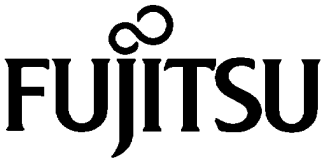
MB86290A

Graphics Controller
Hardware Specifications

Revision 2.0b

23 May 2000



1 Overview

1.1 Introduction

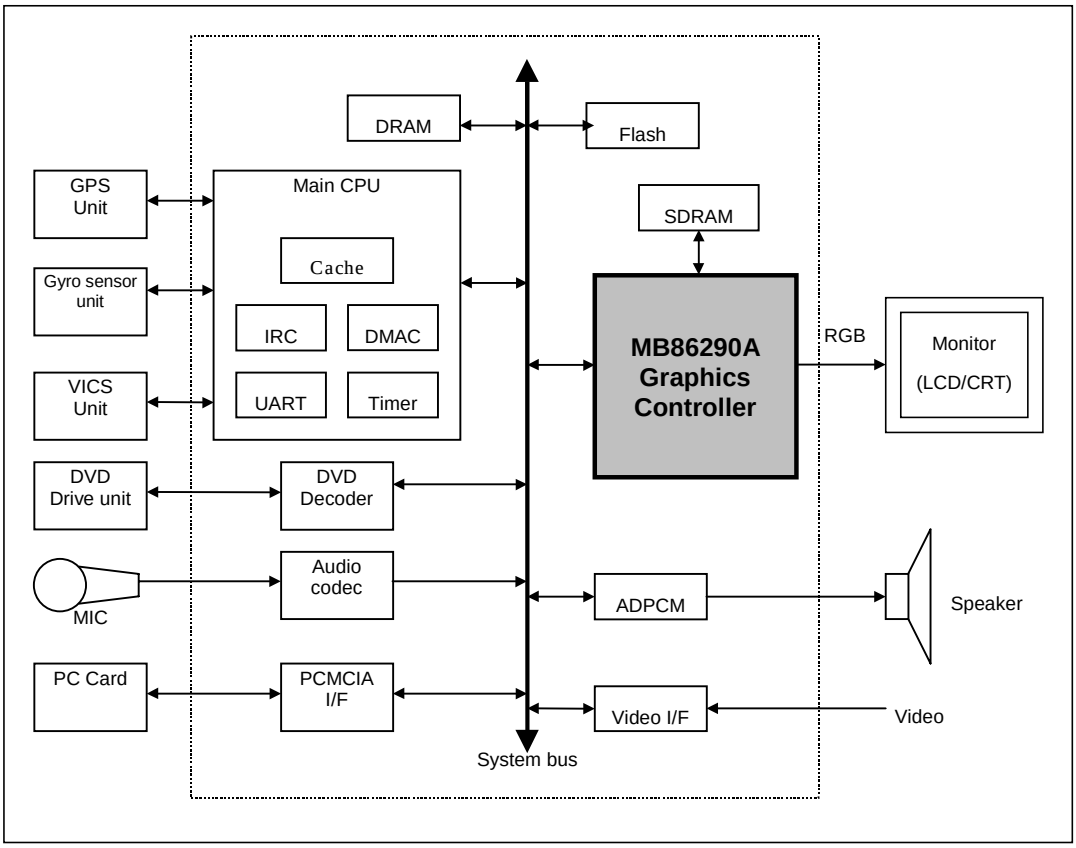
Recent consumer information processing systems, such as car navigation systems, require graphics capabilities for web page browsing and 3D object manipulation. The required performance level for these graphics operations is also increasing. This MB86290A graphics controller provides an optimized solution for these new requirements.

Target applications

- Car navigation systems
- Consumer information processing systems including digital STB
- Mobile IP terminals (Windows CE HPC/PPC)
- Consumer or arcade game machines

1.2 System Configuration

The following figure shows an example of a car navigation system using MB86290A.



System Configuration

1.3 Outline

- High performance

  The maximum operating frequency is 100 MHz. At this speed, the pixel fill rate is 100 MPixels/sec (2D drawing without special effects).

- Flexible display controller

  Display resolutions up to XGA (1024 × 768) and on-chip DAC are supported. The full screen can be split into two separate parts (left/right) each displaying different contents simultaneously. Smooth double-buffer-mode animation is supported. Each part of the screen can be scrolled independently. In addition, up to three screen layers can be overlaid. Alpha blending for transparent display of lower-layer contents is also supported. This function can be used to blend a navigation map onto a text window.

- 2D Rendering

  Anti-aliasing and alpha blending are supported to display high-quality graphics on low-resolution monitors.

- 3D Rendering

  Professional 3D rendering features, including perspective texture mapping, Gouraud shading, etc., are supported.

- Others

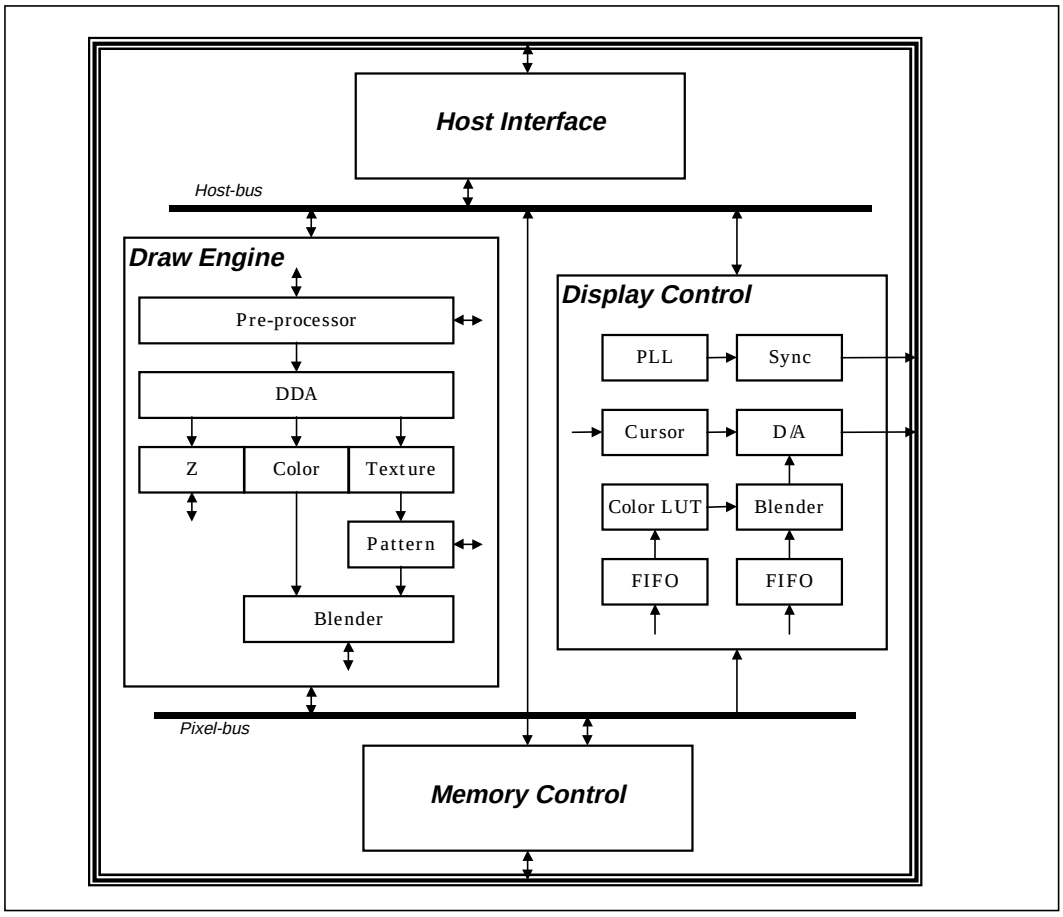
  CMOS 0.25-µm technology

  HQFP240 Package (lead pitch 0.5 mm)

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

1.4 Block Diagram

The MB86290A block diagram is shown below:



MB86290A Block Diagram

1.5 Functional Overview

1.5.1 System Configuration

Host CPU interface

MB86290A can be connected to Hitachi's SH3 or SH4 CPUs and NEC's V832 CPU without any glue logic. The host MB86290A CPU interface can drive the host CPU DMAC and transfer all graphical source data (display list, texture patterns, etc.) from the host (main) memory to it's internal registers (or external frame memory).

Graphics memory

Synchronous DRAM is attached externally. Either the 32-bit or 64-bit mode is supported as the interface with these external SDRAM devices. The external SDRAM operation frequency is the same as MB86290A (up to 100 MHz). Applicable memory device configurations are as follows:

Graphics Memory Device Configuration

Type	Data bus width	# of devices	Total capacity
SDRAM 64 Mbit (x32 bit)	32 bit	1	8 MB
SDRAM 64 Mbit (x32 bit)	64 bit	2	16 MB
SDRAM 64 Mbit (x16Bit)	64 bit	4	32 MB

Display signals

MB86290A has three channels of 8-bit D/A converters and outputs analog RGB signals. Superimposing is possible by applying an external sync signal.

1.5.2 Display Controller

Screen resolution

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

Screen Resolutions

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

Display colors

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

TV/Video display

MB86290A can output a graphics image synchronized with external TV/video display signals. The graphics image can be overlapped at any area on the TV/video display window. MB86290A outputs a control signal to switch the display window externally. This scheme supports both interlace and non-interlace.

Overlay

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

Hardware cursor

MB86290A supports two separate hardware cursor functions. Each of these hardware cursors is specified as a 64 × 64-pixel area. Each pixel of these hardware cursors is 8 bits and uses the indirect mode look-up table.

1.5.3 Frame Control

Double buffer scheme

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

Scroll scheme

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

Windows display

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

1.5.4 2D Drawing

2D Primitives

MB86290A provides automatic drawing of various primitives and patterns (drawing surfaces) to frame memory in either indirect color (8 bits/pixel referencing appropriate palette) or direct color (16 bits/pixel) mode. Alpha blending and anti-aliasing features are useful when the direct color mode is selected.

A triangle is drawn in a single color, mapped with a style image formed by a single color or 2D pattern (tiling), or mapped with a texture pattern by designating coordinates of the 2D pattern at each vertex (texture mapping). Alpha blending can be applied either per entire shape in single color mode or per pixel in tiling/texture mapping mode. When an object is drawn in single color or filled with a 2D pattern (without using Gouraud shading or texture mapping), dedicated primitives, such as Fast2DLine and Fast2DTriangle, are used. Only vertex coordinates are set for these primitives. Fast2Dtriangle is also used to draw polygons.

2D Primitives

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

Polygon draw

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

BLT/Rectangle fill

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

BLT Attributes

Attribute	Description
Raster operation	Selects two source logical operation mode

Pattern (Text) drawing

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

Pattern (Text) Drawing Attributes

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

Clipping

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

1.5.5 3D Drawing

3D Primitives

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

3D Primitives

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

3D Drawing attributes

MB86290Ahas various professional 3D graphics features, including Gouraud shading and texture mapping with bi-linear filtering/automatic perspective correction, and provides high- quality realistic 3D drawing. A built-in sophisticated texture mapping unit delivers fast pixel calculations. This unit also delivers color blending between the shading color and texture color as well as alpha blending per pixel.

Hidden surface management

MB86290A supports the Z buffer for hidden surface management.

1.5.6 Special Effects

Anti-aliasing

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

Line drawing

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

Line Draw Attributes

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

Alpha blending

Alpha blending blends two separate colors to provide a transparency effect. MB86290A supports two types of alpha blending; blending two different colors at drawing, and blending overlay planes at display. Transparent color is not used for these blending options.

Alpha Blending

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

Shading

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

Texture mapping

MB86290A supports texture mapping to map a style pattern onto the surface of 3D polygons. Perspective correction is calculated automatically. For 2D pattern texture mapping, MB86290A has a built-in buffer memory for a field of up to 64 × 64 pixels. Texture mapping is performed at high speeds while texture patterns are stored in this buffer. The texture pattern can also be stored in the graphics memory. In this case, a large pattern of up to 256 × 256 pixels can be used.

Texture Mapping

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

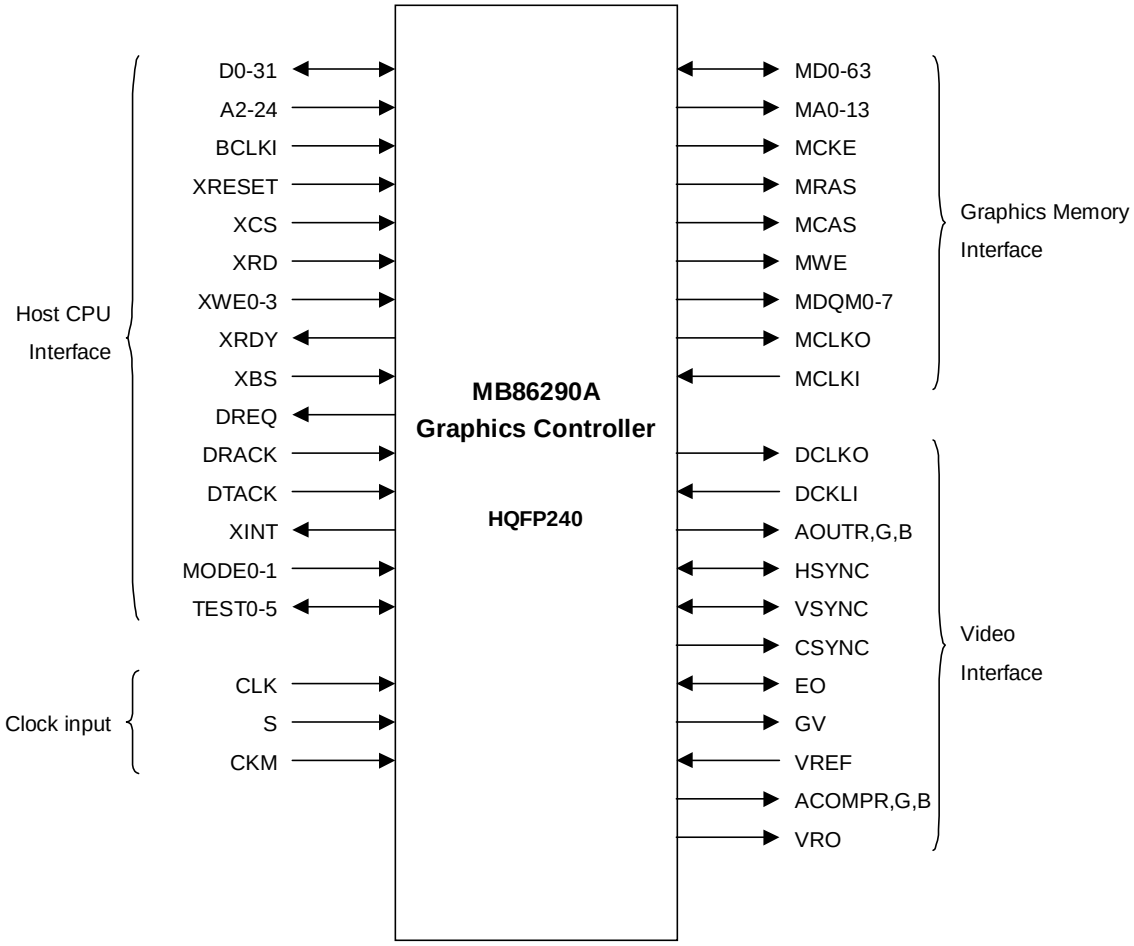
1.5.7 Display List

MB86290A is operated by feeding display lists which consists of a set of display commands, arguments and pattern data for them. Normally, these display lists are stored either in off- screen frame memory (part of MB86290A's local buffer) or host (main) memory that the DMAC of the host CPU can access directly. MB86290A reads these display lists, decodes the commands, and executes them after reading all the necessary arguments. By executing this operation set until the end of the display list, all graphics operations, including image/object drawing and display control, are separated from the CPU. Of course, the CPU program can also feed the display list information directly to MB86290A's designated registers.

2 Signal Pins

2.1 Signals

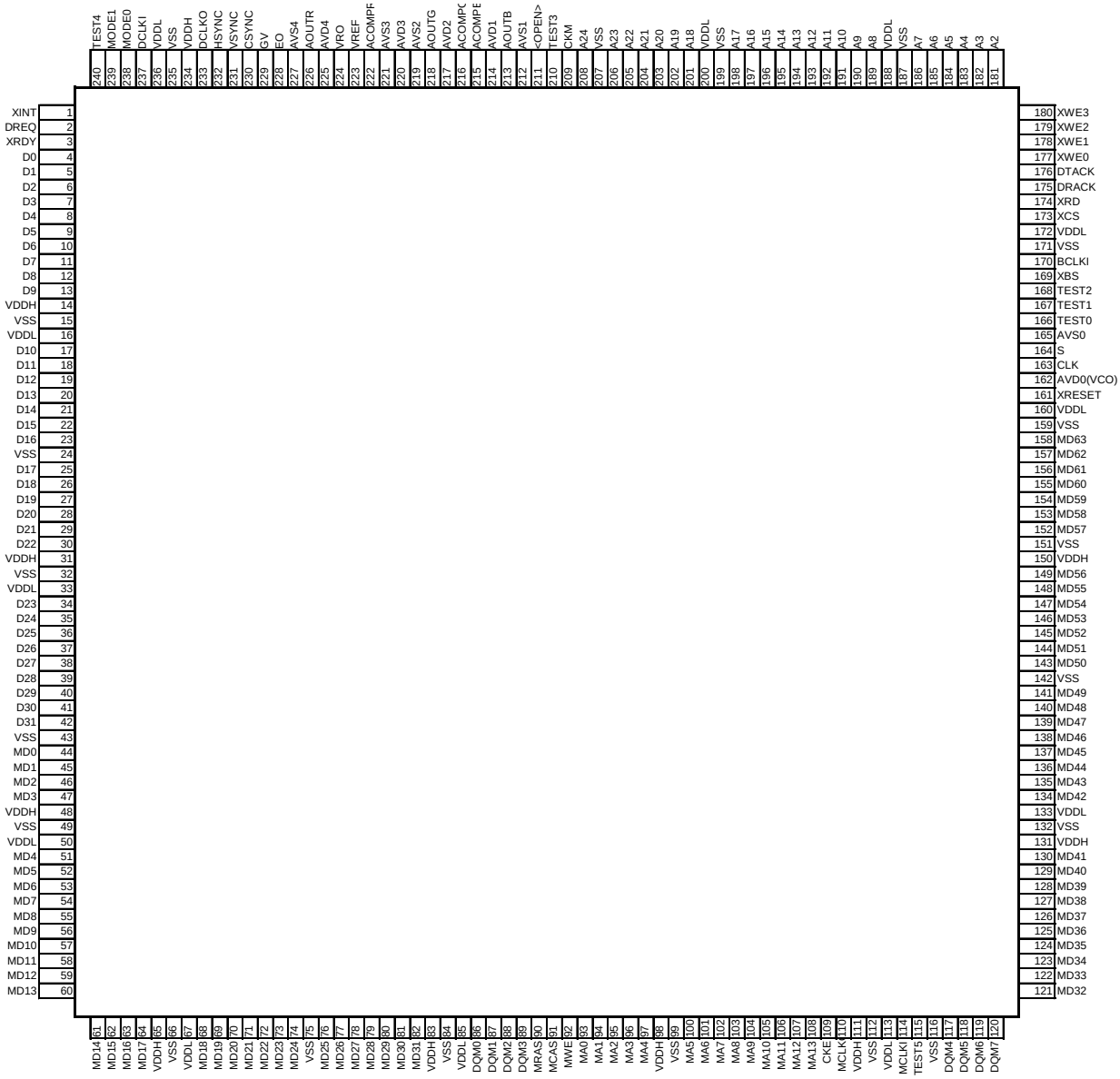
2.1.1 Signals



MB86290A Signals

2.2 Pin Assignment

2.2.1 Pin Assignment Diagram



MB86290A Pin Assignment

2.2.2 Pin Assignment Table

No.	Pin Name	No.	Pin Name	No.	Pin Name	No.	Pin Name
1	XINT	61	MD14	121	MD32	181	A2
2	DREQ	62	MD15	122	MD33	182	A3
3	XRDY	63	MD16	123	MD34	183	A4
4	D0	64	MD17	124	MD35	184	A5
5	D1	65	VDDH	125	MD36	185	A6
6	D2	66	VSS	126	MD37	186	A7
7	D3	67	VDDL	127	MD38	187	VSS
8	D4	68	MD18	128	MD39	188	VDDL
9	D5	69	MD19	129	MD40	189	A8
10	D6	70	MD20	130	MD41	190	A9
11	D7	71	MD21	131	VDDH	191	A10
12	D8	72	MD22	132	VSS	192	A11
13	D9	73	MD23	133	VDDL	193	A12
14	VDDH	74	MD24	134	MD42	194	A13
15	VSS	75	VSS	135	MD43	195	A14
16	VDDL	76	MD25	136	MD44	196	A15
17	D10	77	MD26	137	MD45	197	A16
18	D11	78	MD27	138	MD46	198	A17
19	D12	79	MD28	139	MD47	199	VSS
20	D13	80	MD29	140	MD48	200	VDDL
21	D14	81	MD30	141	MD49	201	A18
22	D15	82	MD31	142	VSS	202	A19
23	D16	83	VDDH	143	MD50	203	A20
24	VSS	84	VSS	144	MD51	204	A21
25	D17	85	VDDL	145	MD52	205	A22
26	D18	86	DQM0	146	MD53	206	A23
27	D19	87	DQM1	147	MD54	207	VSS
28	D20	88	DQM2	148	MD55	208	A24
29	D21	89	DQM3	149	MD56	209	CKM
30	D22	90	MRAS	150	VDDH	210	TEST3
31	VDDH	91	MCAS	151	VSS	211	<OPEN>
32	VSS	92	MWE	152	MD57	212	ACOMPB
33	VDDL	93	MA0	153	MD58	213	AVD1
34	D23	94	MA1	154	MD59	214	AOUTB
35	D24	95	MA2	155	MD60	215	AVS1
36	D25	96	MA3	156	MD61	216	ACOMPG
37	D26	97	MA4	157	MD62	217	AVD2
38	D27	98	VDDH	158	MD63	218	AOUTG
39	D28	99	VSS	159	VSS	219	AVS2
40	D29	100	MA5	160	VDDL	220	AVD3
41	D30	101	MA6	161	XRESET	221	AVS3
42	D31	102	MA7	162	$AVD0_{(VCO)}$	222	AVS4
43	VSS	103	MA8	163	CLK	223	AOUTR
44	MD0	104	MA9	164	S	224	AVD4
45	MD1	105	MA10	165	AVS0	225	VRO
46	MD2	106	MA11	166	TEST0	226	VREF
47	MD3	107	MA12	167	TEST1	227	ACOMPR
48	VDDH	108	MA13	168	TEST2	228	EO
49	VSS	109	CKE	169	XBS	229	GV
50	VDDL	110	MCLKO	170	BCLKI	230	CSYNC
51	MD4	111	VDDH	171	VSS	231	VSYNC
52	MD5	112	VSS	172	VDDL	232	HSYNC
53	MD6	113	VDDL	173	XCS	233	DCLKO
54	MD7	114	MCLKI	174	XRD	234	VDDH
55	MD8	115	TEST5	175	DRACK	235	VSS
56	MD9	116	VSS	176	DTACK	236	VDDL
57	MD10	117	DQM4	177	XWE0	237	DCLKI
58	MD11	118	DQM5	178	XWE1	238	MODE0
59	MD12	119	DQM6	179	XWE2	239	MODE1
60	MD13	120	DQM7	180	XWE3	240	TEST4

VSS/AVS: Ground VDDH: 3.3-V power supply VDDL: 2.5-V power supply

AVD: 2.5-V Analog power supply AVD(VCO): 2.5-V PLL power supply

Note 1: Do not connect anything to pin 211 <OPEN>

Note 2: These power supply layers (AVD/AVD(VCO)/VDDL) are recommended to physically isolate each other on the PCB.

2.3 Signal Descriptions

2.3.1 Host CPU Interface

Host CPU Interface Signals

Signal Name	I/O	Description
MODE0-1	Input	Host CPU Mode selection
XRESET	Input	Hardware reset
D0-31	In/Out	Host CPU bus data
A2-A24	Input	Host CPU bus address (In the V832 mode, A[24] is connected to XMWR.)
BCLKI	Input	Host CPU bus clock
XBS	Input	Bus cycle start
XCS	Input	Chip select
XRD	Input	Read strobe
XWE0	Input	Write strobe for D0-D7
XWE1	Input	Write strobe for D8-D15
XWE2	Input	Write strobe for D16-D23
XWE3	Input	Write strobe for D24-D31
XRDY	Output Tri-state	Wait request signal (In the SH3 mode, when this signal is 0, it indicates the wait state; in the SH4 and V832 modes, when this signal is 1, it indicates the wait state.)
DREQ	Output	DMA request signal (This signal is low-active in both the SH mode and V832 mode.)
DRACK/DMAAK	Input	Acknowledge signal issued in response to DMA request (DMAAK is used in the V832 mode; this signal is high-active in both the SH mode and V832 mode.)
DTACK/XTC	Input	DMA transfer strobe signal (XTC is used in the V832 mode. In the SH mode, this signal is high-active; in the V832 mode, it is low-active.)
XINT	Output	Interrupt signal issued to host CPU (In the SH mode, this signal is low-active; in the V832 mode, it is high-active)
TEST0-5	Input	Test signals

- MB86290A can be connected to the Hitachi SH4 (SH7750), SH3 (SH7709/09A) and NEC V832. In the SRAM interface mode, MB86290A can be used with any other CPU as well. The host CPU is specified by the MODE pins.

MODE 1	MODE 2	CPU
L	L	SH3
L	H	SH4
H	L	V832
H	H	Reserved

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

MB86290A Signal Pins	V832 Signal Pins
A24	XMWR
DTACK	XTC
DRACK	DMAAK

2.3.2 Video Interface

Video Interface Signals

Signal Name	I/O	Description
DCLKO	Output	Dot clock signal for display
DCLKI	Input	Dot clock input for external synchronization
AOUTR	Analog output	Analog signal (R) output
AOUTG	Analog output	Analog signal (G) output
AOUTB	Analog output	Analog signal (B) output
HSYNC	I/O*1	Horizontal sync signal output Horizontal sync input in external sync mode
VSYNC	I/O*1	Vertical sync signal output Vertical sync input in external sync mode
CSYNC	Output	Composite sync signal output
EO	I/O*1	Even/odd field identification output <In the external synchronous mode>, this signal is input for even/odd field identification input.
GV	Output	Graphics/Video switch
VREF	Analog input	Reference voltage input
ACOMPR	Analog output	R Signal complement output
ACOMPG	Analog output	G Signal complement output
ACOMPB	Analog output	B Signal complement output
VRO	Analog output	Reference current output

*1: Tolerates 5-V input voltage level

- Contains 8-bit precision D/A converters and outputs analog RGB signals
- Uses CSYNC signal and external circuits to generate composite video signal
- Can output analog RGB signals synchronously to external video signal
- Can synchronize to either DCLKI signal input or internal dot clock
- HSYNC and VSYNC reset to output mode. These signals must be pulled up externally.
- AOUTR, AOUTG and AOUTB must be terminated at 75 Ω.
- 1.1 V is input to VREF. A bypass capacitor (with good high-frequency characteristics) must be inserted between VREF and AVS.
- ACOMPR, ACOMPG and ACOMPB are tied to analog VDD via 0.1-µF ceramic capacitors.
- VRO must be pulled down to analog ground by a 2.7-kΩ resistor.
- HSYNC, VSYNC and EO can tolerate input voltage levels of 5 V. However, NEVER input 5 V to these pins when power is not supplied to MB86290A. (See the maximum voltage specification in the electrical characteristics.)
- When producing a non-interlaced display in the external synchronous mode, input 0 to the EO pin by using a pull-down resistor, etc.
- The GV signal switches graphics and video at chroma key operation. When video I is selected, the L level is output.

2.3.3 Graphics Memory Interface

Graphics Memory Interface Signals

Signal Name	I/O	Description
MD0-63	In/Out	Graphics memory bus data
MA0-13	Output	Graphics memory bus address
CKE	Output	Clock enable
MRAS	Output	Row address strobe
MCAS	Output	Column address strobe
MWE	Output	Write enable
MDQM0-7	Output	Data mask
MCLKO	Output	Graphics memory clock output
MCLKI	Input	Graphics memory clock input

- This interface is used to transfer data from/to external memory. 64-Mbit SDRAM can be used without glue logic.
- The data bus width is set to either 64 or 32 bits. In the 32-bit mode, MD32-63 and MDQM4-7 must be kept open.
- MCLKI and MCLKO are tied to each other externally.

2.3.4 Clock Input

Clock Input Signals

Signal Name	I/O	Description
CLK	Input	Clock input signal
S	Input	PLL reset signal
CKM	Input	Clock mode signal

- Inputs source clock for generating internal operation clock and display dot clock. Normally, 4 Fsc(= 14.31818 MHz) is input. An internal PLL generates the internal operation clock of 100.22726 MHz and the display base clock of 200.45452 MHz.
- For the internal operation clock, use either the output clock of the internal PLL (x7 of input clock) or the bus clock input (BCLK1) from the host CPU. When the host CPU bus speed is 100 MHz, the BCLK1 input should be selected.

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

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

3 Host Interface

3.1 Operation Mode

3.1.1 Host CPU Mode

Select the host CPU by setting the MODE signals as follows:

CPU Type Setting

MODE1	MODE0	CPU
L	L	SH3
L	H	SH4
H	L	V832
H	H	Reserved

3.1.2 Endian

MB86290A operates in little-endian mode. All the register address descriptions in these specifications are byte address in little endian. When using a big-endian CPU, note that the byte or word addresses are different from these descriptions.

3.2 Access Mode

3.2.1 SRAM Interface

Data can be transferred to/from MB86290A using a typical SRAM access protocol. MB86290A internal registers, internal memory and external memory are all mapped to the physical address field of the host CPU. The host CPU can access any of them like a normal memory device. Since MB86290A uses a hardware wait using the XRDY signal output, the respective hardware wait option of the host CPU must be enabled.

CPU Read

The host CPU reads data from internal registers and memory of MB86290A in double-word (32 bit) units.

CPU Write

The host CPU writes data to internal registers and memory of MB86290A in byte units.

3.2.2 FIFO Interface

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

3.3 DMA Transfer

3.3.1 Data Transfer Unit

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

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

3.3.2 Address Mode

Dual address mode

DMA is performed at memory-to-memory transfer between host memory (source) and MB86290A internal registers, memory, or external memory (destination). Both the host memory address and destination address is used. In the SH4 mode, the 1 double-word transfer (32 bits) and 8 double-word transfer (32 bytes) can be used.

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

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

Single address mode (FIFO interface)

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

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

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

3.3.3 Bus Mode

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

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

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

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

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

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

3.3.4 DMA Transfer Request

- Single-address mode

  DMA is started when the MB86290A issues an external request to DMAC of the host processor.

  Set the transfer count in the transfer count register of the MB86290A and then issue DREQ.

  Fix the CPU destination address to the FIFO address.

- Dual-address mode

  DMA is started by two procedures: the MB86290A issues an external request to DMAC of the host processor, or the CPU itself is started (auto request mode, etc.). Set the transfer count in the transfer count register of MB86290A and then issue DREQ.

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

3.3.5 Ending DMA Transfer

- SH3/SH4

  When the MB86290A transfer count register is set to 0, DMA transfer ends and DREQ is negated.

- V832

  When the XTC signal from the CPU is low-asserted while the DMAAK signal to MB86290A is high-asserted, the end of DMA transfer is recognized and DREQ is negated.

The end of DMA transfer is detected in two ways: the DMA status register (DST) is polled, and an interrupt to end the drawing command (FD000000h) is added to the display list and the interrupt is detected.

3.4 Interrupt Request

MB86290A issues interrupt requests to the host CPU. The following events issue interrupt requests. An interrupt request caused by each of these events is enabled/disabled independently by IMR (Interrupt Mask Register).

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

3.5 Transfer of Local Display List

This is the mode in which the MB86290A internal bus is used to transfer the display list stored in the graphics memory to the FIFO interface.

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

How to transfer list: Store the display list in the local memory of the MB86290A, set the transfer source local address (LSA) and the transfer count (LCO), and then issue a request (LREQ). Whether or not the local display list is currently being transferred is checked using the local transfer status register (LSTA).

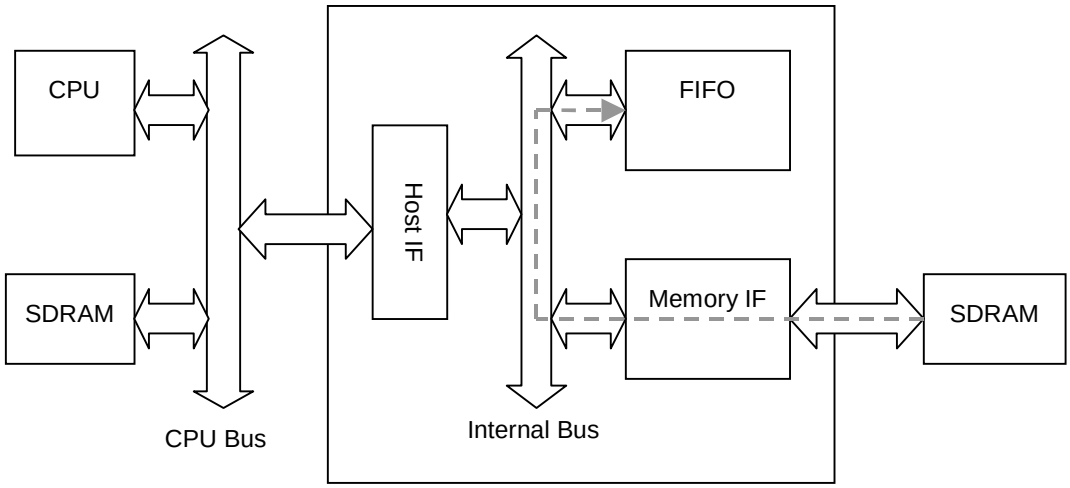


Fig. 3.1 Transfer Path for Local Display List

3.6 Memory Map

The following table shows the memory map of MB86290A to the host CPU address field. The physical address is mapped differently in each CPU type (SH3, SH4 or V832).

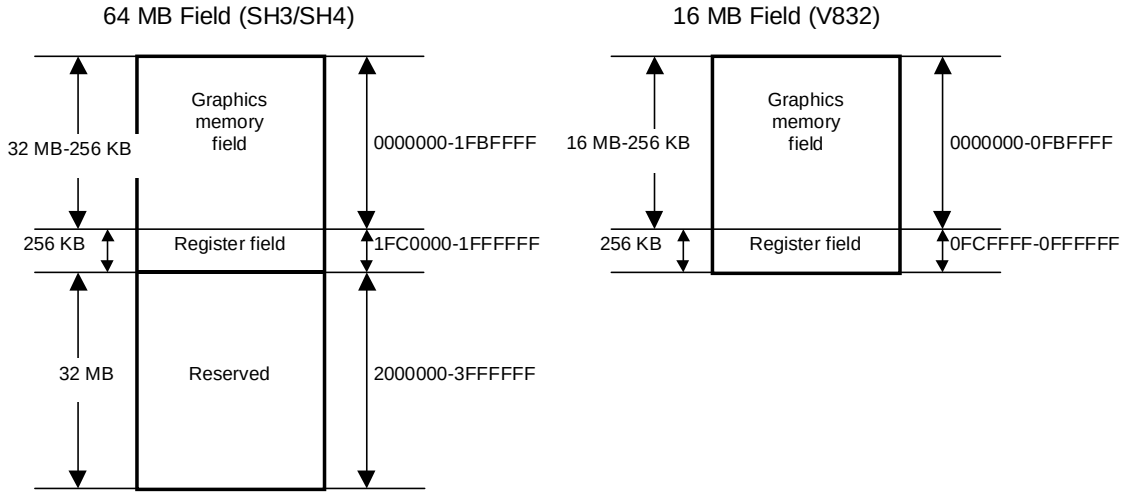


Fig. 3.2 Memory Map

Table 3-1 Address Mapping in SH3/SH4 Mode

Size	Resource	Base address	(Name)
32 MB to 256 KB	Graphics memory	00000000	
64 KB	Host interface registers	01FC0000	(HostBase)
64 KB	Display engine registers	01FD0000	(DisplayBase)
64 KB	Internal texture memory	01FE0000	(TextureBase)
64 KB	Drawing engine registers	01FF0000	(DrawBase)
32 KB	Reserved *	02000000	

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

Table 3-2 Address Mapping in V832 Mode

Size	Resource	Base address	(Name)
32 MB to 256KB	Graphics memory	00000000	
64 KB	Host interface registers	00FC0000	(HostBase)
64 KB	Display engine registers	00FD0000	(DisplayBase)
64 KB	Internal texture memory	00FE0000	(TextureBase)
64 KB	Drawing engine registers	00FF0000	(DrawBase)

4 Graphics Memory

4.1 Configuration

MB86290A uses local external memory (Graphics Memory) for drawing and display management. The configuration of this Graphics Memory is described as follows:

4.1.1 Data Type

MB86290A handles the following types of data. Display list can be stored in the host (main) memory as well. Texture-tiling pattern and text pattern can be defined by a display list as well.

Drawing frame

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

Display frame

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

Z buffer

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

Polygon draw flag buffer

This is a work field for random shape drawing of multiple vertices. 1 bit/pixel should be defined for the drawing shape. This flag buffer must be cleared prior to drawing.

Display list

This is a set of commands and parameters executed by MB86290A.

Texture pattern

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

Cursor pattern

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

4.1.2 Memory Layout

Each of these data items can be allocated anywhere in the Graphics Memory according to the respective register setting.

4.1.3 Memory Data Format

Direct color

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

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

Indirect color

The color index code is in 8 bits.

7	6	5	4	3	2	1	0
Color Code							

Z value

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

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

Polygon draw flag

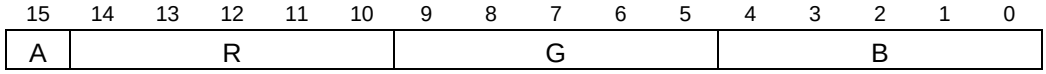
This is binary data describing each pixel in 1 bit.

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

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

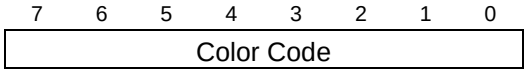
Texture/tiling pattern (direct color)

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



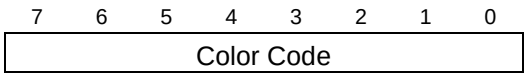
Tiling pattern (indirect color)

This is a color index code in 8 bits.



Cursor pattern

This is a color index code in 8 bits.



4.2 Frame Management

4.2.1 Single Buffer

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

4.2.2 Double Buffer

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

4.3 Memory Access

4.3.1 Memory Access by Host CPU

The Graphics Memory is mapped to the host CPU physical address field. The host CPU can access the Graphics Memory of MB86290A like a typical memory device.

4.3.2 Priority of Memory Access

The Graphics Memory accesses priority is as follows:

1. Refresh
2. Display
3. Host CPU Access
4. Drawing

5 Display Controller

5.1 Overview

Display control

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

Video timing generator

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

Color look-up

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

Cursor

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

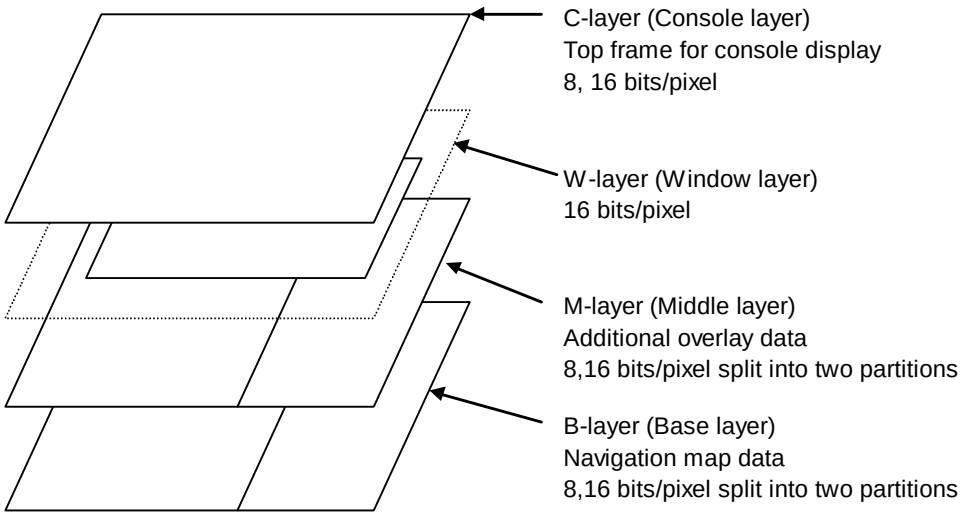
External synchronization control

Graphics display can be synchronized with the external video display timing.

5.2 Display Function

5.2.1 Layer Configuration

MB86290A supports four layers of display frames (C, W, M and B). Furthermore, the M and B layers can be split into two separate windows at any position (L frame and R frame). All these six frames are assigned as logically separated fields in the Graphics Memory.



Configuration of Display Layers

When the resolution exceeding the VGA (640 x 480) is required, the layer count or pixel data which can be simultaneously displayed is restricted according to the capability of frame memory for supplying data.

5.2.2 Overlay

Simple priority mode

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

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

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

Blend mode

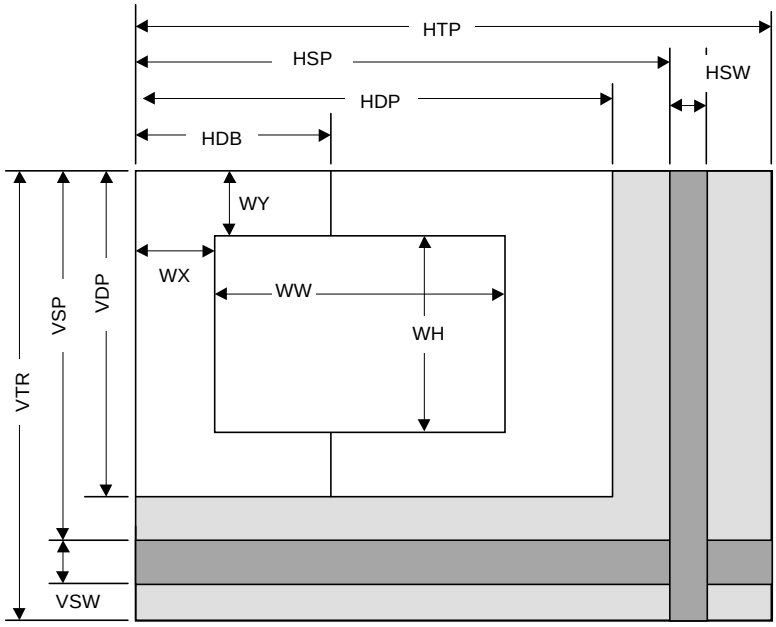
The W, M and B layers are managed in the same way as the simple priority mode described above. The result of the W/M/B layer priority color is blended with the C-layer color according to the blending ratio specified in the register. This mode is applied when the alpha bit of that pixel in the C layer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

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

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

5.2.3 Display Parameters

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



Display Parameters

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

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

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

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

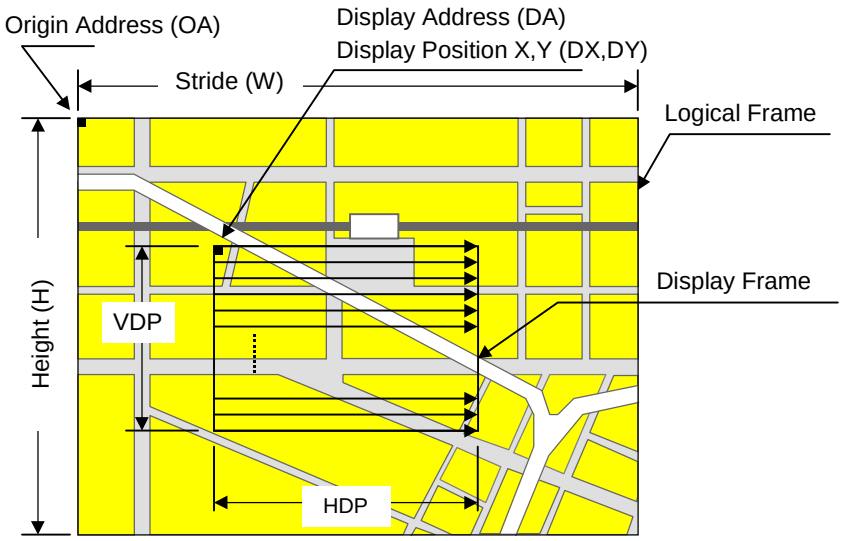
$HDP - HDB > 4$ (in direct color mode), 8 (in indirect color mode)

5.2.4 Display Position Control

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

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

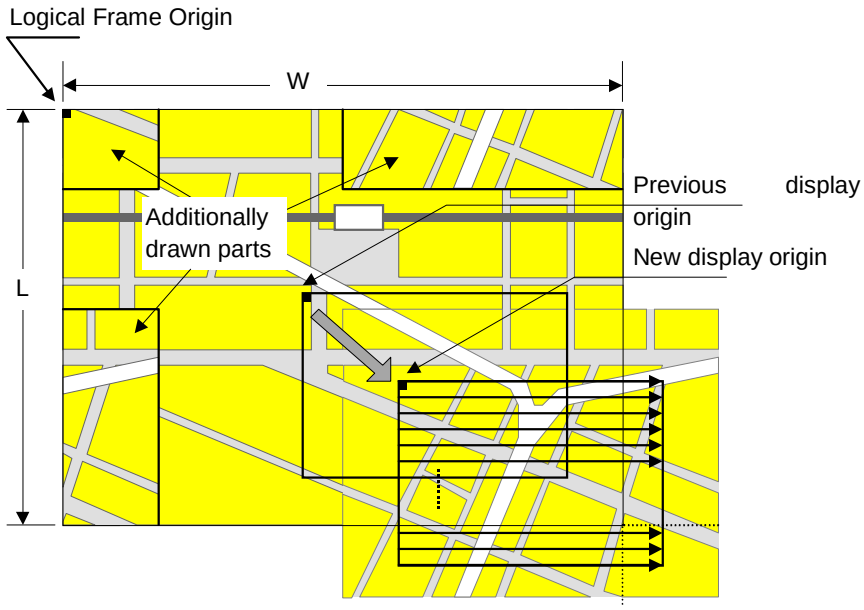
The correlation between the logical graphics field and physical display position is defined as follows:



Display Position Parameters

OA	Origin Address	Base address of logical graphics field. Memory address of top left edge pixel in logical graphics field
W	Stride	Width of logical graphics field. Defined in 64-byte boundary
H	Height	Height of logical graphics field. Total raster (pixel) count of field
DA	Display Address	Display base address. Top left position address of display frame
DX DY	Display Position	Display base 2D coordinate

MB86290A scans the logical graphics field as if the entire field is rolled over in both the horizontal and vertical directions. By using this function, if the display frame crosses the border of the logical graphics field, the part outside the border is covered with the other side of the logical graphics field, which is assumed to be connected cyclically as shown below:



Wrap Around Management of Display Frame

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

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

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

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

$$0 \leq DY < H$$

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

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

5.3 Display Color

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

5.3.1 Color Look-up Table

MB86290A has two color look-up tables (pallets) for the indirect color mode. Each pallet has 256 entries. A color data item contains 18 bits of data (RGB 6 bit, respectively), which is correlated to each color code specified in 8-bit data. Therefore, each pallet can show 256 colors at one time out of 262,144 color selections.

C-layer palette

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

M/B-layer palette

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

5.3.2 Chroma-key Operation

MB86290A performs superimpose using the chroma-key function. When the key color of this chroma-key operation matches the color of the C layer during the display scan period, the GV signal output becomes L level. The graphics signal output from MB86290A and the external video signal can be switched by using this signal.

5.4 Cursor

5.4.1 Cursor Display Function

MB86290A can display two hardware cursors simultaneously. Each cursor is specified as 64 x 64 pixels, and the style pattern is set in the Graphics Memory. Only the indirect color mode (8 bits/pixel) can be used and the C-layer pallet is used for the color look-up. However, transparent color management (transparent color code setting and management of code 0) is different from ordinary C-layer pixels−alpha blending cannot be used for the cursor color and the alpha bit in the color data registered to the color palette is ignored.

5.4.2 Cursor Management

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

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

5.5 Processing Flow for Display Data

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

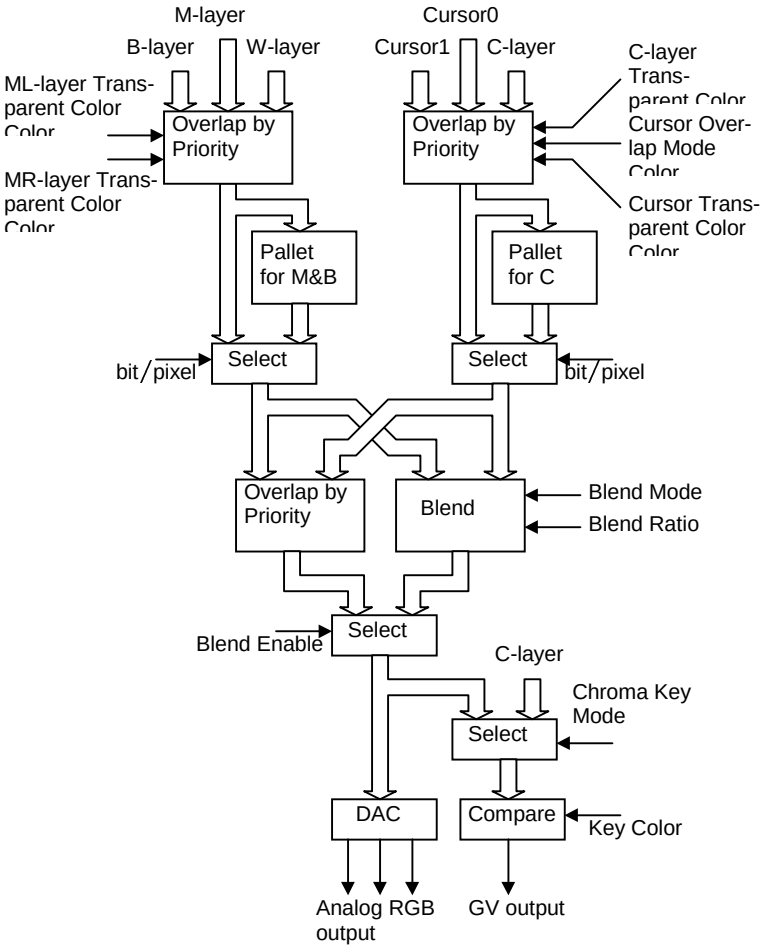


Fig. 5.1 Display data processing flow

ML-layer Transparent Color

Specifies transparent color code for left side of M layer

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

ML-layer Transparent Color

Specifies transparent color code for right side of M layer

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

C-layer Transparent Color

Specifies transparent color code for C layer

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

Cursor Transparent Color

Specifies transparent color code for cursor

Cursor Priority Mode

Specifies whether or not to display cursor above C layer

Blend Mode

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

Blend ratio

Specifies blend ratio with accuracy of 1/16

Blend Enable

Specifies whether or not to use Blend

Chroma Key Mode

Selects display data used to compare chroma keys

The data for the C-layer or final tier can be selected.

Key Color

Sets color code compared with display data

When display data matches the color code, 0 is output to the GV pin.

5.6 Synchronization Control

5.6.1 Applicable Display Resolution

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

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

Pixel frequency = 14.31818 MHz × 14 x reference clock division rate (when internal PLL selected)

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

Horizontal frequency = Pixel frequency / Horizontal total pixel count

Vertical frequency = Horizontal frequency / Vertical total raster count

5.6.2 Interlace Display

The MB86290A can generate both a non-interlace display and an interlace display. For the interlace display, the 1st, 3rd, ... (2n+1)th rasters of the display screen are output to odd fields, and 2nd, 4th, ... 2n-th rasters of the display screen are output to even fields.

5.6.3 External Synchronization

Display scan can also be synchronized to external HSYNC/VSYNC signals. When the external synchronization mode is set at the register, MB86290A starts sampling the HSYNC signal input and displays the graphics output synchronized to the external video signals. Either the internal display base clock or DCLKI input can be used for this sampling clock. Also, by using the chroma-key function, superimpose is performed with external circuitry as follows:

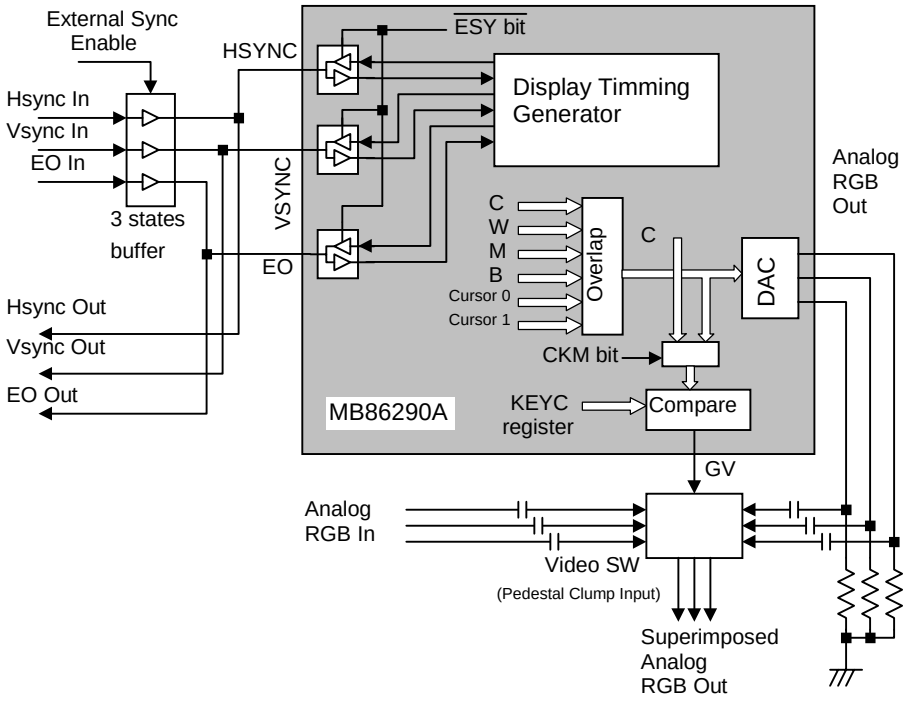
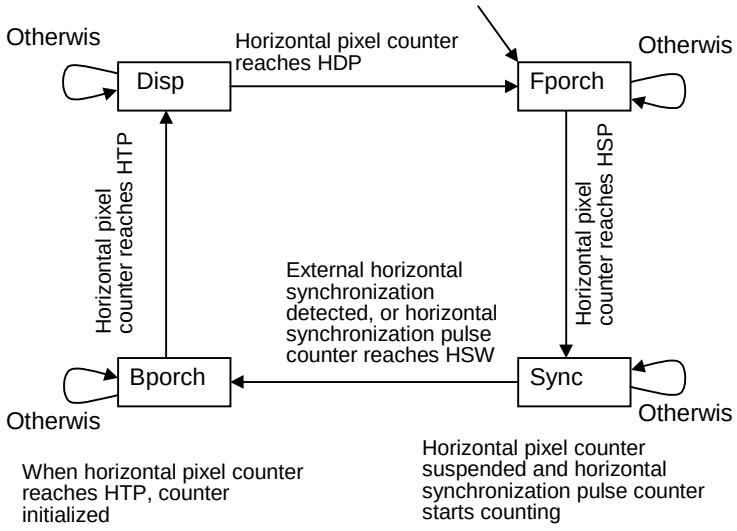


Fig. 5.2 Example of External Synchronization Circuit

The external synchronous mode is set using the ESY bit of the DCM register. When the external synchronous mode is set, the HSYNC, VSYNC, and EO pins of the MB86290A are placed in the input mode. After this, supply external sync signals by using the tristate buffer. Also, when exiting from the external synchronous mode, cut the external synchronous input and then set the internal ESY bit of the MB86290A to OFF.

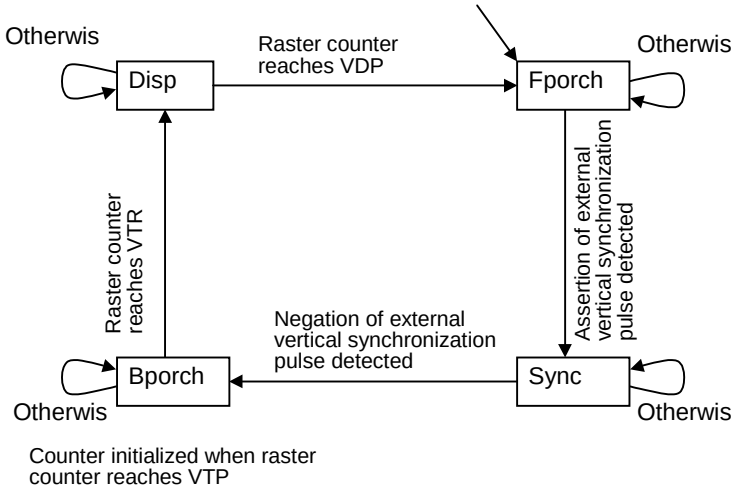
With the MB86290A sync signal output set to ON, avoid setting the buffer for external sync signals to ON. Use the above procedure to control so that the concurrent-ON duration will not occur.

Horizontal synchronization is controlled by the following state transitions:



State transitions are controlled mainly using the count values of the horizontal pixel counter. The display duration is equivalent to the Disp state. When the value of the horizontal pixel counter reaches the setting of the HDP register, the display duration ends, causing a transition from the Disp state to the Fporch state (front porch). With the Fporch state established, when the value of the horizontal pixel counter reaches the setting of the HSP register, a transition is made to the Sync state. In the Sync state, external horizontal synchronization signals are supplied. The MB86290A detects the negation edge of the external horizontal synchronization pulse to perform synchronization. When the external horizontal synchronization signal is detected, a transition is made to the Bporch state (back porch). In the Sync state, the horizontal pixel counter stops, but the horizontal synchronization pulse counter starts incrementing from 0. When the value of the counter reaches the setting of the HSW register, a transition is made to the Bporch state without detecting the external horizontal synchronization signal. With the Bporch state established, when the value of the horizontal pixel counter reaches the setting of the HTP register, the horizontal pixel counter is reset and a transition is made to the Disp state, starting display of the next raster.

The vertical synchronization is controlled by the following state transitions:



State transitions are mainly controlled using the count values of the raster counter. The display duration is equivalent to the Disp state. When the value of the raster counter reaches the setting of the VDP register, the display duration ends, causing a transition from the Disp state to the Fporch state (front porch). In the Fporch state, the processing waits for the external vertical synchronization pulse to be asserted. When assertion of the external vertical synchronization pulse is detected, a transition is made to the Sync state. In the Sync state, the processing waits for the external vertical synchronization signal to be negated. When the negation is detected, a transition is made to the Bporch state (back porch). With the Bporch state established, when the value of the raster counter reaches the setting of the VTR register, the raster counter is reset and a transition is made to the Disp state, starting display of the next field.

5.7 Video Interface

5.7.1 NTSC Output

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

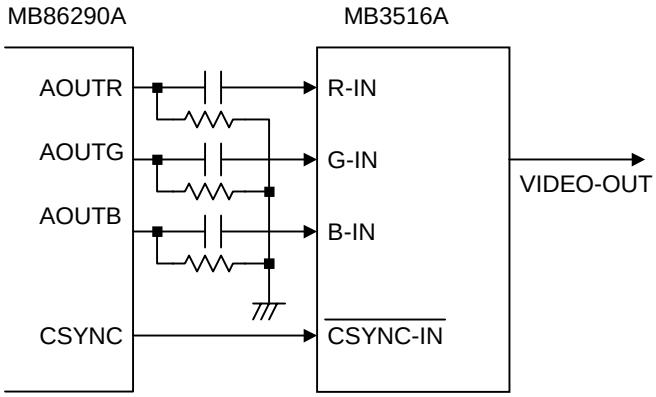


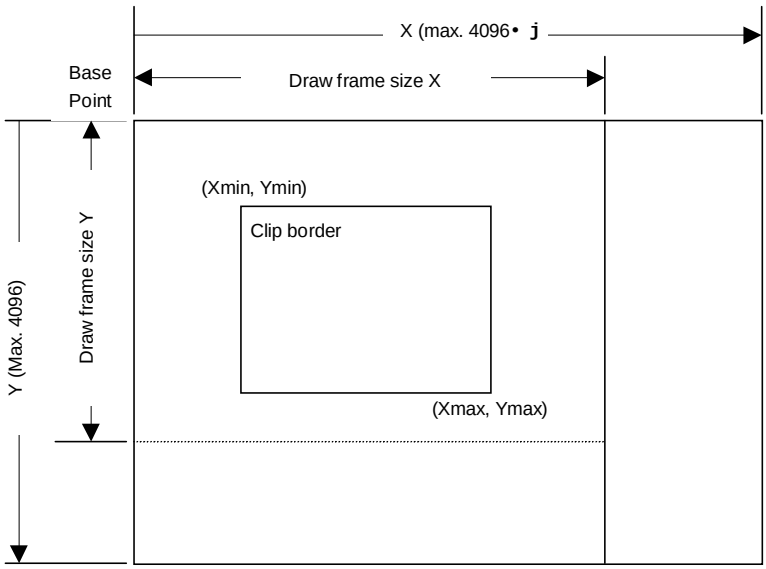
Fig. 5.3 Example of NTSC Encoder Connection

6 Drawing Control

6.1 Coordinates

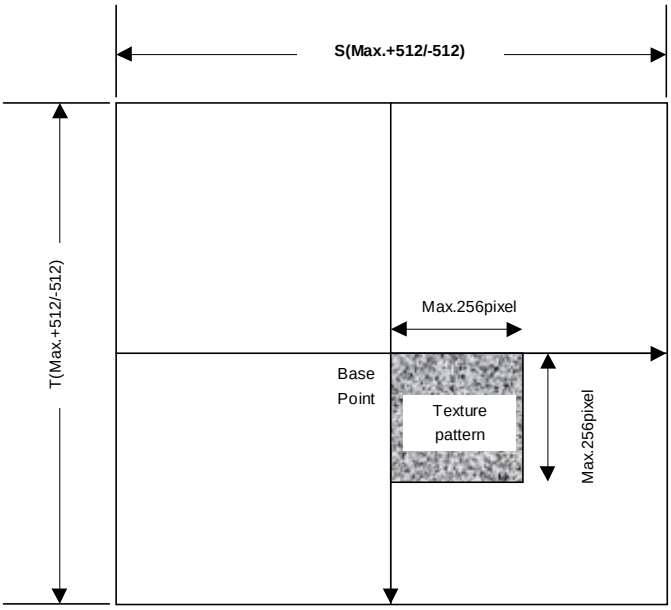
6.1.1 Drawing Coordinate

MB86290A manages a drawing frame as a 2D coordinate with the origin at the top left edge. The maximum coordinate is 4096 x 4096. Each drawing frame is located in the Graphics Memory by setting the address of the origin and width (pixel size of X span). Although the maximum size of Y span does not need to be specified, take care about the memory size allocation so as not to overlap any other frames. Also, setting the clip field (top left and bottom right coordinates in registers) prevents drawing of all images outside the border of the clip window.



6.1.2 Texture Coordinate

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



6.1.3 Frame Buffer

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

Drawing frame

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

Z buffer

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

Polygon draw flag buffer

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

6.2 Polygon Drawing

6.2.1 Drawing Primitives

MB86290A supports the following primitive types:

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

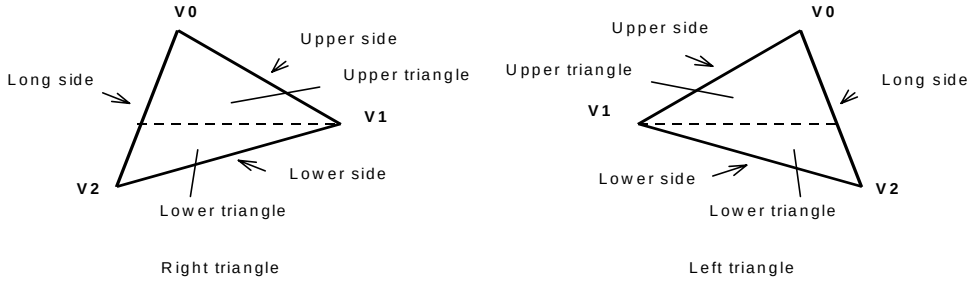
6.2.2 Polygon Drawing

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

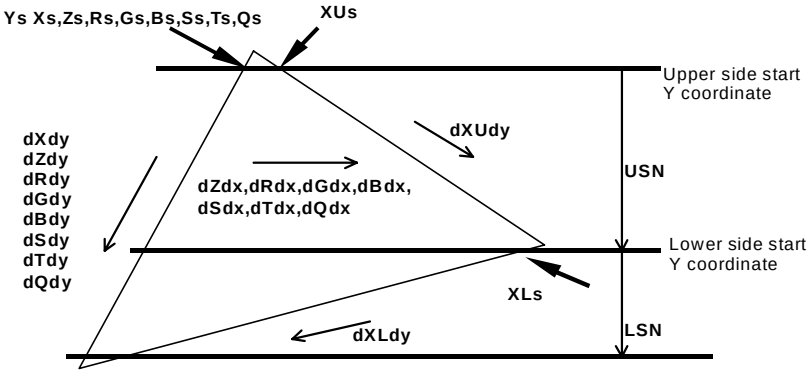
1. Execute PolygonBegin command
   Initialize polygon draw enginew
2. Draw vertices.
   Draw outline of polygon and plot all vertices to polygon draw flag buffer utilizing Fast2Dtriangle primitive.
3. Execute PolygonEnd command.
   Copy shape in polygon draw flag buffer to drawing frame and fill shape with color or specified tiling pattern.

6.2.3 Drawing Parameters

MB86290A differentiates triangles (Right triangle and Left triangle) according to the locations of three vertices as follows (not used for Fast2Dtriangle):



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



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

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

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

6.2.4 Anti-aliasing Function

MB86290A performs anti-aliasing to eliminate jaggies on line edges and make lines appear smooth. To use this function at the edges of primitives, redraw the primitive edges with anti-alias lines.

6.3 Bit Map Operation

6.3.1 BLT

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

(1) From host CPU to Drawing frame memory

(2) From Graphics Memory (other than Drawing frame memory area) to drawing memory

(3) From host CPU to internal texture memory

(4) From Graphics Memory to internal texture memory

When Drawing frame memory is designated as the destination, the result of logical calculation between the source and current value in the designated destination can be stored as well. If part of the source and destination of the BLT field are physically overlapped in the display frame, the start address (from which vertex the BLT field to be transferred) must be set carefully.

Usage caution: When transferring a rectangle from one graphics memory to another graphics memory (drawing frames included), or from the host CPU to the internal texture memory, the width of the rectangle must be at least 5 pixels (in direct color mode) or 9 pixels (in indirect color mode).

6.3.2 Pattern Data Format

MB86290A can handle three bit map data formats: indirect color mode (8 bits/pixel), direct color mode (16 bits/pixel), and binary bit map (1 bit/pixel). The direct color mode is used for texture patterns. Either the indirect or direct color mode is used for tiling patterns. The binary bit map is used for character/font patterns, where foreground color is used for bitmap = 1 pixel, and background color is applied for bitmap = 0 pixels.

6.4 Texture Mapping

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

6.4.1 Texture Size

MB86290A reads texcel data from the specified texture coordinate (S, T) position, and pastes that data at the correlated pixel position of the polygon. The applicable texture data size is 16, 32, 64, 128 or 256 pixels per S and T, respectively. Texture mapping is used only when the direct color mode (16bit/pixel) is used.

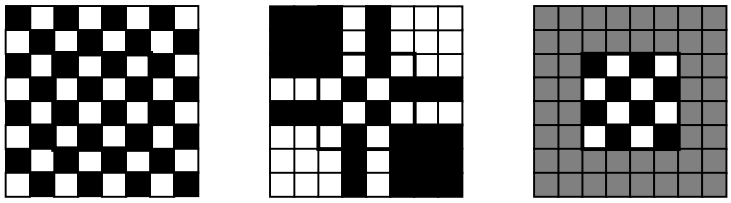
6.4.2 Texture Memory

Texture pattern data is stored in either the MB86290A internal texture buffer or external Graphics Memory. The internal texture buffer size is 8 Kbyte and can hold up to 64 × 64 pixels of texture. If the texture pattern size is smaller than 64 × 64pixels, it is best to store it in the internal texture buffer because the texture mapping speed is faster.

6.4.3 Texture Lapping

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

Repeat



Repeat Cramp Border

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

Cramp

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

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

Border

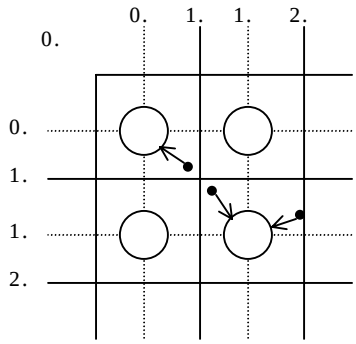
When the applied (S, T) coordinate is either negative or larger than the specified texture pattern size, the outside of the specified texture pattern is rendered in the 'border' color.

6.4.4 Filtering

MB86290A supports two texture filtering modes: point filtering, and bi-linear filtering.

Point filtering

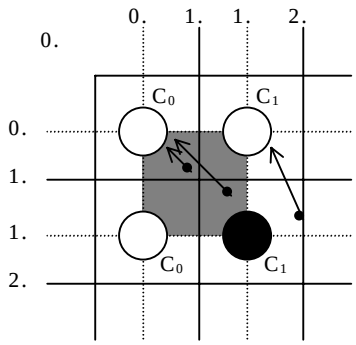
This mode uses the texcel data specified by the (S, T) coordinate. The nearest texcel in the texture pattern is chosen according to the calculated (S, T) coordinate.



Bi-linear filtering

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

Note: This mode can be used when the internal memory is specified as the texture memory mode.



6.4.5 Perspective Correction

This function adjusts the depth distortion of the 3D projection in the texture mapping process. For this adjustment, the 'Q' element of the texture coordinate (Q = 1/W) is defined from the 3D coordinate of the correlated vertex. This Q value is used after normalizing in the range between 0.0 and 1.0.

6.4.6 Texture Blending

MB86290A supports the following three texture blending modes:

Decal

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

Modulate

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

$$C_0 = C_R \times C_P$$

Stencil

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

MSB = 1: Texcel color

MSB = 0: Polygon color

6.5 Rendering

6.5.1 Tiling

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

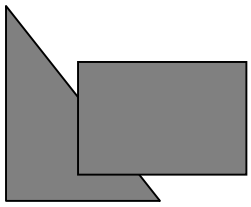


Fig. 6.4 Example of Tiling Operation

6.5.2 Alpha Blending

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

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

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

Normal

Blends post texture mapping color with current frame buffer color

Stencil

Uses MSB of texcel color to select display color:

MSB = 1: Texcel color

MSB = 0: Current frame buffer color

Stencil alpha

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

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

MSB = 0: Current frame buffer color

6.5.3 Logical Calculation

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

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

6.5.4 Hidden Surface Management

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

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

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

6.6 Drawing Attributes

6.6.1 Line Draw Attributes

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

Line Draw Attributes

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

6.6.2 Triangle Draw Attributes

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

Triangle Draw Attributes

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

6.6.3 Texture Attributes

The following attributes apply for texture mapping:

Texture Attributes

Attribute	Description
Texture mode	Select either texture mapping or tiling
Texture memory mode	Select either internal texture buffer or external Graphics Memory to use in texture mapping
Texture filter	Select either point sampling or bi-linear filtering The bilinear filter can be specified when the internal memory is specified as the texture memory mode.
Texture coordinate correction	Select either linear or perspective correction
Texture wrap	Select either repeat or cramp of texture pattern
Texture blend mode	Select either decal or modulate

6.6.4 Character/Font Drawing and BLT Attributes

When character/font pattern draw and BLT draw are performed, the following attributes apply:

Character/Font Pattern and BLT Attributes

Attribute	Description
Character pattern enlarge/shrink	2×2, $\times 2$ horizontal, $1/2 \times 1/2$, $\times 1/2$ horizontal
Character pattern color	Set character color and background color
Logical calculation mode	Specify two source logical calculation mode in BLT operation

6.7 Display List

6.7.1 Overview

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

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

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

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

Display List

6.7.2 Header Format

Format Overview

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

Description of Each Field

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

Vertex Number Specified in Vertex Code

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

6.7.3 Display List Command Overview

The following table lists the MB86290A display list commands.

Type	Draw Command	Description
Nop	-	No operation
Interrupt	-	Interrupt request to host CPU
Sync	-	Synchronization of events
SetRegister	-	Data set to register
SetVertex2i	Normal	Data set to Fast2DTriangle VRTX register
	PolygonBegin	Initialization of border rectangle calculation of multiple vertices random shape
Draw	PolygonEnd	Polygon flag clear (post random shape drawing operation)
	Flush_FB/Z	Flushes drawing pipelines
	All draw commands	Issue draw command
DrawPixel	Pixel	Plot Point
DrawPixelZ	PixelZ	Plot Point with Z value
DrawLine	Xvector	Draw Line (*1)
	Yvector	Draw Line (*2)
	AntiXvector	Draw Line with anti-alias option (*1)
	AntiYvector	Draw Line with anti-alias option (*2)
DrawLine2I DrawLine2iP	ZeroVector	Draw Fast2DLine (start from vertex 0)
	OneVector	Draw Fast2DLine (start from vertex1)
DrawTrap	TrapRight	Draw Right Triangle
	TrapLeft	Draw Left Triangle
DrawVertex2i	TriangleFan	Draw Fast2DTriangle
DrawVertex2iP	FlagTriangleFan	Draw Fast2DTriangle for multiple vertices random shape
DrawRect	BltFill	Fill rectangle with one color or tiling pattern
DrawRectP	ClearPolyFlag	Clear Polygon flag buffer
DrawBitmap	BltDraw	Draw rectangle pattern
DrawBitmapP	Bitmap	Draw binary bit map pattern (character)
BltCopy BltCopyP	TopLeft	BitBlt transfer from left upper vertex
	TopRight	BitBlt transfer from right upper vertex
BltCopy Alternate	BottomLeft	BitBlt transfer from left lower vertex
BltCopy AlternateP	BottomRight	BitBlt transfer from right lower vertex
LoadTexture	LoadTexture	Load texture pattern
	LoadTILE	Load tile pattern
BltTexture	LoadTexture	Load texture pattern from Graphics Memory
	LoadTILE	Load tile pattern from Graphics Memory

Note (*1) $-Pai/4 \leq$ Line angle $\leq Pai/4$

(*2) $-Pai/2 \leq$ Line angle $\leq -Pai/4$, or $Pai/4 \leq$ Line angle $\leq Pai/2$

Type Field Code Table

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

Draw Command Code Table (1)

DrawCommand	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

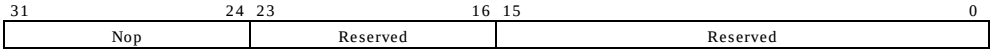
Draw Command Code Table (2)

DrawCommand	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

6.7.4 Details of Display List Commands

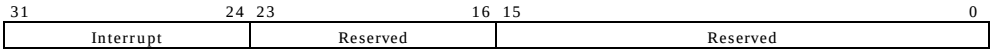
All parameters belonging to their command are set in correlated registers. The definition of each parameter is figured out in the section of each command description.

Nop (Format1)



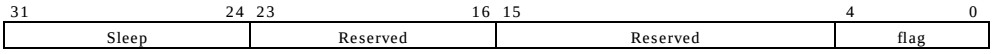
No operation

Interrupt (Format1)



Generates interrupt request to host CPU

Sync (Format9)



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

Flag:

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

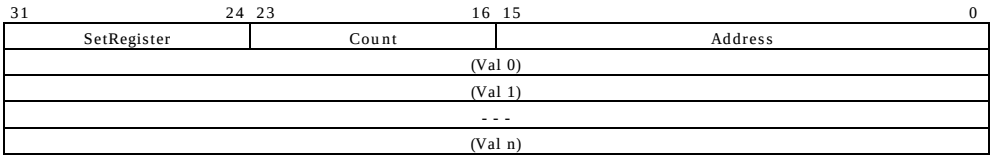
Bit0 VBLANK

VBLANK Synchronization

0 No operation

1 Wait for VSYNC detection

SetRegister (Format2)



Sets data at consecutive registers

Count:

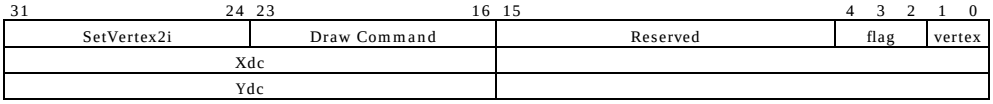
Data word count (in double-word unit)

Address:

Register address

Set the register address as the byte address/4 (address in double-word units).

SetVertex2i (Format8)



Sets vertices data for Fast2DLine or Fast2DTriangle command at registers

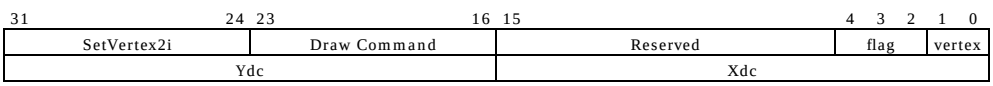
Commands:

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

Flag:

Not used

SetVertex2iP (Format8)



Sets vertices data for Fast2DLine or Fast2DTriangle command to registers

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

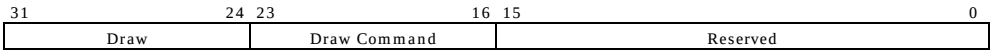
Command:

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

Flag:

Not used

Draw (Format5)



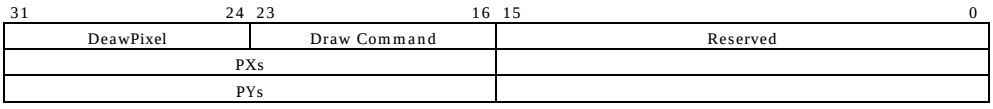
Executes draw command

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

Commands:

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

DrawPixel (Format5)

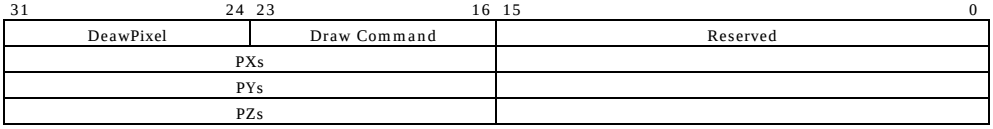


Plots pixel

Command:

Pixel	Plot pixel (without Z value).

DrawPixelZ (Format5)

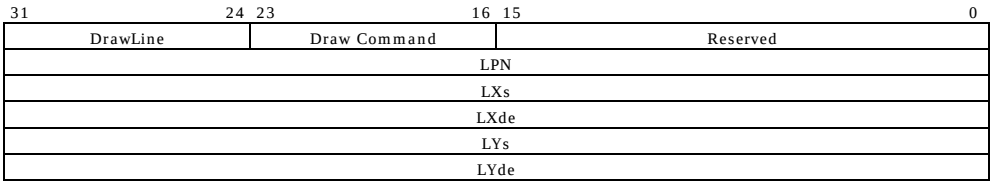


Plots 3D pixel

Command:

PixelZ	Plot pixel (with Z value).

DrawLine (Format5)



Draws line

Start drawing after setting all parameters at line draw registers.

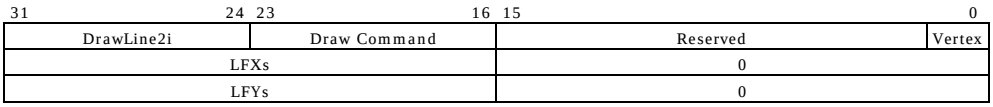
Commands:

Xvector	Draw line (*1).
Yvector	Draw line (*2).
XvectorNoEnd	Draw line without end point (*1).
YvectorNoEnd	Draw line without end point (*2).
XvectorBlpClear	Draw line (*1). Prior to drawing, clear reference position of broken line pattern.
YvectorBlpClear	Draw a line (*2) Prior to drawing, clear reference position of broken line pattern.
XvectorNoEndBlpClear	Draw line without end point (*1). Prior to drawing, clear reference position of broken line pattern.
YvectorNoEndBlpClear	Draw line without end point (*2). Prior to drawing, clear reference position of broken line pattern.
AntiXvector	Draw anti-alias line (*1).
AntiYvector	Draw anti-alias line (*2).
AntiXvectorNoEnd	Draw anti-alias line without end point (*1).
AntiYvectorNoEnd	Draw anti-alias line without end point (*2).
AntiXvectorBlpClear	Draw anti-alias line (*1). Prior to drawing, clear reference position of broken line pattern.
AntiYvectorBlpClear	Draw anti-alias line (*2). Prior to drawing, clear reference position of broken line pattern.
AntiXvectorNoEndBlpClear	Draw anti-alias line without end point (*1). Prior to drawing, clear reference position of broken line pattern.
AntiYvectorNoEndBlpClear	Draw anti-alias line without end point (*2). Prior to drawing, clear reference position of broken line pattern

Note (*1) -Pai/4 ≤ Line angle ≤ Pai/4

(*2) -Pai/2 ≤ Line angle ≤ -Pai/4, orPai/4 ≤ Line angle ≤ Pai/2

DrawLine2i (Format7)



31 – 24	23 – 16	15 – 0	
DrawLine2i	Draw Command	Reserved	Vertex
LFXs		0	
LFYs		0	

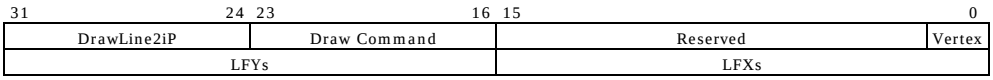
Draws Fast2Dline

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

Commands:

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

DrawLine2iP (Format7)



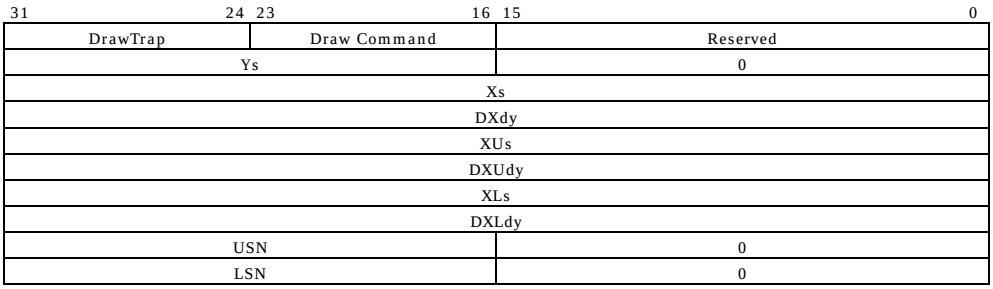
Draws Fast2Dline

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

Commands:

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

DrawTrap (Format5)



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

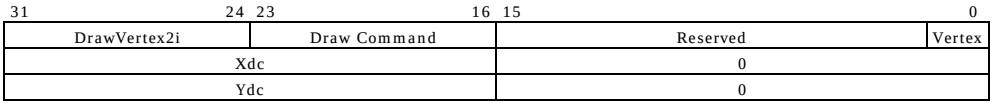
Draws Triangle

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

Commands:

TrapRight	Draw Right Triangle.
TrapLeft	Draw Left Triangle.

DrawVertex2i (Format7)



31 – 24	23 – 16	15 – 1	0
DrawVertex2i	Draw Command	Reserved	Vertex
Xdc		0	
Ydc		0	

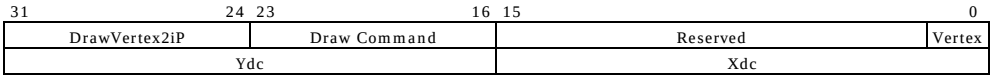
Draws Fast2Dtriangle

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

Commands:

TriangleFan	Draw Fast2Dtriangle.
FlagTriangleFan	Draw Fast2DTriangle for random shape with multiple vertices.

DrawVertex2iP (Format7)



31 – 24	23 – 16	15 – 1	0
DrawVertex2iP	Draw Command	Reserved	Vertex
Ydc		Xdc	

Draws Fast2Dtriangle

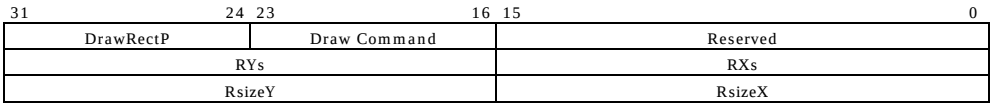
Operation is started after setting all the related parameters at Plane Draw registers

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

Commands:

TriangleFan	Draw Fast2Dtriangle.
FlagTriangleFan	Draw Fast2DTriangle for random shape with multiple vertices.

DrawRectP (Format5)



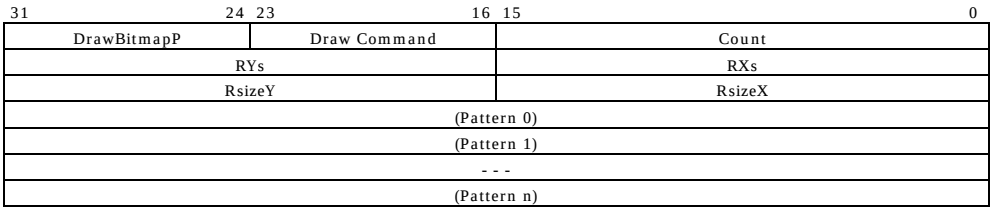
Fills rectangle

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

Commands:

BltFill	Fill rectangle with current color (single) or current tiling pattern.
ClearPolyFlag	Fill polygon flag field with 0. The size is defined in RsizeX,Y.

DrawBitmapP (Format6)

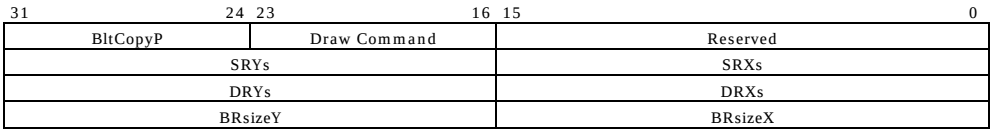


Draws rectangle

Commands:

BltDraw	Draw rectangle of 8 bits/pixel or 16 bits/pixel.
DrawBitmap	Draw binary bitmap character pattern. Bit0 is drawn in transparent or background color, and bit1 is drawn in foreground color. Background color is specified in the BC register, and foreground color is specified in the FC register.

BltCopyP (Format5)



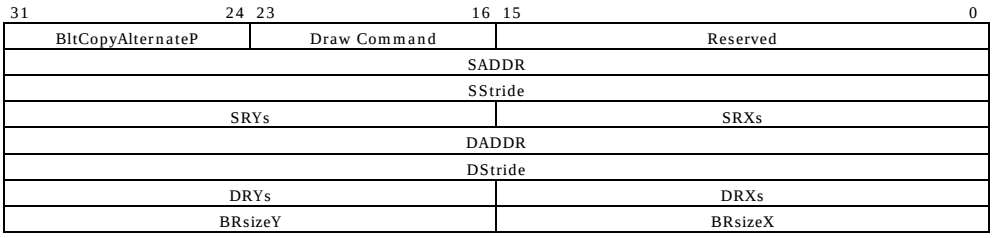
Copies rectangle pattern within one drawing frame

For BRsizeX, specify at least 5 pixels (when direct color mode used) or 9 pixels (when indirect color mode used).

Commands:

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

BltCopyAlternateP (Format5)



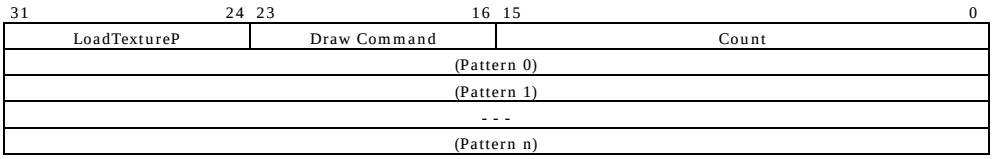
Copies rectangle between two separate drawing frames

For BRsizeX, specify at least 5 pixels (when direct color mode used) or 9 pixels (when indirect color mode used).

Commands:

TopLeft	Start BitBlt transfer from top left vertex.

LoadTextureP (Format6)



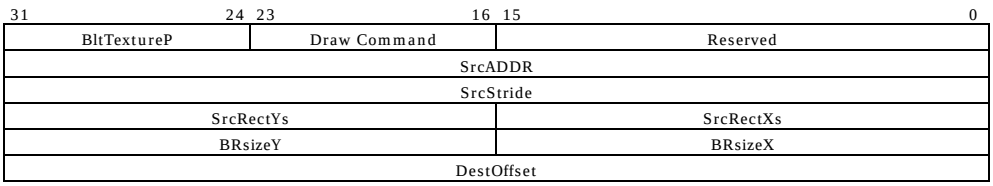
Loads texture or tile pattern into internal texture buffer memory

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

Commands:

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

BltTextureP (Format5)



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

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

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

For BRsizeX, specify at least 5 pixels (when direct color mode used) or 9 pixels (when indirect color mode used).

Commands:

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

7 Registers

7.1 Description

All the terms in this chapter are explained below:

(1) Register address

Indicates address of register

(2) Bit #

Indicates bit number

(3) Bit field name

Indicates name of each bit field in register

(4) R/W

Indicates access attribute (Read/Write) of each field

Each sign shown in this section means the following:

R0	0 always read at read. Write access is Don't care.
W0	Only 0 can be written
R	Enable read
RX	Enable read (read values undefined)
RW	Enable read and write any data
RW0	Enable read and write 0

(5) Default

This section shows the reset defaults for each bit field.

7.1.1 Host Interface Registers

DTC (DMA Transfer Count)

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

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

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

DSU (DMA Set Up)

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

Bit0 DW(DMA Word)

Sets DMA transfer unit

0 1 long words (32 bytes) per DMA transaction

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

Bit1 DBM (DMA Bus request Mode)

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

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

1 DREQ is irrelevant to the cycle steal mode or burst mode, and is negated when the MB86290A cannot receive data (that is, when Ready cannot be returned immediately). When the MB86290A is ready to receive data, DREQ is reasserted (When DMA transfer is performed in the single-address mode, DREQ is controlled automatically).

Bit2 DAM(DMA Address Mode)

Sets DMA addressing mode

0 Dual address mode

1 Single address mode (SH4 only)

DRM (DMA Request Mask)

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

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

DST (DMA STatus)

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

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

DTS (DMA Transfer Stop)

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

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

LSTA (displayList transfer STAtus)

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

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

DRQ (DMA ReQquest)

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

Starts sending external DMA request signal

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

IST (Interrupt STatus)

Register address	HostBaseAddress + 20h							
Bit #	7	6	5	4	3	2	1	0
Bit field name	Reserved			FSYNC	SYNCERR	VSYNC	CEND	CERR
R/W	R0			RW0	RW0	RW0	RW0	RW0
Default	0			0	0	0	0	0

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

Bit 0 CERR (Command Error Flag)
Draws command execution error interrupt

Bit 1 CEND (Command END)
Draws command complete interrupt

Bit 2 VSYNC (Vertical Sync.)
VSYNC detection interrupt

Bit 3 FSYNC (Frame Sync.)
Indicates frame synchronization interrupt

Bit 4 SYNCERR (Sync. Error)
Indicates external synchronization error interrupt

IMASK (Interrupt MASK)

Register address	HostBaseAddress + 24h							
Bit #	7	6	5	4	3	2	1	0
Bit field name	Reserved			SYNCERRM	FSYNCM	VSYNCM	CENDM	CERRM
R/W	R0			RW	RW	RW	RW	RW
Default	0			0	0	0	0	0

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

Bit 0 CERRM (Command Error Interrupt Mask)
Masks draw command execution error interrupt

Bit 1 CENDM (Command Interrupt Mask)
Masks draw command complete interrupt

Bit 2 VSYNCM (Vertical Sync. Interrupt Mask)
Masks VSYNC detection interrupt

Bit 3 FSYNCM (Frame Sync. Interrupt Mask)
Masks frame synchronization interrupt

Bit 4 SYNCERRM (Sync. Error Interrupt Mask)
Masks external synchronization error interrupt

SRST (Software ReSeT)

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

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

LSA (displayList Source Address)

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

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

LCO (displayList Count)

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

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

LREQ (displayList transfer REQuest)

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

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

7.1.2 Graphics Memory Interface Registers

MMR (Memory I/F Mode Register)

Register address	HostBaseAddress + FFFCh																													
Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve							TRRD	TRC			TRP	TRAS			TRCD	LOWD		RTS			RAW	ASW	CL						
R/W	W0	RX						RW	RW			RW	RW			RW	RW		RW			RW	RW	RW						
Default	00	Don't care						11	1001			11	110			11	10		0111			10	1	011						

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

Bits 2-0 CL (CAS Latency)

Set CAS latency cycles. Set same value at mode register of SDRAM.

011	CL3
010	CL2
Others	Prohibited

Bit 3	ASW (Attached SDRAM bit Width)	
	Sets data bus width of Graphics Memory interface	
	1	64 bit
	0	32 bit
Bits 5-4	RAW (Row Address Width)	
	Set bit width of Row address	
	00	14 bit
	11	13 bit
	Others	Prohibited
Bits 9-6	RTS (Refresh Timing Setting)	
	Set refresh interval	
	1010	1024 clocks
	1001	512 clocks
	1000	256 clocks
	0111	128 clocks
	Others	Prohibited

Bits 11-10 LOWD

Set last data output to next write command input latency

10	2 clocks
Others	Prohibited

Bits 13-12 TRCD

Set Bank Active to CAS latency

11	3 clocks
10	2 clocks
Others	Prohibited

Bits 16-14 TRAS

Set minimum Bank Active cycle

111	7 clocks
110	6 clocks
101	5 clocks
Others	Prohibited

Bits 18-17 TRP

Set Precharge to Bank Active wait time

11	3 clocks
10	2 clocks
Others	Prohibited

Bits 22-19 TRC

Set refresh to Bank Active wait time

1010	10 clocks
1001	9 clocks
1000	8 clocks
0111	7 clocks
Others	Prohibited

Bits 24-23 TRRD

Set Bank Active to next Bank Active wait time

11	3 clocks
10	2 clocks
Others	Prohibited

7.1.3 Display Control Register

DCM (Display Control Mode)

Register address	DisplayBaseAddress + 00h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	CKS	Reserve		SC					Reserve		EO	Reserve	SOF	ESY	SYNC	
R/W	RW	R0		RW					R0		RW	R0	RW	RW	RW	
Default	0	00		11110					00		0	0	0	0	00	

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

Bits 1-0 SYNC (Synchronize)

Set synchronization mode

X0 Non-interlace mode

11 Interlace video mode

Bit 2 ESY (External Synchronize)

Sets external synchronization mode

0 Disable

1 Enable

Bit 3 SF (Synchronize signal output format)

Sets active level of synchronization (VSYNC, HSYNC, CSYNC) signals

0 Low active

1 High active

Bit 5 EO (Even/Odd signal mode)

Defines EO signal output format

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

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

Bits 12-8 SC (Scaling)

Define pre-scaling ratio to generate dot clock

00000 No pre-scaling

00001 1/2

00010 1/3

: :

11110 1/31 (default)

11111 1/32

Bit 15 CKS (Clock Source)

Selects source clock

0 Internal PLL output clock

1 DCLKI input

DCE (Display Controller Enable)

Register address	DisplayBaseAddress + 02h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DEN	Reserved											BE	ME	WE	CE
R/W	RW	R0											RW	RW	RW	RW
Default	0	0											0	0	0	0

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

Bit 0 CE (C layer Enable)

Enables C-layer display

0 Does not display C layer

1 Displays C-layer

Bit 1 WE (W layer Enable)

Enables W-layer display

0 Does not display W-layer

1 Displays W layer

Bit 2 ME (M layer Enable)

Enables M layer display

0 Does not display M layer

1 Displays M layer

Bit 3 BE (BL-layer Enable)

Enables ML-layer display

0 Does not display B layer

1 Displays B layer

Bit 15 DEN (Display Enable)

Enables display

0 Does not output display signal

1 Outputs display signal

HTP (Horizontal Total Pixels)

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

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

HDP (Horizontal Display Period)

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

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

HDB (Horizontal Display Boundary)

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

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

HSP (Horizontal Synchronize pulse Position)

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

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

HSW (Horizontal Synchronize pulse Width)

Register address	DisplayBaseAddress + 0Eh							
Bit #	7	6	5	4	3	2	1	0
Bit field name	Reserved		HSW					
R/W	R0		RW					
Default	0		Don't care					

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

VSW (Vertical Synchronize pulse Width)

Register address	DisplayBaseAddress + 0Fh							
Bit #	7	6	5	4	3	2	1	0
Bit field name	Reserved		VSW					
R/W	R0		RW					
Default	0		Don't care					

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

VTR (Vertical Total Rasters)

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

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

VSP (Vertical Synchronize pulse Position)

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

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

VDP (Vertical Display Period)

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

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

WX (Window position X)

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

This register controls the horizontal position of the left edge of the Window layer. Set the left edge position of the Window layer from the display field start edge in dot-clock units.

WY (Window position Y)

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

This register controls the vertical position of the top edge of the Window layer. Set the top edge position of the Window layer from the display field start edge in raster units.

WW (Window Width)

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

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

WH (Window Height)

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

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

CM (C-layer Mode)

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

Bits 11-0 CH (C-layer Height)

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

Bits 23-16 CW (C-layer memory Width)

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

Bit 31 CC (C-layer Color mode)

Sets color mode used for Console layer

0 Indirect color mode (8 bits/pixel)

1 Direct color mode (16 bits/pixel)

COA(C-layer Origin Address)

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

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

CDA (C-layer Display Address)

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

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

CDX (C-layer Display position X)

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

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

CDY (C-layer Display position Y)

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

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

WM (W-layer Mode)

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

Bits 21-16 WW(W-layer memory Width)

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

Bit 31 WC (W-layer Color mode)

Sets color mode for W-layer

0 Indirect color (8 bits/pixel) mode

1 Direct color (16 bits/pixel) mode

WOA (W-layer Origin Address)

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

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

WDA (W-layer Display Address)

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

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

MLM (ML-layer Mode)

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

Bits 11-0 MLH (ML-layer Height)

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

Bits 23-16 MLW (ML-layer memory Width)

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

Bits 30-29 MLFLP (ML-layer Flip mode)

Set flipping mode for Middle Left (ML) layer

00 Display frame 0

01 Display frame 1

10 Switch frame 0 and 1 back and forth

11 Reserved

Bit 31 MLC (ML-layer Color mode)

Sets color mode for Middle Left (ML) layer

0 Indirect color mode (8 bits/pixel)

1 Direct color mode (16 bits/pixel)

MLOA0 (ML-layer Origin Address 0)

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

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

MLDA0 (ML-layer Display Address 0)

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

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

MLOA1 (ML-layer Origin Address 1)

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

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

MLDA1 (ML-layer Display Address 1)

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

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

MLDX (ML-layer Display position X)

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

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

MLDY (ML-layer Display position Y)

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

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

MRM (MR-layer Mode)

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

Bits 11-0 MRH (MR-layer Height)

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

Bits 23-16 MRW (MR-layer memory Width)

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

Bits 30-29 MRFLP (MR-layer Flip mode)

Set flipping mode for Middle Right (MR) layer

00 Display frame 0

01 Display frame 1

10 Switch frame 0 and 1 back and forth

11 Reserved

Bit 31 MRC (MR-layer Color mode)

Sets color mode for Middle Right (MR) layer

0 Indirect color mode (8 bits/pixel)

1 Direct color mode (16 bits/pixel)

MROA0 (MR-layer Origin Address 0)

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

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

MRDA0 (MR-layer Display Address 0)

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

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

MROA1 (MR-layer Origin Address 1)

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

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

MRDA1 (MR-layer Display Address 1)

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

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

MRDX (MR-layer Display position X)

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

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

MRDY (MR-layer Display position Y)

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

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

BLM (BL-layer Mode)

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

Bits 11-0 BLH (BL-layer Height)

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

Bits 23-16 BLW (BL-layer memory Width)

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

Bits 30-29 BLFLP (BL-layer Flip mode)

Set flipping mode for Base Left (BL) layer

00 Display frame 0

01 Display frame 1

10 Switch frame 0 and 1 back and forth

11 Reserved

Bit 31 BLC (BL-layer Color mode)

Sets color mode for Base Left (BL) layer

0 Indirect color mode (8 bits/pixel)

1 Direct color mode (16 bits/pixel)

BLOA0 (BL-layer Origin Address 0)

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

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

BLDA0 (BL-layer Display Address 0)

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

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

BLOA1 (BL-layer Origin Address 1)

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

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

BLDA1 (BL-layer Display Address 1)

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

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

BLDX (BL-layer Display position X)

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

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

BLDY (BL-layer Display position Y)

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

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

BRM (BR-layer Mode)

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

Bits 11-0 BRH (BR-layer Height)

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

Bits 23-16 BRW (BR-layer memory Width)

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

Bits 30-29 BRFLP (BR-layer Flip mode)

Set flipping mode for Base Right (BR) layer

00 Display frame 0

01 Display frame 1

10 Switch frame 0 and 1 back and forth

11 Reserved

Bit 31 BRC (BR-layer Color mode)

Sets color mode for Base Right (BR) layer

0 Indirect color mode (8 bits/pixel)

1 Direct color mode (16 bits/pixel)

BROA0 (BR-layer Origin Address 0)

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

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

BRDA0 (BR-layer Display Address 0)

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

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

BROA1 (BR-layer Origin Address 1)

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

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

BRDA1 (BR-layer Display Address 1)

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

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

BRDX (BR-layer Display position X)

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

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

BRDY (BR-layer Display position Y)

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

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

CUTC (CUrsor Transparent Control)

Register address	DisplayBaseAddress + A0h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved							CUZT	CUTC							
R/W	R0							RW	RW							
Default	0							Don't care	Don't care							

Bits 7-0 CUTC (Cursor Transparent Code)

Set transparency color code

Bit 8 CUZT (Cursor Zero Transparency)

Defines treatment of color code 0

0 Code 0 transparency color

1 Code 0 not transparency color

CPM (Cursor Priority Mode)

Register address	DisplayBaseAddress + A2h							
Bit #	7	6	5	4	3	2	1	0
Bit field name	Reserved		CEN1	CEN0	Reserved		CUO1	CUO0
R/W	R0		RW	RW	R0		RW	RW
Default	0		0	0	0		0	0

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

Bit 0 CUO0 (Cursor Overlap 0)

Sets display priority between cursor 0 and pixels of Console layer

0 Put cursor 0 at bottom of Console layer.

1 Put cursor 0 at top of Console layer.

Bit 1 CUO1 (Cursor Overlap 1)

Sets display priority between cursor 1 and pixels of Console layer

0 Put cursor 1 at bottom of Console layer.

1 Put cursor 1 at top of Console layer.

Bit 4 CEN0 (Cursor Enable 0)

Sets display enable of cursor 0

0 Disable

1 Enable

Bit 5 CEN1 (Cursor Enable 1)

Sets display enable of cursor 1

0 Disable

1 Enable

CUOA0 (Cursor-0 Origin Address)

Register address	DisplayBaseAddress + A4h																													
Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserve						CUOA0																									
R/W	R0						RW																								R0	
Default	0						Don't care																								0000	

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

CUX0 (Cursor-0 X position)

Register address	DisplayBaseAddress + A8h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUX0											
R/W	R0				RW											
Default	0				Don't care											

This register controls the horizontal position of the cursor-0 pattern left edge. Set the left-edge position of the cursor-0 pattern from the start edge of the display field in dot-clock units.

CUY0 (Cursor-0 Y position)

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

This register controls the vertical position of the cursor-0 pattern top edge. Set the top edge position of the cursor-0 pattern from the start edge of the display field in raster units.

CUOA1 (Cursor-1 Origin Address)

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

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

CUX1 (Cursor-1 X position)

Register address	DisplayBaseAddress + B0h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUX1											
R/W	R0				RW											
Default	0				Don't care											

This register controls the horizontal position of the cursor-1 pattern left edge. Set the left edge position of the cursor-0 pattern from the start edge of the display field in dot-clock units.

CUY1 (Cursor-1 Y position)

Register address	DisplayBaseAddress + B2h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved				CUY1											
R/W	R0				RW											
Default	0				Don't care											

This register controls the vertical position of the cursor-1 pattern top edge. Set the top edge position of the cursor-0 pattern from the start edge of the display field in raster units.

BRATIO (Blend Ratio)

Register address	DisplayBaseAddress + B4h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	BRS	Reserved							BRATIO				Reserved			
R/W	RW	R0							RW				R0			
Default	0	0							0				0000			

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

Bits 7-4 BRATIO (Blend Ratio)

Set blending ratio

0000 Coefficient = 0

0001 Coefficient = 1/16

: :

1111 Coefficient = 15/16

Bit 15 BRS (Blend Ratio Select)

Selects formula for alpha blending

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

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

BMODE (Blend MODE)

Register address	DisplayBaseAddress + B6h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved														Reserve	BLEND
R/W	R0														R0	RW
Default	0														0	0

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

Bit 0 BLEND

Overlays mode between C and B/M/W

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

1 Blending mode

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

KEYC (Key Color)

Register address	DisplayBaseAddress + B8h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	KYEN	KYC														
R/W	RW	RW														
Default	0	Don't care														

Bits 14-0 KYC (Key Color)

Set key color for chroma-key operation. Bits 7-0 used in indirect color mode. Bits 7-0 are used when the indirect color mode (8 bits/pixel) and the chroma key mode are set to the C-layer color.

Bit 15 KYEN (chroma-Key Enable)

Enables/disables chroma-key operation

0 Disable chroma-key operation (H always output from GV pin).

1 Enable chroma-key operation.

CKM (Chroma Key Mode)

Register address	DisplayBaseAddress + BAh															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	Reserved															KCS
R/W	R0															RW
Default	0															0

Bit 0 KCS (Key Color Select)

Selects key color as C-layer color or display color

0 Set key color as display color.

1 Set key color as C-layer color.

(See *Section 5.5*.)

CTC (C-layer Transparent Control)

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

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

Bits 14-0 CTC (C-layer Transparent Color)

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

Bit 15 CZT (C-layer Zero Transparency)

Sets treatment for code 0 in Console layer

0 Code 0 not transparent color

1 Code 0 transparent color

MRTC (MR-layer Transparent Control)

Register address	DisplayBaseAddress + C0h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MRZT	MRTC														
R/W	RW	RW														
Default	0	Don't care														

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

Bits 14-0 MRTC (MR-layer Transparent Color)

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

Bit 15 MRZT (MR-layer Zero Transparency)

Sets treatment for code 0 in MR-layer

0 Code 0 not transparent color

1 Code 0 transparent color

MLTC (ML-layer Transparent Control)

Register address	DisplayBaseAddress + C2h															
Bit #	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	MLZT	MLTC														
R/W	RW	RW														
Default	0	Don't care														

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

Bits 14-0 MLTC (ML-layer Transparent Color)

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

Bit 15 MLZT (ML-layer Zero Transparency)

Sets treatment for code 0 in ML-layer

0 Code 0 not transparent color

1 Code 0 transparent color

CPAL0-255 (C-layer Pallet 0-255)

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

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

Bits 7-2 B (Blue)

Set blue color element

Bit 15-10 G (Green)

Set green color element

Bits 23-18 R (Red)

Set red color element

Bit 31 A (Alpha)

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

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

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

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

Bits 7-2 B (Blue)

Set blue color element

Bits 15-10 G (Green)

Set green color element

Bits 23-18 R (Red)

Set red color element

7.1.4 Draw Control Registers

CTR (Control Register)

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

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

Bits 1-0 PS (Pixel engine Status)

Indicate status of pixel engine unit

00 Idle

01 Busy

10 Reserved

11 Reserved

Bits 5-4 DS (DDA Status)

Indicate status of DDA

00 Idle

01 Busy

10 Reserved

11 Reserved

Bits 9-8 SS (Setup Status)

Indicate status of Set up unit

00 Idle

01 Busy

10 Reserved

11 Reserved

Bit 12 FE (FIFO Empty)

Indicates status of display list FIFO

0 Valid data

1 No valid data

Bit 13 FF (FIFO Full)

Indicates fullness of display list FIFO

0 Not full

1 Full

Bit 14 NF (FIFO Near Full)

Indicates entries of display list FIFO

0 Empty entries equal to or more than half

1 Empty entries less than half

Bits 20-15 FCNT(FIFO Counter)

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

Bit 22 CE (Display List Command Error)

Indicates command error detection

0 Normal

1 Command error detected

Bit 23 PE (Display List Packet code Error)

Indicates packet code error detection

0 Normal

1 Packet code error detected

Bit 24 FO (FIFO Overflow)

Indicates FIFO overflow status

0 Normal

1 FIFO overflow detected

IFSR (Input FIFO Status Register)

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

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

IFCNT (Input FIFO Counter)

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

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

SST (Setup engine Status)

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

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

DST (DDA Status)

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

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

PST (Pixel engine Status)

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

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

EST (Error Status)

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

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

7.1.5 Draw mode Parameter Registers

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

MDR0 (Mode Register for miscellaneous)

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

Bits 1-0 BSH (Bitmap Scale Horizontal)

Set horizontal zoom ratio of bitmap draw

00 x1

01 x2

10 x1/2

01 Reserved

Bits 3-2 BSV (Bitmap Scale Vertical)

Set vertical zoom ratio of bitmap draw

00 x1

01 x2

10 x1/2

01 Reserved

Bit 8 CX (Clip X enable)

Sets X-coordinate clipping mode

0 Disable

1 Enable

Bit 9 CY (Clip Y enable)

Sets Y-coordinate clipping mode

0 Disable

1 Enable

Bit 15 CF (Color Format)

Sets drawing color format of current draw frame

0 Indirect color mode (8 bits/pixel)

1 Direct color mode (16 bits/pixel)

MDR1 (Mode Register for LINE)

Register address	DrawBaseAddress + 424h																													
Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name				LW									BL							LOG				BM		ZW	ZCL			ZC		
R/W				RW									RW							RW				RW		RW	RW			RW		
Default				00000									0							0011				0		0	0000			0		

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

Bit 2 ZC (Z Compare mode)

Sets Z comparison mode

0 Disable

1 Enable

Bits 5-3 ZCL (Z Compare Logic)

Select type of Z comparison

000 NEVER

001 ALWAYS

010 LESS

011 LEQUAL

100 EQUAL

101 GEQUAL

110 GREATER

111 NOTEQUAL

Bit 6 ZW (Z Write mask)

Sets ZWRITEMASK

0 Compare Z values and overwrite result to Z buffer.

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

Bits 8-7 BM (Blend Mode)

Set blend mode

00 Normal (source copy)

01 Alpha blending

10 Logical calculation enable

11 Reserved

Bits 12-9 LOG (Logical operation)

Set type of logical calculation

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

Bits 28-24 LW (Line Width)

Set line width

00000	1 pixel
00001	2 pixels
:	:
11111	32 pixels

MDR2 (Mode Register for Polygon)

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

This register controls the polygon draw mode.

Bit 0 SM (Shading Mode)

Sets shading mode

0 Flat shading

1 Gouraud shading

Bit 2 ZC (Z Compare mode)

Sets Z comparison mode

0 Disable

1 Enable

Bits 5-3 ZCL (Z Compare Logic)

Select type of Z comparison

000 NEVER

001 ALWAYS

010 LESS

011 LEQUAL

100 EQUAL

101 GEQUAL

110 GREATER

111 NOTEQUAL

Bit 6 ZW (Z Write mask)

Sets ZWRITEMASK

0 Compare Z values and overwrite result to Z buffer

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

Bits 8-7 BM (Blend Mode)

Set blend mode

00 Normal (source copy)

01 Alpha blending

10 Logical calculation enable

11 Reserved

Bits 12-9 LOG (Logical operation)

Set type of logical calculation

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

Bits 29-28 TT (Texture-Tile Select)

Select texture or tile pattern

00	Not used
01	Enable tiling operation
10	Enable texture mapping
11	Reserved

MDR3 (Mode Register for Texture)

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

This register controls the texture mapping mode.

Bit 0 TBU (Texture Buffer)

Selects texture memory (internal buffer always used in tiling)

0 External Graphics Memory

1 Internal texture buffer

Bit 3 TC (Texture coordinates Correct)

Controls perspective correction mode

0 Disable

1 Enable

Bit 5 TF (Texture Filtering)

Sets texture filtering mode

0 Point sampling

1 Bi-linear filtering

Bits 9-8 TWT (Texture Wrap T)

Set texture T-coordinate wrapping mode

00 Repeat

01 Cramp

10 Border

11 Reserved

Bits 11-10 TWS (Texture Wrap S)

Set texture S coordinate wrapping mode

00 Repeat

01 Cramp

10 Border

11 Reserved

Bits 17-16 TBL (Texture Blend mode)

Set texture blending mode

00 Decal

01 Modulate

10 Stencil

11 Reserved

Bits 21-20 TAB (Texture Alpha Blend mode)

Set texture alpha blending mode. The stencil alpha mode is used only when the BM bits in the MDR1 register are set to 01 (alpha blending). If any other mode is set at the BM bit field, the stencil alpha mode is treated as the stencil mode.

00 Normal

01 Stencil

10 Stencil alpha

11 Reserved

MDR4 (Mode Register for BLT)

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

This register controls the BitBLT. Mode.

Bits 8-7 BM (Blend Mode)

Set blend mode

00 Normal (source copy)

01 Reserved

10 Logical calculation enable

11 Reserved

Bits 12-9 LOG (Logical operation)

Set logical calculation type

0000 CLEAR

0001 AND

0010 AND REVERSE

0011 COPY

0100 AND INVERTED

0101 NOP

0110 XOR

0111 OR

1000 NOR

1001 Reserved

1010 INVERT

1011 OR REVERSE

1100 COPY INVERTED

1101 OR INVERTED

1110 NAND

1111 SET

FBR (Frame buffer Base)

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

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

XRES (X Resolution)

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

This register controls the drawing frame horizontal resolution.

ZBR (Z-buffer Base)

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

This register controls the Z-buffer base address.

TBR (Texture memory Base)

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

This register controls the texture memory base address.

PFBR (2D Polygon Flag-Buffer Base)

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

This register controls the polygon flag buffer base address.

CXMIN (Clip X minimum)

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

This register controls the clip frame minimum X position.

CXMAX (Clip X maximum)

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

This register controls the clip frame maximum X position.

CYMIN (Clip Y minimum)

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

This register controls the clip frame minimum Y position.

CYMAX (Clip Y maximum)

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

This register controls the clip frame maximum Y position.

TXS (Texture Size)

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

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

Bits 8-0 TXSM (Texture Size M)

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

000000100	M=4
000001000	M=8
000010000	M=16
000100000	M=32
001000000	M=64
010000000	M=128
100000000	M=256
Others	Prohibited

Bits 24-16 TXSN (Texture Size N)

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

000000100	N=4
000001000	N=8
000010000	N=16
000100000	N=32
001000000	N=64
010000000	N=128
100000000	N=256
Others	Prohibited

TIS (Tile Size)

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

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

Bits 6-0 TISM (Title Size M)

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

0.000100	M=4
0001000	M=8
0010000	M=16
0100000	M=32
1000000	M=64
Others	Prohibited

Bits 22-16 TISN (Title Size N)

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

0000100	N=4
0001000	N=8
0010000	N=16
0100000	N=32
1000000	N=64
Others	Prohibited

TOA (Texture Buffer Offset address)

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

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

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

FC (Foreground Color)

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

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

Bits 15-0 FGC (Foreground Color)

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

BC (Background Color)

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

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

Bits 14-0 BGC (Background Color)

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

Bit 15 BT (Background Transparency)

Sets transparent mode of background color

0 Draw background in color used in BGC field.

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

ALF (Alpha Factor)

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

This register controls the alpha blending ratio.

BLP (Broken Line Pattern)

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

This register controls the broken-line pattern. The bit 1 set in the broken-line pattern is drawn in the foreground color and bit 0 is drawn in the background color. The actual line pattern is pasted from MSB to LSB to the line to be drawn. If the length of the applied line is longer than 32 bits, the same line pattern is wrapped around in 32-bit units. The current position (bit #) of the line pattern used for the line is set in the BLPO register.

TBC (Texture Border Color)

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

This register controls the texture mapping border color.

Bits 14-0 BC (Border Color)

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

BLPO (Broken Line Pattern Offset)

Register address	DrawBaseAddress + 3E0h																														
Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name																												BCR				
R/W																												RW				
Default																												11111				

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

7.1.6 Triangle Draw Registers

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

(XY-coordinate register)

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ys	0000h	S	S	S	S	Int												0															
Xs	0004h	S	S	S	S	Int												Frac															
dXdy	0008h	S	S	S	S	Int												Frac															
XUs	000ch	S	S	S	S	Int												Frac															
dXUdy	0010h	S	S	S	S	Int												Frac															
XLs	0014h	S	S	S	S	Int												Frac															
dXLdy	0018h	S	S	S	S	Int												Frac															
USN	001bh	0	0	0	0	Int												0															
LSN	0020h	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 (X, Y) coordinates for triangle drawing

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

(Color register)

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

Address Offset from DrawBaseAddress

S Sign bit or sign extension

0 Not used or 0 extension

Int Integer or integer part of fixed point data

Frac Fraction part of fixed point data

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

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

(Z-coordinate register)

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

Address Offset from DrawBaseAddress

S Sign bit or sign extension

0 Not used or 0 extension

Int Integer or integer part of fixed point data

Frac Fraction part of fixed point data

Sets Z-coordinate for 3D triangle drawing

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

(Texture coordinate register)

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Ss	00c0h	S	S	S	S	S	S	S	Int																						Frac		
dSdx	00c4h	S	S	S	S	S	S	S	Int																						Frac		
dSdy	00c8h	S	S	S	S	S	S	S	Int																						Frac		
Ts	00cch	S	S	S	S	S	S	S	Int																						Frac		
dTdx	00d0h	S	S	S	S	S	S	S	Int																						Frac		
dTdy	00d4h	S	S	S	S	S	S	S	Int																						Frac		
Qs	00d8h	0	0	0	0	0	0	0	Int																						Frac		
dQdx	00dch	S	S	S	S	S	S	S	Int																						Frac		
dQdy	00e0h	S	S	S	S	S	S	S	Int																						Frac		

Address Offset from DrawBaseAddress

S Sign bit or sign extension

0 Not used or 0 extension

Int Integer or integer part of fixed point data

Frac Fraction part of fixed point data

Sets texture coordinate parameters for triangle drawing

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

7.1.7 Line Draw Registers

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

(Coordinate register)

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

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

Sets coordinate parameters for line drawing

LPN	Pixel length of line (*1)
LXs	X-coordinate position of line draw start vertex (-Pai/4 ≤ Line angle ≤ Pai/4) Set truncated integer value of X-coordinate. (Other than above) Set current integer part of fixed point X-coordinate data.
LXde	Line angle data for X axis (-Pai/4 ≤ Line angle ≤ Pai/4) Increment or decrement according to drawing direction. (Other than above) Set fraction part of DX/DY.
LYs	Y-coordinate position of line Pai draw start vertex (-Pai/4 ≤ Line angle ≤ Pai/4) Set current integer part of fixed point Y-coordinate data. (Other than above) Set truncated integer value of Y-coordinate.
LYde	Line angle data for Y-axis (-Pai/4 ≤ Line angle ≤ Pai/4) Set fraction part of dY/dX. (Other than above) Increment or decrement according to drawing direction.
LZs	Z-coordinate position of line draw start vertex
LZde	Z angle

(*1) If -Pai/4 ≤ Line angle ≤ Pai/4: Horizontal length of line in pixel units

Other than above: Vertical length of line in pixel units

(Color register)

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LRs	015ch	0	0	0	0	0	0	0	0	Int								Frac															
LRde	0160h	S	S	S	S	S	S	S	S	Int								Frac															
LGs	0164h	0	0	0	0	0	0	0	0	Int								Frac															
LGde	0168h	S	S	S	S	S	S	S	S	Int								Frac															
LBs	016ch	0	0	0	0	0	0	0	0	Int								Frac															
LBde	0170h	S	S	S	S	S	S	S	S	Int								Frac															

Address Offset from DrawBaseAddress

S Sign bit or sign extension

0 Not used or 0 extension

Int Integer or integer part of fixed point data

Frac Fraction part of fixed point data

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

LRs	R value at line draw start vertex
LRde	Differential value of R element
LGs	G value at line draw start vertex
LGde	Differential value of G element
LBs	B value at line draw start vertex
LBde	Differential value of B element

7.1.8 Pixel Plot Registers

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

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

Address Offset from DrawBaseAddress

S Sign bit or sign extension

0 Not used or 0 extension

Int Integer or integer part of fixed point data

Frac Fraction part of fixed point data

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

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

7.1.9 Rectangle Draw Registers

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

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

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

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

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

7.1.10 Blt Registers

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

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

SADDR	Sets start address of source field in byte boundary.
SStride	Sets horizontal size of source field
SRXs	Sets start X-coordinate position of source rectangle
SRYs	Sets start Y-coordinate position of source rectangle
DADDR	Sets start address of destination rectangle in byte boundary
DStride	Sets horizontal size of destination field
DRXs	Sets start X-coordinate position of destination rectangle
DRYs	Sets start Y-coordinate position of destination rectangle
BRsizeX	Sets horizontal size of rectangle
BRsizeY	Sets vertical size of rectangle

7.1.11 Fast2DLine Draw Registers

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

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

Address Offset from DrawBaseAddress

S Sign bit or sign extension

0 Not used or 0 extension

Int Integer or integer part of fixed point data

Frac Fraction part of fixed point data

Sets coordinate parameters of both end points for Fast2DLine drawing

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

7.1.12 Fast2DTriangle Draw Registers

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

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
X0dc	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 coordinate parameters of three vertices for Fast2DTriangle drawing

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

7.1.12 DisplayList FIFO Registers

DFIFO (Displaylist FIFO)

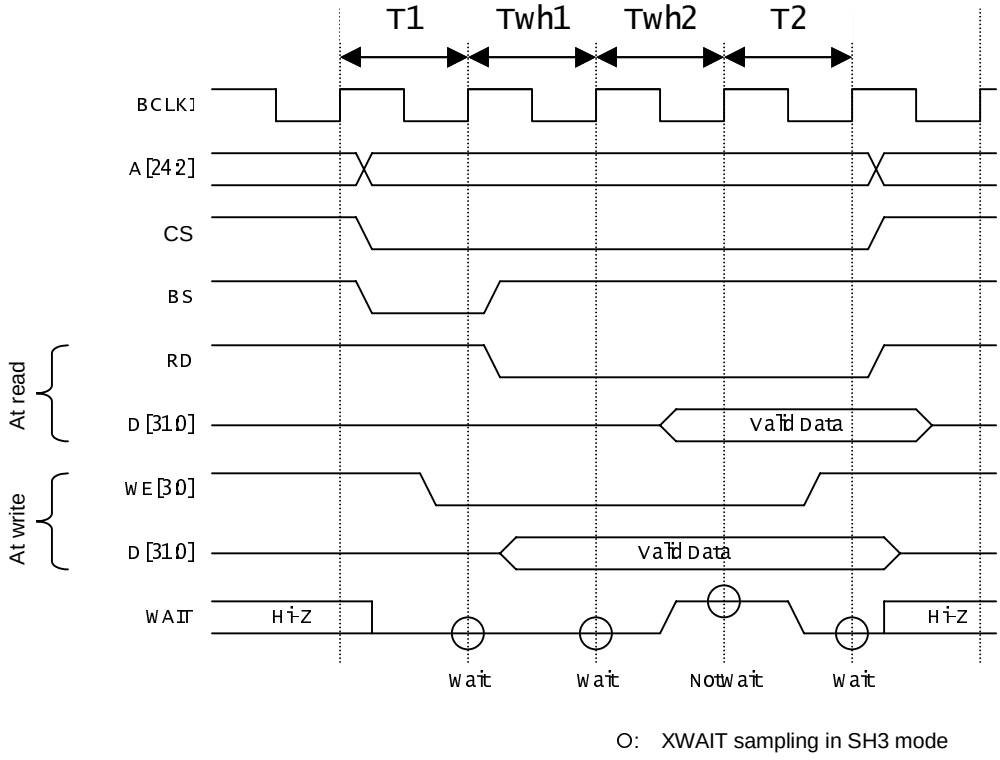
Register address	DrawBaseAddress + 4A0h																																
Bit #	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit field name	DFIFO																																
R/W	W																																
Default	Don't care																																

FIFO registers for DisplayList transfer

8 Timing Diagram

8.1 Host Interface

8.1.1 CPU Read/Write Timing Diagram for SH3 Mode



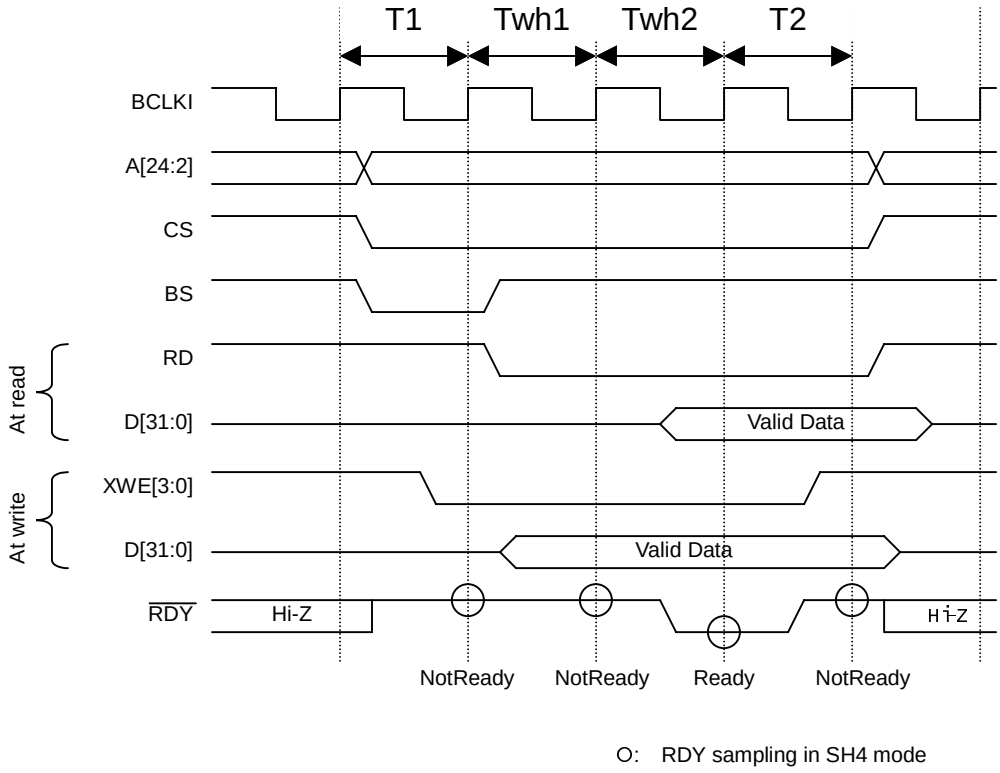
T1: Read/write start cycle (RDY in wait state)

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

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

Fig. 8.1 CPU Read/Write Timing Diagram for SH3 Mode

8.1.2 CPU Read/Write Timing Diagram for SH4 Mode



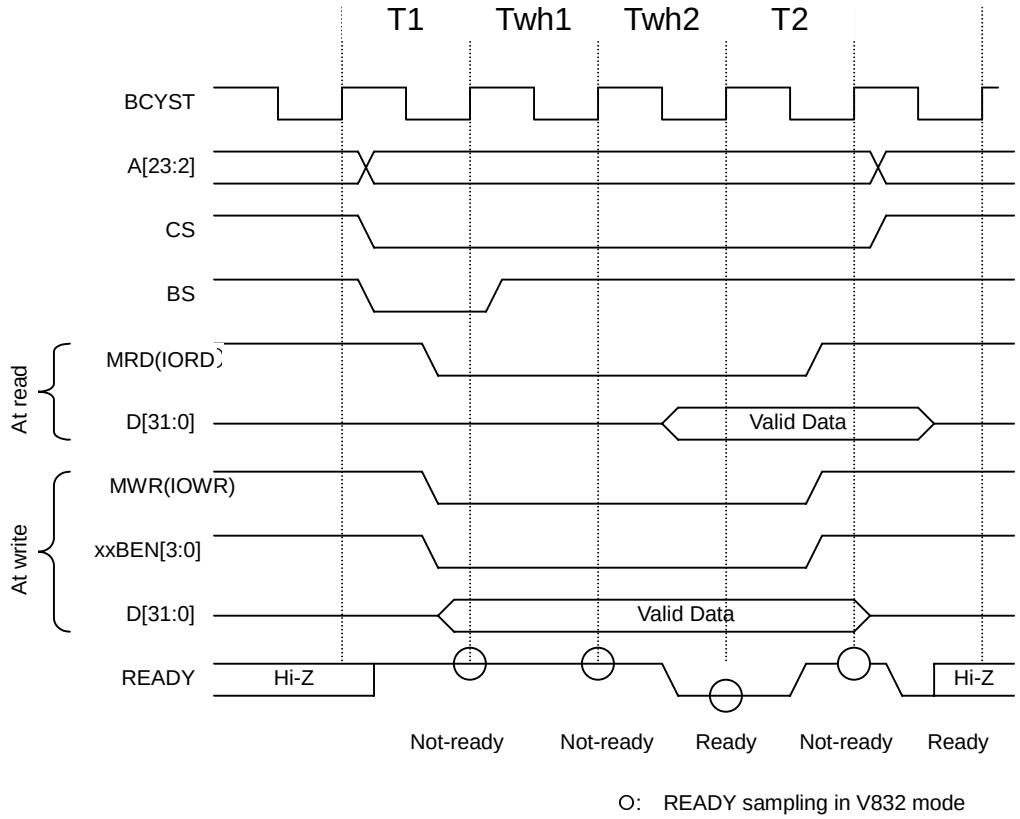
T1: Read/write start cycle (RDY is in the not-ready state.)

Twh*: Cycles inserted by hardware (RDY asserts Ready as soon as the preparations are made.)

T2: Read/write end cycle (RDY ends in the not-ready state.)

Fig. 8.2 CPU Read/Write Timing Diagram for SH4 Mode

8.1.3 CPU Read/Write Timing Diagram in V832 Mode



T1: Read/write start cycle (READY is in the not-ready state.)

Twh*: Cycles inserted by hardware (READY asserts Ready as soon as the preparations are made.)

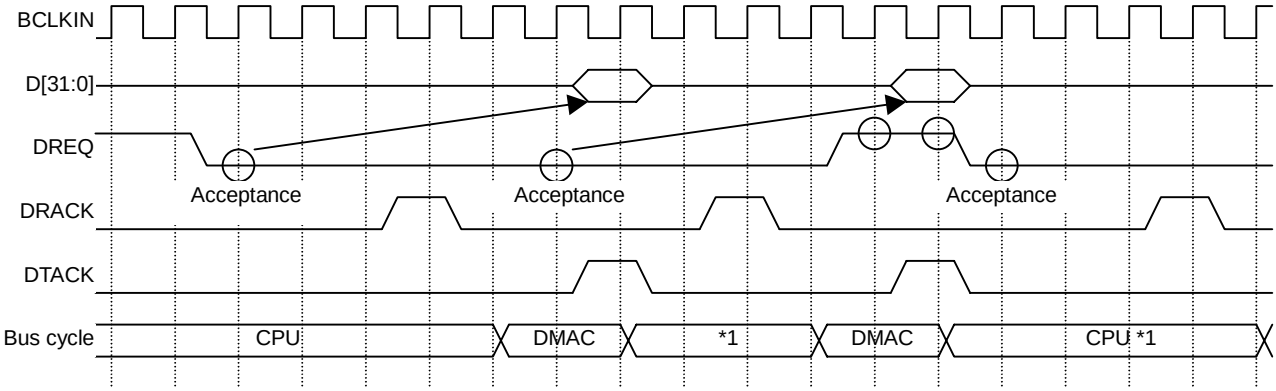
T2: Read/write end cycle (READY is in the not-ready state.)

READY is placed in the ready state and then set to Hi-Z.

Note: The xxBEN signal is used only when performing a write from the CPU; it is not used when performing a read from the CPU.

Fig. 8.3 CPU Read/Write Timing Diagram in V832 Mode

8.1.4 SH4 Single-address DMA Write (Transfer of 1 Long Word)



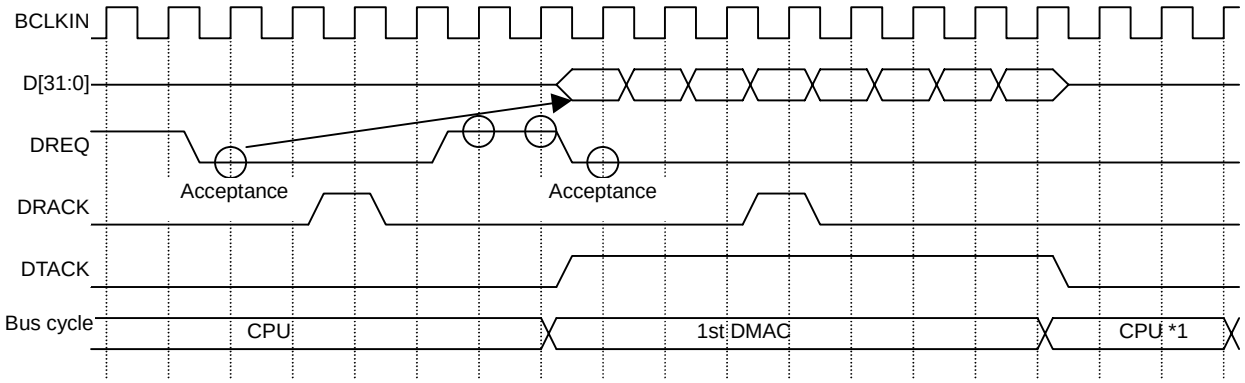
○: DREQ sampling and channel priority determination for SH mode (DREQ = level detection)

*1: In the cycle steal mode, even when DREQ is already asserted at the 2nd DREQ sampling, the right to use the bus is returned to the CPU once. In the burst mode, DMAC secures the right to use the bus unless DREQ is negated.

Fig. 8.4 SH4 Single-address DMA Write (Transfer of 1 Long Word)

The MB86290A writes data according to the DTACK assert timing. When data cannot be received, the DREQ signal is automatically negated. And then the DREQ signal is reasserted as soon as data becomes ready to be received.

8.1.5 SH4 Single-address DMA Write (Transfer of 8 Long Words)



○: DREQ sampling and channel priority determination for SH mode (DREQ = level detection)

*1: In the cycle steal mode, even when DREQ is already asserted at the 2nd DREQ sampling, the right to use the bus is returned to the CPU once. In the burst mode, DMAC secures the right to use the bus unless DREQ is negated.

Fig. 8.5 SH4 Single-address DMA Write (Transfer of 8 Long Words)

The MB86290A writes data in accordance with the DTACK assert timing. When data cannot be received, the DREQ signal is negated automatically. And then the DREQ signal is reasserted as soon as data becomes ready to be received.

8.1.6 SH3/4 Dual-address DMA (Transfer of 1 Long Word)

For the MB86290A, the read/write operation is performed according to the SRAM protocol.

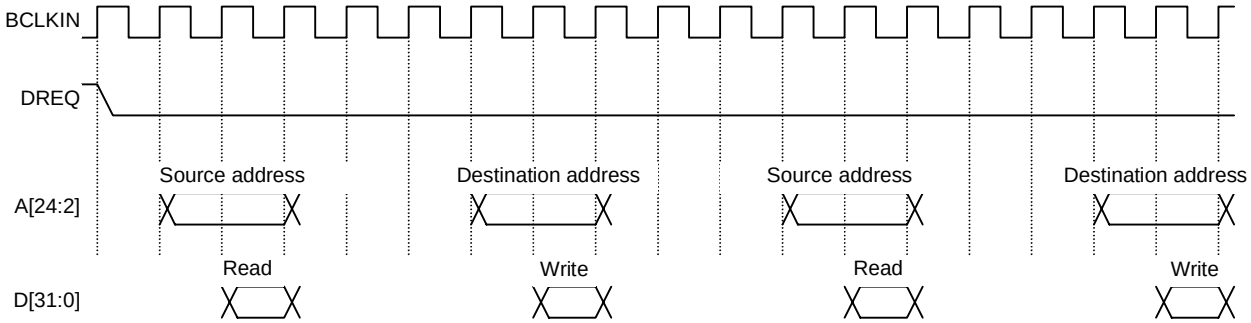


Fig. 8.6 SH3/4 Dual-address DMA (Transfer of 1 Long Word)

In the dual-address mode, the DREQ signal is kept asserted until the transfer ends by default. Consequently, to negate the DREQ signal when the MB86290A cannot return the Ready signal immediately, set the DBM register.

8.1.7 SH3/4 Dual-Address DMA (Transfer of 8 Long Words)

For the MB86290A, the read/write operation is performed according to the SRAM protocol.

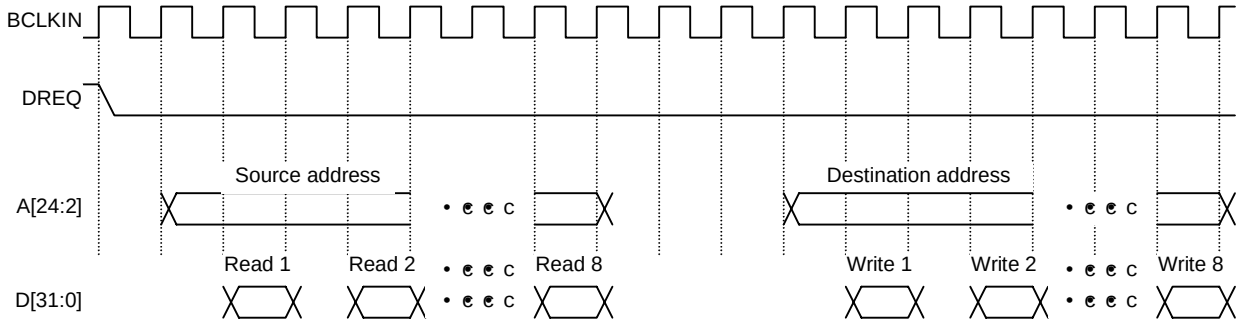


Fig. 8.7 SH3/4 Dual-address DMA (Transfer of 8 Long Words)

In the dual-address mode, the DREQ signal is kept asserted until the transfer ends by default. Consequently, to negate the DREQ signal when the MB86290A cannot return the Ready signal immediately, set the DBM register.

8.1.8 V832 DMA Transfer

For the MB86290A, the read/write operation is performed according to the SRAM protocol.

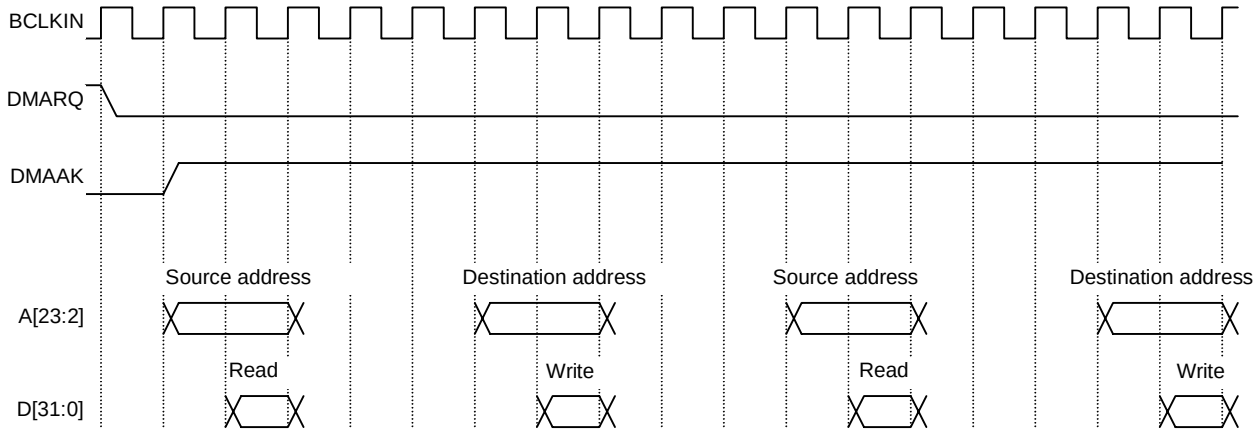
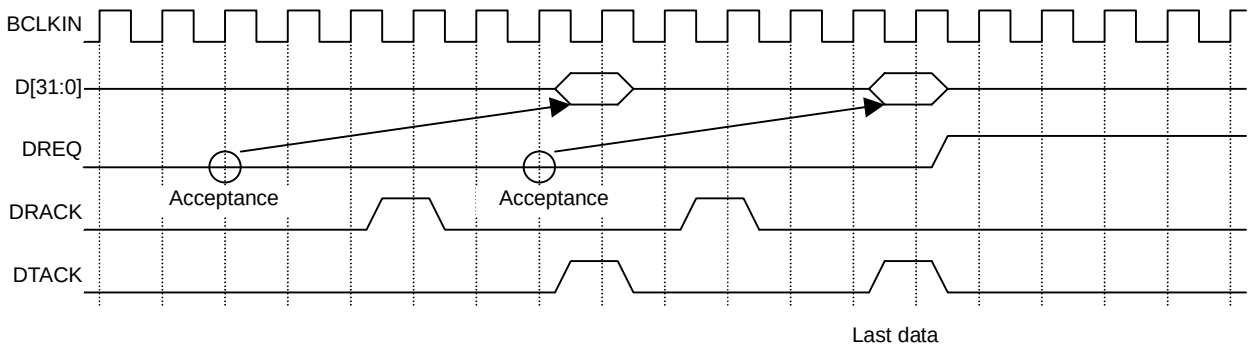


Fig. 8.8 V832 DMA Transfer

During DMA transfer, the DREQ signal is kept asserted until the transfer ends by default. Consequently, to negate the DREQ signal when the MB86290A cannot return the Ready signal immediately, set the DBM register.

8.1.9 SH4 Single-address DMA Transfer End Timing



○: DREQ sampling and channel priority determination for SH mode (DREQ = level detection)

Fig. 8.9 SH4 Single-address DMA Transfer End Timing

DREQ is negated three cycles after DRACK is written as the last data.

8.1.10 SH3/4 Dual-address DMA Transfer End Timing

For the MB86290A, the read/write operation is performed according to the SRAM protocol.

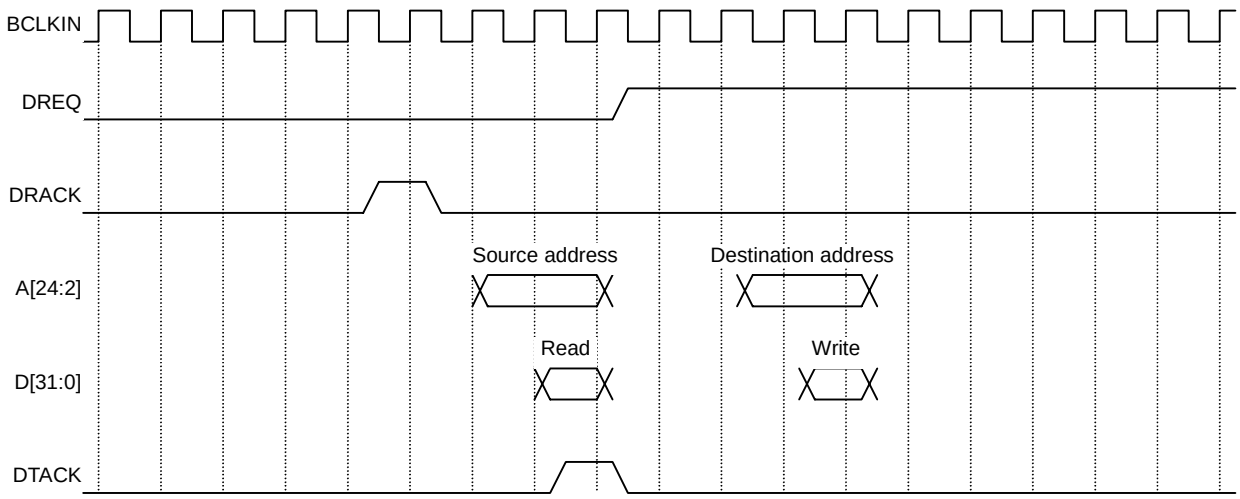


Fig. 8.10 SH3/4 Dual-address DMA Transfer End Timing

DREQ is negated three cycles after DRACK is written as the last data.

8.1.11 V832 DMA Transfer End Timing

For the MB86290A, the read/write operation is performed according to the SRAM protocol.

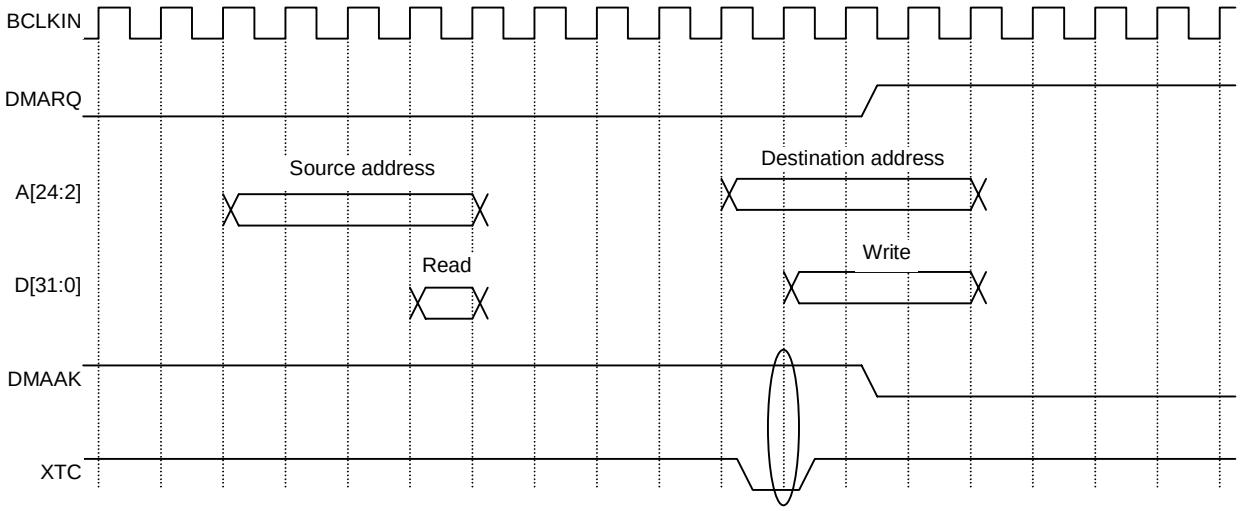


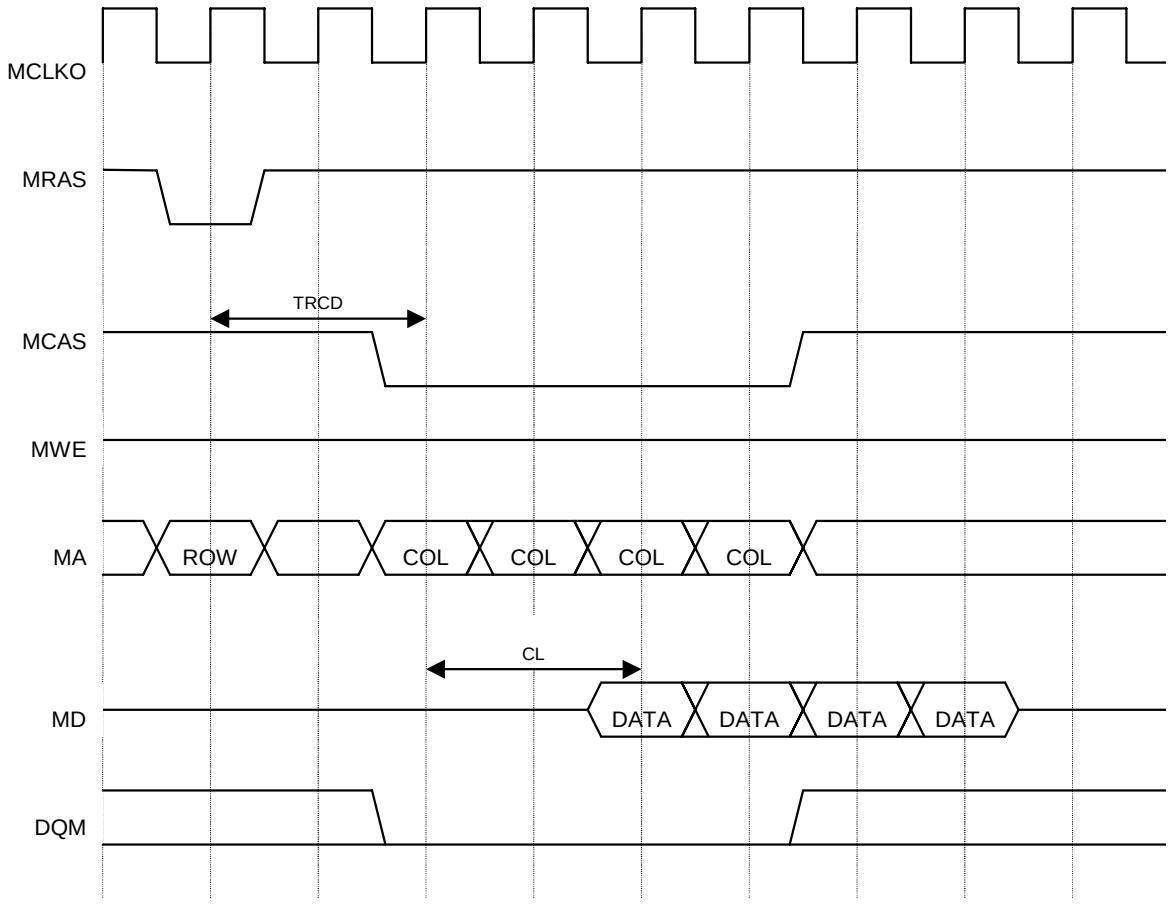
Fig. 8.11 V832 DMA Transfer End Timing

DMMAK and XTC are ANDed inside the MB86290A to end DMA.

8.2 Graphics Memory Interface

The access timing for the MB86290A and the graphics memory is explained.

8.2.1 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

Note: This timing is used when CL2 is operating.

Fig. 8.2.1 Timing of Read Access to Same Row Address

This timing diagram shows that the same row address of SDRAM is read-accessed four times from the MB86290A. The Read command is issued after TRCD has elapsed after the ACTV command was issued.

Data that is output after CL has elapsed after the Read command was issued is written to the MB86290A.

8.2.2 Timing of Read Access to Different Row Addresses

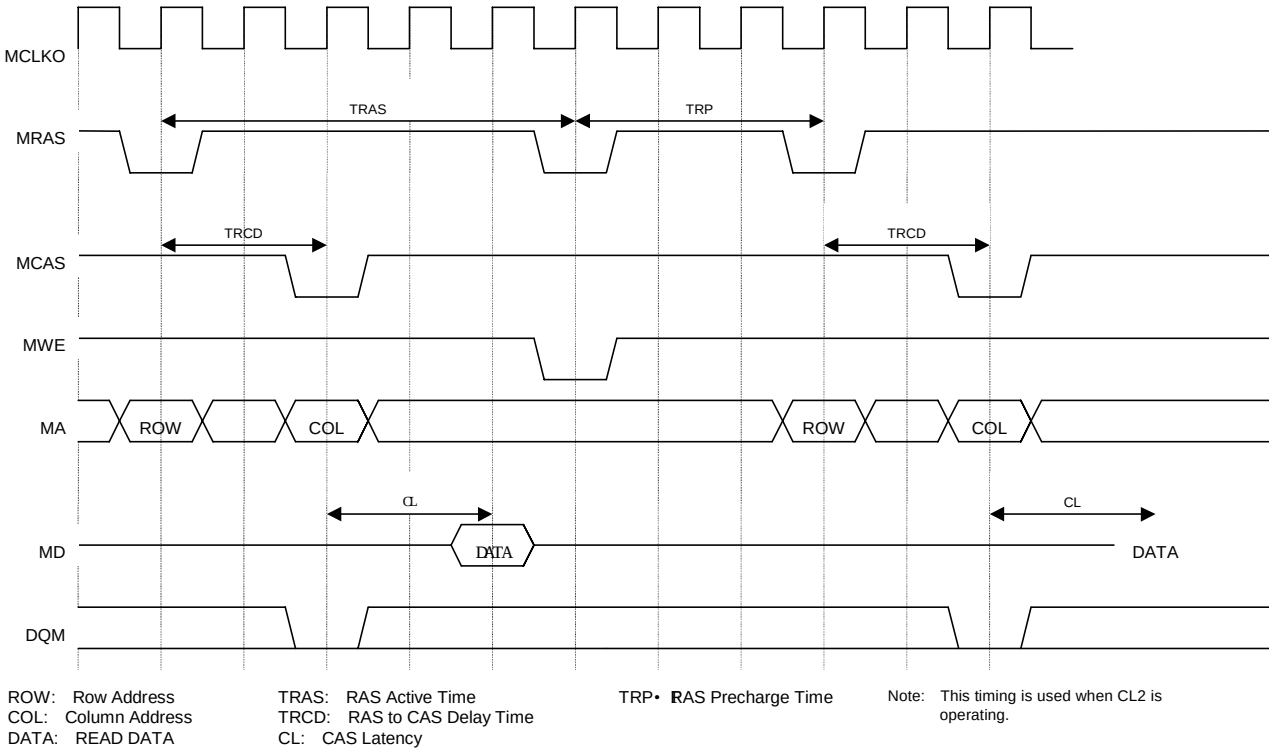
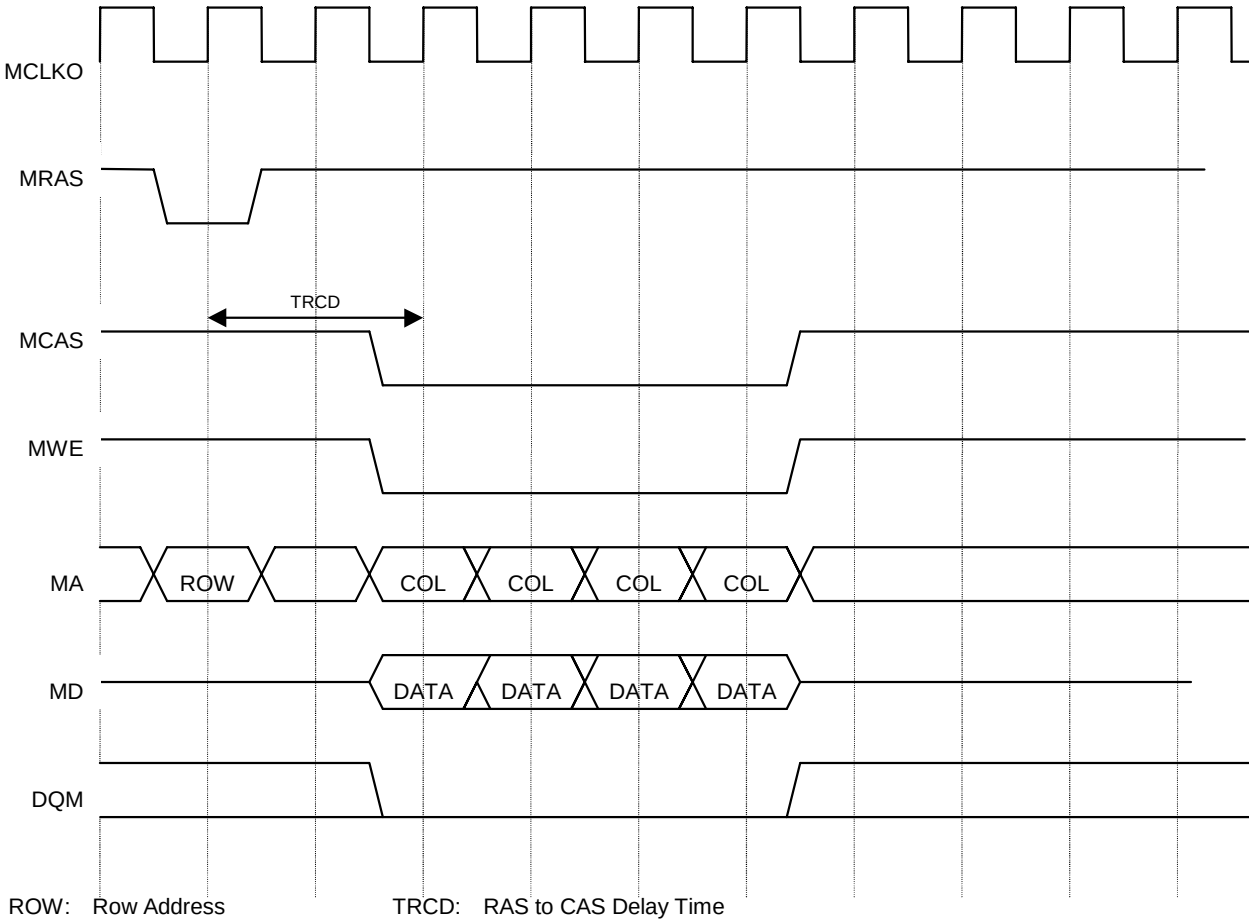


Fig. 8.2.1 Timing of Read Access to Different Row Addresses

This timing diagram shows that different row addresses of SDRAM are read-accessed from the MB86290A. An SDRAM page boundary is located between the address to be read first and the address to be read next. Consequently, the Precharge command is issued at the timing that meets the TRAS condition, and then after TRP has elapsed, the ACTV command is reissued and the Read command is issued.

8.2.3 Timing of Write Access to Same Row Address



ROW: Row Address
COL: Column Address
DATA: READ DATA
TRCD: RAS to CAS Delay Time

Fig. 8.2.3 Timing of Write Access to Same Row Address

This timing diagram shows that the same row address of SDRAM is write-accessed four times from MB86290A. The Write command is issued after TRCD has elapsed after the ACTV command is issued. Then, data is written to SDRAM.

8.2.4 Timing of Write Access to Different Row Addresses

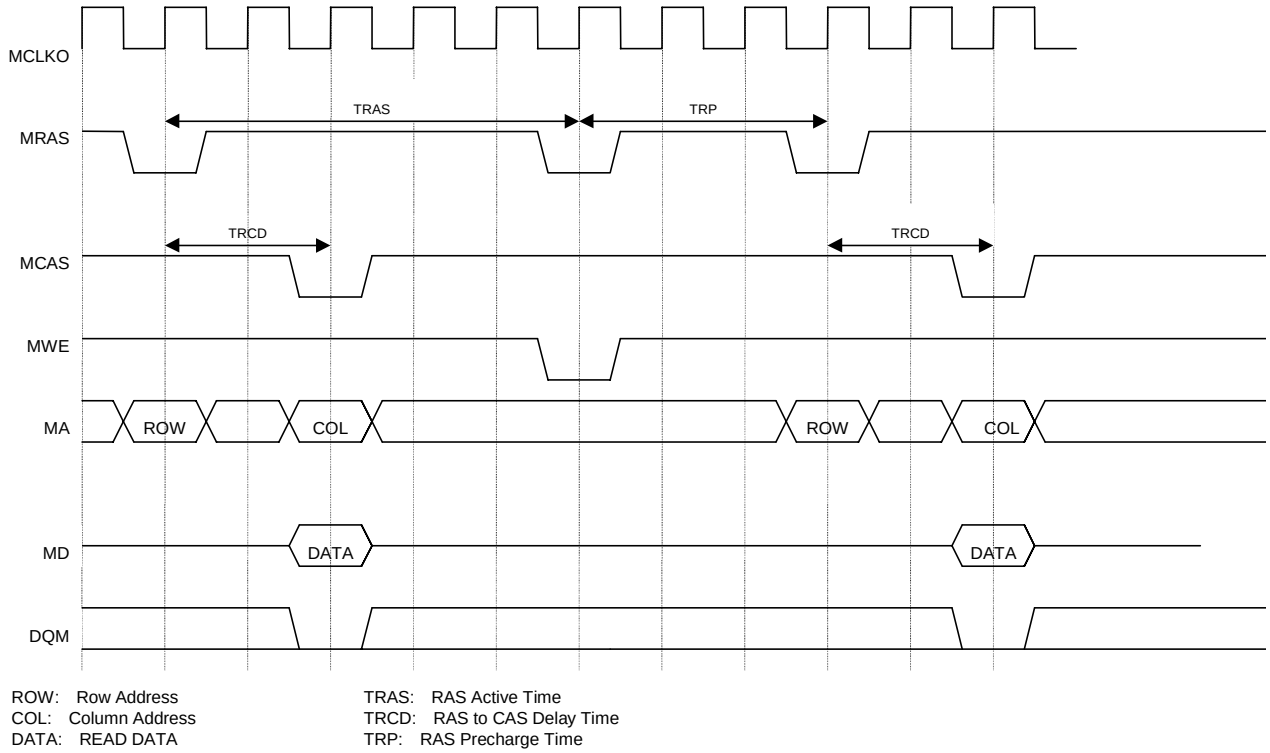


Fig. 8.2.4 Timing of Write Access to Different Row Addresses

This timing diagram shows that different row addresses of SDRAM are write-accessed from the MB86290A. An SDRAM page boundary is located between the address to be written to first and the address to be written to next. Consequently, the Precharge command is issued at the timing that meets the TRAS condition, and then after TRP has elapsed, the ACTV command is reissued and the Read command is issued.

8.2.5 Timing of Read/Write Access to Same Row Address

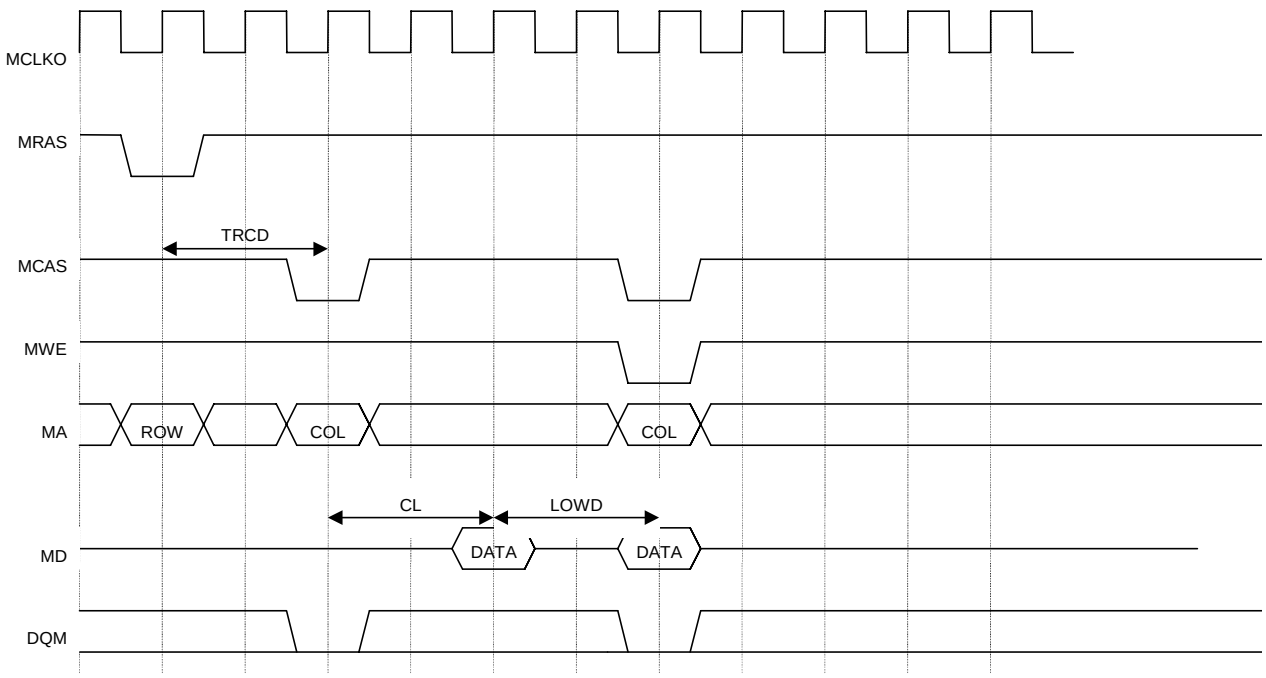
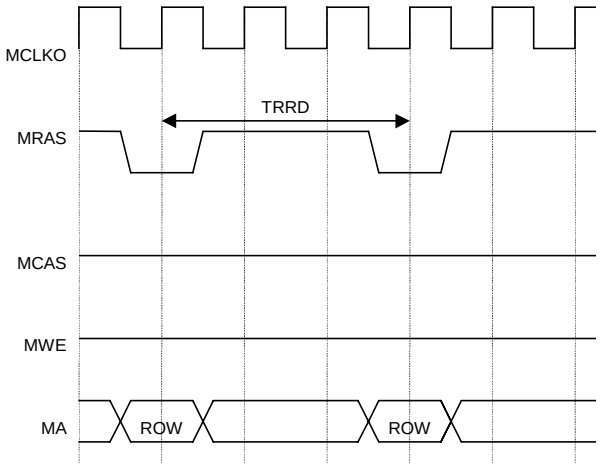


Fig. 8.2.5 Timing of Read/Write Access to Same Row Address

This timing diagram shows that a row address of SDRAM is read-accessed from the MB86290A, and then immediately afterwards the same row address is write-accessed from the MB86290A. The Write command is issued after LOWD has elapsed after read data is output from SDRAM.

8.2.6 Delay between ACTV Commands



ROW: Row Address
TRRD: RAS to RAS Bank Active Delay Time

Fig. 8.2.6 Delay between ACTV Commands

The ACTV command is issued to the SDRAM row address from the MB86290A after TRRD has elapsed after the previous ACTV command is issued.

8.2.7 Delay between Refresh Command and Next ACTV Command

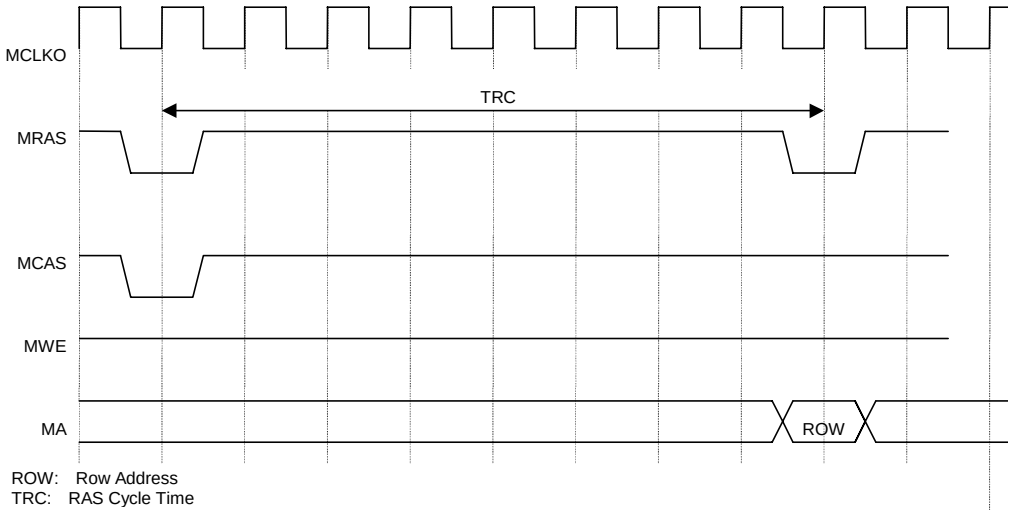
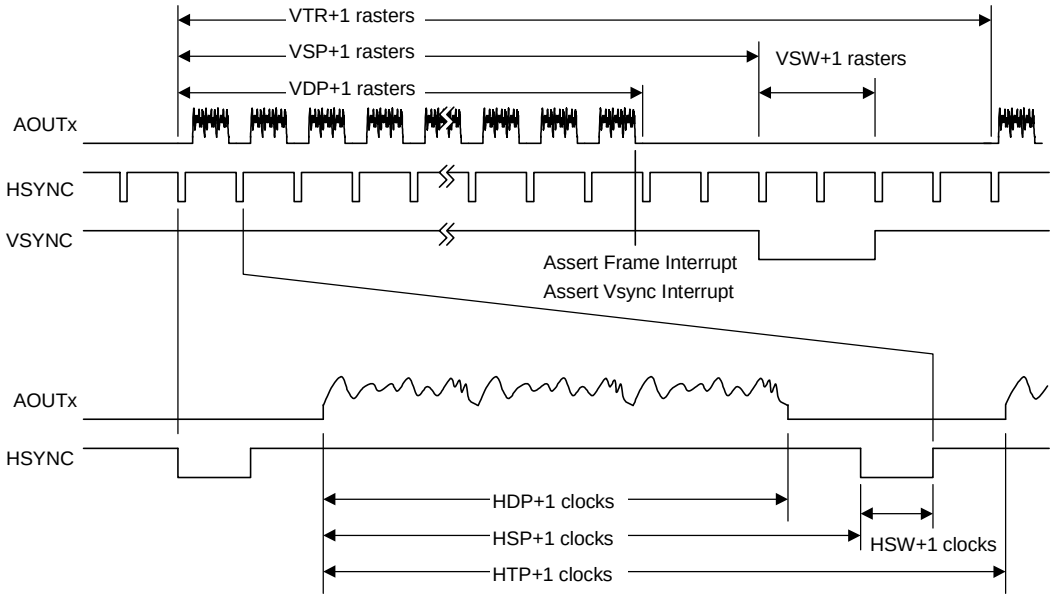


Fig. 8.2.7 Delay between Refresh Command and next ACTV Command

The ACTV command is issued after TRC has elapsed after the Refresh command is issued.

8.3 Display Timing

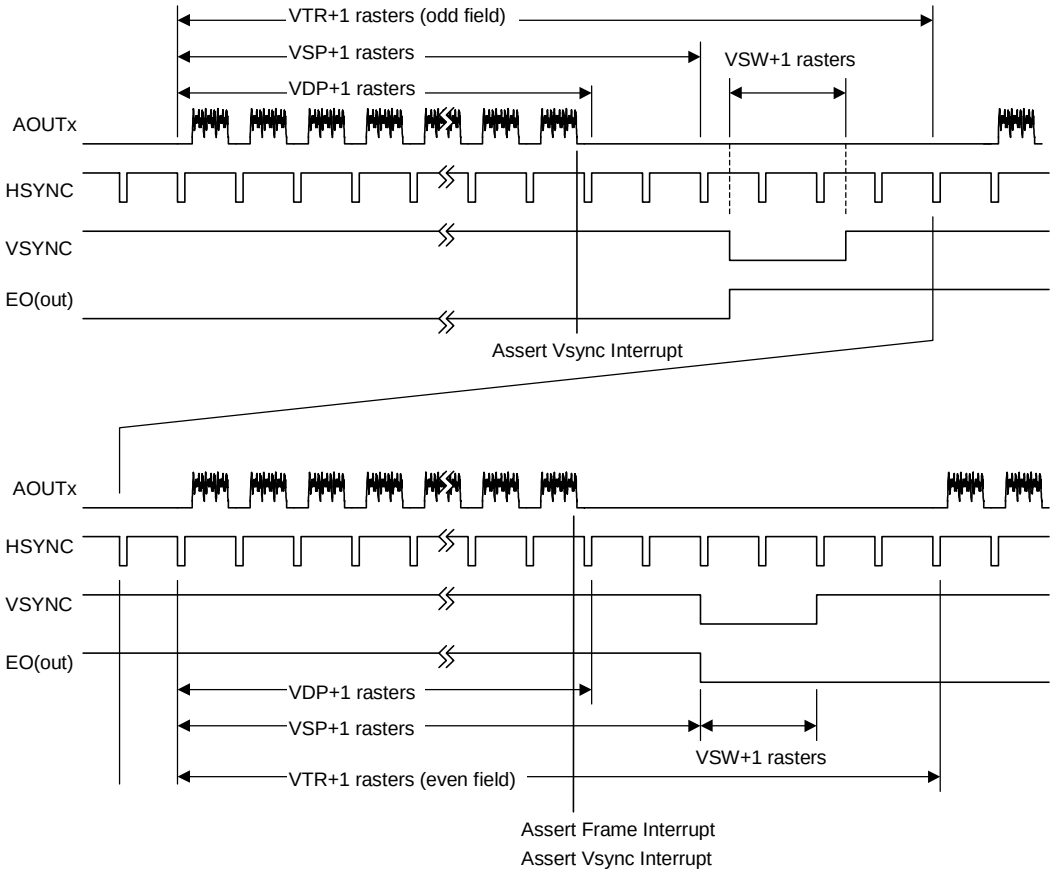
8.3.1 Non-interlaced Video Mode



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

The VSYNC/frame interrupt is asserted when display of the last raster ends. When updating display parameters, synchronize with the frame interrupt so no display disturbance occurs. Calculation for the next frame is started immediately after the vertical synchronization pulse is asserted, so the parameters must be updated by the time that calculation is started.

8.3.2 Interlaced Video Mode



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

Cautions

8.4 CPU Cautions

(1) Enable the hardware wait for the areas to which the MB86290A is linked. Set the software wait count to 1.
(2) When starting DMA by issuing an external request, do so after setting the transfer count register (DTCR) and mode setting register (DSUR) of the MB86290A to the same value as the CPU setting. In the V832 mode, there is no need to set DTCR.
(3) When MB86290A is read-/write-accessed from the CPU during DMA transfer, do not access the registers and memories related to DMA transfer. If these registers and memories are accessed, reading and writing of the correct value is not assured.
(4) In the SH mode, only the lowers 32 Mbytes are used (A[25] is not used), so do not access the uppers 32 Mbytes. When linking other devices to the uppers 32 Mbytes, create Chip Select for the MB86290A by using glue logic.
(5) Set DREQ (DMARQ) to detection.
(6) Set the SH-mode DACK/DRACK to high active output, V832-mode DMAAK to high active, and V832-mode TC to low active.

8.5 SH3 Mode

(1) When the RDY pin is low, it is in the wait state.
(2) DMA transfer in the single-address mode is not supported.
(3) DMA transfer in the dual-address mode supports the direct address transfer mode, but does not support the indirect address transfer mode.
(4) 16-byte DMA transfer in the dual-address mode is not supported.
(5) The $\overline{\text{INT}}$ signal is low active.

8.6 SH4 Mode

(1) When the RDY pin is low, it is in the ready state.
(2) At DMA transfer in the single-address mode, transfer from the main memory (SH-mode memory) to FIFO of the MB86290A can be performed, but transfer from the MB86290A to the main memory cannot be performed.
(3) DMA transfer in the single-address mode is performed in units of 32 bits or 32 bytes.
(4) SH4-mode 32-byte DMA transfer in the dual-address mode supports inter-memory transfer, but does not support transfer from memory to FIFO.
(5) The INT signal is low active.

8.7 V832 Mode

(1) When the RDY pin is low, it is in the ready state.
(2) Set the active level of DMAAK to high-active in V832 mode.
(3) DMA transfer supports the single transfer mode and demand transfer mode.
(4) The INT signal is high-active. Set the V832-mode registers to high-level trigger.

8.8 DMA Transfer Modes Supported by SH3, SH4, and V832

Table 8-1 Table of DMA Transfer Modes supported by SH3, SH4, and V832

	Single-address mode	Dual-address mode
SH3	SH 3 does not support the single-address mode.	SH3 supports the direct address transfer mode; it does not support the indirect address transfer mode. Transfer is performed in 32-bit units. SH3 supports the cycle steal mode and burst mode.
SH4	Transfer is performed in units of 32 bits or 32 bytes. SH4 supports the cycle steal mode and burst mode.	Transfer is performed in 32-bit units. Transfer to memory is performed in 32-byte units. SH4 supports transfer to FIFO. SH4 supports the cycle steal mode and burst mode.
V832		Transfer is performed in 32-bit units. V832 supports the single transfer mode and demand transfer mode.

9 Electrical Characteristics (Preliminary Target Specifications)

9.1 Absolute Maximum Ratings

Maximum Ratings

Parameter	Symbol	Maximum Rating	Unit
Supply voltage	VDDL *1 VDDH	-0.5 < VDDL < 3.0 -0.5 < VDDH < 4.0	V
Input voltage	VI VIV *2	-0.5 < VI < VDDH+0.5 (<4.0) -0.5 < VI < VDDH+4.0 (<6.0)	V
Output current	IO	+13 / -13	mA
Power current	IPOW	60	mA
Ambient temperature	TOP	0 < TOP < 70 (-40 < TOP < 85) *3	°C
Storage temperature	TST	-55 < TST < +125	°C

*1 Includes analog power supply and PLL power supply

*2 HSYNC, VSYNC, EO input

*3 Temperature extended version

9.2 Recommended Operating Conditions

9.2.1 Recommended Operating Conditions

Recommended Operating Conditions

Parameter	Symbol	Specifications			Unit
		Min.	Typ.	Max.	
Supply voltage	VDDL *1 VDDH	2.3 3.0	2.5 3.3	2.7 3.6	V
Input high voltage	VIH VIHV *2	2.0 2.0		VDDH+0.3 5.5	V
Input low voltage	VIL VILV *2	-0.3 -0.3		0.8 0.8	V
Input voltage to VREF	VREF	1.05	1.10	1.15	V
VRO External resistance	RVRO		2.7		k•
AOUT External resistance *3	RAOUT		75		•
ACOMP External capacitance *4	CACOMP		0.1		µF
Ambient temperature	TOP	0		70	°C

*1 Includes analog power supply and PLL power supply

*2 HSYNC, VSYNC, EO input

*3 AOUTR, AOUTG, AOUTB pins

*4 ACOMPR, ACOMPG, ACOMPB pins

Parameter	Symbol	Specifications			Unit
		Min.	Typ.	Max.	
Supply voltage	VDDL *1 VDDH	2.6 3.5	2.5 3.3	2.7 3.6	V
Ambient temperature	TA	–40		7.0	°C
Others	T				

9.2.2 Power-on Precautions

- There is no restriction on the order of power-on/power-off between VDDL and VDDH. However, do not supply only VDDH for more than a few seconds.
- Do not supply HSYNC, VSYNC and EC signals while the voltage supply is OFF. (See the recommended input voltage in the section on absolute maximum ratings.)
- After power-on, hold the S input at the 'L' level for at least 500 ns. Then, after setting the S-input to the 'H' level, hold the XRESET input at the 'L' level for at least 300 µs.

9.3 DC Characteristics

Condition: VDDL = 2.5 ±0.2 V, VDDH = 3.3 ±0.3 V, VSS = 0.0 V, Ta = 0-70˚C

Parameter	Symbol	Specifications			Unit
		Min.	Typ.	Max.	
Output high voltage[*1]	VOH	VDDH-0.2		VDDH	V
Output low voltage[*2]	VOL	0.0		0.2	V
Output high current	IOH1[*3] IOH2[*4] IOH3[*5]	-2.0 -4.0 -8.0			mA
Output low current	IOL1[*3] IOL2[*4] IOL3[*5]	2.0 4.0 8.0			mA
AOUT Output current[*6] Full scale Zero scale	IAOUT	 9.90 0	 10.42 2	 10.94 20	 mA µA
AOUT Voltage[*7]	VAOUT	-0.1		1.1	V
Input leakage current	IL			+5/-5	µA
Load capacitance	C			16	pF

*1 IOH = -100 µA

*2 IOL = 100 µA

*3 Output current of MD0-63, MDQM0-7

*4 Output current of all signals except *3 and *5 (not including analog signals)

*5 Output current of MCLKO

*6 Output current of AOUTR, AOUTG and AOUTB (VREF = 1.10 V, RVRO = 2.7 kΩ)
(The formula for full-scale output current calculation is (VREF/RVRO) x 25.575.)

*7 AOUTR, AOUTG and AOUTB pins

9.4 AC Characteristics

9.4.1 Host Interface

Clock

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
BCLKI Frequency	f_{BCLKI}				100	MHz
BCLKI H-width	t_{HBCLKI}		1			ns
BCLKI L-width	t_{LBCLKI}		1			ns

Host interface signals

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
Address set up time	t_{ADS}		3.0			ns
Address hold time	t_{ADH}		1.0			ns
BS Set up time	t_{BSS}		3.5			ns
BS Hold time	t_{BSH}		0.0			ns
CS Set up time	t_{CSS}		3.5			ns
CS Hold time	t_{CSH}		0.0			ns
RD Set up time	t_{RDS}		3.0			ns
RD Hold time	t_{RDH}		1.0			ns
WE Set up time	t_{WES}		3.0			ns
WE Hold time	t_{WEH}		1.0			ns
Write data set up time	t_{WDS}		5.0			ns
Write data hold time	t_{WDH}		1.0			ns
DTACK Set up time	t_{DAKS}		3.0			ns
DTACK Hold time	t_{DAKH}		1.0			ns
DRACK Set up time	t_{DRKS}		3.0			ns
DRACK Hold time	t_{DRKH}		1.0			ns
Read data delay time (for XRD)	t_{RDDZ}		4.0		8.5	ns
Read data delay time	t_{RDD}		4.0		9.5	ns
RDY Delay time (for XCS) SH	t_{RDYDZ}		3.0		9.0	ns
RDY Delay time (for XCS) V832	t_{RDYDZ}		3.0		8.5	ns
RDY Delay time	t_{RDYD}		3.5		7	ns
INT Delay time	t_{INTD}		-		10	ns
DREQ Delay time	t_{DQRD}		3.5		7	ns
MODE Hold time	t_{MODH}	*1			20	ns

*1 Hold time requirement for RESET release

9.4.2 Video Interface

Clock

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
CLK Frequency	f_{CLK}			14.32		MHz
CLK H-width	t_{HCLK}		25			ns
CLK L-width	t_{LCLK}		25			ns
DCLKI Frequency	f_{DCLKI}				67	MHz
DCLKI H-width	t_{HDCLKI}		5			ns
DCLKI L-width	t_{LDCLKI}		5			ns
DCLKO frequency	f_{DCLKO}				67	MHz

Input signals

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
HSYNC Input pulse width	$t_{WHSYNC0}$	*1	3			clock
	$t_{WHSYNC1}$	*2	3			clock
HSYNC Input set up time	t_{SHSYNC}	*2	10			ns
HSYNC Input hold time	t_{HHSYNC}	*2	10			ns
VSYNC Input pulse width	$t_{WHSYNC1}$		1			HSYNC period
EO Input set up time	t_{SEO}	*3	10			ns
EO Input hold time	t_{HEO}	*3	10			ns

*1 In PLL synchronization mode (CKS = 0), base clock output from internal PLL (period = 1/14*fCLK)

*2 In DCLKI synchronization mode (CKS = 1), base clock = DCLKI

*3 For VSYNC negation edge

Output signals

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
EO Output delay time	t_{DEO}	*4			10	ns
HSYNC Output delay time	t_{DHSYNC}				10	ns
VSYNC Output delay time	t_{DVSYNC}				10	ns
CSYNC Output delay time	t_{DCSYNC}				10	ns
GV Output delay time	t_{DGV}				10	ns

*4 EO output changes at timing of VSYNC assertion

9.4.3 Graphics Memory Interface

Clock

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
MCLKO Frequency	t_{MCLKO}				100	ns
MCLKO H-pulse width	t_{HMCLKO}		1			ns
MCLKO L-pulse width	t_{LMCLKO}		1			ns
MCLKI Delay	t_{DMCLKI}				100	MHz
MCLKI H-Frequency	t_{HMCLKI}		1			ns
MCLKI H-pulse width	t_{LMCLKI}		1			ns
	t_{OID}		1		4	ns

Input/Output signals

Parameter	Symbol	Condition	Specifications			Unit
			Min.	Typ.	Max.	
MA, MRAS, MCAS, MWE, CKE Setup time	t_{MADS}	*1	3.5			ns
MA, MRAS, MCAS, MWE, CKE Hold time	t_{MADH}	*1	1			ns
MDQM Data setup time	t_{MDQMDS}	*1	3.5			ns
MDQM Data hold time	t_{MDQMDH}	*1	1			ns
MD Output data setup time	t_{MDODS}	*1	3.5			ns
MD Output data hold time	t_{MDODH}	*1	1			ns
MD Input data setup time	t_{MDIDS}	*2	3			ns
MD Input data hold time	t_{MDIDH}	*2	1			ns

*1: Setup hold time for MCLKO

*2: Setup hold time for MCLKI

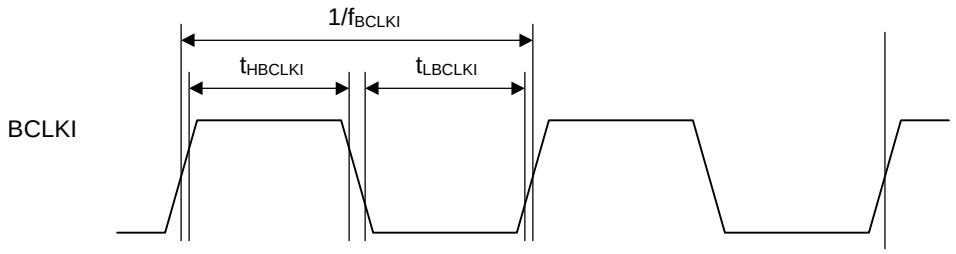
9.4.4 PLL Specifications

Parameter	Specifications	Description
Input frequency (typ.)	14.31818 MHz	
Output frequency	200.45452 MHz	x 14
Duty ratio	101.3~93.1%	H/L Pulse width ratio of PLL output
Jitter	180~-150ps	Frequency tolerant of two consecutive clock cycles

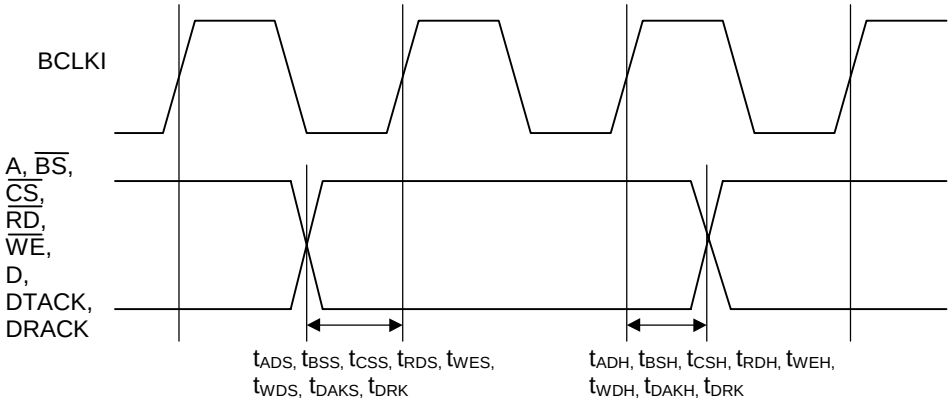
9.5 Timing Diagram

9.5.1 Host Interface

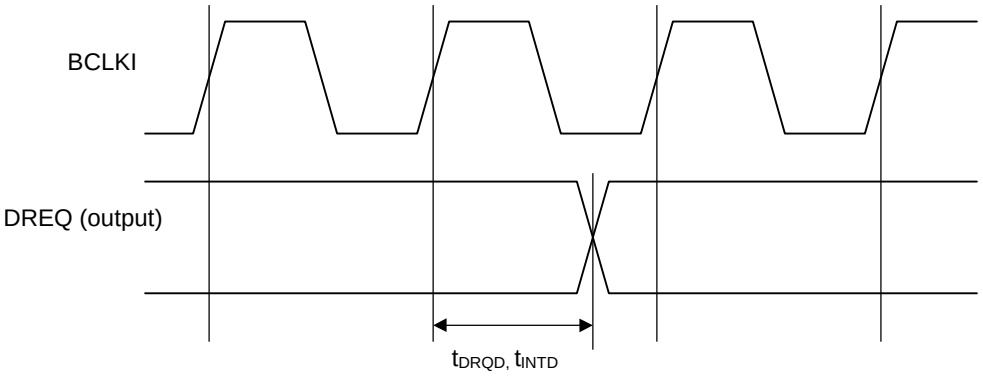
Clock



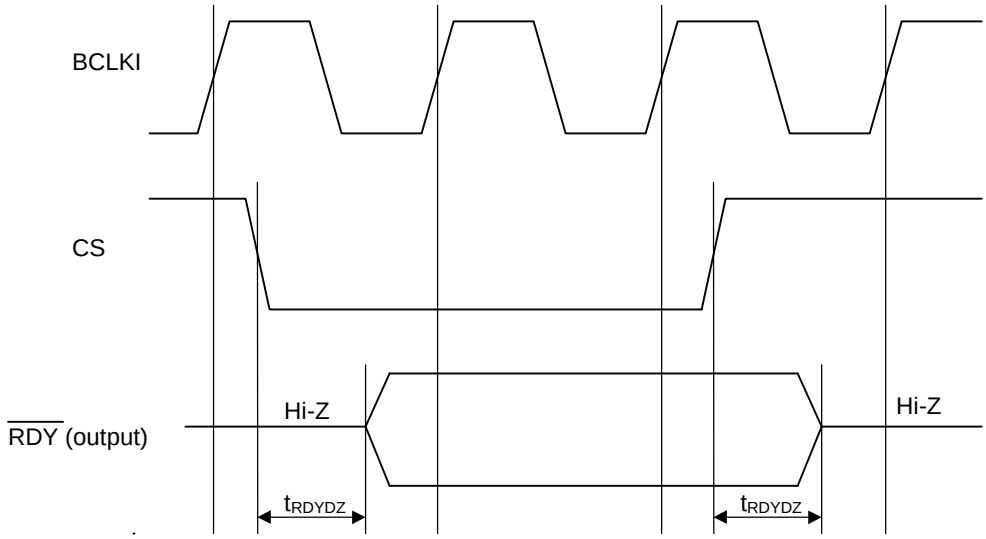
Input signal setup/hold times



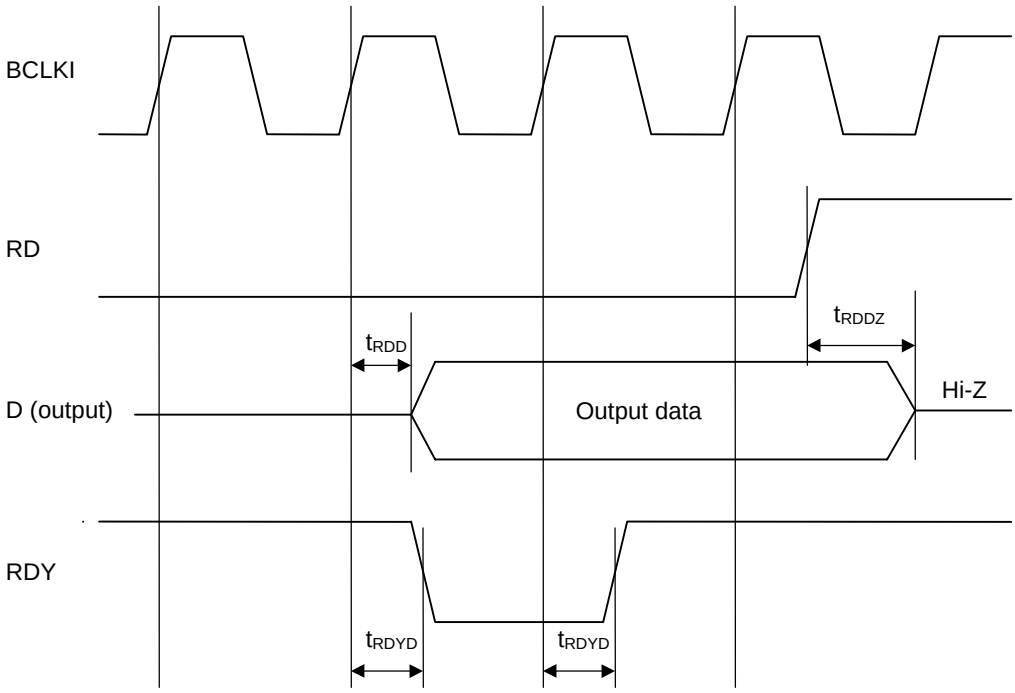
DREQ output delay times



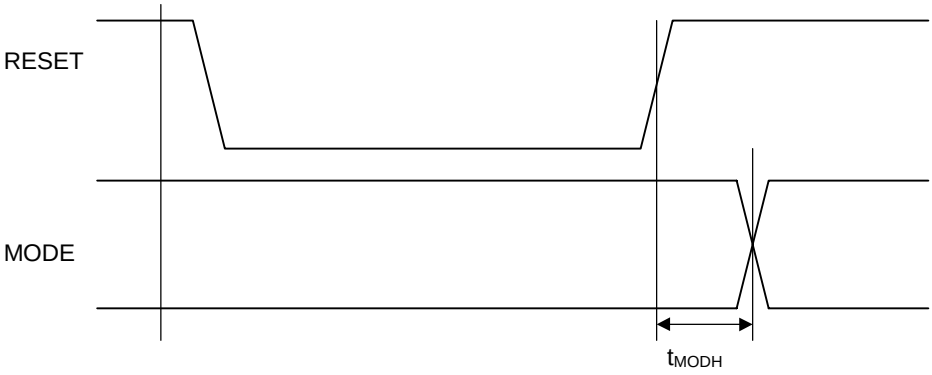
$\overline{\text{RDY}}$ Delay (for $\overline{\text{CS}}$)



RDY, D Output delay

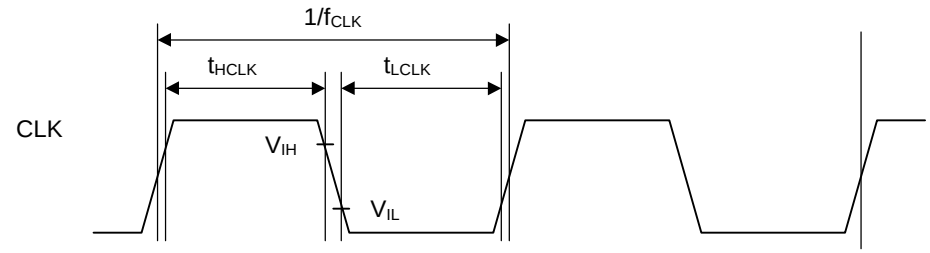


MODE Signal hold time

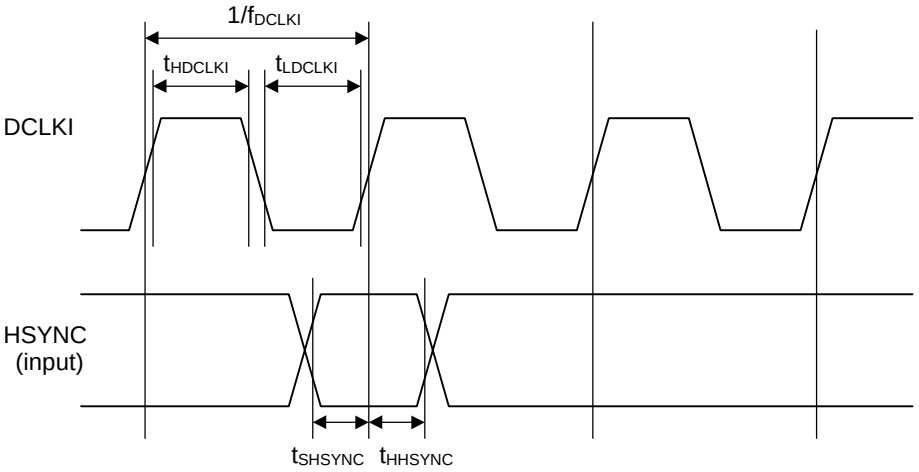


9.5.2 Video Interface

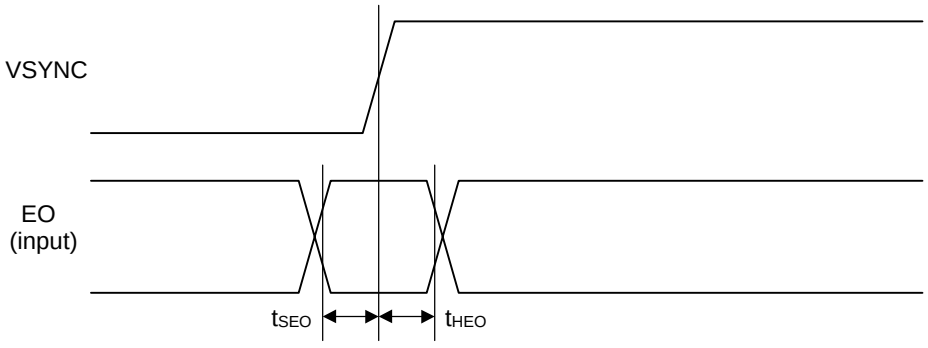
Clock



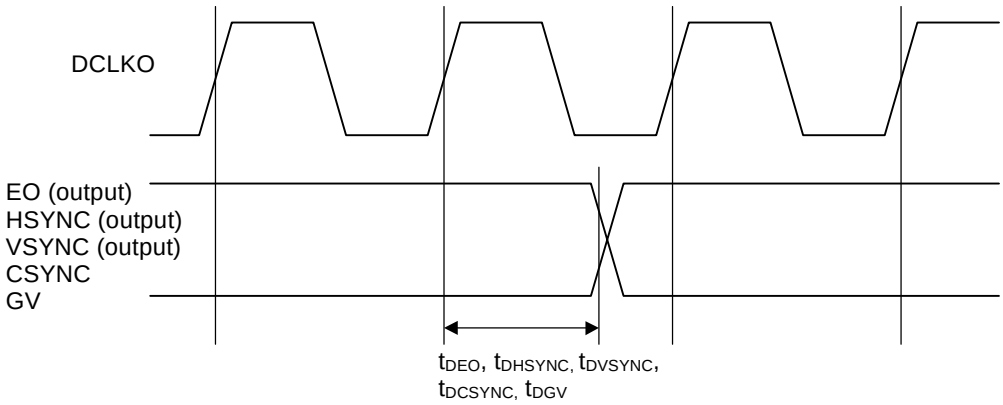
HSYNC Signal setup/hold



EO Signal setup/hold

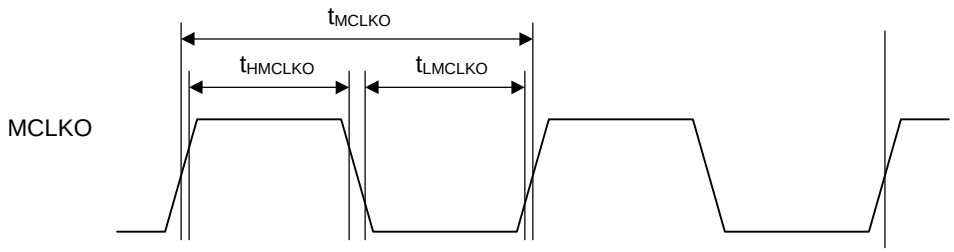


Output signal delay

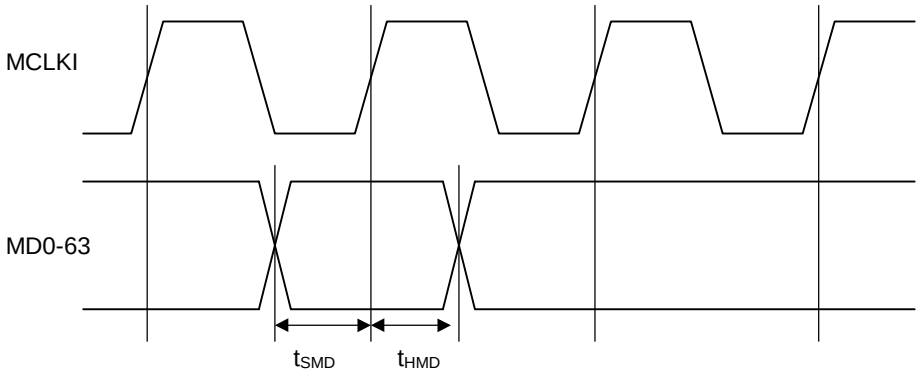


9.5.3 Graphics Memory Interface

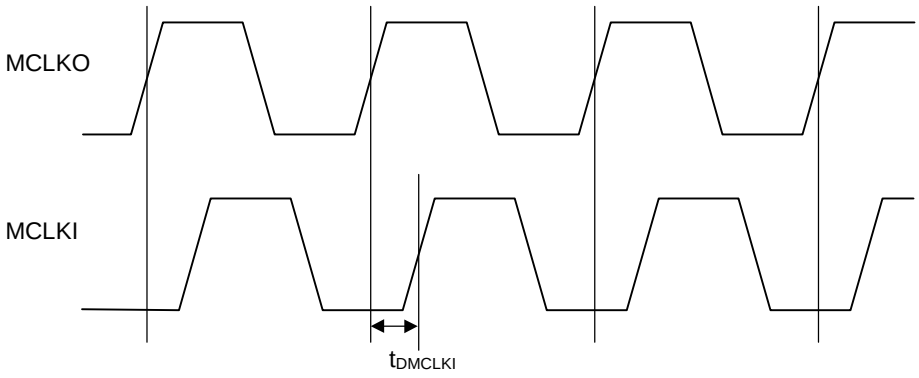
Clock



Input signal setup/hold times



MCLKI Signal delay



Output signal delay