MB86290 Series Graphics Driver
Users Manual
Rev.1.3

FUJITSU LIMITED

Overview

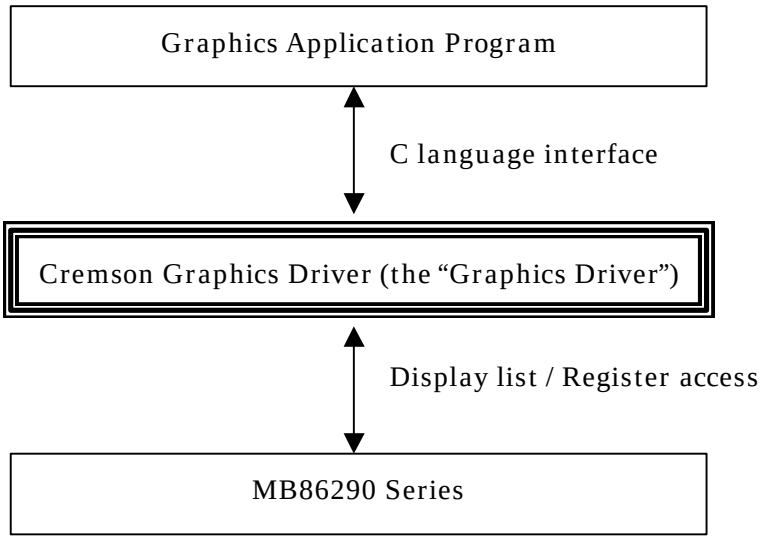
This manual describes about the operation mechanism of MB86290 Series graphics driver and application interface. For further understanding of the description, its need knowledge of MB86290 Series hardware specifications. Please refer to each “Graphics Controller Hardware Specifications” at the same time.

Index

1 MB86290 Series Graphics Driver Overview

1.1 Overview

The MB86290 Series Graphics Driver (the “Graphics Driver”) is a set of commands written in C language to assist graphics application programs utilizing the “MB86290 Series” Graphics Display Controller. The “Graphics Driver” is to interface between application programs (or graphical libraries) and hardware. By using this graphics driver, application programs can be made without concerning the code to access to hardware registers. This specification describes about the interface between application program and the “Graphics Driver”.



1.2 Configuration

1.2.1 Driver commands

Driver commands are to interface to the “Graphics Driver” from application programs. There are approximately 130 types of commands are supported, depending on each function, such as various primitive draw, display control, and so on. Application programs are able to use various hardware functions featured by the MB86290 Series, by calling these driver commands.

1.2.2 Display list

Display list is a series of commands to let MB86290 Series perform a drawing operation. The “Graphics Driver” dispatches a display list to the MB86290 Series and make it work.

1.2.3 System dependent commands

System dependent commands are to perform such operations depending on the target system or application program, as DMA operation procedure in the “Graphics Driver”. These commands must be designed by application designers for each target system, according to the interface defined by the “Graphics Driver”.

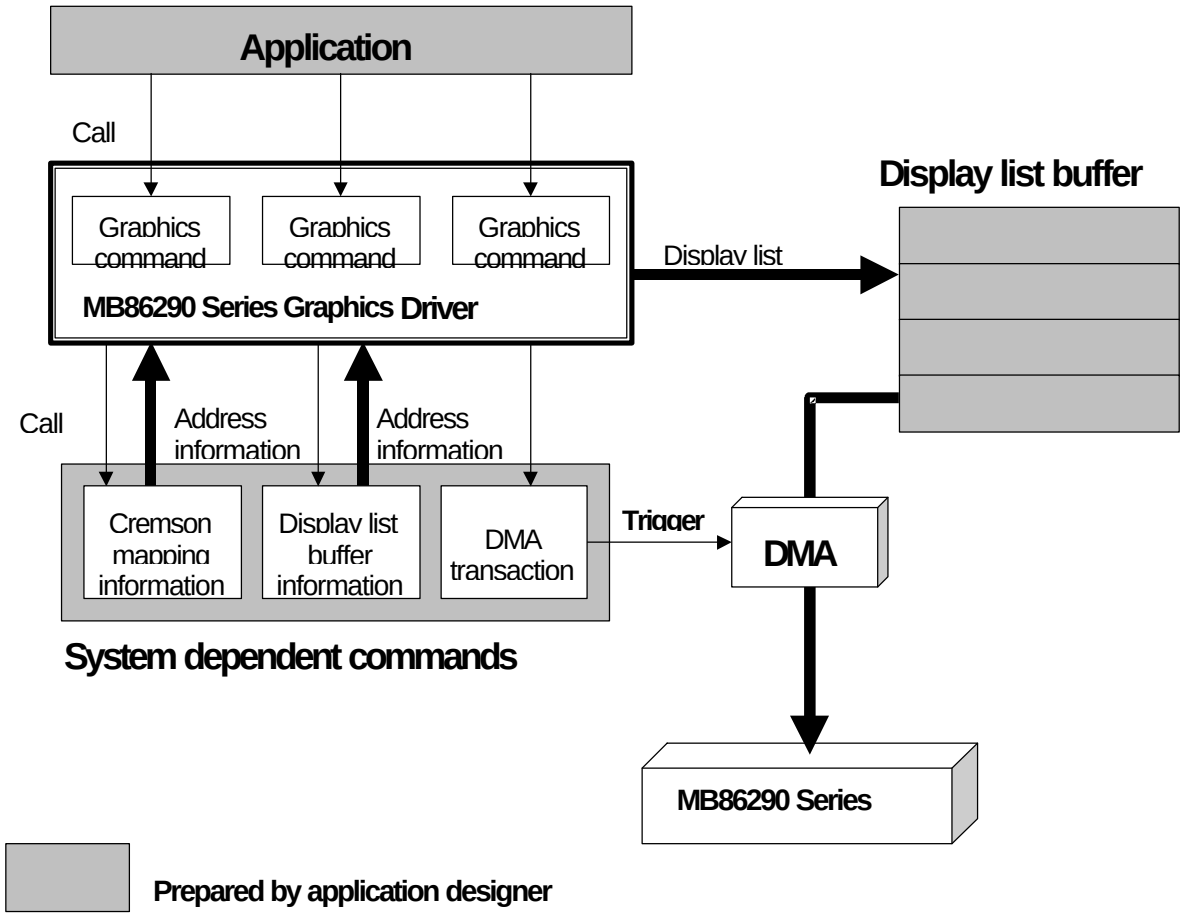
2 Operation of MB86290 Series Graphics Driver

2.1 Rendering Scheme

The following figure indicates the procedure of the “Graphics Driver”. Drawing operation of MB86290 Series is started by sending respective command and it’s arguments. To draw one object, multiple commands need to be sent. Typically a bunch of commands to be required to draw one object are set together and transferred consecutively in effective ways such as DMA. This bunch of commands is called display list.

The “Graphics Driver” stores a display list in display list buffer. Then according to the order from the application program, that display list is transferred to MB86290 Series.

The system dependent commands shown in the figure below handle such target system or application dependent operations, as acquisition of system information required by the “Graphics Driver” (such as the logical address of MB86290 Series resources, location of display list buffer, size of display list buffer and so on), trigger of DMA transactions. Because of the dependency to the target system and application program, the “Graphics Driver” only defines the interface to these system dependent commands. The actual commands are developed by each application designer according to these interface specifications.

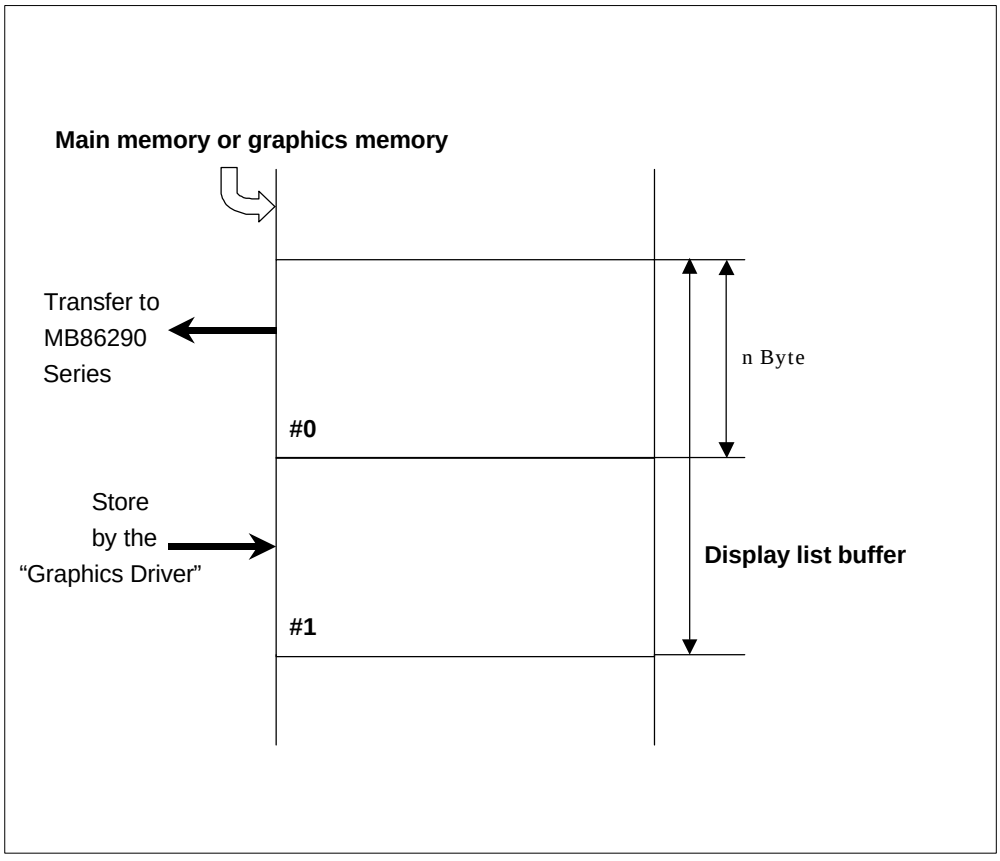


2.2 Management of Display List

The “Graphics Driver” store generated display lists to display list buffer and manages them. The display list buffer is a certain amount of memory area allocated on either the local memory of MB86290 Series (the graphics memory) or the host CPU memory. Acquisition of this area is done by application program. Created a display list is stored in the display list buffer till either obvious transfer order of it is made by the application program or no more open space will be available in the display list buffer. (The trigger timing of a display list transfer is referred to “2.3.2 Trigger of a display list transfer”).

Display list buffer is formed by 1 or 2 blocks of memory field. When it is formed in 2 blocks, each block is used alternatively. The biggest advantage of this 2 blocks configuration is that the host CPU can create a new display list while MB86290 Series executes rendering operations according to the current display list. To make this scheme, the display list needs to be transferred to MB86290 Series via DMA.

If the display list buffer is configured in 1 block, once the display list transfer is started, following display list creation cannot be started until current transfer operation will complete and the buffer will be open. In 2 blocks configuration, when 1 block is completely full of display lists, transfer is started. But prior to wait this transfer start, following display lists can be filled into the alternative block.



2.3 Transfer of Display List

2.3.1 Methods of display list transfer

Display lists are transferred to MB86290 Series by a system dependent command. For the display list transfer, the following three options are available. Depends on the target system configuration, each application designer should choose the most appropriate option:

- DMA transfer
- Display list read by MB86290 Series
- Display list write by host CPU

2.3.2 Trigger of display list transfer

Display list transfer is started by the following events:

- GdcFlush command call
- GdcSync command call
- GdcVFlush command call
- GdcVSync command call
- Not enough space available in the display list buffer to fill the display list to be generated at the execution of respective driver command

2.4 Sync Mode and Async Mode

The “Graphics Driver” has two operation modes, Sync mode and Async mode. In sync mode, regardless the method of the display list transfer, each driver command transfers it's generated display list to MB86290 Series immediately, and returns back to the application after the completion of MB86290 Series' rendering operations according to the display list. This mode is mainly used in the debug of application programs.

In async mode, each driver command returns back to the application right after it's set of display list to the display list buffer. Display list transfer is performed in the condition as described in 2.3.2 Trigger of the display list transfer. In this mode, display list creation by the “Graphics Driver”, the display list read by MB86290 Series, and it's execution work independently. Normally in application program execution, this mode is recommended. Switching of sync mode and async mode is performed by GdcExecMode command.

2.5 Operation of System Dependent Commands

The “Graphics Driver” performs target system/application dependent operation and acquires such information by calling system dependent commands. The system dependent commands must be designed by each application designer according to the command interface specified by the “Graphics Driver”. The system dependent commands handle the following operations (the command interface of the system dependent commands is referred to “7. System Dependent Commands”:

(1) Acquisition of the mapping address of MB86290 Series register areas

Get the address allocation information of MB86290 Series register areas and feed these information back to the “Graphics Driver” to access to various physical MB86290 Series registers.

(2) Sets the display list buffer

Informs the address allocation and size of the display list buffer created by the application program to the “Graphics Driver”.

(3) Display list transfer

Transfer display lists to MB86290 Series according to the selected transfer option.

3 For Application Program Development

This section describes mandatory operations and reminders in the application program development utilizing the “Graphics Driver”.

3.1 Mandatory Operations

3.1.1 Reads header file

The “Graphics Driver” includes the following header files (.h) to be used by the application programs. Whenever driver command is called, gdc.h must be included. Since the gdctypes.h is already included in the gdc.h, application program does not need to include it directly.

- gdc.h : statement of driver command prototype
- gdctypes.h : definition of the data type applied in the drivers

3.1.2 Gets display list buffer area

The display list buffer area must be acquired by the application program. The buffer size should be 32byte boundary. When DMA is applied to transfer display lists, the address allocation of the buffer area and the block size must be aware. The address allocation and the buffer size should be defined not to conflict any source address restrictions of the DMA controller (if any).

When DMA is adopted, always the source address is the top address of each display list buffer block (if the display list buffer is configured in 2 blocks, top address is for the 1st block. And the top address for the 2nd block is “top address+1/2 byte count of the total display list buffer area).

3.1.3 Creates system dependent commands

The system dependent commands should be designed according to the command interface specified in “7. System Dependent Commands”.

3.2 Reminder

3.2.1 Prohibition of re-entrant

The “Graphics Driver” is not configured to allow re-entrance. NOT to call the “Graphics Driver” from multiple tasks. If multiple tasks must call the “Graphics Driver” simultaneously, these calling conventions must be managed exclusively, and avoid driver command call from one task before the completion of the driver command operation called by the other task.

4 Driver Commands

4.1 System Control Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcQueryVersion	Version number check of the “Graphics Driver”	Y	Y
2	GdcInitialize	Graphics driver initialization	Y	Y
3	GdcGeoInitialize	Initialize geometry engine	N	Y
4	GdcFlush	Drawing by display lists (Async)	Y	Y
5	GdcSync	Drawing by display lists (Sync)	Y	Y
6	GdcVFlush	Vertical blanking interval palling (Async)	Y	Y
7	GdcVSync	Vertical blanking interval palling (Sync)	Y	Y
8	GdcGeoSync	Vertical blanking interval palling (Sync)	N	Y
9	GdcInterrupt	Interrupts request to host CPU	Y	Y
10	GdcGeoInterrupt	Interrupts request to host CPU from geometry engine	N	Y
11	GdcExecMode	Sets execution mode of display list operation	Y	Y
12	GdcSetDMAMode	Sets DMA mode	Y	Y
13	GdcGetFIFOStatus	Gets display list FIFO status	Y	Y
14	GdcGeoGetFIFOStatus	Gets geometry display list FIFO status	N	Y
15	GdcGetFIFORemain	Gets number of display list FIFO open entries	Y	Y
16	GdcGeoGetFIFORemain	Gets number of geometry display list FIFO open entries	N	Y
17	GdcGetFIFOErrorStatus	Gets display list FIFO error status	Y	Y
18	GdcGeoGetFIFOErrorStatus	Gets geometry display list FIFO error status	N	Y
19	GdcClearFIFOErrorStatus	Clear display list FIFO error status	Y	Y
20	GdcGeoClearFIFOErrorStatus	Clears geometry display list FIFO error status	N	Y

(*) Y : can be used

N : can not be used

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
21	GdcGetInterruptStatus	Gets interrupt status	Y	N
22	GdcGeoGetInterruptStatus	Gets interrupt status	N	Y
23	GdcClearInterruptStatus	Clears interrupt status	Y	N
24	GdcGeoClearInterruptStatus	Clears interrupt status	N	Y
25	GdcSetInterruptMask	Sets interrupt mask	Y	N
26	GdcGeoSetInterruptMask	Sets interrupt mask	N	Y
27	GdcSetMemoryMode	Sets memory interface mode	Y	Y
28	GdcSoftwareReset	Resets by software	Y	Y
29	GdcGetErrCode	Gets error code	Y	Y

(*) Y : can be used

N : can not be used

4.2 Display Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcDispClock	Sets display clock mode	Y	Y
2	GdcDispTiming	Sets display timing parameters	Y	Y
3	GdcDispTimingWindow	Sets W-layer display position	Y	Y
4	GdcDispDividePos	Sets border position of screen partition	Y	Y
5	GdcDispDimension	Sets display frame attribute	Y	Y
6	GdcDispOn	Asserts video signal output	Y	Y
7	GdcDispOff	Negates video signal output	Y	Y
8	GdcDispLayerOn	Asserts screen display	Y	Y
9	GdcDispLayerOff	Negates screen display	Y	Y
10	GdcDispPos	Sets display start position	Y	Y
11	GdcDispDoFlip	Flips display bank	Y	Y
12	GdcOverlayPriorityMode	Sets overlay display mode	Y	Y
13	GdcOverlayBlend	Sets blend parameter for overlay blend	Y	Y

(*) Y : can be used

N: can not be used

4.3 Color Control Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcColorPalette	Sets palette colors	Y	Y
2	GdcColorTransparent	Sets transparent color	Y	Y
3	GdcColorZeroMode	Sets color code 0 mode	Y	Y
4	GdcChromaKeyMode	Sets Chroma-key mode	Y	Y
5	GdcColorKey	Sets key color for Chroma-key	Y	Y

(*) Y : can be used

N: can not be used

4.4 Cursor Control Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcCursorAddress	Sets cursor pattern memory address	Y	Y
2	GdcCursorPattern	Sets cursor pattern	Y	Y
3	GdcCursorDisplay	Controls cursor display	Y	Y
4	GdcCursorPos	Sets cursor display position	Y	Y
5	GdcCursorPriority	Sets cursor display priority mode	Y	Y
6	GdcCursorColorTransparent	Sets cursor transparent color	Y	Y
7	GdcCursorColorZeroMode	Sets cursor color code 0 mode	Y	Y

(*) Y : can be used

N: can not be used

4.5 Draw Frame Control Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcDrawDimension	Sets draw frame	Y	Y
2	GdcBufferCreateZ	Sets Z buffer start address	Y	Y
3	GdcBufferCreateC	Sets start address of polygon draw flag buffer	Y	Y
4	GdcBufferClearZ	Clears Z buffer	Y	Y
5	GdcBufferClearC	Clears polygon draw flag buffer	Y	Y
6	GdcDrawClipFrame	Sets draw clip border	Y	Y

(*) Y : can be used

N: can not be used

4.6 Primitive Draw Control Commands for Device Coordinates

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcPrimType	Sets drawing procedure	Y	Y
2	GdcPrimEnd	Completes drawing procedure	Y	Y
3	GdcTexCoord2D	Sets coordinates of 2D texture	Y	Y
4	GdcTexCoord2Df		Y	Y
5	GdcTexCoord2DNf		Y	Y
6	GdcTexCoord3D	Sets coordinates of 3D texture	Y	Y
7	GdcTexCoord3Df		Y	Y
8	GdcTexCoord3DNf		Y	Y
9	GdcDrawVertex2D	Sets coordinates of 2D vertex	Y	Y
10	GdcDrawVertex2Di		Y	Y
11	GdcDrawVertex3D	Sets coordinates of 3D vertex	Y	Y
12	GdcDrawVertex3Df		Y	Y
13	GdcDrawPrimitive	Draws multiple 3D triangles	Y	Y

(*) Y : can be used

N: can not be used

4.7 Primitive Draw Control Commands for Object Coordinates

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcGeoPrimType	Sets drawing procedure	N	Y
2	GdcGeoPrimEnd	Completes drawing procedure	N	Y
3	GdcGeoDrawVertex2D	Sets XY coordinates of vertex	N	Y
4	GdcGeoDrawVertex2Df			
5	GdcGeoDrawVertex2Di			
6	GdcGeoDrawVertex3D	Sets XYZ coordinates of vertex	N	Y
7	GdcGeoDrawVertex3Df			
8	GdcGeoDrawVertex3Di			
9	GdcGeoTexCoord2D	Sets texture coordinates	N	Y
10	GdcGeoTexCoord2Df			
11	GdcGeoTexCoord2DN			
12	GdcGeoTexCoord2DNf			
13	GdcVertexColor3f	Sets color of vertex	N	Y
14	GdcVertexColor32			

(*) Y : can be used

N: can not be used

4.8 Draw Attribute Control Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcColor	Sets vertex color/foreground color	Y	Y
2	GdcColorI		Y	Y
3	GdcBackColor	Sets background color	Y	Y
4	GdcBackColorI		Y	Y
5	GdcSetAttrMisc	Sets draw attribute	Y	Y
6	GdcSetAttrLine	Sets line draw attribute	Y	Y
7	GdcSetAttrSurf	Sets surface draw attribute	Y	Y
8	GdcGeoSetAttrSurf	Sets surface draw attribute for object coordinates	N	Y
9	GdcSetAttrTexture	Sets texture mapping attribute	Y	Y
10	GdcSetAttrBlt	Sets BitBlt attribute (Sets blend mode)	Y	Y
		Sets BitBlt attribute (Sets transparent mode)	N	Y
11	GdcSetAlpha	Sets alpha blend ratio	Y	Y
12	GdcSetLinePattern	Sets broken line pattern	Y	Y
13	GdcSetTextureBorder	Sets texture border color	Y	Y
14	GdcSetRop	Sets logical calculation mode	Y	Y

(*) Y : can be used

N: can not be used

4.9 Attribute Control Commands for Object Coordinate

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcGeoSetAttrMisc	Sets miscellaneous attribute	N	Y
2	GdcGeoLoadMatrix	Sets matrix	N	Y
3	GdcGeoLoadMatrixf			
4	GdcGeoNdcDcViewportCoef	Sets coefficients of NdcDc transformation for XY	N	Y
5	GdcGeoNdcDcViewportCoeff			
6	GdcGeoNdcDcDepthCoef	Sets coefficients of NdcDc transformation for Z	N	Y
7	GdcGeoNdcDcDepthCoeff			
8	GdcGeoViewVolumeXYClip	Sets view volume boundary for XY	N	Y
9	GdcGeoViewVolumeXYClipf			
10	GdcGeoViewVolumeZClip	Sets view volume boundary for Z	N	Y
11	GdcGeoViewVolumeZClipf			
12	GdcGeoViewVolumeWminClip	Sets view volume boundary for W	N	Y
13	GdcGeoViewVolumeWminClipf			

(*) Y : can be used

N: can not be used

4.10 Texture Pattern Management Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcTextureMemoryMode	Sets texture memory mode	Y	Y
2	GdcTextureLoadInt	Loads texture/tile pattern to internal buffer	Y	Y
3	GdcTextureLoadExt	Loads texture pattern to graphics memory	Y	Y
4	GdcTextureDimension	Sets texture information	Y	Y
5	GdcBltTexture	Loads Blt texture to internal texture buffer for MB86290A	Y	N
6	GdcGeoBltTexture	Loads Blt texture to internal texture buffer for MB86291	N	Y

(*) Y: can be used

N: can not be used

4.11 Binary Pattern Draw Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcBitPatternDraw	Draws binary pattern	Y	Y
2	GdcBitPatternMode	Sets enlarge/shrink mode	Y	Y

(*) Y: can be used

N: can not be used

4.12 BLT Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcBltCopy	Copies BitBlt pattern within current draw frame	Y	Y
2	GdcBltCopyAlt	Copies BitBlt pattern between any draw frame	Y	Y
3	GdcBltCopyAltSync		Y	Y
4	GdcBltDraw	Draws BitBlt pattern	Y	Y
5	GdcBltFill	Fills BitBlt field	Y	Y
6	GdcBltColorTransparent	Sets transparent color of transparent BitBlt	N	Y

(*) Y : can be used

N: can not be used

4.13 Video Capture Control Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcCapSetVideoCaptureMode	Sets mode of video capture	N	Y
2	GdcCapGetErrorStatus	Gets error status of video capture	N	Y
3	GdcCapClearErrorStatus	Clears error status of video capture	N	Y
4	GdcCapSetVideoCaptureBuffer	Sets video capture buffer	N	Y
5	GdcCapSetImageArea	Sets range of image	N	Y
6	GdcCapSetWindowMode	Sets w-layer mode	N	Y
7	GdcCapSetVideoCaptureScale	Sets scale of video capture	N	Y
8	GdcCapSetAttrMisc	Sets attribute of video capture	N	Y
9	GdcCapSetInputDataCountNTSC	Sets number of video capture data for NTSC	N	Y
10	GdcCapSetInputDataCountPAL	Sets number of video capture data for PAL	N	Y

(*) Y : can be used

N: can not be used

4.14 System Dependent Commands

No.	Command name	Function	Graphics LSI (*)	
			MB86290A	MB86291
1	GdcSetDisplayListBuffer	Sets display list buffer	Y	Y
2	GdcFlushDisplayList	Transfers a display list	Y	Y
3	GdcGetHostRegisterAddress	Gets host interface register area address	Y	Y
4	GdcGetDispRegisterAddress	Gets display control register area address	Y	Y
5	GdcGetDrawRegisterAddress	Gets draw control register area address	Y	Y

(*) Y : can be used

N: can not be used

5 Data Format

The data types specified by the “Graphics Driver” are described as follows:

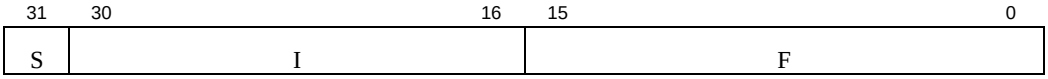
5.1 Data Type

Format	Description
GDC_FIXED32	32bit signed fixed point (16bit integer and 16bit fraction)
GDC_FIXED_SCALE	16bit unsigned fixed point for Capture Scale (5bit integer and 11bit fraction)
GDC_SFLOAT	32bit single precision float (IEEE754 compliant)
GDC_ULONG	32bit unsigned integer
GDC_LONG	32bit signed integer
GDC_USHORT	16bit unsigned integer
GDC_SHORT	16bit signed integer
GDC_UCHAR	8bit signed integer
GDC_COLOR32	32bit unsigned integer (32bit color format)
GDC_COL32	32bit unsigned integer (palette color format)
GDC_COL16	16bit unsigned integer (16bit color format)
GDC_COL8	8bit unsigned integer (8bit color format)
GDC_LPPCOL32	Pointer for GDC_COL32 format data
GDC_LPCOL16	Pointer for GDC_COL16 format data
GDC_LPCOL8	Pointer for GDC_COL8 format data
GDC_LPLONG	Pointer for GDC_LONG format data
GDC_LPBINIMAGE	Pointer for 32bit unsigned integer data (binary pattern data)
GDC_VERTEX	GDC_SFLOAT format vertex data structure
GDC_BOOL	True/false

5.2 Data Structure

5.2.1 GDC_FIXED32 [32 bit fixed point]

A fixed-point data with sign described in sign 1bit, integer 15bit, fraction 16bit.



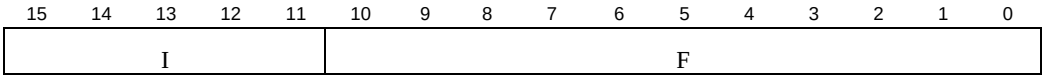
S: Sign(1bit)

I: Integer(15bit)

F: Fraction(16bit)

5.2.2 GDC_FIXED_SCALE [Capture scale format]

A capture scale data described in integer 5bit, fraction 11bit.
It used by GdcCapSetVideoCaptureScale.



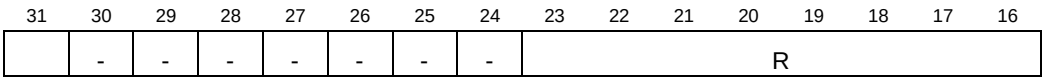
I: Integer(5bit)

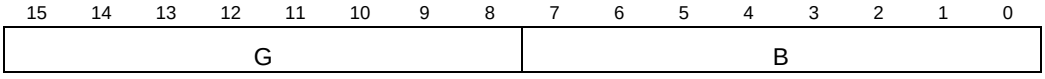
F: Fraction(11bit)

5.2.3 GDC_COLOR32 [32bit color]

A color data described in 8bit per R, G and B respectively.

It used by GdcVertexColor32.

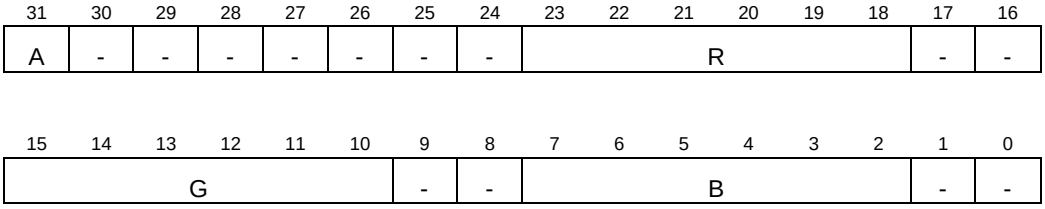




R,G,B: color bit(8bit)

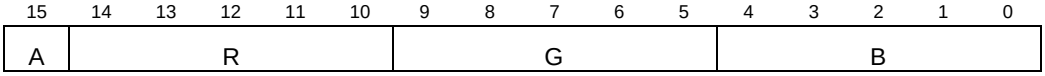
5.2.4 GDC_COL32 [Palette color format]

A color data described in 6bit per R, G and B respectively. For C layer palette, bit 31 is an alpha bit.



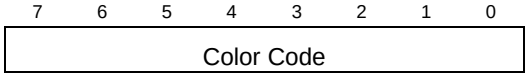
5.2.5 GDC_COL16 [16bit color format]

A color data described in 5bit per R, B and G respectively. When this color data format is applied to texture, bit15 is used as an alpha bit.



5.2.6 GDC_COL8 [8bit color format]

A color index code in 8bit.



5.2.7 GDC_VERTEX [GDC_SFLOAT format vertex data structure]

A structure data is packed in vertex coordinates, texture coordinates and RGB value.

It used by GdcDrawPrimitive.

GDC_VERTEX Structure

Parameter	Description
GDC_SFLOAT x	X coordinates of vertex
GDC_SFLOAT y	Y coordinates of vertex
GDC_SFLOAT z	Z coordinates of vertex
GDC_SFLOAT r	R value of vertex color
GDC_SFLOAT g	G value of vertex color
GDC_SFLOAT b	B value of vertex color
GDC_SFLOAT u	U coordinates of vertex texture
GDC_SFLOAT v	V coordinates of vertex texture
GDC_SFLOAT rw	Reciprocal W coordinates of vertex texture
Long work	Reserved

6 Driver Command Reference

6.1 System Control Commands

6.1.1 GdcQueryVersion [Version number check]

Format	void GdcQueryVersion (int *major, int *minor)	
Parameter	major	Major version number
	minor	Minor version number
Return value	None	
Description	Indicates current version number of the “Graphics Driver”.	
	This command can be used by all graphics LSI.	

6.1.2 GdcInitialize [Graphics driver initialization]

Format	int GdcInitialize(void)	
Parameter	None	
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Initialize the “Graphics Driver”.	
	This command can be used by all graphics LSI.	

6.1.3 GdcGeoInitialize [Initialize geometry engine]

Format	void GdcGeoInitialize(void)
Parameter	None
Return value	None
Description	Initializes internal resources in the geometric engine.
	When using GdcGeo* command, be sure to call this command after initializing drivers.
	This command is only for MB86291.

6.1.4 GdcFlush [Drawing by display lists (Async)]

Format: void GdcFlush(void)

Parameter: None

Return value: None

Description: Transfers a display list in the display list buffer to MB86290 Series. If DMA is applied, this command is completed without waiting for the end of the list transfer. If CPU writes the display list to MB86290 Series, this command is completed after the end of the list transfer.
This command can be used by all graphics LSI.

6.1.5 GdcSync [Drawing by display lists (Sync)]

Format: void GdcSync(void)

Parameter: None

Return value: None

Description: Transfers a display list in the display list buffer to MB86290 Series and returns to application after MB86290 Series will complete all the display list operation.
This command can be used by all graphics LSI.

6.1.6 GdcVFlush [Vertical blanking interval palling (Async)]

Format: void GdcVFlush(void)

Parameter: None

Return value: None

Description: Transfers a display list after attaching a command for waiting VSYNC (Sync command) to the end of it.

When the Sync command is executed, MB86290 Series synchronize the next operation with the vertical blanking interval. By means of this function, disorder of display caused by flipping can be avoided when flipping of draw frame (GdcDispDoFlip command).

Similar to the GdcFlush command, this command does not wait for the completion of MB86290 Series' all display list operations. Therefore, the completion of the Sync command is not guaranteed after returning from this command.

In order to detect the completion of the Sync command, use an interrupt for termination of drawing and so on. The interrupt can be issued by executing GdcInterrupt before this GdcVFlush command.

This command can be used by all graphics LSI.

Note: This command dose not guarantee the punctual synchronous ness with the vertical blanking interval. In order to detect the punctual vertical blanking interval, use VSYNC interrupt.

6.1.7 GdcVSync [Vertical blanking interval palling (Sync)]

Format: void GdcVSync(void)

Parameter: None

Return value: None

Description: Transfers a display list after attaching a command for waiting VSYNC (Sync command) to the end of it.

This command waits for the completion of MB86290 Series' all display list operations and then returns. Right after the call of this command, MB86290 Series' operation is synchronized with the vertical blanking interval. By means of this function, disorder of display caused by flipping can be avoided when flipping of draw frame with GdcDispDoFlip command.

This command can be used by all graphics LSI.

6.1.8 GdcGeoSync [Vertical blanking interval palling (Sync)]

Format void GdcGeoSync (void)

Parameter None

Return value None

Description Stops processes and waits until a vertical blanking begin. The timing of synchronizing with a vertical blanking by a geometry engine differs from a setup engine's one since they use different FIFOs respectively. When a vertical blanking should be synchronized with a geometry engine, this command must be use. When it should be synchronized with a setup engine, the GdcVFlush or GdcVSync command must be used.

This command can be used by all graphics LSI.

6.1.9 GdcInterrupt [Interrupt request to host CPU]

Format void GdcInterrupt(void)

Parameter None

Return value None

Description Generates an interrupt request. When the “Graphics Driver” works in sync mode, an interrupt request is generated immediately after the execution of this command. In async mode, a command to generate an interrupt request is put in a display list.

This command can be used by all graphics LSI.

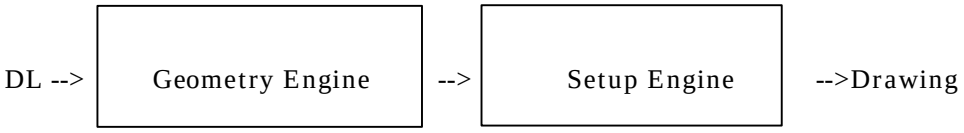
6.1.10 GdcGeoInterrupt [Interrupts request to host CPU from geometry engine]

Format void GdcGeoInterrupt (void)

Parameter None

Return value None

Description Generates an interrupt request from the geometry engine. MB86290 series consists of two blocks internally, that is, a geometry engine and a setup engine. Since they use different FIFOs respectively, the timing of generating an interrupt request by a geometry engine differs from a setup engine's one.



When generating an interrupt request from a geometry engine, this command must be used. When generating from a setup engine, GdcInterrupt must be used.
This command is only for MB86291.

6.1.11 GdcExecMode [Sets execution mode]

Format void GdcExecMode (GDC_UCHAR sync)

Parameter sync Sync/async mode celection

GDC_EXECMODE_SYNC	Sync mode
GDC_EXECMODE_ASYNC	Async mode

Return value None

Description Sets operation mode of the display list execution.
This command can be used by all graphics LSI.

6.1.12 GdcSetDMAMode [Sets DMA mode]

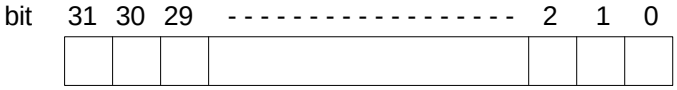
Format	int GdcSetDMAMode(int tran_unit, int bus_mode, int address_mode)		
Parameter	tran_unit	Unit of DMA transfer	
		GDC_DMA_TRANUNIT_4	4 byte
		GDC_DMA_TRANUNIT_32	32 byte
	bus_mode	Bus mode	
		GDC_DMA_BUSMODE_CYCLE	Cycle steal mode
		GDC_DMA_BUSMODE_BURST	Burst mode
	address_mode	Address mode of external DMA request	
		GDC_DMA_ADDRMODE_DUAL	Dual address mode
		GDC_DMA_ADDRMODE_SINGLE	Single address mode
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	
Description	Sets DMA transfer mode to DSU (DMA Set Up) register.		
	This command can be used by all graphics LSI.		

6.1.13 GdcGetFIFOStatus [Gets display list FIFO status]

Format GDC_ULONG GdcGetFIFOStatus(void)

Parameter None

Return value Display list FIFO status (IFSR register value) in the following format:



bit 0: Valid data exists in DL FIFO Yes=0, No=1

bit 1: DL FIFO is not full Yes=0, No=1

bit 2: More than half entries of DL FIFO are empty Yes=0, No=1

Description Reads IFSR (Input FIFO Status Register) and returns current display list FIFO status.

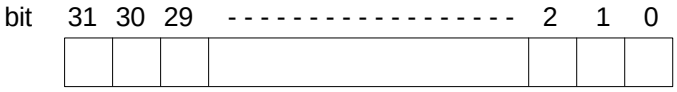
This command can be used by all graphics LSI.

6.1.14 GdcGeoGetFIFOStatus [Gets geometry display list FIFO status]

Format GDC_ULONG GdcGeoGetFIFOStatus(void)

Parameter None

Return value Display list FIFO status in the following format:



bit0 : Valid data exists in geometry DL FIFO Yes=0, No =1

bit1 : Geometry DL FIFO is not full Yes =0, No =1

bit2 : More than half entries of geometry DL FIFO are empty Yes =0, No =1

Description Returns current Geometry display list FIFO status (for test).

This command can be used by all graphics LSI.

6.1.15 GdcGetFIFORemain [Gets number of display list FIFO open entries]

Format	GDC_ULONG GdcGetFIFORemain(void)
Parameter	None
Return value	Number of open entries in the display list FIFO
Description	Reads IFCNT (Input FIFO CouNTer) register and returns the number of open entries in the display list FIFO. This command can be used by all graphics LSI.

6.1.16 GdcGeoGetFIFORemain [Gets number of geometry display list FIFO open entries]

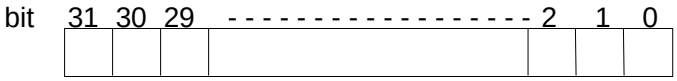
Format	GDC_ULONG GdcGeoGetFIFORemain(void)
Parameter	None
Return value	Number of open entries in the geometry display list FIFO
Description	Returns the number of open entries in the geometry display list FIFO (for test). This command is only for MB86291.

6.1.17 GdcGetFIFOErrorStatus [Gets display list FIFO error status]

Format GDC_ULONG GdcGetFIFOErrorStatus(void)

Parameter None

Return value Display list FIFO error status (IFSR register value) in the following format:



bit 0: Command error No=0, Yes=1

bit 1: Packet error No=0, Yes=1

bit 2: FIFO overflow No=0, Yes=1

Description Reads EST (Error Status Register) and returns display list FIFO error status.

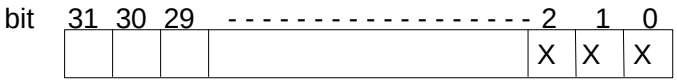
This command can be used by all graphics LSI.

6.1.18 GdcGeoGetFIFOErrorStatus [Gets geometry display list FIFO error status]

Format GDC_ULONG GdcGeoGetFIFOErrorStatus(void)

Parameter None

Return value Display list FIFO error status in the following format:



bit0 : Geometry command error No =0, Yes =1

bit1 : Geometry packet error No =0, Yes =1

bit2 : Geometry FIFO overflow No =0, Yes =1

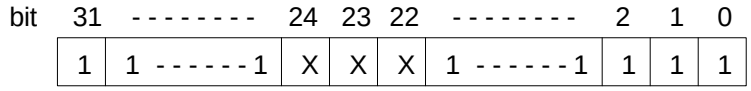
Description Returns display list FIFO error status (for test).

This command is only for MB86291.

6.1.19 GdcClearFIFOErrorStatus [Clears display list FIFO error status]

Format void GdcClearFIFOErrorStatus (GDC_ULONG clear)

Parameter clear Clear pattern (shown below)



bit 22: Command error Clear=0, Hold=1

bit 23: Packet error Clear=0, Hold=1

bit 24: FIFO overflow Clear=0, Hold=1

All other bits: 1

Return value None

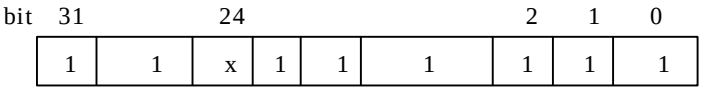
Description Clears the error event indicated by 22-24 bits in CTR (ConTrol Register) by the clear pattern specified as above. To clear an error event, respective bit in the clear pattern for that event is set to 0 and all other bits are set to 1.

This command can be used by all graphics LSI.

6.1.20 GdcGeoClearFIFOErrorStatus [Clears geometry display list FIFO error status]

Format void GdcGeoClearFIFOErrorStatus(GDC_ULONG clear)

Parameter clear Clear pattern (shown below)



bit24 : Geometry FIFO overflow Clear =0, Hold =1

All other bits: 1

Return value None

Description Clears the error event indicated by 24 bits in GCTR (Geometry Control Register) by the clear pattern specified as above. To clear an error event, respective bit in the clear pattern for that event is set to 0 and all other bits are set to 1.

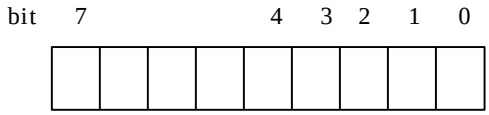
This command is only for MB86291.

6.1.21 GdcGetInterruptStatus [Gets interrupt status]

Format GDC_UCHAR GdcGetInterruptStatus (void)

Parameter None

Return value Interrupts status (IST register value) in the following format:



bit0: Command execution error interrupt	Yes=1, No=0
bit1: Command complete interrupt	Yes=1, No=0
bit2: VSYNC interrupt	Yes=1, No=0
bit3: Frame sync interrupt	Yes=1, No=0
bit4: External sync error interrupt	Yes=1, No=0

Description Reads IST (Interrupt Status Register) and return interrupt status.

This command is only for MB86290A.

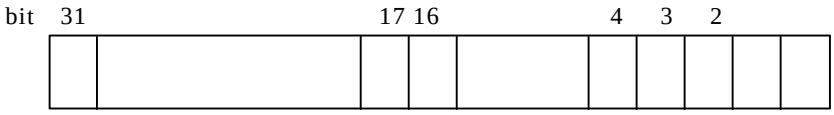
When graphics LSI is MB86291, GdcGeoGetInterruptStatus must be used.

6.1.22 GdcGeoGetInterruptStatus [Gets interrupt status]

Format GDC_ULONG GdcGeoGetInterruptStatus(void)

Parameter None

Return value Interrupts status (IST register value) in the following format:



bit2	: VSYNC interrupt	Yes =1,	No =0
bit3	: Frame sync interrupt	Yes =1,	No =0
bit4	: External sync error interrupt	Yes =1,	No =0
bit16	: Command execution error interrupt	Yes =1,	No =0
bit17	: Command complete interrupt	Yes =1,	No =0

Description Reads IST (Interrupt Status Register) and return interrupt status.

This command is only for MB86291.

When graphics LSI is MB86290A, GdcGetInterruptStatus must be used.

6.1.23 GdcClearInterruptStatus [Clears interrupt status]

Format void GdcClearInterruptStatus (GDC_UCHAR clear)

Parameter clear Clears pattern (shown below)

bit	7			4	3	2	1	0
	1	1	1	x	x	x	x	x

bit0: Command execution error interrupt Clear=0, Hold=1
bit1: Command complete interrupt Clear=0, Hold=1
bit2: VSYNC interrupt Clear=0, Hold=1
bit3: Frame sync interrupt Clear=0, Hold=1
bit4: External sync error interrupt Clear=0, Hold=1
All other bits: 1

Return value None

Description Clears the interrupt event indicated by 0-4 bits in ISR (Interrupt Status Register) by the clear pattern specified as above. To clear an interrupt event, respective bit in the clear pattern for that event is set to 0 and all other bits are set to 1.
This command is only for MB86290A.
When graphics LSI is MB86291, GdcGeoClearInterruptStatus must be used.

6.1.24 GdcGeoClearInterruptStatus [Clears interrupt status]

Format void GdcGeoClearInterruptStatus(GDC_ULONG clear)

Parameter clear Clears pattern (shown below)

Return value

bit 31 17 16 4 3 2

1		x	x		x	x	x	1	1

bit2 : VSYNC interrupt Clear =0, Hold =1

bit3 : Frame sync interrupt Clear =0, Hold =1

bit4 : External sync error interrupt Clear =0, Hold =1

bit16 : Command execution error interrupt Clear =0, Hold =1

bit17 : Command complete interrupt Clear =0, Hold =1

Return value None

Description Clears a request of interrupt, indicated by ISR (Interrupt Status Register), with clear pattern.

To clear an interrupt event, respective bit in the clear pattern for that event is set to 0 and all other bits are set to 1.

This command is only for MB86291.

When graphics LSI is MB86290A, GdcClearInterruptStatus must be used.

6.1.25 GdcSetInterruptMask [Sets interrupt mask]

Format: void GdcSetInterruptMask (GDC_UCHAR mask)

Parameter: mask — Mask pattern (shown below)

bit	7	- - - - -	4	3	2	1	0
	0	0	0	X	X	X	X	X

bit0: Command error interrupt	Mask=1,	Enable=0
bit1: Command complete interrupt	Mask=1,	Enable=0
bit2: VSYNC interrupt	Mask=1,	Enable=0
bit3: Frame sync interrupt	Mask=1,	Enable=0
bit4: External sync error interrupt	Mask=1,	Enable=0

Return value: None

Description: Sets interrupt mask pattern to IMASK (Interrupt MASK) register to disable interrupt requests generated by the respective events.

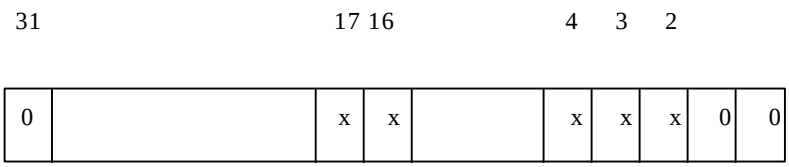
This command is only for MB86290A.

When graphics LSI is MB86291, GdcGeoSetInterruptMask must be used.

6.1.26 GdcGeoSetInterruptMask [Sets interrupt mask]

Format void GdcGeoSetInterruptMask(GDC_ULONG mask)

Parameter mask Mask pattern (shown below)



bit2	: VSYNC interrupt	Mask =0,	Enable =1
bit3	: Frame sync interrupt	Mask =0,	Enable =1
bit4	: External sync error interrupt	Mask =0,	Enable =1
bit16	: Command execution error interrupt	Mask =0,	Enable =1
bit17	: Command complete interrupt	Mask =0,	Enable =1

Return value None

Description Sets interrupt mask pattern to IMASK(Interrupt MASK) to disable interrupt requests generated by the respective events.
This command is only for MB86291.
When graphics LSI is MB86290A, GdcSetInterruptMask must be used.

6.1.27 GdcSetMemoryMode [Sets memory interface mode]

Format void GdcSetMemoryMode (GDC_ULONG memorymode)

Parameter memorymode Mode information of memory interface (MMR register set data)

Return value None

Description Sets memorymode value to MMR (Memory I/F Mode Register) and defines the operation mode of memory interface. Detail of the memorymode is referred to the MB86290 Series hardware specifications.
This command can be used by all graphics LSI.

6.1.28 GdcSoftwareReset [Resets by software]

Format	void GdcSoftwareReset (void)
Parameter	None
Return value	None
Description	Sets “1” to SRST (Software ReSet) register and execute software reset. This command can be used by all graphics LSI.

6.1.29 GdcGetErrCode [Gets error code]

Format Int GdcGetErrCode (void)

Parameter None

Return value Error code

Description Returns an error code when a driver command abnormally ends. Currently this command is applicable to GdcInitialize command only.
This command can be used by all graphics LSI.

Error code

GDC_ERR_DL_BUF_ALLOC	Failure of display list buffer acquisition
GDC_ERR_DL_SIZE	Incorrect buffer size A display list buffer size for 1 block is not a multiply of 32byte, or less than the minimum applicable size
GDC_ERR_DL_NUM	Not appropriate block count Block count is other than 1, 2
GDC_ERR_DATA_TOO_BIG	Too large data
GDC_ERR_INVALID_LAYER	Invalid layer is specified
GDC_ERR_INVALID_BANK	Invalid bank is specified
GDC_ERR_INVALID_COLOR_MODE	Invalid color mode is specified
GDC_ERR_INVALID_CURSOR_NUMBER	Invalid cursor number mode is specified
GDC_ERR_ILLEGAL_DIMENSION	Illegal vertical/horizontal size of pattern data
GDC_ERR_INVALID_ATTRIBUTE	Invalid attribute is specified
GDC_ERR_INVALID_PRIMITIVE	Invalid primitive is specified
GDC_ERR_CREMSON_OPEN_FAILED	Fail to initialize MB86290 series
GDC_ERR_ILLEGAL_VERTEX_COUNT	Illegal number of vertex
GDC_ERR_ILLEGAL_LINE_WIDTH	Illegal width of line
GDC_ERR_NOT_READY	Driver is not initialized
GDC_ERR_INVALID_PARAMETER	Invalid parameter is specified

6.2 Display Commands

6.2.1 GdcDispClock [Sets display clock mode]

Format: void GdcDispClock (GDC_ULONG mode)

Parameter:
- mode: Sets display clock mode. This parameter is directly set to the correlated hardware register of the MB86290 Series. Detail of the DCM register description is referred to the MB86290 Series hardware specifications.

Return value: None

Description: Control display clock and sync mode by setting parameters to Display Control Mode (DCM) register.

- Sets display sync mode
- Sets external sync mode
- Sets signal type
- Sets dot clock frequency
- Sets dot clock source

This command can be used by all graphics LSI.

6.2.2 GdcDispTiming [Sets display timing parameters]

Format: void GdcDispTiming (GDC_USHORT htp, GDC_USHORT hsp,
GDC_USHORT hsw, GDC_USHORT hdp,
GDC_USHORT vtr, GDC_USHORT vsp,
GDC_USHORT vsw, GDC_USHORT vdp)

Parameter:
- htp: Total horizontal pixel count
- hsp: Hsync pulse timing
- hsw: Hsync pulse width
- hdp: Horizontal display pixel count
- vtr: Total vertical raster count
- vsp: Vsync pulse timing
- vsw: Vsync pulse width
- vdp: Vertical display raster count

Return value: None

Description: Sets display window size and display timing parameters.

This command can be used by all graphics LSI.

6.2.3 GdcDispTimingWindow [Sets W-layer display position]

Format	void GdcDispTimingWindow (GDC_USHORT wx, GDC_USHORT wy, GDC_USHORT ww, GDC_USHORT wh)	
Parameter	wx	X coordinates of window display position (X coordinates of top left edge of the window in the display frame)
	wy	Y coordinates of window display position (Y coordinates of top left edge of the window in the display frame)
	ww	Window width
	wh	Window height
Return value	None	
Description	Sets display position of the W(Window) layer. This command can be used by all graphics LSI.	

6.2.4 GdcDispDividePos [Sets border position of screen partition]

Format	void GdcDispDividePos (GDC_USHORT hdb)	
Parameter	hdb	Horizontal pixel count of left window
Return value	None	
Description	Sets the border of left/right layers when screen partition mode is applied. When the value 0 is set, 1 line of right window is displayed as well as the value 1 is set. This command can be used by all graphics LSI.	

6.2.5 GdcDispDimension [Sets display frame attribute]

Format

```
int GdcDispDimension (GDC_UCHAR layer, GDC_UCHAR enable,
                      GDC_UCHAR cmode, GDC_UCHAR fmode,
                      GDC_ULONG loa0, GDC_ULONG loa1,
                      GDC_USHORT lw, GDC_USHORT lh)
```

Parameter

layer	Layer selection	
	GDC_DISP_LAYER_C	C layer
	GDC_DISP_LAYER_W	W layer
	GDC_DISP_LAYER_ML	ML layer
	GDC_DISP_LAYER_MR	MR layer
	GDC_DISP_LAYER_BL	BL layer
	GDC_DISP_LAYER_BR	BR layer
enable	Layer display enable/disable	
	GDC_ENABLE	Layer display enable
	GDC_DISABLE	Layer display disable
cmode	Color mode selection	
	GDC_16BPP_FORMAT	16bit color mode
	GDC_8BPP_FORMAT	8bit color mode
fmode	Flipping mode selection	
	GDC_FLIPMODE_0	Display Bank 0
	GDC_FLIPMODE_1	Display Bank 1
	GDC_FLIPMODE_AUTO	Display both banks alternately
loa0	Base address of logical frame (Bank 0)	
loa1	Base address of logical frame (Bank 1)	
lw	Logical frame width	
lh	Logical frame height	

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description

Sets attributes of logical frame independently for C, W, ML, MR, BL and BR. In B and M layer, if either L(Left) or R(Right) layer is enables, the other side layer is also enabled automatically. For W layer, cmode, fmode, loa1 and lh are not applied.

This command can be used by all graphics LSI.

6.2.6 GdcDispOn [Asserts video signal output]

Format: void GdcDispOn (void)

Parameter: None

Return value: None

Description: Outputs video signals.

Screen display is started at this command call, so this command must be called after all the rest display parameters are set. Nothing is displayed prior to this command call.

This command can be used by all graphics LSI.

6.2.7 GdcDispOff [Negates video signal output]

Format: void GdcDispOff (void)

Parameter: None

Return value: None

Description: Disables screen display of video signals.

This command can be used by all graphics LSI.

6.2.8 GdcDispLayerOn [Asserts screen display]

Format: int GdcDispLayerOn (GDC_UCHAR layer)

Parameter: Layer — Layer selection

GDC_DISP_LAYER_C	C layer
GDC_DISP_LAYER_W	W layer
GDC_DISP_LAYER_M	M layer
GDC_DISP_LAYER_B	BL layer

Return value: GDC_TRUE — Complete

GDC_FALSE — Incomplete

Description: Display the layer specified by the layer parameter.

This command can be used by all graphics LSI.

6.2.9 GdcDispLayerOff [Negates screen display]

Format int GdcDispLayerOff (GDC_UCHAR layer)

Parameter layer Layer selection

GDC_DISP_LAYER_C	C layer
GDC_DISP_LAYER_W	W layer
GDC_DISP_LAYER_M	M layer
GDC_DISP_LAYER_B	BL layer

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Description Stops the display of specified layer.

This command can be used by all graphics LSI.

6.2.10 GdcDispPos [Sets display start position]

Format int GdcDispPos (GDC_UCHAR layer, GDC_UCHAR bank,
GDC_USHORT dx, GDC_USHORT dy)

Parameter layer Layer selection

GDC_DISP_LAYER_C	C layer
GDC_DISP_LAYER_W	W layer
GDC_DISP_LAYER_ML	ML layer
GDC_DISP_LAYER_MR	MR layer
GDC_DISP_LAYER_BL	BL layer
GDC_DISP_LAYER_BR	BR layer

bank Logical frame bank selection

GDC_DISP_BANK_0	Bank 0
GDC_DISP_BANK_1	Bank 1

dx X coordinates of display start position

dy Y coordinates of display start position

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Description Sets the display start position by the distance from the base position of logical frame.

Bank, bank is not applied to W(Window) layer.

This command can be used by all graphics LSI.

6.2.11 GdcDispDoFlip [Flips display bank]

Format int GdcDispDoFlip (GDC_UCHAR layer, GDC_UCHAR bank)

Parameter

layer	Layer selection	
	GDC_DISP_LAYER_ML	ML layer
	GDC_DISP_LAYER_MR	MR layer
	GDC_DISP_LAYER_BL	BL layer
	GDC_DISP_LAYER_BR	BR layer
bank	Logical frame bank selection	
	GDC_DISP_BANK_0	Bank 0
	GDC_DISP_BANK_1	Bank 1

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description Executes flipping (of double buffer Bank0/Bank1).

This command can be used by all graphics LSI.

6.2.12 GdcOverlayPriorityMode [Sets overlay display mode]

Format int GdcOverlayPriorityMode (GDC_UCHAR mode)

Parameter

mode	C layer overlay mode	
	GDC_OVERLAY_C_PRIORITY	Simple priority mode
	GDC_OVERLAY_C_BLEND	Blend mode

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description Sets overlay display mode. When simple priority mode is selected, C layer is displayed at the top of all the layers all the time. When blend mode is selected, after displaying all the rest layers according to the priority order, C layer color is transparently blended with the rest layers.

This command can be used by all graphics LSI.

6.2.13 GdcOverlayBlend [Sets blend parameter for overlay blend]

Format int GdcOverlayBlend (GDC_UCHAR select, GDC_UCHAR blend)

Parameter

enable	Overlay blend selection	
	GDC_BLEND_RATIO_C	Blend target is C-layer color
	GDC_BLEND_RATIO_WMB	Blend target is WMB-layer color
blend	Blending ratio (only upper 4bits are valid)	

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description Sets the blend coefficient to determine the C layer color when the overlay mode is blend mode.

The followings are the meanings of blend coefficient and formula to determine the C layer color.

[Blend coefficient]

Blend	Blend coefficient
0x00	0
0x10	1/16
0x20	2/16
0x30	3/16
:	:
0xf0	15/16

[Blend formula]

- For GDC_BLEND_RATIO_C

(C_layer_color * blend_coefficient)
+ (W/M/B_layer_compound_color * (1- blend_coefficient))

- For GDC_BLEND_RATIO_WMB

(C_layer_color * (1- blend_coefficient))
+ (W/M/B_layer_compound_color * blend_coefficient)

This command can be used by all graphics LSI.

6.3 Color Control Commands

6.3.1 GdcColorPalette [Sets palette colors]

Format: int GdcColorPalette (GDC_UCHAR layer, GDC_UCHAR number, GDC_UCHAR size, GDC_LPPCOL32 lpColor)

Parameter	layer	C layer, MBlayer palette selection GDC_C_LAYER_PALETTE Select C layer palette GDC_MB_LAYER_PALETTE Select MB layer palette
	number	Sets the head palette number
	size	Sets the palette number
	lpColor	Pointer to the color data
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets color index code to palette table. If size is set to “0”, all 256 entries of selected palette are set. This command can be used by all graphics LSI.	

6.3.2 GdcColorTransparent [Sets transparent color]

Format: int GdcColorTransparent (GDC_UCHAR layer, GDC_COL16 color)

Parameter	layer	Layer selection GDC_DISP_LAYER_C C layer GDC_DISP_LAYER_ML ML layer GDC_DISP_LAYER_MR MR layer
	color	Transparent color code
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets transparent color code. In indirect color mode (in which color palette is applied), the lower 8bit is used. This command can be used by all graphics LSI.	

6.3.3 GdcColorZeroMode [Sets color code 0 mode]

Format: int GdcColorZeroMode (GDC_UCHAR layer, GDC_UCHAR mode)

Parameter

layer	Layer selection	
	GDC_DISP_LAYER_C	C layer
	GDC_DISP_LAYER_ML	ML layer
	GDC_DISP_LAYER_MR	MR layer
mode	Color 0 mode	
	GDC_COLOR_NOTRANSPARENT	NOT Transparent color
	GDC_COLOR_TRANSPARENT	Transparent color

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description: Selects the color option applied for color 0 (palette entry #0 color code in indirect mode or color value 0 in direct color mode). Color 0 is treat as either transparent color or ordinary color code in indirect mode or black in direct mode.

This command can be used by all graphics LSI.

6.3.4 GdcChromaKeyMode [Sets Chroma-key mode]

Format: int GdcChromaKeyMode (GDC_UCHAR mode, GDC_UCHAR source)

Parameter

mode	Chroma-key mode selection	
	GDC_ENABLE	Chroma-key operation enable
	GDC_DISABLE	Chroma-key operation disable
source	Source key color selection	
	GDC_CHROMAKEY_C	C layer color
	GDC_CHROMAKEY_DISP	Display color

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description: Sets Chroma-key operation and select which color is applied as the key color to be compared (C layer color of display color).

This command can be used by all graphics LSI.

6.3.5 GdcColorKey [Sets key color for Chroma-key]

Format	int GdcColorKey (GDC_COL16 color)	
Parameter	color	Key color for Chroma-key operation
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets the key color for Chroma-key operation. In indirect mode, the lower 8bit of this key color is applied as the color code of the key color. This command can be used by all graphics LSI.	

6.4 Cursor Control Commands

6.4.1 GdcCursorAddress [Sets cursor pattern memory address]

Format int GdcCursorAddress (GDC_UCHAR numCursor, GDC_ULONG ladrs)

Parameter	numCursor	Cursor number
	ladrs	Cursor pattern address
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets the start address of graphics memory where the cursor pattern is stored. This command can be used by all graphics LSI.	

6.4.2 GdcCursorPattern [Sets cursor pattern]

Format int GdcCursorPattern (GDC_UCHAR numCursor, GDC_LPCOL8 lpCursor)

Parameter	numCursor	Cursor
	lpCursor	Pointer of cursor pattern
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets a cursor pattern. Transfer a cursor pattern data in main memory pointed via lpCursor to the graphics memory that start address is designated by the GdcCursorAddress command. This command can be used by all graphics LSI.	

6.4.3 GdcCursorDisplay [Controls cursor display]

Format int GdcCursorDisplay (GDC_UCHAR numCursor, GDC_UCHAR enable)

Parameter	numCursor	Cursor number	
	enable	Cursor display ON/OFF	
		GDC_ENABLE	Cursor display ON
		GDC_DISABLE	Cursor display OFF
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	
Description	Controls cursor display ON or OFF.		
	This command can be used by all graphics LSI.		

6.4.4 GdcCursorPos [Sets cursor display position]

Format int GdcCursorPos (GDC_UCHAR numCursor,
GDC_USHORT x, GDC_USHORT y)

Parameter	numCursor	Cursor number
	x	X coordinates of cursor display position
	y	Y coordinates of cursor display position
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets display position of cursor.	
	This command can be used by all graphics LSI.	

6.4.5 GdcCursorPriority [Sets cursor display priority mode]

Format int GdcCursorPriority (GDC_UCHAR numCursor, GDC_UCHAR mode)

Parameter	numCursor	Cursor number	
	mode	Cursor display priority mode	
		GDC_PRIORITY_C_LAYER	C layer is prioritized
		GDC_PRIORITY_CURSOR	Cursor is prioritized
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	

Description Selects which is prioritized in display, C layer or cursor.

This command can be used by all graphics LSI.

6.4.6 GdcCursorColorTransparent [Sets cursor transparent color]

Format int GdcCursorColorTransparent (GDC_COL8 color)

Parameter	color	Color code to be treat as transparent color
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Description Sets a transparent color code for cursor.

This command can be used by all graphics LSI.

6.4.7 GdcCursorColorZeroMode [Sets cursor color code 0 mode]

Format int GdcCursorColorZeroMode (GDC_UCHAR mode)

Parameter	mode	Color code 0 mode	
		GDC_COLOR_NOTRANSPARENT	NOT Transparent color
		GDC_COLOR_TRANSPARENT	Transparent color
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	

Description Selects the color option applied for color code 0 in cursor pattern. Color code 0 is treat as either transparent color or ordinary color code.

This command can be used by all graphics LSI.

6.5 Draw Frame Control Commands

6.5.1 GdcDrawDimension [Sets draw frame]

Format int GdcDrawDimension (GDC_UCHAR cmode, GDC_ULONG dadrs, GDC_USHORT dw, GDC_USHORT dh)

Parameter	cmode	Color mode
		GDC_16BPP_FORMAT 16bit color mode GDC_8BPP_FORMAT 8bit color mode
	dadrs	Draws frame base address
	dw	Draws frame width
	dh	Draws frame height
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets color mode and size of draw frame.	
	This command can be used by all graphics LSI.	

6.5.2 GdcBufferCreateZ [Sets Z buffer start address]

Format int GdcBufferCreateZ (GDC_ULONG zadrs)

Parameter	zadrs	Z buffer base address
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets the base address of Z buffer. The vertical/horizontal size of Z buffer is assumed to be equal to that of draw frame. For each pixel, 16bit size of data is required for Z buffer.	
	This command can be used by all graphics LSI.	

6.5.3 GdcBufferCreateC [Sets start address of polygon draw flag buffer]

Format int GdcBufferCreateC (GDC_ULONG cadrs)

Parameter	cadrs	Polygon draw control buffer base address
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets the start address of polygon draw control buffer. The vertical/horizontal size of this control buffer is assumed to be equal to that of draw frame. For each pixel, 1bit of data is required for this buffer.	
	This command can be used by all graphics LSI.	

6.5.4 GdcBufferClearZ [Clears Z buffer]

Format int GdcBufferClearZ (void)

Parameter	None	
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Clears Z buffer. Prior to hidden surface manipulation, Z buffer should be cleared.	
	This command can be used by all graphics LSI.	

6.5.5 GdcBufferClearC [Clears polygon draw flag buffer]

Format int GdcBufferClearC (void)

Parameter	None	
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Clears polygon draw flag buffer.	
	This command can be used by all graphics LSI.	

6.5.6 GdcDrawClipFrame [Sets draw clip border]

Format `int GdcDrawClipFrame (GDC_USHORT x0, GDC_USHORT y0, GDC_USHORT x1, GDC_USHORT y1)`

Parameter	x0	X coordinates of left top edge of clip border
	y0	Y coordinates of left top edge of clip border
	x1	X coordinates of right bottom edge of clip border
	y1	Y coordinates of right bottom edge of clip border
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Description Sets clip border of drawing. Clip border is set as a Blt located relatively from the base point of draw frame. Drawing to the area outside of this clip border is not performed. This command can be used by all graphics LSI.

6.6 Primitive Draw Commands for Device Coordinates

6.6.1 GdcPrimType [Starts drawing procedure]

Format int GdcPrimType (GDC_UCHAR type)

Parameter type Sets primitive type

GDC_POINTS	Point
GDC_LINE	Line
GDC_POLYLINE	Poly-line
GDC_LINES_FAST	Fast 2D line
GDC_POLYLINE_FAST	Fast 2D poly-line
GDC_TRIANGLES	Triangle
GDC_TRIANGLE_STRIP	Triangle strip
GDC_TRIANGLE_FAN	Triangle fan
GDC_POLYGON	Polygon
GDC_TRIANGLES_FAST	Fast 2D triangle
GDC_TRIANGLE_STRIP_FAST	Fast 2D triangle strip
GDC_TRIANGLE_FAN_FAST	Fast 2D triangle fan

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Description Sets the primitive type to be drawn by DrawVertex2D or DrawVertex3D command.

Once either of these commands is executed, same type of primitive will keep being drawn till GdcPrimEnd will be executed.

This command can be used by all graphics LSI.

6.6.2 GdcPrimEnd [Completes drawing procedure]

Format void GdcPrimEnd (void)

Parameter None

Return value None

Description Stops drawing the primitive applied by GdcPrimType.

This command can be used by all graphics LSI.

6.6.3 GdcTexCoord2D / 2Df / 2DNf [Sets coordinates of 2D texture]

Format
void GdcTexCoord2D (GDC_FIXED32 u, GDC_FIXED32 v)
void GdcTexCoord2Df (GDC_SFLOAT u, GDC_SFLOAT v)
void GdcTexCoord2DNf (GDC_SFLOAT u, GDC_SFLOAT v)

Parameter
u U coordinates of texture mapped on the vertex
v V coordinates of texture mapped on the vertex

Return value
None

Description
Sets the texture coordinates (2D) for the vertex to be drawn by GdcDrawVertex command. Once this command is executed, the same texture coordinates is continuously applied till another GdcTexCoord command will be executed.
GdcTexCoord2DNf command is applied in case the texture coordinates is normalized (max value of the coordinates for current texture is 1.0).
This command can be used by all graphics LSI.

6.6.4 GdcTexCoord3D / 3Df / 3DNf [Sets coordinates of 3D texture]

Format
void GdcTexCoord3D (GDC_FIXED32 u, GDC_FIXED32 v, GDC_FIXED32 rw)
void GdcTexCoord3Df (GDC_SFLOAT u, GDC_SFLOAT v, GDC_SFLOAT rw)
void GdcTexCoord3DNf (GDC_SFLOAT u, GDC_SFLOAT v, GDC_SFLOAT rw)

Parameter
u U coordinates of texture mapped on the vertex
v V coordinates of texture mapped on the vertex
rw Reciprocal of W coordinates of texture mapped on the vertex

Return value
None

Description
Sets the texture coordinates (3D) for the vertex to be drawn by GdcDrawVertex command. Once this command is executed, the same texture coordinates is continuously applied till another GdcTexCoord command will be executed.
GdcTexCoord3DNf command is applied in case the texture coordinates is normalized (max value of the coordinates for current texture is 1.0).
This command can be used by all graphics LSI.

6.6.5 GdcDrawVertex2D / 2Di [Sets coordinates of 2D vertex]

Format void GdcDrawVertex2D (GDC_FIXED32 x, GDC_FIXED32 y)
void GdcDrawVertex2Di (GDC_LONG x, GDC_LONG y)

Parameter
- x X coordinates of 2D vertex
- y Y coordinates of 2D vertex

Return value None

Description Sets 2D vertex coordinates and draw a designated primitive. Color and texture coordinates defined previously by GdcColor, GdcColorI, GdcBackColor, GdcBackColorI, GdcTexCoord2D and GdcTexCoord3D are applied in this draw operation.

GdcDrawVertex2Di command is applicable to the following primitives:

GDC_LINES_FAST
GDC_POLYLINE_FAST
GDC_POLYGON
GDC_TRIANGLES_FAST
GDC_TRIANGLE_STRIP_FAST
GDC_TRIANGLE_FAN_FAST

This command can be used by all graphics LSI.

6.6.6 GdcDrawVertex3D / 3Df [Sets coordinates of 3D vertex]

Format void GdcDrawVertex3D (GDC_FIXED32 x, GDC_FIXED32 y, GDC_USHORT z)
void GdcDrawVertex3Df (GDC_SFLOAT x, GDC_SFLOAT y, GDC_SFLOAT z)

Parameter
- x X coordinates of 3D vertex
- y Y coordinates of 3D vertex
- z Z coordinates of 3D vertex

Return value None

Description Sets 3D vertex coordinates and draw a designated primitive. Color and texture coordinates defined previously by GdcColor, GdcColorI, GdcBackColor, GdcBackColorI, GdcTexCoord2D and GdcTexCoord3D are applied in this draw operation.

GdcDrawVertex3Df command is applicable to the following primitives:

GDC_TRIANGLES
GDC_TRIANGLE_STRIP
GDC_TRIANGLE_FAN

This command can be used by all graphics LSI.

6.6.7 GdcDrawPrimitive [Draws multiple 3D triangles]

Format: int GdcDrawPrimitive (GDC_ULONG type, GDC_VERTEX lpVertices, int count)

Parameter

type	Sets primitive type	
	GDC_TRIANGLES	Triangle
	GDC_TRIANGLE_STRIP	Triangle strip
	GDC_TRIANGLE_FAN	Triangle fan
lpVertices	Pointer of vertex parameter list (coordinates, color texture coordinates)	
count	Number of vertices	

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description: Draws a primitive specified in the type formed with multiple vertices designated by lpVertices.

This command can be used by all graphics LSI.

6.7 Primitive Draw Control Commands for Object Coordinates

6.7.1 GdcGeoPrimType [Starts drawing procedure]

Format int GdcGeoPrimType (GDC_UCHAR type)

Parameter Type Sets primitive type

GDC_POINTS	points
GDC_LINES	lines
GDC_POLYLINE	polylines
GDC_TRIANGLES	triangle
GDC_TRIANGLE_STRIP	linked strip of triangles
GDC_TRIANGLE_FAN	linked fan of triangles
GDC_POLYGON	polygon

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Description Sets primitive drawn with GdcGeoDrawVertex2D* or GdcGeoDrawVertex3D*.
Once this command is executed, the same primitive is drawn until GdcGeoPrimEnd is executed.
This command is only for MB86291.

6.7.2 GdcGeoPrimEnd [Completes drawing procedure]

Format void GdcGeoPrimEnd (void)

Parameter None

Return value None

Description Terminates a series of processes to draw primitives fallowing GdcGeoPrimType.
This command is only for MB86291.

6.7.3 GdcGeoDrawVertex2D / 2Df / 2Di [Sets XY coordinates of vertex]

Format void GdcGeoDrawVertex2D (GDC_FIXED32 x, GDC_FIXED32 y)
void GdcGeoDrawVertex2Df (GDC_SFLOAT x, GDC_SFLOAT y)
void GdcGeoDrawVertex2Di (GDC_LONG x, GDC_LONG y)

Parameter x X coordinates of the vertex
y Y coordinates of the vertex

Return value None

Description Specifies a vertex coordinates in object coordinates and draw a primitive currently set. In this case, z is treated as zero.
Current values of color and texture coordinates are used in drawing, which has been set by the vertex color setting command and texture coordinates setting command respectively.
This command is only for MB86291.

6.7.4 GdcGeoDrawVertex3D / 3Df / 3Di [Sets XYZ coordinates of vertex]

Format void GdcGeoDrawVertex3D (GDC_FIXED32 x,GDC_FIXED32 y,
GDC_FIXED32 z)
void GdcGeoDrawVertex3Df (GDC_SFLOAT x, GDC_SFLOAT y, GDC_SFLOAT z)
void GdcGeoDrawVertex3Di (GDC_LONG x, GDC_LONG y, GDC_FIXED32 z)

Parameter x X coordinates of the vertex
y Y coordinates of the vertex
z Z coordinates of the vertex

Return value None Complete

Description Sets vertex coordinates in object coordinates and draw a primitive currently set. In this case, z is treated as zero.
Current values of color and texture coordinates are used in drawing, which has been set by the vertex color setting command and texture coordinates setting command respectively.
This command is only for MB86291.

6.7.5 GdcGeoTexCoord2D / 2Df / 2DN / 2DNf [Sets texture coordinates]

Format: void GdcGeoTexCoord2DN (GDC_FIXED32 u, GDC_FIXED32 v)
void GdcGeoTexCoord2DNf (GDC_SFLOAT u, GDC_SFLOAT v)

Parameter:
- u: Texture U coordinates of the vertex
- v: Texture V coordinates of the vertex

Return value: None

Description: Sets a texture coordinates (2 dimensions) of vertex in drawing with the vertex coordinates setting command. Once this command is executed, the same texture coordinates is used in drawing unless texture coordinates is changed by this command.
GdcGeoTexCoord2DN, GdcGeoTexCoord2DNf treat texture coordinates as normalized (1.0 is maximum size of current texture).
This command is only for MB86291.

6.7.6 GdcVertexColor3f / 32 [Sets color of vertex]

Format: void GdcVertexColor3f (GDC_SFLOAT r, GDC_SFLOAT g, GDC_SFLOAT b)
void GdcVertexColor32 (GDC_COLOR32 color)

Parameter:
- r, g, b: Normalized values in which each color elements (r,g,b) are normalized to [0,1].
- color: Packed format in which each color elements (r,g,b) is normalized to [0,255]. In this case, r,g,b are 8 bit respectively.

Return value:
- GDC_TRUE: Complete
- GDC_FALSE: Incomplete

Description: Sets a color of vertex. Once this command is executed, the same color is used in drawing for object coordinates unless the color is changed by this command.
This command is used when shading mode is smooth shading. If the shading mode is flat shading, use GdcColor.
This command is only for MB86291.

6.8 Draw Attribute Control Commands

6.8.1 GdcColor, GdcColorI [Sets vertex color/foreground color]

Format int GdcColor (GDC_COL16 color)
int GdcColorI (GDC_COL8 color)

Parameter	color	Vertex and foreground color
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Description Sets vertex color and foreground color applied for bitmap draw and broken line draw to be executed by GdcDrawVertex command in direct color mode. Once this command is executed, the same color is continuously applied till another GdcColor or GdcColorI command will be executed.
This command can be used by all graphics LSI.

6.8.2 GdcBackColor, GdcBackColorI [Sets background color]

Format int GdcBackColor (GDC_COL16 color)
int GdcBackColorI (GDC_COL8 color)

Parameter	color	Background color
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Description Sets background color applied for binary pattern draw and broken line draw. Once this command is executed, the same color is continuously applied till another GdcBackColor or GdcBackColorI command will be executed.
This command can be used by all graphics LSI.

6.8.3 GdcSetAttrMisc [Sets draw attribute]

Format int GdcSetAttrMisc (GDC_ULONG target, GDC_ULONG param)

Parameter target Attribute

GDC_CLIP Enable/disable clipping

param Parameter belong to each attribute (shown below)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Description Sets draw attribute.

Only clipping can be set as attribute at present. Specify GDC_CLIP as target.

param is among the followings. GDC_CLIP_X_ON and GDC_CLIP_Y_ON can be set at set the same time with OR operator.

This command can be used by all graphics LSI.

GDC_CLIP	Enable/disable clip operation GDC_CLIP_X_ON and GDC_CLIP_Y_ON are applicable at the same time
GDC_CLIP_X_ON	Validate clipping toward X axis
GDC_CLIP_Y_ON	Validate clipping toward Y axis
GDC_CLIP_DISABLE	Invalidate clipping

6.8.4 GdcSetAttrLine [Sets line draw attribute]

Format int GdcSetAttrLine (GDC_ULONG target, GDC_ULONG param)

Parameter target Line draw attribute

GDC_DEPTH_TEST	Z value compare mode
GDC_DEPTH_FUNC	Z value compare type
GDC_DEPTH_WRITE_MASK	Z value write permission mask
GDC_BLEND_MODE	Blending mode
GDC_BROKEN_LINE	Broken line mode
GDC_LINE_WIDTH	Line width
GDC_ANTI_ALIAS	Antialias option
GDC_LINE_ENDPOINT	End of the line control
GDC_BROKEN_LINE_OFFSET	Offset control of broken line pattern (for MB86291)
GDC_BROKEN_LINE_PERIOD	Period set of broken line pattern (for MB86291)

param Parameter belong to each attribute (shown below)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Description Sets attribute for line draw.

This command can be used by all graphics LSI.

However, GDC_BROKEN_LINE_OFFSET and GDC_BROKEN_LINE_PERIOD for setting for target can be only used by MB86291.

GDC_DEPTH_TEST Sets Z value compare mode

GDC_ENABLE	Validates Z value comparison
GDC_DISABLE	Invalidates Z value comparison

GDC_DEPTH_FUNC Selects Z value comparison type

GDC_DEPTH_NEVER	Always NOT drawn
GDC_DEPTH_ALWAYS	Always drawn
GDC_DEPTH_LESS	Drawn if current Z value is less than Z buffer value
GDC_DEPTH_LEQUAL	Drawn if current Z value equal to or less than Z buffer value
GDC_DEPTH_EQUAL	Drawn if current Z value equal to Z buffer value
GDC_DEPTH_GEQUAL	Drawn if current Z value equal to or more than Z buffer value
GDC_DEPTH_GREATER	Drawn if current Z value more than Z buffer value
GDC_DEPTH_NOTEQUAL	Drawn if current Z value is not equal to Z buffer value

GDC_DEPTH_WRITE_MASK Enables write access to Z buffer

If GDC_ENABLE, according to the result of Z value comparison, Z value is written to Z buffer

GDC_ENABLE	Disable Z buffer write
GDC_DISABLE	Enable Z buffer write

GDC_BLEND_MODE	Sets blending mode of pixel write
GDC_BLEND_COPY	Regular draw operation (writes pixel color to draw frame)
GDC_BLEND_ALPHA	Enables alpha blending
GDC_BLEND_ROP	Draws with logical arithmetic
GDC_BROKEN_LINE	Selects broken line mode
GDC_ENABLE	Draws a broken line utilizing applied line pattern
GDC_DISABLE	Draws a solid line
GDC_LINE_WIDTH	Sets line width
GDC_LINE_WIDTH_1	Draws a line of 1 pixel width
GDC_LINE_WIDTH_2	Draws a line of 2 pixel width
:	:
GDC_LINE_WIDTH_32	Draws a line of 32 pixel width
GDC_ANTI_ALIAS	Sets antialias mode
GDC_ENABLE	Enables antialias operation
GDC_DISABLE	Disables antialias operation
GDC_LINE_ENDPOINT	Controls the end point of line in GDC_LINES and GDC_LINES_FAST commands End point is not drawn in GDC_POLYLINE and GDC_POLYLINE_FAST commands regardless this setting
GDC_ENABLE	Draws the end point
GDC_DISABLE	NOT draws the end point
GDC_BROKEN_LINE_OFFSET	Specifies the way of drawing broken line (only for MB86291)
GDC_ENABLE	Starts new drawing broken line pattern
GDC_DISABLE	Continues from the last drawing broken line pattern
GDC_BROKEN_LINE_PERIOD	Sets broken line pattern period (only for MB86291)
GDC_BROKEN_LINE_32	32 bit period
GDC_BROKEN_LINE_24	24 bit period

6.8.5 GdcSetAttrSurf [Sets surface draw attribute]

Format int GdcSetAttrSurf (GDC_ULONG target, GDC_ULONG param)

Parameter target Surface draw attribute

GDC_SHADE_MODE	Shading mode
GDC_DEPTH_TEST	Z value compare mode
GDC_DEPTH_FUNC	Z value compare type
GDC_DEPTH_WRITE_MASK	Z value write mask
GDC_BLEND_MODE	Blending mode
GDC_TEXTURE_SELECT	Texture mode

param Parameter belong to each attribute (shown below)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Description Sets attribute for surface draw (not including texture mapping attribute).

This command can be used by all graphics LSI.

GDC_SHADE_MODE — Sets shading mode

GDC_SHADE_FLAT	Flat shading
GDC_SHADE_SMOOTH	Gouraud shading

GDC_DEPTH_TEST — Sets Z value compare mode

GDC_ENABLE	Validate Z value comparison
GDC_DISABLE	Invalidate Z value comparison

GDC_DEPTH_FUNC — Selects Z value comparison type

GDC_DEPTH_NEVER	Always NOT drawn
GDC_DEPTH_ALWAYS	Always drawn
GDC_DEPTH_LESS	Drawn if current Z value is less than Z buffer value
GDC_DEPTH_LEQUAL	Drawn if current Z value is equal to or less than Z buffer value
GDC_DEPTH_EQUAL	Drawn if current Z value is equal to Z buffer value
GDC_DEPTH_GEQUAL	Drawn if current Z value is equal to or more than Z buffer value
GDC_DEPTH_GREATER	Drawn if current Z value is more than Z buffer value
GDC_DEPTH_NOTEQUAL	Drawn if current Z value is not equal to Z buffer value

GDC_DEPTH_WRITE_MASK — Enables write access to Z buffer

If GDC_ENABLE, according to the result of Z value comparison, Z value is written to Z buffer

GDC_ENABLE	Disables Z buffer write
GDC_DISABLE	Enables Z buffer write

GDC_BLEND_MODE	Sets blending mode of pixel write
GDC_BLEND_COPY	Regular draws operation (writes pixel color to draw frame)
GDC_BLEND_ALPHA	Enables alpha blending
GDC_BLEND_ROP	Draws with logical arithmetic
GDC_TEXTURE_SELECT	Sets texture mapping mode
GDC_SELECT_TEXTURE	Draws with texture mapping
GDC_SELECT_TILE	Draws with tiling
GDC_SELECT_PLAIN	Invalidates texture mapping

6.8.6 GdcGeoSetAttrSurf [Sets surface draw attribute for object coordinates]

Format int GdcGeoSetAttrSurf (GDC_ULONG target, GDC_ULONG param)

Parameter

target	Attribute for surface drawing as target	
	GDC_GEO_FACE_CULL	Enable/disable culling back face of triangle.
	GDC_GEO_FACE_INVERT	Specify direction of surface of triangle.
param	Parameter belong to each attribute (shown below)	

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description Sets surface draw attribute in object coordinates.
It doesn't affect polygons.
Attributes for surface drawing and parameters for them are described below.
This command is only for MB86291.

GCD_GEO_FACE_CULL		Specifies culling back face of triangle.
	GDC_ENABLE	Enables culling back face of triangle.
	GDC_DISABLE	Disables culling back face of triangle.

GCD_GEO_FACE_INVERT		Specifies direction of surface of triangle. Counterclockwise surface is front facing by default.
	GDC_ENABLE	Invert direction of surface from default.
	GDC_DISABLE	Direction of surface is default.

6.8.7 GdcSetAttrTexture [Sets texture mapping attribute]

Format int GdcSetAttrTexture (GDC_ULONG target, GDC_ULONG param)

Parameter target Texture mapping attribute

GDC_TEXTURE_PERSPECTIVE	Perspective correction
GDC_TEXTURE_FILTER	Texture filter
GDC_TEXTURE_WRAP_S	S coordinates wrap
GDC_TEXTURE_WRAP_T	T coordinates wrap
GDC_TEXTURE_BLEND	Texture blend mode
GDC_TEXTURE_ALPHA	Texture alpha mode

param Parameter belong to each attribute (shown below)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Description Sets attribute for texture mapping.

This command can be used by all graphics LSI.

GDC_TEXTURE_PERSPECTIVE	Selects perspective correction mode
GDC_ENABLE	Validates perspective correction
GDC_DISABLE	Invalidates perspective correction
GDC_TEXTURE_FILTER	Selects texture filter mode
GDC_TEXTURE_POINT	Point sampling mode
GDC_TEXTURE_BILINEAR	Bi-linear filtering mode
GDC_TEXTURE_WRAP_S	Defines S coordinates wrapping option when S coordinates value exceed the texture size
GDC_TEXTURE_REPEAT	Repeats the texture pattern
GDC_TEXTURE_CLAMP	Sets out-most texture color
GDC_TEXTURE_BORDER	Sets defined border color
GDC_TEXTURE_WRAP_T	Sets T coordinates wrapping option when T coordinates value exceed the texture size
GDC_TEXTURE_REPEAT	Repeats the texture pattern
GDC_TEXTURE_CLAMP	Sets out-most texture color
GDC_TEXTURE_BORDER	Sets defined border color

GDC_TEXTURE_BLEND	Sets blending mode of texture color and polygon color. This is applicable only when texture mapping mode is selected.
GDC_TEXTURE_DECAL GDC_TEXTURE_MODULATE	Texture color is drawn Blended color is drawn
GDC_TEXTURE_STENCIL	If MSB of texture color is 1, texture color is drawn, otherwise polygon color is drawn
GDC_TEXTURE_ALPHA	Sets alpha blending mode between drawn color and current pixel color of the draw frame This is applicable only when texture mapping and alpha blending are selected
GDC_TEXTURE_ALPHA_ALL	Alpha blend between post texture mapping color and current pixel color of the draw frame
GDC_TEXTURE_ALPHA _STENCIL	If MSB of texture color is 1, texture color is drawn, otherwise not drawn
GDC_TEXALPHA_ALPHA _STENCILALPHA	If MSB of texture color is 1, alpha blend between texture color and current pixel color in the draw frame is performed, otherwise not drawn

6.8.8 GdcSetAttrBlt [Sets BitBlt attribute]

Format int GdcSetAttrBlt (GDC_ULONG target, GDC_ULONG param)

Parameter	target	Bitmap draw attribute
		GDC_BLEND_MODE Blend mode
	param	Parameter belongs to each attribute (shown below)
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets attribute when copying and drawing BitBlt.	
	This blend mode setting function can be used by all graphics LSI.	
	The transparent mode setting function is only for MB86291.	
	Sets the transparent color for the transparent mode with GdcBltColorTransparent.	

GDC_BLEND_MODE	Sets blend mode
GDC_BLEND_COPY	Regular draw operation (writes pixel color to draw frame)
GDC_BLEND_ROP	Draws with logical arithmetic

GDC_TRANSPARENT_MODE	Sets transparent mode (only for MB86291)
GDC_ENABLE	The color which was set by GdcBltColorTransparent regards as transparent color
GDC_DISABLE	The color which was set by GdcBltColorTransparent is not treated as transparent color

6.8.9 GdcSetAlpha [Sets alpha blend ratio]

Format int GdcSetAlpha (GDC_UCHAR alpha)

Parameter	alpha	Alpha blending ratio
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets the ratio of alpha blending.	
	This command can be used by all graphics LSI.	

6.8.10 GdcSetLinePattern [Sets broken line pattern]

Format	int GdcSetLinePattern (GDC_ULONG pattern)	
Parameter	pattern	Broken line pattern
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets line pattern applied for broken line draw.	
	This command can be used by all graphics LSI.	

6.8.11 GdcSetTextureBorder [Sets texture border color]

Format	int GdcSetTextureBorder (GDC_COL16 color)	
Parameter	color	Texture border color
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets the border color of the texture applied in border mode of texture wrap.	
	This command can be used by all graphics LSI.	

6.8.12 GdcSetRop [Sets logical calculation mode]

Format int GdcSetRop (GDC_UCHAR mode)

Parameter mode Logical arithmetic mode

GDC_ROP_CLEAR	all0
GDC_ROP_AND	s & d
GDC_ROP_AND_REVERSE	s & !d
GDC_ROP_COPY	s
GDC_ROP_AND_INVERTED	!s & d
GDC_ROP_NOP	d
GDC_ROP_XOR	s ^ d
GDC_ROP_OR	s | d
GDC_ROP_NOR	!(s | d)
GDC_ROP_EQUIV	!(s ^ d)
GDC_ROP_INVERT	!d
GDC_ROP_OR_REVERSE	s | !d
GDC_ROP_COPY_INVERTED	!s
GDC_ROP_OR_INVERTED	!s | d
GDC_ROP_NAND	!(s & d)
GDC_ROP_SET	all1

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Description Sets logical arithmetic type.

This operation is performed between the pixel color to be drawn and current pixel color inn the draw frame. Result of this operation is to be drawn to the draw frame

This operation is applicable only when GDC_BLEND_ROP option of GDC_BLEND_MODE is selected.

This command can be used by all graphics LSI.

6.9 Attribute Control Commands for Object Coordinates

6.9.1 GdcGeoSetAttrMisc [Sets miscellaneous attribute]

Format int GdcGeoSetAttrMisc (GDC_ULONG target, GDC_ULONG param)

Parameter target Attributes for geometry as target

GDC_GEO_VTX_COL	Enable/disable vertex color in smooth shading.
GDC_GEO_VTX_Z	Enable/disable Z coordinates of vertex in Z value comparison.
GDC_GEO_VTX_ST	Enable/disable ST coordinates of vertex in texture mapping.
GDC_GEO_IN_FORMAT	Input format.

param Parameter belong to each attribute (shown below)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Description Sets miscellaneous attribute for object coordinates.

Attributes and parameters for them are described below.

This command is only for MB86291.

GCD_GEO_VTX_COL		Specifies when using color of vertex. Also, set smooth shading with GdcSetAttrSurf.
	GDC_ENABLE	Enables vertex color.
	GDC_DISABLE	Disables vertex color.
GCD_GEO_VTX_Z		Specifies when using Z coordinates of vertex.
	GDC_ENABLE	Enables Z value.
	GDC_DISABLE	Disables Z value.
GCD_GEO_VTX_ST		Specifies when using texture coordinates of vertex. Also, set texture-mapping mode with GdcSetAttrSurf.
	GDC_ENABLE	Enables ST (texture coordinates).
	GDC_DISABLE	Disables ST (texture coordinates).
GDC_GEO_IN_FORMAT		Specifies input format.
	GDC_GEO_FLOAT_INPUT	Floating-point format.
	GDC_GEO_FIXED_INPUT	Fixed-point format.
	GDC_GEO_INT_INPUT	Integer format.

6.9.2 GdcGeoLoadMatrix[f] [Sets matrix]

Format void GdcGeoLoadMatrix (GDC_FIXED32 *ptMatrix)

void GdcGeoLoadMatrixf (GDC_SFLOAT *ptMatrix)

Parameter *ptMatrix A pointer to an array {m1, m2, m3, …, m16} which corresponds to the 4 x 4 matrix M such as,

$$M=\begin{pmatrix} M1 & m5 & m9 & m13 \\ M2 & m6 & m10 & m14 \\ M3 & m7 & m11 & m15 \\ M4 & m8 & m12 & m16 \end{pmatrix}$$

Return value None

Description Sets a 4 x 4 matrix that transforms an object coordinates to a clip coordinate.

Each element in the matrix is put in the following order.

$$M=\begin{pmatrix} m1 & m5 & m9 & m13 \\ m2 & m6 & m10 & m14 \\ m3 & m7 & m11 & m15 \\ m4 & m8 & m12 & m16 \end{pmatrix}$$

Elements (m4 ,m8, m12, m16) in the matrix specify whether the projection type is orthographic or perspective. Therefore the projection type is set automatically by the result of their values.

If (m4 ,m8, m12, m16) == (0,0,0,1), then orthographic projection.

Else if (m4 ,m8, m12, m16) != (0,0,0,1) then perspective projection.

This command is only for MB86291.

6.9.3 GdcGeoNdcDcViewportCoef[f] [Sets coefficients of NdcDc transformation for XY]

Format
void GdcGeoNdcDcViewportCoef (GDC_FIXED32 scalex, GDC_FIXED32 offsetx, GDC_FIXED32 scaley, GDC_FIXED32 offsety)
void GdcGeoNdcDcViewportCoeff (GDC_SFLOAT scalex, GDC_SFLOAT offsetx, GDC_SFLOAT scaley, GDC_SFLOAT offsety)

Parameter
scalex magnification of X
offsetx offset of X
scaley magnification of Y
offsety offset of Y

Return value
None

Description
Sets the magnifications and offsets of X,Y that is used for transforming Normalized Device Coordinates (NDC) to Device Coordinates (DC).
This command is only for MB86291.

6.9.4 GdcGeoNdcDcDepthCoef[f] [Sets coefficients of NdcDc transformation for Z]

Format
void GdcGeoNdcDcDepthCoef (GDC_FIXED32 scalez, GDC_FIXED32 offsetz)
void GdcGeoNdcDcDepthCoeff (GDC_SFLOAT scalez, GDC_SFLOAT offsetz)

Parameter
scalex magnification of Z
offsetx offset of Z

Return value
None

Description
Sets the magnification and offset of Z that is used for transforming Normalized Device Coordinates (NDC) to Device Coordinates (DC).
This command is only for MB86291.

6.9.5 GdcGeoViewVolumeXYClip[f] [Sets view volume boundary for XY]

Format
void GdcGeoViewVolumeXYClip (GDC_FIXED32 xmin, GDC_FIXED32 xmax, GDC_FIXED32 ymin, GDC_FIXED32 ymax)
void GdcGeoViewVolumeXYClipf (GDC_SFLOAT xmin, GDC_SFLOAT xmax, GDC_SFLOAT ymin, GDC_SFLOAT ymax)

Parameter
xmin minimum clip value of x
xmax maximum clip value of x
ymin minimum clip value of y
ymax maximum clip value of y

Return value
None

Description
Sets the view volume boundary in the clip coordinates for XY.
This command is only for MB86291.

6.9.6 GdcGeoViewVolumeZClip[f] [Sets view volume boundary for Z]

Format
void GdcGeoViewVolumeZClip (GDC_FIXED32 zmin, GDC_FIXED32 zmax)
void GdcGeoViewVolumeZClipf (GDC_SFLOAT zmin, GDC_SFLOAT zmax)

Parameter
zmin minimum clip value of z.
zmax maximum clip value of z.

Return value
None

Description
Sets the view volume boundary in the clip coordinates for Z.
This command is only for MB86291.

6.9.7 GdcGeoViewVolumeWminClip[f] [Sets view volume boundary for w]

Format
void GdcGeoViewVolumeWminClip (GDC_FIXED32 wmin)
void GdcGeoViewVolumeWminClipf (GDC_SFLOAT wmin)

Parameter
wmin minimum clip value of w.

Return value
None

Description
Sets the view volume boundary in the clip coordinates for w.
As the front clip face (zmin) closes with the viewpoint limitlessly, w also approximates to zero limitlessly.
Since w is used to calculate 1/w internally, wmin must be the one that does not occur overflow in division.
w has only minimum value. wmin is not minus value.
This command is only for MB86291.

6.10 Texture Pattern Control Commands

6.10.1 GdcTextureMemoryMode [Sets texture memory mode]

Format int GdcTextureMemoryMode (GDC_UCHAR mode)

Parameter	mode	Texture memory read	
		GDC_TEX_MEM_MODE_EXT	Read from graphics memory
		GDC_TEX_MEM_MODE_INT	Read from internal buffer
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	

Description Sets the source memory to refer texture pattern from, either internal texture buffer or the graphics memory.

This command can be used by all graphics LSI.

6.10.2 GdcTextureLoadInt [Loads texture/tile pattern to internal texture buffer]

Format int GdcTextureLoadInt (GDC_USHORT length, GDC_LPCOL16 lpTexture, GDC_ULONG oadrs)

Parameter	length	Texture pattern size (pixel unit)
	lpTexture	Pointer to refer texture pattern
	oadrs	Offset address of the memory texture pattern is stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Description Loads texture pattern or tile pattern to internal texture buffer.

GDC_COL8 format texture data is assumed to be packed in GDC_COL16 texture format.

This command can be used by all graphics LSI.

6.10.3 GdcTextureLoadExt [Loads texture pattern to graphics memory]

Format	int GdcTextureLoadExt (GDC_ULONG length, GDC_LPCOL16 lpTexture, GDC_ULONG adrs)	
Parameter	length	Texture pattern size (pixel unit)
	lpTexture	Pointer to refer texture pattern
	adrs	Offset address of the memory texture pattern is stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Copies texture pattern to the graphics memory. Prior to this command execution, size of the texture pattern should be set by GdcTextureDimension command. If length is longer than 65536, transfer will not complete successfully. This command can be used by all graphics LSI.	

6.10.4 GdcTextureDimension [Sets texture information]

Format	int GdcTextureDimension (GDC_ULONG adrs, GDC_ULONG tw, GDC_ULONG th)	
Parameter	adrs	Start address of texture pattern (Offset address from the top, if internal texture buffer is used)
	tw	Texture width (Power of 2)
	th	Texture height (Power of 2)
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets the start address and size of texture pattern to refer. This command can be used by all graphics LSI.	

6.10.5 GdcBltTexture [Loads Blt texture to internal texture buffer for MB86290A]

Format	int GdcBltTexture (GDC_ULONG sadrs, GDC_ULONG sstride, GDC_USHORT x, GDC_USHORT y, GDC_USHORT w, GDC_USHORT h, GDC_ULONG oadrs)	
Parameter	Sadr	Memory address of the base point of the source draw frame
	Sstride	Stride (memory size of horizontal span) of the source draw frame
	X	X coordinates of the top left vertex of source Blt
	Y	Y coordinates of the top left vertex of source Blt
	W	Horizontal width of the Blt field
	H	Vertical height of the Blt field
	Oadrs	Offset address of destination memory where texture pattern to be stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Loads texture pattern from the graphics memory to the internal texture buffer. This command is only for MB86290A. When graphics LSI is MB86291, GdcGeoBltTexture must be used.	

6.10.6 GdcGeoBltTexture [Loads Blt texture to internal texture buffer for MB86291]

Format

```
int GdcGeoBltTexture (GDC_ULONG sadrs, GDC_ULONG sstride,
        GDC_USHORT x, GDC_USHORT y,
        GDC_USHORT w, GDC_USHORT h, GDC_ULONG oadrs)
```

Parameter	sadr	Memory address of the base point of the source draw frame
	sstride	Stride (memory size of horizontal span) of the source draw frame
	x	X coordinates of the top left vertex of source Blt
	y	Y coordinates of the top left vertex of source Blt
	w	Horizontal width of the Blt field
	h	Vertical height of the Blt field
	oadrs	Offset address of destination memory where texture pattern to be stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Description

Loads texture pattern from the graphics memory to the internal texture buffer.
This command is only for MB86291.
When graphics LSI is MB86290A, GdcBltTexture must be used.

6.11 Binary Pattern Draw Commands

6.11.1 GdcBitPatternDraw [Draws binary pattern]

Format: int GdcBitPatternDraw (GDC_USHORT x, GDC_USHORT y, GDC_USHORT w, GDC_USHORT h, GDC_LPBINIMAGE lpPattern)

Parameter

x	X coordinates of the draw frame when the top left point of binary pattern is drawn
y	Y coordinates of the draw frame when the top left point of binary pattern is drawn
w	Binary pattern data width
h	Binary pattern data height
lpPattern	Pointer to binary pattern data

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description: Draws a binary pattern. Foreground color (pixel of binary pattern "1") is drawn in the color applied by GdcColor or GdcColorI command, and background color (pixel of binary pattern "0") is drawn in the color applied GdcBackColor or GdcBackColorI command.

This command can be used by all graphics LSI.

6.11.2 GdcBitPatternMode [Sets enlarge/shrink mode]

Format: int GdcBitPatternMode (GDC_UCHAR mode)

Parameter

mode: Enlarge/shrink mode(GDC_BPSCALE_H and GDC_BPSCALE_V are applicable at the same time)

GDC_BPSCALE_H_EQUIV	Horizontal enlarge x1
GDC_BPSCALE_H_TWICE	Horizontal enlarge x2
GDC_BPSCALE_H_HALF	Horizontal enlarge x1/2
GDC_BPSCALE_V_EQUIV	Vertical enlarge x1
GDC_BPSCALE_V_TWICE	Vertical enlarge x2
GDC_BPSCALE_V_HALF	Vertical enlarge x1/2

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description: Sets enlarge/shrink mode for binary pattern draw.

This command can be used by all graphics LSI.

6.12 BLT Commands

6.12.1 GdcBltCopy [Copies BitBlt pattern in current draw frame]

Format: int GdcBltCopy (GDC_USHORT x0, GDC_USHORT y0,
GDC_USHORT x1, GDC_USHORT y1,
GDC_USHORT w, GDC_USHORT h)

Parameter	x0	X coordinates of the top left vertex of source Blt
	y0	Y coordinates of the top left vertex of source Blt
	x1	X coordinates of the bottom right vertex of destination
	y1	Y coordinates of the bottom right vertex of destination
	w	Horizontal width of the Blt field
	h	Vertical height of the Blt field
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Description: Draws bitmap pattern to the draw frame by Blt copy.
The destination field is current draw frame in the graphics memory.
This command can be used by all graphics LSI.

6.12.2 GdcBltCopyAlt, GdcBltCopyAltSync [Copies BitBlt pattern between any draw frame]

Format

```
int GdcBltCopyAlt (GDC_USHORT x0, GDC_USHORT y0,
        GDC_USHORT x1, GDC_USHORT y1,
        GDC_USHORT w, GDC_USHORT h,
        GDC_ULONG sadr, GDC_ULONG sstride,
        GDC_ULONG dadr, GDC_ULONG dstride)
int GdcBltCopyAltSync (GDC_USHORT x0, GDC_USHORT y0,
        GDC_USHORT x1, GDC_USHORT y1,
        GDC_USHORT w, GDC_USHORT h,
        GDC_ULONG sadr, GDC_ULONG sstride,
        GDC_ULONG dadr, GDC_ULONG dstride)
```

Parameter

x0	X coordinates of the top left vertex of source Blt
y0	Y coordinates of the top left vertex of source Blt
x1	X coordinates of the top left vertex of destination
y1	Y coordinates of the top left vertex of destination
w	Horizontal width of the Blt field
h	Vertical height of the Blt field
sadr	Memory address of the base point of the source draw frame
sstride	Stride (memory size of horizontal span) of the source draw frame
dadr	Memory address of the base point of the destination draw frame
dstride	Stride (memory size of horizontal span) of the destination draw frame

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Description

Draws bit map pattern to the draw frame by Blt copy Any draw frame in the graphics memory is applicable to source and destination. Color mode of source and destination must be the same. GdcBltCopyAltSync command is synchronously executed to the vertical blanking interval. Source and destination field must not be overlapped to each other. Clipping operation by GdcDrawClipFrame is not applicable. This command can be used by all graphics LSI.

6.12.3 GdcBltDraw [Draws BitBlt pattern]

Format: int GdcBltDraw (GDC_USHORT x, GDC_USHORT y,
GDC_USHORT w, GDC_USHORT h, GDC_LPLONG lpRect)

Parameter	x	X coordinates of the top left vertex of source Blt
	y	Y coordinates of the top left vertex of source Blt
	w	Horizontal width of the Blt field
	h	Vertical height of the Blt field
	lpRect	Pointer to refer the pattern data
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Description: Draws bitmap pattern to the draw frame by Blt copy. Source field is main memory. Color mode of the source field is assumed to be the same as that of current draw frame.
The data format of lpRect is to be the same as that of current draw frame. Maximum data size of the Blt to be transferred at a time is 65535-2 double words.
This command can be used by all graphics LSI.

6.12.4 GdcBltFill [Fills BitBlt field]

Format: int GdcBltFill (GDC_USHORT x, GDC_USHORT y,
GDC_USHORT w, GDC_USHORT h)

Parameter	x	X coordinates of the top left vertex of source Blt
	y	Y coordinates of the top left vertex of source Blt
	w	Horizontal width of the Blt field
	h	Vertical height of the Blt field
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Description: Fills a Blt field with the foreground color or tile pattern specified by GdcColor or GdcColorI command.
This command can be used by all graphics LSI.

6.12.5 GdcBltColorTransparent [Sets transparent color of transparent BitBlt]

Format	int GdcBltColorTransparent (GDC_COL16 color)	
Parameter	color	Color code treated as transparent color
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Description	Sets the transparent color referred when the Blt pattern is copied/drawn in the transparent mode. When the color mode is indirect using palette, the color mode treated as transparent color using lower 8 bit of color value. This command is only for MB86291.	

6.13 Video Capture Commands

6.13.1 GdcCapSetVideoCaptureMode [Sets mode of video capture]

Format void GdcCapSetVideoCaptureMode (GDC_ULONG mode)

Parameter mode Sets modes of video capture. The value is set to VCM (Video Capture Mode) register as it is. For detailed information about VCM register, refer to the MB86290 Series Hardware Specifications.

Macros representing each mode are prepared. These can be used as the need arises.

Please set each mode by combining the following macros.

Macros	Meaning
GDC_CAP_START	Starts capturing
GDC_CAP_STOP	Stops capturing
GDC_CAP_ENABLE_V_INTERPOLATION	Performs the interpolation of perpendicular direction
GDC_CAP_DISABLE_V_INTERPOLATION	NOT perform the interpolation of perpendicular direction
GDC_CAP_NTSC	Video= NTSC
GDC_CAP_PAL	Video=PAL

Return value None

Description Sets a value to VCM(Video Capture Mode) register, and sets video capture mode.

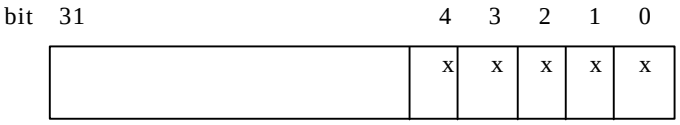
This command is only for MB86291.

6.13.2 GdcCapGetErrorStatus [Gets error status of video capture]

Format GDC_ULONG GdcCapGetErrorStatus(void)

Parameter None

Return value Video Capture Status (VCS) in the following format:



Description Reads VCS (Video Capture Status) register and returns error status.
This command is only for MB86291.

6.13.3 GdcCapClearErrorStatus [Clears error status of video capture]

Format void GdcCapClearErrorStatus(void)

Parameter None

Return value None

Description Sets 0 to VCS (Video Capture Status) register and clears error status.
This command is only for MB86291.

6.13.4 GdcCapSetVideoCaptureBuffer [Sets video capture buffer]

Format: void GdcCapSetVideoCaptureBuffer (GDC_ULONG saddr, GDC_ULONG eaddr, GDC_ULONG stride)

Parameter:

- saddr: Specifies the start address of the video capture buffer by offset value from graphics memory.
- eaddr: Specifies the end address +1 of the video capture buffer by offset value from graphics memory.
- stride: Sets width of memory (stride) for video capture buffer in blocks of 64 byte.

Return value: None

Description: Sets video capture buffer.

The start address needs to be in a 16 byte boundary.

Please specify the end address +1 of the video capture buffer as the end address.

The video capture buffer size need a size which is a part for the picture to take at least.

This command is only for MB86291.

6.13.5 GdcCapSetImageArea [Sets range of image]

Format	void GdcCapSetImageArea (GDC_USHORT x0, GDC_USHORT y0, GDC_USHORT x1, GDC_USHORT y1)	
Parameter	x0	The upper left x coordinates of the picture
	y0	The upper left y coordinates of the picture
	x1	The lower right x coordinates of the picture
v	y1	The lower right y coordinates of the picture
Return value	None	
Description	Sets the range for the image to be written to the video capture buffer. The picture of the range of (x0,y0) and (x1,y1) which are the starting point (0,0) of the upper left of the input picture is written to buffer. Please set coordinates x0<x1 and y0<y1 to specify the range of the picture. This command is only for MB86291.	

6.13.6 GdcCapSetWindowMode [Sets w-layer mode]

Format: void GdcCapSetWindowMode (GDC_ULONG format, GDC_ULONG mode,)

Parameter

format	Sets color format of w-layer.	
	Sets YC modes when using video capture.	
	GDC_CAP_RGB_MODE	RGB mode
	GDC_CAP_YC_MODE	YC mode
mode	Sets whether w-layer is used as a normal display layer or a video capture.	
	GDC_CAP_NORMAL_MODE	Normal mode
	GDC_CAP_CAPTURE_MODE	Capture mode

Return value: None

Description: Sets mode of w-layer. When using video capture, be sure to set mode.
Before execute this command, sets attribute of w-layer by GdcDispDimension beforehand. Color mode supports only 16-bit mode.
This command is only for MB86291.

6.13.7 GdcCapSetVideoCaptureScale [Sets scale of video capture]

Format: void GdcCapSetVideoCaptureScale (GDC_FIXED_SCALE hscale,
GDC_FIXED_SCALE vscale)

Parameter

hscale	horizontal scale
vscale	vertical scale

Return value: None

Description: Sets scales for reducing video capture.
Horizontal and vertical scales specify reduction scale by the style of GDC_FIXED_SCALE. Equivalent, 1/2, and 4/1 magnifications specify 1, 2 and 4 respectively. The style of GDC_FIXED_SCALE is decimal point of fixation of 5-bit of integer part and 11-bit of decimal part.
The range of value is 0xffff(1/31.99951171875 magnifications)-0x0800(equivalent magnifications). Initial value of hscale and vscale is 0x0800(equivalent magnifications) respectively.
This command is only for MB86291.

6.13.8 GdcCapSetAttrMisc [Sets attribute of video capture]

Format	Int GdcCapSetAttrMisc (GDC_ULONG target, GDC_ULONG param)		
Parameter	target	The target attribute	
		GDC_CAP_ODD_MODE	Odd number mode
		GDC_CAP_CNV_MODE	Non-interlace conversion mode
	param	Parameter belongs to each attribute (shown below)	
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	
Description	Sets attribute of video capture.		
	The attribute of video capture and the parameter are shown below.		
	This command is only for MB86291.		

GDC_CAP_ODD_MODE	Specifies the capture method	
	GDC_CAP_EVEN_AND_ODD_MODE	Captures both the odd number and the even number fields.
	GDC_CAP_ODD_ONLY_MODE	Captures only the odd number field.
GDC_CAP_CNV_MODE	Specifies the non-interlace conversion mode of the picture which is captured.	
	GDC_CAP_CNV_BOB_MODE	BOB mode (*1)
	GDC_CAP_CNV_WEAVE_MODE	WEAVE mode (*2)

(*1)BOB mode; The mode is a frame which is the even field of the raster is averaged interpolation then it is added to the odd field.

(*2)WEAVE mode; The mode is a frame which is the odd field and the even field merge on the video capture buffer.

6.13.9 GdcCapSetInputDataCountNTSC [Sets the video capture buffer for NTSC]

Format	Void GdcCapSetInputDataCountNTSC (GDC_ULONG blank_data, GDC_ULONG valid_data)	
Parameter	blank_data	The horizontal blanking interval is specified by the dot clock cycle number.
	valid_data	The data number of the term of validity is specified by the dot clock cycle number.
Return value	None	
Description	Sets the input video stream number at the time of NTSC format. This command is used to detect an error occurred. When the input data is not same as the value set up by this command, an error occurs. The video capture status becomes the value other than zero at this time. Also, capturing is continued when the error occurred This command is only for MB86291.	

6.13.10 GdcCapSetInputDataCountPAL [Sets the video capture buffer for PAL]

Format	Void GdcCapSetInputDataCountPAL (GDC_ULONG blank_data, GDC_ULONG valid_data)	
Parameter	blank_data	The horizontal blanking interval is specified by the dot clock cycle number.
	valid_data	The data number of the term of validity is specified by the dot clock cycle number.
Return value	None	
Description	Sets the input video stream number at the time of PAL format. This command is used to detect an error occurred. When the input data is not same as the value set up by this command, an error occurs. The video capture status becomes the value other than zero at this time. Also, capturing is continued when the error occurred This command is only for MB86291.	

7 System dependent Commands

7.1 Command Interface

7.1.1 GdcSetDisplayListBuffer [Sets display list buffer]

Format int GdcSetDisplayListBuffer (GDC_ULONG **base, GDC_ULONG *total_size, GDC_ULONG *num)

Parameter	base	Pointer to get the start address of the display list buffer field
	total_size	Pointer to get the size of the display list buffer field
	num	Pointer to get the block count of the display list buffer field
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Called by GdcInitialize command

Description This command sets the display list buffer information, such as the start address, size and block configuration of the display list buffer field acquired by the application program, to the “Graphics Driver” through respective pointers. The start address, size (total byte count) and number of blocks must be set to the area which are pointed by the parameters, base, total_size and num respectively. The start address of the display list buffer needs to be the host CPU address, even though the MB86290 Series controls the data transfer. 1 (single DL buffer) or 2 (double DL buffer) is applied as a number of blocks.

[Single DL buffer and double DL buffer]

In single DL buffer mode, the “Graphics Driver” uses the entire DL buffer area as one block. In this mode, when a display list transfer is started, driver command waits for the end of this transfer and then returns to the application. This means the execution time of each command depends on whether or not display list transfers occur.

In double buffer mode, the entire DL buffer area is divided into two equal blocks and these two blocks are used exclusively. When one block is fully filled with display list information, transfer of that display list is started and continuous part of the display list is filled into the other block. In this mode, by using DMA function or master function of MB86290 Series, the “Graphics Driver” does not need to check the completion of display list transfer by itself.

[Error manipulation]

If this command fails to acquire the display list buffer area, please set GDC FALSE to the return value. In this case, GdcInitialize command ends as initialization failure (GDC_FALSE).

[Remark]

- The size of display list buffer must be a multiple of 32byte.

This command can be used by all graphics LSI.

Example

```
#define   BUF_NUM       2
#define   BUF_SIZE      ((2*256*256+32)*BUF_NUM)

int GdcSetDisplayListBuffer(GDC_ULONG **base, GDC_ULONG *total_size,
GDC_ULONG *num){
    /* Acquisition of display list buffer area */
    if( (*base = (GDC_ULONG *)malloc(BUF_SIZE)) == NULL )
        return(GDC_FALSE);          /* Acquisition failure */

        *total_size = BUF_SIZE;
        *num = BUF_NUM;
        return(GDC_TRUE);                   /* Acquisition complete */
}
```

7.1.2 GdcFlushDisplayList [Transfers a display list]

Format	void GdcFlushDisplayList (GDC_ULONG *src, GDC_ULONG count)	
Parameter	src	Source address (display list buffer)
	count	Transfer count
Return value	None	
Called by	GdcSync command	
	GdcFlush command	
	All draw commands	

Description

This command is to transfer a display list of the size specified by “count” started from the source address specified by “src”. The “src” specifies the display list buffer address mapped to the host CPU address field. The unit of “count” is what specified by GdcSetDMAMode command (32byte or 4byte). If GdcSetDMAMode command is not applied since DMA is not used, this unit is set to 4 byte. For the display list transfer, the following three methods are applicable. For each procedure, please refer the description [Display list transfer procedure] as follows:

- DMA transfer
- Master transfer of MB86290 Series
- CPU transfer

This command can be used by all graphics LSI.

[Display list transfer procedure]

*** DMA transfer**

This is a method of display list transfers utilizing the DMA controller of the host CPU (MB86290 Series does not contain a DMA controller). The operation procedure of this case is shown as follows. Prior to call this command, DMA transfer mode must be appropriately set on both DMAC (the host CPU) and MB86290 Series.

(1) Check DMA transfer enable/disable

- Check the appropriate operation mode check of the DMAC and wait till it will be ready to accept a new DMA transaction request.

(2) Set DMA (According to the applied procedure for the DMAC, set the following parameter)

- Source address (the address specified in “src”)
- Destination address (Display list FIFO of the MB86290 Series)
- Transfer count (the value specified in “count”)

(3) Set transfer count (the MB86290 Series side)

- Set DMA transfer count (the value specified in “count”) to DTC (DMA Transfer Count) register

(4) Start DMA transaction

- Appropriate start up operation for the applied DMA controller

(5) Issue the DMA request

- Set 1 to DRQ (DMA ReQuest) register

(6) Wait for the completion of the DMA transfer

- Single DL buffer mode is applied, wait till the end of DMA transaction

[Remark]

When the unit of transfer count is 32byte, if the total byte size of the display list is not a multiple of 32byte, the driver command fills appropriate number of NOP and makes the size to be a multiple of 32byte.

[Example]

```
/* Start address of the host interface register field */
#define HOSTBASE0x########

/* Start address of drawing control register field */
#define DRAWBASE          0x########

#define WRITE_DTC(i)      ( *(GDC_ULONG*)(HOSTBASE+0x00) = (i) )
#define WRITE_DRQ(i)      ( *(GDC_ULONG*)(HOSTBASE+0x18) = (i) )
#define FIFO_ADDRESS      (DRAWBASE+0x4a0)

void GdcFlushDisplayList(GDC_ULONG *src, GDC_ULONG count){
    /* Polling for DMA ready DMA */
    while( DMA_BUSY() );

    /* Transfer count set */
    SET_DMA_COUNT(CHANNEL0, count);

    /* Source address set */
    SET_DMA_SRC(CHANNEL0, src);

    /* Destination address set */
    SET_DMA_DEST(CHANNEL0, FIFO_ADDRESS);

    /* Transfer count set (MB86290 Series) */
    WRITE_DTC(count);

    /* Trigger of DMA transaction */
    DMA_START();

    /* Issue of external DMA request */
    WRITE_DRQ(1);

#ifdef SINGLE_DL_BUFFER
    /* Wait for the next display list buffer write to be ready */
    while( DMA_BUSY() );
#endif
}
```

***Master transfer of MB86290 Series**

This is a method of the display list transfers utilizing the bus master function of MB86290 Series. Transfer count is 4byte unit. In this case, the display list buffer must be located to the graphics memory of MB86290 Series. And the source address "src" must be converted to the local address of MB86290 Series. The operation procedure of this case is shown as follows:

(1) Check transfer enable/disable

- Check the status of LSTA (display List transfer STAtus) register and wait until it will be 0.

(2) Set source address

- Set the source address to LSA (display List Source Address) register. The address to be set to this register is

("src" value) – (start address of host interface register field)

(3) Set transfer count

- Set the transfer count ("count" value) to LCO (display List COunt) register

(4) Start the transaction

- Set 1 to LREQ (display List transfer REQuest) register

(5) Wait for the completion of the transfer (in case of single DL buffer mode)

- Same as (1)

[Example]

```
/* Start address of host interface register field */
#define HOSTBASE            0x########

/* Start address of graphic memory field */
#define MB86290_BASE        0x########

#define READ_LSTA()         *((volatile GDC_ULONG*)(HOSTBASE+0x10))
#define WRITE_LSA(i)        ( *((GDC_ULONG*)(HOSTBASE+0x40)) = (i) )
#define WRITE_LCO(i)        ( *((GDC_ULONG*)(HOSTBASE+0x44)) = (i) )
#define WRITE_LREQ(i)       ( *((GDC_ULONG*)(HOSTBASE+0x48)) = (i) )

void GdcFlushDisplayList(GDC_ULONG *src, GDC_ULONG count){
    GDC_ULONG       src_local;

    /* Polling of transfer ready */
    while( READ_LSTA() );

    /* Source address set */
    src_local = (GDC_ULONG)src – MB86290_BASE;
    WRITE_LSA(src_local);

    /* Transfer count set */
    WRITE_LCO(count);
```

```
        /* Trigger */
        WRITE_LREQ(1);

    #ifdef SINGLE_DL_BUFFER
        /* Wait for next the display list buffer write to be ready */
        while( READ_LSTA() );
    #endif
    }
```

***CPU transfer**

This is a method to write the transfer data (display list) to the display list FIFO of MB86290 Series by software. The operation procedure of this case is shown as follows. Repeat (1) through (4) for the times specified by “count”.

(1) Acquire the display list FIFO status

- Call GdcGetFIFOStatus command and acquire the display list FIFO status information.

(2) Check the display list FIFO status

- Check the empty entries of the display list FIFO from the above status information. If FIFO is full, keep repeating (1) and (2) till open entries will be available.

(3) Transfer 4byte of data from the source address to the display list FIFO

(4) Post increment (+4) source address

[Example]

```
/* Start address of drawing control register field */
#define DRAWBASE            0x########
#define WRITE_FIFO(i)       ( *(volatile GDC_ULONG*)(DRAWBASE+0x4a0) = (i) )
#define FIFO_FULL           0x2

void GdcFlushDisplayList(GDC_ULONG *src, GDC_ULONG count){
    int    i;

    for(i = 0; i < count; i++){
        /* If FIFO is full, wait until open entry will be available */
        while(GdcGetFIFOStatus() & FIFO_FULL);

        /* Transfer data to the FIFO */
        WRITE_FIFO(*src++);
    }
}
```

7.1.3 GdcGetHostRegisterAddress [Gets host interface register area address]

Format	GDC_ULONG *GdcGetHostRegisterAddress (void)
Parameter	None
Return value	Start address of host interface register field
Called by	GdcInitialize command
Description	Start address of host interface register field is returned. This command can be used by all graphics LSI.

7.1.4 GdcGetDispRegisterAddress [Gets display control register area address]

Format	GDC_ULONG *GdcGetDispRegisterAddress (void)
Parameter	None
Return value	Start address of display control register field
Called by	GdcInitialize command
Description	Start address of display control register field is returned. This command can be used by all graphics LSI.

7.1.5 GdcGetDrawRegisterAddress [Gets draw control register area address]

Format	GDC_ULONG *GdcGetDrawRegisterAddress (void)
Parameter	None
Return value	Start address of drawing control register field
Called by	GdcInitialize command
Description	Start address of drawing control register field is returned. This command can be used by all graphics LSI.