SH7722/SH7731 Group

LCD Controller Setup Example

R01AN0669EJ0101
Rev.1.01
Jun 09, 2011

Introduction

This application note describes how to use the LCD controller (LCDC) of the SH7722 and SH7731, and shows a display example by the TFT-LCD module.

Target Device

SH7722, SH7731

Contents

1. Preface

1.1 Specifications

- In this application note, section 2 gives a supplementary explanation of the LCDC description in the hardware manual, and section 3 describes the LCDC bus load factor necessary for system design. Section 4 shows an application example in which four types of images are alternately displayed on the TFT-LCD module and the TFT-LCD module is repeatedly turned on and off.

1.2 Modules Used

- LCDC

1.3 Applicable Conditions

• Evaluation board	SH7722 reference platform (product code: R0P7722TH001ARK) manufactured by Renesas Electronics Corporation		
	External memory	(area 0)	64-MB NOR-type flash memory: S29GL512N10FF1020 from Spansion
		(area 3)	64-MB SDRAM: MT48LC8M16A2B475 from Micron
• MCU	SH7722 (R8A77220AC266BGV)		
• Operating frequency	CPU clock: 200.00 MHz SH bus clock: 100.00 MHz U memory clock: 100.00 MHz Bus clock: 50.00 MHz SDRAM clock: 80.00 MHz Peripheral clock: 25.00 MHz		

Note: The operating frequency is set not at the maximum specification but at a setting that makes the peripheral clock frequency become 25 MHz. For details on the frequency determination method, refer to section 4.2.2.

• Bus width for area 0	16-bit (with the MD3 pin at the low level)
• Bus width for area 3	64-bit
• Clock operating mode	Mode 0 (with the MD0 pin and the MD1 pin at the low level)
• Endian	Little endian (with the MD5 pin at the high level)
• Toolchain	SuperH RISC engine Standard Toolchain Ver.9.3.0.0 from Renesas Electronics
• Compiler options	Default settings of High-performance Embedded Workshop (-cpu=sh4aldsp -endian=little -include="$(PROJDIR)\inc" -object="$(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 Terms Used in This Application Note

- Frame

  A frame is pixel information located in the memory in the same way the image is to be displayed on the screen. A buffer in which frame information is stored is called a frame buffer. The LCDC reads pixel information from the frame buffers and uses it to display images.
- Refresh rate [Hz]

  Indicates how many times the LCD module connected to the LCDC rewrites the screen in 1 second. This is equal to the frequency of the vertical sync signal VSYNC output by the LCDC in 1 second.
- Frame rate [fps]

  Indicates the number of images updated by the image display system in 1 second. The upper limit of the images that can be updated by the LCD module is equal to the refresh rate [Hz].

  When the frame rate is lower than the refresh rate, e.g., frame rate is 30 fps for an LCD module whose VSYNC frequency is 60 Hz, the LCD module displays updated images for 30 times in the period of 60 VSYNC signals.

1.5 Scope of This Application Note

This application note describes the basic usage method of the LCDC without an OS, that is, to continuously display images in the frame buffers onto the LCD module supporting the RGB interface mode. The following functions are not covered by the descriptions in this application note.

- Linked operation of BEU and LCDC
- YCbCr output
- Color palette
- SYS interface
- Sub LCD module display
- Partial screen mode
- Write-back of display data
- VSYNC input mode
- One-shot mode

1.6 Related Application Note

The operation of the reference programs in this application note has been confirmed under setup conditions partially modified from those in the "SH7722/SH7731 Group Application Note: SH7722/SH7731 Example of Initialization (REJ06B0942)". Also refer to that application note.

2. LCDC Operation

2.1 Overview of LCDC

The LCDC is an image system module that can read image data from the frame buffers located in the external memory and display images onto the TFT-LCD module. This application note introduces a setup example in which RGB- or YCbCr-format image data is read and an image is displayed on the LCD module in RGB interface mode.

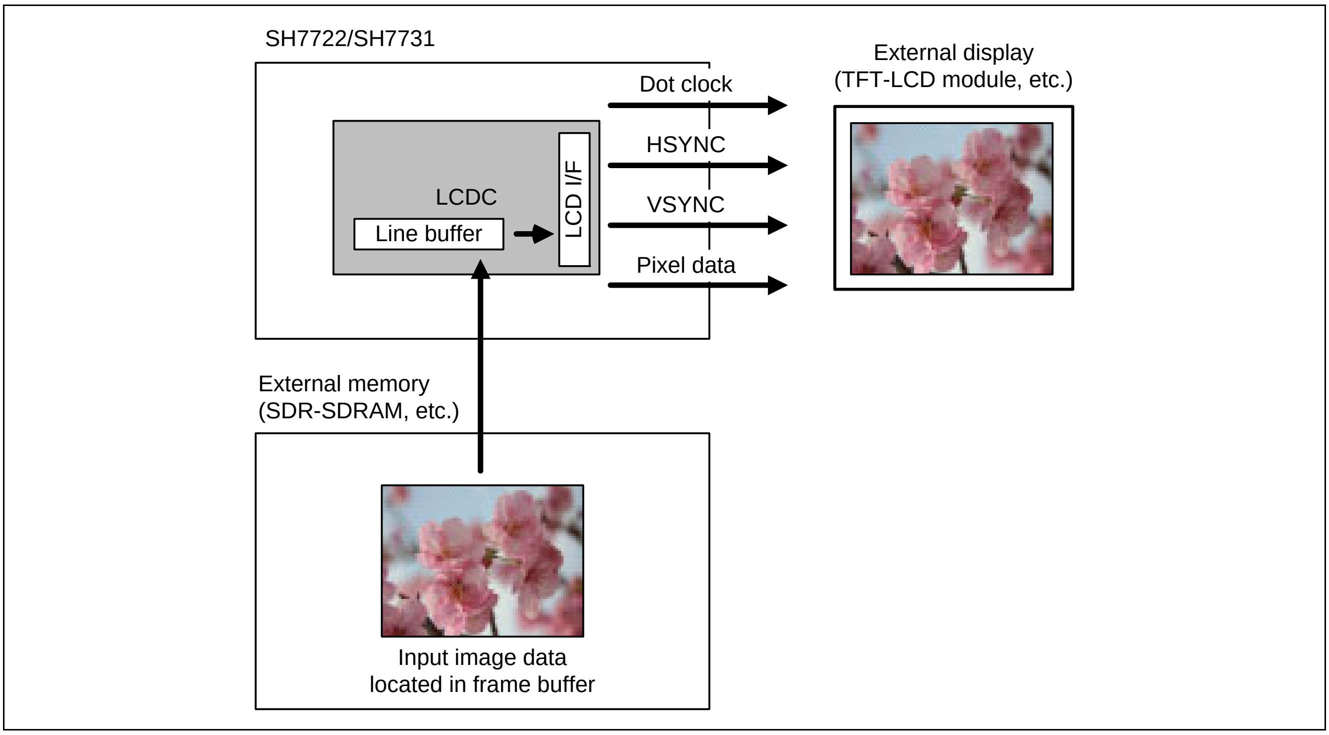


Figure 1 Image of LCDC Operation

2.1.1 Image Data Read

The LCDC reads from the frame buffer one pixel each rightward, starting from the base point of an image that is located at the top-left corner. The pixel information for the amount of the horizontal image size is called one line. After reading for one line has finished, reading is continued after returning to the left end one line down. Reading for one frame is completed when the number of lines equal to the vertical image size has been read. Reading of the next frame is started again from the base point.

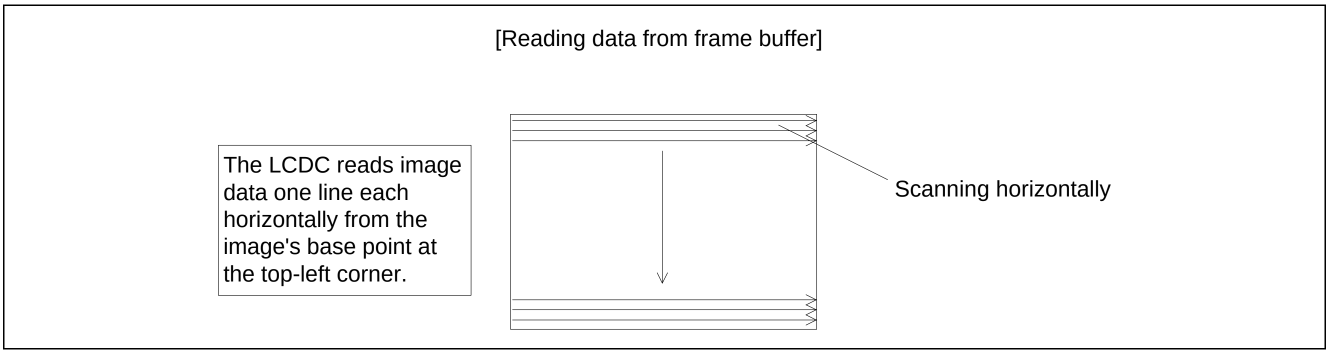

[Reading data from frame buffer]
The LCDC reads image data one line each horizontally from the image's base point at the top-left corner.
Scanning horizontally

Figure 2 Image Data Input

2.1.2 Image Data Output

The LCDC outputs image data in synchronization with the following three sync signals.

- Dot clock (DotCLK):
  The information for one pixel is output in synchronization with the dot clock.
- Horizontal sync signal (HSYNC):
  The information for a single horizontal line of an image is output in synchronization with the horizontal sync signal. The periods that are before and after the sync signal and in which no pixel information is output are referred to as the horizontal front porch and horizontal back porch, respectively.
- Vertical sync signal (VSYNC):
  The information for a single frame of an image is output in synchronization with the vertical sync signal. The periods that are before and after the sync signal and in which no pixel information is output are referred to as the vertical front porch and vertical back porch, respectively.

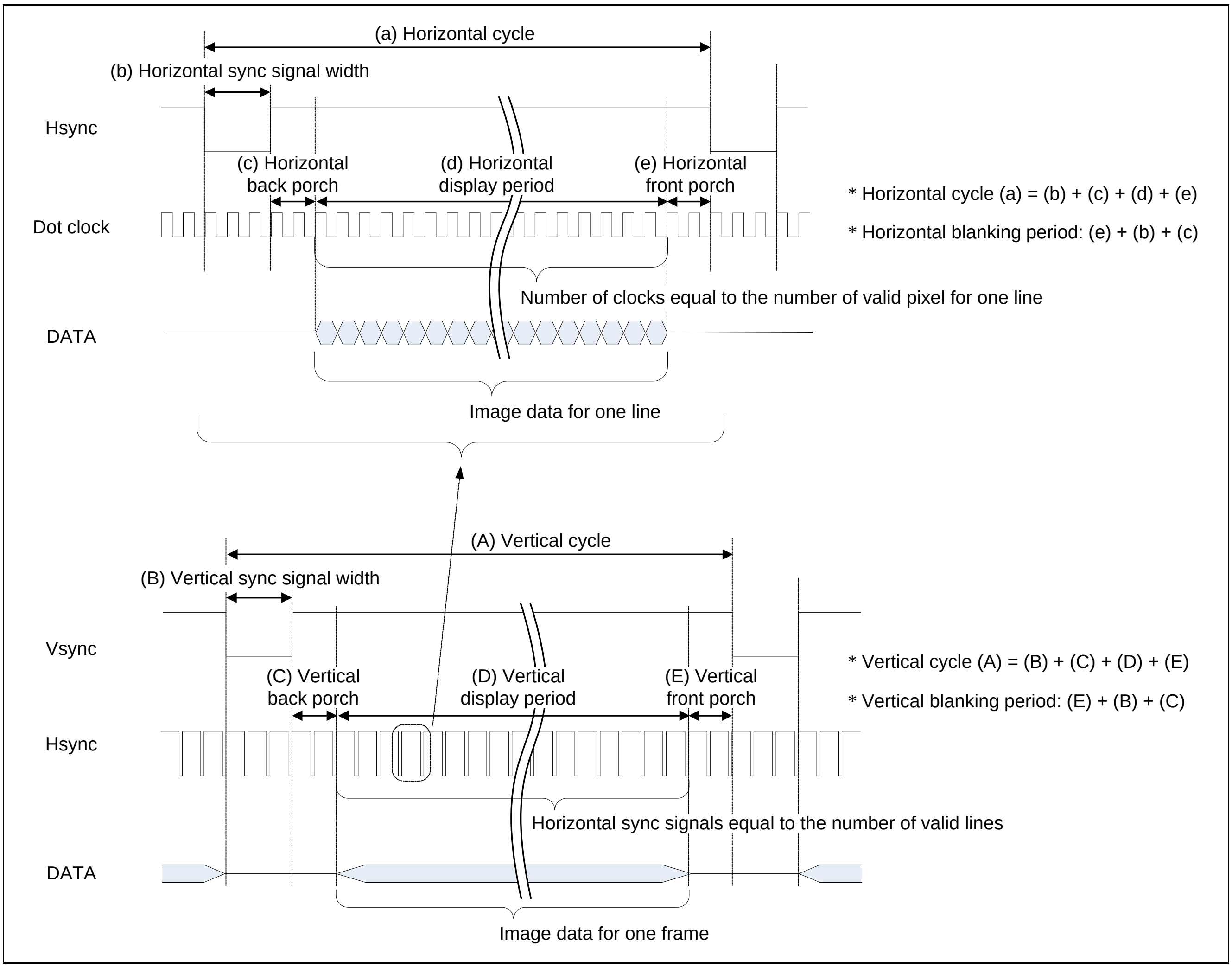

(a) Horizontal cycle
(b) Horizontal sync signal width
Hsync
(c) Horizontal back porch
(d) Horizontal display period
(e) Horizontal front porch
Dot clock
* Horizontal cycle (a) = (b) + (c) + (d) + (e)
* Horizontal blanking period: (e) + (b) + (c)
Number of clocks equal to the number of valid pixel for one line
DATA
Image data for one line
(A) Vertical cycle
(B) Vertical sync signal width
Vsync
(C) Vertical back porch
(D) Vertical display period
(E) Vertical front porch
* Vertical cycle (A) = (B) + (C) + (D) + (E)
* Vertical blanking period: (E) + (B) + (C)
Hsync
Horizontal sync signals equal to the number of valid lines
DATA
Image data for one frame

Figure 3 Input of Image Data and Sync Signals

Each cycle of the sync signal includes an image-display period and a non-display period. The non-display period in the horizontal direction starts from the end of displaying a single line from the left to the right and lasts until the beginning of displaying the next single line after returning to the left. This period is called the horizontal blanking period. The non-display period in the vertical direction starts from the end of displaying a single frame and lasts until the beginning of displaying the next single frame. This period is called the vertical blanking period.

2.2 LCDC Specifications

Table 1 lists the main specifications of the LCDC.

Table 1 LCDC Specifications

Item	Specifications
Maximum displayable size by LCD module Note: The bus occupation rate needs to be calculated in actual display operations. For details, see section 3.	[Horizontal image size] • Up to 1024 pixels when the output data format is the 16-bit RGB565 packed format (2 bytes/pixel) • Up to 512 pixels when the output data format is the 32-bit RGB888 packed format (4 bytes/pixel) [Vertical image size] • Up to 1024 pixels
Input data format	32-bit RGB888 packed format (from the lower bits) 32-bit RGB888 packed format (from the upper bits) 24-bit RGB888 packed format (* See section 2.3.4, Note 2.) 16-bit RGB565 packed format 32-bit RGB666 packed format 16-bit RGB444 packed format 24-bit RGB666 packed format 24-bit BGR666 packed format 24-bit BGR888 packed format 32-bit BGR888 packed format 32-bit RGB565 packed format YCbCr444 YCbCr422 YCbCr420
Output data format	[RGB interface] RGB8 3 cycle/pixel RGB9 2 cycle/pixel RGB12a 2 cycle/pixel RGB12b 1 cycle/pixel RGB16 1 cycle/pixel RGB18 1 cycle/pixel RGB24 1 cycle/pixel
	[SYS interface] Not handled in this application note.
Display data write-back	Not handled in this application note.
LCD driver interface	[RGB interface of main LCD] • Interface using horizontal sync signal and vertical sync signal • The polarity of a signal can be set • The output width and output position of an image signal can be set
	[SYS interface of main LCD] Not handled in this application note.
	[Only SYS interface of sub LCD] Not handled in this application note.
Dot clock	• The source clock should be selected from the following three clocks and it can be output after its frequency has been divided. — Bus clock (Bϕ) — Peripheral clock (Pϕ) — External input clock • A frequency of up to 33.3 MHz can be output. If Bϕ is set to a frequency slower than 33.3 MHz, the value of Bϕ becomes the maximum value of the dot clock. • The division ratio can be selected from 1/1 (no division), 1/2, and 1/3 to 1/60. Division in which the numerator becomes a value other than 1 (e.g., division by 2/5) is not allowed.

2.3 LCDC Setup

The LCDC functions and register setting methods are described here.

2.3.1 Canceling Module Standby State

The SH7722 has a module standby function that starts and stops clock supply to each peripheral module. Since the LCDC is a module whose clock supply is halted after a power-on reset, the module standby state has to be canceled and the clock supply started before setting the LCDC registers.

Table 2 shows the register used for setting a power saving mode.

Table 2 Power-Down Mode Setting

Function	Register
Enables or disables the LCDC power-down mode	MSTP200 bit in module stop register 2 (MSTPCR2)

When canceling the module standby state by software, start setting the LCDC registers after making sure that the module standby state has been canceled by polling until the MSTP200 bit is read as 0. The LCDC registers cannot be written to in the module standby state.

2.3.2 Setting the Source for Data Acquisition

The LCDC selects the memory or BEU as the source for acquiring image data. When the BEU is selected, the LCDC operates together with the BEU. In the SH7731 that does not incorporate the BEU, the source for data acquisition needs not be set, and data is always acquired from the memory. In the SH7722, the source for data acquisition can be selected as the memory or the BEU. If the BEU is selected, BEU-LCDC linked operation is performed (linked operation of the BEU and LCDC is not handled in this application note).

Table 3 shows the LCDC register used for setting the source for data acquisition.

Table 3 Setting the Source for Data Acquisition

Function	Register
Setting the Source for Data Acquisition	LCDC control register 2 (LDCNT2R) • In the SH7731, the source for data acquisition must always be set to memory.

2.3.3 Setting the Output Destination

The LCDC selects from two display destinations: main LCD and sub LCD. The RGB interface or SYS interface can be selected for the main LCD. Only the SYS interface is available for the sub LCD.

In this application note, only the RGB interface of the main LCD is handled.

Table 4 shows the LCDC registers used for setting the output destination.

Table 4 Setting the Output Destination

Function	Register
Setting the Output Destination	LCDC control register 1 (LDCNT1R) LCDC control register 2 (LDCNT2R)

2.3.4 Setting the Frame Buffer

Figure 4 shows the frame buffer information required by the LCDC when reading image data from the memory.

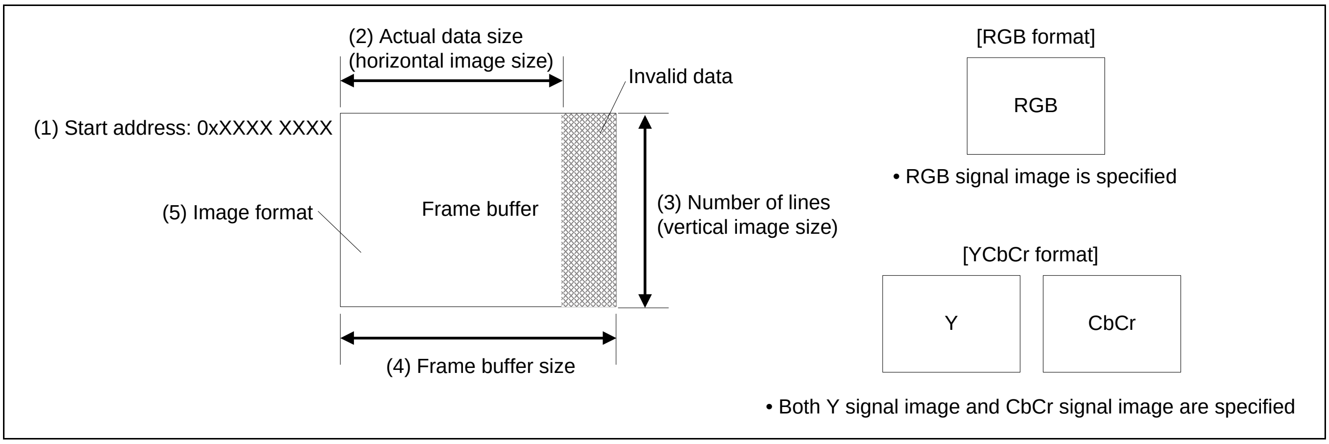


Figure 4 Frame Buffer Configuration

(1) Start address: Set a 8-byte aligned address for the base point address of the frame buffer. The address should be set as a 32-bit physical address with the upper three bits as 0.

(2) Actual data size: Specify the horizontal image size in pixel units. Note that this setting must be a multiple of 8 because the LCDC handles data in the horizontal direction in units of characters (1 character: 8 pixels).

(3) Number of lines: Specify the vertical image size in pixel units.

(4) Frame buffer size: Specify the frame buffer size in byte units, independent of the horizontal image size. The frame buffer size must be equal to or greater than the horizontal image size. When set to a value greater than the horizontal image size, the right end of the frame buffer becomes invalid data.

(5) Image format: The LCDC can handle the image formats listed in "input data format" in table 1.

— Only one frame buffer is used for the RGB format.

— Since the YCbCr format data is handled with the Y signal and CbCr signal separated, two frame buffers are used. In this case, set start address (1) in the register for the start address of the Y signal and the register for the start address of the CbCr signal. Set frame buffer information on the Y signal for (2) to (4).

Notes: 1. If YCbCr has been set as the image format, when an image is output to the RGB interface, the color converted format (YUV → RGB) supports only the ITU-BT.601 standard. The conversion formula can be selected from the following two types.

- Extension type: Conversion of BT.601-range YCbCr (Y: 16 to 235, C: 16 to 240) ↔ full-range RGB (0 to 255)
- Compression type: Conversion of full-range YCbCr (0 to 255) ↔ full-range RGB (0 to 255)

2. Though 24-bpp RGB888 is not listed in table 31.7, RGB Packed Format, in section 31, LCD Controller (LCDC), in the "*SH7722 Group Hardware Manual* (REJ09B0314-0200) Rev. 2.00", or in table 27.7, RGB Packed Format, in section 27, LCD Controller (LCDC), in the "*SH7731 Group Hardware Manual* (REJ09B0488-0100) Rev. 1.00", it can be used by setting PKF[4:0] = B'00010.

3. When the image format of the frame buffer differs from the image format of the LCD module that is connected to the LCDC, the LCDC automatically pads the lower bits with 0 or discards the lower bits. For example, when the image format of the frame buffer is RGB565 while the image format of the LCD module is RGB666, the LCDC adds 0 to the LSB in the R and B data and outputs the data in the RGB666 format. For details on how to set the image format of the LCD module, see section 2.3.8.

Table 5 shows the LCDC registers used for setting the frame buffer.

Table 5 Frame Buffer Setting

Function	Register
Sets the start address (Y signal/RGB signal)	Main LCD display data read start address register 1 (MLDSA1R)
Sets the start address (CbCr signal)	Main LCD display data read start address register 2 (MLDSA2R)
Horizontal image size	See section 2.3.7, Setting the Sync Signals.
Vertical image size	See section 2.3.7, Setting the Sync Signals.
Horizontal size of frame buffer (stride)	Main LCD display data storing memory line size register (MLDMLSR)
Selects the image format (RGB888, RGB565, and YCbCr420, etc.), and selects the YCbCr → RGB conversion formula	Main LCD data format register (MLDDFR)

2.3.5 Setting Data Swapping

The LCDC is a module that operates in big endian. When the SH7722/SH7731 is operated in little endian and data is located in the frame buffers in the memory in little endian, data swapping should be set adequately for each image format. Byte swapping, word swapping, and longword swapping can be individually set when the LCDC reads data from the memory.

Table 6 shows the LCDC register used for setting data swapping.

Table 6 Data Swapping Setting

Function	Register
Selects longword swapping	LCDC input image data swap register (LDDDSR)
Selects word swapping	
Selects byte swapping	

Table 7 shows examples of swapping setting for frequently used image formats.

Table 7 Swapping Setting Examples

Image Format	Setting
32-bit RGB888 packed format (from the lower bits) 32-bit RGB888 packed format (from the upper bits)	LDDDSR = 0x00000004 • Longword swapping is enabled
24-bit RGB888 packed format	LDDDSR = 0x00000007 • Longword swapping is enabled • Word swapping is enabled • Byte swapping is enabled
16-bit RGB565 packed format	LDDDSR = 0x00000006 • Longword swapping is enabled • Word swapping is enabled

2.3.6 Setting the Dot Clock

The dot clock should be set according to the AC characteristics of the LCD module to be used. The dot clock can be generated by dividing the source clock.

(1) Source clock: Can be selected from the three clock types of the bus clock (Bϕ), peripheral clock (Pϕ), and external input clock input from the LCDLCLK pin. Since only an integer ratio (1/1, 1/2, 1/3, ...) can be set for the frequency division ratio, select the source clock whose cycle is an integer multiple of the desired dot clock. Note that the following restrictions are applied.
 — When Pϕ is selected, Pϕ should be set to a lower frequency than Bϕ.
 — The external input clock should have a frequency equal to or lower than Bϕ, including jitter.

(2) Division ratio: Select the division ratio of 1/1 or other than 1/1 (1/2, 1/3, ...). When "other than 1/1" is selected, the division ratio is determined by both the denominator (m) of the division ratio and the dot clock pattern.

How to Determine the Division Ratio

When "other than 1/1" is selected, the pattern set in LCD dot clock pattern registers 1 and 2 for the source clock is output without change as the dot clock, as shown in figure 5.

The relationship between the settings of the bits in the LCD dot clock pattern registers and the dot clock output level is determined by the setting of the DWPOL bit in main LCD module type register 1 as follows:

- DWPOL bit = 1: LCD dot clock pattern register (0, 1) → Dot clock output level (H, L)
- DWPOL bit = 0: LCD dot clock pattern register (0, 1) → Dot clock output level (L, H)

Described in this section is the case in which the DWPOL bit is set to 0. When setting the DWPOL bit to 1, assume that the dot clock output level is inverted.

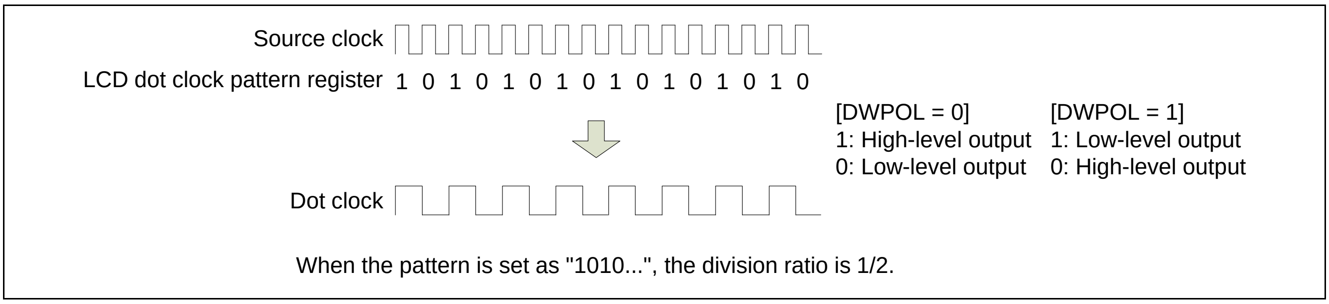


Figure 5 Division Ratio Setting

Normally, a cyclic pattern in which 1 and 0 appear alternately, such as, "1010..." and "110110...", is set as the dot clock pattern. This repetitive cycle is denoted by "A". A = 2 for the pattern of "10,10,...". A = 3 for the pattern of "110,110,...". At this time, the division ratio is 1/A.

(1) To make the division ratio be 1/A, select a denominator (m) divisible by A from among 42, 48, 54, and 60. m can be selected from only these four values. Any value can be selected as long as it is divisible.

(2) Next, set the "1, 0" pattern of the repetitive cycle A in the LCD dot clock pattern registers for the number of bits equal to the value of m. If A = 2, set the "1010..." pattern in which 1 and 0 are alternately lined one bit each. If A = 3, set the "110110..." or "100100..." pattern. When the division ratio is set to an odd number like A = 3, the number of 1s and 0s are not the same so the clock duty ratio will not be 1:1. Confirm that the LCD module used does not have any problems with the duty ratio of the dot clock not being 1:1.

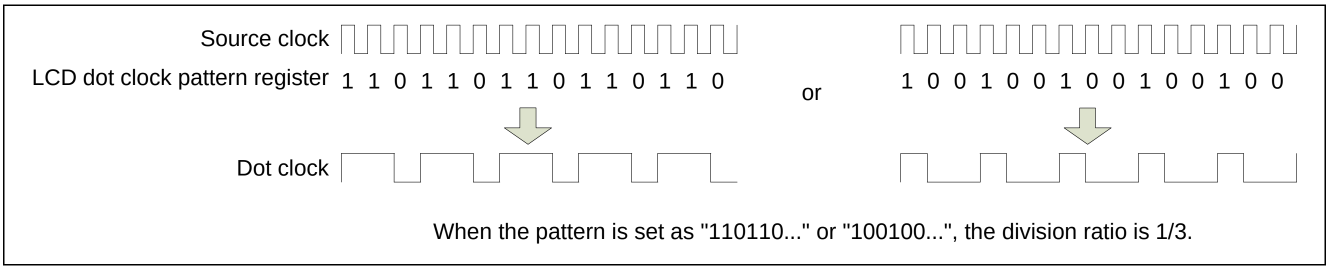
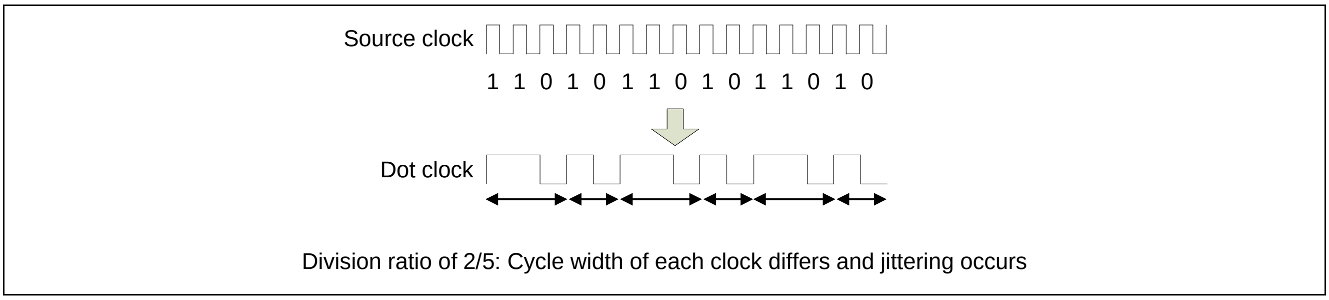

Source clock
LCD dot clock pattern register
1 1 0 1 1 0 1 1 0 1 1 0 1 1 0
or
1 0 0 1 0 0 1 0 0 1 0 0 1 0 0
Dot clock
When the pattern is set as "110110..." or "100100...", the division ratio is 1/3.

Figure 6 Setting a Division Ratio of 1/3

(3) Do not make a setting that causes the numerator to be a value other than 1, e.g., 2/5. When such a setting is made, the cycle width of each clock differs and jittering occurs.



Division ratio of 2/5: Cycle width of each clock differs and jittering occurs

Figure 7 Setting a Division Ratio of 2/5 (Setting Prohibited)

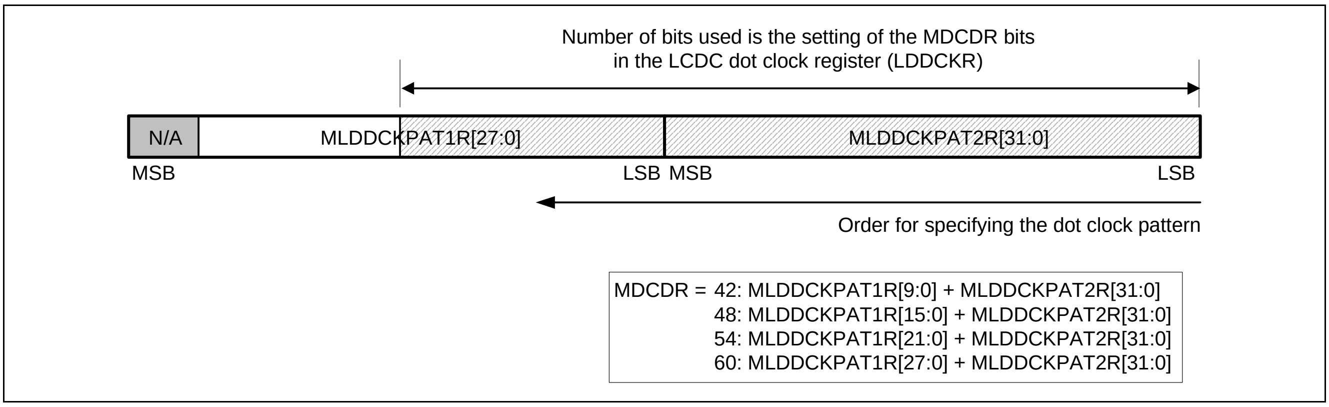
The division ratio can be set from 1/2 to 1/60. If a desired dot clock cannot be obtained by frequency division, input an appropriate clock signal to the LCDLCLK pin and select the external clock as the source clock.

Table 8 shows the LCDC register used for setting dot clock.

Table 8 Dot Clock Setting

Function	Register
Selects the source clock	LCDC dot clock register (LDDCKR)
Selects division by 1/1 or division by a value other than 1/1	
Selects the denominator (m) of the division ratio	
Sets the dot clock pattern	Main LCD dot clock pattern register 1 (MLDDCKPAT1R)
	Main LCD dot clock pattern register 2 (MLDDCKPAT2R)

The denominator (m) of the division ratio is set by the MDCDR bits in the LCDC dot clock register (LDDCKR). Figure 8 shows which bits in LCD dot clock pattern registers 1 and 2 are used.


Number of bits used is the setting of the MDCDR bits
in the LCDC dot clock register (LDDCKR)
N/A
MLDDCKPAT1R[27:0]
MLDDCKPAT2R[31:0]
MSB
LSB
MSB
LSB
Order for specifying the dot clock pattern

MDCDR = 42: MLDDCKPAT1R[9:0] + MLDDCKPAT2R[31:0]
48: MLDDCKPAT1R[15:0] + MLDDCKPAT2R[31:0]
54: MLDDCKPAT1R[21:0] + MLDDCKPAT2R[31:0]
60: MLDDCKPAT1R[27:0] + MLDDCKPAT2R[31:0]

Figure 8 Bits Used in LCD Dot Clock Pattern Registers 1 and 2

Dot clock pattern examples with division ratios from 1/2 to 1/8 are shown in table 9.

Table 9 Division Ratio Setting Examples

Division Ratio	Selectable Denominator (m)	Dot Clock Pattern to be Set (Illustrated with Smallest Possible m)

1/2	42, 48, 54, 60	MLDDCKPAT1R = 0x000002AA,
		MLDDCKPAT2R = 0xAAAAAAAA

1/3	42, 48, 54, 60	H:L = 2:1, MLDDCKPAT1R = 0x0000036D, MLDDCKPAT2R = 0xB6DB6DB6 or H:L = 1:2, MLDDCKPAT1R = 0x00000249, MLDDCKPAT2R = 0x24924924

1/4	48, 60	MLDDCKPAT1R = 0x0000CCCC,
		MLDDCKPAT2R = 0xCCCCCCCC

1/5	60	H:L = 2:1, MLDDCKPAT1R = 0x0E739CE7, MLDDCKPAT2R = 0x39CE739C or H:L = 1:2, MLDDCKPAT1R = 0x0C6318C6, MLDDCKPAT2R = 0x318C6318

1/6	42, 48, 54, 60	MLDDCKPAT1R = 0x0000038E,
		MLDDCKPAT2R = 0x38E38E38

1/7	42, 48	H:L = 4:3, MLDDCKPAT1R = 0x000003C7, MLDDCKPAT2R = 0x8F1E3C78 or H:L = 3:4, MLDDCKPAT1R = 0x00000387, MLDDCKPAT2R = 0x0E1C3870

1/8	48	MLDDCKPAT1R = 0x0000F0F0,
		MLDDCKPAT2R = 0xF0F0F0F0

2.3.7 Setting the Sync Signals

The sync signal parameters in figure 9 should be set according to the characteristics (A) to (E) and (a) to (e) in figure 3 of the LCD module used.

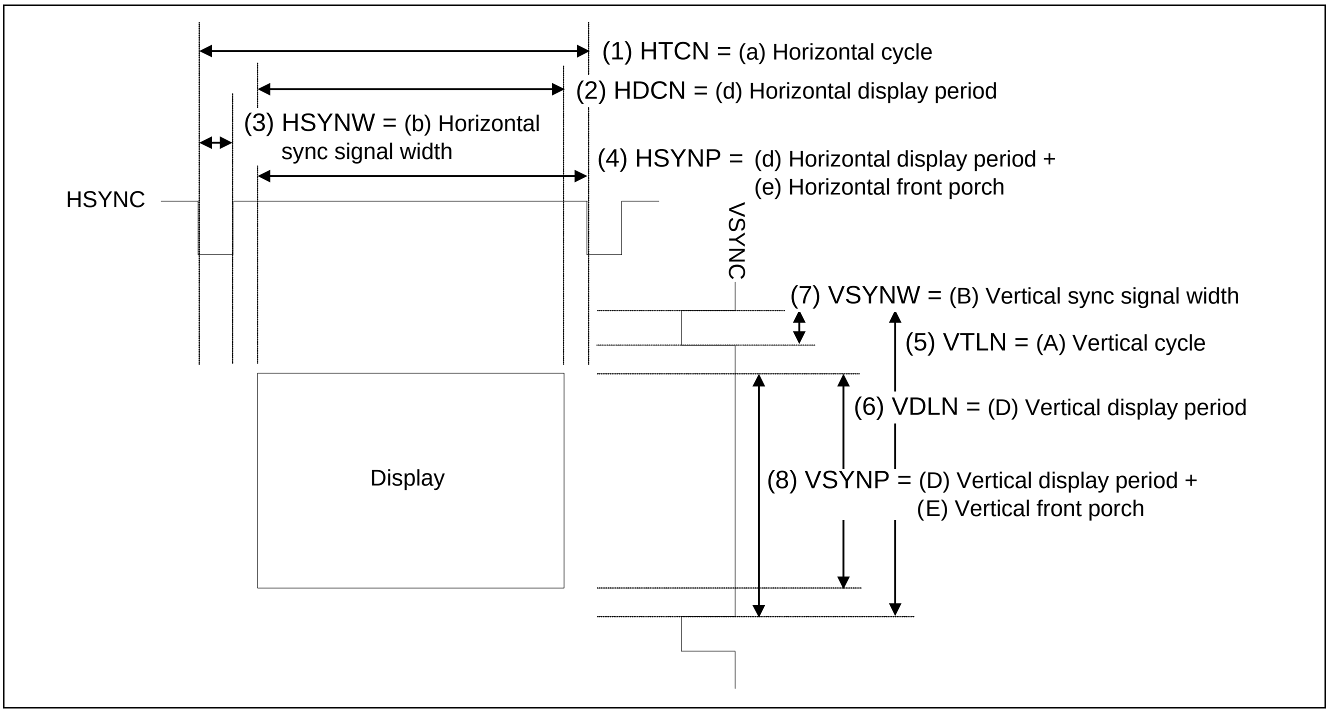

(1) HTCN = (a) Horizontal cycle
(2) HDCN = (d) Horizontal display period
(3) HSYNW = (b) Horizontal sync signal width
(4) HSYNP = (d) Horizontal display period + (e) Horizontal front porch
HSYNC
VSYNC
(7) VSYNW = (B) Vertical sync signal width
(5) VTLN = (A) Vertical cycle
(6) VDLN = (D) Vertical display period
Display
(8) VSYNP = (D) Vertical display period + (E) Vertical front porch

Figure 9 Sync Signal Configuration

(1) HTCN: Sets the horizontal cycle in units of characters.
(2) HDCN: Sets the horizontal display period in units of characters.
(3) HSYNW: Sets the horizontal sync signal width in units of characters.
(4) HSYNP: Sets "horizontal display period + horizontal front porch width" in units of characters.
(5) VTLN: Sets the vertical cycle in units of horizontal sync signals.
(6) VDLN: Sets the vertical display period in units of horizontal sync signals.
(7) VSYNW: Sets the vertical sync signal width in units of horizontal sync signals.
(8) VSYNP: Sets "vertical display period + vertical front porch width" in units of horizontal sync signals.

Notes on Horizontal Direction Settings

- 1 character = 8 pixels (8 dot clocks). Accordingly, each parameter in the horizontal direction can be set only in dot clock cycles as a multiple of 8. In some cases, a parameter in the horizontal direction needs to be set in dot clock cycles as a value other than a multiple of 8, depending on the LCD module used. However, in such a case, specify the nearest multiple of 8, convert that value into units of characters (change it to 1/8 of the value), and set the converted value. Confirm thoroughly that this setting does not cause trouble in the LCD module used. For example, there may be an undisplayed range at either the left or right end.
- The size of horizontal pixel data must be equal to or less than 2048 bytes per line. For example, when the 16-bit RGB565 packed format is specified by setting the PKF bits in MLDDFR to B'00011, the maximum number of horizontal pixels is 1024 (128 characters).
- Make a setting to satisfy the expression of "HTCN ≥ HDCN + 3".
- Make a setting to satisfy the expression of "HTCN ≥ HSYNP + HSYNW".
- Make a setting to satisfy the expression of "HSYNP ≥ HDCN + 1".

Notes on Vertical Direction Settings

- Make a setting to satisfy the expression of "VTLN ≥ VDLN + 1".

Table 10 shows the LCDC registers used for setting the sync signals.

Table 10 Sync Signal Setting

Function	Register
Sets the horizontal sync signal	Main LCD horizontal character number register (MLDHCNR)
	Main LCD horizontal sync signal register (MLDHSYNR)
Sets the vertical sync signal	Main LCD vertical line number register (MLDVLNR)
	Main LCD vertical sync signal register (MLDVSYNR)

2.3.8 Setting the LCD Interface

The LCD module interface (RGB or SYS), transfer mode, signal polarity, etc. should be set according to the specifications of the LCD module used. Set each value with reference to the datasheet for the LCD module used.

(1) Module interface: Selects the RGB interface or SYS interface. This application note shows settings for a case in which the RGB interface is selected.

(2)Main LCD module setting: Selects the bit width and number of transfer cycles of the image data of the LCD module.

(3) Sync signal polarity: Sets the vertical sync signal (VSYNC) and horizontal sync signal (HSYNC) to low-active or high-active. When low-active is set, a low-level signal is output during the period of the sync signal (HSYNC/VSYNC in section 2.3.7). In contrast, a high-level signal is output when high-active is set.

(4) HSYNC signal output control: Sets whether or not to output HSYNC during the VSYNC blanking period.

(5) Dot clock control: Sets whether or not to output the dot clock during the blanking period of VSYNC and HSYNC.

(6) Dot clock polarity: Selects whether the LCD module latches image data at the rising edge of the dot clock or the falling edge of the dot clock.

(7) Display enable signal polarity: Sets the display enable signal (LCDDISP) to low-active or high-active.

(8) Display data polarity: Sets the display data to high-active (bit value is output without change) or low-active (bit value is output after being inverted).

For details, please refer to the section 31.3.8 "Main LCD Module Type Register 1 (MLDMT1R)" in the *SH7722 Group Hardware Manual* (REJ09B0314), or the section 27.3.8 "Main LCD Module Type Register 1 (MLDMT1R)" in the *SH7731 Group Hardware Manual* (REJ09B0488).

Table 11 shows the LCDC registers used for setting the LCD interface.

Table 11 Setting the LCD Interface

Function	Register
• Setting the module interface • VSYNC polarity • HSYNC polarity • Dot clock polarity	Main LCD module type register 1 (MLDMT1R)
Note: Used when the SYS interface is selected. Not handled in this application note.	Main LCD module type register 2 (MLDMT2R)
	Main LCD module type register 3 (MLDMT3R)

2.3.9 Setting the Power Management

The LCDC has a function to operate, in accordance with the specified sequence, three control signals for managing the power supply of the LCD module. Using this function enables power control of the LCD module to be synchronized with the display start and display stop of the LCDC. The external pins that can be controlled by this function are the following three pins (hereafter expressed as power management pins).

- LCDVCPWC pin: First pin to become high at display start and last pin to become low at display stop.
- LCDVEPWC pin: Second pin to become high at display start and second pin to become low at display stop.
- LCDDON pin: Last pin to become high at display start and first pin to become low at display stop.

Table 12 shows the LCDC registers used for setting the power management.

Table 12 Power Management Setting

Function	Register
Sets the timing for the power management pins	Main LCD power management register (MLDPMR)

Transitions of the signal levels of the power management pins at LCDC activation are shown in figure 10.

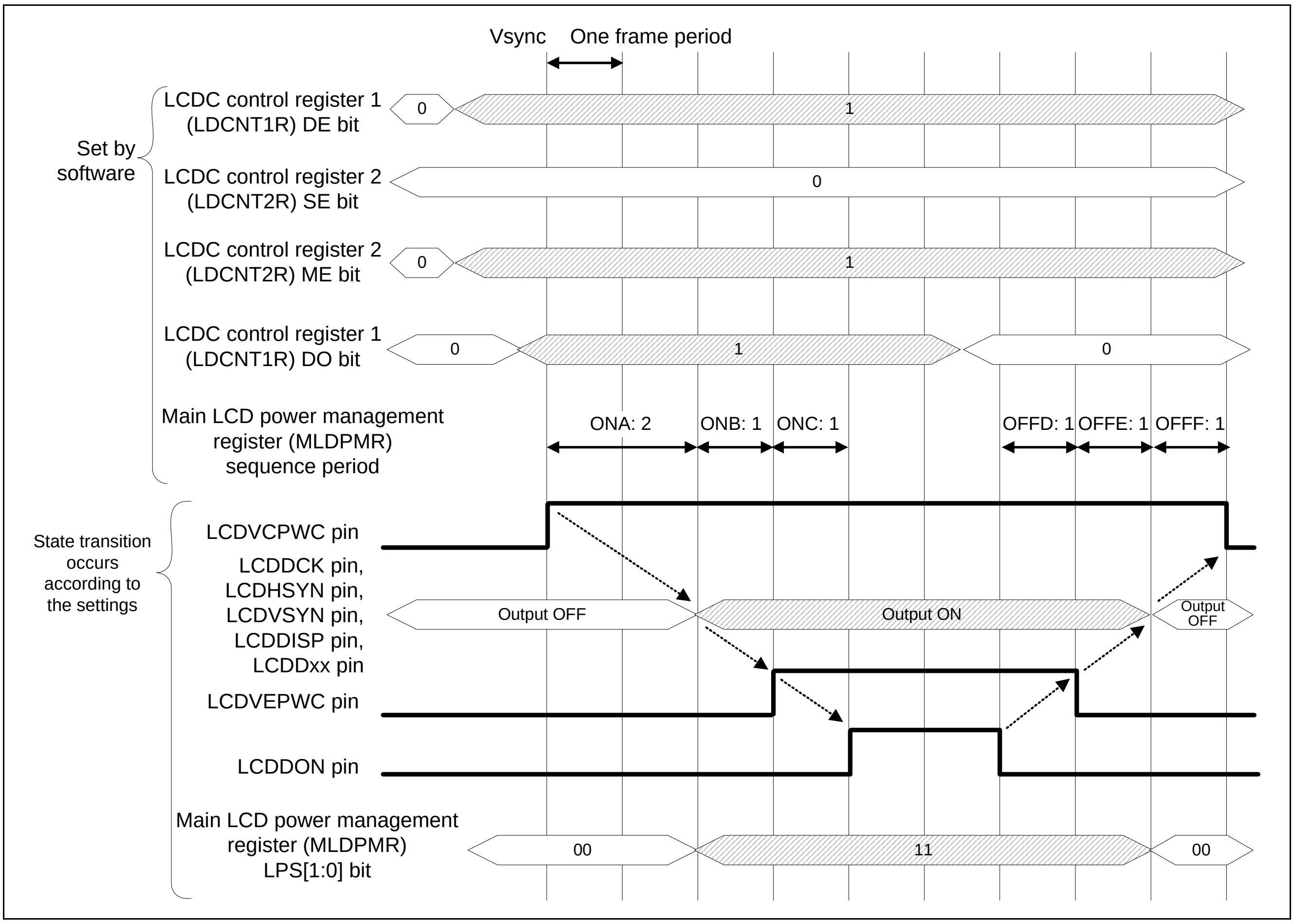

Vsync
One frame period
Set by software
LCDC control register 1 (LDCNT1R) DE bit
0
1
LCDC control register 2 (LDCNT2R) SE bit
0
LCDC control register 2 (LDCNT2R) ME bit
0
1
LCDC control register 1 (LDCNT1R) DO bit
0
1
0
Main LCD power management register (MLDPMR) sequence period
ONA: 2
ONB: 1
ONC: 1
OFFD: 1
OFFE: 1
OFFF: 1
State transition occurs according to the settings
LCDVCPWC pin
LCDDCK pin, LCDHSYN pin, LCDVSYN pin, LCDDISP pin, LCDDxx pin
Output OFF
Output ON
Output OFF
LCDVEPWC pin
LCDDON pin
Main LCD power management register (MLDPMR) LPS[1:0] bit
00
11
00

Figure 10 Transition Example of Power Management Pins

The delay between each signal is a number of frames equal to the value of the bits in MLDPMR for setting the sequence interval. For details, please refer to the section 31.3.26 "Main LCD Power Management Register (MLDPMR)" and the section 31.4.6 "Power Management Function" in the *SH7722 Group Hardware Manual* (REJ09B0314), or the section 27.3.26 "Main LCD Power Management Register (MLDPMR)" and the section 27.4.6 "Power Management Function" in the *SH7731 Group Hardware Manual* (REJ09B0488).

2.4 Activation and Termination of LCDC

Figure 11 shows a flowchart of activating the LCDC and starting display on the main LCD module.

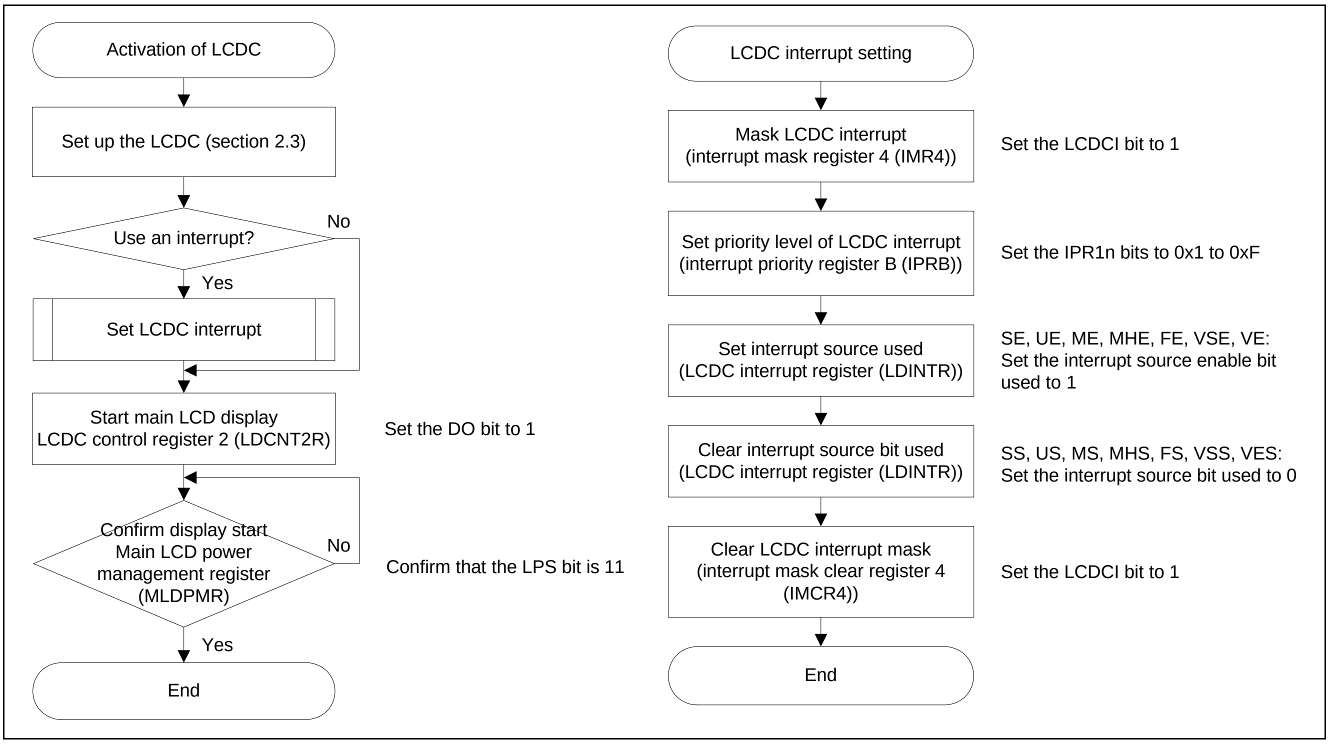

Activation of LCDC
Set up the LCDC (section 2.3)
Use an interrupt?
No
Yes
Set LCDC interrupt
Start main LCD display
LCDC control register 2 (LDCNT2R)
Set the DO bit to 1
Confirm display start
Main LCD power management register (MLDPMR)
No
Confirm that the LPS bit is 11
Yes
End
LCDC interrupt setting
Mask LCDC interrupt (interrupt mask register 4 (IMR4))
Set the LCDCI bit to 1
Set priority level of LCDC interrupt (interrupt priority register B (IPRB))
Set the IPR1n bits to 0x1 to 0xF
Set interrupt source used (LCDC interrupt register (LDINTR))
SE, UE, ME, MHE, FE, VSE, VE: Set the interrupt source enable bit used to 1
Clear interrupt source bit used (LCDC interrupt register (LDINTR))
SS, US, MS, MHS, FS, VSS, VES: Set the interrupt source bit used to 0
Clear LCDC interrupt mask (interrupt mask clear register 4 (IMCR4))
Set the LCDCI bit to 1
End

Figure 11 LCDC Activation Flow

Figure 12 shows a flowchart of terminating the LCDC and stopping display on the main LCD module.

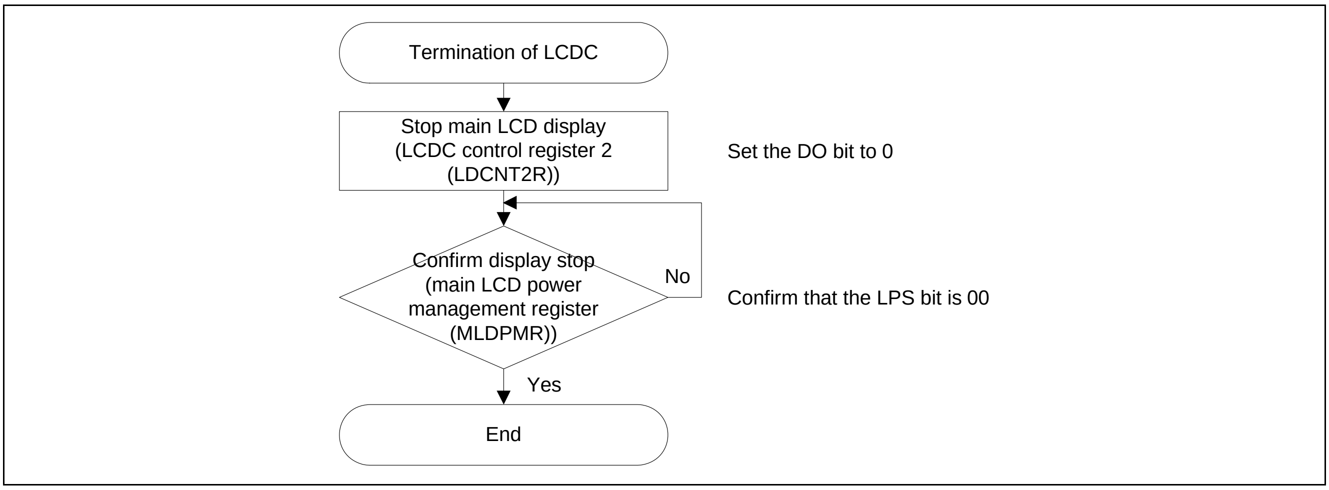

Termination of LCDC
Stop main LCD display (LCDC control register 2 (LDCNT2R))
Set the DO bit to 0
Confirm display stop (main LCD power management register (MLDPMR))
No
Confirm that the LPS bit is 00
Yes
End

Figure 12 LCDC Termination Flow

- LCDC setup which is described in section 2.3 can be made in a random order after the module standby state has been canceled. The setup can be started from any setting.
- When using an LCDC interrupt, start main LCD display after enabling interrupts to be generated. If generation of an LCDC interrupt is enabled (interrupt mask is cleared, etc.) after display has already been started, an interrupt may occur at an undesired timing.
- Starting or stopping of main LCD display is controlled by the DO bit in LCDC control register 2 (LDCNT2R). Since this application note only handles main LCD display, other display enable control bits should be handled as shown in table 13.
- Display start and display stop can be confirmed by the LPS bits in the main LCD power management register (MLDPMR), regardless of whether the power management function is used.

Table 13 Display Enable Bits at Main LCDC Control

LCDC Register Name	Display Enable	Remarks
LCDC control register 1 (LDCNT1R)	DE: Display Enable	Always set to 1.
LCDC control register 2 (LDCNT2R)	ME: Main LCD Enable	Always set to 1.
	SE: Sub LCD Enable	Always set to 0.
	DO: Display On	Set to 1 at display start and set to 0 at display stop.

2.5 LCDC Synchronous Design

If there is only one frame buffer when the image data to be displayed by the LCDC is updated, while the LCDC is reading data from the frame buffer, new frame information is drawn to that same frame buffer. This causes the image to be distorted, such as, screen tearing. The necessary operations to avoid this are to prepare several frame buffers, and change the LCDC reading destination upon completion of drawing to the frame buffer whose contents are to be displayed next. The LCDC facilitates frame buffer switching by having two planes for frame buffer information.

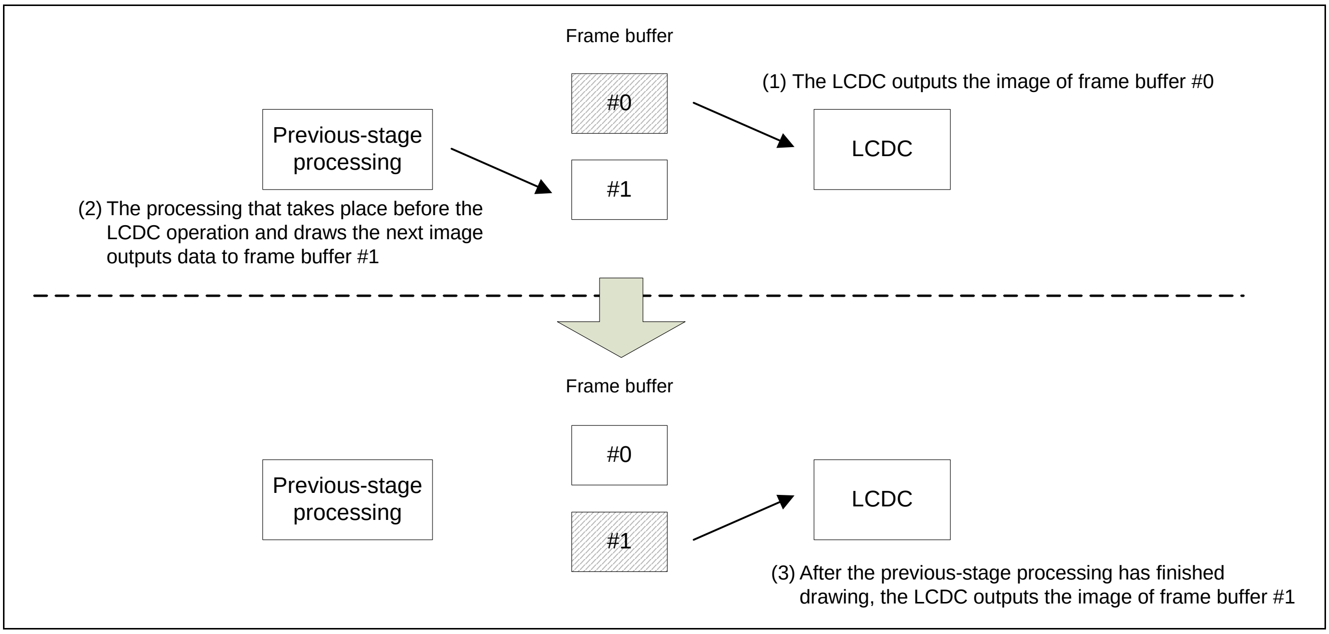

Frame buffer
#0
#1
Previous-stage processing
LCDC
(1) The LCDC outputs the image of frame buffer #0
(2) The processing that takes place before the LCDC operation and draws the next image outputs data to frame buffer #1
Frame buffer
#0
#1
Previous-stage processing
LCDC
(3) After the previous-stage processing has finished drawing, the LCDC outputs the image of frame buffer #1

Figure 13 Overview of Plane Switching Operation

2.5.1 Multiple Buffer Switching Using Register Planes A and B

The LCDC has frame buffer information for two planes: plane A and plane B. The register plane being displayed is called the used plane and the register plane not being displayed is called the unused plane. The information on the frame buffer to be displayed next is set to the unused plane, and upon completion of drawing to the frame buffer, the used plane and unused plane are switched and the new frame buffer is displayed.

For a register with planes A and B, the same value should be set for both planes A and B, except for the address registers (MLDSA1R and MLDSA2R). While the LCDC is activated, only the address register values can be changed.

Table 14 lists the registers which have planes A and B and are handled in this application note. When plane A is displayed, the LCDC operates with the plane A register settings, whereas when plane B is displayed, the LCDC operates with the plane B register settings.

A mirror address can be used to set an unused-plane register. All registers with two planes A and B have mirror addresses, and reading from and writing to a mirror address results in reading from and writing to an unused plane.

Table 14 Registers Having Planes A and B and Appearing in this Application Note

LCDC Display Enable	Remarks
Main LCD display data read start address register 1 (MLDSA1R)	—
Main LCD display data read start address register 2 (MLDSA2R)	—
Main LCD module type register 1 (MLDMT1R)	The same value should be set in planes A and B.
Main LCD data format register (MLDDFR)	The same value should be set in planes A and B.
Main LCD display data storing memory line size register (MLDMLSR)	The same value should be set in planes A and B.
Main LCD horizontal character number register (MLDHCNR)	The same value should be set in planes A and B.
Main LCD horizontal sync signal register (MLDHSYNR)	The same value should be set in planes A and B.
Main LCD vertical line number register (MLDVLNR)	The same value should be set in planes A and B.
Main LCD vertical sync signal register (MLDVSYNR)	The same value should be set in planes A and B.

Note: Though the LCDC input image data swap register (LDDDSR) is stated to have planes A and B in table 31.3, Register Configuration, in section 31, LCD Controller (LCDC), in the "*SH7722 Group Hardware Manual* (REJ09B0314-0200) Rev. 2.00", or in table 27.3, Register Configuration, in section 27, LCD Controller (LCDC), in the "*SH7731 Group Hardware Manual* (REJ09B0488-0100) Rev. 1.00", this is an error.

Plane Switching Method

The register planes can be manually switched by software and automatically switched by hardware.

- Software switching: When the MRS bit in the LCDC register side change control register (LDRCNTR) is inverted from 0 to 1 or from 1 to 0, planes A and B are switched at the same time the next frame display operation is finished (a frame end interrupt (FE) occurs simultaneously with the frame display operation end). When the MRS bit is changed from 0 to 1, plane A is switched to plane B, and when the MRS bit is changed from 1 to 0, plane B is switched to plane A. At this time, the MRC bit in LDRCNTR must be set to 0 to disable hardware switching.

  Figure 14 shows the timing chart for hardware switching. Shown here is an example of flag usage by software in order to achieve synchronization with the previous-stage processing.

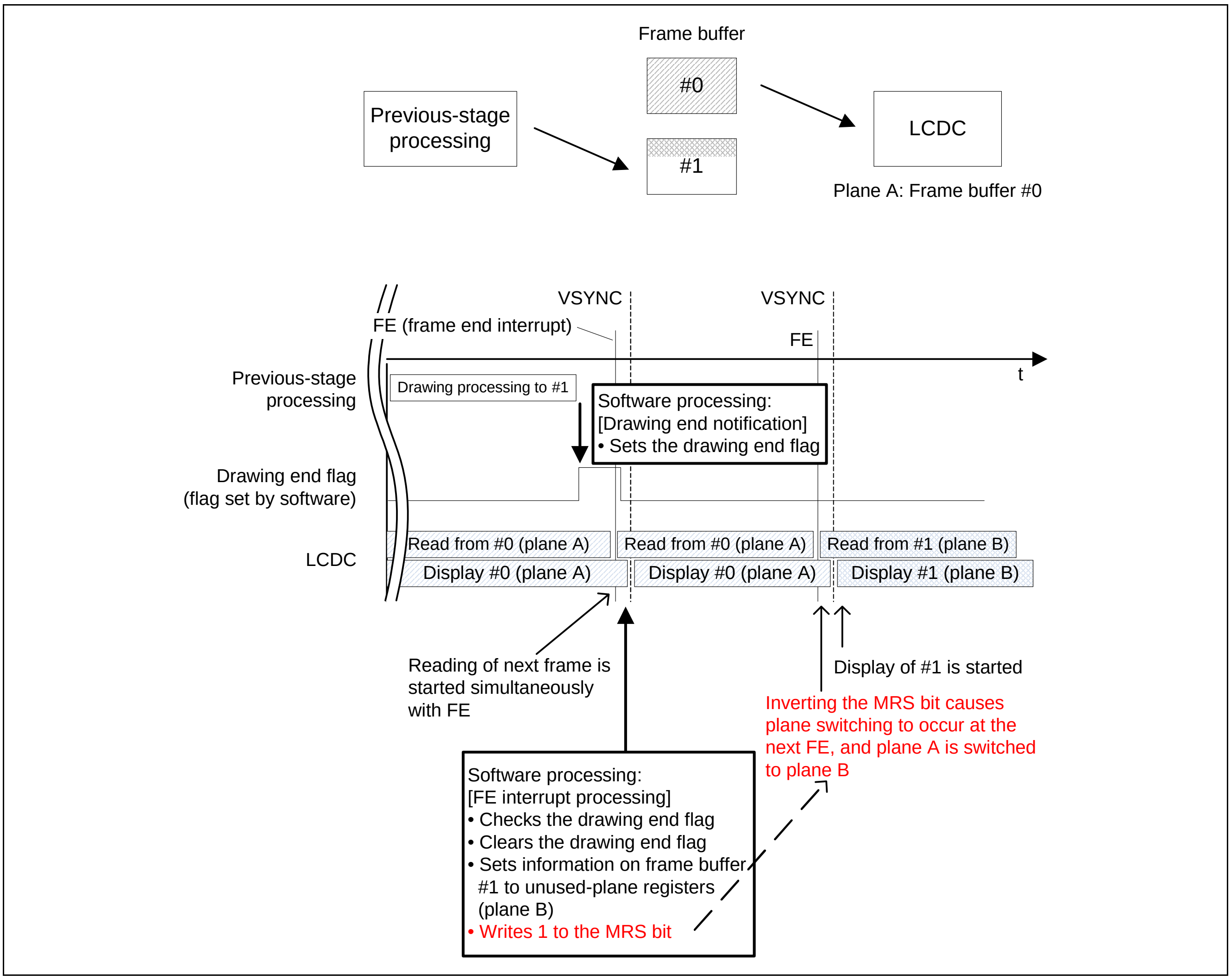

Frame buffer
#0
#1
Previous-stage processing
LCDC
Plane A: Frame buffer #0
VSYNC
VSYNC
FE (frame end interrupt)
FE
t
Previous-stage processing
Drawing processing to #1
Software processing:
[Drawing end notification]
• Sets the drawing end flag
Drawing end flag
(flag set by software)
LCDC
Read from #0 (plane A)
Read from #0 (plane A)
Read from #1 (plane B)
Display #0 (plane A)
Display #0 (plane A)
Display #1 (plane B)
Reading of next frame is started simultaneously with FE
Display of #1 is started
Inverting the MRS bit causes plane switching to occur at the next FE, and plane A is switched to plane B
Software processing:
[FE interrupt processing]
• Checks the drawing end flag
• Clears the drawing end flag
• Sets information on frame buffer #1 to unused-plane registers (plane B)
• Writes 1 to the MRS bit

Figure 14 Timing of Software Switching Occurrence

The drawing end flag is set when the previous-stage processing has completed drawing. The LCDC monitors the drawing end flag by the frame end interrupt processing. If the flag is set, information on the frame buffer displayed next is set to the unused-plane registers and the MRS bit is inverted. Register plane switching occurs simultaneously with the next frame end interrupt, and the image displayed by the LCDC is updated.

- Hardware switching: When the MRC bit in the LCDC register side change control register (LDRCNTR) is set to 1, the LCDC automatically switches planes A and B each time the frame display operation finishes (a frame end interrupt (FE) occurs simultaneously with the frame display operation end). At this time, the MRS bit setting is invalidated.

  Figure 15 shows the timing chart for hardware switching.

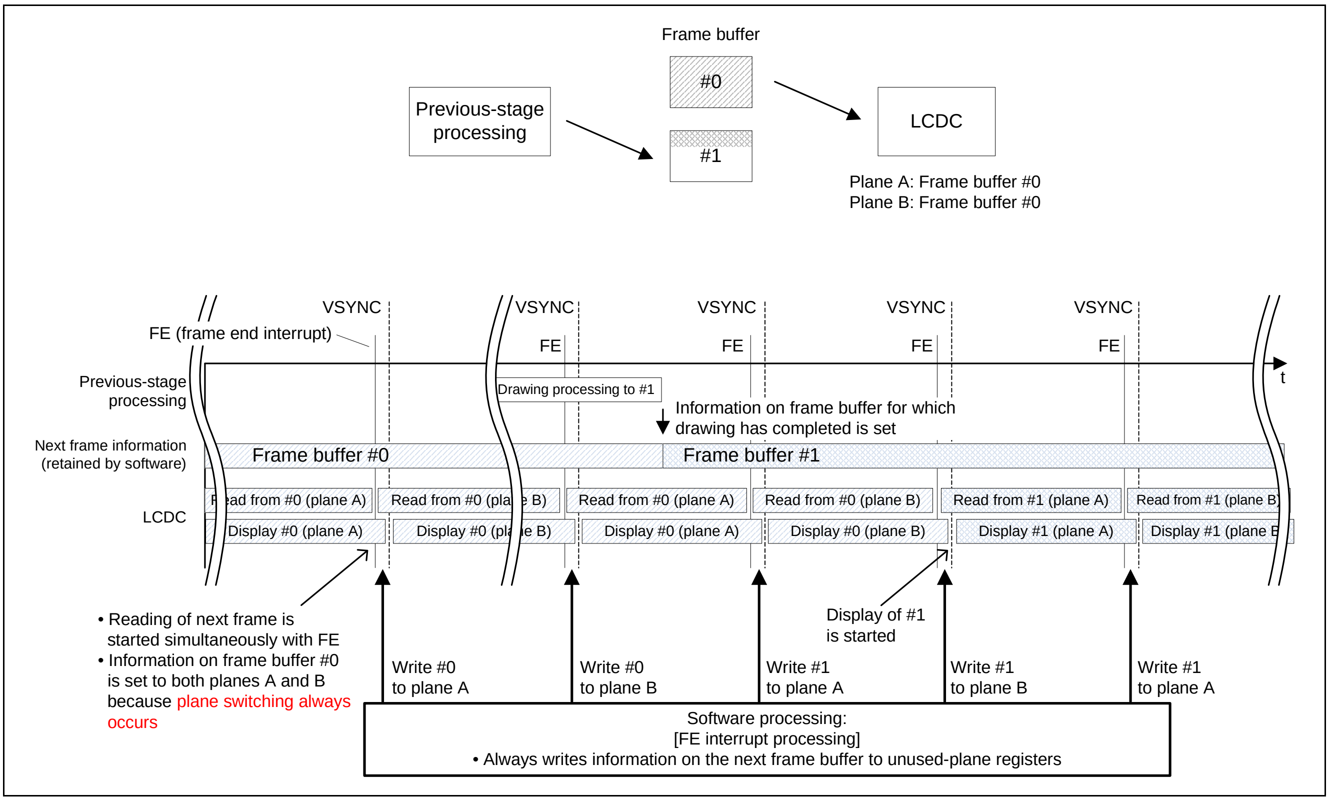


Figure 15 Timing of Hardware Switching Occurrence

Since the planes are switched automatically at the same time a frame end interrupt (FE) occurs in hardware switching, when drawing to the new frame buffer has not completed, both planes A and B need the same frame information. Therefore, in the operation example shown, the frame buffer information that should be displayed next is retained, and the next frame buffer information is set to the unused-plane registers at every frame end interrupt. The same frame buffer information is set to planes A and B in the initial state, and information on the currently displayed frame is set to the unused plane if the frames are not updated.

After the LCDC is activated, the LCDC displays frame buffer #0 while automatically switching planes A and B until the previous-stage processing has completed drawing. After drawing has finished, the previous-stage processing updates the next frame information. After update, the LCDC sets the new frame information to the unused-plane registers by the frame end interrupt processing, and the image to be displayed simultaneously with the next frame end interrupt is updated.

With either switching method, register planes are switched at the same time a frame end interrupt is generated. Accordingly, when the user needs to change the settings of the unused plane, to perform it in the frame end interrupt processing is the best timing for software.

2.5.2 Synchronous Design Example of Asynchronous Previous-Stage Processing and Multiple Buffers of LCDC

This section shows an operation example of a system in which an image is updated asynchronously with VSYNC of the LCDC by the previous-stage processing and that image is displayed by the LCDC, such as captured images or movie of a camera. Three planes are allocated for the frame buffers of the LCDC, and planes are switched by software switching. The LCDC refresh rate is assumed to be 60 Hz, and the frame rate of the previous-stage processing is assumed to be approximately 30 fps.

Information on frame buffer #0 is set to plane A. In the frame end interrupt processing, the drawing end flag of the previous-stage processing is checked. If the next frame buffer can be displayed, information on the next frame buffer #1 is set to the mirror address and the MRS bit in the LCDC register side change control register (LDRCNTR) is inverted. Hereafter, the drawing end flag of the previous-stage processing is checked in the frame end interrupt processing. Depending on the result, the mirror address is set to the register and the MRS bit is inverted, thus causing the planes to be switched.

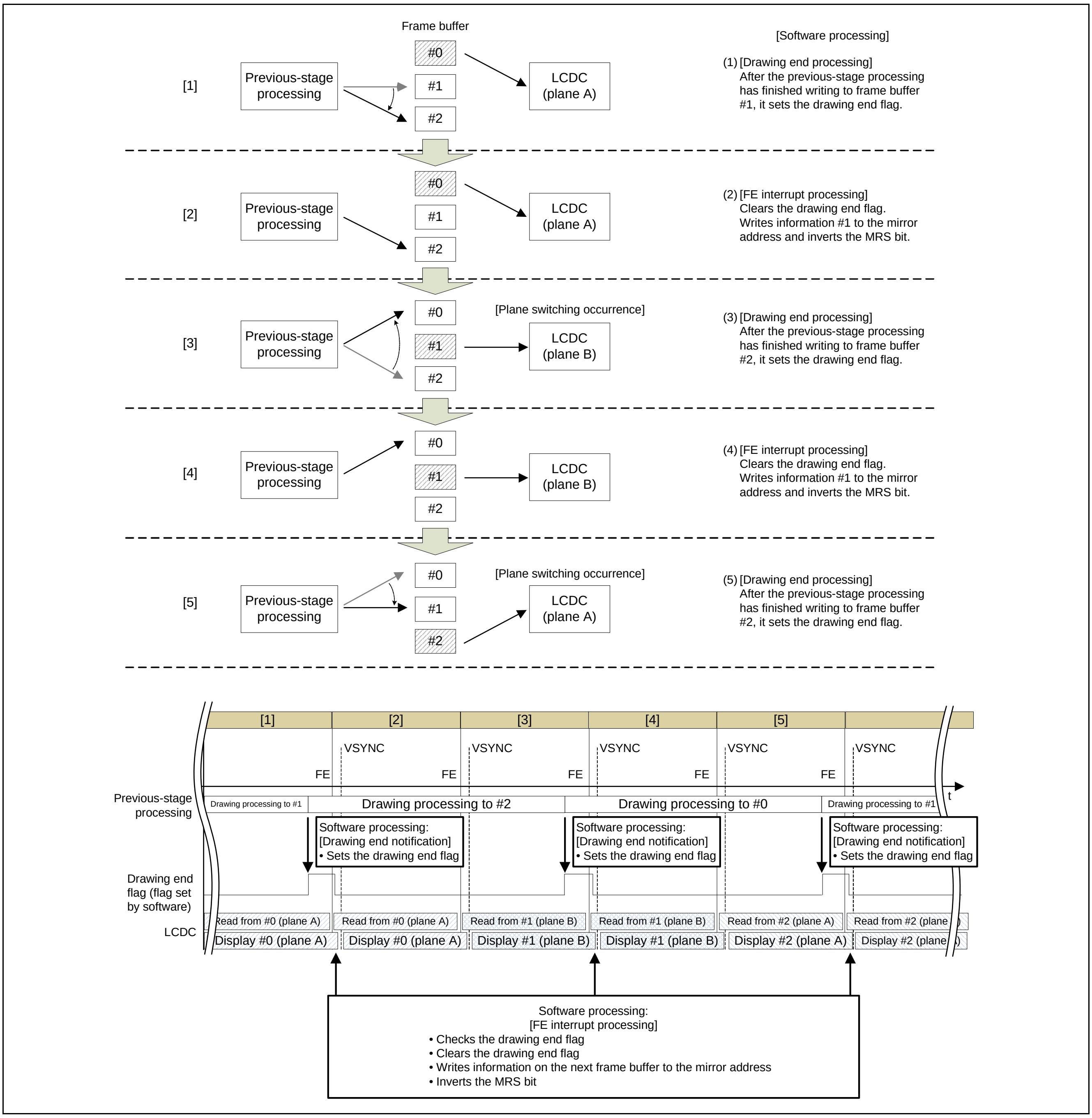

Frame buffer
[Software processing]
[1]
Previous-stage processing
#0
#1
#2
LCDC (plane A)
(1) [Drawing end processing]
After the previous-stage processing has finished writing to frame buffer #1, it sets the drawing end flag.
[2]
Previous-stage processing
#0
#1
#2
LCDC (plane A)
(2) [FE interrupt processing]
Clears the drawing end flag.
Writes information #1 to the mirror address and inverts the MRS bit.
[3]
Previous-stage processing
#0
#1
#2
[Plane switching occurrence]
LCDC (plane B)
(3) [Drawing end processing]
After the previous-stage processing has finished writing to frame buffer #2, it sets the drawing end flag.
[4]
Previous-stage processing
#0
#1
#2
LCDC (plane B)
(4) [FE interrupt processing]
Clears the drawing end flag.
Writes information #1 to the mirror address and inverts the MRS bit.
[5]
Previous-stage processing
#0
#1
#2
[Plane switching occurrence]
LCDC (plane A)
(5) [Drawing end processing]
After the previous-stage processing has finished writing to frame buffer #2, it sets the drawing end flag.
[1]
[2]
[3]
[4]
[5]
VSYNC
VSYNC
VSYNC
VSYNC
VSYNC
FE
FE
FE
FE
FE
t
Previous-stage processing
Drawing processing to #1
Drawing processing to #2
Drawing processing to #0
Drawing processing to #1
Software processing:
[Drawing end notification]
• Sets the drawing end flag
Software processing:
[Drawing end notification]
• Sets the drawing end flag
Software processing:
[Drawing end notification]
• Sets the drawing end flag
Drawing end flag (flag set by software)
LCDC
Read from #0 (plane A)
Read from #0 (plane A)
Read from #1 (plane B)
Read from #1 (plane B)
Read from #2 (plane A)
Read from #2 (plane A)
Display #0 (plane A)
Display #0 (plane A)
Display #1 (plane B)
Display #1 (plane B)
Display #2 (plane A)
Display #2 (plane A)
Software processing:
[FE interrupt processing]
• Checks the drawing end flag
• Clears the drawing end flag
• Writes information on the next frame buffer to the mirror address
• Inverts the MRS bit

Figure 16 Operation Example of Movie Display System Using Three Frame Buffers

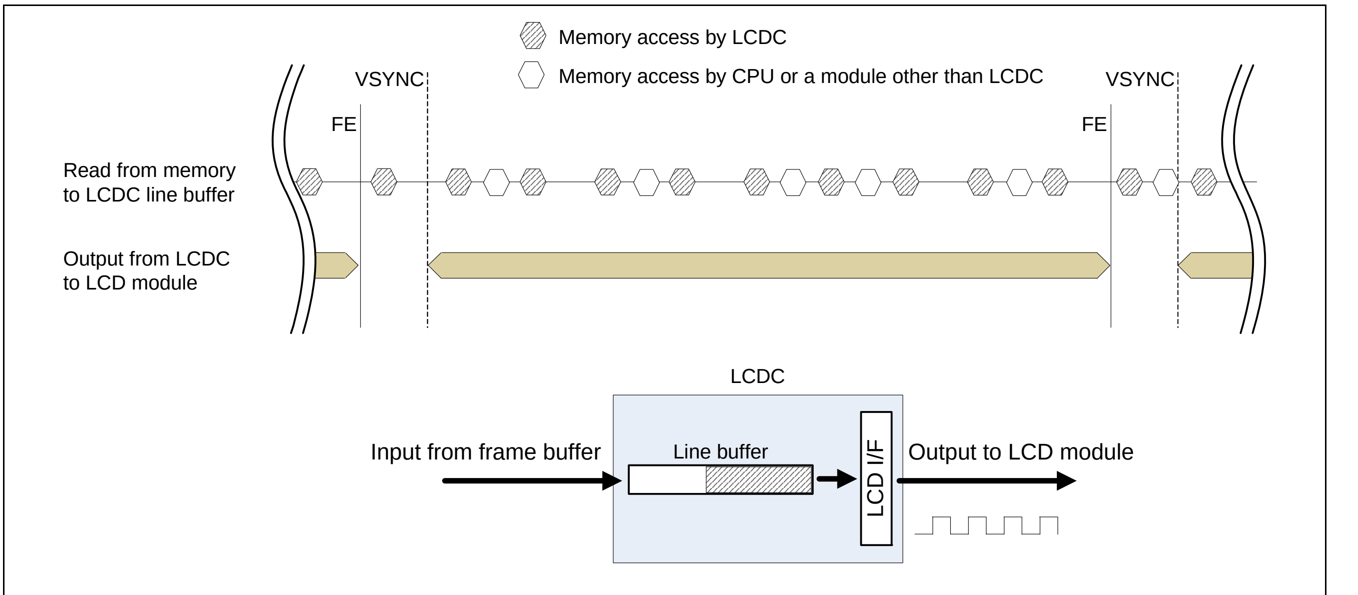
3. Estimating Load on the Bus in System Design

As the SH7722 has an architecture where the video and audio modules and the CPU share the external memory through the SHwy bus, the system design should estimate the load on the bus and ensure that the total load from all modules does not exceed the transfer bandwidth for the memory.

3.1 Bus Occupancy by LCDC

The LCDC provides a function for displaying up to the maximum displayable size by LCD module as shown in table 1. However, the images to be displayed are stored in the external memory connected through the SHwy bus, which is shared with the CPU, so memory access contention occurs among the LCDC, CPU, and other modules. The LCDC should read data from the external memory without delay so that every image is displayed in time. If read access is delayed, the LCDC buffer will underflow and image display will be disturbed.

The LCDC has a line buffer; as soon as display of a frame is completed, the LCDC starts prefetching the next frame. Even during the blanking period, it can read data from memory into the line buffer. Figure 17 shows the best case where the LCDC constantly reads frame buffer data from memory without interruption by other memory access.


Memory access by LCDC
Memory access by CPU or a module other than LCDC
VSYNC
FE
VSYNC
FE
Read from memory to LCDC line buffer
Output from LCDC to LCD module
LCDC
Input from frame buffer
Line buffer
LCD I/F
Output to LCD module

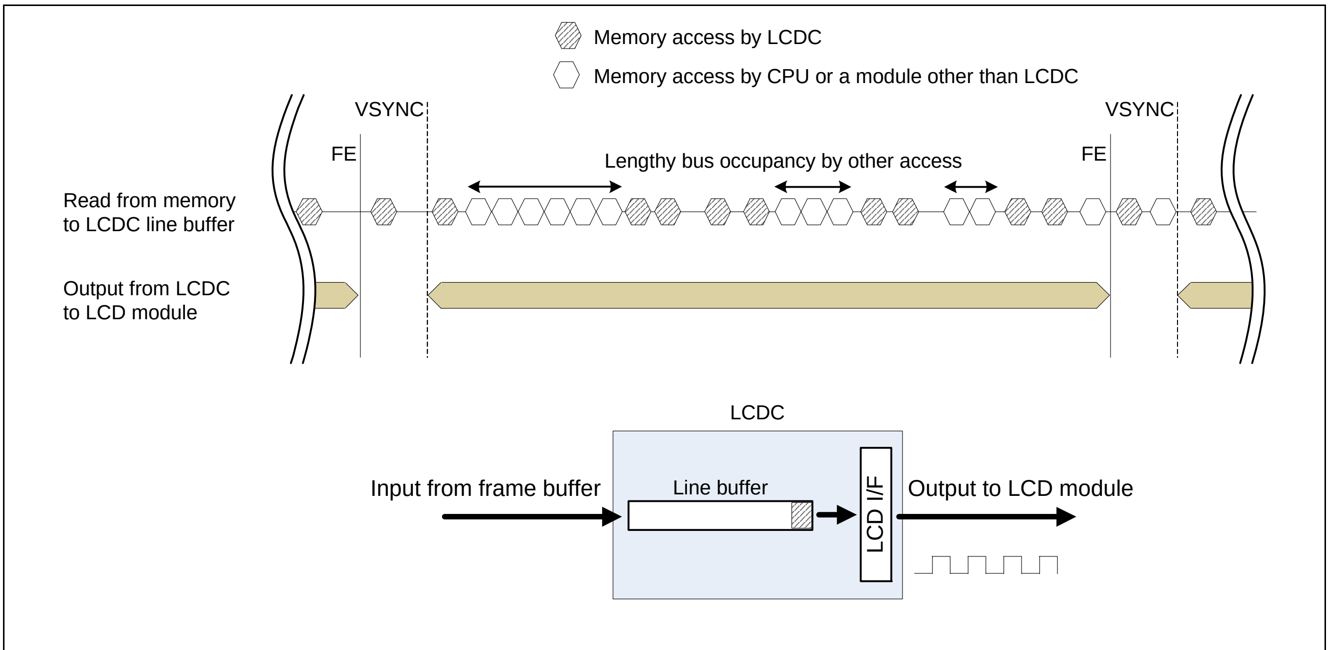
(a) Best case where one-frame data can be read at a constant rate

Figure 17 Best Case for Frame Buffer Reading

The bus bandwidth (byte/s) required by the LCDC in the best case is calculated as follows:

Horizontal image size [pixels] × vertical image size [pixels] × input pixel size [byte/pixel] × refresh rate [fps]

In actual systems, there will always be memory access contention among the LCDC, CPU, and other modules and the best case cannot be achieved (figure 18).


Memory access by LCDC
Memory access by CPU or a module other than LCDC
VSYNC
FE
Lengthy bus occupancy by other access
VSYNC
FE
Read from memory to LCDC line buffer
Output from LCDC to LCD module
LCDC
Input from frame buffer
Line buffer
LCD I/F
Output to LCD module

(b) Case where contention with other memory access occurs and data cannot be read at a constant rate

Figure 18 Frame Buffer Reading in Actual Systems

In the worst case, the LCDC requires enough bus bandwidth to complete reading the data that should be output in one dot-clock period from memory within one dot-clock period. This is the maximum bus bandwidth required by the LCDC.

The rate (byte/s) of image data output from the LCDC is calculated as follows.

Rate of image data output from LCDC [byte/s] =
LCDC dot-clock frequency × (output pixel size / number of transfer cycles *1)

Note: *1. Number of transfer cycles:
This is the amount of transfers per cycle determined by the main LCD module setting in section 2.3.8.
For example, when MIFTYP[3:0] = b'0000 in MLDMT1R, three-time transfer at 24 bpp is selected. Therefore, (output pixel size / number of transfer cycles) = (24 / 8) / 3 = 1 [byte/cycle]. When MIFTYP[3:0] = b'1011, which selects one-time transfer at 24 bpp, (output pixel size / number of transfer cycles) = (24 / 8) / 1 = 3 [byte/cycle].

The maximum bus bandwidth required by the LCDC can be calculated by replacing the output pixel size with the input pixel size in the above equation (according to note 3 in section 2.3.4, when input pixel size ≠ output pixel size, the LCDC pads the lower bits with 0 or discards the lower bits to convert the input pixels into the output pixels).

Rate of image data input to LCDC [byte/s] =
LCDC dot-clock frequency × (input pixel size / number of transfer cycles)

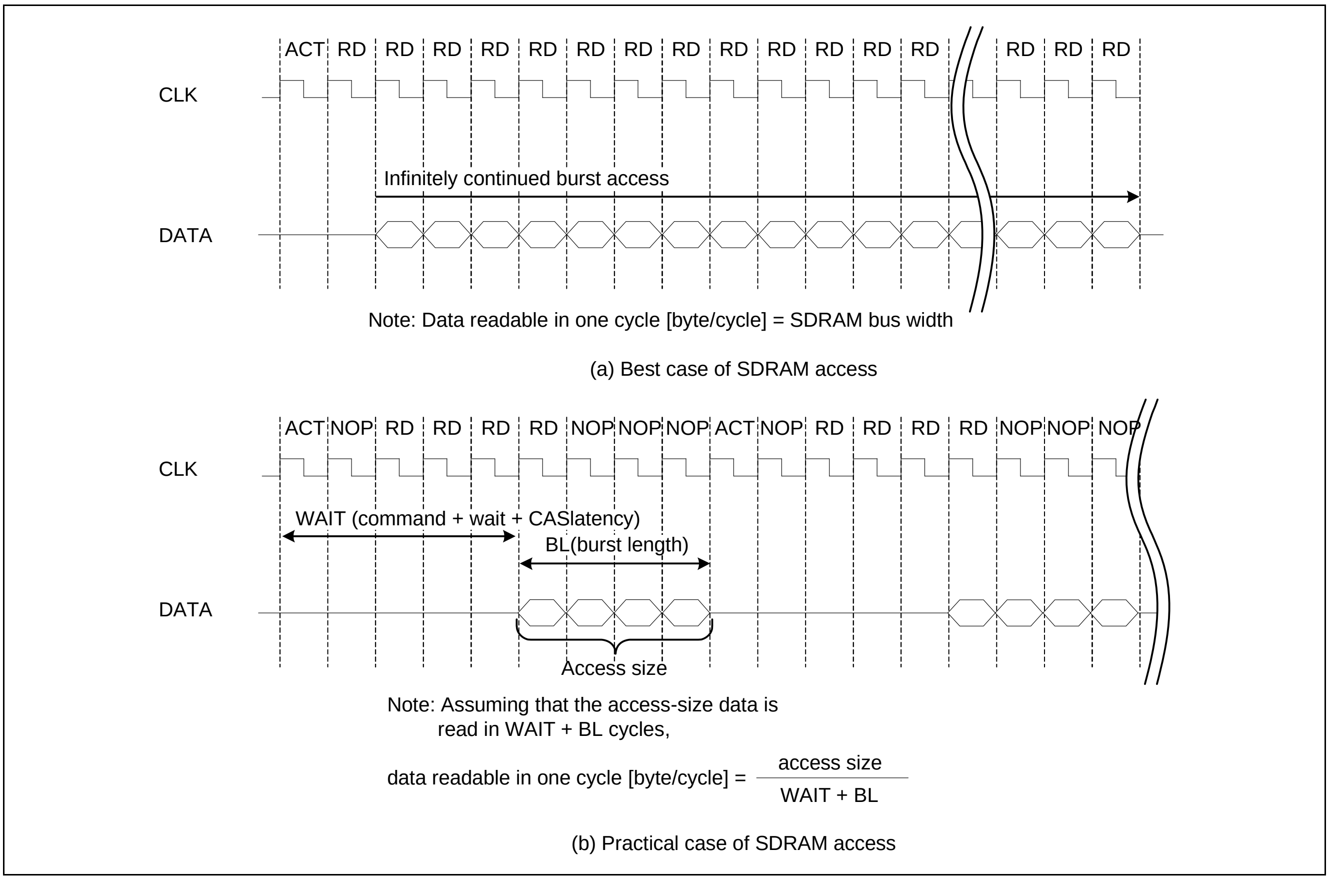
When estimating the load on the bus from the LCDC, assume the worst case and control any other memory access or bus occupancy so that the maximum bus bandwidth requirement is satisfied.

3.2 Effective Bandwidth of SDRAM

As the SH7722/SH7731 internal bus (SHwy bus) has enough transfer bandwidth relative to the external memory, the transfer bandwidth of the external memory will be a bottleneck in a system. When frame buffers are allocated in the SDR-SDRAM (SDRAM), the peak bandwidth of the SDRAM is calculated as follows.

SDRAM peak bandwidth = SDRAM frequency [MHz] × SDRAM bus width [byte/cycle]

In actual applications such as video systems, multiple memory blocks are accessed at the same time. Accordingly, the SDRAM command sequence and CAS latency will be practical matters of concern because they cause access wait.


ACT RD RD RD RD RD RD RD RD RD RD RD RD RD RD RD RD
CLK
Infinitely continued burst access
DATA
ACT NOP RD RD RD RD NOP NOP NOP ACT NOP RD RD RD RD NOP NOP NOP
CLK
WAIT (command + wait + CASlatency)
BL(burst length)
DATA
Access size

Note: Data readable in one cycle [byte/cycle] = SDRAM bus width

(a) Best case of SDRAM access

Note: Assuming that the access-size data is read in WAIT + BL cycles,

$$\text{data readable in one cycle [byte/cycle]} = \frac{\text{access size}}{\text{WAIT + BL}}$$

(b) Practical case of SDRAM access

Figure 19 Examples of SDRAM Access

For example, an estimate of the effective SDRAM bandwidth is shown below assuming the following conditions.

- SDRAM frequency: 106.66 MHz
- SDRAM bus width: 32-bit
- Practical number of access wait cycles (WAIT): 5
  (Estimated by considering the command issuing and CAS latency and assuming that the SBSC is set to bank active mode and the burst count is set to 4)
- Access size: 16-byte
  (Most of the access from the LCDC is 32-byte burst access, but 16-, 8-, or 4-byte burst access also occurs. In this calculation example, 16-byte burst access is used as an average of the access sizes.)
- Number of Bursts (BN): 4

Effective SDRAM bandwidth = SDRAM frequency [MHz] × (data readable in one cycle) [byte/cycle]
= SDRAM frequency [MHz] × access size [byte] / (WAIT + BN) [cycle]
= 188 Mbytes/s

In the earlier stages of design, the target ratio of the load on the bus from the video and audio modules should be 60% or less of the effective bandwidth.

Effective SDRAM bandwidth × 60% = 190 Mbytes/s × 0.6 = 113 Mbytes/s

3.3 Sample Calculation of Load on the Bus from LCDC

When the LCDC displays 640 × 480 RGB565 images on a VGA-size LCD module with the 18-bpp (RGB666) bus width in one-time transfer mode at a 25-MHz dot clock and a 60-Hz refresh rate, the maximum instantaneous data rate required by the LCDC is calculated as follows.

25 [MHz] × (16 / 8) [Byte] / 1 [time transfer] = 50.0 [Mbytes/s]

This rate is faster than the target rate (effective SDRAM bandwidth × 60% = 113 Mbytes/s) in section 3.2.

4. Description of the Sample Application

This section describes an example of pin connection and various settings as a sample application that uses the LCDC to display graphic images.

4.1 TFT-LCD Panel Specifications

The following shows the specifications of the LS037V7DW01 TFT-LCD panel (manufactured by Sharp Corporation) used in this sample application. As the details of the specifications differ depending on the TFT-LCD panel actually used in your system, please refer to the datasheet for the target LCD panel.

4.1.1 General Specifications

Table 15 lists the general specifications of the TFT-LCD panel used in this application.

Table 15 TFT-LCD Panel General Specifications (Excerpt from Datasheet)

Item	Specifications
Resolution	VGA or QVGA (VGA is used in this application)
Pixel format	H 480 × V 640 (Number of dots: H (480 × 3) × V640) * 1 pixel = R + G + B dots
Pixel configuration	R, G, B vertical stripe
Input signal	CMOS RGB (6 bits each, digital)

4.1.2 Pin Functions

Table 16 lists the pin functions of the TFT-LCD panel used in this application.

Table 16 TFT-LCD Panel Pin Functions (Excerpt from Datasheet)

Symbol	Description
RESB	Reset signal
INI	Power on control
DEN	Data enable signal
HSYNC	Horizontal synchronizing signal
VSYNC	Vertical synchronizing signal
CLKIN	System clock signal
R5-0	Red data signals (6 bits, MSB: R5, LSB: R0)
G5-0	Green data signals (6 bits, MSB: G5, LSB: G0)
B5-0	Blue data signals (6 bits, MSB: B5, LSB: B0)

4.1.3 Interface Timing

Table 17 lists the interface timing and characteristics of the TFT-LCD panel used in this application.

Table 17 TFT-LCD Panel Timing Characteristics (Excerpt from Datasheet)

Item		MODE	Symbol	Min.	Typ.	Max.	Unit
CLK	Cycle time	VGA	t_{CLK}	38	39.7	41.7	ns
		QVGA		152	158.8	167	
HSYNC	Cycle time	VGA	t_{HS}	—	648	—	CLK
		QVGA		—	324	—	
	Horizontal back porch	VGA	t_{HBP}	28	78	166	
		QVGA		14	38	82	
	Horizontal front porch	VGA	t_{HFP}	0	88	138	
		QVGA		0	44	68	
	Valid display period	VGA	t_{HHW}	—	480	—	
		QVGA		—	240	—	
	Sync signal width		t_{HSW}	—	2	—	
VSYNC	Cycle time	VGA	t_{VS}	—	648	—	Hsync
		QVGA		—	324	—	
	Vertical back porch	VGA	—	—	(1)	—	
		QVGA		—	(1)	—	
	Vertical front porch	VGA	—	—	(6)	—	
		QVGA		—	(2)	—	
	Valid display period	VGA	—	—	640	—	
		QVGA		—	320	—	
	Sync signal width		t_{VSW}	—	1	—	

Note: The vertical back porch and vertical front porch times are estimated from the timing chart because they are not specified in the AC characteristics table.

Table 18 shows an example of LCDC timing settings for VGA display calculated from the characteristics listed in table 17.

Table 18 Example of VGA Timing Settings

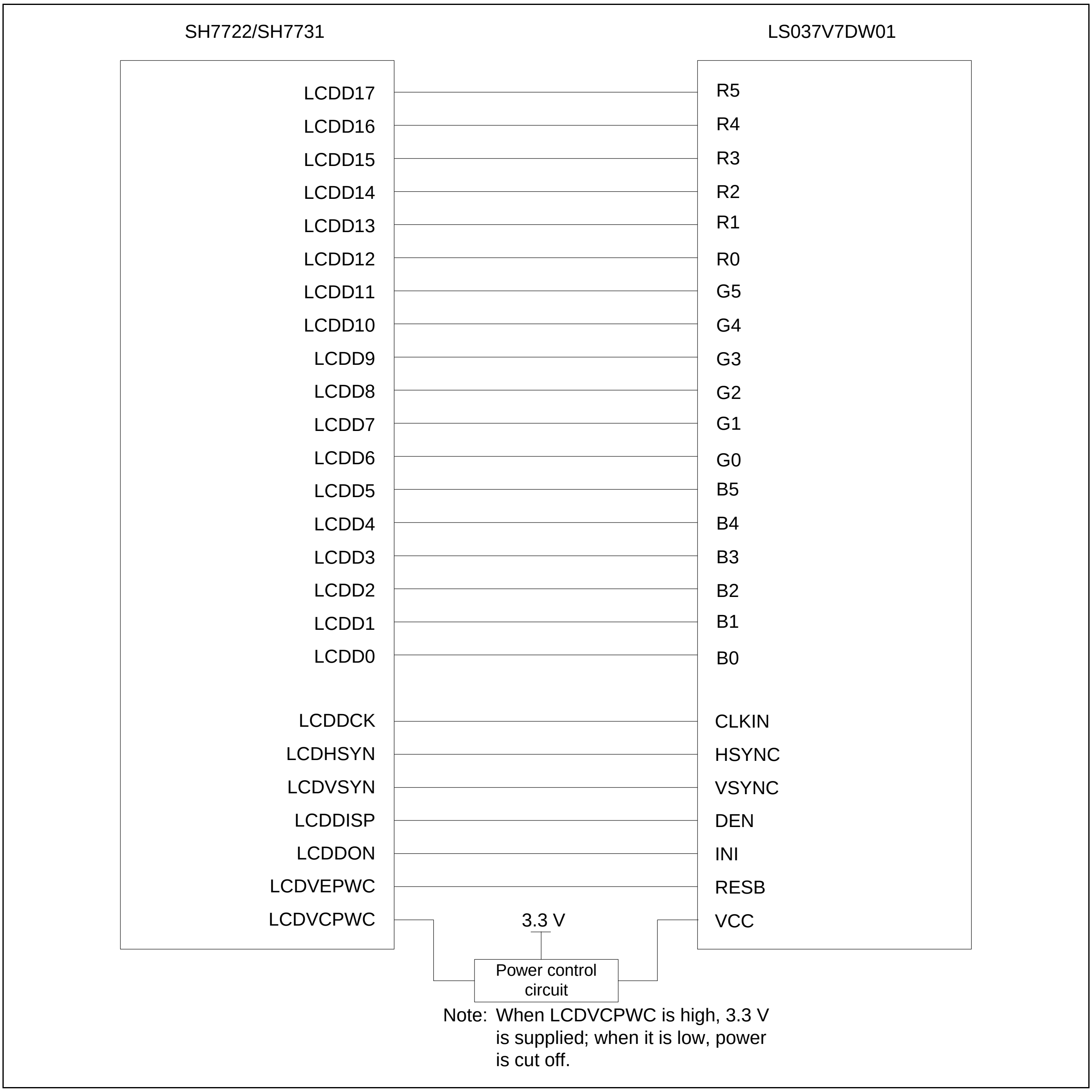
Function	Register	Bit Name	Number of Dot Clock	Setting Value Example
Horizontal timing settings	Main LCD horizontal character number register (MLDHCNR)	HTCN: 1/8 of horizontal cycle	648/8 = 81	0x51
		HDCN: 1/8 of horizontal display period	480/8 = 60	0x3C
	Main LCD horizontal sync signal register (MLDHSYNR)	HSYNW: 1/8 of horizontal sync signal width	2/8 = 1	0x01
		HSYNP: 1/8 of (horizontal display period + horizontal front porch)	568/8 = 71	0x47
Vertical timing settings	Main LCD vertical line number register (MLDVLNR)	VTLN: Vertical cycle	648	0x288
		VDLN: Vertical display period	640	0x280
	Main LCD vertical sync signal register (MLDVSYNR)	VSYNW: Vertical sync signal width	1	0x01
		VSYNP: Vertical display period + vertical front porch	646	0x286

Although the horizontal sync signal width is two clock cycles according to the datasheet, the minimum horizontal sync signal width that can be specified in the LCDC is eight clock cycles (one character). Therefore, the horizontal back porch time is set to 72 clock cycles, which is six clock cycles shorter than the standard 78 clock cycles.

4.2 TFT-LCD Panel Connection Circuit Example

4.2.1 Pin Connection Example

Figure 20 shows the TFT-LCD panel connection circuit example in this application.



Note: When LCDVCPWC is high, 3.3 V is supplied; when it is low, power is cut off.

Figure 20 TFT-LCD Panel Connection Circuit Example

4.2.2 Clock Setting

The TFT-LCD panel in this application needs 25-MHz dot clock supply when operating in VGA mode. As the SH7722 reference platform does not supply an external clock to the LCDLCLK pin, a 25-MHz clock should be generated from Bϕ or Pϕ. Under the conditions described in the "SH7722/SH7731 Group Application Note: SH7722/SH7731 Example of Initialization (REJ06B0942)", Bϕ = 66 MHz and Pϕ = 33 MHz, which cannot generate 25 MHz even after being divided. To generate a 25-MHz clock, this application example sets the operating frequencies as shown in section 1.3, Applicable Conditions, and selects Pϕ as the source for the dot clock.

4.2.3 Power Management Setting

The LCDC power management function is specified as follows according to the power control timing requirements of the TFT-LCD panel used in this application.

To Turn on the Power:

Requirement (1): VCC (+3.3 V) should settle within a 2-frame period:
→ Power-on sequence period A (ONA) is set to 2 to ensure two frames for the VCC settling period.

Requirement (2): The sync signals and RGB data should be output before INI is driven high:
→ Power-on sequence period B (ONB) + power-on sequence period C (ONC) should be 1 or larger. In this example, ONB is set to 1.

Requirement (3): The RESB signal should be kept low for at least 20 μs after VCC (+3.3 V) has settled:
→ ONB is set to 1.

Requirement (4): RESB should be driven high before INI:
→ ONC is set to 1.

To Turn Off the Power

Requirement (5): The sync signal and RGB data output should be stopped and RESB should be driven low after 5-frame period has passed since INI is driven low:
Power-off sequence period A (OFFD) is set to 5.

Requirement (6): Power-off sequence period B (OFFE) and power-off sequence period C (OFFF) are set to 1 in this application although the TFT-LCD panel specifications do not prescribe such requirements.

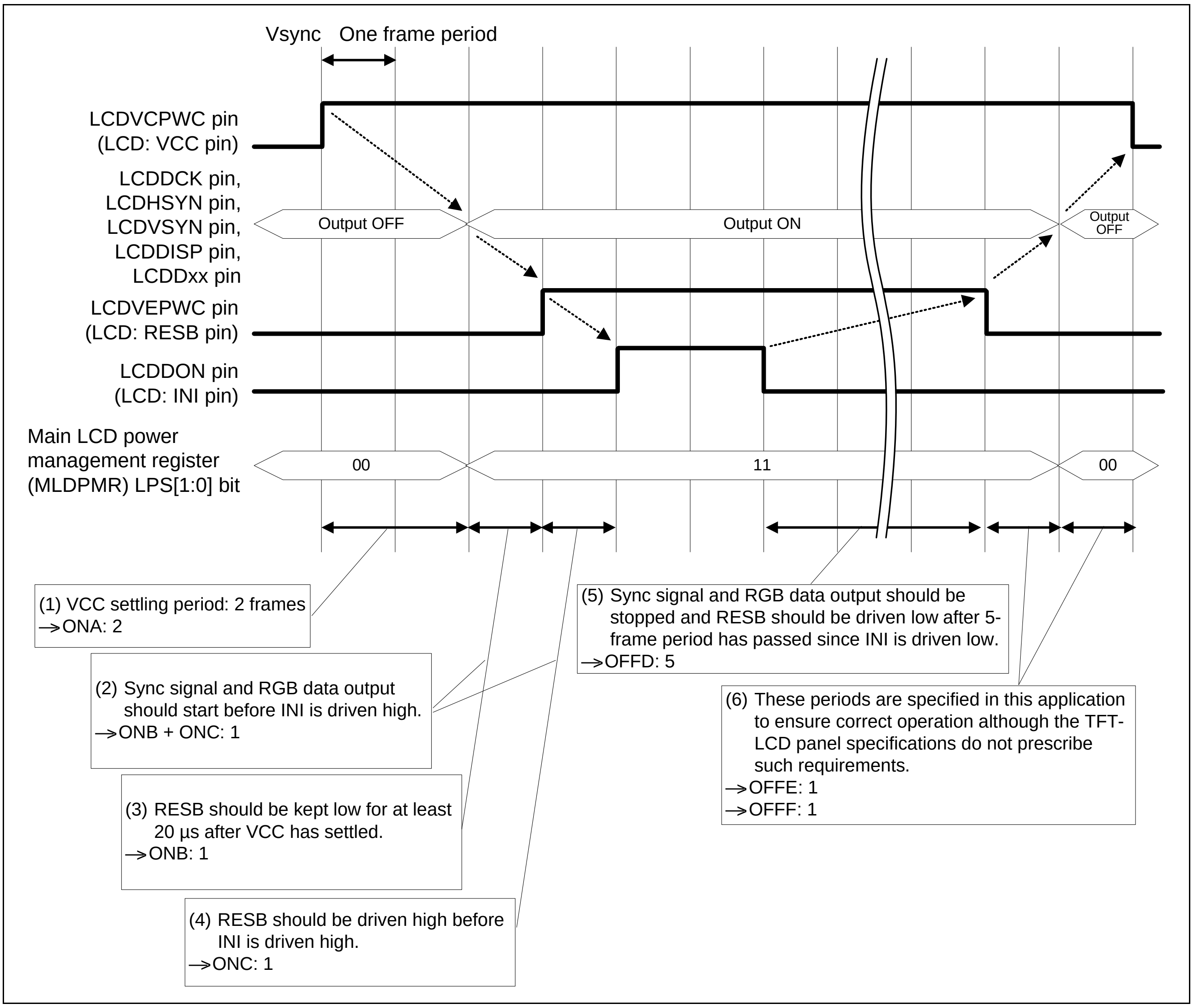

Vsync
One frame period
LCDVCPWC pin
(LCD: VCC pin)
LCDDCK pin,
LCDHSYN pin,
LCDVSYN pin,
LCDDISP pin,
LCDDxx pin
Output OFF
Output ON
Output OFF
LCDVEPWC pin
(LCD: RESB pin)
LCDDON pin
(LCD: INI pin)
Main LCD power
management register
(MLDPMR) LPS[1:0] bit
00
11
00
(1) VCC settling period: 2 frames
→ONA: 2
(2) Sync signal and RGB data output should start before INI is driven high.
→ONB + ONC: 1
(3) RESB should be kept low for at least 20 µs after VCC has settled.
→ONB: 1
(4) RESB should be driven high before INI is driven high.
→ONC: 1
(5) Sync signal and RGB data output should be stopped and RESB should be driven low after 5-frame period has passed since INI is driven low.
→OFFD: 5
(6) These periods are specified in this application to ensure correct operation although the TFT-LCD panel specifications do not prescribe such requirements.
→OFFE: 1
→OFFF: 1

Figure 21 Example of Power Management Setting

4.3 Sample Program Specifications

This section describes the specifications of the sample program and shows the flowchart of each processing.

4.3.1 Specifications

(1) Displays in turn four gradation images (gray-scale, red, green, and blue) drawn by the CPU on the VGA portrait (W480 × H640) TFT-LCD panel.

(2)Activates the LCDC, displays the images in turn with switching among four frame buffers, and then stops the LCDC.

(3) Repeats step (2) infinitely.

(4) Switches the planes by software.

4.3.2 Main Flowchart of the Sample Program

Figure 22 shows the main flowchart of the sample program.

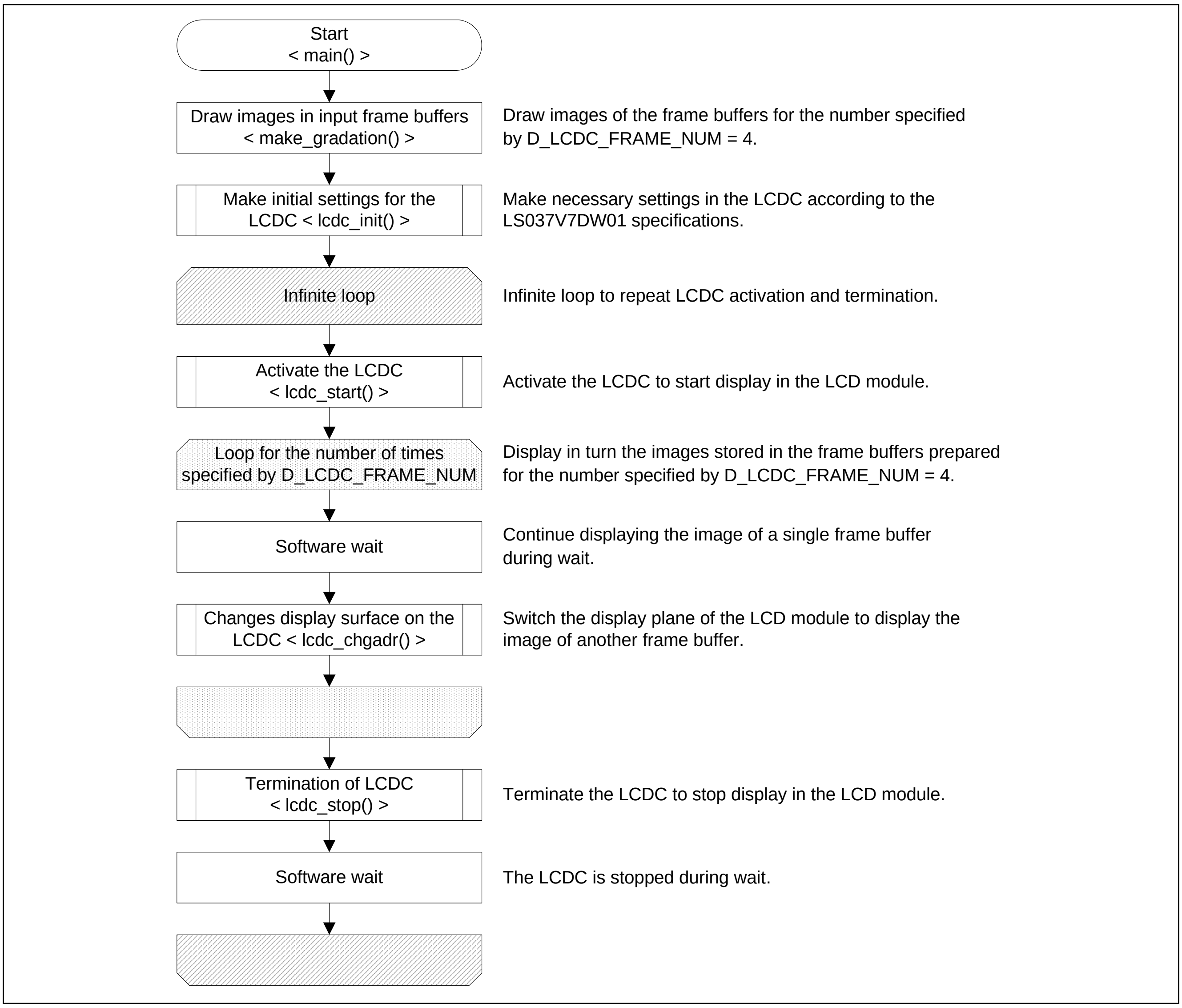
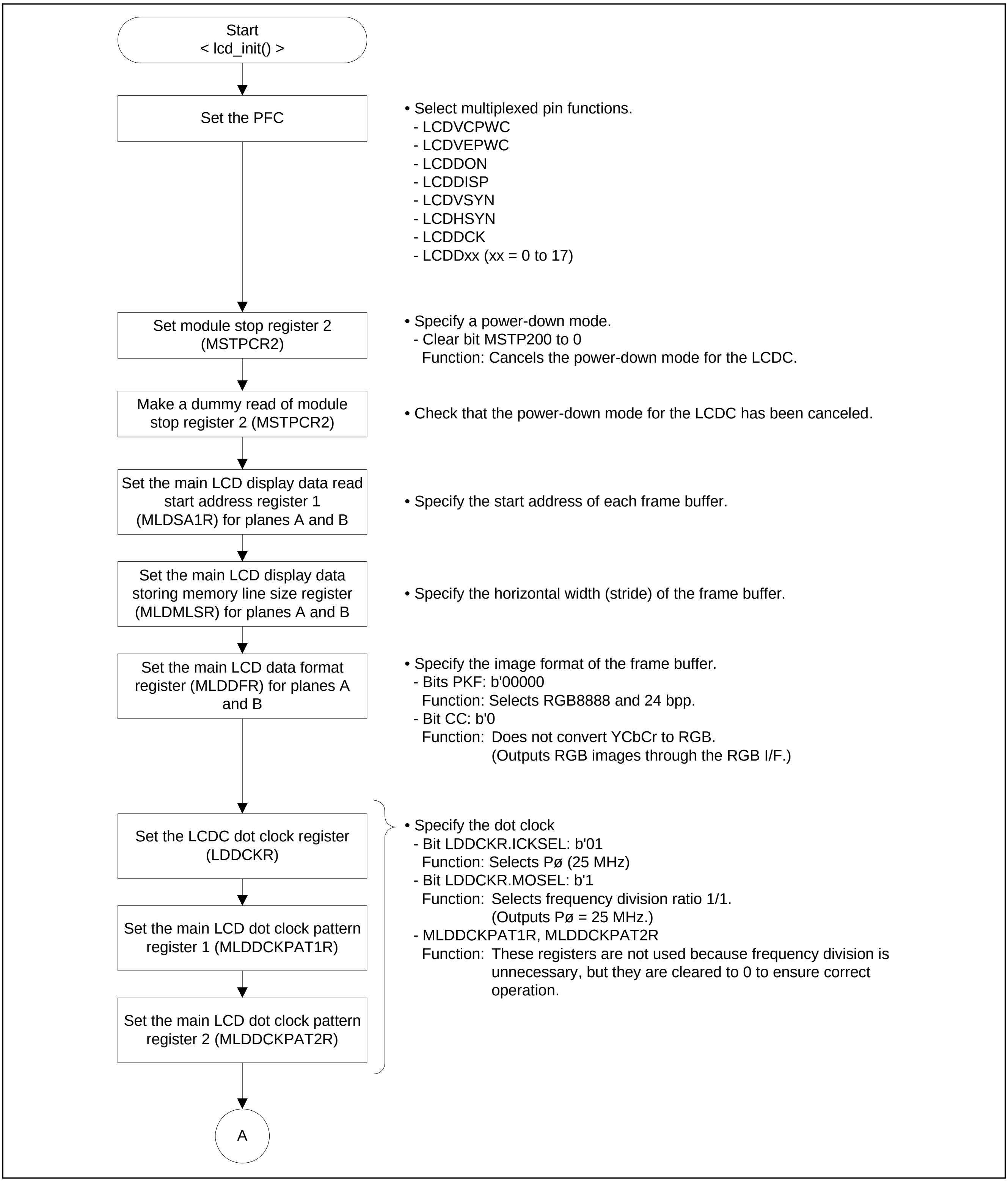

Start
< main() >
Draw images in input frame buffers
< make_gradation() >
Draw images of the frame buffers for the number specified by D_LCDC_FRAME_NUM = 4.
Make initial settings for the LCDC < lcdc_init() >
Make necessary settings in the LCDC according to the LS037V7DW01 specifications.
Infinite loop
Infinite loop to repeat LCDC activation and termination.
Activate the LCDC
< lcdc_start() >
Activate the LCDC to start display in the LCD module.
Loop for the number of times specified by D_LCDC_FRAME_NUM
Display in turn the images stored in the frame buffers prepared for the number specified by D_LCDC_FRAME_NUM = 4.
Software wait
Continue displaying the image of a single frame buffer during wait.
Changes display surface on the LCDC < lcdc_chgadr() >
Switch the display plane of the LCD module to display the image of another frame buffer.
Termination of LCDC
< lcdc_stop() >
Terminate the LCDC to stop display in the LCD module.
Software wait
The LCDC is stopped during wait.

Figure 22 Main Flowchart of Sample Program

4.3.3 Initial Setting of the LCDC

Figure 23 shows the flowchart for initial setting of the LCDC.


Start
< lcd_init() >
Set the PFC
Set module stop register 2 (MSTPCR2)
Make a dummy read of module stop register 2 (MSTPCR2)
Set the main LCD display data read start address register 1 (MLDSA1R) for planes A and B
Set the main LCD display data storing memory line size register (MLDMLSR) for planes A and B
Set the main LCD data format register (MLDDFR) for planes A and B
Set the LCDC dot clock register (LDDCKR)
Set the main LCD dot clock pattern register 1 (MLDDCKPAT1R)
Set the main LCD dot clock pattern register 2 (MLDDCKPAT2R)
A

- Select multiplexed pin functions.
  - LCDVCPWC
  - LCDVEPWC
  - LCDDON
  - LCDDISP
  - LCDVSYN
  - LCDHSYN
  - LCDDCK
  - LCDDxx (xx = 0 to 17)

- Specify a power-down mode.
  - Clear bit MSTP200 to 0
    Function: Cancels the power-down mode for the LCDC.

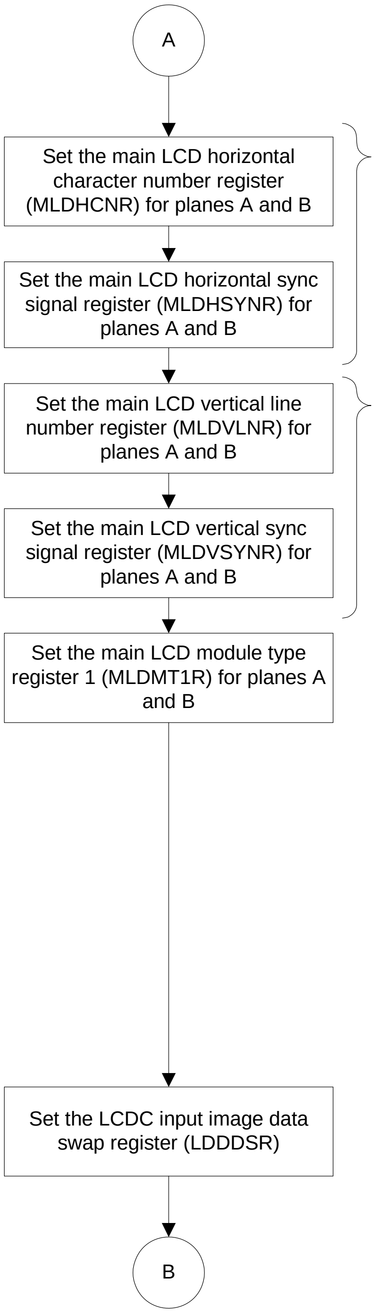
- Check that the power-down mode for the LCDC has been canceled.

- Specify the start address of each frame buffer.

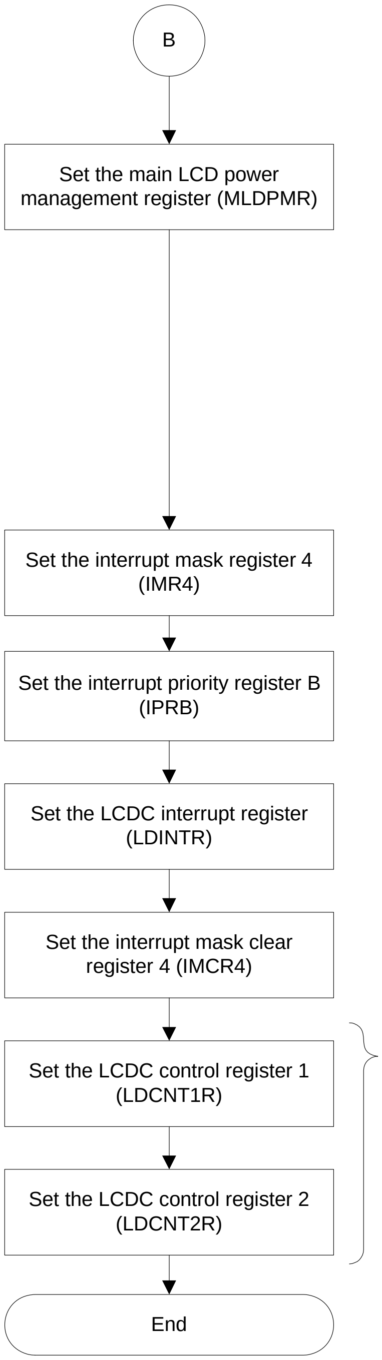
- Specify the horizontal width (stride) of the frame buffer.

- Specify the image format of the frame buffer.
  - Bits PKF: b'00000
    Function: Selects RGB8888 and 24 bpp.
  - Bit CC: b'0
    Function: Does not convert YCbCr to RGB.
    (Outputs RGB images through the RGB I/F.)

- Specify the dot clock
  - Bit LDDCKR.ICKSEL: b'01
    Function: Selects Pø (25 MHz)
  - Bit LDDCKR.MOSEL: b'1
    Function: Selects frequency division ratio 1/1.
    (Outputs Pø = 25 MHz.)
  - MLDDCKPAT1R, MLDDCKPAT2R
    Function: These registers are not used because frequency division is unnecessary, but they are cleared to 0 to ensure correct operation.


A
Set the main LCD horizontal character number register (MLDHCNR) for planes A and B
Set the main LCD horizontal sync signal register (MLDHSYNR) for planes A and B
Set the main LCD vertical line number register (MLDVLNR) for planes A and B
Set the main LCD vertical sync signal register (MLDVSYNR) for planes A and B
Set the main LCD module type register 1 (MLDMT1R) for planes A and B
Set the LCDC input image data swap register (LDDDSR)
B

- Specify the horizontal sync signal.
  - HTCN: 81
    Function: Sets the HSYNC cycle to 81 × 8 = 648 clock cycles.
  - HDCN: 60
    Function: Sets the horizontal display size to 60 × 8 = 480 clock cycles.
  - HSYNW: 1
    Function: Sets the HSYNC signal width to 1 × 8 = 8 clock cycles.
  - HSYNP: 71
    Function: Sets the horizontal display size + horizontal back porch to 71 × 8 = 568 clock cycles.
- Specify the vertical sync signal.
  - VTLN: 648
    Function: Sets the VSYNC cycle to 648 HSYNC cycles (648 lines).
  - VDLN: 640
    Function: Sets the vertical display size to 640 HSYNC cycles (640 lines).
  - VSYNW: 1
    Function: Sets the VSYNC signal width to 1 HSYNC cycle (1 line).
  - VSYNP: 644
    Function: Sets the vertical display size + vertical back porch to 644 HSYNC cycles (644 lines).
- Specify the LCD interface.
  - Bits VPOL, HPOL: b'1
    Function: VSYNC and HSYNC are low-active.
  - Bit DWPOL: b'1
    Function: Data is latched at the rising edge of the dot clock.
  - Bit DIPOL: b'0
    Function: The display enable signal is high-active.
  - Bit DAPOL: b'0
    Function: Display data is high-active.
  - Bit HSCNT: b'0
    Function: HSYNC is output in the vertical blanking period.
  - Bit DWCNT: b'0
    Function: The dot clock is output in the horizontal and vertical blanking periods.
  - Bit IFM: b'0
    Function: RGB interface mode
  - Bits MIFYTP: b'1010
    Function: RGB18 bits, one-time transfer
- Specify data swapping
  - Bit LS: b'1
  - Bit WS: b'0
  - Bit BS: b'0
    Function: Selects longword swapping because RGB8888 24 bpp mode is used.


B
Set the main LCD power management register (MLDPMR)
Set the interrupt mask register 4 (IMR4)
Set the interrupt priority register B (IPRB)
Set the LCDC interrupt register (LDINTR)
Set the interrupt mask clear register 4 (IMCR4)
Set the LCDC control register 1 (LDCNT1R)
Set the LCDC control register 2 (LDCNT2R)
End

• Specify the power management function.
- VE, VE, DO: b'1
  Function: The LCDVCPWC, LCDVEPWC, and LCDDON pins are asserted or negated according to the specified sequence.
- ONA: 2
  Function: Sets sequence period A to 2 frames.
- ONB: 1
  Function: Sets sequence period B to 1 frame.
- ONC: 1
  Function: Sets sequence period C to 1 frame.
- OFFD: 5
  Function: Sets sequence period D to 5 frames.
- OFFE: 1
  Function: Sets sequence period E to 1 frame.
- OFFF: 1
  Function: Sets sequence period F to 1 frame.

• Specify an interrupt mask.
- Bit IMR47: b'1
  Function: Masks LCDC interrupts.

• Specify the interrupt priority level.
- Bits IPR1B: b'1010
  Function: Sets the LCDC interrupt level to 10.

• Make settings for LCDC interrupts.
- Bit FE: b'1
  Function: Enables the LCDC frame end interrupt.
- Bit FS: b'0
  Function: Clears the LCDC frame end interrupt bit.

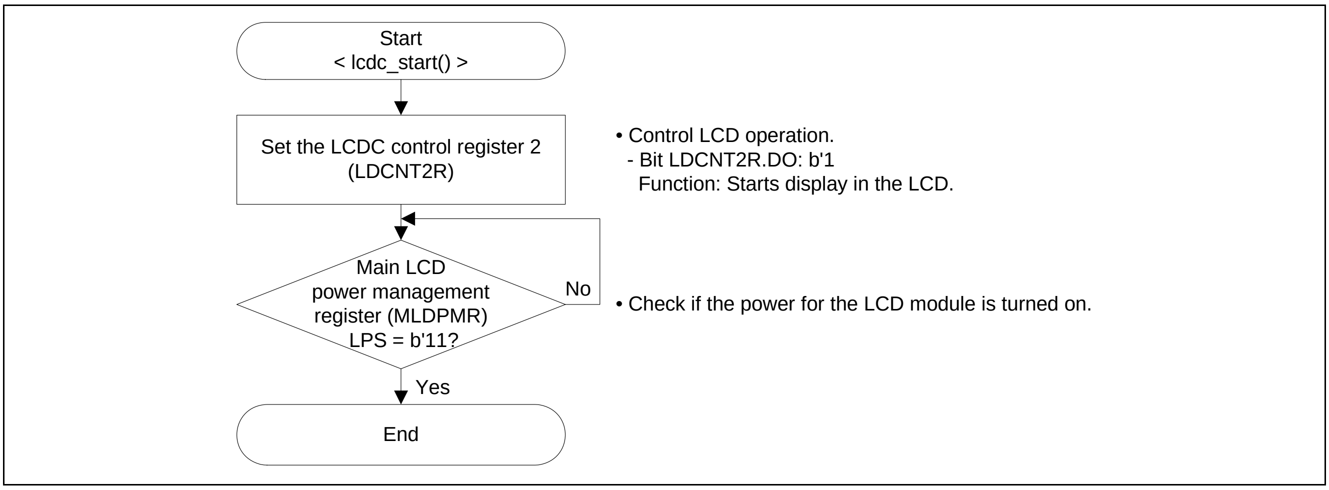
• Clear the interrupt mask.
- Bit IMCR47: b'1
  Function: Clears the LCDC interrupt mask.

• Control LCD operation.
- Bit LDCNT1R.DE: b'1
  Function: Enables data output.
- Bit LDCNT2R.ME: b'1
  Function: Enables the main LCD.

Figure 23 Flowchart for LCDC Initial Setting in Sample Program

4.3.4 Starting and Stopping LCDC Display

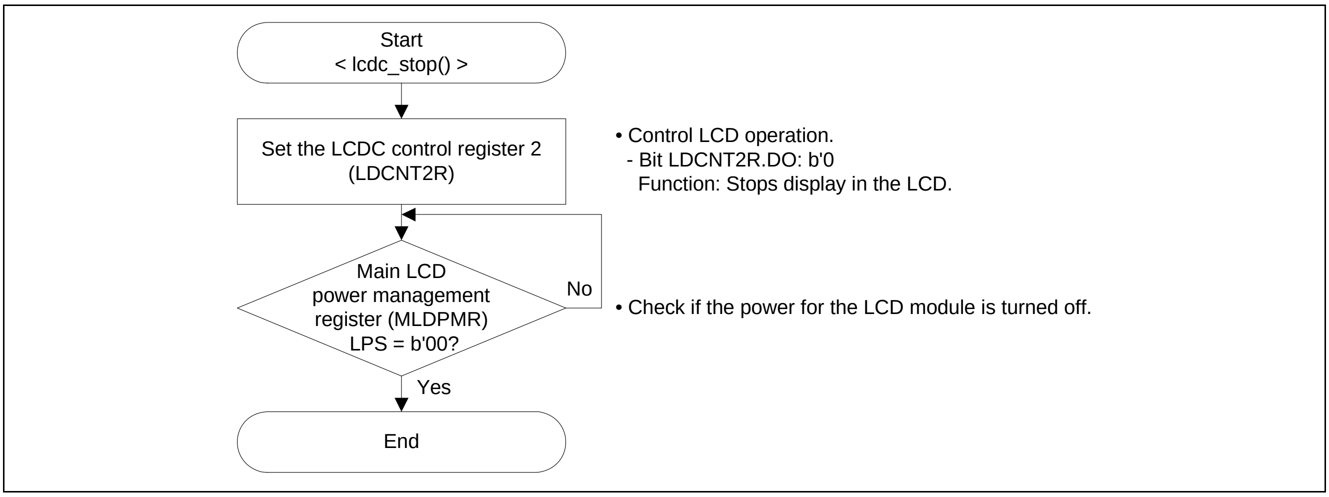
Figure 24 shows the flowchart for starting LCDC display.


Start
< lcdc_start() >
Set the LCDC control register 2 (LDCNT2R)
Main LCD power management register (MLDPMR) LPS = b'11?
No
Yes
End

- Control LCD operation.
  - Bit LDCNT2R.DO: b'1
    Function: Starts display in the LCD.
- Check if the power for the LCD module is turned on.

Figure 24 Flowchart for LCDC Activation in Sample Program

Figure 25 shows the flowchart for stopping LCDC display.


Start
< lcdc_stop() >
Set the LCDC control register 2 (LDCNT2R)
Main LCD power management register (MLDPMR) LPS = b'00?
No
Yes
End

- Control LCD operation.
  - Bit LDCNT2R.DO: b'0
    Function: Stops display in the LCD.
- Check if the power for the LCD module is turned off.

Figure 25 Flowchart for LCDC Termination in Sample Program

4.3.5 LCDC Plane Switching Settings

Figure 26 shows the flowchart for switching LCDC planes. This processing only sets the plane switching flag. The settings necessary for LCDC plane switching are made in the frame end interrupt processing.

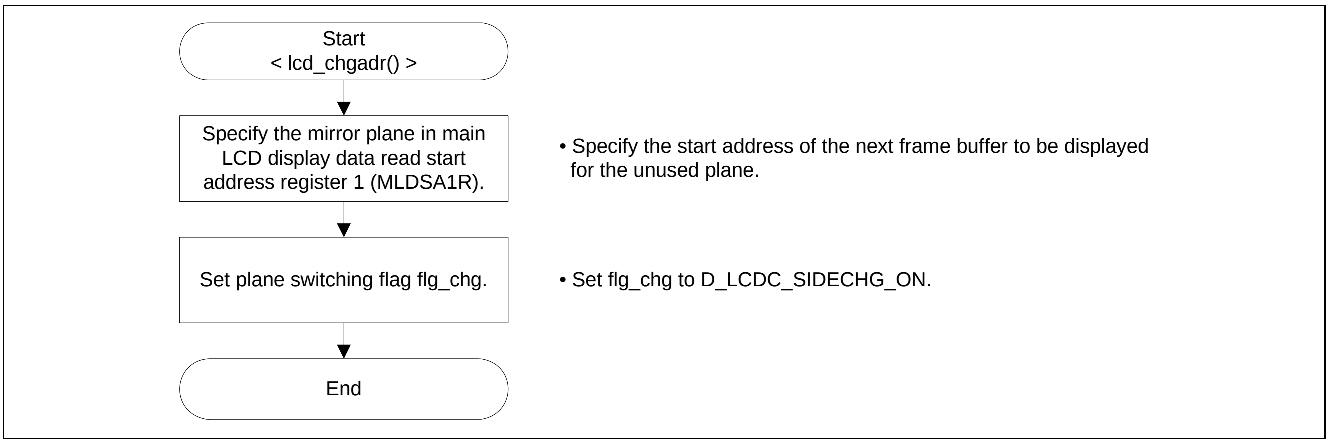
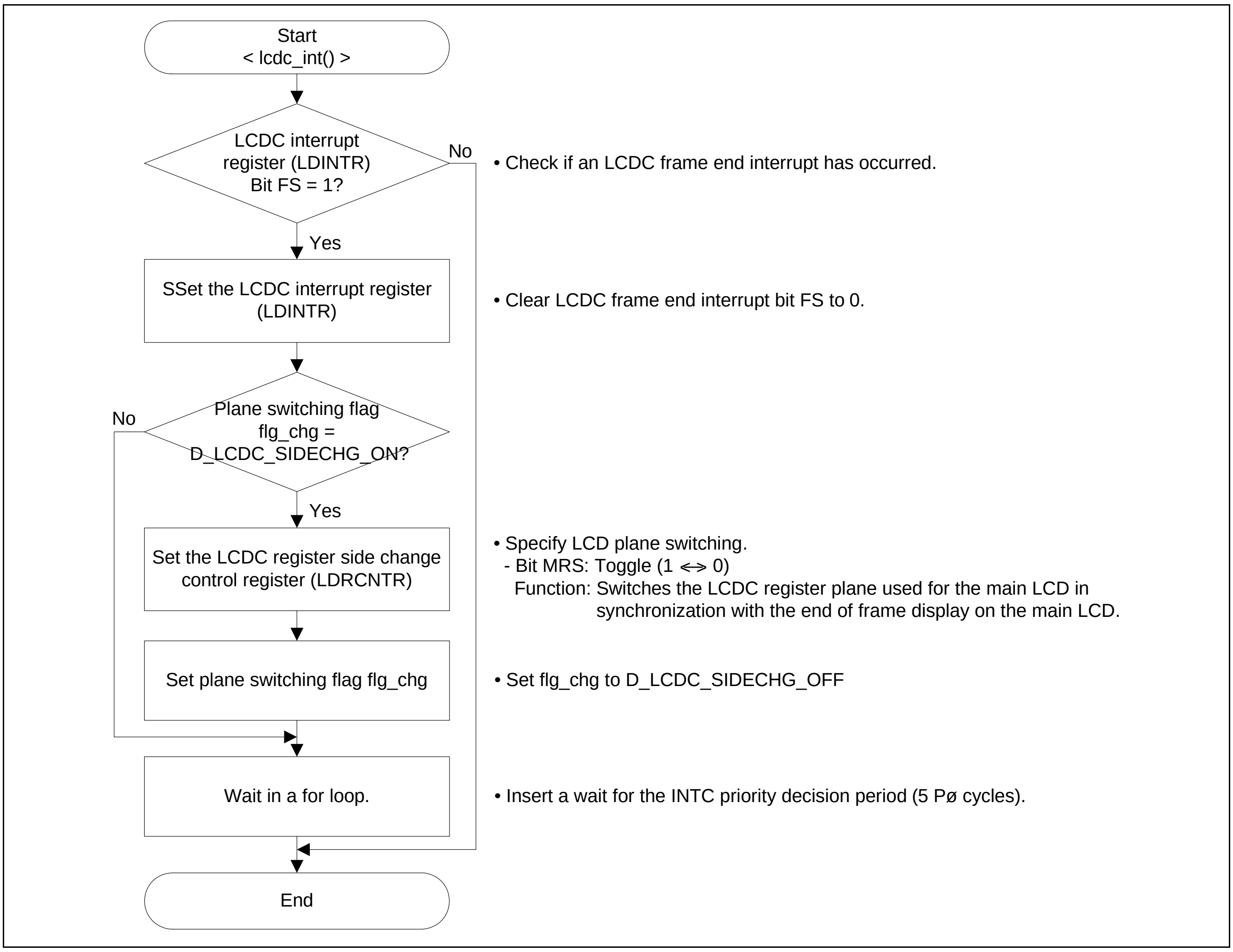

Start
< lcd_chgadr() >
Specify the mirror plane in main LCD display data read start address register 1 (MLDSA1R).
• Specify the start address of the next frame buffer to be displayed for the unused plane.
Set plane switching flag flg_chg.
• Set flg_chg to D_LCDC_SIDECHG_ON.
End

Figure 26 Flowchart for LCDC Plane Switching in Sample Program

4.3.6 LCDC Frame End Interrupt Settings

Figure 27 shows the flowchart for LCDC frame end interrupt processing. When the plane switching flag is set, this processing makes the necessary settings for plane switching in the LCDC. Plane switching is actually done at the end of the next frame. A wait is inserted for the INTC priority decision period (5 Pϕ clock cycles) so that clearing of end interrupt bit FS is reflected in the INTC.


Start
< lcdc_int() >
LCDC interrupt register (LDINTR) Bit FS = 1?
No
Yes
SSet the LCDC interrupt register (LDINTR)
Plane switching flag flg_chg = D_LCDC_SIDECHG_ON?
No
Yes
Set the LCDC register side change control register (LDRCNTR)
Set plane switching flag flg_chg
Wait in a for loop.
End

- Check if an LCDC frame end interrupt has occurred.
- Clear LCDC frame end interrupt bit FS to 0.
- Specify LCD plane switching.
  - Bit MRS: Toggle (1 <-> 0)
    Function: Switches the LCDC register plane used for the main LCD in synchronization with the end of frame display on the main LCD.
- Set flg_chg to D_LCDC_SIDECHG_OFF
- Insert a wait for the INTC priority decision period (5 Pø cycles).

Figure 27 Flowchart for LCDC Frame End Interrupt in Sample Program

4.3.7 Section Allocation

Table 19 shows the allocation of the sections used in this application.

Table 19 Section Allocation

Section Name	Description	Area	Address (Virtual Address)	
P	Program area (with no specification)	ROM	0x00003000	P0 area (caching enabled and MMU address translation enabled)
C	Constant area	ROM		
C$BSEC	Structure for uninitialized data area addresses	ROM		
C$DSEC	Structure for initialized data area addresses	ROM		
D	Initialized data (initial values)	ROM		
PROMC	Area for program stored in ROM	ROM		
B	Uninitialized data area	RAM	0x0C000000	
R	Initialized data area	RAM		
PRAMC	Area for copying program stored in ROM	RAM		
INTHRAM	Area for copying interrupt and exception handlers	RAM		
INTPRAM	Area for copying interrupt functions	RAM		
S	Stack area	RAM	0x0FFFF9F0	
PINTHandler	Exception and interrupt handlers	ROM	0x80000800	P1 area (caching enabled and MMU address translation disabled)
VECTTBL	Reset vector table Interrupt vector table	ROM		
INTTBL	Interrupt mask table	ROM		
PIntPRG	Interrupt functions	ROM		
B_LCD_BUFF	LCDC frame buffers	RAM	0x8E000000	
SP_S	Stack dedicated for TLB-miss handler	RAM	0x8FFFFDF0	
RSTHandler	Reset handlers	ROM	0xA0000000	P2 area (caching disabled and MMU address translation disabled)
PResetPRG	Reset programs	ROM		
PnonCACHE	Program area (cache-disabled access)	ROM		

5. Listing of the Sample Program

(1) Sample Program Listing: "main.c"

```
/******************************************************************************
* DISCLAIMER

* This software is supplied by Renesas Electronics Corporation. and is only
* intended for use with Renesas products. No other uses are authorized.

* This software is owned by Renesas Electronics Corporation. and is protected under
* all applicable laws, including copyright laws.

* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
* REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
* INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
* PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
* DISCLAIMED.

* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
* FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
* AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

* Renesas reserves the right, without notice, to make changes to this
* software and to discontinue the availability of this software.
* By using this software, you agree to the additional terms and
* conditions found by accessing the following link:
* http://www.renesas.com/disclaimer
******************************************************************************/
/* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.          */
/*""FILE COMMENT""*********** Technical reference data ****************
* System Name  : SH7722,SH7731 Sample Program
* File Name    : resetprg.c
* Abstract     : Sample Program for Setting the SH7722,SH7731 LCDC
* Version      : Ver 1.00
* Device       : SH7722,SH7731
* Tool-Chain   : High-performance Embedded Workshop (Version 4.05.01.001)
*              : C/C++ Compiler Package for SuperH Family (V.9.03 release00)
* OS           : None
* H/W Platform : R0P7722TH001ARK for SH7722 Reference Platform
* Description  : Sample Program for Setting the SH7722,SH7731 LCDC
*              :
* Operation    :
* Limitation   :
*              :
***********************************************************************
* History      : 6.July.2009 Ver. 1.00 First Release
*""FILE COMMENT END""**************************************************/

/***********************************************************************/
/*                                                                     */
/*  FILE        :SH7731_ini.c                                          */
/*  DATE        :Wed, Apr 22, 2009                                     */
/*  DESCRIPTION :Main Program                                          */
/*  CPU TYPE    :SH7722,SH7731                                         */
/*                                                                     */
/*  This file is generated by Renesas Project Generator (Ver.4.9).     */
/*                                                                     */
/***********************************************************************/

#include <stdio.h>
```

```
#include <machine.h>
#include "iodefine.h"
#include "lcdc.h"
#include "framebuf.h"

//#include "typedefine.h"
#ifdef __cplusplus
//#include <ios>                        // Remove the comment when you use ios
//_SINT ios_base::Init::init_cnt;       // Remove the comment when you use ios
#endif

void main(void);
#ifdef __cplusplus
extern "C" {
void abort(void);
}
#endif

#pragma section ROMC                // Section to be executed in RAM
/*""FUNC COMMENT""***************************************************
* ID                       :
* Outline                  : Sample program main
*                          : (produces a display from the LCDC)
* Include                  :
* Declaration              : void main (void)
* Description              : This program displays a VGA image on the main LCD and then, after a software
*                          : wait defined by WAIT_LCDC_CHGSIDE followed by repeated switching of the
*            : display between the A and B planes TIMES_LCDC_CHGSIDE times, stops the
*              display.
*            : The display is restarted after a software wait defined by WAIT_LCDC_STOP.
             :
* Limitation               :
*                          :
* Argument                 : none
* Return Value             : none
* Calling Functions        :
*""FUNC COMMENT END""**********************************************/
void main(void)
{
    unsigned long i;
    unsigned long j;

    /* Draws the image in frame buffers */
    make_gradation();

    /* LCDC initialization : VGA display on the main LCD */
    lcdc_init( pg_tbl_lcdc_buf[D_LCDC_FRAME_NUM-1] );


    while(1)
    {
      /* LCDC activation : Starts display on the main LCD */
        lcdc_start();

        for(i=0;i<D_LCDC_FRAME_NUM;i++)
        {
            for(j=0;j<D_LCDC_WAIT_CHGSIDE;j++)
            {
                /* Wait */
            }

            /* Changes display surface on the LCDC */
            lcdc_chgadr( pg_tbl_lcdc_buf[i] );
        }
        /* Stopping the LCDC : Stops display on the main LCD */
        lcdc_stop();
        for(j=0;j<D_LCDC_WAIT_STOP;j++)
        {
            /* Wait */
        }
    }


}
#pragma section

#ifdef __cplusplus
void abort(void)
{

}
#endif
```

(2) Sample Program Listing: "framebuf.c"

```
/*****************************************************************************
* DISCLAIMER

* This software is supplied by Renesas Electronics Corporation. and is only
* intended for use with Renesas products. No other uses are authorized.

* This software is owned by Renesas Electronics Corporation. and is protected under
* all applicable laws, including copyright laws.

* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
* REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
* INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
* PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
* DISCLAIMED.

* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
* FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
* AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

* Renesas reserves the right, without notice, to make changes to this
* software and to discontinue the availability of this software.
* By using this software, you agree to the additional terms and
* conditions found by accessing the following link:
* http://www.renesas.com/disclaimer
******************************************************************************/
/* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.         */
/*""FILE COMMENT""*********** Technical reference data ****************
* System Name  : SH7722,SH7731 Sample Program
* File Name    : resetprg.c
* Abstract     : Sample Program for Setting the SH7722,SH7731 LCDC
* Version      : Ver 1.00
* Device       : SH7722,SH7731
* Tool-Chain   : High-performance Embedded Workshop (Version 4.05.01.001)
*              : C/C++ Compiler Package for SuperH Family (V.9.03 release00)
* OS           : None
* H/W Platform : R0P7722TH001ARK for SH7722 Reference Platform
* Description  : Sample Program for Setting the SH7722,SH7731 LCDC
*              :
* Operation    :
* Limitation   :
*              :
***********************************************************************
* History      : 21.Oct.2009 Ver. 1.00 First Release
*""FILE COMMENT END""**************************************************/


#include <machine.h>
#include "iodefine.h"
#include "framebuf.h"

/* ==== Frame buffers ==== */
#pragma section _LCD_BUFF
unsigned long g_lcdc_buf0[D_LCDC_HEIGHT][D_LCDC_WIDTH];
unsigned long g_lcdc_buf1[D_LCDC_HEIGHT][D_LCDC_WIDTH];
unsigned long g_lcdc_buf2[D_LCDC_HEIGHT][D_LCDC_WIDTH];
unsigned long g_lcdc_buf3[D_LCDC_HEIGHT][D_LCDC_WIDTH];
#pragma section

/* ==== Frame buffer table ==== */
```

```
void* pg_tbl_lcdc_buf[4] = {
    g_lcdc_buf0,
    g_lcdc_buf1,
    g_lcdc_buf2,
    g_lcdc_buf3
};


/* ==== Prototype declaration ==== */
void make_gradation(void);


/*""FUNC COMMENT""***************************************************
* ID                         :
* Outline                    : Sample program main
*                            : (produces a display from the LCDC)
* Include                    :
* Declaration                : void make_gradation(void)
* Description                : This function creates a 24-bpp gradated image in RGB8888 format within the
              frame buffer.
*                            :
* Limitation                 :
*                            :
* Argument                   : none
* Return Value               : none
* Calling Functions          :
*""FUNC COMMENT END""**********************************************/
void make_gradation(void)
{
    static unsigned long i;
    static unsigned long j;
    unsigned long pixel0 = 0;
    unsigned long pixel1 = 0;
    unsigned long pixel2 = 0;
    unsigned long pixel3 = 0;

    for(i=0;i<D_LCDC_HEIGHT;i++)
    {
        for(j=0;j<D_LCDC_WIDTH;j++)
        {
            g_lcdc_buf0[i][j] = pixel0;
            g_lcdc_buf1[i][j] = pixel1;
            g_lcdc_buf2[i][j] = pixel2;
            g_lcdc_buf3[i][j] = pixel3;

            if( ( j != 0 ) && ( (j%60) == 0 ) )
            {
                pixel0 += D_LCDC_PATTERN0;
                pixel1 += D_LCDC_PATTERN1;
                pixel2 += D_LCDC_PATTERN2;
                pixel3 += D_LCDC_PATTERN3;
            }
        }
        pixel0 = 0;
        pixel1 = 0;
        pixel2 = 0;
        pixel3 = 0;
    }
}


/* End of File */
```

(3) Sample Program Listing: "framebuf.h"

```
/******************************************************************************
* DISCLAIMER

* This software is supplied by Renesas Electronics Corporation. and is only
* intended for use with Renesas products. No other uses are authorized.

* This software is owned by Renesas Electronics Corporation. and is protected under
* all applicable laws, including copyright laws.

* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
* REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
* INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
* PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
* DISCLAIMED.

* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
* FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
* AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

* Renesas reserves the right, without notice, to make changes to this
* software and to discontinue the availability of this software.
* By using this software, you agree to the additional terms and
* conditions found by accessing the following link:
* http://www.renesas.com/disclaimer
******************************************************************************/
/* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.         */
/*""FILE COMMENT""*********** Technical reference data ****************
* System Name  : SH7722,SH7731 Sample Program
* File Name    : resetprg.c
* Abstract     : Sample Program for Setting the SH7722,SH7731 LCDC
* Version      : Ver 1.00
* Device       : SH7722,SH7731
* Tool-Chain   : High-performance Embedded Workshop (Version 4.05.01.001)
*              : C/C++ Compiler Package for SuperH Family (V.9.03 release00)
* OS           : None
* H/W Platform : R0P7722TH001ARK for SH7722 Reference Platform
* Description  : Sample Program for Setting the SH7722,SH7731 LCDC
*              :
* Operation    :
* Limitation   :
*              :
***********************************************************************
* History      : 21.Oct.2009 Ver. 1.00 First Release
*""FILE COMMENT END""**************************************************/

#ifndef _FRAMEBUF_H_
#define _FRAMEBUF_H_

/* ==== Macro definition ==== */
#define   D_LCDC_FRAME_NUM        4

#define   D_LCDC_WIDTH        480 /* Width of the image */
#define   D_LCDC_HEIGHT       640 /* Hight of the image */
#define   D_LCDC_BPP          4   /* Number of bytes per pixel */
#define D_LCDC_STRIDE       ( D_LCDC_WIDTH * D_LCDC_BPP )

#define D_LCDC_PATTERN0 0x00200000
#define D_LCDC_PATTERN1 0x00002000
#define D_LCDC_PATTERN2 0x00000020
```

```
#define D_LCDC_PATTERN3 0x00202020

/* ==== Frame buffer ==== */
extern void* pg_tbl_lcdc_buf[D_LCDC_FRAME_NUM];

#endif /* _FRAMEBUF_H_ */
/* End of File */
```

(4) Sample Program Listing: "lcdc.c"

```
/******************************************************************************
* DISCLAIMER

* This software is supplied by Renesas Electronics Corporation. and is only
* intended for use with Renesas products. No other uses are authorized.

* This software is owned by Renesas Electronics Corporation. and is protected under
* all applicable laws, including copyright laws.

* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
* REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
* INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
* PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
* DISCLAIMED.

* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
* FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
* AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

* Renesas reserves the right, without notice, to make changes to this
* software and to discontinue the availability of this software.
* By using this software, you agree to the additional terms and
* conditions found by accessing the following link:
* http://www.renesas.com/disclaimer
******************************************************************************/
/* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.         */
/*""FILE COMMENT""*********** Technical reference data ****************
* System Name  : SH7722,SH7731 Sample Program
* File Name    : resetprg.c
* Abstract     : Sample Program for Setting the SH7722,SH7731 LCDC
* Version      : Ver 1.00
* Device       : SH7722,SH7731
* Tool-Chain   : High-performance Embedded Workshop (Version 4.05.01.001)
*              : C/C++ Compiler Package for SuperH Family (V.9.03 release00)
* OS           : None
* H/W Platform : R0P7722TH001ARK for SH7722 Reference Platform
* Description  : Sample Program for Setting the SH7722,SH7731 LCDC
*              :
* Operation    :
* Limitation   :
*              :
***********************************************************************
* History      : 21.Oct.2009 Ver. 1.00 First Release
*""FILE COMMENT END""**************************************************/


#include <machine.h>
#include "iodefine.h"
#include "lcdc.h"
#include "framebuf.h"

long g_flg_chg = D_LCDC_SIDECHG_OFF;

/*""FUNC COMMENT""***************************************************
* ID                       :
* Outline                  : Sample program main
*                          : (produces a display from the LCDC)
* Include                  :
* Declaration              : void lcdc_init(void* framebuf_adr)
* Description              : Initializes the LCDC.
*                          : Makes necessary settings to enable display on the main LCD.
*                          :
*                          :
*                          :
* Limitation               :
*                          :
* Argument                 : none
* Return Value             : none
* Calling Functions        :
*""FUNC COMMENT END""**********************************************/
void lcdc_init(void* framebuf_adr)
{
    unsigned long dummy;

    /* PFC setting */
    /* Specifies in the order of MSEL->PSEL->PCR->HiZ. */
    PFC.MSELCRB.BIT.MSELB8 = 0;
    PFC.PSELD.WORD = ~0x000D;
    PFC.PHCR.WORD = 0;
    PFC.PLCR.WORD = 0;
    PFC.PMCR.WORD = 0;
    PFC.PRCR.WORD &= ~0x000F;
    PFC.HIZCRA.WORD &= ~0x01C0;

    /* Cancels the power-down mode. */
    LOWP.MSTPCR2 &= ~0x00000001;
    dummy = LOWP.MSTPCR2; /* Makes a dummy read to confirm that the power-down mode has been canceled. */

    /* Frame buffer setting */
    LCDCA.MLDSA1R = (unsigned long)D_LCDC_PHY_ADR(framebuf_adr);
    LCDCB.MLDSA1R = (unsigned long)D_LCDC_PHY_ADR(framebuf_adr);

    LCDCA.MLDMLSR = D_LCDC_STRIDE;
    LCDCB.MLDMLSR = D_LCDC_STRIDE;

    LCDCA.MLDDFR.BIT.CC = 0; /* Conversion from YCbCr to RGB is not performed. */
    LCDCA.MLDDFR.BIT.PKF = 0; /* RGB8888 24bpp */
    LCDCB.MLDDFR.BIT.CC = 0;
    LCDCB.MLDDFR.BIT.PKF = 0;

    /* Dot clock setting */
    LCDCA.LDDCKR.BIT.ICKSEL = 1;          /* Pφ */
    LCDCA.LDDCKR.BIT.MOSEL = 1;           /* Frequency division 1/1 */
    LCDCA.MLDDCKPAT1R = 0x00000000;
    LCDCA.MLDDCKPAT2R = 0x00000000;

    /* Sync signal setting */
    LCDCA.MLDHCNR.BIT.HTCN      = 81;
    LCDCA.MLDHCNR.BIT.HDCN      = 60;
    LCDCA.MLDHSYNR.BIT.HSYNW    = 1;
    LCDCA.MLDHSYNR.BIT.HSYNP    = 71;
    LCDCA.MLDVLNR.BIT.VTLN      = 648;
    LCDCA.MLDVLNR.BIT.VDLN      = 640;
    LCDCA.MLDVSYNR.BIT.VSYNW    = 1;
    LCDCA.MLDVSYNR.BIT.VSYNP    = 646;

    LCDCB.MLDHCNR.BIT.HTCN      = 81;
    LCDCB.MLDHCNR.BIT.HDCN      = 60;
    LCDCB.MLDHSYNR.BIT.HSYNW    = 1;
    LCDCB.MLDHSYNR.BIT.HSYNP    = 71;
    LCDCB.MLDVLNR.BIT.VTLN      = 648;
    LCDCB.MLDVLNR.BIT.VDLN      = 640;
    LCDCB.MLDVSYNR.BIT.VSYNW    = 1;
    LCDCB.MLDVSYNR.BIT.VSYNP    = 646;

    /* LCD interface setting */
    LCDCA.MLDMT1R.LONG = 0x1C00000A;       /* Main LCD:RGB interface/18bit/active_low */
    LCDCB.MLDMT1R.LONG = 0x1C00000A;

    /* Data swapping setting */
    LCDCA.LDDDSR.LONG = 0x00000004; /* Longword swapping */

    /* Power management setting */
    LCDCA.MLDPMR.BIT.ONA  = 2;
    LCDCA.MLDPMR.BIT.ONB  = 1;
    LCDCA.MLDPMR.BIT.ONC  = 1;
    LCDCA.MLDPMR.BIT.OFFD  = 5;
    LCDCA.MLDPMR.BIT.OFFE  = 1;
    LCDCA.MLDPMR.BIT.OFFF  = 1;
    LCDCA.MLDPMR.BIT.VC  = 1;
    LCDCA.MLDPMR.BIT.VE  = 1;
    LCDCA.MLDPMR.BIT.DO  = 1;

    /* Interrupt setting */
    /* Masks the LCDC interrupts. */
    INTC0.IMR4.BYTE |= D_LCDC_INT_MASK;
    /* Enables the LCDC interrupt to be used. */
    INTC0.IPRB = 0x0100;        /* Interrupt priority level 1 */
    LCDCA.LDINTR.LONG = D_LCDC_INT_FE_ON; /* Enables frame end interrupts. */
    /* Clears the LCDC interrupt source. */
    LCDCA.LDINTR.LONG &= ~D_LCDC_INT_FS_FLG;
    /* Clears the mask for the LCDC interrupts. */
    INTC0.IMCR4 = D_LCDC_INT_MASK;

    /* Enables the main LCD */
    LCDCA.LDCNT1R.LONG = 0x00000001; /* Enables display. */
    LCDCA.LDCNT2R.LONG = 0x00000002; /* Enables the main LCD. */
}

/*""FUNC COMMENT""***************************************************
* ID                       :
* Outline                  : Sample program main
*                          : (produces a display from the LCDC)
* Include                  :
* Declaration              : void lcdc_start(void)
* Description              : Activates the LCDC.
*                          :
* Limitation               :
*                          :
* Argument                 : none
* Return Value             : none
* Calling Functions        :
*""FUNC COMMENT END""*********************************************/
void lcdc_start(void)
{
    /*DO = 1*/
    LCDCA.LDCNT2R.LONG |= 0x00000001; /* Initializes the LCDC */

    /* Waits until the status changes to "displaying". */
    while( LCDCA.MLDPMR.BIT.LPS != 0x00000003 )
    {
        /* DO NOTHING */
    }
}

```

```
/*""FUNC COMMENT""***************************************************
 * ID                        :
 * Outline                   : Sample program main
 *                           : (produces a display from the LCDC)
 * Include                   :
 * Declaration               : void lcdc_chgadr(unsigned long mirror_adr)
 * Description               : Initializes the LCDC.
 *                           :
 * Limitation                :
 *                           :
 * Argument                  : none
 * Return Value              : none
 * Calling Functions         :
 *""FUNC COMMENT END""**********************************************/
void lcdc_chgadr(void* mirror_adr)
{
    /* Specifies the mirror addresses. */
    LCDCM.MLDSA1R = (unsigned long)D_LCDC_PHY_ADR(mirror_adr);
    g_flg_chg = D_LCDC_SIDECHG_ON;
}

/*""FUNC COMMENT""***************************************************
 * ID                        :
 * Outline                   : Sample program main
 *                           : (produces a display from the LCDC)
 * Include                   :
 * Declaration               : void lcdc_stop(void)
 * Description               : Stops the LCDC.
 *                           :
 * Limitation                :
 *                           :
 * Argument                  : none
 * Return Value              : none
 * Calling Functions         :
 *""FUNC COMMENT END""**********************************************/
void lcdc_stop(void)
{
    /*DO = 0*/
    LCDCA.LDCNT2R.LONG &= ~0x00000001; /* Stops display on the main LCD. */

    /* Wwaits until the status changes to "stopped". */
    while( LCDCA.MLDPMR.BIT.LPS != 0x00000000 )
    {
        /* DO NOTHING */
    }
}

/* End of File */
```

(5) Sample Program Listing: "lcdc.h"

```
/*******************************************************************************
* DISCLAIMER

* This software is supplied by Renesas Electronics Corporation. and is only
* intended for use with Renesas products. No other uses are authorized.

* This software is owned by Renesas Electronics Corporation. and is protected under
* all applicable laws, including copyright laws.

* THIS SOFTWARE IS PROVIDED "AS IS" AND RENESAS MAKES NO WARRANTIES
* REGARDING THIS SOFTWARE, WHETHER EXPRESS, IMPLIED OR STATUTORY,
* INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, FITNESS FOR A
* PARTICULAR PURPOSE AND NON-INFRINGEMENT.  ALL SUCH WARRANTIES ARE EXPRESSLY
* DISCLAIMED.

* TO THE MAXIMUM EXTENT PERMITTED NOT PROHIBITED BY LAW, NEITHER RENESAS
* ELECTRONICS CORPORATION. NOR ANY OF ITS AFFILIATED COMPANIES SHALL BE LIABLE
* FOR ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES
* FOR ANY REASON RELATED TO THE THIS SOFTWARE, EVEN IF RENESAS OR ITS
* AFFILIATES HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

* Renesas reserves the right, without notice, to make changes to this
* software and to discontinue the availability of this software.
* By using this software, you agree to the additional terms and
* conditions found by accessing the following link:
* http://www.renesas.com/disclaimer
*******************************************************************************/
/* Copyright (C) 2010. Renesas Electronics Corporation., All Rights Reserved.         */
/*""FILE COMMENT""*********** Technical reference data ****************
* System Name  : SH7722,SH7731 Sample Program
* File Name    : resetprg.c
* Abstract     : Sample Program for Setting the SH7722,SH7731 LCDC
* Version      : Ver 1.00
* Device       : SH7722,SH7731
* Tool-Chain   : High-performance Embedded Workshop (Version 4.05.01.001)
*              : C/C++ Compiler Package for SuperH Family (V.9.03 release00)
* OS           : None
* H/W Platform : R0P7722TH001ARK for SH7722 Reference Platform
* Description  : Sample Program for Setting the SH7722,SH7731 LCDC
*              :
* Operation    :
* Limitation   :
*              :
***********************************************************************
* History      : 21.Oct.2009 Ver. 1.00 First Release
*""FILE COMMENT END""*****************************************************/

#ifndef _LCDC_H_
#define _LCDC_H_

/* ==== Macro definition ==== */
#define   D_LCDC_WAIT_CHGSIDE         (200000000/2*1) /* Frequency ÷(number of instructions in the for
                  loop)×seconds */
#define   D_LCDC_WAIT_STOP            (200000000/2*1) /* Frequency ÷(number of instructions in the for
                  loop)×seconds */

#define D_LCDC_PHY_ADR(x)           ( (unsigned long)(x) & ~0xE0000000 )

#define D_LCDC_INT_FE_ON  0x00000400
#define D_LCDC_INT_FS_FLG  0x00000004
#define D_LCDC_INT_MASK  0x01
```

```

enum {
    D_LCDC_SIDECHG_OFF = 0,
    D_LCDC_SIDECHG_ON,
};

/* ==== Function declaration ==== */
void lcdc_init(void* framebuf_adr);
void lcdc_start(void);
void lcdc_chgadr(void* mirror_adr);
void lcdc_stop(void);
void lcdc_int(void);

/* ==== Variable declaration ==== */
extern long g_flg_chg;

#endif /* _LCDC_H_ */
/* End of File */
```

(6) Sample Program Listing: "vhandler.src"

The vhandler.src file is modified from that described in the "SH7722/SH7731 Group Application Note: SH7722/SH7731 Example of Initialization (REJ06B0942)" as follows.

- The CPG settings are modified so that the applicable conditions in section 1.3 are satisfied.
- As B3ϕ = 80 MHz, the timing settings in the SBSC are modified.

For details of the settings, refer to the source code.

```

…Omitted…

             mov.l   #H'A4150000,r0 ;set FRQCR address
             mov.l   #H'05022538,r1 ;  * Clockin = 33.333MHz, CKIO = 66.6MHz,
                                    ;  * I Clock = 200MHz, U Clock = 100MHz,
                                    ;  * SH Clock = 100MHz, B Clock = 50MHz,
                                    ;  * B3 Clock = 80MHz, P Clock = 25MHz
             mov.l   r1,@r0

…Omitted…

SDRAM_INIT:
             mov.l   #H'FE400008,r0 ;set SDCR0 address
             mov.l   #H'00020809,r1 ;set for SDRAM(Micron MT48LC8M16A2B475)
                                    ;64bit bus-width, row 12bit, column 9bit
             mov.l   r1,@r0

             mov.l   #H'FE40000C,r0 ;set SDWCR address
             mov.l   #H'0014248A,r1 ;tRRD 2cyc
                                    ;tRAS 4cyc
                                    ;tRP  2cyc
                                    ;tRCD 2cyc
                                    ;CL   2cyc
                                    ;tWR  2cyc
                                    ;tRC  6cyc
             mov.l   r1,@r0

             mov.l   #H'FE400010,r0 ;set SDPCR address
             mov.l   #H'00000087,r1 ;default
             mov.l   r1,@r0

             mov.l   #H'FE400018,r0 ;set RTCNT address
             mov.l   #H'a55a0000,r1
             mov.l   r1,@r0

             mov.l   #H'FE40001C,r0 ;set RTCOR address
             mov.l   #H'a55a004C,r1 ;refresh rate
             mov.l   r1,@r0

             mov.l   #H'A4050186,r0 ;set SBSCR address
             mov.W   #H'0000,r1     ;Low speed
             mov.W   r1,@r0


             mov.l   #H'000030d4,r0
LOOP1:
             dt      r0
             bf      LOOP1 ;200μs wait
             nop
             nop

             mov.l   #H'FE400014,r0 ;set RTCSR address
             mov.l   #H'a55a0010,r1 ;B3φ/16
             mov.l   r1,@r0

             mov.l   #H'FE500100,r0  ;set SDMR3(64bit bus-width, CL=2, burstR/W(burst length=1))
             mov.b   #H'00,r1
             mov.b   r1,@r0

             mov.l   #SDRAM_INIT_END,r0
             jmp     @r0
             nop

             .pool

…Omitted…
```

(7) Sample Program Listing: "vecttbl.src"

This program specifies the interrupt priority level that can be accepted during LCDC interrupt processing execution.

As the LCDC interrupt priority level is set to 1, the acceptable priority level is also set to 1 so that another LCDC interrupt is not accepted during earlier LCDC interrupt processing.

```

…Omitted…

;LCDC
             ; H'580 LCDC LCDCI
             .data.b    H'10

…Omitted…
```

(8) Sample Program Listing: "intprg.c"

This program registers the LCDC interrupt processing function as an interrupt handler.

```

…Omitted…

#define I_DIV_P         8 /* Iφ:Pφ = 8:1 */
#define INST_NUM        2 /* instruction number of for loop */
#define PCLK_5CYC       ( 5 * I_DIV_P / 2)

…Omitted…

/* H'580 LCD Controler interrupt */
/*""FUNC COMMENT""***************************************************
* ID                       :
* Outline                  : Sample program main
*                          : (produces a display from the LCDC)
* Include                  :
* Declaration              : void lcdc_int(void)
* Description              : Initializes LCDC.
*                          : Makes necessary settings to enable display on the main LCD.
*                          :
*                          :
*                          :
* Limitation               :
*                          :
* Argument                 : none
* Return Value             : none
* Calling Functions        :
*""FUNC COMMENT END""**********************************************/
void INT_LCDC_LCDCI(void)
{
    unsigned long i;

    if( LCDCA.LDINTR.LONG & D_LCDC_INT_FS_FLG)
    {
        LCDCA.LDINTR.LONG &= ~D_LCDC_INT_FS_FLG;

        if( g_flg_chg == D_LCDC_SIDECHG_ON )
        {
            LCDCA.LDRCNTR.LONG ^= 0x00000002; /* Inverts MRS. */
         g_flg_chg = D_LCDC_SIDECHG_OFF;
        }

      /* Waits for the INTC priority decision period. */
      for(i=0;i<PCLK_5CYC;i++)
      {
         /* DO NOTHING */
      }
    }
}

…Omitted…
```

6. Execution Results

When the above sample application is executed, the following processes will be repeated.

- The power for the LCD module is turned on.
- The gray gradation pattern is displayed.
- The red gradation pattern is displayed.
- The green gradation pattern is displayed.
- The blue gradation pattern is displayed.
- The power for the LCD module is turned off.
- The power for the LCD module is turned on (the above steps are repeated infinitely).

7. Documents for Reference

- Software Manual
  SH-4A Software Manual (REJ09B0003)
  (The most up-to-date versions of the documents are available on the Renesas Electronics Website.)

- Hardware Manual
  SH7722 Group Hardware Manual (REJ09B0314)
  (The most up-to-date versions of the documents are available on the Renesas Electronics Website.)

Website and Support

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

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

Revision Record

		Description	
Rev.	Date	Page	Summary
1.00	Sep 21, 2010	—	First edition issued
1.01	Jun 09, 2011	26	Sample calculation of the maximum data rate required by the LCDC is corrected.

General Precautions in the Handling of MPU/MCU Products

The following usage notes are applicable to all MPU/MCU products from Renesas. For detailed usage notes on the products covered by this manual, refer to the relevant sections of the manual. If the descriptions under General Precautions in the Handling of MPU/MCU Products and in the body of the manual differ from each other, the description in the body of the manual takes precedence.

1. Handling of Unused Pins
   Handle unused pins in accord with the directions given under Handling of Unused Pins in the manual.
   — The input pins of CMOS products are generally in the high-impedance state. In operation with an unused pin in the open-circuit state, extra electromagnetic noise is induced in the vicinity of LSI, an associated shoot-through current flows internally, and malfunctions occur due to the false recognition of the pin state as an input signal become possible. Unused pins should be handled as described under Handling of Unused Pins in the manual.
2. Processing at Power-on
   The state of the product is undefined at the moment when power is supplied.
   — The states of internal circuits in the LSI are indeterminate and the states of register settings and pins are undefined at the moment when power is supplied.
   In a finished product where the reset signal is applied to the external reset pin, the states of pins are not guaranteed from the moment when power is supplied until the reset process is completed.
   In a similar way, the states of pins in a product that is reset by an on-chip power-on reset function are not guaranteed from the moment when power is supplied until the power reaches the level at which resetting has been specified.
3. Prohibition of Access to Reserved Addresses
   Access to reserved addresses is prohibited.
   — The reserved addresses are provided for the possible future expansion of functions. Do not access these addresses; the correct operation of LSI is not guaranteed if they are accessed.
4. Clock Signals
   After applying a reset, only release the reset line after the operating clock signal has become stable. When switching the clock signal during program execution, wait until the target clock signal has stabilized.
   — When the clock signal is generated with an external resonator (or from an external oscillator) during a reset, ensure that the reset line is only released after full stabilization of the clock signal. Moreover, when switching to a clock signal produced with an external resonator (or by an external oscillator) while program execution is in progress, wait until the target clock signal is stable.
5. Differences between Products
   Before changing from one product to another, i.e. to one with a different type number, confirm that the change will not lead to problems.
   — The characteristics of MPU/MCU in the same group but having different type numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different type numbers, implement a system-evaluation test for each of the products.

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.

RENESAS

SALES OFFICES Renesas Electronics Corporation http://www.renesas.com

Refer to "http://www.renesas.com/" for the latest and detailed information.

Renesas Electronics America Inc.
2880 Scott Boulevard Santa Clara, CA 95050-2554, U.S.A.
Tel: +1-408-588-6000, Fax: +1-408-588-6130

Renesas Electronics Canada Limited
1101 Nicholson Road, Newmarket, Ontario L3Y 9C3, Canada
Tel: +1-905-898-5441, Fax: +1-905-898-3220

Renesas Electronics Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K
Tel: +44-1628-585-100, Fax: +44-1628-585-900

Renesas Electronics Europe GmbH
Arcadiastrasse 10, 40472 Düsseldorf, Germany
Tel: +49-211-65030, Fax: +49-211-6503-1327

Renesas Electronics (China) Co., Ltd.
7th Floor, Quantum Plaza, No.27 ZhiChunLu Haidian District, Beijing 100083, P.R.China
Tel: +86-10-8235-1155, Fax: +86-10-8235-7679

Renesas Electronics (Shanghai) Co., Ltd.
Unit 204, 205, AZIA Center, No.1233 Lujiazui Ring Rd., Pudong District, Shanghai 200120, China
Tel: +86-21-5877-1818, Fax: +86-21-6887-7858 / -7898

Renesas Electronics Hong Kong Limited
Unit 1601-1613, 16/F., Tower 2, Grand Century Place, 193 Prince Edward Road West, Mongkok, Kowloon, Hong Kong
Tel: +852-2886-9318, Fax: +852 2886-9022/9044

Renesas Electronics Taiwan Co., Ltd.
13F, No. 363, Fu Shing North Road, Taipei, Taiwan
Tel: +886-2-8175-9600, Fax: +886 2-8175-9670

Renesas Electronics Singapore Pte. Ltd.
1 harbourFront Avenue, #06-10, keppel Bay Tower, Singapore 098632
Tel: +65-6213-0200, Fax: +65-6278-8001

Renesas Electronics Malaysia Sdn.Bhd.
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No. 18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: +60-3-7955-9390, Fax: +60-3-7955-9510

Renesas Electronics Korea Co., Ltd.
11F., Samik Lavied' or Bldg., 720-2 Yeoksam-Dong, Kangnam-Ku, Seoul 135-080, Korea
Tel: +82-2-558-3737, Fax: +82-2-558-5141

Colophon 1.1