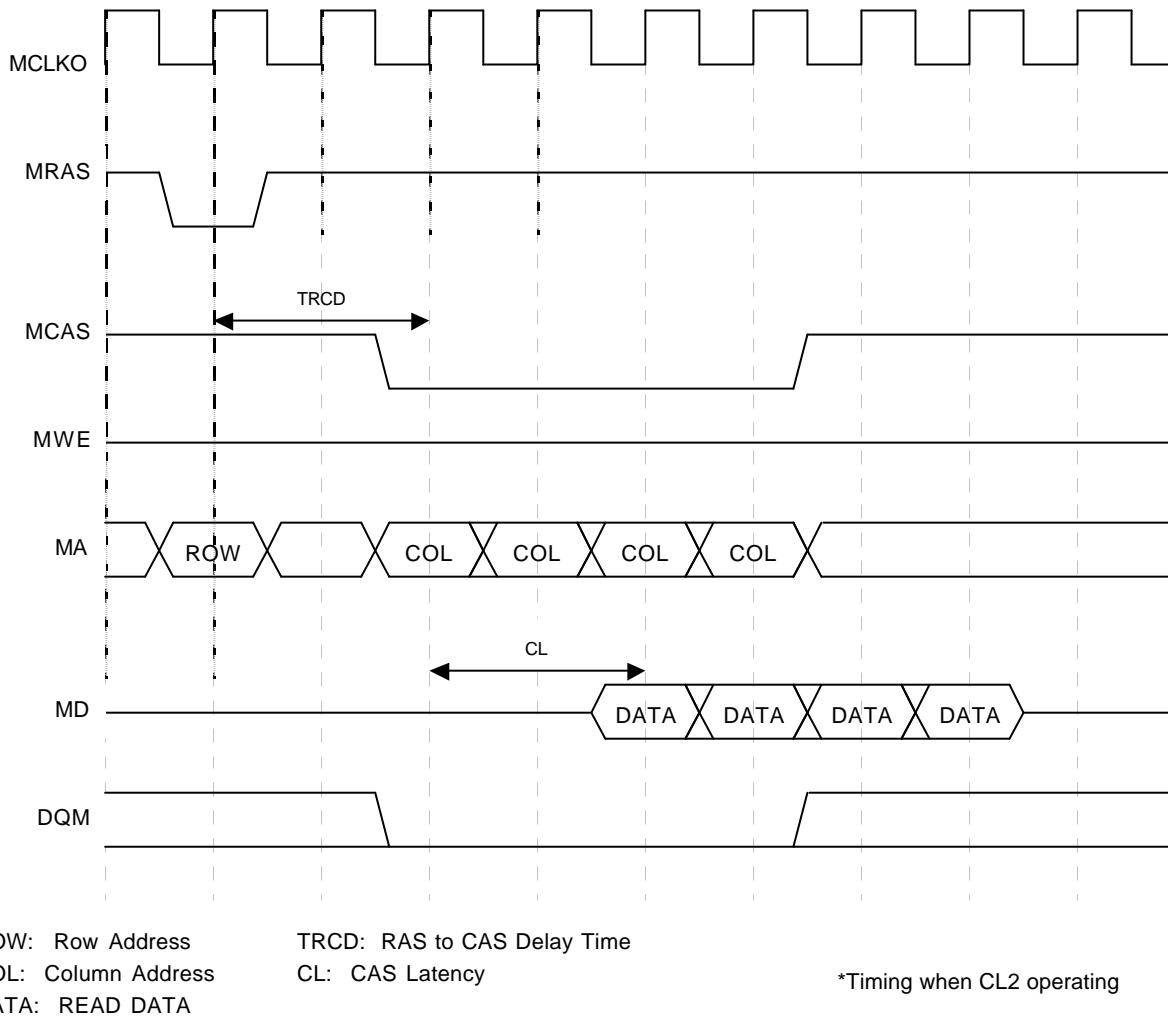
Example: DMA operation when DMA transfer performed twice

- (1) The CPU accesses the DREQ issue register (DRQ) of Coral to issue DREQ.
- (2) The right to use bus is transferred from the CPU to the DMAC.
- (3) In the first DMAC cycle, write is performed to CORAL and DREQ is negated; DREQ is reasserted in the next cycle.
- (4) The right to use bus is returned to the CPU and the DREQ edge is detected, so the right to use bus is transferred to the DMAC.
- (5) The second write operation is performed and DREQ is negated, but DREQ is reasserted because CORAL does not recognize that the transfer has ended.
- (6) The right to use bus is transferred to the CPU, so the CPU writes to the DTS register of CORAL to negate DREQ.

## 11.2 Graphics Memory Interface

The CORAL access timing and graphics memory access timing are explained here.

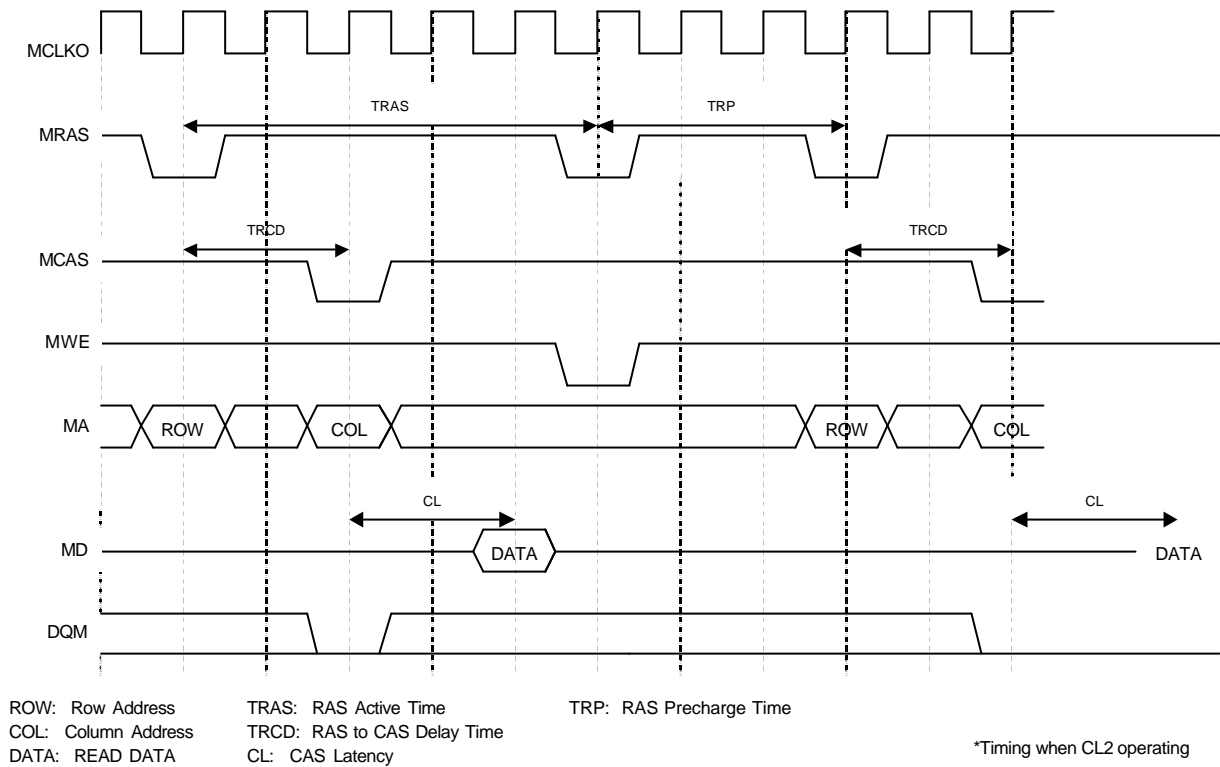
### 11.2.1 Timing of read access to same row address



**Fig. 10.19 Timing of Read Access to Same Row Address**

The above timing diagram shows that read access is made four times from CORAL to the same row address of SDRAM. The **ACTV** command is issued and then the **READ** command is issued after TRCD elapses. Then data that is output after the elapse of CL after the **READ** command is issued is captured into CORAL.

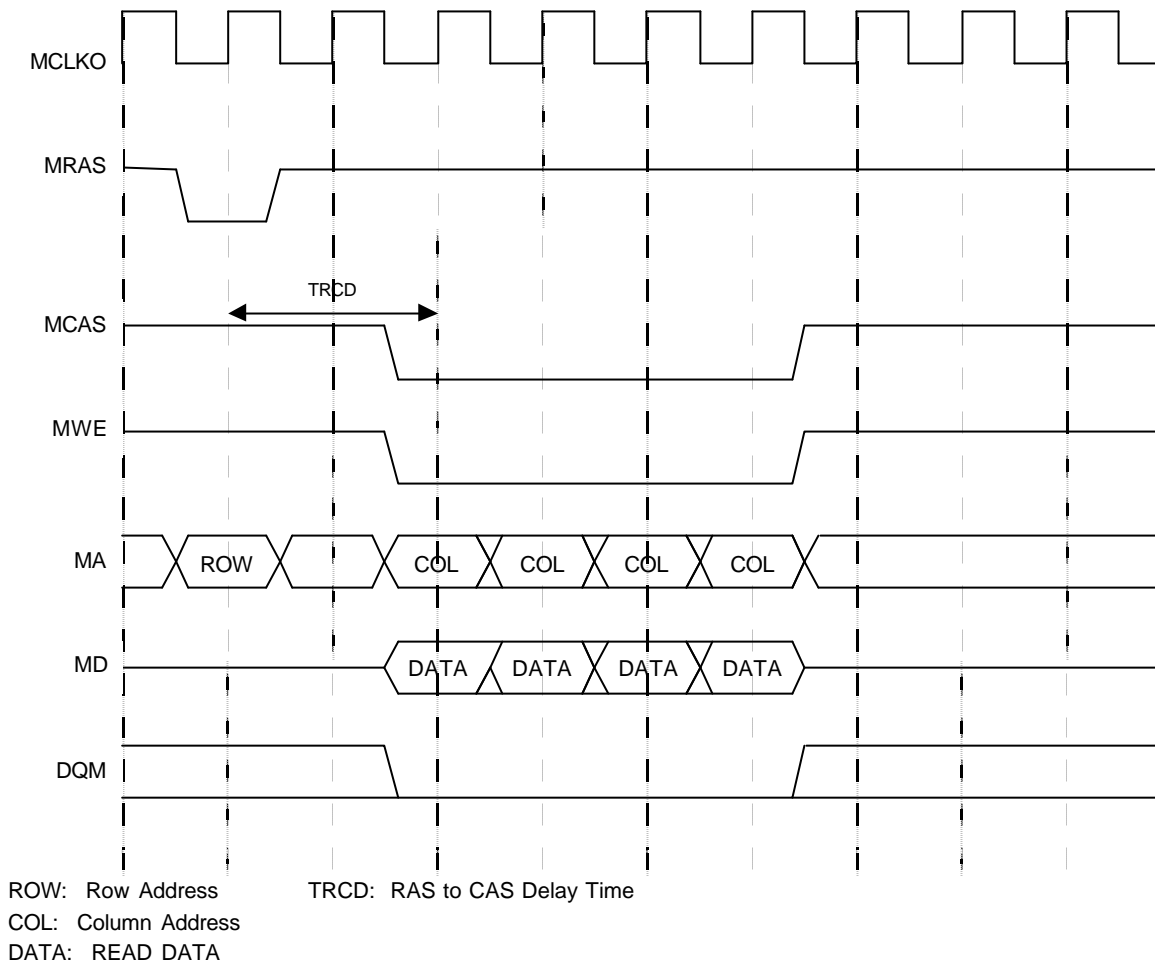
### 11.2.2 Timing of read access to different row addresses



**Fig. 10.20 Timing of Read Access to Different Row Addresses**

The above timing diagram shows that read access is made from CORAL to different row addresses of SDRAM. The first and next address to be read fall across an SDRAM page boundary, so the **Pre-charge** command is issued at the timing satisfying TRAS, and then after the elapse of TRP, the **ACTV** command is reissued, and then the **READ** command is issued.

### 11.2.3 Timing of write access to same row address



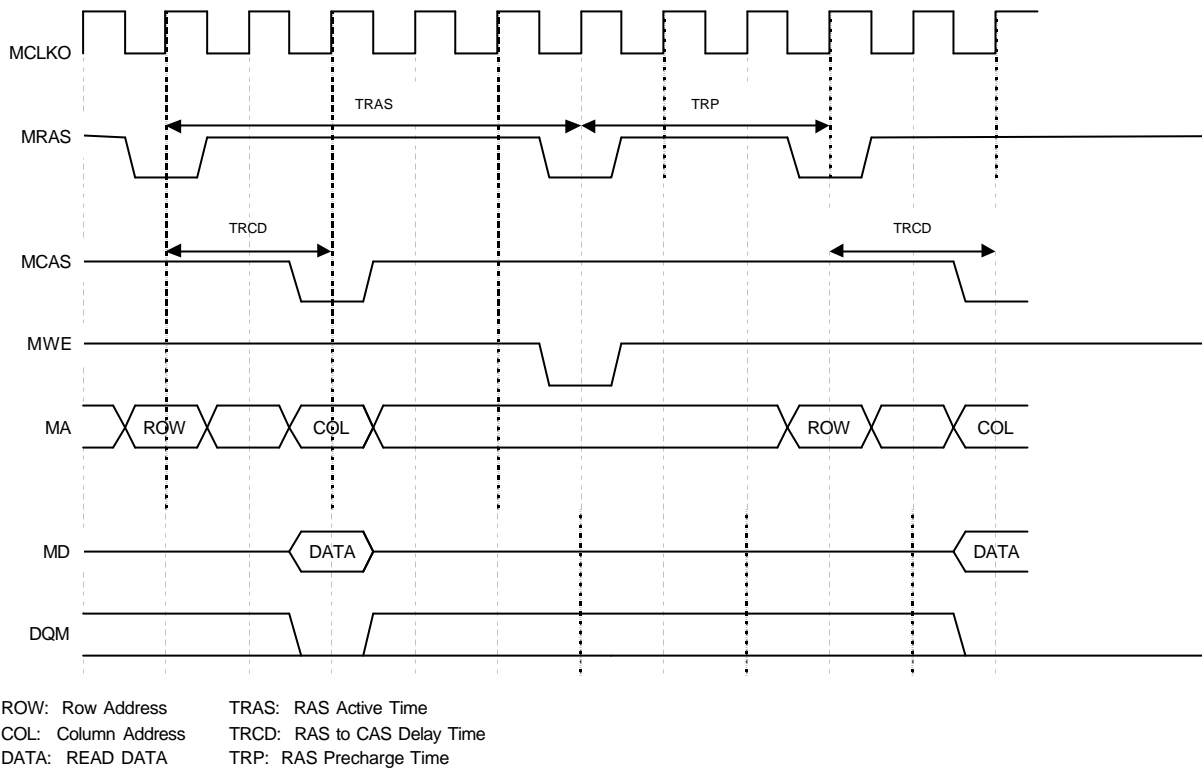
**Fig. 10.21 Timing of Write Access to Same Row Address**

The above timing diagram shows that write access is made from times from CORAL to the same row address of SDRAM.

The **ACTV** command is issued, and then after the elapse of TRCD, the **WRITE** command is issued to write to SDRAM.



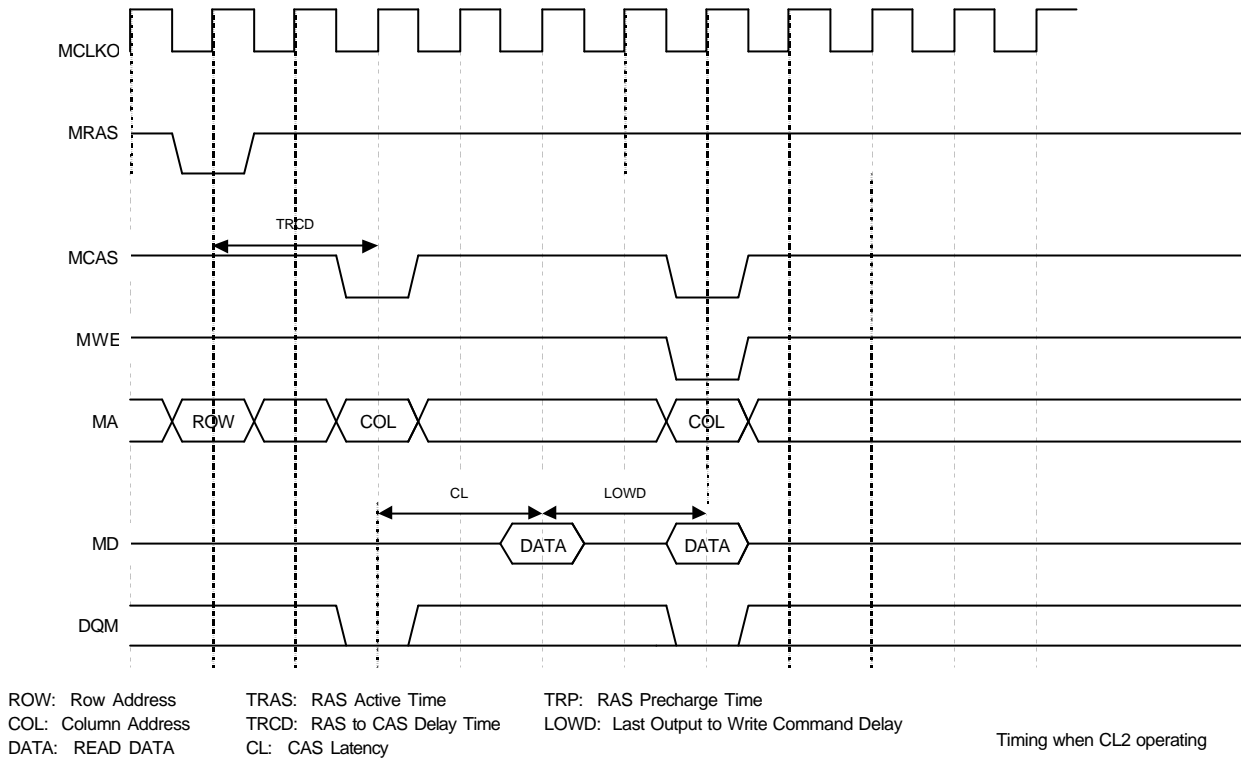
### 11.2.4 Timing of write access to different row addresses



**Fig. 10.22 Timing of Write Access to Different Row Addresses**

The above timing diagram shows that write access is made from CORAL to different row addresses of SDRAM. The first and next address to be write fall across an SDRAM page boundary, so the **Pre-charge** command is issued at the timing satisfying TRAS, and then after the elapse of TRP, the **ACTV** command is reissued, and then the **WRITE** command is issued.

### 11.2.5 Timing of read/write access to same row address

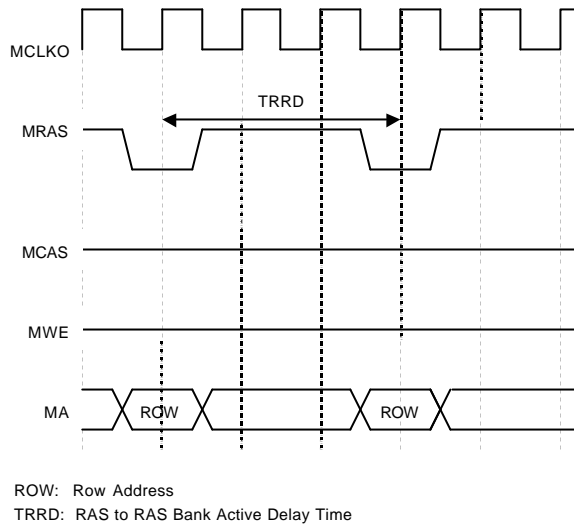


**Fig. 10.23 Timing of Read/Write Access to Same Row Address**

The above timing diagram shows that write access is made immediately after read access is made from CORAL to the same row address of SDRAM.

Read data is output from SDRAM, LOWD elapses, and then the **WRITE** command is issued.

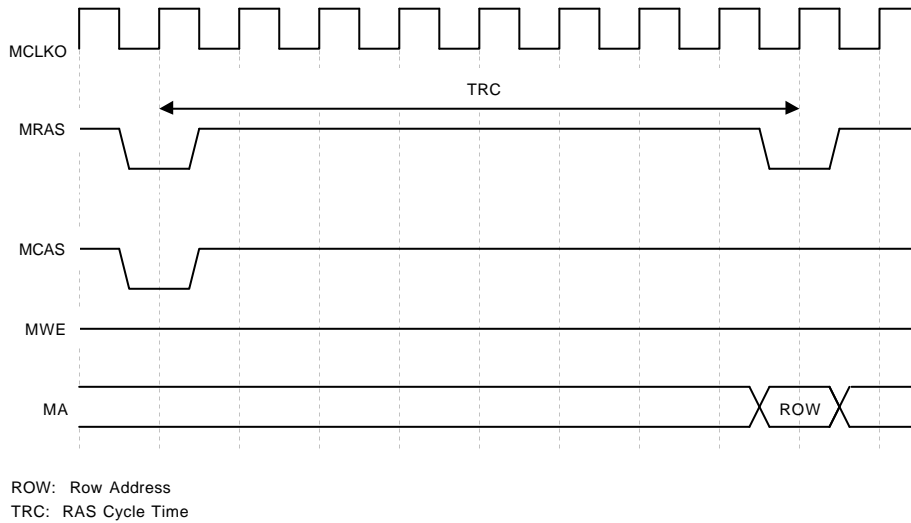
### 11.2.6 Delay between ACTV commands



**Fig.10.24 Delay between ACTV Commands**

The ACTV command is issued from CORAL to the row address of SDRAM after the elapse of **TRRD** after issuance of the previous **ACTV** command.

### 11.2.7 Delay between Refresh command and next ACTV command

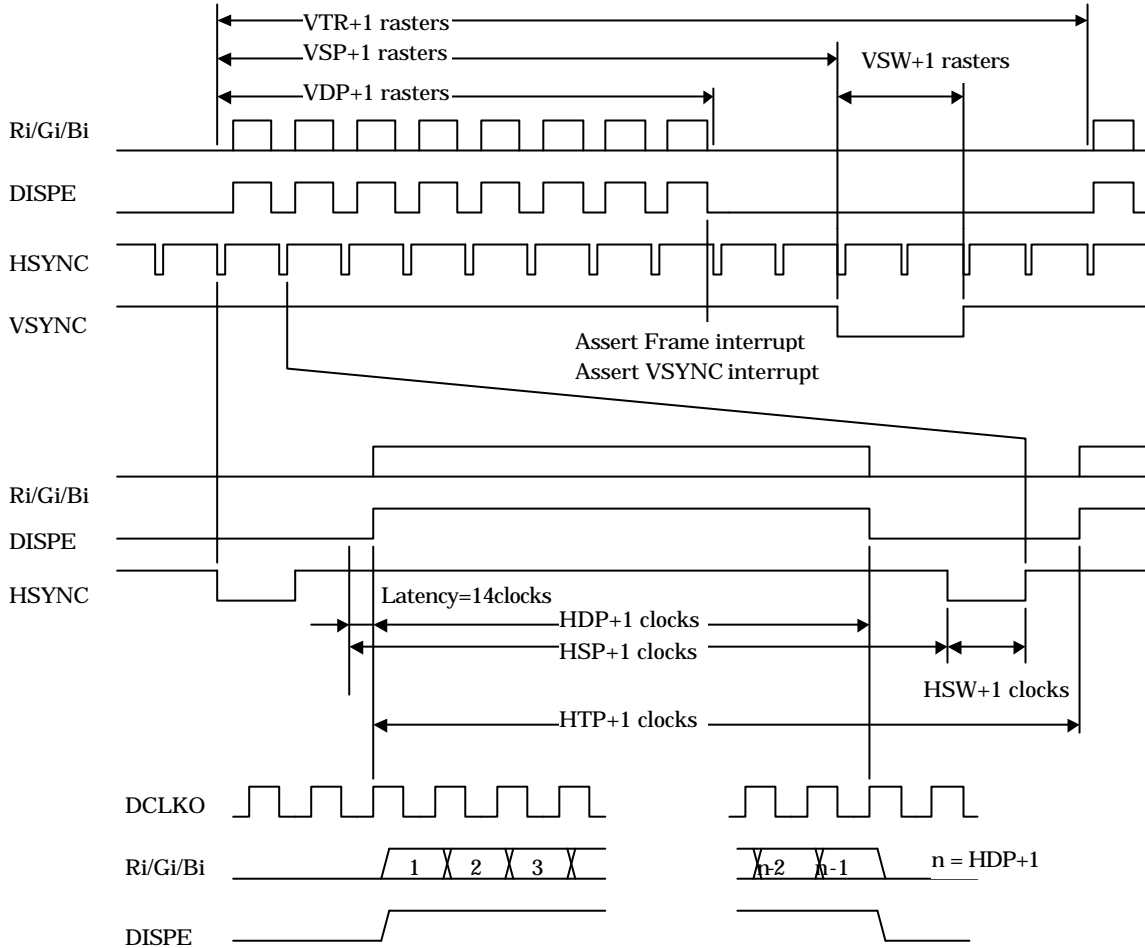


**Fig. 10.25 Delay between Refresh Command and Next ACTV Command**

The **ACTV** command is issued after the elapse of TRC after issuance of the **Refresh** command.

## 11.3 Display Timing

### 11.3.1 Non-interlace mode

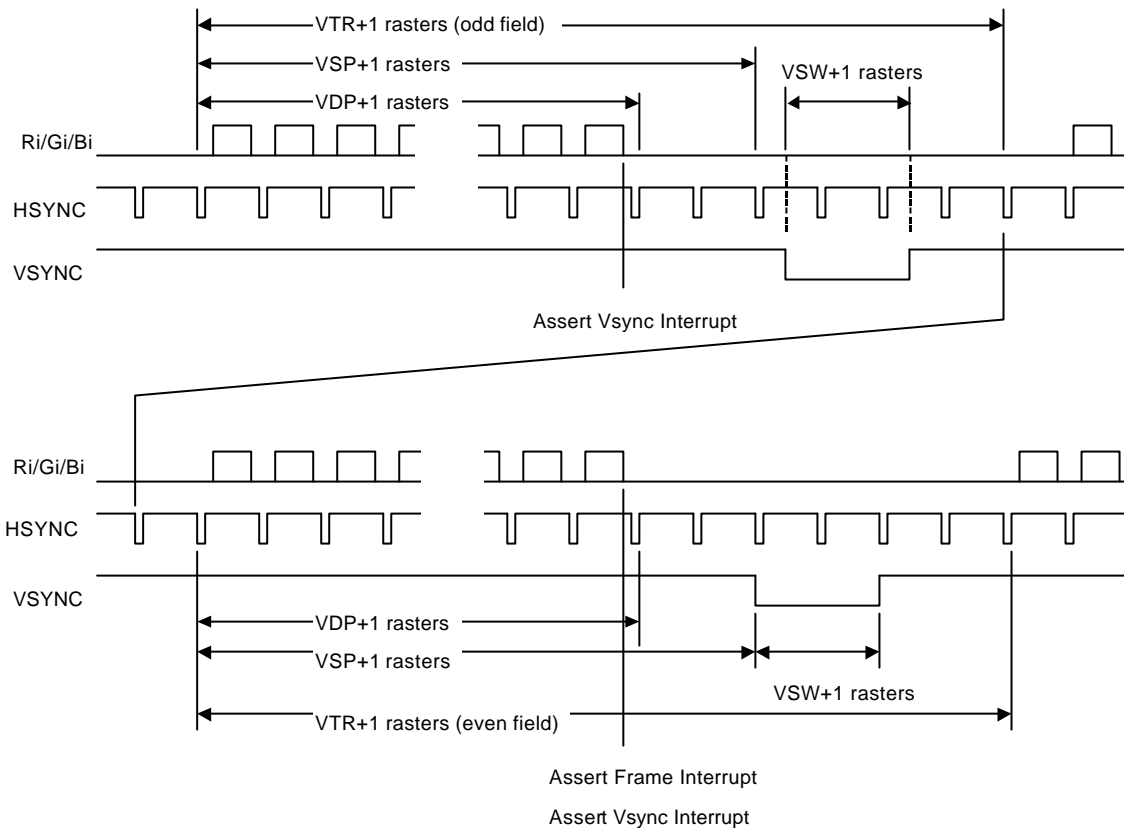


**Fig. 10.26 Non-interlace Timing**

In the above diagram, VTR, HDP, etc., are the setting values 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 Interlace video mode



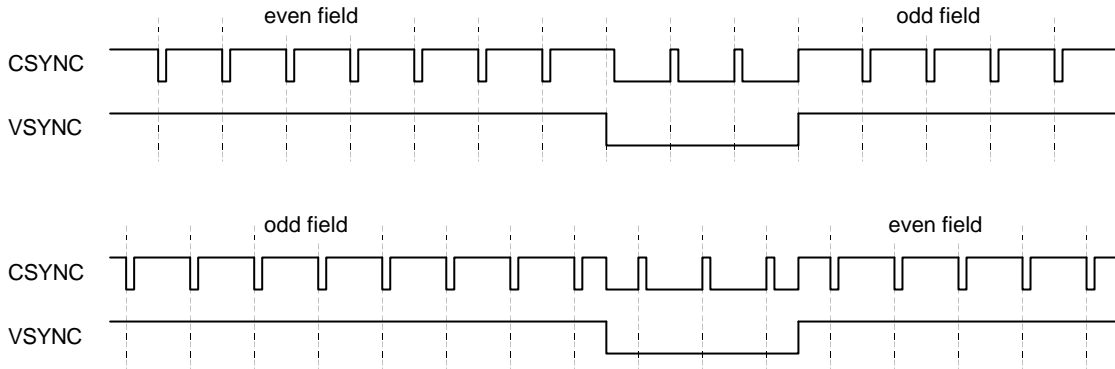
**Fig. 10.27 Interlace Video Timing**

In the above diagram, VTR, HDP, etc., are the setting values of their associated registers.

The interlace mode also operates at the same timing as the interlace video mode. The only difference between the two modes is the output image data.

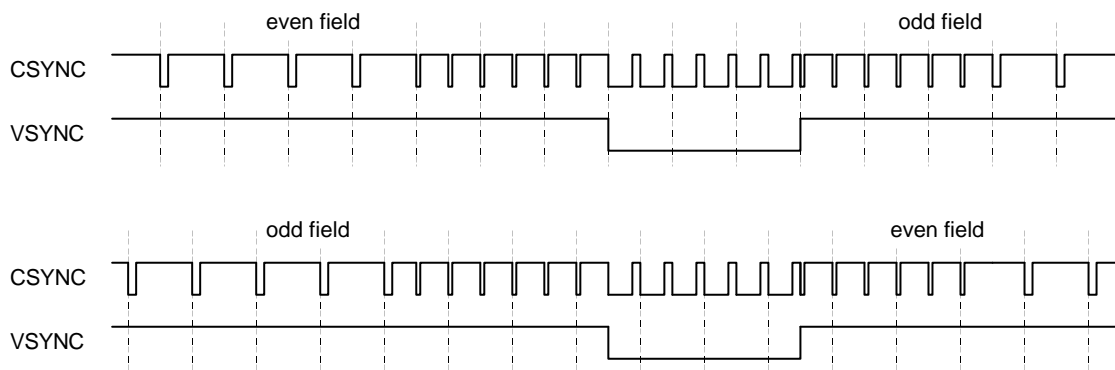
### 11.3.3 Composite synchronous signal

When the EEQ bit of the DCM register is “0”, the CSYNC signal output waveform is as shown below.



**Fig 10.28 Composite Synchronous Signal without Equalizing Pulse**

When the EEQ bit of the DCM register is “1”, the equalizing pulse is inserted into the CSYNC signal, producing the waveform shown below.



**Fig 10.29 Composite Synchronous Signal with Equalizing Pulse**

The equalizing pulse is inserted when the vertical blanking time period starts. It is also inserted three times after the vertical synchronization time period has elapsed.

## CAUTIONS

### 11.4 CPU Cautions

- 1) Enable the hardware wait for the areas to which CORAL is connected. When the normally not ready mode (RDY\_MODE = 0) is used, set the software wait count to "1". When the normally ready mode (RDY\_MODE = 1) is used, set the count to "2". When the normally ready mode is used (RDY\_MODE = 1) and BS\_MODE = L, set the software wait to 2. When the normally ready mode is enabled and **BS\_MODE = H**, set the software wait to "3".
- 2) When starting DMA by issuing an external request, do so after setting the transfer count register (DTCR) and mode setting register (DSUR) of CORAL to the same value as the CPU setting. In the dual DMA without ACK mode or V832 mode, there is no need to set DTCR.
- 3) When CORAL 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) Set DREQ (DMARQ) to "Low" level detection.
- 5) Set the DACK/DRACK of SH to high active output, DMAAK of V832 to high active, and XTC of V832 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 asserts low active signal.

## 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 memory) to FIFO of CORAL can be performed, but transfer from CORAL 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 asserts low active signal.

## 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 and demand transfer modes.
- 4) The XINT signal asserts high active signal. 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 low active signal.

## 11.9 Supported DMA Transfer Modes

	Single address mode	Dual address mode
SH3	Not supported	Direct address transfer mode supported; indirect address transfer mode not supported. Transfer is performed in 32-bit units. Cycle steal mode and burst mode supported.
SH4	Transfer performed in units of 32 bits or 32 bytes Cycle steal mode and burst mode supported	Transfer is performed in 32-bit units. Transfer to memory is performed in 32-byte units. Transfer to FIFO not supported. Cycle steal mode and burst mode supported.
V832		Transfer is performed in 32-bit units. Single transfer mode and demand transfer mode supported.
SPARC lite		



## 12 ELECTRICAL CHARACTERISTICS

### 12.1 Introduction

The values in this chapter are the final specification for CORAL-LQ.

### 12.2 Maximum Rating

Maximum Rating

Parameter	Symbol	Maximum rating	Unit
Power supply voltage	$V_{DDL}^{*1}$ $V_{DDH}$	$-0.5 < V_{DDL} < 2.5$ $-0.5 < V_{DDH} < 4.0$	V
Input voltage	$V_I$	$-0.5 < V_I < V_{DDH} + 0.5 (< 4.0)$	V
Output current	$I_O$	$\pm 13$	mA
Ambient for storage temperature	TST	$-55 < TST < +125$	°C

\*1 Includes PLL power supply

## 12.3 Recommended Operating Conditions

### 12.3.1 Recommended operating conditions

**Recommended Operating Conditions**

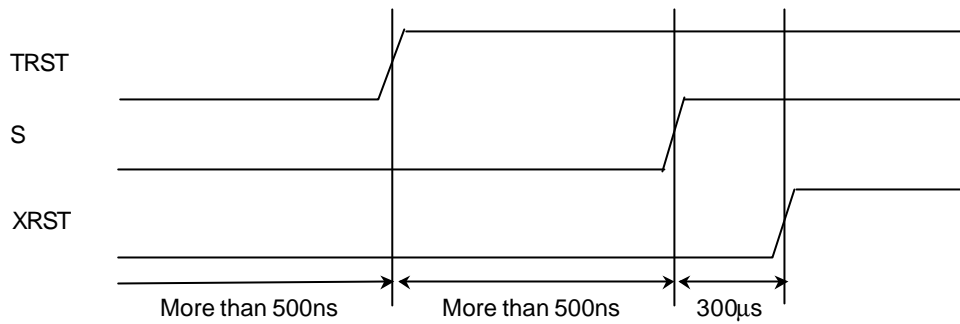
Parameter	Symbol	Rating			Unit
		Min.	Typ.	Max.	
Supply voltage	$V_{DDL}$ *1	1.65	1.8	1.95	V
	$V_{DDH}$	3.0	3.3	3.6	
Input voltage (High level)	$V_{IH}$	2.0		$V_{DDH}+0.3$	V
Input voltage (low level)	$V_{IL}$	-0.3		0.8	V
Ambient temperature for operation	TA	-40		85	°C

\*1 Includes PLL power supply

### 12.3.2 Note at power-on

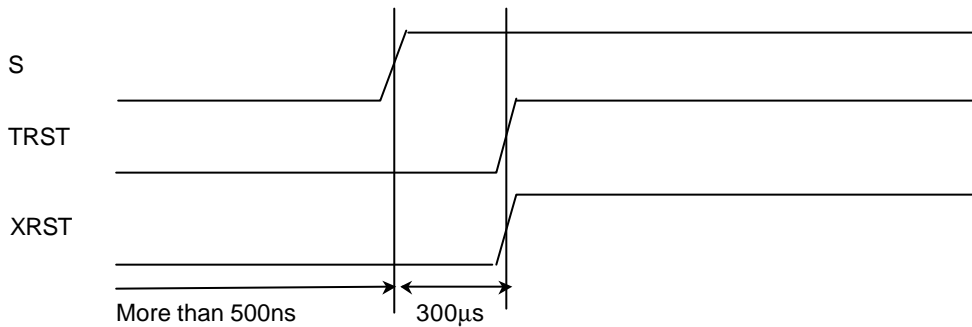
- There is no restriction on the sequence of power-on/power-off between  $V_{DDL}$  and  $V_{DDH}$ . However, do not apply only  $V_{DDH}$  for more than a few seconds.
- Do not input HSYNC, VSYNC, and EO signals when the power supply voltage is not applied. (See the input voltage item in **Maximum rating**.)
- There are three reset sequences as described next page. And please input at least 10 BCLK cycles to BCLK pin before XRST negated.

(1) TRST, S, XRST are changed from “Low” to “High” levels in this order:



Immediately after power-on, input the “Low” level to the TRST pin for 500 ns or more. After the TRST pin is set to “High” level, input the “Low” level to the S pin for 500 ns or more. After the S pin is set to “High” level, input the “Low” level to the XRST pin for 300 µs or more.

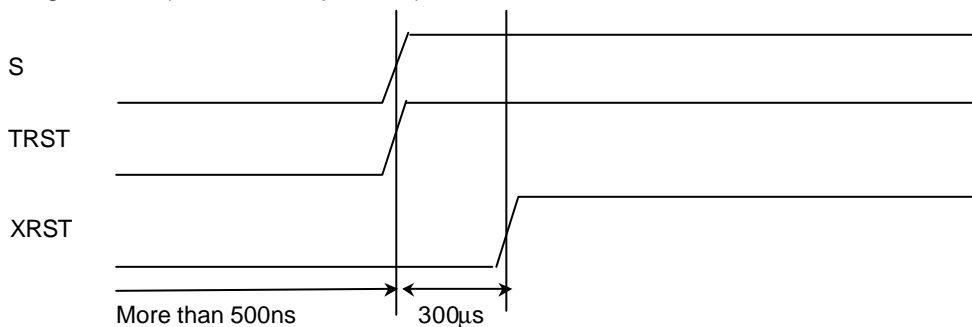
(2) S is changed from “Low” to “High” levels and then TRST and XRST are changed from “Low” to “High” levels simultaneously (TRST = XRST is possible):



Immediately after power-on, input the “Low” level to the S pin for 500 ns or more. After the S pin is set to “High” level, input the “Low” level to the TRST and XRST pins for 300 µs or more.

There is no restriction on the input sequence to the XRST and TRST pins.

(3) S and TRST are changed from “Low” to “High” levels simultaneously and then XRST is changed from “Low” to “High” levels (S = TRST is possible):



Immediately after power-on, input the “Low” level to the S and TRST pins for 500 ns or more. After the S and TRST pins are set to “High” level, input the “Low” level to the XRST pin for 300 µs or more.

There is no restriction on the input sequence to the S and TRST pins.

## 12.4 DC Characteristics

### 12.4.1 DC Characteristics

Measuring condition:  $V_{DDL} = 1.8 \pm 1.5 \text{ V}$ ,  $V_{DDH} = 3.3 \pm 0.3 \text{ V}$ ,  $V_{SS} = 0.0 \text{ V}$ ,  $T_a = -40 \text{ to } +85^\circ\text{C}$

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
Output voltage <sup>*1</sup> ("High" level)	$V_{OH}$	$I_{OH}=-100\mu\text{A}$	$V_{DDH}-0.2$		$V_{DDH}$	V
Output voltage <sup>*2</sup> ("Low" level)	$V_{OL}$	$I_{OL}=100\mu\text{A}$	0.0		0.2	V
Output current ("High" level)	--	$V_{DDH}=3.3\text{V}\pm 0.3\text{V}$	(*1)			mA
Output current ("Low" level)	--	$V_{DDH}=3.3\text{V}\pm 0.3\text{V}$	(*1)			mA
Input leakage current	IL				$\pm 5$	$\mu\text{A}$
Pin capacitance	C				16	pF

\*1: Please refer "V-I characteristics diagram".

**L Type:** Output characteristics of MD0-63, MDQM0-7, R2-7, G2-7, B2-7 pins

**M Type:** Output characteristics of pins other than signals indicated by L type and H type

**H Type:** Output characteristics of XINT, DREQ, XRDY, MCLKO pins

### 12.4.2 V-I characteristics diagram

Condition MAX: Process=Slow, Ta=85°C, V<sub>DD</sub>=3.6V  
 TYP: Process=Typical, Ta=25°C, V<sub>DD</sub>=3.3V  
 MIN: Process=Fast, Ta=-40°C, V<sub>DD</sub>=3.0V

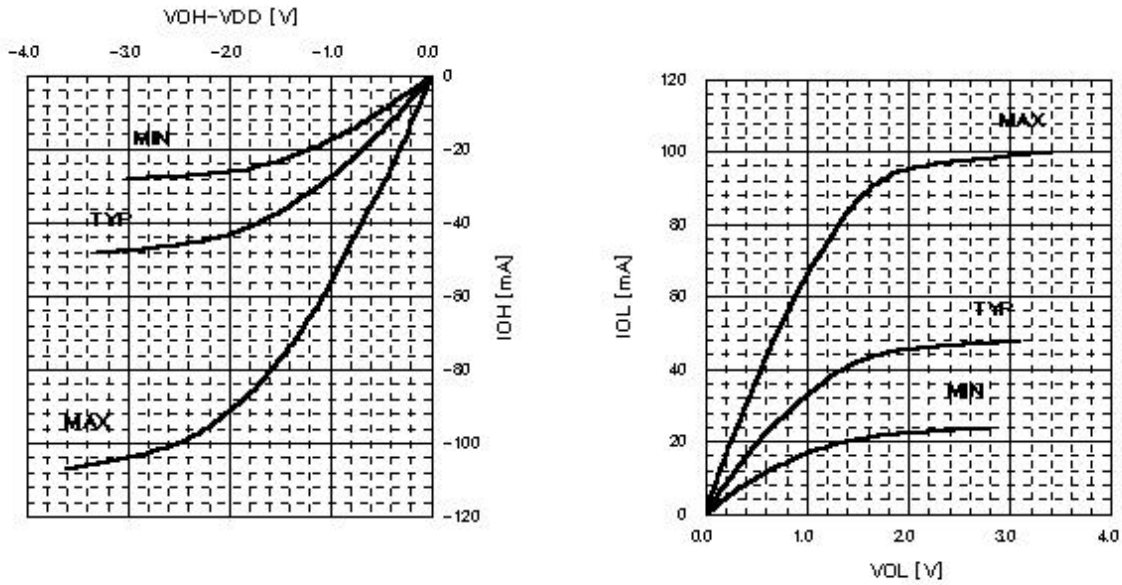


Fig. V-I characteristics L, M type

Condition MAX: Process=Slow, Ta=85°C, V<sub>DD</sub>=3.6V  
 TYP: Process=Typical, Ta=25°C, V<sub>DD</sub>=3.3V  
 MIN: Process=Fast, Ta=-40°C, V<sub>DD</sub>=3.0V

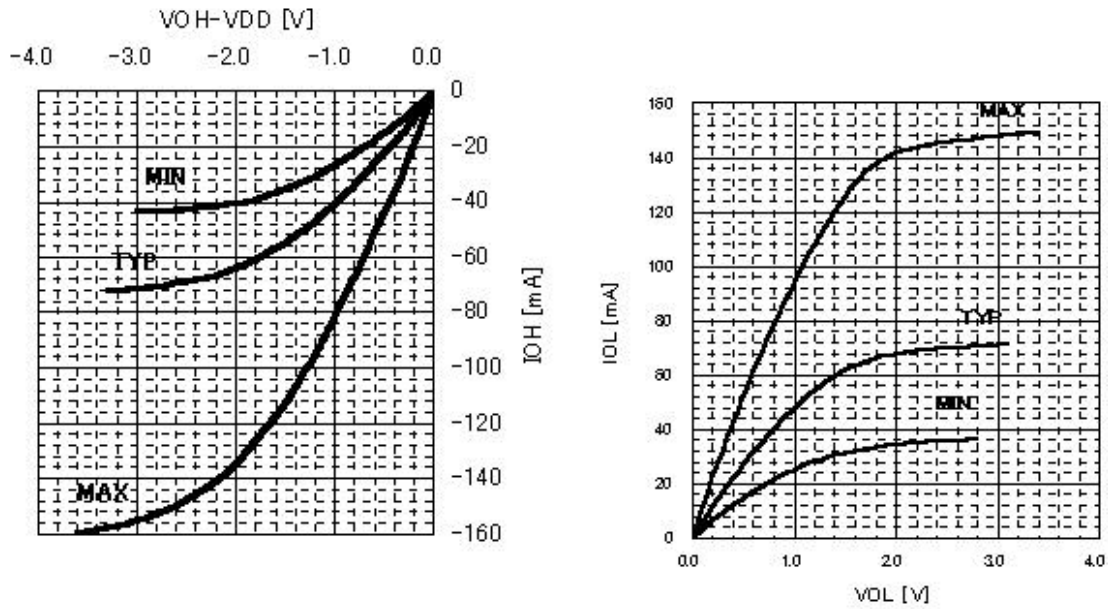


Fig. V-I characteristics H type

## 12.5 AC Characteristics

### 12.5.1 Host interface

#### Clock

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
BCLKI frequency	f <sub>BCLKI</sub>				100	MHz
BCLKI H-width	t <sub>HCLKI</sub>		1			ns
BCLKI L-width	t <sub>LCLKI</sub>		1			ns

#### Host interface signals

(Operating condition: external load = 20 pF)

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
Address set up time	t <sub>ADS</sub>		3.0			ns
Address hold time	t <sub>ADH</sub>		0.0			ns
XBS Set up time	t <sub>BSS</sub>		3.0			ns
XBS Hold time	t <sub>BSH</sub>		0.0			ns
XCS Set up time	t <sub>CSS</sub>		3.0			ns
XCS Hold time	t <sub>CSH</sub>		0.0			ns
XRD Set up time	t <sub>RDS</sub>		3.0			ns
XRD Hold time	t <sub>RDH</sub>		0.0			ns
XWE Set up time	t <sub>WES</sub>		5.0			ns
XWE Hold time	t <sub>WEH</sub>		0.0			ns
Write data set up time	t <sub>WDS</sub>		3.5			ns
Write data hold time	t <sub>WDH</sub>		0.0			ns
DTACK Set up time	t <sub>DAKS</sub>		3.0			ns
DTACK Hold time	t <sub>DAKH</sub>		0.0			ns
DRACK Set up time	t <sub>DRKS</sub>		3.0			ns
DRACK Hold time	t <sub>DRKH</sub>		0.0			ns
Read data delay time (for XRD)	t <sub>RDDZ</sub>		4.5		10.5	ns
Read data delay time	t <sub>RDD</sub>	*2	4.5		9.5	ns
XRDY Delay time (for XCS)	t <sub>RDYDZ</sub>		3.5		7.0	ns
XRDY Delay time	t <sub>RDYD</sub>		2.5		6.0	ns
XINT Delay time	t <sub>INTD</sub>		3.0		7.0	ns
DREQ Delay time	t <sub>DQRD</sub>		3.5		7.0	ns
MODE Hold time	t <sub>MODH</sub>	*1			20.0	ns

\*1 Hold time required for canceling reset

\*2 Valid data is output at assertion of XRDY and is retained until XRD is negated.

## 12.5.2 Video interface

### Clock

Parameter	Symbol	Condition	Rating			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	Rating			Unit
			Min.	Typ.	Max.	
HSYNC Input pulse width	$t_{WHSYNC0}$	*1	3			clock
	$t_{WHSYNC1}$	*2	3			clock
HSYNC Input setup time	$t_{SHSYNC}$	*2	10			ns
HSYNC Input hold time	$t_{HHSYNC}$	*2	10			ns
VSYNC Input pulse width	$t_{WHSYNC1}$		1			HSYNC 1 cycle

\*1 Applied only in PLL synchronization mode (CKS = 0), reference clock output from internal PLL (cycle = 1/14\*CLK)

\*2 Applied only in DCLKI synchronization mode (CKS = 1), reference clock = DCLKI

### Output signals

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
RGB Output delay time	$T_{RGB}$		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

### 12.5.3 Graphics memory interface

**Condition:** Clock frequency=133MHz, 100MHz, BCLK. Printed-wiring is isometry.

**An assumed external capacitance**

Parameter	An assumed external capacitance			Unit
	Min	Typ	Max	
Board pattern	5.0		15.0	pF
SDRAM (CLK)	2.5		4.0	pF
SDRAM (D)	4.0		6.5	pF
SDRAM (A, DQM)	2.5		5.0	pF

#### Clock

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
MCLKO Frequency	$f_{MCLKO}$				*1	MHz
MCLKO H-width	$t_{HMCLKO}$		1.0			ns
MCLKO L-width	$t_{LMCLKO}$		1.0			ns
MCLKI Frequency	$f_{MCLKI}$				*1	MHz
MCLKI H-width	$t_{HMCLKI}$		1.0			ns
MCLKI L-width	$t_{LMCLKI}$		1.0			ns

\*1 For the bus-asynchronous mode, the frequency is 1/3 of the oscillation frequency of the internal PLL. For the bus-synchronous mode, the frequency is the same as the frequency of BCLKI.

#### Input signals

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
MD Input data setup time	$t_{MDIDS}$	*2	2.0			ns
MD Input data hold time	$t_{MDIDH}$	*2	0.7			ns

\*2 It means against MCLKI.



There are some cases regarding AC specifications of output signals.

The following tables shows typical six cases of external SDRFAM capacitance.

(1) External SDRAM capacitance case 1

External SDRAM capacitance

SDRAM x1	Total capacitance	Unit
MCLKO	9.9pF (DRAM CLK 2.5pF, Board pattern 5pF)	pF
MA,MRAS,MCAS,MWE	7.5pF (DRAM A.DQM 2.5pF, Board pattern 5pF)	pF
MD,DQM	9.0pF (DRAM D 4pF, Board pattern 5pF)	pF

Output signals

Parameter	Symbol	Condition	Rating *1			Unit
			Min.	Typ.	Max.	
MCLKI signal delay time against MCLKO	t <sub>DID</sub>		0		4.2	ns
MA, MRAS, MCAS, MWE Access time	t <sub>MAD</sub>		1.0		5.0	ns
MDQM Access time	t <sub>MDQMD</sub>		1.1		5.4	ns
MD Output access time	t <sub>MDOD</sub>		1.1		5.4	ns

(2) External SDRAM capacitance case 2

External SDRAM capacitance

SDRAM x1	Total capacitance	Unit
MCLKO	25.4pF (DRAM CLK 4.0pF, Board pattern 15pF)	pF
MA,MRAS,MCAS,MWE	20.0pF (DRAM A.DQM 5pF, Board pattern 15pF)	pF
MD,DQM	21.5pF (DRAM D 6.5pF, Board pattern 15pF)	pF

Output signals

Parameter	Symbol	Condition	Rating *1			Unit
			Min.	Typ.	Max.	
MCLKI signal delay time against MCLKO	t <sub>DID</sub>		0		3.5	ns
MA, MRAS, MCAS, MWE Access time	t <sub>MAD</sub>		1.0		5.2	ns
MDQM Access time	t <sub>MDQMD</sub>		1.2		5.5	ns
MD Output access time	t <sub>MDOD</sub>		1.2		5.5	ns

**(3) External SDRAM capacitance case 3**

**External SDRAM capacitance**

SDRAM x2	Total capacitance	Unit
MCLKO	12.4pF (DRAM CLK 2.5pF x2, Board pattern 5pF)	pF
MA,MRAS,MCAS,MWE	10.0pF (DRAM A.DQM 2.5pF x2, Board pattern 5pF)	pF
MD,DQM	9.0pF (DRAM D 4pF, Board pattern 5pF)	pF

**Output signals**

Parameter	Symbol	Condition	Rating *1			Unit
			Min.	Typ.	Max.	
MCLKI signal delay time against MCLKO	t <sub>DID</sub>		0		4.1	ns
MA, MRAS, MCAS, MWE Access time	t <sub>MAD</sub>		1.0		5.0	ns
MDQM Access time	t <sub>MDQMD</sub>		1.1		5.2	ns
MD Output access time	t <sub>MDOD</sub>		1.1		5.2	ns

**(4) External SDRAM capacitance case 4**

**External SDRAM capacitance**

SDRAM x2	Total capacitance	Unit
MCLKO	29.4pF (DRAM CLK 4.0pF x2, Board pattern 15pF)	pF
MA,MRAS,MCAS,MWE	25.0pF (DRAM A.DQM 5pF x2, Board pattern 15pF)	pF
MD,DQM	21.5pF (DRAM D 6.5pF, Board pattern 15pF)	pF

**Output signals**

Parameter	Symbol	Condition	Rating *1			Unit
			Min.	Typ.	Max.	
MCLKI signal delay time against MCLKO	t <sub>DID</sub>		0		3.4	ns
MA, MRAS, MCAS, MWE Access time	t <sub>MAD</sub>		1.1		5.4	ns
MDQM Access time	t <sub>MDQMD</sub>		1.1		5.5	ns
MD Output access time	t <sub>MDOD</sub>		1.1		5.5	ns

(5) External SDRAM capacitance case 5

External SDRAM capacitance

SDRAM x4	Total capacitance	Unit
MCLKO	17.4pF (DRAM CLK 2.5pF x4, Board pattern 5pF)	pF
MA,MRAS,MCAS,MWE	15.0pF (DRAM A.DQM 2.5pF x4, Board pattern 5pF)	pF
MD,DQM	9.0pF (DRAM D 4pF, Board pattern 5pF)	pF

Output signals

Parameter	Symbol	Condition	Rating *1			Unit
			Min.	Typ.	Max.	
MCLKI signal delay time against MCLKO	t <sub>DID</sub>		0		3.9	ns
MA, MRAS, MCAS, MWE Access time	t <sub>MAD</sub>		1.0		5.2	ns
MDQM Access time	t <sub>MDQMD</sub>		1.0		5.0	ns
MD Output access time	t <sub>MDOD</sub>		1.0		5.0	ns

(6) External SDRAM capacitance case 6

External SDRAM capacitance

SDRAM x4	Total capacitance	Unit
MCLKO	37.3pF (DRAM CLK 4.0pF x4, Board pattern 15pF)	pF
MA,MRAS,MCAS,MWE	35.0pF (DRAM A.DQM 5pF x4, Board pattern 15pF)	pF
MD,DQM	21.5pF (DRAM D 6.5pF, Board pattern 15pF)	pF

Output signals

Parameter	Symbol	Condition	Rating *1			Unit
			Min.	Typ.	Max.	
MCLKI signal delay time against MCLKO	t <sub>DID</sub>		0		3.4	ns
MA, MRAS, MCAS, MWE Access time	t <sub>MAD</sub>		1.2		5.7	ns
MDQM Access time	t <sub>MDQMD</sub>		1.0		5.3	ns
MD Output access time	t <sub>MDOD</sub>		1.0		5.3	ns

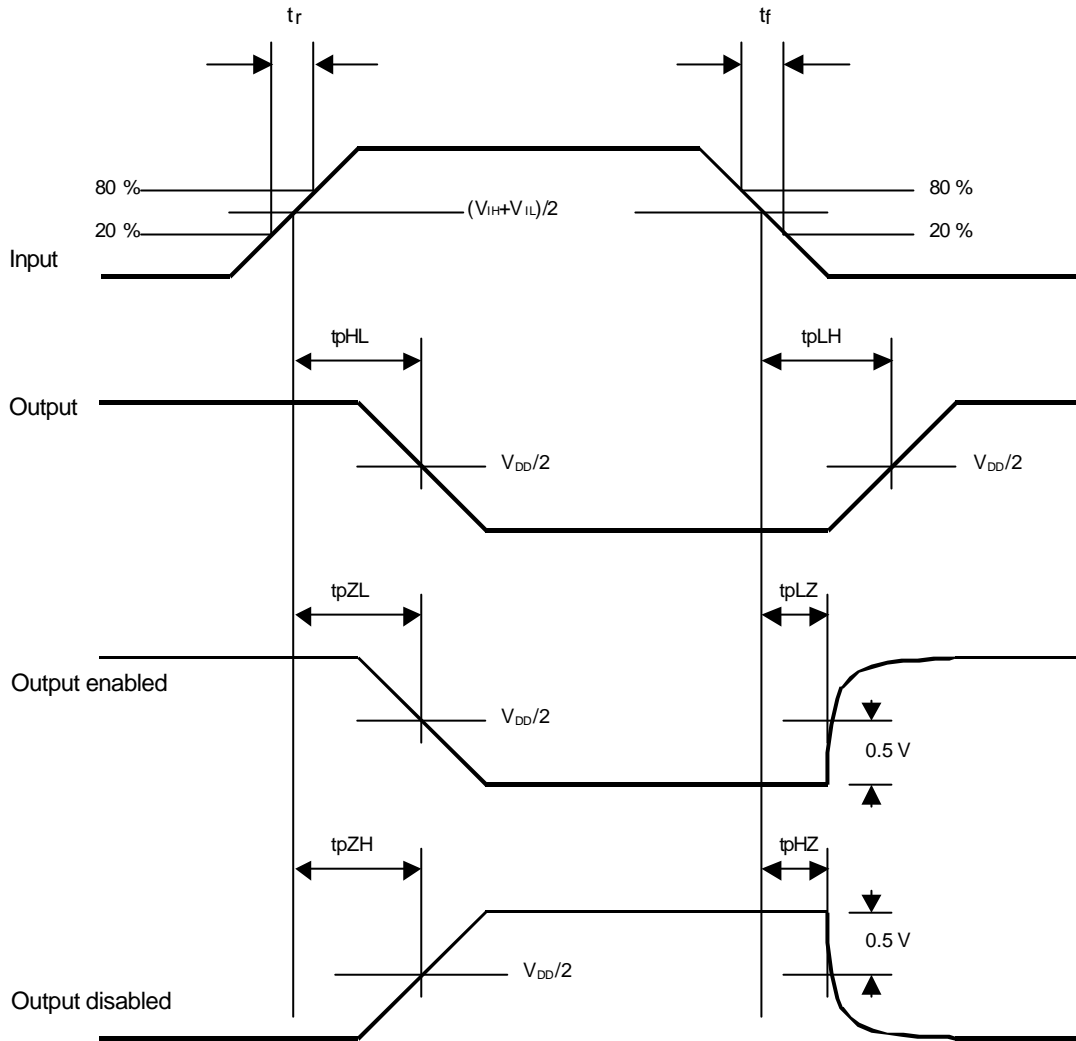
### 12.5.4 PLL specifications

Parameter	Rating	Description
Input frequency (typ.)	14.31818 MHz	
Output frequency	400.9090 MHz	×28
Duty ratio	101.6 to 93.0%	H/L Pulse width ratio of PLL output
Jitter	60 to -60 ps	Frequency tolerant of two consecutive clock cycles

CLKSEL1	CLKSEL1	Input frequency	Assured operation range (*1)
L	L	13.5 MHz	13.365 to 13.5 MHz
L	H	14.32 MHz	14.177 to 14.32 MHz
H	L	17.73 Hz	17.553 to 17.73 MHz

\*1 Assured operation input frequency range: Standard value -1%

## 12.6 AC Characteristics Measuring Conditions



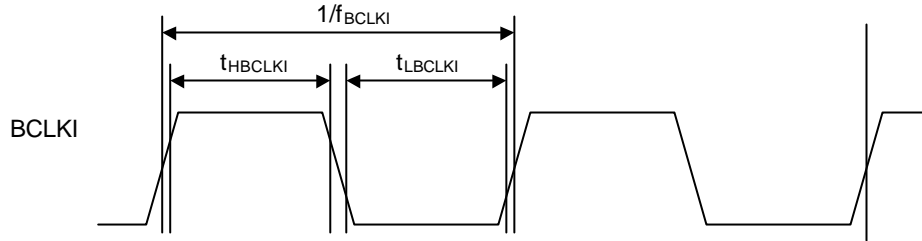
$T_r, t_f \leq 5 \text{ ns}$

$V_{IH} = 2.0 \text{ V}, V_{IL} = 0.8 \text{ V}$  (3.3-V CMOS interface input)

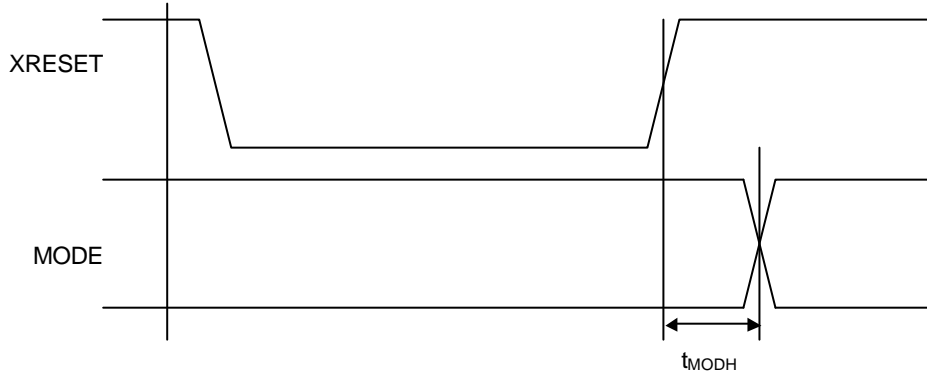
## 12.7 Timing Diagram

### 12.7.1 Host interface

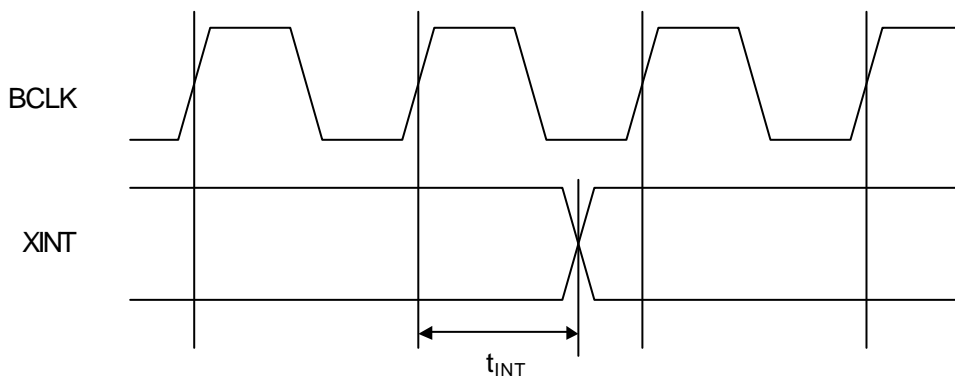
#### Clock



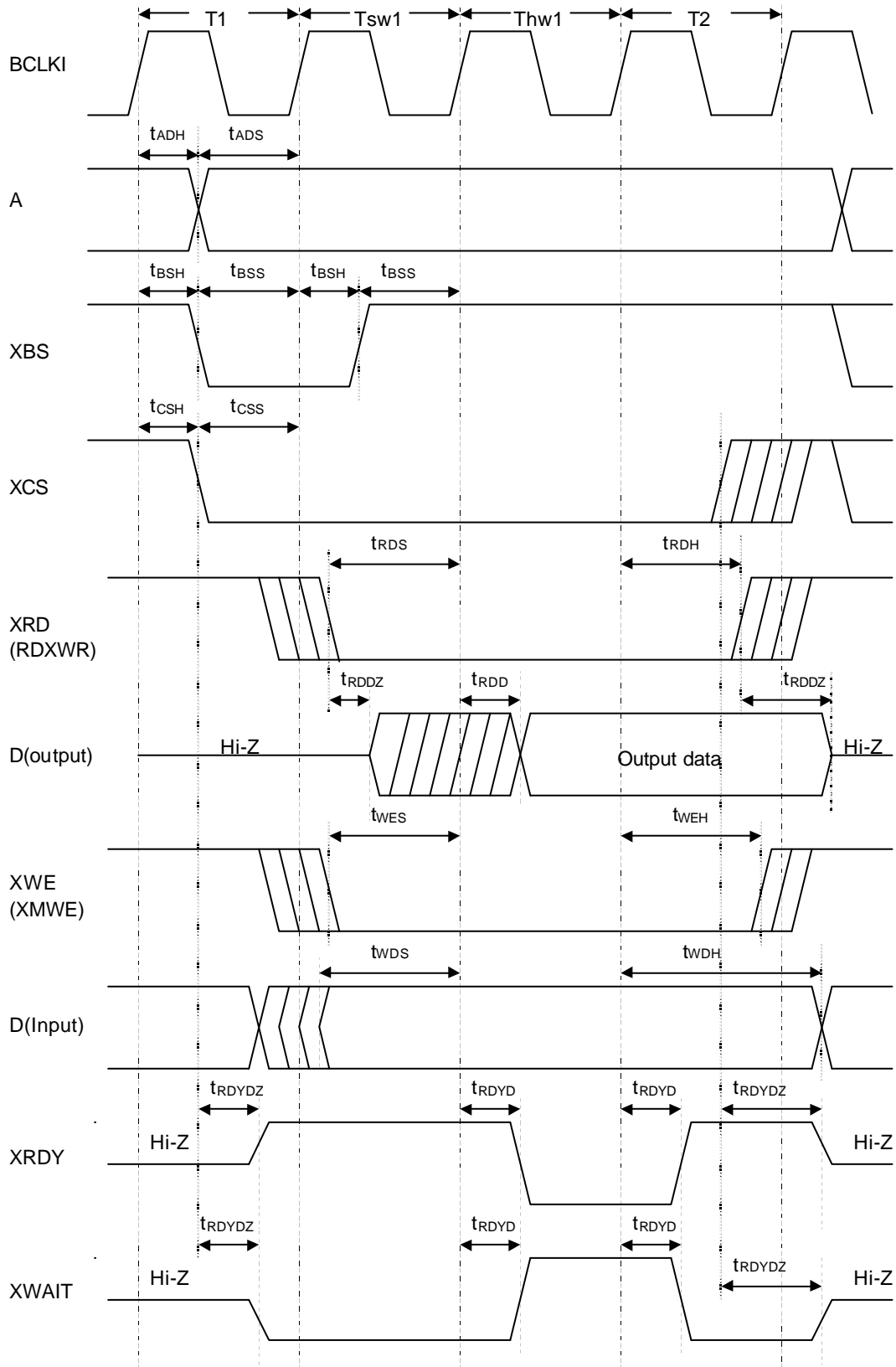
#### MODE hold time



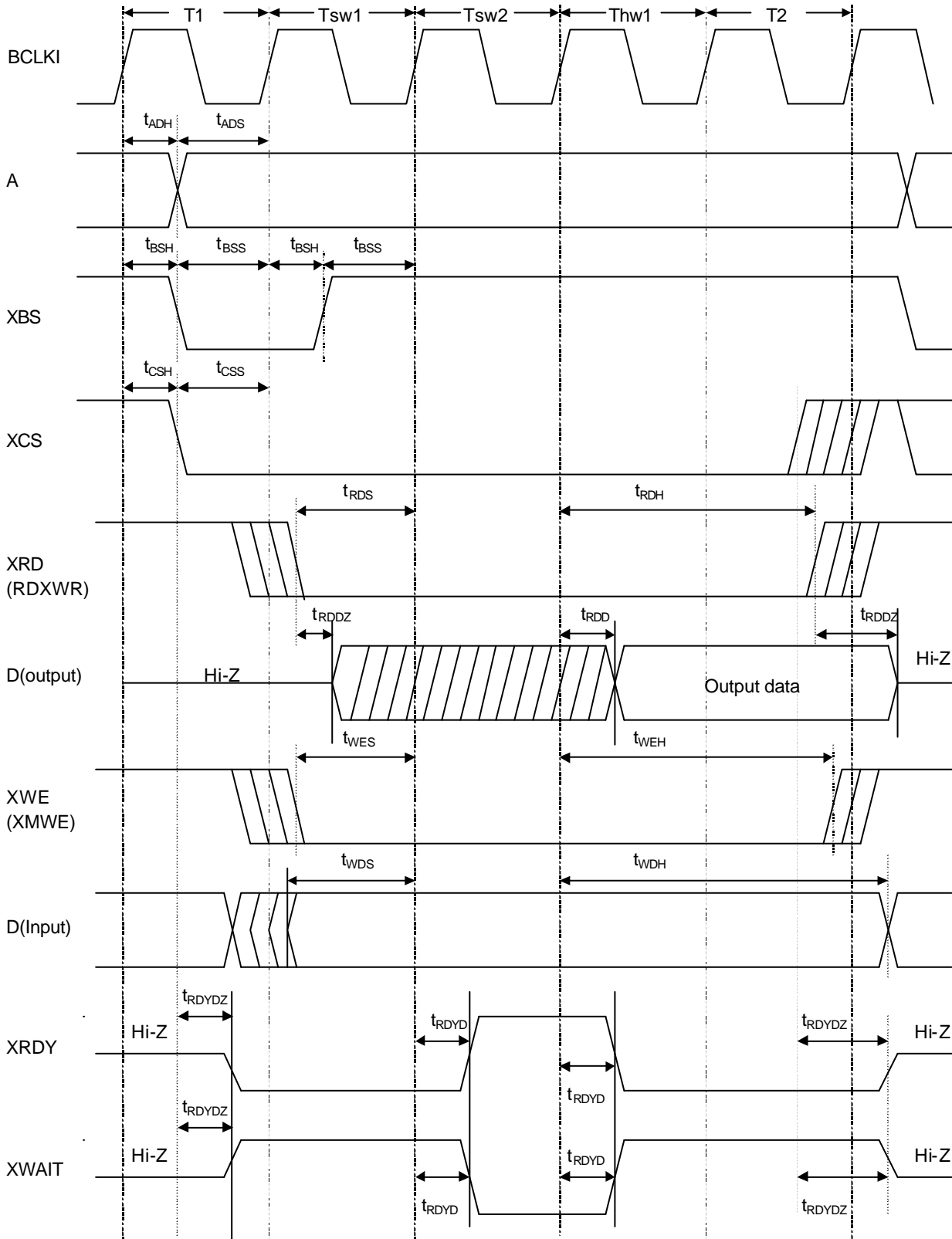
#### XINT output delay times



Host bus AC timing (Normally Not Ready)

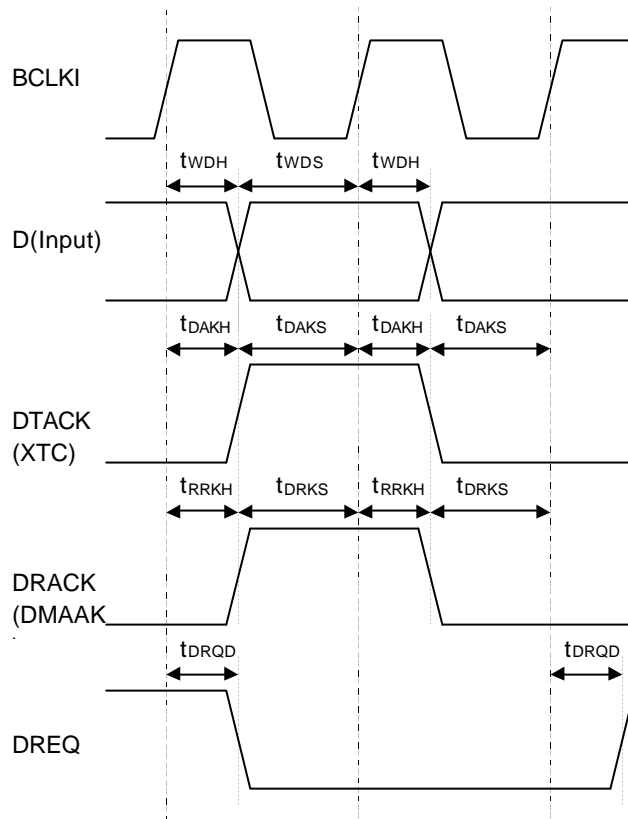


Host bus AC timing (Normally Ready)





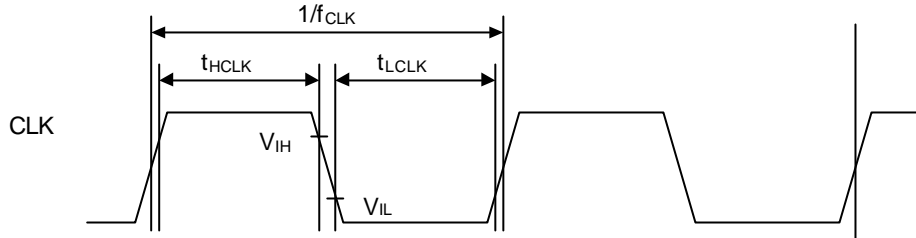
**DMA AC timing**



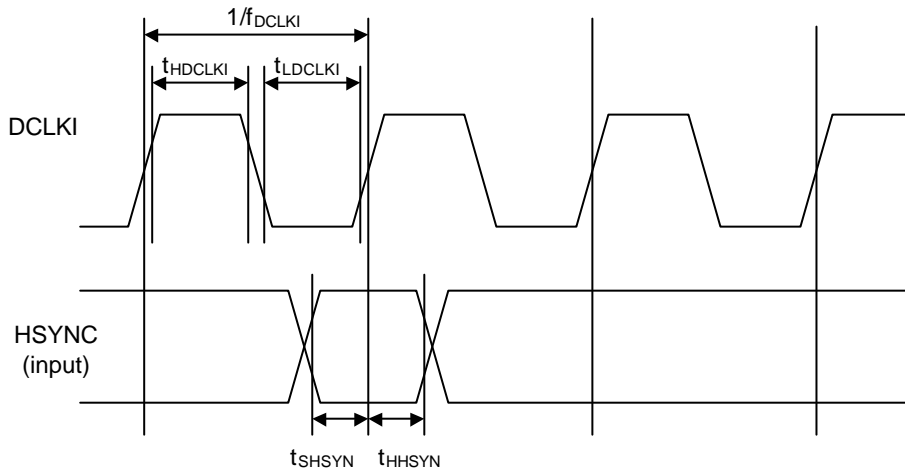
\*: The above timing diagram for the D pin is that of when a single DMA is used.  
 When a dual DMA is used, see the host bus-timing diagram.

## 12.7.2 Video interface

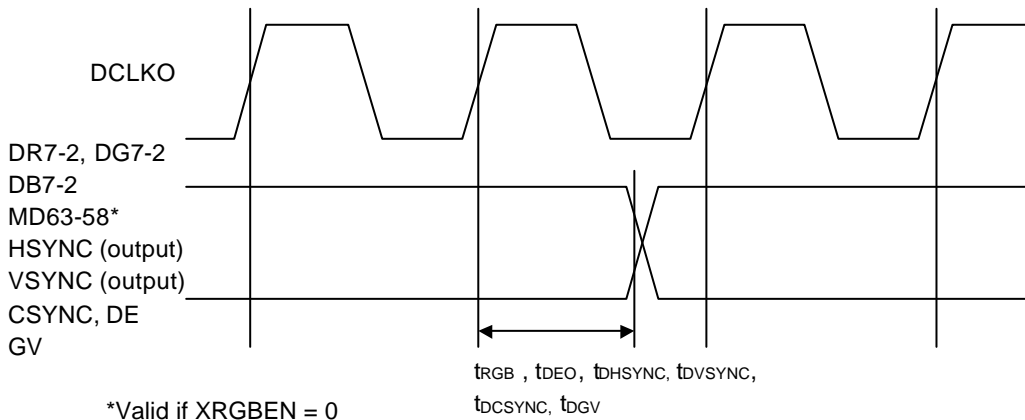
### Clock



### HSYNC signal setup/hold

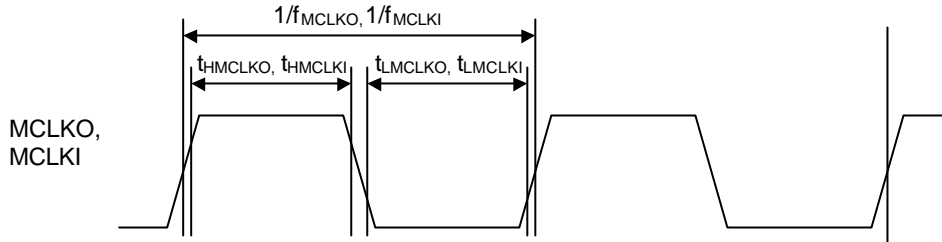


### Output signal delay

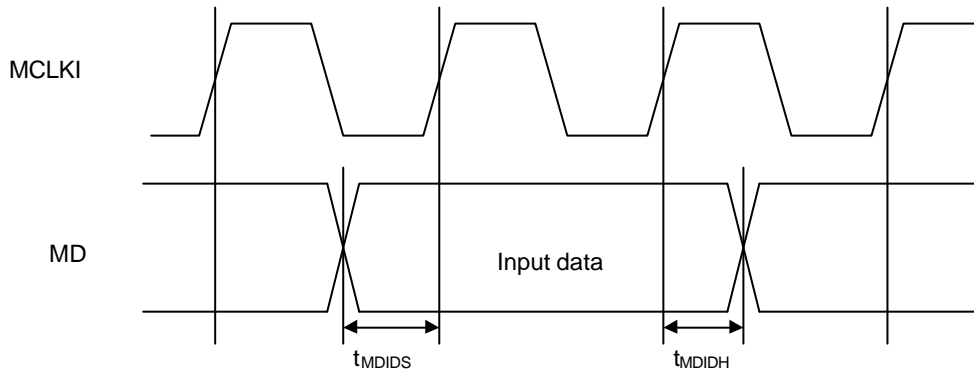


### 12.7.3 Graphics memory interface

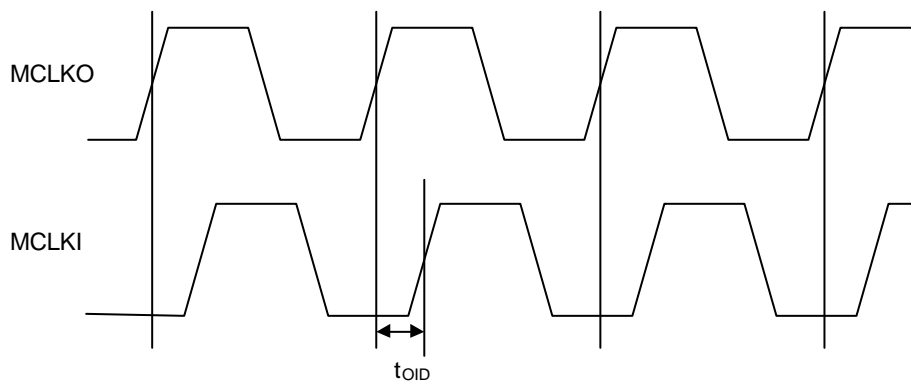
#### Clock



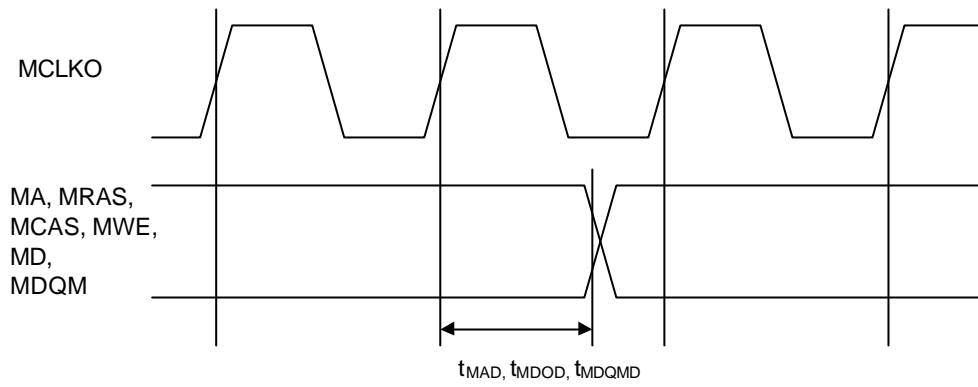
#### Input signal setup/hold time



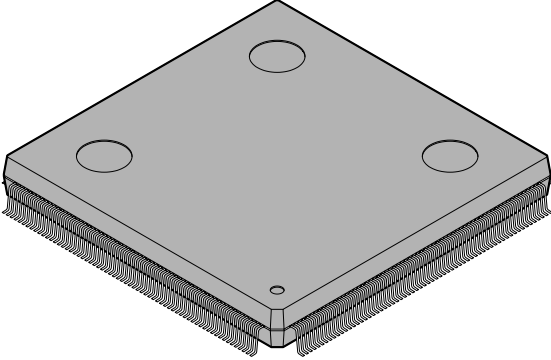
#### MCLKI signal delay

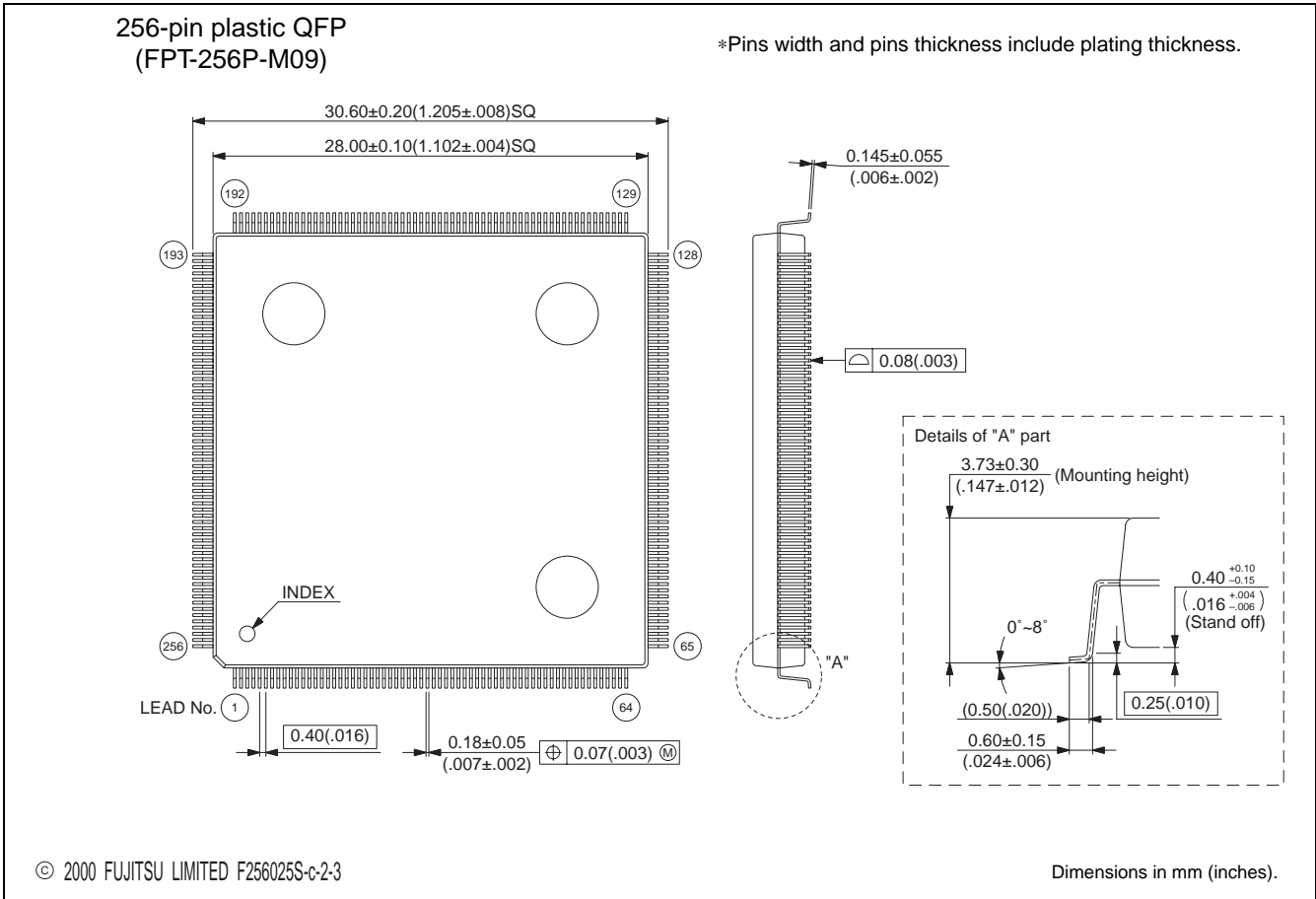


### Output signal delay



FPT-256P-M09

<p>256-pin plastic QFP</p>  <p>(FPT-256P-M09)</p>	Lead pitch	0.40 mm
	Package width × package length	28.0 × 28.0 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	4.03 mm MAX
	Weight	5.74g
	Remark	Low heat resistance type
	Code(Reference)	P-FQFP256-28 × 28-0.40



The contents of this document are subject to change without notice.  
Customers are advised to consult with FUJITSU sales representatives before ordering.  
FUJITSU is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of the information or package dimensions in this document.