To our customers,

Old Company Name in Catalogs and Other Documents

On April 1st, 2010, NEC Electronics Corporation merged with Renesas Technology Corporation, and Renesas Electronics Corporation took over all the business of both companies. Therefore, although the old company name remains in this document, it is a valid Renesas Electronics document. We appreciate your understanding.

Renesas Electronics website: http://www.renesas.com

April 1st, 2010
Renesas Electronics Corporation

Issued by: Renesas Electronics Corporation (http://www.renesas.com)

Send any inquiries to http://www.renesas.com/inquiry.

Notice

1. All information included in this document is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas Electronics products listed herein, please confirm the latest product information with a Renesas Electronics sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas Electronics such as that disclosed through our website.
2. Renesas Electronics does not assume any liability for infringement of patents, copyrights, or other intellectual property rights of third parties by or arising from the use of Renesas Electronics products or technical information described in this document. No license, express, implied or otherwise, is granted hereby under any patents, copyrights or other intellectual property rights of Renesas Electronics or others.
3. You should not alter, modify, copy, or otherwise misappropriate any Renesas Electronics product, whether in whole or in part.
4. Descriptions of circuits, software and other related information in this document are provided only to illustrate the operation of semiconductor products and application examples. You are fully responsible for the incorporation of these circuits, software, and information in the design of your equipment. Renesas Electronics assumes no responsibility for any losses incurred by you or third parties arising from the use of these circuits, software, or information.
5. When exporting the products or technology described in this document, you should comply with the applicable export control laws and regulations and follow the procedures required by such laws and regulations. You should not use Renesas Electronics products or the technology described in this document for any purpose relating to military applications or use by the military, including but not limited to the development of weapons of mass destruction. Renesas Electronics products and technology may not be used for or incorporated into any products or systems whose manufacture, use, or sale is prohibited under any applicable domestic or foreign laws or regulations.
6. Renesas Electronics has used reasonable care in preparing the information included in this document, but Renesas Electronics does not warrant that such information is error free. Renesas Electronics assumes no liability whatsoever for any damages incurred by you resulting from errors in or omissions from the information included herein.
7. Renesas Electronics products are classified according to the following three quality grades: "Standard", "High Quality", and "Specific". The recommended applications for each Renesas Electronics product depends on the product's quality grade, as indicated below. You must check the quality grade of each Renesas Electronics product before using it in a particular application. You may not use any Renesas Electronics product for any application categorized as "Specific" without the prior written consent of Renesas Electronics. Further, you may not use any Renesas Electronics product for any application for which it is not intended without the prior written consent of Renesas Electronics. Renesas Electronics shall not be in any way liable for any damages or losses incurred by you or third parties arising from the use of any Renesas Electronics product for an application categorized as "Specific" or for which the product is not intended where you have failed to obtain the prior written consent of Renesas Electronics. The quality grade of each Renesas Electronics product is "Standard" unless otherwise expressly specified in a Renesas Electronics data sheets or data books, etc.

   "Standard": Computers; office equipment; communications equipment; test and measurement equipment; audio and visual equipment; home electronic appliances; machine tools; personal electronic equipment; and industrial robots.

   "High Quality": Transportation equipment (automobiles, trains, ships, etc.); traffic control systems; anti-disaster systems; anti-crime systems; safety equipment; and medical equipment not specifically designed for life support.

   "Specific": Aircraft; aerospace equipment; submersible repeaters; nuclear reactor control systems; medical equipment or systems for life support (e.g. artificial life support devices or systems), surgical implantations, or healthcare intervention (e.g. excision, etc.), and any other applications or purposes that pose a direct threat to human life.
8. You should use the Renesas Electronics products described in this document within the range specified by Renesas Electronics, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas Electronics shall have no liability for malfunctions or damages arising out of the use of Renesas Electronics products beyond such specified ranges.
9. Although Renesas Electronics endeavors to improve the quality and reliability of its products, semiconductor products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Further, Renesas Electronics products are not subject to radiation resistance design. Please be sure to implement safety measures to guard them against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas Electronics product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other appropriate measures. Because the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
10. Please contact a Renesas Electronics sales office for details as to environmental matters such as the environmental compatibility of each Renesas Electronics product. Please use Renesas Electronics products in compliance with all applicable laws and regulations that regulate the inclusion or use of controlled substances, including without limitation, the EU RoHS Directive. Renesas Electronics assumes no liability for damages or losses occurring as a result of your noncompliance with applicable laws and regulations.
11. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written consent of Renesas Electronics.
12. Please contact a Renesas Electronics sales office if you have any questions regarding the information contained in this document or Renesas Electronics products, or if you have any other inquiries.

(Note 1) "Renesas Electronics" as used in this document means Renesas Electronics Corporation and also includes its majority-owned subsidiaries.

(Note 2) "Renesas Electronics product(s)" means any product developed or manufactured by or for Renesas Electronics.

SH7764 Group

Example of Rendering Using 2-D Graphics Engine

Introduction

This application note presents a sample program that uses the 2-D graphics engine (G2D) of the SH7764 to perform rendering.

Target Device

SH7764 (Renesas Technology R0K507764E001BR)

Contents

1. Introduction

1.1 Specifications

The 2-D graphics engine is used to draw lines and rectangles, and perform BitBlt operations, to a frame buffer located in external memory (SDRAM).

1.2 Functions Used

2-D graphics engine (G2D)

1.3 Applicable Conditions

• MCU	SH7764
• Operating frequency	CPU clock: 324 MHz SuperHyway clock: 108 MHz Peripheral clock: 54 MHz Bus clock: 108 MHz
• Integrated development environment	Renesas Technology
• C compiler	Renesas Technology SuperH RISC Engine Family C/C++ Compiler Package, Ver. 9.03, Release 00
• Compile options	High-performance Embedded Workshop default settings -cpu=sh4a -endian=little -include="$(WORKSPDIR)\inc" -object="$(CONFIGDIR)\$(FILELEAF).obj" -debug -optimize=0 -gbr=auto -chgincpath -errorpath -global_volatile=0 -opt_range=all -infinite_loop=0 -del_vacant_loop=0 -struct_alloc=1 –nologo

1.4 Related Application Notes

Application notes related to this application note are listed below. Please refer to them in conjunction with this application note.

- SH7764 Group: *SH7764 Example of Initial Setting*
- SH7764 Group: *Video Display Controller TFT-LCD Interfacing Example 1*
- SH7764 Group: *LCD Controller TFT-LCD Interfacing Example*

2. Description of Sample Program

The sample program provides an example of the use of the G2D to perform rendering.

2.1 G2D Operation

2.1.1 Features

- On-chip geometry engine for coordinate transformation
  Hardware that performs coordinate transformation (4 × 4 matrix operation + Z clipping + perspective W division) for the input vertex
- Extended 2-D functions
  High-functionality bold line drawing, antialias line drawing, and BitBlt type commands with ROP/alpha blending
- Enhanced control command functions
  Two command systems: GOSUB/RET and INT command, and enhanced WPR and TRAP command functions
- Backward compatibility with Q2SD on the functional level

2.1.2 Coordinate Systems

The G2D supports four types of two-dimensional coordinates (screen coordinates, rendering coordinates, two-dimensional source coordinates, and work coordinates), and one type of one-dimensional coordinates (one-dimensional source coordinates).

Screen coordinates are display control coordinates. The screen coordinate positive directions are right for the X-axis and down for the Y-axis. Either 16 bits (16 bits/pixel) or 8 bits (8 bits/pixel) can be selected as the screen coordinate data width. The maximum screen coordinate values are X = 4095 and Y = 4095.

Rendering coordinates are drawing control coordinates. Rendering coordinates are shifted horizontally and vertically relative to screen coordinates by the offset amounts specified in drawing commands. Drawing commands use these coordinates to perform rendering operations. Either 16 bits (16 bits/pixel) or 8 bits (8 bits/pixel) can be selected as the data width of each rendering coordinate.

Two-dimensional source coordinates are drawing control coordinates. When a drawing command is executed with SS = 1, these are the source coordinates specified by the drawing command. Either 16 bits (16 bits/pixel) or 8 bits (8 bits/pixel) can be selected as the data width of each two-dimensional source coordinate.

Figure 1 illustrates screen coordinates and figure 2 rendering coordinates.

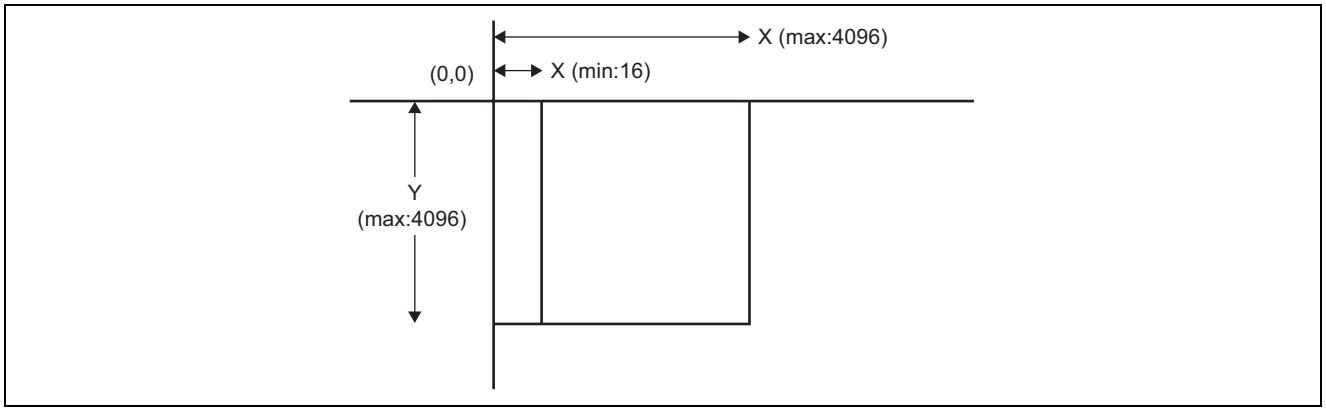

Figure 1 Screen Coordinates

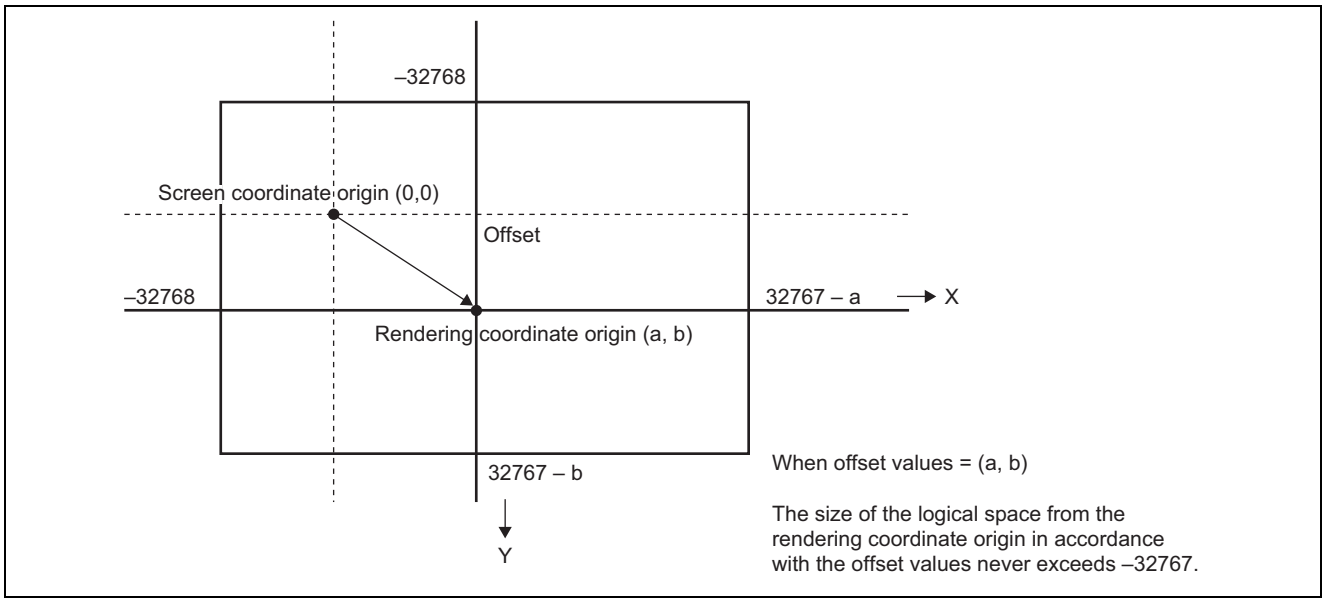

Figure 2 Rendering Coordinates

2.1.3 Data Formats

Figure 3 illustrates the data formats.

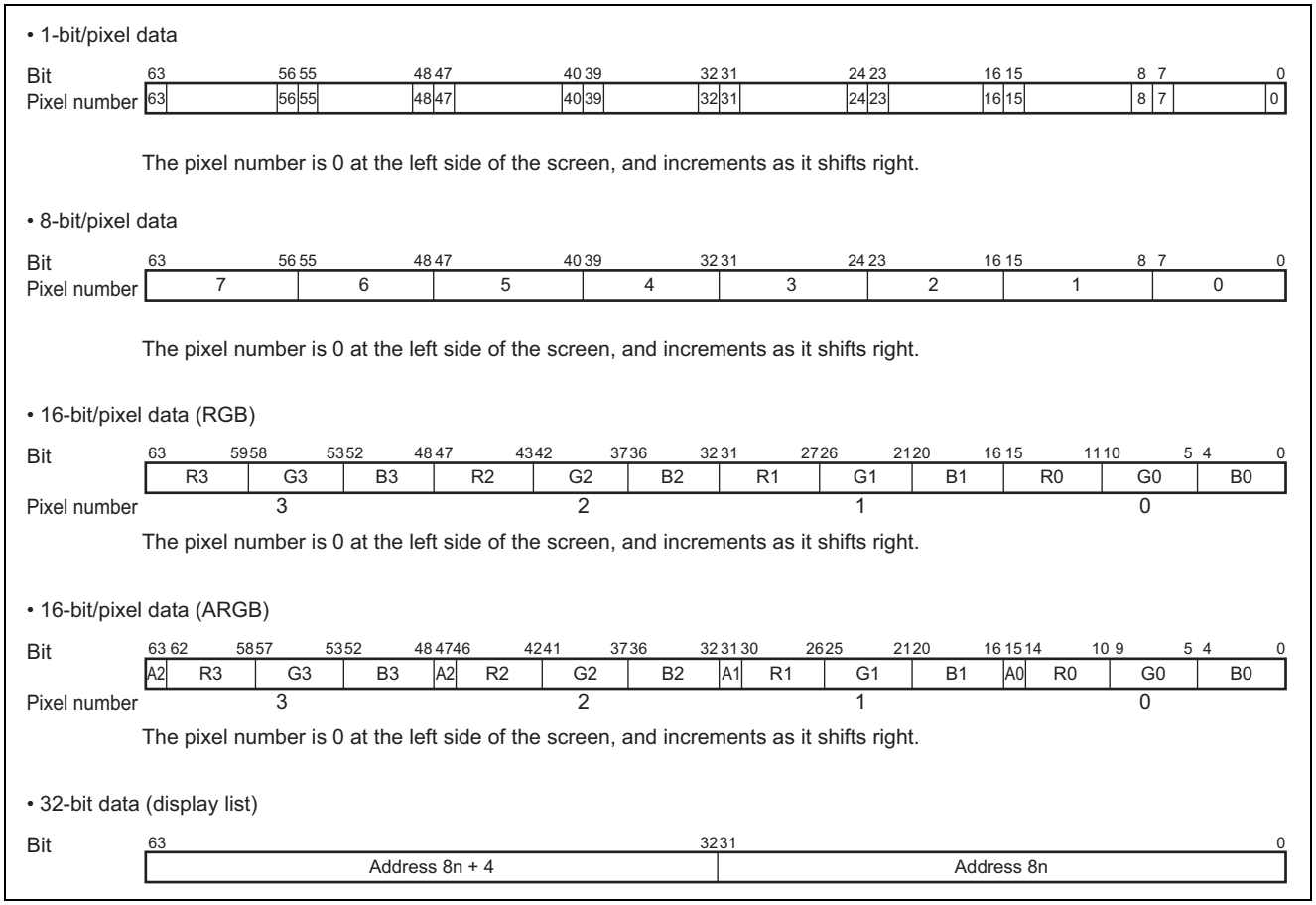

Figure 3 Data Formats

2.1.4 List of Commands and Rendering Attributes

Figure 4 is a list of commands and rendering attributes. It is followed by overview descriptions of the various rendering attributes. For a detailed description of rendering attributes, see *SH7764 Group Hardware Manual*.

	OP CODE								Draw Mode							
									b15	b14	b13	b12	b11	b10	b9	b8
Command	b31	b30	b29	b28	b27	b26	b25	b24	MTRE	reserve	CLIP	RCLIP	STRANS	DTRANS/ LINKE	WORK/ LREL	SS
POLYGON4A	1	0	0	0	0	0	1	0	MTRE		CLIP	RCLIP	STRANS		WORK	SS
POLYGON4B					0	0	0	1	MTRE		CLIP	RCLIP	STRANS		WORK	SS
POLYGON4C					0	0	0	0	MTRE		CLIP	RCLIP			WORK	
LINEA	1	0	1	1	0	0	1	0	MTRE		CLIP	RCLIP	STRANS			SS (0)
LINEB					0	0	0	1	MTRE		CLIP	RCLIP	STRANS			SS (0)
LINEC					0	0	0	0	MTRE		CLIP	RCLIP		LINKE	LREL	
LINED					0	0	1	1	MTRE		CLIP	RCLIP		LINKE	LREL	
RLINEA					0	1	1	0	MTRE		CLIP	RCLIP	STRANS			SS (0)
RLINEB					0	1	0	1	MTRE		CLIP	RCLIP	STRANS			SS (0)
RLINEC					0	1	0	0	MTRE		CLIP	RCLIP		LINKE	LREL	
RLINED					0	1	1	1	MTRE		CLIP	RCLIP		LINKE	LREL	
FTRAPC	1	1	0	1	0	0	0	0	MTRE		CLIP	RCLIP		LINKE	LREL	
RFTRAPC					0	1	0	0	MTRE		CLIP	RCLIP		LINKE	LREL	
CLRWC	1	1	1	0	0	0	0	0	MTRE		CLIP	RCLIP				
LINEWC	1	1	1	1	0	0	0	0	MTRE		CLIP	RCLIP				
RLINEWC					0	1	0	0	MTRE		CLIP	RCLIP				
BITBLTA	1	0	1	0	0	0	1	0	MTRE		CLIP	RCLIP	STRANS	DTRANS	WORK	SS
BITBLTB					0	0	0	1	MTRE		CLIP	RCLIP	STRANS	DTRANS	WORK	SS
BITBLTC					0	0	0	0	MTRE		CLIP	RCLIP		DTRANS	WORK	
Test mode	1	0	1	0	1	0	0	0	This setting is for internal verification only and should not be used. This setting does not cause the command error (CER) flag to be set.							

Figure 4 Rendering Attributes (1)

Command	OP CODE								Draw Mode							
									b7	b6	b5	b4	b3	b2	b1	b0
	b31	b30	b29	b28	b27	b26	b25	b24	REL	STYLE/ SRCDIRX	BLKE/ SRCDIRY	NET/EDG/ DSTDIRX	EOS/ DSTDIRY	COOF	AA/ αE	CLKW/ SαE
POLYGON4A	1	0	0	0	0	0	1	0	REL	STYLE	BLKE	NET	EOS	COOF	αE	SαE
POLYGON4B					0	0	0	1	REL	STYLE	BLKE	NET	EOS	COOF	αE	
POLYGON4C					0	0	0	0			BLKE	NET	EOS	COOF	αE	
LINEA	1	0	1	1	0	0	1	0	REL	STYLE (1)		NET	EOS	COOF	AA	
LINEB					0	0	0	1	REL	STYLE (1)		NET	EOS	COOF	AA	
LINEC					0	0	0	0				NET	EOS	COOF	AA	
LINED					0	0	1	1							AA (1)	CLKW
RLINEA					0	1	1	0	REL	STYLE (1)		NET	EOS	COOF	AA	
RLINEB					0	1	0	1	REL	STYLE (1)		NET	EOS	COOF	AA	
RLINEC					0	1	0	0				NET	EOS	COOF	AA	
RLINED					0	1	1	1							AA (1)	CLKW
FTRAPC	1	1	0	1	0	0	0	0			BLKE (1)	EDG	EOS			
RFTRAPC					0	1	0	0			BLKE (1)	EDG	EOS			
CLRWC	1	1	1	0	0	0	0	0			BLKE (1)					
LINEWC	1	1	1	1	0	0	0	0					EOS			
RLINEWC					0	1	0	0					EOS			
BITBLTA	1	0	1	0	0	0	1	0	REL	SRCDIR X	SRCDIR Y	DSTDIR X	DSTDIR Y	COOF	αE	SαE
BITBLTB					0	0	0	1	REL	SRCDIR X	SRCDIR Y	DSTDIR X	DSTDIR Y	COOF	αE	
BITBLTC					0	0	0	0				DSTDIR X	DSTDIR Y	COOF	αE	
Test mode	1	0	1	0	1	0	0	0	This setting is for internal verification only and should not be used. This setting does not cause the command error (CER) flag to be set.							

Figure 4 Rendering Attributes (2)

Command	OP CODE								Draw Mode							
	b31	b30	b29	b28	b27	b26	b25	b24	b15	b14	b13	b12	b11	b10	b9	b8
TRAP	0	0	0	0	0	0	0	0								
NOP/INT	0	0	0	0	1	0	0	0	INT							
VBKEM	0	0	0	1	0	0	0	0								
WPR	0	0	0	1	1	0	0	0						LINKE	LREL	
JUMP	0	0	1	0	1	0	0	0								
GOSUB	0	0	1	1	0	0	0	0								
RET	0	0	1	1	1	0	0	0								
LCOFS	0	1	0	0	0	0	0	0								
RLCOFS	0	1	0	0	0	1	0	0								
MOVE	0	1	0	0	1	0	0	0								
RMOVE	0	1	0	0	1	1	0	0								
Test mode	0	1	0	1	0	0	0	0	This setting is for internal verification only and should not be used. This setting does not cause the command error (CER) flag to be set.							

Command	OP CODE								Draw Mode							
	b31	b30	b29	b28	b27	b26	b25	b24	b7	b6	b5	b4	b3	b2	b1	b0
TRAP*	0	0	0	0	0	0	0	0			Flip5	Flip4	Flip3	Flip2	Flip1	Flip0
NOP/INT	0	0	0	0	1	0	0	0	INT No							
VBKEM	0	0	0	1	0	0	0	0								
WPR	0	0	0	1	1	0	0	0					ByteM3	ByteM2	ByteM1	ByteM0
JUMP	0	0	1	0	1	0	0	0	REL							
GOSUB	0	0	1	1	0	0	0	0	REL							No
RET	0	0	1	1	1	0	0	0								No
LCOFS	0	1	0	0	0	0	0	0								
RLCOFS	0	1	0	0	0	1	0	0								
MOVE	0	1	0	0	1	0	0	0								
RMOVE	0	1	0	0	1	1	0	0								
Test mode	0	1	0	1	0	0	0	0	This setting is for internal verification only and should not be used. This setting does not cause the command error (CER) flag to be set.							

Note: * The TRAP command's rendering attribute field Flip[5:0] in is not supported on the SH7764.

Figure 4 Rendering Attributes (3)

1. Source Transparency Specification (STRANS)
   When referencing source data, the STRANS bit can be used to select transparency or non-transparency on an individual drawing command basis. The source transparency specification can be used with the OLYGON4A, OLYGON4B, LINEA, LINEB, RLINEA, RLINEB, BITBLTA, and BITBLTB commands. The STRANS bit should be cleared to 0 in other commands.

2. Destination Transparency Specification (DTRANS)
   When referencing destination data, the DTRANS bit can be used to select transparency or non-transparency on an individual drawing command basis. The destination transparency specification can be used with the BITBLTA, BITBLTB, and BITBLTC commands. The DTRANS bit should be cleared to 0 in other commands.

3. Source Style Specification (STYLE)
   The STYLE bit can be used to select, on an individual drawing command basis, whether to enlarge or reduce the source data or repeatedly reference it. The source style specification can be used with the POLYGON4A, POLYGON4B, LINEA, LINEB, RLINEA, and RLINEB commands. The STYLE bit should be cleared to 0 in other commands.

4. Clipping Specification (CLIP)
   The G2D can perform clipping area management. There are three types of clipping area: system clipping area, user clipping area, and relative user clipping area.
   The system clipping area has a fixed drawing range. The system clipping area is always enabled, regardless of attribute specifications.
   A user clipping area can be designated as desired within the system clipping area. Whether or not clipping is performed in that area can be selected on an individual drawing command basis with the rendering attribute CLIP bit.
   Clipping is set with screen coordinates. Since the clipping area is undefined after power-on, set the clipping area with the WPR command at the beginning of the display list that is executed first.

5. Relative Clipping Specification (RCLIP)
   A relative user clipping area can be designated as desired within the system clipping area at a relative setting with respect to the local offset. Whether or not clipping is performed in that area can be selected on an individual command basis with the rendering attribute RCLIP bit.
   If both the RCLIP and CLIP bits are set to 1 simultaneously, the region where the two clipping areas overlap is drawn.

6. Net Drawing Specification (NET)
   The NET bit can be used to select, on an individual drawing command basis, whether or not net drawing is to be performed. Net drawing is a function for drawing only pixels at coordinates for which the condition *rendering coordinates* $X + Y = EOS$ (0: even number, 1: odd number) is true. The net drawing specification can be used with the POLYGON4 type, LINEA, LINEB, LINEC, RLINEA, RLINEB, and RLINEC commands. The NET bit should be cleared to 0 in other commands.

7. Even/Odd Select Specification (EOS)
   Even pixels are selected when EOS = 0, and odd pixels when EOS = 1. The even/odd select specification is used together with the net drawing specification (NET).

8. Work Specification (WORK)
   When drawing is performed at rendering coordinates with a POLYGON4 type or BITBLT type command, the WORK bit can be used to select, on an individual drawing command basis, whether or not binary work data is to be referenced.
   When binary work data referencing is selected, drawing is performed if the work data for the pixel corresponding to the rendering coordinates is 1, but not if the work data is 0. The same shape as that drawn at work coordinates can thus be drawn at rendering coordinates. The work specification can be used with POLYGON4 type and BITBLT type commands. The WORK bit should be cleared to 0 in the other commands.

9. Source Address Specification (SS)
   The SS bit is used to select whether the source is to be referenced at a two-dimensional source area address or at the address indicated by the Base Address parameter in the display list. The source address specification can be used with the POLYGON4A, POLYGON4B, BITBLTA, and BITBLTB commands. The SS bit should be cleared to 0 in other commands.

10. Source Coordinate Relative Address Specification (REL)
    Setting the REL bit to 1 in the POLYGON4A, POLYGON4B, BITBLTA, BITBLTB, LINEA, LINEB, RLINEA, RLINEB, JUMP, and GOSUB commands enables source referencing or branching to be performed at an address relative to (before or after) the command code.

11. Edge Drawing (EDG)
    With the FTRAP and RFTRAP commands, setting the EDG bit to 1 enables edge lines to be drawn after completion of trapezoid painting to the work area.

12. Color Offset (COOF)
    The color offset specification can be used with the POLYGON4 type, LINEA, LINEB, LINEC, RLINEA, RLINEB, RLINEC, and BITBLT type commands. In 16-bit/pixel drawing, if the rendering attribute COOF bit is set to 1, the result of adding the value in the COFSR register to the value of the source data (color expanded data for a binary source and the specified color for the monochrome specification) is drawn.

13. Source Direction X, Y (SRCDIRX, SRCDIRY)
    The source direction X and Y specifications can be used with the BITBLTA and BITBLTB commands to select the directions in which to scan the source data.

14. Destination Direction X, Y (DSTDIRX, DSTDIRY)
    The destination direction X and Y specifications can be used with the BITBLTA, BITBLTB, and BITBLTC commands to select the directions in which to draw the destination data.

15. Antialias Enable (AA)
    The antialias enable specification can be used with LINE type and RLINE type commands to reduce aliasing. The antialias enable specification is enabled only in 16-bit/pixel drawing.

16. Alpha Blend Enable (αE)
    The alpha blend enable specification can be used with POLYGON4 type and BITBLT type commands. The source data and ground data are alpha blended and drawn. The alpha blend enable specification is enabled only in 16-bit/pixel drawing. In POLYGON4 type commands, the alpha blend enable specification is enabled only when BLKE = 1. In BITBLT type commands, the alpha blend enable specification is enabled only when the ROP code is H'CC (source copy).

17. Source Alpha Enable (SαE)
    The source alpha enable specification can be used with the POLYGON4A, and BITBLTA commands. It is used together with αE.

18. Block Enable (BLKE)
    The block enable specification can be used with POLYGON4 type commands. When BLKE = 1, the input vertex coordinates (DXn, DYn) are internally transformed to circumscribed rectangle coordinates (DXn', DYn') and drawing is performed. When coordinate transformation is to be performed, the transformed vertices are internally converted into a rectangle and drawn.

19. Coordinate Transformation Enable (MTRE)
    The coordinate transformation enable specification can be used with all drawing commands. Setting the MTRE bit to 1 when the coordinate transformation enable bit (GTE) in the coordinate transformation control register (GTRCR) is set to 1 performs coordinate transformation for the input vertex.

20. Link Specification Enable (LINKE)
    The link specification enable specification can be used with the LINEC, LINED, RLINEC, RLINED, FTRAPC, RFTRAPC, and WPR commands. From the memory address specified by the LINK address, the vertex coordinates are read with the LINEC, LINED, RLINEC, RLINED, FTRAPC, and RFTRAPC commands, and the register write data is read with the WPR command.

21. Link Address Relative Specification (LREL)
    The link address relative specification can be used with the LINEC, LINED, RLINEC, RLINED, FTRAPC, RFTRAPC, and WPR commands.

22. Clockwise (CLKW)
    The clockwise specification can be used with the LINED and RLINED commands. The CLKW bit is used to specify whether the order in giving the N vertices is clockwise or counterclockwise.

23. Raster Operation (ROP)
    The raster operation specification can be used with BITBLT type commands. The ROP code is specified in the ROP field, which is a BITBLT command parameter. When alpha blending is enabled (αE = 1), set the ROP code to H'CC.

2.2 Display List

The SH7764 performs drawing at rendering coordinates and work coordinates based on a collection of drawing commands called a display list. An overview description of the drawing commands used in the sample program is presented below. For a detailed description, see *SH7764 Group Hardware Manual*.

Unless otherwise specified, the parameter notations used in the command descriptions have the meanings indicated below.

- TXS, TYS: Source starting point
- TDX, TDY: Source size
- DXn, DYn: Rendering coordinates (absolute coordinates)
- TXOFS, TYOFS: Source offset
- COLOR: 8 or 16 bits/pixel color specification
- n: Number of vertices
- W: Line width

2.2.1 POLYGON4A

Transfers multi-value (8 or 16 bits/pixel) source data to the user-specified quadrilateral rendering coordinates. When SS = 1, set 8 or more pixels as the TDX value. If TXOFS or TYOFS is set, the source is referenced at a location shifted by the offset amount.

Figure 5 shows a command format example, and figure 6 shows an example of the command's operation.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1000_0010								Reserve (all 0)								Draw Mode															
0	0	0	0	TXS (0 ≤ TXS ≤ 4088)												0	0	0	0	TYS (0 ≤ TYS ≤ 4095)											
0	0	0	0	TDX (0 ≤ TDX ≤ 4095)												0	0	0	0	TDY (0 ≤ TDY ≤ 4095)											
0	0	0	0	TXOFS (0 ≤ TXOFS ≤ TDX – 1)												0	0	0	0	TYOFS (0 ≤ TYOFS ≤ TDY – 1)											
Sign	DX1 (–32768 ≤ DX1 ≤ 32767)															Sign	DY1 (–32768 ≤ DY1 ≤ 32767)														
Sign	DX2 (–32768 ≤ DX2 ≤ 32767)															Sign	DY2 (–32768 ≤ DY2 ≤ 32767)														
Sign	DX3 (–32768 ≤ DX3 ≤ 32767)															Sign	DY3 (–32768 ≤ DY3 ≤ 32767)														
Sign	DX4 (–32768 ≤ DX4 ≤ 32767)															Sign	DY4 (–32768 ≤ DY4 ≤ 32767)														

Figure 5 POLYGON4A SS = 1 Command Format Example

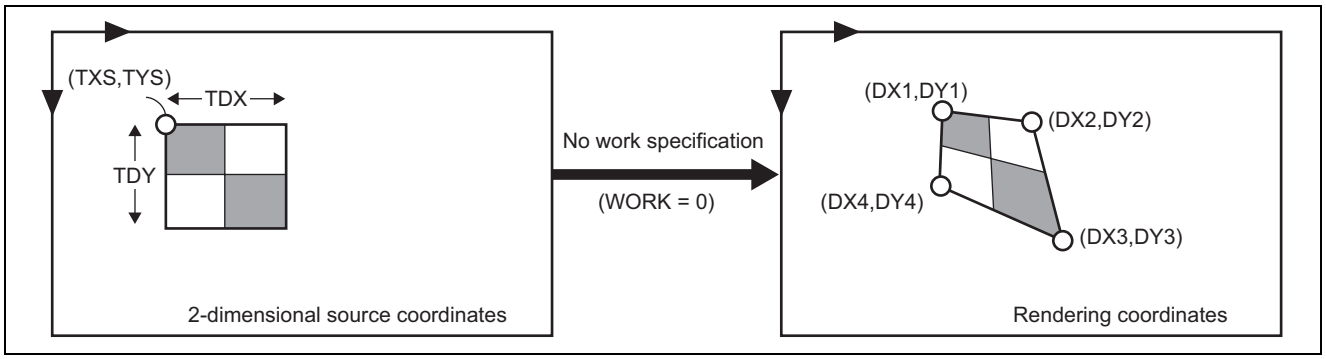

Figure 6 POLYGON4A Operation Example

2.2.2 POLYGON4C

Performs any four-vertex drawing at rendering coordinates with a monochrome specification.

Figure 7 shows a command format example, and figure 8 shows an example of the command's operation.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1000_0000								Reserve (all 0)								Draw Mode															
all 0																Color															
Sign	DX1 (–32768 ≤ DX1 ≤ 32767)															Sign	DY1 (–32768 ≤ DY1 ≤ 32767)														
Sign	DX2 (–32768 ≤ DX2 ≤ 32767)															Sign	DY2 (–32768 ≤ DY2 ≤ 32767)														
Sign	DX3 (–32768 ≤ DX3 ≤ 32767)															Sign	DY3 (–32768 ≤ DY3 ≤ 32767)														
Sign	DX4 (–32768 ≤ DX4 ≤ 32767)															Sign	DY4 (–32768 ≤ DY4 ≤ 32767)														

Figure 7 POLYGON4C Command Format Example

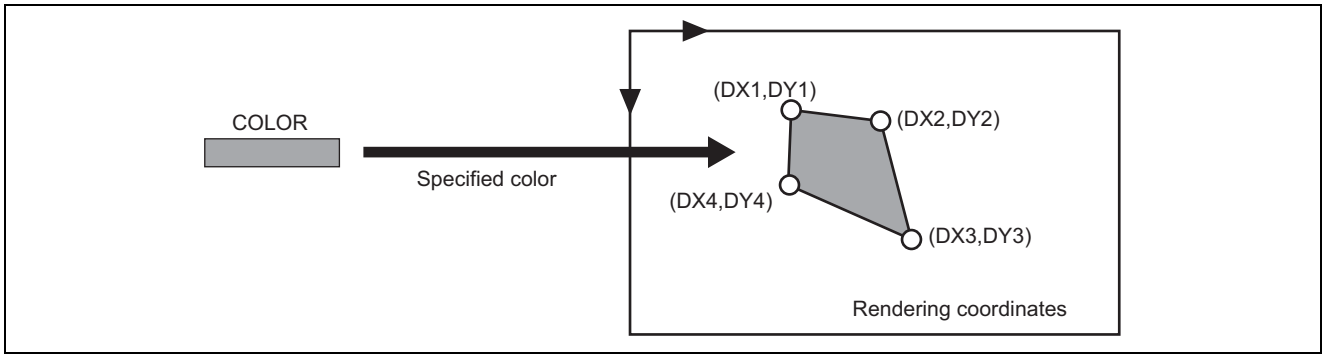

Figure 8 POLYGON4C Operation Example

2.2.3 LINEC

Draws a polygonal line with any width in the destination area with a monochrome specification.

Draws a polygonal line from vertex 1 (DX1, DY1), through vertex 2 (DX2, DY2), …, to vertex n (DXn, DYn). When a value greater than 1 is set in W, a bold line is drawn.

Figure 9 shows a command format example, and figure 10 shows an example of the command's operation.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 1011_0000								Reserve (all 0)								Draw Mode															
Color																n (2 ≤ n ≤ 65535)															
Reserve (all 0)																0	0	0	0	0	0	0	0	0	0	W (0, 2 ≤ W ≤ 63)					
Sign	DX1 (–32768 ≤ DX1 ≤ 32767)															Sign	DY1 (–32768 ≤ DY1 ≤ 32767)														
Sign	•															Sign	•														
Sign	•															Sign	•														
Sign	DXn (–32768 ≤ DXn ≤ 32767)															Sign	DYn (–32768 ≤ DYn ≤ 32767)														

Figure 9 LINEC LINKE = 0 Command Format Example

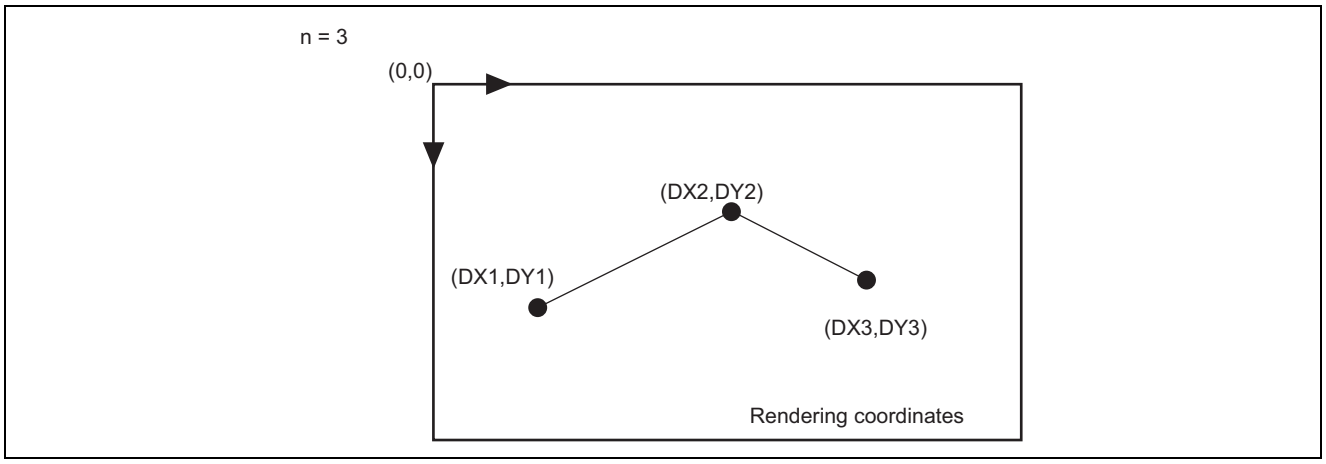

Figure 10 LINEC Operation Example

2.2.4 LCOFS

Sets the offset values (local offset) of the destination area and work area. After the local offset values are set, these offset values are added in all subsequent coordinate specifications made in drawing commands. These settings must be made at the start of the display list (the initial values are undefined).

Figure 11 shows a command format example, and figure 12 shows an example of the command's operation.

- XO: Local offset value, rendering coordinate (absolute coordinate), work coordinate (absolute coordinate)
- YO: Local offset value, rendering coordinate (absolute coordinate), work coordinate (absolute coordinate)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0100_0000								Reserve (all 0)								Draw Mode															
XO (–32768 ≤ XO ≤ 32767)																YO (–32768 ≤ YO ≤ 32767)															

Figure 11 LCOFS Command Format Example

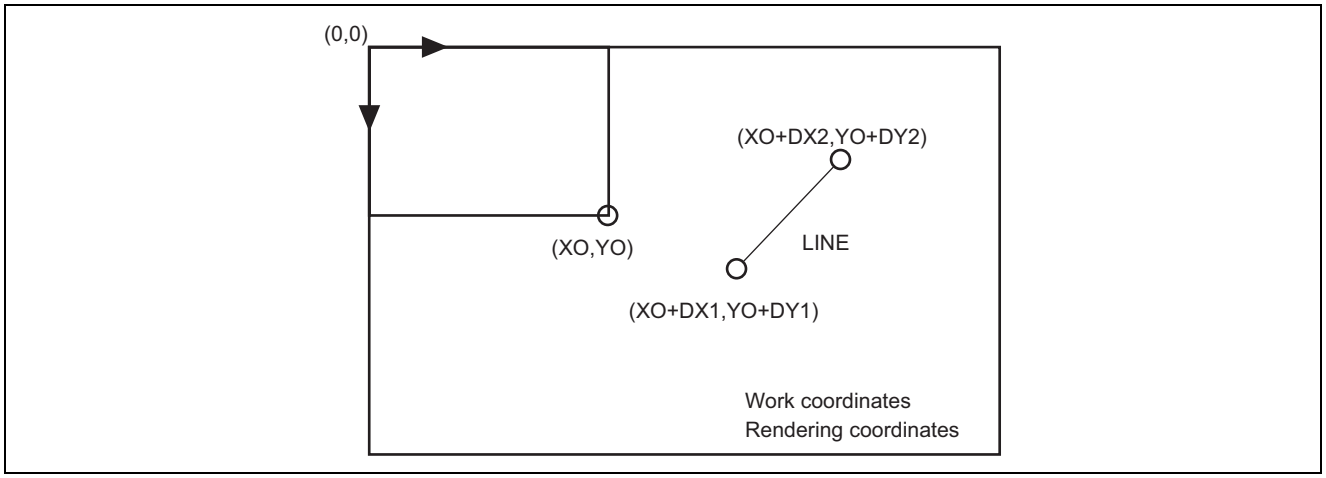

Figure 12 LCOFS Operation Example

2.2.5 WPR

Sets a value in a specific address-mapped register.

Figure 13 shows the command format.

- W reg No: Register number
- n – 1: Write data count
- Data n: Write data

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0001_1000								Reserve (all 0)								Draw Mode															
Reserve (all 0)								n – 1 (0 ≤ n – 1 ≤ 255)								0	0	0	0	0	0	0	W Reg No								
Data 0																															
•																															
•																															
Data n – 1																															

Figure 13 WPR Command Format Example

2.2.6 TRAP

Indicates the end of the display list.

Halts drawing operation and sets the TRA bit in the status register (SR) to 1. If the TRE bit in the interrupt enable register (IER) is set to 1, an interrupt is sent to the CPU. This command must be placed at the end of the display list.

Figure 14 shows a command format example, and figure 15 shows an example of the command's operation.

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OP CODE = 0000_0000								Reserve (all 0)								Draw Mode															

Figure 14 TRAP Command Format Example

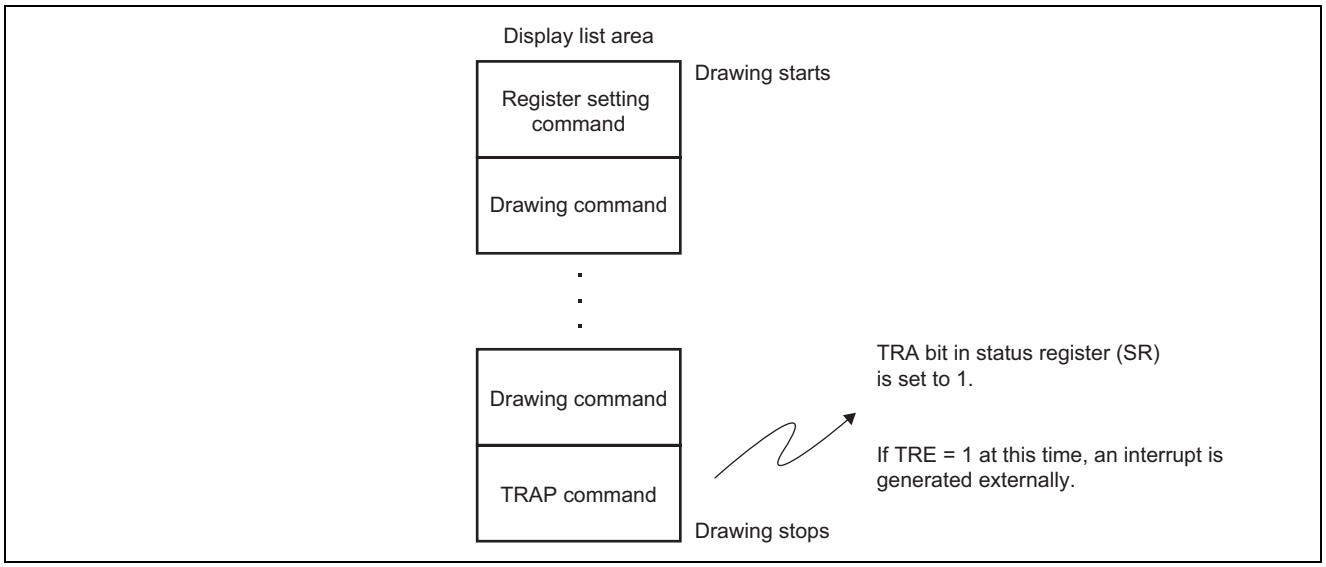

Figure 15 TRAP Operation Example

2.3 Command Fetching

When 1 is written to the rendering start bit (RS) in the system control register (SCLR), the G2D itself fetches the display list from the external memory (SDRAM) and starts drawing. It is therefore necessary to store in the SDRAM the display list and the source data it uses before the start of command fetching is triggered by the RS bit. The G2D starts command fetching from the address specified in the display list start address register (DLSAR) and then continues fetching commands sequentially. The command fetching of the G2D is stopped by placing the TRAP command at the end of the display list. The arrangement of the commands is in accordance with the command format.

Figure 16 shows an example display list.

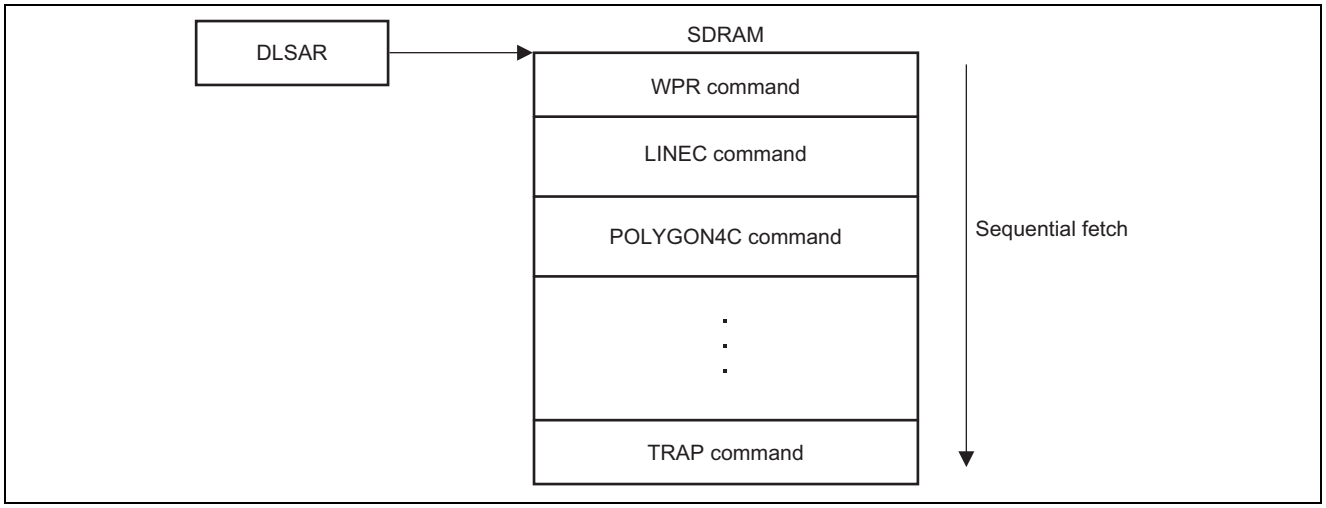

Figure 16 Display List Example

2.4 Specifications of Sample Program

The specifications of the sample program and flowcharts of its operations are presented below.

2.4.1 Specifications

- The G2D is used to draw lines and rectangles, and perform BitBlt operations, to a frame buffer area set to WVGA resolution (H 800 × V 480) and 16-bit pixels.
- The display list uses three planes and the frame buffer two planes.
- Blue and red rectangles, and a white line, are drawn to the second plane of the frame buffer.
  1. The POLYGON4C command is used to draw a blue rectangle that fills the entire screen and then to draw a red rectangle in the center of the screen.
  2. The LINEC command is used to draw a white line from the upper left corner to the bottom right corner of the screen.
- The contents of the second plane of the frame buffer are transferred to the first plane of the frame buffer using a BitBlt operation.
  1. The POLYGON4A command is used to perform a BitBlt operation for the entire screen. (Portions that are not red are set as transparent.)
  2. The POLYGON4A command is used to perform a BitBlt operation for the entire screen.

  Of the image drawn to the second plane of the frame buffer, the first BitBlt operation draws only the red portions to the first plane of the frame buffer. The next BitBlt draws operation draws the entire image.

Note: The sample program performs rendering to the frame buffer area. To show the contents of the frame buffer on a display panel, use the display controller (VDC2) or LCD controller (LCDC) of the SH7764.

Figure 17 shows an overview of the operation of the sample program. Display lists 1 to 3 are executed, causing the images rendered to the first plane of the frame buffer to be transferred by BitBlt operations to the second plane.

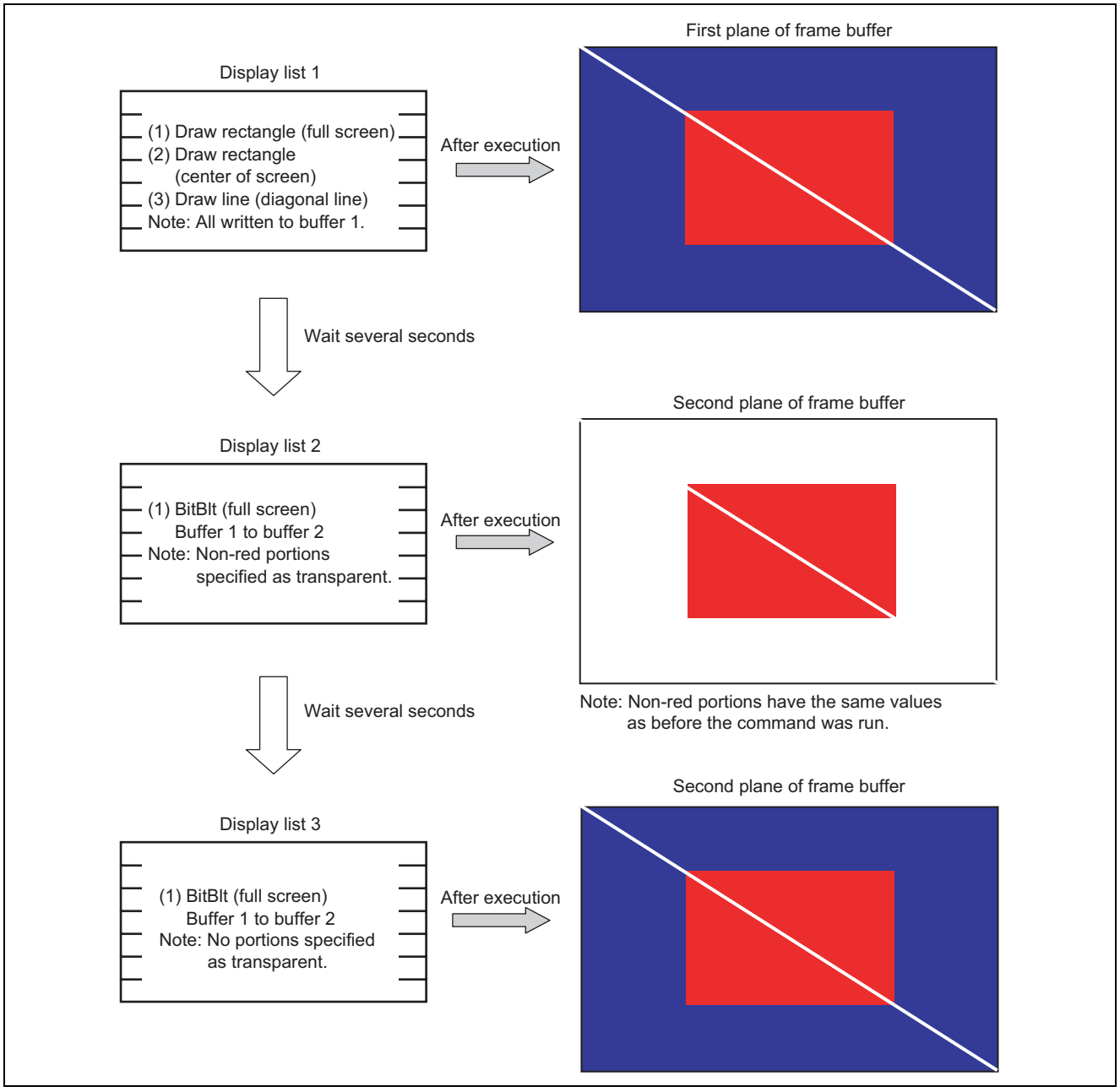

Figure 17 Sample Program Operation Overview

2.4.2 Main Sequence of Sample Program

Figure 18 shows the main sequence of the sample program. The G2D initial settings routine shown in figure 19 is run, then commands are set in display lists 1 to 3. Then the display list start address setting routine shown in figure 20 is run, and rendering to the frame buffer area takes place when the rendering start setting shown in figure 21 is run.

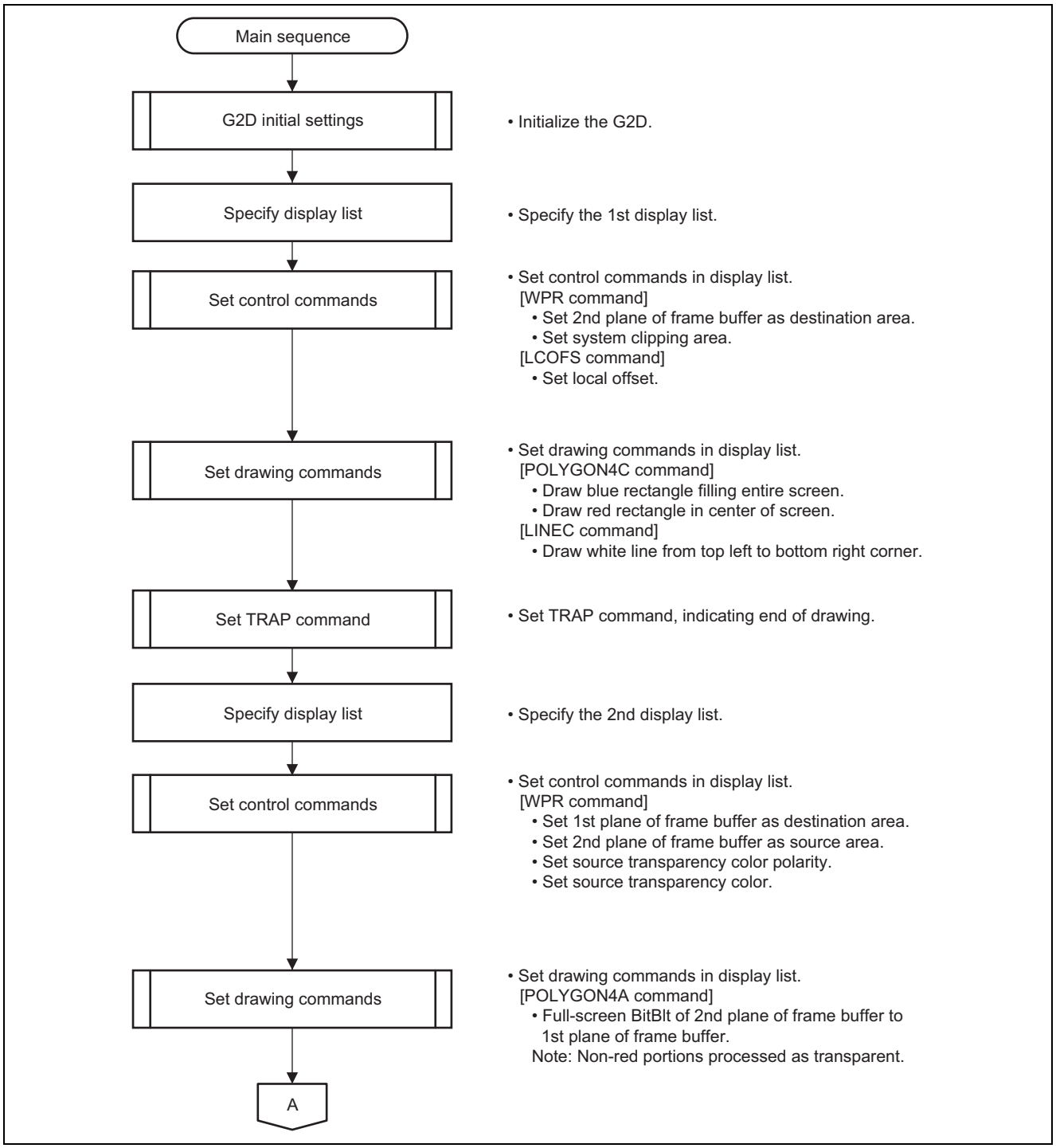

Figure 18 Main Sequence of Sample Program (1)

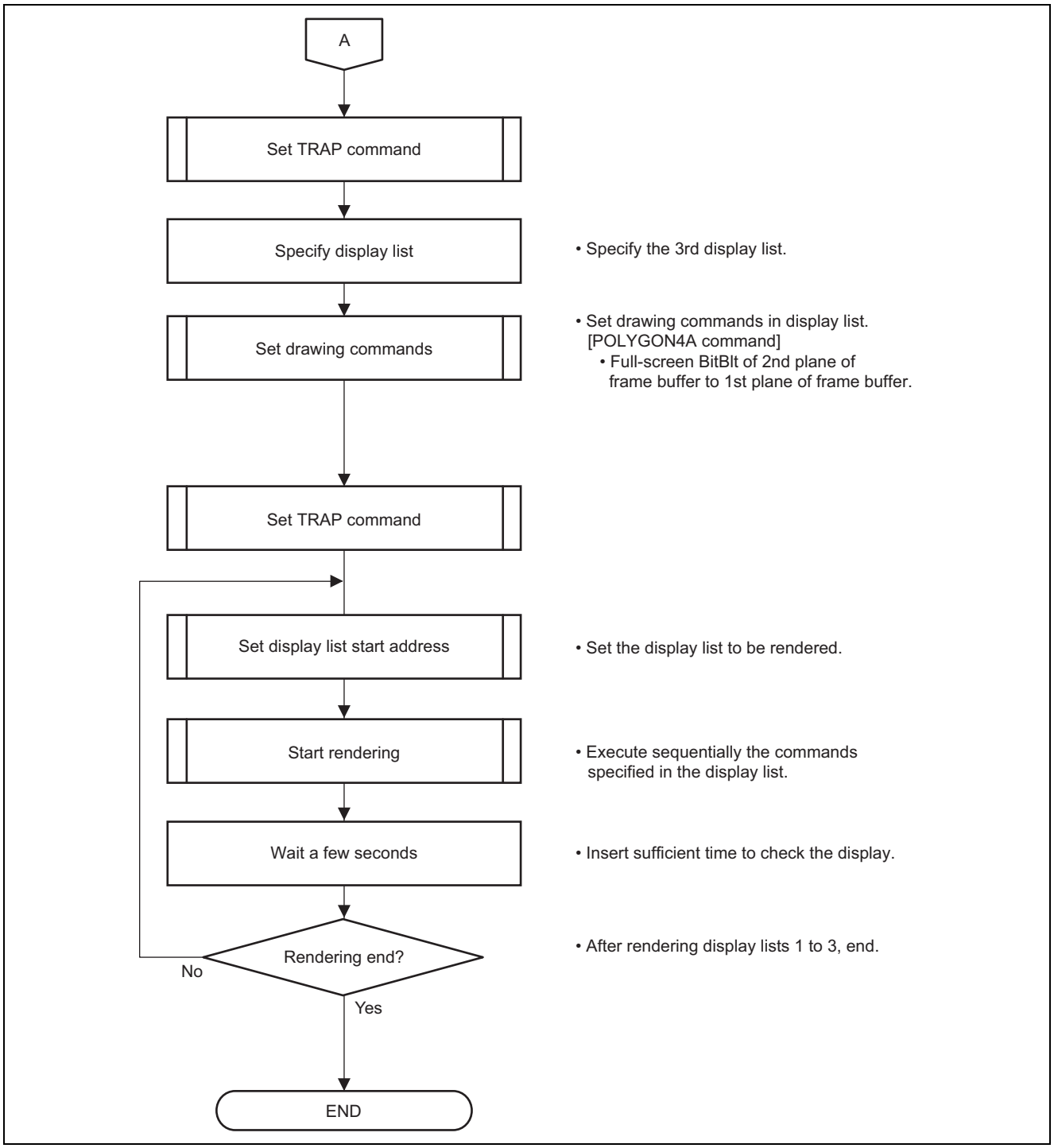

Figure 18 Main Sequence of Sample Program (2)

2.4.3 G2D Initial Settings

Figure 19 shows example G2D initial settings.

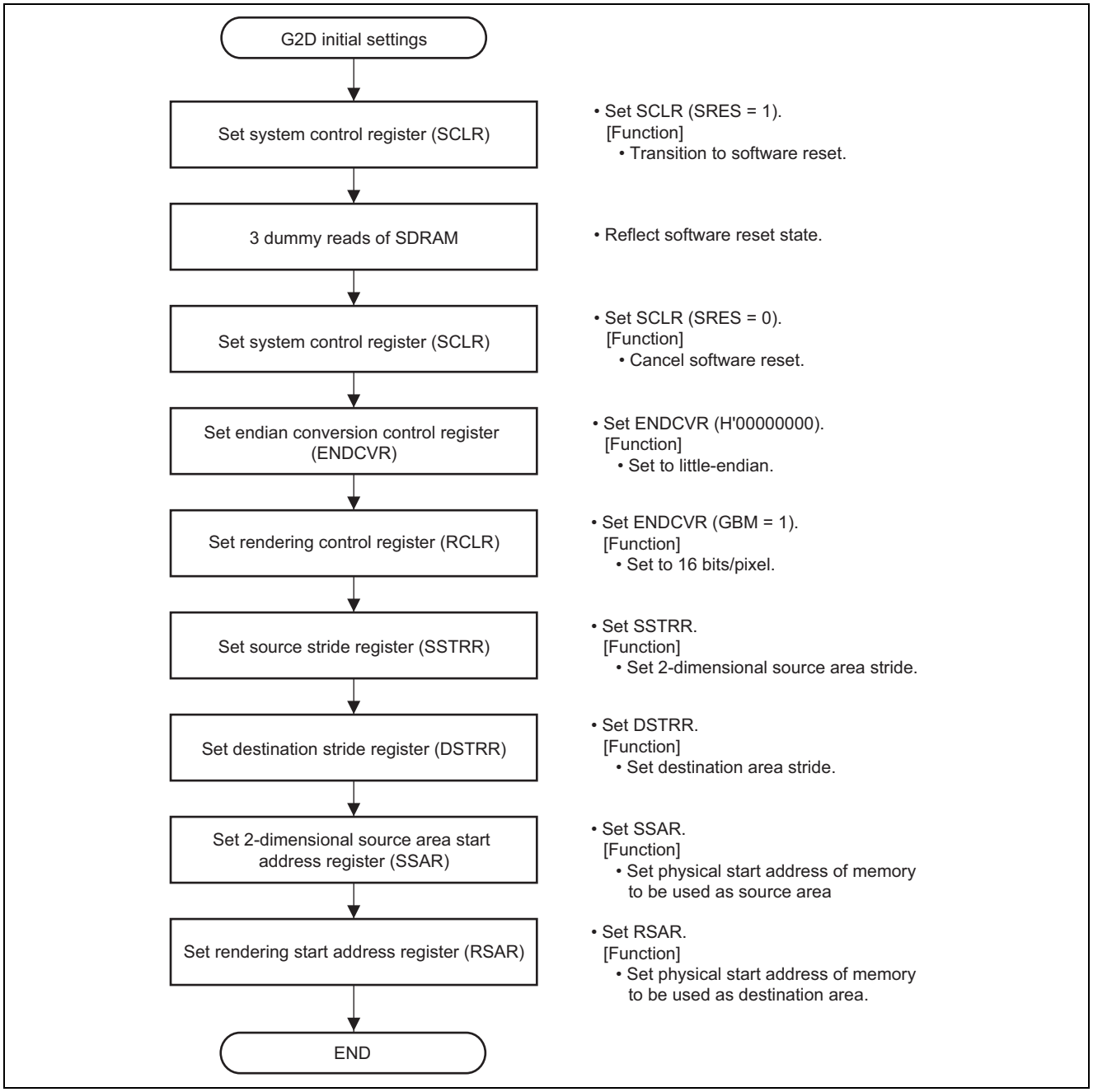

Figure 19 G2D Initial Settings Example

2.4.4 Display List Start Address Setting

Figure 20 shows an example of display list start address setting.

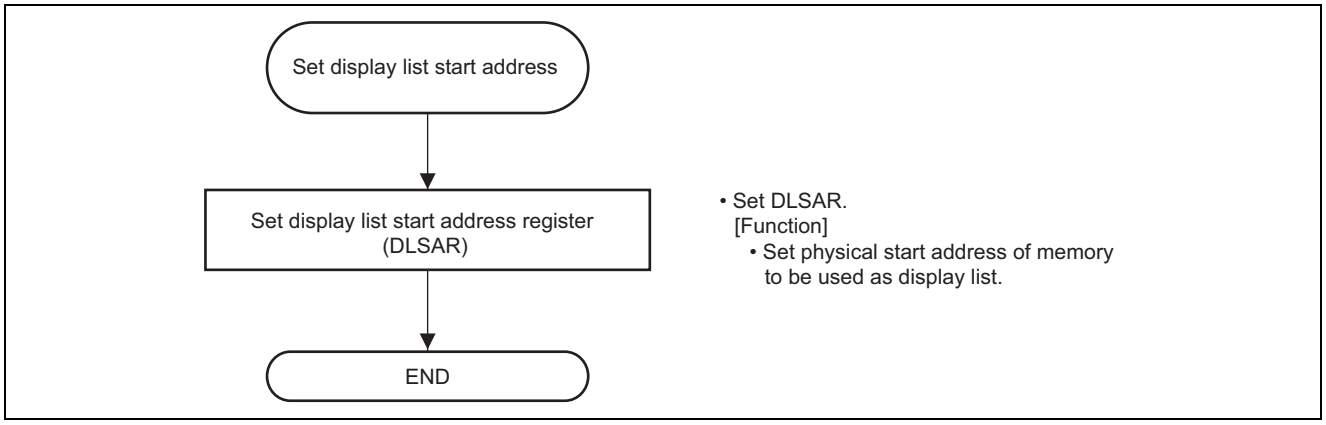

Figure 20 Display List Start Address Setting Example

2.4.5 Rendering Start Setting

Figure 21 shows an example of rendering start setting.

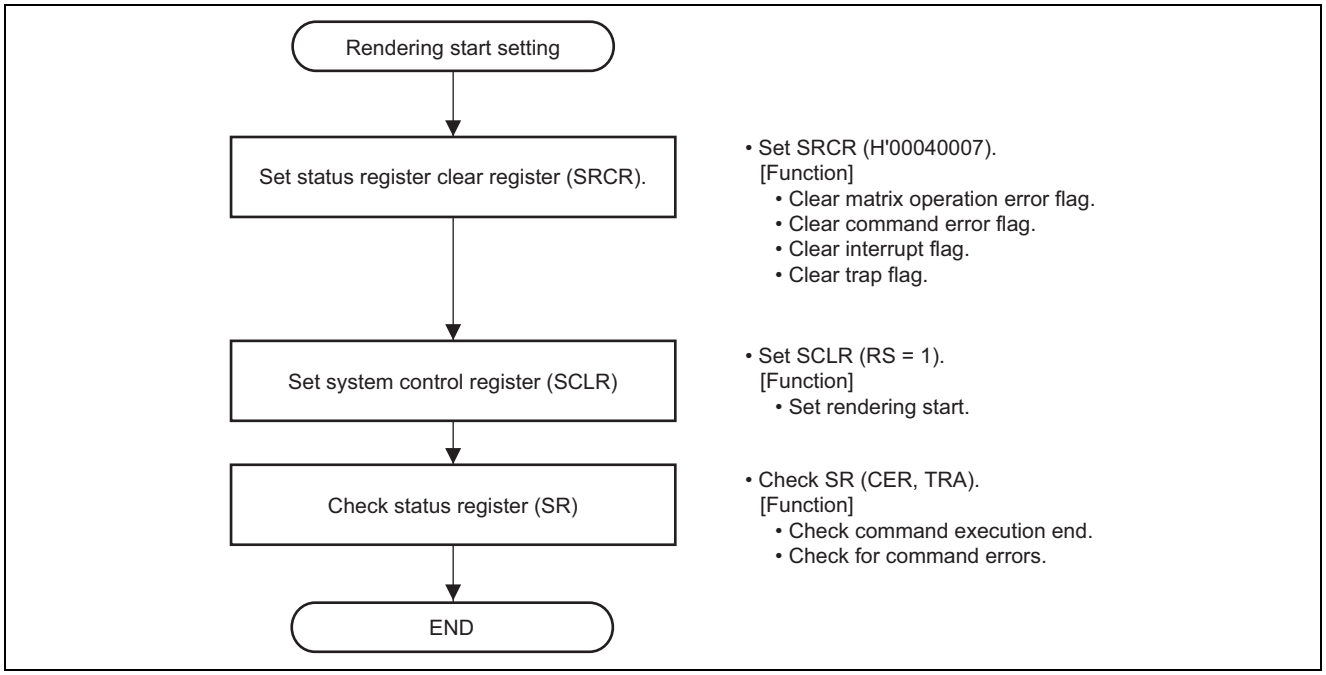

Figure 21 Rendering Start Setting Example

3. Sample Program “g2d.c”

3.1 Sample Program Listing: Macro Definition

```
1.  /*""FILE COMMENT""************ Technical reference data *************************
2.  *
3.  *       System Name : SH7764 Sample Program
4.  *       File Name   : g2d.c
5.  *       Abstract    : G2D usage example
6.  *       Version     : 1.00.00
7.  *       Device      : SH7764
8.  *       Tool-Chain  : High-performance Embedded Workshop (Ver.4.05.01).
9.  *                   : C/C++ compiler package for the SuperH RISC engine family
10. *                   :                               (Ver.9.03 Release00).
11. *       OS          : none
12. *       H/W Platform: R0K507764E001BR
13. *       Disclaimer  :
14. *                     Note
15. *                     The entirety of this sample program is intended for reference,
16. *                     and its operation is not guaranteed. Customers should use this
17. *                     sample program as a technical reference when developing their
18. *                     own software.
19. *
20. *       The information described here may contain technical inaccuracies or
21. *       typographical errors. Renesas Technology Corporation and Renesas Solutions
22. *       assume no responsibility for any damage, liability, or other loss rising
23. *       from these inaccuracies or errors.
24. *
25. *       Copyright (C) 2009 Renesas Technology Corp. All Rights Reserved
26. *       AND Renesas Solutions Corp. All Rights Reserved
27. *
28. *       History     : July.01,2009 Ver.1.00.00
29. *""FILE COMMENT END""*************************************************************/
30. #include "iodefine.h"
31.
32. /* ==== Macro definitions ==== */
33. #define TFT_PANEL_CLOCK     800         /* Number of horizontal pixels */
34. #define TFT_PANEL_LINE      480         /* Number of vertical pixels */
35. #define DISPLIST_BUF_SIZE   64          /* Display list size */
36. #define OFFSET_SCLMAR       0x0D0       /* SCLMAR register offset */
37. #define OFFSET_SSAR         0x04C       /* SSAR register offset */
38. #define OFFSET_RSAR         0x050       /* RSAR register offset */
39. #define OFFSET_STCR         0x080       /* STCR register offset */
40. #define OFFSET_RCLR         0x0C0       /* RCLR register offset */
41. #define RGB565_RED          0xF800      /* RGB565 red data */
42. #define RGB565_BLUE         0x001F      /* RGB565 blue data */
43. #define RGB565_WHITE        0xFFFF      /* RGB565 white data */
```

3.2 Sample Program Listing: Function Prototype Declarations/Variable Definitions

```
/* ==== Function prototype declarations ==== */
void g2d_main(void);
void g2d_initial(void);
void g2d_displist_set(unsigned long addr);
unsigned long * g2d_com_WPR_set(unsigned long * displist, short RegNo,
             unsigned long Data);
unsigned long * g2d_com_LCOFS_set(unsigned long * displist, short x_offset,
             short y_offset);
unsigned long * g2d_com_LINEC_set(unsigned long * displist, short x1,
             short y1, short x2, short y2, short color, short width);
unsigned long * g2d_com_POLYGON4C_set(unsigned long * displist, short x,
             short y, short w, short h, short color);
unsigned long * g2d_com_POLYGON4A_set(unsigned long * displist, short x,
             short y, short w, short h, short trans);
void g2d_com_TRAP_set(unsigned long * displist);
int g2d_start_rendering(void);
void delay(void);

/* ==== Variable definitions ==== */
#pragma section _G2D_FRAME_BUFFER /* Placement at 16-byte boundary of cache disabled space */
unsigned short frame_buffer[2][TFT_PANEL_LINE][TFT_PANEL_CLOCK];
#pragma section

#pragma section _DISPLIST_START     /* Placement in cache disabled space */
unsigned long displist[3][DISPLIST_BUF_SIZE];
#pragma section
```

3.3 Sample Program Listing: Main Process

```
/*""FUNC COMMENT""*****************************************************************
 * ID           :
 * Outline      : G2D main process
 *-----------------------------------------------------------------------------
 * Include      :
 *-----------------------------------------------------------------------------
 * Declaration : void g2d_main(void);
 *-----------------------------------------------------------------------------
 * Function     : Draws lines and rectangles in the frame buffer, then performs BitBlt operations.
 *-----------------------------------------------------------------------------
 * Argument     : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void g2d_main(void)
{
  unsigned long *displist_addr;
  long i;

  /* ==== Initialize G2D module ==== */
  g2d_initial();

  /* ==== Set 1st display list ==== */
  displist_addr = displist[0];

  /* ---- Set control commands ---- */
  displist_addr = g2d_com_WPR_set(displist_addr, OFFSET_SCLMAR,
                 ((TFT_PANEL_CLOCK << 16) | TFT_PANEL_LINE));
      /* Set system clipping area */
      /* Always set the start of the display list with the first command */
  displist_addr = g2d_com_LCOFS_set(displist_addr, 0, 0);
      /* Set local offset */
  displist_addr = g2d_com_WPR_set(displist_addr, OFFSET_RSAR,
                  (unsigned long)frame_buffer[1]);
      /* Set destination area */

  /* ---- Set drawing commands ---- */
  displist_addr = g2d_com_POLYGON4C_set(displist_addr, 0, 0,
                 TFT_PANEL_CLOCK, TFT_PANEL_LINE, RGB565_BLUE);
      /* Draw full-screen blue rectangle */
  displist_addr = g2d_com_POLYGON4C_set(displist_addr, TFT_PANEL_CLOCK / 4,
                   TFT_PANEL_LINE / 4, TFT_PANEL_CLOCK / 2, TFT_PANEL_LINE / 2,
                    RGB565_RED);
      /* Draw red rectangle in center of screen */
  displist_addr = g2d_com_LINEC_set(displist_addr, 0, 0, TFT_PANEL_CLOCK,
                   TFT_PANEL_LINE, RGB565_WHITE, 10);
      /* Draw white line from top left to bottom right corner */

  /* ---- Set TRAP command ---- */
  g2d_com_TRAP_set(displist_addr);

  /* ==== Set 2nd display list 2 ==== */
```

```
122.  displist_addr = displist[1];
123.
124.   /* ---- Set control commands ---- */
125.   displist_addr = g2d_com_WPR_set(displist_addr, OFFSET_RSAR,
126.                   (unsigned long)frame_buffer[0]);
127.      /* Set destination area */
128.  displist_addr = g2d_com_WPR_set(displist_addr, OFFSET_SSAR,
129.                   (unsigned long)frame_buffer[1]);
130.      /* Set source area */
131.  displist_addr = g2d_com_WPR_set(displist_addr, OFFSET_RCLR, 0x02040000);
132.      /* Set source transparency color polarity */
133.  displist_addr = g2d_com_WPR_set(displist_addr, OFFSET_STCR, RGB565_RED);
134.      /* Set source transparency color */
135.
136.  /* ---- Set drawing commands ---- */
137.  displist_addr = g2d_com_POLYGON4A_set(displist_addr, 0, 0,
138.                   TFT_PANEL_CLOCK, TFT_PANEL_LINE, 1);
139.      /* Set BitBlt command in display list (apply transparency to non-red areas) */
140.
141.  /* ---- Set TRAP command ---- */
142.  g2d_com_TRAP_set(displist_addr);
143.
144.  /* ==== Set 3rd display list ==== */
145.  displist_addr = displist[2];
146.
147.  /* ---- Set drawing commands ---- */
148.  displist_addr = g2d_com_POLYGON4A_set(displist_addr, 0, 0,
149.                   TFT_PANEL_CLOCK, TFT_PANEL_LINE, 0);
150.      /* Set BitBlt command in display list */
151.
152.  /* ---- Set TRAP command ---- */
153.   g2d_com_TRAP_set(displist_addr);
154.
155.   /* ==== Start rendering (display lists 1 to 3) ==== */
156.   for(i=0; i<3; i++){
157.      g2d_displist_set((unsigned long)displist[i]);
158.      g2d_start_rendering();
159.      delay();
160.   }
161.  }
```

3.4 Sample Program Listing: G2D Initial Settings

```
/*""FUNC COMMENT""*****************************************************************
 * ID           :
 * Outline      : G2D initialization
 *-----------------------------------------------------------------------------
 * Include      :
 *-----------------------------------------------------------------------------
 * Declaration : void g2d_control_initial(void);
 *-----------------------------------------------------------------------------
 * Function     : Initializes G2D registers.
 *-----------------------------------------------------------------------------
 * Argument     : void
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void g2d_initial(void)
{
    long i;
    unsigned long dummy;

    G2D.SCLR.BIT._SRES = 1;           /* Reset G2D */
    for(i=0;i<3;i++){
        dummy = *(unsigned long*)frame_buffer[0]; /* Reflect reset*/
    }
    G2D.SCLR.BIT._SRES = 0;           /* Cancel reset */

    G2D.ENDCVR.BIT._BITSWAP = 0;      /* Set endian conversion */

    G2D.RCLR.BIT._GBM = 1;            /* Set rendering control (16 bits/pixel) */

    G2D.SSTRR.BIT._SSTRIDE = (TFT_PANEL_CLOCK);      /* Set source stride */

    G2D.DSTRR.BIT._DSTRIDE = (TFT_PANEL_CLOCK);
                                      /* Set destination stride */
    G2D.SSAR = (unsigned long *)((unsigned long)frame_buffer[0] & 0x1FFFFFF0);
                                      /* Set source area */

    G2D.RSAR = (unsigned long *)((unsigned long)frame_buffer[1] & 0x1FFFFFF0);
                                      /* Set destination area */
}
```

3.5 Sample Program Listing: Display List Start Address Setting

```
/*""FUNC COMMENT""*****************************************************************
 * ID           :
 * Outline      : Display list start address setting
 *-----------------------------------------------------------------------------
 * Include      :
 *-----------------------------------------------------------------------------
 * Declaration : void g2d_displist_set(unsigned long addr);
 *-----------------------------------------------------------------------------
 * Function     : Sets display list start address.
 *-----------------------------------------------------------------------------
 * Argument     : unsigned long addr
 *-----------------------------------------------------------------------------
 * Return Value: void
 *""FUNC COMMENT END""*************************************************************/
void g2d_displist_set(unsigned long addr)
{
    G2D.DLSAR = (unsigned long *)(addr & 0x1FFFFFF0);
}
```

3.6 Sample Program Listing: Rendering Start Setting

```
/*""FUNC COMMENT""*****************************************************************
 * ID           :
 * Outline      : Start rendering
 *-----------------------------------------------------------------------------
 * Include      :
 *-----------------------------------------------------------------------------
 * Declaration : void g2d_start_rendering(void);
 *-----------------------------------------------------------------------------
 * Function     : Starts rendering by G2D.
 *-----------------------------------------------------------------------------
 * Argument     : void
 *-----------------------------------------------------------------------------
 * Return Value:  0: success
 *              : -1: failure
 *""FUNC COMMENT END""*************************************************************/
int g2d_start_rendering(void)
{
    G2D.SRCR.LONG = (0x00040000|(0x00000004|0x00000002|0x00000001));
        /* Clear status */
    G2D.SCLR.BIT._RS = 1;                /* Start rendering */

    while(1){
        if(G2D.SR.BIT._TRA)              /* Wait for command execution end */
            break;
        if(G2D.SR.BIT._CER)
            return (-1);                 /* Generate command error */
    }
    return (0);
}
```

3.7 Sample Program Listing: WPR Command Setting

```
/*""FUNC COMMENT""*****************************************************************
 * ID           :
 * Outline      : WPR command setting
 *-------------------------------------------------------------------------------
 * Include      :
 *-------------------------------------------------------------------------------
 * Declaration : void g2d_com_WPR_set(void);
 *-------------------------------------------------------------------------------
 * Function     : Uses WPR command to set register values.
 *-------------------------------------------------------------------------------
 * Argument     : unsigned long *displist
 *              : short RegNo
 *              : unsigned long Data
 *-------------------------------------------------------------------------------
 * Return Value: unsigned long *
 *""FUNC COMMENT END""*************************************************************/
unsigned long * g2d_com_WPR_set(unsigned long * displist, short RegNo, unsigned long Data)
{
    *displist = 0x18000000;              /* Set OP CODE and draw mode */
    displist++;

    *displist = RegNo & 0x000001FF;      /* Set data count and W Reg No */
    displist++;

    *displist = Data;                    /* Set data */
    displist++;

    return displist;
}
```

3.8 Sample Program Listing: LINEC Command Setting

```
/*""FUNC COMMENT""*****************************************************************
 * ID           :
 * Outline      : LINEC command setting
 *-----------------------------------------------------------------------------
 * Include      :
 *-----------------------------------------------------------------------------
 * Declaration : void g2d_com_LINEC_set(void);
 *-----------------------------------------------------------------------------
 * Function     : Sets LINEC command in display list.
 *-----------------------------------------------------------------------------
 * Argument     : short x1
 *              : short y1
 *              : short x2
 *              : short y2
 *              : short color
 *              : short width
 *-----------------------------------------------------------------------------
 * Return Value: unsigned long *
 *""FUNC COMMENT END""*************************************************************/
unsigned long * g2d_com_LINEC_set(unsigned long * displist, short x1, short y1,
                                  short x2, short y2, short color, short width)
{
    *displist = 0xB0000002;              /* Set OP CODE and draw mode */
    displist++;

    *displist = ((color << 16) | (2 & 0x0000FFFF));  /* Set color and number of vertices */
    displist++;

    *displist = width & 0x0000003F;                  /* Set line width */
    displist++;

    *displist = ((x1 << 16) | (y1 & 0x0000FFFF));    /* Rendering coordinate */
    displist++;

    *displist = ((x2 << 16) | (y2 & 0x0000FFFF));    /* Rendering coordinate */
    displist++;

    return displist;
}
```

4. Reference Documents

- Hardware Manual
  SH7764 Hardware Manual
  (The latest version can be downloaded from the Renesas Technology Web site.)
- Software Manual
  SH-4A Software Manual
  (The latest version can be downloaded from the Renesas Technology Web site.)

Website and Support

Renesas Technology Website
http://www.renesas.com/

Inquiries
http://www.renesas.com/inquiry
csc@renesas.com

Revision Record

		Description	
Rev.	Date	Page	Summary
1.00	Jan.29.10	—	First edition issued

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.