SI0000Q320J01-04

MB86290 Series Graphics Driver Users Manual Rev.2.3

FUJITSU LIMITED

1. The contents of this document are subject to change without notice. Therefore, please confirm that information given in this document is the newest in Fujitsu Limited sales representatives in the case of using.

2. Fujitsu Limited is unable to assume responsibility for infringement of any patent rights or other rights of third parties arising from the use of this information or figures.

3. No part of the publication may be copied reproduced in any form or by any means, or transferred to any third party without prior written consent of Fujitsu Limited.

4. If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

Revision history

Date	Rev.	Page	Change
Feb. 23,2000	1.0	87	(1)First release
May 18, 2001	1.3	116	(1)Adds API for object coordinates (for MB86291) (2)Adds API for video capture (for MB86291) (3)Adds API for I^{2}C control (for MB86291S)
Jun. 18, 2001	1.4	118	(1)Adds GdcCapSetLPFMode command (2)The number of transmission byte is set as the return value of GdcFlush, GdcSync, GdcVFlush, and GdcVSync (3)Adds GdcVerticalSync command
Aug. 15, 2001	1.5	113	(1)Deletes GdcGeoSync,GdcGeoInterrupt,GdcGeoGetFIFOStatus,GdcGetGetFIFORemain,GdcGeoGetFIFOErrorStatus,andGdcGeoClearFIFOErrorStatus command
Sept. 13, 2001	1.6	113	(1)Adds argument (gdc_type) to the GdcGeoInitialize command

(Revision history:1/3)

Revision history

Date	Rev.	Page	Change
June, 13, 2002	1.7	169	(1)Adds GDC_TYPE_MB86291A macro to appoint with an argument of the GdcGeoInitialize command (2)Changes the font (3)Adds commands for MB86293 (4)Adds the GdcClipMode command, and delete the GdcSetAttrMisc command of the same function (5)Added argument (gdc_type) to the GdcInitialize command (6)Deleted argument (gdc_type) from the GdcGeoInitialize command (7)Changed argument name (dma_request) of the GdcSetDMAMode command (8)Added the GdcCancelDisplayList command (9)Added a drawing order control mode setting function of main / border / shadow primitive to the GdcGeoSetAttrLine command (10)Changed a sentence of notes (11)Updated by an abstract description of the “Graphics Driver” (12)Added a description of the GdcCapSetVideoCaptureScale command (13)Updated from GDC_COL32 to GDC_COLOR32 by a type description of an argument used with the following commands - GdcColorTransparent command - GdcColor command - GdcBackColor command - GdcGeoShadowColor command - GdcGeoShadowBackColor command - GdcGeoBorderColor command - GdcGeoBorderBackColor command (14)Deleted the GdcColorI command and the GdcBackColorI command (8 bits and 16 bits color can be set by the GdcColor and the GdcBackColor respectively) (15)Changed a type of an argument used with the following commands from the GDC_COL16 to the GDC_COLOR32 - GdcSetTextureBorder command - GdcBltColorTransparent command (16)Because could clear a Display List FIFO error only by reset, deleted the GdcClearFIFOErrorStatus command

Revision history

Date	Rev.	Page	Change
June, 13, 2002	1.7	169	(17)The following commands used in order to delete the GdcSync and a GdcVSync command and to check drawing end instead are added. - GdcGetPixelEngineStatus - GdcGeoGetPixelEngineStatus - GdcGetLocalDisplayListTransferStatus
June, 27, 2002	2.3	171	(1)Description of GdcGeoLoadMatrix command is corrected.

Introduction

*A purpose and a target reader of this document

This document describes the mechanisms of MB86290 Series Graphics Driver (the “Graphics Driver”) and application interfaces.

This document is written for an engineer developing a graphics application using the “Graphics Driver”.

A description of this document has premised the reader who has understood specification of MB86290 Series Graphics Controller (the “Graphics Controller”) and technology about graphics.

If needed refer to the specification of Graphics Controller, or graphics-related books.

*Graphics-related technology which will be the requisite for an understanding of this document

Device coordinate system, object coordinate system, coordinate transformation, conversion matrix, clipping, polygon, shading, Z-buffer method, shade surface elimination, texture mapping, tilling, anti-aliasing, alpha blending, chrome-key composition, palette color, etc.

*Specifications of the “Graphics Controller”

For hardware specifications of the “Graphics Controller” and programming, refer to the following documents.

- Graphics Controller Specifications
- Application Note

These documents are prepared for every “Graphics Controller”.

Each “Graphics Controller” and document names are described in the table 1.

Table 1. List of documents

Graphics Controller	Document title
MB86290A	MB86290A Graphics Controller Hardware Specifications
	Cremson Application Note
MB86291/86291S	MB86291 <SCARLET> Graphics Controller Specifications
	MB86291 <SCARLET> Application Note
	I^{2}C Interface Specification
MB86291A	MB86291A <SCARLET2> Graphics Controller Specifications
	MB86291 <SCARLET> Application Note
MB86292/86292S	MB86292 <ORCHID> Graphics Controller Specifications
	MB86292 <ORCHID> Application Note
	I^{2}C Interface Specification
MB86293	MB86293 <CORAL-LQ> Graphics Controller Specifications

A specific “Graphics Controller” group may be shown as follows in this document.

MB86291 or later : MB86291/86291S/86291A/86292/86292S/86293 Graphics Controller

MB86291/86292 : MB86291/86291S/86291A/86292/86292S Graphics Controller

MB86293 or later : MB86293 Graphics Controller or later

Index

1 MB86290 Series Graphics Driver Overview

This section describes abstract of the MB86290 Series Graphics Driver.

1.1 Overview

The MB86290 Series Graphics Driver (the "Graphics Driver") is a set of commands to assist graphics application programs utilizing the MB86290 Series Graphics Controller (the "Graphics Controller"). Each command to use the "Graphics Driver" from a graphics application program is called driver command. Abstract of the "Graphics Driver" is shown by figure 1.1.

By using this Graphics Driver, graphics application programs can be made without concerning the code to access to hardware registers and management of Display List (refer to "2.1 Rendering Scheme").

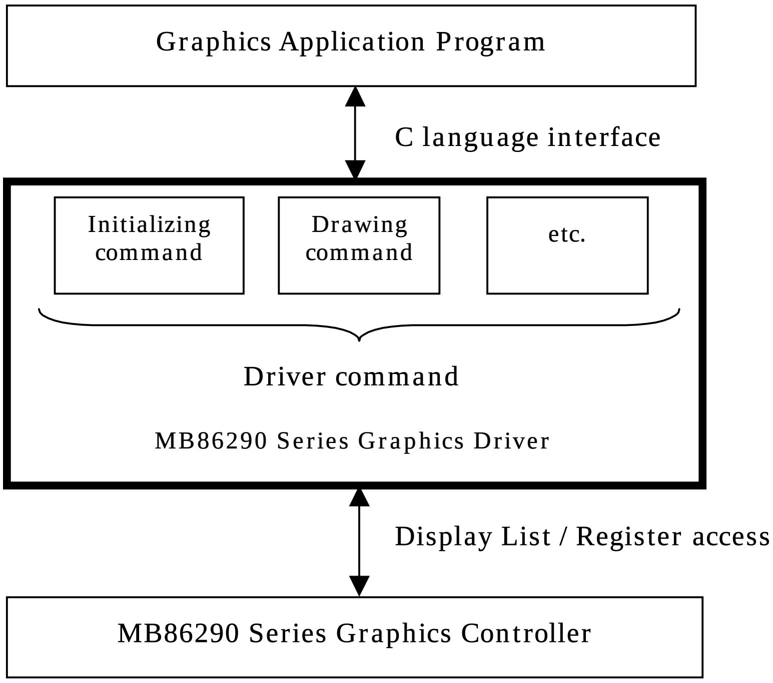

Graphics Application Program
C language interface
Initializing command
Drawing command
etc.
Driver command
MB86290 Series Graphics Driver
Display List / Register access
MB86290 Series Graphics Controller

Figure 1.1 Overview of the "Graphics Driver"

2 Operation of MB86290 Series Graphics Driver

This section describes rendering scheme by the “Graphics Driver”.

2.1 Rendering Scheme

Drawing operation of the “Graphics Controller” is started by sending respective command and it’s parameters. 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 transfer or transfer of Local Display List. This bunch of commands is called Display List.

Rendering scheme is shown in the figure 2.1. The drawing function of the “Graphics Driver” once stores the generated Display List in Display List buffer (refer to "2.2 Management of Display List") in order to transmit a Display List collectively by DMA etc (1). Then, transferring to the “Graphics Controller” is started with Display List transferring command (2).

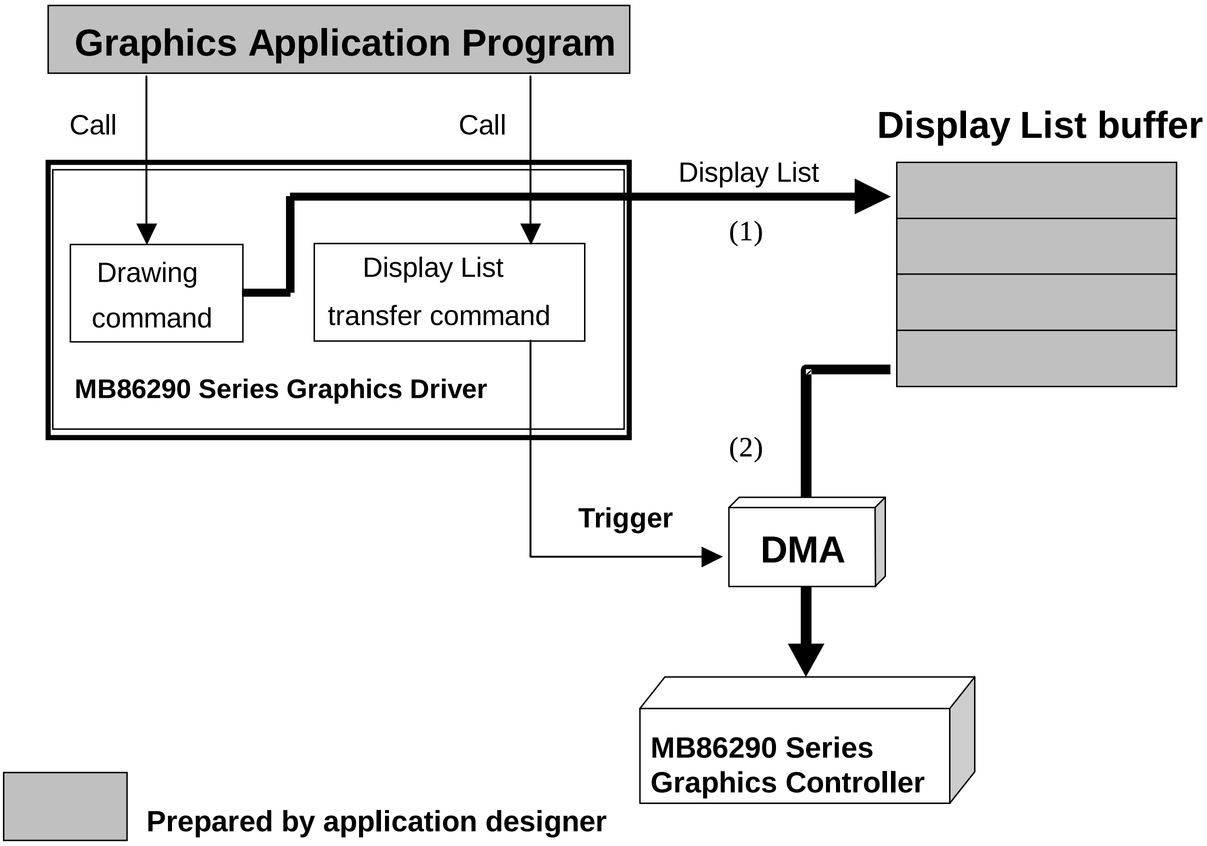

Graphics Application Program
Call
Call
Display List buffer
Display List
(1)
Drawing command
Display List transfer command
MB86290 Series Graphics Driver
(2)
Trigger
DMA
MB86290 Series Graphics Controller
Prepared by application designer

Figure 2.1 Rendering scheme

2.2 Management of Display List

The “Graphics Driver” stores generated Display List to Display List buffer and manages them. The Display List buffer is a certain amount of memory area allocated on either the local memory (the graphics memory) of the “Graphics Controller” or the host CPU memory. Acquisition of this area is done by graphics application program. Created 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 Display List transfer is referred to “2.3.2 Trigger of a Display List transfer”).

Display List buffer has the method of using the whole as one area, and the method of dividing two and using (refer to figure 2.2). 1 block configuration and 2 blocks configuration are described in the following.

(1)1 block configuration

In 1 block configuration, the whole Display List buffer is used as one area. 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.

(2)2 blocks configuration

In 2 blocks configuration, each block is used alternatively. In 2 blocks configuration, when 1 block is completely full of Display List, transfer is started. But prior to wait this transfer start, following Display List can be stored into the alternative block.

The biggest advantage of this 2 blocks configuration is that the host CPU can create a new Display List while the “Graphics Controller” executes rendering operations according to the current Display List. To make this scheme, the Display List needs to be transferred to the “Graphics Controller” via DMA or transfer of Local Display List.

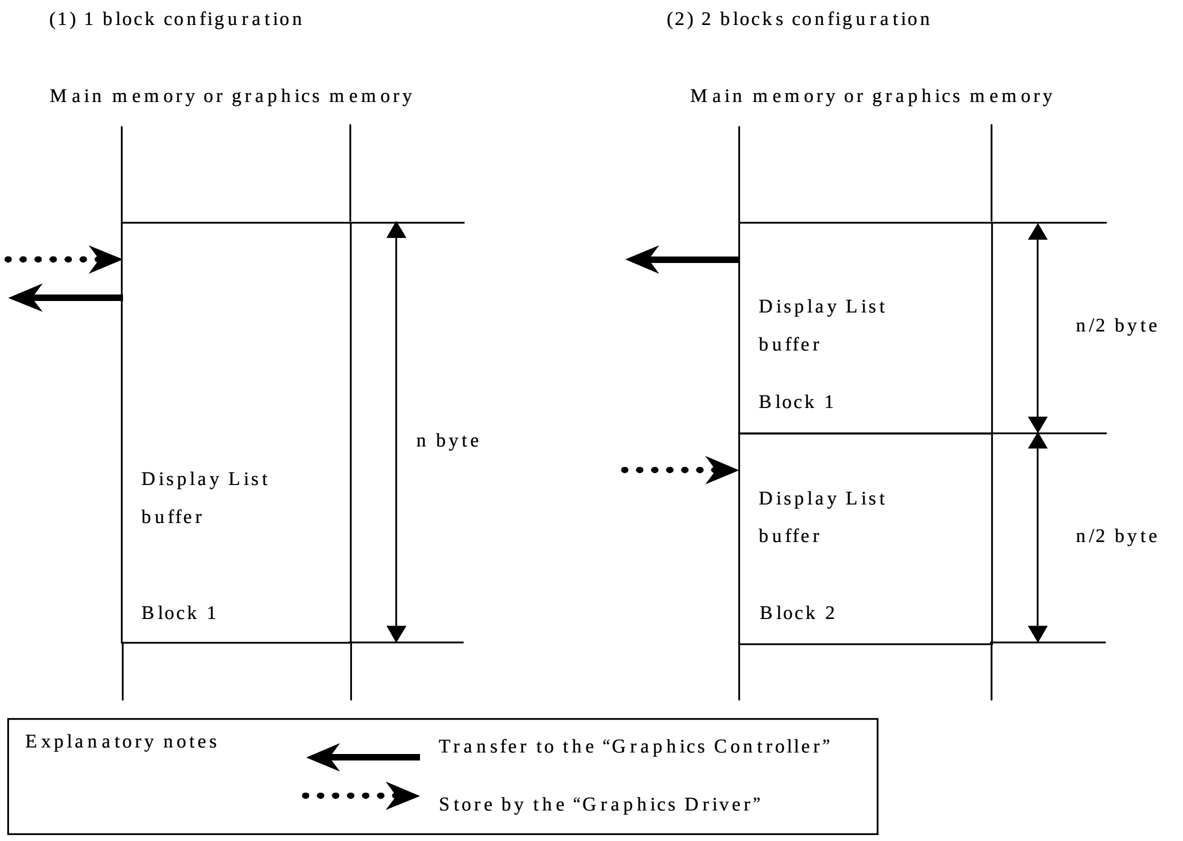

(1) 1 block configuration
Main memory or graphics memory
n byte
Display List
buffer
Block 1
(2) 2 blocks configuration
Main memory or graphics memory
Display List
buffer
Block 1
n/2 byte
Display List
buffer
Block 2
n/2 byte
Explanatory notes
Transfer to the “Graphics Controller”
Store by the “Graphics Driver”

Figure 2.2 Configuration of Display List buffer

2.3 Transfer of Display List

2.3.1 Methods of Display List transfer

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
- Transfer of Local Display List
- 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
- GdcVFlush 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. Default mode is async mode. Switching of Sync mode and Async mode is performed by the GdcExecMode command.

(1)Async mode

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 Display List transfer”. In this mode, Display List creation by the “Graphics Driver”, Display List transfer to the “Graphics Controller” and rendering, and it’s execution work independently.

(2)Sync mode

In Sync mode, regardless the method of the Display List transfer, each driver command transfers it’s generated Display List to the “Graphics Controller” immediately, and returns back to the application after the completion of the “Graphics Controller” rendering operations according to the Display List. This mode is mainly used in the debug of graphics application programs.

2.5 System Dependent Commands

The system dependents command is a command to process procedures such as DMA transfer that depends on a target system and a graphics application program in the “Graphics Driver”. The relationship between system dependent commands and the “Graphics Driver” is shown in the figure 2.5.

The system dependent commands must be designed by each application designer according to the command interface specified by the “Graphics Driver”. (the command interface of the system dependent commands is referred to “8. System Dependent Commands Interface”.)

The following procedures must be implemented in the system depend commands.

(1) Acquisition of the mapping address of the “Graphics Controller” register areas

Gets the address allocation information of the “Graphics Controller” register areas and feed these information back to the “Graphics Driver” to access to various registers of the “Graphics Controller”.

(2) Setting of 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

Transfers Display List to the “Graphics Controller” according to method of “2.3.1 Methods of Display List transfer."

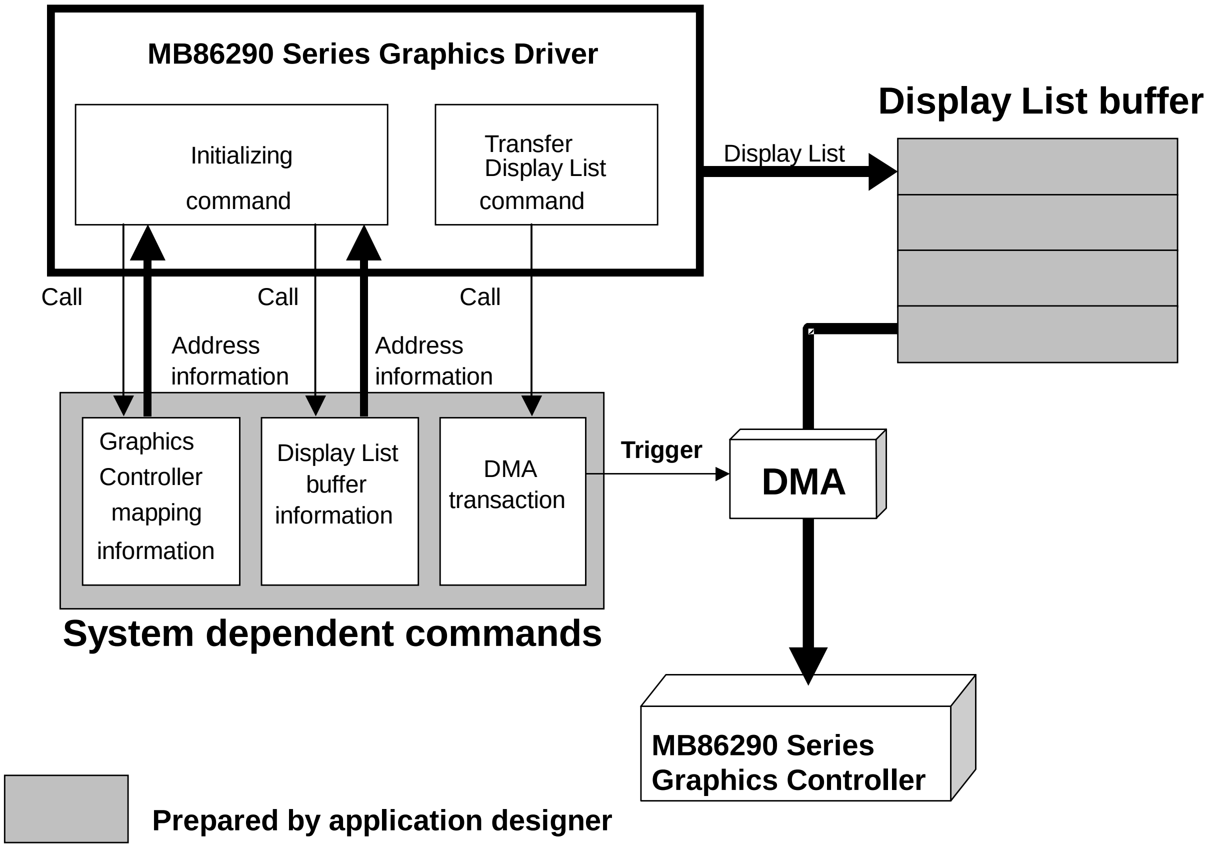

MB86290 Series Graphics Driver
Initializing command
Transfer Display List command
Display List
Display List buffer
Call
Call
Call
Address information
Address information
Graphics Controller mapping information
Display List buffer information
DMA transaction
Trigger
DMA
System dependent commands
MB86290 Series Graphics Controller
Prepared by application designer

Figure 2.5 System dependent commands and relation with the "Graphics Driver"

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 Header files

The “Graphics Driver” is prepared the following header files to application designers.

Whenever driver command is called, gdc.h must be included. Since the gdctypes.h is already included in the gdc.h, application designers do not need to include it directly.

- gdc.h	: Declaration of driver command prototype
- gdctypes.h	: Definition of the data type applied in the “Graphics Driver”

3.1.2 Gets Display List buffer

The Display List buffer must be acquired by the application program. The buffer size should be 32byte boundary. When DMA is applied to transfer Display List, the address allocation of the buffer 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, DMA transfer source address of the 1st block is a top address of Display List buffer. And DMA transfer source address of the 2nd block is “top address+1/2 byte count of the total Display List buffer).

3.1.3 Creates system dependent commands

The system dependent commands should be designed according to the command interface specified in “8. System Dependent Commands”.

3.2 Reminder

3.2.1 Prohibition of re-entrant

The “Graphics Driver” is not supported to re-entrant. NOT to call the “Graphics Driver” from multiple tasks. If multiple tasks 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

This section describes each “Graphics Controller” can use any driver commands.

Driver commands are shown in the following.

- System control commands
- Display commands
- Color control commands
- Cursor control commands
- Drawing frame control commands
- Primitive drawing control commands for device coordinates
- Primitive drawing control commands for object coordinates
- Drawing attribute control commands
- Attribute control commands for object coordinates
- Texture pattern management commands
- Binary pattern drawing commands
- Blt commands
- Video capture control commands
- I^{2}C control commands

4.1 System Control Commands

System control commands list is shown in the table 4.1.

Table 4.1 System control commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcInitialize	"Graphics Driver" initialization	Y	Y	Y
2	GdcQueryVersion	Version number check of the "Graphics Driver"	Y	Y	Y
3	GdcSetInternalClock	Changes internal clock frequency	N	N	Y
4	GdcInitDevice	Initializes the "Graphics Controller"	Y	Y	Y
5	GdcGeoInitialize	Initialize geometry engine	N	Y	Y
6	GdcFlush	Drawing by Display List (Async)	Y	Y	Y
7	GdcVFlush	Vertical blanking interval palling (Async)	Y	Y	Y
8	GdcVerticalSync	Adds vertical blanking interval command	Y	Y	Y
9	GdcInterrupt	Interrupts request to host CPU	Y	Y	Y
10	GdcExecMode	Sets execution mode of Display List operation	Y	Y	Y
11	GdcSetDMAMode	Sets DMA mode	Y	Y	Y
12	GdcGetFIFOStatus	Gets Display List FIFO status	Y	Y	Y
13	GdcGetFIFORemain	Gets number of Display List FIFO open entries	Y	Y	Y
14	GdcGetFIFOErrorStatus	Gets Display List FIFO error status	Y	Y	Y
15	GdcGetInterruptStatus	Gets interrupt status	Y	N	N
16	GdcGeoGetInterruptStatus	Gets interrupt status	N	Y	Y
17	GdcClearInterruptStatus	Clears interrupt status for MB86290A	Y	N	N

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
18	GdcGeoClearInterruptStatus	Clears interrupt status for MB86291 or later	N	Y	Y
19	GdcSetInterruptMask	Sets interrupt mask for MB86290A	Y	N	N
20	GdcGeoSetInterruptMask	Sets interrupt mask for MB86291 or later	N	Y	Y
21	GdcGeoGetFIFOStatus	Gets geometry Display List FIFO status	N	Y	Y
22	GdcGeoGetFIFORemain	Gets number of geometry Display List FIFO open entries	N	Y	Y
23	GdcSetMemoryMode	Sets memory interface mode	Y	Y	Y
24	GdcSoftwareReset	Resets by software	Y	Y	Y
25	GdcGetErrCode	Gets error code	Y	Y	Y
26	GdcSetRegisterLocation	Changes address of registers location	N	N	Y
27	GdcSetBurstMode	Sets burst transfer mode of drawing	N	N	Y
28	GdcQueryChipID	Queries about chip ID	N	N	Y
29	GdcCancelDisplayList	Cancels Display List	Y	Y	Y
30	GdcGetPixelEngineStatus	Gets pixel engine status	Y	N	N
31	GdcGeoGetPixelEngineStatus	Gets geometry pixel engine status	N	Y	Y
32	GdcGetLocalDisplayListTransferStatus	Gets transfer of Local Display List status	Y	Y	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.2 Display Commands

Display commands list is shown in the table 4.2.

Table 4.2 Display commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcDispClock	Sets display clock mode	Y	Y	Y
2	GdcDispTiming	Sets display timing parameters	Y	Y	Y
3	GdcDispTimingWindow	Sets W-layer display position	Y	Y	Y
4	GdcDispDividePos	Sets border position of screen partition	Y	Y	Y
5	GdcDispDimension	Sets display frame attribute	Y	Y	Y
6	GdcDispOn	Asserts video signal output	Y	Y	Y
7	GdcDispOff	Negates video signal output	Y	Y	Y
8	GdcDispLayerOn	Asserts screen display	Y	Y	Y
9	GdcDispLayerOff	Negates screen display	Y	Y	Y
10	GdcDispPos	Sets display start position	Y	Y	Y
11	GdcDispDoFlip	Flips display bank	Y	Y	Y
12	GdcOverlayPriorityMode	Sets overlay display mode	Y	Y	Y
13	GdcOverlayBlend	Sets blend parameter for overlay blend	Y	Y	Y
14	GdcDispDisplayMode	Sets display mode	N	N	Y
15	GdcDispDisplayLayerMode	Sets layer display mode	N	N	Y
16	GdcDispSetBackColor	Sets background color	N	N	Y
17	GdcDispSetLayerWindow	Sets position and size of the window mode layer	N	N	Y
18	GdcLayerOverlayPriorityMode	Sets overlay display mode in every layer	N	N	Y
19	GdcLayerOverlayBlend	Sets blend mode in every layer	N	N	Y
20	GdcDisplLayerOrder	Sets layer display order	N	N	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.3 Color Control Commands

Color control commands list is shown in the table 4.3.

Table 4.3 Color control commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcColorPalette	Sets palette colors	Y	Y	Y
2	GdcColorTransparent	Sets transparent color	Y	Y	Y
3	GdcColorZeroMode	Sets color code 0 mode	Y	Y	Y
4	GdcChromaKeyMode	Sets Chroma-key mode	Y	Y	Y
5	GdcColorKey	Sets key color for Chroma-key	Y	Y	Y
6	GdcColorPaletteOffset	Sets of the color palette offset	N	N	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.4 Cursor Control Commands

Cursor control commands list is shown in the table 4.4.

Table 4.4 Cursor control commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcCursorAddress	Sets cursor pattern memory address	Y	Y	Y
2	GdcCursorPattern	Sets cursor pattern	Y	Y	Y
3	GdcCursorDisplay	Controls cursor display	Y	Y	Y
4	GdcCursorPos	Sets cursor display position	Y	Y	Y
5	GdcCursorPriority	Sets cursor display priority mode	Y	Y	Y
6	GdcCursorColorTransparent	Sets cursor transparent color	Y	Y	Y
7	GdcCursorColorZeroMode	Sets cursor color code 0 mode	Y	Y	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.5 Drawing Frame Control Commands

Drawing frame control commands list is shown in the table 4.5.

Table 4.5 Drawing frame control commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcDrawDimension	Sets drawing frame	Y	Y	Y
2	GdcSetZPrecision	Sets precision of Z value	N	N	Y
3	GdcBufferCreateZ	Sets Z buffer start address	Y	Y	Y
4	GdcBufferCreateC	Sets start address of polygon drawing flag buffer	Y	Y	Y
5	GdcBufferClearZ	Clears Z buffer	Y	Y	Y
6	GdcBufferClearC	Clears polygon drawing flag buffer	Y	Y	Y
7	GdcDrawClipFrame	Sets drawing clip border	Y	Y	Y
8	GdcSetAlphaMapBase	Sets alpha map area start address	N	N	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.6 Primitive Drawing Control Commands for Device Coordinates

Primitive drawing control commands for device coordinates list is shown in the table 4.6.

Table 4.6 Primitive drawing control commands for device coordinates list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcPrimType	Sets drawing procedure	Y	Y	Y
2	GdcPrimEnd	Completes drawing procedure	Y	Y	Y
3	GdcTexCoord2D	Sets coordinates of 2D texture (GDC_FIXED32 type)	Y	Y	Y
4	GdcTexCoord2Df	Sets coordinates of 2D texture (GDC_SFLOAT type)	Y	Y	Y
5	GdcTexCoord2DNf	Sets normalized coordinates of 2D texture (GDC_SFLOAT type)	Y	Y	Y
6	GdcTexCoord3D	Sets coordinates of 3D texture (GDC_FIXED32 type)	Y	Y	Y
7	GdcTexCoord3Df	Sets coordinates of 3D texture (GDC_SFLOAT type)	Y	Y	Y
8	GdcTexCoord3DNf	Sets normalized coordinates of 3D texture (GDC_SFLOAT type)	Y	Y	Y
9	GdcDrawVertex2D	Sets coordinates of 2D vertex (GDC_FIXED32 type)	Y	Y	Y
10	GdcDrawVertex2Di	Sets coordinates of 2D vertex (GDC_LONG type)	Y	Y	Y
11	GdcDrawVertex3D	Sets coordinates of 3D vertex (GDC_FIXED32 type)	Y	Y	Y
12	GdcDrawVertex3Df	Sets coordinates of 3D vertex (GDC_SFLOAT type)	Y	Y	Y
13	GdcDrawPrimitive	Draws multiple 3D triangles	Y	Y	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.7 Primitive Drawing Control Commands for Object Coordinates

Primitive drawing control commands for object coordinates list is shown in the table 4.7.

Table 4.7 Primitive drawing control commands for object coordinates list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcGeoPrimType	Sets drawing procedure	N	Y	Y
2	GdcGeoPrimEnd	Completes drawing procedure	N	Y	Y
3	GdcGeoDrawVertex2D	Sets XY coordinates of vertex (GDC_FIXED32 type)	N	Y	Y
4	GdcGeoDrawVertex2Df	Sets XY coordinates of vertex (GDC_SFLOAT type)	N	Y	Y
5	GdcGeoDrawVertex2Di	Sets XY coordinates of vertex (GDC_LONG type)	N	Y	Y
6	GdcGeoDrawVertex3D	Sets XYZ coordinates of vertex (GDC_FIXED32 type)	N	Y	Y
7	GdcGeoDrawVertex3Df	Sets XYZ coordinates of vertex (GDC_SFLOAT type)	N	Y	Y
8	GdcGeoDrawVertex3Di	Sets XYZ coordinates of vertex (GDC_LONG type)	N	Y	Y
9	GdcGeoTexCoord2DN	Sets texture coordinates (GDC_FIXED32 type)	N	Y	Y
10	GdcGeoTexCoord2DNf	Sets texture coordinates (GDC_SFLOAT type)	N	Y	Y
11	GdcVertexColor3f	Sets color of vertex (GDC_SFLOAT type)	N	Y	Y
12	GdcVertexColor32	Sets color of vertex (GDC_COLOR32 type)	N	Y	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.8 Drawing Attribute Control Commands

Drawing attribute control commands list is shown in the table 4.8.

Table 4.8 Drawing attribute control commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcColor	Sets vertex color/foreground color	Y	Y	Y
2	GdcBackColor	Sets background color	Y	Y	Y
3	GdcClipMode	Sets clipping mode	Y	Y	Y
4	GdcSetAttrLine	Sets line drawing attribute	Y	Y	Y
5	GdcSetAttrSurf	Sets surface drawing attribute	Y	Y	Y
6	GdcSetAttrTexture	Sets texture mapping attribute	Y	Y	Y
7	GdcSetAttrBlt	Sets BitBlt attribute	Y	Y	Y
8	GdcSetAlpha	Sets alpha blend ratio	Y	Y	Y
9	GdcSetLinePattern	Sets broken line pattern	Y	Y	Y
10	GdcSetTextureBorder	Sets texture border color	Y	Y	Y
11	GdcSetRop	Sets logical calculation mode	Y	Y	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.9 Attribute Control Commands for Object Coordinate

Attribute control commands for object coordinate list is shown in the table 4.9.

Table 4.9 Attribute control commands for object coordinate list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcGeoSetAttrMisc	Sets miscellaneous attribute	N	Y	Y
2	GdcGeoSetAttrLine	Sets line drawing attribute for object coordinates	N	Y	Y
3	GdcGeoSetAttrSurf	Sets surface drawing attribute for object coordinates	N	Y	Y
4	GdcGeoLoadMatrix	Sets matrix (GDC_FIXED32 type)	N	Y	Y
5	GdcGeoLoadMatrixf	Sets matrix (GDC_SFLOAT type)	N	Y	Y
6	GdcGeoNdcDcViewportCoef	Sets coefficients of NdcDc transformation for xy (GDC_FIXED32 type)	N	Y	Y
7	GdcGeoNdcDcViewportCoeff	Sets coefficients of NdcDc transformation for xy (GDC_SFLOAT type)	N	Y	Y
8	GdcGeoNdcDcDepthCoef	Sets coefficients of NdcDc transformation for z (GDC_FIXED32 type)	N	Y	Y
9	GdcGeoNdcDcDepthCoeff	Sets coefficients of NdcDc transformation for z (GDC_SFLOAT type)	N	Y	Y
10	GdcGeoViewVolumeXYClip	Sets view volume boundary for xy (GDC_FIXED32 type)	N	Y	Y
11	GdcGeoViewVolumeXYClipf	Sets view volume boundary for xy (GDC_SFLOAT type)	N	Y	Y
12	GdcGeoViewVolumeZClip	Sets view volume boundary for z (GDC_FIXED32 type)	N	Y	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
13	GdcGeoViewVolumeZClipf	Sets view volume boundary for z (GDC_SFLOAT type)	N	Y	Y
14	GdcGeoViewVolumeWminClip	Sets view volume boundary for w (GDC_FIXED32 type)	N	Y	Y
15	GdcGeoViewVolumeWminClipf	Sets view volume boundary for w (GDC_SFLOAT type)	N	Y	Y
16	GdcGeoSetLogOutBase	Sets top address for log output of device coordinate	N	N	Y
17	GdcGeoSetLogOutMode	Sets log output mode of the device coordinate	N	N	Y
18	GdcGeoShadowXY	Sets xy offset of shadow	N	N	Y
19	GdcGeoOverlapZ	Sets Z value of primitives (body / shadow / border / correction in top-left rule non-applicable mode)	N	N	Y
20	GdcGeoShadowColor	Sets color of shadow	N	N	Y
21	GdcGeoShadowBackColor	Sets background color of shadow	N	N	Y
22	GdcGeoBorderColor	Sets color of border	N	N	Y
23	GdcGeoBorderBackColor	Sets background color of border	N	N	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.10 Texture Pattern Management Commands

Texture pattern management commands list is shown in the table 4.10.

Table 4.10 Texture pattern management commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcTextureMemoryMode	Sets texture memory mode	Y	Y	Y
2	GdcTextureLoadInt	Loads texture/tile pattern to internal texture memory	Y	Y	Y
3	GdcTextureLoadExt	Loads texture pattern to the graphics memory	Y	Y	Y
4	GdcTextureLoadExt8	Loads 8bpp texture pattern to the graphics memory	N	N	Y
5	GdcTextureLoadExt16	Loads 16bpp texture pattern to the graphics memory	Y	Y	Y
6	GdcTextureLoadExt24	Loads 24bpp texture pattern to the graphics memory	N	N	Y
7	GdcTextureLoadInt16Fast	Loads 16bpp texture pattern to internal texture memory	N	N	Y
8	GdcTextureLoadExt16Fast	Loads 16bpp texture pattern to the graphics memory	N	N	Y
9	GdcTextureDimension	Sets texture information	Y	Y	Y
10	GdcBltTexture	Loads Blt texture to internal texture memory for MB86290A	Y	N	N
11	GdcGeoBltTexture	Loads Blt texture to internal texture memory for MB86291	N	Y	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.11 Binary Pattern Drawing Commands

Binary pattern drawing commands list is shown in the table 4.11.

Table 4.11 Binary pattern drawing commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcBitPatternDraw	Draws binary pattern	Y	Y	Y
2	GdcBitPatternMode	Sets enlarge/shrink mode	Y	Y	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.12 BLT Commands

Blt commands list is shown in the table 4.12.

Table 4.12 Blt commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291292	293
1	GdcBltCopy	Copies BitBlt pattern within current drawing frame	Y	Y	Y
2	GdcBltCopyAlt	Copies BitBlt pattern between any drawing frame (Async)	Y	Y	Y
3	GdcBltCopyAltSync	Copies BitBlt pattern between any drawing frame (Sync)	Y	Y	Y
4	GdcBltDraw	Draws BitBlt pattern	Y	Y	Y
5	GdcBltFill	Fills BitBlt field	Y	Y	Y
6	GdcBltColorTransparent	Sets transparent color of transparent BitBlt	N	Y	Y
7	GdcBltCopyAltAlpha	Copies BitBlt pattern with alpha blending	N	N	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.13 Video Capture Control Commands

Video capture control commands list is shown in the table 4.13.

Table 4.13 Video capture control commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcCapSetVideoCaptureMode	Sets mode of video capture	N	Y	Y
2	GdcCapGetErrorStatus	Gets error status of video capture	N	Y	Y
3	GdcCapClearErrorStatus	Clears error status of video capture	N	Y	Y
4	GdcCapSetVideoCaptureBuffer	Sets video capture buffer	N	Y	Y
5	GdcCapSetImageArea	Sets range of image	N	Y	Y
6	GdcCapSetWindowMode	Sets W layer mode	N	Y	Y
7	GdcCapSetVideoCaptureScale	Sets scale of video capture	N	Y	Y
8	GdcCapSetAttrMisc	Sets attribute of video capture	N	Y	Y
9	GdcCapSetInputDataCountNTSC	Sets number of video capture data for NTSC	N	Y	Y
10	GdcCapSetInputDataCountPAL	Sets number of video capture data for PAL	N	Y	Y
11	GdcCapSetLPFMode	Sets low pass filter mode	N	Y	Y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

Y : can be used

N : can not be used

4.14 I^{2}C Control Commands

I^{2}C control commands list is shown in the table 4.14.

Table 4.14 I^{2}C control commands list

No.	Command name	Function	Graphics Controller (*1)		
			290A	291/292	293
1	GdcI2CGetBusStatus	Gets I^{2}C bus status	N	y	y
2	GdcI2CSetBusControl	Controls I^{2}C bus	N	y	y
3	GdcI2CGetBusControlStatus	Gets I^{2}C bus control status	N	y	y
4	GdcI2CSetClock	Sets I^{2}C clock	N	y	y
5	GdcI2CSetData	Sets transfer data	N	y	y

(*1)290A:MB86290A, 291/292:MB86291/86291A/86291S/86292/86292S, 293:MB86293

y : These commands can be used only following Graphics Controllers.

MB86291S,MB86292S or MB86293 or later which supported video capture function.

N : can not be used

5 Data Format

This section describes the data types and data structures specified by “Graphics Driver”.

5.1 Data Type

Data types to define in the “Graphics Driver” is shown in the table 5.1.

Table 5.1 Data type list

Format	Description
GDC_FIXED32	32 bits signed fixed point (1 bits sign,15 bits integer and 16 bits fraction)
GDC_FIXED_SCALE	16 bits unsigned fixed point for Capture Scale (5 bits integer and 11 bits fraction)
GDC_SFLOAT	32 bits single precision float (IEEE754 compliant)
GDC_ULONG	32 bits unsigned integer
GDC_LONG	32 bits signed integer
GDC_USHORT	16 bits unsigned integer
GDC_SHORT	16 bits signed integer
GDC_UCHAR	8 bits signed integer
GDC_COLOR32	32 bits unsigned integer (32 bits color format)
GDC_COL32	32 bits unsigned integer (palette color format)
GDC_COL16	16 bits unsigned integer (16 bits color format)
GDC_COL8	8 bits unsigned integer (8 bits color format)
GDC_LPPCOL32	Pointer for GDC_COL32 format data
GDC_LPCOL24	Pointer for GDC_COLOR32 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 32 bits unsigned integer data (binary pattern data)
GDC_VERTEX	GDC_SFLOAT format vertex data structure
GDC_BOOL	True/false(value of GDC_TRUE / GDC_FALSE) [Note] It is not used as a type of the return value of the driver command

5.2 Data Structure

Data structures at to define in the “Graphics Driver” is shown in the following.

5.2.1 GDC_FIXED32 [32 bits fixed point]

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

31	30	16	15	0
S	I		F	

S: Sign (1 bit)

0:Positive number or zero

1:Negative number

I: Integer (15 bits)

F: Fraction (16 bits)

Figure 5.2.1 GDC_FIXED32 format

5.2.2 GDC_FIXED_SCALE [Capture scale]

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

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

I: Integer (5 bits)

F: Fraction (11 bits)

Figure 5.2.2 GDC_FIXED_SCALE format

5.2.3 GDC_COLOR32 [32 bits color]

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

It used by the GdcVertexColor32 command.

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td colspan="8">A</td><td colspan="8">R</td></tr>
</table>

<table>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="8">G</td><td colspan="8">B</td></tr>
</table>

A:Alpha bit (8 bits)

Sets blend ratio of vertex

R,G,B:

Color bit (8 bits)

Figure 5.2.3 GDC_COLOR32 format

5.2.4 GDC_COL32 [Palette color]

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

<table>
<tr><th>31</th><th>30</th><th>29</th><th>28</th><th>27</th><th>26</th><th>25</th><th>24</th><th>23</th><th>22</th><th>21</th><th>20</th><th>19</th><th>18</th><th>17</th><th>16</th></tr>
<tr><td>A</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td colspan="6">R</td><td>0</td><td>0</td></tr>
</table>

<table>
<tr><th>15</th><th>14</th><th>13</th><th>12</th><th>11</th><th>10</th><th>9</th><th>8</th><th>7</th><th>6</th><th>5</th><th>4</th><th>3</th><th>2</th><th>1</th><th>0</th></tr>
<tr><td colspan="6">G</td><td>0</td><td>0</td><td colspan="6">B</td><td>0</td><td>0</td></tr>
</table>

A:Alpha bit (1 bit)

When blend mode is available, sets mode of blend

0:Blending

1:Not blending

R,G,B:

Color bit (6 bits)

Figure 5.2.4 GDC_COL32 format

5.2.5 GDC_COL16 [16 bits color]

A color data described in 5 bits per R, B and G respectively. When this color data format is applied to texture, bit 15 is used as an alpha bit.

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

A:Alpha bit (1 bit)

When blend mode is available, sets mode of blend or stencil processing

0:Blending or stencil processing

1:Not blending or stencil processing

R,G,B:

Color bit (5 bits)

Figure 5.2.5 GDC_COL16 format

5.2.6 GDC_COL8 [8 bits color]

A index code to refer to color palette in 8 bits.

7	6	5	4	3	2	1	0
Index code							

Figure 5.2.6 GDC_COL8 format

5.2.7 GDC_VERTEX [GDC_SFLOAT format vertex data structure]

GDC_VERTEX structure is shown in the table 5.2.7.

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

It used by the GdcDrawPrimitive command.

Table 5.2.7 GDC_VERTEX structure

Type	Parameter	Description
GDC_SFLOAT	x	x coordinates of vertex (for device coordinates)
GDC_SFLOAT	y	y coordinates of vertex (for device coordinates)
GDC_SFLOAT	z	z coordinates of vertex (for device coordinates)
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 texture coordinates of vertex
GDC_SFLOAT	v	v texture coordinates of vertex
GDC_SFLOAT	rw	Reciprocal w texture coordinates of vertex
long	work	Reserved

6 Driver Command Reference

This section describes calling conventions interface and processing contents of driver commands.

6.1 Explanatory notes

Each item of a driver command reference is described as below.

Format	Prototype declaration of a command.

Parameter	Description of a parameter.

Return value	The return values and the description of them.

Error code	The error code and a name of an error when a command ends abnormally.

When the command ends normally, the error code is not set.

In this case the error code just before is held.

The error code is got by the GdcGetErrCode command.

This item is omitted if the command has no return values.

In some commands errors don't occur even if the commands have the return values.

Description	Description of the command.

6.2 System Control Commands

6.2.1 GdcInitialize [“Graphics driver” initialization]

Format int GdcInitialize(int gdc_type)

Parameter gdc_type A type of the “Graphics Controller”

GDC_TYPE_MB86290A	In case of using MB86290A
GDC_TYPE_MB86291	In case of using MB86291
GDC_TYPE_MB86291A	In case of using MB86291A
GDC_TYPE_MB86292	In case of using MB86292
GDC_TYPE_MB86293	In case of using MB86293

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code - GDC_ERR_DL_BUF_ALLOC (Failure of Display List buffer acquisition)

- GDC_ERR_DL_SIZE (Incorrect buffer size)

- GDC_ERR_DL_NUM (Not appropriate block count)

Description Initialize the “Graphics Driver”.

This command precedes all driver commands, and call only once.

If this command is called, the following system dependent commands will be called inside.

- GdcSetDisplayListBuffer command
- GdcGetHostRegisterAddress command
- GdcGetDispRegisterAddress command
- GdcGetDrawRegisterAddress command

For details about initialization processing of the “Graphics Driver”, refer to "initialize.c" file in sample directory.

This command can be used by all “Graphics Controller”.

6.2.2 GdcQueryVersion [Version number check]

Format	void GdcQueryVersion (int *version, int *level)	
Parameter	version	Pointer to store version number
	level	Pointer to store level number
Return value	None	

Description

Indicates current version and level number of the “Graphics Driver”.

Version number and level number are numerical value.

For example, when the version number is 1, and the level number is 9, the following numbers are stored in each parameter.

*version = 1

*level = 9

This command can be used by all “Graphics Controller”.

6.2.3 GdcSetInternalClock [Changes internal clock frequency]

Format void GdcSetInternalClock(GDC_ULONG geo_clock, GDC_ULONG other_clock)

Parameter geo_clock Clock frequency for geometry engine

GDC_CLOCK_166MHZ 166MHz
GDC_CLOCK_133MHZ 133MHz
GDC_CLOCK_100MHZ 100MHz

other_clock Clock frequency for besides geometry engine

GDC_CLOCK_133MHZ 133MHz
GDC_CLOCK_100MHZ 100MHz

Return value None

Description Sets internal clock frequency. The combination of parameters are shown in the table 6.2.3.

Table 6.2.3 Internal clock frequency combinations

		geo_clock		
		166MHz	133MHz	100MHz
other_clock	133MHz	OK	OK	NG
	100MHz	OK	OK	OK

OK:Possible

NG:Impossible (not supported)

Be sure to execute this command just after initialization by the GdcInitialize command of the “Graphics Driver”.

In order to steady status of the “Graphics Controller”, 200 microsecond intervals are necessary after setting internal clock frequency before executing the following procedures.

This command is for MB86293 or later.

6.2.4 GdcInitDevice [Initializes the “Graphics Controller”]

Format	int GdcInitDevice (void)	
Parameter	None	
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Description

Initializes each register of the “Graphics Controller”.

Be sure to execute this command once before calling other driver commands.

However, be sure to execute the following commands before calling the GdcInitDevice command.

- GdcInitialize command
- GdcSetInternalClock command
- GdcSoftwareReset command
- GdcSetRegisterLocation command
- GdcSetMemoryMode command
- GdcSetBurstMode command

This command can be used by all “Graphics Controller”.

6.2.5 GdcGeoInitialize [Initialize geometry engine]

Format int GdcGeoInitialize(void)

Parameter None

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)

Description Initializes internal resources in the geometric engine.

When using drawing commands for object coordinates, be sure to call this command after initializing the "Graphics Driver".

However, be sure to execute the following commands before calling the GdcGeoInitialize command.

- GdcInitialize command
- GdcSetInternalClock command
- GdcSoftwareReset command
- GdcSetRegisterLocation command
- GdcSetMemoryMode command
- GdcSetBurstMode command
- GdcInitDevice command

This command is for MB86291 or later.

6.2.6 GdcFlush [Drawing by Display List (Async)]

Format GDC_ULONG GdcFlush(void)

Parameter None

Return value The transmitted number of byte

Description Transfers a Display List in the Display List buffer to the "Graphics Controller". If DMA is applied, this command is completed without waiting for the end of the list transfer. If CPU writes the Display List to the "Graphics Controller", this command is completed after the end of the list transfer.

The transmitted number of byte is set to return value.

This command can be used by all "Graphics Controller".

6.2.7 GdcVFlush [Vertical blanking interval palling (Async)]

Format	GDC_ULONG GdcVFlush(void)
Parameter	None
Return value	The transmitted number of byte

Description

Transfers a Display List after attaching a command for waiting Sync command (*1) to the end of it.

When the Sync command is executed, the "Graphics Controller" 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 drawing frame (the 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.

The transmitted number of byte is set to return value.

This command can be used by all "Graphics Controller".

(*1) Sync command: It waits for drawing processing of the "Graphics Controller" until a vertical blanking interval period comes.

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.

6.2.8 GdcVerticalSync [Adds vertical blanking interval command]

Format	int GdcVerticalSync(void)	
Parameter	None	
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)	

Description

Attaches a command for waiting VSYNC (Sync command) to the end of a Display List.

Transmission of a Display List is not performed.

This command can be used by all "Graphics Controller".

6.2.9 GdcInterrupt [Interrupt request to host CPU]

Format int GdcInterrupt(void)

Parameter None

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)

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 Controller".

6.2.10 GdcExecMode [Sets execution mode]

Format void GdcExecMode (GDC_UCHAR sync)

Parameter sync Sync/Async mode selection

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 Controller".

6.2.11 GdcSetDMAMode [Sets DMA mode]

Format int GdcSetDMAMode(int tran_unit, int dma_request, int address_mode,
int ack_mode)

Parameter	tran_unit	Unit of DMA transfer	
		GDC_DMA_TRANUNIT_4	4 byte
		GDC_DMA_TRANUNIT_32	32 byte
	dma_request	DMA request	
		GDC_DMA_REQUEST_NEGATE During transferring, when the “Graphics Controller” cannot receive data, DMA request is invalid (negate), and if it will be in the state where data is receivable, it will be valid (assert)	
		GDC_DMA_REQUEST_NO_NEGATE Not negate while DMA is transferring Display List	
	address_mode	Address mode of external DMA request	
		GDC_DMA_ADDRMODE_DUAL	Dual address mode
		GDC_DMA_ADDRMODE_SINGLE	Single address mode
	ack_mode	ACK mode	
		GDC_DMA_ACKMODE	Uses ACK (detect DMA request at a low level signal)
		GDC_DMA_NO_ACKMODE	Not use ACK (detect DMA request at an edge)
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	

Error code GDC_ERR_INVALID_PARAMETER(Invalid parameter is specified)

Description Sets DMA transfer mode to DSU (DMA Set Up) register.

This command can be used by all “Graphics Controller”.

However, “ack_mode” is available only for the MB86293 or later.

In case of MB86290A/291/292, sets GDC_DMA_ACKMODE (uses ACK) to “ack_mode”.

6.2.12 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 31	- - -	2	1	0
0	- - -	x	x	x

bit 0:	Valid data exists in Display List FIFO	=0
	Valid data not exists in Display List FIFO	=1
bit 1:	Display List FIFO is full	=0
	Display List FIFO is not full	=1
bit 2:	More than half entries of Display List FIFO are empty	=0
	More than half entries of Display List FIFO are not empty	=1

All other bits:0

Figure 6.2.12 Display List FIFO status

Description Reads IFSR (Input FIFO Status Register) and returns current Display List FIFO status.

This command can be used by all “Graphics Controller”.

6.2.13 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 Controller”.

6.2.14 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 31	- - -	2	1	0
0	- - -	x	x	x

bit 0:	Command error (type code is not normal)	No=0,	Yes=1
bit 1:	Packet error (command code is not normal)	No=0,	Yes=1
bit 2:	FIFO overflow	No=0,	Yes=1

All other bits:0

Figure 6.2.14 Display List FIFO error status

Description Reads EST (Error Status Register) and returns Display List FIFO error status.

This command can be used by all “Graphics Controller”.

6.2.15 GdcGetInterruptStatus [Gets interrupt status]

Format GDC_UCHAR GdcGetInterruptStatus (void)

Parameter None

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

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

bit 0:	Command execution error interrupt	Yes=1,	No=0
bit 1:	Command complete interrupt	Yes=1,	No=0
bit 2:	VSYNC interrupt	Yes=1,	No=0
bit 3:	Frame sync interrupt	Yes=1,	No=0
bit 4:	External sync error interrupt	Yes=1,	No=0

All other bits:0

Figure 6.2.15 Interrupt status

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

This command is only for MB86290A.

When the “Graphics Controller” is MB86291 or later, GdcGeoGetInterruptStatus must be used.

6.2.16 GdcGeoGetInterruptStatus [Gets interrupt status for MB86291 or later]

Format GDC_ULONG GdcGeoGetInterruptStatus(void)

Parameter None

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

bit 31	- - -	4	3	2	1	0
0	- - -	x	x	x	x	x

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

All other bits:1

Figure 6.2.16 Interrupt status

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

This command is for MB86291 or later.

When the "Graphics Controller" is MB86290A, GdcGetInterruptStatus must be used.

6.2.17 GdcClearInterruptStatus [Clears interrupt status for MB86290A]

Format void GdcClearInterruptStatus (GDC_UCHAR clear)

Parameter clear Clear pattern (shown below)

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

bit 0:	Command execution error interrupt	Clear=0,	Hold=1
bit 1:	Command complete interrupt	Clear=0,	Hold=1
bit 2:	VSYNC interrupt	Clear=0,	Hold=1
bit 3:	Frame sync interrupt	Clear=0,	Hold=1
bit 4:	External sync error interrupt	Clear=0,	Hold=1

All other bits:1

Figure 6.2.17 Clear pattern format

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 the “Graphics Controller” is MB86291 or later, the GdcGeoClearInterruptStatus command must be used.

6.2.18 GdcGeoClearInterruptStatus [Clears interrupt status for MB86291 or later]

Format void GdcGeoClearInterruptStatus(GDC_ULONG clear)

Parameter clear Clear pattern (shown below)

Return value

bit 31	- - -	4	3	2	1	0
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

Figure 6.2.18 Clear pattern format

Return value None

Description Clears a request of interrupt, indicated by IST (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 for MB86291 or later.

When the “Graphics Controller” is MB86290A, the GdcClearInterruptStatus command must be used.

6.2.19 GdcSetInterruptMask [Sets interrupt mask for MB86290A]

Format void GdcSetInterruptMask (GDC_UCHAR mask)

Parameter mask Mask pattern (shown below)

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

bit 0:	Command error interrupt	Mask=1,	Enable=0
bit 1:	Command complete interrupt	Mask=1,	Enable=0
bit 2:	VSYNC interrupt	Mask=1,	Enable=0
bit 3:	Frame sync interrupt	Mask=1,	Enable=0
bit 4:	External sync error interrupt	Mask=1,	Enable=0

All other bits:0

Figure 6.2.19 Mask pattern format

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 the “Graphics Controller” is MB86291 or later, the GdcGeoSetInterruptMask command must be used.

6.2.20 GdcGeoSetInterruptMask [Sets interrupt mask for MB86291 or later]

Format void GdcGeoSetInterruptMask(GDC_ULONG mask)

Parameter mask Mask pattern (shown below)

bit 31	- - -	4	3	2	1	0
0	- - -	x	x	x	x	x

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

All other bits:0

Figure 6.2.20 Mask pattern format

Return value None

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

This command is for MB86291 or later.

When the “Graphics Controller” is MB86290A, the GdcSetInterruptMask command must be used.

6.2.21 GdcGeoGetFIFOStatus [Gets geometry Display List FIFO status]

Format GDC_ULONG GdcGeoGetFIFOStatus(void)

Parameter None

Return value Geometry Display List FIFO status in the following format:

bit 31	- - -	2	1	0
0	- - -	x	x	x

bit 0:	Valid data exists in geometry Display List FIFO	=0
	Valid data not exists in geometry Display List FIFO	=1
bit 1:	Geometry Display List FIFO is full	=0
	Geometry Display List FIFO is not full	=1
bit 2:	More than half entries of geometry Display List FIFO are empty	=0
	More than half entries of geometry Display List FIFO are not empty	=1

All other bits:0

Figure 6.2.21 Display List FIFO status

Description Returns current geometry Display List FIFO status.

This command is for MB86291 or later.

6.2.22 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.

This command is for MB86291 or later.

6.2.23 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 Controller”.

6.2.24 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.

In addition, cancels Display List stored in Display List buffer.

In order to steady status of the “Graphics Controller”, 32 bus clock cycles intervals are necessary after software reset before executing the following procedures.

This command can be used by all “Graphics Controller”.

6.2.25 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 Controller”.

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
GDC_ERR_DL_NUM	applicable size
	Not appropriate block count
GDC_ERR_DATA_TOO_BIG	Block count is other than 1, 2
GDC_ERR_INVALID_LAYER	Too large data
GDC_ERR_INVALID_BANK	Invalid layer is specified
GDC_ERR_INVALID_COLOR_MODE	Invalid bank is specified
GDC_ERR_INVALID_CURSOR_NUMBER	Invalid color mode is specified
GDC_ERR_ILLEGAL_DIMENSION	Invalid cursor number is specified
GDC_ERR_INVALID_ATTRIBUTE	Illegal vertical/horizontal size of pattern data
GDC_ERR_INVALID_PRIMITIVE	Invalid attribute is specified
GDC_ERR_CREMSON_OPEN_FAILED	Invalid primitive is specified
GDC_ERR_ILLEGAL_VERTEX_COUNT	Fail to initialize “Graphics Controller”
GDC_ERR_ILLEGAL_LINE_WIDTH	Illegal number of vertex
GDC_ERR_NOT_READY	Illegal width of line
GDC_ERR_INVALID_PARAMETER	“Graphics Driver” is not initialized
	Invalid parameter is specified

6.2.26 GdcSetRegisterLocation [Changes address of registers location]

Format void GdcSetRegisterLocation (GDC_ULONG locate)

Parameter locate Register location

GDC_REG_LOCATE_CENTER	Center location
GDC_REG_LOCATE_BOTTOM	The last location

Return value None

Description In SH-3/SH-4 mode sets this parameter when changing the register location from center (0x1fc0000) to the last (0x3fc0000).

Execute this command after having called the following commands.

Be sure to execute the following commands before calling the GdcSetRegisterLocation command.

- GdcInitialize command
- GdcSetInternalClock command
- GdcSoftwareReset command

In order to steady status of the “Graphics Controller”, 20 bus clock cycles intervals are necessary after changes address of register location before executing the following procedures.

This command is for MB86293 or later.

6.2.27 GdcSetBurstMode [Sets burst transfer mode of drawing]

Format void GdcSetBurstMode(GDC_ULONG mode)

Parameter mode Burst transfer mode

GDC_ENABLE	Enable
GDC_DISABLE	Disable

Return value None

Description Sets burst transfer mode of drawing.

If the bus occupation rate of drawing becomes high, all the 6 layers may not be displayed.

In this case, sets burst transfer mode to disable.

Be sure to execute the GdcSetMemoryMode commands before calling this command.

This command is for MB86293 or later.

6.2.28 GdcQueryChipID [Queries about chip ID]

Format void GdcQueryChipID (int *chip_no, int *version)

Parameter chip_no Pointer to the area which stores chip number

version Pointer to the area which stores chip version number

Return value None

Description Indicates chip number and chip version number.

Chip number and chip version number are numerical value.

For details about each number, refer to the hardware specification of the “Graphics Controller” of use.

This command is for MB86293 or later.

6.2.29 GdcCancelDisplayList [Cancels Display List]

Format void GdcCancelDisplayList (void)

Parameter None

Return value None

Description Cancels the Display List stored in the Display List buffer.

This command can be used by all “Graphics Controller”.

6.2.30 GdcGetPixelEngineStatus [Gets pixel engine status]

Format GDC_ULONG GdcGetPixelEngineStatus (void)

Parameter None

Return value Pixel engine status in the following format:

bit 31	- - -	2	1	0
	- - -		x	x

bit 1-0: Rendering is complete =00

Rendering is executing =01

All other bits: unsettled

Figure 6.2.30 Pixel engine status

Description Returns pixel engine status.

This command is only for MB86290A.

When the “Graphics Controller” is MB86291 or later, the GdcGeoGetPixelEngineStatus command must be used.

6.2.31 GdcGeoGetPixelEngineStatus [Gets geometry pixel engine status]

Format GDC_ULONG GdcGeoGetPixelEngineStatus (void)

Parameter None

Return value Geometry pixel engine status in the following format:

bit 31	- - -	2	1	0
	- - -		x	x

bit 1-0: Rendering is complete =00

Rendering is executing =01

All other bits: unsettled

Figure 6.2.31 Geometry pixel engine status

Description Returns pixel engine status.

This command is only for MB86291 or later.

When the “Graphics Controller” is MB8629A, the GdcGetPixelEngineStatus command must be used.

6.2.32 GdcGetLocalDisplayListTransferStatus [Gets Local Display List transfer status]

Format GDC_ULONG GdcGetLocalDisplayListTransferStatus (void)

Parameter None

Return value Transfer of Local Display List status in the following format:

bit 31	- - -	2	1	0
0	- - -	0	0	x

bit 0: Transfer is complete =0

Transfer is executing =1

All other bits:0

Figure 6.2.32 Local Display List transfer status

Description Returns transfer of Local Display List status.

This command can be used by all “Graphics Controller”.

6.3 Display Commands

6.3.1 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 "Graphics Controller". For details of the DCM register description, refer to the "Graphics Controller" hardware specification of use.

Return value None

Description Controls 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 Controller".

6.3.2 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 Controller".

6.3.3 GdcDispTimingWindow [Sets W-layer display position]

Format	void GdcDispTimingWindow (GDC_USHORT x, GDC_USHORT y, GDC_USHORT w, GDC_USHORT h)	
Parameter	x	x coordinate in the device coordinates
	y	y coordinate in the device coordinates
	w	Window width (pixel unit)
	h	Window height (pixel unit)
Return value	None	

Description

Sets display position of the W(Window) layer.

x and y should specify the display position at the upper left of a window frame.

This command can be used by all “Graphics Controller”.

6.3.4 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 Controller”.

6.3.5 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

[When the "Graphics Controller" is MB86293 or later, following functions are available]

GDC_DISP_LAYER_L0	L0 layer
GDC_DISP_LAYER_L1	L1 layer
GDC_DISP_LAYER_L2	L2 layer
GDC_DISP_LAYER_L3	L3 layer
GDC_DISP_LAYER_L4	L4 layer
GDC_DISP_LAYER_L5	L5 layer

enable Layer display enable/disable

GDC_ENABLE	Layer display enable
GDC_DISABLE	Layer display disable

cmode Color mode selection

GDC_24BPP_FORMAT	24 bits color mode (*1)
GDC_16BPP_FORMAT	16 bits color mode
GDC_8BPP_FORMAT	8 bits 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 of bank 0

loa1 Base address of logical frame of bank 1

lw Logical frame width (pixel unit)

lh Logical frame height (pixel unit)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code GDC_ERR_INVALID_LAYER (Invalid layer is specified)

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.

When layer specification is W layer, each variable of cmode, fmode, and loa1 and lh is not used.

When the following macros are set to display or not to display layers, both ML and MR, both BL and BR layers are displayed simultaneously.

- GDC_DISP_LAYER_ML
- GDC_DISP_LAYER_MR
- GDC_DISP_LAYER_BL
- GDC_DISP_LAYER_BR

When the following macros are set to display or not to display layers, layers can be displayed or not displayed individually.

- GDC_DISP_LAYER_L2
- GDC_DISP_LAYER_L3
- GDC_DISP_LAYER_L4
- GDC_DISP_LAYER_L5

When the L5 layer is used as a blend coefficient layer, this layer must be displayed in 8 bits color mode.

In case of extend display mode, L4 and L5 layer are not available in 8 bits color mode.

When the L5 layer is used as a blend coefficient layer, this layer must be displayed in 8 bits color mode (except L5 layer is used as a blend coefficient layer).

This command can be used by all “Graphics Controller”.

(*1)The 24 bits color mode function is for MB86293 or later, and this mode can be used with the “Graphics Controller” supporting the 24 bits color mode option.

6.3.6 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 Controller”.

6.3.7 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 Controller”.

6.3.8 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	B layer

[When the “Graphics Controller” is MB86293 or later, following functions are available]

GDC_DISP_LAYER_L0	L0 layer
GDC_DISP_LAYER_L1	L1 layer
GDC_DISP_LAYER_L2	L2 layer
GDC_DISP_LAYER_L3	L3 layer
GDC_DISP_LAYER_L4	L4 layer
GDC_DISP_LAYER_L5	L5 layer

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code GDC_ERR_INVALID_LAYER (Invalid layer is specified)

Description Displays the layer specified by the layer parameter.

When the following macros are set to display layers, both ML and MR, both BL and BR layers are displayed simultaneously.

- GDC_DISP_LAYER_M
- GDC_DISP_LAYER_B

When the L5 layer is used as a blend coefficient layer, this layer must be displayed.

This command can be used by all “Graphics Controller”.

6.3.9 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	B layer

[When the “Graphics Controller” is MB86293 or later, following functions are available]

GDC_DISP_LAYER_L0	L0 layer
GDC_DISP_LAYER_L1	L1 layer
GDC_DISP_LAYER_L2	L2 layer
GDC_DISP_LAYER_L3	L3 layer
GDC_DISP_LAYER_L4	L4 layer
GDC_DISP_LAYER_L5	L5 layer

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code GDC_ERR_INVALID_LAYER (Invalid layer is specified)

Description Stops the display of specified layer.

When the following macros are set to not display layers, both ML and MR, both BL and BR layers are not displayed simultaneously.

- GDC_DISP_LAYER_M
- GDC_DISP_LAYER_B

This command can be used by all “Graphics Controller”.

6.3.10 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_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

Error code - GDC_ERR_INVALID_LAYER (Invalid layer is specified)

- GDC_ERR_INVALID_BANK (Invalid bank is specified)

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

This command can be used by all “Graphics Controller”.

6.3.11 GdcDispDoFlip [Flips display bank]

Format int GdcDispDoFlip (GDC_UCHAR layer, GDC_UCHAR bank)

Parameter layerq 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

Error code - GDC_ERR_INVALID_LAYER (Invalid layer is specified)

- GDC_ERR_INVALID_BANK (Invalid bank is specified)

Description Switching displays (flipping).

This command can be used by all “Graphics Controller”.

6.3.12 GdcOverlayPriorityMode [Sets overlay display mode]

Format int GdcOverlayPriorityMode (GDC_UCHAR mode)

Parameter mode C layer overlay mode

GDC_OVERLAY_C_PRIORITY	Simple priority mode (default)
GDC_OVERLAY_C_BLEND	Blend mode

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code None

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 Controller”.

6.3.13 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 W/M/B layer color

blend Blending ratio (only upper 4 bits are valid)

Effective values are 0x00-0xf0

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Error code None

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 Controller”.

6.3.14 GdcDispDisplayMode [Sets display mode]

Format int GdcDispDisplayMode (GDC_UCHAR mode)

Parameter mode Display mode

GDC_STANDARD_MODE	Standard display mode (default)
GDC_OVERLAY_EXT_MODE	Extend overlay mode
GDC_WINDOW_MODE	Window mode
GDC_EXTEND_MODE	Extend display mode

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code None

Description Sets display mode of all layers.

This command is for MB86293 or later.

Difference among these modes are shown in table 6.3.14.

Table 6.3.14 Function list

Function	Extend display mode (Window mode + Extend overlay mode)	Window mode	Extend overlay mode	Standard display mode
Overlay	6 layer	6 layer	4 layer + right and left division	4 layer + right and left division
Displaying window	6 layer	6 layer	1 layer	1 layer
Displaying background color	OK	OK	OK	NG
Changes layer order	OK	NG	OK	NG
Palette number	4	2	4	2

Displayed images of extend display mode and standard display mode are shown in the figure 6.3.14

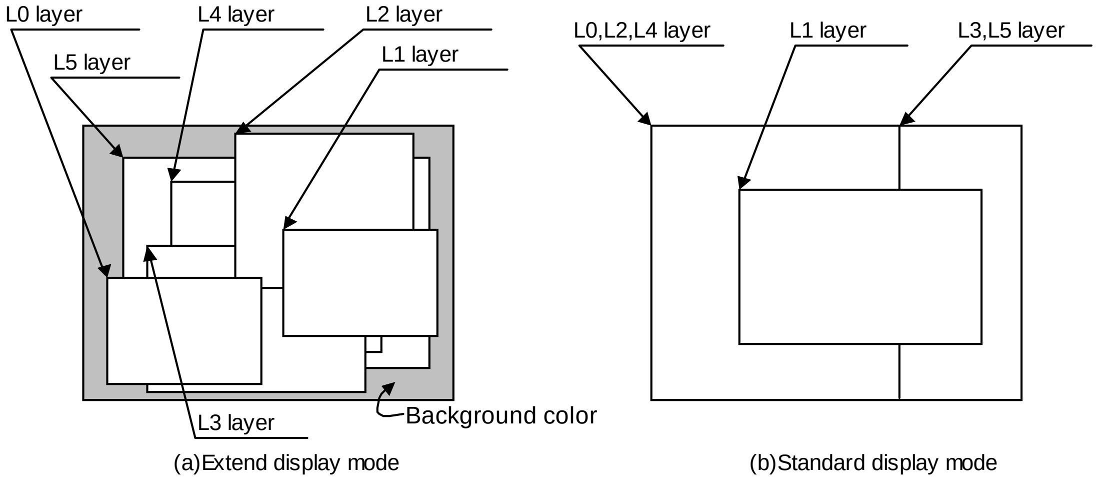


Figure 6.3.14 Displayed images of extend display mode and standard display mode

6.3.15 GdcDispDisplayLayerMode [Sets layer display mode]

Format int GdcDispDisplayLayerMode (GDC_UCHAR layer, GDC_ULONG mode)

Parameter layer Layer selection

GDC_DISP_LAYER_L0	L0 layer
GDC_DISP_LAYER_L1	L1 layer
GDC_DISP_LAYER_L2	L2 layer
GDC_DISP_LAYER_L3	L3 layer
GDC_DISP_LAYER_L4	L4 layer
GDC_DISP_LAYER_L5	L5 layer

mode Layer display mode

GDC_STANDARD_MODE	Standard display mode (default)
GDC_OVERLAY_EXT_MODE	Extend overlay mode
GDC_WINDOW_MODE	Window mode
GDC_EXTEND_MODE	Extend display mode

Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_INVALID_LAYER (Invalid layer is specified)

Description Changes display mode by each layer.

In extend display mode setting, window mode and an extend overlay mode can be set at the same time. GDC_EXTEND_MODE has the following meanings.

GDC_EXTEND_MODE = GDC_WINDOW_MODE | GDC_OVERLAY_EXT_MODE

When L5 layer is used as a blend coefficient layer, this layer must be set extend display mode.

This command is for MB86293 or later.

6.3.16 GdcDispSetBackColor [Sets background color]

Format int GdcDispSetBackColor (GDC_COLOR32 color)

Parameter	color	24 bits background color
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code None

Description Sets background color.

This command is for MB86293 or later.

6.3.17 GdcDispSetLayerWindow [Sets position and size of the window mode layer]

Format int GdcDispSetLayerWindow (GDC_UCHAR layer,
GDC_USHORT x, GDC_USHORT y,
GDC_USHORT w, GDC_USHORT h)

Parameter layer Layer selection

GDC_DISP_LAYER_L0	L0 layer
GDC_DISP_LAYER_L1	L1 layer
GDC_DISP_LAYER_L2	L2 layer
GDC_DISP_LAYER_L3	L3 layer
GDC_DISP_LAYER_L4	L4 layer
GDC_DISP_LAYER_L5	L5 layer

	x	x coordinates in the device coordinates
	y	y coordinates in the device coordinates
	w	Window width (pixel unit)
	h	Window height (pixel unit)
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_INVALID_LAYER (Invalid layer is specified)

Description Sets x and y position and width and height of window mode layer.

Setting to L1 layer is also available by the GdcDispTimingWindow command.

This command is for MB86293 or later.

6.3.18 GdcLayerOverlayPriorityMode [Sets overlay display mode in every layer]

Format int GdcLayerOverlayPriorityMode (GDC_UCHAR layer, GDC_UCHAR mode)

Parameter layer Layer select

GDC_DISP_LAYER_L0	L0 layer
GDC_DISP_LAYER_L1	L1 layer
GDC_DISP_LAYER_L2	L2 layer
GDC_DISP_LAYER_L3	L3 layer
GDC_DISP_LAYER_L4	L4 layer
GDC_DISP_LAYER_L5	L5 layer

mode Overlay mode

GDC_OVERLAY_PRIORITY	Overlay with transparent color (default)
GDC_OVERLAY_BLEND	Overlay with blend

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code GDC_ERR_INVALID_LAYER (Invalid layer is specified)

Description Sets overlay display mode by each layer.

Setting of C layer by the GdcOverlayPriorityMode command reflects on L0 layer.

On the contrary, setting to L0 layer by this command refracts on C layer.

When L5 layer is used as a blend coefficient layer, this layer is not target of overlaying by transparent color or blend.

This command is for MB86293 or later.

6.3.19 GdcLayerOverlayBlend [Sets blend mode in every layer]

Format int GdcLayerOverlayBlend (GDC_UCHAR layer, GDC_UCHAR select,
GDC_UCHAR correct, GDC_UCHAR source, GDC_UCHAR blend)

Parameter layer Layer selection

GDC_DISP_LAYER_L0	L0 layer
GDC_DISP_LAYER_L1	L1 layer
GDC_DISP_LAYER_L2	L2 layer
GDC_DISP_LAYER_L3	L3 layer
GDC_DISP_LAYER_L4	L4 layer
GDC_DISP_LAYER_L5	L5 layer

select Selects of blend calculating method

GDC_BLEND_CURRENT_RATIO
layer color * blend ratio + lower layer color
*(1 - blend ratio)

GDC_BLEND_ONE_MINUS_CURRENT_RATIO
layer color * (1 - blend ratio)
+ lower layer color * blend ratio

correct Correction by 1/256 value

GDC_BLEND_NO_CORRECT
Uses the blend ratio

GDC_BLEND_CORRECT
When the blend ratio is not 0, add 1/256
(When using the blend ratio of 100%)

source Selects source data of the blend ratio

GDC_BLEND_RATIO_CONSTANT
Uses the fixed value as blend ratio specified by “blend”

GDC_BLEND_RATIO_L5
Uses pixel value of L5 layer for the blend ratio

blend Blend ratio coefficient

(used when ”source” is GDC_BLEND_RATIO_CONSTANT)

Effective values are 0-255

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code GDC_ERR_INVALID_LAYER (Invalid layer is specified)

Description Sets the blend mode when overlay priority mode is “overlay with blend (GDC_OVERLAY_BLEND)”.

Setting by the GdcOverlayBlend command reflect on L0 layer.

“correct” and “source” are set to GDC_BLEND_NO_CORRECT and GDC_BLEND_RATIO_CONSTANT respectively when the mode is set by the GdcOverlayBlend command.

Setting of “source” is not valid when L5 layer is set.

When the L5 layer is used as a blend coefficient layer, set L5 layer as extend display mode by the GdcDispDisplayLayerMode command.

When the L1 layer is capture mode and L5 layer is used as blend coefficient layer, L1 layer is not blended correctly.

This command is for MB86293 or later.

6.3.20 GdcDispLayerOrder [Sets layer display order]

Format int GdcDispLayerOrder(GDC_UCHAR layer0, GDC_UCHAR layer1,

GDC_UCHAR layer2, GDC_UCHAR layer3,

GDC_UCHAR layer4, GDC_UCHAR layer5)

Parameter layer0 Selects top level layer

GDC_DISP_LAYER_L0	L0 layer
GDC_DISP_LAYER_L1	L1 layer
GDC_DISP_LAYER_L2	L2 layer
GDC_DISP_LAYER_L3	L3 layer
GDC_DISP_LAYER_L4	L4 layer
GDC_DISP_LAYER_L5	L5 layer
GDC_NO_LAYER	Not select layer

layer1 Selects the second level layer (as for the value, the same in case of layer0)

layer2 Selects the third level layer (as for the value, the same in case of layer0)

layer3 Selects the fourth level layer (as for the value, the same in case of layer0)

layer4 Selects the fifth level layer (as for the value, the same in case of layer0)

layer5 Selects the sixth level layer (as for the value, the same in case of layer0)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code GDC_ERR_INVALID_LAYER (Invalid layer is specified)

Description Sets display order of layers. Sets different layer in each level.

The layer which was not selected is not displayed.

When the L5 layer is used as a blend coefficient layer, this layer should not be selected.

This command is for MB86293 or later.

6.4 Color Control Commands

6.4.1 GdcColorPalette [Sets palette colors]

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

Parameter layer C layer, MB layer 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

Error code GDC_ERR_INVALID_LAYER (Invalid layer is specified)

Description Sets color index code to palette table. If size is set to “0”, all 256 entries of selected palette are set.

Setting of palette is available without regard to display mode.

However, L2,L3 layer palettes are available only in extend display mode(only in MB86293 or later).

This command can be used by all “Graphics Controller”.

6.4.2 GdcColorTransparent [Sets transparent color]

Format	int GdcColorTransparent (GDC_UCHAR layer, GDC_COLOR32 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	
Error code	GDC_ERR_INVALID_LAYER (Invalid layer is specified)		

Description

Sets transparent color code. In indirect color mode (in which color palette is applied), the lower 8 bits is used.

L0, L2, L3 layers correspond to C, ML and MR layers.

The following values are used by color mode of layer.

- 8 bits color mode	: lower 8 bits of “color”
- 16 bits color mode	: lower 16 bits of “color”
- 24 bits color mode	: lower 24 bits of “color”

This command can be used by all “Graphics Controller”.

6.4.3 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

Error code GDC_ERR_INVALID_COLOR_MODE (Invalid color mode is specified)

Description Select handling of color value 0 (palette number 0) from the following.

- Not treaded as transparent color, and treated as normal color value (palette number)
- Treated as transparent color

This command can be used by all “Graphics Controller”.

6.4.4 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

Error code None

Description Sets whether Chroma-key is operated. When Chroma-key is operated, select target to be compared between C layer color and display color. When Chroma-key is not operated, setting of target of key color to be compared has no meaning.

This command can be used by all “Graphics Controller”.

6.4.5 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
Error code	None	
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 Controller”.	

6.4.6 GdcColorPaletteOffset [Sets of the color palette offset]

Format	int GdcColorPalette (GDC_UCHAR layer, GDC_UCHAR number, GDC_UCHAR size, GDC_LPPCOL32 lpColor)	
Parameter	layer	Palette selection

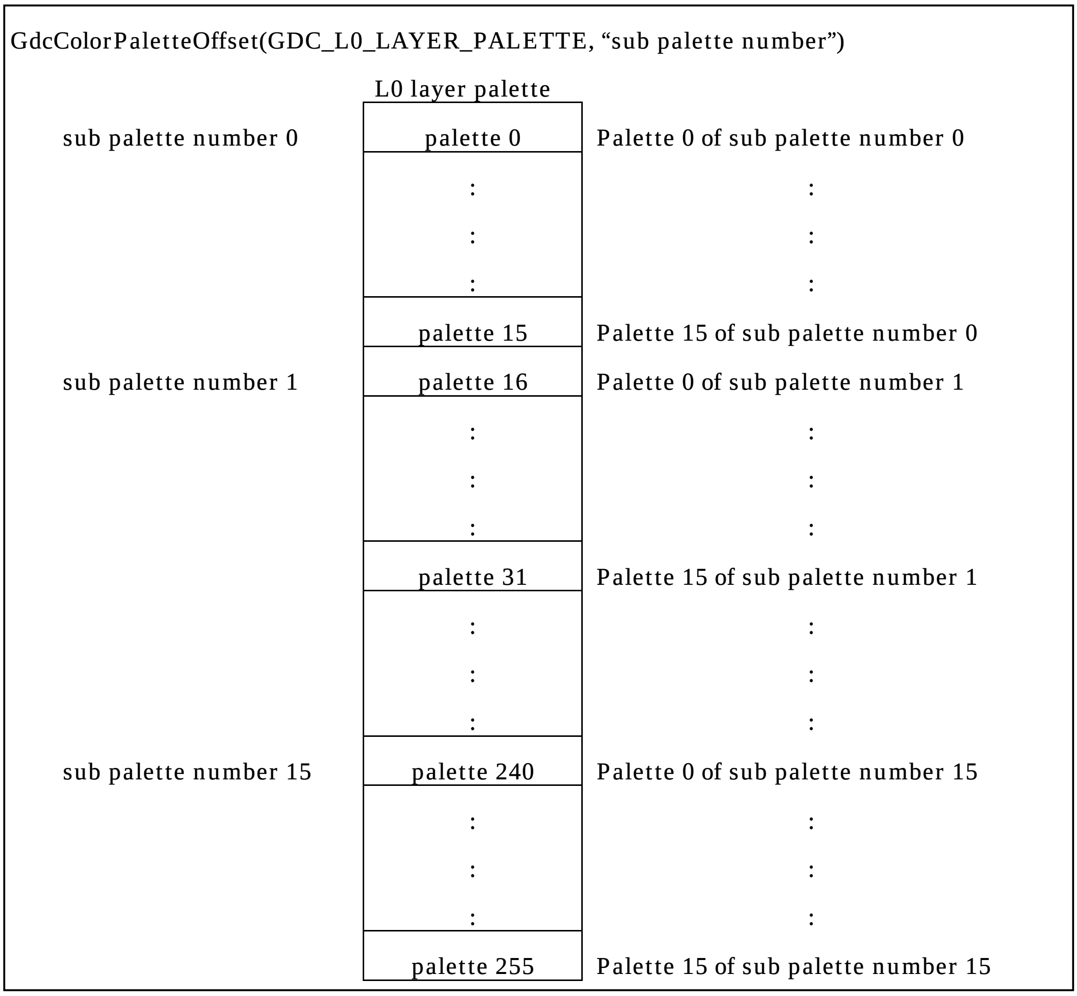
GDC_C_LAYER_PALETTE	Selects C layer palette
GDC_MB_LAYER_PALETTE	Selects MB layer palette
GDC_L0_LAYER_PALETTE	Selects L0 layer palette (same as C layer palette)
GDC_L1_LAYER_PALETTE	Selects L1 layer palette (same as MB layer palette)
GDC_L2_LAYER_PALETTE	Selects L2 layer palette
GDC_L3_LAYER_PALETTE	Selects L3 layer palette

	sub_no	Sub palette number (Serial number of 16 divided parts of 256 palette) The range of value is 0-15, and more than 16 are not valid
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_INVALID_LAYER (Invalid layer is specified)	
Description	Sets the sub palette number of the color palette. Sub palette number is the serial number of 16 divided parts of 256 palette. This command is for MB86293 or later.	

[Example of usage of the GdcColorPaletteOffset command]

Example for changing color set number to each sub palette numbers is shown in the figure 6.4.6.

In this example, 16 sub palettes have been already assigned to L0 layer palette.



GdcColorPaletteOffset(GDC_L0_LAYER_PALETTE, "sub palette number")

	L0 layer palette	
sub palette number 0	palette 0	Palette 0 of sub palette number 0
	:	:
	:	:
	:	:
	palette 15	Palette 15 of sub palette number 0
sub palette number 1	palette 16	Palette 0 of sub palette number 1
	:	:
	:	:
	:	:
	palette 31	Palette 15 of sub palette number 1
	:	:
	:	:
	:	:
sub palette number 15	palette 240	Palette 0 of sub palette number 15
	:	:
	:	:
	:	:
	palette 255	Palette 15 of sub palette number 15

Figure 6.4.6 Example of usage of the GdcColorPaletteOffset command

6.5 Cursor Control Commands

6.5.1 GdcCursorAddress [Sets cursor pattern memory address]

Format int GdcCursorAddress (GDC_UCHAR numCursor, GDC_ULONG ladrs)

Parameter	numCursor	Cursor number (0 or 1)
	ladrs	Cursor pattern address
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_INVALID_CURSOR_NUMBER (Invalid cursor number is specified)

Description Sets the start address of the graphics memory where the cursor pattern is stored.

This command can be used by all “Graphics Controller”.

6.5.2 GdcCursorPattern [Sets cursor pattern]

Format int GdcCursorPattern (GDC_UCHAR numCursor, GDC_LPCOL8 lpCursor)

Parameter	numCursor	Cursor number (0 or 1)
	lpCursor	Pointer of cursor pattern
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_INVALID_CURSOR_NUMBER (Invalid cursor number is specified)

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.

Size of cursor pattern is fixed to 64*64.

This command can be used by all “Graphics Controller”.

6.5.3 GdcCursorDisplay [Controls cursor display]

Format	int GdcCursorDisplay (GDC_UCHAR numCursor, GDC_UCHAR enable)		
Parameter	numCursor	Cursor number (0 or 1)	
	enable	Cursor display ON/OFF	
		GDC_ENABLE	Cursor display ON
		GDC_DISABLE	Cursor display OFF
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	
Error code	GDC_ERR_INVALID_CURSOR_NUMBER (Invalid cursor number is specified)		
Description	Controls cursor display ON or OFF. This command can be used by all “Graphics Controller”.		

6.5.4 GdcCursorPos [Sets cursor display position]

Format	int GdcCursorPos (GDC_UCHAR numCursor, GDC_USHORT x, GDC_USHORT y)	
Parameter	numCursor	Cursor number (0 or 1)
	x	x coordinates of cursor display position
	y	y coordinates of cursor display position
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_INVALID_CURSOR_NUMBER (Invalid cursor number is specified)	
Description	Sets display position of cursor. This command can be used by all “Graphics Controller”.	

6.5.5 GdcCursorPriority [Sets cursor display priority mode]

Format	int GdcCursorPriority (GDC_UCHAR numCursor, GDC_UCHAR mode)	
Parameter	numCursor	Cursor number (0 or 1)
	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
Error code	GDC_ERR_INVALID_CURSOR_NUMBER (Invalid cursor number is specified)	
Description	Selects which is prioritized in display, C layer or cursor. This command can be used by all “Graphics Controller”.	

6.5.6 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
Error code	None	
Description	Sets a transparent color code for cursor. This command can be used by all “Graphics Controller”.	

6.5.7 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	

Error code GDC_ERR_INVALID_COLOR_MODE (Invalid color mode is specified)

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 Controller”.

6.6 Drawing Frame Control Commands

6.6.1 GdcDrawDimension [Sets drawing frame]

Format	int GdcDrawDimension (GDC_UCHAR cmode, GDC_ULONG dadrs, GDC_USHORT dw, GDC_USHORT dh)		
Parameter	cmode	Color mode	
		GDC_24BPP_FORMAT	24 bits color mode
		GDC_16BPP_FORMAT	16 bits color mode (default)
		GDC_8BPP_FORMAT	8 bits color mode
	dadrs	Drawing frame base address	
	dw	Drawing frame width (pixel unit)	
	dh	Drawing frame height (pixel unit)	
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	
Error code	GDC_ERR_INVALID_COLOR_MODE (Invalid color mode is specified)		
Description	Sets color mode and size of drawing frame. This command can be used by all “Graphics Controller”.		

6.6.2 GdcSetZPrecision [Sets precision of Z value]

Format	int GdcSetZPrecision (GDC_ULONG mode)		
Parameter	mode	Precision of Z value	
		GDC_Z_16BIT	Precision of Z value is 16 bits (default)
		GDC_Z_8BIT	Precision of Z value is 8 bits
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	
Error code	GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)		
Description	Sets precision of Z value. This command is for MB86293 or later.		

6.6.3 GdcBufferCreateZ [Sets Z buffer base address]

Format	int GdcBufferCreateZ (GDC_ULONG zadrs)	
Parameter	zadrs	Z buffer base address
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)	

Description

Sets the base address of Z buffer. The vertical/horizontal size of Z buffer is assumed to be equal to that of drawing frame.

When precision of Z value is 16 bits, memory size of 16 bits per 1 pixel is needed.

When precision of Z value is 8 bits, memory size of 8 bits per 1 pixel is needed.

This command can be used by all "Graphics Controller".

6.6.4 GdcBufferCreateC [Sets base address of polygon drawing flag buffer]

Format	int GdcBufferCreateC (GDC_ULONG cadrs)	
Parameter	cadrs	Polygon drawing control buffer base address
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)	

Description

Sets the base address of polygon drawing control buffer. The vertical/horizontal size of this control buffer is assumed to be equal to that of drawing frame. For each pixel, 1bit of data is required for this buffer.

This command can be used by all "Graphics Controller".

6.6.5 GdcBufferClearZ [Clears Z buffer]

Format	int GdcBufferClearZ (void)	
Parameter	None	
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)	

Description

Clears Z buffer. Prior to hidden surface manipulation, Z buffer should be cleared.

This command can be used by all "Graphics Controller".

6.6.6 GdcBufferClearC [Clears polygon drawing flag buffer]

Format	int GdcBufferClearC (void)	
Parameter	None	
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)	
Description	Clears polygon drawing flag buffer. This command can be used by all “Graphics Controller”.	

6.6.7 GdcDrawClipFrame [Sets drawing 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
Error code	GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)	
Description	Sets clip border of drawing. Clip border is set as a Blt located relatively from the base point of drawing frame. Drawing to the area outside of this clip border is not performed. This command can be used by all “Graphics Controller”.	

6.6.8 GdcSetAlphaMapBase [Sets base address of alpha map area]

Format	int GdcSetAlphaMapBase (GDC_ULONG adrs)	
Parameter	adrs	Alpha map area base address
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets the base address of alpha map area.

Alpha map area is an alpha coefficient area to be used by the GdcBltCopyAltAlpha command. An offset from the graphics memory top must be set to “adrs”.

For details about alpha map, refer to hardware specification of the “Graphics Controller”.

This command is for MB86293 of later.

6.7 Primitive Drawing Commands for Device Coordinates

6.7.1 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

Error code - GDC_ERR_INVALID_PRIMITIVE (Invalid primitive is specified)

- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets the primitive type to be drawn by the GdcDrawVertex2D[i] or the GdcDrawVertex3D[f] command. Once either of these commands is executed, same type of primitive will keep being drawn till the GdcPrimEnd command will be executed.

This command can be used by all “Graphics Controller”.

6.7.2 GdcPrimEnd [Completes drawing procedure]

Format void GdcPrimEnd (void)

Parameter None

Return value None

Description Stops drawing the primitive applied by the GdcPrimType command.

This command can be used by all “Graphics Controller”.

6.7.3 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 the vertex coordinate setting command. Once this command is executed, the same texture coordinates is continuously applied till this command will be executed.

The GdcTexCoord2D command must be used when the type of texture coordinate is GDC_FIXED32.

The GdcTexCoord2Df command must be used when the type of texture coordinate is GDC_SFLOAT.

The GdcTexCoord2DNf command must be used when the type of texture coordinate is GDC_SFLOAT and normalized. In this case, the range of texture coordinate must be within 0.0 to 1.0. The minimum size of texture coordinate is 0.0 and the maximum size is 1.0.

This command is applicable to the following primitives:

GDC_TRIANGLES
GDC_TRIANGLE_STRIP
GDC_TRIANGLE_FAN

This command can be used by all “Graphics Controller”.

6.7.4 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 the vertex coordinate setting command. Once this command is executed, the same texture coordinates is continuously applied till this command will be executed.

The GdcTexCoord3D command must be used when the type of texture coordinate is GDC_FIXED32.

The GdcTexCoord3Df command must be used when the type of texture coordinate is GDC_SFLOAT.

The GdcTexCoord3DNf command must be used when the type of texture coordinate is GDC_SFLOAT and normalized. In this case, the range of texture coordinate must be within 0.0 to 1.0. The minimum size of texture coordinate is 0.0 and the maximum size is 1.0.

This command is applicable to the following primitives:

GDC_TRIANGLES
GDC_TRIANGLE_STRIP
GDC_TRIANGLE_FAN

This command can be used by all “Graphics Controller”.

6.7.5 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 drawing a designated primitive.

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.

The GdcDrawVertex2D command must be used when the type of vertex coordinate is GDC_FIXED32.

The GdcDrawVertex2Di command must be used when the type of vertex coordinate is GDC_LONG.

The 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 Controller”.

6.7.6 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 drawing a designated primitive.

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.

The GdcDrawVertex3D command must be used when the type of vertex coordinate is GDC_FIXED32.

The GdcDrawVertex3Df command must be used when the type of vertex coordinate is GDC_SFLOAT.

The GdcDrawVertex3Df command is applicable to the following primitives:

GDC_TRIANGLES
GDC_TRIANGLE_STRIP
GDC_TRIANGLE_FAN

When drawing a polygon primitive (GDC_POLYGON), z coordinates of a parameter is disregarded.

This command can be used by all “Graphics Controller”.

6.7.7 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

Error code None

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

This command can be used by all “Graphics Controller”.

6.8 Primitive Drawing Control Commands for Object Coordinates

6.8.1 GdcGeoPrimType [Starts drawing procedure]

Format int GdcGeoPrimType (GDC_UCHAR type)

Parameter Type Sets primitive type

GDC_POINTS	Point
GDC_LINES	Line
GDC_POLYLINE	Poly-line
GDC_TRIANGLES	Triangle
GDC_TRIANGLE_STRIP	Triangle strip
GDC_TRIANGLE_FAN	Triangle fan
GDC_POLYGON	Polygon

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code - GDC_ERR_INVALID_PRIMITIVE (Invalid primitive is specified)

- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets primitive drawn with GdcGeoDrawVertex2D[f/i] or GdcGeoDrawVertex3D[f/i].

Once this command is executed, the same primitive is drawn until the GdcGeoPrimEnd command is executed.

This command is for MB86291 or later.

6.8.2 GdcGeoPrimEnd [Completes drawing procedure]

Format void GdcGeoPrimEnd (void)

Parameter None

Return value None

Description Terminates a series of processes to draw primitives fallowing the GdcGeoPrimType command.

This command is for MB86291 or later.

6.8.3 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 drawing 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.

The GdcGeoDrawVertex2D command must be used when the type of vertex coordinate is GDC_FIXED32.

The GdcGeoDrawVertex2Df command must be used when the type of vertex coordinate is GDC_SFLOAT.

The GdcGeoDrawVertex2Di command must be used when the type of vertex coordinate is GDC_LONG.

This command is for MB86291 or later.

6.8.4 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 drawing 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.

The GdcGeoDrawVertex3D command must be used when the type of vertex coordinate is GDC_FIXED32.

The GdcGeoDrawVertex3Df command must be used when the type of vertex coordinate is GDC_SFLOAT.

The GdcGeoDrawVertex3Di command must be used when the type of vertex coordinate is GDC_LONG.

However, when drawing a polygon primitive (GDC_POLYGON) by MB86291/86292, z coordinates of a parameter is disregarded.

This command is for MB86291 or later.

6.8.5 GdcGeoTexCoord2DN / 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. This command treat texture coordinates as normalized (1.0 is maximum size of current texture).

The GdcGeoTexCoord2DN command must be used when the type of texture coordinate is GDC_FIXED32.

The GdcGeoTexCoord2DNf command must be used when the type of texture coordinate is GDC_SFLOAT.

This command is applicable to the following primitives:

GDC_TRIANGLES
GDC_TRIANGLE_STRIP
GDC_TRIANGLE_FAN

However, MB86293 or later can also be used the following primitives:

GDC_POLYGON

This command is for MB86291 or later.

6.8.6 GdcVertexColor3f / 32 [Sets color of vertex]

Format	void GdcVertexColor32 (GDC_COLOR32 color) void GdcVertexColor3f (GDC_SFLOAT r, GDC_SFLOAT g, GDC_SFLOAT b)	
Parameter	color	Packed format in which each color elements (r,g,b) is normalized to [0,255]. In this case, r,g,b are 8 bits respectively.
	r, g, b	Normalized values in which each color elements (r,g,b) are normalized to [0,1].
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.

The GdcVertexColor32 command must be used when the type of vertex color is GDC_COLOR32.

The GdcVertexColor3f command must be used when the type of vertex color is GDC_SFLOAT.

When drawing a polygon (GDC_POLYGON), setup of this command is disregarded.

This command is for MB86291 or later.

6.9 Drawing Attribute Control Commands

6.9.1 GdcColor [Sets vertex color/foreground color]

Format	int GdcColor (GDC_COLOR32 color)	
Parameter	color	Vertex and foreground color
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	None	

Description

Sets vertex color and foreground color applied for bitmap drawing and broken line drawing to be executed by set coordinates of vertex command. Once this command is executed, the same color is continuously applied till this command will be executed.

The following values are used by color mode.

- 8 bits color mode	: lower 8 bits of “color”
- 16 bits color mode	: lower 16 bits of “color”
- 24 bits color mode	: lower 24 bits of “color”

This command can be used by all “Graphics Controller”.

6.9.2 GdcBackColor [Sets background color]

Format	int GdcBackColor (GDC_COLOR32 color)	
Parameter	color	Background color
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	None	

Description

Sets background color applied for binary pattern drawing and broken line drawing. Once this command is executed, the same color is continuously applied till this command will be executed.

The following values are used by color mode.

- 8 bits color mode : lower 8 bits of "color"
- 16 bits color mode : lower 16 bits of "color"
- 24 bits color mode : lower 24 bits of "color"

In order to make background color transparent, sets the following bit as 1 by color mode.

- 8 bits color mode : bit 15 of "color"
- 16 bits color mode : bit 15 of "color"
- 24 bits color mode : bit 31 of "color"

This command can be used by all "Graphics Controller".

6.9.3 GdcClipMode [Sets clipping mode]

Format int GdcClipMode (GDC_ULONG mode)

Parameter mode Clipping mode

GDC_CLIP_X_ON	Validates clipping toward x axis
GDC_CLIP_Y_ON	Validates clipping toward y axis
GDC_CLIP_DISABLE	Invalidates clipping

Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)	

Description

Sets clipping mode.

GDC_CLIP_X_ON and GDC_CLIP_Y_ON can be set at the same time with OR operator.

This command can be used by all "Graphics Controller".

6.9.4 GdcSetAttrLine [Sets line drawing attribute]

Format int GdcSetAttrLine (GDC_ULONG target, GDC_ULONG param)

Parameter target Line drawing 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

[When the “Graphics Controller” is MB86291 or later, following functions are available]

GDC_BROKEN_LINE_OFFSET	Offset control of broken line pattern
GDC_BROKEN_LINE_PERIOD	Period set of broken line pattern

[When the “Graphics Controller” is MB86293 or later, following functions are available]

GDC_SHADOW_DEPTH_TEST	Z value compare mode of shadow
GDC_SHADOW_DEPTH_FUNC	Z value compare type of shadow
GDC_SHADOW_DEPTH_WRITE_MASK	Z value write permission mask of shadow
GDC_SHADOW_BLEND_MODE	Blending mode of shadow
GDC_SHADOW_BROKEN_LINE	Broken line mode of shadow
GDC_SHADOW_LINE_WIDTH	Line width of shadow
GDC_SHADOW_BROKEN_LINE_PERIOD	Period set of broken line pattern of shadow
GDC_BORDER_DEPTH_TEST	Z value compare mode of border
GDC_BORDER_DEPTH_FUNC	Z value compare type of border
GDC_BORDER_DEPTH_WRITE_MASK	Z value write permission mask of border
GDC_BORDER_BLEND_MODE	Blending mode of border
GDC_BORDER_BROKEN_LINE	Broken line mode of border
GDC_BORDER_LINE_WIDTH	Line width of border
GDC_BORDER_BROKEN_LINE_PERIOD	Period set of broken line pattern of border

param Parameter corresponding to target (*1)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

- GDC_ERR_ILLEGAL_LINE_WIDTH (Illegal width of line)

- GDC_ERR_INVALID_ATTRIBUTE (Invalid attribute is specified)

Description Sets attribute for line drawing.

This command can be used by all “Graphics Controller”.

(*1) Line drawing attribute (target) and parameter (param) corresponding to each line drawing attribute is shown below.

[Explanatory notes]

Line drawing attribute	Description of line drawing attribute
Parameter 1 that can set	Description of parameter 1
Parameter 2 that can set	Description of parameter 2
:	:
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 drawing operation (writes pixel color to drawing 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 bits period.
GDC_BROKEN_LINE_24	24 bits period.
GDC_SHADOW_DEPTH_TEST	Sets Z value compare mode of shadow (only for MB86293 or later).
GDC_ENABLE	Validates Z value comparison.
GDC_DISABLE	Invalidates Z value comparison.
GDC_SHADOW_DEPTH_FUNC	Selects Z value comparison type of shadow (only for MB86293 or later).
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_SHADOW_DEPTH_WRITE_MASK	Enables write access to Z buffer of shadow (only for MB86293 or later).
GDC_ENABLE	Disable Z buffer write.
GDC_DISABLE	Enable Z buffer write.
GDC_SHADOW_BLEND_MODE	Sets blending mode of pixel write of shadow (only for MB86293 or later).
GDC_BLEND_COPY	Regular drawing operation (writes pixel color to drawing frame).
GDC_BLEND_ALPHA	Enables alpha blending.
GDC_BLEND_ROP	Draws with logical arithmetic.
GDC_SHADOW_BROKEN_LINE	Selects broken line mode of shadow (only for MB86293 or later).
GDC_ENABLE	Draws a broken line utilizing applied line pattern.
GDC_DISABLE	Draws a solid line.
GDC_SHADOW_LINE_WIDTH	Sets line width of shadow (only for MB86293 or later).
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_SHADOW_BROKEN_LINE_PERIOD	Sets broken line pattern period of shadow (only for MB86293 or later).
GDC_BROKEN_LINE_32	32 bits period.
GDC_BROKEN_LINE_24	24 bits period.

GDC_BORDER_DEPTH_TEST	Sets Z value compare mode of border (only for MB86293 or later).
GDC_ENABLE	Validates Z value comparison.
GDC_DISABLE	Invalidates Z value comparison.
GDC_BORDER_DEPTH_FUNC	Selects Z value comparison type of border (only for MB86293 or later).
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_BORDER_DEPTH_WRITE_MASK	Enables write access to Z buffer of border (only for MB86293 or later).
GDC_ENABLE	Disable Z buffer write.
GDC_DISABLE	Enable Z buffer write.
GDC_BORDER_BLEND_MODE	Sets blending mode of pixel write of border (only for MB86293 or later).
GDC_BLEND_COPY	Regular drawing operation (writes pixel color to drawing frame).
GDC_BLEND_ALPHA	Enables alpha blending.
GDC_BLEND_ROP	Draws with logical arithmetic.
GDC_BORDER_BROKEN_LINE	Selects broken line mode of border (only for MB86293 or later).
GDC_ENABLE	Draws a broken line utilizing applied line pattern.
GDC_DISABLE	Draws a solid line.
GDC_BORDER_LINE_WIDTH	Sets line width of border (only for MB86293 or later).
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_BORDER_BROKEN_LINE_PERIOD	Sets broken line pattern period of border (only for MB86293 or later).
GDC_BROKEN_LINE_32	32 bits period.
GDC_BROKEN_LINE_24	24 bits period.

6.9.5 GdcSetAttrSurf [Sets surface drawing attribute]

Format int GdcSetAttrSurf (GDC_ULONG target, GDC_ULONG param)

Parameter target Surface drawing 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

[When the “Graphics Controller” is MB86293 or later, following functions are available]

GDC_ALPHA_SHADE_MODE	Alpha shading mode
GDC_SHADOW_DEPTH_TEST	Z value compare mode of shadow
GDC_SHADOW_DEPTH_FUNC	Z value compare type of shadow
GDC_SHADOW_DEPTH_WRITE_MASK	Z value write mask of shadow
GDC_SHADOW_BLEND_MODE	Blending mode of shadow
GDC_NON_TOPLEFT_SHADE_MODE	Shading mode of non top-left primitive
GDC_NON_TOPLEFT_DEPTH_TEST	Z value compare mode of non top-left primitive
GDC_NON_TOPLEFT_DEPTH_FUNC	Z value compare type of non top-left primitive
GDC_NON_TOPLEFT_DEPTH_WRITE_MASK	Z value write mask of non top-left primitive
GDC_NON_TOPLEFT_BLEND_MODE	Blending mode of non top-left primitive
GDC_NON_TOPLEFT_TEXTURE_SELECT	Texture mode of non top-left primitive
GDC_NON_TOPLEFT_ALPHA_SHADE_MODE	Alpha shading mode of non top-left primitive

param Parameter corresponding to target (*1)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

- GDC_ERR_INVALID_ATTRIBUTE (Invalid attribute is specified)

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

This command can be used by all “Graphics Controller”.

(*1) Surface drawing attribute (target) and parameter (param) corresponding to each surface drawing attribute is shown below.

[Explanatory notes]

Surface drawing attribute		Description of surface drawing attribute
	Parameter 1 that can set	Description of parameter 1
	Parameter 2 that can set	Description of parameter 2
	:	:

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 drawing operation (writes pixel color to drawing 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.
GDC_ALPHA_SHADE_MODE		Sets alpha shading mode (only for MB86293 or later).
	GDC_SHADE_FLAT	Flat shading.
	GDC_SHADE_SMOOTH	Gouraud shading.

GDC_SHADOW_DEPTH_TEST	Sets Z value compare mode of shadow (only for MB86293 or later).
GDC_ENABLE	Validate Z value comparison.
GDC_DISABLE	Invalidate Z value comparison.
GDC_SHADOW_DEPTH_FUNC	Selects Z value comparison type of shadow (only for MB86293 or later).
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_SHADOW_DEPTH_WRITE_MASK	Enables write access to Z buffer of shadow (only for MB86293 or later). 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_SHADOW_BLEND_MODE	Sets blending mode of pixel write of shadow (only for MB86293 or later).
GDC_BLEND_COPY	Regular drawing operation (writes pixel color to drawing frame).
GDC_BLEND_ALPHA	Enables alpha blending.
GDC_BLEND_ROP	Draws with logical arithmetic.

GDC_NON_TOPLEFT_SHADE_MODE	Sets shading mode of non top-left primitive (only for MB86293 or later).
GDC_SHADE_FLAT	Flat shading.
GDC_SHADE_SMOOTH	Gouraud shading.
GDC_NON_TOPLEFT_DEPTH_TEST	Sets Z value compare mode of non top-left primitive (only for MB86293 or later).
GDC_ENABLE	Validate Z value comparison.
GDC_DISABLE	Invalidate Z value comparison.
GDC_NON_TOPLEFT_DEPTH_FUNC	Selects Z value comparison type of non top-left primitive (only for MB86293 or later).
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_NON_TOPLEFT_DEPTH_WRITE_MASK	Enables write access to Z buffer of non top-left primitive (only for MB86293 or later). 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_NON_TOPLEFT_BLEND_MODE	Sets blending mode of pixel write of non top-left primitive (only for MB86293 or later).
GDC_BLEND_COPY	Regular drawing operation (writes pixel color to drawing frame).
GDC_BLEND_ALPHA	Enables alpha blending.
GDC_BLEND_ROP	Draws with logical arithmetic.
GDC_NON_TOPLEFT_TEXTURE_SELECT	Sets texture mapping mode of non top-left primitive (only for MB86293 or later).
GDC_SELECT_TEXTURE	Draws with texture mapping.
GDC_SELECT_TILE	Draws with tiling.
GDC_SELECT_PLAIN	Invalidates texture mapping.
GDC_NON_TOPLEFT_ALPHA_SHADE_MODE	Sets alpha shading mode of non top-left primitive (only for MB86293 or later).
GDC_SHADE_FLAT	Flat shading.
GDC_SHADE_SMOOTH	Gouraud shading.

6.9.6 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

[When the “Graphics Controller” is MB86293 or later, following function is available]

GDC_TEXTURE_FAST_MODE	Bi-linear filtering fast mode

param Parameter corresponding to target (*1)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

- GDC_ERR_INVALID_ATTRIBUTE (Invalid attribute is specified)

Description Sets attribute for texture mapping.

This command can be used by all “Graphics Controller”.

(*1) Texture mapping attribute (target) and parameter (param) corresponding to each texture mapping attribute is shown below.

[Explanatory notes]

Texture mapping attribute	Description of texture mapping attribute
Parameter 1 that can set	Description of parameter 1
Parameter 2 that can set	Description of parameter 2
:	:

GDC_TEXTURE_PERSPECTIVE	Selects perspective correction mode.
GDC_ENABLE	Validates perspective correction.
GDC_DISABLE	Invalidates perspective correction (default).
GDC_TEXTURE_FILTER	Selects texture filter mode.
GDC_TEXTURE_POINT	Point sampling mode (default).
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 (default).
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 (default).
- 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 — Texture color is drawn (default).
- GDC_TEXTURE_MODULATE — 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 drawing 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 drawing frame (default).
- 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 drawing frame is performed, otherwise not drawn.

GDC_TEXTURE_FAST_MODE — Sets bi-linear fast mode (only for MB86293 or later).

- GDC_ENABLE — Texture mapping is executed at fast speed using a default * 4 as texture area.
- GDC_DISABLE — A default texture area is used (default).

6.9.7 GdcSetAttrBlt [Sets BitBlt attribute]

Format int GdcSetAttrBlt (GDC_ULONG target, GDC_ULONG param)

Parameter target Bitmap drawing attribute

GDC_BLEND_MODE Blend mode

[When the "Graphics Controller" is MB86291 or later, following function is available]
GDC_TRANSPARENT_MODE Transparent mode

param Parameter corresponding to target (*1)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code - GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)

- GDC_ERR_INVALID_ATTRIBUTE (Invalid attribute is specified)

Description Sets attribute when copying and drawing BitBlt.

This blend mode setting function can be used by all "Graphics Controller".

The transparent mode setting function is only for MB86291 or later.

Sets the transparent color for the transparent mode with the GdcBltColorTransparent command.

(*1) Bitmap drawing attribute (target) and parameter (param) corresponding to each bitmap drawing attributes are shown below.

[Explanatory notes]

Bitmap drawing attribute	Description of bitmap drawing attributes
Parameter 1 that can set	Description of parameter 1
Parameter 2 that can set	Description of parameter 2
:	:

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

GDC_TRANSPARENT_MODE	Sets transparent mode (only for MB86291 or later).
GDC_ENABLE	The color which was set by the GdcBltColorTransparent command regards as transparent color.
GDC_DISABLE	The color which was set by the GdcBltColorTransparent command is not treated as transparent color (default).

6.9.8 GdcSetAlpha [Sets alpha blend ratio]

Format	int GdcSetAlpha (GDC_UCHAR alpha)	
Parameter	alpha	Alpha blending ratio (0 - 255)
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)	
Description	Sets the ratio of alpha blending.	

This command can be used by all "Graphics Controller".

6.9.9 GdcSetLinePattern [Sets broken line pattern]

Format	int GdcSetLinePattern (GDC_ULONG pattern)	
Parameter	pattern	Broken line pattern
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)	
Description	Sets broken line pattern with 32 bits pattern when drawing broken line.	

Upper 24 bits pattern values are set as a broken line pattern, when the period of broken line pattern is 24 bits.

Broken line is drawn with foreground color when the bit of pattern is 1, and drawn with background color when the bit is 0.

Example of broken line with broken line pattern "0xaaaaaaaa" is shown in the figure 6.9.9.

FBFBFBFBFBFBFBFBFBFBFBFBFBFBFBFB

[meaning of symbol]
F: Foreground color
B: Background color

Figure 6.9.9 Example of broken line

This command can be used by all "Graphics Controller".

6.9.10 GdcSetTextureBorder [Sets texture border color]

Format	int GdcSetTextureBorder (GDC_COLOR32 color)	
Parameter	color	Texture border color
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)	

Description

Sets the border color of the texture applied in border mode of texture wrap.

The following values are used by color mode.

- 8 bits color mode	: lower 8 bits of “color”
- 16 bits color mode	: lower 16 bits of “color”
- 24 bits color mode	: lower 24 bits of “color”

This command can be used by all “Graphics Controller”.

6.9.11 GdcSetRop [Sets logical calculation mode]

Format int GdcSetRop (GDC_UCHAR mode)

Parameter mode Logical arithmetic mode

GDC_ROP_CLEAR	All bits are set to 0
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	All bits are set to 1

s:drawing value
d:destination value

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets logical arithmetic type.

This operation is performed between the pixel color to be drawn and current pixel color in the drawing frame. Result of this operation is to be drawn to the drawing 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 Controller”.

6.10 Attribute Control Commands for Object Coordinates

6.10.1 GdcGeoSetAttrMisc [Sets miscellaneous attribute]

Format int GdcGeoSetAttrMisc (GDC_ULONG target, GDC_ULONG param)

Parameter target Geometry attribute

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 corresponding to target (*1)

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

- GDC_ERR_INVALID_ATTRIBUTE (Invalid attribute is specified)

Description Sets miscellaneous attribute for object coordinates.

Attributes and parameters for them are described below.

This command is for MB86291 or later.

(*1) Miscellaneous attribute (target) and parameter (param) corresponding to each miscellaneous attribute is shown below.

[Explanatory notes]

Miscellaneous attribute	Description of miscellaneous attribute
Parameter 1 that can set	Description of parameter 1
Parameter 2 that can set	Description of parameter 2
:	:

GCD_GEO_VTX_COL	Specifies when using color of vertex. Smooth shading mode (GDC_SHADE_SMOOTH) should also set smooth shading by the shade mode (GDC_SHADE_MODE) of the GdcSetAttrSurf command.
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. Texture mapping (GDC_SELECT_TEXTURE) should also set texture mapping by the texture mapping mode (GDC_TEXTURE_SELECT) of the GdcSetAttrSurf command.
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.

6.10.2 GdcGeoSetAttrLine [Sets line drawing attribute for object coordinates]

Format int GdcGeoSetAttrLine (GDC_ULONG target, GDC_ULONG param)

Parameter target Line drawing attribute

GDC_GEO_THICK_LINE_CORRECT	Sets correction mode of thick line connection
GDC_GEO_BROKEN_LINE_CORRECT	Sets correction mode of broken line pattern
GDC_GEO_BROKEN_LINE_CORRECT_LENGTH	Sets the number of broken line pattern address fixation pixel
GDC_GEO_UNIFORM_LINE_WIDTH	Sets uniform mode of line width
GDC_GEO_THICK_LINE_VERTICAL	Sets thick/broken line vertical mode
GDC_GEO_BORDER_LINE	Sets drawing mode of border primitive
GDC_GEO_SHADOW_MODE	Sets drawing mode of shadow primitive

param Parameter corresponding to target (*1)

Return value GDC_TRUE Complete

GDC_FALSE Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

- GDC_ERR_INVALID_ATTRIBUTE (Invalid attribute is specified)

Description Sets line drawing attribute in object coordinates.

This command is for MB86293 or later.

(*1) Line drawing attribute (target) and parameter (param) corresponding to each line drawing attribute is shown below.

[Explanatory notes]

Line drawing attribute	Description of line drawing attribute
Parameter 1 that can set	Description of parameter 1
Parameter 2 that can set	Description of parameter 2
:	:

GDC_GEO_THICK_LINE_CORRECT	Sets thick line connection correct mode.
GDC_ENABLE	Enables thick line connection correct.
GDC_DISABLE	Disables thick line connection correct.
GDC_GEO_BROKEN_LINE_CORRECT	Sets broken line connection correct mode.
GDC_ENABLE	Refer to the same broken line pattern with front and back number pixel of broken line connection part (“broken line pattern address fixation mode”). Number of pixel is set by GDC_GEO_BROKEN_LINE_CORRECT_LENGTH.
GDC_DISABLE	Not correct broken line pattern (Default).
GDC_GEO_BROKEN_LINE_CORRECT_LENGTH	Sets the number of broken line pattern address fixation pixel. A recommended value is the same as line width. This parameter is available when the correction mode of broken line pattern is “broken line pattern address fixation mode”.
0-32	Number of pixel (default is 0).
GDC_GEO_UNIFORM_LINE_WIDTH	Sets uniform mode of line width.
GDC_ENABLE	Enables uniform of line width.
GDC_DISABLE	Disables uniform of line width.
GDC_GEO_THICK_LINE_VERTICAL	Sets thick/broken line vertical mode.
GDC_ENABLE	Draws perpendicular section of thick/broken line to an ideal line.
GDC_DISABLE	Draws perpendicular section of thick/broken line to a base axis.
GDC_GEO_BORDER_LINE	Sets drawing mode of border primitive.
GDC_ENABLE	Draws border primitive.
GDC_DISABLE	Not draw border primitive.

GDC_GEO_SHADOW_MODE	Sets drawing mode of shadow primitive.
GDC_ENABLE	Draws shadow primitive.
GDC_DISABLE	Not draw shadow primitive.

[Note] Following combinations are not available (not supported)

1. GDC_GEO_THICK_LINE_VERTICAL = GDC_ENABLE
   GDC_GEO_UNIFORM_LINE_WIDTH = GDC_DISABLE
2. GDC_GEO_THICK_LINE_VERTICAL = GDC_DISABLE
   GDC_GEO_THICK_LINE_CORRECT = GDC_ENABLE
3. GDC_GEO_THICK_LINE_VERTICAL = GDC_DISABLE
   GDC_GEO_BORDER_LINE = GDC_ENABLE
4. GDC_GEO_THICK_LINE_VERTICAL = GDC_DISABLE
   GDC_GEO_UNIFORM_LINE_WIDTH = GDC_ENABLE
5. GDC_GEO_THICK_LINE_VERTICAL = GDC_DISABLE
   GDC_GEO_BROKEN_LINE_CORRECT = GDC_ENABLE

6.10.3 GdcGeoSetAttrSurf [Sets surface drawing attribute for object coordinates]

Format int GdcGeoSetAttrSurf (GDC_ULONG target, GDC_ULONG param)

Parameter target Surface drawing attribute

GDC_GEO_FACE_CULL	Enable/disable culling back face of triangle
GDC_GEO_FACE_INVERT	Specify direction of surface of triangle

[When the “Graphics Controller” is MB86293 or later, following functions are available]

GDC_GEO_NON_TOPLEFT	Sets drawing algorithm
GDC_GEO_SHADOW_MODE	Sets drawing mode of shadow primitive

param Parameter corresponding to target (*1)

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

- GDC_ERR_INVALID_ATTRIBUTE (Invalid attribute is specified)

Description Sets surface drawing attribute in object coordinates.

It doesn’t affect polygons.

Attributes for surface drawing and parameters for them are described below.

This command is for MB86291 or later.

(*1) Surface drawing attribute (target) and parameter (param) corresponding to each surface drawing attribute is shown below.

[Explanatory notes]

Surface drawing attribute	Description of surface drawing attribute
Parameter 1 that can set	Description of parameter 1
Parameter 2 that can set	Description of parameter 2
:	:

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.
GCD_GEO_NON_TOPLEFT	Sets drawing algorithm.
GDC_ENABLE	Non top left applying rule is used.
GDC_DISABLE	Non top left applying rule is not used.
GDC_GEO_SHADOW_MODE	Sets drawing mode of shadow primitive.
GDC_ENABLE	Draws shadow primitive.
GDC_DISABLE	Not draw shadow primitive.

6.10.4 GdcGeoLoadMatrix[f] [Sets matrix]

Format int GdcGeoLoadMatrix (GDC_FIXED32 *ptMatrix)

int 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	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

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 for MB86291 or later.

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

Format int GdcGeoNdcDcViewportCoef (GDC_FIXED32 scalex, GDC_FIXED32 offsetx,
GDC_FIXED32 scaley, GDC_FIXED32 offsety)
int 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	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

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 for MB86291 or later.

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

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

Parameter	scalez	Magnification of z
	offsetz	Offset of z
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets the magnification and offset of z that is used for transforming Normalized Device Coordinates (NDC) to Device Coordinates (DC).

This command is for MB86291 or later.

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

Format int GdcGeoViewVolumeXYClip (GDC_FIXED32 xmin, GDC_FIXED32 xmax,
GDC_FIXED32 ymin, GDC_FIXED32 ymax)

int 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	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets the view volume boundary in the clip coordinates for x, y.

This command is for MB86291 or later.

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

Format int GdcGeoViewVolumeZClip (GDC_FIXED32 zmin, GDC_FIXED32 zmax)

int GdcGeoViewVolumeZClipf (GDC_SFLOAT zmin, GDC_SFLOAT zmax)

Parameter	zmin	Minimum clip value of z
	zmax	Maximum clip value of z
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets the view volume boundary in the clip coordinates for z.

This command is for MB86291 or later.

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

Format int GdcGeoViewVolumeWminClip (GDC_FIXED32 wmin)

int GdcGeoViewVolumeWminClipf (GDC_SFLOAT wmin)

Parameter	wmin	Minimum clip value of w
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

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 for MB86291 or later.

6.10.10 GdcGeoSetLogOutBase [Sets base address for log output of device coordinates]

Format	int GdcGeoSetLogOutBase (GDC_ULONG adrs)	
Parameter	adrs	Base address for log output of device coordinates (specified by offset address from top of the graphics memory)
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)	

Description

Sets base address for log output of device coordinates.

"adrs" must be specified by offset address from top of the graphics memory.

Log format consists of packed number of x and y coordinates of vertex (refer to figure 6.10.10).

bit

31 30 29	28	27 16	15 14 13	12	11 0
0 0 0	S	y	0 0 0	S	x

S:signed bit

y:y coordinates values (integer)

x:x coordinates values (integer)

All other bits:0

Figure 6.10.10 Log format of device coordinates

Log needs 4 byte for every vertex.

Therefore, memory area must be allocated with size of number of vertex * 4 byte.

These vertices are specified between the GdcGeoPrimType command and the GdcGeoPrimEnd command.

Address for log output is added by 4 byte whenever one log is outputted.

If this memory area is used repeatedly, sets the base address for log output again.

Base address for log output must be set after drawing is finished by "drawing command end interruption" or the GdcGetPixelEngienStatus command or the GdcGeoGetPixelEngineStatus command.

This command is for MB86293 or later.

6.10.11 GdcGeoSetLogOutMode [Sets log output mode of the device coordinates]

Format int GdcGeoSetLogOutMode (GDC_ULONG mode)

Parameter mode Log output mode of device coordinate

GDC_GEO_LOGOUT_ENABLE	Outputs logs
GDC_GEO_LOGOUT_DISABLE	Not output logs (default)
GDC_GEO_LOGOUT_ONLY	Output logs without drawing

Return value

GDC_TRUE	Complete
GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets log output mode of device coordinates. Each mode is explained below.

[GDC_GEO_LOGOUT_ENABLE mode]

Log is outputted and drawing is executed.

[GDC_GEO_LOGOUT_DISABLE mode]

Log is not outputted and drawing is executed.

[GDC_GEO_LOGOUT_ONLY mode]

Log is outputted and drawing is not executed.

This mode is available only when drawing point primitive.

When “GDC_GEO_LOGOUT_ONLY” is specified, log is not outputted when primitive except point is drawn.

This command is for MB86293 or later.

6.10.12 GdcGeoShadowXY [Sets xy offset of shadow]

Format	int GdcGeoShadowXY (GDC_ULONG type, GDC_LONG offsetx, GDC_LONG offsety)		
Parameter	type	A kind of primitive	
		GDC_GEO_SHADOW	Shadow primitive
		GDC_GEO_SHADOW_COMPOSITION	Shadow composition primitive
	offsetx	x offset of shadow (or shadow composition) primitive for body primitive	
	offsety	y offset of shadow (or shadow composition) primitive for body primitive	
Return value	GDC_TRUE	Complete	
	GDC_FALSE	Incomplete	
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)		

Description

Sets offset of shadow from body. Offset must be specified with pixel unit.

When offset is positive number, position of x is right side of body, y is lower side of body.

When offset is negative number, position of x is left side of body, y is upper side of body.

Offset position of shadow form body must be set before drawing shadow primitive.

Shadow primitive drawing function is available in object coordinates drawing.

This command is for MB86293 or later.

6.10.13 GdcGeoOverlapZ [Sets Z value of primitives (body / shadow / border / correction in top-left rule non-applicable mode)]

Format

```
int   GdcGeoOverlapZ (GDC_ULONG origin_offset,
                      GDC_ULONG non_topleft_offset,
                      GDC_ULONG border_offset,
                      GDC_ULONG shadow_offset)
```

Parameter	origin_offset	Z value of body primitive
	non_topleft_offset	Z value of correction primitive in top-left rule non-applicable mode
	border_offset	Z value of border primitive
	shadow_offset	Z value of shadow primitive
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets Z value of body, shadow, border and correction primitive in top-left rule non-applicable mode.

Z value must be set before drawing these primitives.

When precision of Z value is 8 bits, lower 8 bits of each parameter is effective.

When precision of Z value is 16 bits, lower 16 bits of each parameter is effective.

This command is for MB86293 or later.

6.10.14 GdcGeoShadowColor [Sets color or shadow]

Format	int GdcGeoShadowColor (GDC_COLOR32 color)	
Parameter	color	Sets color of shadow
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)	

Description

Sets color of shadow. This setting is effective in drawing lines with shadow, triangles with shadow, and polygons with shadow.

Once this command is executed, the same color is continuously applied till this command will be executed.

The following values are used by color mode.

- 8 bits color mode	: lower 8 bits of "color"
- 16 bits color mode	: lower 16 bits of "color"
- 24 bits color mode	: lower 24 bits of "color"

This command is for MB86293 or later.

6.10.15 GdcGeoShadowBackColor [Sets background color of shadow]

Format int GdcGeoShadowBackColor (GDC_COLOR32 color)

Parameter	color	Sets background color of shadow
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)

Description Sets background color of shadow. Background color of shadow is effective when drawing lines with shadow and it's shadow is broken line.

Background color is corresponding to 0 in bits of broken line pattern when drawing shadow as broken line (refer to figure 6.10.15).

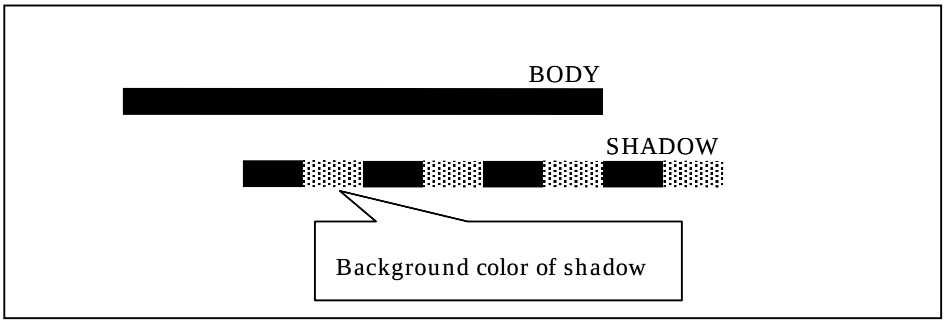


Figure 6.10.15 Relationship between foreground and background color of shadow when the broken line pattern is "0xf0f0f0f0"

Once this command is executed, the same color is continuously applied till this command will be executed.

The following values are used by color mode.

- 8 bits color mode : lower 8 bits of "color"
- 16 bits color mode : lower 16 bits of "color"
- 24 bits color mode : lower 24 bits of "color"

This command is for MB86293 or later.

6.10.16 GdcGeoBorderColor [Sets color or border]

Format int GdcGeoBorderColor (GDC_COLOR32 color)

Parameter	color	Sets color of border
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets color of border. This setting is effective in drawing lines with border.

Once this command is executed, the same color is continuously applied till this command will be executed.

The following values are used by color mode.

- 8 bits color mode	: lower 8 bits of “color”
- 16 bits color mode	: lower 16 bits of “color”
- 24 bits color mode	: lower 24 bits of “color”

This command is for MB86293 or later.

6.10.17 GdcGeoBorderBackColor [Sets background color of border]

Format	int GdcGeoBorderBackColor (GDC_COLOR32 color)	
Parameter	color	Sets background color of border
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)

Description Sets background color of border. Background color of border is effective when drawing lines with border and it's border is broken line.

Background color is corresponding to 0 in bits of broken line pattern when drawing border as broken line (refer to figure 6.10.17).

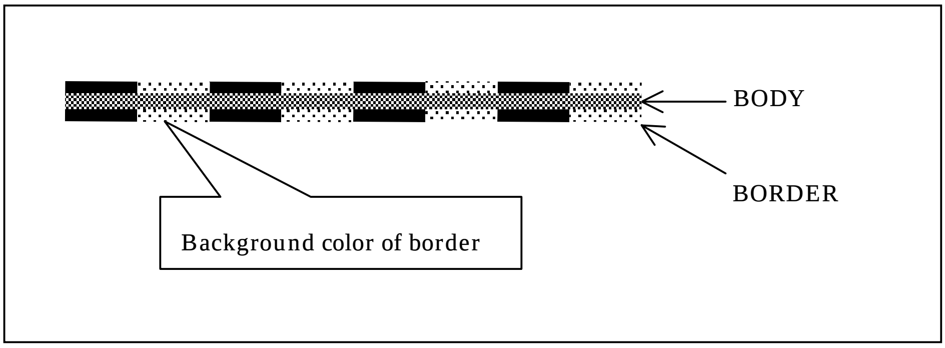


Figure 6.10.17 Relationship between foreground and background color of border when the broken line pattern is "0xf0f0f0f0"

Once this command is executed, the same color is continuously applied till this command will be executed.

The following values are used by color mode.

- 8 bits color mode	: lower 8 bits of "color"
- 16 bits color mode	: lower 16 bits of "color"
- 24 bits color mode	: lower 24 bits of "color"

This command is for MB86293 or later.

6.11 Texture Pattern Control Commands

6.11.1 GdcTextureMemoryMode [Sets texture memory mode]

Format int GdcTextureMemoryMode (GDC_UCHAR mode)

Parameter mode Texture memory read

GDC_TEX_MEM_MODE_EXT	Read from the graphics memory
GDC_TEX_MEM_MODE_INT	Read from internal texture memory

Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

- GDC_ERR_INVALID_PARAMETER(Invalid parameter is specified)

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

This command can be used by all “Graphics Controller”.

6.11.2 GdcTextureLoadInt [Loads texture/tile pattern to internal texture memory]

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

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

- GDC_ERR_DATA_TOO_BIG (Too large data)

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

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

This command can be used by all “Graphics Controller”.

6.11.3 GdcTextureLoadExt [Loads texture pattern to the 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

Error code
- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)
- GDC_ERR_DATA_TOO_BIG (Too large data)

Description

Copies texture pattern to the graphics memory.

Prior to this command execution, size of the texture pattern should be set by the GdcTextureDimension command.

If length is longer than 65536, transfer will not complete successfully.

This command can be used by all “Graphics Controller”.

6.11.4 GdcTextureLoadExt8 [Loads 8bpp texture pattern to the graphics memory]

Format	int GdcTextureLoadExt8 (GDC_LPCOL8 lpTexture, GDC_ULONG adrs)	
Parameter	lpTexture	Pointer to refer texture pattern
	adrs	Offset address of the memory texture pattern is stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code
- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)
- GDC_ERR_DATA_TOO_BIG (Too large data)

Description

Copies 8bpp texture pattern to the graphics memory.

An offset from the graphics memory top must be set to “adrs”.

Prior to this command execution, size of the texture pattern should be set by the GdcTextureDimension command.

This command is for MB86293 or later.

6.11.5 GdcTextureLoadExt16 [Loads 16bpp texture pattern to the graphics memory]

Format	int GdcTextureLoadExt16 (GDC_LPCOL16 lpTexture, GDC_ULONG adrs)	
Parameter	lpTexture	Pointer to refer texture pattern
	adrs	Offset address of the memory texture pattern is stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)	
	- GDC_ERR_DATA_TOO_BIG (Too large data)	
Description	Copies 16bpp texture pattern to the graphics memory. An offset from the graphics memory top must be set to “adrs”. Prior to this command execution, size of the texture pattern should be set by the GdcTextureDimension command. This command is for MB86293 or later.	

6.11.6 GdcTextureLoadExt24 [Loads 24bpp texture pattern to the graphics memory]

Format	int GdcTextureLoadExt24 (GDC_LPCOL24 lpTexture, GDC_ULONG adrs)	
Parameter	lpTexture	Pointer to refer texture pattern
	adrs	Offset address of the memory texture pattern is stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)	
	- GDC_ERR_DATA_TOO_BIG (Too large data)	
Description	Copies 24bpp texture pattern to the graphics memory. An offset from the graphics memory top must be set to “adrs”. Prior to this command execution, size of the texture pattern should be set by the GdcTextureDimension command. This command is for MB86293 or later.	

6.11.7 GdcTextureLoadInt16Fast [Loads texture pattern to internal texture memory]

Format	int GdcTextureLoadInt16Fast (GDC_LPCOL16 lpTexture, GDC_ULONG adrs)	
Parameter	lpTexture	Pointer to refer texture pattern
	adrs	Offset address of the memory texture pattern is stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)	
	- GDC_ERR_DATA_TOO_BIG (Too large data)	
Description	Copies 16bpp texture pattern to internal texture memory. An offset from the internal texture memory top must be set to “adrs”. Prior to this command execution, size of the texture pattern should be set by the GdcTextureDimension command. This command is for MB86293 or later.	

6.11.8 GdcTextureLoadExt16Fast [Loads texture pattern to the graphics memory]

Format	int GdcTextureLoadExt16Fast (GDC_LPCOL16 lpTexture, GDC_ULONG adrs)	
Parameter	lpTexture	Pointer to refer texture pattern
	adrs	Offset address of the memory texture pattern is stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)	
	- GDC_ERR_DATA_TOO_BIG (Too large data)	
Description	Copies 16bpp texture pattern to internal texture memory. An offset from the internal texture memory top must be set to “adrs”. Prior to this command execution, size of the texture pattern should be set by the GdcTextureDimension command. This command is for MB86293 or later.	

6.11.9 GdcTextureDimension [Sets texture / tile information]

Format int GdcTextureDimension (GDC_ULONG adrs, GDC_ULONG adrs,
GDC_ULONG tw, GDC_ULONG th)

Parameter	adrs	Start address of texture/tile pattern
	oadrs	Offset
	tw	Pattern data width (power of 2)
	th	Pattern data height (power of 2)
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

- GDC_ERR_ILLEGAL_DIMENSION (Illegal vertical/horizontal size of pattern data)

Description Sets following texture information .

- Start address of texture/tile pattern
- Offset
- Pattern data width
- Pattern data height

“adrs” and “oadrs” must be specified with suitable value for the stored position of the referred pattern according to the table 6.11.9a.

Table 6.11.9a Values to be specified to “adrs” and “oadrs”

Stored position of referred pattern	adrs	oadrs
Internal texture	0	Offset from top address of internal texture memory
Extend texture	Start address of texture pattern from top address of the graphics memory	0
Internal tile	0	Offset from top address of internal texture memory
Extend tile	Base address of stored area of tile pattern	Offset from base address of stored area of tile pattern

[Note]

Available destination address of texture/tile pattern is changed according to the “Graphics Controller”.

This command can be used by all “Graphics Controller”.

Range of pattern data width and pattern data height according to the “Graphics Controller” are shown in the table 6.11.9b.

Table 6.11.9b Range of pattern data width and pattern data height

	MB86290A/291/292	MB86293 or later
Internal texture memory	16,32,64	16,32,64
The graphics memory	16,32,64,128,256	16-4096 (power of 2)

This command can be used by all “Graphics Controller”.

However, external tile function is only for MB86293 or later with which internal texture memory is not equipped.

6.11.10 GdcBltTexture [Loads Blt texture to internal texture memory 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 drawing frame
	sstride	Stride (memory size of horizontal span) of the source drawing 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 (pixel unit)
	h	Vertical height of the Blt field (pixel unit)
	oadrs	Offset address of destination memory where texture pattern to be stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code
- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)
- GDC_ERR_ILLEGAL_DIMENSION (Illegal vertical/horizontal size of pattern data)

Description
Loads texture pattern from the graphics memory to the internal texture memory.
This command is only for MB86290A.
When the “Graphics Controller” is MB86291, GdcGeoBltTexture must be used.

6.11.11 GdcGeoBltTexture [Loads Blt texture to internal texture memory for MB86291 or later]

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 drawing frame
	sstride	Stride (memory size of horizontal span) of the source drawing 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 (pixel unit)
	h	Vertical height of the Blt field (pixel unit)
	oadrs	Offset address of destination memory where texture pattern to be stored
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)
- GDC_ERR_ILLEGAL_DIMENSION (Illegal vertical/horizontal size of pattern data)

Description Loads texture pattern from the graphics memory to the internal texture memory.
This command is for MB86291 or later.
When the “Graphics Controller” is MB86290A, GdcBltTexture must be used.

6.12 Binary Pattern Drawing Commands

6.12.1 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 drawing frame when the top left point of binary pattern is drawn
	y	y coordinates of the drawing frame when the top left point of binary pattern is drawn
	w	Binary pattern data width (pixel unit)
	h	Binary pattern data height (pixel unit)
	lpPattern	Pointer to binary pattern data
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)
- GDC_ERR_DATA_TOO_BIG (Too large data)

Description Draws a binary pattern. Foreground color (pixel of binary pattern “1”) is drawn in the color applied by the GdcColor command, and background color (pixel of binary pattern “0”) is drawn in the color applied the GdcBackColor command.

This command can be used by all “Graphics Controller”.

6.12.2 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

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Sets enlarge/shrink mode for binary pattern drawing.

This command can be used by all “Graphics Controller”.

6.13 BLT Commands

6.13.1 GdcBltCopy [Copies BitBlt pattern in current drawing 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 (pixel unit)
	h	Vertical height of the Blt field (pixel unit)
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code
- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)
- GDC_ERR_ILLEGAL_DIMENSION (Illegal vertical/horizontal size of pattern data)

Description
Draws bitmap pattern to the drawing frame by Blt copy.

The destination field is current drawing frame in the graphics memory.

This command can be used by all “Graphics Controller”.

6.13.2 GdcBltCopyAlt, GdcBltCopyAltSync [Copies BitBlt pattern between any drawing 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 (pixel unit)
	h	Vertical height of the Blt field (pixel unit)
	sadr	Memory address of the base point of the source drawing frame
	sstride	Stride (memory size of horizontal span) of the source drawing frame
	dadr	Memory address of the base point of the destination drawing frame
	dstride	Stride (memory size of horizontal span) of the destination drawing frame
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code

- GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)
- GDC_ERR_ILLEGAL_DIMENSION (Illegal vertical/horizontal size of pattern data)

Description

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

This command can be used by all “Graphics Controller”.

6.13.3 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 (pixel unit)
	h	Vertical height of the Blt field (pixel unit)
	lpRect	Pointer to refer the pattern data
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code - GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)
- GDC_ERR_DATA_TOO_BIG (Too large data)

Description Draws bitmap pattern to the drawing 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 drawing frame.

The data format of lpRect is to be the same as that of current drawing 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 Controller”.

6.13.4 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 (pixel unit)
	h	Vertical height of the Blt field (pixel unit)
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Fills a Blt field with the foreground color or tile pattern specified by the GdcColor command.

This command can be used by all “Graphics Controller”.

6.13.5 GdcBltColorTransparent [Sets transparent color of transparent BitBlt]

Format	int GdcBltColorTransparent (GDC_COLOR32 color)	
Parameter	color	Color code treated as transparent color
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete
Error code	GDC_ERR_NOT_READY ("Graphics Driver" is not initialized)	

Description

Sets the transparent color referred when the Blt pattern is copied/drawn in the transparent mode.

The following values are used by color mode.

- 8 bits color mode : lower 8 bits of "color"
- 16 bits color mode : lower 16 bits of "color"
- 24 bits color mode : lower 24 bits of "color"

This command is for MB86291 or later.

6.13.6 GdcBltCopyAltAlpha [Copies BitBlt pattern with alpha blending]

Format int GdcBltCopyAltAlpha(GDC_USHORT x0, GDC_USHORT y0,

```
                              GDC_USHORT x1, GDC_USHORT y1,
                              GDC_USHORT bx, GDC_USHORT by,
                              GDC_USHORT w, GDC_USHORT h,
                              GDC_ULONG sadr, GDC_ULONG sstride,
                              GDC_ULONG bstride)
```

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
	bx	x coordinates of the top left vertex of alpha map area
	by	y coordinates of the top left vertex of alpha map area
	w	Horizontal width of the Blt field (pixel unit)
	h	Vertical height of the Blt field (pixel unit)
	sadr	Memory address of the base point of the source drawing frame
	sstride	Stride (memory size of horizontal span) of the source drawing frame
	bstride	Stride of the alpha map area (pixel unit)
Return value	GDC_TRUE	Complete
	GDC_FALSE	Incomplete

Error code GDC_ERR_NOT_READY (“Graphics Driver” is not initialized)

Description Execute alpha blending with source Blt area and destination area.

Copies alpha blended area to the x1,y1 coordinates of current drawing frame.

Source Blt area is specified with sadrs,sstride,x0,y0,w,h and destination Blt area is specified with x1,y1.

Alpha blend coefficient area is specified with bstride, bx, by, w, h and it’s top address is specified by the GdcAlphaMapBase command.

This command is for MB86293 or later.

6.14 Video Capture Commands

Please use a subsequent video capture command (GdcCap*) after initializing a digital video decoder with an I2C control command (GdcI2C*), when using the digital video decoder connected with an I2C interface.

For details information, refer to the “Application Note”.

6.14.1 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 details information about VCM register, refer to the “Graphics Controller” hardware specifications of using.

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

Sets each mode by combining in the table 6.14.1 macros.

Table 6.14.1 Video capture mode control macro list

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 for MB86291 or later.

6.14.2 GdcCapGetErrorStatus [Gets error status of video capture]

Format GDC_ULONG GdcCapGetErrorStatus(void)

Parameter None

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

31	- - -	4	3	2	1	0
	- - -	x	x	x	x	x

bit 0-4: Error status Yes = except 00000, No = all 0

All other bits:unsetled

Figure 6.14.2 Error status of video capture

Description Reads VCS (Video Capture Status) register and returns error status.

This command is for MB86291 or later.

6.14.3 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 for MB86291 or later.

6.14.4 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 the graphics memory
	eaddr	Specifies the end address +1 of the video capture buffer by offset value from the 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 for MB86291 or later.

6.14.5 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
	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 for MB86291 or later.

6.14.6 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 (default)
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 (default)
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 the GdcDispDimension command beforehand. Color mode supports only 16 bits mode.

This command is for MB86291 or later.

6.14.7 GdcCapSetVideoCaptureScale [Sets scale of video capture]

Format void GdcCapSetVideoCaptureScale (GDC_FIXED_SCALE hscale,
GDC_FIXED_SCALE vscale)

Parameter hscale Horizontal scaling rate (default = 0x0800)

vscale Vertical scaling rate (default = 0x0800)

Return value None

Description Sets scales for enlarging and reducing video capture.

Sets horizontal scaling rate (hscale) which is cast to “GDC_FIXED_SCALE” type.

Sets vertical scaling rate(vscale) which is cast to “GDC_FIXED_SCALE” type.

This scaling rate is mode by dividing number of horizontal pixels of captured image by number of horizontal pixels of image to be displayed.

The style of GDC_FIXED_SCALE is decimal point of fixation of 5 bits of integer part and 11 bits of decimal part.

The range of value is as follows.

--MB86291/86292:0xffff(1/31.99951171875magnifications)
-0x0800(equivalent magnifications).

--MB86293 or later:0xffff(1/31.99951171875magnifications)
-0x0400(2 times magnifications).

Initial value of hscale and vscale is 0x0800(equivalent magnifications) respectively.

Examples of calculation of scaling rate is shown below.

[Example of calculation of scaling rate]

(1)When the image of size 720*576 is reduced to the size 648*490, scaling rate is calculated as below.

--Reduction of horizontal direction

720pixel to 648pixel

720/648=1.111

1.111*2048=2275 (expressed in hexadecimal is 0x08e3)

--Reduction of vertical direction

576line to 490line

576/490=1.176

1.176*2048=2275 (expressed in hexadecimal is 0x0968)

--A value to be set to “hscale” is “0x08e3”.

--A value to be set to “vscale” is “0x0968”.

(2)When 1/n times, scaling rate is calculated as below.

--hscale = n * 2048

--vscale = n * 2048

(3)When n times, scaling rate is calculated as below.

--hscale = 1/n * 2048

--vscale = 1/n * 2048

This command is for MB86291 or later.

However enlarge function is available only with the “Graphics Controller” which is MB86293 or later and supports video capture function.

6.14.8 GdcCapSetAttrMisc [Sets attribute of video capture]

Format	int GdcCapSetAttrMisc (GDC_ULONG target, GDC_ULONG param)	
Parameter	target	Video capture attribute GDC_CAP_ODD_MODE Odd number mode GDC_CAP_CNV_MODE Non-interlace conversion mode
	param	Parameter corresponding to target (*1)
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 for MB86291 or later.	

(*1) Video capture attribute (target) and parameter (param) corresponding to each video capture attributes are shown below.

[Explanatory notes]

Video capture attribute	Description of video capture attributes
Parameter 1 that can set	Description of parameter 1
Parameter 2 that can set	Description of parameter 2
:	:

GCD_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.
GCD_CAP_CNV_MODE	Specifies the non-interlace conversion mode of the picture which is captured.
GDC_CAP_CNV_BOB_MODE	BOB mode (*2).
GDC_CAP_CNV_WEAVE_MODE	WEAVE mode (*3).

(*2)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.

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

6.14.9 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 for MB86291 or later.

6.14.10 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 for MB86291 or later.

6.14.11 GdcCapSetLPFMode [Sets low pass filter mode]

Format void GdcCapSetLPFMode (GDC_ULONG vlpf_y, GDC_ULONG vlpf_c,
GDC_ULONG hlpf_y, GDC_ULONG hlpf_c)

Parameter	vlpf_y	The vertical LPF coefficient code for luminosity signals (0-2)
	vlpf_c	The vertical LPF coefficient code for different color signals (0-2)
	hlpf_y	The horizontal LPF coefficient code for luminosity signals (0-3)
	hlpf_c	The horizontal LPF coefficient code for different color signals (0-3)

Return value None

Description The mode (formula) of the low pass filter at the time of a video capture is set up.

One of the following of coefficients can be specified in vlpf_y of an argument, and vlpf_c. When a coefficient is 0, it outputs as it is, without covering a low pass filter.

The initial value of vlpf_y and vlpf_c are 0.

Table 6.14.11a Low pass filter formula (for vlpf_y and vlpf_c)

Coefficient	Low pass filter formula
0	y[i,j] = x[i,j]
1	y[i,j] = x[i,j-1]/4 + x[i,j]/2 + x[i,j+1]/4
2	y[i,j] = x[i,j-1]*3/16 + x[i,j]*5/8 + x[i,j+1]*3/16

where

x[i,j] : input pixel value at i-th column and j-th raster

y[i,j] : output pixel value at i-th column and j-th raster

One of the following of coefficients can be specified in hlpf_y of an argument, and hlpf_c. When a coefficient is 0, it outputs as it is, without covering a low pass filter.

The initial value of hlpf_y and hlpf_c are 0.

Table 6.14.11b Low pass filter formula (for hlpf_y and hlpf_c)

Coefficient	Low pass filter formula
0	y[i,j] = x[i,j]
1	y[i,j] = x[i-1,j]/4 + x[i,j]/2 + x[i+1,j]/4
2	y[i,j] = x[i-1,j]*3/16 + x[i,j]*5/8 + x[i+1,j]*3/16
3	y[i,j] = x[i-2,j]*3/32 + x[i-1,j]/4 + x[i,j]*5/16 + x[i+1,j]/4 + x[i+2,j]*3/32

where

x[i,j] : input pixel value at i-th column and j-th raster

y[i,j] : output pixel value at i-th column and j-th raster

This command is for MB86291 or later.

6.15 I^{2}C Control Commands

6.15.1 GdcI2CGetBusStatus [Gets I^{2}C bus status]

Format GDC_ULONG GdcI2CGetBusStatus (void)

Parameter None

Return value I^{2}C bus status in the following format (value of BSR register):

bit 31	---	7	6	5	4	3	2	1	0
		x	x	x	x	x	x	x	x

bit7:Detects START/STOP condition
0:STOP condition
1:START condition (The bus is in use)
bit6:Detects repeated START condition
0:Repeated START condition was not detected
1:START condition was detected again while bus is in use
bit5:Indicates Arbitration lost
0:Arbitration lost was not detected
1:Arbitration lost occurred during master transmission
bit4:Status of Acknowledge
0:No acknowledge
1:Acknowledge
bit3:Status of data transfer
0:Receiving status
1:Transmitting status
bit2:Detects addressing
0:Addressing was not performed in a slave mode
1:Addressing was performed in a slave mode
bit1:Detects “General call address (00h)”
0:”General call address(00h)” was not received in a slave mode
1:”General call address(00h)” was received in a slave mode
bit0:Detects the first byte
0:Received data is not the 1st byte
1:Receiverd data is the 1st byte (address data)

Description Reads BSR (Bus Status Register) and returns I^{2}C bus status.

This command is for MB86291S or MB86292S.

6.15.2 GdcI2CSetBusControl [Controls I^2C bus]

Format void GdcI2CSetBusControl (GDC_UCHAR param)

Return value param The parameter for I^2C bus control (the following format)

bit 7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x

bit7:Flag bit for request of bus error interruption
  0:Clears a request flag of bus error interruption
  1:Don’t care
bit6:Permits bus error interruption
  0:Prohibition of bus error interruption
  1:Permission of bus error interruption
bit5:Generates START condition
  0:Don’t care
  1:START condition is generated again at the time of master transmission
bit4:Selects master/slave mode
  0:Becomes a slave mode after generation of STOP condition and completing transfer
  1:Becomes a master mode, generates START condition and starts transfer
bit3:Permits generation of acknowledge at the time of data reception
  0:Acknowledge is not generated
  1:Acknowledge is generated
bit2: Permits generation of acknowledge at the time of “General call address(00h)” reception
  0: Acknowledge is not generated
  1: Acknowledge is generated
bit1:Permits interruption
  0:Prohibition of interrupt
  1:Permission of interrupt
bit0:Flag bit for request of interruption for transfer end
  0:Clears the flag
  1:Don’t care

Return value None

Description Controls I^2C bus by value of parameter for I^2C bus control.

This command is for MB86291S or MB86292S.

6.15.3 GdcI2CGetBusControlStatus [Gets I^{2}C bus control status]

Format GDC_ULONG GdcI2CGetBusControlStatus (void)

Parameter None

Return value I^{2}C bus control status in the following format (value of BCR register):

bit 7	6	5	4	3	2	1	0
x	x		x	x	x	x	x

bit7: Flag bit for request of bus error interruption
0:A bus error was not detected
1:Invalid START condition or STOP condition was detected while data transfer
bit6: Permission of bus error interruption
0:Prohibition of bus error interruption
1:Permission of bus error interruption
bit4: Master / slave mode
0:Slave
1:Master
bit3: Permission of generating acknowledge at the time of data reception
0:Acknowledge is not generated
1:Acknowledge is generated
bit2: Permission of generating acknowledge at the time of “General call address (00h)” reception
0:Acknowledge is not generated
1:Acknowledge is generated
bit1: Permission of interruption
0:Prohibition of interrupt
1:Permission of interrupt
bit0: Flag bit for request of interruption for transfer end
0:The transfer is not ended
1:It is set when 1 byte transfer including the acknowledge bit is completed and it corresponds to the following conditions
- It is a bus master
- It is an addressed slave
- It received “General call address (00h)”
- It was going to generate START condition while other systems by which arbitration lost happened used the bus

Description Reads BCR (Bus Control Register) and returns I^{2}C bus control status.

This command is for MB86291S or MB86292S.

6.15.4 GdcI2CSetClock [Sets I^{2}C clock]

Format void GdcI2CSetClock (GDC_UCHAR param)

Parameter param Parameter for setup of I^{2}C clock (the following format)

bit 7	6	5	4	3	2	1	0
	x	x	x	x	x	x	x

bit6:Selects standard-mode / high-speed-mode
0:Standard-mode
1:High-speed-mode
bit5:Permits I^{2}C operation
0:Prohibition of operation
1:Permission of operation
bit4-0:Frequency of a transfer clock
00000-11111

Table 6.15.4 I^{2}C clock parameter format

For details information about I^{2}C clock, refer to the chapter of the “clock control register” of “I^{2}C Interface specification”.

Return value None

Description Writes parameter to CCR (Clock Control Register) and sets transfer clock.

This command is for MB86291S or MB86292S.

6.15.5 GdcI2CSetData [Sets transfer data]

Format void GdcI2CSetData (GDC_UCHAR param)

Parameter param Value of transfer data

Return value None

Description Writes value of transfer data to DAR (DAta Register).

This command is for MB86291S or MB86292S.

7 System dependent Commands

This section describes which the "Graphics Controller" a system dependence command can be used with it. It is necessary for a graphics application developer to make these.

7.1 System Dependent Commands

System dependent commands list is shown in the table 7.1.

Table 7.1 System dependent commands list

No.	Command name	Function
1	GdcSetDisplayListBuffer	Sets Display List buffer
2	GdcFlushDisplayList	Transfers a Display List
3	GdcGetHostRegisterAddress	Gets host interface register area address
4	GdcGetDispRegisterAddress	Gets display control register area address
5	GdcGetDrawRegisterAddress	Gets drawing control register area address

8 System dependent Commands Interface

This section describes calling conventions interface and processing contents of a system dependent command.

8.1 Explanatory notes

Each item of a driver command reference seems to become following.

Format	Prototype declaration of a command.
Parameter	Description of a parameter.
Return value	The return value and the description.
Called by	The command name of a calling origin.
Description	Description of the command.

8.2 Command Interface

8.2.1 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 Effective values are 1 or 2
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 (1 block) or 2 (2 blocks) is applied as a number of blocks.

[1 block configuration]

In 1 block configuration, the “Graphics Driver” uses the entire Display List buffer area as 1 block. In 1 block configuration, 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.

[2 blocks configuration]

In 2 blocks configuration, the entire Display List buffer area is divided into 2 equal blocks and these 2 blocks are used exclusively. When 1 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 2 blocks configuration, by using DMA function or master function of the “Graphics Controller”, 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, sets GDC FALSE to the return value. In this case, the 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 Controller”.

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 */
    /* A malloc function is a library function of ANSI regulation.
    In this driver, since it does not provide, prepares according to the environment
    of use. */
    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 */
}
```

8.2.2 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 GdcFlush, GdcVFlush command

All drawing 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 the GdcSetDMAMode command (32byte or 4byte). If the 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, refer to the description [Display List transfer procedure] as follows:

- DMA transfer
- Transfer of Local Display List
- CPU transfer

This command can be used by all “Graphics Controller”.

[Display List transfer procedure]

* DMA transfer

This is a method of Display List transfers utilizing the DMA controller of the host CPU (the “Graphics Controller” 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 the “Graphics Controller”.

(1) Checks DMA transfer enable/disable

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

(2) Sets 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 “Graphics Controller”).
- Transfer count (the value specified in “count”).

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

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

(4) Starts DMA transaction

- Appropriate start up operation for the applied DMA controller.

(5) Issues the DMA request

- Sets 1 to DRQ (DMA ReQuest) register.

(6) Waits for the completion of the DMA transfer

- 1 block configuration Display List buffer 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 */
/* Please set a suitable value to the following ########
   according to the environment of use*/
#define HOSTBASE 0x########

/* Start address of drawing control register field */
/* Please set a suitable value to the following ########
   according to the environment of use*/
#define DRAWBASE            0x########

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

#ifdef GDC_MB86290A
#define FIFO_ADDRESS        (DRAWBASE+0x4a0) /* for MB86290A */
#else
#define FIFO_ADDRESS        (DRAWBASE+0x8400) /* for MB86291 or later */
#endif

void GdcFlushDisplayList(GDC_ULONG *src, GDC_ULONG count){
    /* Polling for DMA ready DMA */
    /* Please create DMA_BUSY function according to the environment of use */
    while( DMA_BUSY() );

    /* Sets transfer count */
    /* Please create SET_DMA_COUNT function according to the environment of use */
    SET_DMA_COUNT(CHANNEL0, count);

    /* Sets source address */
    /* Please create SET_DMA_SRC function according to the environment of use */
    SET_DMA_SRC(CHANNEL0, src);

    /* Sets destination address */
    /* Please create SET_DMA_DEST function according to the environment of use */
    SET_DMA_DEST(CHANNEL0, FIFO_ADDRESS);

    /* Sets transfer count (Graphics Controller) */
    WRITE_DTC(count);

    /* Trigger of DMA transaction */
    /* Please create DMA_START function according to the environment of use */
    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
}
```

*Transfer of Local Display List

This is a method of the Display List transfers utilizing the bus master function of the “Graphics Controller”. Transfer count is 4byte unit. In this case, the Display List buffer must be located to the graphics memory of the “Graphics Controller”. And the source address “src” must be converted to the local address of the “Graphics Controller”. The operation procedure of this case is shown as follows:

(1) Checks transfer enable/disable
- Checks the status of LSTA (display List transfer STAtus) register and wait until it will be 0.

(2) Sets source address
- Sets 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) Sets transfer count
- Sets the transfer count (“count” value) to LCO (display List COunt) register.

(4) Starts the transaction
- Sets 1 to LREQ (display List transfer REQuest) register.

(5) Waits for the completion of the transfer (in case of single DL buffer mode)
- Same as (1).

[Example]

```
/* Start address of host interface register field */
/* Please set a suitable value to the following ########
   according to the environment of use*/
#define HOSTBASE            0x########

/* Start address of graphics memory field */
/* Please set a suitable value to the following ########
   according to the environment of use*/
#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) Acquires the Display List FIFO status

- Calls the GdcGetFIFOStatus command and acquire the Display List FIFO status information.

(2) Checks the Display List FIFO status

- Checks 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) Transfers 4byte of data from the source address to the Display List FIFO

(4) Posts increment (+4) source address

[Example]

```
/* Start address of drawing control register field */
/* Please set a suitable value to the following ########
   according to the environment of use*/
#define DRAWBASE            0x########
#define FIFO_FULL           0x2

#ifdef GDC_MB86290A
/* for MB86290A */
#define WRITE_FIFO(i)       ( *(volatile GDC_ULONG*)(DRAWBASE+0x4a0) = (i) )
#else
/* for MB86291 */
#define WRITE_FIFO(i)       ( *(volatile GDC_ULONG*)(DRAWBASE+0x8400) = (i) )
#endif

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);

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

}
```

8.2.3 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 Controller”.

8.2.4 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 Controller”.

8.2.5 GdcGetDrawRegisterAddress [Gets drawing 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 Controller”.