

# **MB86295S <CORAL P>**

## **PCI Graphics Controller Specification**

Revision 1.1  
8th January, 2003



Copyright © FUJITSU LIMITED 2002

**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.

#### All Rights Reserved

The contents of this document are subject to change without notice. Customers are advised to consult with FUJITSU sales representatives before ordering. The information and circuit diagrams in this document are presented as examples of semiconductor device applications, and are not intended to be incorporated in devices for actual use. Also, FUJITSU is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of this information or circuit diagrams. The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).

Please note that Fujitsu will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products. Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

## Update history

| Date       | Version | Page count | Change  |
|------------|---------|------------|---|
| 22.8.2002  | 0.1     | 266        | First edition (update from Coral-LQ specification)                          |
| 26.8.2002  | 0.2     | 272        | Video capture description added   |
| 12.11.2002 | 0.3     | 274        | Minor updates to host interface description. Addition of waveforms/timing.  |
| 27.11.2002 | 0.4     | 276        | Refer diff03vs04.txt file.  |
| 2.12.2002  | 0.4a    | 277        | Video Input Register update   |
| 6.12.2002  | 1.0     | 283        | First release   |
| 26.12.2002 | 1.0a    | 282        | Delete the description of two host interface registers.                     |
| 8.1.2003   | 1.1     | 300        | I <sup>2</sup> C interface and PCI configuration register description added |

## CONTENTS

|   |           |
|---|-----------|
| <b>1. GENERAL</b>   | <b>1</b>  |
| 1.1 Preface.....  | 1         |
| 1.2 Features.....   | 2         |
| 1.3 Block Diagram.....                                    | 3         |
| 1.4 Functional Overview.....                              | 4         |
| 1.4.1 Host CPU interface.....                             | 4         |
| 1.4.2 External memory interface.....                      | 5         |
| 1.4.3 Display controller.....                             | 6         |
| 1.4.4 Video capture function.....                         | 8         |
| 1.4.5 Geometry processing.....                            | 9         |
| 1.4.6 2D Drawing.....                                     | 10        |
| 1.4.7 3D Drawing.....                                     | 12        |
| 1.4.8 Special effects.....                                | 13        |
| 1.4.9 Others.....   | 15        |
| <b>2. PINS</b>  | <b>16</b> |
| 2.1 Signals.....  | 16        |
| 2.1.1 Signal lines.....                                   | 16        |
| 2.2 Pin Assignment.....                                   | 17        |
| 2.2.1 Pin assignment diagram.....                         | 17        |
| 2.2.2 Pin assignment table.....                           | 18        |
| 2.3 Pin Function.....                                     | 26        |
| 2.3.1 Host CPU interface.....                             | 26        |
| 2.3.2 Video output interface.....                         | 28        |
| 2.3.3 Video capture interface.....                        | 29        |
| 2.3.4 I <sup>2</sup> C interface.....                     | 30        |
| 2.3.5 Graphics memory interface.....                      | 31        |
| 2.3.6 Clock input.....                                    | 32        |
| 2.3.7 Test pins.....                                      | 33        |
| 2.3.8 Reset sequence.....                                 | 33        |
| 2.3.9 How to switch internal operating frequency.....     | 33        |
| <b>3. HOST INTERFACE</b>                                  | <b>34</b> |
| 3.1 Standard PCI Slave Accesses.....                      | 34        |
| 3.1.1 PCI Slave Write.....                                | 34        |
| 3.1.2 PCI Slave Read.....                                 | 34        |
| 3.2 Burst Controller Accesses (including PCI Master)..... | 34        |
| 3.2.1 Transfer Modes.....                                 | 35        |
| 3.2.2 Burst Controller Control/Status.....                | 36        |
| 3.3 FIFO Transfers.....                                   | 37        |
| 3.4 GPIO/Serial Interface.....                            | 37        |
| 3.4.1 GPIO.....   | 37        |
| 3.4.2 Serial Interface.....                               | 37        |
| 3.5 Interrupt.....  | 38        |
| 3.5.1 Internal Bus/FIFO timeout.....                      | 38        |
| 3.5.2 Address Error Interrupt.....                        | 39        |
| 3.6 Memory Map.....                                       | 39        |
| <b>4. I<sup>2</sup>C Interface Controller</b>             | <b>41</b> |
| 4.1 Features.....   | 41        |
| 4.2 Block diagram.....                                    | 42        |

|   |           |
|---|-----------|
| 4.2.1 Block Diagram.....                          | 42        |
| 4.2.2 Block Function Overview.....                | 43        |
| 4.3 Example application .....                     | 44        |
| 4.3.1 Connection Diagram.....                     | 44        |
| 4.4 Function overview .....                       | 45        |
| 4.4.1 START condition.....                        | 45        |
| 4.4.2 STOP condition.....                         | 45        |
| 4.4.3 Addressing.....                             | 46        |
| 4.4.4 Synchronization of SCL.....                 | 46        |
| 4.4.5 Arbitration .....                           | 47        |
| 4.4.6 Acknowledge.....                            | 47        |
| 4.4.7 Bus error.....                              | 47        |
| 4.4.8 Initialize .....                            | 48        |
| 4.4.9 1-byte transfer from master to slave.....   | 49        |
| 4.4.10 1-byte transfer from slave to master ..... | 50        |
| 4.4.11 Recovery from bus error.....               | 51        |
| 4.5 Note .....                                    | 52        |
| <b>5. DISPLAY CONTROLLER</b>                      | <b>53</b> |
| 5.1 Overview .....                                | 53        |
| 5.2 Display Function.....                         | 54        |
| 5.2.1 Layer configuration.....                    | 54        |
| 5.2.2 Overlay.....                                | 55        |
| 5.2.3 Display parameters.....                     | 57        |
| 5.2.4 Display position control.....               | 58        |
| 5.3 Display Color.....                            | 60        |
| 5.4 Cursor .....                                  | 61        |
| 5.4.1 Cursor display function.....                | 61        |
| 5.4.2 Cursor control.....                         | 61        |
| 5.5 Display Scan Control .....                    | 62        |
| 5.5.1 Applicable display.....                     | 62        |
| 5.5.2 Interlace display.....                      | 63        |
| 5.6 Video Interface, NTSC/PAL Output.....         | 64        |
| <b>6. Video Capture</b>                           | <b>65</b> |
| 6.1 Input Formats.....                            | 65        |
| 6.2 ITU RBT -656 input.....                       | 65        |
| 6.2.1 YUV input format.....                       | 65        |
| 6.2.2 Synchronous Control.....                    | 65        |
| 6.2.3 Non-interlace Transformation .....          | 66        |
| 6.2.4 Area Allocation.....                        | 66        |
| 6.3 RGB input.....                                | 67        |
| 6.3.1. RGB input modes .....                      | 67        |
| 6.3.2. RGB Input Signals.....                     | 67        |
| 6.3.3. Captured Range .....                       | 68        |
| 6.3.4. Direct Input Mode Operation.....           | 69        |
| 6.3.5 Multiplex Input Mode Operation.....         | 69        |
| 6.3.6. Even/Odd field Recognition.....            | 70        |
| 6.3.7. Conversion Operation.....                  | 70        |
| 6.4 Scaling .....                                 | 71        |
| 6.4.1 Downscaling Function.....                   | 71        |
| 6.4.2 Upscaling Function.....                     | 71        |

|   |           |
|---|-----------|
| <b>7. GEOMETRY ENGINE</b>   | <b>72</b> |
| 7.1 Geometry Pipeline.....  | 72        |
| 7.1.1 Processing flow.....  | 72        |
| 7.1.2 Model-view-projection (MVP) transformation (OC→CC coordinate transformation)..... | 73        |
| 7.1.3 3D-2D transformation (CC→NDC coordinate transformation).....                      | 73        |
| 7.1.4 View port transformation (NDC→DC coordinate transformation) .....                 | 74        |
| 7.1.5 View volume clipping.....   | 74        |
| 7.1.6 Back face culling.....  | 76        |
| 7.2 Data Format.....  | 77        |
| 7.2.1 Data format.....  | 77        |
| 7.3 Setup Engine .....  | 78        |
| 7.3.1 Setup processing.....   | 78        |
| 7.4 Log Output of Device Coordinates.....   | 78        |
| 7.4.1 Log output mode.....  | 78        |
| 7.4.2 Log output destination address .....  | 78        |
| <b>8. DRAWING PROCESSING</b>  | <b>79</b> |
| 8.1 Coordinate System.....  | 79        |
| 8.1.1 Drawing coordinates.....  | 79        |
| 8.1.2 Texture coordinates.....  | 80        |
| 8.1.3 Frame buffer.....   | 80        |
| 8.2 Figure Drawing.....   | 81        |
| 8.2.1 Drawing primitives .....  | 81        |
| 8.2.2 Polygon drawing function .....  | 81        |
| 8.2.3 Drawing parameters.....   | 82        |
| 8.2.4 Anti-aliasing function.....   | 83        |
| 8.3 Bit Map Processing.....   | 84        |
| 8.3.1 BLT.....  | 84        |
| 8.3.2 Pattern data format .....   | 84        |
| 8.4 Texture Mapping.....  | 85        |
| 8.4.1 Texture size.....   | 85        |
| 8.4.2 Texture memory.....   | 85        |
| 8.4.3 Texture color.....  | 85        |
| 8.4.4 Texture lapping.....  | 86        |
| 8.4.5 Filtering.....  | 87        |
| 8.4.6 Perspective correction.....   | 87        |
| 8.4.7 Texture blending .....  | 88        |
| 8.4.8 Bi-linear high-speed mode.....  | 88        |
| 8.5 Rendering .....   | 90        |
| 8.5.1 Tiling.....   | 90        |
| 8.5.2 Alpha blending.....   | 90        |
| 8.5.3 Logic operation.....  | 91        |
| 8.5.4 Hidden plane management.....  | 91        |
| 8.6 Drawing Attributes.....   | 92        |
| 8.6.1 Line drawing attributes .....   | 92        |
| 8.6.2 Triangle drawing attributes .....   | 92        |
| 8.6.3 Texture attributes .....  | 92        |
| 8.6.4 BLT attributes.....   | 93        |
| 8.6.5 Character pattern drawing attributes .....  | 93        |
| 8.7 Bold Line .....   | 94        |
| 8.7.1 Starting and ending points.....   | 94        |

|            |  |            |
|------------|--|------------|
| 8.7.2      | Broken line pattern.....                       | 95         |
| 8.7.3      | Edging.....                                    | 96         |
| 8.7.4      | Interpolation of bold line joint.....          | 96         |
| 8.8        | DISPLAY LIST.....                              | 97         |
| 8.8.1      | Overview.....                                  | 97         |
| 8.8.2      | Header format .....                            | 98         |
| 8.8.3      | Parameter format.....                          | 98         |
| 8.8.4      | Geometry command list .....                    | 99         |
| 8.8.5      | Explanation of geometry commands .....         | 102        |
| 8.9        | Rendering Command.....                         | 112        |
| 8.9.1      | Command list.....                              | 112        |
| 8.9.2      | Details of rendering commands .....            | 116        |
| <b>9.</b>  | <b>PCI Configuration Registers</b>             | <b>127</b> |
| 9.1        | PCI Configuration register list.....           | 127        |
| 9.2        | PCI Configuration Registers Descriptions.....  | 128        |
| <b>10</b>  | <b>Local Memory Registers</b>                  | <b>131</b> |
| 10.1       | Local memory register list.....                | 131        |
| 10.1.1     | Host interface register list.....              | 131        |
| 10.1.2     | I <sup>2</sup> C interface register list ..... | 133        |
| 10.1.3     | Graphics memory interface register list.....   | 133        |
| 10.1.4     | Display controller register list.....          | 134        |
| 10.1.5     | Video capture register list.....               | 139        |
| 10.1.6     | Drawing engine register list.....              | 141        |
| 10.1.7     | Geometry engine register list.....             | 147        |
| 10.2       | Explanation of Local Memory Registers.....     | 148        |
| 10.2.1     | Host interface registers .....                 | 149        |
| 10.2.2     | I <sup>2</sup> C Interface Registers .....     | 162        |
| 10.2.3     | Graphics memory interface registers .....      | 168        |
| 10.2.4     | Display control register.....                  | 171        |
| 10.2.5     | Video capture registers.....                   | 219        |
| 10.2.6     | Drawing control registers.....                 | 231        |
| 10.2.7     | Drawing mode registers.....                    | 234        |
| 10.2.8     | Triangle drawing registers.....                | 250        |
| 10.2.9     | Line drawing registers.....                    | 253        |
| 10.2.10    | Pixel drawing registers .....                  | 254        |
| 10.2.11    | Rectangle drawing registers.....               | 254        |
| 10.2.12    | Blt registers .....                            | 256        |
| 10.2.13    | High-speed 2D line drawing registers.....      | 257        |
| 10.2.14    | High-speed 2D triangle drawing registers ..... | 258        |
| 10.2.15    | Geometry control register .....                | 259        |
| 10.2.16    | Geometry mode registers .....                  | 261        |
| 10.2.17    | Display list FIFO registers.....               | 268        |
| <b>11.</b> | <b>TIMING DIAGRAM</b>                          | <b>269</b> |
| 11.1       | Host Interface .....                           | 269        |
| 11.1.1     | PCI Interface.....                             | 269        |
| 11.1.2     | EEPROM Timing.....                             | 270        |
| 11.1.3     | Serial Interface Timing.....                   | 271        |
| 11.2       | I <sup>2</sup> C Interface .....               | 272        |
| 11.3       | Graphics Memory Interface.....                 | 273        |
| 11.3.1     | Timing of read access to same row address..... | 273        |

|            |  |            |
|------------|--|------------|
| 11.3.2     | Timing of read access to different row addresses .....   | 274        |
| 11.3.3     | Timing of write access to same row address .....         | 275        |
| 11.3.4     | Timing of write access to different row addresses.....   | 276        |
| 11.3.5     | Timing of read/write access to same row address.....     | 277        |
| 11.3.6     | Delay between ACTV commands.....                         | 278        |
| 11.3.7     | Delay between Refresh command and next ACTV command..... | 278        |
| 11.4       | Display Timing .....                                     | 279        |
| 11.4.1     | Non-interlace mode.....                                  | 279        |
| 11.4.2     | Interlace video mode .....                               | 280        |
| 11.4.3     | Composite synchronous signal.....                        | 281        |
| <b>12.</b> | <b>ELECTRICAL CHARACTERISTICS</b>                        | <b>282</b> |
| 12.1       | Introduction .....                                       | 282        |
| 12.2       | Maximum Rating.....                                      | 282        |
| 12.3       | Recommended Operating Conditions.....                    | 283        |
| 12.3.1     | Recommended operating conditions .....                   | 283        |
| 12.3.2     | Note at power-on.....                                    | 283        |
| 12.4       | DC Characteristics.....                                  | 284        |
| 12.5       | AC Characteristics.....                                  | 285        |
| 12.5.1     | Host interface.....                                      | 285        |
| 12.5.2     | I <sup>2</sup> C Interface.....                          | 287        |
| 12.5.3     | Video interface.....                                     | 288        |
| 12.5.4     | Graphics memory interface.....                           | 289        |
| 12.5.5     | PLL specifications.....                                  | 296        |
| 12.6       | AC Characteristics Measuring Conditions.....             | 297        |
| 12.7       | Timing Diagram.....                                      | 298        |
| 12.7.1     | Host interface.....                                      | 298        |
| 12.7.2     | Video interface.....                                     | 299        |
| 12.7.3     | Graphics memory interface.....                           | 300        |



# 1. GENERAL

## 1.1 Preface

The MB86295S <CORAL P> is a graphics controller with PCI host interface.

Note:

This device has a I<sup>2</sup>C interface. Purchase of Fujitsu I<sup>2</sup>C components conveys a license under the Philips I<sup>2</sup>C Patent Right to use these components in an I<sup>2</sup>C system, provided that the system conforms to the I<sup>2</sup>C Standard Specification as defined by Philips.

## 1.2 Features

- Geometry engine

Geometry engine supports the geometry processing that is compatible\*\*1 with ORCHID (MB86292). Using the display list created by ORCHID enables drawing. Heavy processing of geometric operations such as coordinates conversions or clipping performed by this device can reduce the CPU loads dramatically. \*\*1(Floating point setup command is tbd)

- 2D and 3D Drawing

The MB86295 has a drawing function that is compatible with the CREMSON (MB86290A). It can draw data using the display list created for CREMSON.

The MB86295 also supports 3D rendering, such as texture mapping with perspective collection and Gouraud shading, alpha blending, and anti-aliasing for drawing smooth lines.

- Digital video capture

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

- Display controller

The MB86295 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 6-layer overlay, 4-sided for palette are expanded.

- Host CPU interface

The MB86295 has a 32 bit, 33MHz PCI interface fully compliant to PCI version 2.1.

- External memory interface

SDRAM and FCRAM can be connected.

- Optional function

Final device can be selected from the combination of geometry high-/low-speed version and video capture function provided/ not provided.

- Others

CMOS technology 0.18μm

BGA256 Package

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

### 1.3 Block Diagram

CORAL general block diagram is shown below:

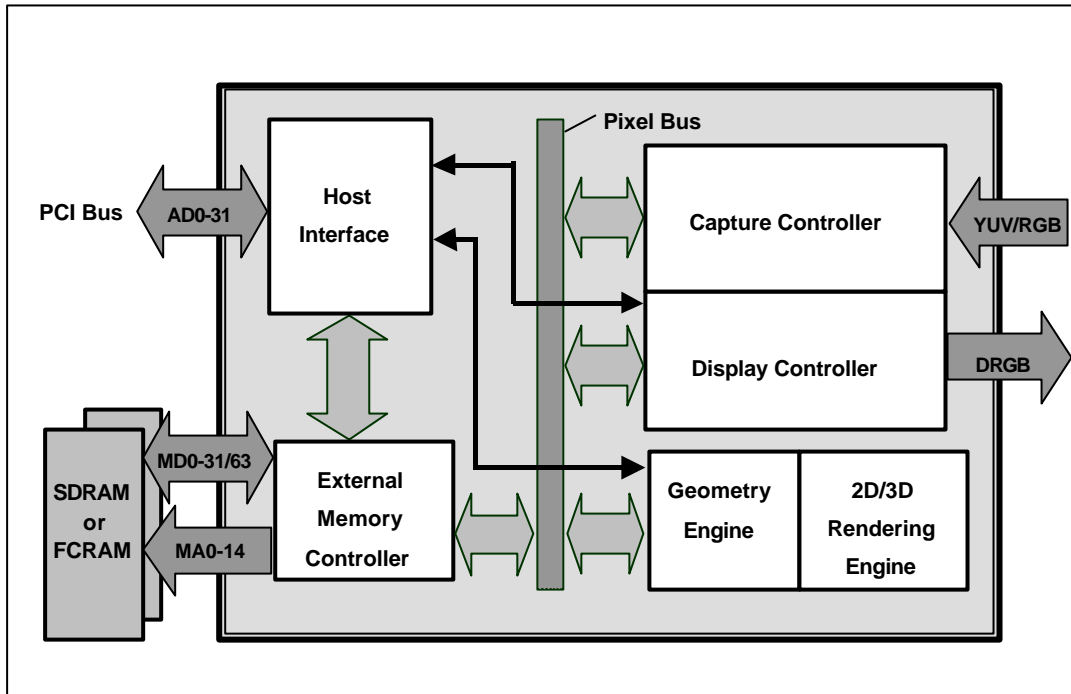


Fig.1.1 CORAL P Block Diagram

## 1.4 Functional Overview

### 1.4.1 Host CPU interface

#### Supported CPU

The MB86295 can be connected to any CPU with a 32MHz 32-bit PCI v2.1 host interface.

#### Configuration

EEPROM configuration supported

Serial interface for external device control through PCI interface

#### PCI Slave

Supports burst reads/writes of up to 8 double words (32 bytes).

Supports multi-burst transfers with automatic pre-fetch.

#### PCI Master

Supports transfers of up to  $2^{24}-1$  double words in bursts of between 1 and 8 double words.

Supports all combinations of transfer (PCI->PCI, PCI->Internal, Internal->PCI)

Host notification on burst complete and/or transfer complete

Optional external burst initiation control

#### Internal DMA

Supports transfers of up to  $2^{24}-1$  double words in bursts of between 1 and 8 double words.

#### Interrupt

Vertical (frame) synchronous detection

Field synchronous detection

External synchronous error detection

Drawing command error

Drawing command execution end

Burst/Transfer complete

### 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 (x32 Bits)  | 32 Bits        | 2         | 4 Mbytes       |
| FCRAM 16 Mbits (x32 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-/8-bit digital video output is provided. When selecting each 8 bits output, usable external memory bus width is 32 bits only.

#### 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

MB8629x 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.
- Supports direct color mode which specifies RGB with 24 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).

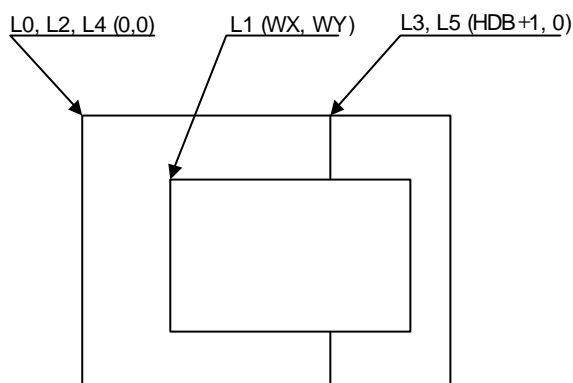
The transparent mode or the blend mode can be selected for overlay.

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

Window display can be performed for the W-layer.

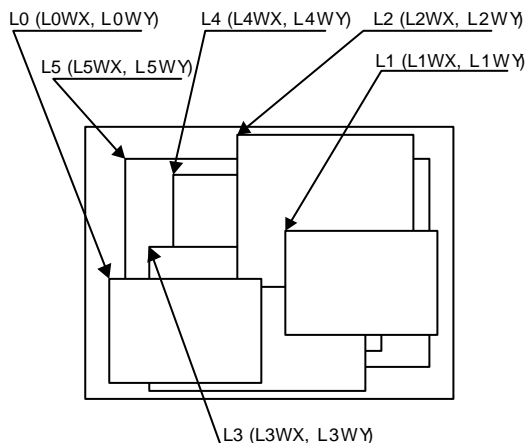
Two palettes are provided: C-layer and M-/B-layer.

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 Video capture function

##### Video input

- The input format is either ITU RBT-656 or RGB.
- The 8-bit video input pin and the external digital video decoder can be connected.
- Video data is stored in graphics memory once and then displayed on the screen in synchronization with the display scan.

##### Scaling

- A scale-up factor 1 to 2 can be used. PAL or NTSC images can be displayed on a wide screen.
- A scale-down factor 1 to 1/32 can be used.
- Picture-in-picture can be used to display drawn images and video images on the same screen.



### **1.4.5 Geometry processing**

The MB86295 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.

## 1.4.6 2D Drawing

### 2D Primitives

MB8629x 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<br>(provisional name) | Draws continuous bold line<br>This primitive is used when interpolating the bold line joint. |
| Triangle                              | Draws triangle   |
| High-speed 2DLine                     | Draws lines<br>Compared to line, this reduces the host CPU processing load.                  |
| Arbitrary polygon                     | Draws arbitrary closed polygon containing concave shapes<br>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 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**

| <b>Attribute</b>       | <b>Description</b>  |
|------------------------|---|
| 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**

| <b>Attribute</b> | <b>Description</b>   |
|------------------|--|
| Enlarge          | Vertically $2 \times 2$<br>Horizontally $\times 2$<br>Vertically and Horizontally $\times 2$ |
| Shrink           | Vertically $1/2 \times 1/2$<br>Horizontally $1/2$<br>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.

## 1.4.7 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

MB8629x supports the Z buffer for hidden plane management.

### 1.4.8 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**

| <b>Attribute</b> | <b>Description</b>                             |
|------------------|--|
| 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<br>While one primitive (polygon, pattern, etc.), being drawn, registered transparent ratio applied<br>A transparent coefficient set for each vertex. A linear-interpolated transparent coefficient applied. |
| Overlay display | Blends top layer pixel color with lower layer pixel color<br>Transparent coefficient set in particular register<br>Registered transparent coefficient applied during one frame scan  |

## Shading

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

**Texture mapping**

MB86295 supports texture mapping to map a image pattern onto the surface of plane. For 2D pattern texture mapping, MB86295 has a built-in pattern memory for a field of up to 64 × 64 pixels (at 16-bit color), which performs high-speed texture mapping. The texture pattern can also be laid out in the graphics memory. In this case, max. 4096 × 4096 pixels can be used.

Drawing of 8-/16-/24-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.

**Texture Mapping**

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

**1.4.9 Others**

**Direct color**

24-bit direct color is supported in addition to 16-bit direct color as a drawing input data. The 24-bit direct color data is laid out on the memory by 32-bit-aligned.

**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.

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

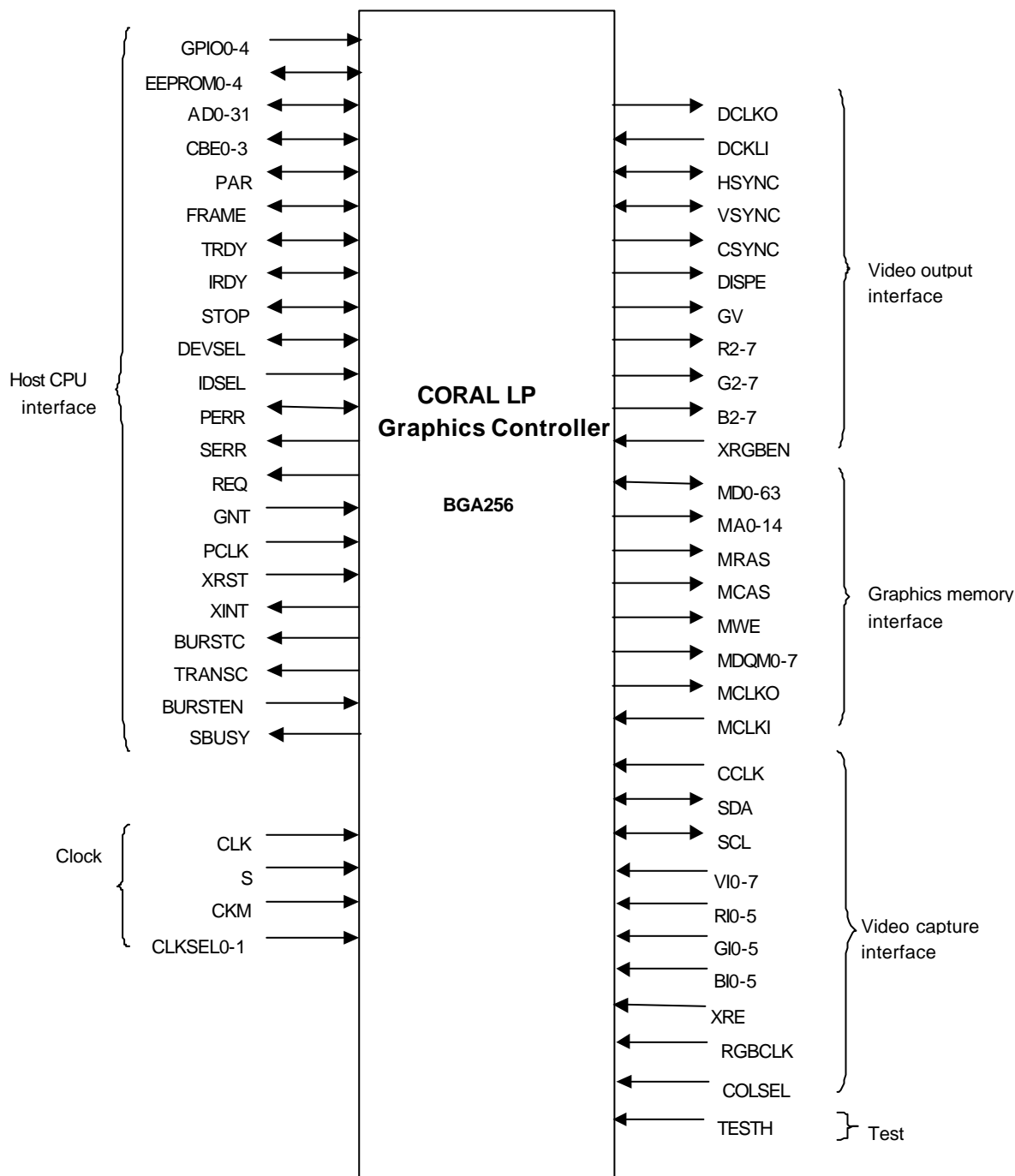


Fig. 2.1 CORAL LP Signal Lines



## 2.2 Pin Assignment

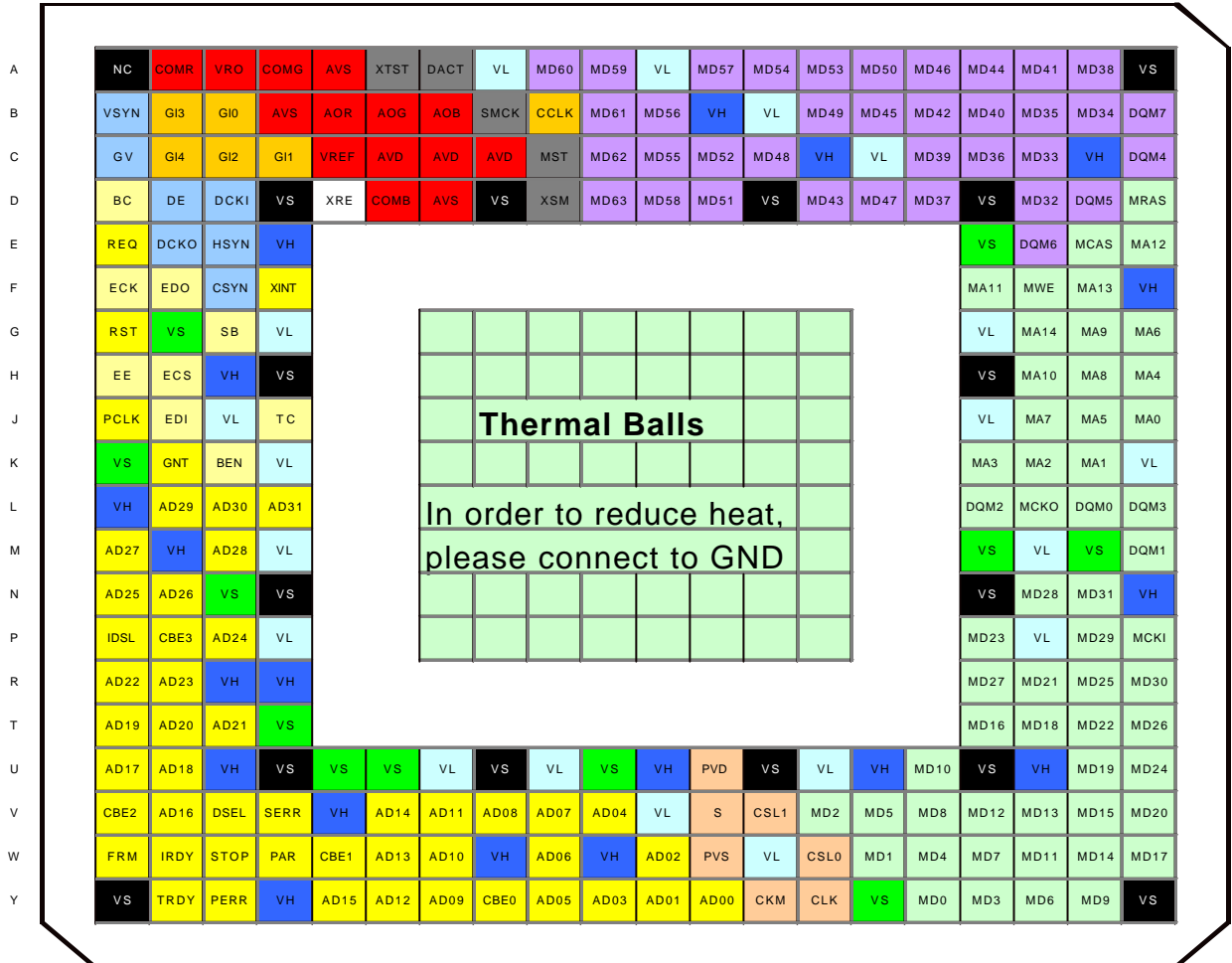
### 2.2.1 Pin assignment diagram

INDEX

TOP VIEW

BGA256

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20



|                     |
|---------------------|
| PCI Interface Pins  |
| Other Host I/f Pins |

|                       |
|-----------------------|
| Memory I/f Pins       |
| Muxed Memory I/f Pins |

|           |
|-----------|
| DAC Pins  |
| Disp Pins |

|              |
|--------------|
| Clock Pins   |
| Capture Pins |
| Test Pins    |

## 2.2.2 Pin assignment table

| JEDEC Number |   | Pin Name | I/O                    | Function  |
|--------------|---|----------|------------------------|---|
| B            | 2 | GI3      | Input                  | RGB Input Green[3]. May also be configured as GPIO input.   |
| C            | 2 | GI4      | Input                  | RGB Input Green[4]. May also be configured as GPIO input.   |
| D            | 3 | DCKI     | Input                  | Video output interface dot clock input.   |
| E            | 4 | VH       | -                      | VDDH - 3.3V power supply.   |
| B            | 1 | VSYN     | I/O                    | Video output interface vertical sync output. Vertical sync input in external sync mode.   |
| E            | 3 | HSYN     | I/O                    | Video output interface horizontal sync output. Horizontal sync input in external sync mode.   |
| D            | 2 | DE       | Output                 | Video output interface display enable period.   |
| C            | 1 | GV       | Output                 | Video output interface graphics/video switch.   |
| F            | 3 | CSYN     | Output                 | Video output interface composite sync output.   |
| E            | 2 | DCKO     | Output                 | Video output interface dot clock signal for display.  |
| D            | 4 | VS       | -                      | VSS - ground.   |
| G            | 4 | VL       | -                      | VDDL 1.8V power supply.   |
| G            | 3 | SB       | I/O                    | Host interface Slave Busy signal. May also be configured as GPIO input/output. In addition this signal is used as RGB input Green[5] and serial interface strobe depending on configuration.  |
| D            | 1 | BC       | I/O                    | Host interface Burst Complete signal. May also be configured as GPIO input/output. In addition this signal is used as RGB input Red[0] and serial interface strobe depending on configuration.  |
| F            | 2 | EDO      | I/O                    | PCI configuration EEPROM data output. May also be configured as GPIO input/output. In addition this signal is used as RGB input Red[1] and serial interface data out depending on configuration.  |
| E            | 1 | REQ      | Output                 | PCI request.  |
| F            | 4 | XINT     | Output<br>(open drain) | External interrupt. By default (and PCI standard) it is active low. However it may be configured as active high if desired.   |
| H            | 3 | VH       | -                      | VDDH 3.3V power supply.   |
| G            | 2 | VS       | -                      | VSS - ground.   |
| F            | 1 | ECK      | I/O                    | PCI configuration EEPROM clock output. May also be configured as GPIO input/output. In addition this signal is used as RGB input Red[2] and serial interface clock out depending on configuration.  |
| H            | 2 | ECS      | I/O                    | PCI configuration EEPROM select output. May also be configured as GPIO input/output. In addition this signal is used as RGB input Red[3] depending on configuration.  |
| J            | 4 | TC       | I/O                    | Host interface transfer complete. May also be configured as GPIO input/output. Note that the state of this pin is latched at external reset to help provide initial I/O configuration. If it is in an active high state then the EEPROM enable register bit is set. |
| J            | 3 | VL       | -                      | VDDL 1.8V power supply.   |
| G            | 1 | XRST     | Input                  | Device reset.   |

|   |   |      |        |  |
|---|---|------|--------|--|
| H | 4 | VS   | -      | VSS - ground.  |
| J | 2 | EDI  | I/O    | PCI configuration EEPROM data input. May also be configured as GPIO input/output. In addition this signal is used as RGB input Red[4] and serial interface data in depending on configuration.   |
| H | 1 | EE   | I/O    | PCI configuration EEPROM enable. May also be configured as GPIO input/output. In addition this signal is used as RGB input Red[5] depending on configuration.  |
| K | 3 | BEN  | I/O    | Host interface burst enable used as an external trigger of the host interface burst controller. May also be configured as GPIO input/output. Note that the state of this pin is latched at external reset to help provide initial I/O configuration. If it is in an active high state then the RGB input enable register bit is set. |
| K | 2 | GNT  | Output | PCI grant.   |
| J | 1 | PCLK | Input  | PCI clock (33MHz).   |
| K | 4 | VL   | -      | VDDL 1.8V power supply.  |
| K | 1 | VS   | -      | VSS - ground.  |
| L | 1 | VH   | -      | VDDH 3.3V power supply.  |
| M | 1 | AD27 | I/O    | PCI address/data bit 27.   |
| L | 2 | AD29 | I/O    | PCI address/data bit 29.   |
| L | 3 | AD30 | I/O    | PCI address/data bit 30.   |
| L | 4 | AD31 | I/O    | PCI address/data bit 31.   |
| N | 1 | AD25 | I/O    | PCI address/data bit 25.   |
| M | 2 | VH   | -      | VDDH 3.3V power supply.  |
| N | 4 | VS   | -      | VSS - ground.  |
| P | 1 | IDSL | Input  | PCI Initialisation Device Select (IDSEL).  |
| M | 3 | AD28 | I/O    | PCI address/data bit 28.   |
| M | 4 | VL   | -      | VDDL 1.8V power supply.  |
| N | 2 | AD26 | I/O    | PCI address/data bit 26.   |
| R | 1 | AD22 | I/O    | PCI address/data bit 22.   |
| P | 2 | CBE3 | I/O    | PCI command/byte enable 3.   |
| N | 3 | VS   | -      | VSS - ground.  |
| R | 4 | VH   | -      | VDDH 3.3V power supply.  |
| T | 1 | AD19 | I/O    | PCI address/data bit 19.   |
| R | 2 | AD23 | I/O    | PCI address/data bit 23.   |
| P | 3 | AD24 | I/O    | PCI address/data bit 24.   |
| U | 1 | AD17 | I/O    | PCI address/data bit 17.   |
| P | 4 | VL   | -      | VDDL 1.8V power supply.  |
| Y | 1 | VS   | -      | VSS - ground.  |
| T | 2 | AD20 | I/O    | PCI address/data bit 20.   |
| R | 3 | VH   | -      | VDDH 3.3V power supply.  |
| V | 1 | CBE2 | I/O    | PCI command/byte enable 2.   |
| U | 2 | AD18 | I/O    | PCI address/data bit 18.   |
| T | 3 | AD21 | I/O    | PCI address/data bit 21.   |
| W | 1 | FRM  | I/O    | PCI Frame.   |
| T | 4 | VS   | -      | VSS - ground.  |
| V | 2 | AD16 | I/O    | PCI address/data bit 16.   |

|   |    |      |                        |   |
|---|----|------|------------------------|---|
| U | 3  | VH   | -                      | VDDH 3.3V power supply.   |
| V | 3  | DSEL | I/O                    | PCI Device Select (DEVSEL).   |
| W | 2  | IRDY | I/O                    | PCI Initiator Ready.  |
| W | 3  | STOP | I/O                    | PCI Stop.   |
| V | 4  | SERR | Output<br>(open drain) | PCI System Error.   |
| U | 5  | VS   | -                      | VSS - ground.   |
| Y | 2  | TRDY | I/O                    | PCI Target Ready.   |
| V | 5  | VH   | -                      | VDDH 3.3V power supply.   |
| W | 4  | PAR  | I/O                    | PCI Parity.   |
| Y | 3  | PERR | I/O                    | PCI Parity Error.   |
| V | 6  | AD14 | I/O                    | PCI address/data bit 14.  |
| W | 5  | CBE1 | I/O                    | PCI command/byte enable 1.  |
| U | 4  | VS   | -                      | VSS - ground.   |
| U | 7  | VL   | -                      | VDDL 1.8V power supply.   |
| V | 7  | AD11 | I/O                    | PCI address/data bit 11.  |
| Y | 4  | VH   | -                      | VDDH 3.3V power supply.   |
| W | 6  | AD13 | I/O                    | PCI address/data bit 13.  |
| Y | 5  | AD15 | I/O                    | PCI address/data bit 15.  |
| U | 6  | VS   | -                      | VSS - ground.   |
| V | 8  | AD08 | I/O                    | PCI address/data bit 8.   |
| W | 7  | AD10 | I/O                    | PCI address/data bit 10.  |
| Y | 6  | AD12 | I/O                    | PCI address/data bit 12.  |
| W | 8  | VH   | -                      | VDDH 3.3V power supply.   |
| U | 9  | VL   | -                      | VDDL 1.8V power supply.   |
| V | 9  | AD07 | I/O                    | PCI address/data bit 7.   |
| Y | 7  | AD09 | I/O                    | PCI address/data bit 9.   |
| U | 8  | VS   | -                      | VSS - ground.   |
| W | 9  | AD06 | I/O                    | PCI address/data bit 6.   |
| Y | 8  | CBE0 | I/O                    | PCI command/byte enable 0.  |
| V | 10 | AD04 | I/O                    | PCI address/data bit 4.   |
| W | 10 | VH   | -                      | VDDH 3.3V power supply.   |
| Y | 9  | AD05 | I/O                    | PCI address/data bit 5.   |
| U | 10 | VS   | -                      | VSS - ground.   |
| Y | 10 | AD03 | I/O                    | PCI address/data bit 3.   |
| Y | 11 | AD01 | I/O                    | PCI address/data bit 1.   |
| Y | 12 | AD00 | I/O                    | PCI address/data bit 0.   |
| W | 11 | AD02 | I/O                    | PCI address/data bit 2.   |
| V | 11 | VL   | -                      | VDDL 1.8V power supply.   |
| U | 11 | VH   | -                      | VDDH 3.3V power supply.   |
| Y | 13 | CKM  | Input                  | Clock Mode. If low then the output from the internal PLL is used as the internal clock. If high then the PCI clock is used. |
| W | 12 | PVS  | -                      | PLL Ground.   |
| U | 13 | VS   | -                      | VSS - ground.   |
| Y | 14 | CLK  | Input                  | Clock input.  |
| V | 12 | S    | Input                  | PLL reset.  |
| U | 12 | PVD  | -                      | PLL 1.8V power supply.  |
| W | 13 | VL   | -                      | VDDL 1.8V power supply.   |

|   |    |      |        |                               |
|---|----|------|--------|-------------------------------|
| Y | 15 | VS   | -      | VSS - ground.                 |
| W | 14 | CSL0 | Input  | Clock rate selection 0.       |
| V | 13 | CSL1 | Input  | Clock rate selection 1.       |
| U | 15 | VH   | -      | VDDH 3.3V power supply.       |
| Y | 16 | MD0  | I/O    | Graphics memory data bit 0.   |
| W | 15 | MD1  | I/O    | Graphics memory data bit 1.   |
| V | 14 | MD2  | I/O    | Graphics memory data bit 2.   |
| Y | 17 | MD3  | I/O    | Graphics memory data bit 3.   |
| U | 14 | VL   | -      | VDDL 1.8V power supply.       |
| Y | 20 | VS   | -      | VSS – ground.                 |
| W | 16 | MD4  | I/O    | Graphics memory data bit 4.   |
| V | 15 | MD5  | I/O    | Graphics memory data bit 5.   |
| Y | 18 | MD6  | I/O    | Graphics memory data bit 6.   |
| W | 17 | MD7  | I/O    | Graphics memory data bit 7.   |
| V | 16 | MD8  | I/O    | Graphics memory data bit 8.   |
| Y | 19 | MD9  | I/O    | Graphics memory data bit 9.   |
| U | 16 | MD10 | I/O    | Graphics memory data bit 10.  |
| W | 18 | MD11 | I/O    | Graphics memory data bit 11.  |
| V | 17 | MD12 | I/O    | Graphics memory data bit 12.  |
| V | 18 | MD13 | I/O    | Graphics memory data bit 13.  |
| W | 19 | MD14 | I/O    | Graphics memory data bit 14.  |
| V | 19 | MD15 | I/O    | Graphics memory data bit 15.  |
| U | 18 | VH   | -      | VDDH 3.3V power supply.       |
| T | 17 | MD16 | I/O    | Graphics memory data bit 16.  |
| W | 20 | MD17 | I/O    | Graphics memory data bit 17.  |
| T | 18 | MD18 | I/O    | Graphics memory data bit 18.  |
| U | 19 | MD19 | I/O    | Graphics memory data bit 19.  |
| V | 20 | MD20 | I/O    | Graphics memory data bit 20.  |
| R | 18 | MD21 | I/O    | Graphics memory data bit 21.  |
| T | 19 | MD22 | I/O    | Graphics memory data bit 22.  |
| U | 17 | VS   | -      | VSS - ground.                 |
| P | 17 | MD23 | I/O    | Graphics memory data bit 23.  |
| P | 18 | VL   | -      | VDDL 1.8V power supply.       |
| U | 20 | MD24 | I/O    | Graphics memory data bit 24.  |
| R | 19 | MD25 | I/O    | Graphics memory data bit 25.  |
| T | 20 | MD26 | I/O    | Graphics memory data bit 26.  |
| R | 17 | MD27 | I/O    | Graphics memory data bit 27.  |
| N | 18 | MD28 | I/O    | Graphics memory data bit 28.  |
| P | 19 | MD29 | I/O    | Graphics memory data bit 29.  |
| R | 20 | MD30 | I/O    | Graphics memory data bit 30.  |
| N | 19 | MD31 | I/O    | Graphics memory data bit 31.  |
| M | 17 | VS   | -      | VSS - ground.                 |
| M | 18 | VL   | -      | VDDL 1.8V power supply.       |
| P | 20 | MCKI | Input  | Graphics memory clock input.  |
| N | 17 | VS   | -      | VSS - ground.                 |
| M | 19 | VS   | -      | VSS - ground.                 |
| N | 20 | VH   | -      | VDDH 3.3V power supply.       |
| L | 18 | MCKO | Output | Graphics memory clock output. |

|   |    |      |        |   |
|---|----|------|--------|---|
| L | 19 | DQM0 | Output | Graphics memory data mask 0.  |
| M | 20 | DQM1 | Output | Graphics memory data mask 1.  |
| L | 17 | DQM2 | Output | Graphics memory data mask 2.  |
| L | 20 | DQM3 | Output | Graphics memory data mask 3.  |
| K | 20 | VL   | -      | VDDL 1.8V power supply.   |
| J | 20 | MA0  | Output | Graphics memory address bit 0.  |
| K | 19 | MA1  | Output | Graphics memory address bit 1.  |
| K | 18 | MA2  | Output | Graphics memory address bit 2.  |
| K | 17 | MA3  | Output | Graphics memory address bit 3.  |
| H | 20 | MA4  | Output | Graphics memory address bit 4.  |
| J | 19 | MA5  | Output | Graphics memory address bit 5.  |
| H | 17 | VS   | -      | VSS - ground.   |
| G | 20 | MA6  | Output | Graphics memory address bit 6.  |
| J | 18 | MA7  | Output | Graphics memory address bit 7.  |
| J | 17 | VL   | -      | VDDL 1.8V power supply.   |
| H | 19 | MA8  | Output | Graphics memory address bit 8.  |
| F | 20 | VH   | -      | VDDH 3.3V power supply.   |
| G | 19 | MA9  | Output | Graphics memory address bit 9.  |
| H | 18 | MA10 | Output | Graphics memory address bit 10.   |
| F | 17 | MA11 | Output | Graphics memory address bit 11.   |
| E | 20 | MA12 | Output | Graphics memory address bit 12.   |
| F | 19 | MA13 | Output | Graphics memory address bit 13.   |
| G | 18 | MA14 | Output | Graphics memory address bit 14.   |
| D | 20 | MRAS | Output | Graphics memory row address strobe.   |
| G | 17 | VL   | -      | VDDL 1.8V power supply.   |
| A | 20 | VS   | -      | VSS - ground.   |
| E | 19 | MCAS | Output | Graphics memory column address strobe.  |
| F | 18 | MWE  | Output | Graphics memory write enable.   |
| C | 20 | DQM4 | Output | Graphics memory data mask 4.  |
| D | 19 | DQM5 | Output | Graphics memory data mask 5.  |
| E | 18 | DQM6 | Output | Graphics memory data mask 6. May also be configured as Blue[0] for the RGB output.  |
| B | 20 | DQM7 | Output | Graphics memory data mask 7. May also be configured as Blue[1] for the RGB output.  |
| E | 17 | VS   | -      | VSS - ground.   |
| C | 19 | VH   | -      | VDDH 3.3V power supply.   |
| D | 18 | MD32 | I/O    | Graphics memory data bit 32. May also be configured as Blue[2] for the RGB output.  |
| C | 18 | MD33 | I/O    | Graphics memory data bit 32. May also be configured as Blue[3] for the RGB output.  |
| B | 19 | MD34 | I/O    | Graphics memory data bit 32. May also be configured as Blue[4] for the RGB output.  |
| B | 18 | MD35 | I/O    | Graphics memory data bit 32. May also be configured as Blue[5] for the RGB output.  |
| C | 17 | MD36 | I/O    | Graphics memory data bit 32. May also be configured as Blue[6] for the RGB output.  |
| D | 16 | MD37 | I/O    | Graphics memory data bit 32. May also be configured as Blue[7] for the RGB output.  |
| A | 19 | MD38 | I/O    | Graphics memory data bit 32. May also be configured as Green[0] for the RGB output. |

|   |    |      |     |  |
|---|----|------|-----|--|
| C | 16 | MD39 | I/O | Graphics memory data bit 32. May also be configured as Green[1] for the RGB output.  |
| B | 17 | MD40 | I/O | Graphics memory data bit 32. May also be configured as Green[2] for the RGB output.  |
| A | 18 | MD41 | I/O | Graphics memory data bit 32. May also be configured as Green[3] for the RGB output.  |
| C | 15 | VL   | -   | VDDL 1.8V power supply.  |
| B | 16 | MD42 | I/O | Graphics memory data bit 32. May also be configured as Green[4] for the RGB output.  |
| D | 17 | VS   | -   | VSS - ground.  |
| D | 14 | MD43 | I/O | Graphics memory data bit 32. May also be configured as Green[5] for the RGB output.  |
| C | 14 | VH   | -   | VDDH 3.3V power supply.  |
| A | 17 | MD44 | I/O | Graphics memory data bit 32. May also be configured as Green[6] for the RGB output.  |
| B | 15 | MD45 | I/O | Graphics memory data bit 32. May also be configured as Green[7] for the RGB output.  |
| A | 16 | MD46 | I/O | Graphics memory data bit 32. May also be configured as Red[0] for the RGB output.R0  |
| D | 15 | MD47 | I/O | Graphics memory data bit 32. May also be configured as Red[1] for the RGB output.R1  |
| C | 13 | MD48 | I/O | Graphics memory data bit 32. May also be configured as Red[2] for the RGB output.R2  |
| B | 14 | MD49 | I/O | Graphics memory data bit 32. May also be configured as Red[3] for the RGB output.R3  |
| A | 15 | MD50 | I/O | Graphics memory data bit 32. May also be configured as Red[4] for the RGB output.R4  |
| B | 13 | VL   | -   | VDDL 1.8V power supply.  |
| D | 12 | MD51 | I/O | Graphics memory data bit 51. May also be configured as Red[5] for the RGB output.R5  |
| C | 12 | MD52 | I/O | Graphics memory data bit 52. May also be configured as Red[6] for the RGB output.R6  |
| A | 14 | MD53 | I/O | Graphics memory data bit 53. May also be configured as Red[7] for the RGB output. R7   |
| D | 13 | VS   | -   | VSS - ground.  |
| B | 12 | VH   | -   | VDDH 3.3V power supply.  |
| A | 13 | MD54 | I/O | Graphics memory data bit 54. May also be configured as I <sup>2</sup> C serial data (SDA).   |
| C | 11 | MD55 | I/O | Graphics memory data bit 55. May also be configured as I <sup>2</sup> C serial clock (SCL).  |
| B | 11 | MD56 | I/O | Graphics memory data bit 56. May also be configured as ITU-RBT-656 video capture data input bit 0 (V10). When the RGB input is enabled this pin acts as Blue[0]. |
| A | 12 | MD57 | I/O | Graphics memory data bit 57. May also be configured as ITU-RBT-656 video capture data input bit 1 (V11). When the RGB input is enabled this pin acts as Blue[1]. |
| D | 11 | MD58 | I/O | Graphics memory data bit 58. May also be configured as ITU-RBT-656 video capture data input bit 2 (V12). When the RGB input is enabled this pin acts as Blue[2]. |

|   |    |      |        |  |
|---|----|------|--------|--|
| A | 11 | VL   | -      | VDDL 1.8V power supply.  |
| A | 10 | MD59 | I/O    | Graphics memory data bit 59. May also be configured as ITU-RBT-656 video capture data input bit 3 (VI3). When the RGB input is enabled this pin acts as Blue[3]. |
| A | 9  | MD60 | I/O    | Graphics memory data bit 60. May also be configured as ITU-RBT-656 video capture data input bit 4 (VI4). When the RGB input is enabled this pin acts as Blue[4]. |
| B | 10 | MD61 | I/O    | Graphics memory data bit 61. May also be configured as ITU-RBT-656 video capture data input bit 5 (VI5). When the RGB input is enabled this pin acts as Blue[5]. |
| C | 10 | MD62 | I/O    | Graphics memory data bit 62. May also be configured as ITU-RBT-656 video capture data input bit 6 (VI6). When the RGB input is enabled this pin acts as HSYNC.   |
| D | 10 | MD63 | I/O    | Graphics memory data bit 63. May also be configured as ITU-RBT-656 video capture data input bit 7 (VI7). When the RGB input is enabled this pin acts as VSYNC.   |
| A | 8  | VL   | -      | VDDL 1.8V power supply.  |
| B | 9  | CCLK | Input  | ITU-RBT-656 video capture clock input.   |
| D | 8  | VS   | -      | VSS - ground.  |
| A | 7  | DACT | Input  | Test signal.   |
| C | 9  | MST  | Input  | Test signal.   |
| D | 9  | XSM  | Input  | Test Signal.   |
| B | 8  | SMCK | Input  | Test Signal.   |
| A | 6  | XTST | Input  | Test Signal.   |
| B | 7  | AOB  | Output | Analog Signal (B) output   |
| C | 8  | AVD2 | -      | Analog Power Supply(3.3V)  |
| D | 6  | COMB | Output | Analog B Signal Compensation pin   |
| A | 5  | AVS2 | -      | Analog Ground  |
| B | 6  | AOG  | Output | Analog Singnal (G) output  |
| C | 7  | AVD1 | -      | Analog Power Supply(3.3V)  |
| A | 4  | COMG | Output | Analog G Signal Compensation pin   |
| D | 7  | AVS1 | -      | Analog Ground  |
| A | 1  | NC   | -      | Not connected.   |
| B | 5  | AOR  | Output | Analog Singnal (R) output  |
| C | 6  | AVD0 | -      | Analog Power Supply(3.3V)  |
| A | 3  | VRO  | Output | Analog Reference current output  |
| B | 4  | AVS0 | -      | Analog Ground  |
| C | 5  | VREF | Input  | Analog Reference Voltage input   |
| A | 2  | COMR | Output | Analog R Signal Compensation pin   |
| D | 5  | XRE  | Input  | RGB output/video input/I <sup>2</sup> C enable.  |
| B | 3  | GI0  | GI0    | RGB Input Green[0]. May also be configured as GPIO input.  |
| C | 4  | GI1  | GI1    | RGB Input Green[1]. May also be configured as GPIO input.  |
| C | 3  | GI2  | GI2    | RGB Input Green[2]. May also be configured as GPIO input.  |



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 (1.8 V)    |
| OPEN                 | : Do not connect anything.    |
| TESTH                | : Input a 3.3 V-power supply. |
| AVS                  | : Analog Ground               |
| AVD                  | : Analog power supply (3.3 V) |

- It is recommended that  $PLL V_{DD}$  should be isolated on the PCB.
- It is recommended that AVD 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   |
|----------|------------------------|---|
| AD0-31   | In/Out                 | PCI Address/Data  |
| CBE0-3   | In/Out                 | PCI Bus Command/Byte Enable   |
| PAR      | In/Out                 | PCI Parity  |
| FRM      | In/Out                 | PCI Cycle Frame   |
| TRDY     | In/Out                 | PCI Target Ready  |
| IRDY     | In/Out                 | PCI Initiator Ready   |
| STOP     | In/Out                 | PCI Stop  |
| DSEL     | In/Out                 | PCI Device Select   |
| IDSEL    | Input                  | PCI Initialisation Device Select  |
| PERR     | In/Out                 | PCI Parity Error  |
| SERR     | Output<br>(Open Drain) | System Error  |
| REQ      | Output                 | PCI Bus Master Request  |
| GNT      | Input                  | PCI Bus Grant   |
| PCLK     | Input                  | PCI Clock – 33MHz   |
| XRST     | Input                  | System Reset (including PCI)  |
| XINT     | Output<br>(Open Drain) | Interrupt   |
| BC       | Output                 | Burst Complete. Indicates a burst is complete when using the DMA/Burst Controller.<br><br>This pin may also be configured as a GPIO Input/Output and acts as RI0 (Red Input 0) when the RGB Input is enabled.   |
| TC       | Output                 | Transfer Complete. Indicates that a whole transfer is complete when using the DMA/Burst Controller.<br><br>This may also be configured as a GPIO Input/Output.<br><br>In addition this pin may be used to automatically enable the EEPROM at the reset phase. To do this a pull up should be applied. |
| BEN      | Input                  | Enables the Burst Controller to start/continue execution.<br><br>This pin may also be configured as a GPIO Input/Output.<br><br>In addition this pin may be used to automatically enable the RGB Input pins as RGB inputs. To do this a pull up should be applied.                                    |
| SB       | Output                 | Slave Busy. Indicates that the PCI Slave is busy completing a write transfer.<br><br>This pin may also be configured as a GPIO Input/Output, the Serial Interface Strobe Output and acts as GI5 (Green Input 5) when the RGB Input is enabled.  |

|       |        |   |
|-------|--------|---|
| EE    | Input  | EEPROM Enable. Enables the PCI EEPROM Configuration.<br>This pin may also be configured as a GPIO Input/Output and acts as RI5 (Red Input 5) when the RGB Input is enabled. |
| ECS   | Output | EEPROM Chip Select . This pin may also be configured as a GPIO Input/Output and acts as RI3 (Red Input 3) when the RGB Input is enabled.                                    |
| ECK   | Output | EEPROM Clock. This pin may also be configured as a GPIO Input/Output, the Serial Interface Data Input and acts as RI2 (Red Input 2) when the RGB Input is enabled.          |
| EDO   | Output | EEPROM Data Out. This pin may also be configured as a GPIO Input/Output, the Serial Interface Data Output and acts as RI1 (Red Input 1) when the RGB Input is enabled.      |
| EDI   | Input  | EEPROM Data In. This pin may also be configured as a GPIO Input/Output, the Serial Interface Data Input and acts as RI4 (Red Input 4) when the RGB Input is enabled.        |
| GI0-4 | Input  | GPIO Inputs. These pins also act as GI0-4 (Green Inputs 0-4) when the RGB Input is enabled.   |

The EE, ECK, ECS, EDO, EDI, BC, TC, SB and BEN signals can all be configured as GPIO inputs/outputs and default to GPIO inputs at reset unless otherwise specified by the reset control pins (TC, BEN) which can be used to enable the EEPROM or the RGB input. The GI0-4 signals can be GPIO inputs only, which is their default state unless the RGB input is enabled in which case they are used as Green[0-4].

The Host Interface also has a serial interface function built in. This uses the EDI/EDO signals as data in/out, the ECK pin as a serial clock output and the SB pin as a strobe output. The serial interface may only be used when neither the EEPROM nor the RGB input is in use.

Once the device has been reset all configuration of the host interface related pins is done using the IO Mode register (IOM).

Note that to enable the RGB input the XRE signal must be active low and also the appropriate register in the capture engine must be configured.

## 2.3.2 Video output interface

**Table 2-2 Video Output Interface Pins**

| Pin name | I/O           | Description   |
|----------|---------------|---|
| DCKO     | Output        | Dot clock signal for display  |
| DCKI     | Input         | Dot clock signal input  |
| HSYN     | I/O           | Horizontal sync signal output<br>Horizontal sync input <in external sync mode>                                  |
| VSYN     | I/O           | Vertical sync signal output<br>Vertical sync input <in external sync mode>                                      |
| CSYN     | Output        | Composite sync signal output  |
| DE       | Output        | Display enable period signal  |
| GV       | Output        | Graphics/video switch   |
| R7-0     | Output        | Digital picture (R) output. . These pins are multiplexed MD53-46. These pins are available when XRE=0.          |
| G7-0     | Output        | Digital picture (G) output. . These pins are multiplexed MD45-38. These pins are available when XRE=0.          |
| B7-0     | Output        | Digital picture (B) output. These pins are multiplexed MD37-32 and DQM7-6. These pins are available when XRE=0. |
| XRE      | Input         | Signal to switch between digital RGB output, capture signals /memory bus (MD 63-32, DQM7-6)                     |
| AOR      | Analog Output | Analog Signal (R) output  |
| AOG      | Analog Output | Analog Signal (G) output  |
| AOB      | Analog Output | Analog Signal (B) output  |
| COMR     | Analog        | Analog (R) Compensation output  |
| COMG     | Analog        | Analog (G) Compensation output  |
| COMB     | Analog        | Analog (B) Compensation output  |
| VREF     | Analog        | Analog Voltage Reference input  |
| VRO      | Analog        | Analog Reference Current output   |

It is possible to output digital RGB when XRE = 0 (Memory bus = 32bit).

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.

AOR, AOG and AOB must be terminated at 75 ohm.

1.1 V is input to VREF. A bypass capacitor ( with good high-frequency characteristics ) must be inserted between VREF and AVS.

COMR, COMG and COMB are tied to analog VDD via 0.1 uF ceramic capacitors.

VRO must be pulled down to analog ground by a 2.7 k ohm resistor.

### 2.3.3 Video capture interface

#### 1. ITU-656 Input Signals

**Table 2-3 Video Capture Interface Pins**

| Pin name | I/O   | Description  |
|----------|-------|--|
| CCLK     | Input | Digital video input clock signal input                                 |
| V17-0    | Input | ITU656 Digital video data input. These pins are multiplexed MD63-MD56. |

Inputs ITU-RBT-656 format digital video signal

Digital video data input can be used only when the XRE pin is “0”. MD63-MD56 are assigned as the digital video data input pins.

When video capture is not used and the XRE pin is 0, input the “High” level to MD63-MD56.

#### 2. RGB Input Signals

The signals used for video capture are not assigned on dedicated pins but share the same pins with other functions. There is a set of signals corresponding to the RGB capture modes.

##### (1) Direct Input Mode

| Name   | IO | Function   |
|--------|----|--|
| RGBCLK | In | Clock for RGB input. This pin is multiplexed CCLK.                             |
| RI5-0  | In | Red component value. These pins are multiplexed EE, EDI, ECS, ECK, EDO and BC. |
| GI5-0  | In | Green component value. These pins are multiplexed SB and GPI4-GPI0.            |
| BI5-0  | In | Blue component value. These pins are multiplexed MD61-MD56.                    |
| VSYNCI | In | Vertical sync for RGB capture. This pin is multiplexed MD63.                   |
| HSYNCI | In | Horizontal sync for RGB capture. This pin is multiplexed MD63.                 |

Note :

- the RGB bit of VCM(video capture mode) register enables RGB input mode of video capture.

### 2.3.4 I<sup>2</sup>C interface

| Pin name | I/O | Description  |
|----------|-----|--|
| SDA      | I/O | I <sup>2</sup> C or Video capture test signal. This pin is multiplexed MD54. |
| SCL      | I/O | I <sup>2</sup> C or Video capture test signal. This pin is multiplexed MD55. |

I<sup>2</sup>C interface signals can be used only when the XRE pin is "0". MD55-MD54 are assigned as the I<sup>2</sup>C interface pins.

When I<sup>2</sup>C interface is not used and the XRE pin is 0, input the "High" level to MD63-MD56.

Note)

Input voltage level is 3.3V. Please be careful, it does not support to 5V input.

(The device whose output voltage is 5V is not connectable.)

### 2.3.5 Graphics memory interface

Graphics memory interface pins

| Pin name    | I/O    | Description  |
|-------------|--------|--|
| MD31 - MD0  | I/O    | Graphics memory bus data   |
| MD53 - MD32 | I/O    | Graphics memory bus data or digital R7-0, G7-0, B7-2 output (when XRE = 0) |
| MD55 - MD54 | I/O    | Graphics memory bus data or SCL, SDA (when XRE=0)                          |
| MD63 - MD56 | I/O    | Graphics memory bus data or video input (when XRE=0)                       |
| MA0 to 14   | Output | Graphics memory bus data   |
| MRAS        | Output | Row address strobe   |
| MCAS        | Output | Column address strobe  |
| MWE         | Output | Write enable   |
| DQM5 - DQM0 | Output | Data mask  |
| DQM7 - DQM6 | Output | Data mask or digital B1-0 output (when XRE = 0)                            |
| 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 SD RAM (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. .

Connect MCLKI to MCLK0.

When XRE is fixed at "1", MD63 - MD32 and DQM7 - DQM6 can be used as graphics memory interface.

When XRE is fixed at "0", these signals can be used as digital RGB output and digital video data input.

### 2.3.6 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        |
| CSL [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   | PCI bus clock selected            |

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

| CSL1 | CSL0 | 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                         |                     |                         |



### 2.3.7 Test pins

**Table 2-5 Test Pins**

| Pin name | I/O   | Description        |
|----------|-------|--------------------|
| TESTH    | Input | Input 3.3-V power. |

### 2.3.8 Reset sequence

See *Section 10.3.2*.

### 2.3.9 How to switch 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 2-6 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

## 3. HOST INTERFACE

The Coral LP has a 33MHz, 32-bit PCI host interface compliant to PCI version 2.1. It includes both PCI master and PCI slave functions and an internal DMA/burst controller for multi-burst transfers of large quantities of data between all combinations of PCI data space and Coral LP internal data space. PCI EEPROM configuration is also supported.

Additional functions provided by the host interface are optional host interface status/control signals which may aid in the reduction of PCI retries, the provision of general purpose IO (GPIO) signals for control of external devices via the PCI interface including support for a simple serial interface.

### 3.1 Standard PCI Slave Accesses

An external PCI master will access the Coral LP as a PCI slave.

#### 3.1.1 PCI Slave Write

For a PCI slave write, data will be “posted” into a temporary buffer from where it is written to the target internal client. This temporary buffer is 8 dwords deep. PCI slave writes of any size are supported but typically a retry will occur after each 8 dword burst. Note that when writing to the display list FIFO a burst should be no more than 16 dwords (64 bytes) due to FIFO address space limitations.

When the write from the temporary buffer to the internal client is being performed the Slave Busy (SB) signal becomes active. While this is happening PCI accesses will be rejected. If the SB signal is used then PCI retries may be reduced.

#### 3.1.2 PCI Slave Read

For a PCI slave read the read requested will be passed to an internal client from where data will be fetched into the temporary buffer (8 dwords deep). Typically a retry will occur to actually fetch the data. In order to fetch the correct number of words from the read address the burst size must be specified. This is done by writing to the Slave Burst Read Size (SRBS) register. Bursts of between 1 and 8 dwords are supported. If the PCI master retries and reads less than the specified burst size then the remaining dwords will be discarded. This means that the Slave Burst Read Size can be permanently configured as 8 dwords. However there will be an increased latency on the pre-fetch stage if this is done.

### 3.2 Burst Controller Accesses (including PCI Master)

The Coral LP host interface includes a burst controller which can be used for transferring large quantities of contiguous data between all combinations (source/destination) of PCI data space and Coral LP internal data space. Control/status monitoring is done through internal registers with the optional aid of external signals – Burst Complete (BC), Transfer Complete (TC) and Burst Enable (BEN).

A transfer can be any number of dwords from 1 to  $16777215 (2^{24}-1)$  dwords, split up into a number of individual bursts of size from 1 to 8 dwords. If the transfer size is not an integer multiple of the burst size then the final burst of the transfer will be less than the configured burst size. A transfer is from a source address to a destination address with the source/destination being in either PCI or Coral LP data space as appropriate to the transfer mode. After each burst of a transfer the source and/or the destination address may be incremented (or not) by the burst size enabling transfers both to/from

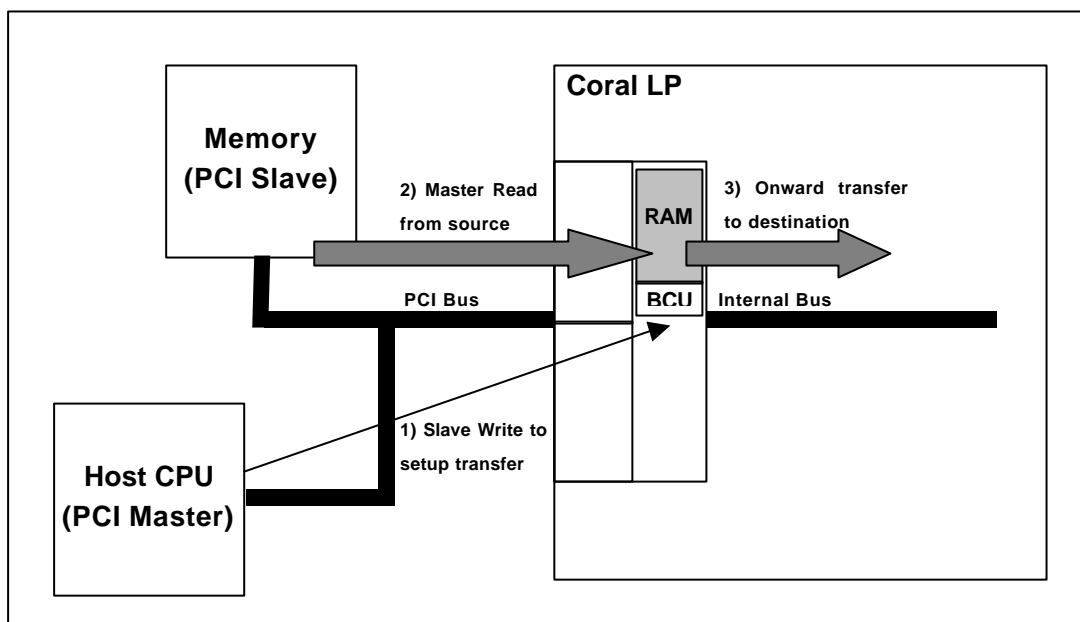
memory and also FIFO-like sources/destinations. Note that when writing to the display list FIFO, the destination address should be configured to **not** increment between bursts.

### 3.2.1 Transfer Modes

There are 6 transfer modes configurable through the Burst Setup Register (BSR). These are:

| Mode | Function  |
|------|---|
| 000b | Slave Mode PCI to Coral LP. In this mode a PCI master writes bursts of data directly into a temporary buffer from where it is transferred to the destination address by the Burst Controller. While this can also be accomplished using simple PCI Slave writes there are benefits in using this mode when transferring large quantities of data. For a normal PCI write the Coral LP PCI slave interface is blocked until the write to the destination address has completed. Depending on the destination there may be some delay in doing this. Using the burst controller the data is transferred out of the PCI interface into the temporary buffer from where it is transferred to the destination. In this case the PCI slave interface is quickly cleared and so other operations can take place or the next burst can be written in. |
| 001b | Slave Mode Coral LP to PCI. In this mode the burst controller reads data from a Coral LP internal address into its temporary buffer and then waits for the data to be read using a PCI slave read from this buffer's address. While this can also be accomplished using simple PCI Slave reads there are benefits in using this mode when transferring large quantities of data. A normal PCI read will typically be accomplished by a PCI read request followed by a retry to fetch the data. Using this mode the burst controller can be used to automatically fetch the next data to be read. Depending on internal latencies this should reduce the number of retries.  |
| 010b | Coral LP to Coral LP. In this mode data is read from a source address internal to Coral LP into a temporary buffer, from where it is written to a destination, also internal to Coral LP. An example of where this mode may be used is to transfer display list data from graphics memory to the display list FIFO.   |
| 011b | Reserved.   |
| 100b | PCI to Coral LP (PCI Master read). In this mode the source address is in PCI data space and the destination address internal to Coral LP. For each burst of the transfer "burst size" dwords of data are read as a PCI Master read into a temporary buffer, from where they are written to the internal destination address. An example of where this mode will be used is display list transfer to the FIFO/graphics memory.   |
| 101b | Coral LP to PCI (PCI Master write). In this mode the source address is internal to Coral LP and the destination address is in PCI data space. For each burst of the transfer "burst size" dwords of data are fetched from an internal address into a temporary buffer, from where they are written to the destination address using a PCI master write. An example of where this mode may be used is to transfer graphics memory data to external PCI memory.   |
| 110b | PCI to PCI (PCI Master read/write). This mode is effectively a PCI to PCI DMA. Data is read from a source address in PCI data space into a temporary buffer from where it is written to the destination address, also in PCI data space.  |
| 111b | Reserved.   |

The figure below illustrates a PCI to Coral (Master Read) transfer. The Host CPU will program up the BCU registers (using normal PCI Slave writes) and trigger the transfer. The Coral then reads data from the source memory as a PCI Master and writes to the destination inside the Coral.



All other BCU transfers use the BCU RAM in a similar way but with source/destination dependent on transfer type.

### 3.2.2 Burst Controller Control/Status

All setup/control and status for the burst controller can be done through registers. These provide ways of specifying the parameters for a burst (source/destination address, address increment (or not) and burst/transfer size. In addition, a transfer can be started/paused/aborted and also its progress monitored using the enable and status registers.

The key status indicators are Burst Complete and Transfer Complete, which become active at the end of each burst/transfer respectively. These may either be active high or toggle state at the end of each burst/transfer. When active high they will have to be cleared after each burst/transfer. This may be done using a clear on read mode (default) or by manually writing to the appropriate register.

The burst/transfer complete indications are also available through the main interrupt status register (IST) and can trigger the main external interrupt (XINT). If being used for this they must be configured as active high (ie. not toggle mode). In addition burst/transfer complete can be made available as external signals (BC/TC) for connection directly to an external device (eg. through some form of GPIO or interrupt).

Normally a transfer will be configured and enabled using internal registers. However it is possible to configure the transfer but not actually start it. An external signal (BEN) can then be used to trigger the transfer and pause it between bursts. This may be useful, for example, when doing PCI Master reads from a client which takes time to pre-fetch more data for the next burst.

### **3.3 FIFO Transfers**

Unlike Coral LQ/Coral LB there are no specific transfer mechanisms to write data into the display list FIFO. A write to the FIFO interface occurs automatically when it is specified as a destination address either for a PCI Slave Write or in a Burst Controller transfer. If this is not desired, and the main internal bus should be used, then the Override FIFO Use register may be set. Under normal circumstances there should be no need to use this feature.

As previously stated when the FIFO address is specified as the destination in the Burst Controller the destination should not be incremented after each burst. This will not happen automatically and must be specifically configured. In addition when writing to the FIFO using a PCI Slave Write the FIFO address space is limited to 16 dwords (64 bytes). This means that a PCI Slave Write burst to the FIFO must not be more than 16 dwords, otherwise data will be written to invalid locations for retries after 2 bursts of 8 dwords.

In normal mode when writing to the FIFO, data is written to the Geometry Engine FIFO from where it is transferred either directly to the Draw Engine FIFO or to the Geometry Engine, depending on the command. If the Geometry Engine is not in use then a direct write to the Draw Engine FIFO can be accomplished by setting Cremson Mode (CM register).

When the burst controller is used to transfer data to the FIFO the rate of bursts is controlled using the current FIFO status. When the FIFO is nearly full the next burst will not occur until data is processed by the Geometry/Draw Engine. This guarantees that there will always be space for the next burst of data. If this feature is not required then it can be disabled using the FIFO Burst Mode (FBM) register.

### **3.4 GPIO/Serial Interface**

The Host Interface supports optional register mapped General Purpose IO (GPIO) and Serial Interface functions.

#### **3.4.1 GPIO**

Depending on configuration there are up to 14 GPIO signals. 5 of these (GI0, GI1, GI2, GI3, GI4) are inputs only. The remainder (BEN,SB,TC,BC,EE,ECS,ECK,EDI, EDO) may be either input or output. All reset to GPIO inputs unless otherwise configured using the reset configuration mechanism to enable the EEPROM/RGB input.

Operation of the GPIO is simply through the reading of the GPIO Data (GD) register for GPIO Inputs and writing to this register (with write mask) for the GPIO Outputs. GPIO Inputs may be configured selectively to trigger an external interrupt (via the interrupt status register (IST)) when they change state (0->1 or 1->0 transition).

#### **3.4.2 Serial Interface**

A simple serial interface is available depending on configuration. This uses the EDI/EDO pins as serial data input/output, the ECK as the serial clock output and SB as the serial interface strobe. The serial data out signal may be tri-stated when not in use.

Up to 8 bits of data is shifted out/in based on the serial clock. This may be  $1/16$ ,  $1/32$ ,  $1/64$  or  $1/128$  of the main internal clock. The clock polarity may be specified to be high/low and it may be gated when the serial interface is inactive.

The strobe signal has configurable polarity and may be active only for the first cycle of a transfer or the complete transfer. It may also be disabled completely. Configured strobe settings may be overridden on a transfer by transfer basis if required.

An interrupt may be generated when a transfer is complete.

### **3.5 Interrupt**

The Coral LP MB86295 issues interrupt requests to the host CPU. The following interrupt triggers may enabled/disabled using the Interrupt Mask Register (IMASK).

- Vertical synchronization detect
- Field synchronization detect
- External synchronization error detect
- Drawing command error
- Drawing command execution end
- Internal Bus/FIFO Timeout
- Serial Interface transfer complete
- GPIO input change
- Burst Complete
- Transfer Complete
- Host Interface Fatal (PCI error)
- Address Error (invalid address accessed)

In addition the I<sup>2</sup>C interface can trigger an interrupt, but this is non-maskable through the IMASK register.

By default the external interrupt is active low (PCI standard) and is open drain. If required it may be configured to be active high using the Interrupt Polarity (IP) register.

Once an interrupt is detected by the host it can read the interrupt status register (IST) to determine the source of the interrupt. The exception to this is the I<sup>2</sup>C interrupt. Once read the interrupt status register must be cleared by writing 0 to the appropriate bit/bits (selective clearing is possible). Note that the Burst Complete/Transfer Complete interrupts must be cleared by writing to the Burst Status (BST) register.

#### **3.5.1 Internal Bus/FIFO timeout**

When accessing an internal client through the internal bus or writing to the FIFO it is possible that an unacceptable delay (possibly a lockup situation) occurs. This should not normally happen, but as a safety feature a timeout is available to allow for graceful termination of the offending access. Separate timeout periods for the internal bus and FIFO can be programmed and enabled (using the BTV, FTV and TCS registers).

When an access is made to a client and no response is obtained within the specified timeout period then the access is terminated and an interrupt generated. The Timeout Control/Status (TCS) register may be read to determine the offending client. Depending on circumstance a soft or firm reset may then be issued (through the SRST or FRST register) to clear the problem.

### 3.5.2 Address Error Interrupt

Certain addresses are invalid depending on operation. For example the Burst Controller cannot access the Host Interface internal registers. If an attempt is made to do this then the access will be terminated and an Address Error Interrupt triggered.

### 3.6 Memory Map

The local memory base address of Cora+LP is determined by Memory Base Address Register 0 (PCI Byte Address=0x10) in PCI Configuration Registers.

The following shows the local memory map of Coral LP to the host CPU memory space.

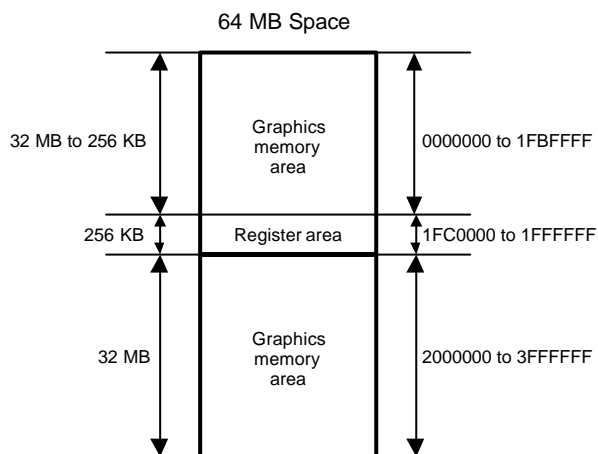


Fig. 3.1 Memory Map

Table 3-4 Address Space

| Size            | Resource   | Base address           | (Name)                               |
|-----------------|--|------------------------|--------------------------------------|
| 32 MB to 256 KB | Graphics Memory  | 00000000               |                                      |
| 64 KB           | Host interface registers<br>(I <sup>2</sup> C interface registers) | 01FC0000<br>(01FCC000) | (HostBase)<br>(I <sup>2</sup> CBase) |
| 32 KB           | Display registers  | 01FD0000               | (DisplayBase)                        |
| 32 KB           | Video capture registers  | 01FD8000               | (CaptureBase)                        |
| 64 KB           | Internal texture memory  | 01FE0000               | (TextureBase)                        |
| 32 KB           | Drawing registers  | 01FF0000               | (DrawBase)                           |
| 32 KB           | Geometry engine registers  | 01FF8000               | (GeometryBase)                       |
| 32 MB           | Graphics memory  | 02000000               |                                      |

If required 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. Coral LP may be accessed after about 20 bus clocks after writing 1 to RSW.

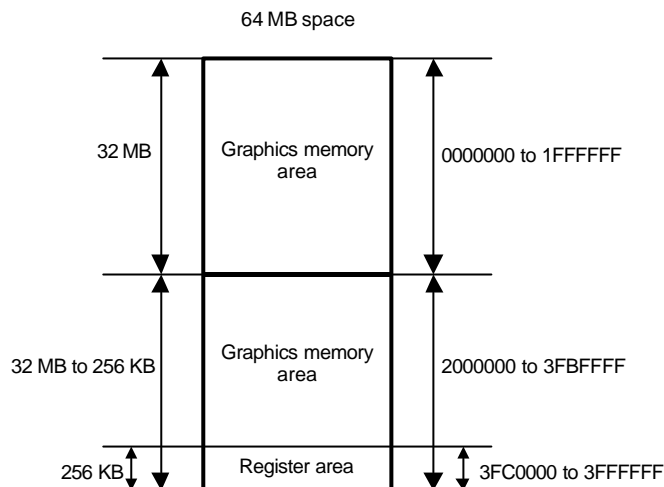


Fig. 3.2 Alternate Memory Map

Table 3-5 Alternate Address Mapping

| Size            | Resource   | Base address           | (Name)                               |
|-----------------|--|------------------------|--------------------------------------|
| 64 MB to 256 KB | Graphics memory  | 00000000               |                                      |
| 64 KB           | Host interface registers<br>(I <sup>2</sup> C interface registers) | 03FC0000<br>(03FCC000) | (HostBase)<br>(I <sup>2</sup> CBase) |
| 32 KB           | Display registers  | 03FD0000               | (DisplayBase)                        |
| 32 KB           | Video capture registers  | 03FD8000               | (CaptureBase)                        |
| 64 KB           | Internal texture memory  | 03FE0000               | (TextureBase)                        |
| 32 KB           | Drawing registers  | 03FF0000               | (DrawBase)                           |
| 32 KB           | Geometry engine registers  | 03FF8000               | (GeometryBase)                       |



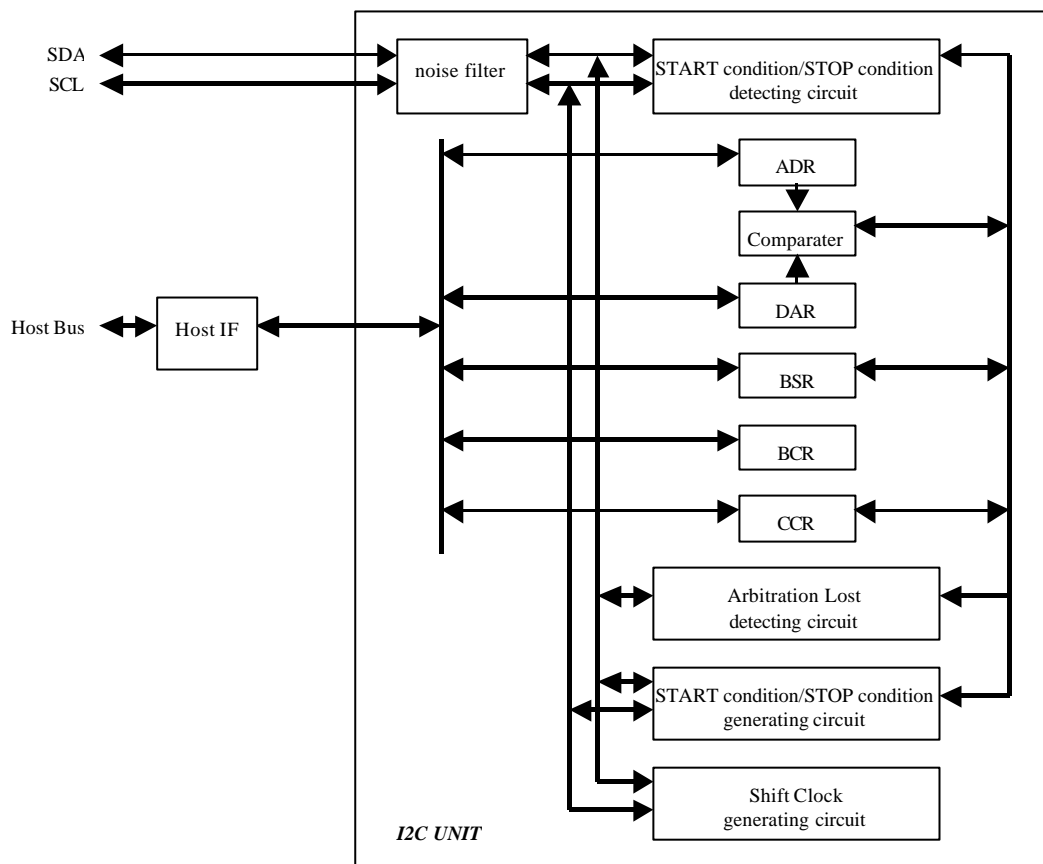
## 4. I<sup>2</sup>C Interface Controller

### 4.1 Features

- Master transmission and receipt
- Slave transmission and receipt
- Arbitration
- Clock synchronization
- Detection of slave address
- Detection of general call address
- Detection of transfer direction
- Repeated generation and detection of START condition
- Detection of bus error
- Correspondence to standard-mode (100kbit/s ) / high-speed-mode (400kbit/s)

## 4.2 Block diagram

### 4.2.1 Block Diagram



## 4.2.2 Block Function Overview

### **START condition / STOP condition detecting circuit**

This circuit performs detection of START condition and STOP condition from the state of SDA and SCL.

### **START condition / STOP condition generating circuit**

This circuit performs generation of START condition and STOP condition by changing the state of SDA and SCL.

### **Arbitration Lost detecting circuit**

This circuit compares the data output to SDA line with the data input into SDA line at the time of data transmission, and it checks whether these data is in agreement. When not in agreement, it generates arbitration lost.

### **Shift Clock generating circuit**

This circuit performs generating timing count of the clock for serial data transfer, and output control of SCL clock by setup of a clock control register.

### **Comparater**

Comparater compares whether the received address and the self-address appointed to be the address register is in agreement, and whether the received address is a global address.

### **ADR**

ADR is the 7-bit register which appoints a slave address.

### **DAR**

DAR is the 8-bit register used by serial data transfer.

### **BSR**

BSR is the 8-bit register for the state of I2C bus etc. This register has following functions:

- detection of repeated START condition
- detection of arbitration lost
- storage of acknowledge bit
- data transfer direction
- detection of addressing
- detection of general call address
- detection of the 1st byte

### **BCR**

BCR is the 8-bit register which performs control and interruption of I2C bus. This register has following functions:

- request / permission of interruption
- generation of START condition
- selection of master / slave
- permission to generate acknowledge

### CCR

CCR is the 7-bit register used by serial data transfer. This register has following functions:

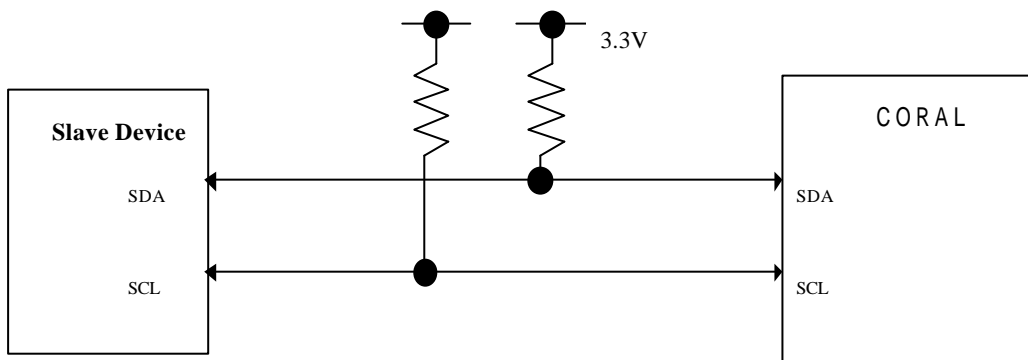
- permission of operation
- setup of a serial clock frequency
- selection of standard-mode / high-speed-mode

### Noise filter

This noise filter consists of a 3 step shift register. When all three value that carried out the continuation sampling of the SCL/SDA input signals is "1", the filter output is "1". Conversely when all three value is "0", the filter output is "0". To other samplings it holds the state before 1 clock.

## 4.3 Example application

### 4.3.1 Connection Diagram



### 4.4 Function overview

Two bi-directional buses, serial data line (SDA) and serial clock line (SCL), carry information at I2C-bus. Scarlet I2C interface has SDA input (SDAI) and SDA output (SDAO) for SDA and is connected to SDA line via open-drain I/O cell. And this interface also has SCL input (SCLI) and SCL output (SCLO) for SCL line and is connected to SCL line via open-drain I/O cell. The wired theory is used when the interface is connected to SDA line and SCL line.

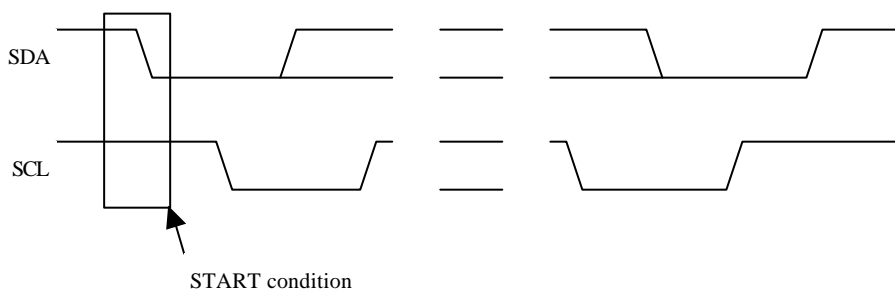
#### 4.4.1 START condition

If "1" is written to MSS bit while the bus is free, this module will become a master mode and will generate START condition simultaneously. In a master mode, even if a bus is in a use state (BB=1), START condition can be generated again by writing "1" to SCC bit.

There are two conditions to generate START condition.

- "1" writing to MSS bit in the state where the bus is not used (MSS=0 & BB=0 & INT=0 & AL=0)
- "1" writing to SCC bit in the interruption state in a master mode (MSS=1 & BB=1 & INT=1 & AL=0)

If "1" writing is performed to MSS bit in an idol state, AL bit will be set to "1". "1" writing to MSS bit other than the above is disregarded.

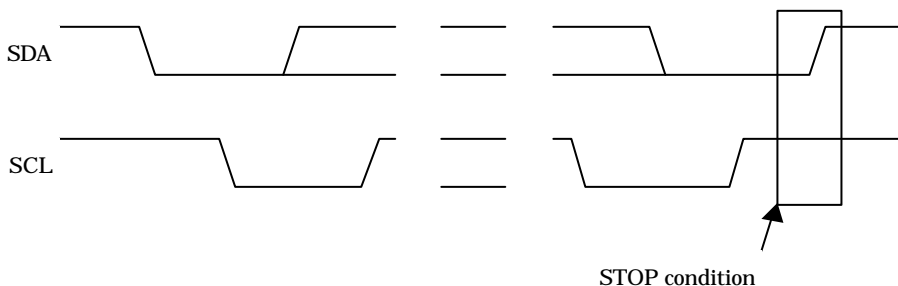


#### 4.4.2 STOP condition

If "0" is written to MSS bit in a master mode (MSS=1), this module will generate STOP condition and will become a slave mode.

There is a condition to generate STOP condition.

- "0" writing to MSS bit in the interruption state in a master mode (MSS=1 & BB=1 & INT=1 & AL=0)
- "0" writing to MSS bit other than the above is disregarded.

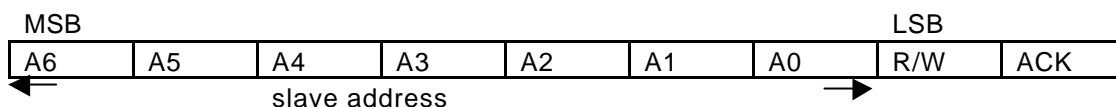


### 4.4.3 Addressing

In a master mode, it is set to BB="1" and TRX="0" after generation of START condition, and the contents of DAR register are output from MSB. When this module receives acknowledge after transmission of address data, the bit-0 of transmitting data (bit-0 of DRA register after transmission) is reversed and it is stored in TRX bit.

#### - Transfer format of slave address

A transfer format of slave address is shown below:



#### - Map of slave address

A map of slave address is shown below:

| slave address   | R/W | Description               |
|---|-----|---------------------------|
| 0 0 0 0 0 0 0   | 0   | General call address      |
| 0 0 0 0 0 0 0   | 1   | START byte                |
| 0 0 0 0 0 0 1   | X   | CBUS address              |
| 0 0 0 0 0 1 0   | X   | Reserved                  |
| 0 0 0 0 0 1 1   | X   | Reserved                  |
| 0 0 0 0 1 X X   | X   | Reserved                  |
| 0 0 0 1 X X X<br> <br> <br> <br> <br> <br>1 1 1 0 X X X | X   | Available slave address   |
| 1 1 1 1 0 X X   | X   | 10-bit slave addressing*1 |
| 1 1 1 1 1 X X   | X   | Reserved                  |

\*1 This module does not support 10-bit slave address.

### 4.4.4 Synchronization of SCL

When two or more I2C devices turn into a master device almost simultaneously and drive SCL line, each devices senses the state of SCL line and adjusts the drive timing of SCL line automatically in accordance with the timing of the latest device.

#### 4.4.5 Arbitration

When other masters have transmitted data simultaneously at the time of master transmission, arbitration takes place. When its own transmitting data is "1" and the data on SDA line is "0", the master considers that the arbitration was lost and sets "1" to AL. And if the master is going to generate START condition while the bus is in use by other master, it will consider that arbitration was lost and will set "1" to AL.

When the START condition which other masters generated is detected by the time the master actually generated START condition, even when it checked the bus is in nonuse state and wrote in MSS="1", it considers that the arbitration was lost and sets "1" to AL.

When AL bit is set to "1", a master will set MSS="0" and TRX="0" and it will be a slave receiving mode. When the arbitration is lost (it has no royalty of a bus), a master stops a drive of SDA. However, a drive of SCL is not stopped until 1 byte transfer is completed and interruption is cleared.

#### 4.4.6 Acknowledge

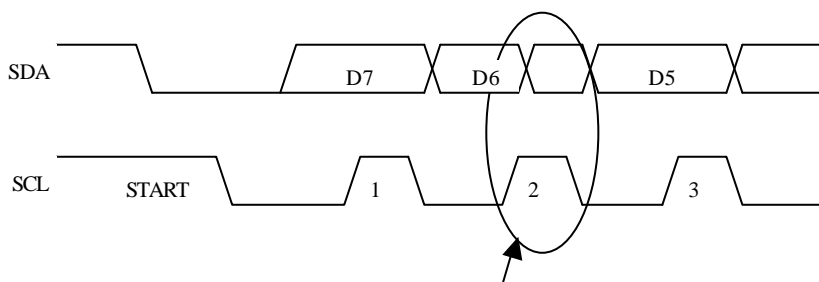
Acknowledge is transmitted from a reception side to a transmission side. At the time of data reception, acknowledge is stored in LRB bit by ACK bit.

When the acknowledge from a master reception side is not received at the time of slave transmission, it sets TRX="0" and becomes slave receiving mode. Thereby, a master can generate STOP condition when a slave opens SCL.

#### 4.4.7 Bus error

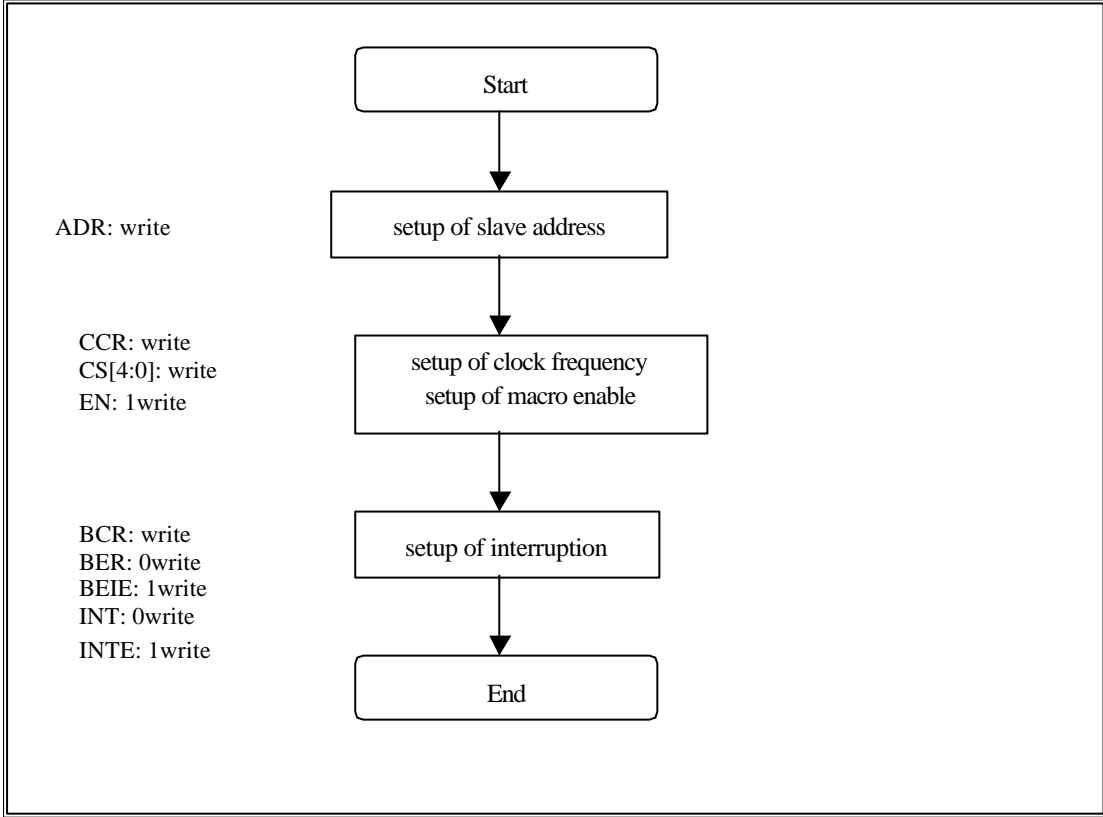
When the following conditions are satisfied, it is judged as a bus error, and this interface will be in a stop state.

- Detection of the basic regulation violation on I2C-bus under data transfer (including ACK bit)
- Detection of STOP condition in a master mode
- Detection of the basic regulation violation on I2C-bus at the time of bus idol



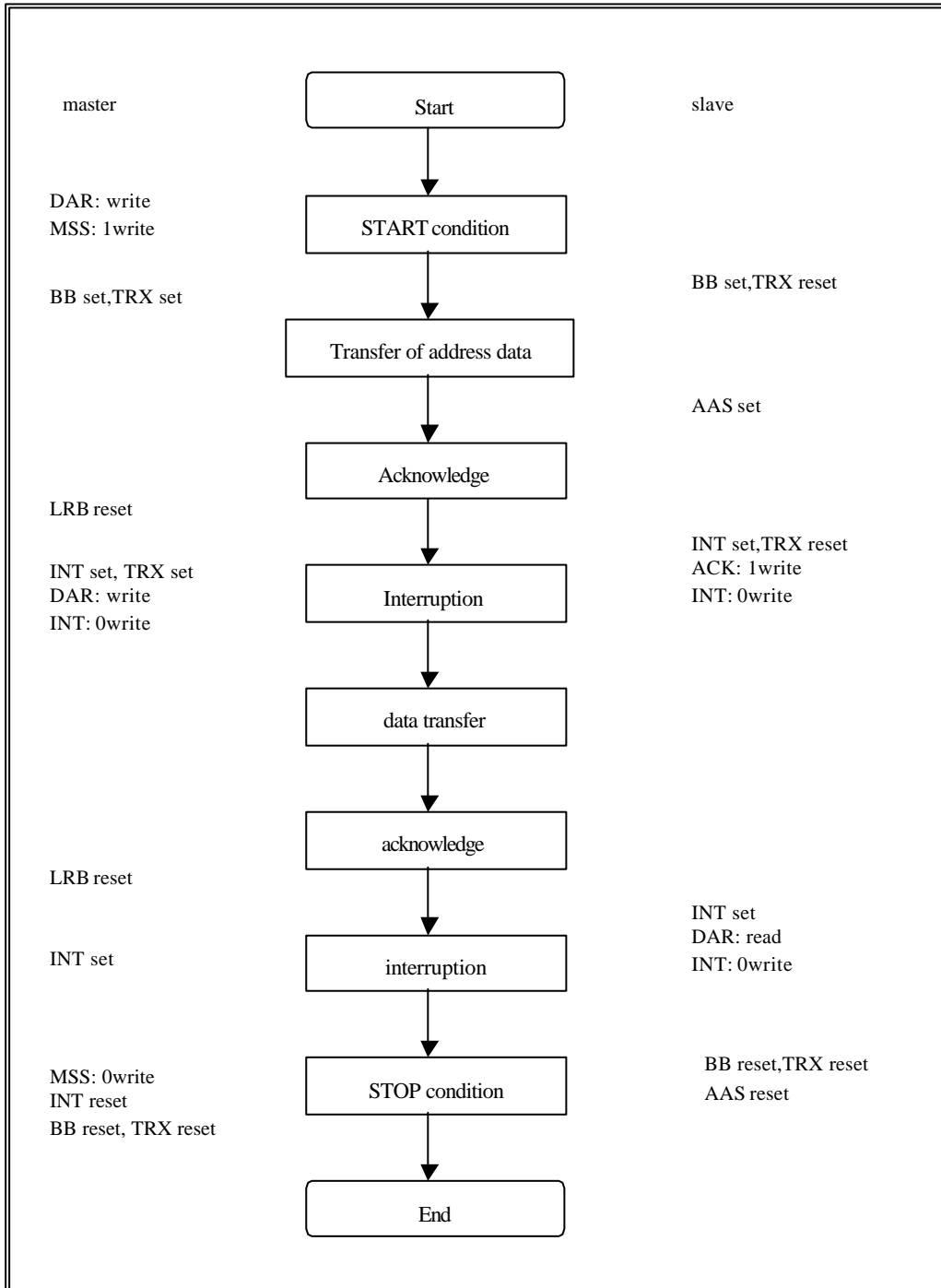
SDA changed under data transmission (SCL=H). It becomes bus error.

4.4.8 Initialize

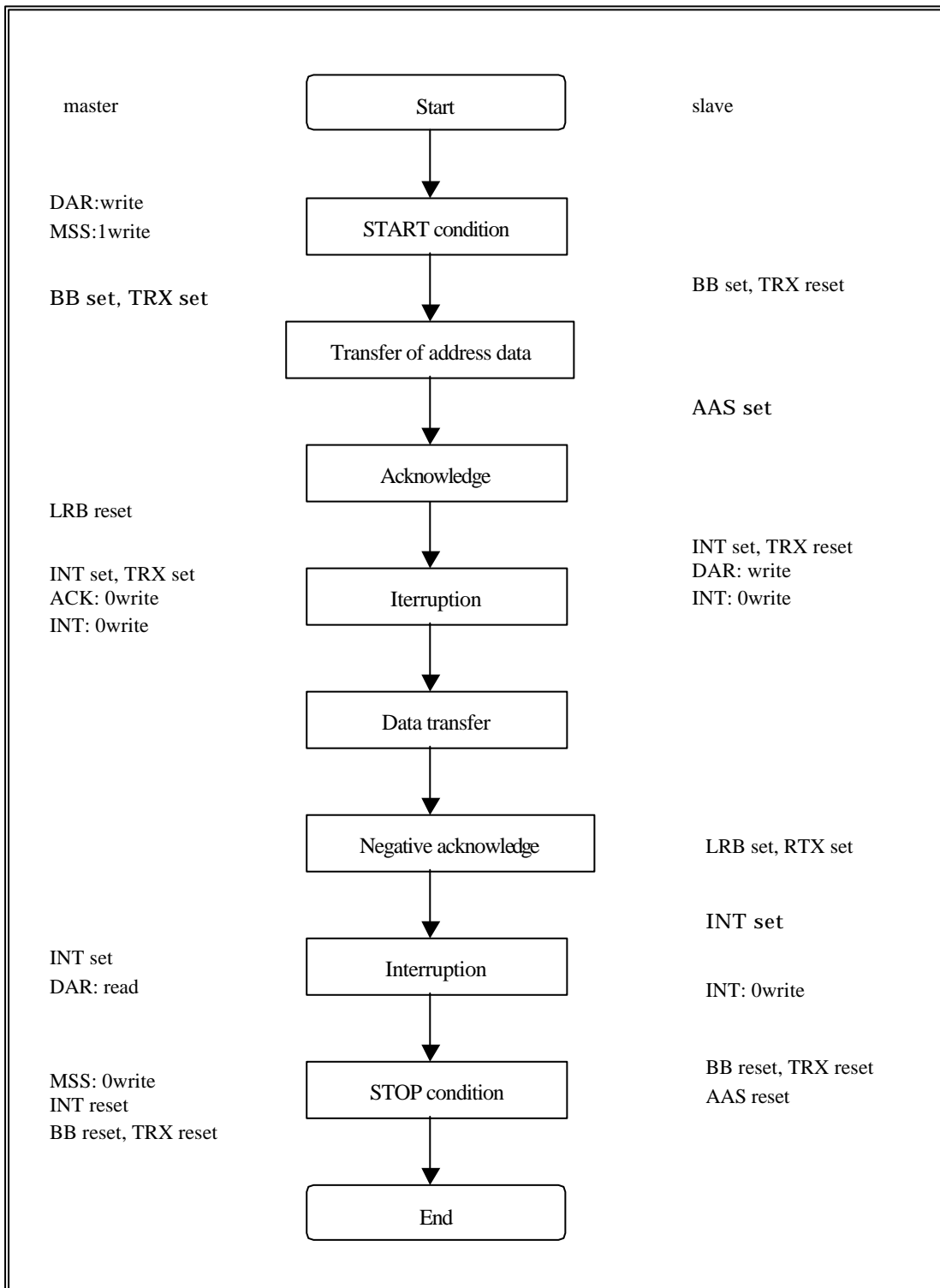




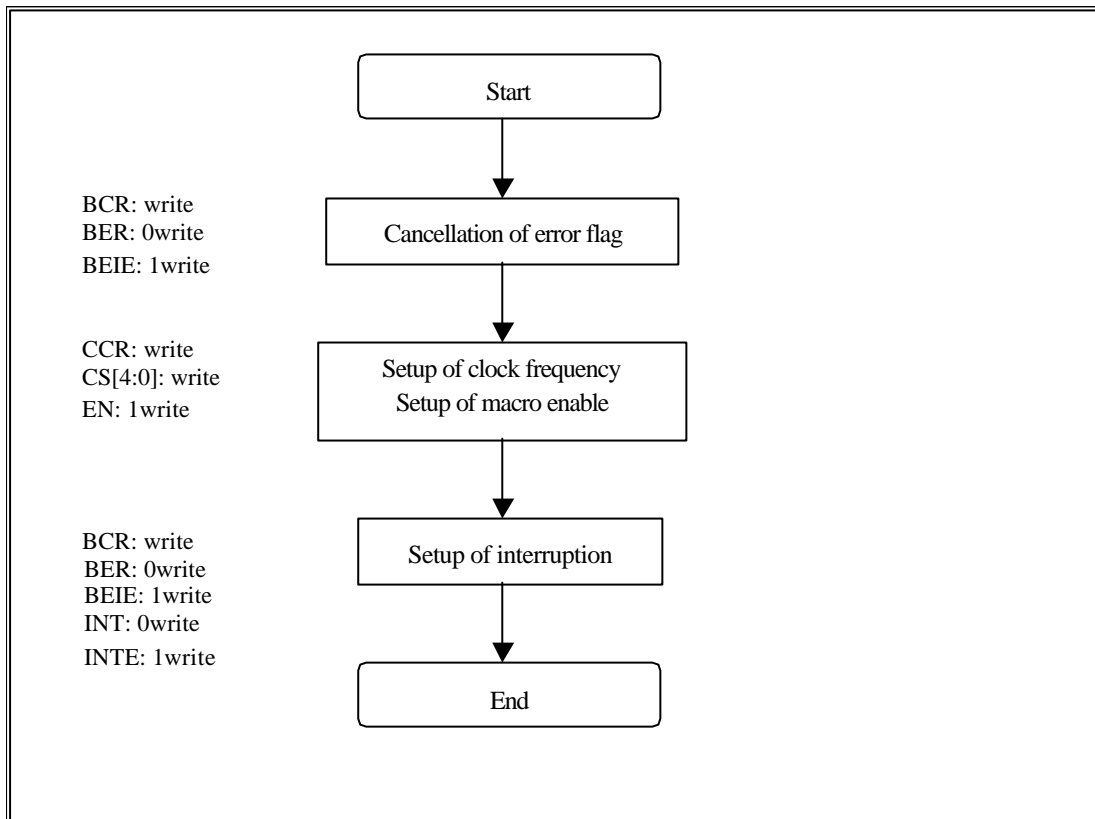
4.4.9 1-byte transfer from master to slave



4.4.10 1-byte transfer from slave to master



#### 4.4.11 Recovery from bus error



## 4.5 Note

### A ) About a 10-bit slave address

This module does not support the 10-bit slave address. Therefore, please do not specify the slave address of from 78H to 7bH to this module. If it is specified by mistake, a normal transfer cannot be performed although acknowledge bit is returned at the time of 1 byte reception.

### B ) About competition of SCC, MSS, and INT bit

Competition of the following byte transfer, generation of START condition, and generation of STOP condition happens by the simultaneous writing of SCC, MSS, and INT bit. At this time the priority is as follows.

- 1) The following byte transfer and generation of STOP condition  
If "0" is written to INT bit and "0" is written to MSS bit, priority will be given to "0" writing to MSS bit and STOP condition will be generated.
- 2) The following byte transfer and generation of START condition  
If "0" is written to INT bit and "1" is written to SCC bit, priority will be given to "1" writing to SCC bit and START condition will be generated.
- 3) Generation of START condition and generation of STOP condition  
The simultaneous writing of "1" in SCC bit and "0" to MSS bit is prohibition.

### C ) About setup of S serial transfer clock

When the delay of the positive edge of SCL terminal is large or when the clock is extended by the slave device, it may become smaller than setting value (calculation value) because of generation of overhead.

## 5. DISPLAY CONTROLLER

### 5.1 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).

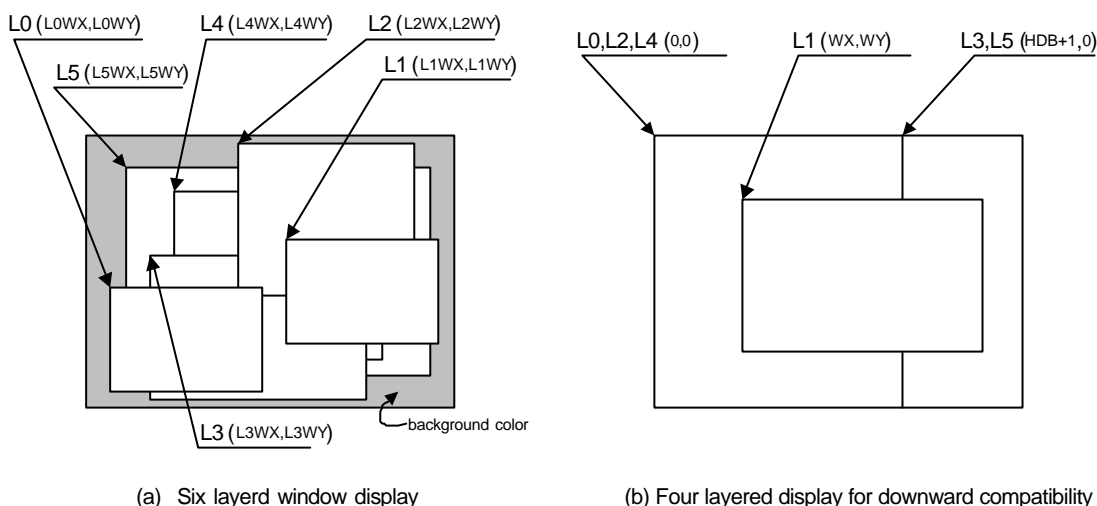
#### Cursor

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

## 5.2 Display Function

### 5.2.1 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.



### 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  | (L0WX, L0WY)                  | (0, 0)             | (L0WW, L0WH + 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) |

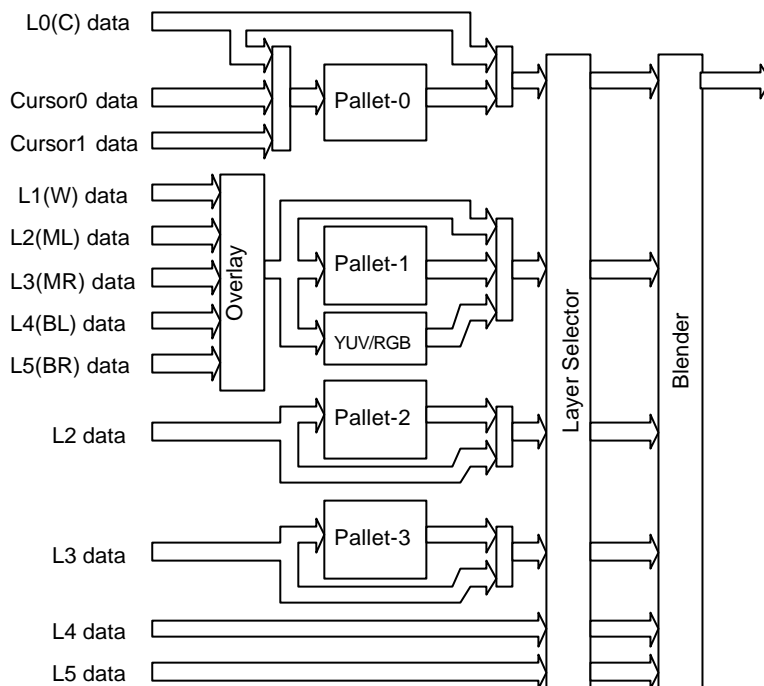
C, W, ML, MR, BL, and BR above mean layers for previous products. The window mode or the compatibility mode can be selected for each layer. It is possible to use new functions through minor program changes by allowing the coexistence of display modes instead of separating them completely.

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.

## 5.2.2 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 to 8 bits/pixel.



### 5.2.3 Display parameters

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

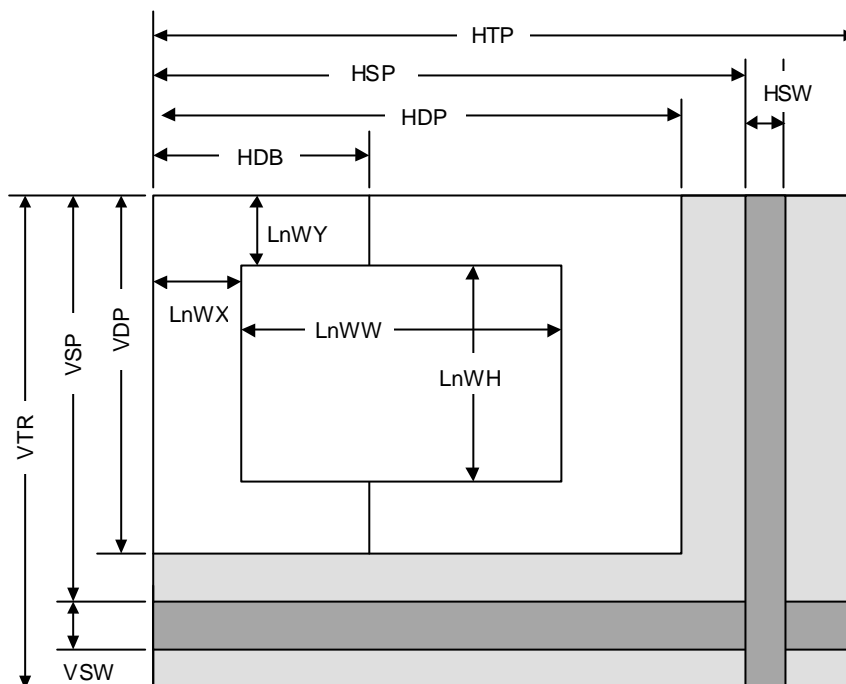


Fig. 5.1 Display Parameters

|      |                                       |
|------|---------------------------------------|
| 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:

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

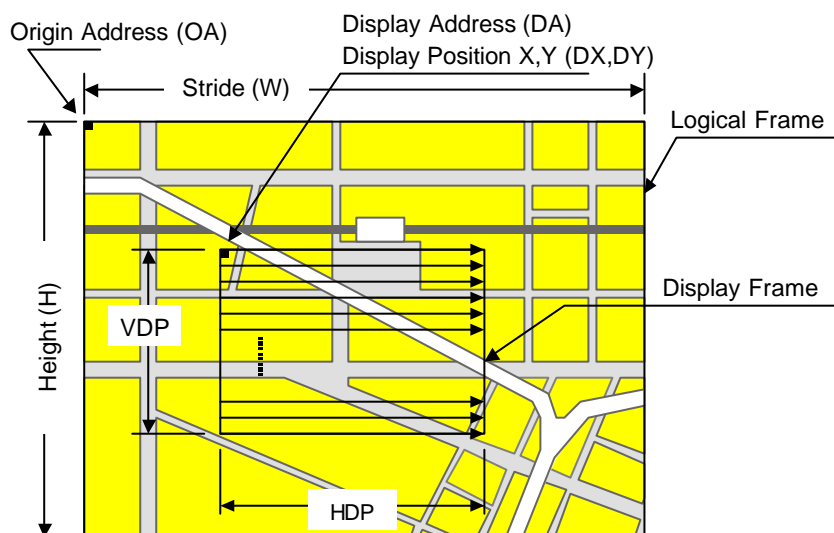
$$0 < VDP < VSP < VSP + VSW + 1 < VTR$$

### 5.2.4 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

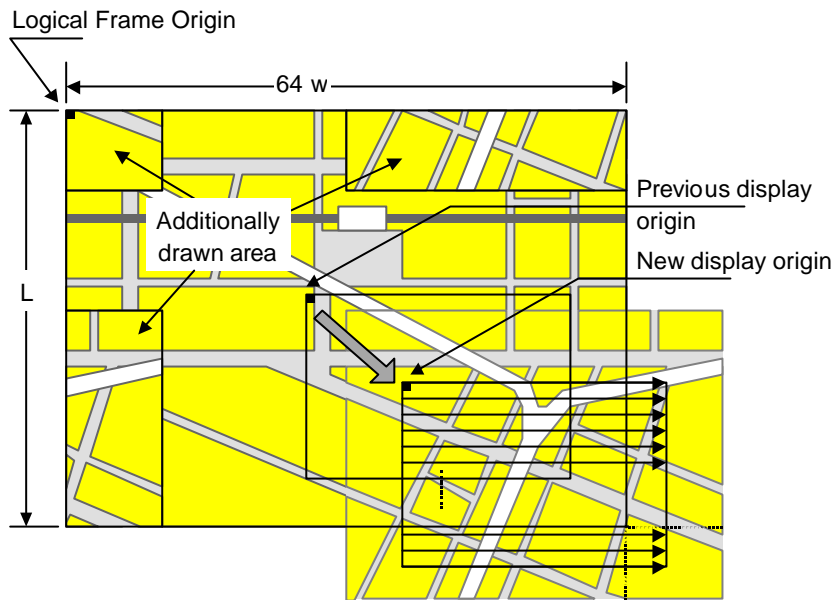
The relation between the logical graphics space and display position is defined as follows:



**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<br>DY | Display Position | Display origin coordinates.<br>Coordinates in logical frame space of display frame origin               |

MB8629x 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.

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

### 5.3 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.

## **5.4 Cursor**

### **5.4.1 Cursor display function**

CORAL can display two hardware cursors. Each cursor is specified as  $64 \times 64$  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.

### **5.4.2 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.

## 5.5 Display Scan Control

### 5.5.1 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

### 5.5.2 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.

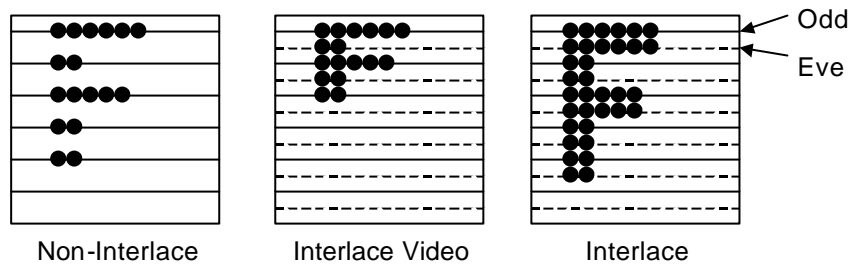
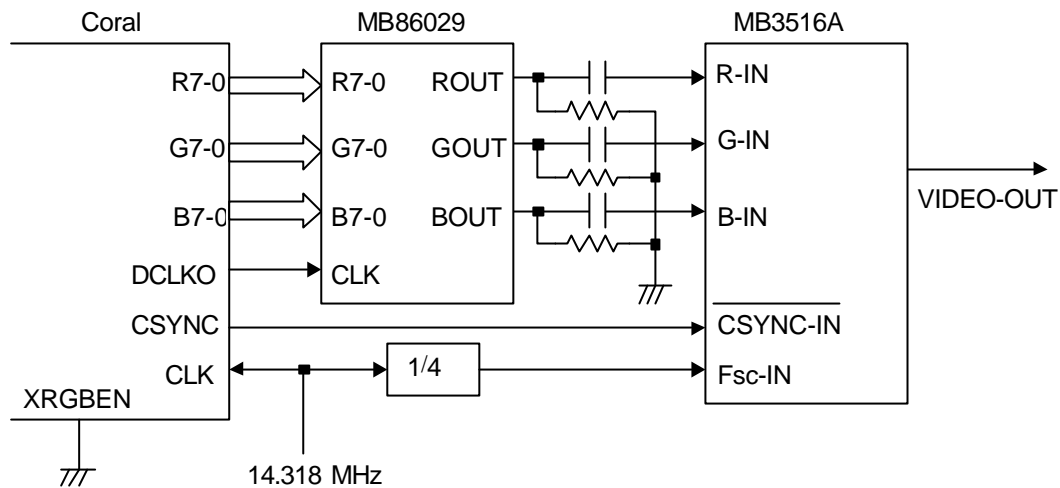


Fig. 5.4 Display Difference between Synchronization Modes

## 5.6 Video Interface, NTSC/PAL Output

To achieve NTSC/PAL signals, a NTSC/PAL encoder must be connected externally as shown below:



**Fig. 5.6 Example of NTSC/PAL Encoder Connection**

The digital NTSC/PAL 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/PAL encoder.

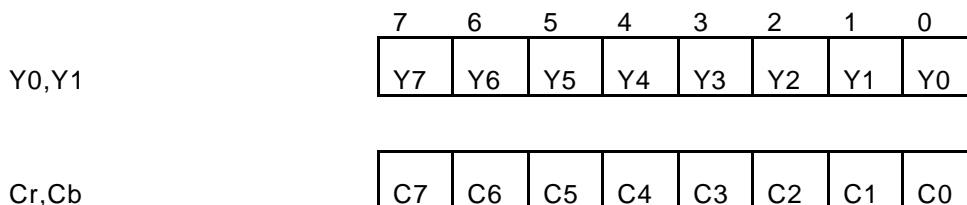
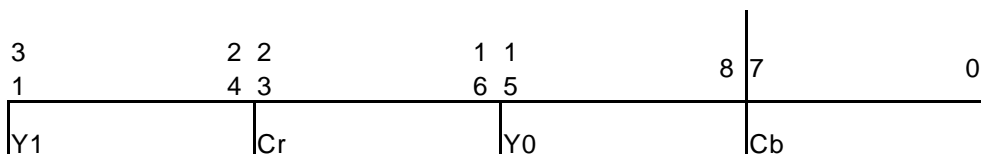


## 6. Video Capture

### 6.1 Input Formats

The video capture unit of MB86295 'Coral-P' accepts YUV422 video data primarily, but RGB video data is also accepted via an internal RGB preprocessor which converts RGB to YUV422.

Captured pixels are stored in YCbCr format in graphics memory, 16 bits per pixel. The video data is converted to RGB when it is displayed.



### 6.2 ITU RBT-656 input

#### 6.2.1 YUV input format

The ITU RBT-656 format is widely used for digital transmission of NTSC and PAL signals. The format corresponds to YUV422. Interlaced video display signals can be captured and displayed non-interlaced with linear interpolation.

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

#### 6.2.2 Synchronous Control

Writing video data in memory and scanning for display are executed simultaneously. The memory of the video capture unit is controlled by a ring buffer controller. If the frame rate of video capture is

different from the display frame rate, frames are skipped or the same frame is continuously displayed automatically to match the two frame rates.

When the expected control code in input video stream is not detected, an error is generated. The error status is returned in a register. When control code is not detected, pictures are taken in continuously by predicting the timing by code input previously.

### **6.2.3 Non-interlace Transformation**

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

#### **- BOB Mode**

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

#### **- WEAVE Mode**

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

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

### **6.2.4 Area Allocation**

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

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

$$\text{CBLA} = \text{CBOA} + 64n \times \text{CBS}$$

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

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

## 6.3 RGB input

### 6.3.1. RGB input modes

RGB video data is accepted via an internal RGB preprocessor which converts RGB to YUV422. There are two RGB modes : direct input mode and multiplex input mode. One pixel is transferred in ONE clock in direct input mode while one pixel is transferred in TWO clocks in multiplex input mode.

The **direct mode** is suitable for relatively high speed non-interlaced video signals but the de-interlacing operation is not available in this mode. The maximum input rate is 40Mpixel/sec. RGB component data is 6bit.

The **multiplex mode** is suitable for interlaced or relatively low speed video signal and de-interlacing operation is available. RGB component data is 8bit.

The mode will be controlled by the RGB bit of VCM(video capture mode) register.

### 6.3.2. RGB Input Signals

The signals used for RGB video capture are not assigned dedicated terminals but share same pins with other functions. There are two set of signals corresponding to two modes.

Direct Input Mode :

| Name   | IO | Function                        |
|--------|----|---------------------------------|
| RGBCLK | In | Clock for RGB input             |
| RI5-0  | In | Red component value             |
| GI5-0  | In | Green component value           |
| BI5-0  | In | Blue component value            |
| VSYNCI | In | Vertical sync for RGB capture   |
| HSYNCI | In | Horizontal sync for RGB capture |

Multiplex Input Mode :

| Name   | IO | Function                        |
|--------|----|---------------------------------|
| RGBCLK | In | Clock for RGB input             |
| RBI7-0 | In | Red and blue component value    |
| GI7-0  | In | Green component value           |
| COLSEL | In | Select Red and Blue             |
| VSYNCI | In | Vertical sync for RGB capture   |
| HSYNCI | In | Horizontal sync for RGB capture |

Note :

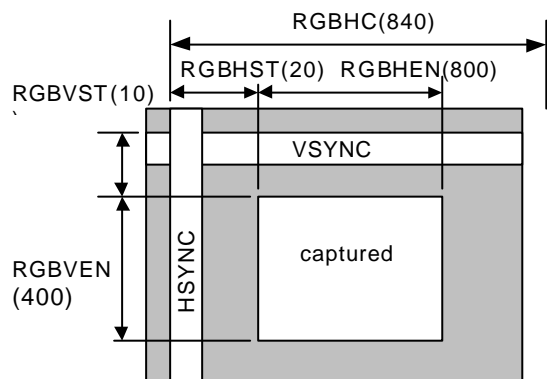
- input pins are shared with the ITU656 input and memory data bus.
- the MPX bit of the VCM(video capture mode) register selects which mode is used.

### 6.3.3. Captured Range

Instead of embedded sync code method used in ITU656 mode, the capture range in RGB mode is specified by the following register parameters :

- 1) RGB input mode of capture : Set RGB666 input flag in VCM.
- 2) HSYNC Cycle : Set the number of HSYNC Cycles in RGBHC.
- 3) Horizontal Enable area : Set enable area start position and enable picture size into RGBHST and RGBHEN.
- 4) Vertical Enable area : Set enable area start position and enable picture size into RGBVST and RGBVEN.

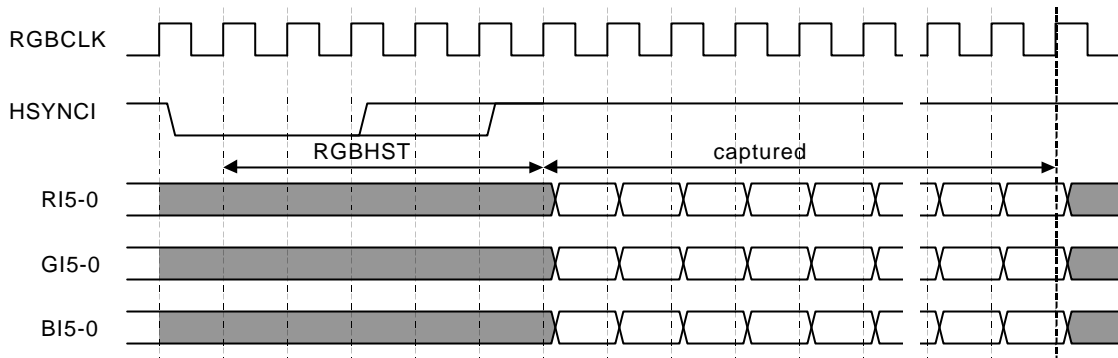
For example, if input picture size is 800x400, then parameters for each register are decided as follow :



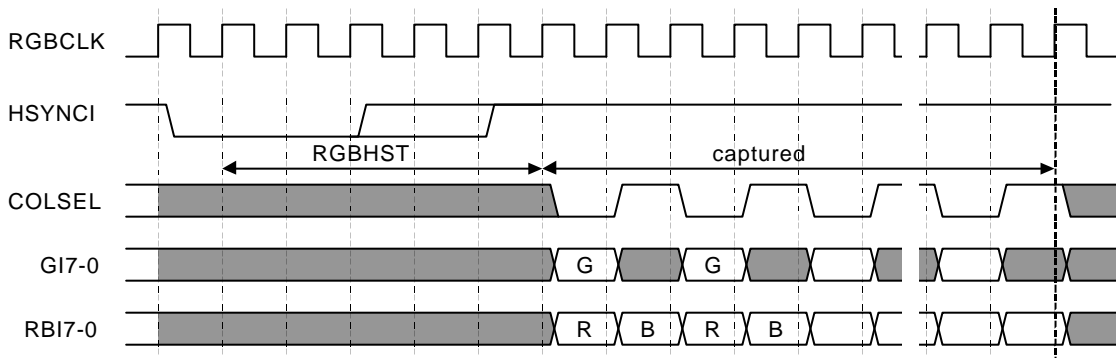
#### 5) Convert Matrix Coefficient

In order to change the color conversion matrix, set up RGBCMY, RGBCb, RGBCr and RGBCMb.

### 6.3.4. Direct Input Mode Operation

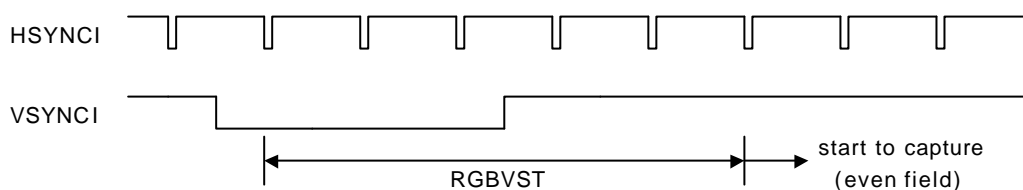
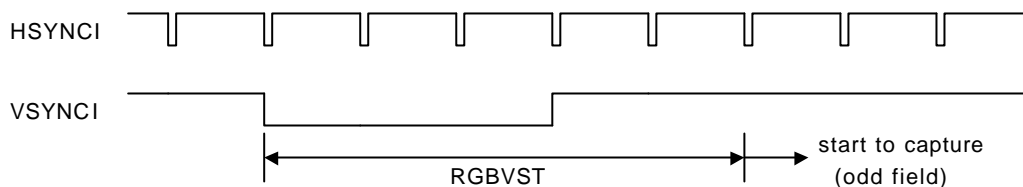


### 6.3.5 Multiplex Input Mode Operation



### 6.3.6. Even/Odd field Recognition

In multiplex input mode, interlaced RGB video data can be accepted and de-interlaced. A field is recognized as even or odd by the relative pulse position of H-sync and V-sync.



### 6.3.7. Conversion Operation

RGB input data is converted to YcrCb by the following matrix operation :

$$\begin{aligned}
 Y &= a_{11} * R + a_{12} * G + a_{13} * B + b_1 \\
 Cr &= a_{21} * R + a_{22} * G + a_{23} * B + b_2 & a_{ij} : & \text{10bit signed real ( lower 8bit is fraction )} \\
 Cb &= a_{31} * R + a_{32} * G + a_{33} * B + b_3 & b_i : & \text{8bit unsigned integer}
 \end{aligned}$$

- Note :
- Each coefficient can be defined by registers.
  - Cb and Cr components are reduced to half after this operation to form in 4:2:2 format.

## 6.4 Scaling

### 6.4.1 Downscaling Function

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

|                                   |                            |                     |
|-----------------------------------|----------------------------|---------------------|
| Reduction in vertical direction   | 576 → 490 lines            | $576/490 = 1.176$   |
|                                   | $1.176 \times 2048 = 2408$ | → 0968 <sub>H</sub> |
| Reduction in horizontal direction | 720 → 648 pixels           | $720/648 = 1.111$   |
|                                   | $1.111 \times 2048 = 2275$ | → 08E3 <sub>H</sub> |

Therefore, 096808E3<sub>H</sub> is set in CSC.

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

### 6.4.2 Upscaling Function

Coral is able to enlarge the size of a video capture picture by the factor of 2 in both the horizontal and vertical directions. This feature can be used to realize full-screen modes of video input streams which have a resolution less than actual display size. In order to use magnify (up-scaling) mode, the horizontal and vertical factor must be less than one. Do not specify different scaling ways (reduction/enlargement) for horizontal and vertical factors ! Also initialize the following registers as follows :

Set the magnify flag in the L1-layer mode register of the display controller.

Set the picture source size (before magnification) into CMSHP and CMSVL.

Set the final picture size (after magnification) into CMDHP and CMDVL.

An example of the expressions for setting an enlargement in the vertical and horizontal directions is shown below :

If the input picture size is 480x360 and the display picture size is 640x480, then the parameters for each register are as follows.

HSCALE=(480/640)\*2048=0x0600

VSCALE=(360/480)\*2048=0x0600

CMSHP=0x00f0

CMSVL=0x0168

CMDHP=0x0140

CMDVL=0x01e0

L1WW=0x0280

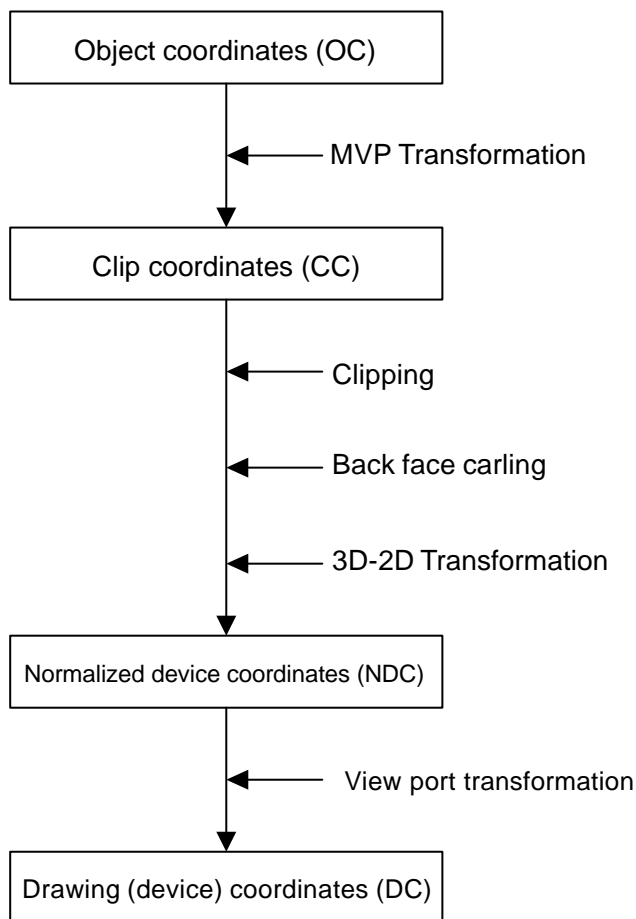
L1WH=0x01df

## 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

$$X_{dc} = X_{Scaling} * X_{ndc} + X_{Offset}$$

$$Y_{dc} = Y_{Scaling} * Y_{ndc} + Y_{Offset}$$

$$Z_{dc} = Z_{Scaling} * Z_{ndc} + Z_{Offset}$$

### 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.

$$X_{min} * W_{cc} \leq X_{cc} \leq X_{max} * W_{cc}$$

$$Y_{min} * W_{cc} \leq Y_{cc} \leq Y_{max} * W_{cc}$$

$$Z_{min} * W_{cc} \leq Z_{cc} \leq Z_{max} * W_{cc}$$

$$W_{min} \leq W_{cc}$$

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 = 0x7f7fffff, minimum value = 0xff7fffff

\*2: Xmin, Ymin, Zmin, Wmin

\*3: Xmax, Ymax, Zmax

An example of the G\_ViewVolumeZclip packet is shown below.

```
0xf1012010 //Setting of GMDR0
0x00000000 //Data format: Floating point data format
0x45000000 //G_ViewVolumeZclip packet
0xff7fffff //Zmin.float setting value (minimum value of IEEE single-precision floating point)
0x7f7fffff //Zmax.float setting value (maximum value of IEEE single-precision floating point)
```

**Example of G\_ViewVolumeZclip Packet when Z Clipping Off**

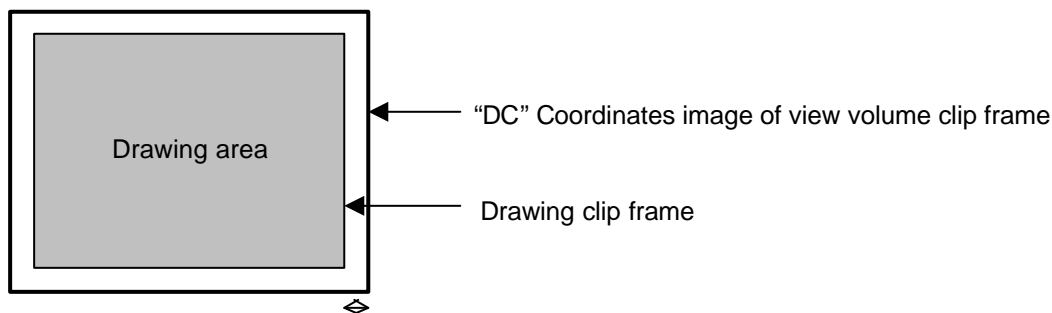
**“W” clipping at orthogonal projection transformation**

“W” at orthogonal projection transformation (GMDR0[0] = 0) is treated as “Wcc=1.0”. For this reason, to suppress “W” clipping, the set “Wmin” value must be 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 culling

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

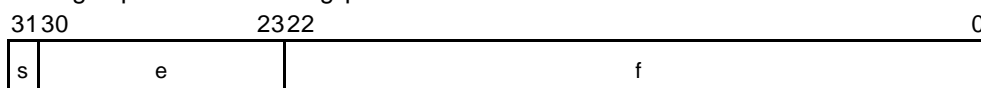
## 7.2 Data Format

### 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



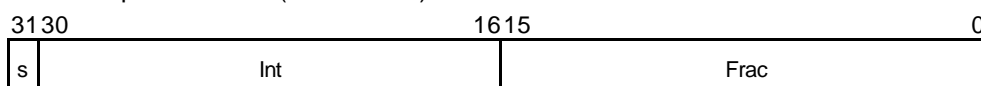
s: Sign bit (1 bit)

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)

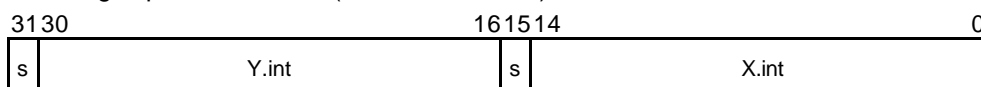


s: Sign bit (1 bit)

int: Integer (15 bits)

frac: Fraction (16 bits)

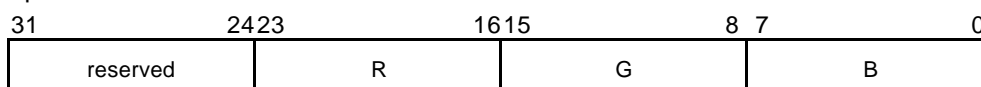
- (3) Signed integer packed format (SINT16.SINT16)



s: Sign bit (1 bit)

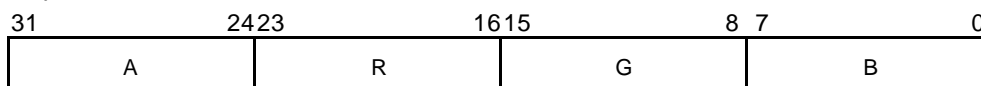
int: Integer (15 bits)

- (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 `nclip_Points.int` 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 using the device coordinates log pointer.

Log pointer is auto-increment-pointer, increment with log output.

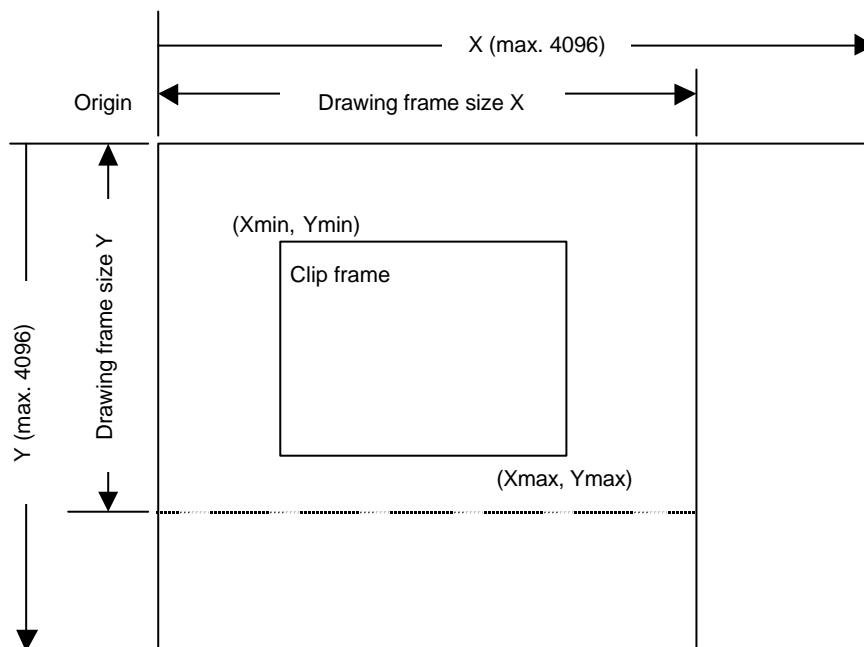
## 8. DRAWING PROCESSING

### 8.1 Coordinate System

#### 8.1.1 Drawing coordinates

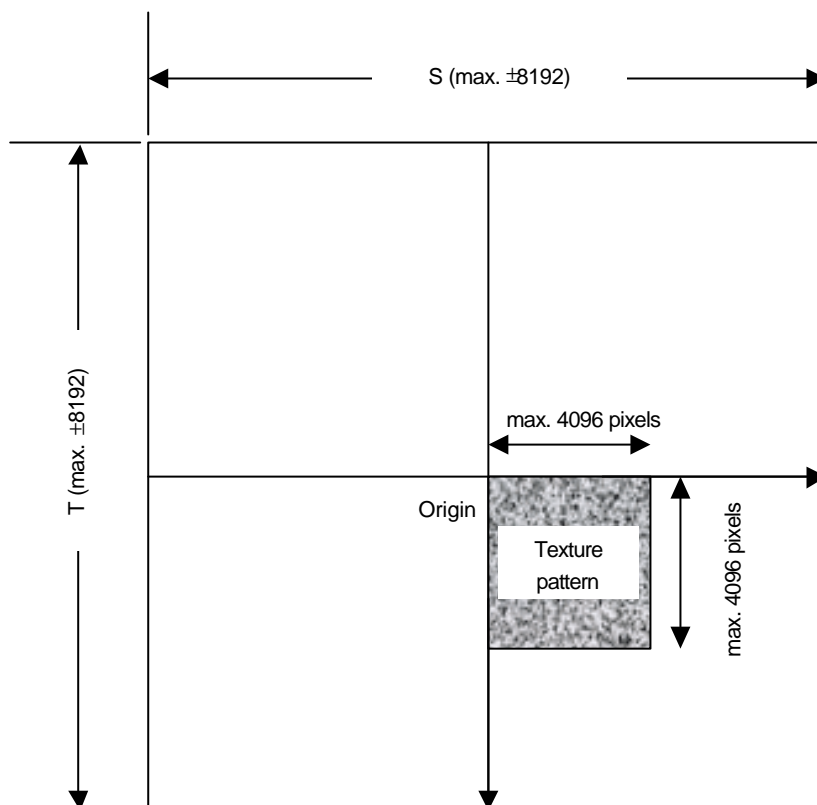
After the calculation of coordinates by the geometry engine, CORAL draws data in the drawing frame in the graphics memory that finally uses the drawing coordinates (device coordinates).

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 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, clamp 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. 1 bit 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**

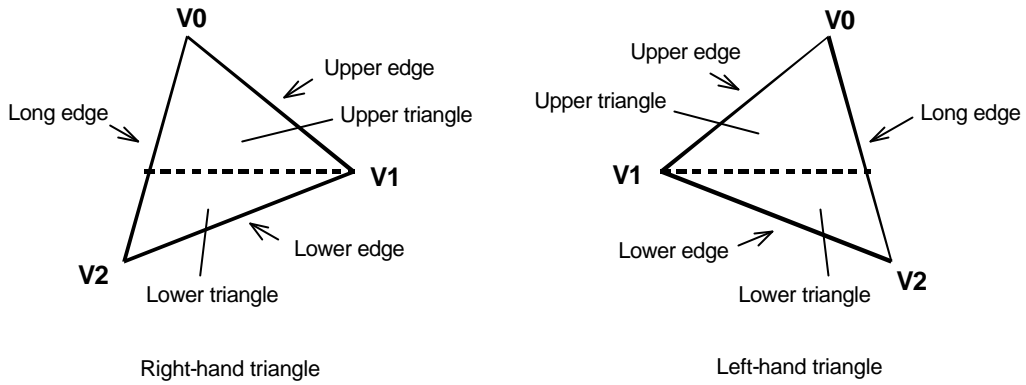
An irregular polygon (including concave shape) is drawn by hardware in the following manner:

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.

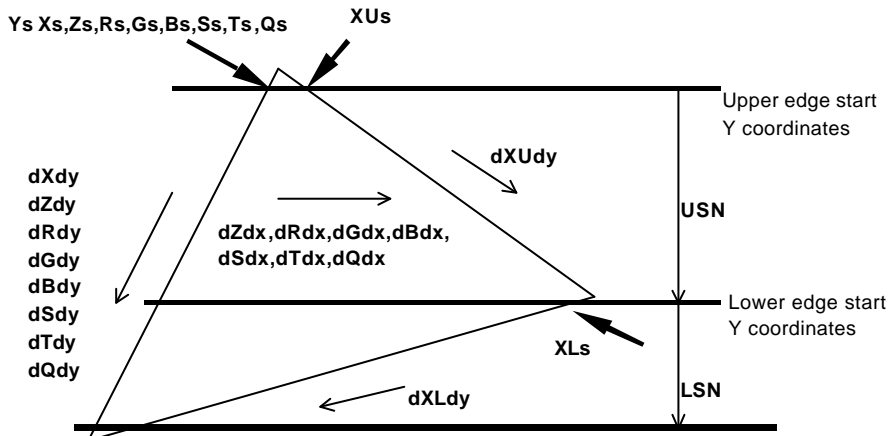
### 8.2.3 Drawing parameters

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.

## 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
3. Transfer from host CPU to internal texture memory
4. Transfer from Graphics Memory to internal texture memory

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, 24 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 16 to 4096 pixels represented as an exponent of 2.

### **8.4.2 Texture memory**

Texture pattern data is stored in either CORAL internal texture RAM or externally connected Graphics Memory. The CORAL texture RAM can store up to  $64 \times 64$  pixels of texture (at 16-bit color). If the texture pattern size is smaller than  $64 \times 64$  pixels, it is best to store it in the internal texture buffer because the texture mapping speed is faster.

Note the following point when using the texture:

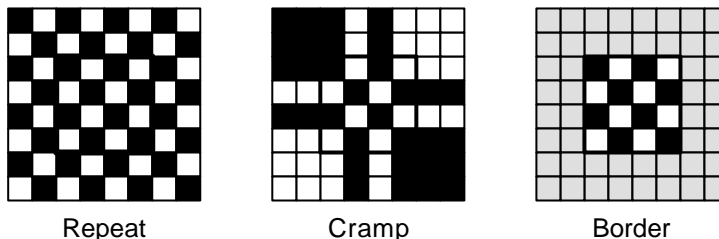
- When access (e.g., CPU read/write) is made to the internal texture RAM other than the display list during drawing, the drawing results are not assured.

### **8.4.3 Texture color**

Drawing of 8-/16-/24-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.

### 8.4.4 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

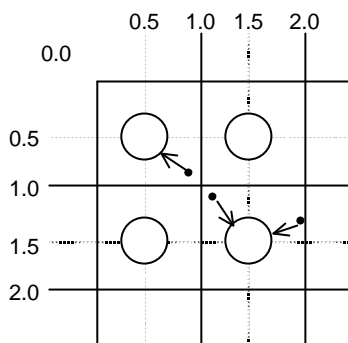
When the applied (S, T) coordinate is either negative or larger than the specified texture pattern size, the outside of the specified texture pattern is rendered in the 'border' color.

### 8.4.5 Filtering

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

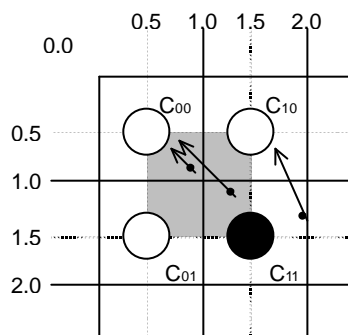
#### Point filtering

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.



### 8.4.6 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 is smaller than -2048 or larger than 2048, normal drawing results are less likely to be obtained.

### 8.4.7 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

### 8.4.8 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 wrapping mode is limited to REPEAT and 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)**

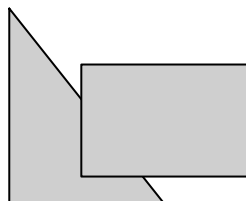
|   |             |             |             |             |
|---|-------------|-------------|-------------|-------------|
|   | 0           | 1           | 6           | 7           |
| 0 | 00 01 08 09 | 01 02 09 10 | 06 07 14 15 | 07 00 15 08 |
| 1 | 08 09 16 17 | 09 10 17 18 | 14 15 12 13 | 15 08 23 16 |
| 2 | 16 17 24 25 | 17 18 25 26 | 22 23 30 31 | 23 16 31 24 |
| 3 | 24 25 32 33 | 25 26 33 34 | 30 31 38 39 | 31 24 39 32 |
| 4 | 32 33 40 41 | 33 34 41 42 | 38 39 46 47 | 39 32 47 40 |
| 5 | 40 41 48 49 | 41 42 49 50 | 46 47 54 55 | 47 40 55 48 |
| 6 | 48 49 56 57 | 49 50 57 58 | 54 55 62 63 | 55 48 63 56 |
| 7 | 56 57 00 01 | 57 58 01 02 | 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. Since the tiling pattern is stored in the texture memory, this function and texture mapping cannot be used at the same time. Also, 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, 24 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 8bit 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

### 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 Z-compare mode setting. Define the Z-buffer 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 2D Triangle). Texture mapping and tiling have separated texture attributes:

**Triangle Drawing Attributes**

| Drawing Attribute          | Description                                   |
|----------------------------|---|
| Shading                    | Gouraud shading or flat shading selectable    |
| Alpha blending             | Set alpha blending enable/disable per polygon |
| Alpha blending coefficient | Set color blending ratio of alpha blending    |

### 8.6.3 Texture attributes

In texture mapping, the following attributes apply:

**Texture Attributes**

| Drawing Attribute              | Description   |
|--------------------------------|---|
| Texture mode                   | Select either texture mapping or tiling   |
| Texture memory mode            | Select either internal texture buffer or external Graphics Memory to use in texture mapping |
| Texture filter                 | Select either point sampling or bi-linear filtering   |
| Texture 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 |

#### 8.6.5 Character pattern drawing attributes

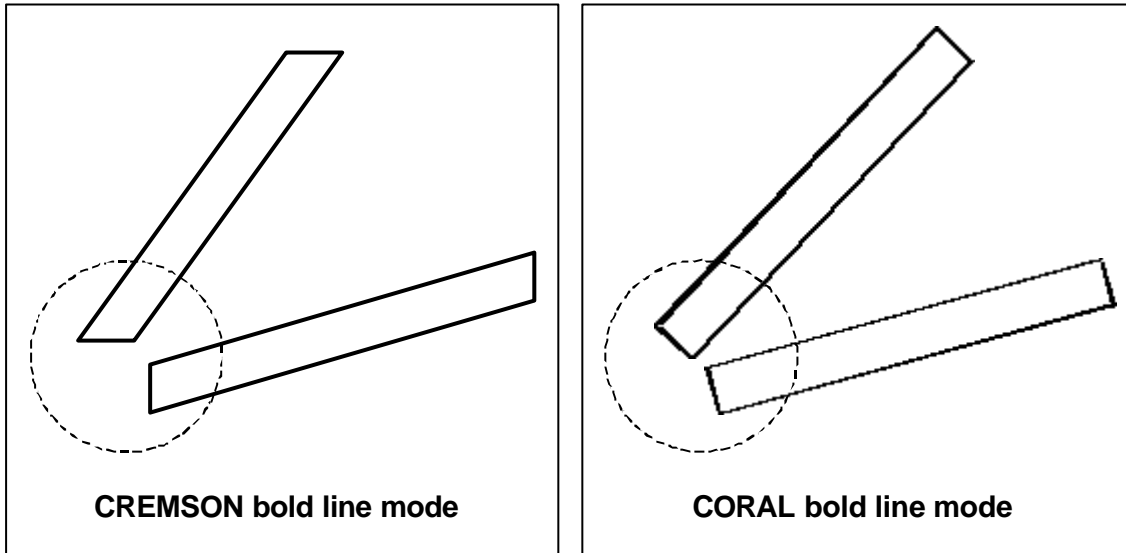
##### Character Pattern Drawing

| 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

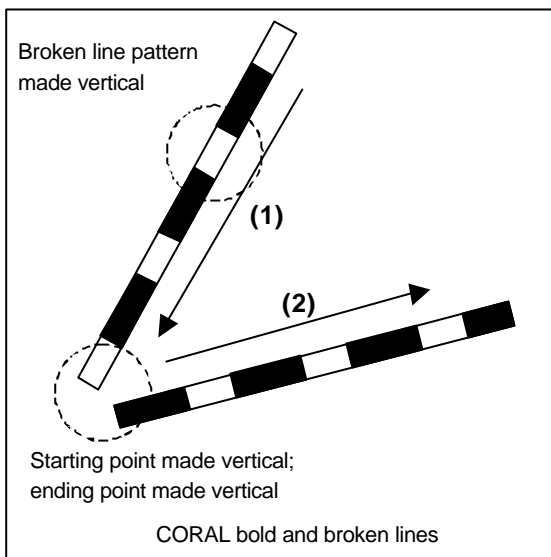
### 8.7.1 Starting and ending points

- 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 line is generated by different algorithm. Thus drawing position is little bit different from other primitive.



### 8.7.2 Broken line pattern

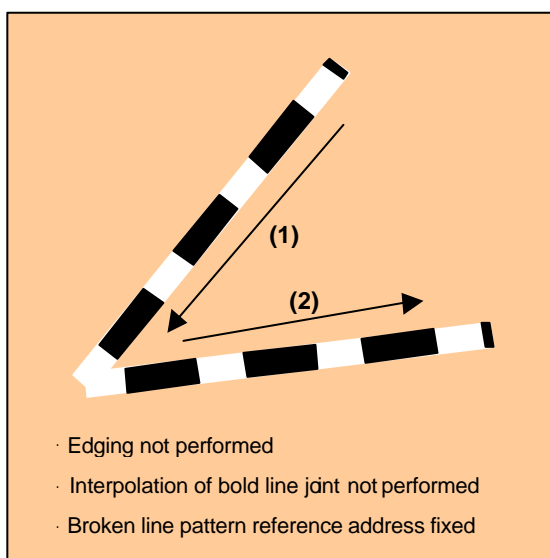
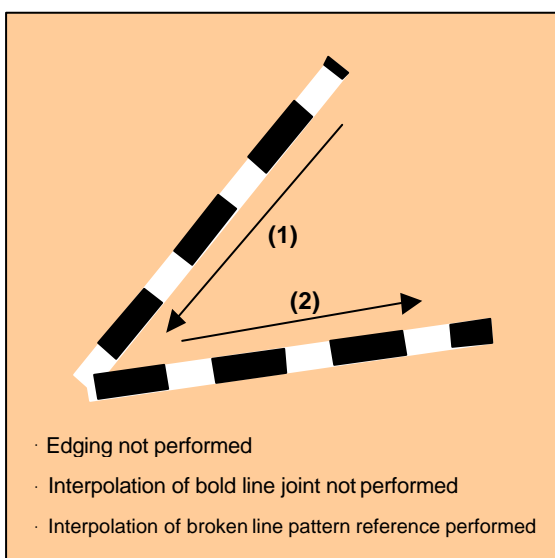
- The broken line pattern vertical to the theoretical line (the CORAL broken line pattern) is supported.
- In the CREMSON bold line mode, lines can be drawn using the broken line pattern vertical to the CREMSON-compatible principal axis (the CREMSON broken line pattern), and can also be drawn using the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



### Interpolation of broken line pattern

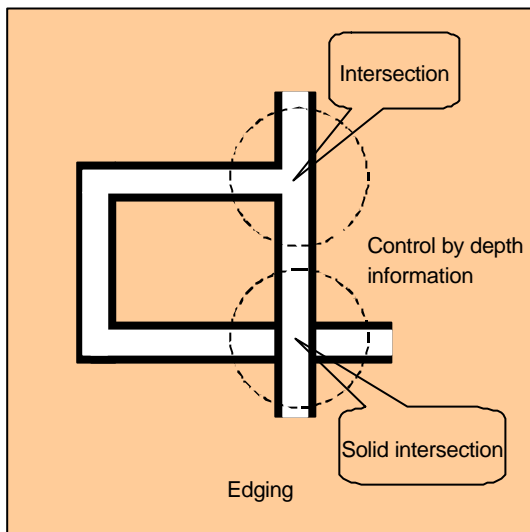
Two types of interpolation modes are supported:

- 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.



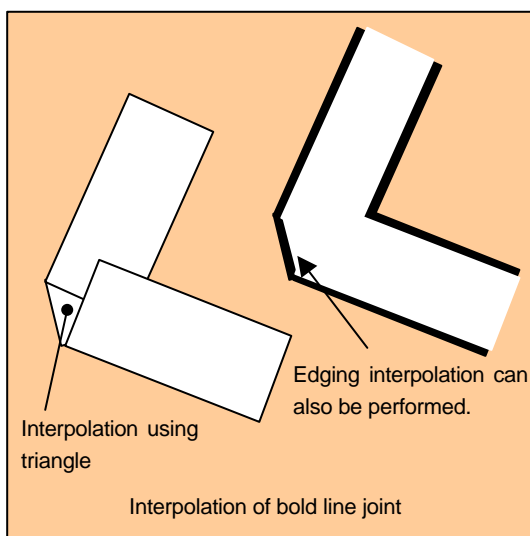
### 8.7.3 Edging

- 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)





## 8.8 DISPLAY LIST

### 8.8.1 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**

### 8.8.2 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    |  |  |    |    |  |      |  |        |  |        |  |  |   |

**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       |

### 8.8.3 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.

### 8.8.4 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<br>This command is used at execution of the CORAL extended function.  |
| G_BeginECont       | —  | Specifies primitive type (vertex processing in same mode as previous mode)<br>This command is used at execution of the CORAL extended function.        |
| G_End              | —  | Ends primitive<br>This command is used at execution of G_Begin or G_BeginCont  |
| G_EndE             | —  | Ends primitive<br>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<br>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) Floating point setup type → Integer setup type

This function is deleted. (Coral Series)

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

(2) Integer setup type

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 |

(3) “Unclipped” integer setup type

This command does not clip the view volume.

Only “XY” is enabled as the input parameter.

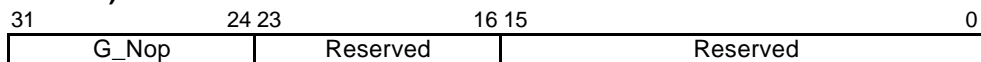
In setup processing, “XY” is calculated in the integer format.

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 |

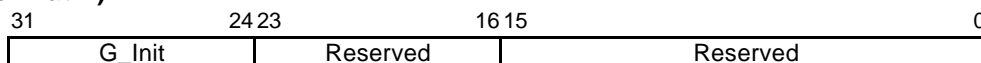
### 8.8.5 Explanation of geometry commands

#### G\_Nop (Format 1)



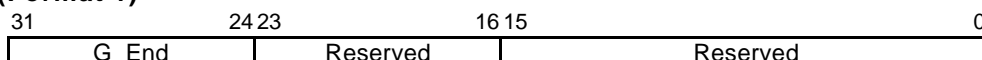
No operation

#### G\_Init (Format 1)



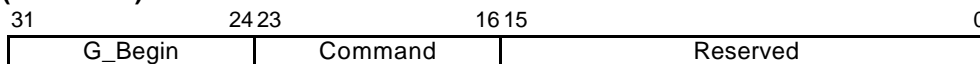
The **G\_Init** command initializes geometry engine. Execute this command before processing.

#### 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\*)

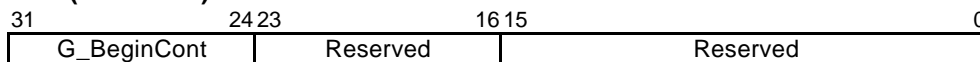
| (ST,Z,C)         | Point | Line | Triangle | Polygon |
|------------------|-------|------|----------|---------|
| (0,0,0)          | ○     | ○    | ○        | ○       |
| Other than above | ∕     | ∕    | ∕        | ∕       |

Primitives other than unclipped primitives

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

\*1: Shading is not assured.

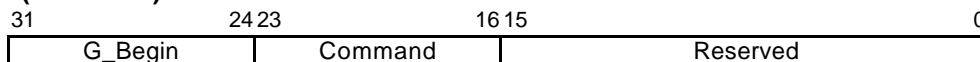
**G\_BeginCont (Format 1)**



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

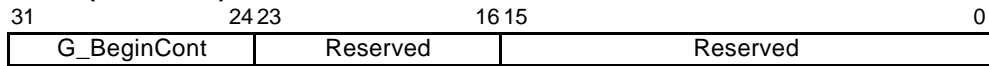
| (ST,Z,C) | Point | Line | Triangle | Polygon*2 |
|----------|-------|------|----------|-----------|
| (0,0,0)  | ○     | ○    | ○        | ○         |
| (0,0,1)  | ∕     | ∕    | ○        | ∕         |
| (0,1,0)  | ○     | ○    | ○        | ○         |
| (0,1,1)  | ∕     | ∕    | ○        | ∕     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

\*1: Shading is not assured.

\*2: Texture and depth quality is less than Triangle



**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”.

**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 | <b>ShadowXY</b> command sets the XY offset of the shade primitive relative to the body primitive.  |
| ShadowXYcomposition | 0000_0001 | <b>ShadowXYcomposition</b> command sets the XY offset of the shade synthetic primitive relative to the body primitive.<br>It command synthesizes a shade from the relationship between the XY offset set using <b>ShadowXY</b> 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.

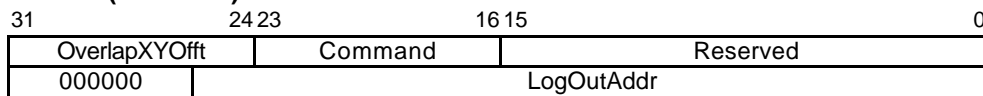
At this time, the following relationship must be satisfied when, for example, GREATER is specified for the Z value comparison mode:

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

Command:

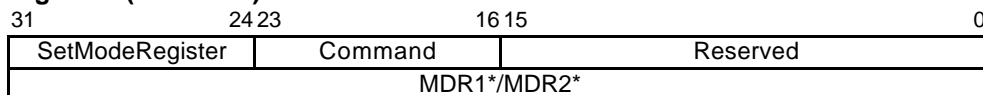
| Command     | Code      | Explanation   |
|-------------|-----------|---|
| Origin      | 0000_0000 | <b>Origin</b> command sets the Z offset of the body primitive. When drawing one primitive below the other primitive (for example, when drawing a solid intersection), this Z offset is changed. When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

**DC\_LogOutAddr (Format5)**



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

**SetModeRegister (Format5)**

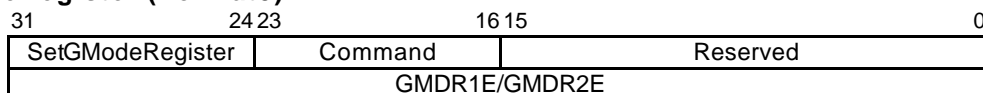


The **SetModeRegister** command sets the mode register for shade primitive, for edge primitive, and for top-left non-applicable primitive. At drawing of these primitives, also set the mode register (MDR1/MDR2) for the body primitive, using this packet.

Command:

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

**SetGModeRegister (Format5)**

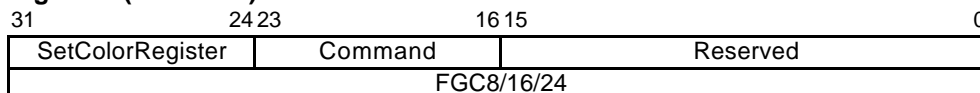


The **SetGModeRegister** command sets the geometry extended mode register.

Command:

| Command | Code      | Explanation  |
|---------|-----------|--|
| GMDR1E  | 0001_0000 | <b>GMDR1E</b> command sets GMDR1E and at the same time, updates GMDR1. |
| GMDR2E  | 0010_0000 | <b>GMDR2E</b> command sets GMDR2E and at the same time, updates GMDR2. |

**SetColorRegister (Format5)**

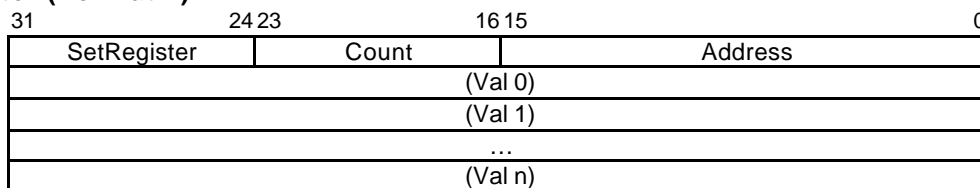


The **SetColorRegister** command sets the foreground color and background color of the body primitive, shade primitive, and edge primitive.

Commands:

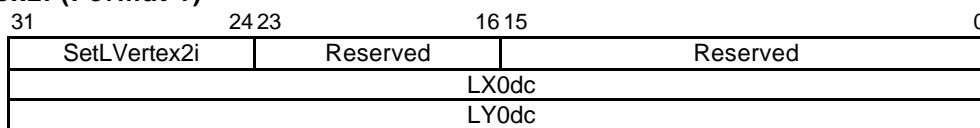
| Command         | Code      | Explanation   |
|-----------------|-----------|---|
| ForeColor       | 0000_0000 | <b>ForeColor</b> command sets the foreground color for the body primitive.        |
| BackColor       | 0000_0001 | <b>BackColor</b> command sets the background color for the body primitive.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

**SetRegister (Format 2)**



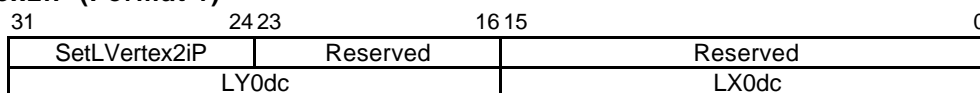
The **SetRegister** command is upper compatible with CREMSON **SetRegister**. It can specify the address of a register in the geometry engine.

**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.

## 8.9 Rendering Command

### 8.9.1 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<br>DrawLine2iP            | ZeroVector      | Draws high-speed 2DLine (with vertex 0 as starting point)                  |
|                                      | OneVector       | Draws high-speed 2DLine (with vertex 1 as starting point)                  |
| DrawTrap                             | TrapRight       | Draws right triangle   |
|                                      | TrapLeft        | Draws left triangle  |
| DrawVertex2i<br>DrawVertex2iP        | TriangleFan     | Draws high-speed 2DTriangle  |
|                                      | 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 Blit   |
|                                      | Bitmap          | Draws binary bit map (character)   |
| BlitCopyP<br>BlitCopy-<br>AlternateP | TopLeft         | Blit transfer from top left coordinates                                    |
|                                      | TopRight        | Blit transfer from top right coordinates                                   |
|                                      | BottomLeft      | Blit transfer from bottom left coordinates                                 |
|                                      | BottomRight     | Blit transfer from bottom right coordinates                                |
| LoadTextureP                         | LoadTexture     | Loads texture pattern  |
|                                      | LoadTILE        | Loads tile pattern   |
| BlitTextureP                         | LoadTexture     | Loads texture pattern from local memory                                    |
|                                      | LoadTILE        | Loads tile pattern from local memory                                       |
| BlitCopyAlt-<br>AlphaBlendP          | —               | Alpha blending is supported (see the alpha map).<br>BlitCopyAlternateP     |



Type Code Table

| Type                   | Code      |
|------------------------|-----------|
| DrawPixel              | 0000_0000 |
| DrawPixelZ             | 0000_0001 |
| DrawLine               | 0000_0010 |
| DrawLine2i             | 0000_0011 |
| DrawLine2iP            | 0000_0100 |
| DrawTrap               | 0000_0101 |
| DrawVertex2i           | 0000_0110 |
| DrawVertex2iP          | 0000_0111 |
| DrawRectP              | 0000_1001 |
| DrawBitmapP            | 0000_1011 |
| BitCopyP               | 0000_1101 |
| BitCopyAlternateP      | 0000_1111 |
| LoadTextureP           | 0001_0001 |
| BlitTextureP           | 0001_0011 |
| BlitCopyAltAlphaBlendP | 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)

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

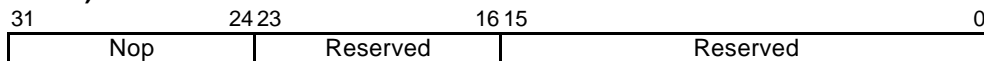
**Command Code Table (2)**

| <b>Command</b>  | <b>Code</b> |
|-----------------|-------------|
| BlitFill        | 010_00001   |
| BlitDraw        | 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   |

### 8.9.2 Details of rendering commands

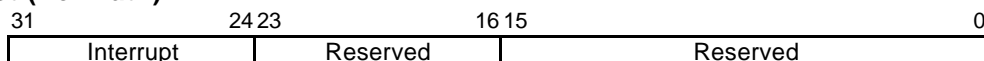
All parameters belonging to their command are stored in relevant registers. The definition of each parameter is explained in the section of each command.

#### Nop (Format1)



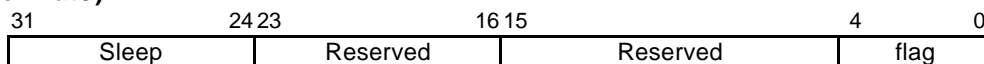
No operation

#### 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)**

|      |         |          |   |
|------|---------|----------|---|
| 31   | 24 23   | 16 15    | 0 |
| Draw | Command | Reserved |   |

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       Flushes drawing data in the drawing pipeline into the graphics memory. Place this command at the end of the display list.
- Flush\_Z        Flushes Z value data in the drawing pipeline into the graphics memory. When using the Z buffer, place this command together with the **Flush\_FB** command at the end of the display list.

**DrawPixel (Format5)**

|           |         |          |   |
|-----------|---------|----------|---|
| 31        | 24 23   | 16 15    | 0 |
| DeawPixel | Command | Reserved |   |
| PXs       |         |          |   |
| PYs       |         |          |   |

The **DrawPixel** command draws pixel.

Command:

- Pixel            Draws pixel without Z value.

**DrawPixelZ (Format5)**

|           |         |          |   |
|-----------|---------|----------|---|
| 31        | 24 23   | 16 15    | 0 |
| DeawPixel | Command | Reserved |   |
| PXs       |         |          |   |
| PYs       |         |          |   |
| PZs       |         |          |   |

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        |        |

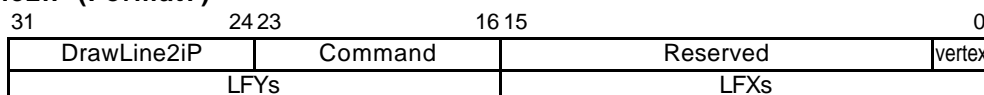
The **DrawLine2i** command draws high-speed 2DLine. It starts drawing after setting parameters at the high-speed 2DLine drawing registers. Integer data can only be used for coordinates.

## 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.

**BltCopyP (Format5)**

|          |         |          |   |
|----------|---------|----------|---|
| 31       | 24 23   | 16 15    | 0 |
| BltCopyP | Command | Reserved |   |
| SRYs     |         | SRXs     |   |
| DRYs     |         | DRXs     |   |
| BysizeY  |         | BysizeX  |   |

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     |   |
| BysizeY           |         | BysizeX  |   |

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

Command:

- TopLeft                      Starts BitBlt transfer from top left coordinates.

**LoadTextureP (Format6)**

|              |         |       |   |
|--------------|---------|-------|---|
| 31           | 24 23   | 16 15 | 0 |
| LoadTextureP | Command | Count |   |
| (Pattern 0)  |         |       |   |
| (Pattern 1)  |         |       |   |
| ...          |         |       |   |
| (Pattern n)  |         |       |   |

The **LoadTextureP** command loads texture or tile pattern into internal texture buffer.

It stores a texture pattern into the texture buffer based on the current pattern size (TXS/TIS) and offset address (XBO).

Commands:

- LoadTexture                 Stores texture pattern into internal texture buffer.
- LoadTile                    Stores tile pattern into internal texture buffer.

**BlitTextureP (Format5)**

|              |         |           |   |
|--------------|---------|-----------|---|
| 31           | 24 23   | 16 15     | 0 |
| BlitTextureP | Command | Reserved  |   |
| SrcADDR      |         |           |   |
| SrcStride    |         |           |   |
| SrcRectYs    |         | SrcRectXs |   |
| BysizeY      |         | BysizeX   |   |
| DestOffset   |         |           |   |

The **BlitTextureP** command loads texture or tile pattern into texture buffer from Graphics Memory. It stores a texture pattern into the texture buffer current pattern size (TXS/TIS) and offset address (XBO).

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

Commands:

- LoadTexture                 Stores texture pattern into internal texture buffer.
- LoadTile                    Stores tile pattern into internal texture buffer.

**BltCopyAltAlphaBlendP (Format5)**

|                   |       |          |          |
|-------------------|-------|----------|----------|
| 31                | 24 23 | 16 15    | 0        |
| BltCopyAlternateP |       | Command  | Reserved |
| SADDR             |       |          |          |
| SStride           |       |          |          |
| SRYs              |       | SRXs     |          |
| BlendStride       |       |          |          |
| BlendRYs          |       | BlendRXs |          |
| DRYs              |       | DRXs     |          |
| BsizeY            |       | BsizeX   |          |

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.

## 9. PCI Configuration Registers

For the Coral-LP, the PCI Configuration registers are divided into two subgroups:

1. Device specific registers (eg. Vendor ID). These should not normally be modified by the user. These registers can be loaded from EEPROM.
2. Application specific registers (eg. PCI Command Register). These can be modified by the user and must be programmed using PCI Configuration cycles as they can not be loaded from the EEPROM. However an EEPROM loadable 32 bit register is available for the user.

For the EEPROM loadable configuration registers, the Coral-LP uses Byte Addresses which are used on the PCI bus. However, when in 16 bit data mode the EEPROM requires word addresses. The EEPROM preloaded using the 16 bit word addresses shown in the below.

### 9.1 PCI Configuration register list

| 31:24                  | 23:16       | 15:8                 | 7:0            | PCI Byte Address | EEPROM Word Address |    |
|------------------------|-------------|----------------------|----------------|------------------|---------------------|----|
| DEVICE ID              |             | VENDER ID            |                | 00               | 01                  | 00 |
| STATUS                 |             | COMMND               |                | 04               | -                   | -  |
| CLASS CODE             |             |                      | REVISION ID    | 08               | 05                  | 04 |
| BIST                   | HEADER TYPE | MASTER LATENCY TIMER | CACHELINE SIZE | 0C               | 07                  | -  |
| BASE ADDRESS REGISTER0 |             |                      |                | 10               | -                   | -  |
| RESERVED               |             |                      |                | 14               | -                   | -  |
| RESERVED               |             |                      |                | 18               | -                   | -  |
| RESERVED               |             |                      |                | 1C               | -                   | -  |
| RESERVED               | RESERVED    | RESERVED             | RESERVED       | 20               | -                   | -  |
| RESERVED               | RESERVED    | RESERVED             | RESERVED       | 24               | -                   | -  |
| RESERVED               | RESERVED    | RESERVED             | RESERVED       | 28               | -                   | -  |
| SUBSYSTEM ID           |             | SUBSYSTEM VENDOR ID  |                | 2C               | 17                  | 16 |
| RESERVED               | RESERVED    | RESERVED             | RESERVED       | 30               | -                   | -  |
| RESERVED               |             |                      |                | 34               | -                   | -  |
| RESERVED               |             |                      |                | 38               | -                   | -  |
| MAX LAT                | MIN GNT     | INTERRUPT PIN        | INTERRUPT LINE | 3C               | 1F                  | 1E |
| RESERVED               |             | RETRY TIME OUT       | TRDY TIME OUT  | 40               | -                   | -  |
| USER REGISTER          |             |                      |                | 44               | 23                  | 22 |

## 9.2 PCI Configuration Registers Descriptions

In the following sections, the following abbreviations in the “Type” field apply:

**RO:** Register is Read-only, not loadable via EEPROM.

**ER:** Register is Read-only, loadable via EEPROM.

**RW:** Register is Read/Writable using PCI configuration transactions; not loadable via EEPROM.

For further information about these fields, please refer to the PCI Specification v2.1, Section6.

### Vendor ID Register

| Bit  | Type | Reset Value | Description   |
|------|------|-------------|---|
| 15-0 | ER   | 10CFh       | Identifies the vendor of the IC. The Reset Value represents the vendor ID of Fujitsu Limited. |

### Device ID Register

| Bit  | Type | Reset Value | Description   |
|------|------|-------------|---|
| 15-0 | ER   | 2019h       | ID of Fujitsu Limited PCI device (Coral device ID). |

### PCI Command Register

| Bit   | Type | Reset Value | Description   |
|-------|------|-------------|---|
| 15-10 | -    | 0           | Reserved  |
| 9     | RW   | 0           | Fast Back-to-Back Master Enable. This is not supported by the Coral-LP and should be set to '0'   |
| 8     | RW   | 0           | System Error Enable. This is supported by the Coral-LP.   |
| 7     | -    | 0           | Reserved  |
| 6     | RW   | 0           | Parity Error Enable. This is supported by the Coral-LP.   |
| 5     | -    | 0           | Reserved  |
| 4     | RW   | 0           | Memory Write and Invalidate Enable. This feature is not supported in master mode, but in slave mode the Coral-LP will convert any Memory Write and Invalidate commands to Memory Write commands. This bit should be set to '0'. |
| 3     | -    | 0           | Reserved  |
| 2     | RW   | 0           | Bus Master Enable. This bit must be set to '1' by the user for correct operation.   |
| 1     | RW   | 0           | Memory Access Enable. This bit must be set to '1' by the user for correct operation.  |
| 0     | RW   | 0           | I/O Access Enable. The Coral-LP does not do I/O Accesses.   |



**PCI Status Register**

| Bit  | Type   | Reset Value | Description   |
|------|--------|-------------|---|
| 15   | Status | 0           | Parity Error has been detected by the Coral-LP.   |
| 14   | Status | 0           | System Error has been signaled by the Coral-LP.   |
| 13   | Status | 0           | Received Master Abort. Set to '1' when a PCI Master terminates a user to the Coral-LP transaction with Master Abort.      |
| 12   | Status | 0           | Received Target Abort. Set to '1' when the Coral-LP has initiated a transaction that has been terminated by Target Abort. |
| 11   | Status | 0           | Target Abort has been signaled by the Coral-LP.   |
| 10-9 | RO     | 01          | Device Select Timing. Indicates the timing of the <b>DEVSEL#</b> signal when the Coral-LP responds as a PCI Target.       |
| 8    | Status | 0           | Data Parity Error detected.   |
| 7    | RO     | 1           | Fast Back-to-Back Capable Status Flag.  |
| 6    | -      | 0           | Reserved  |
| 5    | RO     | 0           | 66MHz Capable Flag.   |
| 4-0  | -      | -           | Reserved  |

**Revision ID Register**

| Bit | Type | Reset Value | Description                  |
|-----|------|-------------|------------------------------|
| 7-0 | ER   | 01h         | Revision ID of the Coral-LP. |

**PCI Class Code Register**

| Bit  | Type | Reset Value | Description  |
|------|------|-------------|--|
| 23-0 | ER   | 038000h     | Class Code of the Coral-LP. The Reset value means "Display Controller" of non-specific type. |

**Casheline Size Register**

| Bit | Type | Reset Value | Description     |
|-----|------|-------------|-----------------|
| 7-0 | RW   | 0           | Casheline Size. |

**Master Latency Timer Register**

| Bit | Type | Reset Value | Description   |
|-----|------|-------------|---|
| 7-2 | RW   | 0           | Master Latency Timer Count Value. This register sets the minimum number of PCI clocks the Coral-LP is guaranteed access to the PCI bus. After the count has expired, the Coral-LP releases the PCI bus as soon as another PCI Master is granted the bus by the bus arbiter. |
| 1-0 | -    | 0           | Reserved  |

**Header Type Register**

| Bit | Type | Reset Value | Description   |
|-----|------|-------------|---|
| 7-0 | ER   | 0           | As defined in the PCI Specification, Section 6.2.1. |

**BIST Register**

| Bit | Type | Reset Value | Description  |
|-----|------|-------------|--|
| 7-0 | -    | 0           | This field is not used by the Coral-LP, so it is hard-wired to zero. |

**Memory Base Address Register**

| Bit | Type | Reset Value | Description   |
|-----|------|-------------|---|
| 31  | RW   | 0           | Memory Base Address. This determines the address of the first Coral-LP non PCI register. The Coral-LP will respond as a Target to accesses in the address range:<br>(memory_base_address) to (memory_base_address + 3FF0000H) |

**Subsystem Vendor ID Register**

| Bit  | Type | Reset Value | Description   |
|------|------|-------------|---|
| 15-0 | ER   | 0           | Subsystem Vendor ID. This register can be loaded from EEPROM. |

**Subsystem ID Register**

| Bit  | Type | Reset Value | Description   |
|------|------|-------------|---|
| 15-0 | ER   | 0           | Subsystem ID. This register can be loaded from EEPROM |

**Interrupt Line Register**

| Bit | Type | Reset Value | Description   |
|-----|------|-------------|---|
| 7-0 | RW   | 0           | Interrupt Line Register. Used to convey interrupt line routing information. |

**Interrupt Pin Register**

| Bit | Type | Reset Value | Description   |
|-----|------|-------------|---|
| 7-0 | RW   | 1           | Identifies which PCI Interrupt pin the Coral-LP is connected to. The default value of this indicate that the Coral-LP is connected to the INTA line, which is the usual setting for this field. |

**Min Grant Register**

| Bit | Type | Reset Value | Description   |
|-----|------|-------------|---|
| 7-0 | ER   | 0           | Identifies the maximum length of PCI burst period the Coral-LP needs. This should be left at the reset setting. |

**Max Latency Register**

| Bit | Type | Reset Value | Description  |
|-----|------|-------------|--|
| 7-0 | ER   | 0           | Specifies how often the Coral-LP needs to access the bus. This should be left at the reset settings. |

**TRDY Timeout Value Register**

| Bit | Type | Reset Value | Description   |
|-----|------|-------------|---|
| 7-0 | RW   | 80h         | Sets the number of PCI clocks the Coral-LP will wait for TRDY, when acting as a Bus Master. |

**Retry Timeout Value Register**

| Bit | Type | Reset Value | Description  |
|-----|------|-------------|--|
| 7-0 | RW   | 80h         | Sets the number of retries of the Coral-LP will perform when acting as a Bus Master. |

**User Programmable Register**

| Bit  | Type | Reset Value | Description                |
|------|------|-------------|----------------------------|
| 31-0 | ER   | 0           | User programmable register |

## 10 Local Memory Registers

### 10.1 Local memory 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    |
|--------|-------|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|-------|-----|----|----|----|----|---|---|-------|---|---|---|---|---|---|------|
| 001C   | MRO   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   | MRO  |
| 0020   | IST   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        | IST   |    |    |    |    |    |    |    | IST   |    |    |    |    |    |    |    | IST   |     |    |    |    |    |   |   | IST   |   |   |   |   |   |   |      |
| 0024   | IMASK |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        | IMASK |    |    |    |    |    |    |    | IMASK |    |    |    |    |    |    |    | IMASK |     |    |    |    |    |   |   | IMASK |   |   |   |   |   |   |      |
| 002C   |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    | SRST  |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   | SRST |
| 0038   | CCF   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    | CGE   | COT |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
| 005C   |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    | RSW   |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   | RSW  |
| 0070   | IP    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   | IP   |
| 0074   | BTV   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    | BTV   |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
| 0078   | FTV   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    | FTV   |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
| 007C   | OFU   |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   | OFU  |
| 00A0   | FRST  |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   | FRST |
| 00A4   | SRBS  |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   |      |
|        |       |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |       |     |    |    |    |    |   |   |       |   |   |   |   |   |   | SRBS |

|             |            |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|-------------|------------|-----|-----|-------|--|--|--|--|--|--|--|--|--|--|------|-------|-------|-------|-----|-----|-------|------|--|--|-----|-----|-----|-----|-----|-------|-----|-----|--|--|--|--|--|--|--|--|--|--|--|--|---|---|
| 00A8        | IOM        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             | GIM        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       | GD    |       |     |     |       |      |  |  | SER | RGB | BEE | SBE | TCE | BCE   | EEE |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| 00AC        | GD         |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             | GWE        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       | GD    |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| 00B0        | SIC        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             |            |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       | CKP   | CKG   | CKD |     |       |      |  |  |     |     |     |     |     |       |     | DOE |  |  |  |  |  |  |  |  |  |  |  |  | S | S |
| 00B4        | SID        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             |            |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       | FSL   | FS    | TLS |     |       |      |  |  |     |     | RWD |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| 00F0        | CID        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             |            |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       | CN    |       |     |     |       |      |  |  | VER |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| 8000        | BSA        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             | SA         |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| 8004        | BDA        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             | DA         |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| 8008        | BCR        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             | STRT       | NDA | NSA | BSIZE |  |  |  |  |  |  |  |  |  |  |      | TSIZE |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| 800C        | BSR        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             |            |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       | XCOR  | IMODE | TCM | BCM | EXTEN | MODE |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| 8010        | BER        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             |            |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       | ABORT |       |     |     |       |      |  |  |     |     |     |     |     | EXTST | BEN |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| 8014        | BST        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
|             | TC         | BC  |     |       |  |  |  |  |  |  |  |  |  |  | TCNT |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| 8040        | BCB        |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |
| ...<br>805C | RWDATA * 8 |     |     |       |  |  |  |  |  |  |  |  |  |  |      |       |       |       |     |     |       |      |  |  |     |     |     |     |     |       |     |     |  |  |  |  |  |  |  |  |  |  |  |  |   |   |

### 10.1.2 I<sup>2</sup>C interface register list

Base = I<sup>2</sup>CBase

| 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    | Reserved           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | BSR |   |   |   |   |   |   |   |
| 004    | Reserved           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | BCR |   |   |   |   |   |   |   |
| 008    | Reserved           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | CCR |   |   |   |   |   |   |   |
| 00C    | Reserved           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | ADR |   |   |   |   |   |   |   |
| 010    | Reserved           |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | DAR |   |   |   |   |   |   |   |
| 014    | Access Prohibition |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
| 018    | Access Prohibition |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |
| 01C    | Access Prohibition |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |     |   |   |   |   |   |   |   |

### 10.1.3 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   | DTC |     |    |    |    |      |    |     |    |     |    |      |    |      |    |      |    |     |    |     |    |     |   |    |   |   |   |   |   |   |   |   |
|        |     | TWR |    | ID |    | TTRD |    | TRC |    | TRP |    | TRAS |    | TRCD |    | LOWD |    | RTS |    | SAW |    | ASW |   | CL |   |   |   |   |   |   |   |   |

### 10.1.4 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 | L0E | OXS | DCS |     |                 |    |   |   |   |   |   |      |      |   |   | SC | EEO | EDE | EOF | EOD | SF | ESY | SYNC |    |     |     |     |     |    |     |
| 100    | DCEE (Display Controller Extend Enable) |       |      |     |    |    |    |     |    |    |    |    |    |                 |                             |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
|        | DEN                                     |       |      |     |    |    |    |     |    |    |    |    |    | L5E             | L4E                         | L3E | L2E | L1E | L0E | OXS | DCS             |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      | SC | EEO | EDE | EOF | EOD | SF | ESY |
| 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    | LOM (L0 Mode)                           |       |      |     |    |    |    |     |    |    |    |    |    |                 |                             |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
|        | LOC                                     |       |      |     |    |    |    |     |    |    |    |    |    | LOS (L0 Stride) |                             |     |     |     |     |     | LOH (L0 Height) |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
| 024    | LOOA (L0 Origin Address)                |       |      |     |    |    |    |     |    |    |    |    |    |                 |                             |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
| 028    | L0DA (L0 Display Address)               |       |      |     |    |    |    |     |    |    |    |    |    |                 |                             |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
| 02C    | L0DY (L0 Display Y)                     |       |      |     |    |    |    |     |    |    |    |    |    |                 | L0DX (L0 Display X)         |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
| 110    | LOEM (L0 Extend Mode)                   |       |      |     |    |    |    |     |    |    |    |    |    |                 |                             |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
|        | LOEC                                    |       |      |     |    |    |    |     |    |    |    |    |    | LOPB            |                             |     |     |     |     |     |                 |    |   |   |   |   |   | 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 | L1M |    |    |    |     |    |    |    |    |    | L1S (L1 Stride) |                             |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
| 034    | L1DA (L1 Display Address)               |       |      |     |    |    |    |     |    |    |    |    |    |                 |                             |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
| 120    | L1EM (L1 Extend Mode)                   |       |      |     |    |    |    |     |    |    |    |    |    |                 |                             |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
|        | L1EC                                    |       |      |     |    |    |    |     |    |    |    |    |    | L1PB            |                             |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
| 040    | L2M (L2 Mode)                           |       |      |     |    |    |    |     |    |    |    |    |    |                 |                             |     |     |     |     |     |                 |    |   |   |   |   |   |      |      |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
|        | L2C                                     | L2FLP |      |     |    |    |    |     |    |    |    |    |    | L2S (L2 Stride) |                             |     |     |     |     |     | 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            |                             |     |     |     |     |     |                 |    |   |   |   |   |   | L2OM | L2WP |   |   |    |     |     |     |     |    |     |      |    |     |     |     |     |    |     |
| 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 |    | L3S (L3 Stride) |    |    |    |    |    |    |    |    |    |      |      | L3H (L3 Height)        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 05C    | L3OA0 (L3 Origin Address 0)  |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 060    | L3DA0 (L3 Display Address 0) |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 064    | L3OA1 (L3 Origin Address 1)  |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 068    | L3DA1 (L3 Display Address 1) |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 06C    | L3DY (L3 Display Y)          |       |    |                 |    |    |    |    |    |    |    |    |    |      |      | L3DX (L3 Display X)    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 140    | L3EM (L3 Extend Mode)        |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        | L3EC                         | L3PB  |    |                 |    |    |    |    |    |    |    |    |    | L3OM | L3WP |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 144    | L3WY (L3 Window Y)           |       |    |                 |    |    |    |    |    |    |    |    |    |      |      | L3WX (L3 Window X)     |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 148    | L3WH (L3 Window Height)      |       |    |                 |    |    |    |    |    |    |    |    |    |      |      | L3WW (L3 Window Width) |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 070    | L4M (L4 Mode)                |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        | L4C                          | L4FLP |    | L4S (L4 Stride) |    |    |    |    |    |    |    |    |    |      |      | L4H (L4 Height)        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 074    | L4OA0 (L4 Origin Address 0)  |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 078    | L4DA0 (L4 Display Address 0) |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 07C    | L4OA1 (L4 Origin Address 1)  |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 080    | L4DA1 (L4 Display Address 1) |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 084    | L4DY (L4 Display Y)          |       |    |                 |    |    |    |    |    |    |    |    |    |      |      | L4DX (L4 Display X)    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 150    | L4EM (L4 Extend Mode)        |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        | L4EC                         | L4WB  |    |                 |    |    |    |    |    |    |    |    |    | L4OM | L4WP |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 154    | L4WY (L4 Window Y)           |       |    |                 |    |    |    |    |    |    |    |    |    |      |      | L4WX (L4 Window X)     |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 158    | L4WH (L4 Window Height)      |       |    |                 |    |    |    |    |    |    |    |    |    |      |      | L4WW (L4 Window Width) |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 088    | L5M (L5 Mode)                |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        | L5C                          | L5FLP |    | L5S (L5 Stride) |    |    |    |    |    |    |    |    |    |      |      | L5H (L5 Height)        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 08C    | L5OA0 (L5 Origin Address 0)  |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 090    | L5DA0 (L5 Display Address 0) |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 094    | L5OA1 (L5 Origin Address 1)  |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 098    | L5DA1 (L5 Display Address 1) |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 09C    | L5DY (L5 Display Y)          |       |    |                 |    |    |    |    |    |    |    |    |    |      |      | L5X (L5 Display X)     |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 110    | L5EM (L5 Extend Mode)        |       |    |                 |    |    |    |    |    |    |    |    |    |      |      |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        | L5EC                         | L5WB  |    |                 |    |    |    |    |    |    |    |    |    | L5OM | L5WP |                        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 164    | L5WY (L5 Window Y)           |       |    |                 |    |    |    |    |    |    |    |    |    |      |      | L5WX (L5 Window X)     |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 168    | L5WH (L5 Window Height)      |       |    |                 |    |    |    |    |    |    |    |    |    |      |      | L5WW (L5 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 |
|--------|----------------------------------|----|-------|----|------|----|------|----|-----------------------------------|----|------|----|------|----|----|----|------|----|----|----|------|----|---|---|---|---|---|---|---|---|---|---|
| 0A0    | CSIZE                            |    |       |    | CPM  |    |      |    | CUTC (Cursor Transparent Control) |    |      |    |      |    |    |    |      |    |    |    |      |    |   |   |   |   |   |   |   |   |   |   |
|        | CSIZ1                            |    | CSIZ0 |    | CUE1 |    | CUE0 |    | CUO1                              |    | CUO0 |    | 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                             |    |       |    | LOBS |    |      |    | 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                              |    |      |    | L5BP |    |    |    | 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)       |                             |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |  |                             |    |    |    |    |    |    |    |    |    |    |    |    |    |    | L0ZT                                | L0TC (L0 Transparent Color) |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 0C0    | L2TR (L2 Transparent Control)          |                             |    |    |    |    |    |    |    |    |    |    |    |    |    |    | L3TR (L3 Transparent Control)       |                             |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        | L2ZT                                   | L2TC (L2 Transparent Color) |    |    |    |    |    |    |    |    |    |    |    |    |    |    | L3ZT                                | 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) |                             |    |    |    |    |   |   |   |   |   |   |   |   |   |   |



### 10.1.5 Video capture register list

Base = CaptureBase

| 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    | VCM (Video Capture Mode)                  |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        | VE  |    |    |    |       |    | CM |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     | VS |        |    |  |  |  |  |  |  |  |  |  |
| 004    | CSC(Capture SScale)                       |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        | VSCI                                      |    |    |    | VSCF  |    |    |    |        |    |    |    | HSCI  |    |        |    | HSCF |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
| 008    | VCS(Video Capture Status)                 |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    | CE     |    |  |  |  |  |  |  |  |  |  |
| 010    | CBM(Capture Buffer Mode)                  |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        | ∞   |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   | CBW |    |        |    |  |  |  |  |  |  |  |  |  |
| 014    | CBOA(Capture Bauffer Origin Address)      |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    |        |    |    |    |       |    | CBOA   |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
| 018    | CBLA(Capture Buffer Limit Address)        |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    |        |    |    |    |       |    | CBLA   |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
| 01C    |   |    |    |    |       |    |    |    |        |    |    |    |       |    | CIVSTR |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    | CIHSTR |    |  |  |  |  |  |  |  |  |  |
| 020    |   |    |    |    |       |    |    |    |        |    |    |    |       |    | CIVEND |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    | CIHEND |    |  |  |  |  |  |  |  |  |  |
| 028    | CHP(Capture Horizontal Pixel)             |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    | CHP    |    |  |  |  |  |  |  |  |  |  |
| 02C    | CVP(Capture Vertical Pixel)               |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    | CVPP   |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    | CVPN   |    |  |  |  |  |  |  |  |  |  |
| 040    | CLPF(Capture Low Pass Filter)             |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    | CVLPF |    |    |    |        |    |    |    | CHLPF |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
| 048    | CMSS(Capture Magnify Source Size)         |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    | CMSHP  |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    | CMSVL  |    |  |  |  |  |  |  |  |  |  |
| 04C    | CMDSD(Capture Magnify Display Size)       |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    | CMDHP  |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    | CMDVL  |    |  |  |  |  |  |  |  |  |  |
| 080    | RGBHC( RGB input HSYNC Cycle)             |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    | RGBHC  |    |  |  |  |  |  |  |  |  |  |
| 084    | RGBHEN( RGB input Horizontal Enable Area) |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    | RGBHST |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    | RGBHEN |    |  |  |  |  |  |  |  |  |  |
| 088    | RGBVEN( RGB input Vertical Enable Area)   |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    | RGBVST |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    | RGBVEN |    |  |  |  |  |  |  |  |  |  |
| 090    | RGRS( RGB input SYNC)                     |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    |      |    |    |    |    |    |   |   |   |   |   |   |   |     |    |        |    |  |  |  |  |  |  |  |  |  |
|        |   |    |    |    |       |    |    |    |        |    |    |    |       |    |        |    | RM   |    |    |    |    |    |   |   |   |   |   |   |   |     |    | HP     | VP |  |  |  |  |  |  |  |  |  |

| 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 |
|--------|--|----|----|----|----|----|----|----|------|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| 0C0    | RGBCMY(RGB Color convert Matrix Y coefficient)   |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        | a11  |    |    |    |    |    |    |    | a11  |    |    |    |    |    |    |    | a11  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 0C4    | RGBCMCb(RGB Color convert Matrix Cb coefficient) |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        | a21  |    |    |    |    |    |    |    | a22  |    |    |    |    |    |    |    | a23  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 0C8    | RGBCMCr(RGB Color convert Matrix Cr coefficient) |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        | a31  |    |    |    |    |    |    |    | a32  |    |    |    |    |    |    |    | a33  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 0CC    | RGBCMb(RGB Color convert Matrix b coefficient)   |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        | b1   |    |    |    |    |    |    |    | b2   |    |    |    |    |    |    |    | b3   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 4000   | CDCN(Capture Data Count for NTSC)                |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |  |    |    |    |    |    |    |    | BDCN |    |    |    |    |    |    |    | VDCN |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 4004   | CDCP(Capture Data Count for PAL)                 |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|        |  |    |    |    |    |    |    |    | BDCP |    |    |    |    |    |    |    | VDCP |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

### 10.1.6 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<br>(000) | Ys    |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | Int |    |    |    |    |     |    |    | 0    |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 004<br>(001) | Xs    |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | Int |    |    |    |    |     |    |    | Frac |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 008<br>(002) | dXdY  |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | Int |    |    |    |    |     |    |    | Frac |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 00C<br>(003) | XUs   |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | Int |    |    |    |    |     |    |    | Frac |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 010<br>(004) | dXUdy |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | Int |    |    |    |    |     |    |    | Frac |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 014<br>(005) | XLs   |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | Int |    |    |    |    |     |    |    | Frac |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 018<br>(006) | dXLdy |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | Int |    |    |    |    |     |    |    | Frac |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 01C<br>(007) | USN   |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | 0     | 0  | 0  | 0  | Int |    |    |    |    |     |    |    | 0    |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 020<br>(008) | LSN   |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | 0     | 0  | 0  | 0  | Int |    |    |    |    |     |    |    | 0    |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 040<br>(010) | Rs    |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | 0     | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | Int |    |    |      |    |    |    |    | Frac |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 044<br>(011) | dRdx  |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | s   | s  | s  | s  | s  | Int |    |    |      |    |    |    |    | Frac |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 048<br>(012) | dRdy  |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | s   | s  | s  | s  | s  | Int |    |    |      |    |    |    |    | Frac |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 04C<br>(013) | Gs    |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | 0     | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | Int |    |    |      |    |    |    |    | Frac |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 050<br>(014) | dGdx  |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | s   | s  | s  | s  | s  | Int |    |    |      |    |    |    |    | Frac |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 054<br>(015) | dGdy  |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | s   | s  | s  | s  | s  | Int |    |    |      |    |    |    |    | Frac |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 058<br>(016) | Bs    |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | 0     | 0  | 0  | 0  | 0   | 0  | 0  | 0  | 0  | Int |    |    |      |    |    |    |    | Frac |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 05C<br>(017) | dBdx  |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | s  | s  | s  | s   | s  | s  | s  | s  | Int |    |    |      |    |    |    |    | Frac |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
| 060<br>(018) | dBdy  |    |    |    |     |    |    |    |    |     |    |    |      |    |    |    |    |      |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |
|              | s     | 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    |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|--------------|------|-----|----|-----|-----|----|----|----|-----|------|----|----|----|----|----|----|------|----|------|------|----|----|---|---|---|---|---|---|---|---|---|------|------|--|--|--|--|--|--|--|--|--|--|--|--|--|--|
| 080<br>(020) | Zs   |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | 0    | Int |    |     |     |    |    |    |     |      |    |    |    |    |    |    | Frac |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 084<br>(021) | dZdx |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | Int |    |     |     |    |    |    |     |      |    |    |    |    |    |    | Frac |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 088<br>(022) | dZdy |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | Int |    |     |     |    |    |    |     |      |    |    |    |    |    |    | Frac |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0C0<br>(030) | Ss   |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | Int |     |    |    |    |     |      |    |    |    |    |    |    |      |    | Frac |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0C4<br>(031) | dSdx |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | Int |     |    |    |    |     |      |    |    |    |    |    |    |      |    | Frac |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0C8<br>(032) | dSdy |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | Int |     |    |    |    |     |      |    |    |    |    |    |    |      |    | Frac |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0CC<br>(033) | Ts   |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | Int |     |    |    |    |     |      |    |    |    |    |    |    |      |    | Frac |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0D0<br>(034) | dTdx |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | Int |     |    |    |    |     |      |    |    |    |    |    |    |      |    | Frac |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0D4<br>(035) | dTdy |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | Int |     |    |    |    |     |      |    |    |    |    |    |    |      |    | Frac |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0D8<br>(036) | Qs   |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | 0    | 0   | 0  | 0   | 0   | 0  | 0  | 0  | INT | Frac |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0DC<br>(037) | dQdx |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | s   | s   | s  | s  | s  | INT | Frac |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 0E0<br>(038) | dQdy |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | s   | s   | s  | s  | s  | INT | Frac |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 140<br>(050) | LPN  |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | 0    | 0   | 0  | 0   | Int |    |    |    |     |      |    |    |    |    |    |    |      |    |      | 0    |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 144<br>(051) | LXs  |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | s   | Int |    |    |    |     |      |    |    |    |    |    |    |      |    |      | Frac |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 148<br>(052) | LXde |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | s   | s   | s  | s  | s  | s   | s    | s  | s  | s  | s  | s  | s  | s    | s  | s    | s    | s  | s  | s | s | s | s | s | s | s | s | Ξ | Frac |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 14C<br>(053) | LYs  |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | s   | Int |    |    |    |     |      |    |    |    |    |    |    |      |    |      | Frac |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 150<br>(054) | LYde |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | s   | s  | s   | s   | s  | s  | s  | s   | s    | s  | s  | s  | s  | s  | s  | s    | s  | s    | s    | s  | s  | s | s | s | s | s | s | s | s | s | Ξ    | Frac |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 154<br>(055) | LZs  |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | Int |    |     |     |    |    |    |     |      |    |    |    |    |    |    | Frac |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 158<br>(056) | LZde |     |    |     |     |    |    |    |     |      |    |    |    |    |    |    |      |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|              | s    | Int |    |     |     |    |    |    |     |      |    |    |    |    |    |    | Frac |    |      |      |    |    |   |   |   |   |   |   |   |   |   |      |      |  |  |  |  |  |  |  |  |  |  |  |  |  |  |







**FUJISTU LIMITED PRELIMINARY AND CONFIDENTIAL**

| 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)  |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FGC8/16/24 |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| 484    | BC    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| (121)  |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    | BGC8/16/24 |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| 488    | ALF   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| (122)  |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    | A          |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| 48C    | BLP   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| (123)  |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| 494    | TBC   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |            |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |
| (129)  |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    | BC16/24    |    |    |    |    |    |    |   |   |   |      |   |   |   |   |   |   |

| 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 |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| (150)  | 0     | 0  | 0  | 0  | Int |    |    |    |    |    |    |    |    |    |    |    | 0  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 54C    | LY1dc |    |    |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| (151)  | 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 Local Memory Registers

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

#### MRO (Mirror Register Override)

|                  |                         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |     |
|------------------|-------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|-----|
| Register address | HostBaseAddress + 001Ch |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |     |
| 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                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | MRO |
| R/W              | R0                      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | RW  |
| Initial value    | 0                       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   | 0   |

Writing a “1b” to this register overrides use of the Geometry/Draw Engine Mirror registers which reside in the host interface. Access to the Mirror registers is faster than the source registers in the Geometry/Draw Engines. For normal operation this register need not be used and should be kept as “0b”.

#### 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   | IST                   |    |     |    | *1  | IST | Reserved |    |    |    |    |      | Resv | Reserved |    |    |    |    |    |    |    |    |     | IST | IST |   |   |   |   |   |   |   |
| R/W              | RW0                   | R  | RW0 | R0 | RW0 | R0  |          |    |    |    |    | R0W0 | R0   |          |    |    |    |    |    |    |    |    | RW0 | RW0 |     |   |   |   |   |   |   |   |
| Initial value    | 0                     | 0  | 0   | 0  | 0   | 0   |          |    |    |    |    | 0    | 0    |          |    |    |    |    |    |    |    |    | 0   | 0   |     |   |   |   |   |   |   |   |

\*1 Reserved

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.
- Bit 24           TIM (Timeout)  
Indicates that an internal FIFO or Bus timeout has occurred. The TCS (Timeout Control/Status) register may be read to determine the cause of the timeout.
- Bit 26           SII (Serial Interface Interrupt)  
Indicates a serial interface write/read has completed.
- Bit 27           GI (GPIO Interrupt)  
Indicates that a GPIO input has changed state (0->1 or 1->0)

- Bit 28 BC (Burst Complete)  
Indicates that a burst has completed (as part of a Burst Control Unit transfer). Note that this bit is cleared by writing to the BST (Burst Status) register, not the IST.
- Bit 29 TC (Transfer Complete)  
Indicates that a transfer is complete (as controlled by the Burst Control Unit). Note that this bit is cleared by writing to the BST (Burst Status) register, not the IST.
- Bit 30 HF (HIF Fatal)  
Indicates that a fatal error occurred in a PCI transfer.
- Bit 31 AE (Address Error)  
Indicates that an invalid address was specified for an access (eg. Host Interface registers as a BCU source address).

**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   | IMASK                 |    |    |    |    |    | *1 | IMASK | Reserved |    |    |    |    |    | Resv | Reserved |    |    |    |    |    | IMASK | IMASK |   |   |   |   |   |   |   |   |   |
| R/W              | RW                    |    |    |    |    |    | R0 | RW    | R0       |    |    |    |    |    | R0W0 | R0       |    |    |    |    |    | RW    | RW    |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                     |    |    |    |    |    | 0  | 0     | 0        |    |    |    |    |    | 0    | 0        |    |    |    |    |    | 0     | 0     |   |   |   |   |   |   |   |   |   |

\*1 Reserved

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
- Bit 24 TIMM (Timeout Mask)  
Masks timeout interrupt.
- Bit 26 SIIM (Serial Interface Interrupt)  
Masks serial interface interrupt.
- Bit 27 GIM (GPIO Interrupt)  
Masks GPIO interrupt.
- Bit 28 BCM (Burst Complete)  
Masks Burst Complete interrupt.
- Bit 29 TCM (Transfer Complete)  
Masks Transfer complete interrupt.

- Bit 30 HFM (HIF Fatal)  
Masks HIF fatal interrupt.
- Bit 31 AEM (Address Error)  
Masks address error interrupt.

**SRST (Software ReSeT)**

|                  |                                   |   |   |   |   |   |   |      |
|------------------|-----------------------------------|---|---|---|---|---|---|------|
| Register address | HostBaseAddress + 2C <sub>H</sub> |   |   |   |   |   |   |      |
| 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.

**CCF (Change of Clock Frequency)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |          |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|----------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | HostBaseAddress + 0038 <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                                   |    |    |    |    |    |    |    |    |    |    |    |    |    | 10  | 01  | 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.

**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   |

Setting this register will move 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.

**IP (Interrupt Polarity)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |    |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|----|---|
| Register address | HostBaseAddress + 0070 <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                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | IP |   |
| R/W              | R0                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW |   |
| Initial value    | 0                                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | 0  |   |

In normal mode (with IP “0b”) the interrupt polarity is low (PCI standard). If an active high interrupt is required then this may be configured by setting this register to “1b”.

**OFU (Override FIFO Use)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |     |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|-----|---|
| Register address | HostBaseAddress + 007C <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                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | OFU |   |
| R/W              | R0                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW  |   |
| Initial value    | 0                                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | 0   |   |

In normal mode (with OFU “0b”) any write to the FIFO address will use the FIFO interface. Setting this bit to “1b” will override this and a standard bus access will be used. Under normal circumstances this register should be kept as “0b”.

**FRST (Firm ReSeT)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |      |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|------|---|
| Register address | HostBaseAddress + 00A0 <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                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | FRST |   |
| R/W              | R0                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | RW   |   |
| Initial value    | 0                                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   | 0    |   |

Writing a “1b” to this register will trigger a Firm Reset. This resets the complete device (as far as possible) including the PCI Interface.

**SRBS (Slave Burst Read Size)**



|                  |                         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |   |   |
|------------------|-------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|------|---|---|---|
| Register address | HostBaseAddress + 00A4H |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |   |   |
| 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                |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | SRBS |   |   |   |
| R/W              | R0                      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW   |   |   |   |
| Initial value    | 0                       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | 0    |   |   |   |

This register specifies the length of a burst read through the PCI Slave Interface as SRBS+1. By default this register is set to “000b” indicating a burst read length of 1 dword. The maximum setting is 7 (“111b”) and indicates a burst read length of 8 dwords.

**IOM (IO Mode)**

|                  |                         |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |     |     |     |     |     |    |   |   |
|------------------|-------------------------|----|-----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|-----|-----|-----|-----|-----|-----|----|---|---|
| Register address | HostBaseAddress + 00A8H |    |     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |     |     |     |     |     |     |    |   |   |
| 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   | Resv.                   |    | GIM |    |    |    |    |    |    |    |    |    |    |    |    |    | GD |    |    |    |    |    | SER | RGB | BEE | SBE | TCE | BCE | EEE |    |   |   |
| R/W              | R0                      |    | RW  |    |    |    |    |    |    |    |    |    |    |    |    |    | RW |    |    |    |    |    | RW  | RW  | RW  | RW  | RW  | RW  | RW  |    |   |   |
| Initial value    | 0                       |    | 0   |    |    |    |    |    |    |    |    |    |    |    |    |    | 0  |    |    |    |    |    | 0   | *1  | 0   | 0   | 0   | 0   | 0   | *2 |   |   |

\*1 – initial reset value specified by Burst Enable pin state at reset.

\*2 – initial reset value specified by Transfer Complete pin state at reset.

This register determines the function of those Coral LP pins under the control of the host interface. It also defines the direction (input/output) of any GPIO.

- Bit 0      EEE (EEPROM Enable)  
If set then the PCI EEPROM Configuration function is enabled. This field takes its reset value from the Transfer Complete pin at system reset. Note that if the RGB input is enabled then the EEPROM interface is disabled regardless of the value of this register. If this field is “0b” (and the RGB input is not enabled) then the EEPROM pins operate either as serial interface pins or GPIO as determined by the SER field.
- Bit 1      BCE (Burst Complete Enable)  
If set to “1b” then the BURSTC pin operates as Burst Complete. Otherwise if set to “0b” it operates as a GPIO. If the RGB input is enabled this field is ignored and the BURSTC pin operates as an RGB input pin.
- Bit 2      TCE (Transfer Complete Enable)  
If set to “1b” then the TRANSC pin operates as Transfer Complete. Otherwise if set to “0b” it operates as GPIO.
- Bit 3      SBE (Slave Busy Enable)  
If set to “1b” then the SBUSY pin operates as Slave Busy. Otherwise if set to “0b” it operates as a GPIO. If the RGB input is enabled this field is ignored and the SBUSY pin operates as an RGB input pin.
- Bit 4      BEE (Burst Enable Enable)  
If set to “1b” then the BURSTEN pin operates as Burst Enable. Otherwise if set to “0b” it operates as GPIO.
- Bit 5      RGB (RGB input enable)  
If set to “1b” then the RGB input is enabled. This field takes its reset value from the Burst Enable pin at system reset and overrides all other IO enable fields.
- Bit 6      SER (SERial Interface enable)  
If set to “1b” then the serial interface is enabled. This field is ignored if either the RGB input or EEPROM is enabled. For the serial interface strobe signal to be used the SBE field must also be clear (“0b”).

- Bit 15 to GD (GPIO Direction)  
Bit 7 Specifies the direction of pins acting as GPIO. If a bit is “0b” then the pin acts as an input. Otherwise if set to “1b” it operates as an output. The mapping to pins is:  
Bit 7: EDO  
Bit 8: EDI  
Bit 9: ECK  
Bit 10: ECS  
Bit 11: EE  
Bit 12: BURSTC  
Bit 13: TRANSC  
Bit 14: SBUSY  
Bit 15: BURSTEN
- Bit 29 to GIM (GPIO Interrupt Mask)  
Bit 16 Masks (enables) interrupt triggering on a GPIO pin by pin basis. If a bit is set to “1b” then a change in stage of that pin (0->1 or 1->0) can trigger an interrupt via the IST register. Otherwise if set to “0b” no interrupt will be triggered. Care should be taken to disable interrupts on pins not operating as GPIO inputs, otherwise unwanted interrupts may occur. The mapping to pins is:  
Bit 16: EDO  
Bit 17: EDI  
Bit 18: ECK  
Bit 19: ECS  
Bit 20: EE  
Bit 21: BURSTC  
Bit 22: TRANSC  
Bit 23: SBUSY  
Bit 24: BURSTEN  
Bit 25: GI1  
Bit 26: GI2  
Bit 27: GI3  
Bit 28: GI4  
Bit 29: GI5

**GD (GPIO Data)**

|                  |                                     |    |    |    |    |    |    |    |     |    |    |    |    |    |    |    |      |    |        |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|-----|----|----|----|----|----|----|----|------|----|--------|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | HostBaseAddress + 00AC <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                            |    |    |    |    |    |    |    | GWE |    |    |    |    |    |    |    | Resv |    | GD     |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | R0                                  |    |    |    |    |    |    |    | W   |    |    |    |    |    |    |    | R0   |    | RW     |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                   |    |    |    |    |    |    |    | 0   |    |    |    |    |    |    |    | 0    |    | 0 (*1) |    |    |    |   |   |   |   |   |   |   |   |   |   |

\*1 – initial value will be affected by state of GPIO pins

This register contains the GPIO read/write data field and the write mask when setting GPIO outputs.

Bit 13 to Bit 0 GD (GPIO Data)

This field is used for both reading the value of GPIO inputs and specifying the value for GPIO outputs. When writing to this field only those pins with the corresponding bit set in the GWE field will be changed. The bit positions refer to the following pins:

- Bit 0: EDO
- Bit 1: EDI
- Bit 2: ECK
- Bit 3: ECS
- Bit 4: EE
- Bit 5: BURSTC
- Bit 6: TRANSC
- Bit 7: SBUSY
- Bit 8: BURSTEN
- Bit 9: GI1
- Bit 10: GI2
- Bit 11: GI3
- Bit 12: GI4
- Bit 13: GI5

Bit 24 to Bit 16 GWE (GPIO Write Enable)

When writing values to the GPIO Outputs using the GD field, this field specifies those bits which are being written to. If a bit in this field is “1b” then the corresponding bit will be written to. Otherwise if a bit it “0b” the corresponding bit will remain unchanged. The bit positions refer to the following pins:

- Bit 16: EDO
- Bit 17: EDI
- Bit 18: ECK
- Bit 19: ECS
- Bit 20: EE
- Bit 21: BURSTC
- Bit 22: TRANSC
- Bit 23: SBUSY
- Bit 24: BURSTEN

**SIC (Serial Interface Control)**

|                  |                         |    |    |    |    |    |    |    |    |    |    |    |    |    |     |          |    |    |    |    |    |     |          |   |   |    |    |    |   |   |   |   |
|------------------|-------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----------|----|----|----|----|----|-----|----------|---|---|----|----|----|---|---|---|---|
| Register address | HostBaseAddress + 00B0H |    |    |    |    |    |    |    |    |    |    |    |    |    |     |          |    |    |    |    |    |     |          |   |   |    |    |    |   |   |   |   |
| 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                |    |    |    |    |    |    |    |    |    |    |    | 0  | 0  | CKD | Reserved |    |    |    |    |    | DOE | Reserved |   |   | SD | SP | SL |   |   |   |   |
| R/W              | R0                      |    |    |    |    |    |    |    |    |    |    |    | RW | RW | RW  | R0       |    |    |    |    |    | RW  | R0       |   |   | RW | RW | RW |   |   |   |   |
| Initial value    | 0                       |    |    |    |    |    |    |    |    |    |    |    | 0  | 0  | 0   | 0        |    |    |    |    |    | 0   | 0        |   |   | 0  | 0  | 0  |   |   |   |   |

This register provides control for the serial interface protocol and clock.

- Bit 0      SL (Strobe Length)  
 If set to "0b" then the strobe signal is only active for one cycle at the start of a transfer. Otherwise if set to "1b" it is active for the duration of the cycle. Note that this field may be overridden for a single transaction using the FS/FSL fields in the SID register.
- Bit 1      SP (Strobe Polarity)  
 If set to "0b" then strobe is active low. Otherwise if set to "1b" it is active high.
- Bit 2      SD (Strobe Disable)  
 If set to "1b" then the serial interface strobe is disabled. Note that this field may be overridden for a single transaction using the FS field in the SID register.
- Bit 8      DOE (Data Output Enable control)  
 If set to "0b" then the Data Out signal is driven permanently even when transactions are not in progress. If set to "1b" then the Data Out is driven only during active cycles.
- Bit 17 to  
 Bit 16      CKD (Clock Divisor)  
 This field specifies the serial interface clock divisor. The main system clock is divided down by one of the following factors:  
 00b: 16  
 01b: 32  
 10b: 64  
 11b: 128  
 Based on a 133MHz internal clock these yield frequencies of approximately 8.3MHz, 4.1MHz, 2.0 MHz and 1.0MHz respectively.
- Bit 18      CKG (Clock Gating)  
 When set to "1b" the serial interface clock is only active during active transfers. Otherwise if set to "0b" it is active continuously. Note that the CKP field specifies the inactive value when the clock is static.
- Bit 19      CKP (Clock Polarity)  
 When set to "0b" data/strobe are clocked out on a falling edge of the serial interface clock and data in is clocked in on the next falling edge. When clock gating is enabled (by setting the CKG field) the static level is low.  
 When set to "1b" data/strobe are clocked out on a rising edge of the serial interface clock and data in is clocked in on the next falling edge. When clock gating is enabled (by setting the CKG field) the static level is high.

**SID (Serial Interface Data)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |    |    |    |    |    |     |   |   |   |   |   |   |   |   |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----|----|----|----|----|-----|---|---|---|---|---|---|---|---|---|
| Register address | HostBaseAddress + 00B4 <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                            |    |    |    |    |    |    |    |    |    |    |    |    |    | FS | FS | TLS |    |    |    |    |    | RWD |   |   |   |   |   |   |   |   |   |
| R/W              | R0                                  |    |    |    |    |    |    |    |    |    |    |    |    |    | RW | RW | RW  |    |    |    |    |    | RW  |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                   |    |    |    |    |    |    |    |    |    |    |    |    |    | 0  | 0  | 0   |    |    |    |    |    | 0   |   |   |   |   |   |   |   |   |   |

This register is used to write/read serial interface data, enable a transfer and monitor a transfers progress.

- Bit 0 to Bit 7      RWD (Read/Write Data)  
When written to specifies the serial output data. When read it contains the serial interface input data. Note that data will be shifted out top bit (bit 7) first down to the bottom bit (bit 0) last. Read data will be shifted in to the bottom bit and shifted up by by each bit of the transfer. For transfer of length 8 this will yield consistent read/write data. For transfers of less than 8 bits then identical read and write data will appear different.
- Bit 15 to Bit 8      TLS (Transfer Length/Status)  
Specifies the length of a transfer and can be used to monitor its status. For each bit of a transfer this field is shifted up by one until it is "0000000b". For example, to specify a transfer of 8 bits "0000001b" should be written. To specify a transfer of 3 bits "0010000" should be written.
- Bit 16      FS (Force Strobe)  
For a single transfer this field can be used to override settings in the SIC register. If set to "1b" then a strobe will be done with a length specified in the FSL field.
- Bit 17      FSL (Force Strobe Length)  
For a single transfer if the FS field is set this field overrides the SL field in the SIC register and specifies the Strobe Length for the transfer. A value of "0b" specifies a strobe only for the first active cycle of the transfer. A value of "1b" specifies a strobe active for the whole transfer.

**CID (Chip ID register)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |           |    |    |    |    |    |           |    |   |   |   |   |   |   |   |   |   |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|-----------|----|----|----|----|----|-----------|----|---|---|---|---|---|---|---|---|---|---|
| Register address | HostBaseAddress + 00f0 <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 |    |    |    |    |    | 0000_0110 |    |   |   |   |   |   |   |   |   |   |   |

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  
0000\_0011    Reserved  
0000\_0100    Reserved for LB

0000\_0101 Reserved  
 0000\_0110 Reserved for LP (Coral LP value)  
 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

**BSA (Burst Source Address)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | HostBaseAddress + 8000 <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   | SA                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | RW                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

This register specifies the initial source address for a transfer controlled by the Burst Control Unit. Its interpretation (internal Coral/external PCI) will depend on the transfer mode specified in the BSR register.

**BDA (Burst Destination Address)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | HostBaseAddress + 8004 <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   | DA                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | RW                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

This register specifies the initial destination address for a transfer controlled by the Burst Control Unit. Its interpretation (internal Coral/external PCI) will depend on the transfer mode specified in the BSR register.

**BCR (Burst Control Register)**

|                  |                                     |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |   |   |   |   |   |   |   |   |   |   |  |  |
|------------------|-------------------------------------|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|---|---|---|---|---|---|---|---|---|---|--|--|
| Register address | HostBaseAddress + 8008 <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   | SL                                  | LN | AN | *1 | BSIZE |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | TSIZE |   |   |   |   |   |   |   |   |   |   |  |  |
| R/W              | RW                                  | RW | RW | R0 | RW    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW    |   |   |   |   |   |   |   |   |   |   |  |  |
| Initial value    | 0                                   | 0  | 0  | 0  | 0     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0     |   |   |   |   |   |   |   |   |   |   |  |  |

\*1 - Reserved

This register specifies the length and address manipulation performed for a transfer. It can also be used to start a transfer.

Bit 23 to 0 TSIZE  
 This field specifies the overall transfer length as a number of dwords. A transfer will be split up into a number of bursts whose length is specified by the BSIZE field.

- Bit 27 to 24    BSIZE (Burst Size)  
 This field specifies the length of a BCU controlled burst as a number of dwords. One or more bursts will make up an overall transfer. Note that if TSIZE is not an exact multiple of BSIZE the final burst of a transfer will be less than BSIZE.
  
- Bit 29            NSA (New Source Address)  
 If this bit is set to “1b” then after each burst the source address is incremented by the burst size. This means that a large continuous section of memory can be transferred. If this bit is “0b” then successive bursts will always be from the initial specified start address. This mode could be used if transferring data from a FIFO like interface.
  
- Bit 30            NDA (New Destination Address)  
 If this bit is set to “1b” then after each burst the destination address is incremented by the burst size. This means that data can be transferred into a large continuous section of memory. If this bit is “0b” then successive bursts will always be to the initial specified destination address. This mode should be used when transferring data to the FIFO.
  
- Bit 31            STRT (STaRT transfer)  
 When set to “1b” a transfer is started. Otherwise the transfer will wait until triggered wither through the Burst Enable Register (BER) or via the external burst enable signal.

**BSR (Burst Setup Register)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |       |     |     |       |      |   |   |   |   |   |   |   |   |   |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|-------|-----|-----|-------|------|---|---|---|---|---|---|---|---|---|---|
| Register address | HostBaseAddress + 800C <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                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | XCOR | IMODE | TCM | BCM | EXTEN | MODE |   |   |   |   |   |   |   |   |   |   |
| R/W              | RO                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW   | RW    | RW  | RW  | RW    | RW   |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0    | 0     | 0   | 0   | 0     | 0    |   |   |   |   |   |   |   |   |   |   |

This register specifies the type of a transfer (interpretation of the addresses) and specifies the setup of control signals/status bits.

- Bit 2 to 0        MODE (transfer MODE)  
 This field specifies the mode of the transfer and thus the interpretation of the source/destination addresses.  
 000b: Slave Mode PCI to Coral  
 001b: Slave Mode Coral to PCI  
 010b: Coral to Coral (internal transfer)  
 011b: Reserved  
 100b: PCI to Coral (PCI Master read)  
 101b: Coral to PCI (PCI Master write)  
 110b: PCI to PCI (PCI Master read/write external DMA transfer)  
 111b: Reserved  
 Refer to Chapter 3 for a detailed explanation of these modes.
  
- Bit 3            EXTEN (EXTernal ENable)  
 If set to “1b” then the external BURSTEN (Burst Enable) signal may be used to initiate and pause a transfer. Otherwise if set to “0b” the external BURSTEN signal is ignored.
  
- Bit 4            BCM (Burst Complete Mask)  
 If set to “1b” then the external BURSTC signal will be active. Otherwise if set to “0b” it will remain inactive low. Note that this bit does not affect the Burst Complete indication in the main interrupt status register (IST) or the triggering of the main external interrupt.
  
- Bit 5            TCM (Transfer Complete Mask)  
 If set to “1b” then the external TRANSC signal will be active. Otherwise if set to “0b” it will remain inactive low. Note that this bit does not affect the Transfer Complete indication in the main interrupt status register (IST) or the triggering of the main external interrupt.

- Bit 6 IMODE (Interrupt Mode)  
 This bit controls how the external BURSTC/TRANSC signals operate. If set to “0b” they are active high. Otherwise if set to “1b” they toggle at each change of state removing the need for the host to read/write the status register to clear them down.  
 Note that when using the Burst Complete/Transfer Complete indications via the main interrupt status register this field should always be “0b”.
- Bit 7 XCOR (*not* Clear On Read)  
 If set to “0b” then the Burst Complete/Transfer Complete fields in the Burst Status register are clear on read. Otherwise if set to “1b” they must be manually written.

**BER (Burst Enable Register)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |       |    |            |    |    |    |    |   |          |   |   |       |     |   |   |   |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-------|----|------------|----|----|----|----|---|----------|---|---|-------|-----|---|---|---|---|
| Register address | HostBaseAddress + 8010 <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                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    | ABORT | *1 | Reserved   |    |    |    |    |   | Reserved |   |   | EXTST | BEN |   |   |   |   |
| R/W              | R0                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    | W     | R0 | RX         |    |    |    |    |   | R0       |   |   | R     | RW  |   |   |   |   |
| Initial value    | 0                                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 0     | 0  | Don't Care |    |    |    |    |   | 0        |   |   | 0     | 0   |   |   |   |   |

\*1 - Reserved

This register can be used to enable/pause/abort a transfer. It can also be used to monitor the state of the external Burst Enable signal.

- Bit 0 BEN (Burst ENable)  
 When set to “1b” a transfer is enabled. This bit will also become set if the STRT bit in the BCR register is set. During a transfer this may be cleared to “0b” to pause/halt a transfer at the next boundary between bursts. Setting it back to “1b” will re-enable the transfer from the position it had reached.
- Bit 1 EXTST (External Status)  
 Provided the state of the external Burst Enable signal.
- Bit 16 ABORT  
 Under some circumstances clearing the BEN field may not halt a transfer. This will happen if the Burst Controller is waiting for an external PCI Master to take some action. In this case writing “1b” to the ABORT field will cancel the transfer. The transfer will not be able to be re-started.

**BST (Burst STatus)**

|                  |                                     |    |          |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|-------------------------------------|----|----------|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | HostBaseAddress + 8014 <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   | TC                                  | BC | Reserved |    |    |    |    |    |    |    |    |    |    |    |    |    | TCNT |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | R                                   | R  | R0       |    |    |    |    |    |    |    |    |    |    |    |    |    | R    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                   | 0  | 0        |    |    |    |    |    |    |    |    |    |    |    |    |    | 0    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

This register is used to monitor the state of the current transfer.

- Bit 23 to 0 TCNT (Transfer CouNT)  
 Gives the current transfer count as a number of dwords remaining to be transferred.
- Bit 30 BC (Burst Complete)  
 Indicates the state of a burst. Note that when in active high mode this field will remain high following a burst unless it is cleared either by a clear on read or by writing 0 to it.



Bit 31 TC (Transfer Complete)  
 Indicates the state of the current transfer. When set to "1b" the transfer is complete.

**BCB (Burst Controller Buffer)**

|                  |                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|-------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | HostBaseAddress + 8040 <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   | RWDATA * 8                          |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | RW                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                   |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

This buffer is used by the Burst Controller as a temporary store while executing transfers. The user should only need to access it when using modes "000b" and "001b" – the PCI slave modes. These can be used to transfer large quantities of data to/from the Coral LP in PCI Slave mode with automatic pre-fetch/write of data with address incrementing.

## 10.2.2 I<sup>2</sup>C Interface Registers

### BSR (Bus Status Register)

| Register address | I2C Base Address + 000h |     |    |     |     |     |     |     |
|------------------|-------------------------|-----|----|-----|-----|-----|-----|-----|
| Bit No           | 7                       | 6   | 5  | 4   | 3   | 2   | 1   | 0   |
| Bit field name   | BB                      | RSC | AL | LRB | TRX | AAS | GCA | FBT |
| R/W              | R                       | R   | R  | R   | R   | R   | R   | R   |
| Default          | 0                       | 0   | 0  | 0   | 0   | 0   | 0   | 0   |

All bits on this register are cleared while bit EN on CCR register is "0".

- Bit7** BB (Bus Busy)  
Indicate state of I2C-bus  
0: STOP condition was detected.  
1: START condition (The bus is in use.) was detected.
- Bit6** RSC (Repeated START Condition)  
Indicate repeated START condition  
This bit is cleared by writing "0" to INT bit, the case of not addressed in a slave mode, the detection of START condition under bus stop, and the detection of STOP condition.  
0: Repeated START condition was not detected.  
1: START condition was detected again while the bus was in use.
- Bit5** AL(Arbitration Lost)  
Detect Arbitration lost  
This bit is cleared by writing "0" to INT bit.  
0: Arbitration lost was not detected.  
1: Arbitration occurred during master transmission, or "1" writing was performed to MSS bit while other systems were using the bus.
- Bit4** LRB (Last Received Bit)  
Store Acknowledge  
This bit is cleared by detection of START condition or STOP condition.
- Bit3** TRX (Transmit / Receive)  
Indicate data receipt and data transmission.  
0: receipt  
1: transmission
- Bit2** AAS (Address As Slave)  
Detect addressing  
This bit is cleared by detection of START condition or STOP condition.  
0: Addressing was not performed in a slave mode.  
1: Addressing was performed in a slave mode.
- Bit1** GCA (General Call Address)  
Detect general call address (00h)  
This bit is cleared by detection of START condition or STOP condition.  
0: General call address was not received in a slave mode.  
1: General call address was received in a slave mode.
- Bit0** FBT (First Byte Transfer)  
Detect the 1st byte  
Even if this bit is set to "1" by detection of START condition, it is cleared by writing "0" on INT bit or by not being addressed in a slave mode.  
0: Received data is not the 1st byte.  
1: Received data is the 1st byte (address data).

**BCR (Bus Control Register)**

|                  |                          |      |       |     |     |      |      |     |
|------------------|--------------------------|------|-------|-----|-----|------|------|-----|
| Register address | I2C Base Address + 0004h |      |       |     |     |      |      |     |
| Bit No           | 7                        | 6    | 5     | 4   | 3   | 2    | 1    | 0   |
| Bit field name   | BER                      | BEIE | SCC   | MSS | ACK | GCAA | INTE | INT |
| R/W              | R/W0                     | R/W  | R0/W1 | R/W | R/W | R/W  | R/W  | R/W |
| Default          | 0                        | 0    | 0     | 0   | 0   | 0    | 0    | 0   |

- Bit7           BER (Bus Error)  
 Flag bit for request of bus error interruption  
 When this bit is set, EN bit on CCR register will be cleared, this module will be in a stop state and data transfer will be discontinued.  
write case  
 0: A request of buss error interruption is cleared.  
 1: Don't care.

read case  
 0: A bus error was not detected.  
 1: Undefined START condition or STOP condition was detected while data transfer.
- Bit6           BEIE (Bus Error Interruption Enable)  
 Permit bus error interruption  
 When both this bit and BER bit are "1", the interruption is generated.  
 0: Prohibition of bus error interruption  
 1: Permission of bus error interruption
- Bit5           SCC (Start Condition Continue)  
 Generate START condition  
write case  
 0: Don't care.  
 1: START condition is generated again at the time of master transmission.
- Bit4           MSS (Master Slave Select)  
 Select master / slave mode  
 When arbitration lost is generated in master transmission, this bit is cleared and this module becomes a slave mode.  
 0: This module becomes a slave mode after generating STOP condition and completing transfer.  
 1: This module becomes a master mode, generates START condition and starts transfer.
- Bit3           ACK (ACKnowledge)  
 Permit generation of acknowledge at the time of data reception  
 This bit becomes invalid at the time of address data reception in a slave mode.  
 0: Acknowledge is not generated.  
 1: Acknowledge is generated.
- Bit2           GCAA(General Call Address Acknowledge)  
 Permit generation of acknowledge at the time of general call address reception  
 0: Acknowledge is not generated.  
 1: Acknowledge is generated.
- Bit1           INTE (INTerrupt Enable)  
 Permit interruption  
 When this bit is "1" interruption is generated if INT bit is "1".  
 0: Prohibition of interrupt  
 1: Permission of interrupt
- Bit0           INT (INTerrupt)  
 Flag bit for request of interruption for transfer end  
 When this bit is "1" SCL line is maintained at "L" level. If this bit is cleared by being

written "0", SCL line is released and the following byte transfer is started. Moreover, it is reset to "0" by generating of START condition or STOP condition at the time of a master.

write case

0: The flag is cleared.

1: Don't care.

read case

0: The transfer is not ended.

1: It is set when 1 byte transfer including the acknowledge bit is completed and it corresponds to the following conditions.

- It is a bus master.

- It is an addressed slave.

- It was going to generate START condition while other systems by which arbitration lost happened used the bus.

**Competition of SCC, MSS and INT bit**

Competition of the following byte transfer, generation of START condition and generation of STOP condition happens by the simultaneous writing of SCC, MSS and INT bit. The priority at this case is as follows.

1) The following byte transfer and generation of STOP condition

If "0" is written to INT bit and "0" is written to MSS bit, priority will be given to "0" writing to MSS bit and STOP condition will be generated.

2) The following byte transfer and generation of START condition

If "0" is written to INT bit and "1" is written to SCC bit, priority will be given to "1" writing to SCC bit and START condition will be generated.

3) Generation of START condition and STOP condition

The simultaneous writing of "1" to SCC bit and "0" to MSS bit is prohibition.

**CCR (Clock Control Register)**

|                  |                          |     |     |     |     |     |     |     |
|------------------|--------------------------|-----|-----|-----|-----|-----|-----|-----|
| Register address | I2C Base Address + 0008h |     |     |     |     |     |     |     |
| Bit No           | 7                        | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| Bit field name   | -                        | HSM | EN  | CS4 | CS3 | CS2 | CS1 | CS0 |
| R/W              | R1                       | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Default          | 1                        | 0   | 0   | -   | -   | -   | -   | -   |

- Bit7 Nonuse  
"1" is always read at read.
- Bit6 HSM (High Speed Mode)  
Select standard-mode / high-speed-mode  
0: Standard-mode  
1: High-speed-mode
- Bit5 EN (Enable)  
Permission of operation  
When this bit is "0", each bit of BSR and BCR register (except BER and BEIE bit) is cleared. This bit is cleared when BER bit is set.  
0: Prohibition of operation  
1: Permission of operation
- Bit4 CS4 - 0 (Clock Period Select4 - 0)  
Set up the frequency of a serial transfer clock  
Frequency fscl of a serial transfer clock is shown as the following formula.  
Please set up fscl not to exceed the value shown below at the time of master operation.  
standard-mode: 100KHz  
high-speed-mode: 400KHz

**standard-mode**

$$fscl = \frac{A}{(2 \times m)+2}$$

**high-speed-mode**

$$fscl = \frac{A}{int(1.5 \times m)+2}$$

A: I2C system clock = 16.6MHz

<Notes>

+2 cycles are minimum overhead to confirm that the output level of SCL terminal changed. When the delay of the positive edge of SCL terminal is large or when the clock is extended by the slave device, it becomes larger than this value.

The value of m becomes like the following page to the value of CS 4-0.

| CS4 | CS3 | CS2 | CS1 | CS0 | m        |            |
|-----|-----|-----|-----|-----|----------|------------|
|     |     |     |     |     | standard | high-speed |
| 0   | 0   | 0   | 0   | 0   | 65       | inhibited  |
| 0   | 0   | 0   | 0   | 1   | 66       | inhibited  |
| 0   | 0   | 0   | 1   | 0   | 67       | inhibited  |
| 0   | 0   | 0   | 1   | 1   | 68       | inhibited  |
| 0   | 0   | 1   | 0   | 0   | 69       | inhibited  |
| 0   | 0   | 1   | 0   | 1   | 70       | inhibited  |
| 0   | 0   | 1   | 1   | 0   | 71       | inhibited  |
| 0   | 0   | 1   | 1   | 1   | 72       | inhibited  |
| 0   | 1   | 0   | 0   | 0   | 73       | 9          |
| 0   | 1   | 0   | 0   | 1   | 74       | 10         |
| 0   | 1   | 0   | 1   | 0   | 75       | 11         |
| 0   | 1   | 0   | 1   | 1   | 76       | 12         |
| 0   | 1   | 1   | 0   | 0   | 77       | 13         |
| 0   | 1   | 1   | 0   | 1   | 78       | 14         |
| 0   | 1   | 1   | 1   | 0   | 79       | 15         |
| 0   | 1   | 1   | 1   | 1   | 80       | 16         |
| 1   | 0   | 0   | 0   | 0   | 81       | 17         |
| 1   | 0   | 0   | 0   | 1   | 82       | 18         |
| 1   | 0   | 0   | 1   | 0   | 83       | 19         |
| 1   | 0   | 0   | 1   | 1   | 84       | 20         |
| 1   | 0   | 1   | 0   | 0   | 85       | 21         |
| 1   | 0   | 1   | 0   | 1   | 86       | 22         |
| 1   | 0   | 1   | 1   | 0   | 87       | 23         |
| 1   | 0   | 1   | 1   | 1   | 88       | 24         |
| 1   | 1   | 0   | 0   | 0   | 89       | 25         |
| 1   | 1   | 0   | 0   | 1   | 90       | 26         |
| 1   | 1   | 0   | 1   | 0   | 91       | 27         |
| 1   | 1   | 0   | 1   | 1   | 92       | 28         |
| 1   | 1   | 1   | 0   | 0   | 93       | 29         |
| 1   | 1   | 1   | 0   | 1   | 94       | 30         |
| 1   | 1   | 1   | 1   | 0   | 95       | 31         |
| 1   | 1   | 1   | 1   | 1   | 96       | 32         |

**Address Register(ADR)**

|                  |                          |     |     |     |     |     |     |     |
|------------------|--------------------------|-----|-----|-----|-----|-----|-----|-----|
| Register address | I2C Base Address + 000Ch |     |     |     |     |     |     |     |
| Bit No           | 7                        | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| Bit field name   | -                        | A6  | A5  | A4  | A3  | A2  | A1  | A0  |
| R/W              | R1                       | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Default          | 1                        | -   | -   | -   | -   | -   | -   | -   |

Bit7            Nonuse  
                  "1" is always read at read.

Bit6 - 0        A6 - 0 (Address6 - 0)  
                  Store slave address  
                  In a slave mode it is compared with DAR register after address data reception, and  
                  when in agreement, acknowledge is transmitted to a master.

**Data Register(DAR)**

|                  |                          |     |     |     |     |     |     |     |
|------------------|--------------------------|-----|-----|-----|-----|-----|-----|-----|
| Register address | I2C Base Address + 0010h |     |     |     |     |     |     |     |
| Bit No           | 7                        | 6   | 5   | 4   | 3   | 2   | 1   | 0   |
| Bit field name   | D7                       | D6  | D5  | D4  | D3  | D2  | D1  | D0  |
| R/W              | R/W                      | R/W | R/W | R/W | R/W | R/W | R/W | R/W |
| Default          | -                        | -   | -   | -   | -   | -   | -   | -   |

Bit7 - 0        D7 - 0 (Data7 - 0)  
                  Store serial data  
                  This is a data register for serial data transfer. The data is transferred from MSB. At  
                  the time of data reception (TRX=0) the data output is set to "1".  
                  The writing side of this register is a double buffer. When the bus is in use (BB=1),  
                  the write data is loaded to the register for serial transfer for every transfer. At the  
                  time of read-out, the receiving data is effective only when INT bit is set because  
                  the register for serial transfer is read directly at this time.

### 10.2.3 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<br>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   | 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

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   |
|               | Other than the above | Setting disabled   |
| <br>          |                      |  |
| 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  |
| <br>          |                      |  |
| 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   |
| <br>          |                      |  |
| 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  |
| <br>          |                      |  |
| 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.<br>"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  |

### 10.2.4 Display control register

#### DCM (Display Control 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 | Reserved | SC    |    |    |   | EEQ | ODE | Reserved | Reserved | SF | ESY | SYNC |   |   |
| R/W              | RW  | RW0      | RX       | RW    |    |    |   | RW  | RW  | RX       | RX       | RW | RW  | RW   |   |   |
| Initial value    | 0   | 0        | X        | 11110 |    |    |   | 0   | 0   | X        | X        | 0  | 0   | 00   |   |   |

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  
       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 output format)  
 Sets format of synchronization (VSYNC, HSYNC) signals  
 0:    Negative logic output  
 1:    Positive logic output

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 + 102H |          |    |    |    |    |   |   |   |   |   |     |     |     |     |     |     |
| 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;  $2 \times$  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.



**L0M (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 |    |    |    | CW         |    |    |    |    |    |    |    | 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 L0W (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

**L0EM (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 |    |    |    | LOPB |    |    |    | 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

**L0OA (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                             |    |    |    | L0OA       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |    |   |   |   |
| R/W              | R0                                   |    |    |    | RW         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |    |   |   |   |
| Initial value    | 0                                    |    |    |    | Don't care |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | R0 |   |   |   |

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.

**L0DA (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                             |    |    |    | L0DA       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 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.

**L0DX (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                             |    |    |    | L0DX       |    |   |   |   |   |   |   |   |   |   |   |
| 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.

**L0DY (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                             |    |    |    | L0DY       |    |   |   |   |   |   |   |   |   |   |   |
| 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.



**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   | L0EC                                  |    | Reserved |    |    |    |    | LOPB |    |    | Reserved |    |    |    |    |    |    |    |    |    |    |    |     |   |   |   |   |   |
| R/W              | RW                                    |    | R0       |    |    |    |    | RW   |    |    | R0       |    |    |    |    |    |    |    |    |    |    |    |     |   |   |   |   |   |
| Initial value    |                                       |    | 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 LOC  
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  |    |    |    |    |    |   |   | LOWX       |   |   |   |   |   |   |   |
| 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  |    |    |    |    |    |   |   | LOWY       |   |   |   |   |   |   |   |
| 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  |    |    |    | LOWW       |    |   |   |   |   |   |   |   |   |   |   |
| 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  |    |    |    | LOWH       |    |   |   |   |   |   |   |   |   |   |   |
| 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 + 40H |       |    |          |    |    |    |            |    |    |    |    |    |    |    |          |    |    |    |            |    |   |   |   |   |   |   |   |   |   |   |  |
| 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        |    |    |    | 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 + 130 <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   | L2EC                                  |    | Reserved |    |    |    |    |    | L2PB |    |    |    | Reserved |    |    |    |    |    |    |    |    |    | L2OM  | LOWP |   |   |   |   |
| 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         |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | R0 |   |   |   |   |   |   |   |
| 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         |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | R0 |   |   |   |   |   |   |   |
| 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 + 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                              |    |    |    | 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 + 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                              |    |    |    | 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 + 13C <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    |     |   |   |   |   |   |

- 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 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |    |   |   |   |

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 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |    |   |   |   |

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 + 140 <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 + 142 <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 + 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                              |    |    |    | 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 + 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                              |    |    |    | 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 |    |    |    |    |    | L4PB |    |    |    |    |    | Reserved |    |    |    |    |    | L4OM | L4WP |     |   |   |   |   |   |
| R/W              | RW                                    |    | R0       |    |    |    |    |    | RW   |    |    |    |    |    | R0       |    |    |    |    |    | RW   | RW   |     |   |   |   |   |   |
| Initial value    | 00                                    |    | 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         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | R0   |   |   |   |
| Initial value    | 0                                    |    |    |    | Don't care |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | 0000 |   |   |   |

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         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | R0   |   |   |   |
| Initial value    | 0                                    |    |    |    | Don't care |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | 0000 |   |   |   |

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 + 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   | L5EC                                  |    | Reserved |    |    |    |    |    | L5PB |    |    |    |    |    | Reserved |    |    |    |    |    | L5OM | L5WP |     |   |   |   |   |   |
| R/W              | RW                                    |    | R0       |    |    |    |    |    | RW   |    |    |    |    |    | R0       |    |    |    |    |    | RW   | RW   |     |   |   |   |   |   |
| Initial value    | 00                                    |    | 0        |    |    |    |    |    | 0    |    |    |    |    |    | 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 23 to 20    L5PB (L5 layer Palette Base)  
 Shows the value added to the index when subtracting palette of L5 layer. 16 times of setting value is added.
  
- 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                             |    |    |    | BROA0      |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |      |   |   |   |
| 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         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | R0   |   |   |   |
| Initial value    | 0                                    |    |    |    | Don't care |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | 0000 |   |   |   |

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         |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | R0   |   |   |   |
| Initial value    | 0                                    |    |    |    | Don't care |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | 0000 |   |   |   |

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 + 180H |    |    |       |      |     |    |    |      |    |     |    |      |    |    |     |      |    |    |    |      |   |   |   |      |     |   |   |   |   |     |  |  |  |
| 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  |   |   |   |   |     |  |  |  |
| 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              |                                      |    |    |    |       |    |    |    |      |      |      |      |          |    |    |    |   |   |   |   |      |   |   |   |   |   |  |  |
| Initial value    |                                      |    |    |    |       |    |    |    |      |      |      |      |          |    |    |    |   |   |   |   |      |   |   |   |   |   |  |  |

This register specifies the blend parameters for the L0 layer. This register corresponds to BRATIO or BMODE for previous products.

- Bit 7 to 0      LOBR (L0 layer Blend Ratio)  
Sets the blend ratio. Basically, the blend ratio is setting value/256.
  
- Bit 13          LOBP (L0 layer Blend Plane)  
Specifies that the L5 layer is the blend plane.  
0      Value of LOBR used as blend ratio  
1      Pixel of L5 layer used as blend ratio
  
- Bit 14          LOBI (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          LOBS (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          LOBE (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 LOBE, 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              |                                       |    |    |    |       |    |    |    |    |    |      |      |      |      |          |    |   |   |   |      |   |   |   |   |   |   |
| Initial value    |                                       |    |    |    |       |    |    |    |    |    |      |      |      |      |          |    |   |   |   |      |   |   |   |   |   |   |

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              |                                       |    |    |    |       |    |    |    |      |      |      |      |          |    |    |    |   |   |   |   |      |   |   |   |   |   |  |  |
| Initial value    |                                       |    |    |    |       |    |    |    |      |      |      |      |          |    |    |    |   |   |   |   |      |   |   |   |   |   |  |  |

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              |                                       |    |    |    |       |    |    |    |      |      |      |      |          |    |    |    |   |   |   |   |      |   |   |   |   |   |  |  |
| Initial value    |                                       |    |    |    |       |    |    |    |      |      |      |      |          |    |    |    |   |   |   |   |      |   |   |   |   |   |  |  |

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              |                                       |    |    |    |       |    |    |    |      |      |      |      |          |    |    |    |   |   |   |   |      |   |   |   |   |   |  |  |
| Initial value    |                                       |    |    |    |       |    |    |    |      |      |      |      |          |    |    |    |   |   |   |   |      |   |   |   |   |   |  |  |

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                                    | Don't care |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

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                                    | Don't care |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

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                                    | Don't care |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

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 |    |    |     |    | L0ETC |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| R/W              | RW                                    | R0       |    |    |     |    | 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    |                                       |          |    |    |     |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |

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    |                                       |          |    |    |     |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |  |  |  |

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 L3EZT = 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        L3EZT (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 L4EZT = 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        L4EZT (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

**LOPAL0-255 (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   |    |    |    |    |    |    |    | RW         |    |    |    |    |    |    |    | RW |    |    |    |    |    |   |   | RW         |   |   |   |   |   |   |   |
| Initial value    | 0000000  |    |    |    |    |    |    |    | Don't care |    |    |    |    |    |    |    | 00 |    |    |    |    |    |   |   | Don't care |   |   |   |   |   |   |   |

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 CPALn 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         | 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 |   |   |   |   |   |   |   |            |  |  |  |  |  |  |  |    |  |  |  |  |  |  |  |
| Initial value    | Don't care   |    |    |    |    |    |    |    | 0000000 |    |    |    |    |    |    |    | 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         | 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

### 10.2.5 Video capture registers

#### VCM (Video Capture Mode)

|                  |                                       |     |          |    |    |    |          |    |    |    |          |    |    |    |    |    |    |    |    |    |    |    |   |   |    |     |   |   |   |   |   |   |
|------------------|---------------------------------------|-----|----------|----|----|----|----------|----|----|----|----------|----|----|----|----|----|----|----|----|----|----|----|---|---|----|-----|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 00 <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   | VIE                                   | VIS | Reserved |    |    | CM | Reserved |    |    | VI | Reserved |    |    |    |    |    |    |    |    |    |    |    |   |   | VS | Rsv |   |   |   |   |   |   |
| R/W              | RW                                    | RW  | RX       |    |    | RW | RX       |    |    | RW | RX       |    |    |    |    |    |    |    |    |    |    |    |   |   | RW | RX  |   |   |   |   |   |   |
| Initial value    | 0                                     | X   |          |    | 00 | X  |          |    | 0  | X  |          |    |    |    |    |    |    |    |    |    |    |    |   | 0 | X  |     |   |   |   |   |   |   |

This register sets the video capture mode.

- Bit 31            VIE (Video Input Enable)

Enables video capture function

0:    Does not capture video

1:    Captures video
- Bit 30            VIS (Video Input Select)

0    RBT656

1    RGB666
- Bit 25 to 24    CM (Capture Mode)

Sets video capture mode

To capture vides, set these bits to "11".

00:   Initial value

01:   Reserved

10:   Reserved

11:   Capture
- Bit 20            VI (Vertical Interpolation)

Sets whether to perform vertical interpolation

0:    Performs vertical interpolation

          The graphics are enlarged vertically by two times

1:    Does not perform vertical interpolation
- Bit 1            VS (Video Select)

Selects NTSC or PAL

0:    NTSC

1:    PAL

**CSC (Capture Scale)**

|                  |                                       |    |    |    |    |             |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |             |   |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|-------------|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|-------------|---|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 04 <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   | VSCI                                  |    |    |    |    | VSCF        |    |    |    |    |    |    |    |    |    |    | HSCI  |    |    |    |    | HSCF        |   |   |   |   |   |   |   |   |   |   |
| R/W              | RW                                    |    |    |    |    | RW          |    |    |    |    |    |    |    |    |    |    | RW    |    |    |    |    | RW          |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 00001                                 |    |    |    |    | 00000000000 |    |    |    |    |    |    |    |    |    |    | 00001 |    |    |    |    | 00000000000 |   |   |   |   |   |   |   |   |   |   |

This register sets the video capture enlargement/reduction ratio.

- Bit 31 to 27 VSCI (Vertical SCAle Integer)  
Sets integer part of vertical enlargement/reduction ratio
- Bit 26 to 16 VSCF (Vertical Scale Fraction)  
Sets fraction part of vertical enlargement/reduction ratio
- Bit 15 to 11 HSCI (Horizontal SCAle Integer)  
Sets integer part of horizontal enlargement/reduction ratio
- Bit 10 to 0 HSCF (Horizontal SCAle Fraction)  
Sets fraction part of horizontal enlargement/reduction ratio

Note :

Simultaneous upscaling and downscaling is not possible (eg HSCALE=0x1000,VSCALE=0x0600).  
No scaling (HSCALE=0x0800, VSCALE=0x800) is the default setting.

**VCS (Video Capture Status)**

|                  |                                       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |       |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|-------|---|---|---|
| Register address | CaputureBaseAddress + 08 <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                              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | CE    |   |   |   |
| R/W              | RX                                    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | RW    |   |   |   |
| Initial value    | Don't care                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   | 00000 |   |   |   |

This register indicates the ITU-RBT656 SAV and EAV status.

To detect error codes, set NTSC/PAL in the VS bit of VCM. If NTSC is set, reference the number of data in the capture data count register (CDCN). If PAL is set, reference the number of data in the capture data counter register (CDCP). If the reference data does not match the stream data , or undefined Fourth word of SAV/EAV codes are detected, bits 4 to 0 of the video capture status register (VCS) will be values as follows.

Bits 4-0 CE (Capture Error)

Indicates error occurred during video capture

|      |   |          |
|------|---|----------|
| Bit4 | 1 : RBT.656 H code error (End)            | 0 : true |
| Bit3 | 1 : RBT.656 H code error (Start)          | 0 : true |
| Bit2 | 1 : RBT.656 undefined error (Code Bit7-0) | 0 : true |
| Bit1 | 1 : RBT.656 undefined error (Code Bit7-4) | 0 : true |
| Bit0 | 1 : RBT.656 undefined error (Code Bit7)   | 0 : true |

**CBM (vide Capture Buffer Mode)**

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

Bit 23 to 16    CBW (Capture Buffer memory Width)  
 Sets memory width (stride) of capture buffer in 64 bytes

Bit 31            OO (Odd Only mode)  
 Specifies whether to capture odd fields only  
 0:    Normal mode  
 1:    Odd only mode

**CBOA (video Capture Buffer Origin Address)**

|                  |                                       |    |    |    |    |    |    |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 14 <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                              |    |    |    |    |    |    |    | CBOA       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |   |   |   |   |   |   |   |
| R/W              | RX                                    |    |    |    |    |    |    |    | RW         |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | R0 |   |   |   |   |   |   |   |
| Initial value    | Don't care                            |    |    |    |    |    |    |    | Don't care |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | 0  |   |   |   |   |   |   |   |

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

**CBLA (video Capture Buffer Limit Address)**

|                  |                                       |    |    |    |    |    |    |    |            |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|----|----|------------|----|----|----|----|----|----|----|----|----|----|----|----|----|---|---|----|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 18 <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                              |    |    |    |    |    |    |    | CBLA       |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |    |   |   |   |   |   |   |   |
| R/W              | RX                                    |    |    |    |    |    |    |    | RW         |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | R0 |   |   |   |   |   |   |   |
| Initial value    | Don't care                            |    |    |    |    |    |    |    | Don't care |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   | 0  |   |   |   |   |   |   |   |

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

CBLA must be larger than CBOA.

**CIHSTR (Capture Image Horizontal STaRt)**

|                  |                                       |    |    |    |    |    |            |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|------------|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 1C <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                              |    |    |    |    |    | CIHSTR     |   |   |   |   |   |   |   |   |   |
| R/W              | RX                                    |    |    |    |    |    | RW         |   |   |   |   |   |   |   |   |   |
| Initial value    | Don't care                            |    |    |    |    |    | Don't care |   |   |   |   |   |   |   |   |   |

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

**CIVSTR (Capture Image Vertical STaRt)**

|                  |                                       |    |    |    |    |    |            |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|------------|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 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                              |    |    |    |    |    | CIVSTR     |   |   |   |   |   |   |   |   |   |
| R/W              | RX                                    |    |    |    |    |    | RW         |   |   |   |   |   |   |   |   |   |
| Initial value    | Don't care                            |    |    |    |    |    | Don't care |   |   |   |   |   |   |   |   |   |

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

**CIHEND (Capture Image Horizontal END)**

|                  |                                       |    |    |    |    |    |            |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|------------|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 20 <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                              |    |    |    |    |    | CIHEND     |   |   |   |   |   |   |   |   |   |
| R/W              | RX                                    |    |    |    |    |    | RW         |   |   |   |   |   |   |   |   |   |
| Initial value    | Don't care                            |    |    |    |    |    | Don't care |   |   |   |   |   |   |   |   |   |

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

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

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

**CIVEND (Capture Image Vertical END)**

|                  |                                       |    |    |    |    |    |            |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|------------|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 22 <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                              |    |    |    |    |    | CIVEND     |   |   |   |   |   |   |   |   |   |
| R/W              | RX                                    |    |    |    |    |    | RW         |   |   |   |   |   |   |   |   |   |
| Initial value    | Don't care                            |    |    |    |    |    | Don't care |   |   |   |   |   |   |   |   |   |

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

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

**CHP (Capture Horizontal Pixel)**

|                  |                                       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |                                      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|----|--------------------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 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                              |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | CHP                                  |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | RX                                    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RW                                   |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | X                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 168 <sub>H</sub> (360 <sub>D</sub> ) |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

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

**CVP (Capture Vertical Pixel)**

|                  |                                       |    |    |    |    |    |    |    |                                      |    |    |    |    |    |    |    |          |    |    |    |    |    |   |   |                                      |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|----|----|--------------------------------------|----|----|----|----|----|----|----|----------|----|----|----|----|----|---|---|--------------------------------------|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 2C <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                              |    |    |    |    |    |    |    | CVPP                                 |    |    |    |    |    |    |    | Reserved |    |    |    |    |    |   |   | CVPN                                 |   |   |   |   |   |   |   |
| R/W              | RX                                    |    |    |    |    |    |    |    | RW                                   |    |    |    |    |    |    |    | RX       |    |    |    |    |    |   |   | RW                                   |   |   |   |   |   |   |   |
| Initial value    | X                                     |    |    |    |    |    |    |    | 271 <sub>H</sub> (625 <sub>D</sub> ) |    |    |    |    |    |    |    | X        |    |    |    |    |    |   |   | 20D <sub>H</sub> (525 <sub>D</sub> ) |   |   |   |   |   |   |   |

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

Bit 25 to 16 CVPP (Capture Vertical Pixel for PAL)  
Set count of vertical pixels of output image in PAL format used

Bit 9 to 0 CVPN (Capture Vertical Pixel for NTSC)  
Set count of vertical pixels of output image in NTSC format used

**CLPF (Capture Low Pass Filter)**

|                  |                                       |    |    |    |       |    |    |    |         |    |    |    |       |    |    |    |         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|-------|----|----|----|---------|----|----|----|-------|----|----|----|---------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 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   | Reserve                               |    |    |    | CVLPF |    |    |    | Reserve |    |    |    | CHLPF |    |    |    | Reserve |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | R0                                    |    |    |    | R/W   |    |    |    | R0      |    |    |    | R/W   |    |    |    | R0      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                     |    |    |    | 0     |    |    |    | 0       |    |    |    | 0     |    |    |    | 0       |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

This register sets the Low Pass Filter Coefficient. It specifies independently in 2-bit coefficient code with a luminance signal (Y) and a color-difference signal (C). A coefficient is a right-and-left symmetrical coefficient.

A Vertical low path filter consists of FIR filters of three taps. A coefficient is specified in the following register.

Bit 27 to 26 CVLPF\_Y (Capture Vertical LPF coefficient Y)

Sets Y part of vertical LPF coefficient code

| CVLPF_Y | K0      | K1    | K2   |
|---------|---------|-------|------|
| 2'b00   | 0       | 1     | 0    |
| 2'b01   | 1/4     | 2/4   | 1/4  |
| 2'b10   | 3/16    | 10/16 | 3/16 |
| 2'b11   | Reserve |       |      |

Bit 25 to 24 CVLPF\_C (Capture Vertical LPF coefficient C)

Sets C part of vertical LPF coefficient code

| CVLPF_C | K0      | K1    | K2   |
|---------|---------|-------|------|
| 2'b00   | 0       | 1     | 0    |
| 2'b01   | 1/4     | 2/4   | 1/4  |
| 2'b10   | 3/16    | 10/16 | 3/16 |
| 2'b11   | Reserve |       |      |

A horizontal low path filter consists of FIR filters of five taps. A coefficient is specified in the following register.

Bit 19 to 18 CHLPF\_YI (Capture Horizontal LPF coefficient Y)

Sets Y part of horizontal coefficient code

| CHLPF_Y | K0   | K1   | K2    | K3    | K4   |
|---------|------|------|-------|-------|------|
| 2'b00   | 0    | 0    | 1     | 0     | 0    |
| 2'b01   | 0    | 1/4  | 2/4   | 1/4   | 0    |
| 2'b10   | 0    | 3/16 | 10/16 | 3/16  | 0    |
| 2'b11   | 3/32 | 8/32 | 10/32 | 10/32 | 3/32 |

Bit 17 to 16 CHLPF\_C (Capture Horizontal LPF coefficient C)

Sets C part of horizontal coefficient code

| CHLPF_C | K0   | K1   | K2    | K3    | K4   |
|---------|------|------|-------|-------|------|
| 2'b00   | 0    | 0    | 1     | 0     | 0    |
| 2'b01   | 0    | 1/4  | 2/4   | 1/4   | 0    |
| 2'b10   | 0    | 3/16 | 10/16 | 3/16  | 0    |
| 2'b11   | 3/32 | 8/32 | 10/32 | 10/32 | 3/32 |

LPF will be turned off if coefficient code 2'b00 are set up.



**CDCN (Capture Data Count for NTSC)**

|                  |   |    |    |    |                                      |    |    |    |    |    |    |    |          |    |    |    |                         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|---|----|----|----|--------------------------------------|----|----|----|----|----|----|----|----------|----|----|----|-------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 4000 <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                                |    |    |    | BDCN                                 |    |    |    |    |    |    |    | Reserved |    |    |    | VDCN                    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | RX                                      |    |    |    | RW                                   |    |    |    |    |    |    |    | RX       |    |    |    | RW                      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | X                                       |    |    |    | 10f <sub>H</sub> (271 <sub>D</sub> ) |    |    |    |    |    |    |    | X        |    |    |    | 5A3 <sub>H</sub> (1443) |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

This register sets the count of data of the input video stream in NTSC format.

Bit 25 to 16 BDCN (Blanking Data Count for NTSC)  
Sets count of data processed during blanking period in NTSC format

Bit 10 to 0 VDCN (Valid Data Count for NTSC)  
Sets count of data processed during valid period in NTSC format

**CDCP (Capture Data Count for PAL)**

|                  |   |    |    |    |                                      |    |    |    |    |    |    |    |          |    |    |    |                         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|---|----|----|----|--------------------------------------|----|----|----|----|----|----|----|----------|----|----|----|-------------------------|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 4004 <sub>H</sub> |    |    |    |                                      |    |    |    |    |    |    |    |          |    |    |    |                         |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Bit #            | 31                                      | 30 | 29 | 28 | 27                                   | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19       | 18 | 17 | 16 | 15                      | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field name   | Reserved                                |    |    |    | BDCP                                 |    |    |    |    |    |    |    | Reserved |    |    |    | VDCP                    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | RX                                      |    |    |    | RW                                   |    |    |    |    |    |    |    | RX       |    |    |    | RW                      |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | X                                       |    |    |    | 11B <sub>H</sub> (283 <sub>D</sub> ) |    |    |    |    |    |    |    | X        |    |    |    | 5A3 <sub>H</sub> (1443) |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

This register sets the count of data of the input video stream in PAL format.

Bit 25 to 16 BDCP (Blanking Data Count for PAL)  
Sets count of data processed during blanking period in PAL format

Bit 10 to 0 VDCP (Valid Data Count for PAL)  
Sets count of data processed during valid period in PAL format

**CMSS (Capture Magnify Source Size)**

|                  |                                      |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
|------------------|--------------------------------------|----|----|----|----------------|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|-------|----|---|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress +48 <sub>H</sub> |    |    |    |                |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| Bit #            | 31                                   | 30 | 29 | 28 | 27             | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11    | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field name   | reserved                             |    |    |    | CMSHP          |    |    |    |    |    |    |    |    |    |    |    | reserved |    |    |    | CMSVL |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | R                                    |    |    |    | R/W            |    |    |    |    |    |    |    |    |    |    |    | R        |    |    |    | R/W   |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                    |    |    |    | X <sub>H</sub> |    |    |    |    |    |    |    |    |    |    |    | 0        |    |    |    | X     |    |   |   |   |   |   |   |   |   |   |   |

Bit 27 to 16 CMSHP(Capture Magnify Source Horizontal pixel)  
 This register sets the number of horizontal pixels of the image input before Magnify scaling. Specify the number of horizontal pixels in 2-pixel units.

Bit 11 to 0 CMSVL(Capture Magnify Source Vertical line)  
 This register sets the number of vertical lines of the image input before Magnify scaling.

**CMDS (Capture Magnify Display Size)**

|                  |                                       |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|-------|----|---|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 4C <sub>H</sub> |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |          |    |    |    |       |    |   |   |   |   |   |   |   |   |   |   |
| Bit #            | 31                                    | 30 | 29 | 28 | 27    | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15       | 14 | 13 | 12 | 11    | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field name   | reserved                              |    |    |    | CMDHP |    |    |    |    |    |    |    |    |    |    |    | reserved |    |    |    | CMDVL |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | R                                     |    |    |    | R/W   |    |    |    |    |    |    |    |    |    |    |    | R        |    |    |    | R/W   |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                     |    |    |    | X     |    |    |    |    |    |    |    |    |    |    |    | 0        |    |    |    | X     |    |   |   |   |   |   |   |   |   |   |   |

Bit 27 to 16 CMDHP(Capture Magnify Display Horizontal pixel)  
 This register sets the number of horizontal pixels of the image output after Magnify scaling. Specify the number of horizontal pixels in 2-pixel units.

Bit 11 to 10 CMDVL(Capture Magnify Display Vertical line)  
 This register sets the number of vertical lines of the image output after Magnify scaling.

**RGBHC (RGB input HSYNC Cycle)**

|                  |                                      |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|--------------------------------------|----|----|----|----|----|----|----|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress +80 <sub>H</sub> |    |    |    |    |    |    |    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Bit #            | 31                                   | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20    | 19 | 18 | 17 | 16 | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field name   | Reserved                             |    |    |    |    |    |    |    |    |    |    | RGBHC |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | R                                    |    |    |    |    |    |    |    |    |    |    | R/W   |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                    |    |    |    |    |    |    |    |    |    |    | X     |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Bit 11 to 0 RGBHC(RGB input HSYNC Cycle)  
 This register sets the number of HSYNC cycles of the RGB input.

**RGBHEN (RGB input Horizontal Enable area)**

|                  |                                       |    |        |    |    |    |    |    |    |    |    |    |    |          |    |        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|--------|----|----|----|----|----|----|----|----|----|----|----------|----|--------|----|----|----|----|----|----|---|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 84 <sub>H</sub> |    |        |    |    |    |    |    |    |    |    |    |    |          |    |        |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Bit #            | 31                                    | 30 | 29     | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18       | 17 | 16     | 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field name   | Reserved                              |    | RGBHST |    |    |    |    |    |    |    |    |    |    | Reserved |    | RGBHEN |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | R                                     |    | R/W    |    |    |    |    |    |    |    |    |    |    | R        |    | R/W    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                     |    | X      |    |    |    |    |    |    |    |    |    |    | 0        |    | X      |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

Bit 27 to 16 RGBHST(RGB input Horizontal Enable area Start position)  
 This register sets the position of horizontal active area start position. Setting - 4 is the line count for the start position.

Bit 10 to 0 RGBHEN(RGB input Horizontal Enable area Size)  
 This register sets the number of horizontal active area size of the RGB input. Specify the number of horizontal pixels in 2-pixel units.

**RGBVEN (RGB input Vertical Enable area)**

|                  |                                       |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |          |    |    |    |   |   |   |        |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|----|--------|----|----|----|----|----|----|----|----|----|----|----------|----|----|----|---|---|---|--------|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + 88 <sub>H</sub> |    |    |    |    |    |    |        |    |    |    |    |    |    |    |    |    |    |          |    |    |    |   |   |   |        |   |   |   |   |   |   |
| Bit #            | 31                                    | 30 | 29 | 28 | 27 | 26 | 25 | 24     | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 | 15 | 14 | 13       | 12 | 11 | 10 | 9 | 8 | 7 | 6      | 5 | 4 | 3 | 2 | 1 | 0 |
| Bit field name   | Reserved                              |    |    |    |    |    |    | RGBVST |    |    |    |    |    |    |    |    |    |    | Reserved |    |    |    |   |   |   | RGBVEN |   |   |   |   |   |   |
| R/W              | R                                     |    |    |    |    |    |    | R/W    |    |    |    |    |    |    |    |    |    |    | R        |    |    |    |   |   |   | R/W    |   |   |   |   |   |   |
| Initial value    | 0                                     |    |    |    |    |    |    | X      |    |    |    |    |    |    |    |    |    |    | 0        |    |    |    |   |   |   | X      |   |   |   |   |   |   |

Bit 27 to 16 RGBVST(RGB input Vertical Enable area start Position)  
 This register sets the position of vertical active area start position. Setting - 1 is the line count for the start position.

Bit 9 to 0 RGBVEN(RGB input Vertical Enable area Size)  
 This register sets the number of vertical active area size.

RGBS (RGB input SYNC)

|                  |                                       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |          |    |    |    |    |    |   |   |   |   |     |    |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|-----|----------|----|----|----|----|----|---|---|---|---|-----|----|---|---|---|---|
| Register address | CaputureBaseAddress + 90 <sub>H</sub> |    |    |    |    |    |    |    |    |    |    |    |    |    |    |     |          |    |    |    |    |    |   |   |   |   |     |    |   |   |   |   |
| Bit #            | 31                                    | 30 | 29 | 28 | 27 | 26 | 25 | 24 | 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16  | 15       | 14 | 13 | 12 | 11 | 10 | 9 | 8 | 7 | 6 | 5   | 4  | 3 | 2 | 1 | 0 |
| Bit field name   | Reserved                              |    |    |    |    |    |    |    |    |    |    |    |    |    |    | RM  | Reserved |    |    |    |    |    |   |   |   |   | HP  | VP |   |   |   |   |
| R/W              | R                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | R/W | R        |    |    |    |    |    |   |   |   |   | R/W |    |   |   |   |   |
| Initial value    | 0                                     |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 1   | 0        |    |    |    |    |    |   |   |   |   | 0   | 0  |   |   |   |   |

- Bit 16 RM (RGB Input Mode select)  
Sets Direct RGB input mode  
0: Reserved  
1: RGB666 Direct input mode
- Bit 1 HP (HSYNC Polarity)  
0 Negedge is set to HSYNC  
1 Posedge is set to HSYNC
- Bit 0 VP (VSYNC Polarity)  
0: Negedge is set to VSYNC  
1: Posedge is set to VSUNC

**Conversion Operation**

RGB data is converted to YUV by the following matrix expression :

$$\begin{aligned}
 Y &= a_{11} * R + a_{12} * G + a_{13} * B + b_1 \\
 Cb &= a_{21} * R + a_{22} * G + a_{23} * B + b_2 \\
 Cr &= a_{31} * R + a_{32} * G + a_{33} * B + b_3
 \end{aligned}$$

$a_{ij}$  10bit signed real ( lower 8bit is fraction )  
 $b_i$  8bit unsigned integer

Each coefficients can be defined by following registers.  
 Cb and Cr components are reduced half after this operation to form the 4:2:2 format.

**RGBCMY (RGB Color convert Matrix Y coefficient)**

|                  |                                       |    |    |    |    |    |    |    |    |    |    |                         |    |    |    |    |    |    |    |    |    |    |                        |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|----|----|----|----|----|-------------------------|----|----|----|----|----|----|----|----|----|----|------------------------|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + C0 <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   | a11                                   |    |    |    |    |    |    |    |    |    | Re | a12                     |    |    |    |    |    |    |    |    |    | Re | a13                    |   |   |   |   |   |   |   |   |   |
| R/W              | RW                                    |    |    |    |    |    |    |    |    |    | R  | RW                      |    |    |    |    |    |    |    |    |    | R  | RW                     |   |   |   |   |   |   |   |   |   |
| Initial value    | 0001000010 <sub>b</sub>               |    |    |    |    |    |    |    |    |    | 0  | 0010000000 <sub>b</sub> |    |    |    |    |    |    |    |    |    | 0  | 000011001 <sub>b</sub> |   |   |   |   |   |   |   |   |   |

This register sets the RGB color convert matrix coefficient.

Bit 31 to 22 a11  
 10bit signed real (lower8bit is fraction)

Bit 20 to 11 a12  
 10bit signed real (lower8bit is fraction)

Bit 9 to 0 a13  
 10bit signed real (lower8bit is fraction)

**RGBCMc (RGB Color convert Matrix Cb coefficient)**

|                  |                                       |    |    |    |    |    |    |    |    |    |    |                         |    |    |    |    |    |    |    |    |    |    |                         |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|----|----|----|----|----|-------------------------|----|----|----|----|----|----|----|----|----|----|-------------------------|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + C4 <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   | a21                                   |    |    |    |    |    |    |    |    |    | Re | a22                     |    |    |    |    |    |    |    |    |    | Re | a23                     |   |   |   |   |   |   |   |   |   |
| R/W              | RW                                    |    |    |    |    |    |    |    |    |    | R  | RW                      |    |    |    |    |    |    |    |    |    | R  | RW                      |   |   |   |   |   |   |   |   |   |
| Initial value    | 1111011010 <sub>b</sub>               |    |    |    |    |    |    |    |    |    | 0  | 1110110110 <sub>b</sub> |    |    |    |    |    |    |    |    |    | 0  | 0001110000 <sub>b</sub> |   |   |   |   |   |   |   |   |   |

This register sets the RGB color convert matrix coefficient.

Bit 31 to 22 A21  
 10bit signed real (lower8bit is fraction)

Bit 20 to 11 A22  
 10bit signed real (lower8bit is fraction)

Bit 9 to 0 A23  
 10bit signed real (lower8bit is fraction)

**RGBCMCr (RGB Color convert Matrix Cr coefficient)**

|                  |                                       |    |    |    |    |    |    |    |    |    |    |                         |    |    |    |    |    |    |    |    |    |    |                         |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|----|----|----|----|----|----|-------------------------|----|----|----|----|----|----|----|----|----|----|-------------------------|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + C8 <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   | A31                                   |    |    |    |    |    |    |    |    |    | Re | A32                     |    |    |    |    |    |    |    |    |    | Re | A33                     |   |   |   |   |   |   |   |   |   |
| R/W              | RW                                    |    |    |    |    |    |    |    |    |    | R  | RW                      |    |    |    |    |    |    |    |    |    | R  | RW                      |   |   |   |   |   |   |   |   |   |
| Initial value    | 0001110000 <sub>b</sub>               |    |    |    |    |    |    |    |    |    | 0  | 1110100010 <sub>b</sub> |    |    |    |    |    |    |    |    |    | 0  | 1111101110 <sub>b</sub> |   |   |   |   |   |   |   |   |   |

This register sets the RGB color convert matrix coefficient.

Bit 31 to 22    A31  
                   10bit signed real (lower8bit is fraction)

Bit 20 to 11    A32  
                   10bit signed real (lower8bit is fraction)

Bit 9 to 0        A33  
                   10bit signed real (lower8bit is fraction)

**RGBCMb (RGB Color convert Matrix b coefficient)**

|                  |                                       |                        |    |    |    |    |    |    |    |    |     |                        |    |    |    |    |    |    |    |    |     |                        |   |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|------------------------|----|----|----|----|----|----|----|----|-----|------------------------|----|----|----|----|----|----|----|----|-----|------------------------|---|---|---|---|---|---|---|---|---|---|
| Register address | CaputureBaseAddress + CC <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   | R                                     | B1                     |    |    |    |    |    |    |    |    | Res | b2                     |    |    |    |    |    |    |    |    | Res | b3                     |   |   |   |   |   |   |   |   |   |   |
| R/W              | R                                     | RW                     |    |    |    |    |    |    |    |    | R   | RW                     |    |    |    |    |    |    |    |    | R   | RW                     |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                     | 000010000 <sub>b</sub> |    |    |    |    |    |    |    |    | 0   | 010000000 <sub>b</sub> |    |    |    |    |    |    |    |    | 0   | 010000000 <sub>b</sub> |   |   |   |   |   |   |   |   |   |   |

This register sets the RGB color convert matrix coefficient.

Bit 30 to 22    B1  
                   9bit unsigned integer

Bit 19 to 11    B2  
                   9bit unsigned integer

Bit 8 to 0        B3  
                   9bit unsigned integer

### 10.2.6 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 | PE | CE | FCNT   |    |    |    |    |    |    | NF | FF | FE | SS |    |   | DS |   |   | PS |   |   |   |   |   |
| R/W              |                                    |    |    |    |    |    |    | RW | RW | RW | 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>H</sub>)

Bit 22 CE (Display List Command Error)  
Indicates command error occurrence

- 0 Normal
- 1 Command error detected

Bit 23 PE (Display List Packet code Error)  
Indicates packet code error occurrence

- 0 Normal
- 1 Packet code 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 miller 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.

### 10.2.7 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/for TopLeft)**

|                  |                                    |    |    |    |       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |   |      |    |    |    |    |   |   |   |
|------------------|------------------------------------|----|----|----|-------|----|----|----|----|----|----|----|----|----|----|----|----|----|----|------|----|----|----|---|------|----|----|----|----|---|---|---|
| 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 | AS |    |    |   |   |   |
| R/W              |                                    |    |    |    | RW    |    |    |    |    |    |    | RW | RW |    |    |    |    |    |    | RW   |    | RW | RW |   | RW   | RW |    | RW | RW |   |   |   |
| Initial value    |                                    |    |    |    | 00000 |    |    |    |    |    |    | 0  | 0  |    |    |    |    |    |    | 0011 |    | 0  | 0  |   | 0000 | 0  | 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, and for the top-left non-applicable primitive.

The value after a drawing that involves the shade primitive, the edge primitive, or the top-left non-applicable primitive is the value set for MDR1.

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)<br>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)<br>Selects line type                   |
|              | 0     | Solid line  |
|              | 1     | Broken line   |
| Bit 20       |       | BP (Broken line Period)<br>Selects broken line cycle    |
|              | 0:    | 32 bits   |
|              | 1:    | 24 bits   |
| Bit 28 to 24 |       | LW (Line Width)<br>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   |                                    |    |    |    |    |    |    |    |    |    |    | LOG  |    | BM | ZW | ZCL  |    |    | ZC | AS | SM |    |   |   |   |   |   |   |   |   |   |   |
| R/W              |                                    |    |    |    |    |    |    |    |    |    |    | RW   |    | RW | RW | RW   |    |    | RW | RW | RW |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    |                                    |    |    |    |    |    |    |    |    |    |    | 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.

(Must set SM=AS=TT=0 for MDR2S)

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 |    | TBU |   |   |   |   |   |   |   |   |   |
| R/W              |                                    |    |    |    |    |    |    | RW | RW  |    |    | RW  |    |    | RW  |    | RW  |    | RW |    | RW |    | RW  |   |   |   |   |   |   |   |   |   |
| Initial value    |                                    |    |    |    |    |    |    | 0  | 00  |    |    | 00  |    |    | 00  |    | 00  |    | 0  |    | 0  |    | 0   |   |   |   |   |   |   |   |   |   |

This register sets the texture mapping mode.

- Bit 0                    TBU (Texture Buffer)

Selects whether to use the internal buffer or graphics memory as texture memory. Internal buffer is always used for tiling.

0    External (frame) Graphics Memory

1    Internal buffer
  
- 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 + 430H |    |    |    |    |    |    |    |    |    |    |    |      |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |
| 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

**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    | 100000000000                       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | 100000000000 |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

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 (Texture Buffer 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.

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   |                                    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    | FGC |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| 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.

24 bit color mode:

- Bit 23 to 0    FGC24 (Foreground 24 bit Color)  
This field sets the 24-bit direct color for the foreground.
- Bit 31 to 24   These bits are not used.  
32-bit units are used for memory, but these bits are reflected in bit 31 to 24 (MSB side).

**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   | BGC8/16/24  |
| 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

**24 bit color mode:**

- Bit 23 to 0    BGC24 (Background 24 bit Color)  
Sets 24-bit direct color for the background
- Bit 30 to 24   Not used  
32-bit units are used for memory, but these bits are reflected in bit 31 to 24 (MSB side)
- Bit 31        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)



**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   | BC16/24                            |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              | RW                                 |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    | 0                                  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

This register sets the border color for texture mapping.

16 bit color mode:

- Bit 15 to 0 BC16 (Border Color)  
Sets the 16-bit direct color for the texture border color

24 bit color mode:

- Bit 23 to 0 BC24 (Border Color)  
Sets the 24-bit direct color for the texture border color
- Bit 31 to 24 Not used  
32-bit units are used for memory but these bits are reflected in bit 31 to 24 (MSB side)

**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    | 1111                               |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |     |   |   |   |

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.



**(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  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| dRdx     | 0044 <sub>H</sub> | S  | S  | S  | S  | S  | S  | S  | S  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| dRdy     | 0048 <sub>H</sub> | S  | S  | S  | S  | S  | S  | S  | S  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Gs       | 004C <sub>H</sub> | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| dGdx     | 0050 <sub>H</sub> | S  | S  | S  | S  | S  | S  | S  | S  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| dGdy     | 0054 <sub>H</sub> | S  | S  | S  | S  | S  | S  | S  | S  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| Bs       | 0058 <sub>H</sub> | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| dBdx     | 005C <sub>H</sub> | S  | S  | S  | S  | S  | S  | S  | S  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| dBdy     | 0060 <sub>H</sub> | S  | S  | S  | S  | S  | S  | S  | S  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |

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  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| dZdx     | 0084 <sub>h</sub> | S  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |
| dZdy     | 008C <sub>h</sub> | S  |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |  |  |

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   |



### 10.2.10 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 |

### 10.2.11 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                        |



### 10.2.12 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<br>For indirect color, set a palette code in the lower 8 bits. |





### 10.2.14 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 |

### 10.2.15 Geometry control register

#### GCTR (Geometry Control Register)

|                  |                                       |    |    |    |    |     |    |        |    |    |    |    |    |    |     |    |    |     |    |    |     |    |    |   |   |   |   |   |   |   |   |   |
|------------------|---------------------------------------|----|----|----|----|-----|----|--------|----|----|----|----|----|----|-----|----|----|-----|----|----|-----|----|----|---|---|---|---|---|---|---|---|---|
| Register address | GeometryBaseAddress + 00 <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                              |    |    |    | 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   |    | 100000 |    |    |    | 0  | 0  | 1  | X   |    | 00 | X   |    | 00 | X   |    | 00 |   |   |   |   |   |   |   |   |   |

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

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

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>H</sub>) of display list FIFO (DFIFOD)
- Bit 24      FO (FIFO Overflow)  
Indicates whether FIFO overflow occurred  
0      Normal  
1      FIFO overflow

### 10.2.16 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 |    | RW | RW | RW | RW |   |   |   |   |   |   |   |   |   |
| 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   |                                       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| R/W              |                                       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |
| Initial value    |                                       |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |    |   |   |   |   |   |   |   |   |   |   |

This register sets the geometry mode at line drawing.

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   | PO                 | LV |    |    |    |    |    |    |    |    |    | TC |    |    |    | BC |    | UW | BM | TM |    |    | BP | SP |   |   |   | BO |   | EP |   | AA |
| R/W              | W                  | W  |    |    |    |    |    |    |    |    |    | W  |    |    |    | W  |    | W  | W  | W  |    |    | W  | W  |   |   |   | W  |   | W  |   | W  |
| Initial value    | 0                  | 0  |    |    |    |    |    |    |    |    |    | 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”.

- Bit 31 PO (Primitive Order Control)  
 Sets the draw order for body/edge/shadow  
 0 Body -> Edge -> shadow (faster)  
 1 Shadow -> Edge -> Body (quality for anti-alias)
  
- Bit 30 LV (Line Version Control)  
 Sets the Coral Line algorithm version  
 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 line joint not performed  
 1 Interpolation of bold line joint performed
  
- 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
  
- Bit 14 UW (Uniform line Width)  
 Sets the line width equalization mode  
 0 Equalization of line width not performed  
 1 Equalization of line width performed
  
- 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).  
 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)  
 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  
 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

Bit 8 SP (Shadow Primitive)  
 Sets the drawing mode for the shadow primitive  
 0 Shadow primitive not drawn  
 1 Shadow primitive drawn

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  
 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.

- 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)

### 10.2.17 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

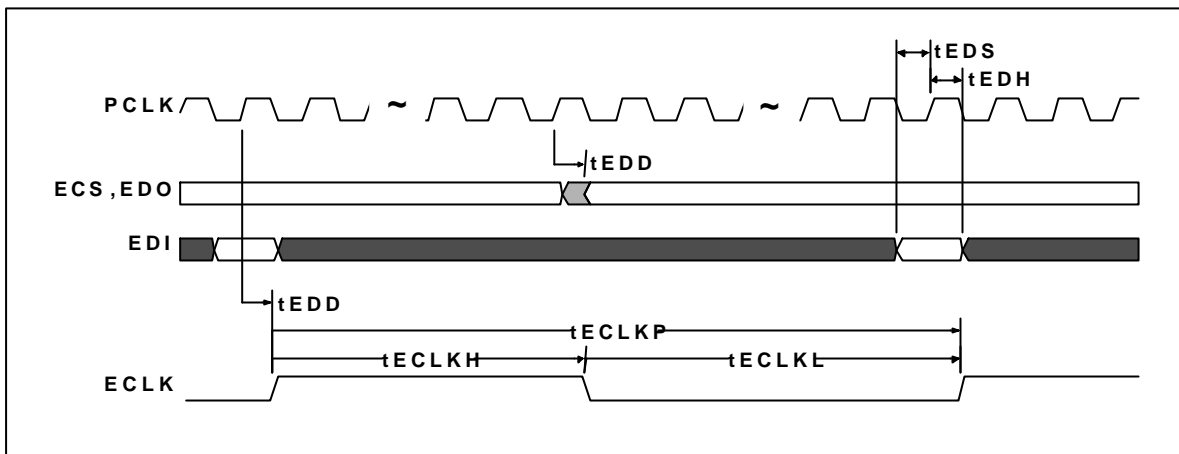
## **11. TIMING DIAGRAM**

### **11.1 Host Interface**

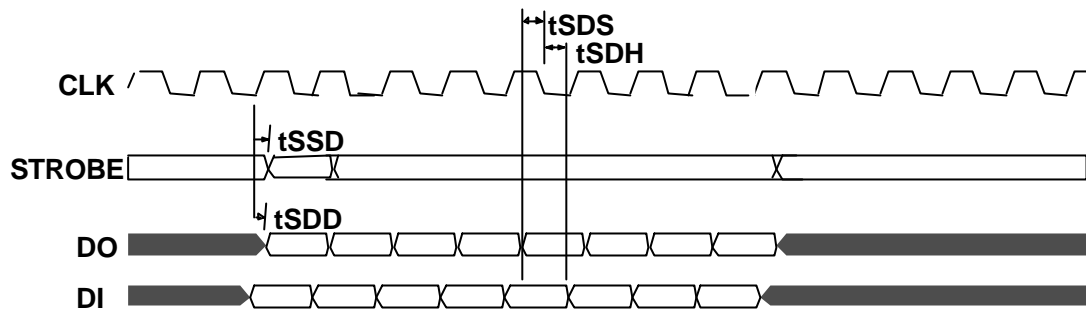
#### **11.1.1 PCI Interface**

Standard PCI V2.1.

11.1.2 EEPROM Timing



### 11.1.3 Serial Interface Timing



## 11.2 I<sup>2</sup>C Interface

### I<sup>2</sup>C Bus Timing

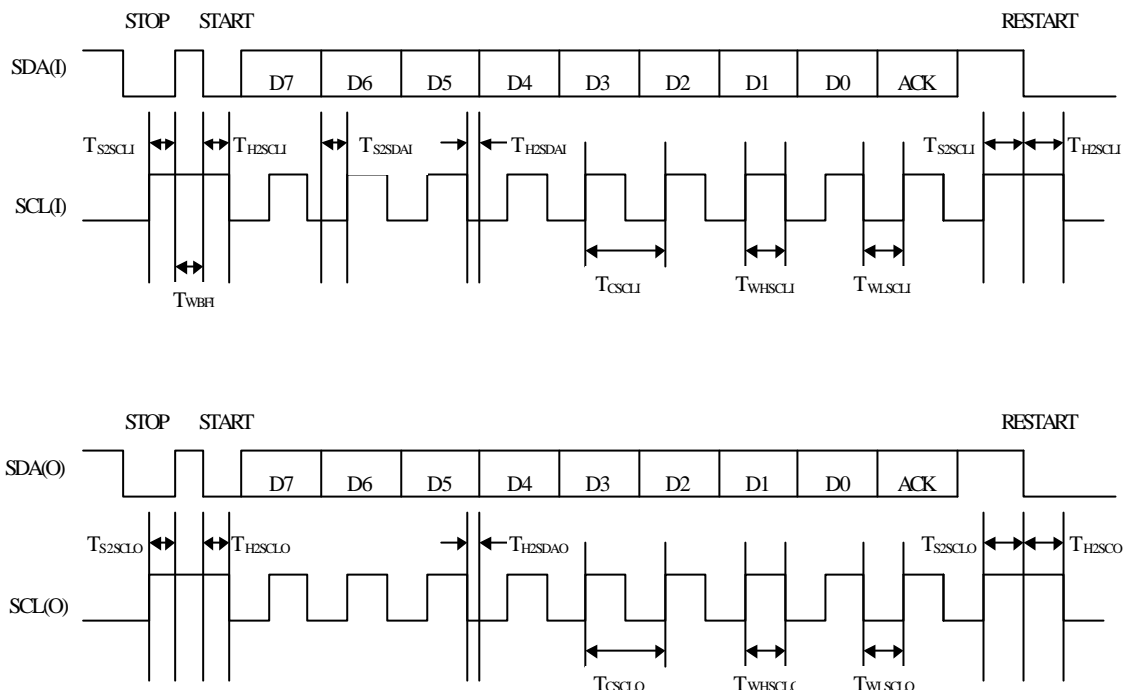


Fig.11.1 I2C bus timing

### Interruption Timing

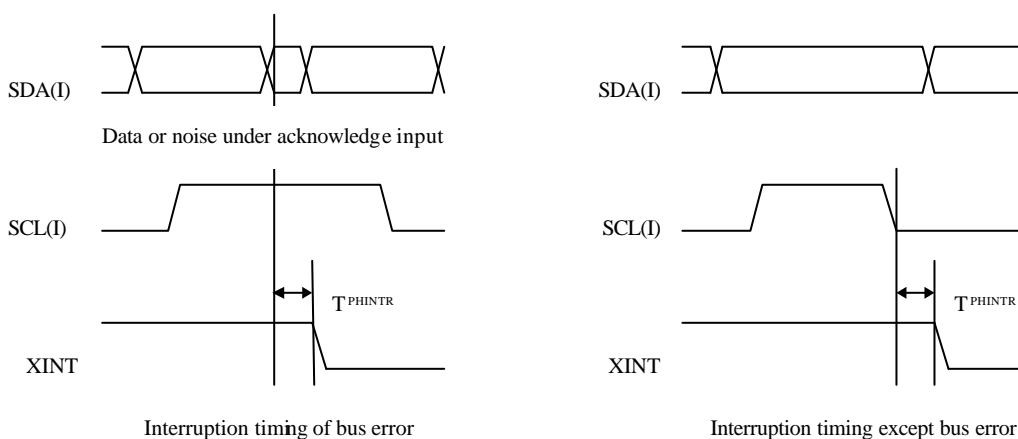


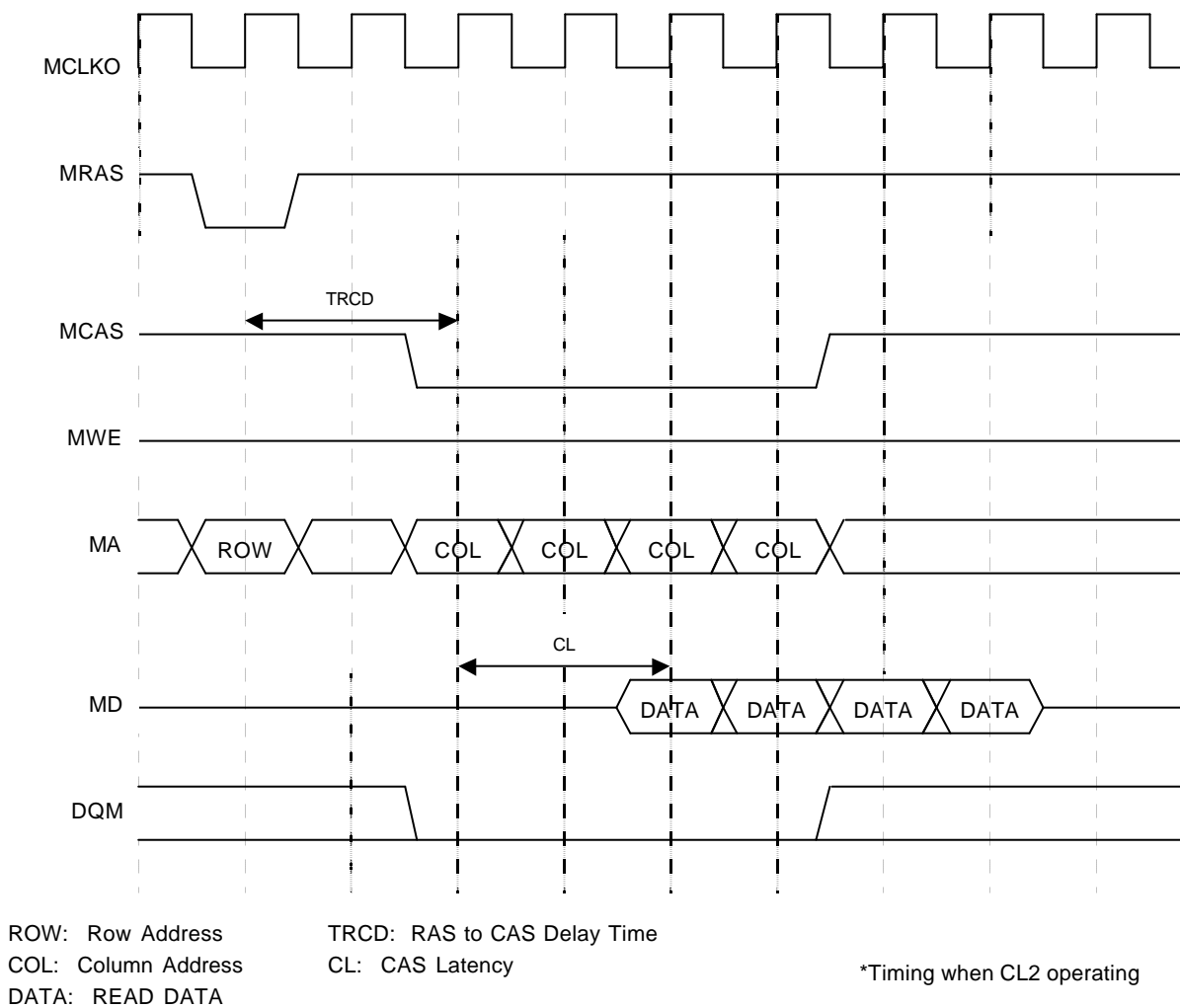
Fig.11.2 Interruption timing



### 11.3 Graphics Memory Interface

The CORAL access timing and graphics memory access timing are explained here.

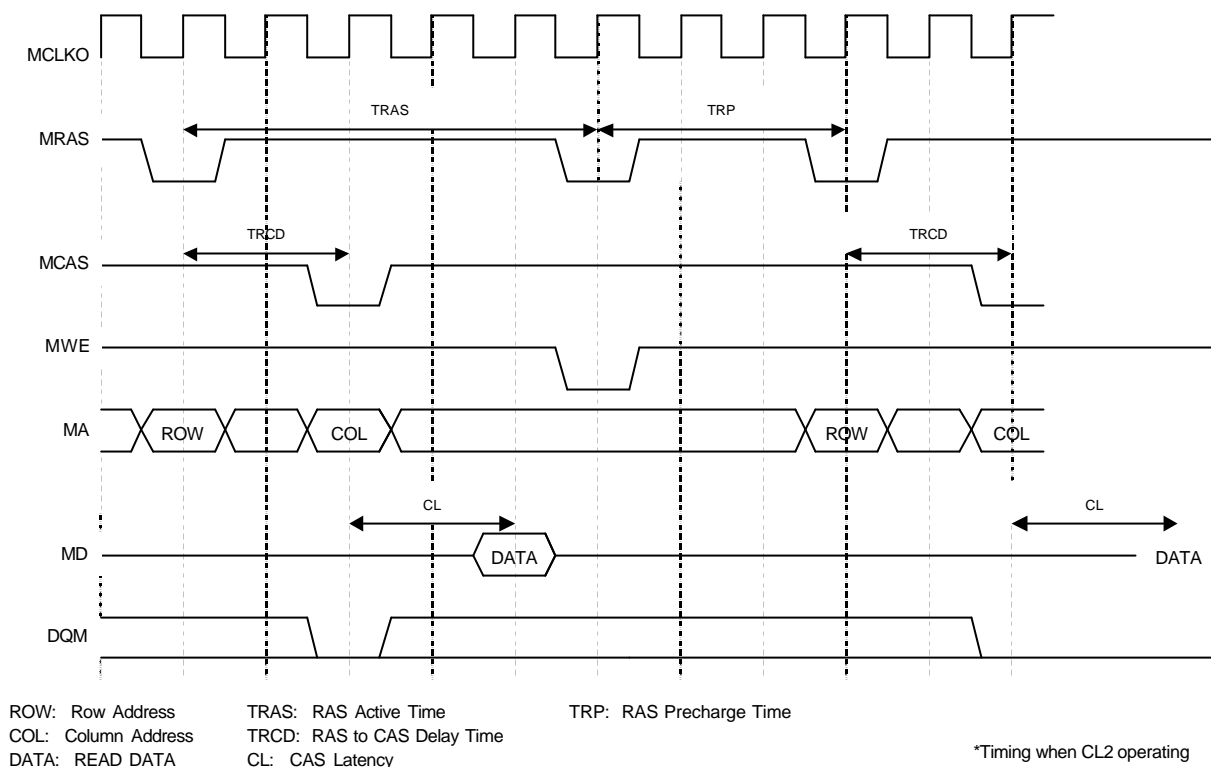
#### 11.3.1 Timing of read access to same row address



**Fig. 11.3 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.

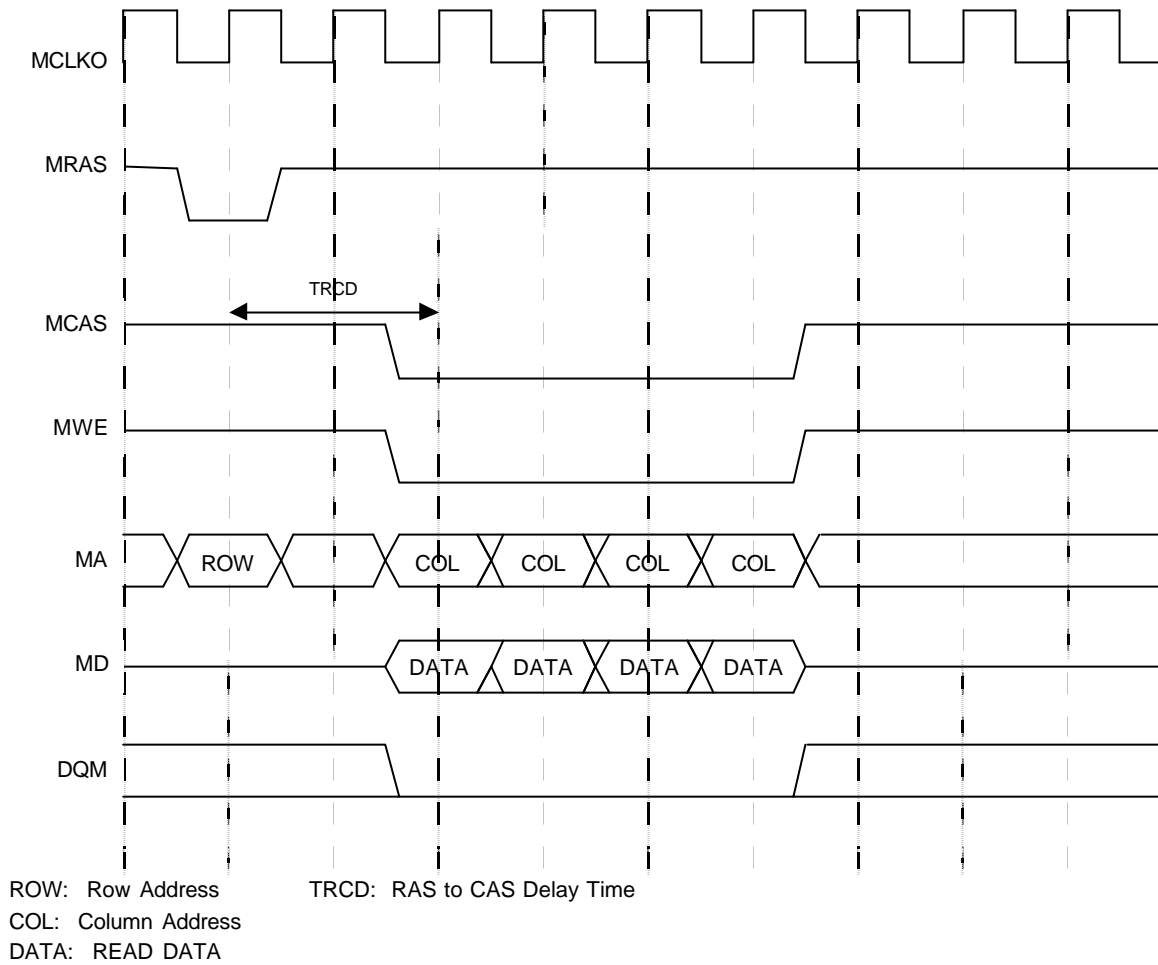
### 11.3.2 Timing of read access to different row addresses



**Fig. 11.4 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.

### 11.3.3 Timing of write access to same row address

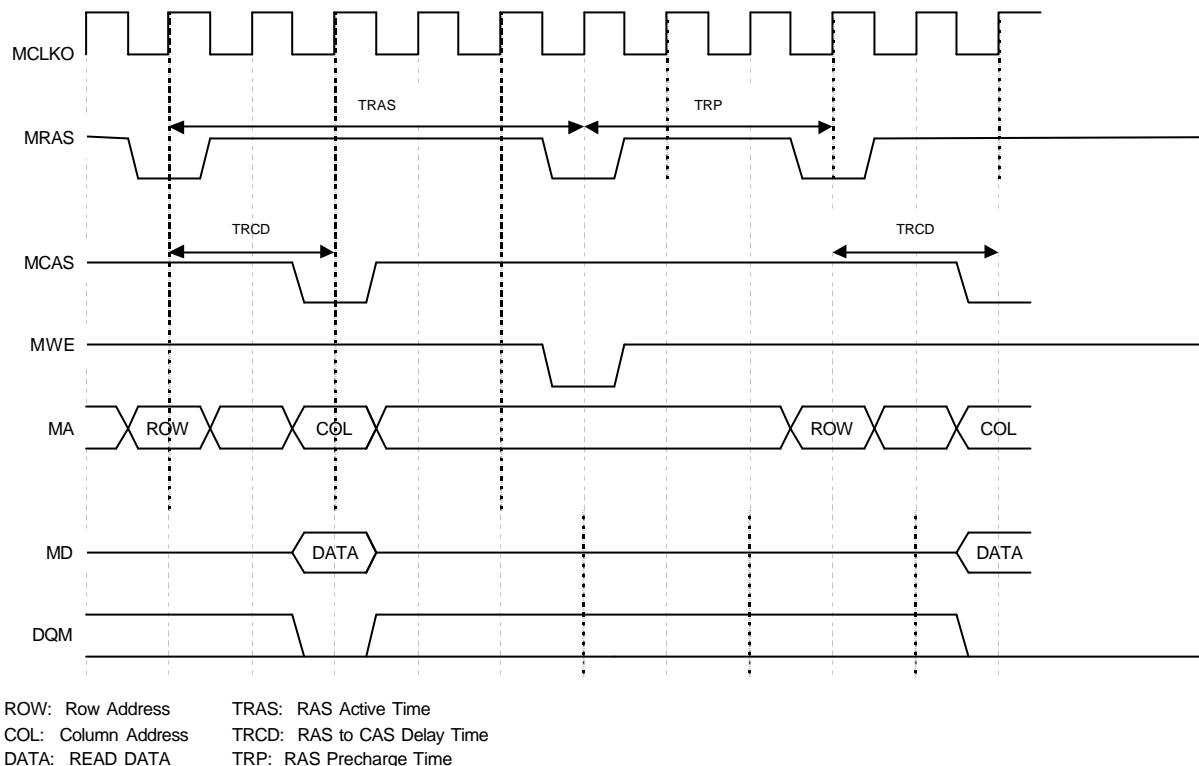


**Fig. 11.5 Timing of Write Access to Same Row Address**

The above timing diagram shows that write access is made form times form 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.

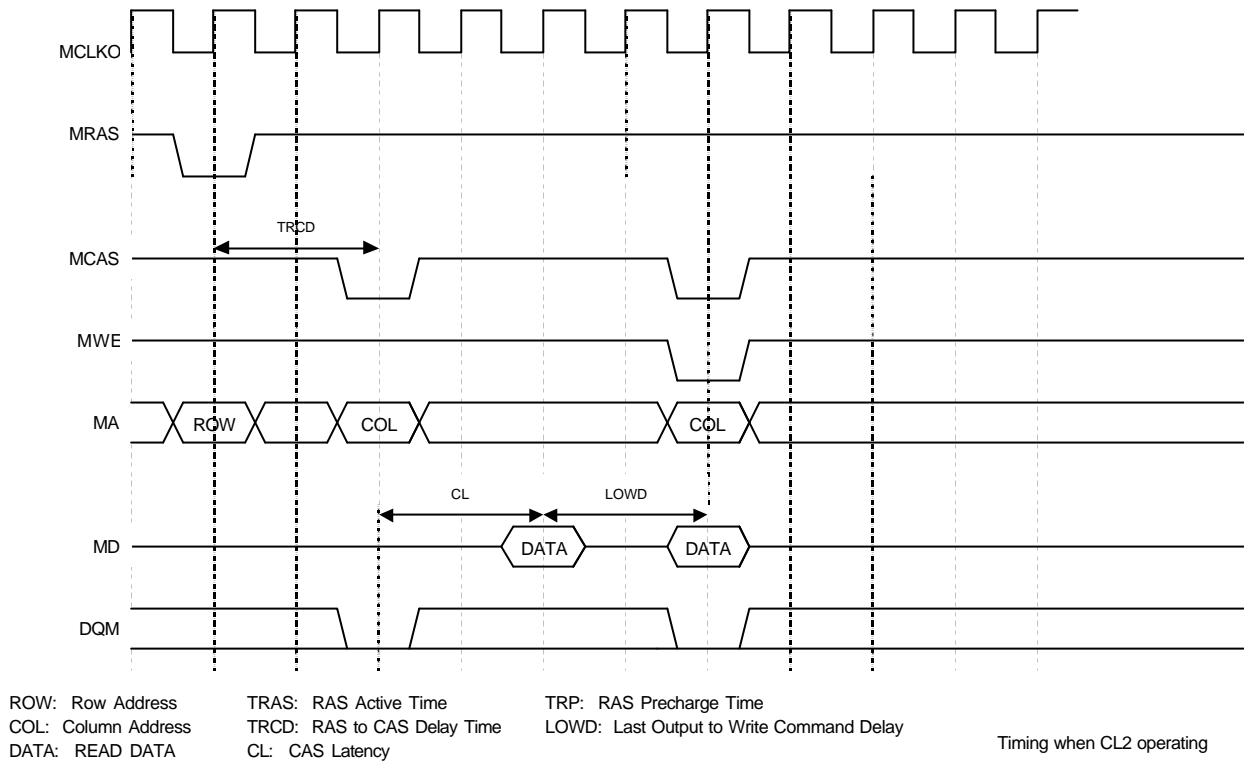
### 11.3.4 Timing of write access to different row addresses



**Fig. 11.6 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.

### 11.3.5 Timing of read/write access to same row address

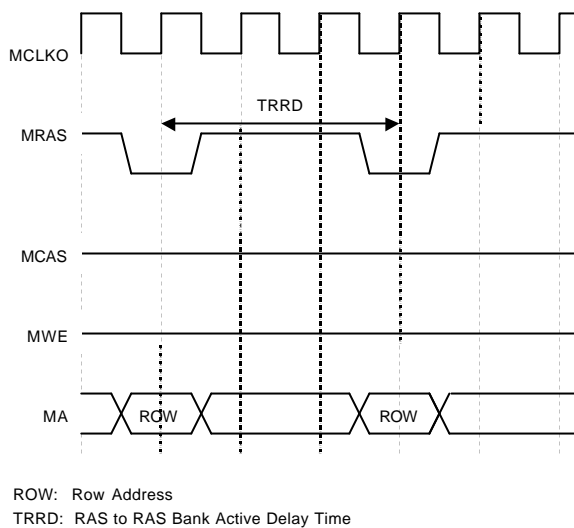


**Fig. 11.7 Timing of Read/Write Access to Same Row Address**

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.

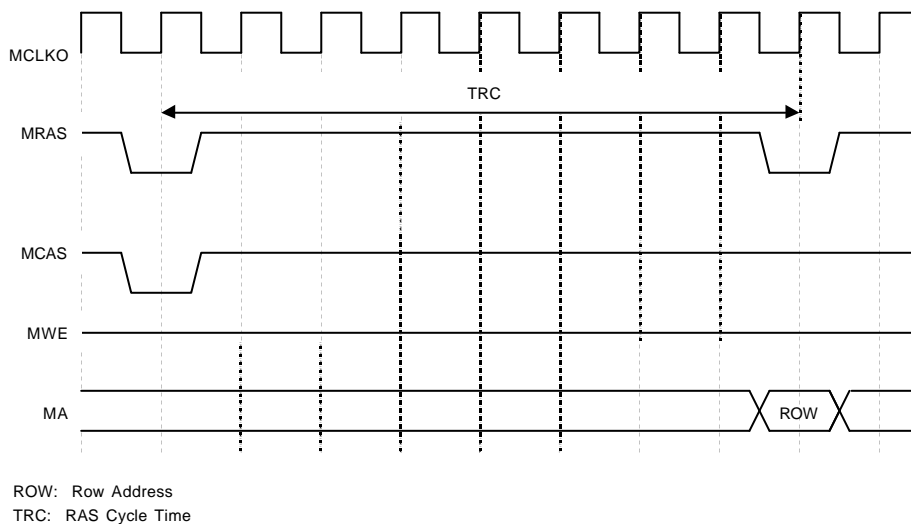
### 11.3.6 Delay between ACTV commands



**Fig.11.8 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.

### 11.3.7 Delay between Refresh command and next ACTV command



**Fig. 11.9 Delay between Refresh Command and Next ACTV Command**

The **ACTV** command is issued after the elapse of TRC after issuance of the **Refresh** command.

## 11.4 Display Timing

### 11.4.1 Non-interlace mode

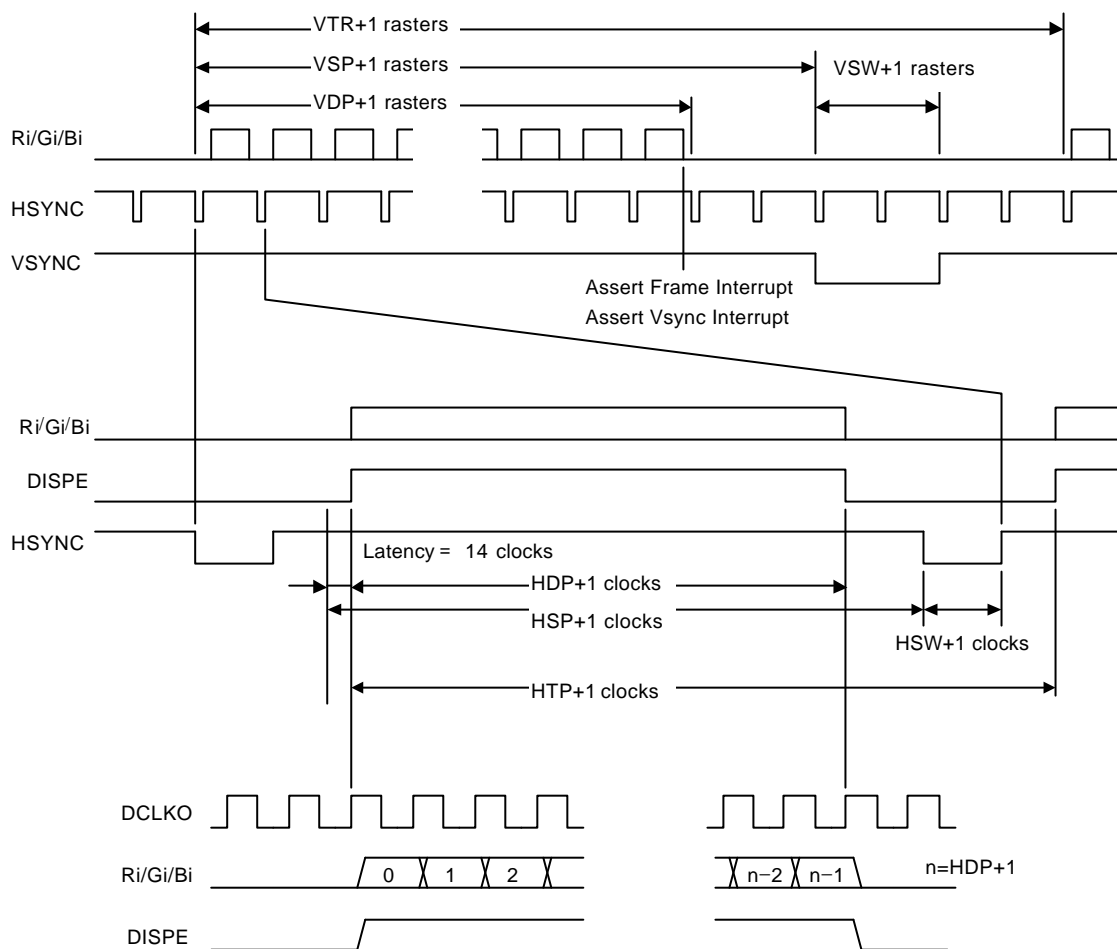
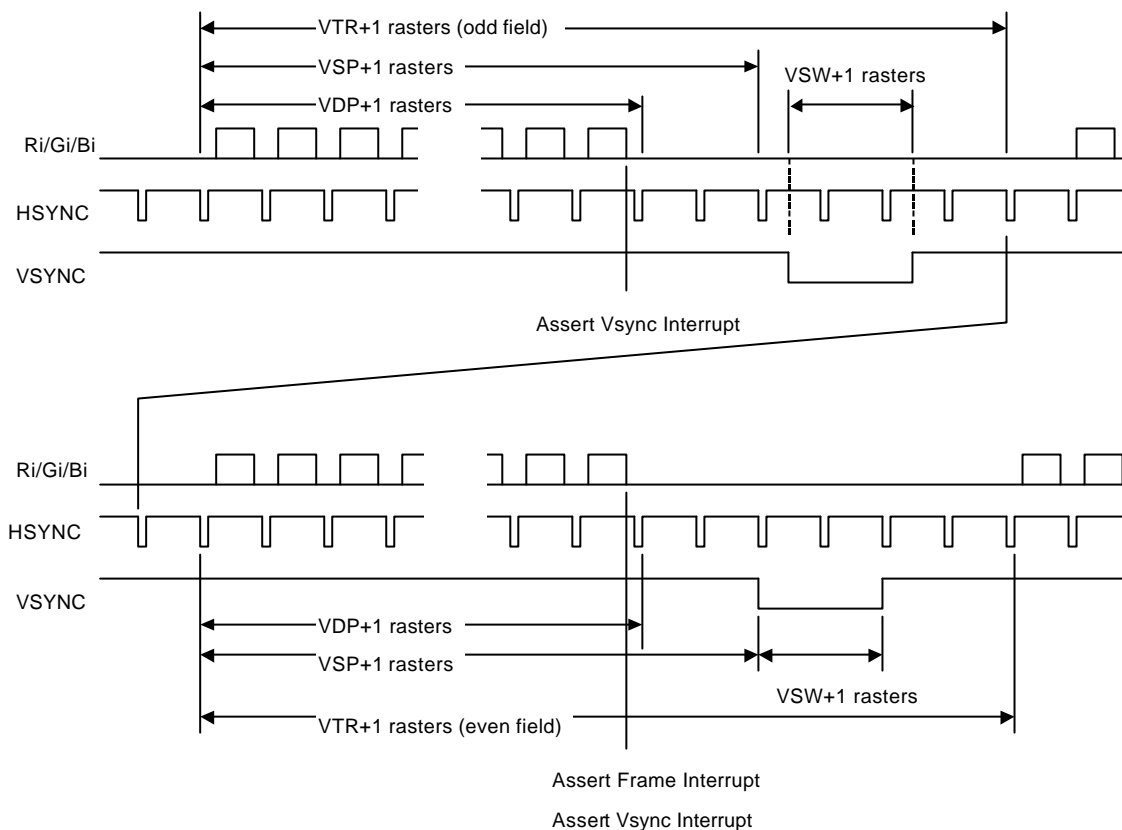


Fig. 11.10 Non-interlace Timing

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.

### 11.4.2 Interlace video mode



**Fig. 11.11 Interlace Video Timing**

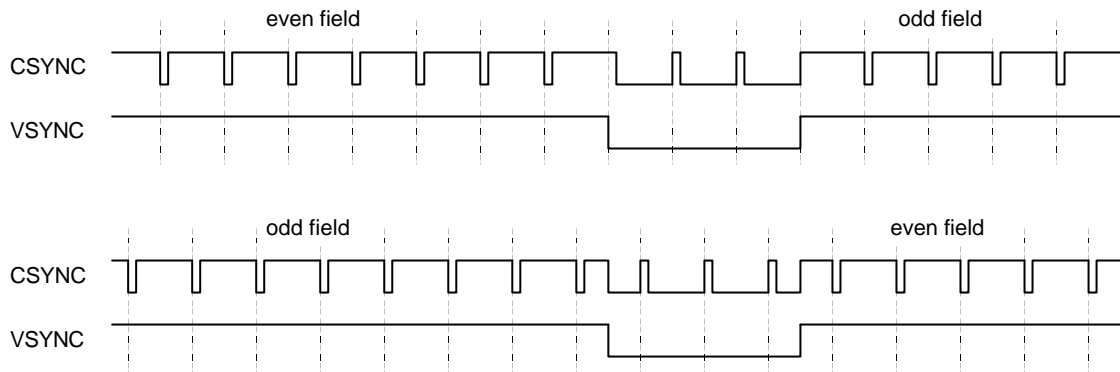
In the above diagram, VTR, HDP, etc., are the setting values of their associated registers.

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.



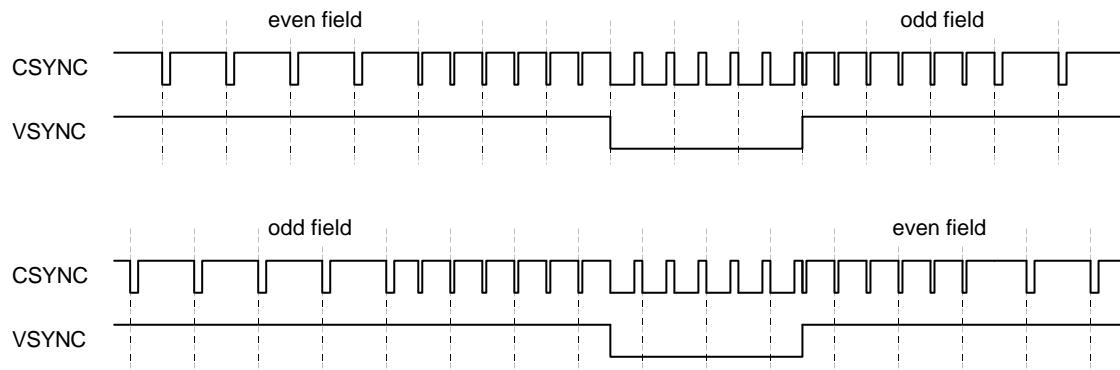
### 11.4.3 Composite synchronous signal

When the EEQ bit of the DCM register is “0”, the CSYNC signal output waveform is as shown below.



**Fig 11.12 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 11.13 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.

## 12. ELECTRICAL CHARACTERISTICS

### 12.1 Introduction

The values in this chapter are valid for the final specification of MB86295.

### 12.2 Maximum Rating

Maximum Rating

| Parameter                       | Symbol                      | Maximum rating                                   | Unit |
|---------------------------------|-----------------------------|--|------|
| Power supply voltage            | $V_{DDL}^{*1}$<br>$V_{DDH}$ | $-0.5 < V_{DDL} < 2.5$<br>$-0.5 < V_{DDH} < 4.0$ | V    |
| Input voltage                   | $V_I$                       | $-0.5 < V_I < V_{DDH} + 0.5 (< 4.0)$             | V    |
| Output current                  | $I_O$                       | $\pm 13$   | mA   |
| Power pin current               | IPOW                        | 68   | 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

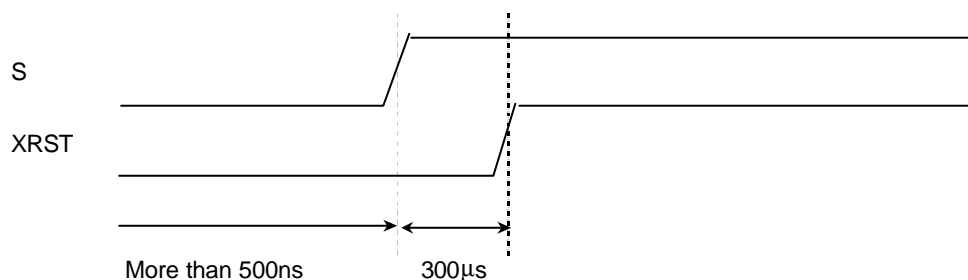
| Parameter                         | Symbol       | Rating |      |               | Unit |
|-----------------------------------|--------------|--------|------|---------------|------|
|                                   |              | Min.   | Typ. | Max.          |      |
| Supply voltage                    | $V_{DDL}$ *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 reset sequences is as follows:

S is changed from “Low” to “High” levels and then XRST is changed from “Low” to “High” level:



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 XRST pins for 300 µs or more.

## 12.4 DC Characteristics

Measuring condition:  $V_{DDL} = 1.8 \pm 1.5$  V,  $V_{DDH} = 3.3 \pm 0.3$  V,  $V_{SS} = 0.0$  V,  $T_a = 0-70^\circ\text{C}$

| Parameter                                      | Symbol  | Rating               |      |           | Unit          |
|--|---|----------------------|------|-----------|---------------|
|  |   | Min.                 | Typ. | Max.      |               |
| Output voltage <sup>*1</sup><br>("High" level) | $V_{OH}$  | $V_{DDH}-0.2$        |      | $V_{DDH}$ | V             |
| Output voltage <sup>*2</sup><br>("Low" level)  | $V_{OL}$  | 0.0                  |      | 0.2       | V             |
| Output current<br>("High" level)               | $I_{OH1}$ <sup>*3</sup><br>$I_{OH2}$ <sup>*4</sup><br>$I_{OH3}$ <sup>*5</sup> | -2.0<br>-4.0<br>-8.0 |      |           | mA            |
| Output current<br>("Low" level)                | $I_{OL1}$ <sup>*3</sup><br>$I_{OL2}$ <sup>*4</sup><br>$I_{OL3}$ <sup>*5</sup> | 2.0<br>4.0<br>8.0    |      |           | mA            |
| Input leakage current                          | IL  |                      |      | $\pm 5$   | $\mu\text{A}$ |
| Pin capacitance                                | C   |                      |      | 16        | pF            |

\*1  $I_{OH} = -100 \mu\text{A}$

\*2  $I_{OL} = 100 \mu\text{A}$

\*3 Output characteristics of MD0 to 63 and MDQM0 to 7 signals

\*4 Output characteristics of signals other than signals indicated by \*3 and \*5

\*5 Output characteristic of XINT and MCLK0 signals

## 12.5 AC Characteristics

### 12.5.1 Host interface

#### PCI Interface

| Parameter                                   | Signal   | Abbrev.     | Values |     |     | Units |
|---|--|-------------|--------|-----|-----|-------|
|   |  |             | Min    | Typ | Max |       |
| PCI Clock Period                            | PCLK   | $t_{PCLKP}$ | 30     |     |     | ns    |
| PCI Clock Low Time                          | PCLK   | $t_{PCLKL}$ | 11     |     |     | ns    |
| PCI Clock High Time                         | PCLK   | $t_{PCLKH}$ | 11     |     |     | ns    |
| PCI Input Setup<br>(bussed signals)         | AD[31:0],<br>C/BE[3:0],<br>PAR,<br>FRAME,<br>IRDY, TRDY,<br>STOP,<br>IDSEL,<br>DEVSEL,<br>PERR               | $t_{PS}$    | 7      |     |     | ns    |
| PCI Input Setup<br>(point-to-point signals) | GNT  | $t_{PSP}$   | 10     |     |     | ns    |
| PCI Input Hold                              | AD[31:0],<br>C/BE[3:0],<br>PAR,<br>FRAME,<br>IRDY, TRDY,<br>STOP,<br>IDSEL,<br>DEVSEL,<br>PERR, GNT          | $t_{PH}$    | 0      |     |     | ns    |
| PCI Output Delay                            | AD[31:0],<br>C/BE[3:0],<br>PAR,<br>FRAME,<br>IRDY, TRDY,<br>STOP,<br>IDSEL,<br>DEVSEL,<br>PERR,<br>SERR, REQ | $t_{PD}$    | 2      |     | 11  | ns    |

#### PCI EEPROM Interface

| Parameter         | Signal | Abbrev.   | Values |     |     | Units |
|-------------------|--------|-----------|--------|-----|-----|-------|
|                   |        |           | Min    | Typ | Max |       |
| EEPROM Data Setup | EDI    | $T_{EDS}$ | 5      |     |     | ns    |
| EEPROM Data Hold  | EDI    | $T_{EDH}$ | 5      |     |     | ns    |

|                           |                  |             |      |  |    |    |
|---------------------------|------------------|-------------|------|--|----|----|
| EEPROM Data Delay         | EDO, ECK,<br>ECS | $T_{EDD}$   | 3    |  | 20 | ns |
| EEPROM Clock<br>Period    | ECK              | $T_{ECLKP}$ | 1000 |  |    | ns |
| EEPROM Clock Low<br>Time  | ECK              | $T_{ECLKL}$ | 500  |  |    | ns |
| EEPROM Clock High<br>Time | ECK              | $t_{ECLKH}$ | 500  |  |    | ns |

### Serial Interface

| Parameter           | Signal | Abbrev.   | Values |     |     | Units |
|---------------------|--------|-----------|--------|-----|-----|-------|
|                     |        |           | Min    | Typ | Max |       |
| Serial Strobe Delay | SB     | $T_{SSD}$ | -      |     | -   | ns    |
| Serial Data Data    | EDO    | $T_{SDD}$ | -      |     | -   | ns    |
| Serial Data Setup   | EDI    | $T_{SDS}$ | -      |     |     | ns    |
| Serial Data Hold    | EDI    | $T_{SDH}$ | -      |     |     | ns    |

12.5.2 I<sup>2</sup>C InterfaceI<sup>2</sup>C bus timing

| symbol              |                    | MIN        | MAX                          | unit               |
|---------------------|--------------------|------------|------------------------------|--------------------|
| T <sub>S2SDAI</sub> | SDA(I) setup time  | standard   | 250                          | ns                 |
|                     |                    | high-speed | 100                          | ns                 |
| T <sub>H2SDAI</sub> | SCL(I) hold time   | standard   | 0                            | ns                 |
|                     |                    | high-speed | 0                            | ns                 |
| T <sub>CSCLI</sub>  | SCL(I) cycle time  | standard   | 10.0                         | us                 |
|                     |                    | high-speed | 2.5                          | us                 |
| T <sub>WHSCLI</sub> | SCL(I) H period    | standard   | 4.0                          | us                 |
|                     |                    | high-speed | 0.6                          | us                 |
| T <sub>WLSCLI</sub> | SCL(I) L period    | standard   | 4.7                          | us                 |
|                     |                    | high-speed | 1.3                          | us                 |
| T <sub>CSALO</sub>  | SCL(O) cycle time  | standard   | $2*m+2^{(r2)}$               | PCLK* <sub>1</sub> |
|                     |                    | high-speed | $\text{int}(1.5*m)+2^{(r2)}$ | PCLK* <sub>1</sub> |
| T <sub>WHALO</sub>  | SCL(O) H period    | standard   | $m+2^{(r2)}$                 | PCLK* <sub>1</sub> |
|                     |                    | high-speed | $\text{int}(0.5*m)+2^{(r2)}$ | PCLK* <sub>1</sub> |
| T <sub>WLSALO</sub> | SCL(O) L period    | standard   | $m^{(r2)}$                   | PCLK* <sub>1</sub> |
|                     |                    | high-speed | $m^{(r2)}$                   | PCLK* <sub>1</sub> |
| T <sub>W2SCLI</sub> | SCL(I) setup time  | standard   | 4.0                          | us                 |
|                     |                    | high-speed | 0.6                          | us                 |
| T <sub>H2SCLI</sub> | SCL(I) hold time   | standard   | 4.7                          | us                 |
|                     |                    | high-speed | 1.3                          | us                 |
| T <sub>WBF1</sub>   | bus free time      | standard   | 4.7                          | us                 |
|                     |                    | high-speed | 1.3                          | us                 |
| T <sub>S2SALO</sub> | SCL(O) set up time | standard   | $m+2^{(r2)}$                 | PCLK* <sub>1</sub> |
|                     |                    | high-speed | $\text{int}(0.5*m)+2^{(r2)}$ | PCLK* <sub>1</sub> |
| T <sub>H2SALO</sub> | SCL(O) hold time   | standard   | $m-2^{(r2)}$                 | PCLK* <sub>1</sub> |
|                     |                    | high-speed | $\text{int}(0.5*m)-2^{(r2)}$ | PCLK* <sub>1</sub> |
| T <sub>H2SDAO</sub> | SDA(O) hold time   | 5          |                              | PCLK* <sub>1</sub> |

\*1 PCLK is an internal clock of I2C module. (16.6MHz)

\*2 Refer to the clock control register (CCR) for the value of m.

## Timing of interrupt

| symbol              |                               | MIN | MAX | unit |
|---------------------|-------------------------------|-----|-----|------|
| T <sub>PHINTR</sub> | XINT delay (bus error)        |     | 4   | PCLK |
| T <sub>PHINTR</sub> | XINT delay (except bus error) |     | 4   | PCLK |

### 12.5.3 Video interface

#### Clock

| Parameter       | Symbol              | Condition | Rating |        |      | Unit |
|-----------------|---------------------|-----------|--------|--------|------|------|
|                 |                     |           | Min.   | Typ.   | Max. |      |
| CLK Frequency   | f <sub>CLK</sub>    |           |        | 14.318 |      | MHz  |
| CLK H-width     | t <sub>HCLK</sub>   |           | 25     |        |      | ns   |
| CLK L-width     | t <sub>LCLK</sub>   |           | 25     |        |      | ns   |
| DCLKI Frequency | f <sub>DCLKI</sub>  |           |        |        | 67   | MHz  |
| DCLKI H-width   | t <sub>HDCLKI</sub> |           | 5      |        |      | ns   |
| DCLKI L-width   | t <sub>LDCLKI</sub> |           | 5      |        |      | ns   |
| DCLKO frequency | f <sub>DCLKO</sub>  |           |        |        | 67   | MHz  |

#### Input signals

| Parameter               | Symbol               | Condition | Rating |      |      | Unit             |
|-------------------------|----------------------|-----------|--------|------|------|------------------|
|                         |                      |           | Min.   | Typ. | Max. |                  |
| HSYNC Input pulse width | t <sub>WHSYNC0</sub> | *1        | 3      |      |      | clock            |
|                         | t <sub>WHSYNC1</sub> | *2        | 3      |      |      | clock            |
| HSYNC Input setup time  | t <sub>SHSYNC</sub>  | *2        | 10     |      |      | ns               |
| HSYNC Input hold time   | t <sub>HHSYNC</sub>  | *2        | 10     |      |      | ns               |
| VSYNC Input pulse width | t <sub>WHSYNC1</sub> |           | 1      |      |      | HSYNC<br>1 cycle |

\*1 Applied only in PLL synchronization mode (CKS = 0), reference clock output from internal PLL (cycle = 1/14\*f<sub>CLK</sub>)

\*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 <sub>RGB</sub>    |           | 2      |      | 11   | ns   |
| DISPE Output delay time | t <sub>DEO</sub>    |           | 2      |      | 10   | ns   |
| HSYNC Output delay time | t <sub>DHSYNC</sub> |           | 2      |      | 10   | ns   |
| VSYNC Output delay time | t <sub>DVSYNC</sub> |           | 2      |      | 10   | ns   |
| CSYNC Output delay time | t <sub>DCSYNC</sub> |           | 2      |      | 10   | ns   |
| GV Output delay time    | t <sub>DGV</sub>    |           | 2      |      | 10   | ns   |



## 12.5.4 Graphics memory interface

### 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 twelve cases of external SDRFAM capacitance.

### (1) External SDRAM capacitance case 1

#### External SDRAM capacitance

| SDRAM x1         | Total capacitance                           | Unit |
|------------------|---|------|
| MCLKO            | 9.8pF (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 | $t_{DID}$   |           | 0         |      | 4.2  | ns   |
| MA, MRAS, MCAS, MWE Access time       | $t_{MAD}$   |           | 1.0       |      | 5.0  | ns   |
| MDQM Access time                      | $t_{MDQMD}$ |           | 1.1       |      | 5.4  | ns   |
| MD Output access time                 | $t_{MDOD}$  |           | 1.1       |      | 5.4  | ns   |

### (2) External SDRAM capacitance case 2

#### External SDRAM capacitance

| SDRAM x1         | Total capacitance                           | Unit |
|------------------|---|------|
| MCLKO            | 24.8pF (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 | $t_{DID}$   |           | 0         |      | 3.5  | ns   |
| MA, MRAS, MCAS, MWE Access time       | $t_{MAD}$   |           | 1.0       |      | 5.2  | ns   |
| MDQM Access time                      | $t_{MDQMD}$ |           | 1.2       |      | 5.5  | ns   |
| MD Output access time                 | $t_{MDOD}$  |           | 1.2       |      | 5.5  | ns   |

**(3) External SDRAM capacitance case 3****External SDRAM capacitance**

| SDRAM x2         | Total capacitance                               | Unit |
|------------------|---|------|
| MCLKO            | 12.3pF (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 | $t_{DID}$   |           | 0         |      | 4.1  | ns   |
| MA, MRAS, MCAS, MWE Access time       | $t_{MAD}$   |           | 1.0       |      | 5.0  | ns   |
| MDQM Access time                      | $t_{MDQMD}$ |           | 1.1       |      | 5.2  | ns   |
| MD Output access time                 | $t_{MDOD}$  |           | 1.1       |      | 5.2  | ns   |

**(4) External SDRAM capacitance case 4****External SDRAM capacitance**

| SDRAM x2         | Total capacitance                              | Unit |
|------------------|--|------|
| MCLKO            | 28.8pF (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 | $t_{DID}$   |           | 0         |      | 3.4  | ns   |
| MA, MRAS, MCAS, MWE Access time       | $t_{MAD}$   |           | 1.1       |      | 5.4  | ns   |
| MDQM Access time                      | $t_{MDQMD}$ |           | 1.1       |      | 5.5  | ns   |
| MD Output access time                 | $t_{MDOD}$  |           | 1.1       |      | 5.5  | ns   |

**(5) External SDRAM capacitance case 5****External SDRAM capacitance**

| <b>SDRAM x4</b>  | <b>Total capacitance</b>                        | <b>Unit</b> |
|------------------|---|-------------|
| MCLKO            | 17.3pF (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**

| <b>Parameter</b>                      | <b>Symbol</b> | <b>Condition</b> | <b>Rating *1</b> |             |             | <b>Unit</b> |
|---------------------------------------|---------------|------------------|------------------|-------------|-------------|-------------|
|                                       |               |                  | <b>Min.</b>      | <b>Typ.</b> | <b>Max.</b> |             |
| MCLKI signal delay time against MCLKO | $t_{DID}$     |                  | 0                |             | 3.9         | ns          |
| MA, MRAS, MCAS, MWE Access time       | $t_{MAD}$     |                  | 1.0              |             | 5.2         | ns          |
| MDQM Access time                      | $t_{MDQMD}$   |                  | 1.0              |             | 5.0         | ns          |
| MD Output access time                 | $t_{MDOD}$    |                  | 1.0              |             | 5.0         | ns          |

**(6) External SDRAM capacitance case 6****External SDRAM capacitance**

| <b>SDRAM x4</b>  | <b>Total capacitance</b>                       | <b>Unit</b> |
|------------------|--|-------------|
| MCLKO            | 36.8pF (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**

| <b>Parameter</b>                      | <b>Symbol</b> | <b>Condition</b> | <b>Rating *1</b> |             |             | <b>Unit</b> |
|---------------------------------------|---------------|------------------|------------------|-------------|-------------|-------------|
|                                       |               |                  | <b>Min.</b>      | <b>Typ.</b> | <b>Max.</b> |             |
| MCLKI signal delay time against MCLKO | $t_{DID}$     |                  | 0                |             | 3.4         | ns          |
| MA, MRAS, MCAS, MWE Access time       | $t_{MAD}$     |                  | 1.2              |             | 5.7         | ns          |
| MDQM Access time                      | $t_{MDQMD}$   |                  | 1.0              |             | 5.3         | ns          |
| MD Output access time                 | $t_{MDOD}$    |                  | 1.0              |             | 5.3         | ns          |

**(7) External SDRAM capacitance case 7****External SDRAM capacitance**

| <b>SDRAM x1</b> | <b>Total capacitance</b> | <b>Unit</b> |
|-----------------|--------------------------|-------------|
|-----------------|--------------------------|-------------|

|                  |   |    |
|------------------|---|----|
| MCLKO            | 10.0pF (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 | $t_{DID}$   |           | 0         |      | 4.2  | ns   |
| MA, MRAS, MCAS, MWE Access time       | $t_{MAD}$   |           | 1.0       |      | 5.0  | ns   |
| MDQM Access time                      | $t_{MDQMD}$ |           | 1.1       |      | 5.4  | ns   |
| MD Output access time                 | $t_{MDOD}$  |           | 1.1       |      | 5.4  | ns   |

**(8) External SDRAM capacitance case 8****External SDRAM capacitance**

| SDRAM x1         | Total capacitance                           | Unit |
|------------------|---|------|
| MCLKO            | 25.0pF (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 | $t_{DID}$   |           | 0         |      | 3.5  | ns   |
| MA, MRAS, MCAS, MWE Access time       | $t_{MAD}$   |           | 1.0       |      | 5.2  | ns   |
| MDQM Access time                      | $t_{MDQMD}$ |           | 1.2       |      | 5.5  | ns   |
| MD Output access time                 | $t_{MDOD}$  |           | 1.2       |      | 5.5  | ns   |

**(9) External SDRAM capacitance case 9****External SDRAM capacitance**

| SDRAM x2         | Total capacitance                               | Unit |
|------------------|---|------|
| MCLKO            | 12.5pF (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 | $t_{DID}$   |           | 0         |      | 4.1  | ns   |
| MA, MRAS, MCAS, MWE Access time       | $t_{MAD}$   |           | 1.0       |      | 5.0  | ns   |
| MDQM Access time                      | $t_{MDQMD}$ |           | 1.1       |      | 5.2  | ns   |
| MD Output access time                 | $t_{MDOD}$  |           | 1.1       |      | 5.2  | ns   |

**(10) External SDRAM capacitance case 10****External SDRAM capacitance**

| SDRAM x2         | Total capacitance                              | Unit |
|------------------|--|------|
| MCLKO            | 29pF (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 | $t_{DID}$   |           | 0         |      | 3.4  | ns   |
| MA, MRAS, MCAS, MWE Access time       | $t_{MAD}$   |           | 1.1       |      | 5.4  | ns   |
| MDQM Access time                      | $t_{MDQMD}$ |           | 1.1       |      | 5.5  | ns   |
| MD Output access time                 | $t_{MDOD}$  |           | 1.1       |      | 5.5  | ns   |

(11) External SDRAM capacitance case 11

External SDRAM capacitance

| SDRAM x4         | Total capacitance                               | Unit |
|------------------|---|------|
| MCLKO            | 17.5pF (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   |

(12) External SDRAM capacitance case 12

External SDRAM capacitance

| SDRAM x4         | Total capacitance                              | Unit |
|------------------|--|------|
| MCLKO            | 37.0pF (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   |

## 12.5.5 PLL specifications

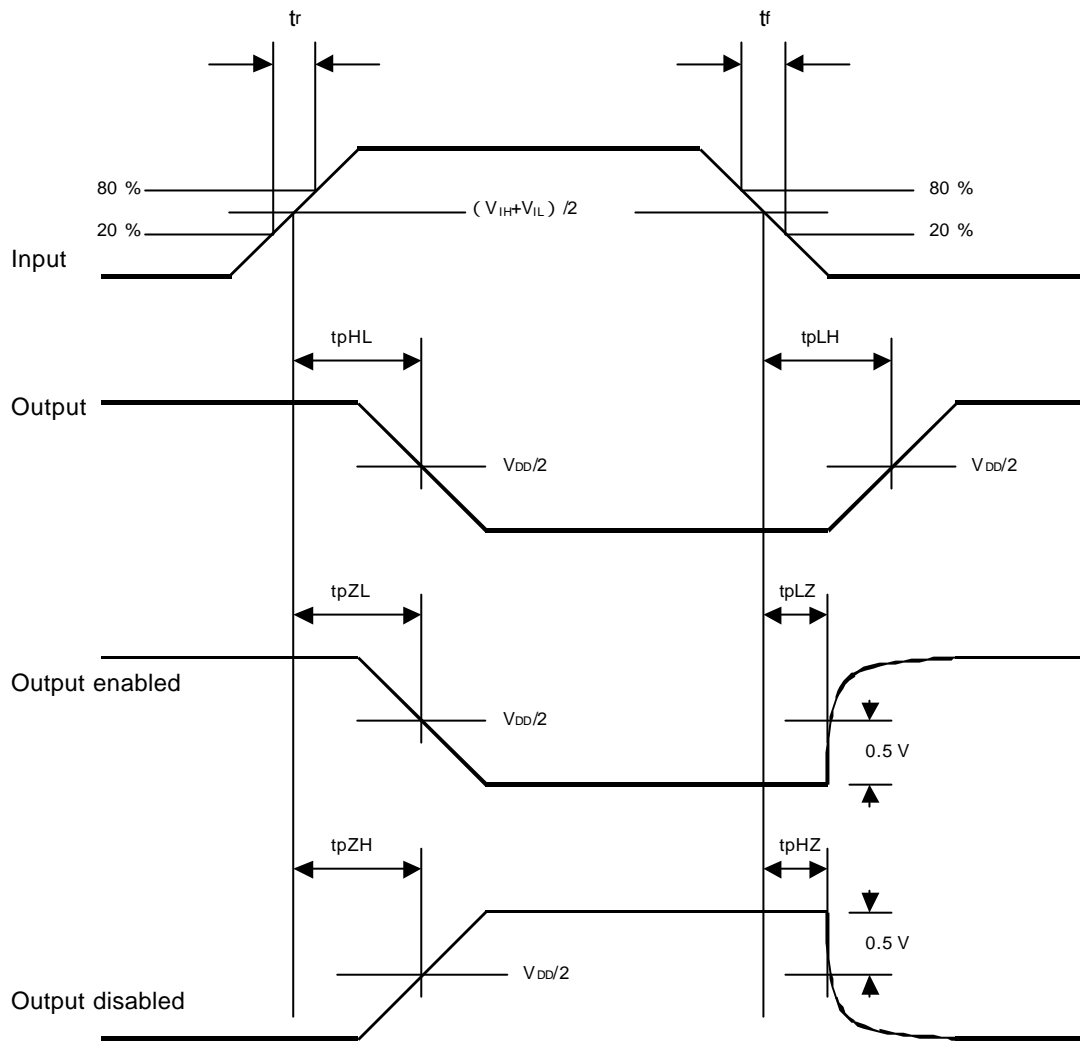
| Parameter              | Rating         | Description  |
|------------------------|----------------|--|
| Input frequency (typ.) | 14.31818 MHz   |  |
| Output frequency       | 400.9090 MHz   | × 28   |
| Duty ratio             | 101.6 to 93.0% | H/L Pulse width ratio of PLL output                |
| 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



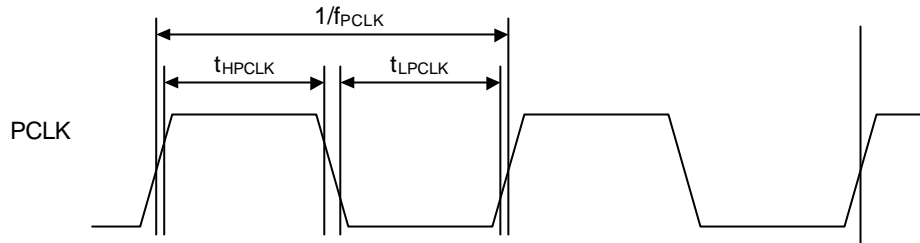
$T_r, t_f \leq 5 \text{ ns}$

$V_{IH} = 2.0 \text{ V}, V_{IL} = 0.8 \text{ V}$  (3.3-V CMOS interface input)

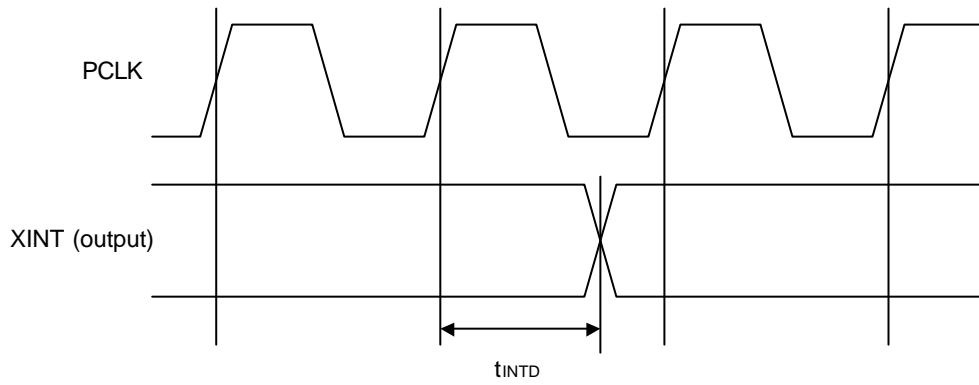
## 12.7 Timing Diagram

### 12.7.1 Host interface

#### Clock

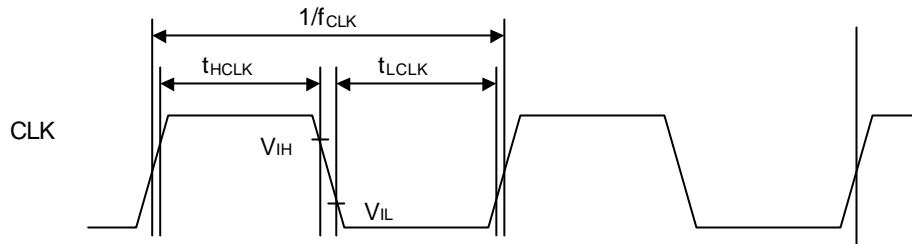


#### XINT output delay times

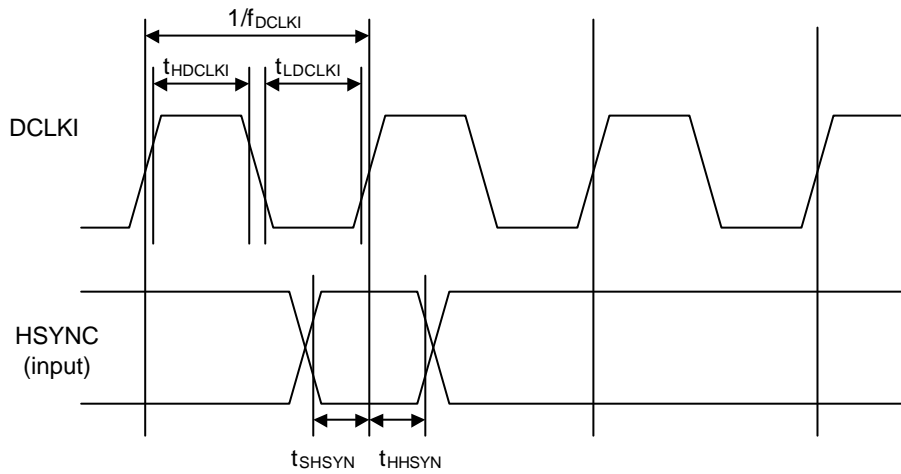


## 12.7.2 Video interface

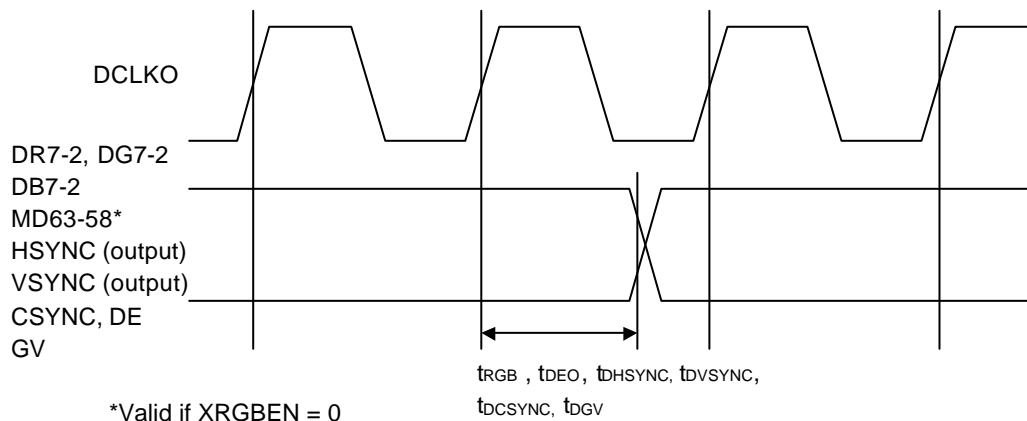
### Clock



### HSYNC signal setup/hold

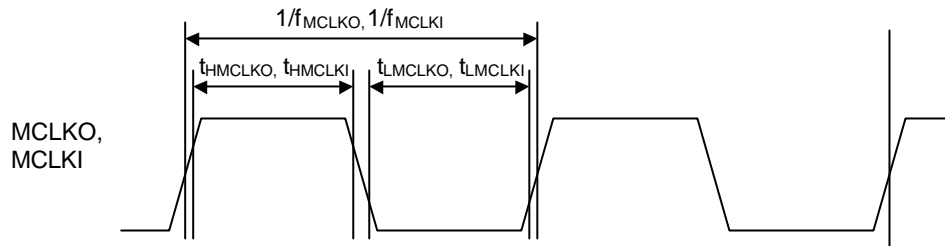


### Output signal delay

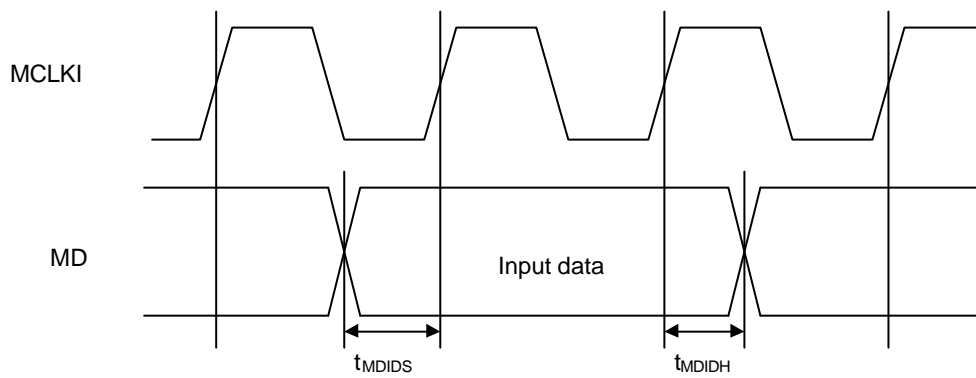


### 12.7.3 Graphics memory interface

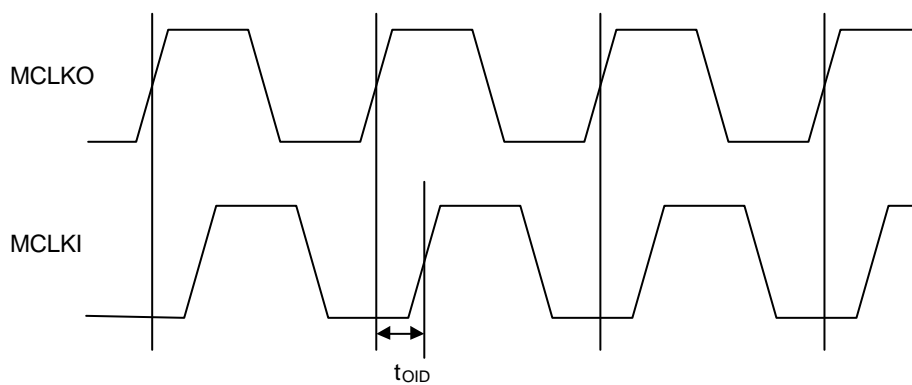
#### Clock



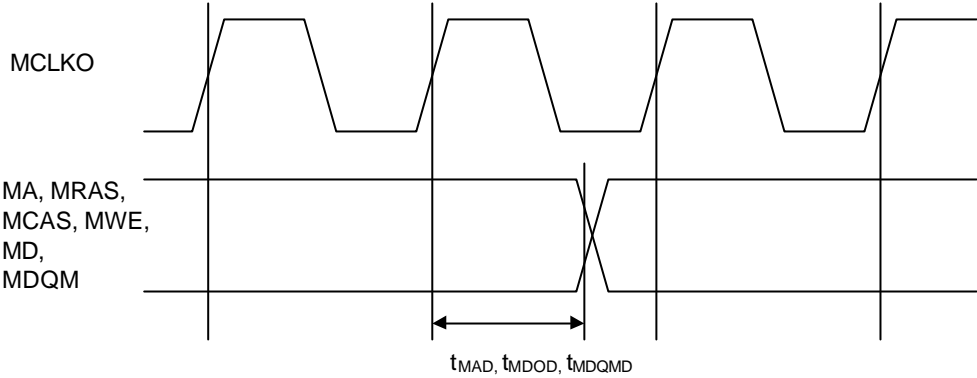
#### Input signal setup/hold time



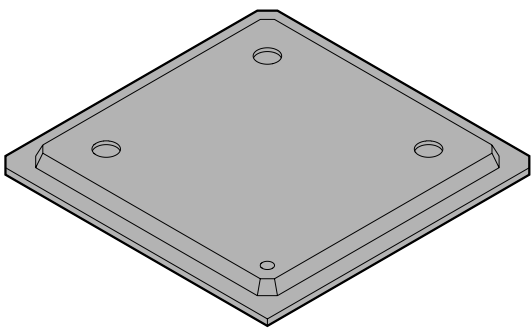
#### MCLKI signal delay

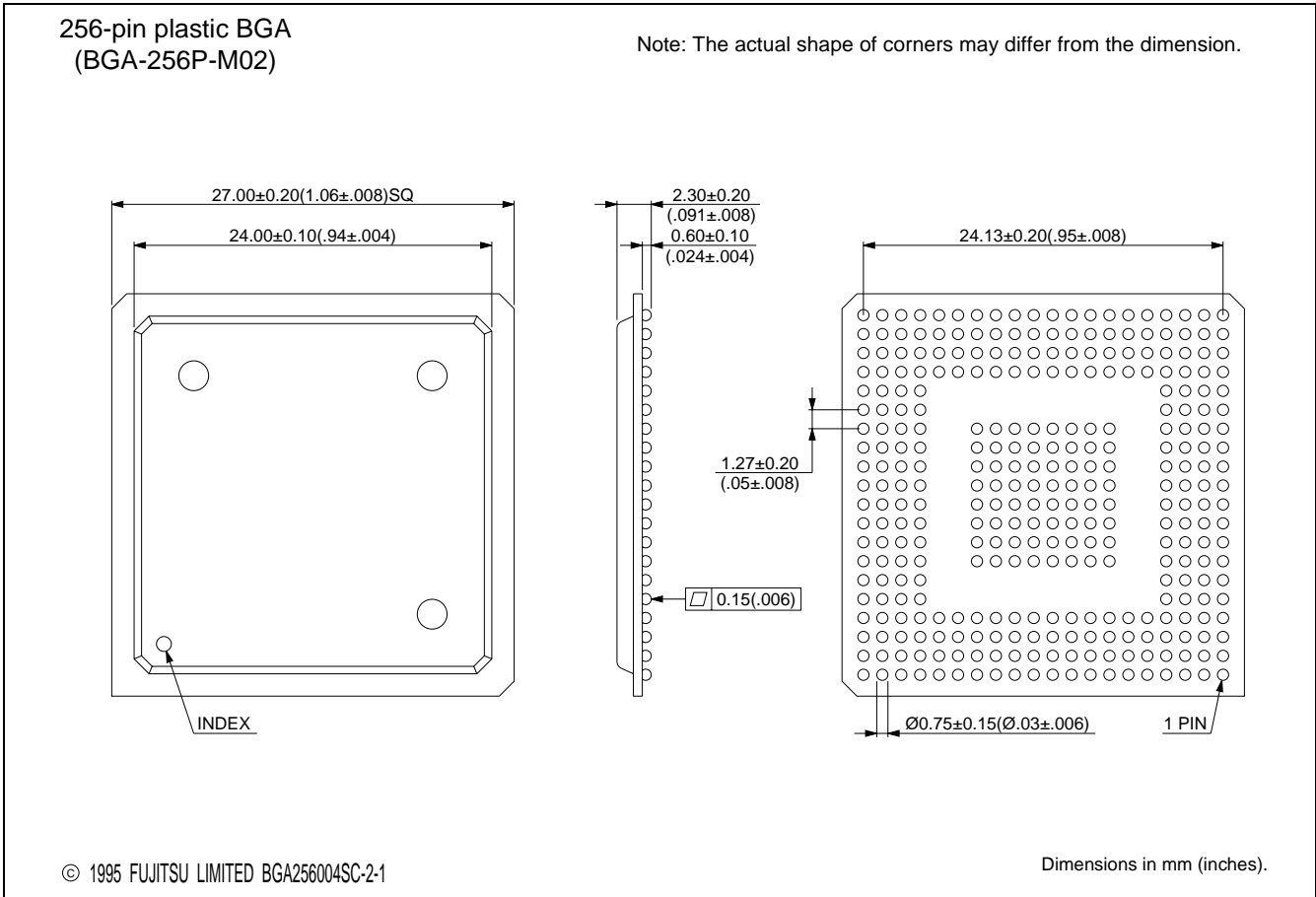


Output signal delay



**BGA-256P-M02**

|  |                |              |
|--|----------------|--------------|
| <p>256-pin plastic BGA</p>  <p>(BGA-256P-M02)</p> | Lead pitch     | 50 mil       |
|  | Pin matrix     | 20           |
|  | Sealing method | Plastic mold |
|  |                |              |
|  |                |              |
|  |                |              |
|  |                |              |



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.