MB86296S <CORAL PA>
PCI Graphics Controller Specification

Revision 1.1

25 September, 2007

FUJITSU

<Notes>

- 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.

⚠ CAUTION

Burns There is a danger of burns because the IC surface is heated depending on the IC operating conditions. In this case, take safety measures.

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
28.2.2005	1.0	346	First edition
25.9.2007	1.1	356	Correct MMR Register (RAW > SAW) and some lingual improvements

CONTENTS

1. GENERAL

1.1 Preface

The MB86296S <CORAL-PA> is a graphics controller with a PCI host interface.

Note:

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

1.2 Features

- Geometry Engine

  The Geometry Engine supports the geometry processing that is basically compatible**1 with ORCHID (MB86292). Display lists generated for ORCHID can be processed. Extensive geometric operation processing such as coordinate conversions or clipping which normally load the CPU dramatically can be reduced using the Geometry Engine. **1 (Floating point setup command is changed or deleted. G_BeginCont command is deleted. GMDR0 CF&DF table mapping is changed ... etc)

- 2D and 3D Drawing

  MB86296's drawing functionality is compatible to the CREMSON (MB86290A). It can draw data using the display lists created for CREMSON (however internal texture RAM is deleted).

  The MB86296 also supports 3D rendering, such as texture mapping with perspective correction 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 images in graphics memory; it can display drawn images and video images on the same screen.

- Display controller

  The MB86296 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, 6-layer overlay functionality, 4-siding for palette are expanded.

- Host CPU interface

  The MB86296 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)

  Current consumption (TYPICAL)

  1.8 V : 500mA

  3.3 V : 100mA

1.3 Block Diagram

The CORAL-PA general block diagram is shown below:

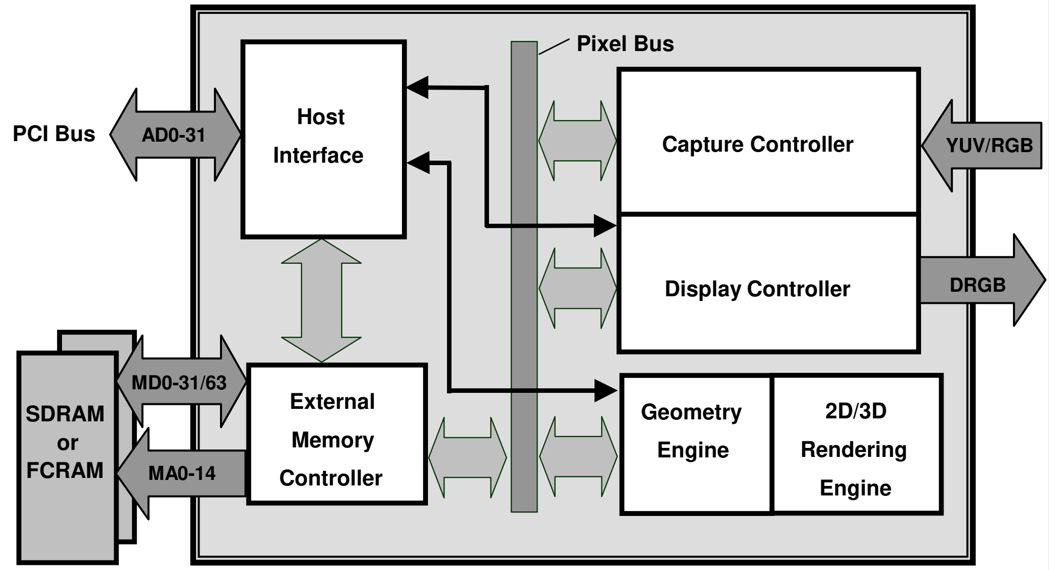


Fig.1.1 CORAL-PA Block Diagram

1.4 Functional Overview

1.4.1 Host CPU interface

Supported CPU

The MB86296 can be connected to any CPU with a 33MHz 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

Register update

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 (x16 Bits)	32 Bits	2	4 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 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 256 Mbits (x16 Bits)	32 Bits	2	64 Mbytes

1.4.3 Display controller

Video data output

Analog RGB video output is provided as well as 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

MB86296S 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

The 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).

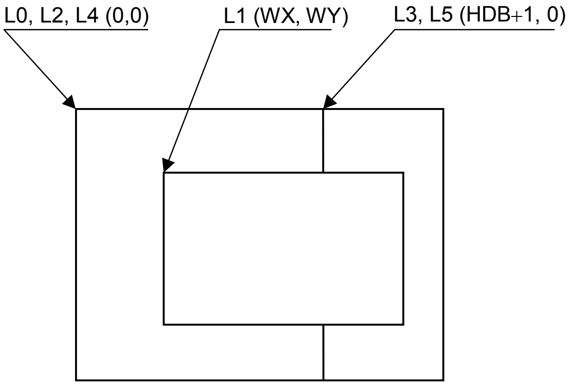
The transparent mode or the blend mode can be selected for overlay.

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

Window display can be performed for the W-layer.

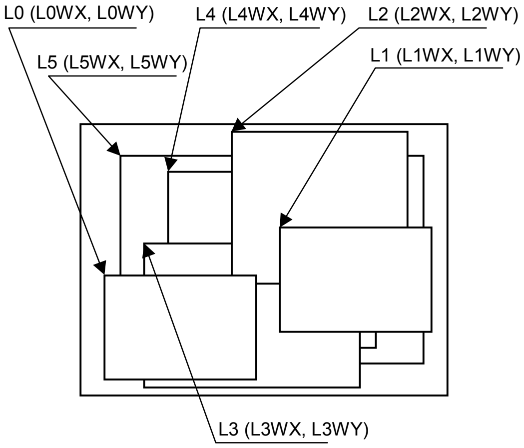
Two palettes are provided: C-layer and M-/B-layer.

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 RGB666.
- 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 MB86296 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 × 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

Backfacing triangles 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

MB86296S can perform 2D drawing for graphics memory (drawing plane) in direct color mode or indirect color mode.

Wide bold lines and broken lines can be drawn. Smooth diagonal lines can also be drawn using anti-aliasing.

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). With 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 functions can be used. Only vertex coordinates are set for these primitives. High-speed 2DTriangle is also used to draw polygons.

2D Primitives

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

Arbitrary polygon drawing

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

BLT/Rectangle drawing

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

BLT Attributes

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

Pattern (Text) drawing

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

Pattern (Text) Drawing Attributes

Attribute	Description
Enlarge	Vertically × 2 Horizontally × 2 Vertically and Horizontally × 2
Shrink	Vertically × 1/2 Horizontally 1/2 Vertically and Horizontally 1/2

Drawing clipping

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

1.4.7 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

MB86296S 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

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

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

Shading

Supports the shading primitive.

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

Alpha blending

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

There are two ways of specifying alpha blending for drawing:

(1) Set a transparent coefficient to the register; the transparent coefficient is applied for transparency processing of one plane.

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

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

Alpha Blending

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

Gouraud Shading

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

Gray Scale Gouraud Shading

Gray scale gouraud shading can be used in the in-direct color mode to draw a blend coefficient layer.

Texture mapping

MB86296 supports texture mapping to map a image pattern onto the surface of plane. For 2D pattern texture mapping, MB86296 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-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 decal can be specified for the blend mode.

Texture Mapping

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

1.4.9 Others

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

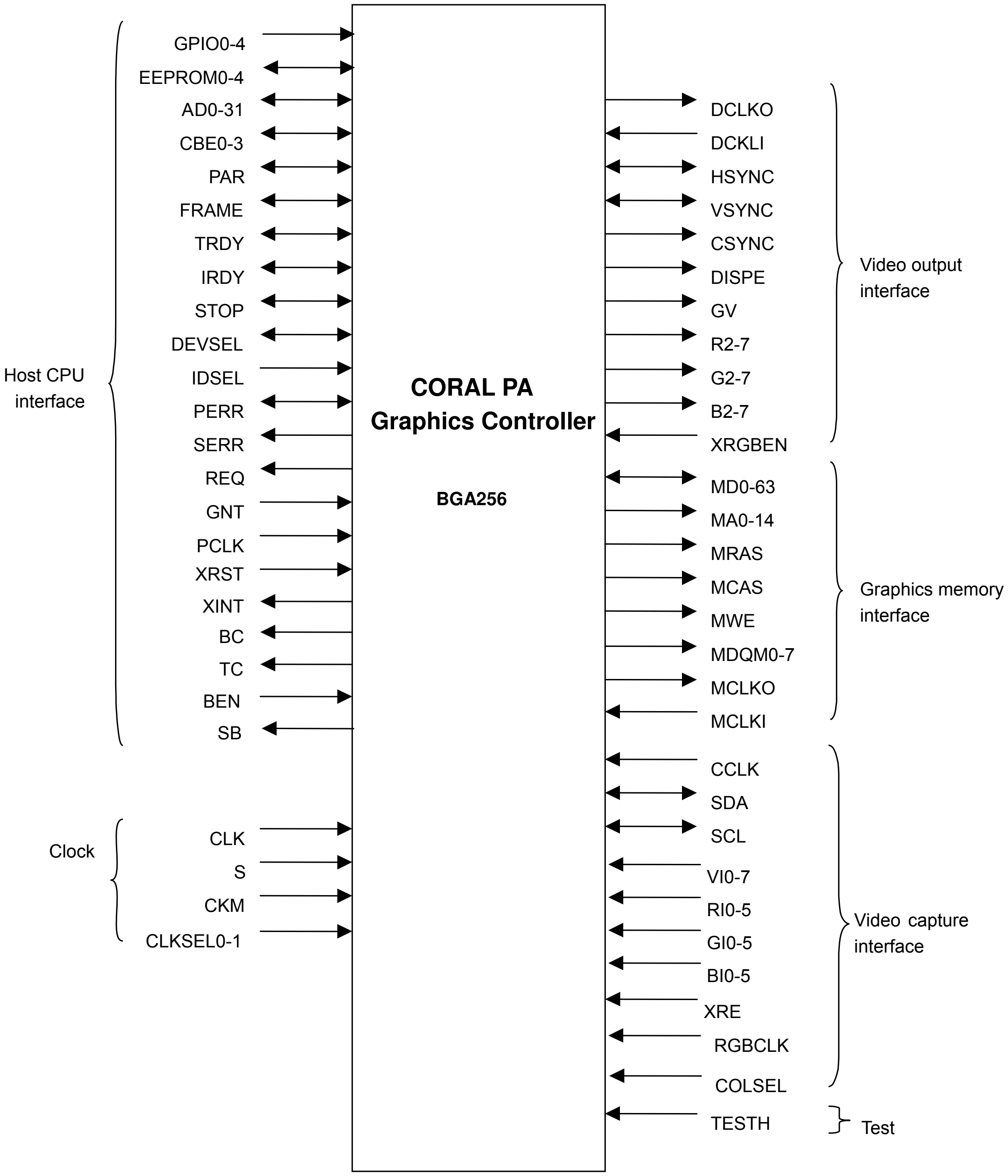
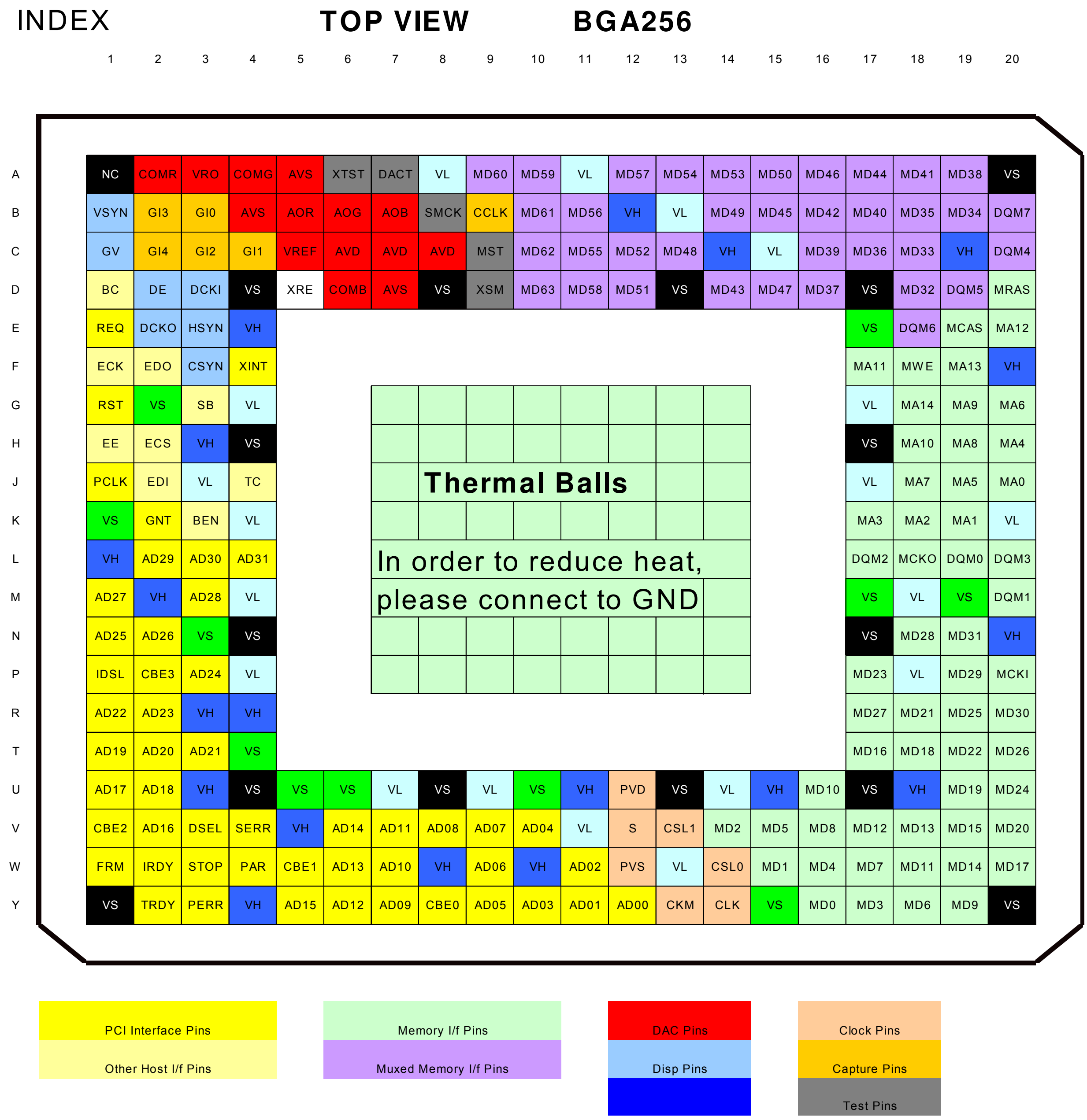


Fig. 2.1 CORAL PA Signal Lines

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
A	NC	COMR	VRO	COMG	AVS	XTST	DACT	VL	MD60	MD59	VL	MD57	MD54	MD53	MD50	MD46	MD44	MD41	MD38	VS
B	VSYN	GI3	GI0	AVS	AOR	AOG	AOB	SMCK	CCLK	MD61	MD56	VH	VL	MD49	MD45	MD42	MD40	MD35	MD34	DQM7
C	GV	GI4	GI2	GI1	VREF	AVD	AVD	AVD	MST	MD62	MD55	MD52	MD48	VH	VL	MD39	MD36	MD33	VH	DQM4
D	BC	DE	DCKI	VS	XRE	COMB	AVS	VS	XSM	MD63	MD58	MD51	VS	MD43	MD47	MD37	VS	MD32	DQM5	MRAS
E	REQ	DCKO	HSYN	VH													VS	DQM6	MCAS	MA12
F	ECK	EDO	CSYN	XINT													MA11	MWE	MA13	VH
G	RST	VS	SB	VL													VL	MA14	MA9	MA6
H	EE	ECS	VH	VS													VS	MA10	MA8	MA4
J	PCLK	EDI	VL	TC													VL	MA7	MA5	MA0
K	VS	GNT	BEN	VL													MA3	MA2	MA1	VL
L	VH	AD29	AD30	AD31													DQM2	MCKO	DQM0	DQM3
M	AD27	VH	AD28	VL													VS	VL	VS	DQM1
N	AD25	AD26	VS	VS													VS	MD28	MD31	VH
P	IDSL	CBE3	AD24	VL													MD23	VL	MD29	MCKI
R	AD22	AD23	VH	VH													MD27	MD21	MD25	MD30
T	AD19	AD20	AD21	VS													MD16	MD18	MD22	MD26
U	AD17	AD18	VH	VS	VS	VS	VL	VS	VL	VS	VH	PVD	VS	VL	VH	MD10	VS	VH	MD19	MD24
V	CBE2	AD16	DSEL	SERR	VH	AD14	AD11	AD08	AD07	AD04	VL	S	CSL1	MD2	MD5	MD8	MD12	MD13	MD15	MD20
W	FRM	IRDY	STOP	PAR	CBE1	AD13	AD10	VH	AD06	VH	AD02	PVS	VL	CSL0	MD1	MD4	MD7	MD11	MD14	MD17
Y	VS	TRDY	PERR	VH	AD15	AD12	AD09	CBE0	AD05	AD03	AD01	AD00	CKM	CLK	VS	MD0	MD3	MD6	MD9	VS

Thermal Balls

In order to reduce heat, please connect to GND

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].
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 (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 (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^2C serial data (SDA).
C	11	MD55	I/O	Graphics memory data bit 55. May also be configured as I^2C 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 (VI0). 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 (VI1). 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 (VI2). 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^2C 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}/PLLV_{SS} : Ground
V_{DDH} : 3.3-V power supply
V_{DDL}/PLLV_{DD} : 1.8-V power supply
PLLV_{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 PLLV_{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 (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 (Open Drain)	Interrupt
BC	Output	Burst Complete. Indicates a burst is complete when using the DMA/Burst Controller. 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. This may also be configured as a GPIO Input/Output. 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. This pin may also be configured as a GPIO Input/Output. 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. 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. 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 clock Output 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 Horizontal sync input <in external sync mode>
VSYN	I/O	Vertical sync signal output 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 resister.

When not using DAC, it is possible to connect all of analog pins(AVD, AOUTR,G,B, ACOMPR,G,B, VREF, VRO) to GND.

The 16bit/pixel color mode and 8bit/pixel color mode are converted to digital R:G:B=8:8:8 as the below.

A) 16bit/pixel color mode

R:G:B=5:5:5 data in graphics memory		Digital R:G:B=8:8:8
0		0
1-31		Add 111 to lower 3bits Formula=X*8+7

B) 8bit/pixel color mode

R:G:B=6:6:6 data in color palette		Digital R:G:B=8:8:8
0		0
1-63		Add 11 to lower 2bits Formula=X*4+3

The Y,Cb,Cr mode is converted to R:G:B=8:8:8 directly.

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
VI7-0	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.

Direct Input Mode

Pin name	I/O	Description
RGBCLK	Input	Clock for RGB input. This pin is multiplexed CCLK.
RI5-0	Input	Red component value. These pins are multiplexed EE, EDI, ECS, ECK, EDO and BC.
GI5-0	Input	Green component value. These pins are multiplexed SB and GPI4-GPI0.
BI5-0	Input	Blue component value. These pins are multiplexed MD61-MD56.
VSYNCI	Input	Vertical sync for RGB capture. This pin is multiplexed MD63.
HSYNCI	Input	Horizontal sync for RGB capture. This pin is multiplexed MD62.

Note :

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

2.3.4 I^{2}C interface

Pin name	I/O	Description
SDA	I/O	I^{2}C or Video capture test signal. This pin is multiplexed MD54.
SCL	I/O	I^{2}C or Video capture test signal. This pin is multiplexed MD55.

I^{2}C interface signals can be used only when the XRE pin is “0”. MD55-MD54 are assigned as the I^{2}C interface pins.

When I^{2}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. Even if don't use an internal PLL (use BCLKI as internal clock and use DCLKI as dot clock), don't stop the PLL (Not fixed the S pin to low level).

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	Inputs 33.33-MHz clock frequency	× 12	399.96

Please connect the crystal oscillator directly with the terminal CLK.

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 15.3.2***.

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. PROCEDURE OF THE HARDWARE INITIALIZATION

3.1. Hardware reset

1.Do the hardware reset. (see section 15.3.2)

2.After the hardware reset, set the CCF(Change of Frequency) register (section 13.2.1).
In being unstable cycle after the hardware reset, keep 32 bus cycles open.

3.Set the graphics memory interface register, MMR (Memory I/F Mode Register).
After setting the CCF register, take 200 us to set the MMR register.
In being unstable memory access cycle, keep 32 bus cycles open.

4.Other registers, except for the CCF register and the MMR register, should be set after setting the CCF register.
In case of not using memory access, the MMR register could be set in any order after the CCF register is set.

3.2. Re-reset

1. Reset XRST signal.

2. See section 3.1 for registers setting after the procedure of re-reset.

3.3. Software reset

1. Set the value of the SRST register (see section 13.2.1) for re-reset.

2. It is not necessary to reset the CCF register and the MMR register again.

4. HOST INTERFACE

The Coral PA 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 PA 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.

4.1 Standard PCI Slave Accesses

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

4.1.1 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 16 dwords deep. PCI slave writes of any size are supported but typically a retry will occur after each 16 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.

Coral PA does not perform any fast back to back transactions.

4.1.2 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.

Note: Data is not guaranteed when the burst transfer more than the burst length set as the SRBS register is performed.

4.2 Burst Controller Accesses (including PCI Master)

The Coral PA 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 PA 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. However, as for burst length, it is set to 2-8 only at the time of “Slave Mode Coral PA to PCI” mentioned later. 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 PA 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.

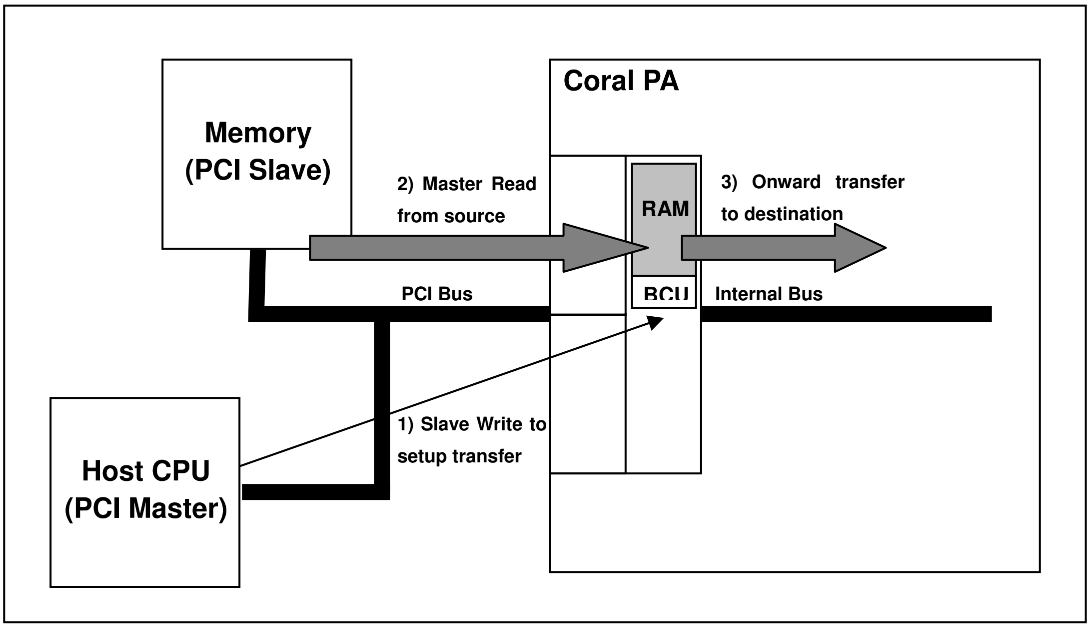
4.2.1 Transfer Modes

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

Mode	Function
000b	Slave Mode PCI to Coral PA. 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 PA 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 PA to PCI. In this mode the burst controller reads data from a Coral PA 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 PA to Coral PA. In this mode data is read from a source address internal to Coral PA into a temporary buffer, from where it is written to a destination, also internal to Coral PA. 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 PA (PCI Master read). In this mode the source address is in PCI data space and the destination address internal to Coral PA. 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 PA to PCI (PCI Master write). In this mode the source address is internal to Coral PA 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.

When Coral PA is master (bcu mode: 100b, 101b, 110b), Coral PA cannot issue an odd address to PCI area. If the beginning address is set to the odd address in 64-bit boundary, Coral PA issues the previous even address. Note: The odd address in 64-bit boundary means 0x004, 0x00C, 0x014….

In master read, Coral PA begins to read the previous even address and read the setting of burst size (BSIZE of BCR register) plus 1.

In master write, Coral PA begins to write the previous even address with disable write byte enable and write the setting of burst size (BSIZE of BCR register) plus 1.

4.2.2 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 though 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.

4.3 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.

4.4 GPIO/Serial Interface

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

4.4.1 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).

4.4.2 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.

4.5 Interrupt

The Coral PA MB86296 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
- Register update
- 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^2C 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^2C 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.

4.5.1 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.

4.6 Memory Map

The local memory base address of Coral-PA 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 PA to the host CPU memory space.

Note: Burst read which follows a Host interface registers from a Graphics memory domain and follows a Graphics memory domain from a Geometry Engine registers is prohibition.

Ex.)Bust size=8 don’t read 1fbffe4-1fbfffc and 1ffffe4-1ffffc

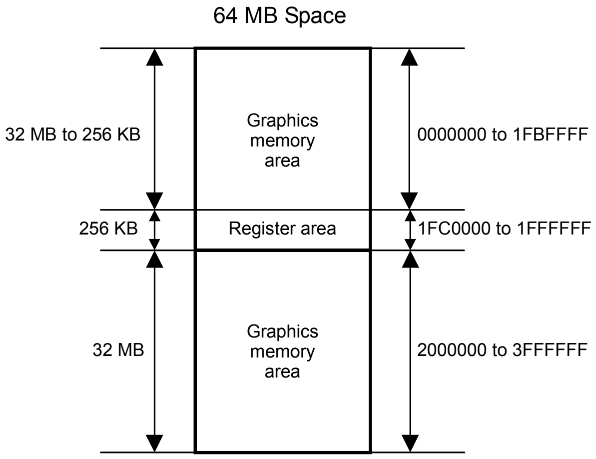


Fig. 3.1 Memory Map

Table 3-4 Address Space

Size	Resource	Base address	(Name)
32 MB to 256 KB	Graphics Memory	00000000	
64 KB	Host interface registers (I^2C interface registers)	01FC0000 (01FCC000)	(HostBase) (I^2CBase)
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 PA may be accessed after about 20 bus clocks after writing 1 to RSW.

Note: Burst read which follows a Host interface registers from a Graphics memory domain is prohibition.

Ex.)Bust size=8 don’t read 3fbffe4-3fbfffc

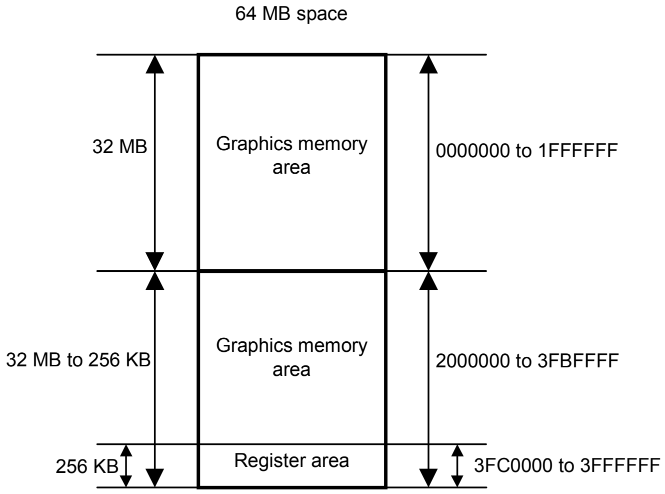


Fig. 3.2 Alternate Memory Map

Table 3-5 Alternate Address Mapping

Size	Resource	Base address	(Name)
64 MB to 256 KB	Graphics memory	00000000	
64 KB	Host interface registers (I^2C interface registers)	03FC0000 (03FCC000)	(HostBase) (I^2CBase)
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)

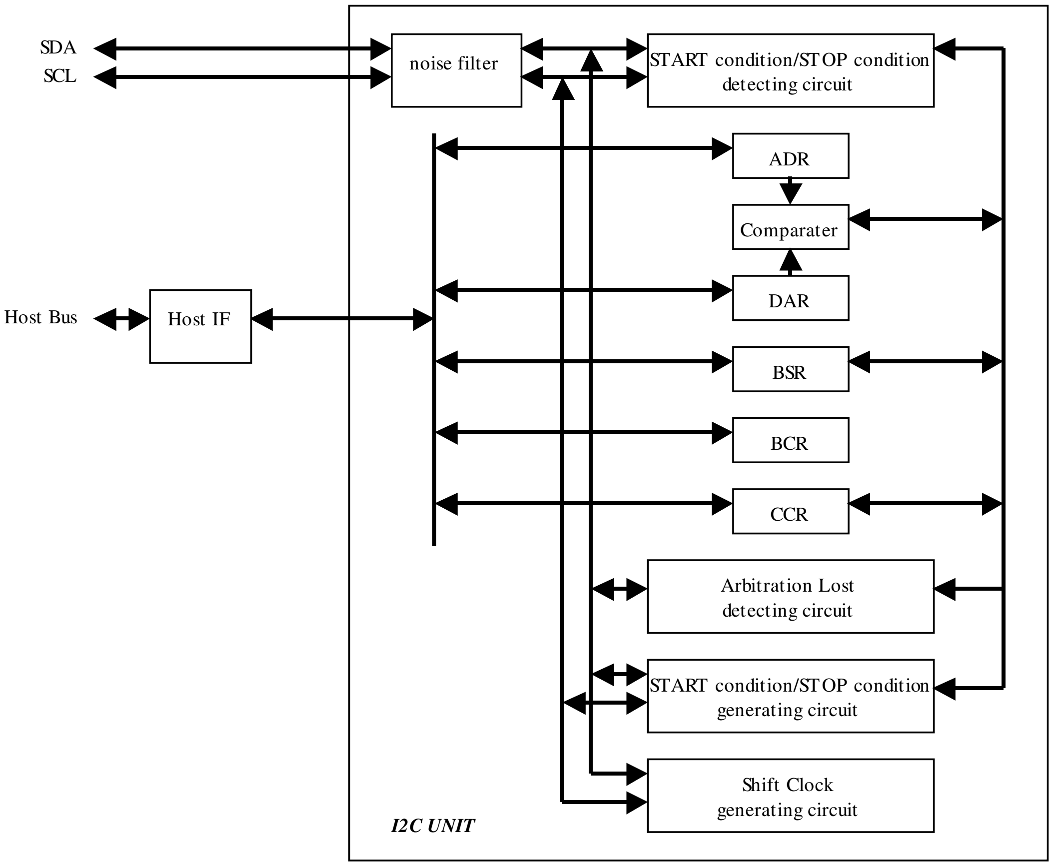
5. I^{2}C Interface Controller

5.1 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)

5.2 Block diagram

5.2.1 Block Diagram



5.2.2 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

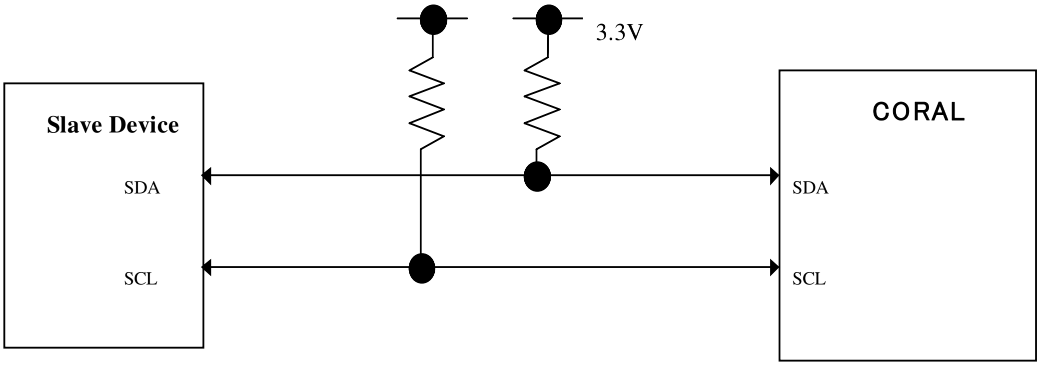
CCR is the 7-bit register used by serial data transfer. This register has following functions:
- permission of operation
- setup of a serial clock frequency
- selection of standard-mode / high-speed-mode

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.

5.3 Example application

5.3.1 Connection Diagram



5.4 Function overview

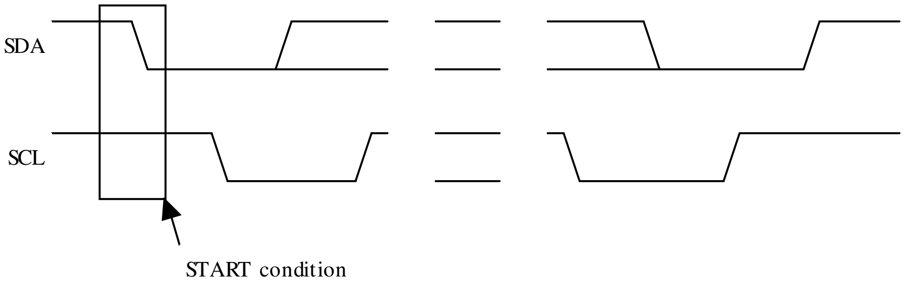
Two bi-directional buses, serial data line (SDA) and serial clock line (SCL), carry information at I2C-bus. Scarlet I2C interface has SDA input (SDAI) and SDA output (SDAO) for SDA and is connected to SDA line via open-drain I/O cell. And this interface also has SCL input (SCLI) and SCL output (SCLO)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

5.4.1 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.

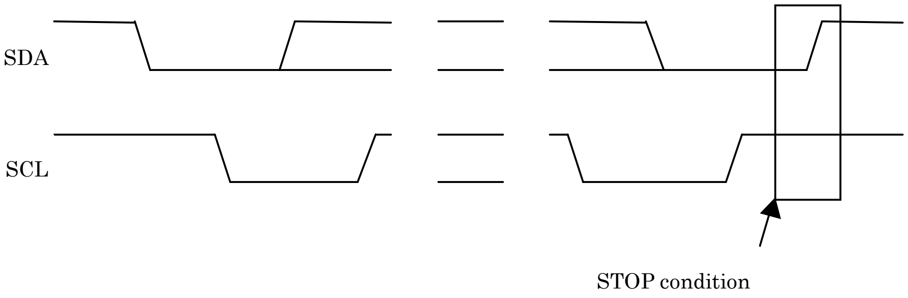
There are two conditions to generate START condition.
- “1” writing to MSS bit in the state where the bus is not used (MSS=0 & BB=0 & INT=0 & AL=0)
- “1” writing to SCC bit in the interruption state in a master mode (MSS=1 & BB=1 & INT=1 & AL=0)

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.



5.4.2 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.



5.4.3 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:

MSB							LSB	
A6	A5	A4	A3	A2	A1	A0	R/W	ACK

slave address

- **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 1XX	X	Reserved
0 0 0 1 XXX ⋮ 1 1 1 0 XXX	X	Available slave address
1 1 1 1 0 XX	X	10-bit slave addressing*1
1 1 1 1 1 XX	X	Reserved

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

5.4.4 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.

5.4.5 Arbitration

When other masters have transmitted data simultaneously at the time of master transmission, arbitration takes places. 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.

5.4.6 Acknowledge

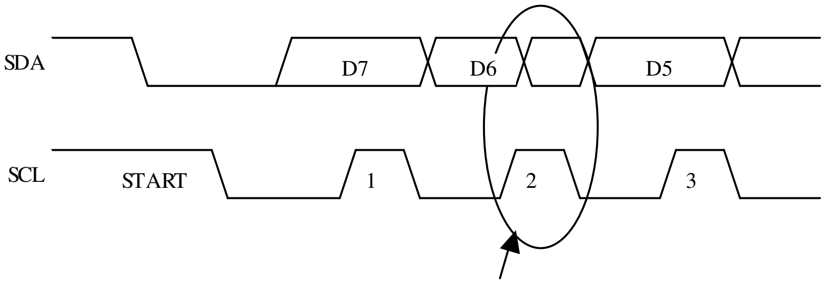
Acknowledge is transmitted from a reception side to a transmission side. At the time of data reception, acknowledge is stored in LRB bit by ACK bit.

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.

5.4.7 Bus error

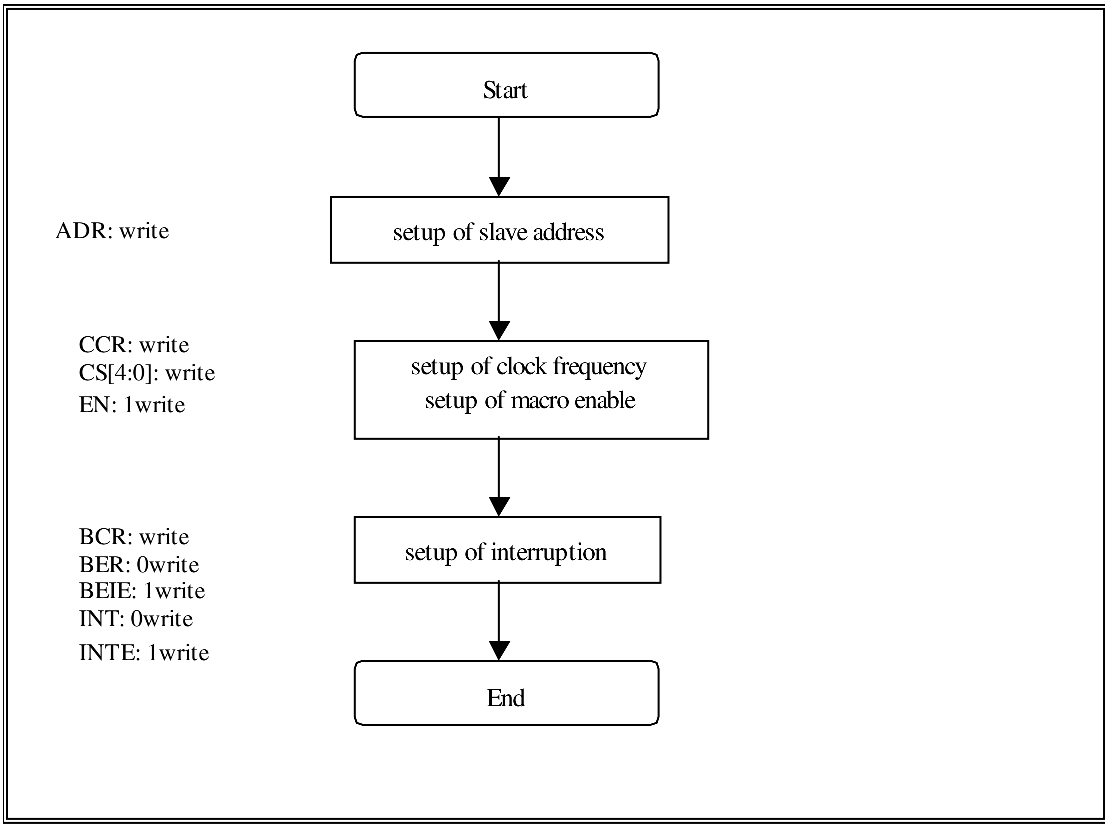
When the following conditions are satisfied, it is judged as a bus error, and this interface will be in a stop state.

- 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

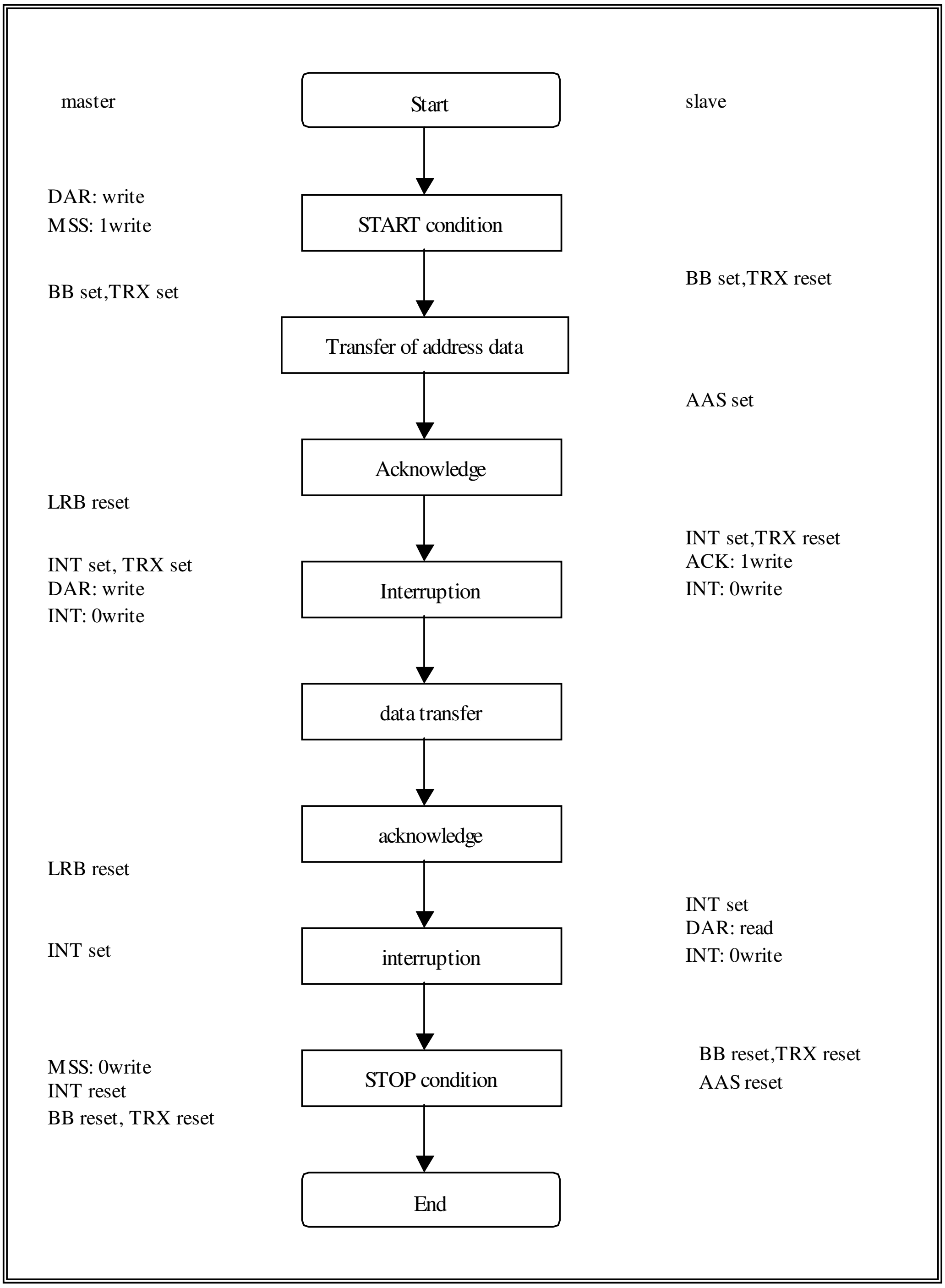


SDA changed under data transmission (SCL=H). It becomes bus error.

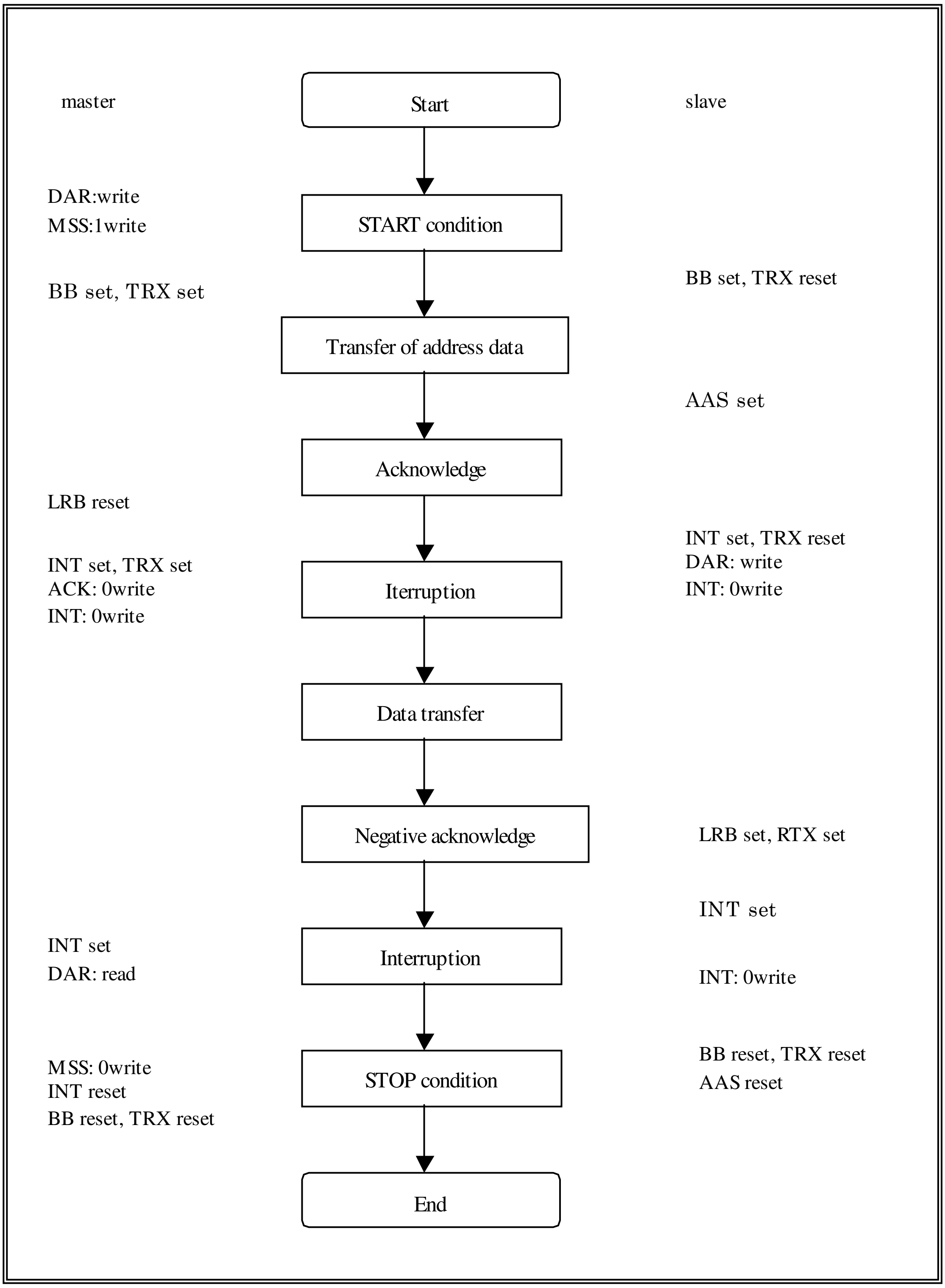
5.4.8 Initialize



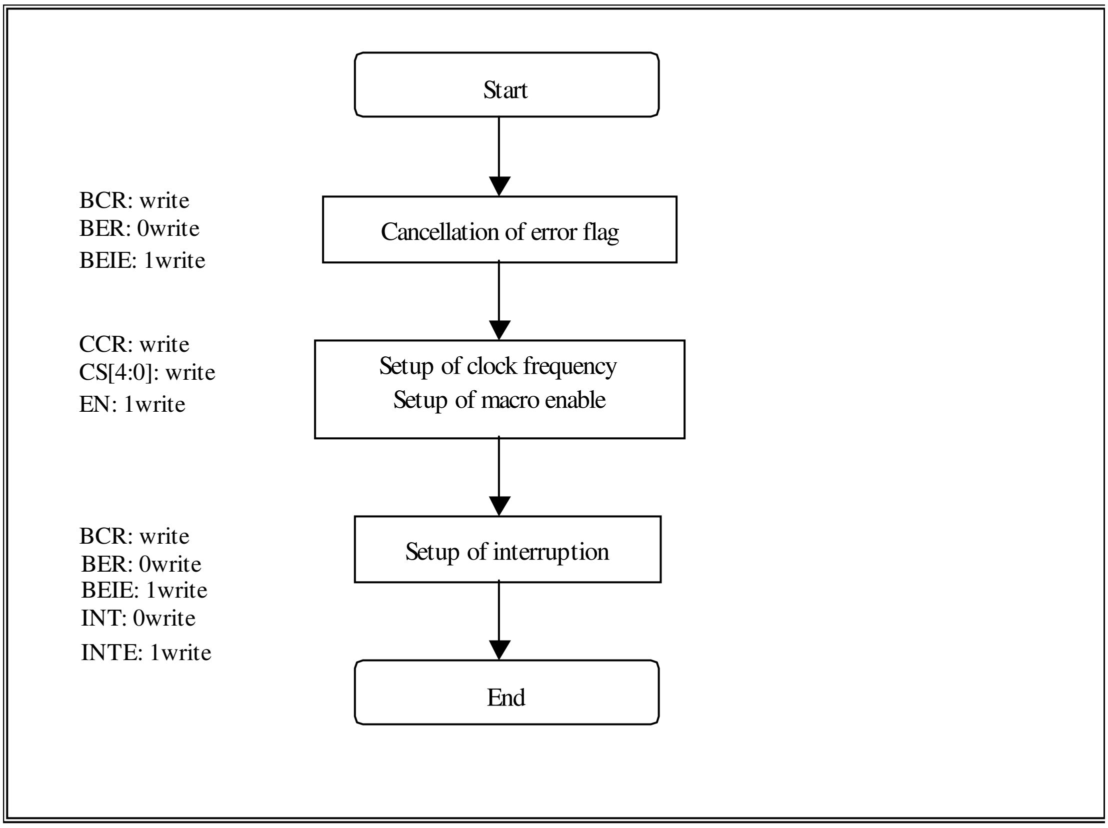
5.4.9 1-byte transfer from master to slave



5.4.10 1-byte transfer from slave to master



5.4.11 Recovery from bus error



5.5 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.

6. Graphics Memory

6.1. Configuration

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

6.1.1. Data type

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

Drawing Frame

This is a rectangular image data field for 2D/3D drawing. The Coral is able to have plural drawing frames and display a part of these area if it is set to be bigger than display size. The maximum size is 4096x4096 pixel in 32 pixel units. And both indirect color (8 bits / pixel) and direct color (16 bits / pixel) mode are applicable.

Display Frame

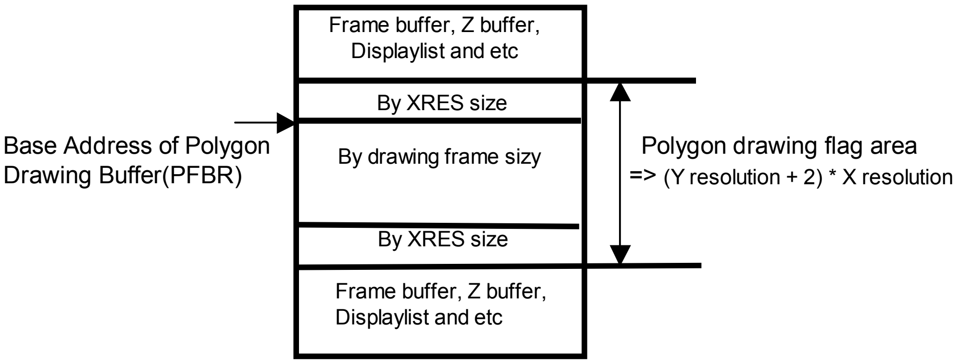
This is a rectangle picture area for display. The Coral is able to set display layer up to 6 layers.

Z Buffer

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

Polygon Drawing Flag Buffer

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



Specially, when you use Polygon with Shadow, required area is depending on geometory view volume clip parameter. (Normally depending on drawing clipping parameter) Above "Y resolution" is "Possible_view_clipped_Max_Ydc-Possible_view_clipped_Min_Ydc+1+6". (+6 mergin must be needed)

Displaylist Buffer

The displaylist is a list of drawing commands and parameters.

Texture Pattern

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

Cursor Pattern

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

6.1.2. Memory Mapping

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

6.1.3. Data Format

Direct Color (16 bits / pixel)

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

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

Indirect Color (8 bits / pixel)

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

7	6	5	4	3	2	1	0
Color Code							

Z Value

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

1) 16 bits mode

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

2) 8 bits mode

7	6	5	4	3	2	1	0
Unsigned Integer							

Polygon Drawing Flag

This data format is 1 bit per 1 pixel.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
P15	P14	P13	P12	P11	P10	P9	P8	P7	P6	P5	P4	P3	P2	P1	P0
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
P31	P30	P29	P28	P27	P26	P25	P24	P23	P22	P21	P20	P19	P18	P17	P16

Texture / Tile Pattern

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

1) Direct color mode (16 bits / pixel)

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

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

2) Indirect color mode (8 bits / pixel)

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

7	6	5	4	3	2	1	0
Color Code							

Cursor Pattern

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

7	6	5	4	3	2	1	0
Color Code							

Video Capture data

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

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y0								Cb							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Y1								Cr							

Direct Color (32 bits / pixel)

This data format is described RGB as each 8 bit. Bit31 is used for alpha bit of layer blending.

But the Coral does not support this color mode drawing. Therefore please draw this layer by CPU writing.

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

6.2. Frame Management

6.2.1. Single Buffer

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

6.2.2. Double Buffer

Two drawing frames are set. While one frame is displayed, drawing is done at the other frame. Flicker-less animation can be performed by flipping these two frames back and forth. Flipping is done in the blanking time period. There are two flipping modes: automatically at every scan frame period, and by user control. The double buffer is assigned independently for the L2, L3, L4, L5 layers.

6.3. Memory Access

6.3.1. Memory Access by host CPU

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

6.3.2. Priority of memory accessing

The priority of Graphics memory accessing is the follows:

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

6.4. Connection with memory

6.4.1. Connection with memory

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

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

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

64M bit SDRAM(x16 bit)

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

64M bit SDRAM(x32 bit)

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

128M bit SDRAM(x16 bit)

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

128M bit SDRAM(x32 bit)

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

256M bit SDRAM(x16 bit)

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

16M bit FCRAM(x16 bit)

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

7. DISPLAY CONTROLLER

7.1 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 × 240 to 1024 × 768).

Color look-up

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

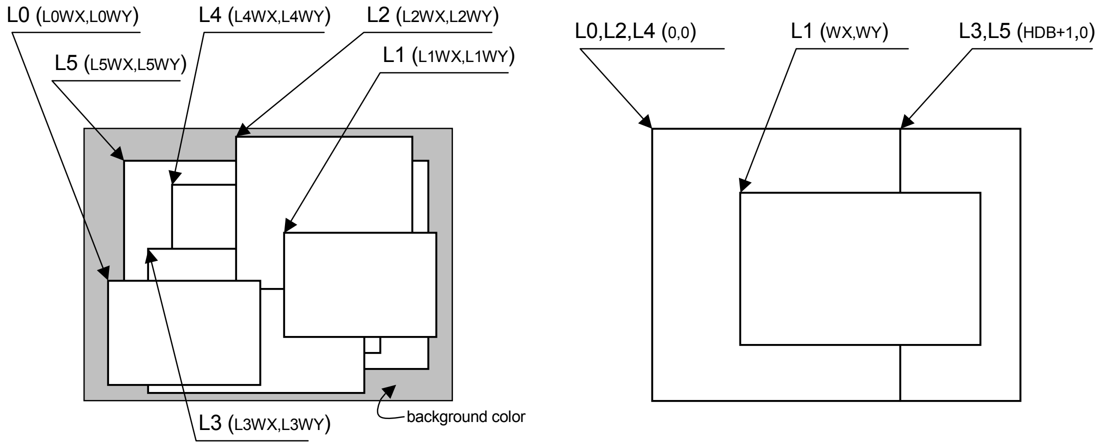
Cursor

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

7.2 Display Function

7.2.1 Layer configuration

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



(a) Six layerd window display

(b) Four layered display for downward compatibility

Configuration of Display Layers

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

Layer correspondence		Coordinates of starting point		Width/height	
		Window mode	Compatibility mode	Window mode	Compatibility mode
L0	C	(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)

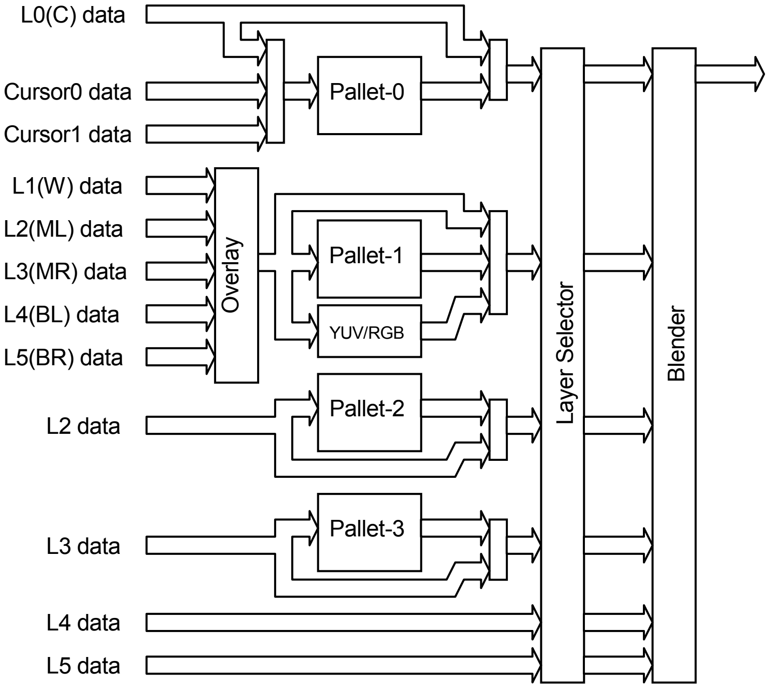
C, W, ML, MR, BL, and BR above mean layers for previous products. The window mode or the compatibility mode can be selected for each layer. It is possible to use new functions through minor program changes by allowing the coexistence of display modes instead of separating them completely.

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.

7.2.2 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.

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

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:

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

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, widow display mode and extend overlay mode.

7.2.3 Display parameters

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

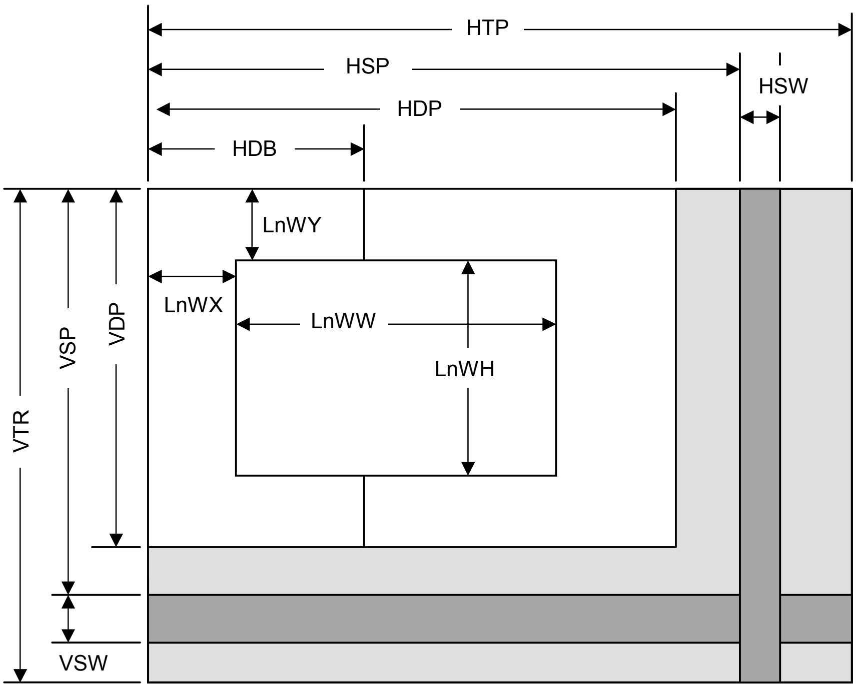


Fig. 5.1 Display Parameters

Note: The actual parameter settings are little different from the above. The details, please refer “14.3.1 Interlaced mode”.

HTP	Horizontal Total Pixels
HSP	Horizontal Synchronize pulse Position
HSW	Horizontal Synchronize pulse Width
HDP	Horizontal Display Period
HDB	Horizontal Display Boundary
VTR	Vertical Total Raster
VSP	Vertical Synchronize pulse Position
VSW	Vertical Synchronize pulse Width
VDP	Vertical Display Period
LnWX	Layer n Window position X
LnWY	Layer n Window position Y
LnWW	Layer n Window Width
LnWH	Layer n Window Height

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

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

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

7.2.4 Display position control

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

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

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

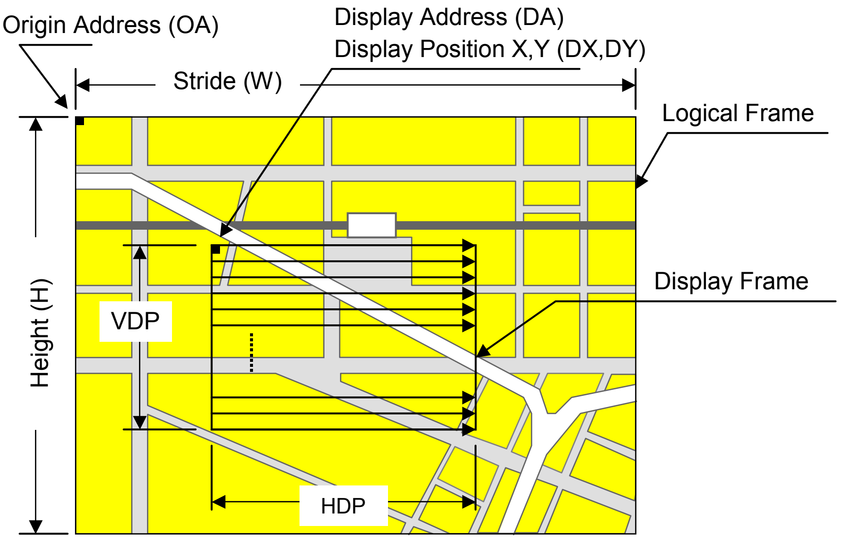


Fig. 5.2 Display Position Parameters

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

MB86296S 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:

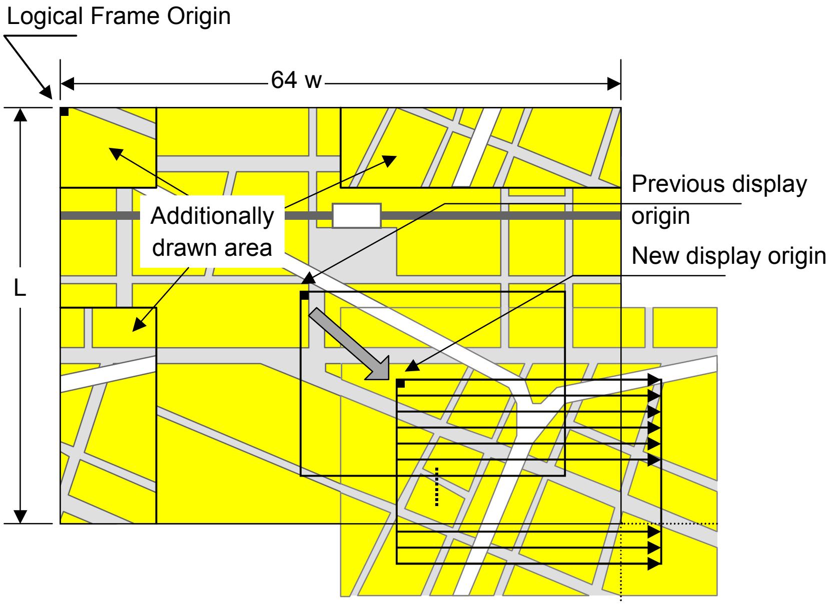


Fig. 5.3 Wrap Around of Display Frame

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

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

$0 \leq DX < w \times 64 \times 8/bpp$ (bpp = 8 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.

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

7.3 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 L3 layer. This palette can be used only for the extended mode.

7.4 Cursor

7.4.1 Cursor display function

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

7.4.2 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.

7.5 Display Scan Control

7.5.1 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

7.5.2 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.

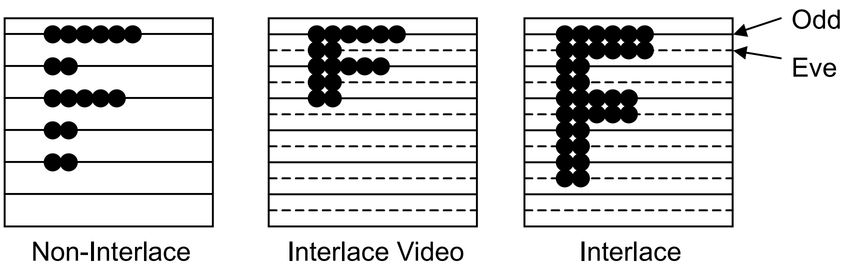


Fig. 5.4 Display Difference between Synchronization Modes

7.6 Video Interface, NTSC/PAL Output

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

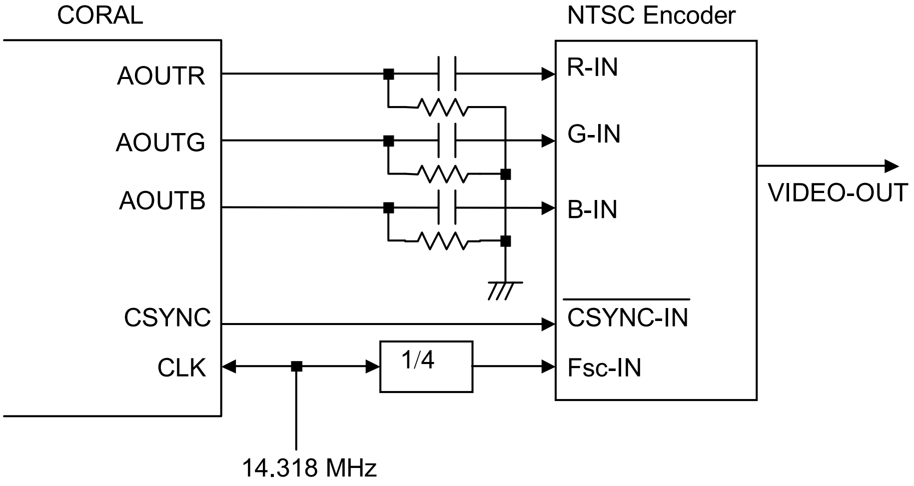


Fig. 5.6 Example of NTSC/PAL Encoder Connection

Note) The neither CSYNC and VSYNC pins are impossible to output the 2.5H width signal.

7.7 Programmable YCbCr/RGB conversion for L1-layer display

L1-layer can display video data in YCbCr format but RGB conversion coefficients are hard-wired and fixed about previous products. Coral-PA can program RGB conversion coefficients by registers.

YCbCr data is converted by following expression.

$R = a_{11}*Y + a_{12}*(Cb-128) + a_{13}*(Cr-128) + b_1$

$G = a_{21}*Y + a_{22}*(Cb-128) + a_{23}*(Cr-128) + b_2$

$B = a_{31}*Y + a_{32}*(Cb-128) + a_{33}*(Cr-128) + b_3$

a_{ij} ---- 11bit signed real (lower 8bit is fraction, two's complement)

b_i ----- 9bit signed integer (two's complement)

It is represented by matrix operation.

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \mathbf{A} \begin{pmatrix} Y \\ Cb\text{-}128 \\ Cr\text{-}128 \end{pmatrix} + \boldsymbol{b} \quad \text{where} \quad \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \ \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

These parameters are set on registers shown bellow.

L1YCR0 (a_{12},a_{11}), L1YCR1(b_1,a_{13})

L1YCG0 (a_{22},a_{21}), L1YCG1(b_2,a_{23})

L1YCB0 (a_{32},a_{31}), L1YCB1(b_3,a_{33})

Same conversion with previous products is applied by initial values of these registers after reset.

The register values just after reset is as follow.

a_{11} = 0x12b (299/256) , a_{12} = 0x0, a_{13} = 0x198 (408/256)

a_{21} = 0x12b (299/256), a_{22} = 0x79c (-100/256), a_{23} = 0x72f (-209/256)

a_{31} = 0x12b (299/256), a_{32} = 0x204 (516/256), a_{33} = 0x0

b_1= b_2= b_3= 0x1f0 (-16)

It is possible to control brightness, contrast, hue , color saturation by change these parameters.

Addition of a constant value into $\boldsymbol{b}$ means inclease of brightness.

Multiplication of a constant scalar value greater than one into $\mathbf{A}$ means increase of contrast.

Two dimentional rotation of Cb-128 and Cr-128 means change of hue.

Color saturation is intensity of color, relative to Y-component.

New coefficients including these changes can be got by following expression.

$$\mathbf{A} = c_1 \mathbf{A}_0 \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(t) & \sin(t) \\ 0 & -\sin(t) & \cos(t) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & c_2 \end{pmatrix} = \mathbf{A}_0 \begin{pmatrix} c_1 & 0 & 0 \\ 0 & \cos(t)c_1c_2 & \sin(t)c_1c_2 \\ 0 & -\sin(t)c_1c_2 & \cos(t)c_1c_2 \end{pmatrix}$$

$$\boldsymbol{b} = \boldsymbol{b}_0 + \begin{pmatrix} c_3 \\ c_3 \\ c_3 \end{pmatrix}$$

$\mathbf{A}_0$, $\boldsymbol{b}_0$: initial value

c_1: contrast parameter, 1 is standard. 1.2 is stronger, for example.

c_2: color saturation parameter, 1 is standard. 0 means mono chrome image.

c_3: brightness parameter, 0 is standard.

t : hue rotation parameter, 0-deg is standard

Note: new a_{ij} and b_i should be clipped in valid range of value for corresponding registers.

7.8 DCLKO shift

1) Delay

DCLKO delay function is available if internal PLL is used for DCLK. DCKD field in DCM3 register defines delay value by internal PLL clock cycle.

DCKD	delay
000000	No additional delay
000010	+2 PLL clock
000100	+3 PLL clock
000110	+4 PLL clock
:	:
111110	+33 PLL clock

2) Inversion

DCLKO inversion is also available with/without delay function. This function is effective with no relation to DCLK clock source.

CKinv-bit of DCM3 enables this function.

7.9 Syncronous register update of display

To update position related parameters without disturbing display, it is need to update synchronously with VSYNC interrupt and finish at a time.

This synchronous register update mode eases this limitation. In this mode, written parameters are hold in intermediate registers and update at once synchronously with VSYNC.

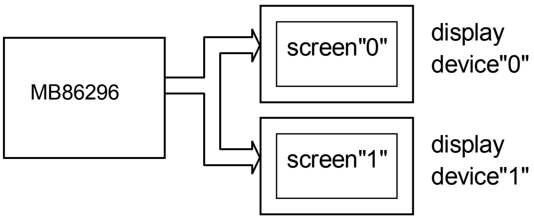
RUM-bit of DCM2 register enables this mode.

RUF-bit of DCM2 register controls start of update and shows whether update is done or not.

7.10 Dual Display

7.10.1 Overview

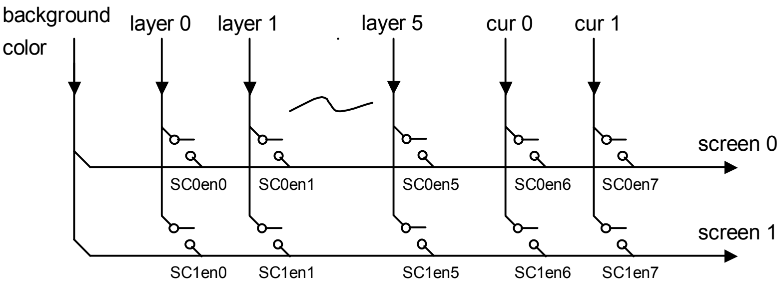
This function enables to display two screens on two display devices. It is possible to control which layer is included in a screen. It is assumed here that display device "0" has screen "0" and display device "1" has screen "1".



7.10.2 Destination Control

A layer or cursor can be included in both screens or one screen. If a layer is NOT included into a screen, this layer is treated as "transparent" . If all bits of a screen are set "0", then background color is displayed on the screen.

This destination control can be thought virtually as crosspoint switch shown next



MDen (multi display enable) bit of MDC(multi display control) register enables this function.

SC0en (screen"0" enable) field of MDC register defines which layers and cursors are included in screen “0”.

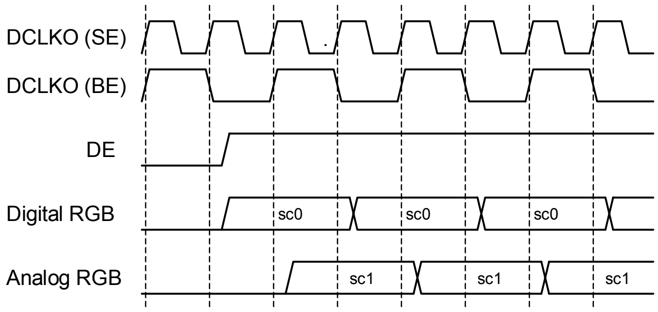
SC1en (screeen"1" enable) field of MDC register defines which layers and cursors are included in screen “1”.

bit-0 ---- L0 is included
bit-1 ---- L1 is included
:
bit-5 ---- L5 is included
bit-6 ---- Cursor0 is included
bit-7 ---- Cursor1 is included

7.10.3 Output Signal Control

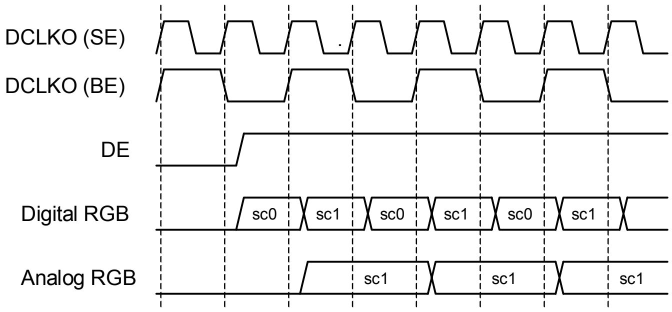
There are two mode to output two screens. In parallel mode, one screen is output at digital RGB while another screen is output at analog RGB. In multiplex mode, two screens are multiplexed and output at digital RGB.

(1) parallel output mode



Note: Analog RGB is shown as corresponding data value

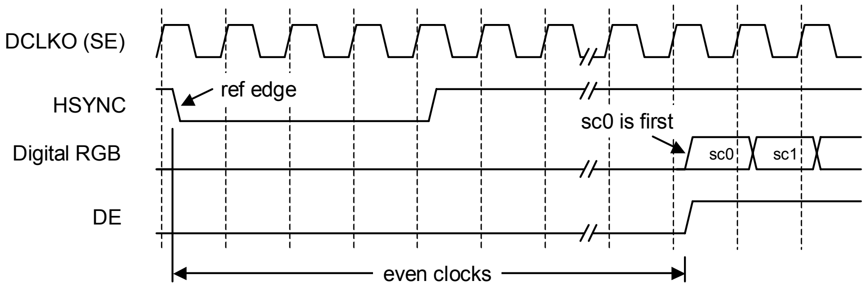
(2) multiplex output mode



Note: Analog RGB is shown as corresponding data value

In BE (bi-edge) DCLKO mode, two ouput phases can be identified both edge of DCLKO.

In SE (single-edge) DCLKO mode, two output phases cab be identified an edge of HSYNC or DE.



POM(parallel output mode) bit in DCM3 register defines which output mode is used, parallel or multiplex. POM=0 means multiplex, POM=1 means parallel, respectively.

CKed(clock edge) bit in DCM3 register defines which DCLKO clock mode is used, BE(bi-edge) or SE(single-edge). DCKed=0

7.10.4 Output Circuit Example

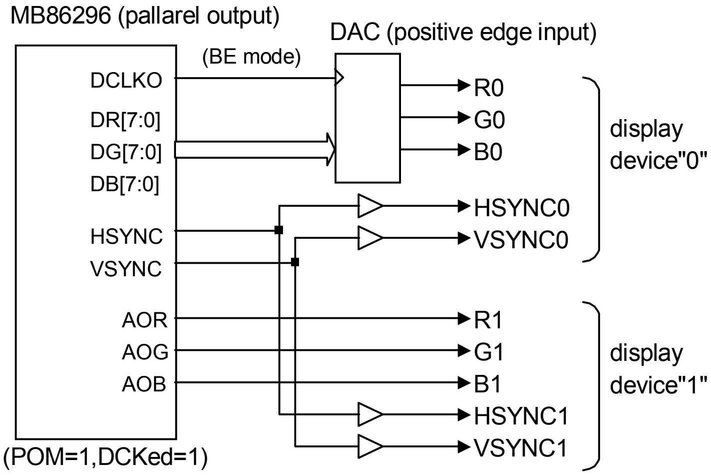
There are three types of output circuit for dual display, primalry.

Parallel, Digital Multiplex(SE), Digital Multiplex(BE)

Here these three examples are described.

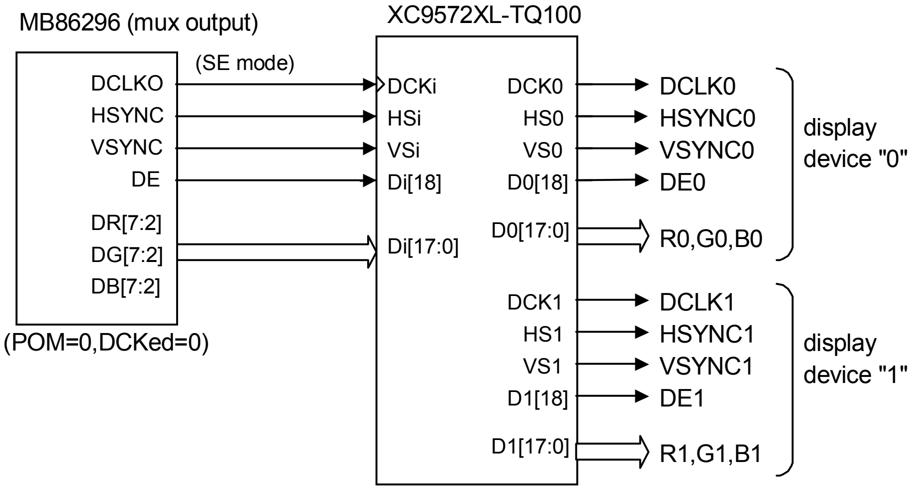
(1) Parallel output

Two screens are given as analog signals in this example.

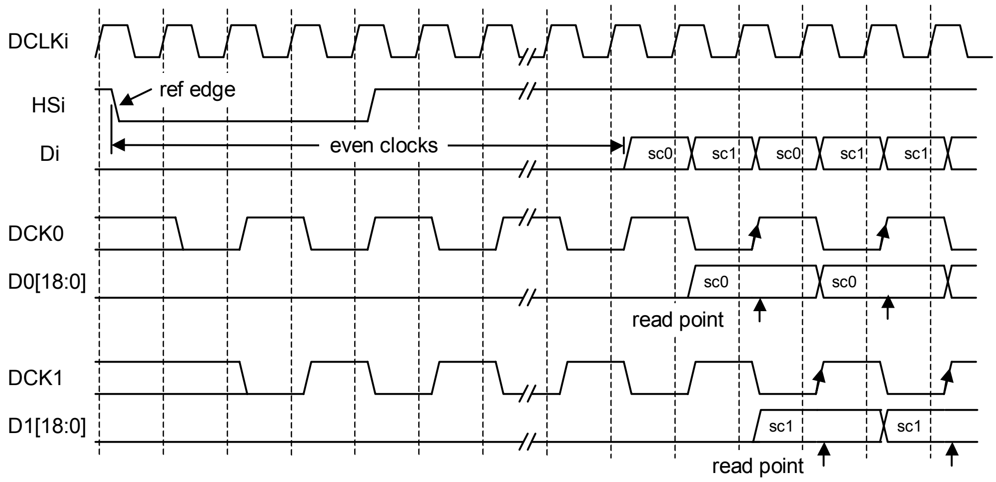


(2) Multiplexed digital output with SE mode DCLKO

In this case, CPLD can be used to demultiplex two digital streams of each screen. In following example, one economical CPLD demultiplexes RGB 6bit/component video data stream.

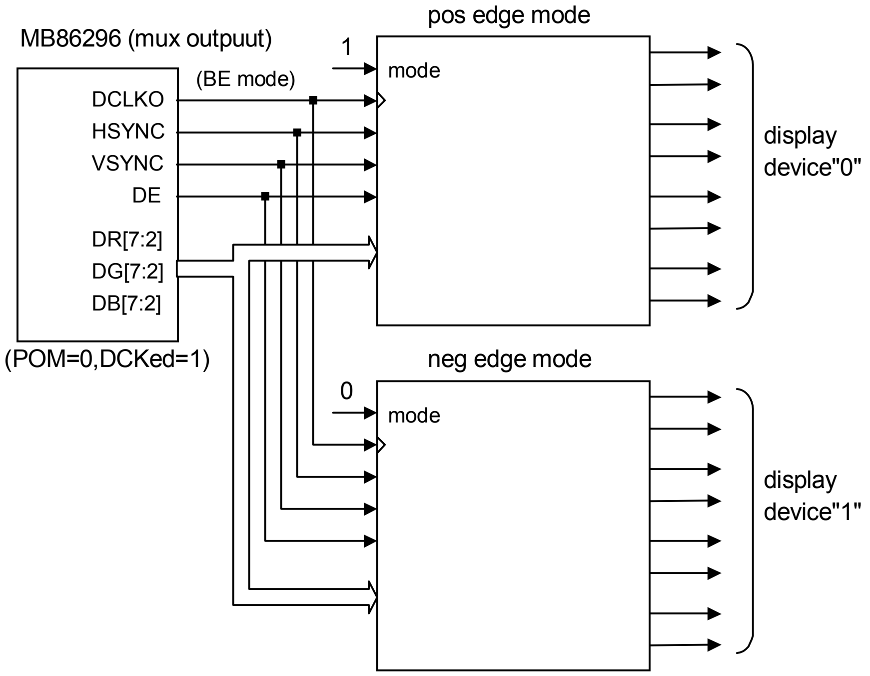


```
module XC9572XL ( DCKi, HSi,VSi,Di,  DCK0,HS0,VS0,D0,  DCK1,HS1,VS1,D1 );
    input DCKi,HSi,VSi;
    input[18:0] Di;
    output DCK0,HS0,VS0, DCK1,HS1,VS1;
    output[18:0] D0,D1;
    reg HS0,HS1, VS0,VS1, DCK0,DCK1;
    reg[18:0] D0,D1;
    always @(posedge DCKi) begin
        HS0 <= HSi; HS1 <= HS0;
        VS0 <= VSi; VS1 <= VS0;
        DCK0 <= (HS0&!HSi)? 0: !DCK0; // sync to ref edge : flip
        DCK1 <= DCK0;
        if(DCK0) D0 <= Di;
        if(DCK1) D1 <= Di;
    end
endmodule
```



(3) Multiplexed digital output with BE mode DCLKO

If a receiving device can select data strobe edge, it can be used to demultiplex two screens with rising and falling edge of DCLKO.



7.10.5 Display Clock and Timing

It is need to supply display clock of twice frequency for dual display function to work. VGA display uses 25MHz display clock, typically in single display mode while 50MHz display clock is need for dual display mode. The timing parameters such as HTP except scaling ratio (SC) are same.

Maximum display clock frequency determines maximum available resolution. It is 800 x 480. 66MHz DCLK clock is need for it.

7.10.6 Limitation

Two display devices has same scan rate and resolution with common sync signals.

The external sync mode can not be used in dual display mode.

8. Video Capture

8.1 Video Capture function

8.1.1 Input data Formats

The digital video stream of ITU RBT-656 or RGB666 format conformity is inputted (for details refer to **8.5 external video signal input conditions**).

8.1.2 Capturing of Video Signal

"Coral-PA" becomes effective when VIE of a video capture mode register (VCM) is 1, and it is CCLK. Synchronizing with a clock, video stream data is captured from 8-bit VI pin or 20-bit RGB input pin.

8.1.3 Non-interlace Transformation

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

- BOB Mode

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

In order to choose BOB mode, while enable vertical interpolation in VI bit of a VCM (Video Capture Mode) register, the L1IM bit of L1M (L1-layer Mode) register is set as 0.

- WEAVE Mode

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

In order to choose WEAVE mode, while disable vertical interpolation in VI bit of a VCM (Video Capture Mode) register, the L1IM bit of L1M (L1-layer Mode) register is set as 1.

8.2 Video Buffer

8.2.1 Data Form

The video capture unit of MB86296 “Coral-PA” 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.

31 24	23 16	15 8	7 0
Y1	Cr	Y0	Cb

	7	6	5	4	3	2	1	0
Y0,Y1	Y7	Y6	Y5	Y4	Y3	Y2	Y1	Y0
Cr,Cb	C7	C6	C5	C4	C3	C2	C1	C0

8.2.2 Synchronous Control

Writing to the graphics memory of video image data and scan for a display are performed independently. The graphics memory for video captures is controlled by the ring buffer system. It displays the frame, when the image data for one frame can be preparing on a memory. When the frame rate of a video capture differs from the frame rate of a display, the continuation display of top omission or the same frame occurs.

8.2.3 Area Allocation

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

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

CBLA = CBOA + 64 (n-2) × CBW

In addition, the head addresses of n+1 raster are 64n×CBW, and CBLA+2 raster becomes a buffer domain. For reduced display, allocate the buffer area of the reduced frame size.

8.2.4 Window Display

The captured video picture is displayed using L1 layer. The whole or a part captured picture can be displayed as the whole screen or a window.

When performing a capture display, L1 layer is set as capture synchronous mode (L1CS=1). In this mode, L1 layer display displays the newest frame in a video capture buffer. Usually, the display address used in the mode is disregarded.

The stride of L1 layer needs to be in agreement with the stride of a video capture buffer. When not in agreement, the picture distorted aslant is displayed.

The display size of L1 layer is made in agreement with the picture size after reduction of a video capture. Invalid data will be displayed if the display size of L1 layer is set up more greatly than capture picture size.

Although selection of a RGB display and a YCbCr display can be performed in L1 layer, in performing a video capture, it chooses YcbCr form (L1YC=1).

8.2.5 Interlace Display

It is possible to display the picture taken in to the video capture buffer in WEAVE mode in an interlace. A setup confirms WEAVE mode and chooses an interlace & video display with display scan.

However, when display scan is asynchronous, flicker will come out in a scene with a motion. In order to prevent this, OO (Odd Only) bit of a CBM (Capture Buffer Mode) register is set as 1.

When synchronizing display scan with a capture, a capture input and a display output can be made to correspond to 1 to 1. In this case, the difference of flicker of an input and an output is lost. Please refer to "8.8 Capture synchronous display."

8.2.6 RGB555 Mode

As an alternative method, a special RGB555-mode can be used which is dedicated for applications where grabbed pictures should be processed further. In this mode, a single buffer is used instead of a ring buffer. In addition, data is directly stored in Coral's RGB555 format in the L1-Layer (see settings of the CBM register). This makes it possible to copy rectangular areas from the L1-layer directly to the texture buffer or to other memory locations using the BitBlt function. Note that the input and output frame rate should be identical if a single buffer method is used and that the lower bits are ignored to form the RGB555 format.

8.3 Scaling

8.3.1 Down-scaling 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_H to $FFFF_H$. Set the vertical direction at bit 31 to bit 16 of the capture scale register (CSC) and the horizontal direction at bits 15 to bit 00. The initial value for this register is 08000800_H (once). An example of the expressions for setting a reduction in the vertical and horizontal directions is shown below.

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

Therefore, $096808E3_H$ is set in CSC.

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

8.3.2 Up-scaling 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

Note:

- Smooth continuation operation to Down Scaling mode and Up Scaling mode cannot be performed. The picture disorder of some arises at the time of a change. This is the restrictions for Up Scaling mode and Down Scaling mode using the same interpolate circuit.

8.3.2 Flow of image processing

As for the capture image displayed on L1 layer window, image processing is performed by the following flow.

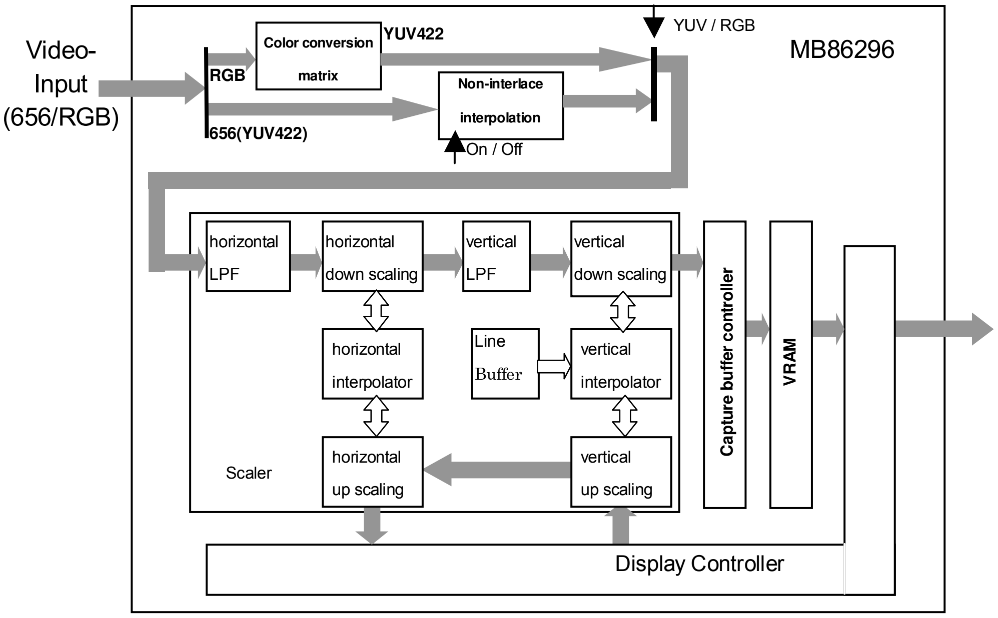


Figure 8.1 Flow of image processing

Non-interlace interpolation processing

When VI of a video capture mode register (VCM) is 0, an interlace screen is interpolated vertically using the data in the same field. A screen is doubled vertically. When VI is 1, it is not interpolated vertically.

Horizontal low-pass filter processing

As a preprocessing when scaling down a picture horizontally, a low-pass filter can be covered horizontally. Regardless of scaling up and scaling down of a picture, ON/OFF is possible for a level low path filter (LPF).

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

CHLPF_Y	Horizontal LPF Luminance element and RGB element coefficient code
CHLPF_C	Horizontal LPF chrominance element coefficient code

The coefficient is specified by the coefficient code in two bits independently by luminance (Y) signal and chrominance (Cb and Cr) signals. The coefficient is a symmetric coefficient.

CHLPF_x	K0	K1	K2	K3	K4
00	0	0	1	0	0
01	0	1/4	2/4	1/4	0
10	0	3/16	10/16	3/16	0
11	3/32	8/32	10/32	10/32	3/32

Horizontal LPF becomes turning off (through) because of the setting of the coefficient code "00".

Note:

- In the case of Native RGB mode (NRGB=1), only a setup of CHLPF_Y code becomes effective.

Down and Up scaling processing of horizontal direction

Please set bit15-00 of capture scale register (CSC) to do the down and up scaling processing of horizontal direction.

Horizontal direction is scaled down before writing in VRAM. Horizontal direction is scaled up after reading from VRAM.

The interpolation filter processing of luminance (Y) signal is done by cubic interpolation (Cubic Interpolate) method. The interpolation filter processing of chrominance (Cb and Cr) signal is done by BiLinear interpolation (BiLinear Interpolate) method. The interpolation filter processing of Native-RGB signal is done by cubic interpolation (Cubic Interpolate) method.

Vertical low-pass filter processing

The low-pass filter can be put on the vertical direction as a preprocessing when the image is scaled down to the vertical direction. Vertical low-pass filter (LPF) can be set to turning on regardless of the scaling up or down of the vertical direction.

A vertical low-pass filter is composed of the FIR filter of three taps. The coefficient is specified by the following register.

CVLPF_Y	Vertical LPF Luminance element and RGB element coefficient code
CVLPF_C	Vertical LPF chrominance element coefficient code

The coefficient is specified by the coefficient code in two bits independently by luminance (Y) signal and chrominance (Cb and Cr) signals. The coefficient is a symmetric coefficient.

CVLPF_x	K0	K1	K2
00	0	1	0
01	1/4	2/4	1/4
10	3/16	10/16	3/16
11	Prohibition of setting		

Vertical LPF becomes turning off (through) because of the setting of the coefficient code "00".

Note:

- In the case of Native RGB mode (NRGB=1), only a setup of CVLPF_Y code becomes effective.

Down and up scaling processing of Vertical direction

Please set bit31-16 of capture scale register (CSC) to do the down and up scaling processing in the vertical direction.

The vertical direction is scaled down before writing in VRAM. The vertical direction is scaled up after reading from VRAM.

The interpolation filter processing of luminance (Y) signal is done by cubic interpolation (Cubic Interpolate) method. The interpolation filter processing of chrominance (Cb and Cr) signal is done by BiLinear interpolation (BiLinear Interpolate) method. The interpolation filter processing of Native-RGB signal is done by cubic interpolation (Cubic Interpolate) method.

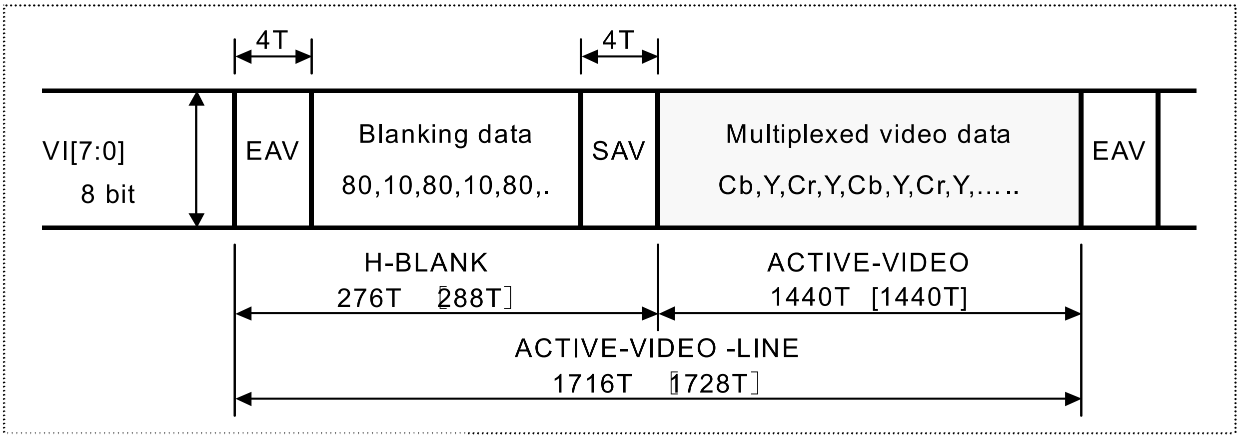
8.4 External video signal input conditions

8.4.1 RTB656 YUV422 input format

The ITU R.BT-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.

1) RTB656 input format VI[7:0]

Synchronous code and image data (Cb,Y,Cr,Y) are input as data of eight multiple bits synchronizing with 27MHz clock, and an valid pixel is transmitted while placed between a synchronous code named SAV and EAV.



SAV : Beginning code of active video data (4 Byte)

EAV : End code of active video data (4 Byte)

T : 27MHz

[ ] : 625/50 series (PAL)

BLANKING PERIOD			TIMING REF-CODE				720 PIXELS YUV4:2:2 DATA									TIMING REF-CODE				BLANKING PERIOD		
…	80	10	FF	00	00	SAV	Cb0	Y0	Cr0	Y1	Cb2	Y2	…	Cr718	Y719	FF	00	00	EAV	80	10	…

2) RTB656 synchronous code (4 Byte) format

Word / Bit	SYNC code (static)			EAV/SAV
	first	second	third	forth
7	1	0	0	1 (static)
6	1	0	0	F 0:first field 1:second field
5	1	0	0	V 0:ACTIVE-VIDEO 1:VBI
4	1	0	0	H 0:SAV 1:EAV
3	1	0	0	P3 Guard bit
2	1	0	0	P2 Guard bit
1	1	0	0	P1 Guard bit
0	1	0	0	P0 Guard bit

3) SAV/EAV timing base signal

Bit Function	7 static	6 F	5 V	4 H	3 P3	2 P2	1 P1	0 P0
80	1	0	0	0	0	0	0	0
9D	1	0	0	1	1	1	0	1
AB	1	0	1	0	1	0	1	1
B6	1	0	1	1	0	1	1	0
C7	1	1	0	0	0	1	1	1
DA	1	1	0	1	1	0	1	0
EC	1	1	1	0	1	1	0	0
F1	1	1	1	1	0	0	0	1

80 : SAV code of first field valid pixel period (Active-video)

9D : EAV code of first field valid pixel period (Active-video)

AB : SAV code of first field vertical retrace line period

B6 : EAV code of first field vertical retrace line period

C7 : SAV code of second field valid pixel period (Active-video)

DA : EAV code of second field valid pixel period (Active-video)

EC : SAV code of second field vertical retrace line period

F1 : EAV code of second field vertical retrace line period

8.4.2 RGB input format

There are the two data-processing methods in RGB input video capture function. One is the method of processing with Native RGB. Another is the method of converting RGB into YUV422 by the internal RGB pre processor.

RGB input function 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 66Mpixel/sec. RGB component data is 6bit.

Note:

- In Native RGB mode, NRGB=1 is set up.

1) RGB Input Signals

The signals used for RGB video capture are not assigned dedicated terminals but share same pins with other functions.

Name	I/O	Function
RGBCLK	Input	Clock for RGB input
RI5-0	Input	Red component value
GI5-0	Input	Green component value
BI5-0	Input	Blue component value
VSYNCI	Input	Vertical sync for RGB capture
HSYNCI	Input	Horizontal sync for RGB capture

Note :

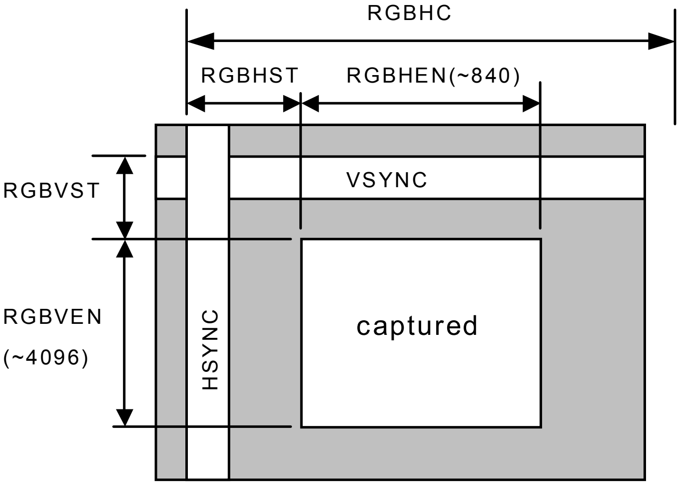
- input pins are shared with the ITU656 input and memory data bus.
- the VIS bit of the VCM (video capture mode) register selects which mode (ITU656 or RGB) is used.

2) 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:

a) RGB input mode of capture: Set RGB666 input flag(VIS) in VCM.
In Native RGB mode, NRGB in VCM =1 is set up.
b) HSYNC Cycle: Set the number of HSYNC Cycles in RGBHC.
c) Horizontal Enable area: Set enables area start position and
enable picture size into RGBHST and RGBHEN.
d) Vertical Enable area: Set enables area start position and
enable picture size into RGBVST and RGBVEN.

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



RGBHC	RGB input Hsync Cycle
RGBHST	RGB input Horizontal enable area STart position
RGBHEN	RGB input Horizontal enable area size
RGBVST	RGB input Vertical ENable area STart position
RGBVEN	RGB input Vertical ENable area size

Note: The actual parameter settings are little different from the above. The details, please refer “Explanation of Registers”.

e) Convert Matrix Coefficient

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

Note:

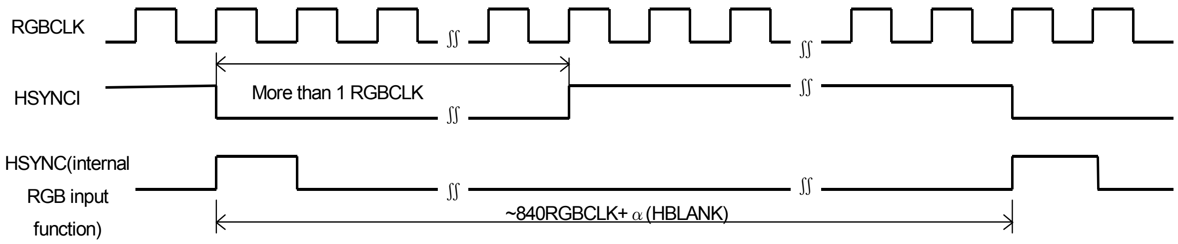
- The maximum horizontal enable area size(RGBHEN) which can be captured is 840 pixels. This is the restriction by line buffer size in a video capture module.

3) Input Operation

At the time of a RGB input, the synchronization of data is taken by VSYNC and SYNCI, which are inputted with Data RI, GI and BI.

■ Input rule of HSYNCI

The positive or negative edge of VINHSYNC is considered as a horizontal synchronization by register setup(HP). Input the signal of 1 or more RGBCLKs –(840+α)RGBCLK cycle.



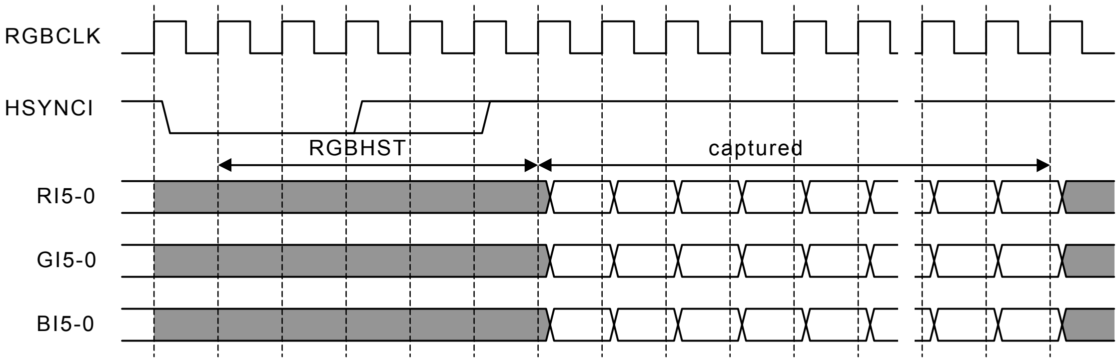
Note:

- The maximum horizontal enable area size(RGBHEN) which can be captured is 840 pixels. This is the restriction by line buffer size in a video capture module.

■ Valid data input rule to HSYNC

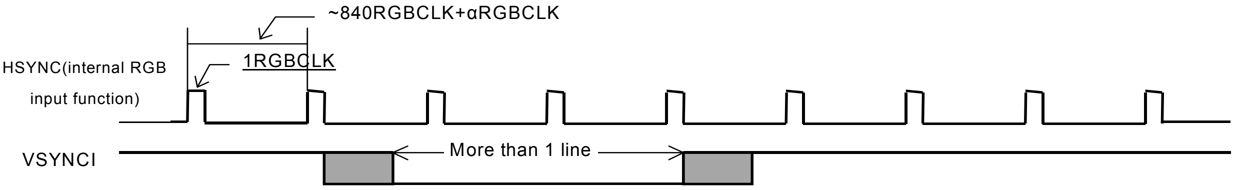
The valid image data input rule to HSYNC is shown.

Input data is inputted synchronizing with HSYNC of each line. (The synchronization of data needs to make a synchronization establish by HSYNC in each line unit. Since the sampling clock of image data is generated from HSYNC, it is because a clock may have jitter per line.)



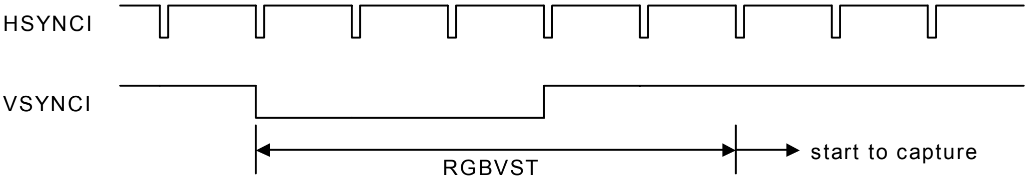
■ Input rule of VINVSYNC

A VSYNCI signal is synchronizing with HSYNCI. Moreover, VSYNCI is sampled by HSYNCI , and it considers as a VSYNC signal. Width is made into at least one line or more although a VSYNCI signal does not need to synchronize with HSYNC at this time. The positive or negative of VSYNCI is set to VSYNC by register setup(VP).



■ valid line input rule to HSYNC

The valid image data input rule to VSYNC is shown.



4) Conversion Operation

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

$Y = a11*R + a12*G + a13*B + b1$

$Cb = a21*R + a22*G + a23*B + b2$ a_{ij} : 10bit signed real (lower 8bit is fraction)

$Cr = a31*R + a32*G + a33*B + b3$ b_i: 8bit unsigned integer

Note:
- registers can define each coefficient.
- Cb and Cr components are reduced to half after this operation to form in 4:2:2 format.

8.5 Input Video Signal Parameter Setup

A parameter setup of an input video signal changes with video formats inputted. A register to be set up is shown in the following figure.

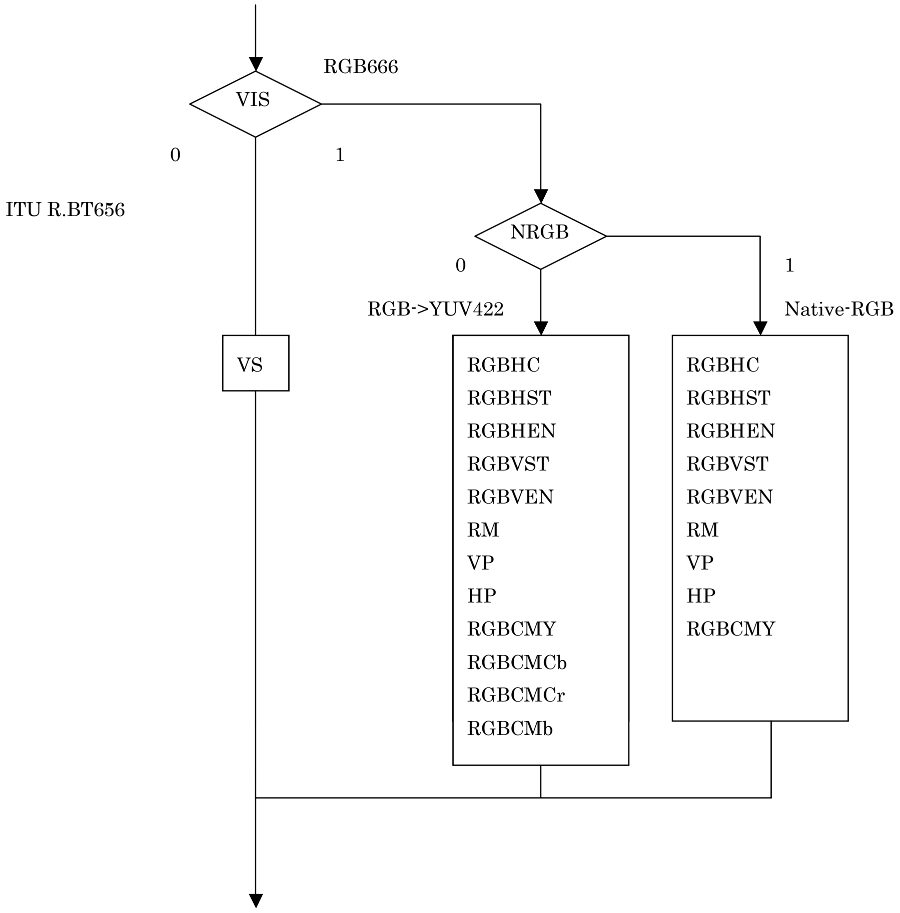


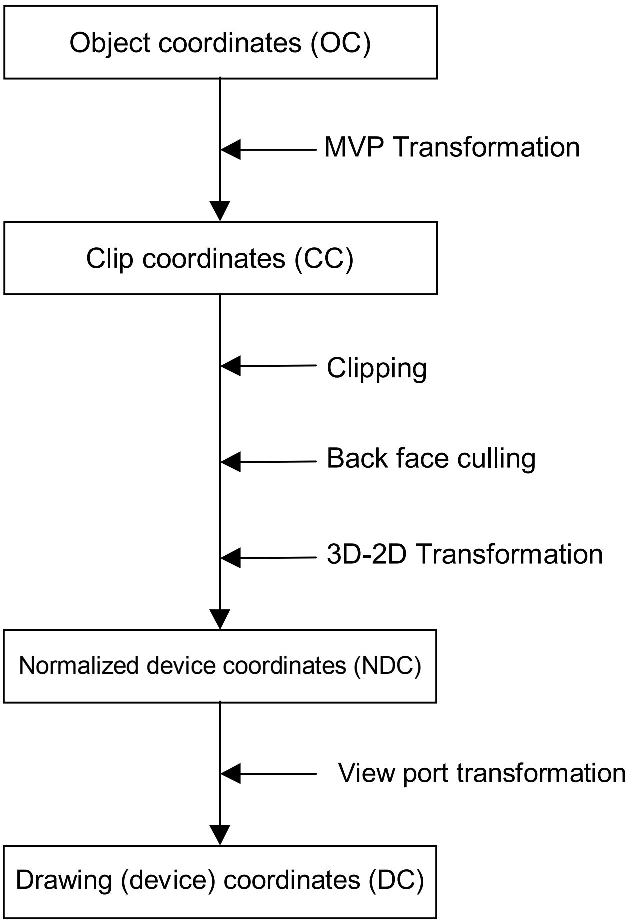
Figure 8.2 A register required for a setup according to format

9. GEOMETRY ENGINE

9.1 Geometry Pipeline

9.1.1 Processing flow

The flow of geometry is shown below.



Calculation is done by “32bit integer”,”32bit fixed-point-integer” or “32bit floating-point”. There is a limitation by itself. And algorithm also has limitation. Not all possible parameter or data can proceed correctly.

9.1.2 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". (Work only for C=0,Z=0 and ST=0 (XY only vertex) mode)

OC: Object Coordinates

CC: Clip Coordinates

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

Ma0 to Md3: OC → CC Matrix

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

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

9.1.3 3D-2D transformation (CC→NDC coordinate transformation)

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

NDC: Normalized Device Coordinates

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

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

9.1.4 View port transformation (NDC→DC coordinate transformation)

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

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

DC: Device Coordinates

Xdc = X_Scaling*Xndc + X_Offset

Ydc = Y_Scaling*Yndc + Y_Offset

Zdc = Z_Scaling*Zndc + Z_Offset

9.1.5 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.

$$\begin{aligned} Xmin*Wcc &\le Xcc \le Xmax*Wcc \\ Ymin*Wcc &\le Ycc \le Ymax*Wcc \\ Zmin*Wcc &\le Zcc \le Zmax*Wcc \\ Wmin &\le Wcc \end{aligned}$$

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". (Work only for C=0,Z=0 and ST=0 (XY only vertex) mode.)

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

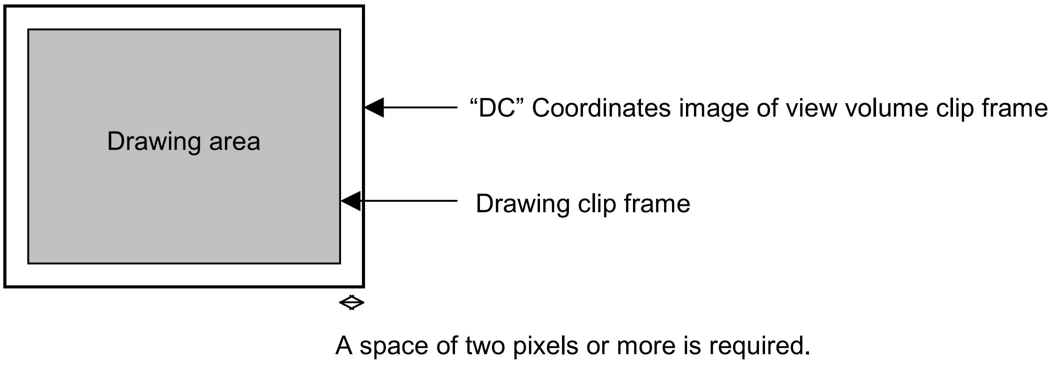
For the following reasons, the clip boundary values of the view volume should be set so that the values after DC coordinate transformation will be larger than the drawing clip frame (2 pixels or more).

(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.

9.1.6 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 carling is required. When back face culling is not required such as in ‘line,’ ‘point,’ and ‘polygon primitive,’ GMDR2[0] must be set to 0.

9.2 Data Format

9.2.1 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

31	30 23	22 0
s	e	f

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)

31	30 16	15 0
s	Int	Frac

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

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

31	30 16	15	14 0
s	Y.int	s	X.int

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

(4) RGB packed format

31 24	23 16	15 8	7 0
reserved	R	G	B

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

(5) ARGB packed format

31 24	23 16	15 8	7 0
A	R	G	B

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

9.3 Setup Engine

9.3.1 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.

9.4 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).

9.4.1 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.

9.4.2 Log output destination address

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

9.4.3 Log output format

The log format consists of packed number of X and Y coordinates of vertex.

bit

31	30 16	15	14 0
S	Y	S	X

S : signed bit

Y : Y coordinates values (integer)

X : X coordinates values (integer

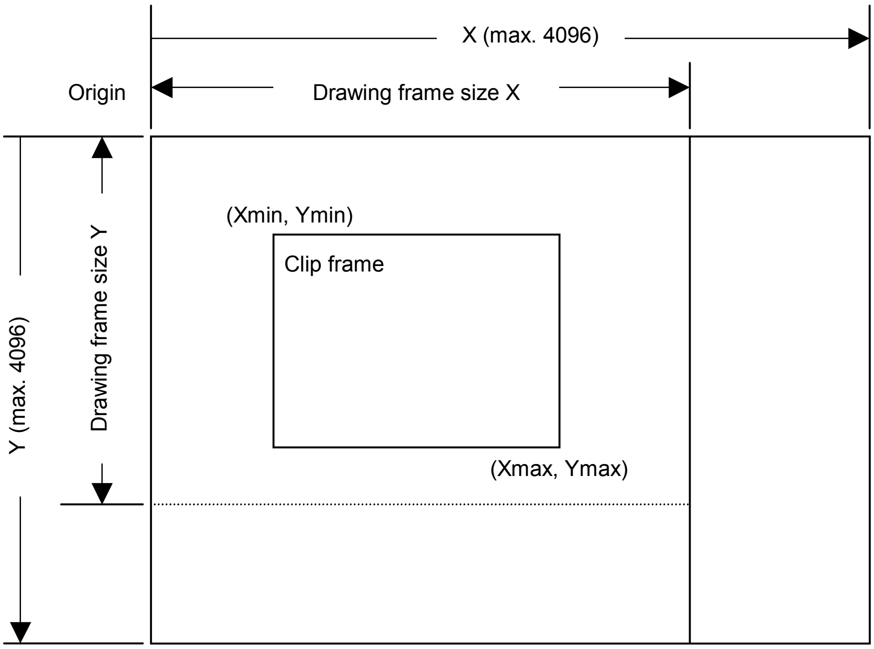
10. DRAWING PROCESSING

10.1 Coordinate System

10.1.1 Drawing coordinates

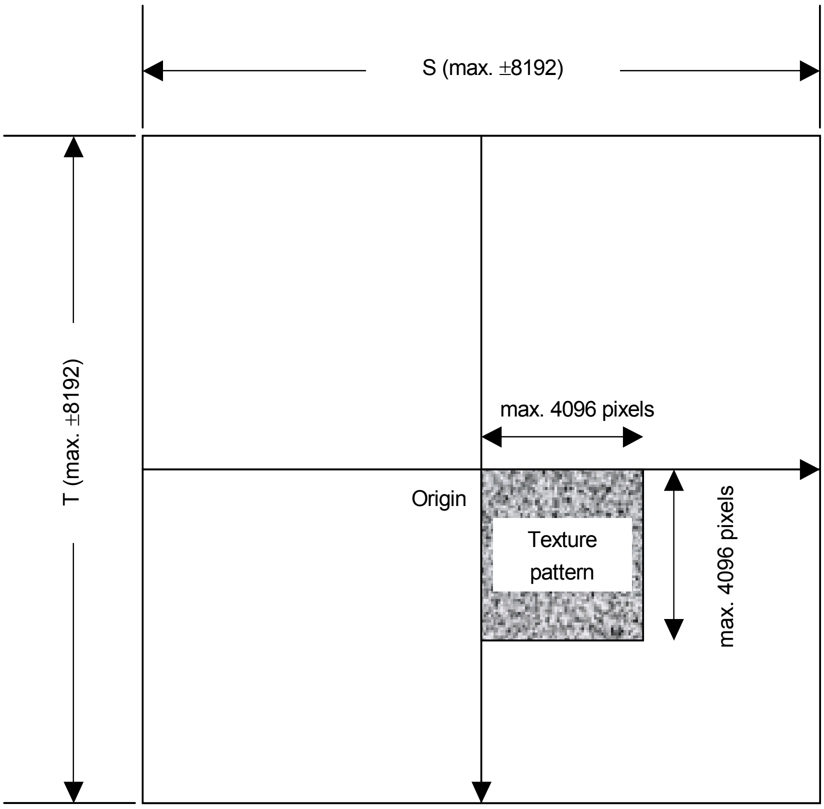
After the calculation of coordinates by the geometry engine, CORAL draws data in the drawing frame in the graphics memory that finally uses the drawing coordinates (device coordinates).

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 × 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.



10.1.2 Texture coordinates

Texture coordinate is a 2D coordinate system represented as S and T (S: horizontal, T: vertical). Any integer in a range of –8192 to +8191 can be used as the S and T coordinates. The texture coordinates is correlated to the 2D coordinates of a vertex. One texture pattern can be applied to up to 4096 × 4096 pixels. The pattern size is set in the register. When the S and T coordinates exceed the maximum pattern size, the repeat, cramp or border color option is selected.



10.1.3 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.

10.2 Figure Drawing

10.2.1 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

10.2.2 Polygon drawing function

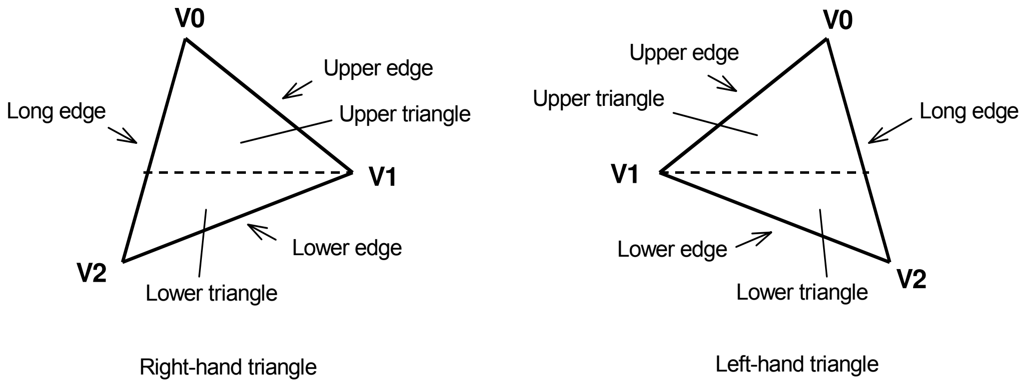
An irregular polygon (including concave shape) is drawn by hardware in the following manner:

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.

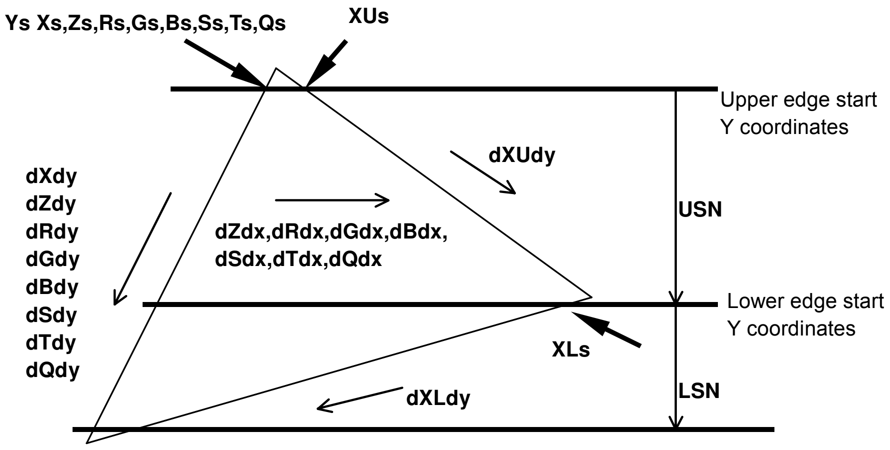
10.2.3 Drawing parameters

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

10.2.4 Anti-aliasing function

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

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

10.3 Bit Map Processing

10.3.1 BLT

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

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

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

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

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

10.3.2 Pattern data format

CORAL can handle three bit map data formats: indirect color mode (8 bits/pixel), direct color mode (16 bits/pixel), and binary bit map (1 bit/pixel).

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

10.4 Texture Mapping

10.4.1 Texture size

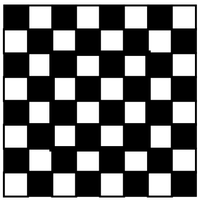
CORAL reads texel corresponding to the specified texture coordinates (S, T), and draws that data at the correlated pixel position of the polygon. For the S and T coordinates, the selectable texture data size is any value in the range from 4 to 4096 pixels represented as an exponent of 2.

10.4.2 Texture color

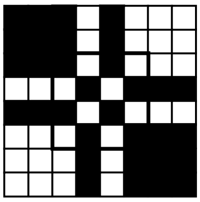
Drawing of 8-/16-bit direct color is supported for the texture pattern. For drawing 8-bit direct color, only point sampling can be specified for texture interpolation; only decal can be specified for the blend mode.

10.4.3 Texture Wrapping

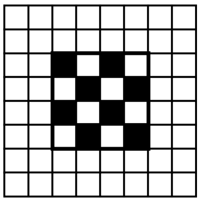
If a negative or larger than the specified texture pattern size is specified as the texture coordinates (S, T), according to the setting, one of these options (repeat, cramp or border) is selected for the 'out-of-range' texture mapping. The mapping image for each case is shown below:



Repeat



Cramp



Border

Repeat

This just simply masks the upper bits of the applied (S, T) coordinates. When the texture pattern size is 64 × 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 > Texture X size − 1	S = Texture X size − 1

Border

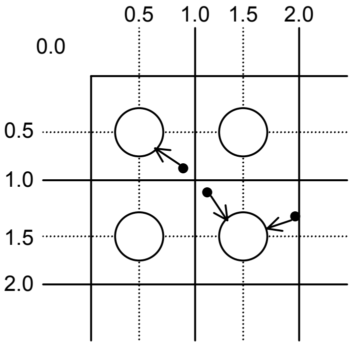
When the applied (S, T) coordinate is either negative or larger than the specified texture pattern size, the outside of the specified texture pattern is rendered in the 'border' color.

10.4.4 Filtering

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

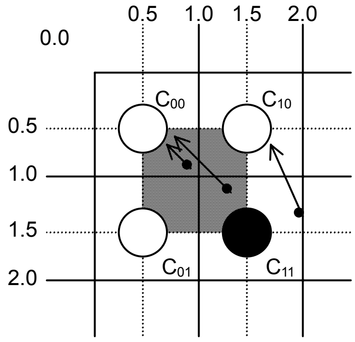
Point filtering

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



Bi-linear filtering

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



10.4.5 Perspective correction

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

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

10.4.6 Texture blending

CORAL supports the following three blend modes for texture mapping:

Decal

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):

$$C_0 = C_T \times C_P$$

Stencil

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

MSB = 1: Texture color

MSB = 0: Polygon color

10.4.7 Bi-linear high-speed mode

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

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

The 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)

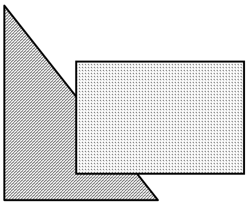
	0				1					6				7			
0	00	01	08	09	01	02	09	10	to	06	07	14	15	07	00	15	08
1	08	09	16	17	09	10	17	18	to	14	15	12	13	15	08	23	16
2	16	17	24	25	17	18	25	26	to	22	23	30	31	23	16	31	24
3	24	25	32	33	25	26	33	34	to	30	31	38	39	31	24	39	32
4	32	33	40	41	33	34	41	42	to	38	39	46	47	39	32	47	40
5	40	41	48	49	41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

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

10.5 Rendering

10.5.1 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 × 64 pixels. (at 16-bit color)



Example of Tiling

10.5.2 Alpha blending

Alpha blending blends the drawn in frame buffer to-be-drawn pixel or pixel already according to the alpha value set in the alpha register. This function cannot be used simultaneously with logic operation drawing. It can be used only when the direct color mode (16 bits/pixel) is used. The blended color C is calculated as shown below when the color of the pixel to be drawn is C_P, the color of frame buffer is C_F, and the alpha value is A:

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

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

Normal

Blends post texture mapping color with frame buffer color

Stencil

Uses MSB of texel color for ON/OFF control:

MSB = 1: Texel color

MSB = 0: Frame buffer color

Stencil alpha

Uses MSB of texel color for α/OFF control:

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

MSB = 0: Frame buffer color

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

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

10.5.3 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

10.5.4 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

10.6 Drawing Attributes

10.6.1 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

10.6.2 Triangle drawing attributes

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

Triangle Drawing Attributes

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

How to set gray scale gouraud shading

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

10.6.3 Texture attributes

In texture mapping, the following attributes apply:

Texture Attributes

Drawing Attribute	Description
Texture mode	Select either texture mapping or tiling
Texture filter	Select either point sampling or bi-linear filtering
Texture coordinates correction	Select either linear or perspective correction
Texture wrap	Select either repeat or cramp of texture pattern
Texture blend mode	Select either decal or modulate
Bi-linear high-speed mode	Texture data is created in a dedicated format to perform high-speed bi-linear filtering.

10.6.4 BLT attributes

In BLT drawing, the following attributes apply:

BLT Attributes

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

10.6.5 Character pattern drawing attributes

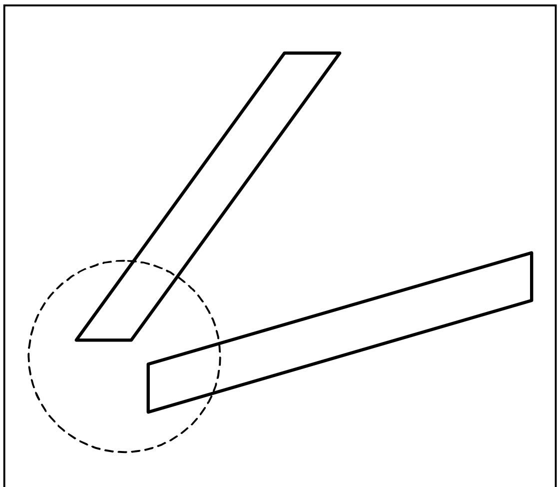
Character Pattern Drawing

Drawing Attribute	Description
Character pattern enlarge/shrink	Vertical and Horizontal × 2, Horizontal × 2, Vertical and Horizontal × 1/2, Horizontal × 1/2
Character pattern color	Set character color and background color
Transparency/non-transparency	Set background color to transparency/non-transparency

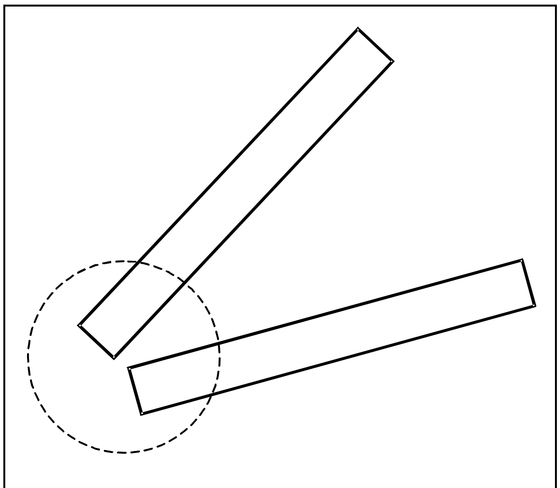
10.7 Bold Line

10.7.1 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 form other primitive.



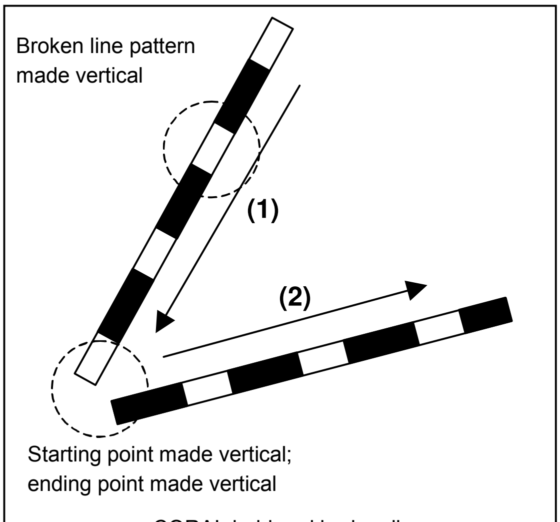
CREMSON bold line mode



CORAL bold line mode

10.7.2 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.

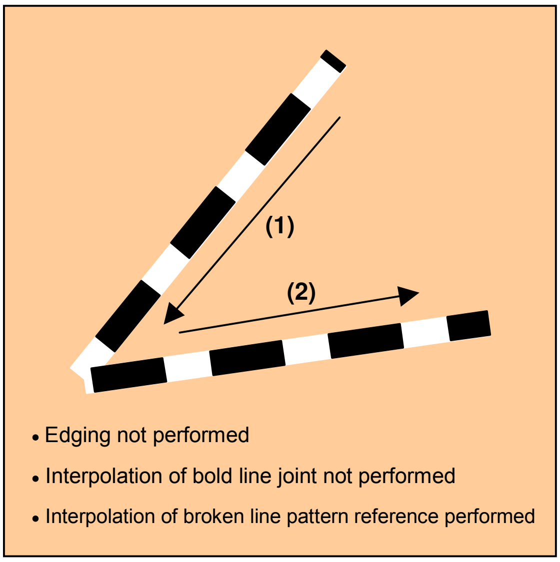


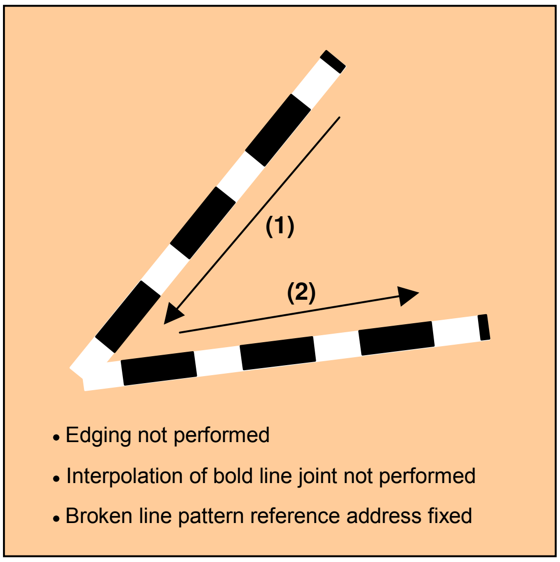
CORAL bold and broken lines

Interpolation of broken line pattern

Two types of interpolation modes are supported:

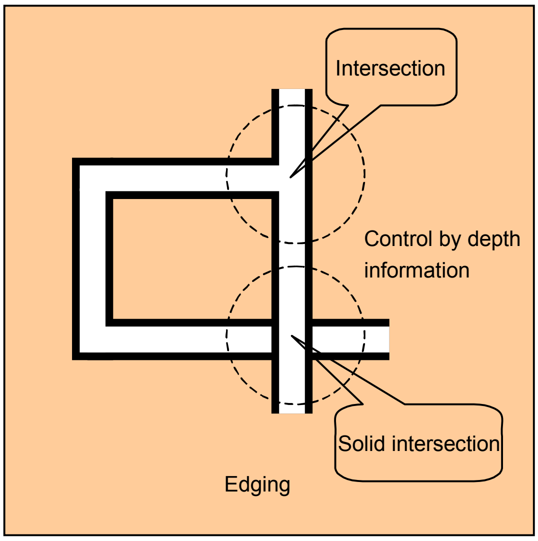
- No interpolation mode: Interpolation is not performed.
- Broken line pattern reference address fix mode: The same broken line pattern is referenced for several pixels before and after the joint of the bold line. Any pixel count can be set by the user.





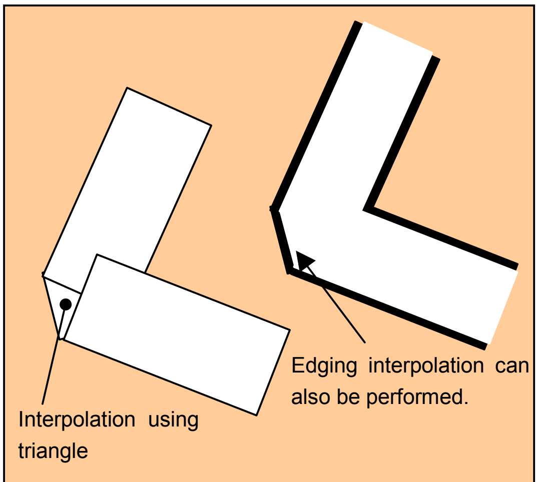
10.7.3 Edging

- 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.



10.7.4 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.
- Only LineStrip primitive can interpolate, and clipping sometimes breaks LineStrip.
- Caution: Sometime joint shape looks not perfect. (using approximate calculation)



Interpolation of bold line joint

10.8 Shadowing

10.8.1 Shadowing

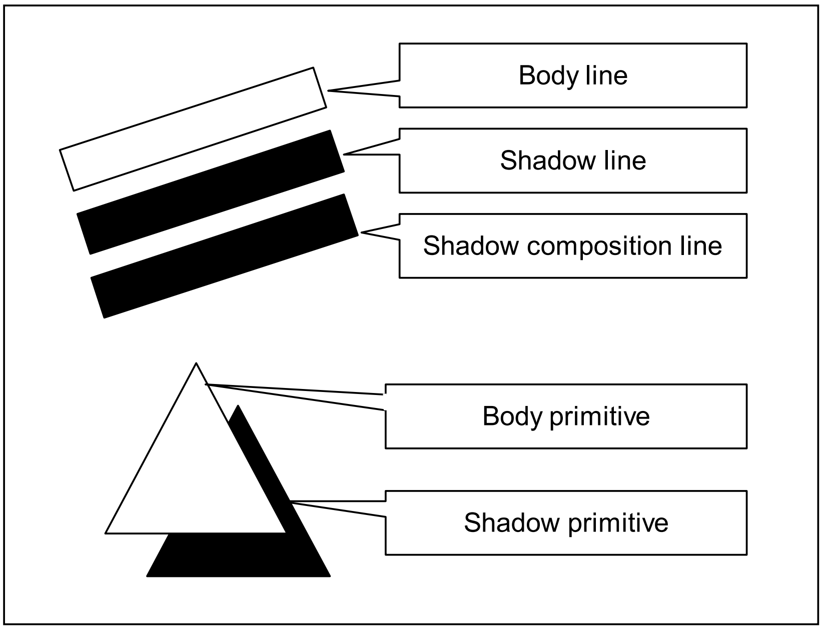
The Coral supports a shadow primitive which is same shape as a body.
A shadow is drawn in a position shifted for a device coordinate(X, Y) by setting the OverlapXY command. And by setting the OverlapZ, it is possible to control a drawing result to avoid twice rendering in alpha blend or logical calculation.

- Line

Two shadow lines are drawn in a line shadowing. One is a shadow line and another is a shadow composition line. A shadow composition line is used for avoiding an overlap with body line. And drawing priority can be set for rendering performance or anti-aliasing.

- Triangle and polygon

A shadow primitive are drawn in a triangle and polygon shadowing. Drawing priority is fixed as a body primitive is first.



11 DISPLAY LIST

11.1 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

11.1.1 Header format

The format of the display list header is shown below.

Format List

<table>
<tr><th>Format</th><th>31 24</th><th>23 16</th><th colspan="3">15 0</th></tr>
<tr><td>Format 1</td><td>Type</td><td>Reserved</td><td colspan="3">Reserved</td></tr>
<tr><td>Format 2</td><td>Type</td><td>Count</td><td colspan="3">Address</td></tr>
<tr><td>Format 3</td><td>Type</td><td>Reserved</td><td colspan="2">Reserved</td><td>Vertex</td></tr>
<tr><td>Format 4</td><td>Type</td><td>Reserved</td><td>Reserved</td><td>Flag</td><td>Vertex</td></tr>
<tr><td>Format 5</td><td>Type</td><td>Command</td><td colspan="3">Reserved</td></tr>
<tr><td>Format 6</td><td>Type</td><td>Command</td><td colspan="3">Count</td></tr>
<tr><td>Format 7</td><td>Type</td><td>Command</td><td colspan="2">Reserved</td><td>Vertex</td></tr>
<tr><td>Format 8</td><td>Type</td><td>Command</td><td>Reserved</td><td>Flag</td><td>Vertex</td></tr>
<tr><td>Format 9</td><td>Type</td><td>Reserved</td><td>Reserved</td><td colspan="2">Flag</td></tr>
<tr><td>Format 10</td><td>Type</td><td>Reserved</td><td colspan="3">Count</td></tr>
<tr><td rowspan="2">Format 11</td><td>Type</td><td>Reserved</td><td colspan="3">Reserved</td></tr>
<tr><td colspan="5">Count</td></tr>
</table>

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

11.1.2 Parameter format

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

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

Rendering command parameters conform to the Coral-PA data format.

11.2 Geometry Commands

11.2.1 Geometry command list

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

Type	Command	Description
G_Nop	—	No operation
G_Begin	See ***Geometry command code table (1)(2)***.	Specifies primitive type and pre-processes
G_BeginE	See ***Geometry command code table (3)(4)***.	Specifies primitive type and pre-processes This command is used at execution of the CORAL extended function.
G_End	—	Ends primitive This command is used at execution of G_Begin
G_EndE	—	Ends primitive This command is used at execution of G_BeginE
G_Vertex	—	Sets vertex parameter and draws
G_VertexLOG	—	Sets vertex parameter and draws Outputs device coordinates
G_VertexNopLOG	—	Only outputs device coordinates
G_Init	—	Initialize geometry engine
G_Viewport	—	Scale to screen coordinates (X, Y) and set origin offset
G_DepthRange	—	Scale to screen coordinates (Z) and set origin offset
G_LoadMatirix	—	Load geometric transformation matrix
G_ViewVolumeXYClip	—	Set boundary value (X, Y) of view volume clip
G_ViewVolumeZClip	—	Set boundary value (Z) of view volume clip
G_ViewVolumeWClip	—	Set boundary value (W) of view volume clip
OverlapXYOfft	See ***Command table***.	Sets XY offset at shading
OverlapZOfft	See ***Command table***.	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 ***Command table***.	Sets drawing extended mode register
SetGModeRegister	See ***Command table***.	Sets geometry extended mode register
SetColorRegister	See ***Command table***.	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_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_EndE	1110_0011

Geometry command code table

GMDR0.FX bit is expanded for CORAL-PA.

Work Only for G_Begin/Triangle(s,_Strip,_Fan)

(1) Integer setup type (for G_Begin)

In setup processing, “XY” is calculated in the integer format and other parameters are calculated in the floating-point format. (*.int)

or

In setup processing, “XY” is calculated in the floating-point format and other parameters are calculated in the floating-point format. (*.float)

Code	**Command(GMDR0.FX==0)**	**Command(GMDR0.FX==1)**
0001_0000	Points.int	Points.int
0001_0001	Lines.int	Lines.int
0001_0010	Polygon.int	Polygon.int
0001_0011	Triangles.int	Triangles.float
0001_0101	Line_Strip.int	Line_Strip.int
0001_0111	Triangle_Strip.int	Triangle_Strip.float
0001_1000	Triangle_Fan.int	Triangle_Fan.float

(2) “Unclipped” integer setup type(for G_Begin)

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. (*.int)

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

（GMDR0.FX has no mean for nclip）

Code	**Command(GMDR0.FX==0)**	**Command(GMDR0.FX==1)**
0011_0000	nclip_Points.int	nclip_Points.int
0011_0001	nclip_Lines.int	nclip_Lines.int
0011_0010	nclip_Polygon.int	nclip_Polygon.int
0011_0011	nclip_Triangles.int	nclip_Triangles.int
0011_0101	nclip_Line_Strip.int	nclip_Line_Strip.int
0011_0111	nclip_Triangle_Strip.int	nclip_Triangle_Strip.int
0011_1000	nclip_Triangle_Fan.int	nclip_Triangle_Fan.int

(3) Integer setup type (for G_BeginE)

In setup processing, "XY" is calculated in the integer format and other parameters are calculated in the floating-point format. (GMDR0.FX has no mean for G_BeginE)

Code	**Command(GMDR0.FX==0)**	**Command(GMDR0.FX==1)**
0001_0000	Points.int	Points.int
0001_0001	Lines.int	Lines.int
0001_0010	Polygon.int	Polygon.int
0001_0011	Triangles.int	Triangles.int
0001_0101	Line_Strip.int	Line_Strip.int
0001_0111	Triangle_Strip.int	Triangle_Strip.int
0001_1000	Triangle_Fan.int	Triangle_Fan.int

(4) "Unclipped" integer setup type(for G_BeginE)

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.

(GMDR0.FX has no mean for G_BeginE)

Code	**Command(GMDR0.FX==0)**	**Command(GMDR0.FX==1)**
0011_0000	nclip_Points.int	nclip_Points.int
0011_0001	nclip_Lines.int	nclip_Lines.int
0011_0010	nclip_Polygon.int	nclip_Polygon.int
0011_0011	nclip_Triangles.int	nclip_Triangles.int
0011_0101	nclip_Line_Strip.int	nclip_Line_Strip.int
0011_0111	nclip_Triangle_Strip.int	nclip_Triangle_Strip.int
0011_1000	nclip_Triangle_Fan.int	nclip_Triangle_Fan.int

11.2.2 Explanation of geometry commands

G_Nop (Format 1)

31 24	23 16	15 0
G_Nop	Reserved	Reserved

No operation

G_Init (Format 1)

31 24	23 16	15 0
G_Init	Reserved	Reserved

The ***G_Init*** command initializes geometry engine. Execute this command before processing.

G_End (Format 1)

31 24	23 16	15 0
G_End	Reserved	Reserved

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

G_Begin (Format 5)

31 … 24	23 … 16	15 … 0
G_Begin	Command	Reserved

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*** command and ***G_End*** command.

Only ***G_Vertex*** or ***SetRegister for FC/BC (XY Only vertex)*** can placed between ***G_Begin*** and ***G_End***.

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)	×	×	○	×

G_BeginE (Format 5)

31 24	23 16	15 0
G_Begin	Command	Reserved

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
- Polygon with Z or texture

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 and the ***G_EndE*** command.

Only ***G_Vertex/G_VertexLOG/G_VertexNopLOG*** or ***SetColorRegister(XY only vertex)*** or ***OverLapZofft*** can placed between ***G_BeginE*** and ***G_EndE***.

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

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			
A.int	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			
A.int	R.ing	G.int	B.int
S.fixed			
T.fixed			

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

- Required parameters depend on the setting of the ***GMDR0*** register. Proper values must be set as the mode values of the ***MDR0*** to ***MDR4*** registers to be finally reflected at drawing. That is, when "Z" comparison is made (ZC bit of MDR1 or MDR2 = 1), the Z bit of the GMDR0 register must be set to 1. When Gouraud shading is performed (SM bit of MDR2 = 1), the C bit of the GMDR0 register must be set to 1. When texture mapping is performed (TT bits of MDR2 = 10), the ST bit of the GMDR0 register must be set to 1.

- When the Z bit of the GMDR0 register is 0, input "Z" (Zoc) is treated as "0".

- Use values normalized to 0 and 1 as texture coordinates (S, T).

- When the color RGB is floating-point format, use values normalized to 0 and 1 as the 8-bit color value. For the packed RGB, use the 8-bit color value directly.

- The GMDR1 register is valid only for line drawing; it is ignored in primitives other than line.

- The GMDR2 register matters only when a triangle (excluding a polygon) is drawn. At primitives other than triangle, set "0".

- The use of both G_BeginE to G_EndE, and G_VertexLOG/NopLOG is not assured.

- G_VertexNopLOG, except for the primitive as point is not assured.
- A vertex data is processed at every time. For example, the Coral draws interpolation of bold line joint, edging line, shadows at every vertices.
- Alpha parameter can be provided only packed ARGB format.

G_Viewport (Format 1)

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

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/fixed		
Z_Offset.float/fixed		

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

G_LoadMatrix (Format 1)

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

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

G_ViewVolumeXYClip (Format 1)

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

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

G_ViewVolumeZClip (Format 1)

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

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

G_ViewVolumeWClip (Format 1)

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

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

OverlapXYOfft (Format5)

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

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

Command:

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

OverlapZOfft (Format5)

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

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

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

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

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

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

Command:

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

DC_LogOutAddr (Format5)

<table>
<tr><td colspan="2">31 24</td><td>23 16</td><td>15 0</td></tr>
<tr><td colspan="2">OverlapXYOfft</td><td>Command</td><td>Reserved</td></tr>
<tr><td>000000</td><td colspan="3">LogOutAddr</td></tr>
</table>

The **DC_LogOutAddr** command sets the starting address of the log output destination of the device coordinates.

SetModeRegister (Format5)

<table>
<tr><td>31 24</td><td>23 16</td><td>15 0</td></tr>
<tr><td>SetModeRegister</td><td>Command</td><td>Reserved</td></tr>
<tr><td colspan="3">MDR1*/MDR2*</td></tr>
</table>

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

Command:

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

SetGModeRegister (Format5)

<table>
<tr><td>31 24</td><td>23 16</td><td>15 0</td></tr>
<tr><td>SetGModeRegister</td><td>Command</td><td>Reserved</td></tr>
<tr><td colspan="3">GMDR1E/GMDR2E</td></tr>
</table>

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

Command:

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

SetColorRegister (Format5)

31 24 23 16 15 0

SetColorRegister	Command	Reserved
FGC8/16/24		

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

Commands:

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

SetRegister (Format 2)

31 24 23 16 15 0

SetRegister	Count	Address
(Val 0)		
(Val 1)		
…		
(Val n)		

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

SetLVertex2i (Format 1)

31 24 23 16 15 0

SetLVertex2i	Reserved	Reserved
LX0dc		
LY0dc		

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

SetLVertex2iP (Format 1)

31 24 23 16 15 0

SetLVertex2iP	Reserved	Reserved
LY0dc		LX0dc

The **SetLVertex2iP** command supports packed XY of SetLVertex2i.

11.3 Rendering Command

11.3.1 Command list

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

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

Type Code Table

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
BltCopyP	0000_1101
BltCopyAlternateP	0000_1111
LoadTextureP	0001_0001
BltTextureP	0001_0011
BltCopyAltAlphaBlendP	0001_1111
SetVertex2i	0111_0000
SetVertex2iP	0111_0001
Draw	1111_0000
SetRegister	1111_0001
Sync	1111_1100
Interrupt	1111_1101
Nop	1111_1111

Command Code Table (1)

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

Command Code Table (2)

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

11.3.2 Details of rendering commands

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

Nop (Format1)

31 24	23 16	15 0
Nop	Reserved	Reserved

No operation

Interrupt (Format1)

31 24	23 16	15 0
Interrupt	Reserved	Reserved

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

Sync (Format9)

31 24	23 16	15	4 0
Sleep	Reserved	Reserved	flag

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)

31 24	23 16	15	0
DrawLine2iP	Command	Reserved	vertex
LFYs		LFXs	

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. Please set XRES(X resolution) to in 8 byte units when using this command.

Commands:

BltFill	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. Must set RXs[3:0] and RsizeX[3:0] as 0000. (16pixel aligned) Drawing clipping is not work for this command.

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. Please set XRES(X resolution) to in 8 byte units when using this command.

Commands:

BltDraw	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. The “RsizeX” has to be up to 2016 in this command.

DrawBitmapLargeP (Format11)

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

The ***DrawBitmapP*** command draws rectangle patterns.

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

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

Please set XRES(X resolution) to in 8 byte units when using this command.

Commands:

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

BltCopyP (Format5)

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

The ***BltCopyP*** command copies rectangle pattern within drawing frame. Please set XRES(X resolution) to in 8 byte units when using this command.

Commands:

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

BltCopyAlternateP (Format5)

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

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

Please set XRES(X resolution) to in 8 byte units when using this command. And please set SStride and DStride to in 8 byte units.

Command:

TopLeft	Starts BitBlt transfer from top left coordinates. Drawing clipping is not wok for this command.

BltCopyAltAlphaBlendP (Format5)

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

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

Please set XRES(X resolution) to in 8 byte units when using this command. And please set SStride and BlendStride to in 8 byte units.

Command:

reserved	Set 0000_0000 to maintain future compatibility.

12. PCI Configuration Registers

For the Coral-PA, 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-PA 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.

12.1 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

12.2 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	201Eh	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-PA and should be set to ‘0’
8	RW	0	System Error Enable. This is supported by the Coral-PA.
7	-	0	Reserved
6	RW	0	Parity Error Enable. This is supported by the Coral-PA.
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-PA 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-PA 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-PA.
14	Status	0	System Error has been signaled by the Coral-PA.
13	Status	0	Received Master Abort. Set to ‘1’ when a PCI Master terminates a user to the Coral-PA transaction with Master Abort.
12	Status	0	Received Target Abort. Set to ‘1’ when the Coral-PA has initiated a transaction that has been terminated by Target Abort.
11	Status	0	Target Abort has been signaled by the Coral-PA.
10-9	RO	01	Device Select Timing. Indicates the timing of the **DEVSEL#** signal when the Coral-PA responds as a PCI Target.
8	Status	0	Data Parity Error detected.
7	RO	0	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-PA.

PCI Class Code Register

Bit	Type	Reset Value	Description
23-0	ER	038000h	Class Code of the Coral-PA. 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-PA is guaranteed access to the PCI bus. After the count has expired, the Coral-PA 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-PA, 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-PA non PCI register. The Coral-PA will respond as a Target to accesses in the address range: (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-PA is connected to. The default value of this indicate that the Coral-PA 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-PA 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-PA 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-PA 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-PA will perform when acting as a Bus Master.

User Programmable Register

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

13 Local Memory Registers

13.1 Local memory register list

13.1.1 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
0020	IST																															
	IST														IST												IST					
0024	IMASK																															
	IMASK														IMASK												IMASK					
002C																									SRST							
																																SRST
0038	CCF																															
													CGE		COT																	
005C																									RSW							
																																RSW
00A0	FRST																															
																																FRST
00A4	SRBS																															
																														SRBS		

<table>
<tr><td rowspan="2">00A8</td><td colspan="32">IOM</td></tr>
<tr><td></td><td></td><td colspan="14">GIM</td><td colspan="9">GD</td><td>SER</td><td>RGB</td><td>BEE</td><td>SBE</td><td>TCE</td><td>BCE</td><td>EEE</td></tr>
<tr><td rowspan="2">00AC</td><td colspan="32">GD</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="9">GWE</td><td></td><td></td><td colspan="14">GD</td></tr>
<tr><td rowspan="2">00B0</td><td colspan="32">SIC</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>CKP</td><td>CKG</td><td colspan="2">CKD</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>DOE</td><td></td><td></td><td></td><td></td><td></td><td>SD</td><td>SP</td><td>SL</td></tr>
<tr><td rowspan="2">00B4</td><td colspan="32">SID</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>FSL</td><td>FS</td><td colspan="8">TLS</td><td colspan="8">RWD</td></tr>
<tr><td rowspan="2">00F0</td><td colspan="32">CID</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="8">CN</td><td colspan="8">VER</td></tr>
<tr><td rowspan="2">8000</td><td colspan="32">BSA</td></tr>
<tr><td colspan="32">SA</td></tr>
<tr><td rowspan="2">8004</td><td colspan="32">BDA</td></tr>
<tr><td colspan="32">DA</td></tr>
<tr><td rowspan="2">8008</td><td colspan="32">BCR</td></tr>
<tr><td>STRT</td><td>NDA</td><td>NSA</td><td></td><td colspan="4">BSIZE</td><td colspan="24">TSIZE</td></tr>
<tr><td rowspan="2">800C</td><td colspan="32">BSR</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>XCOR</td><td>IMODE</td><td>TCM</td><td>BCM</td><td>EXTEN</td><td colspan="3">MODE</td></tr>
<tr><td rowspan="2">8010</td><td colspan="32">BER</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>ABORT</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>EXTST</td><td>BEN</td></tr>
<tr><td rowspan="2">8014</td><td colspan="32">BST</td></tr>
<tr><td>TC</td><td>BC</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="24">TCNT</td></tr>
<tr><td rowspan="2">8040
…
805C</td><td colspan="32">BCB</td></tr>
<tr><td colspan="32">RWDATA * 8</td></tr>
</table>

13.1.2 I^2C interface register list

Base = I^2CBase

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 Prohibitation																															
018	Access Prohibitation																															
01C	Access Prohibitation																															

13.1.3 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		TRRD		TRC				TRP		TRAS			TRCD		LOWD		RTS				SAW		ASW	CL		

13.1.4 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	DCM0 (Display Control Mode 0)																															
	DEN												L45E	L23E	L1E	L0E	CKS	DCS		SC					EEQ	EDE	EOF	EOD	SF	ESY	SYNC	
100	DCM1 (Display Control Mode 1)																															
	DEN										L5E	L4E	L3E	L2E	L1E	L0E	CKS	DCS	SC						EEQ	EDE	EOF	EOD	SF	ESY	SYNC	
104	DCM2 (Display Control Mode 2)																															
																															RUF	RUM
108	DCM3 (Display Control Mode 3)																															
																						POM	DCKed	DCKinv			DCKD					
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)											

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
020	L0M (L0 Mode)																															
	L0C								L0W (L0 memory Width)												L0H (L0 Height)											
024							L0OA0 (L0 Origin Address 0)																									
028							L0DA0 (L0 Display Address 0)																									
02C					L0DY (L0 Display Y)																L0DX (L0 Display X)											
110	L0EM (L0 Extend Mode)																															
	L0EC								L0PB																							L0WP
114					L0WY (L0 Window Y)																L0WX (L0 Window X)											
118					L0WH (L0 Window Height)																L0WW (L0 Window Width)											
030	L1M (L1 Mode)																															
	L1C	L1YC	L1CS	L1IM					L1W (L1 memory Width)																							
034							L1DA(L1 layer Display Address)																									
120	L1EM (L1 Extend Mode)																															
	L1EC			L1CP			VMAG		L1PB																							
124					L1WY (L1 Window Y)																L1WX (L1 Window X)											
128					L1WH (L1 Window Height)																L1WW (L1 Window Width)											
044	L2M (L2 Mode)																															
	L2C	L2FLP							L2W(L2 memory Width)												L0H (L0 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							L3W (L3 memory Width)												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							L4W (L4 memory Width)												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																														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							L5W (L5 memory Width)												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)											
160	L5EM (L5 Extend Mode)																															
	L5EC																														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									CPM								CUTC (Cursor Transparent Control)															
											CUE1	CUE0			CUO1	CUO0								CUZT	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)											
170	MDC (Multi Display Control)																															
	MDen																SC1en								SC0en							
180	DLS (Display Layer Select)																															
									DLS5				DLS4				DLS3				DLS2				DLS1				DLS0			
184									DBGC (Display Back Ground Color)																							
0B4	L0BLD (L0 Blend)																															
																L0BE	L0BS	L0BI	L0BP						L0BR							
188	L1BLD (L1 Blend)																															
																L1BE	L1BS	L1BI	L1BP						L1BR							
18C	L2BLD (L2 Blend)																															
																L2BE	L2BS	L2BI	L2BP						L2BR							
190	L3BLD (L3 Blend)																															
																L3BE	L3BS	L3BI	L3BP						L3BR							
194	L4BLD (L4 Blend)																															
																L4BE	L4BS	L4BI	L4BP						L4BR							
198	L5BLD (L5 Blend)																															
																L5BE	L5BS	L5BI							L5BR							

<table>
<tr><th>Offset</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="2">0BC</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="16">L0TC (L0 layer Transparent Control)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>L0ZT</td><td colspan="15">L0TC (L0 layer Transparent Color)</td></tr>
<tr><td rowspan="2">0C0</td><td colspan="16">L2TC (L2 layer Transparency Control)</td><td colspan="16">L3TC (L3 layer Transparency Control)</td></tr>
<tr><td>L2ZT</td><td colspan="15">L2TC (L2 layer Transparent Color)</td><td>L3ZT</td><td colspan="15">L3TC (L3 layer Transparent Color)</td></tr>
<tr><td rowspan="2">1A0</td><td colspan="32">L0ETC (L0 layer Extend Transparency Control)</td></tr>
<tr><td>L0EZT</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="24">L0ETC (L0 layer Extend Transparent Color)</td></tr>
<tr><td rowspan="2">1A4</td><td colspan="32">L1ETC (L1 layer Transparent Extend Control)</td></tr>
<tr><td>L1EZT</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="24">L1ETC (L1 layer Extend Transparent Color)</td></tr>
<tr><td rowspan="2">1A8</td><td colspan="32">L2ETC (L2 layer Transparent Extend Control)</td></tr>
<tr><td>L2EZT</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="24">L2ETC (L2 layer Extend Transparent Color)</td></tr>
<tr><td rowspan="2">1AC</td><td colspan="32">L3ETC (L3 layer Transparent Extend Control)</td></tr>
<tr><td>L3EZT</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="24">L3ETC (L3 layer Extend Transparent Color)</td></tr>
<tr><td rowspan="2">1B0</td><td colspan="32">L4ETC (L4 layer Extend Transparent Control)</td></tr>
<tr><td>L4EZT</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="24">L4ETC (L4 layer Extend Transparent Color)</td></tr>
<tr><td rowspan="2">1B4</td><td colspan="32">L5ETC (L5 layer Extend Transparent Control)</td></tr>
<tr><td>L5EZT</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="24">L5ETC (L5 layer Extend Transparent Color)</td></tr>
<tr><td rowspan="2">1E0</td><td colspan="32">L1YCR0 (L1 layer YC to Red Coefficient 0)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="11">a12</td><td></td><td></td><td></td><td></td><td></td><td colspan="11">a11</td></tr>
<tr><td rowspan="2">1E4</td><td colspan="32">L1YCR1 (L1 layer YC to Red Coefficient 1)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="9">b1</td><td></td><td></td><td></td><td></td><td></td><td colspan="11">a13</td></tr>
<tr><td rowspan="2">1E8</td><td colspan="32">L1YCG0 (L1 layer YC to Green Coefficient 0)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="11">a22</td><td></td><td></td><td></td><td></td><td></td><td colspan="11">a21</td></tr>
<tr><td rowspan="2">1EC</td><td colspan="32">L1YCG1 (L1 layer YC to Green Coefficient 1)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="9">b2</td><td></td><td></td><td></td><td></td><td></td><td colspan="11">a23</td></tr>
<tr><td rowspan="2">1F0</td><td colspan="32">L1YCB0 (L1 layer YC to Blue Coefficient 0)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td colspan="11">a32</td><td></td><td></td><td></td><td></td><td></td><td colspan="11">a31</td></tr>
<tr><td rowspan="2">1F4</td><td colspan="32">L1YCB1 (L1 layer YC to Blue Coefficient 1)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="9">b3</td><td></td><td></td><td></td><td></td><td></td><td colspan="11">a33</td></tr>
</table>

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
400	L0PAL0																															
	A								R								G								B							
404	L0PAL1																															
:	:																															
7FC	L0PAL255																															
800	L1PAL0																															
	A								R								G								B							
804	L1PAL1																															
:	:																															
BFC	L1PAL255																															
1000	L2PAL0																															
	A								R								G								B							
1004	L2PAL1																															
:	:																															
13FC	L2PAL255																															
1400	L3PAL0																															
	A								R								G								B							
1404	L3PAL1																															
:	:																															
17FC	L3PAL255																															

13.1.5 Video capture register list

Base = CaptureBase

<table>
<tr><th>Offset</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="2">000</td><td colspan="32">VCM (Video Capture Mode)</td></tr>
<tr><td>VIE</td><td>VIS</td><td></td><td>VIC</td><td></td><td></td><td colspan="2">CM</td><td></td><td></td><td></td><td>VI</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>NRGB</td><td>VS</td><td></td></tr>
<tr><td rowspan="2">004</td><td colspan="32">CSC (Capture SCale)</td></tr>
<tr><td colspan="5">VSCI</td><td colspan="11">VSCF</td><td colspan="5">HSCI</td><td colspan="11">HSCF</td></tr>
<tr><td rowspan="2">008</td><td colspan="32">VCS (Video Capture Status)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="5">CE</td></tr>
<tr><td rowspan="2">010</td><td colspan="32">CBM (Capture Buffer Mode)</td></tr>
<tr><td>OOM</td><td>SBUF</td><td>CRGB</td><td></td><td></td><td></td><td></td><td></td><td colspan="8">CBW (stride)</td><td></td><td>C24</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>CBST</td></tr>
<tr><td>014</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="22">CBOA (Capture Buffer Origin Address)</td></tr>
<tr><td>018</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="22">CBLA (Capture Buffer Limit Address)</td></tr>
<tr><td>01C</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="10">CIVSTR</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="10">CIHSTR</td></tr>
<tr><td>020</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="10">CIVEND</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="10">CIHEND</td></tr>
<tr><td rowspan="2">028</td><td colspan="32">CHP (Capture Horizontal Pixel)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="10">CHP</td></tr>
<tr><td rowspan="2">02C</td><td colspan="32">CVP (Capture Vertical Pixel)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="10">CVPP</td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="10">CVPN</td></tr>
<tr><td rowspan="2">048</td><td colspan="32">CMSS (Capture Magnify Source Size)</td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="12">CMSHP</td><td></td><td></td><td></td><td></td><td colspan="12">CMSVL</td></tr>
<tr><td rowspan="2">040</td><td colspan="32">CLPF (Capture Low Pass Filter)</td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="4">CVLPF</td><td></td><td></td><td></td><td></td><td colspan="4">CHLPF</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td rowspan="2">04C</td><td colspan="32">CMDS (Capture Magnify Display Size)</td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="12">CMDHP</td><td></td><td></td><td></td><td></td><td colspan="12">CMDVL</td></tr>
<tr><td rowspan="2">080</td><td colspan="32">RGBHC(RGB input HSYNC Cycle)/VIN_HSSIZE</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="14">RGBHC</td></tr>
<tr><td rowspan="2">084</td><td colspan="32">RGBHEN(RGB input Horizontal Enable Area)</td></tr>
<tr><td></td><td></td><td></td><td></td><td colspan="12">RGBHST</td><td></td><td></td><td colspan="14">RGBHEN</td></tr>
<tr><td rowspan="2">088</td><td colspan="32">RGBVEN(RGB input Vertical Enable Area)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="9">RGBVST</td><td></td><td></td><td></td><td colspan="13">RGBVEN</td></tr>
<tr><td rowspan="2">090</td><td colspan="32">RGBS(RGB input SYNC)</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>RM</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td>HP</td><td>VP</td></tr>
<tr><td rowspan="2">0C0</td><td colspan="32">RGBCMY(RGB Color convert Matrix Y coefficient)</td></tr>
<tr><td colspan="10">a11</td><td></td><td colspan="10">a12</td><td></td><td colspan="10">a13</td></tr>
<tr><td rowspan="2">0C4</td><td colspan="32">RGBCMCb(RGB Color convert Matrix Cb coefficient)</td></tr>
<tr><td colspan="10">a21</td><td></td><td colspan="10">a22</td><td></td><td colspan="10">a23</td></tr>
<tr><td rowspan="2">0C8</td><td colspan="32">RGBCMCr(RGB Color convert Matrix Cr coefficient)</td></tr>
<tr><td colspan="10">a31</td><td></td><td colspan="10">a32</td><td></td><td colspan="10">a33</td></tr>
<tr><td rowspan="2">0CC</td><td colspan="32">RGBCMb(RGB Color convert Matrix b coefficient)</td></tr>
<tr><td></td><td colspan="9">b1</td><td></td><td></td><td colspan="9">b2</td><td></td><td></td><td colspan="9">b3</td></tr>
</table>

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
4000	CDCN(Capture Data Count for NTSC)																															
				BDCN																VDCN												
4004	CDCP(Capture Data Count for PAL)																															
				BDCP																VDCP												

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

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
080	Zs																															
(020)	0	Int																Frac														
084	dZdx																															
(021)	s	Int																Frac														
088	dZdy																															
(022)	s	Int																Frac														
0C0	Ss																															
(030)	s	s	s	Int													Frac															
0C4	dSdx																															
(031)	s	s	s	Int													Frac															
0C8	dSdy																															
(032)	s	s	s	Int													Frac															
0CC	Ts																															
(033)	s	s	s	Int													Frac															
0D0	dTdx																															
(034)	s	s	s	Int													Frac															
0D4	dTdy																															
(035)	s	s	s	Int													Frac															
0D8	Qs																															
(036)	0	0	0	0	0	0	0	INT	Frac																							
0DC	dQdx																															
(037)	s	s	s	s	s	s	s	INT	Frac																							
0E0	dQdx																															
(038)	s	s	s	s	s	s	s	INT	Frac																							
140	LPN																															
(050)	0	0	0	0	Int												0															
144	LXs																															
(051)	s	s	s	s	Int												Frac															
148	LXde																															
(052)	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	Int	Frac															
14C	LYs																															
(053)	s	s	s	s	Int												Frac															
150	LYde																															
(054)	s	s	s	s	s	s	s	s	s	s	s	s	s	s	s	Int	Frac															
154	LZs																															
(055)	s	Int																Frac														
158	LZde																															
(056)	s	Int																Frac														

<table>
<tr><th>Offset</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="2">180
(060)</td><td colspan="32">PXdc</td></tr>
<tr><td>s</td><td>s</td><td>s</td><td>s</td><td colspan="12">Int</td><td colspan="16">Frac</td></tr>
<tr><td rowspan="2">184
(061)</td><td colspan="32">PYdc</td></tr>
<tr><td>s</td><td>s</td><td>s</td><td>s</td><td colspan="12">Int</td><td colspan="16">Frac</td></tr>
<tr><td rowspan="2">188
(062)</td><td colspan="32">PZdc</td></tr>
<tr><td>s</td><td colspan="16">Int</td><td colspan="15">Frac</td></tr>
<tr><td rowspan="2">200
(080)</td><td colspan="32">RXs</td></tr>
<tr><td>s</td><td>s</td><td>s</td><td>s</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">204
(081)</td><td colspan="32">RYs</td></tr>
<tr><td>s</td><td>s</td><td>s</td><td>s</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">208
(082)</td><td colspan="32">RsizeX</td></tr>
<tr><td>s</td><td>s</td><td>s</td><td>s</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">20C
(083)</td><td colspan="32">RsizeY</td></tr>
<tr><td>s</td><td>s</td><td>s</td><td>s</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">240
(090)</td><td colspan="32">SADDR</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="25">Address</td></tr>
<tr><td rowspan="2">244
(091)</td><td colspan="32">SStride</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">248
(092)</td><td colspan="32">SRXs</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">24C
(093)</td><td colspan="32">SRYs</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">250
(094)</td><td colspan="32">DADDR</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="25">Address</td></tr>
<tr><td rowspan="2">254
(095)</td><td colspan="32">DStride</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">258
(096)</td><td colspan="32">DRXs</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">25C
(097)</td><td colspan="32">DRYs</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">250
(098)</td><td colspan="32">BRsizeX</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">254
(099)</td><td colspan="32">BRsizeY</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="12">Int</td><td colspan="16">0</td></tr>
<tr><td rowspan="2">258
(09A)</td><td colspan="32">TColor</td></tr>
<tr><td colspan="16">0</td><td colspan="16">Color</td></tr>
<tr><td rowspan="2">3E0
(0f8)</td><td colspan="32">BLPO</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="5">BCR</td></tr>
</table>

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
400	CTR																															
(100)								FD	FE	CE		FCNT						NF	FF	FE			SS				DS				PS	
404	IFSR																															
(–)																														FD	FE	CE
408	IFCNT																															
(–)																											FCNT					
40C	SST																															
(–)																															SS	
410	DS																															
(–)																															DS	
414	PST																															
(–)																															PS	
418	EST																															
(–)																														FD	PE	CE
420	MDR0																															
(108)												ZP				CF							CY	CX					BSV		BSH	
424	MDR1/MDR1S/MDR1B/MDR1TL																															
(109)				LW								BP	BL							LOG				BM		ZW	ZCL			ZC	AS	SM
428	MDR2/MDR2S/MDR2TL																															
(10a)			TT																	LOG				BM		ZW	ZCL			ZC	AS	SM
42C	MDR3																															
(10b)								BA			TAB				TBL						TWS		TWT				TF		TC			
430	MDR4																															
(10c)																				LOG				BM							TE	
43C	MDR7																															
(10f)																										LTH	EZ	GG		PGH	PTH	PZH

<table>
<tr><th>Offset</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td rowspan="2">440
(110)</td><td colspan="32">FBR</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="26">FBASE</td></tr>
<tr><td rowspan="2">444
(111)</td><td colspan="32">XRES</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="12">XRES</td></tr>
<tr><td rowspan="2">448
(112)</td><td colspan="32">ZBR</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="26">ZBASE</td></tr>
<tr><td rowspan="2">44C
(113)</td><td colspan="32">TBR</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="26">TBASE</td></tr>
<tr><td rowspan="2">450
(114)</td><td colspan="32">PFBR</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="26">PFBASE</td></tr>
<tr><td rowspan="2">454
(115)</td><td colspan="32">CXMIN</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="12">CLIPXMIN</td></tr>
<tr><td rowspan="2">458
(116)</td><td colspan="32">CXMAX</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="12">CLIPXMAX</td></tr>
<tr><td rowspan="2">45C
(117)</td><td colspan="32">CYMIN</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="12">CLIPYMIN</td></tr>
<tr><td rowspan="2">460
(118)</td><td colspan="32">CYMAX</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="12">CLIPYMAX</td></tr>
<tr><td rowspan="2">464
(119)</td><td colspan="32">TXS</td></tr>
<tr><td></td><td></td><td></td><td colspan="13">TXSN</td><td></td><td></td><td></td><td colspan="13">TXSM</td></tr>
<tr><td rowspan="2">468
(11a)</td><td colspan="32">TIS</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="7">TISN</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="7">TISM</td></tr>
<tr><td rowspan="2">46C
(11b)</td><td colspan="32">TOA</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="12">XBO</td></tr>
<tr><td rowspan="2">470
(11C)</td><td colspan="32">SHO</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="25">SHOFFS</td></tr>
<tr><td rowspan="2">474
(11D)</td><td colspan="32">ABR</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="26">ABASE</td></tr>
<tr><td rowspan="2">480
(120)</td><td colspan="32">FC</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="8"></td><td colspan="8">FGC8/16/24</td><td colspan="8"></td></tr>
<tr><td rowspan="2">484
(121)</td><td colspan="32">BC</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="8"></td><td colspan="8">BGC8/16/24</td><td colspan="8"></td></tr>
<tr><td rowspan="2">488
(122)</td><td colspan="32">ALF</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="8">A</td></tr>
<tr><td>48C
(123)</td><td colspan="32">BLP</td></tr>
<tr><td rowspan="2">494
(129)</td><td colspan="32">TBC</td></tr>
<tr><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td colspan="8"></td><td colspan="16">BC16/24</td></tr>
</table>

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															

13.1.7 Geometry engine register list

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

Base = GeometryBase

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 (–)	GCTR																															
								FO				FCNT						NF	FF	FE			GS				SS				PS	
040 (2010)	GMDR0																															
				FX																					CF	DF			ST	Z	C	F
044 (2011)	GMDR1																															
																												BO		EP		AA
–	GMDR1E																															
	PO	LV										TC				BC		UW	BM	TM			BP	SP				BO		EP		AA
048 (2012)	GMDR2																															
																														FD		CF
–	GMDR2E																															
																						TL		SP						FD		CF
400 (–)	DFIFOG																															

13.2 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.

13.2.1 Host interface registers

IST (Interrupt STatus)

Register address	HostBaseAddress + 20_H									
Bit number	31:30	29:28	27:26	25	24:23:22:21:20:19:18	17:16	15:14:13:12:11:10:9	8:7:6	5:4:3:2	1:0
Bit field name	IST			*1	Reserved	Resv	Reserved	Reserved	IST	IST
R/W	RW0	R	RW0	R0	R0	R0W0	R0	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 5	REGUD (Register update) Indicates register update 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 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	Reserved	Resv	Reserved	Reserved	IMASK	IMASK
R/W	RW	R0	R0	R0W0	R0	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 5	REGUD (Register update) Masks register update 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 + $2C_H$							
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_H					
Bit number	31 30 29 28 27 26 25 24 23 22 21 20	19	18	17	16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	CGE		COT		Reserved
R/W	RW0	RW		RW		RW0
Initial value	0	00		00		0

This register changes the operating frequency.

Bit 19 and 18	CGE (Clock select for Geometry Engine)
	Selects the clock for the geometry engine
	11 Reserved
	10 166 MHz
	01 133 MHz
	00 100 MHz

Bit 17 and 16	COT (Clock select for the others except-geometry engine)
	Selects the clock for other than the geometry engine
	11 Reserved
	10 Reserved
	01 133 MHz
	00 100 MHz

Notes:

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

RSW (Register location Switch)

Register address	HostBaseAddress + 5C$_H$							
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.

FRST (Firm ReSeT)

Register address	HostBaseAddress + 00A0$_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved																															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 Read Burst 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. However, it is set to 2-8 only at the time of “Slave Mode Coral PA to PCI”.

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	*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 PA 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 it’s reset value from the Transfer Complete pin at system reset. Note that if the RGB input is enabled then the EEPROM interface us 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 BC 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 BC pin operates as an RGB input pin.

Bit 2 TCE (Transfer Complete Enable)

If set to “1b” then the TC 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 SB 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 SB pin operates as an RGB input pin.

Bit 4 BEE (Burst Enable Enable)

If set to “1b” then the BEN 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 Bit 7	GD (GPIO Direction) 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: BC Bit 13: TC Bit 14: SB Bit 15: BEN
Bit 29 to Bit 16	GIM (GPIO Interrupt Mask) 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: BC Bit 22: TC Bit 23: SB Bit 24: BEN Bit 25: GI0 Bit 26: GI1 Bit 27: GI2 Bit 28: GI3 Bit 29: GI4

GD (GPIO Data)

Register address	HostBaseAddress + 00AC$_H$			
Bit number	31 30 29 28 27 26 25	24 23 22 21 20 19 18 17 16	15 14	13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	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: BC
Bit 6: TC
Bit 7: SB
Bit 8: BEN
Bit 9: GI0
Bit 10: GI1
Bit 11: GI2
Bit 12: GI3
Bit 13: GI4

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: BC
Bit 22: TC
Bit 23: SB
Bit 24: BEN

SIC (Serial Interface Control)

<table>
<tr><td>Register address</td><td colspan="32">HostBaseAddress + 00B0$_H$</td></tr>
<tr><td>Bit number</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Bit field name</td><td colspan="12">Reserved</td><td>CKP</td><td>CKG</td><td colspan="2">CKD</td><td colspan="7">Reserved</td><td>DOE</td><td colspan="5">Reserved</td><td>SD</td><td>SP</td><td>SL</td></tr>
<tr><td>R/W</td><td colspan="12">R0</td><td>RW</td><td>RW</td><td colspan="2">RW</td><td colspan="7">R0</td><td>RW</td><td colspan="5">R0</td><td>RW</td><td>RW</td><td>RW</td></tr>
<tr><td>Initial value</td><td colspan="12">0</td><td>0</td><td>0</td><td colspan="2">0</td><td colspan="7">0</td><td>0</td><td colspan="5">0</td><td>0</td><td>0</td><td>0</td></tr>
</table>

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 foe 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$_H$				
Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18	17	16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Bit field name	Reserved	FSL	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 “00000000b”. For example, to specify a transfer of 8 bits “00000001b” should be written. To specify a transfer of 3 bits “00100000” 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$_H$		
Bit number	31 : 30 : 29 : 28 : 27 : 26 : 25 : 24 : 23 : 22 : 21 : 20 : 19 : 18 : 17 : 16	15 : 14 : 13 : 12 : 11 : 10 : 9 : 8	7 : 6 : 5 : 4 : 3 : 2 : 1 : 0
Bit field name	Reserved	CN	VER
R/W	R0	R	R
Initial value	0	0000_0011	0000_1000

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 for LB
0000_0100	Reserved
0000_0101	Reserved
0000_0110	Reserved for LP (Coral LP value)
others	Reserved
0000_1000	Reserved for PA (Coral PA value)

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_H	
Bit number	31:30:29:28:27:26:25:24	23:22:21:20:19:18:17:16:15:14:13:12:11:10:9:8:7:6:5:4:3:2:1:0
Bit field name	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_H	
Bit number	31:30:29:28:27:26:25:24	23:22:21:20:19:18:17:16:15:14:13:12:11:10:9:8:7:6:5:4:3:2:1:0
Bit field name	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_H					
Bit number	31	30	29	28	27:26:25:24	23:22:21:20:19:18:17:16:15:14:13:12:11:10:9:8:7:6:5:4:3:2:1:0
Bit field name	STRT	NDA	NSA	*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. When Coral PA is master (MODE of BSR register : 100b,101b,110b), Coral PA can not issue an odd address to PCI area. If the beginning address is set to the odd address in 64-bit boundary, Coral PA issues the previous enen address. Note: The odd address in 64-bit boundary means 0x004, 0x00C, 0x014….

In master read, Coral PA begins to read the previous even address and read the setting of burst size(BSIZE of BCR register) plus 1.

In master write, Coral PA begins to write the previous even address with disable write byte enable and write the setting of burst size(BSIZE of BCR register) plus 1.

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 + $800C_H$						
Bit number	31:30:29:28:27:26:25:24 23:22:21:20:19:18:17:16:15:14:13:12:11:10: 9 : 8	7	6	5	4	3	2 : 1 : 0
Bit field name	Reserved	XCOR	IMODE	TCM	BCM	EXTEN	MODE
R/W	R0	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 BEN (Burst Enable) signal may be used to initiate and pause a transfer. Otherwise if set to “0b” the external BEN signal is ignored.

Bit 4 BCM (Burst Complete Mask)

If set to “1b” then the external BC 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 TC 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 BC/TC 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_H						
Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17	16	15	14 13 12 11 10 9 8	7 6 5 4 3 2	1	0
Bit field name	Reserved	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_H			
Bit number	31	30	29 28 27 26 25 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	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_H
Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	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 PA in PCI Slave mode with automatic pre-fetch/write of data with address incrementing.

13.2.2 I^{2}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 wad 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 (INTrrupt) 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

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

high-speed-mode

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

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.

13.2.3 Graphics memory interface registers

MMR (Memory I/F Mode Register)

Register address	HostBaseAddress + FFFC$_H$														
Bit number	31	30	29 : 28 : 27	26	25	24 : 23	22 : 21 : 20 : 19	18 : 17	16 : 15 : 14	13 : 12	11 : 10	9 : 8 : 7	6 : 5 : 4	3	2 : 1 : 0
Bit field name	*1	tWR	Reserved	*1	*1	TRRD	TRC	TRP	TRAS	TRCD	LOWD	RTS	SAW	ASW	CL
R/W	RW	RW	R	R1 W0	R	RW	RW	RW	RW	RW	RW	RW	RW	RW	RW
Initial value	0	0	Don't care	1	0	00	0000	00	000	00	00	000	000	0	000

*1: Reserved

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

This register is not initialized at a software reset.

Bit 2 to 0 CL (CAS Latency)

Sets the CAS latency. Write the same value as this field, to the mode register for SDRAM

011	CL3
010	CL2
Other than the above	Setting disabled

Bit 3 ASW (Attached SDRAM bit Width)

Sets the bit width of the data bus (memory bus width mode)

1	64 bit
0	32 bit

Bit 6 to 4 SAW (SDRAM Address Width)

Sets the bit width of the SDRAM address

001	15 bit BANK 2 bit ROW 13 bit COL 9 bit SDRAM
111	14 bit BANK 2 bit ROW 12 bit COL 9 bit SDRAM
110	14 bit BANK 2 bit ROW 12 bit COL 8 bit SDRAM
101	13 bit BANK 2 bit ROW 11 bit COL 8 bit SDRAM
100	12 bit BANK 1 bit ROW 11 bit COL 8 bit FCRAM
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

Bit 13 and 12 TRCD

Sets the wait time secured from the bank active to CAS. The clock count is used to express the wait time.

11	3 clocks
10	2 clocks
01	1 clock
00	0 clock

Bit 16 to 14 TRAS

Sets the minimum time for 1 bank active. The clock count is used to express the minimum time.

111	7 clocks
110	6 clocks
101	5 clocks
100	4 clocks
011	3 clocks
010	2 clocks
Other than the above	Setting disabled

Bit 18 and 17 TRP

Sets the wait time secured from the pre-charge to the bank active. The clock count is used to express the wait time.

11	3 clocks
10	2 clocks
01	1 clock

Bit 22 to 19 TRC

This field sets the wait time secured from the refresh to the bank active. The clock count is used to express the wait time.

1010	10 clocks
1001	9 clocks
1000	8 clocks
0111	7 clocks
0110	6 clocks
0101	5 clocks
0100	4 clocks

0011	3 clocks
Other than the above	Setting disabled

Bit 24 and 23 TRRD

Sets the wait time secured from the bank active to the next bank active. The clock count is used to express the wait time.

11	3 clocks
10	2 clocks

Bit 26 Reserved

Always write “0” at write.

“1” is always read at read.

Bit 30 TWR

Sets the write recovery time (the time from the write command to the read or to the pre-charge command).

1	2 clocks
0	1 clock

13.2.4 Display control register

DCM0/1 (Display Control Mode 0/1)

Register address	DisplayBaseAddress + 00_H (DCM0)																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DEN	STOP	Reserve										L45E	L23E	L1E	L0E	CKS	Resv		SC					EEQ	ODE	Resv	Resv	SF	ESY	SYNC	
R/W	RW	RW	RX										RW	RW	RW	RW	RW	R0		RW					RW	RW	RW	R0	RW	RW	RW	
Initial value	0	0											0	0	0	0	0	0		1110					0	0	0	0	0	0	00	

Register address	DisplayBaseAddress + 100_H (DCM1)																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DEN	STOP	Reserve								L5E	L4E	L3E	L2E	L1E	L0E	CKS	Resv	SC						EEQ	ODE	Resv	Resv	SF	ESY	SYNC	
R/W	RW	RW	RX								RW	RW	RW	RW	RW	RW	RW	R0	RW						RW	RW	RW	R0	RW	RW	RW	
Initial value	0	0	X								0	0	0	0	0	0	0	0	11101						0	0	0	0	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 but it is one substance. The differences between the two registers are the format of the frequency division rate setting (SC) and layer enable. The two formats exist to maintain backword compatibility with previous products.

Bit 1 to 0 SYNC (Synchronize)

Set synchronization mode

X0 Non-interlace mode

10 Interlace mode

11 Interlace video mode

Bit 2 ESY (External Synchronize)

Sets external synchronization mode

0: External synchronization disabled

1: External synchronization enabled

Bit 3 SF (Synchronize signal format)

Sets format of synchronization (VSYNC, HSYNC) signals

0: Negative logic

1: Positive logic

Bit 7 EEQ (Enable Equalizing pulse)

Sets CCYNC signal mode

0: Does not insert equalizing pulse into CCYNC signal

1: Inserts equalizing pulse into CCYNC signal

Bit 13 to 8 SC (Scaling)

Divides display reference clock by the preset ratio to generate dot clock

Offset = 0		Offset = 100_H	
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

Bit 16 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 17 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 18 L23E (L2 & L3 layer Enable) ------ DCM0

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

L2E (L2 layer Enable) ------ DCM1

Enables L2 layer display

0: Does not display L2 layer

1: Displays L2 layer

Bit 19 L45E (L4 & L5 layer Enable) ------ DCM0

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

L3E (L3 layer Enable)) ------ DCM1

Enables L3 layer display

0: Does not display L3 layer

1: Displays L3 layer

Bit 20 L4E (L4 layer Enable)

Enables L4 layer display

0: Does not display L4 layer

1: Displays L4 layer

Bit 21 L5E (L5 layer Enable)

Enables L5 layer display

0: Does not display L5 layer

1: Displays L5 layer

Bit 31 DEN (Display Enable)

Enables display

0: Does not output display signal

1: Outputs display signal

DCM2 (Display Control Mode 2)

Register address	DisplayBaseAddress + 104_H				
Bit number	31 30 29 28 27 28 17 16 15 14 13 12 11 10 9 8 7 6 5	4	3 2	1	0
Bit field name	Reserve	Reserv	Reserv	RUF	RUM
R/W	R0	R0	R0	RW	RW
Initial value	0	0	0	0	0

Bit0 RUM (Register Update Mode)

The mode reflects the register value synchronizing with vertical synchronization is selected.

0: The register update is in real time reflected in the internal control circuit. The display falls into disorder when updating it for the display period.

1: It value of the register spreads to the internal control circuit synchronizing with vertical synchronization. The simultaneity is controlled with the following RUF flags.

Bit1 RUF (Register Update Flag)

The value is directed to be updated in the following vertical synchronization in writing 1 in this flag. If the update ends, it becomes 0.

0: Initial or update end

1: Vertical synchronous waiting

DCM3 (Display Control Mode 3)

Register address	DisplayBaseAddress + 108_H							
Bit number	31 30 29 17 16 15 14 13 12	11	10	9	8	7	6	5 4 3 2 1 0
Bit field name	reserve	resv	POM	DCKed	DCKinv	Reserve		DCKD
R/W	R0	R0	RW	RW	RW	RW0	R0	RW
Initial value	0	0	0	0	0	0	0	000000

Bit5-0 DCKD (Display Clock Delay)

This defines additional delay time by internal PLL clock period.

000000	No additional delay
000010	+2 PLL clock
000100	+3 PLL clock
000110	+4 PLL clock
:	:
111110	+33 PLL clock
xxxxx1	reserve

Bit8 DCKinv (Display Clock inversion)

0: DCLKO output signal is not inverted

1: DCLKO output signal is inverted.

Bit9 DCKed (Display clock edge)

This defines which edge mode is used.

0: single edge mode in which positive edge is used for digital RGB output.

1: bi-edge mode in which positive edge and negative edge are used for digital RGB output to identify two data streams.

Bit10 POM (Parallel output Mode)

This defines a way to output two data streams for two display

0: multiplex output mode in which two data streams are multiplexed and goes to the digital RGB output.

1: parallel output mode in which one data stream go to the digital RGB output and another data stream goes to the analog RGB output.

HTP (Horizontal Total Pixels)

Register address	DisplayBaseAddress + 06_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				HTP											
R/W	R0				RW											
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_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				HDP											
R/W	R0				RW											
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 + $0A_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				HDB											
R/W	R0				RW											
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 + $0C_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				HSP											
R/W	R0				RW											
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 + $0E_H$							
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 + $0F_H$							
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_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				VTR											
R/W	R0				RW											
Initial value	0				Don't care											

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

VSP (Vertical Synchronize pulse Position)

Register address	DisplayBaseAddress + 14_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				VSP											
R/W	R0				RW											
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_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				VDP											
R/W	R0				RW											
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$_H$					
Bit number	31	30 29	28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	L0C	Reserved	Reserved	L0W	Reserved	L0H
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$_H$				
Bit number	31 30	29 28 27 26 25 24	23 22 21 20	19 18 17 16 15 14 13 12 11 10 ----- 4 3 2 1	0
Bit field name	L0EC	Reserved	L0PB	Reserved	L0WP
R/W	RW	R0	RW	R0	RW
Initial value		0		0	0

Bit 0 L0 WP (L0 layer Window Position enable)

Selects the display position of L0 layer

0 Compatibility mode display (C layer supported)

1 Window display

Bit 23 to 20 L0PB (L0 layer Palette Base)

Shows the value added to the index when subtracting palette of L0 layer. 16 times of setting value is added.

Bit 31 and 30 L0EC (L0 layer Extended Color mode)

Sets extended color mode for L0 layer

00 Mode determined by L0C

01 Direct color (24 bits/pixel) mode

1x Reserved

L0OA (L0 layer Origin Address)

<table>
<tr><td>Register address</td><td colspan="32">DisplayBaseAddress + 24_H</td></tr>
<tr><td>Bit number</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Bit field name</td><td colspan="6">Reserved</td><td colspan="26">L0OA</td></tr>
<tr><td>R/W</td><td colspan="6">R0</td><td colspan="22">RW</td><td colspan="4">R0</td></tr>
<tr><td>Initial value</td><td colspan="6">0</td><td colspan="22">Don’t care</td><td colspan="4">0000</td></tr>
</table>

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)

<table>
<tr><td>Register address</td><td colspan="32">DisplayBaseAddress + 28_H</td></tr>
<tr><td>Bit number</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Bit field name</td><td colspan="6">Reserved</td><td colspan="26">L0DA</td></tr>
<tr><td>R/W</td><td colspan="6">R0</td><td colspan="26">RW</td></tr>
<tr><td>Initial value</td><td colspan="6">0</td><td colspan="26">Don’t care</td></tr>
</table>

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)

<table>
<tr><td>Register address</td><td colspan="16">DisplayBaseAddress + $2C_H$</td></tr>
<tr><td>Bit number</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Bit field name</td><td colspan="4">Reserved</td><td colspan="12">L0DX</td></tr>
<tr><td>R/W</td><td colspan="4">R0</td><td colspan="12">RW</td></tr>
<tr><td>Initial value</td><td colspan="4">0</td><td colspan="12">Don’t care</td></tr>
</table>

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)

<table>
<tr><td>Register address</td><td colspan="16">DisplayBaseAddress + $2E_H$</td></tr>
<tr><td>Bit number</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Bit field name</td><td colspan="4">Reserved</td><td colspan="12">L0DY</td></tr>
<tr><td>R/W</td><td colspan="4">R0</td><td colspan="12">RW</td></tr>
<tr><td>Initial value</td><td colspan="4">0</td><td colspan="12">Don’t care</td></tr>
</table>

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.

L0WX (L0 layer Window position X)

Register address	DisplayBaseAddress + 114_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L0WX											
R/W	R0				RW											
Initial value	0															

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

L0WY (L0 layer Window position Y)

Register address	DisplayBaseAddress + 116_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L0WY											
R/W	R0				RW											
Initial value	0															

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

L0WW (L0 layer Window Width)

Register address	DisplayBaseAddress + 118_H															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L0WW											
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”.

L0WH (L0 layer Window Height)

Register address	DisplayBaseAddress + $11A_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L0WH											
R/W	R0				RW											
Initial value	0				Don’t care											

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

L1M (L1-layer Mode)

Register address	DisplayBaseAddress + 30_H						
Bit number	31	30	29	28	27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 – – – 5 4 3 2 1 0
Bit field name	L1C	L1YC	L1CS	L1IM	Reserved	L1W	Reserved
R/W							R0
Initial value							0

Bit 23 to 16 L1W (L1 layer memory Width)

Sets the memory width (stride) of the logic frame of the W layer in unit of 64 bytes

Bit 28 L1IM (L1 layer Interlace Mode)

Sets video capture mode when L1CS in capture mode

0: Normal mode

1: For non-interlace display, displays captured video graphics in WEAVE mode
For interlace and video display, buffers are managed in frame units (pair of odd field and even field).

Bit 29 L1CS (L1 layer Capture Synchronize)

Sets whether the layer is used as normal display layer or as video capture

0: Normal mode

1: Capture mode

Bit 30 L1YC (L1 layer YC mode)

Sets color format of L1 layer
The YC mode must be set for video capture.

0: RGB mode

1: YC mode

Bit 31 L1C (L1 layer Color mode)
Sets color mode for L1 layer

0: Indirect color (8 bits/pixel) mode

1: Direct color (16 bits/pixel) mode

L1EM (L1 layer Extended Mode)

Register address	DisplayBaseAddress + 120$_H$																											
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	– – –	4	3	2	1	0
Bit field name	L1EC		Reserved				DM		L1PB				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 25 to 24 L1DM (L1 layer Display Magnify Mode)

00 Normal Mode (no scaling or shrink scaling)

01 Reserved

10 Magnify Scaling

11 Reserved

Bit 31 to 30 L1EC (L1 layer Extended Color mode) Sets extended color mode for L1 layer

00 Mode determined by L1C

01 Direct color (24 bits/pixel) mode

1x Reserved

L1DA (L1 layer Display Address)

Register address	DisplayBaseAddress + 34$_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						L1DA																									
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$_H$ (DisplayBaseAddress + 18$_H$)															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L1WX											
R/W	R0				RW											
Initial value	0				Don’t care											

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

L1WY (L1 layer Window position Y)

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

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

L1WW (L1 layer Window Width)

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

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

L1WH (L1 layer Window Height)

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

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

L2M (L2 layer Mode)

Register address	DisplayBaseAddress + 40_H					
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)

<table>
<tr><td>Register address</td><td colspan="28">DisplayBaseAddress + 130_H</td></tr>
<tr><td>Bit number</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>-----</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Bit field name</td><td colspan="2">L2EC</td><td colspan="6">Reserved</td><td colspan="4">L2PB</td><td colspan="14">Reserved</td><td>L2OM</td><td>L0WP</td></tr>
<tr><td>R/W</td><td colspan="2">RW</td><td colspan="6">R0</td><td colspan="4">RW</td><td colspan="14">R0</td><td>RW</td><td>RW</td></tr>
<tr><td>Initial value</td><td colspan="2">00</td><td colspan="6">0</td><td colspan="4">0</td><td colspan="14">0</td><td>0</td><td>0</td></tr>
</table>

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_H		
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4	3 2 1 0
Bit field name	Reserved	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_H	
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	L2DA0
R/W	R0	RW
Initial value	0	Don’t care

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

L2OA1 (L2 layer Origin Address 1)

Register address	DisplayBaseAddress + $4C_H$		
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4	3 2 1 0
Bit field name	Reserved	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_H	
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	L2DA1
R/W	R0	RW
Initial value	0	Don’t care

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

L2DX (L2 layer Display position X)

Register address	DisplayBaseAddress + 54_H	
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_H															
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_H															
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_H															
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 + $13A_H$															
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 + $13C_H$															
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_H					
Bit number	31	30 29	28 27 26 25 24 23 22	21 20 19 18 17 16	15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	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$_H$					
Bit number	31 30	29 28 27 26 25 24	23 22 21 20	19 18 17 16 15 14 13 12 11 10 – – – 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$_H$		
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4	3 2 1 0
Bit field name	Reserved	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$_H$	
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	L3DA0
R/W	R0	RW
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$_H$		
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4	3 2 1 0
Bit field name	Reserved	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.

L3OA1 (L3 layer Display Address 1)

Register address	DisplayBaseAddress + 68$_H$	
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	L3DA1
R/W	R0	RW
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$_H$	
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$_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				L3DY											
R/W	R0				RW											
Initial value	0				Don't care											

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

L3WX (L3 layer Window position X)

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

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

L3WY (L3 layer Window position Y)

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

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

L3WW (L3 layer Window Width)

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

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

L3WH (L3-layer Window Height)

Register address	DisplayBaseAddress + 14A$_H$															
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_H					
Bit number	31	30 29	28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	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$_H$					
Bit number	31 : 30	29 : 28 : 27 : 26 : 25 : 24	23 : 22 : 21 : 20	19 : 18 : 17 : 16 : 15 : 14 : 13 : 12 : 11 : 10 : − − − : 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_H		
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4	3 2 1 0
Bit field name	Reserved	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_H	
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	L4DA0
R/W	R0	RW
Initial value	0	Don't care

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

L4OA1 (L4 layer Origin Address 1)

Register address	DisplayBaseAddress + $7C_H$		
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4	3 2 1 0
Bit field name	Reserved	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.

L4OA1 (L4 layer Display Address 1)

Register address	DisplayBaseAddress + 80_H	
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	L4DA1
R/W	R0	RW
Initial value	0	Don't care

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

L4DX (L4 layer Display position X)

Register address	DisplayBaseAddress + 84_H	
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_H															
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_H															
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_H															
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_H															
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 + $15A_H$															
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$_H$					
Bit number	31	30 29	28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12	11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	L5C	L5FLP	Reserved	L5W	Reserved	L5H
R/W	RW	RW	R0	RW	R0	RW
Initial value			0	Don't care	0	Don't care

Bit 11 to 0 L5H (L5 layer Height)

Specifies the height of the logic frame of the L5 layer in pixel units. Setting value + 1 is the height

Bit 23 to 16 L5W (L5 layer memory Width)

Sets the memory width (stride) logic frame of the L5 layer in 64-byte units

Bit 30 and 29 L5FLP (L5 layer Flip mode)

Sets flipping mode for L5 layer

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

Bit 31 L5C (L5 layer Color mode)

Sets the color mode for L5 layer

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

L5EM (L5 layer Extended Mode)

Register address	DisplayBaseAddress + 160_H					
Bit number	31 30	29 28 27 26 25 24	23 22 21 20	19 18 17 16 15 14 13 12 11 10 – – – 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 + $8C_H$		
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4	3 2 1 0
Bit field name	Reserved	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_H	
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	L5DA0
R/W	R0	RW
Initial value	0	Don't care

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

L5OA1 (L5 layer Origin Address 1)

Register address	DisplayBaseAddress + 94_H		
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4	3 2 1 0
Bit field name	Reserved	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.

L5OA1 (L5 layer Display Address 1)

Register address	DisplayBaseAddress + 98_H	
Bit number	31 30 29 28 27 26	25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	L5DA1
R/W	R0	RW
Initial value	0	Don't care

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

L5DX (L5 layer Display position X)

Register address	DisplayBaseAddress + $9C_H$	
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 + $9E_H$															
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_H															
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_H															
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_H															
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 + $16A_H$															
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$_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved							CUZT	CUTC							
R/W	R0							RW	RW							
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 non-transparency color

1 Code 0 as transparency color

CPM (Cursor Priority Mode)

Register address	DisplayBaseAddress + A2$_H$							
Bit number	7	6	5	4	3	2	1	0
Bit field name	Reserved		CEN1	CEN0	Reserved		CUO1	CUO0
R/W	R0		RW	RW	R0		RW	RW
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$_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						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$_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUX0											
R/W	R0				RW											
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$_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUY0											
R/W	R0				RW											
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_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved						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 + $B0_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUX1											
R/W	R0				RW											
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 + $B2_H$															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUY1											
R/W	R0				RW											
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.

MDC (Multi Display Control)

Register address	DisplayBaseAddress + 170$_H$			
Bit number	31	30 29 28 24 23 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
Bit field name	MDen	reserve	SC1en	SC0en
R/W	RW	R0	RW	RW
Initial value	0	0	X	X

This register controls dual display mode.

Bit 0 SC0en0 (screen 0 enable 0)
- 0: L0 is not included into screen 0
- 1: L0 is included into screen 0

Bit 1 SC0en1 (screen 0 enable 1)
- 0: L1 is not included into screen 0
- 1: L1 is included into screen 0

≀

Bit 5 SC0en5 (screen 0 enable 5)
- 0: L5 is not included into screen 0
- 1: L5 is included into screen 0

Bit 6 SC0en6 (screen 0 enable 6)
- 0: Cursor0 is not included into screen 0
- 1: Cursor0 is included into screen 0

Bit 7 SC0en7 (screen 0 enable 7)
- 0: Cursor1 is not included into screen 0
- 1: Cursor1 is included into screen 0

Bit 8 SC1en0 (screen 1 enable 0)
- 0: L0 is not included into screen 1
- 1: L0 is included into screen 1

Bit 9 SC1en1 (screen 1 enable 1)
- 0: L1 is not included into screen 1
- 1: L1 is included into screen 1

≀

Bit 13 SC1en5 (screen 1 enable 5)
- 0: L5 is not included into screen 1
- 1: L5 is included into screen 1

Bit 14	SC1en6 (screen 1 enable 6) 0: Cursor 0 is not included into screen 1 1: Cursor 0 is included into screen 1
Bit 15	SC1en7 (screen 1 enable 7) 0: Cursor 1 is not included into screen 1 1: Cursor 1 is included into screen 1
Bit 31	MDen (multi display enable) This enables multi or dual display mode 0: Single display mode 1: Dual display mode

DLS (Display Layer Select)

Register address	DisplayBaseAddress + 180$_H$

Bit number	31 30 29 ----- 25 24	23	22 21 20	19	18 17 16	15	14 13 12	11	10 9 8	7	6 5 4	3	2 1 0
Bit field name	Reserved	DLS5		DLS4		DLS3		DLS2		DLS1		DSL0	
R/W	R0	R0	RW	R0	RW	R0	RW	R0	RW	R0	RW	R0	RW
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$_H$			
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$_H$						
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	L0BE	L0BS	L0BI	L0BP	Reserved	L0BR
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 L0BR (L0 layer Blend Ratio)

Sets the blend ratio. Basically, the blend ratio is setting value/256.

Bit 13 L0BP (L0 layer Blend Plane)

Specifies that the L5 layer is the blend plane.

0 Value of L0BR used as blend ratio

1 Pixel of L5 layer used as blend ratio

Bit 14 L0BI (L0 layer Blend Increment)

Selects whether or not 1/256 is added when the blend ratio is not “0”.

0 Blend ratio calculated as is

1 1/256 added when blend ratio ≠ 0

Bit 15 L0BS (L0 layer Blend Select)

Selects the blend calculation expression.

0 Upper image × Blend ratio + Lower image × (1 – Blend ratio)

1 Upper image × (1 – Blend ratio) + Lower image × Blend ratio

Bit 16 L0BE (L0 layer Blend Enable)

This bit enables blending.

0 Overlay via transparent color

1 Overlay via blending

Before blending, the blend mode must be specified using L0BE, and alpha must also be enabled for L0 layer display data. For direct color, alpha is specified using the MSB of data; for indirect color, alpha is specified using the MSB of palette data.

L1BLD (L1 Blend)

Register address	DisplayBaseAddress + 188_H					
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 $\neq 0$

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$_H$						
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 $\neq 0$

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_H						
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$_H$						
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$_H$															
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$_H$															
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$_H$	
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$_H$		
Bit number	31	30 29 28 --- 24	23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	L0ETZ	Reserved	L0TEC
R/W	RW	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 + $1A4_H$		
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 + $1A8_H$		
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$_H$		
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$_H$		
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 + $1B4_H$		
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

L1YCR0 (L1 layer YC to Red coefficient 0)

Register address	DisplayBaseAddress + $1E0_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved					a12											Reserved					a11										
R/W	R0					RW											R0					RW										
Initial value	0					000 0000 0000											0					001 0010 1011										

This register defines YCbCr/RGB converstion parameters for red component.

Bit 10 to 0 a11

11bit signed real. lower 8bit is fraction. two's complement.

Bit 26 to 16 a12

11bit signed real. lower 8bit is fraction. two's complement.

Refer 7.7 for detail.

L1YCR1 (L1 layer YC to Red coefficient 1)

Register address	DisplayBaseAddress + $1E4_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved							b1									Reserved					a13										
R/W	R0							RW									R0					RW										
Initial value	0							1 1111 0000									0					001 1001 1000										

This register defines YCbCr/RGB converstion parameters for red component.

Bit 10 to 0 a13

11bit signed real. lower 8bit is fraction. two's complement.

Bit 24 to 16 b1

9bit signed integer. two's complement.

Refer 7.7 for detail.

L1YCG0 (L1 layer YC to Green coefficient 0)

Register address	DisplayBaseAddress + $1E8_H$			
Bit number	31 30 29 28 27	26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11	10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	a22	Reserved	a21
R/W	R0	RW	R0	RW
Initial value	0	111 1001 1100	0	001 0010 1011

This register defines YCbCr/RGB converstion parameters for green component.

Bit 10 to 0 a21

11bit signed real. lower8bit is fraction. two's complement.

Bit 26 to 16 a22

11bit signed real. lower 8bit is fraction. two's complement.

Refer 7.7 for detail.

L1YCG1 (L1 layer YC to Green coefficient 1)

Register address	DisplayBaseAddress + $1EC_H$			
Bit number	31 30 29 28 27 26 25	24 23 22 21 20 19 18 17 16	15 14 13 12 11	10 9 8 7 6 5 4 3 2 1 0
Bit field name	Reserved	b2	Reserved	a23
R/W	R0	RW	R0	RW
Initial value	0	1 1111 0000	0	111 0010 1111

This register defines YCbCr/RGB converstion parameters for green component.

Bit 10 to 0 a23

11bit signed real. lower 8bit is fraction. two's complement.

Bit 24 to 16 b2

9bit signed integer. two's complement.

Refer 7.7 for detail.

L1YCB0 (L1 layer YC to Blue coefficient 0)

Register address	DisplayBaseAddress + 1F0$_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved					a32											Reserved					a31										
R/W	R0					RW											R0					RW										
Initial value	0					010 0000 0100											0					001 0010 1011										

This register defines YCbCr/RGB converstion parameters for blue component.

Bit 10 to 0 a31

11bit signed real. lower 8bit is fraction. two's complement.

Bit 26 to 16 a32

11bit signed real. lower 8bit is fraction. two's complement.

Refer 7.7 for detail.

L1YCB1 (L1 layer YC to Blue coefficient 1)

Register address	DisplayBaseAddress + 1F4$_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved							b3									Reserved					a33										
R/W	R0							RW									R0					RW										
Initial value	0							1 1111 0000									0					000 0000 0000										

This register defines YCbCr/RGB converstion parameters for blue component.

Bit 10 to 0 a33

11bit signed real. lower 8bit is fraction. two's complement.

Bit 24 to 16 b3

9bit signed integer. two's complement.

Refer 7.7 for detail.

L0PAL0-255 (L0 layer Palette 0-255)

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

These are color palette registers for L0 layer and cursors. In the indirect color mode, a color code in the display frame indicates the palette register number, and the color information set in that register is applied as the display color of that pixel. This register corresponds to the 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$_H$ -- DisplayBaseAddress + BFF$_H$							
Bit number	31	30 29 28 27 26 25 24	23 22 21 20 19 18	17 16	15 14 13 12 11 10	9 8	7 6 5 4 3 2	1 0
Bit field name	A		R		G		B	
R/W	RW	R0	RW	R0	RW	R0	RW	R0
Initial value	Don't care	0000000	Don't care	00	Don't care	00	Don't care	00

These are color palette registers for L1 layer and cursors. In the indirect color mode, a color code in the display frame indicates the palette register number, and the color information set in that register is applied as the display color of that pixel. This register corresponds to the MBPALn register for previous products.

Bit 7 to 2 B (Blue)

Sets blue color component

Bit 15 to 10 G (Green)

Sets green color component

Bit 23 to 18 R (Red)

Sets red color component

Bit 31 A (Alpha)

Specifies whether or not to perform blending with lower layers when the blending mode is enabled.

0 Blending not performed even when blending mode enabled
Overlay is performed via transparent color.

1 Blending performed

L2PAL0-255 (L2 layer Palette 0-255)

<table>
<tr><td>Register address</td><td colspan="32">DisplayBaseAddress + 1000_H -- DisplayBaseAddress + 13FF_H</td></tr>
<tr><td>Bit number</td><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td>Bit field name</td><td colspan="8">A</td><td colspan="8">R</td><td colspan="8">G</td><td colspan="8">B</td></tr>
<tr><td>R/W</td><td>RW</td><td colspan="7">R0</td><td colspan="6">RW</td><td colspan="2">R0</td><td colspan="6">RW</td><td colspan="2">R0</td><td colspan="6">RW</td><td colspan="2">R0</td></tr>
<tr><td>Initial value</td><td>Don't care</td><td colspan="7">0000000</td><td colspan="6">Don't care</td><td colspan="2">00</td><td colspan="6">Don't care</td><td colspan="2">00</td><td colspan="6">Don't care</td><td colspan="2">00</td></tr>
</table>

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$_H$ -- DisplayBaseAddress + 17FF$_H$							
Bit number	31	30 29 28 27 26 25 24	23 22 21 20 19 18	17 16	15 14 13 12 11 10	9 8	7 6 5 4 3 2	1 0
Bit field name	A		R		G		B	
R/W	RW	R0	RW	R0	RW	R0	RW	R0
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

13.2.5 Video capture registers

VCM (Video Capture Mode)

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

This register sets the video capture mode. This register is not initialized by software reset.

Bit1 VS (Video Select)
NTSC or PAL is selected for the code error detection. (only the RTB656 is input.)
0 NTSC
1 PAL

Bit2 NRGB(Native RGB input on)
Native RGB mode is set up.
0 RGB video data is accepted via an internal RGB preprocessor which converts RGB to YUV422
1 Native RGB

Bit20 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

Bit25-24 CM (Capture Mode)
Sets video capture mode. To capture vides, set these bits to “11”.
00 Initial value
01 Reserved
10 Reserved
11 Capture

Bit28 VICE (Video Input Clock Enable)
Capture clock enable
0 Enable
1 Disable

Bit30 VIS(Video Input Select)
0 RBT656/601
1 RGB

Bit31 VIE (Video Input Enable)
Enables video capture function
0 Does not capture video
1 Captures video

-Procedure of video capture clock Stop-

1) 0 is written in bit31 (VIE) of the VCM register, and the video capture function is invalidated.

2) 1 is written in bit28 (VICE) of the VCM register, and Stop does video capture clock.

-Procedure of video capture clock beginning-

1) 0 is written in bit28 (VICE) of the VCM register, and video capture clock is made effective.

2) 1 is written in bit31 (VIE) of the VCM register, and the video capture function is made effective.

CSC (Capture SCale)

Register address	CaptureBaseAddress + 04h			
Bit number	31 30 29 28 27	26 25 24 23 22 21 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	R/W	R/W	R/W	R/W
Initial value	00001	00000000000	00001	00000000000

This register sets the video capture upscaling/downscaling ratio.

Bit10-0	HSCF (Vertical SCale Fraction) The decimal part of a horizontal upscaling/downscaling ratio is set.
Bit15-11	HSCI (Horizontal Scale Integer) The integer part of a horizontal upscaling/downscaling ratio is set.
Bit26-16	VSCF (Vertical SCale Fraction) The decimal part of a vertical upscaling/downscaling ratio is set.
Bit31-27	VSCI (Vertical SCale Integer) The integer part of a vertical upscaling/downscaling ratio is set.

Note:

- Smooth continuation operation to Down Scaling mode and Up Scaling mode cannot be performed. The picture disorder of some arises at the time of a change. This is the restrictions for Up Scaling mode and Down Scaling mode using the same interpolate circuit.

CBM (video Capture Buffer Mode)

Register address	CaptureBaseAddress + 10h															
Bit number	31	30	29	28	27 … 24	23 22 … 17 16	15	14	13	12	11	10 9 8	7 6 5	4	3 2 1	0
Bit field name	OO	S-BUF	C-RGB	PAU	Reserve	CBW	resv	C24	BED	CSW	resv	SSS	SSM	HRV	reserve	C-BST
R/W	R/W	R/W	R/W	R/W	RX	R/W	RX	R/W	R/W	R/W	RX	R/W	R/W	R/W	RX	R/W
Initial value	0	X	X	0	X	X	X	0	0	0	X	000	000	0	X	0

Bit0 CBST (Capture Burst)
The burst-length at the capture Write is specified. Because long burst-length is good the access efficiency, 1 is recommended to be set.
0 Normal burst write (4word)
1 Long burst write (8word)

Bit4 HRV (H-reverse)
The horizontal reversing mode specification
0 Normal operation mode
1 Horizontal reversing mode

Bit7-5 SSM (Single Shot Mode)
Single shot mode
000 Normal operation mode
001 Single shot/odd field mode
010 Single shot/even field mode
011 Single shot/both field mode (with field distinction)
111 Single shot/both field mode2 (without field distinction)

Bit10-8 SSS (Single Shot Status)
The state of single shot operation is shown.
000 Initial state
001 Odd field mode / under capture
010 Even field mode / under capture
100 Both field mode / under first field capture
101 Both field mode / under second field capture

Bit12 CSW (Color Swap)
The byte position of a color ingredient is replaced.
0 Without exchange
1 With exchange

Bit13 BED (Big EnDian)
Endian is reversed
0 Little endian (enable display)
1 Big endian (disable display)

Bit14 C24 (Color 24bit/pixel)
It specifies wherther 24bit/pixel or 16bit/pixel is used in RGB capture.
It is effective in native RGB capture (NRGB=1) or converted RGB capture(CRGB=1).
0 16bit/pixel
1 24bit/pixel

Bit23-16 CBW (Capture Buffer memory Width)
Sets memory width (stride) of capture buffer in 64 bytes

Bit28 PAU (PAUse)
It is shown that capture operation is Stop temporarily. 0 can be written and it can cancel.
0 Under operation
1 Stop temporarily

Bit29 CRGB (Capture RGB write)
It specifies whether YCbCr to RGB conversion is applied or not before writing into the capture buffer.
There are two formats of RGB or RGB=5:5:5 (16 bits/pixel) and RGB = 8:8:8 (24 bit/pixel) format, depending to C24-bit value described above.
0 YCbCr (without conversion)
1 RGB

Bit30 SBUF (Single Buffer)
It specifies managing a capture buffer by the single buffer system.
0 Normal mode (ring buffer)
1 Single buffer mode

Bit31	OO (Odd Only mode)		
	Specifies whether to capture odd fields only		
	0	Normal mode	
	1	Odd only mode	

Note: This register is not initialized by soft reset.

CBOA (video Capture Buffer Origin Address)

Register address	CaptureBaseAddress + 14h		
Bit number	31 - 30 - 29 - 28 - 27 - 26 - 25 - 24	23 - 22 - 21 - 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	R/W	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	CaptureBaseAddress + 18h		
Bit number	31 - 30 - 29 - 28 - 27 - 26 - 25 - 24	23 - 22 - 21 - 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	R/W	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	CaptureBaseAddress + 1Ch															
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				R/W											
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 downscaling, apply this setting to the post-reduction image coordinates.

-Note: The even number is set at the YUV mode.

CIVSTR (Capture Image Vertical STaRt)

Register address	CaptureBaseAddress + 1Eh															
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				R/W											
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 downscaling, apply this setting to the post-reduction image coordinates.

CIHEND (Capture Image Horizontal END)

Register address	CaptureBaseAddress + 20h															
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				R/W											
Initial value	X				X											

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 downscaling, 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.

-Note: In the YUV mode, horizontal pixel size (CIHEND-CIHSTR) sets the even number.

CIVEND (Capture Image Vertical END)

Register address	CaptureBaseAddress + 22h															
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				R/W											
Initial value	X				X											

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 downscaling, 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.

CVCNT (Capture Vertical Count)

Register address	CaptureBaseAddress + 300h															
Bit number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CVCNT											
R/W	R0				R											
Initial value	0				Don't care											

Y coordinates of the raster which is carrying out the capture are shown. Only read-out is possible.

CHP (Capture Horizontal Pixel)

Register address	CaptureBaseAddress + 28h	
Bit number	31 - 30 - 29 - 28 - 27 - 26 - 25 - 24 - 23 - 22 - 21 - 20 - 19 - 18 - 17 - 16 - 15 - 14 - 13 - 12 - 11 - 10	9 - 8 - 7 - 6 - 5 - 4 - 3 - 2 - 1 - 0
Bit field name	Reserved	CHP
R/W	RX	R/W
Initial value	X	0x168 (360)

This register sets the count of horizontal pixels of the image output after scaling. Specify the count of horizontal pixels in 2 pixels. Maximum is 840 pixels (setting value is 0x1A4)

CVP (Capture Vertical Pixel)

Register address	CaptureBaseAddress + 2c$_{H}$			
Bit number	31 : 30 : 29 : 28 : 27 : 26	25 : 24 : 23 : 22 : 21 : 20 : 19 : 18 : 17 : 16	15 : 14 : 13 : 12 : 11 : 10	9 : 8 : 7 : 6 : 5 : 4 : 3 : 2 : 1 : 0
Bit field name	Reserved	CVPP	Reserved	CVPN
R/W	RX	RW	RX	RW
Initial value	X	271$_{H}$ (625$_{D}$)	X	20D$_{H}$ (525$_{D}$)

This register sets the count of vertical pixels of the image output after scaling. The fields 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	CaptureBaseAddress + 40h				
Bit number	31 - 30 - 29 - 28	27 - 26 - 25 - 24	23 - 22 - 21 - 20	19 - 18 - 17 - 16	15 - 14 - 13 - 12 - 11 - 10 - 9 - 8 - 7 - 6 - 5 - 4 - 3 - 2 - 1 - 0
Bit field name	Reserved	CVLPF	Reserved	CHLPF	Reserved
R/W	RX	R/W	RX	R/W	RX
Initial value	0	0	0	0	X

This register sets the Low Pass Filter Coefficient. The vertical low pass filter consists of FIR filters of three taps. The horizontal low pass filter consists of FIR filters of five taps. It specifies independently in 2-bit coefficient code with a luminance signal (Y) and a chrominance signal (Cb and Cr) . A low pass filter is OFF (through) in a setup of each coefficient code “00”.

Bit 17 to 16 CHLPF_C (Capture Horizontal LPF coefficient C)

CHLPF_C	K0	K1	K2	K3	K4
00	0	0	1	0	0
01	0	1/4	2/4	1/4	0
10	0	3/16	10/16	3/16	0
11	3/32	8/32	10/32	10/32	3/32

Bit 19 to 18 CHLPF_Y (Capture Horizontal LPF coefficient Y)

CHLPF_Y	K0	K1	K2	K3	K4
00	0	0	1	0	0
01	0	1/4	2/4	1/4	0
10	0	3/16	10/16	3/16	0
11	3/32	8/32	10/32	10/32	3/32

Bit 25 to 24 CVLPF_C (Capture Vertical LPF coefficient C)

CVLPF_C	K0	K1	K2
00	0	1	0
01	1/4	2/4	1/4
10	3/16	10/16	3/16
11	Reserved		

Bit 27 to 26 CVLPF_Y (Capture Vertical LPF coefficient Y)

CVLPF_Y	K0	K1	K2
00	0	1	0
01	1/4	2/4	1/4
10	3/16	10/16	3/16
11	Reserved		

Note:

- In the case of Native RGB mode (NRGB=1), only a setup of CVLPF_Y code becomes effective.

CMSS (Capture Magnify Source Size)

Register address	CaptureBaseAddress + 48h			
Bit number	31 - 30 - 29 - 28	27 - 26 - 25 - 24 - 23 - 22 - 21 - 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	RX	R/W	RX	R/W
Initial value	X	X	X	X

Bit11-0 CMSVL (Capture Magnify Source Vertical Line)
This register sets the number of vertical lines of the image input before Magnify scaling.

Bit27-16 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.

CMDS (Capture Magnify Display Size)

Register address	CaptureBaseAddress + 4Ch			
Bit number	31 - 30 - 29 - 28	27 - 26 - 25 - 24 - 23 - 22 - 21 - 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	RX	R/W	RX	R/W
Initial value	X	X	X	X

Bit11-0 CMDVL (Capture Magnify Display Vertical Line)
This register sets the number of vertical lines of the image output after Magnify scaling.

Bit27-16 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.

In general, this display size has to be same as L1 display size.

CMDVL = L1WH+1, CMDHP*2=L1WW

If a part of L1 layer is clipped by screen, actual display size has to be set into CMDS register.

For example, if L1WX=320, L1WH=640, HDP=639, then right half is clipped and actual display width is 320.

RGBHC(RGB input Hsync Cycle)

Register address	CaptureBaseAddress + 80h	
Bit number	31 - 30 - 29 - 28 - 27 - 26 - 25 - 24 : 23 - 22 - 21 - 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	RX	R/W
Initial value	X	X

Bit13-0 RGBHC
This register sets number of HSYNC cycles of the RGB input. . It is used when it is made a setup which samples VSYNC. The setting value +1 is a level cycle.

RGBHEN(RGB input Horizontal Enable area)

Register address	CaptureBaseAddress + 84h			
Bit number	31 - 30 - 29 - 28 - 27 - 26 - 25 - 24 : 23 - 22 - 21 - 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	RX	R/W	RX	R/W
Initial value	X	X	X	X

It is a parameter for determining effective pixel data.

Bit12-0 RGBHEN(RGB input Horizontal Enable area Size)
Effective pixel data size is set up per pixel. Specify the number of horizontal pixels in 2-pixel units

Bit27-16 RGBHST(RGB input Horizontal Enable area Start position)
The start position of effective pixel data is set up. The setting value -4 is a start position.

Note:

- The maximum horizontal enable area size(RGBHEN) which can be captured is 840 pixels. This is the restriction by line buffer size in a video capture module.

RGBVEN(RGB input Vertical Enable area)

Register address	CaptureBaseAddress + 88h					
Bit number	31 - 30 - 29 - 28 - 27 - 26 - 25 - 24 : 23 - 22 - 21 - 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		Reserved	RGBVST	Reserved	RGBVEN
R/W	RX		R/W	R/W	RX	R/W
Initial value	X		X	X	X	X

It is a parameter for determining effective pixel data.

Bit12-0 RGBVEN(RGB input Vertical Enable area Size)
Set effective line size

Bit24-16 RGBVST(RGB input Vertical Enable area Start position)
The start position of effective line is set up. The setting value -1 is a start position.

RGBS (RGB input Sync)

Register address	CaptureBaseAddress + 90h				
Bit number	31 - 30 - 29 - 28 - 27 - 26 - 25 - 24 - 23 - 22 - 21 - 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	RX	R/W	RX	R/W	R/W
Initial value	X	1	X	0	0

Edge detection of a synchronized signal is set up. It is used at the time of RGB input format.

Bit0 VP (VSYNCI Polarity)
0 Negative edge of VINVSYNC is set to VSYNC.
1 Positive edge of VINVSYNC is set to VSYNC.

Bit1 HP (HSYNCI Polarity)
0 Negative edge of VINHSYNC is set to HSYNC.
1 Positive edge of VINHSYNC is set to HSYNC.

Bit16 RM(RGB Input Mode select)
Sets Direct RGB input mode
0 reserved
1 RGB666 Direct input Mode

Conversion Operation

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

Y = a11*R + a12*G + a13*B + b1
Cb= a21*R + a22*G + a23*B + b2 a_{ij} 10bit signed real (lower 8bit is fraction)
Cr= a31*R + a32*G + a33*B + b3 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	CaptureBaseAddress + $C0_H$				
Bit number	31 30 29 28 27 26 25 24 23 22	21	20 19 18 17 16 15 14 13 12 11	10	9 8 7 6 5 4 3 2 1 0
Bit field name	a11	Re	a12	Re	a13
R/W	RW	R	RW	R	RW
Initial value	0001000010 $_b$	0	0010000000 $_b$	0	0000011001 $_b$

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)

RGBCMCb (RGB Color convert Matrix Cb coefficient)

Register address	CaptureBaseAddress + $C4_H$				
Bit number	31 30 29 28 27 26 25 24 23 22	21	20 19 18 17 16 15 14 13 12 11	10	9 8 7 6 5 4 3 2 1 0
Bit field name	a21	Re	a22	Re	a23
R/W	RW	R	RW	R	RW
Initial value	1111011010 $_b$	0	1110110110 $_b$	0	0001110000 $_b$

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	CaptureBaseAddress + C8$_H$				
Bit number	31 30 29 28 27 26 25 24 23 22	21	20 19 18 17 16 15 14 13 12 11	10	9 8 7 6 5 4 3 2 1 0
Bit field name	A31	Re	A32	Re	A33
R/W	RW	R	RW	R	RW
Initial value	0001110000 $_b$	0	1110100010 $_b$	0	1111101110 $_b$

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	CaptureBaseAddress + CC$_H$					
Bit number	31	30 29 28 27 26 25 24 23 22	21 20	19 18 17 16 15 14 13 12 11	10 9	8 7 6 5 4 3 2 1 0
Bit field name	R	B1	Res	b2	Res	b3
R/W	R	RW	R	RW	R	RW
Initial value	0	000010000 $_b$	0	010000000 $_b$	0	010000000 $_b$

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

【656 Code error detect】

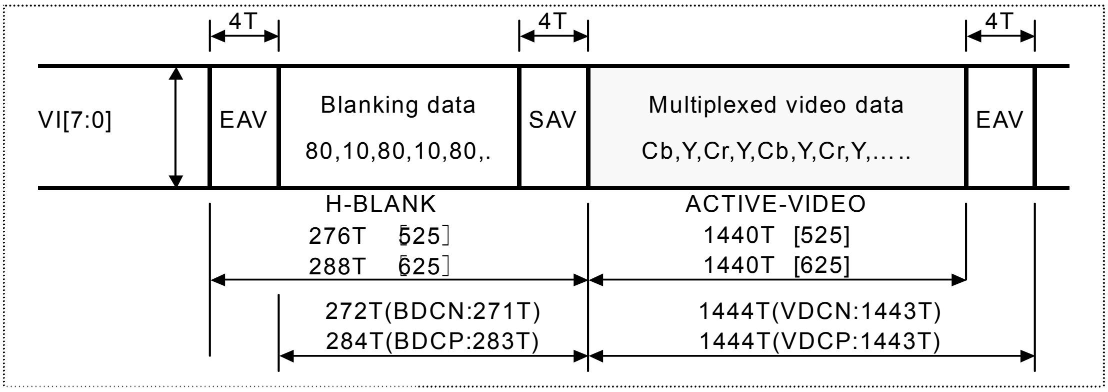
< RBT656 format input only>

CDCN (Capture Data Count for NTSC)

Register address	CaptureBaseAddress + 4000h			
Bit number	31 - 30 - 29	28 - 27 - 26 - 25 - 24 : 23 - 22 - 21 - 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	0x10f(271)	X	0x5A3(1443)

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

Bit12-0	VDCN (Valid Data Count for NTSC) Sets count of data processed during valid period in NTSC format. The setting value +1 is a data number
Bit28-16	BDCN (Blanking Data Count for NTSC) Sets count of data processed during blanking period in NTSC format. The setting value +1 is a data number



The range of VDCN and BDCN is shown in the following figure.

SAV: start of active video timing reference code

EAV: end of active video timing reference code

T: clock period 37 ns nom.

CDCP (Capture Data Count for PAL)

Register address	CaptureBaseAddress + 4004h			
Bit number	31 - 30 - 29	28 - 27 - 26 - 25 - 24 - 23 - 22 - 21 - 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	0x11B(283)	X	0x5A3(1443)

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

Bit12-0	VDCP (Valid Data Count for PAL) Sets count of data processed during valid period in PAL format. The setting value +1 is a data number
Bit28-16	BDCP (Blanking Data Count for PAL) Sets count of data processed during blanking period in PAL format. The setting value +1 is a data number

VCS (Video Capture Status)

Register address	CaptureBaseAddress + 08h	
Bit number	31 - 30 - 29 - 28 - 27 - 26 - 25 - 24 : 23 - 22 - 21 - 20 - 19 - 18 - 17 - 16 - 15 - 14 - 13 - 12 - 11 - 10 - 9 - 8 - 7 - 6 - 5	4 - 3 - 2 - 1 - 0
Bit field name	Reserve	CE
R/W	RX	RW0
Initial value	X	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 6-0 CE0 (Capture Error 0)

Bit0	1 : RBT.656 undefined error (Code Bit7)	0 : true
Bit1	1 : RBT.656 undefined error (Code Bit7-4)	0 : true
Bit2	1 : RBT.656 undefined error (Code Bit7-0)	0 : true
Bit3	1 : RBT.656 long term H code error (SAV)	0 : true
Bit4	1 : RBT.656 long term H code error (EAV)	0 : true

13.2.6 Drawing control registers

CTR (Control Register)

Register address	DrawBaseAddress + 400_H

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

Bit 22 CE (Display List Command Error)

Indicates command error occurrence (Not all error can detect. Need software reset or hardware reset for recovery)

0	Normal
1	Command error detected

Bit 24 FO (FIFO Overflow)

Indicates FIFO overflow occurrence

0	Normal
1	FIFO overflow detected

IFSR (Input FIFO Status Register)

Register address	DrawBaseAddress + 404_H			
Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3	2	1	0
Bit field name		NF	FF	FE
R/W		R	R	R
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_H	
Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6	5 4 3 2 1 0
Bit field name		FCNT
R/W		R
Initial value		011101

This is a mirror register for bits 19 to 15 of the CTR register.

SST (Setup engine Status)

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

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

DST (DDA Status)

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

This is a mirror register for bits 1 to 0 of the CTR register.

EST (Error Status)

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

This is a mirror register for bits 24 to 22 of the CTR register.

13.2.7 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$_H$									
Bit number	31 30 29 28 27 26 25 24 23 22 21	20	19 18 17	16 15	14 13 12 11 10	9	8	7 6 5 4	3 2	1 0
Bit field name		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)

Bit 20 ZP (Z Precision)

Sets the precision of the Z value used for erasing hidden planes.

16 bits/pixel

8 bits/pixel

MDR1/MDR1S/MDR1B (Mode Register for LINE/for Shadow/for Border)

Register address	DrawBaseAddress + 424$_H$											
Bit number	31 30 29	28 27 26 25 24	23 22 21	20	19	18 17 16 15 14 13	12 11 10 9	8 7	6	5 4 3	2	1	0
Bit field name		LW		BP	BL		LOG	BM	ZW	ZCL	ZC	AS	
R/W		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.

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.

Please set ZC bit (bit 2) to 0 when draw BltCopyAltAlphaBlendP command.

Bit 1 AS (Alpha Shading mode)

Sets the shading mode for alpha.

0	Alpha flat shading
1	Alpha Gouraud shading

Bit 2 ZC (Z Compare mode)

Sets Z comparison mode

0	Disabled
1	Enabled

Bit 5 to 3 ZCL (Z Compare Logic)

Selects type of Z comparison

000	NEVER
001	ALWAYS
010	LESS
011	LEQUAL
100	EQUAL
101	GEQUAL
110	GREATER
111	NOTEQUAL

Bit 6 ZW (Z Write mode)

Sets Z write mode

0	Writes Z values.
1	Not write Z values.

Bit 8 to 7 BM (Blend Mode)

Sets blend mode

00	Normal (source copy)
01	Alpha blending
10	Drawing with logic operation

11	Reserved

Bit 12 to 9 LOG (Logical operation)

Sets type of logic operation

0000	CLEAR
0001	AND
0010	AND REVERSE
0011	COPY
0100	AND INVERTED
0101	NOP
0110	XOR
0111	OR
1000	NOR
1001	EQUIV
1010	INVERT
1011	OR REVERSE
1100	COPY INVERTED
1101	OR INVERTED
1110	NAND
1111	SET

Bit 19 BL (Broken Line)

Selects line type

0	Solid line
1	Broken line

Bit 20 BP (Broken line Period)

Selects broken line cycle

0:	32 bits
1:	24 bits

Bit 28 to 24 LW (Line Width)

Sets line width for drawing line

00000	1 pixel
00001	2 pixels
:	:
11111	32 pixels

MDR2/MDR2S/MDR2TL (Mode Register for Polygon/for Shadow/for TopLeft)

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

This register sets the polygon drawing mode.

This register is used for the body primitive, for the shade primitive, and for the top-left non-applicable primitive.

The value after a drawing that involves the shade primitive or the top-left non-applicable primitive is the value set for MDR2.

(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$_H$														
Bit number	31 30 29 28 27 26 25	24	23 22	21 20	19 18	17 16	15 14 13 12	11 10	9 8	7 6	5	4	3	2 1	0
Bit field name		BA		TAB		TBL		TWS	TWT		TF		TC		
R/W		RW		RW		RW		RW	RW		RW		RW		
Initial value		0		00		00		00	00		0		0		

This register sets the texture mapping mode.

Bit 3 TC (Texture coordinates Correct)

Sets texture coordinates correction mode

0	Disabled
1	Enabled

Bit 5 TF (Texture Filtering)

Sets type of texture interpolation (filtering)

0	Point sampling
1	Bi-linear filtering

Bit 9 and 8 TWT (Texture Wrap T)

Sets type of texture coordinates T direction wrapping

00	Cramp
01	Repeat
10	Border
11	Reserved

Bit 11 and 10 TWS (Texture Wrap S)

Sets type of texture coordinates S direction wrapping

00	Cramp
01	Repeat
10	Border
11	Reserved

Bit 17 and 16 TBL (Texture Blend mode)

Sets texture blending mode

00	Decal
01	Modulate
10	Stencil
11	Reserved

Bit 21 and 20 TAB (Texture Alpha Blend mode)

Sets texture blending mode

The stencil mode and the stencil alpha mode are enabled only when the MDR2 register blend mode (BM) is set to the alpha blending mode. If it is not set to the alpha blending mode, the stencil mode and stencil alpha mode perform the same function as the normal mode.

00	Normal
01	Stencil
10	Stencil alpha
11	Reserved

Bit 24 BA (Bilinear Accelerate Mode)

Improves the performance of bi-linear filtering, although a texture area of four times the default texture area is used.

0	Default texture area used
1	Texture area four times default texture area used

MDR4 (Mode Register for BLT)

Register address	DrawBaseAddress + 430_H					
Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13	12 11 10 9	8 7	6 5 4 3 2	1	0
Bit field name		LOG	BM		TE	
R/W		RW	RW		RW	
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 (transparancy copy) Note: Set the blend mode (BM) to normal.

Bit 8 to 7	BM (Blend Mode)	
	Sets blend mode	
	00	Normal (source copy)
	01	Reserved
	10	Drawing with logic operation
	11	Reserved

Bit 12 to 9	LOG (Logical operation)	
	Sets logic operation	
	0000	CLEAR
	0001	AND
	0010	AND REVERSE
	0011	COPY
	0100	AND INVERTED
	0101	NOP
	0110	XOR
	0111	OR
	1000	NOR
	1001	EQUIV
	1010	INVERT
	1011	OR REVERSE
	1100	COPY INVERTED
	1101	OR INVERTED
	1110	NAND
	1111	SET

MDR7 (Mode Register for Extension)

Register address	DrawBaseAddress + $43C_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																												GG				
R/W																												W				
Initial value																												0				

This register used for “Gray Scale Gouraud Shading”. This register is able to use only in 8 bit / pixel mode.

Bit 4 GG (Gray scale Gouraud Shading)

Sets gray scale gouraud shading mode

0: Hard mask on (compatible Orchid)

1: Hard mask off (extension mode)

Note: This register is used for gray scale gouraud shading. This register is changed by internal processing. Please don’t set these bits except GG bit.

In case of gray scale gouraud shading drawing, please set this register to the follows.

1. Set this register to **0x00000050** before drawing.
2. Set this register to **0x00000040** after drawing.

FBR (Frame buffer Base)

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

This register stores the base address of the drawing frame.

XRES (X Resolution)

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

This register sets the drawing frame horizontal resolution.

ZBR (Z buffer Base)

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

This register sets the Z buffer base address.

TBR (Texture memory Base)

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

This register sets the texture memory base address.

PFBR (2D Polygon Flag-Buffer Base)

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

This register sets the polygon flag buffer base address.

CXMIN (Clip X minimum)

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

This register sets the clip frame minimum X position.

CXMAX (Clip X maximum)

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

This register sets the clip frame maximum X position.

CYMIN (Clip Y minimum)

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

This register sets the clip frame minimum Y position.

CYMAX (Clip Y maximum)

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

This register sets the clip frame maximum Y position.

TXS (Texture Size)

Register address	DrawBaseAddress + 464$_H$			
Bit number	31 30 29	28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13	12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name		TXSN		TXSM
R/W		RW		RW
Initial value		000010000000		000010000000

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

Bit 12 to 0 TXSM (Texture Size M)

Sets horizontal texture size. Any power of 2 between 4 and 4096 can be used. Values that are not a power of 2 cannot be used.

0_0000_0000_0100	M=4	0_0010_0000_0000	M=512
0_0000_0000_1000	M=8	0_0100_0000_0000	M=1024
0_0000_0001_0000	M=16	0_1000_0000_0000	M=2048
0_0000_0010_0000	M=32	1_0000_0000_0000	M=4096
0_0000_0100_0000	M=64		
0_0000_1000_0000	M=128		
0_0001_0000_0000	M=256	Other than the above	Setting disabled

Bit 28 to 16 TXSN (Texture Size N)

Sets vertical texture size. Any power of 2 between 4 and 4096 can be used. Values that are not a power of 2 cannot be used.

0_0000_0000_0100	N=4	0_0010_0000_0000	N=512
0_0000_0000_1000	N=8	0_0100_0000_0000	N=1024
0_0000_0001_0000	N=16	0_1000_0000_0000	N=2048
0_0000_0010_0000	N=32	1_0000_0000_0000	N=4096
0_0000_0100_0000	N=64		
0_0000_1000_0000	N=128		
0_0001_0000_0000	N=256	Other than the above	Setting disabled

TIS (Tile Size)

Register address	DrawBaseAddress + 468_H			
Bit number	31 30 29 28 27 26 25 24 23	22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7	6 5 4 3 2 1 0
Bit field name		TISN		TISM
R/W		RW		RW
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 + $46C_H$	
Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13	12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name		XBO
R/W		RW
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_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name								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_H																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name							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_H	
Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name		FGC
R/W		RW
Initial value		0

This register sets the drawing foreground color. This color is for the object color for flat shading and foreground color for bitmap drawing and broken line drawing. All bits set to "1" are drawn in the color set at this register.

8 bit color mode:

Bit 7 to 0	FGC8 (Foreground 8 bit Color) Sets the indirect color for the foreground (color index code).
Bit 31 to 8	These bits are not used.

16 bit color mode:

Bit 15 to 0	FGC16 (Foreground 16 bit Color) This field sets the 16-bit direct color for the foreground. Note that the handling of bit 15 is different from that in ORCHID. Up to ORCHID, bit 15 is "0" for other than bit map and rectangular drawing, but starting with CORAL, the setting value is reflected in memory as is. This bit is also reflected in bit 15 of the 16-bit color at Gouraud shading.
Bit 31 to 16	These bits are not used.

BC (Background Color)

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

ALF (Alpha Factor)

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

This register sets the alpha blending coefficient.

BLP (Broken Line Pattern)

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

TBC (Texture Border Color)

Register address	DrawBaseAddress + 494$_H$		
Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15	14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Bit field name	BC8/16		
R/W	RW		
Initial value	0		

This register sets the border color for texture mapping.

8 bit color mode:

Bit 7 to 0	BC8 (Border Color) Sets the 8-bit direct color for the texture border color

16 bit color mode:

Bit 15 to 0	BC16 (Border Color) Sets the 16-bit direct color for the texture border color Bit15 is used for controlling a stencil and stencil alpha

BLPO (Broken Line Pattern Offset)

Register address	DrawBaseAddress + 3E0$_H$	
Bit number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5	4 3 2 1 0
Bit field name		BCR
R/W		RW
Initial value		11111

This register stores the bit number of the broken-line pattern set to BLP registers, for broken line drawing. This value is decremented at each pixel drawing. Broken line can be drawn starting from any starting position of the specified broken-line pattern by setting any value at this register.

When no write is performed, the position of broken-line pattern is sustained.

13.2.8 Triangle drawing registers

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

(XY coordinates register)

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

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

Sets (X, Y) coordinates for triangle drawing

Ys	Y coordinates start position of long edge
Xs	X coordinates start position of long edge corresponding to Ys
dXdy	X DDA value of long edge direction
XUs	X coordinates start position of upper edge
dXUdy	X DDA value of upper edge direction
XLs	X coordinates start position of lower edge
dXLdy	X DDA value of lower edge direction
USN	Count of spans of upper triangle. If this value is “0”, the upper triangle is not drawn.
LSN	Count of spans of lower triangle. If this value is “0”, the lower triangle is not drawn.

(Color setting register)

<table>
<tr><th>Register</th><th>Address</th><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td>Rs</td><td>0040$_H$</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>dRdx</td><td>0044$_H$</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>dRdy</td><td>0048$_H$</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>Gs</td><td>004C$_H$</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>dGdx</td><td>0050$_H$</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>dGdy</td><td>0054$_H$</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>Bs</td><td>0058$_H$</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>dBdx</td><td>005c$_H$</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>dBdy</td><td>0060$_H$</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>As</td><td>0064$_H$</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>dAdx</td><td>0068$_H$</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
<tr><td>dAdy</td><td>006c$_H$</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td>S</td><td colspan="8">Int</td><td colspan="16">Frac</td></tr>
</table>

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
As	Alpha value at (Xs, Ys, Zs) of long edge corresponding to Ys
dAdx	Alpha DDA value of horizontal direction
dAdy	Alpha DDA value of long edge

(Z coordinates register)

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

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$_H$	S	S	S	Int													Frac															
dSdx	00c4$_H$	S	S	S	Int													Frac															
dSdy	00c8$_H$	S	S	S	Int													Frac															
Ts	00cc$_H$	S	S	S	Int													Frac															
dTdx	00d0$_H$	S	S	S	Int													Frac															
dTdy	00d4$_H$	S	S	S	Int													Frac															
Qs	00d8$_H$	0	0	0	0	0	0	0	Int	Frac																							
dQdx	00dc$_H$	S	S	S	S	S	S	S	Int	Frac																							
dQdy	00e0$_H$	S	S	S	S	S	S	S	Int	Frac																							

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

Sets texture 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

13.2.9 Line drawing registers

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

(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
LPN	0140$_H$	0	0	0	0	Int												0															
LXs	0144$_H$	S	S	S	S	Int												Frac															
LXde	0148$_H$	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	Int	Frac															
LYs	014c$_H$	S	S	S	S	Int												Frac															
LYde	0150$_H$	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	Int	Frac															
LZs	0154$_H$	S	Int																Frac														
LZde	0158$_H$	S	Int																Frac														

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

Sets coordinates parameters for line drawing

LPN	Pixel count of principal axis direction
LXs	X coordinates start position of draw line (In principal axis X) Integer value of X coordinates rounded off (In principal axis Y) X coordinates in form of fixed point data
LXde	Inclination data for X coordinates (In principal axis X) Increment or decrement according to drawing direction (In principal axis Y) Fraction part of DX/DY
LYs	Y coordinates start position of draw line (In principal axis X) Y coordinates in form of fixed point data (In principal axis Y) Integer value of Y coordinates rounded off
LYde	Inclination data for Y coordinates (In principal axis X) Fraction part of DY/DX (In principal axis Y) Increment or decrement according to drawing direction
LZs	Z coordinates start position of line drawing line
LZde	Z Inclination

13.2.10 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$_H$	0	0	0	0	Int												0															
PYdc	0184$_H$	0	0	0	0	Int												0															
PZdc	0188$_H$	0	0	0	0	Int												0															

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

Sets 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

13.2.11 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$_H$	0	0	0	0	Int												0															
RYs	0204$_H$	0	0	0	0	Int												0															
RsizeX	0208$_H$	0	0	0	0	Int												0															
RsizeY	020c$_H$	0	0	0	0	Int												0															

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

Sets 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

13.2.12 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$_H$	0	0	0	0	0	0	0	Address																								
SStride	0244$_H$	0	0	0	0	Int												0															
SRXs	0248$_H$	0	0	0	0	Int												0															
SRYs	024c$_H$	0	0	0	0	Int												0															
DADDR	0250$_H$	0	0	0	0	0	0	0	Address																								
DStride	0254$_H$	0	0	0	0	Int												0															
DRXs	0258$_H$	0	0	0	0	Int												0															
DRYs	025c$_H$	0	0	0	0	Int												0															
BRsizeX	0260$_H$	0	0	0	0	Int												0															
BRsizeY	0264$_H$	0	0	0	0	Int												0															
TColor	0280$_H$	0																Color															

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

Sets parameters for Blt operations

SADDR	Sets start address of source rectangle area in byte address
SStride	Sets stride of source
SRXs	Sets X coordinates start position of source rectangle area
SRYs	Sets Y coordinates start position of source rectangle area
DADDR	Sets start address of destination rectangle area in byte address
DStride	Sets stride of destination
DRXs	Sets X coordinates start position of destination rectangle area
DRYs	Sets Y coordinates start position of destination rectangle area
BRsizeX	Sets horizontal size of rectangle
BRsizeY	Sets vertical size of rectangle
Tcolor	Sets transparent color For indirect color, set a palette code in the lower 8 bits.

13.2.13 High-speed 2D line drawing registers

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

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

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

Sets coordinates of line end points for High-speed 2DLine drawing

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

13.2.14 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 2DTriangle 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

13.2.15 Geometry control register

GCTR (Geometry Control Register)

Register address	GeometryBaseAddress + 00$_H$													
Bit number	31 30 29 28 27 26 25	24	23 22 21	20 19 18 17 16 15	14	13	12	11 10	9 8	7 6	5 4	3 2	1 0	
Bit field name	Reserved	FO	Rsv	FCNT	NF	FF	FE	Rsv	GS	Rsv	SS	Rsv	PS	
R/W	RX	RX	RX	RX	RX	RX	RX	RX	R	RX	R	RX	R	
Initial value	X	0	X	011111	0	0	1	X	00	X	00	X	00	

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

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

Bit 1 and 0 PS (Pixel engine Status)

Indicates status of pixel engine unit

00	Idle
01	Processing
10	Reserved
11	Reserved

Bit 5 and 4 SS (geometry Setup engine Status)

Indicates status of geometry setup engine unit

00	Idle
01	Processing
10	Processing
11	Reserved

Bit 9 and 8 GS (Geometry engine Status)

Indicates status of geometry engine unit

00	Idle
01	Processing
10	Reserved
11	Reserved

Bit 12 FE (FIFO Empty)

Indicates whether the data is contained in display list FIFO (DFIFOD)

0	Data in DFIFOD
1	No data in DFIFOD

Bit 13 FF (FIFO Full)

Indicates whether display list FIFO (DFIFOD) is full or not

0	DFIFOD not full
1	DFIFOD full

Bit 14 NF (FIFO Near Full)

Indicates free space in display list FIFO (DFIFOD)

0 More than half of DFIFOD free

1 Less than half of DFIFOD free

Bit 20 to 15 FCNT (FIFO Counter)

Indicates count of free stages (0 to 011111_B) of display list FIFO (DFIFOD)

Bit 24 FO (FIFO Overflow)

Indicates whether FIFO overflow occurred

0 Normal

1 FIFO overflow

13.2.16 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$_H$																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name				FX																					CF	DF			ST	Z	C	F
R/W				w																					w	W			w	w	w	w
Initial value				0																					0	00			0	0	0	0

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

Bit28 FX (Float Setup eXpand)

Enable Float Setup mode (See Geometry command code table)

Work Only for G_Begin/Triangle(s,_Strip,_Fan)

0 disable

1 enable

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	
	Work only for C=0,Z=0 and ST=0 (XY only vertex) mode	
	0	Orthogonal projection transformation mode
	1	Perspective projection transformation mode

GMDR1 (Geometry Mode Register for Line)

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

This register sets the geometry mode at line drawing. This register is sharing hardware with GMDR1E, so that if GMDR1 is changed, the same bit of GMDR1E is also changed.

Bit 4 BO (Broken line Offset)

Sets broken line reference position

If you want clear initial vertex only SetRegister BLPO before G_Begin and Set 1 for this bit. (Cannot change GMDR1 within G_Begin/G_End)

0	Broken line reference position not cleared for all vertexes.
1	Broken line reference position cleared for all vertexes.

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”.

This register is sharing hardware with GMDR1, so that if GMDR1E is changed, the same bit of GMDR1 is also changed.

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

If you want clear initial vertex only SetRegister BLPO before G_Begin(E) and Set 1 for this bit. (Cannot change GMDR1E within G_Begin(E)/G_End(E))

0	Reference position of dashed-line pattern cleared for all vertexes
1	Reference position of dashed-line pattern not cleared for all vertexes

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

Bit 2 FD (Face Definition)

Sets the face definition

0 Face defined as state with vertexes arranged clockwise

1 Face defined as state with vertexes arranged counterclockwise

Bit 0 CF (Cull Face)

Sets the drawing mode of the back

0 Back drawn

1 Back not drawn (value disabled for polygons)

GMDR2E (Geometry Mode Register for Triangle Extension)

Register address	(SetGModeRegister)																															
Bit number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																						TL		SP						FD		CF
R/W																						W		W						W		W
Initial value																						0		0						0		0

This register sets the geometry processing extended mode at triangle drawing.

In case of TL=1 with texture mapping, please set perspective correction.

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

Bit 10 TL (Top-Left rule mode)

Sets the drawing algorithm

0 Top-left rule applied (compatible with CREMSON)

1 Top-left rule not applied

Bit 8 SP (Shadow Primitive)

Sets the drawing mode for the shadow primitive

0 Shadow primitive not drawn

1 Shadow primitive drawn

Bit 2 FD (Face Definition)

Sets the face definition

0 Face defined as state with vertexes arranged clockwise

1 Face defined as state with vertexes arranged counterclockwise

Bit 0 CF (Cull Face)

Sets the drawing mode of the back

0 Back drawn

1 Back not drawn (value disabled for polygons)

13.2.17 Display list FIFO registers

DFIFOG (Geometry Displaylist FIFO with Geometry)

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

FIFO registers for Display List transfer

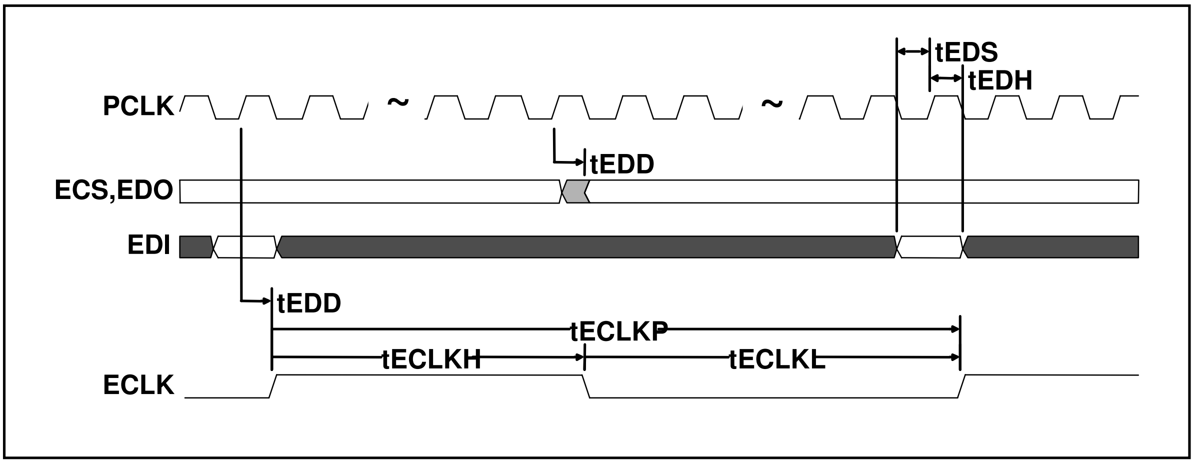
14. TIMING DIAGRAM

14.1 Host Interface

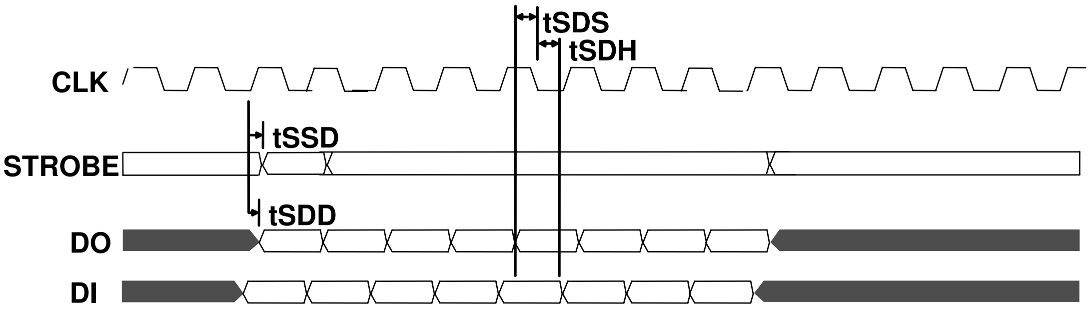
14.1.1 PCI Interface

Standard PCI V2.1.

14.1.2 EEPROM Timing



14.1.3 Serial Interface Timing



14.2 I^{2}C Interface

I^{2}C Bus Timing

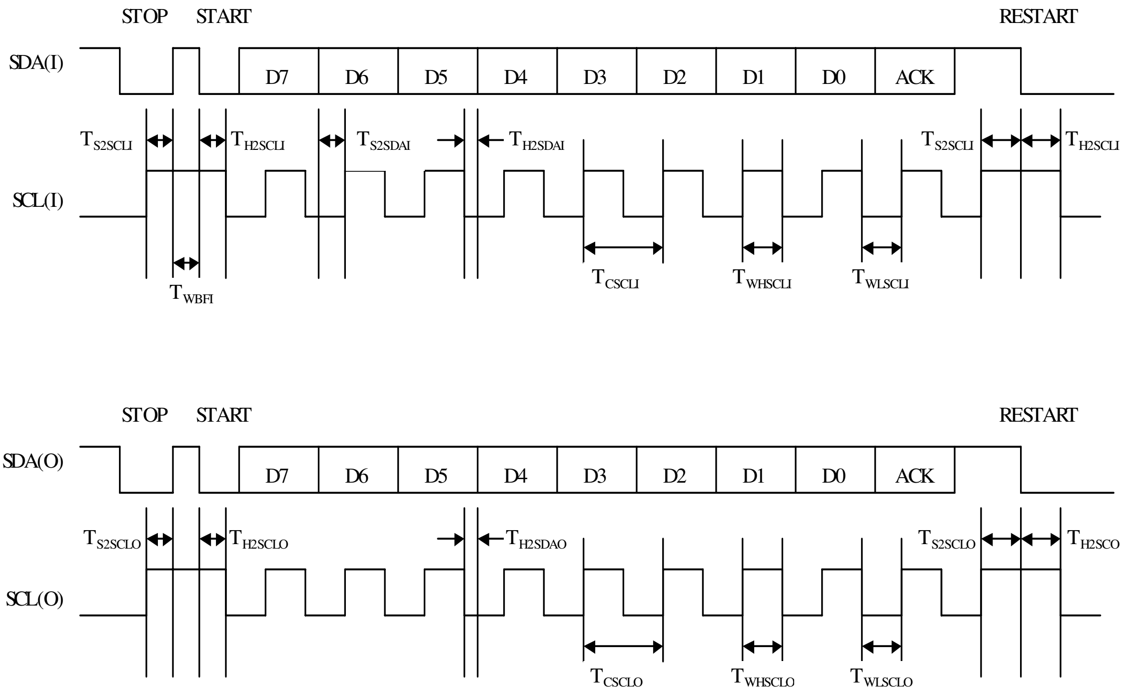
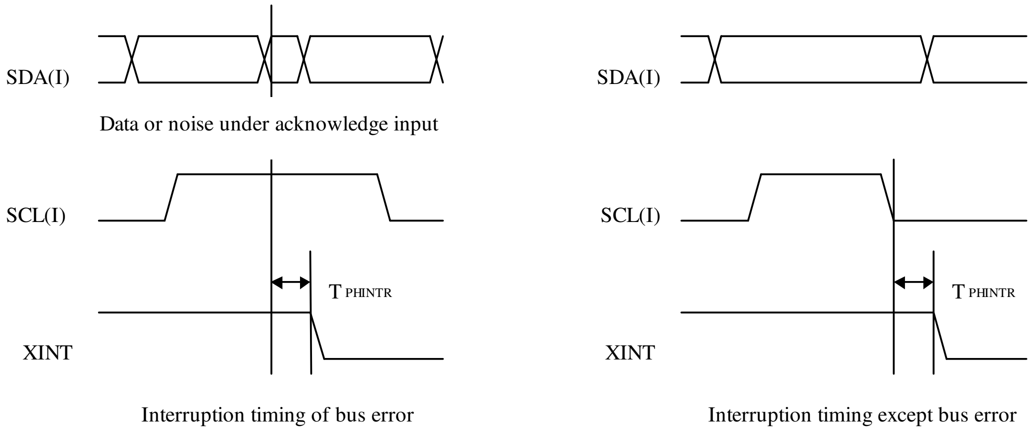


Fig.11.1 I2C bus timing

Interruption Timing



Interruption timing of bus error

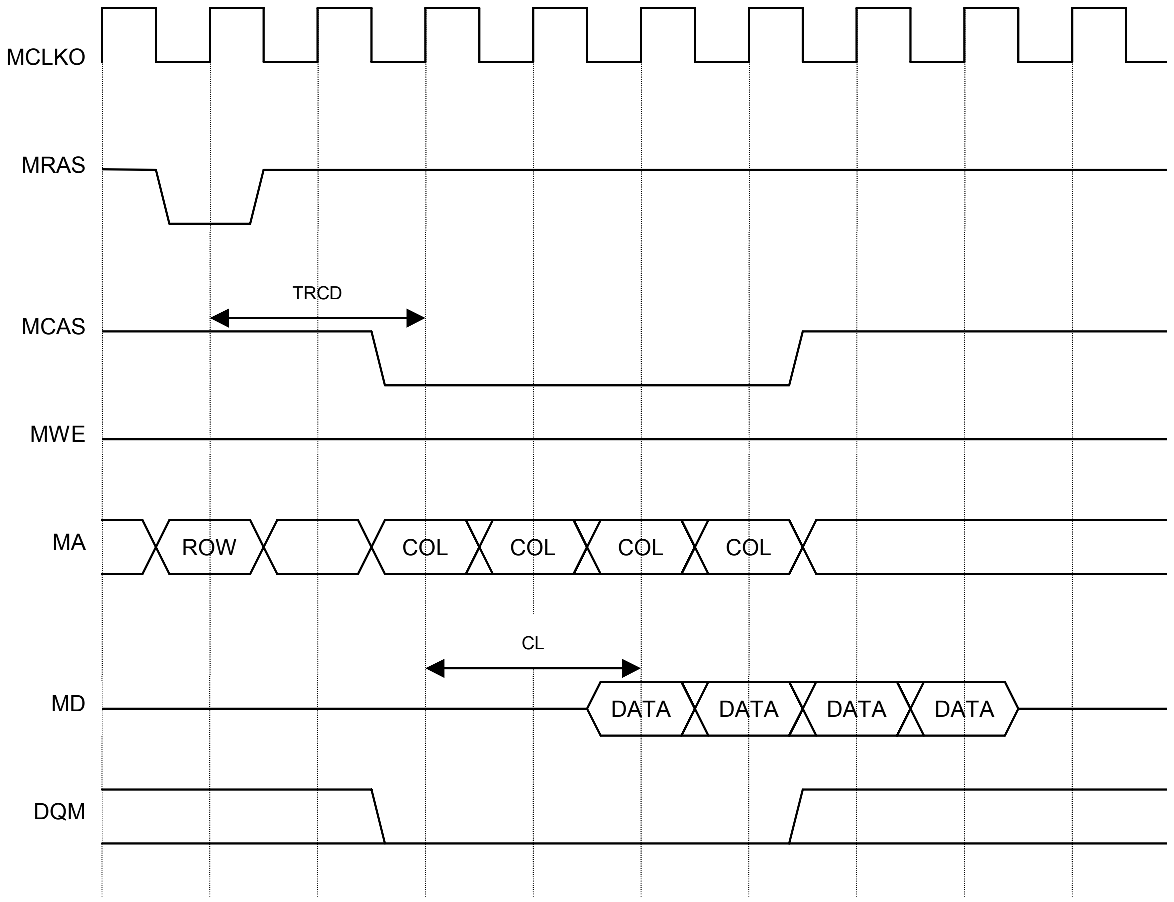
Interruption timing except bus error

Fig.11.2 Interruption timing

14.3 Graphics Memory Interface

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

14.3.1 Timing of read access to same row address



ROW: Row Address
COL: Column Address
DATA: READ DATA

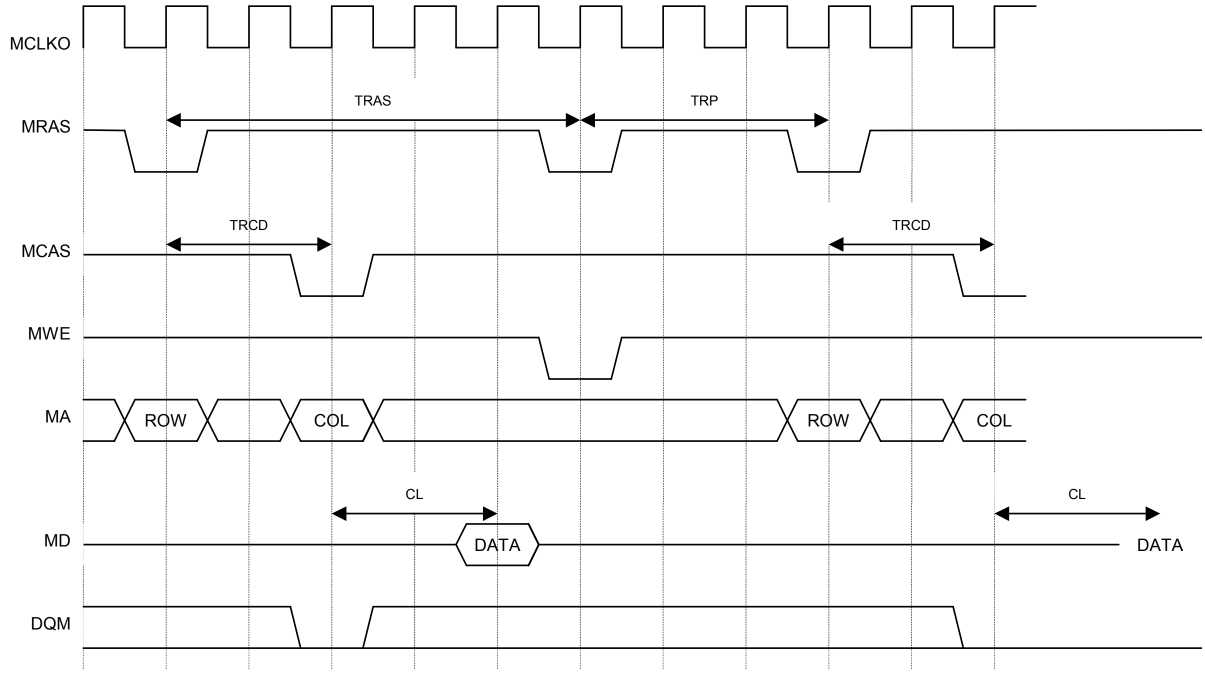
TRCD: RAS to CAS Delay Time
CL: CAS Latency

*Timing when CL2 operating

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.

14.3.2 Timing of read access to different row addresses



ROW: Row Address
COL: Column Address
DATA: READ DATA

TRAS: RAS Active Time
TRCD: RAS to CAS Delay Time
CL: CAS Latency

TRP: RAS Precharge Time

*Timing when CL2 operating

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.

14.3.3 Timing of write access to same row address

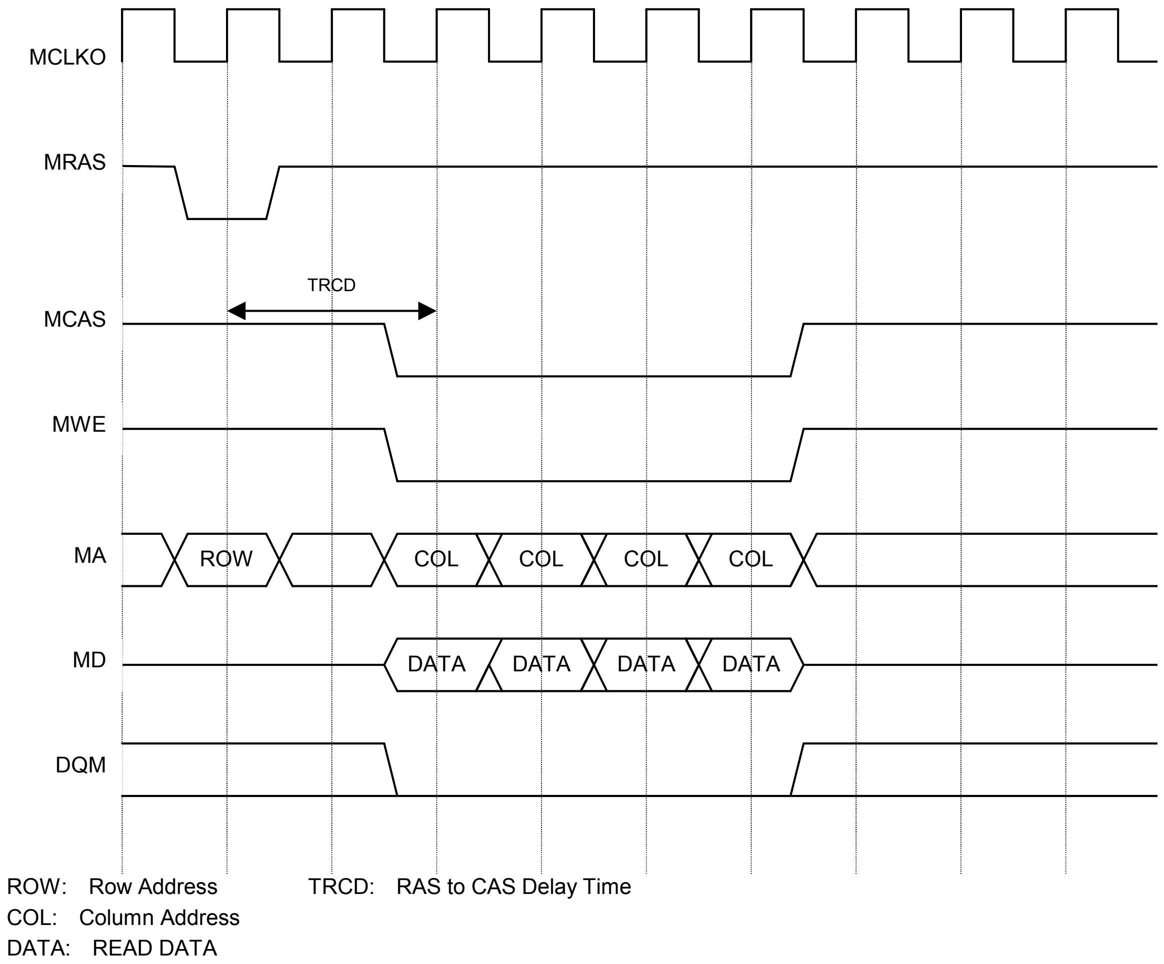
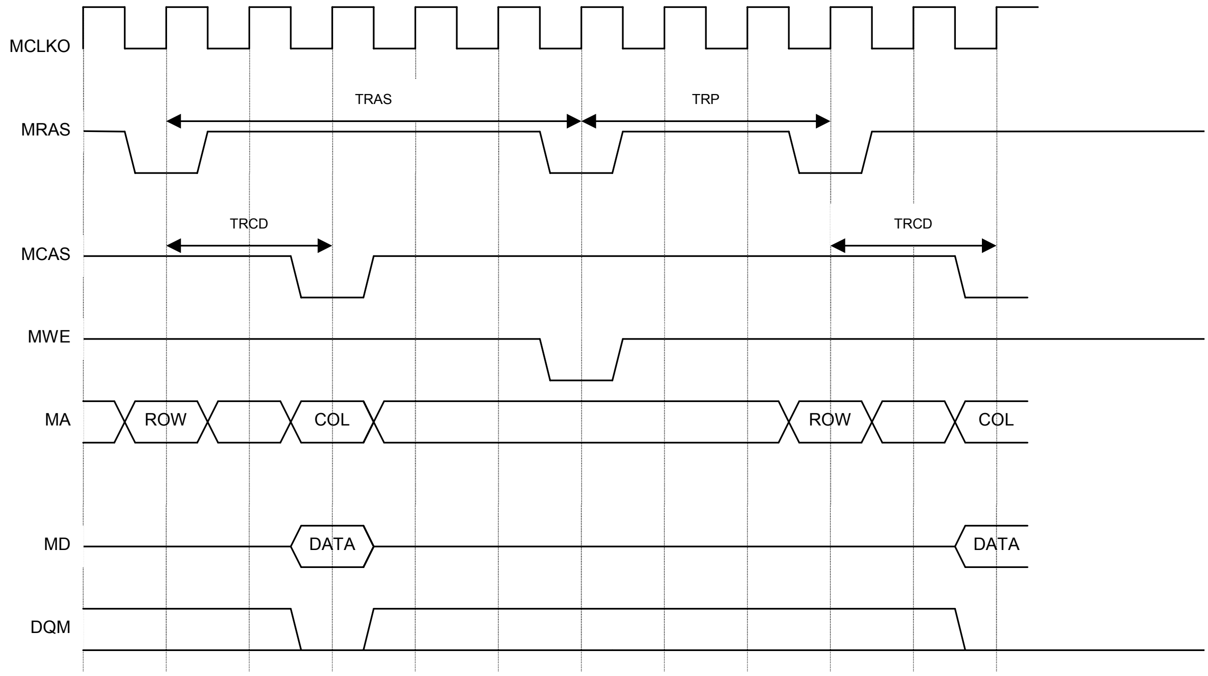


Fig. 11.5 Timing of Write Access to Same Row Address

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.

14.3.4 Timing of write access to different row addresses

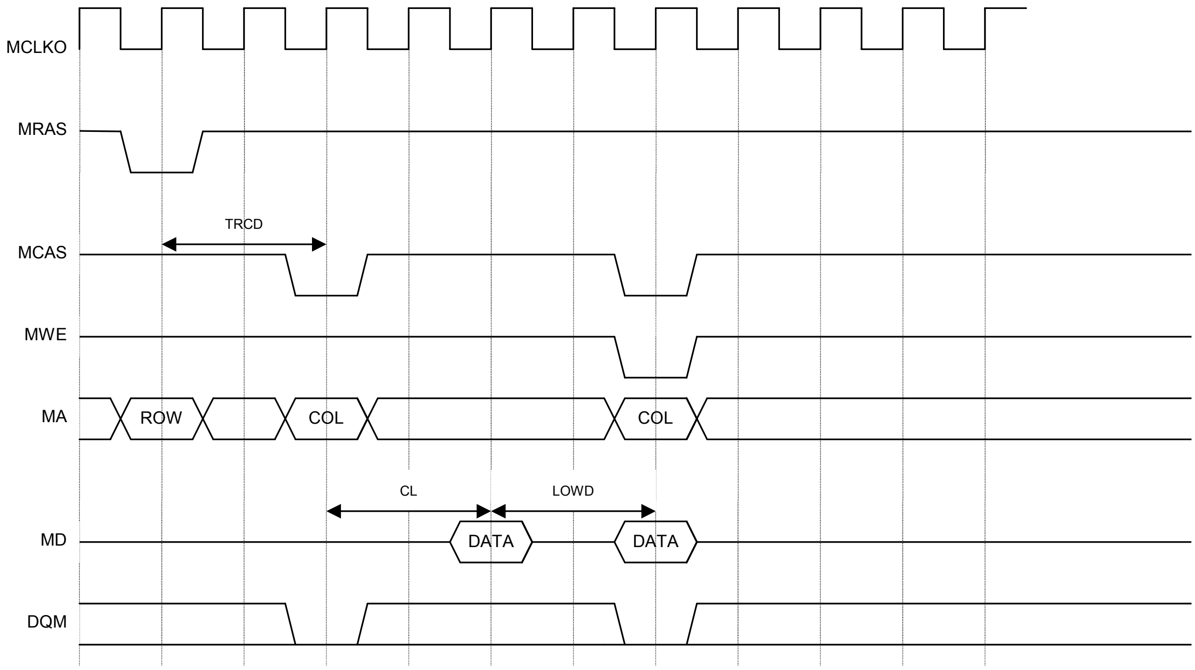


ROW: Row Address
COL: Column Address
DATA: READ DATA
TRAS: RAS Active Time
TRCD: RAS to CAS Delay Time
TRP: RAS Precharge Time

Fig. 11.6 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.

14.3.5 Timing of read/write access to same row address



ROW: Row Address
COL: Column Address
DATA: READ DATA

TRAS: RAS Active Time
TRCD: RAS to CAS Delay Time
CL: CAS Latency

TRP: RAS Precharge Time
LOWD: Last Output to Write Command Delay

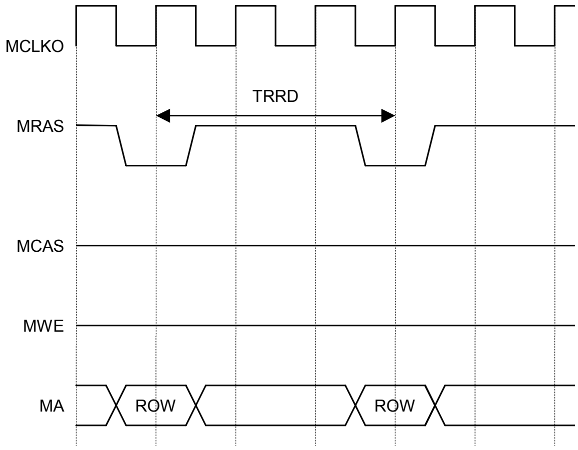
Timing when CL2 operating

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.

14.3.6 Delay between ACTV commands

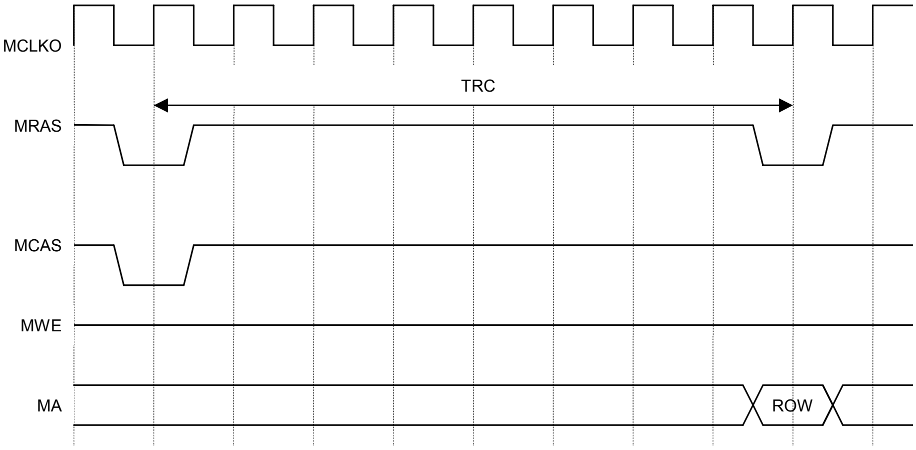


ROW: Row Address
TRRD: RAS to RAS Bank Active Delay Time

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.

14.3.7 Delay between Refresh command and next ACTV command



ROW: Row Address
TRC: RAS Cycle Time

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.

14.4 Display Timing

14.4.1 Non-interlace mode

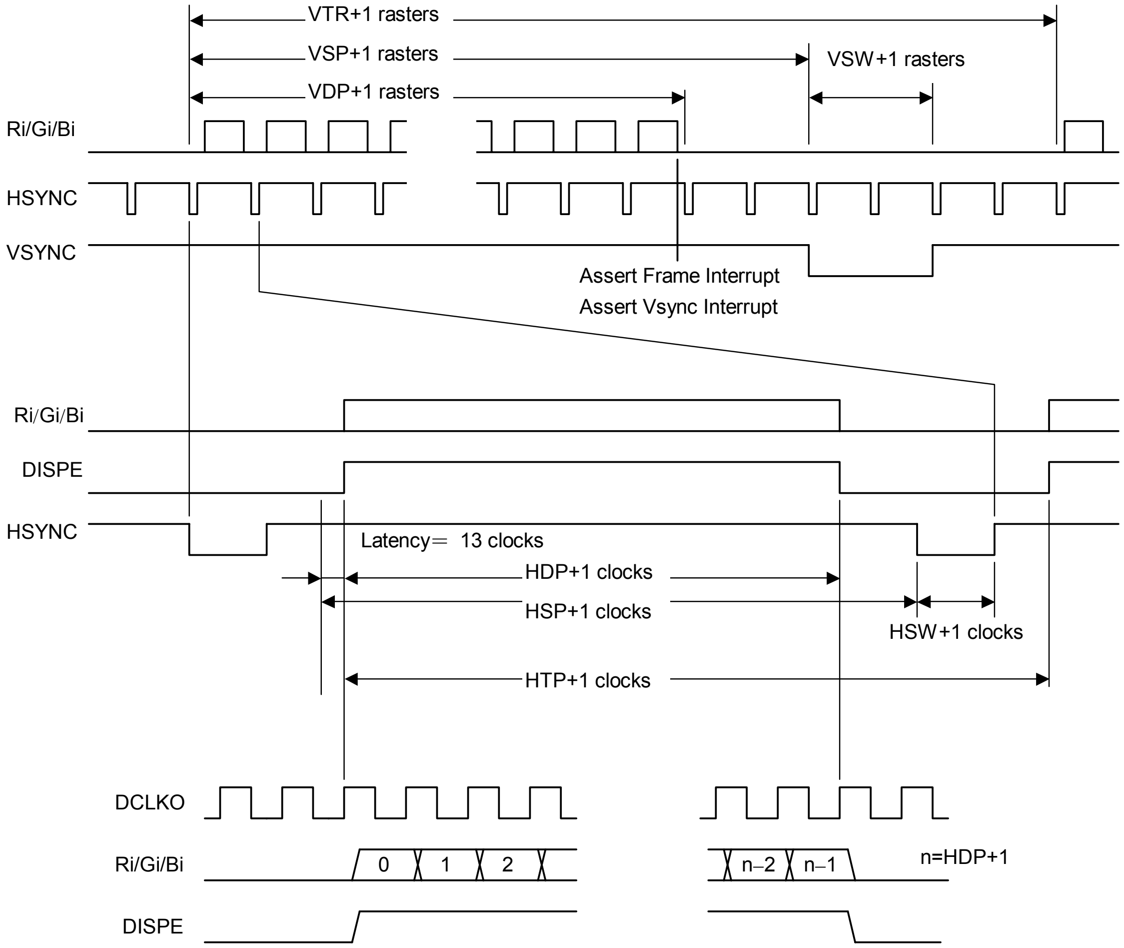


Fig. 11.10 Non-interlace Timing

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.

VSYNC is output 1 dot clock faster than HSYNC.

14.4.2 Interlace video mode

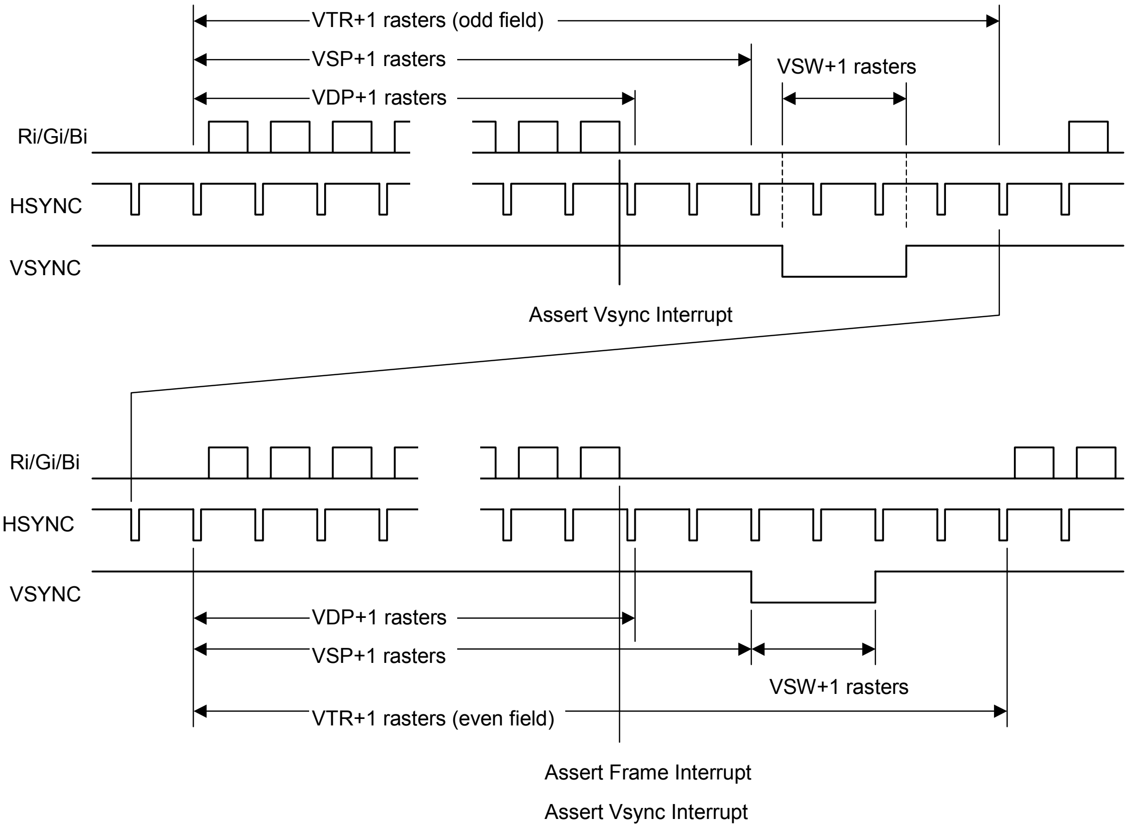


Fig. 11.11 Interlace Video Timing

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.

14.4.3 Composite synchronous signal

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

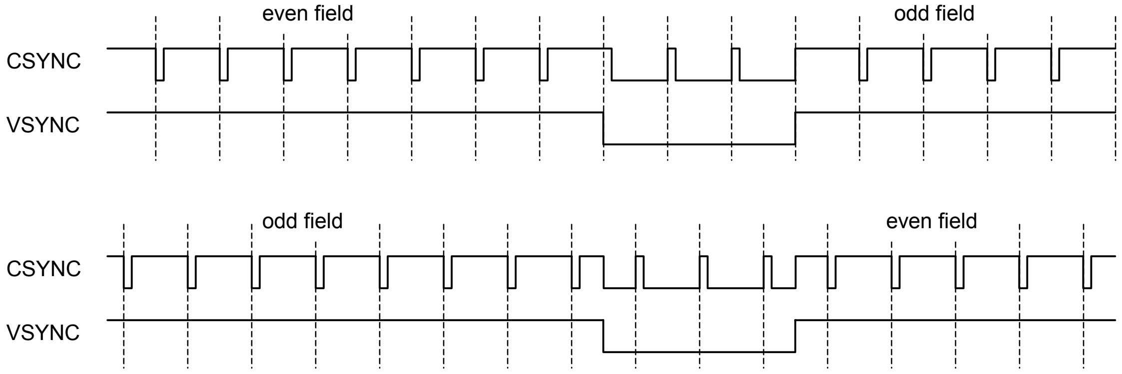


Fig 11.12 Composite Synchronous Signal without Equalizing Pulse

When the EEQ bit of the DCM register is “1”, the equalizing pulse is inserted into the CSYNC signal, producing the waveform shown below.

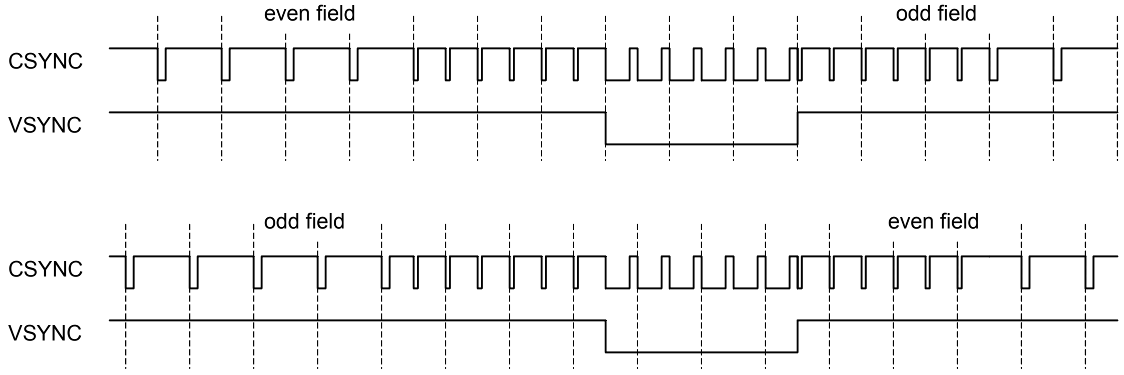


Fig 11.13 Composite Synchronous Signal with Equalizing Pulse

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.

15. ELECTRICAL CHARACTERISTICS

15.1 Introduction

The values in this chapter are valid for the final specification of MB86296.

15.2 Maximum Rating

Maximum Rating

Parameter	Symbol	Maximum rating	Unit
Power supply voltage	V_{DDL} [*1] V_{DDH}	$-0.5 < V_{DDL} < 2.5$ $-0.5 < V_{DDH} < 4.0$	V
Input voltage	V_I	$-0.5 < V_I < V_{DDH}+0.5$ (<4.0)	V
Output current	I_O	±13	mA
Ambient for storage temperature	TST	-55 < TST < +125	°C

*1 Includes PLL power supply

<Notes>

- Semiconductor devices can be permanently damaged by application of stress (voltage, current, temperature, etc) in excess of absolute maximum ratings. Do not exceed these ratings.
- Do not directly connect output pins or bidirectional pins of IC products to each other or VDD or VSS to avoid the breakdown of the device. However direct connection of the output pins or bidirectional pins to each other is possible, if the output pins are designed to avoid a conflict in a timing.
- Because semiconductor devices are particularly susceptible to damaged by static electricity, you must take the measure like ground all fixtures and instruments.
- In CMOS ICs, a latch-up phenomenon is caused when an voltage exceeding Vcc or an voltage below Vss is applied to input or output pins or a voltage exceeding the rating is applied across Vcc and Vss. When a latch-up is caused, the power supply current may be dramatically increased causing resultant thermal break-down of devices. To avoid the latch-up, make sure that the voltage does not exceed the maximum rating.

15.3 Recommended Operating Conditions

15.3.1 Recommended operating conditions

Recommended Operating Conditions

Parameter	Symbol	Rating			Unit
		Min.	Typ.	Max.	
Supply voltage	V_{DDL} [*1] V_{DDH} AVD	1.65 3.0 2.7	1.8 3.3 3.3	1.95 3.6 3.6	V
Current consumption (on 1.8V)	$I_{1.8V}$		500		mA
Current consumption (on 3.3V)	$I_{3.3V}$		100		mA
Input voltage (High level)	V_{IH}	2.0		V_{DDH}+0.3	V
Input voltage (low level)	V_{IL}	−0.3		0.8	V
Input voltage to VREF	VREF	1.05	1.10	1.15	V
VRO External resistance	RREF		2.7		K ohm
AOUT External resistance*[2]	RL		75		ohm
ACOMP External capacitance*[3]	CACOMP		0.1		uF
Ambient temperature for operation	TA	-40		85	°C

*1 Includes PLL power supply

*2 AOUTR, AOUTG, AOUTB pins

*3 ACOMPR, ACOMPG, ACOMPB pins

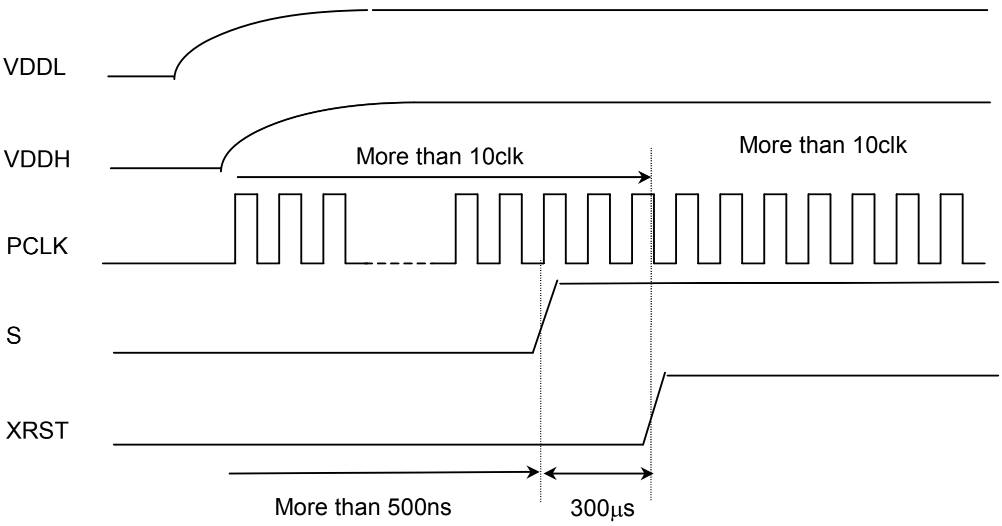
<Note>

- Recommended operating conditions are normal operating ranges for the semiconductor device. All the device's electrical characteristics are warranted when operated within these ranges. Always use semiconductor devices within the recommended operating conditions. Operation outside these ranges may adversely affect reliability and could result in device failure. No warranty is made with respect to uses, operating conditions, or combinations not represented on the manual. Users considering application outside the listed conditions are advised to contact their FUJITSU representative beforehand.

15.3.2 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 and XRST pins 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 continuously.

The S and XRST pins are reset during “Low” level period.
Immediately after power-on, input clock to the PCLK pin for 10 clk or more. The XRST is taken in synchronizing with the PCLK.

15.4 DC Characteristics

15.4.1 DC Characteristics of PCI Buffer

Symbol	Parameter	Condition	PCI Spec		Unit
			Min	Max	
Vcc	Supply Voltage		3.0	3.6	V
Vih	Input High Voltage		0.5Vcc	Vcc+0.5	V
Vil	Input Low Voltage		-0.5	0.3Vcc	V
Vipu	Input Pull-up Voltage		0.7Vcc		V
Iil	Input Leakage Current	0<Vin<Vcc		+/－10	uA
Voh	Output High Voltage	Iout=－0.5mA	0.9Vcc		V
Vol	Output Low Voltage	Iout＝1.5mA		0.1Vcc	V

Symbol	Parameter	Condition	PCI Spec		Unit
			Min	Max	
Ioh	Switching Current High	0＜Vout＜0.3Vcc	Refer “V-I characteristics diagram”		ｍA
		0.3Vcc＜Vout＜0.9Vcc			ｍA
		0.7Vcc<Vout<Vcc			ｍA
Iol	Switching Current Low	Vcc>Vout>0.6Vcc	Refer “V-I characteristics diagram”		ｍA
		0.6Vcc＞Vout＞0.1Vcc			ｍA
		0.18Vcc＞Vout＞0			ｍA

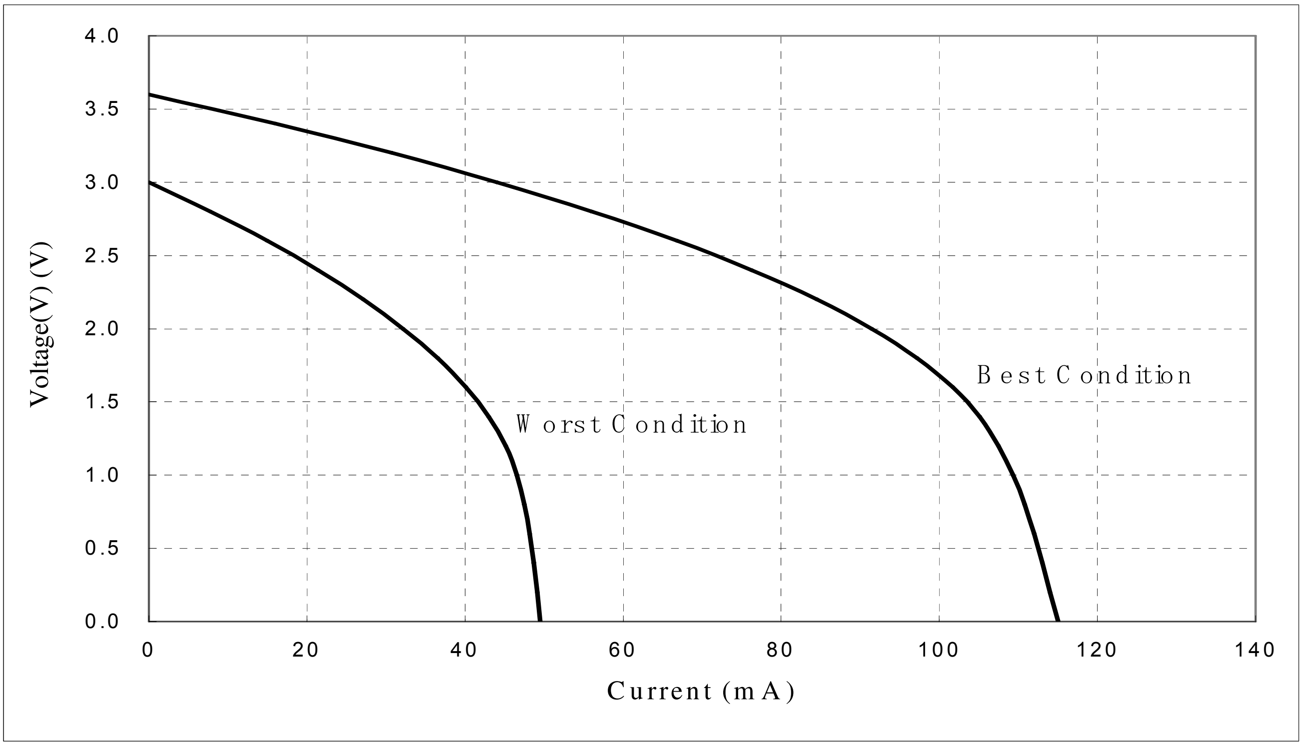


Fig **PCI** Buffer VI Curve (Pull Up)

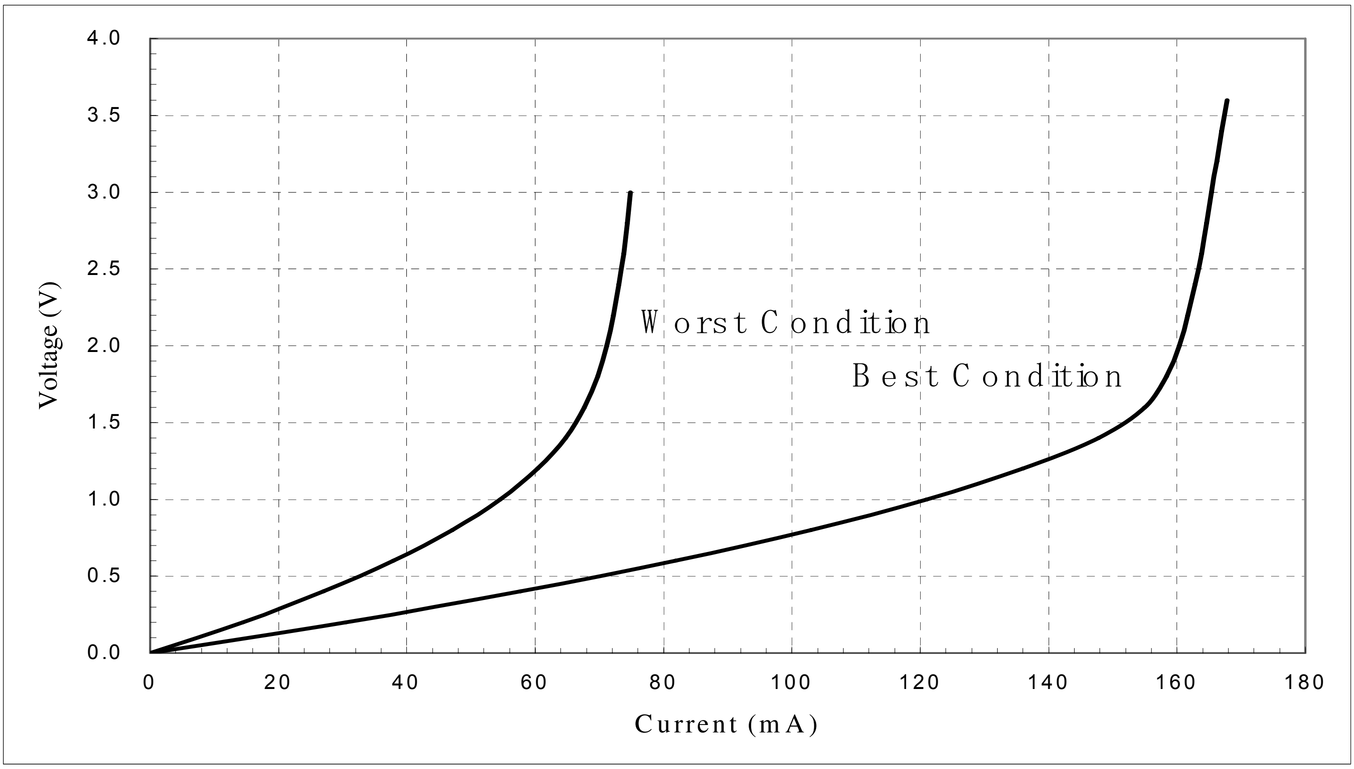


Fig **PCI** Buffer VI Curve (Pull Down)

15.4.2 DC Characteristics of other than PCI buffer

Measuring condition: V_{DDL} = 1.8 ± 0.15 V, V_{DDH} = 3.3 ± 0.3 V, V_{SS} = 0.0 V, Ta = -40 - +85°C

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
Output voltage ("High" level)	V_{OH}	I_{OH}=-100uA	V_{DDH}-0.2		V_{DDH}	V
Output voltage ("Low" level)	V_{OL}	I_{OL}=100uA	0.0		0.2	V
Output current ("High" level)	--	V_{DDH}=3.3V±0.3V	(*1)			mA
Output current ("Low" level)	--	V_{DDH}=3.3V±0.3V	(*1)			mA
AOUT Output current[*2] Full Scale*[3] Zero Scale	IAOUT	VREF=1.1V, RREF=2.7k ohm	9.38 0	10.42 2	11.48 20	mA uA
AOUT Output Voltage[*2]	VAOUT	VREF=1.1V, RREF=2.7k ohm RL=75 ohm	0		0.7815	V
Input leakage current	IL				±5	μA
Pin capacitance	C				16	pF

*1: Please refer "V-I characteristics diagram".

L Type: Output characteristics of MD0-63, MDQM0-7 pins

M Type: Output characteristics of pins other than signals indicated by L type and H type and PCI pins.

H Type: Output characteristics of MCLKO pins

*2: AOUTR, AOUTG, AOUTB pin

*3: Full Scale Output Current = (VREF/RREF) * 25.575

V-I characteristics diagram

Condition MAX: Process=Slow, Ta=85°C, V_{DD}=3.6V
TYP: Process=Typical, Ta=25°C, V_{DD}=3.3V
MIN: Process=Fast, Ta=-40°C, V_{DD}=3.0V

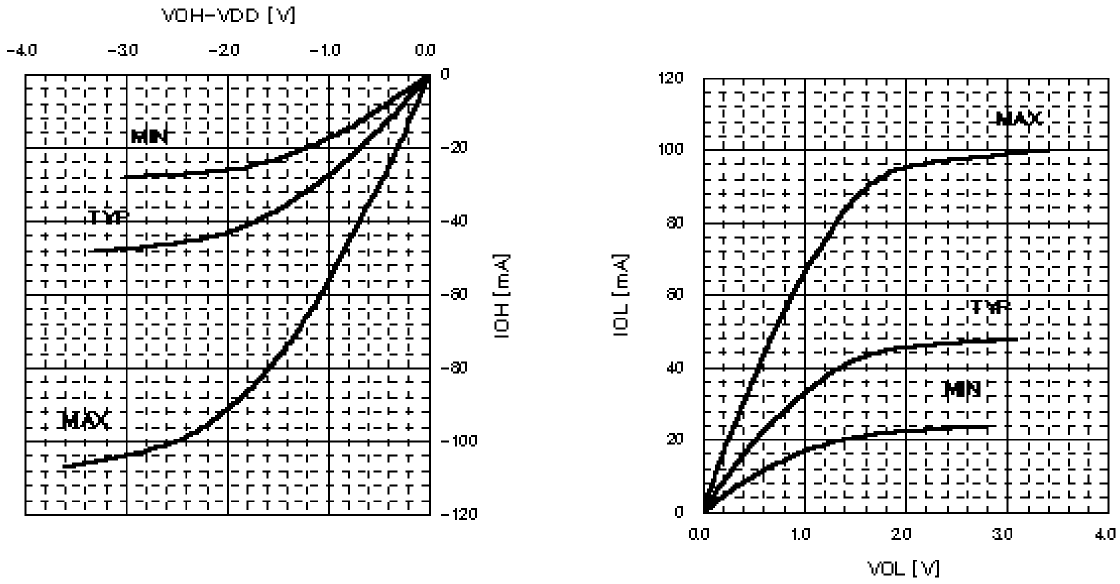


Fig. V-I characteristics L, M type

Condition MAX: Process=Slow, Ta=85°C, V_{DD}=3.6V
TYP: Process=Typical, Ta=25°C, V_{DD}=3.3V
MIN: Process=Fast, Ta=-40°C, V_{DD}=3.0V

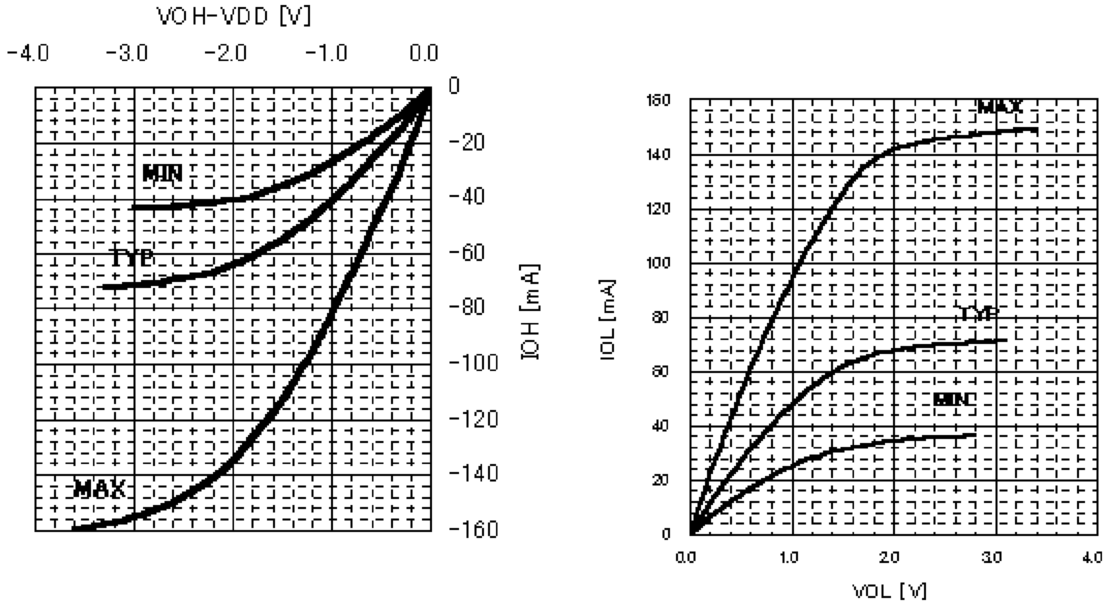


Fig. V-I characteristics H type

15.5 AC Characteristics

15.5.1 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 (bussed signals)	AD[31:0], C/BE[3:0], PAR, FRAME, IRDY, TRDY, STOP, IDSEL, DEVSEL, PERR	t_{PS}	7			ns
PCI Input Setup (point-to-point signals)	GNT	t_{PSP}	10			ns
PCI Input Hold	AD[31:0], C/BE[3:0], PAR, FRAME, IRDY, TRDY, STOP, IDSEL, DEVSEL, PERR, GNT	t_{PH}	0			ns
PCI Output Delay	AD[31:0], C/BE[3:0], PAR, FRAME, IRDY, TRDY, STOP, IDSEL, DEVSEL, PERR, SERR, REQ	t_{PD}	2		11	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

15.5.2 I^2C Interface

I^2C bus timing

symbol			MIN	MAX	unit
T_{S2SDAI}	SDA(I) setup time	standard	250		ns
		high-speed	100		ns
T_{H2SDAI}	SCL(I) hold time	standard	0		ns
		high-speed	0		ns
T_{CSCLI}	SCL(I) cycle time	standard	10.0		us
		high-speed	2.5		us
T_{WHSCLI}	SCL(I) H period	standard	4.0		us
		high-speed	0.6		us
T_{WLSCLI}	SCL(I) L period	standard	4.7		us
		high-speed	1.3		us
T_{CSCLO}	SCL(O) cycle time	standard	2*m+2(*2)		PCLK*1
		high-speed	int(1.5*m)+2(*2)		PCLK*1
T_{WHSCLO}	SCL(O) H period	standard	m+2(*2)		PCLK*1
		high-speed	int(0.5*m)+2(*2)		PCLK*1
T_{WLSCLO}	SCL(O) L period	standard	m(*2)		PCLK*1
		high-speed	m(*2)		PCLK*1
T_{W2SCLI}	SCL(I) setup time	standard	4.0		us
		high-speed	0.6		us
T_{H2SCLI}	SCL(I) hold time	standard	4.7		us
		high-speed	1.3		us
T_{WBFI}	bus free time	standard	4.7		us
		hirh-speed	1.3		us
T_{S2SCLO}	SCL(O) set up time	standard	m+2(*2)		PCLK*1
		high-speed	int(0.5*m)+2(*2)		PCLK*1
T_{H2SCLO}	SCL(O) hold time	standard	m-2(*2)		PCLK*1
		high-speed	int(0.5*m)-2(*2)		PCLK*1
T_{H2SDAO}	SDA(O) hold time		5		PCLK*1

*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_{PHINTR}	XINT delay (bus error)		4	PCLK
T_{PHINTR}	XINT delay (except bus error)		4	PCLK

15.5.3 Video interface

(1) Clock

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
CLK Frequency	f_{CLK}			14.318		MHz
CLK H-width	t_{HCLK}		25			ns
CLK L-width	t_{LCLK}		25			ns
DCLKI Frequency	f_{DCLKI}				67	MHz
DCLKI H-width	t_{HDCLKI}		5			ns
DCLKI L-width	t_{LDCLKI}		5			ns
DCLKO frequency	f_{DCLKO}				67	MHz

(2) Input signals

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
HSYNC Input pulse width	$t_{WHSYNC0}$	*1	3			clock
	$t_{WHSYNC1}$	*2	3			clock
HSYNC Input setup time	t_{SHSYNC}	*2	6			ns
HSYNC Input hold time	t_{HHSYNC}	*2	1			ns
VSYNC Input pulse width	$t_{WHSYNC1}$		1			HSYNC 1 cycle

*1 Applied only in PLL synchronization mode (CKS=0), reference clock output from internal PLL (cycle = 1/14*fCLK)

*2 Applied only in DCLKI synchronization mode (CKS=1), reference clock = DCLKI

(3) Output signals (standard)

This definition is applied for followig mode operations

1) single display & non-inverting DCLKO
   (MDen=0) & (DCKinv=0)
2) dual display & single-edge & non-inverting DCLKO
   (MDen=1) & (DCKed=0) & (DCKinv=0)

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
RGB Output delay time 1	T_{RGB1}		2		9	ns
DISPE Output delay time 1	t_{DEO1}		2		9	ns
HSYNC Output delay time 1	$t_{DHSYNC1}$		2		9	ns
VSYNC Output delay time 1	$t_{DVSYNC1}$		2		9	ns
CSYNC Output delay time 1	$t_{DCSYNC1}$		2		10	ns
GV Output delay time 1	t_{DGV1}		2		9	ns

(4) Output signals (inverting)

This definition is applied for followig mode operations

1) single display & inverting DCLKO
(MDen=0) & (DCKinv=1)
2) dual display & single-edge & inverting DCLKO
(MDen=1) & (DCKed=0) & (DCKinv=1)

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
RGB Output delay time 2	T_{RGB2}		2		9	ns
DISPE Output delay time 2	t_{DEO2}		2		9	ns
HSYNC Output delay time 2	$t_{DHSYNC2}$		2		9	ns
VSYNC Output delay time 2	$t_{DVSYNC2}$		2		9.5	ns
CSYNC Output delay time 2	$t_{DCSYNC2}$		2		10	ns
GV Output delay time 2	t_{DGV2}		2		9	ns

(5) Output signals (bi-edge)

This definition is applied for followig mode operations

dual display & bi-edge
(MDen=1) & (DCKed=1)

Parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
RGB Output delay time 3	T_{RGB3}		1.5		9	ns
DISPE Output delay time 3	t_{DEO3}		1.5		9	ns
HSYNC Output delay time 3	$t_{DHSYNC3}$		1.5		9	ns
VSYNC Output delay time 3	$t_{DVSYNC3}$		1.5		9	ns
CSYNC Output delay time 3	$t_{DCSYNC3}$		1.5		10	ns
GV Output delay time 3	t_{DGV3}		1.5		9	ns
RGB Output delay time 4	T_{RGB4}		1.5		9	ns
DISPE Output delay time 4	t_{DEO4}		1.5		9	ns
HSYNC Output delay time 4	$t_{DHSYNC4}$		1.5		9	ns
VSYNC Output delay time 4	$t_{DVSYNC4}$		1.5		9.5	ns
CSYNC Output delay time 4	$t_{DCSYNC4}$		1.5		10	ns
GV Output delay time 4	t_{DGV4}		1.5		9	ns

15.5.4 Video capture interface

clock

parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
CCLK (RGBCLK) frequency	f_{CCLK}			27	80	MHz
CCLK (RGBCLK) H-width	t_{HCCLKI}		5			ns
CCLK (RGBCLK) L-width	t_{LCCLKI}		5			ns

Input signals

parameter	Symbol	Condition	Rating			Unit
			Min.	Typ.	Max.	
VI setup time	t_{SVI}		6			ns
VI hold time	t_{HVI}		2			ns
HSYNCI setup time	t_{SHSI}		6			ns
HSYNCI hold time	t_{HHSI}		2			ns
VSYNCI setup time	t_{SVSI}		6			ns
VSYNCI hold time	t_{HVSI}		2			ns
RI setup time	t_{SRI}		6			ns
RI hold time	t_{HRI}		2			ns
GI setup time	t_{SGI}		6			ns
GI hold time	t_{HGI}		2			ns
BI setup time	t_{SBI}		6			ns
BI hold time	t_{HBI}		2			ns

15.5.5 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 7

External SDRAM capacitance

SDRAM x1	Total capacitance	Unit
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

(6) 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

(7) 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

(8) 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

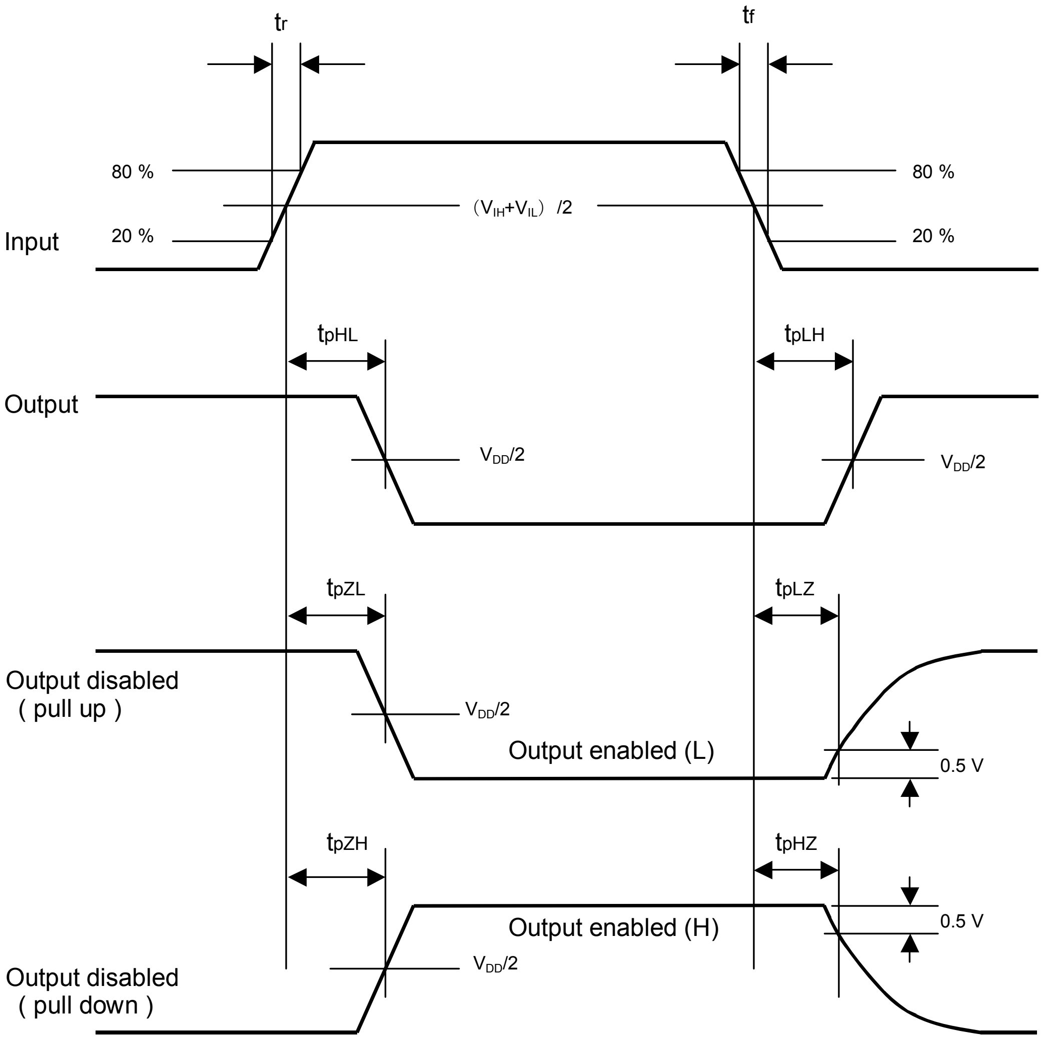
15.5.6 PLL specifications

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	CLKSEL0	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%

15.6 AC Characteristics Measuring Conditions



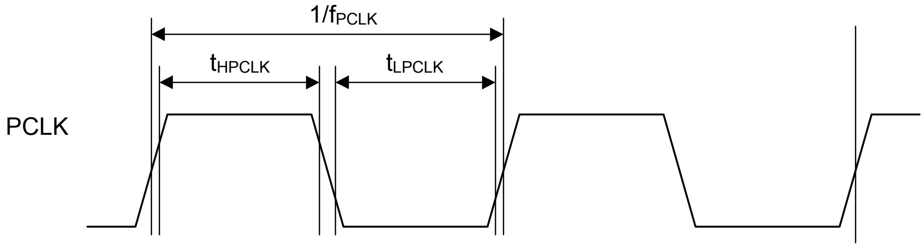
t_r, $t_f \leq 5$ ns

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

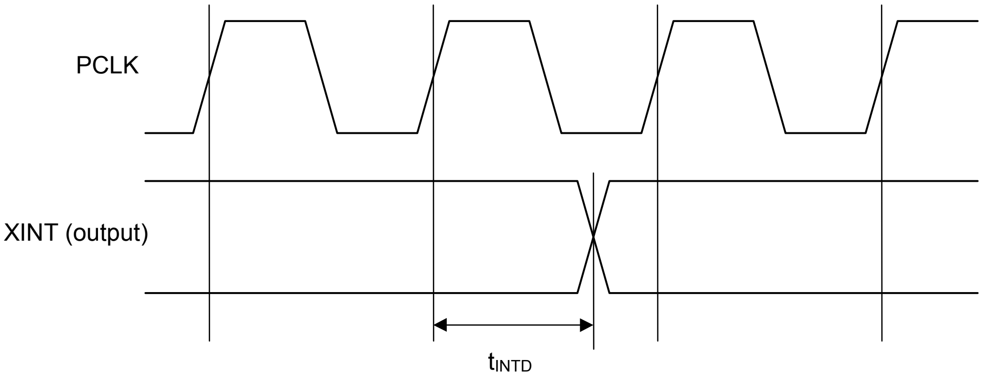
15.7 Timing Diagram

15.7.1 Host interface

Clock

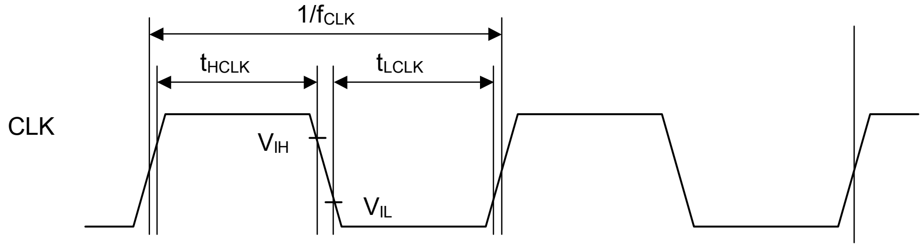


XINT output delay times

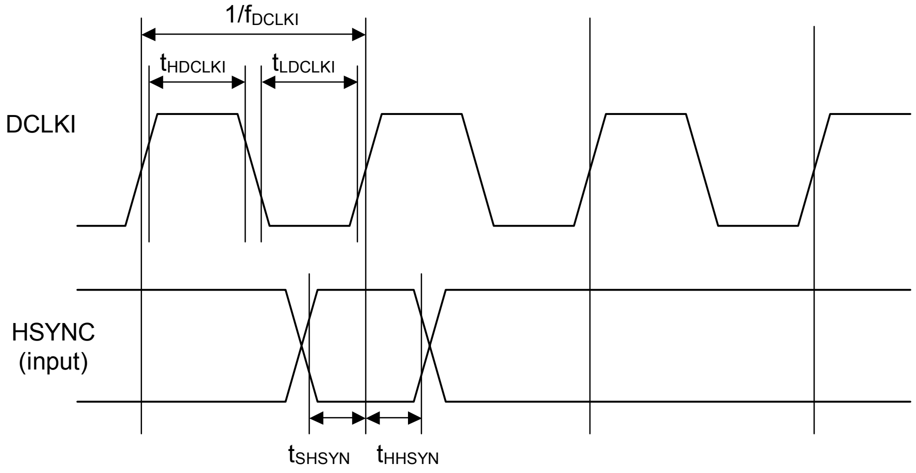


15.7.2 Video interface

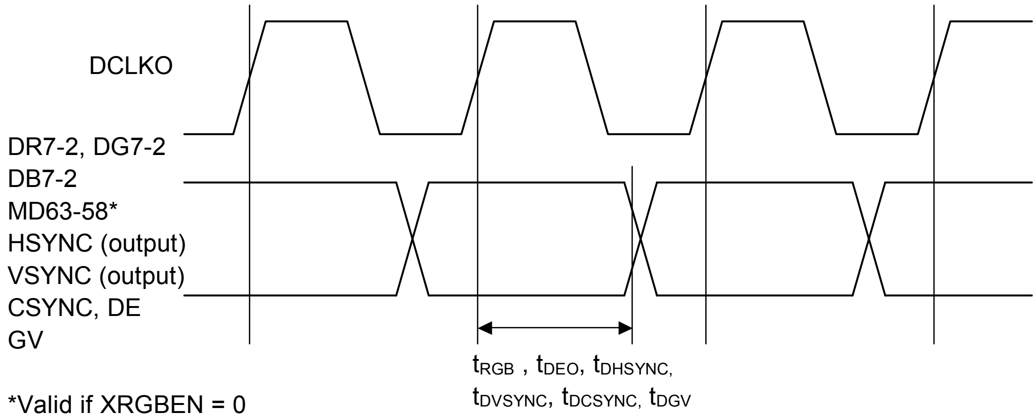
(1) Clock



(2) HSYNC signal setup/hold

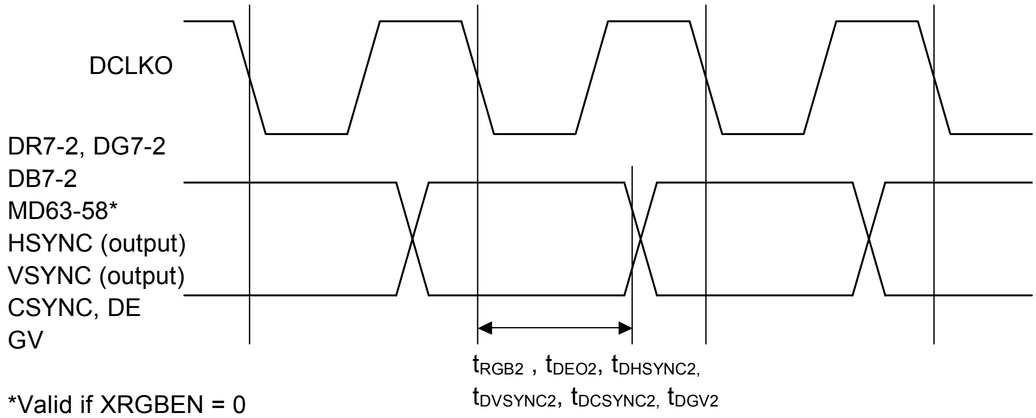


(3) Output signal delay (standard)



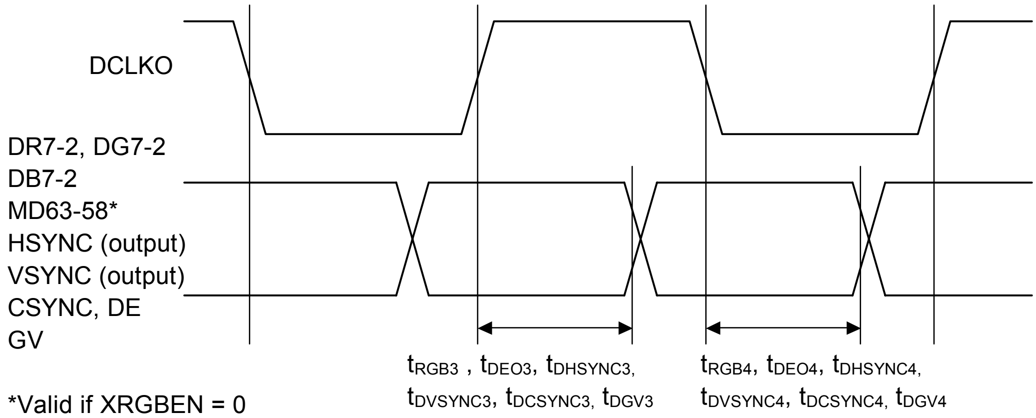
*Valid if XRGBEN = 0

(4) Output signal delay (inverted)



*Valid if XRGBEN = 0

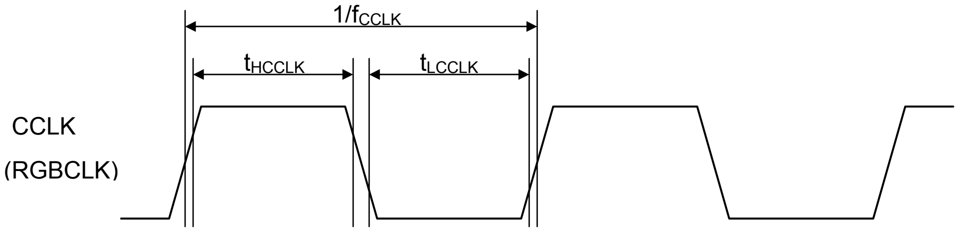
(5) Output signal delay (bi-edge)



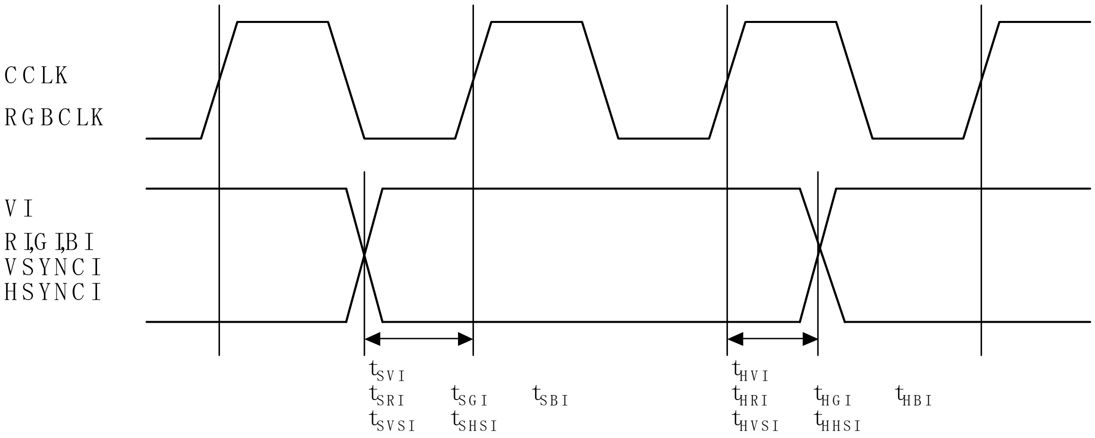
*Valid if XRGBEN = 0

15.7.3 Video capture interface

clock

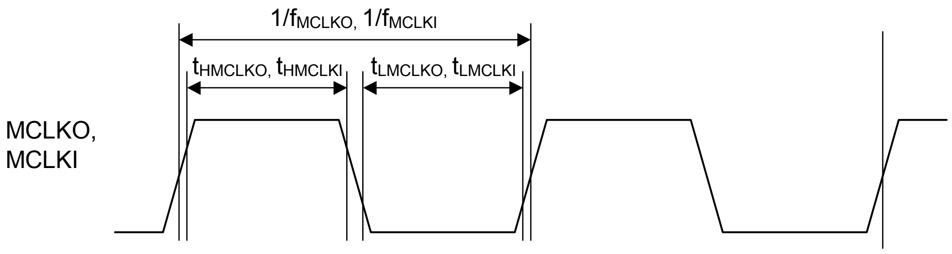


Video input

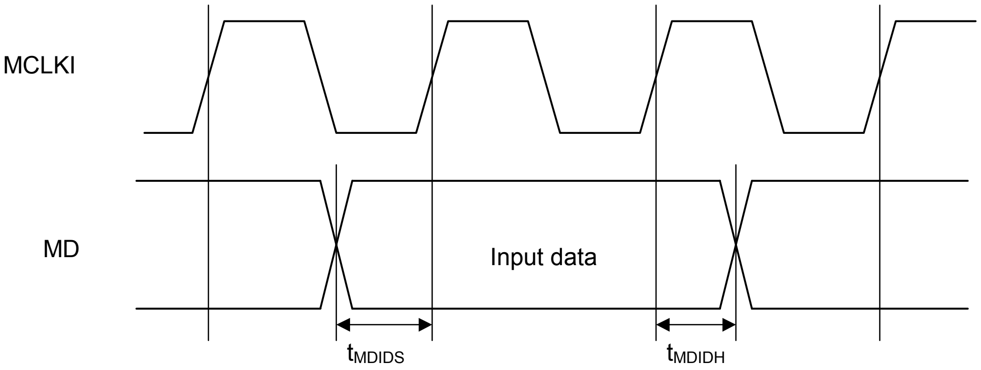


15.7.4 Graphics memory interface

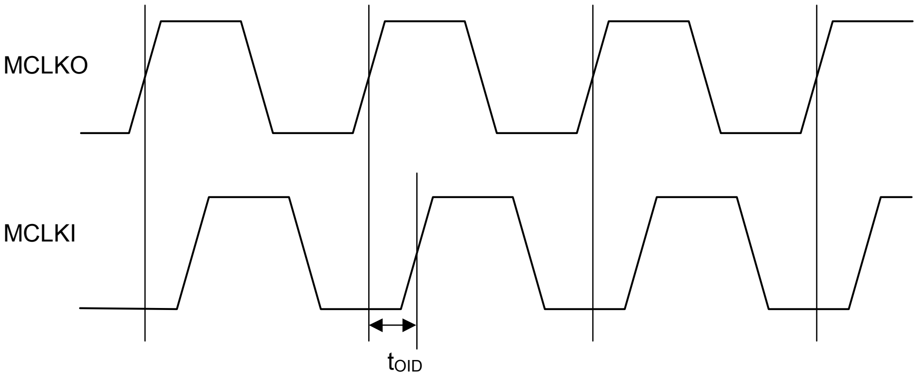
Clock



Input signal setup/hold time



MCLKI signal delay



Output signal delay