User's Manual

# LCD Controller

User's Manual

## Multimedia Processor for Mobile Applications EMMA Mobile ™ EV2



Renesas Electronics
www.renesas.com

Rev.7.00 Dec 2011

# 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.

## 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 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 part number, confirm that the change will not lead to problems.

   — The characteristics of MPU/MCU in the same group but having different part numbers may differ because of the differences in internal memory capacity and layout pattern. When changing to products of different part numbers, implement a system-evaluation test for each of the products.

# How to Use This Manual

## 1. Purpose and Target Readers

This manual is designed to provide the user with an understanding of the hardware functions and electrical characteristics of the MCU. It is intended for users designing application systems incorporating the MCU. A basic knowledge of electric circuits, logical circuits, and MCUs is necessary in order to use this manual.

The manual comprises an overview of the product; descriptions of the CPU, system control functions, peripheral functions, and electrical characteristics; and usage notes.

Particular attention should be paid to the precautionary notes when using the manual. These notes occur within the body of the text, at the end of each section, and in the Usage Notes section.

The revision history summarizes the locations of revisions and additions. It does not list all revisions. Refer to the text of the manual for details.

The following documents apply to the EMMA Mobile EV2. Make sure to refer to the latest versions of these documents. The newest versions of the documents listed may be obtained from the Renesas Electronics Web site.

| Document Type | Description | Document Title | Document No. |
|---|---|---|---|
| Data Sheet | Hardware overview and electrical characteristics | EMMA Mobile EV2 Datasheet | R19DS0010EJxxxx |
| User's manual (1chip) | Hardware whole specifications (pin assignments, memory maps, peripheral function specifications, electrical characteristics, timing charts) and operation description | EMMA Mobile EV2 User's manual 1chip | R19UH0036EJxxxx |
| User's manual (module) | Hardware each macro specifications and operation description. | EMMA Mobile EV2 User's manual each module | See below |

| Document Name | Document No. | Document Name | Document No. |
|---|---|---|---|
| 1chip | R19UH0036EJxxxx | DMA Controller | R19UH0043EJxxxx |
| System Management Unit | R19UH0037EJxxxx | LP-DDR/DDR2 Controller | R19UH0039EJxxxx |
| Timer | R19UH0054EJxxxx | SD Memory Card Interface | R19UH0061EJxxxx |
| System Timer | R19UH0055EJxxxx | SDIO Interface | R19UH0042EJxxxx |
| HD Video Decoder | R19UH0056EJxxxx | CF Card Interface | R19UH0062EJxxxx |
| Rotator | R19UH0057EJxxxx | Unified Serial Interface | R19UH0047EJxxxx |
| Resizer | R19UH0058EJxxxx | UART interface | R19UH0040EJxxxx |
| Image Composer | R19UH0038EJxxxx | USB 2.0 Host Controller | R19UH0045EJxxxx |
| LCD Interface (This manual) | R19UH0044EJxxxx | USB 2.0 Function Contoller | R19UH0034EJxxxx |
| ITU-R BT.656 Interface | R19UH0059EJxxxx | IIC Interface | R19UH0052EJxxxx |
| Digital Terrestrial TV Interface | R19UH0048EJxxxx | General Purpose I/O Interface | R19UH0041EJxxxx |
| Camera Interface | R19UH0060EJxxxx | Pulse Width Modulation Interface | R19UH0063EJxxxx |

4 digits of end shows the version.

## 2. Notation of Numbers and Symbols

The notation conventions for register names, bit names, numbers, and symbols used in this manual are described below.

(1) Register Names, Bit Names, and Pin Names

Registers, bits, and pins are referred to in the text by symbols. The symbol is accompanied by the word “register,” “bit,” or “pin” to distinguish the three categories.

Examples the PM03 bit in the PM0 register

P3_5 pin, VCC pin

(2) Notation of Numbers

The indication “b” is appended to numeric values given in binary format. However, nothing is appended to the values of single bits. The indication “h” is appended to numeric values given in hexadecimal format. Nothing is appended to numeric values given in decimal format.

Examples Binary: 11b or 11

Hexadecimal: EFA0h

Decimal: 1234

## 3. Register Notation

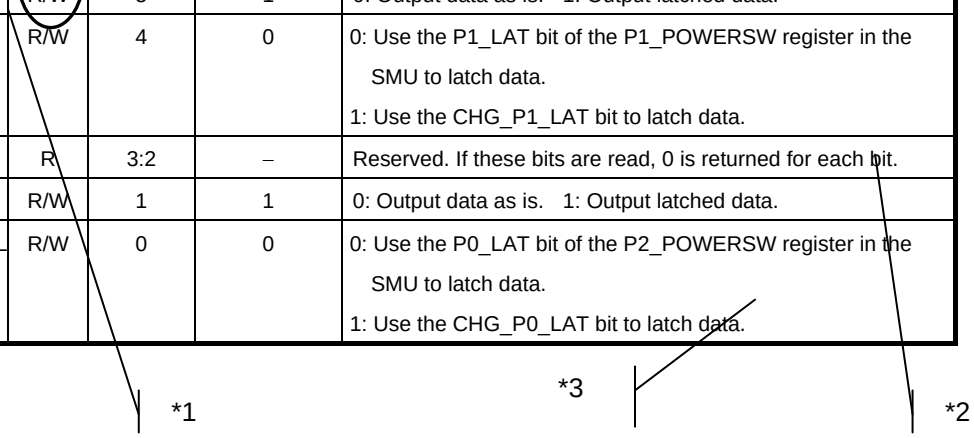
The symbols and terms used in register diagrams are described below.

**x.x.x XXX register**

This register (XXXXXXX: xxxx_xxxxh) …..

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | CHG_P1_LAT | LATCH_P1_SEL | Reserved | | CHG_P0_LAT | CHG_P0_LAT_SEL |

| Name | R/W | Bit No. | After Reset | Description |
|---|---|---|---|---|
| LATCH_P2_SEL | R/W | 8 | 0 | 0: Use the P2_LAT bit of the P2_POWERSW register in the SMU to latch data.<br>1: Use the CHG_P2_LAT bit to latch data. |
| Reserved | R | 7:6 | – | Reserved. If these bits are read, 0 is returned for each bit. |
| CHG_P1_LAT | R/W | 5 | 1 | 0: Output data as is. 1: Output latched data. |
| LATCH_P1_SEL | R/W | 4 | 0 | 0: Use the P1_LAT bit of the P1_POWERSW register in the SMU to latch data.<br>1: Use the CHG_P1_LAT bit to latch data. |
| Reserved | R | 3:2 | – | Reserved. If these bits are read, 0 is returned for each bit. |
| CHG_P0_LAT | R/W | 1 | 1 | 0: Output data as is. 1: Output latched data. |
| CHG_P0_LAT_SEL | R/W | 0 | 0 | 0: Use the P0_LAT bit of the P2_POWERSW register in the SMU to latch data.<br>1: Use the CHG_P0_LAT bit to latch data. |




*1

R/W: Read and Write.

R: Read only.

W: Write only.

-: Nothing is assigned.

*2

Reserved bit.

Reserved bit. Set to specified value.

*3

・ Nothing is assigned.

Nothing is assigned to the bit. As the bit may be used for future functions, if necessary, set to 0.

・ Do not set to a value.

Operation is not guaranteed when a value is set.

・ Function varies according to the operating mode.

The function of the bit varies with the peripheral function mode. Refer to the register diagram for information on the individual modes.

## 4. List of Abbreviations and Acronyms

| Abbreviation | Full Form |
|---|---|
| ACIA | Asynchronous Communication Interface Adapter |
| bps | bits per second |
| CRC | Cyclic Redundancy Check |
| DMA | Direct Memory Access |
| DMAC | Direct Memory Access Controller |
| GSM | Global System for Mobile Communications |
| Hi-Z | High Impedance |
| IEBus | Inter Equipment Bus |
| I/O | Input/Output |
| IrDA | Infrared Data Association |
| LSB | Least Significant Bit |
| MSB | Most Significant Bit |
| NC | Non-Connect |
| PLL | Phase Locked Loop |
| PWM | Pulse Width Modulation |
| SFR | Special Function Register |
| SIM | Subscriber Identity Module |
| UART | Universal Asynchronous Receiver/Transmitter |
| VCO | Voltage Controlled Oscillator |

| Abbreviation | Full Form |
|---|---|
| APB | Advanced Peripheral Bus |
| BG | Back Ground |
| bpp | Bit Per Pixel |
| FIFO | First In, First Out |
| GPIO | General Purpose I/O |
| HDMI | High-Definition Multimedia Interface |
| IMC | Image Composer |
| LSB | Least Significant Bit |
| MEMC | Memory Controller |
| QoS | Quality of Service |
| WB | Write Back |

# Table of Contents

# 1. Overview

The LCD controller outputs synchronization signals and video signals to an LCD panel externally connected to EM/EV. This controller has the YUV output function for HDMI Driver I/F.
Since the LCD controller usually operates in association with the image composer module, some descriptions in this user's manual assume that the reader knows the functions of the image composer module. Also see the **Multimedia Processor for Mobile Applications - Image Composer User's Manual (R19UH0038EJ)**.

## 1.1 Features

The main features of the LCD controller are as follows.

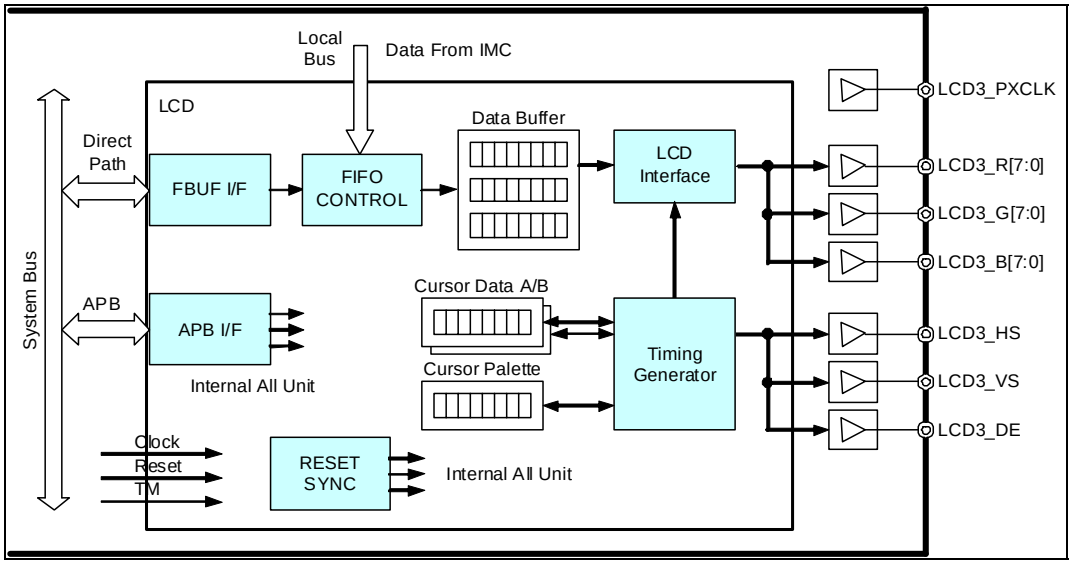
- Output size
  - Horizontal direction : up to 4,090 pixels. (Sync cycle : up to 8,190 CLK)
  - Vertical direction : up to 4,094 lines. (Sync cycle : up to 8,190 HSYNC)
- Output format
  - RGB : RGB565, RGB666, RGB888 (up to 16.77 million colors)
- LCD interface
  - Pixel clock output (PXCLK)
  - Horizontal synchronization signal (HSYNC)
  - Vertical synchronization signal (VSYNC)
  - Data bus enable (DATAENABLE)
  - Data bus (RDATA[7:0], GDATAQ[7:0], BDATA[7:0])
- YUV interface
  - YUV data clock output (YUV_CLK)
  - Horizontal synchronization signal (YUV_HS)
  - Vertical synchronization signal (YUV_VS)
  - Data bus enable (YUV_DE)
  - Data bus (YUV_DATA[15:0])
- Data format
  - Output to LCD panel: RGB565 or RGB666 or RGB888 selectable
  - Input from memory: RGB565 or RGB666 or RGB888 selectable. (RGB888 and in case of YUV output are applicable to only RGB888 input.)

    Operating in conjunction with the IMC, formats of YUV422/YUV420 (YUV Interlace, Y/UV2 plane, Y/U/V3 plane) can also be supported.
- Hardware cursor
  - The H/W cursor function of 64 × 64 pixels and 256 colors (RGB565 palette form and transparent color are included.)
- Data buffer
  - It has 32 bits × 1024 words of data buffer built-in (It's used as FIFO.)
- Corresponding to a power-saving wait mode by DirectPath connection with MEMC macro.
- It's possible to adjust a LCD interface signal to a programmable by register setting.
- Corresponding to the simple QoS function connected in the inner FIFO state.
- The simple double resizing function (It's effective only Direct Path use.)

- It corresponds in an irregular stop period of internal system clock.

## 1.2 Function Block Diagram

**Figure 1-1. Function Block Diagram**



# 2. Pin functions

| Pin Name | I/O | Function | Alternate Function Pin |
|---|---|---|---|
| LCD3_PXCLK | Output | Pixel clock | YUV3_CLK_O, GPIO_018 |
| LCD3_PXCLKB | Output | Pixel clock | GPIO_019 |
| LCD3_CLK_I | Output | Input clock | YUV3_CLK_I, GPIO_020 |
| LCD3_R7 | Output | Red data | TP33_CTRL, GPIO_039 |
| LCD3_R6 | Output | Red data | TP33_CLK, GPIO_038 |
| LCD3_R5 | Output | Red data | GPIO_037 |
| LCD3_R4 | Output | Red data | GPIO_036 |
| LCD3_R3 | Output | Red data | GPIO_035 |
| LCD3_R2 | Output | Red data | GPIO_034 |
| LCD3_R1 | Output | Red data | GPIO_033 |
| LCD3_R0 | Output | Red data | GPIO_032 |
| LCD3_G7 | Output | Green data | YUV3_D7, TP33_DATA7 |
| LCD3_G6 | Output | Green data | YUV3_D6, TP33_DATA6 |
| LCD3_G5 | Output | Green data | YUV3_D5, TP33_DATA5 |
| LCD3_G4 | Output | Green data | YUV3_D4, TP33_DATA4 |
| LCD3_G3 | Output | Green data | YUV3_D3, TP33_DATA3 |
| LCD3_G2 | Output | Green data | YUV3_D2, TP33_DATA2 |
| LCD3_G1 | Output | Green data | YUV3_D1, TP33_DATA1, GPIO_041 |
| LCD3_G0 | Output | Green data | YUV3_D0, TP33_DATA0, GPIO_040 |
| LCD3_B7 | Output | Blue data | YUV3_D15, TP33_DATA15 |
| LCD3_B6 | Output | Blue data | YUV3_D14, TP33_DATA14 |
| LCD3_B5 | Output | Blue data | YUV3_D13, TP33_DATA13 |
| LCD3_B4 | Output | Blue data | YUV3_D12, TP33_DATA12 |
| LCD3_B3 | Output | Blue data | YUV3_D11, TP33_DATA11 |
| LCD3_B2 | Output | Blue data | YUV3_D10, TP33_DATA10 |
| LCD3_B1 | Output | Blue data | YUV3_D9, TP33_DATA9, GPIO_043 |
| LCD3_B0 | Output | Blue data | YUV3_D8, TP33_DATA8, GPIO_042 |
| LCD3_HS | Output | Horizontal synchronization | YUV3_HS, GPIO_021 |
| LCD3_VS | Output | Vertical synchronization | YUV3_VS, GPIO_022 |
| LCD3_DE | Output | Data enable | YUV3_DE, GPIO_023 |

# 3. Registers

## 3.1 Register List

Base address: E004_0000H

**Remark** Among addresses E004_0000H to E004_FFFCH, the addresses not listed in the following tables are reserved. Do not access reserved registers. An undefined value is returned for a read access.

Registers marked with ○ in the Frame Sync column are two-stage registers with which settings made in the registers are latched to the macro and take effect when the frame start signal immediately after the setting change is received; that is, the beginning of a frame processed in the LCD module and it is independent from the VSYNC pin operation. The timing is the same as occurrence of a frame interrupt.
Registers marked with × are registers with which setting changes made in the register take effect immediately. Changing the settings during display output is prohibited.

(1/3)

| Address | Register Name | Symbol | R/W | Frame Sync | After Reset |
|---|---|---|---|---|---|
| Function setting register | | | | | |
| 0000H | Control register | LCD_CONTROL | R/W | × | 0000_0000H |
| 0004H | Simple QoS setting register | LCD_QOS | R/W | × | 0000_0000H |
| 0008H | Data request cycle register | LCD_DATAREQ | R/W | × | 0000_0000H |
| 000CH | Reserved | – | – | – | – |
| Display start register | | | | | |
| 0010H | Display register | LCD_LCDOUT | R/W | ○ | 0000_0000H |
| 0014H | Access bus select register | LCD_BUSSEL | R/W | ○ | 0000_0000H |
| 0018H | Status register | LCD_STATUS | R | ○ | 0000_0000H |
| 001CH | Fixed-color output value register | LCD_BACKCOLOR | R/W | ○ | 0000_0000H |
| Fmare buffer setting register | | | | | |
| 0020H | Display area address register | LCD_AREAADR_ODD | R/W | ○ | 0000_0000H |
| 0024H | Display area address register | LCD_AREAADR_EVEN | R/W | ○ | 0000_0000H |
| 0028H | Address addition value register | LCD_HOFFSET | R/W | ○ | 0000_0000H |
| 002CH | Input format register | LCD_IFORMAT | R/W | ○ | 0000_0000H |
| 0030H | Simple resize register | LCD_RESIZE | R/W | ○ | 0000_0000H |
| 0034H to 003CH | Reserved | – | – | – | – |
| Display setting register | | | | | |
| 0040H | Horizontal direction total register | LCD_HTOTAL | R/W | × | 0000_0000H |
| 0044H | Horizontal direction display area register | LCD_HAREA | R/W | × | 0000_0000H |
| 0048H | Horizontal synchronization edge 1 register | LCD_HEDGE1 | R/W | × | 0000_0000H |
| 004CH | Horizontal synchronization edge 2 register | LCD_HEDGE2 | R/W | × | 0000_0000H |
| 0050H | Vertical direction total register | LCD_VTOTAL | R/W | × | 0000_0000H |

(2/3)

| Address | Register Name | Symbol | R/W | Frame Sync | After Reset |
|---|---|---|---|---|---|
| 0054H | Vertical direction display area register | LCD_VAREA | R/W | × | 0000_0000H |
| 0058H | Vertical synchronization edge 1 register | LCD_VEDGE1 | R/W | × | 0000_0000H |
| 005CH | Vertical synchronization edge 2 register | LCD_VEDGE2 | R/W | × | 0000_0000H |
| Interrupt control register | | | | | |
| 0060H | Interrupt status register | LCD_INTSTATUS | R | × | 0000_0000H |
| 0064H | Interrupt raw status register | LCD_INTRAWSTATUS | R | × | 0000_0000H |
| 0068H | Interrupt enable set register | LCD_INTENSET | R/W | × | 0000_0000H |
| 006CH | Interrupt enable clear register | LCD_INTENCLR | W | × | 0000_0000H |
| 0070H | Interrupt source clear register | LCD_INTFFCLR | W | × | 0000_0000H |
| 0074H | Frame count interrupt setting register | LCD_FRAMECOUNT | R/W | × | 0000_0000H |
| 0078H to 007CH | Reserved | – | – | – | – |
| YUV output relationship register | | | | | |
| 0080H | RGB2YUV transform coefficient register Y0 | LCD_COEF_Y0 | R/W | × | 0000_0000H |
| 0084H | RGB2YUV transform coefficient register Y1 | LCD_COEF_Y1 | R/W | × | 0000_0000H |
| 0088H | RGB2YUV transform coefficient register Y2 | LCD_COEF_Y2 | R/W | × | 0000_0000H |
| 008CH | RGB2YUV transform coefficient register Y3 | LCD_COEF_Y3 | R/W | × | 0000_0000H |
| 0090H | RGB2YUV transform coefficient register U0 | LCD_COEF_U0 | R/W | × | 0000_0000H |
| 0094H | RGB2YUV transform coefficient register U1 | LCD_COEF_U1 | R/W | × | 0000_0000H |
| 0098H | RGB2YUV transform coefficient register U2 | LCD_COEF_U2 | R/W | × | 0000_0000H |
| 009CH | RGB2YUV transform coefficient register U3 | LCD_COEF_U3 | R/W | × | 0000_0000H |
| 00A0H | RGB2YUV transform coefficient register V0 | LCD_COEF_V0 | R/W | × | 0000_0000H |
| 00A4H | RGB2YUV transform coefficient register V1 | LCD_COEF_V1 | R/W | × | 0000_0000H |
| 00A8H | RGB2YUV transform coefficient register V2 | LCD_COEF_V2 | R/W | × | 0000_0000H |
| 00ACH | RGB2YUV transform coefficient register V3 | LCD_COEF_V3 | R/W | × | 0000_0000H |
| 00B0H to 00BCH | Reserved | – | – | – | – |
| 00C0H | BYTELANE Register | LCD_BYTELANE | R/W | × | 0000_0000H |
| 00C4H to 00CCH | Reserved | – | – | – | – |
| 00D0H | Vertical synchronization signal control register | LCD_VSYNC_CONT | R/W | × | 0000_0000H |
| 00D4H | Reserved | – | – | – | – |
| 00D8H | Vertical synchronization edge offset register (Odd field) | LCD_VOFFSET_ODD | R/W | × | 0000_0000H |
| 00DCH | Vertical synchronization edge offset register (Even field) | LCD_VOFFSET_EVEN | R/W | × | 0000_0000H |
| 00E0H | Vertical direction total register (Even field) | LCD_VAREA_EVEN | R/W | × | 0000_0000H |
| 00E4H | Reserved | – | – | – | – |
| 00E8H | Vertical synchronization edge 1 register (Even field) | LCD_VEDGE1_EVEN | R/W | × | 0000_0000H |
| 00ECH | Vertical synchronization edge 2 register (Even field) | LCD_VEDGE2_EVEN | R/W | × | 0000_0000H |
| 00F0H to 01FCH | Reserved | – | – | – | – |

(3/3)

| Address | Register Name | Symbol | R/W | Frame Sync | After Reset |
|---|---|---|---|---|---|
| Cursor relationship register | | | | | |
| 0200H | Cursor enable register | LCD_CURSOR_EN | R/W | ○ | 0000_0000H |
| 0204H | Cursor operation register | LCD_CURSOR_OPE | R/W | ○ | 0000_0000H |
| 0208H | Cursor horizontal position register | LCD_CURSOR_POSH | R/W | ○ | 0000_0000H |
| 020CH | Cursor vertical position register | LCD_CURSOR_POSV | R/W | ○ | 0000_0000H |
| 0210H | Cursor data RAM select register | LCD_CURSOR_RAMSEL | R/W | ○ | 0000_0000H |
| 0214H | Cursor status register | LCD_CURSOR_STATUS | R/W | ○ | 0000_0000H |
| 0218H to 03FCH | Reserved | – | – | – | – |
| 0400H to 07FCH | Cursor palette RAM register | LCD_CURSOR_TABLE | R/W | × | Variable (RAM) |
| 0800H to 0FFCH | Reserved | – | – | – | – |
| 1000H to 1FFCH | Cursor data RAM 0 register | LCD_CURSOR_DATA0 | R/W | × | Variable (RAM) |
| 2000H to 2FFCH | Cursor data RAM 1 register | LCD_CURSOR_DATA1 | R/W | × | Variable (RAM) |
| 3000H to FFFCH | Reserved | – | – | – | – |

## 3.2 Register Details

### 3.2.1 Control register

This register (LCD_CONTROL:E004_0000H) controls the basic LCD controller operation.
Changing the settings during LCD controller operation is prohibited.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| LCLK_SEL | | Reserved | | | | OUT_SEL | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | PI_SEL | OFORMAT | | Reserved | HPOL | VPOL | ENPOL |

(1/2)

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:16 | 0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| LCLK_SEL | R/W | 15:14 | 00b | Input clock select<br>0x : PLL generation CLK<br>10 : Reserved<br>11 : 3.3V system PXCLK input |
| Reserved | R | 13:10 | 0H | Reserved. When these bits are read, 0 is returned for each bit. |
| OUT_SEL | R/W | 9:8 | 00b | Sets the LCD output mode<br>00: RGB output The RGB data designated in OFORMAT is output.<br>01: YUV output<br>10 : Reserved<br>11 : Setting prohibited |
| Reserved | R | 7 | 0 | Reserved. When these bits are read, 0 is returned for each bit. |
| PI_SEL | R/W | 6 | 0 | Set the scan mode.<br>0: Progressive<br>1: Interlace |
| OFORMAT | R/W | 5:4 | 00b | Sets the format of data output from pins. For details, see **4.1.2 Format conversion.**<br>00: RGB888<br>01: RGB666<br>10: RGB565<br>11: Setting prohibited |
| Reserved | R/W | 3 | 0 | Reserved. Set 0 as this bit. |

(2/2)

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| HPOL | R/W | 2 | 0 | Sets the polarity of horizontal synchronization signals. For details, see **4.1.5 Horizontal synchronization signal**.)<br>0: Positive logic<br>1: Negative logic<br>Positive logic is assumed when a low-level pulse is output during a horizontal blanking interval. |
| VPOL | R/W | 1 | 0 | Sets the polarity of vertical synchronization signals. For details, see **4.1.6 Vertical synchronization signal**.<br>0: Positive logic<br>1: Negative logic |
| ENPOL | R/W | 0 | 0 | Sets the active level of enable signals. For details, see **4.1.7 Enable signal.**<br>0: Active High<br>1: Active Low |

### 3.2.2 Simple QoS setting register

This register (LCD_QOS: E004_0004H) sets the simple QoS function of EM/EV.

The simple QoS function is used to prevent overrun and under-run from occurring in an image system function block.

When a QoS request is issued from an image system function block, the bus switch temporarily gives a higher priority for accesses from that QoS request function block, which reduces the access latency.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | QOSEN | QOSVALUE | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| QOSVALUE | | | | | | | |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:11 | 00_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| QOSEN | R/W | 10 | 0 | Sets whether to enable the simple QoS function.<br>0: Disable<br>1: Enable |
| QOSVALUE | R/W | 9:0 | 000H | A QoS request is issued when the free space in FIFO lowers the value set in this register. |

### 3.2.3 Data request cycle register

This register (LCD_DATAREQ: E004_0008H) sets the timing at which a request for reading data from a frame buffer is issued.

The set values are determined based on the available space in the data buffer in the LCD controller.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | DATAREQ | | |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:3 | 0000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| DATAREQ | R/W | 2:0 | 000b | Sets data request output timing with available data size in FIFO. For details, see **4.3.5 Data request cycle setting**.<br>000: 99.2% (1016 words)<br>001: 75.0% (768 words)<br>010: 50.0% (512 words)<br>011: 37.5% (384 words)<br>100: 25.0% (256 words)<br>101: 12.5% (128 words)<br>110: 6.75% (64 words)<br>111: 3.13% (32 words) |

When using DirectPath, 8 words are read by once's data request. Therefore an original value of DATAREQ is 1016 words which subtracted 8 from 1024 with the FIFO size.

When using Local Bus, a data request to IMC uses this register set value.

### 3.2.4 Display register

This register (LCD_LCDOUT: E004_0010H) is used to start data output to the LCD panel.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

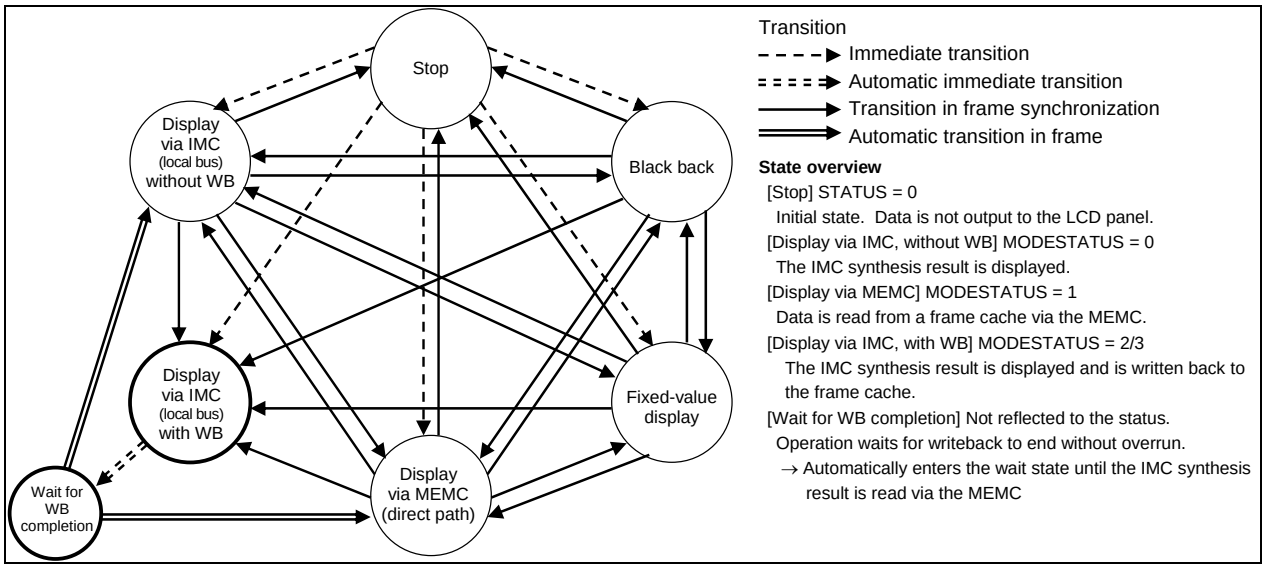
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | LCDOUT |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:1 | 0000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| LCDOUT | R/W | 0 | 0 | Starts display on the LCD panel.<br>0: Stops display.<br>1: Starts display. |

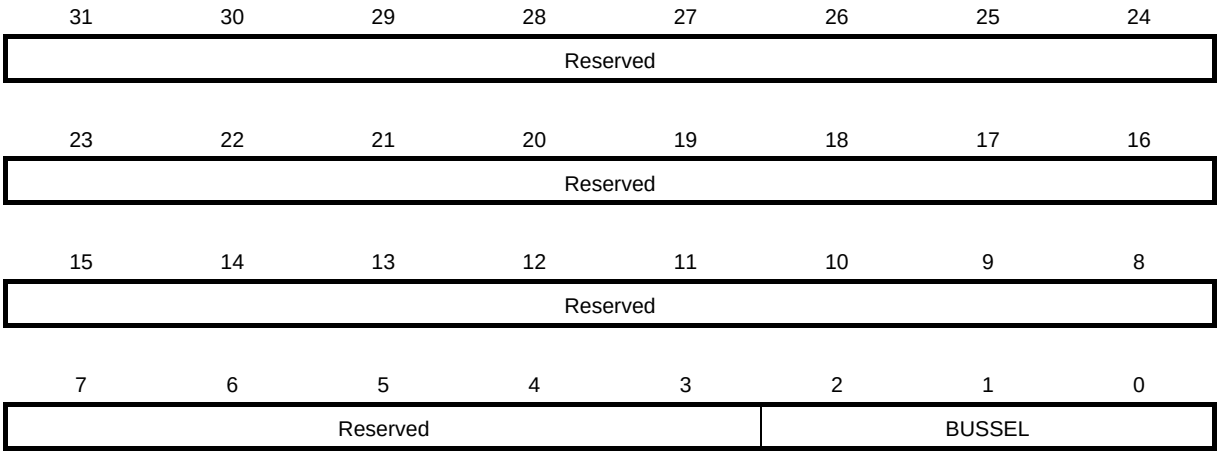
Startup (LCDOUT = 1) takes effect immediately, but stop (LCDOUT = 0) takes effect in frame synchronization. Up to one frame period may be required from issuing of a stop request to the actual stop. The timing at which the LCD display actually stops can be checked by detecting a display stop interrupt.

**Figure 3-1. Status Transition**



### 3.2.5 Access bus select register

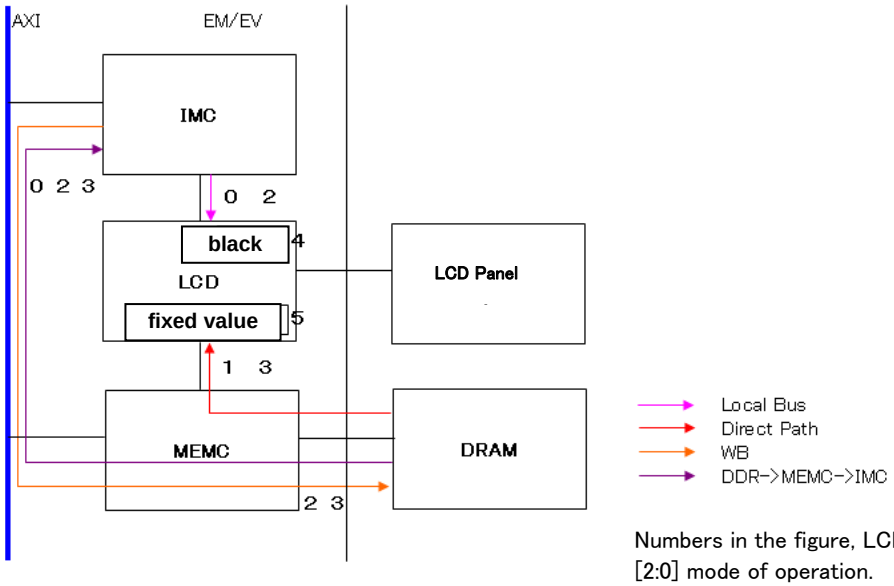
This register (LCD_BUSSEL: E004_0014H) switches the operation modes - normal operation mode in which data is displayed on the LCD via the IMC, or VGA standby mode on the MEMC via the DirectPath or the mode in which fixed values are output for display.
Mode transition triggered by setting this register is performed in frame synchronization. The values set to the BUSSEL bit take effect at the next vertical synchronization interrupt, and are reflected in the MODESTATUS bit described later.




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:3 | 0000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| BUSSEL | R/W | 2:0 | 000b | Sets the LCD controller operation mode.<br>000: Local bus between IMC and LCD controller (without WB)<br>001: DirectPath between MEMC and LCD controller.<br>010: Local bus + WB. Waits for WB completion and automatically switches to local bus mode (without WB).<br>011: Local bus + WB. Waits for WB completion and automatically switches to DirectPath.<br>100: Black back display mode<br>101: Fixed-value display mode (RGB values are set with the LCD_BACKCOLOR register.)<br>110, 111: Setting prohibited |

When the BUSSEL bit is set to 2 or 3, the LCD controller issues a data request to the IMC, at the same time as issuing a WB request. When WB operation requested here completes, the local bus (without WB) (when BUSSEL = 2) , DirectPath (when BUSSEL = 3) is automatically selected for the next frame transfer.
If the WB operation fails (IMC buffer overrun), the LCD controller again issues a data request and a WB request to the IMC at the next frame.

**Figure 3-2. BUSSEL (Operation mode) and data bus**




Numbers in the figure, LCD_BUSSEL register bit [2:0] mode of operation.

### 3.2.6 Status register

This register (LCD_STATUS: E004_0018H) indicates the LCD controller operating status. The LCD controller status can be checked by polling this register. The LCD controller status in a frame is reflected when a frame interrupt occurs.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | MODESTATUS | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | STATUS |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:11 | 00_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| MODESTATUS | R | 10:8 | 000b | Indicates the LCD controller operating status. (Updated upon a frame synchronization interrupt.)<br>000: Operating in local bus (without WB) between IMC and LCD controller.<br>001: Operating in local bus between MEMC and DirectPath<br>010: WB has performed in local bus and waiting for WB completion. Following WB completion, the LCD controller automatically enters operation in local bus mode.<br>011: WB has performed in local bus and waiting for WB completion. The frame following WB completion is automatically set to 001.<br>100: Operating in black back display mode.<br>101: Operating in fixed-value display mode (RGB values are set with the LCD_BACKCOLOR register.)<br>110, 111: – |
| Reserved | R | 7:1 | 00H | Reserved. When these bits are read, 0 is returned for each bit. |
| STATUS | R | 0 | 0 | Indicates the LCD status.<br>0: LCD display is off.<br>1: LCD display is on. |

In black back display mode and fixed-value display mode, the LCD controller can operate individually without using the IMC or MEMC, because pixel data is generated in the LCD controller.

### 3.2.7 Fixed-color output value register

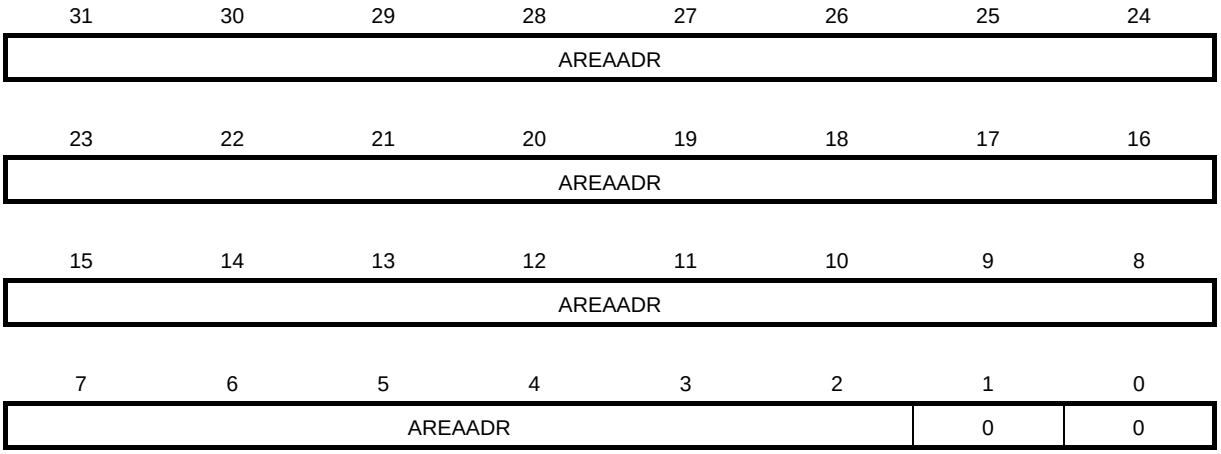
When the fixed-color display mode (BUSSEL = 5) is set in the access bus select register (LCD_BUSSEL), the LCD controller does not read data from a frame buffer but outputs fixed-value data to the LCD panel.
This register (LCD_BACKCOLOR: E004_001CH) sets the fixed values for this mode.
The set values are captured to the circuit at the beginning of reception of a frame.


| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:22 | 00H | Reserved. When these bits are read, 0 is returned for each bit. |
| BGRED | R/W | 23:16 | 00H | Sets the value of red output by the LCD controller in fixed-color display mode. |
| BGGREEN | R/W | 15:8 | 00H | Sets the value of green output by the LCD controller in fixed-color display mode. |
| BGBLUE | R/W | 7:0 | 00H | Sets the value of blue output by the LCD controller in fixed-color display mode. |

When RGB565 or RGB666 is set with the OFORMAT bit of the control register (LCD_CONTROL), the higher bits of the values set to the BGRED, BGGREEN and BGBLUE bits are selected and output to the higher bits of the LCD_R and LCD_B pins.
When the BG layer is set to fixed-color display mode (IMC_BG_FORMAT = 4 and IMC_CONTROL bit[0] = 1) in the IMC macro, the RGB values set in this register are used.
When this register is read, the values that become valid at the next frame (first-stage values) are read.

### 3.2.8 Display area address register

This register (LCD_AREAADR_ODD: E004_0020H and LCD_AREAADR_EVEN : E004_0024H) sets the starting address of frame buffer.
When using in Interlace mode, start address in a odd frame is set as LCD_AREAADR_ODD, start address in a even frame is set as LCD_AREAADR_EVEN.
When using in Progressive mode, start address in a frame buffer is set as LCD_AREAADR_ODD.
The frame buffer set in this register is used in the DirectPath mode. The setting is also used for the buffer for white back, when the IMC macro is operating in LCD-synchronous mode. This is a frame-synchronous register and its set values take effect upon reception of a frame start signal.




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| AREAADR | R/W | 31:2 | 0000_0000H | Sets the starting address of frame buffer.<br>Set the address using the byte address of 32-bit boundary.<br>For details, see **4.3.1 Frame buffer**. |
| – | R | 1:0 | 00b | Fixed to 0. When these bits are read, 0 is returned for each bit. |

When this register is read, the values that become valid at the next frame (first-stage values) are read.

### 3.2.9 Address addition value register

This register (LCD_HOFFSET: E004_0028H) sets the total byte count in the horizontal direction in a frame buffer area.

The frame buffer set in this register is used in the DirectPath mode. The setting is also used for the buffer for white back, when the IMC macro is operating in LCD-synchronous mode. This is a frame-synchronous register and its set values take effect upon reception of a frame start signal.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | HOFFSET | | | | |

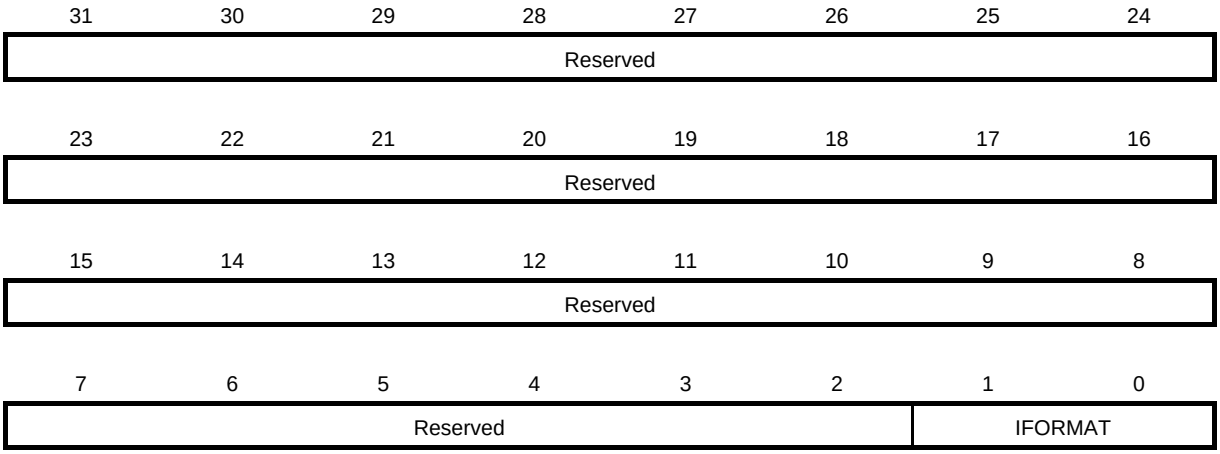
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| HOFFSET | | | | | | 0 | 0 |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 0_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| HOFFSET | R/W | 12:0 | 0000H | Sets the total byte count in the horizontal direction in a frame buffer area. (The lower 2 bits are fixed to 0.) |

When this register is read, the values that become valid at the next frame (first-stage values) are read.

### 3.2.10 Input format register

This register (LCD_IFORMAT: E004_002CH) specifies the image format in a frame buffer area.
The format set in this register is referenced from the IMC macro and used as the output format in the IMC macro.
This is a frame-synchronous register and its set values take effect upon reception of a frame start signal.




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:2 | 0000_000H | Reserved. When these bits are read, 0 is returned for each bit. |
| IFORMAT | R/W | 1:0 | 00b | Sets the input data format. For details, see **4.1.2 Format conversion**.<br>00: RGB888<br>01: RGB666<br>10: RGB565<br>11: Setting prohibited |

When this register is read, the values that become valid at the next frame (first-stage values) are read.
When changing the values of this register during operation, the settings related to frame buffers (start address and address addition value) must also be changed.

### 3.2.11 Simple resize register

This register (LCD_RESIZE: E004_0030H) is expanded double inside the LCD macro and indicated setting is performed. When the data read from a buffer is output to LCD interface.
Only when reading from DirectPath, it's effective. It's ignored by the time of Local Bus mode through IMC.
The set value is just after start-of-frame timing for a V synchronous register, and this becomes effective.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

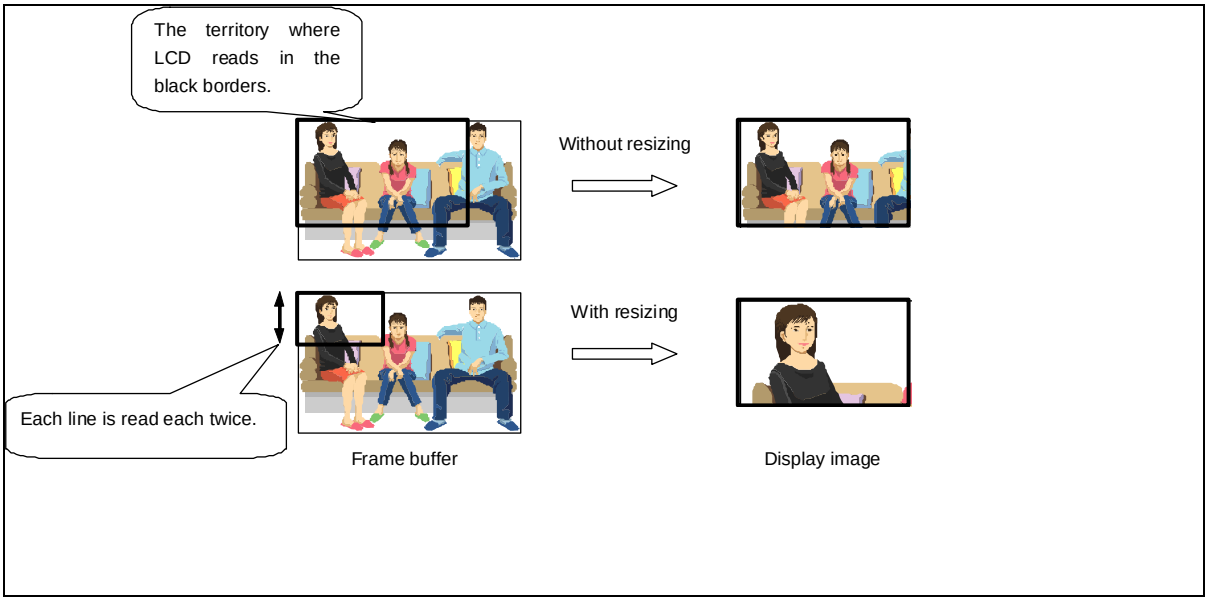
| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | RESIZE |

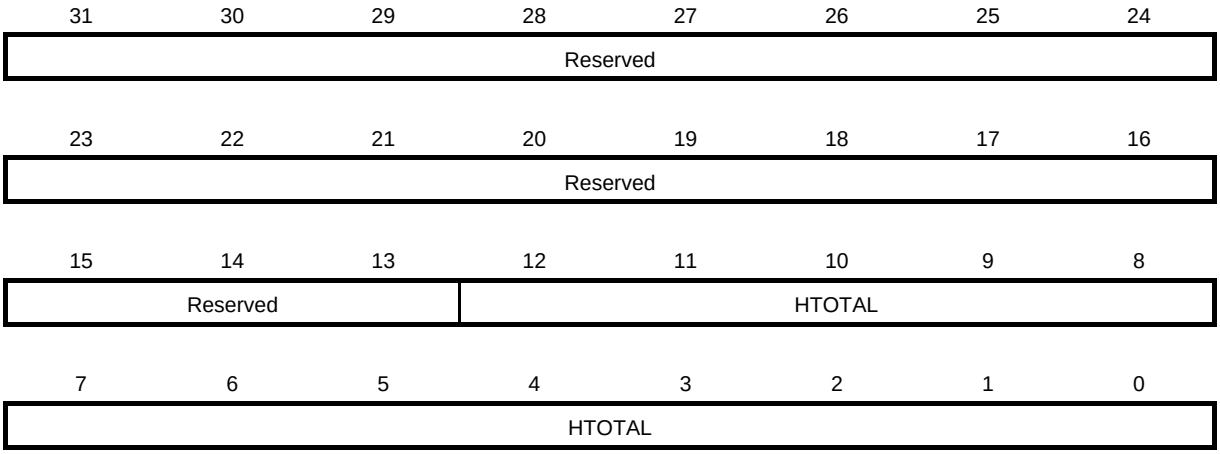
| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:1 | 0000_000H | Reserved. When these bits are read, 0 is returned for each bit. |
| RESIZE | R/W | 0 | 0 | The simple resizing function is made effective..<br>0: Resize invalid<br>1: Resize effective |

When this register is read, the values that become valid at the next frame (first-stage values) are read.
Expansion processing of horizontal direction is performed inside the LCD macro, but 2 lines are to read the same linear data from a frame buffer, and the verticalness direction is achieved. LCD macro will be the read amount of data half as a result.



### 3.2.12 Horizontal direction total register

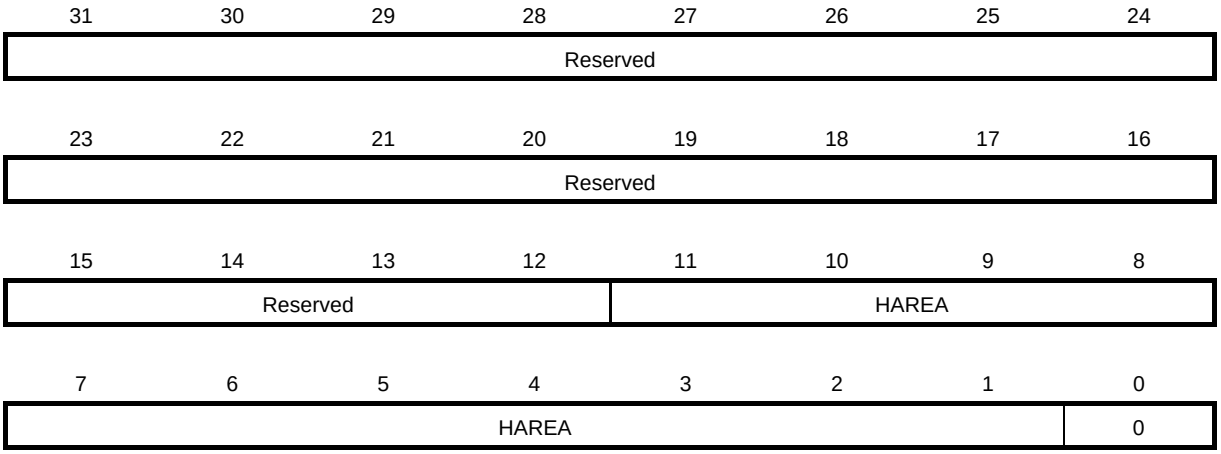
This register (LCD_HTOTAL: E004_0040H) sets the number of pixel clock cycles (HSYNC cycles) in the horizontal direction.




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 0_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| HTOTAL | R/W | 12:0 | 0000H | Sets the number of pixel clock cycles in the horizontal direction.<br>For details, see **4.1.4 Display area, and horizontal and vertical blanks**. |

### 3.2.13 Horizontal direction display area register

This register (LCD_HAREA: E004_0044H) sets the number of display pixels in the horizontal direction, in pixel units.




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:12 | 0_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| HAREA | R/W | 11:0 | 000H | Sets the number of display pixels in the horizontal direction.<br>(The lowest bit is fixed to 0 (2-pixel units).)<br>For details, see **4.1.4 Display area, and horizontal and vertical blanks**. |

### 3.2.14 Horizontal synchronization edge 1 register

This register (LCD_HEDGE1: E004_0048H) sets the position of the first edge of a horizontal synchronization signal by the X coordinate (number of pixel clocks).

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | HEDGE1 | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| HEDGE1 | | | | | | | |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 0_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| HEDGE1 | R/W | 12:0 | 0000H | Sets the position of the first edge of a horizontal synchronization signal by the X coordinate.<br>For details, see **4.1.5 Horizontal synchronization signal**. |

### 3.2.15 Horizontal synchronization edge 2 register

This register (LCD_HEDGE2: E004_004CH) sets the position of the second edge of a horizontal synchronization signal by the X coordinate (number of pixel clocks).

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | HEDGE2 | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| HEDGE2 | | | | | | | |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 0_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| HEDGE2 | R/W | 12:0 | 0000H | Sets the position of the second edge of a horizontal synchronization signal by the X coordinate.<br>For details, see **4.1.5 Horizontal synchronization signal**. |

### 3.2.16 Vertical direction total register

This register (LCD_VTOTAL: E004_0050H) sets the total number of lines in the vertical direction (VSYNC cycles = HSYNC count).

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | VTOTAL | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VTOTAL | | | | | | | |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 0_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| VTOTAL | R/W | 12:0 | 0000H | Sets the total number of lines in the vertical direction.<br>For details, see **4.1.4 Display area, and horizontal and vertical blanks**. |

### 3.2.17 Vertical direction display area register

This register (LCD_VAREA: E004_0054H) sets the number of display lines in the vertical direction.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

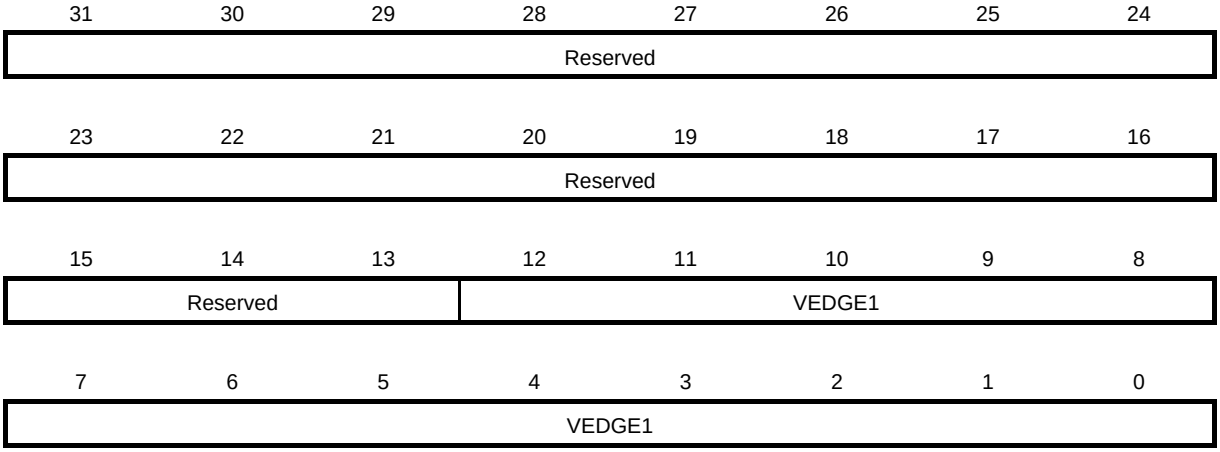
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | VAREA | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| VAREA | | | | | | | |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:12 | 0_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| VAREA | R/W | 11:0 | 000H | Sets the number of display lines in the vertical direction.<br>For details, see **4.1.4 Display area, and horizontal and vertical blanks**. |

### 3.2.18 Vertical synchronization edge 1 register

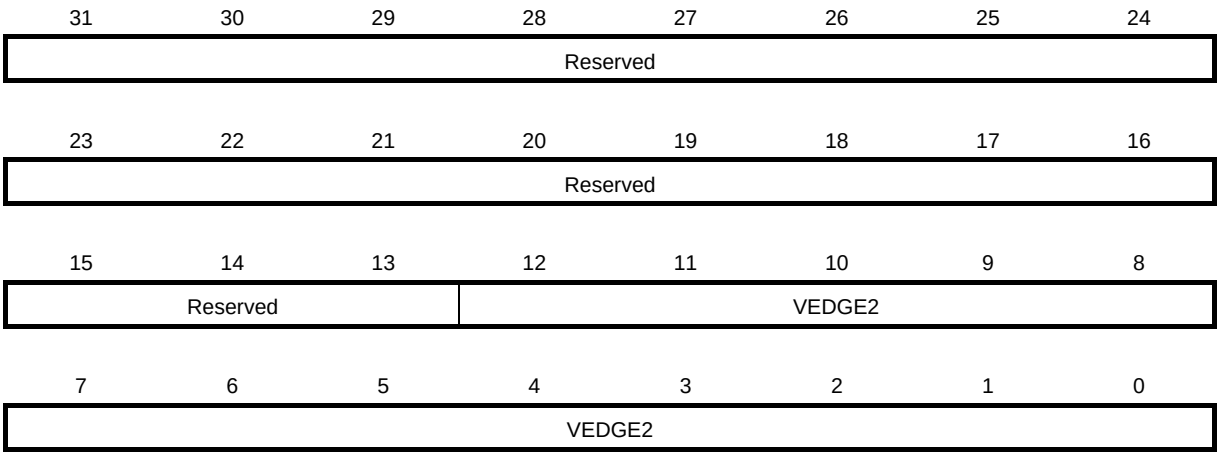
This register (LCD_VEDGE1: E004_0058H) sets the position of the first edge of a vertical synchronization signal by the Y coordinate (HSYNC count).




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 0_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| VEDGE1 | R/W | 12:0 | 0000H | Sets the position of the first edge of a vertical synchronization signal by the Y coordinate.<br>For details, see **4.1.6 Vertical synchronization signal**. |

### 3.2.19 Vertical synchronization edge 2 register

This register (LCD_VEDGE2: E004_005CH) sets the position of the second edge of a vertical synchronization signal by the Y coordinate.




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 0_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| VEDGE2 | R/W | 12:0 | 0000H | Sets the position of the second edge of a vertical synchronization signal by the Y coordinate.<br>For details, see **4.1.6 Vertical synchronization signal**. |

### 3.2.20 Interrupt setting registers

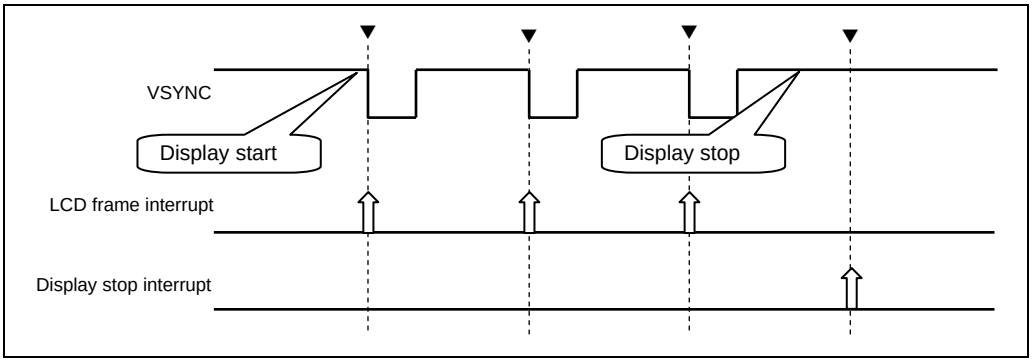
Interrupt setting registers set various interrupt parameters. The LCD controller issues six types of interrupts. Control of each interrupt is assigned to each bit of the interrupt setting register. For details, refer to **Table 3-1**.

**Table 3-1. Interrupts**

| Interrupt Name | Source | Bit Assignment |
|---|---|---|
| FIELD interrupt | This interrupt is issued when indication of FIELD is started in case of Interlace mode. | 5 |
| VGA standby shift end interrupt | This interrupt is issued to report the state that power to the L1 domain can be shut down after WB completion, when BUSSEL is set to 3. | 4 |
| Frame count interrupt | This interrupt is issued for each of the specified number of frames. | 3 |
| Display stop interrupt | This interrupt is issued if the display register value is 0 when frame display ends. | 2 |
| Underrun interrupt | This interrupt is issued when an under-run occurs in the LCD internal buffer. | 1 |
| LCD frame interrupt | This interrupt is issued when a frame display starts. | 0 |

An LCD frame interrupt is issued when a frame display start signal is detected. Therefore, if the register setting is changed immediately after an LCD frame interrupt is issued, the change takes effect at the next frame.
An under-run interrupt is issued when a buffer under-run occurs and capturing of data currently being transferred is stopped. Ordinary operation is resumed when the next frame transfer starts, according to various settings.
When shifting to DirectPath automatically from a WB request, WB access traces that I reached a memory certainly and notifies of the completion in interrupt only in (use method which is usually performed before power supply off in L1 territory).

**Figure 3-3. Issuance Timing of Frame Interrupt and Display Stop Interrupt**



### (1) Interrupt status register

This register (LCD_INTSTATUS: E004_0060H) is a read-only register that indicates the statuses of interrupt sources. The statuses of the interrupt sources enabled with the interrupt enable set register can be read.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | FIELD | WBTRACE | FRMCOUNT | LCDSTOP | UNDERRUN | LCDVS |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:6 | 000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| FIELD | R | 5 | 0 | Indicates the status of FIELD indication start interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |
| WBTRACE | R | 4 | 0 | Indicates the status of the VGA standby shift end interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |
| FRMCOUNT | R | 3 | 0 | Indicates the status of the frame count interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |
| LCDSTOP | R | 2 | 0 | Indicates the status of the display stop interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |
| UNDERRUN | R | 1 | 0 | Indicates the status of the under-run interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |
| LCDVS | R | 0 | 0 | Indicates the status of the LCD frame interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |

**Remark** 0: No interrupt source, 1: Interrupt source occurred

### (2) Interrupt raw status register

This register (LCD_INTRAWSTATUS: E004_0064H) is a read-only register that indicates the statuses of interrupt sources. An interrupt source is set to the register regardless of the settings of the interrupt enable set register and the interrupt enable clear register.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | FIELD RAW | WBTRACE RAW | FRMCOUNT RAW | LCDSTOP RAW | UNDERRUN RAW | LCDVS RAW |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:6 | 000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| FIELDRAW | R | 5 | 0 | Indicates the status of FIELD indication start interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |
| WBTRACERAW | R | 4 | 0 | Indicates the status of the VGA standby shift end interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |
| FRMCOUNTRAW | R | 3 | 0 | Indicates the status of the frame count interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |
| LCDSTOPRAW | R | 2 | 0 | Indicates the status of a display stop interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |
| UNDERRUNRAW | R | 1 | 0 | Indicates the status of an under-run interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |
| LCDVSRAW | R | 0 | 0 | Indicates the status of an LCD frame interrupt.<br>0 : Without interrupt factors<br>1 : With interrupt factors |

**Remark** 0: No interrupt source, 1: Interrupt source occurred

### (3) Interrupt enable set register

This register (LCD_INTENSET: E004_0068H) enables issuance of interrupt requests. Only data of bits to which 1 is written is updated. When the bit corresponding to an interrupt source is set to 1 in this register, the interrupt source is set, the relevant interrupt request is issued and the corresponding bit of the interrupt status register is set to 1. If no bits are set in this register, no interrupt requests are issued even if the interrupt source is set, but the corresponding bit of the interrupt raw status register is set to 1.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | FIELD EN | WBTRACE EN | FRMCOUNT EN | LCDSTOP EN | UNDERRUN EN | LCDVS EN |

(1/2)

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:6 | 000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| FIELDEN | R | 5 | 0 | Indicates whether issuance of FIELD indication start interrupt requests is enabled.<br>0: Not enabled<br>1: Enabled |
| | W | 5 | – | Enables issuance of FIELD indication start interrupt requests.<br>1: Cancels interrupt masking. |
| WBTRACEEN | R | 4 | 0 | Indicates whether issuance of VGA standby shift end interrupt requests is enabled.<br>0: Not enabled<br>1: Enabled |
| | W | 4 | – | Enables issuance of VGA standby shift end interrupt requests.<br>1: Cancels interrupt masking. |
| FRMCOUNTEN | R | 3 | 0 | Indicates whether issuance of frame count interrupt requests is enabled.<br>0: Not enabled<br>1: Enabled |
| | W | 3 | – | Enables issuance of frame count interrupt requests.<br>1: Cancels interrupt masking. |

(2/2)

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| LCDSTOPEN | R | 2 | 0 | Indicates whether issuance of display stop interrupt requests is enabled.<br>0: Not enabled<br>1: Enabled |
| | W | 2 | – | Enables issuance of display stop interrupt requests.<br>1: Cancels interrupt masking. |
| UNDERRUN | R | 1 | 0 | Indicates whether issuance of under-run interrupt requests is enabled.<br>0: Not enabled<br>1: Enabled |
| | W | 1 | – | Enables issuance of under-run interrupt requests.<br>1: Cancels interrupt masking. |
| LCDVSEN | R | 0 | 0 | Indicates whether issuance of LCD frame interrupt requests is enabled.<br>0: Not enabled<br>1: Enabled |
| | W | 0 | – | Enables issuance of LCD frame interrupt requests.<br>1: Cancels interrupt masking. |

### (4) Interrupt enable clear register

This register (LCD_INTENCLR: E004_006CH) is a write-only register that masks issuance of interrupt requests. Only data of bits to which 1 is written is updated. When the bit corresponding to an interrupt source in this register is set to 1, no interrupt requests are issued even if the interrupt source is generated. The status of the corresponding bit in the interrupt status register also remains unchanged. If no bits are set in this register, an interrupt request is issued and the corresponding bit of the interrupt status register is set to 1 when the interrupt source is set.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | FIELD MASK | WBTRACE MASK | FRMCOUNT MASK | LCDSTOP MASK | UNDERRUN MASK | LCDVS MASK |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | – | 31:6 | 000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| FIELDMASK | W | 5 | 0 | Disables issuance of FIELD indication start interrupt requests.<br>1: Masks interrupts. |
| WBTRACEMASK | W | 4 | 0 | Disables issuance of VGA standby shift end interrupt requests.<br>1: Masks interrupts. |
| FRMCOUNTMASK | W | 3 | 0 | Disables issuance of frame count interrupt requests.<br>1: Masks interrupts |
| LCDSTOPMASK | W | 2 | 0 | Disables issuance of display stop interrupt requests.<br>1: Masks interrupts |
| UNDERRUNMASK | W | 1 | 0 | Disables issuance of under-run interrupt requests.<br>1: Masks interrupts |
| LCDVSMASK | W | 0 | 0 | Disables issuance of LCD frame interrupt requests.<br>1: Masks interrupts |

### (5) Interrupt source clear register

This register (LCD_INTFFCLR: E004_0070H) is a write-only register that requests clearing of an interrupt source. Only data of bits to which 1 is written is updated. Setting the bit corresponding to an interrupt source to 1 clears the interrupt source.

If setting and clearing of an interrupt source are performed at the same time, setting takes precedence.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

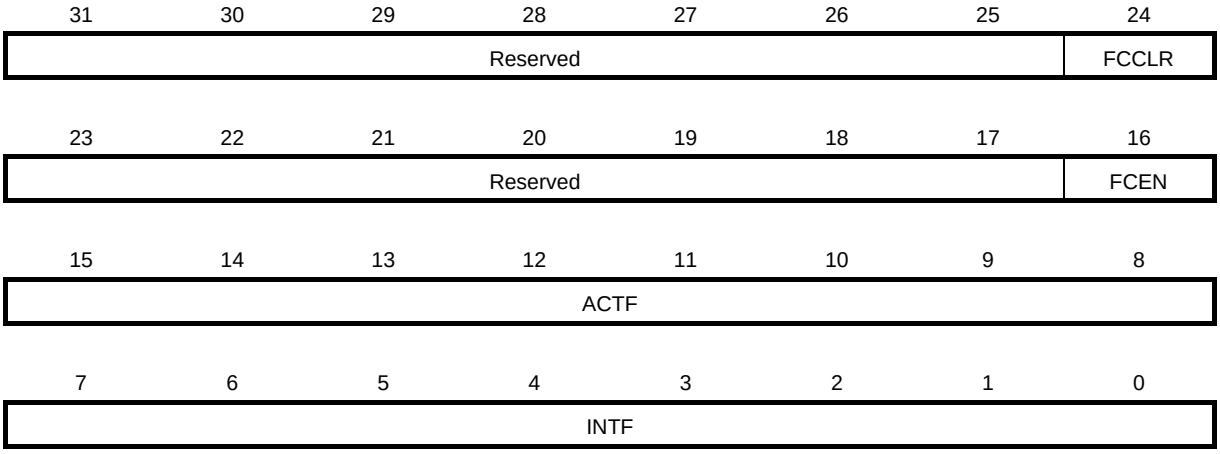
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | FIELD CLR | WBTRACE CLR | FRMCOUNT CLR | LCDSTOP CLR | UNDERRUN CLR | LCDVS CLR |

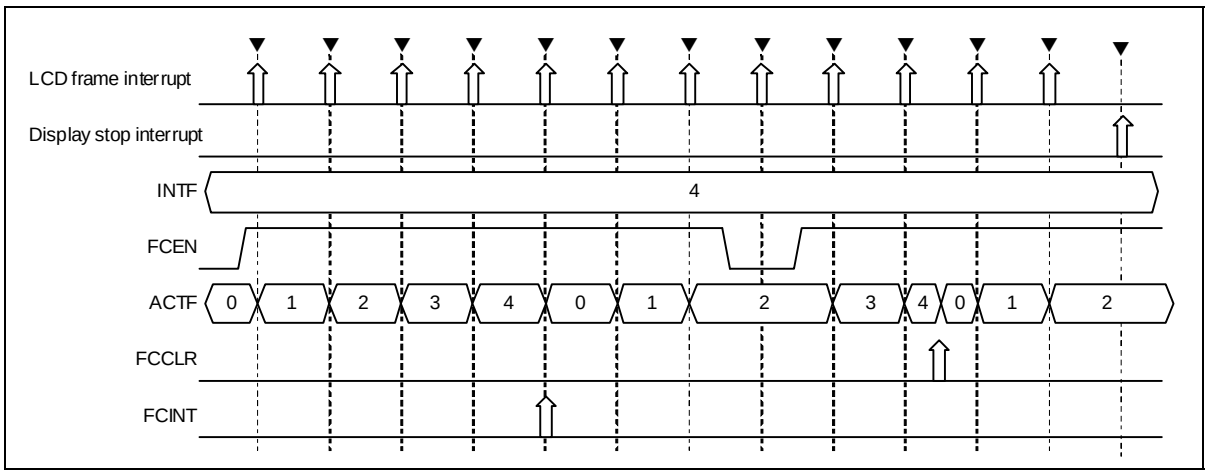
| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | – | 31:6 | 000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| FIELDCLR | W | 5 | 0 | Requests clearing of a FIELD indication start interrupt source.<br>1: Clearing an interrupt source. |
| WBTRACECLR | W | 4 | 0 | Requests clearing of a VGA standby shift end interrupt source.<br>1: Clearing an interrupt source. |
| FRMCOUNTCLR | W | 3 | 0 | Requests clearing of a frame count interrupt source.<br>1: Clearing an interrupt source. |
| LCDSTOPCLR | W | 2 | 0 | Requests clearing of a display stop interrupt source.<br>1: Clearing an interrupt source. |
| UNDERRUNCLR | W | 1 | 0 | Requests clearing of an under-run interrupt source.<br>1: Clearing an interrupt source. |
| LCDVSCLR | W | 0 | 0 | Requests clearing of an LCD frame interrupt source.<br>1: Clearing an interrupt source. |

### (6) Frame count interrupt setting register

This register (LCD_FRAMECOUNT: E004_0074H) is used to perform various settings related to the frame count interrupt.




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | – | 31:25 | 00H | Reserved. When these bits are read, 0 is returned for each bit. |
| FCCLR | W | 24 | 0 | Clears the frame counter value.<br>0: Does not affect the setting. When this bit is read, 0 is returned.<br>1: Initializes the ACTF bit setting. |
| Reserved | – | 23:17 | 00H | Reserved. When these bits are read, 0 is returned for each bit. |
| FCEN | R/W | 16 | 0 | Sets the operation of the frame count function.<br>0: Stops<br>1: Starts operation |
| ACTF | R | 15:8 | 00H | Indicates the number of current frames being counted. |
| INTF | R/W | 7:0 | 00H | Sets the timer count threshold by which a frame count interrupt is issued. |




When an LCD frame interrupt is issued while the FCEN bit is set to 1, the ACTF bit value is incremented. It is not incremented when a display stop interrupt is issued. If an LCD frame interrupt is issued when the values of the ACTF and INTF bits are the same, a frame count interrupt is issued and the ACTF bit is cleared to 0. The ACTF bit is also cleared to 0 when the FCCLR bit is set to 1.

### 3.2.21 RGB2YUV transform coefficient register

This register (LCD_COEF) sets the transform coefficient when changing from YUV to RGB, is established. For details, see Format transform.
Refer to the following for each address.

| | | |
|---|---|---|
| LCD_COEF_Y0 : E004_0080H | LCD_COEF_U0 : E004_0090H | LCD_COEF_V0 : E004_00A0H |
| LCD_COEF_Y1 : E004_0084H | LCD_COEF_U1 : E004_0094H | LCD_COEF_V1 : E004_00A4H |
| LCD_COEF_Y2 : E004_0088H | LCD_COEF_U2 : E004_0098H | LCD_COEF_V2 : E004_00A8H |
| LCD_COEF_Y3 : E004_008CH | LCD_COEF_U3 : E004_009CH | LCD_COEF_V3 : E004_00ACH |

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

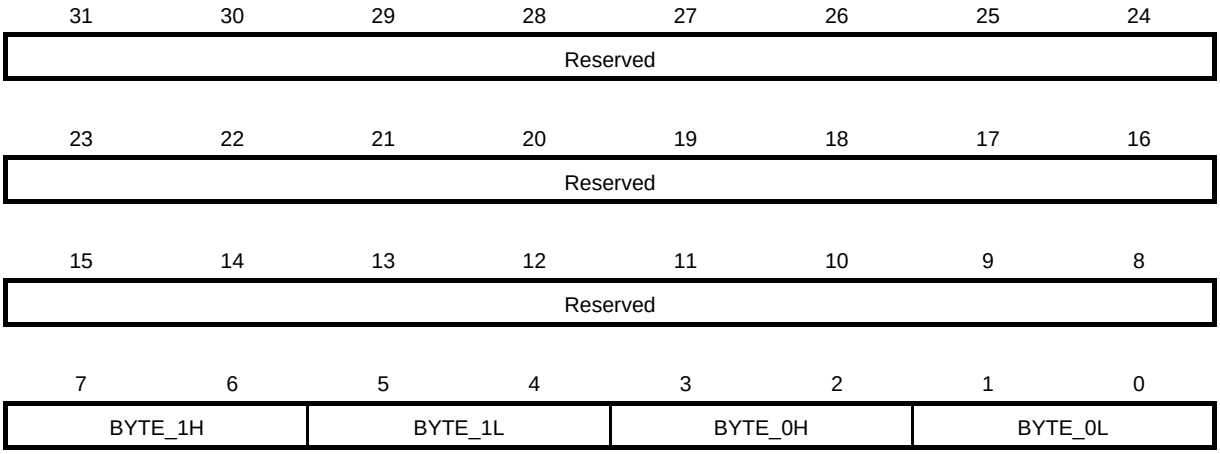
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | COEF | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| COEF | | | | | | | |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:11 | 00_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| COEF | R/W | 10:0 | 000H | The (complement of 2) 11 bit value with a code (-1024-1023) |

### 3.2.22 BYTELANE register

This register (LCD_BYTELANE: E004_00C0H) can choose an order of the output data, when doing YUV output..




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:8 | 00_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| BYTE_1H | R/W | 7:6 | 10b | The data which is output in among the 2Byte and the upper byte output later is chosen.<br>00 : V<br>01 : U<br>10 : Y1<br>11 : Y0 |
| BYTE_1L | R/W | 5:4 | 00b | The data which is output in among the 2Byte and the upper byte output later is chosen.<br>00 : V<br>01 : U<br>10 : Y1<br>11 : Y0 |
| BYTE_0H | R/W | 3:2 | 11b | The data which is output in among the 2Byte and the upper byte output later is chosen.<br>00 : V<br>01 : U<br>10 : Y1<br>11 : Y0 |
| BYTE_0L | R/W | 1:0 | 01b | The data which is output in among the 2Byte and the upper byte output later is chosen.<br>00 : V<br>01 : U<br>10 : Y1<br>11 : Y0 |

Refer to **4.2.1 Image data** about use method of the Bytelane function.

### 3.2.23 Vertical synchronization signal control register

This register (LCD_VSYNC_CONT : E004_00D0H) establishes operation of a vertical sync signal of InterlaceMode.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

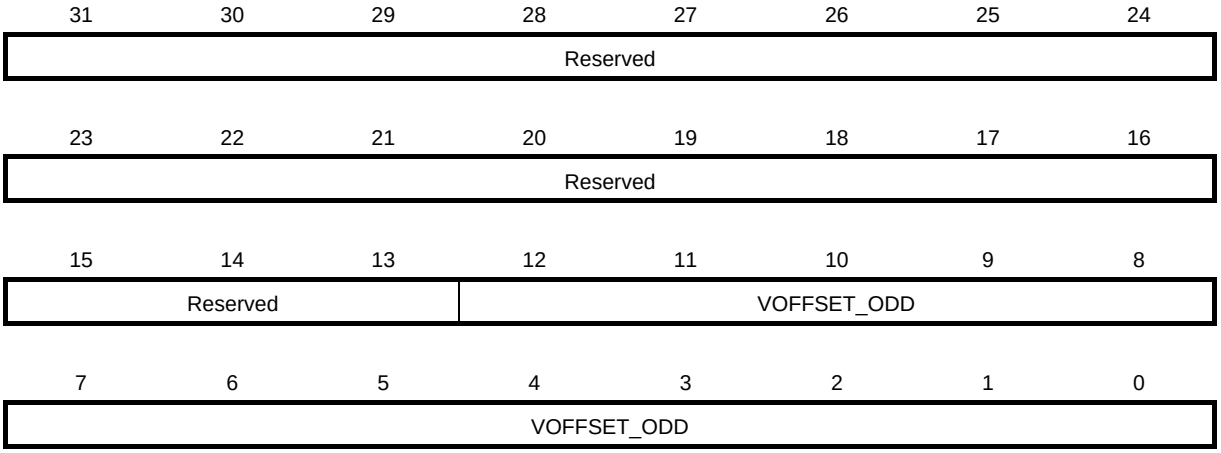
| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | VEDGE_SEL | VTOTAL_SEL | VOFFSET_EN |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:3 | 0000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| VEDGE_SEL | R/W | 2 | 0 | A vertical synchronization edge register of ODDFIELD is chosen in Interlace output Mode.<br>0 : LCD_VEDGE1 / LCD_VEDGE2<br>1 : LCD_VEDGE1_EVEN / LCD_VEDGE2_EVEN |
| VTOTAL_SEL | R/W | 1 | 0 | A vertical direction total register of ODDFIELD is chosen in Interlace output Mode.<br>0 : LCD_VTOTAL<br>1 : LCD_VTOTAL_EVEN |
| VOFFSET_EN | R/W | 1 | 0 | The horizontal direction OFFSET function of the vertical sync signal is established.<br>0 : invalid<br>1 : valid |

### 3.2.24 Vertical synchronization edge offset register (Odd field)

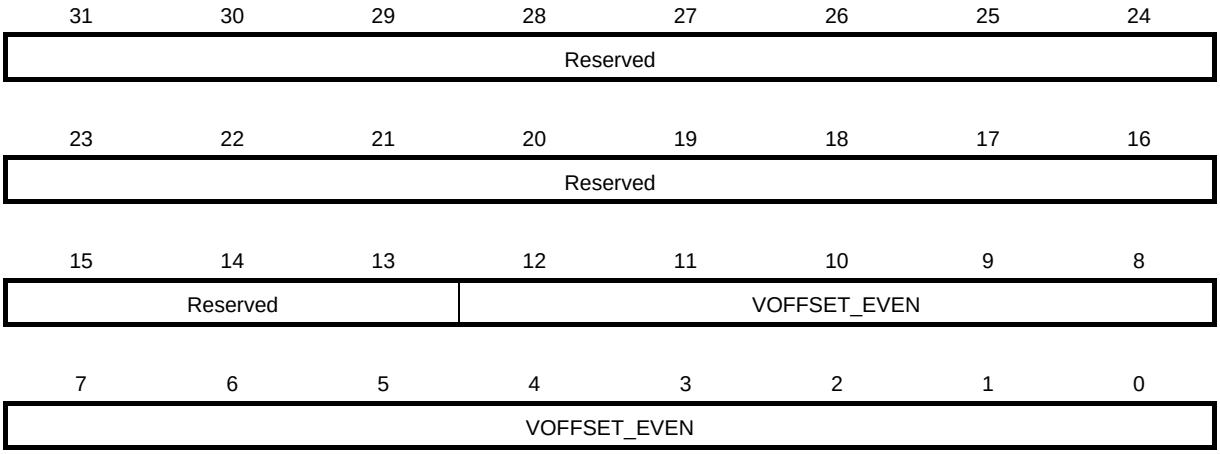
This register (LCD_VOFFSET_ODD : E004_00D8H) sets the horizontal direction location of the vertical sync signal by the X-coordinate (the number of pixel clocks).
This set value is applied in case of the Interlace output odd field and Progressive output when being VOFFSET_EN="1".




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 00_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| VOFFSET_ODD | R/W | 12:0 | 0000H | The horizontal direction location of the vertical sync signal is set. (Odd Field)<br>For details, see **4.1.8 Interlace output mode** |

### 3.2.25 Vertical synchronization edge offset register (Even field)

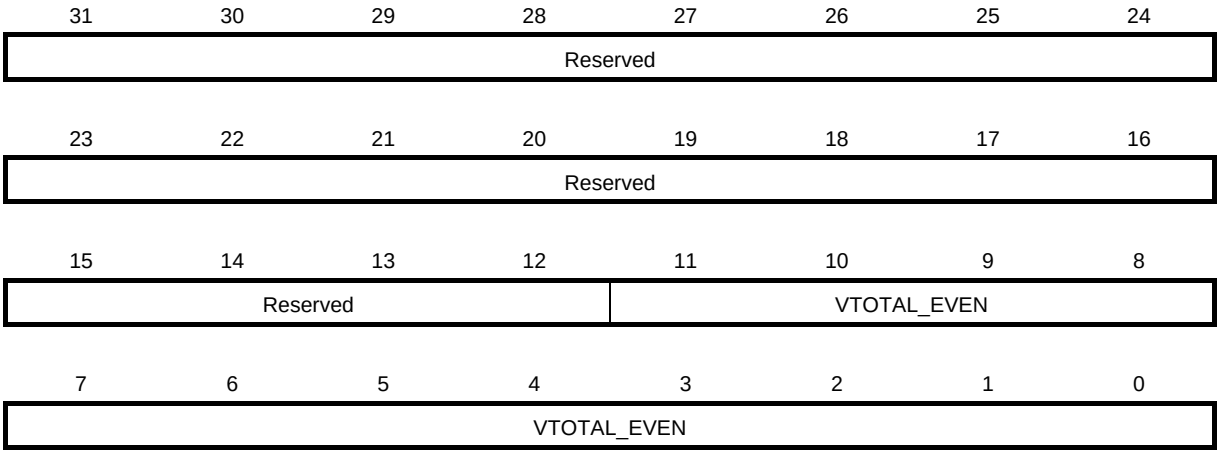
This register (LCD_VOFFSET_EVEN : E004_00DCH) sets the horizontal direction location of the vertical sync signal by the X-coordinate (the number of pixel clocks).
This set value is applied in case of Interlace output even field when being VOFFSET_EN="1".




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 00_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| VOFFSET_EVEN | R/W | 12:0 | 0000H | The horizontal direction location of the vertical sync signal is set. (Even Field)<br>For details, see **4.1.8 Interlace output mode** |

### 3.2.26 Vertical direction total register (Even field)

This register (LCD_VAREA_EVEN : E004_00E0H) establishes general number of lines in a vertical direction (the cycle of the VSYNC= number of the HSYNC).
This set value is applied in case of Interlace output even field when being VTOTAL_SEL="1".

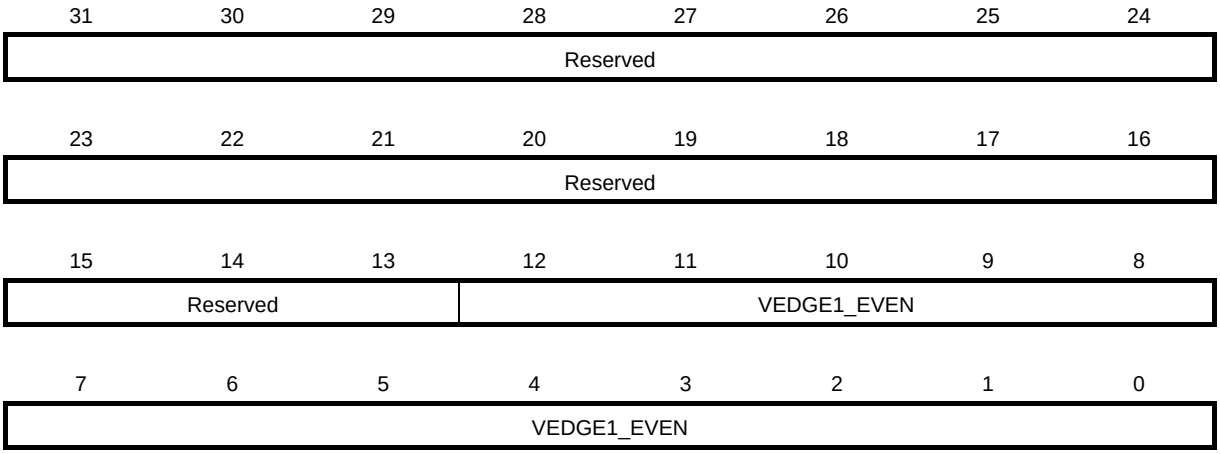



| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:12 | 00_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| VTOTAL_EVEN | R/W | 11:0 | 000H | EVEN FIELD Indication number of lines in a vertical direction is established.<br>For details, see **4.1.4 Display area, and horizontal and vertical blanks** |

### 3.2.27 Vertical synchronization edge1 register (Even field)

This register (LCD_VEDGE1_EVEN : E004_00E8H) sets the 1st edge location of the vertical sync signal by the Y-coordinate (the number of HSYNC).

This set value is applied in case of Interlace output even field when being VEDGE_SEL="1".

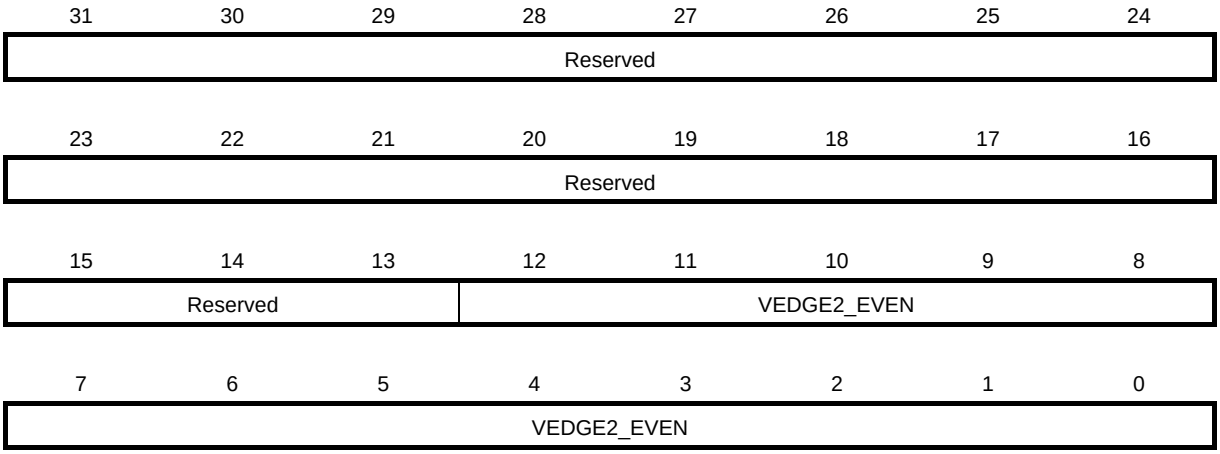



| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 00_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| VEDGE1_EVEN | R/W | 12:0 | 0000H | EVEN FIELD The 1st edge location of the vertical sync signal is set by the Y-coordinate.<br>For details, see **4.1.6 Vertical synchronization signal** |

### 3.2.28 Vertical synchronization edge2 register (Even field)

This register (LCD_VEDGE2_EVEN : E004_00ECH) sets the 2nd edge location of the vertical sync signal by the Y-coordinate (the number of HSYNC).

This set value is applied in case of Interlace output even field when being VEDGE_SEL="1".




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:13 | 00_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| VEDGE2_EVEN | R/W | 12:0 | 0000H | EVEN FIELD The 2nd edge location of the vertical sync signal is set by the Y-coordinate.<br>For details, see **4.1.6 Vertical synchronization signal** |

### 3.2.29 Cursor enable register

This register (LCD_CURSOR_EN: E004_0200H) sets effectively or invalidly of the H/W cursor function. This is a frame-synchronous register and its set values take effect upon reception of a frame start signal.

| 31 | 30 | 29 | 28 | 27 | 26 | 25 | 24 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 23 | 22 | 21 | 20 | 19 | 18 | 17 | 16 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

| 15 | 14 | 13 | 12 | 11 | 10 | 9 | 8 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | |

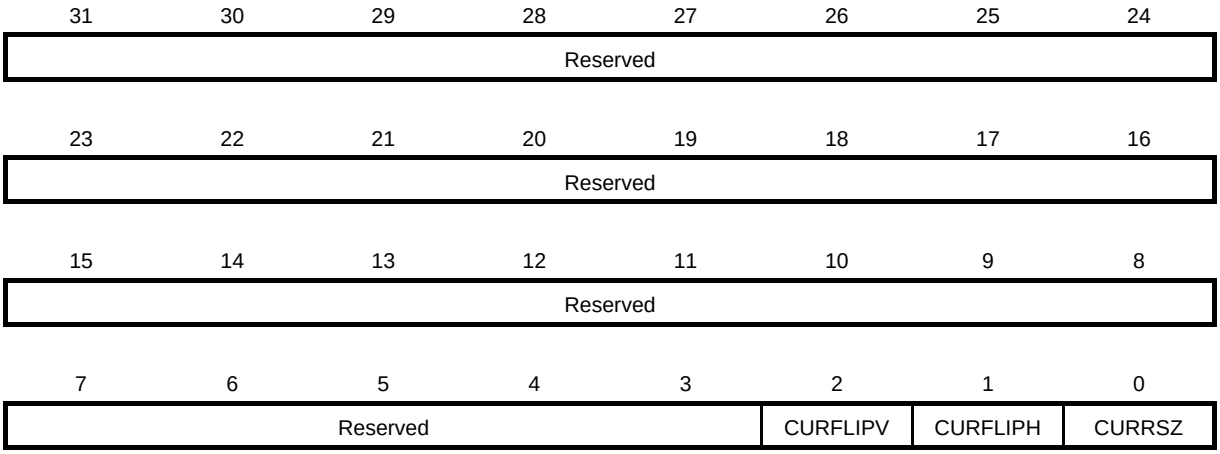
| 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|
| Reserved | | | | | | | CUREN |

| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:1 | 0000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| CUREN | R/W | 0 | 0 | Cursor function effectively or invalidly<br>0 : invalidity<br>1 : effectively |

Refer to Use of the cursor function about use method of the Cursor function.

### 3.2.30 Cursor operation register

This register (LCD_CURSOR_OPE: E004_0204H) sets option of the H/W cursor function.
This register can set effective or invalidly about cursor inverting function and simple double resizing function.
This is a frame-synchronous register and its set values take effect upon reception of a frame start signal.

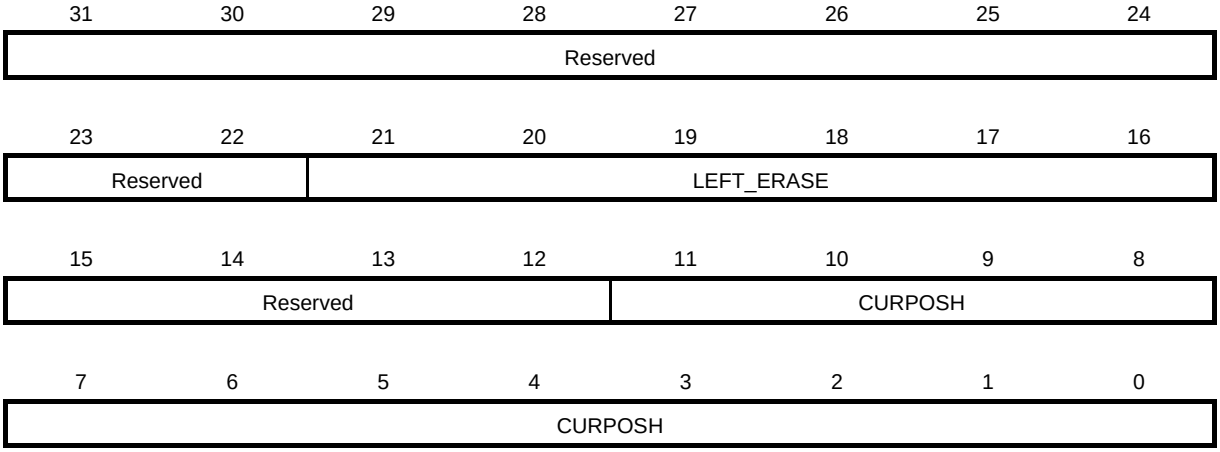



| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:3 | 0000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| CURFLIPV | R/W | 2 | 0 | Set the vertical direction of cursor inverting function<br>0 : not invert<br>1 : invert |
| CURFLIPH | R/W | 1 | 0 | Set the horizontal direction of cursor inverting function<br>0 : not invert<br>1 : invert |
| CURRSZ | R/W | 0 | 0 | Set the cursor simple resizing<br>0 : Simple resizing not use (64 × 64 pixels cursor)<br>1 : Simple resizing use (128 × 128 pixels cursor) |

Refer to Use of the cursor function about use method of the Cursor function.

### 3.2.31 Cursor horizontal position register

This register (LCD_CURSOR_POSH: E004_0208H) sets horizontal starting point position of the H/W cursor. The level starting point position establishes the left side erasure amount of the cursor picture in case of 0. This is a frame-synchronous register and its set values take effect upon reception of a frame start signal.

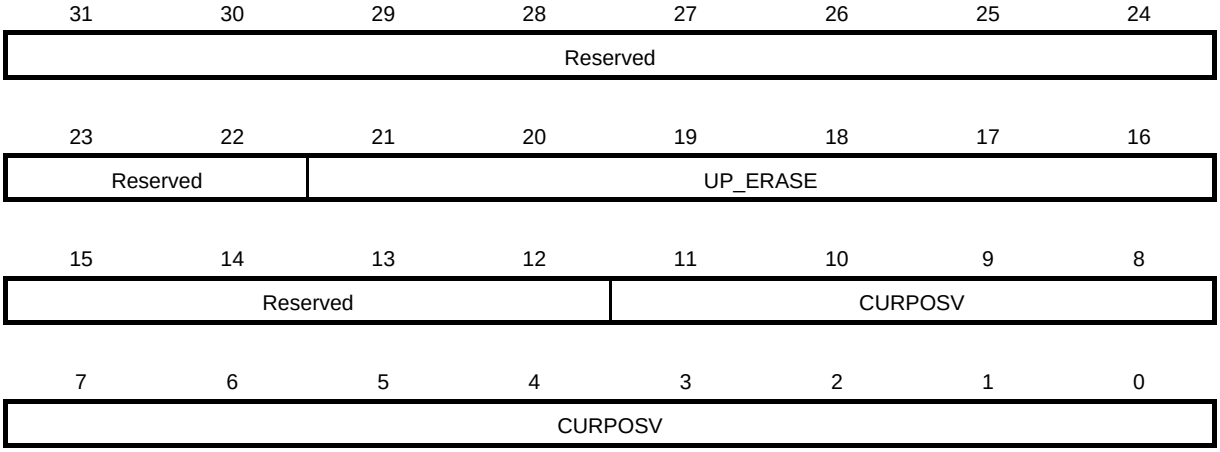



| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:22 | 000H | Reserved. When these bits are read, 0 is returned for each bit. |
| LEFT_ERASE | R/W | 21:16 | 00H | Set the left side erasure amount of the cursor picture.<br>(It's effective only in case of CURPOSH=0.) |
| Reserved | R | 15:12 | 0H | Reserved. When these bits are read, 0 is returned for each bit. |
| CURPOSH | R/W | 11:0 | 000H | Set the cursor horizontal starting point position |

Refer to Use of the cursor function about use method of the Cursor function.

### 3.2.32 Cursor vertical position register

This register (LCD_CURSOR_POSV: E004_020CH) sets vertical starting point position of the H/W cursor. The level starting point position establishes the up side erasure amount of the cursor picture in case of 0. This is a frame-synchronous register and its set values take effect upon reception of a frame start signal.

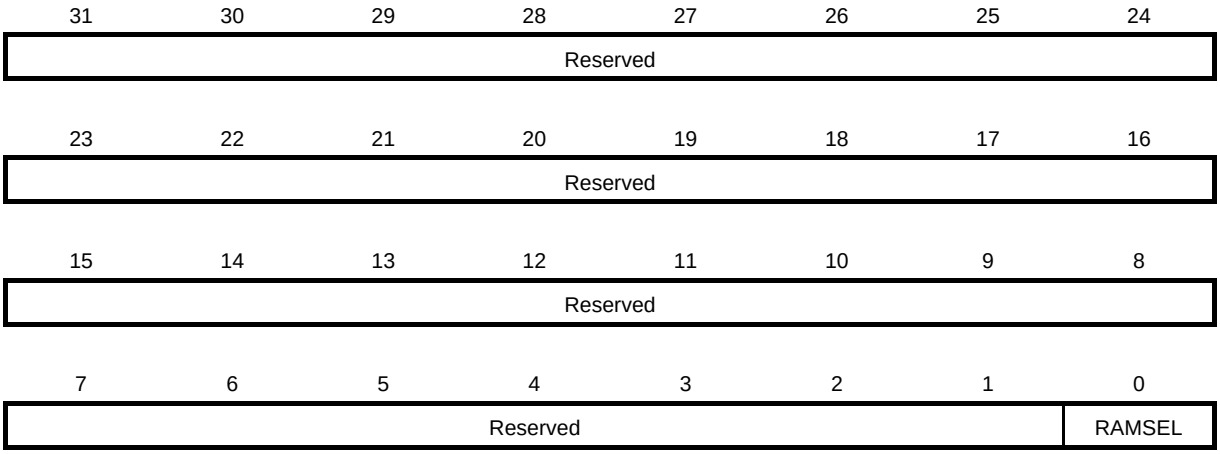



| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:22 | 000H | Reserved. When these bits are read, 0 is returned for each bit. |
| UP_ERASE | R/W | 21:16 | 00H | Set the up side erasure amount of the cursor picture.<br>(It's effective only in case of CURPOSV=0.) |
| Reserved | R | 15:12 | 0H | Reserved. When these bits are read, 0 is returned for each bit. |
| CURPOSV | R/W | 11:0 | 000H | Set the cursor vertical starting point position |

Refer to Use of the cursor function about use method of the Cursor function.

### 3.2.33 Cursor data RAM select register

This register (LCD_CURSOR_RAMSEL: E004_0210H) selects the cursor data RAM used for LCD indication. This is a frame-synchronous register and its set values take effect upon reception of a frame start signal.

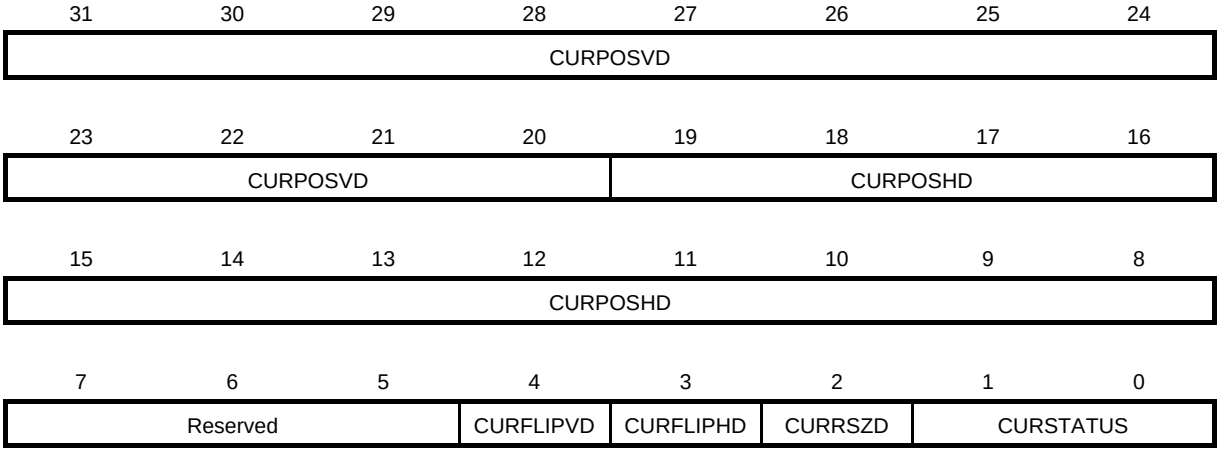



| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:1 | 0000_0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| RAMSEL | R/W | 0 | 0 | Select the cursor data RAM used for LCD indication<br>0 : A side (cursor data RAM 0)<br>1 : B side (cursor data RAM 1) |

Refer to Use of the cursor function about use method of the Cursor function.

### 3.2.34 Cursor status register

This register (LCD_CURSOR_STATUS: E004_0214H) can check the operating state of a cursor. Status of a cursor can be acquired by polling this register. The state of the cursor of the frame is reflected by frame interrupt genesis timing.




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| CURPOSVD | R | 31:20 | 000H | Indicates the cursor vertical starting point position. |
| CURPOSHD | R | 19:8 | 000H | Indicates the cursor horizontal starting point position. |
| Reserved | R | 7:5 | 0H | Reserved. When these bits are read, 0 is returned for each bit. |
| CURFLIPVD | R | 4 | 0 | Indicates the status of cursor vertical invert.<br>0 : Vertical invert invalidity<br>1: Vertical invert effective |
| CURFLIPHD | R | 3 | 0 | Indicates the status of cursor horizontal invert.<br>0 : Horizontal invert invalidity<br>1: Horizontal invert effective |
| CURRSZD | R | 2 | 0 | Indicates the status of cursor simple resizing.<br>0 : Cursor simple resizing invalidity<br>1: Cursor simple resizing effective. |
| CURSTATUS | R | 1:0 | 00b | Indicates the display status of cursor.<br>00 : Cursor nondisplay<br>01: Reserved<br>10 : Cursor in A side (cursor data RAM 0) is being indicated.<br>11 : Cursor in B side (cursor data RAM 1) is being indicated. |

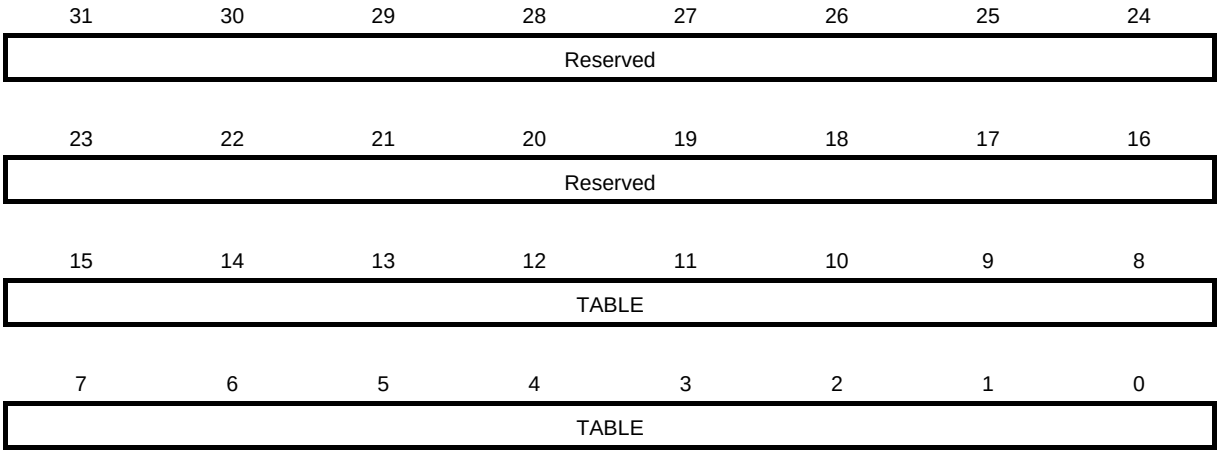
Refer to Use of the cursor function about use method of the Cursor function.

### 3.2.35 Cursor palette RAM register

This register (LCD_CURSOR_TABLE: E004_0400H – E004_07FCH) can access the palette RAM of H/W cursor. It's only when the CURSTATUS bit of the cursor status register (LCD_CURSOR_STATUS) is 0 (cursor non display) even if it's indicating whether the STATUS bit of the status register (LCD_STATUS) is 0 (indication stop condition), that they're accessible.

When doing read, data is stocked in a flip-flop inside the circuit from RAM in the 1st time of read, read cycle in APB, because then it isn't possible to read data of the flip-flop, it's continued twice, and, read, break the 1st time of read data and use the 2nd time of price as read value.

For palette number 0 to be dealt with as transparent color, the set value is ignored.




| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| Reserved | R | 31:16 | 0000H | Reserved. When these bits are read, 0 is returned for each bit. |
| TABLE | R/W | 15:0 | variable | Sets the pixel data of RGB565 format.<br>bit15-11 : Red (5bit)<br>bit10-5 : Green (6bit)<br>bit4-0 : Blue (5bit) |

Note ) With DMA, you can't write.

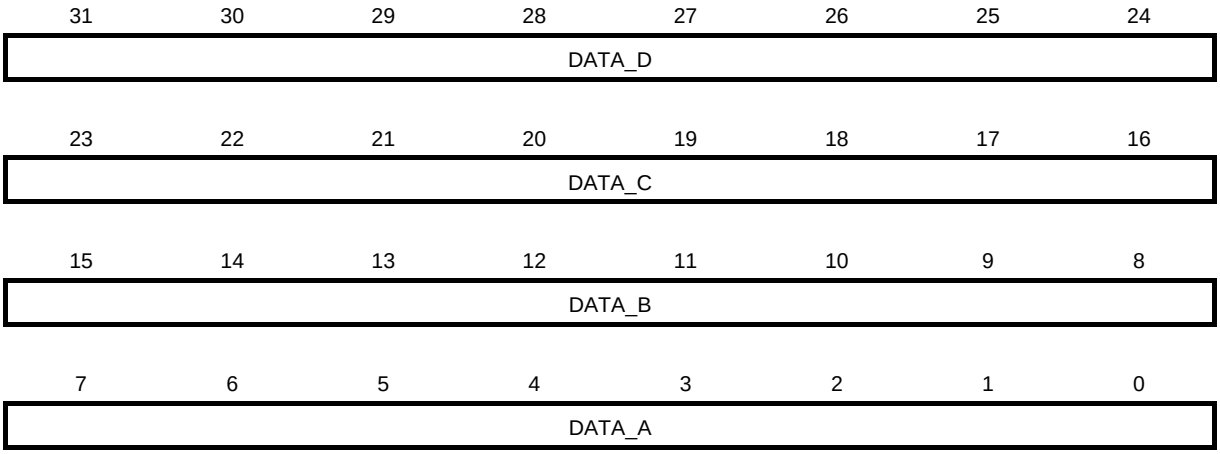
One pixel data is set every one address.

| address | Account |
|---|---|
| E004_0400H | Pixel data of palette 0 |
| E004_0404H | Pixel data of palette 1 |
| : | : |
| E004_07F8H | Pixel data of palette 254 |
| E004_07FCH | Pixel data of palette 255 |

Refer to Use of the cursor function about use method of the Cursor function.

### 3.2.36 Cursor data RAM 0/1 register

This register (LCD_CURSOR_DATA0: E004_1000H – E004_1FFFH) can access the A side (cursor data RAM 0) of memory for cursor storing data.
This register (LCD_CURSOR_DATA1: E004_2000H – E004_2FFFH) can access the B side (cursor data RAM 1) of memory for cursor storing data.
When the CURSTATUS bit of the cursor status register (LCD_CURSOR_STATUS) is 0 (cursor non display) even if it's indicating whether the STATUS bit of the status register (LCD_STATUS) is 0 (indication stop condition), they're accessible to both in A side and a B side. Only the side which isn't chosen is accessible to LCD indication during cursor indication.
When doing read, data is stocked in a flip-flop inside the circuit from RAM in the 1st time of read, read cycle in APB, because then it isn't possible to read data of the flip-flop, it's continued twice, and, read, break the 1st time of read data and use the 2nd time of price as read value.
For palette number 0 to be dealt with as transparent color, the set value is ignored.

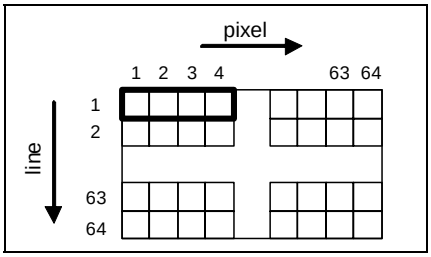



| Name | R/W | Bit No. | After Reset | Function |
|---|---|---|---|---|
| DATA_D | R/W | 31:24 | variable | The palette number which corresponds to right pixel of the horizontal 4pixel unit is established. |
| DATA_C | R/W | 23:16 | variable | The palette number which corresponds to the 2nd pixel from the right of the horizontal 4pixel unit is established. |
| DATA_B | R/W | 15:8 | variable | The palette number which corresponds to the 2nd pixel from the left of the horizontal 4pixel unit is established. |
| DATA_A | R/W | 7:0 | variable | The palette number which corresponds to left pixel of the horizontal 4pixel unit is established. |

Note ) With DMA, you can't write.

Four pixel data is set every one address.

| address | account | address | account |
|---|---|---|---|
| E004_1000H | It correspond pixel for 1 to 4 of line 1. | E004_1FC0H | It correspond pixel for 1 to 4 of line 64. |
| E004_1004H | It correspond pixel for 5 to 8 of line 1. | E004_1FC4H | It correspond pixel for 5 to 8 of line 64. |
| : | | : | |
| E004_1038H | It correspond pixel for 57 to 60 of line 1. | E004_1FF8H | It correspond pixel for 57 to 60 of line 64. |
| E004_103CH | It correspond pixel for 61 to 64 of line 1. | E004_1FFCH | It correspond pixel for 61 to 64 of line 64. |




Refer to Use of the cursor function about use method of the Cursor function.

# 4. Description of Functions

## 4.1 LCD panel Interface

### 4.1.1 Image data

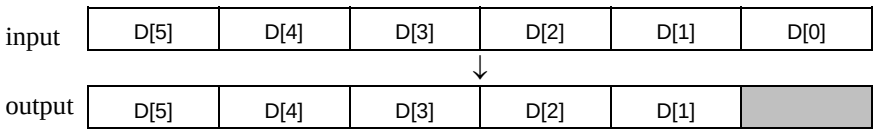
The LCD controller supports an LCD panel in 16 bpp mode (65,536 colors), 18 bpp mode (260,000 colors) and 24 bpp (16,770,000 colors). RGB666, RGB565 and RGB888 are switched by setting the OFORMAT bit of the control register (LCD_CONTROL).
When RGB666/RGB565 was chosen by the output format (OFORMAT), data of 565/666 arranges 8 bits for each of RGB of LCD data on the signal MSB side.

### 4.1.2 Format conversion

In the LCD controller, the format of data input (from the IMC) and output to an LCD panel can be set individually. Since the output data format is mainly determined in accordance with the LCD panel connected, setting of the output data format is assigned to the control register (LCD_CONTROL) (an immediately-reflected register, which is defined in the chapter of the IMC). Setting of the input data format is assigned to individual registers so as to enable switching in frame units.
If a different format is specified to input and output, the components of R and B are converted according to the following rules.
RGB888 and in case of YUV output correspond to only RGB888 input.

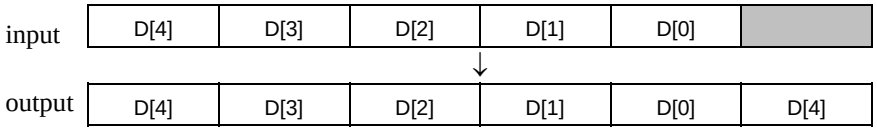
**○ When the input format is RGB666 and output format is RGB565**

The LSB of R and B is discarded and scaled to be 5 bits.



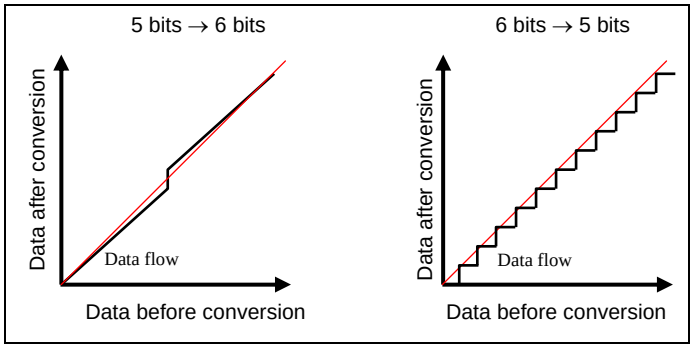

**○ When the input format is RGB565 and output format is RGB666**

The MSB of R and B is added and scaled to be 6 bits.




The next figure shows format conversion processing. It just shows a conceptual diagram because the gradation varies between input and output.

**Figure 4-1. Format Conversion Operation**



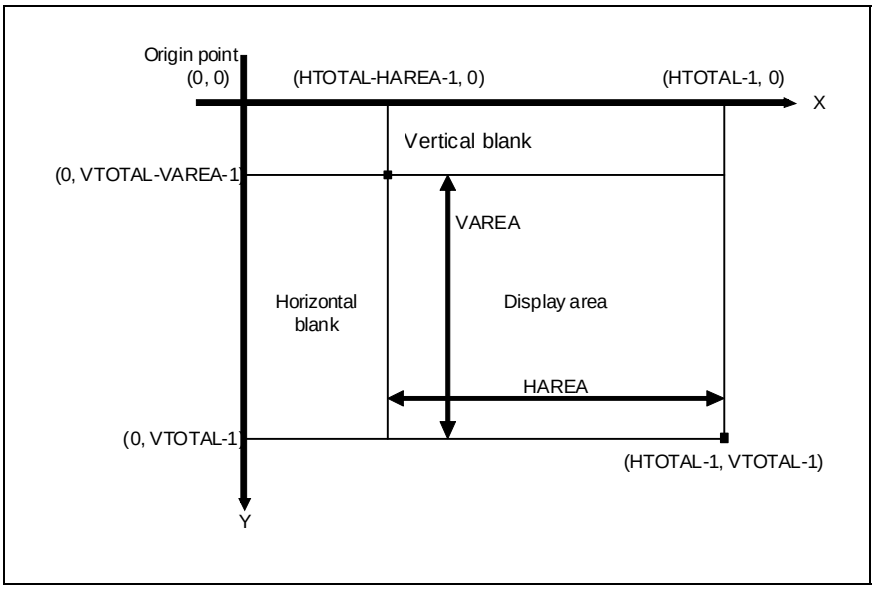

### 4.1.3 LCD clock

The phases of the output clock (PXCLK) and LCD panel interface signal lines (VSYNC, HSYNC, DATAENABLE or R/G/BDATA) can be selected.

### 4.1.4 Display area, and horizontal and vertical blanks

The LCD display areas and horizontal/vertical blanks are defined by using the lower right coordinate system, in pixel clock units. The value of the X coordinate increases as it moves to the right, and the value of the Y coordinate increases as it moves down. The origin point is the higher left (0, 0).
To set the square size in an LCD panel, use the HAREA and VAREA bits of the horizontal/vertical direction display area registers. The HTOTAL and VTOTAL bits of the horizontal/vertical direction total registers define the lower right corner of the square, including horizontal and vertical blanks (non-display area).

**Figure 4-2. Display Area and Horizontal/Vertical Blanks**




Define the parameters by setting the following bits of the relevant LCD controller registers.

**Table 4-1. Parameters Related to Display Size**

| Register | Setting Bits |
|---|---|
| Horizontal direction total register | HTOTAL[12:0] |
| Horizontal direction display area register | HAREA[11:0] |
| Vertical direction total register | VTOTAL[12:0] |
| Vertical direction display area register | VAREA[11:0] |

**Caution Set the parameters so as to satisfy the following expressions.**
**1. HTOTAL > HAREA+4**
**(In the case when the cursor function is used, HTOTAL > HAREA + 6**
**2. VTOTAL > VAREA**

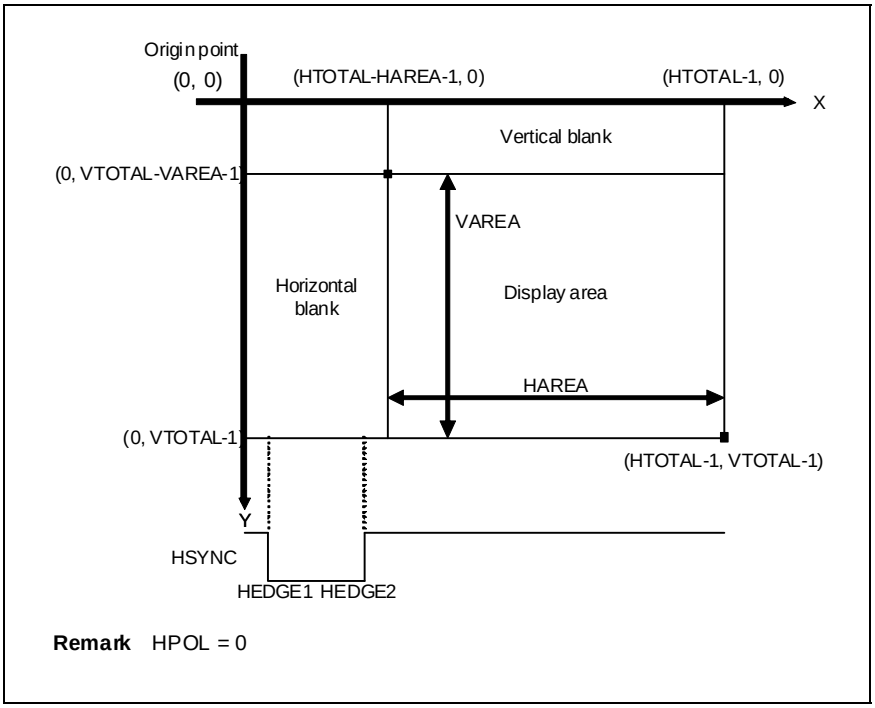
**Data is read from a frame buffer, starting from the origin point (0, 0) as the frame display start position. Pixel data is output to an LCD panel starting from the position of (HTOTAL – HAREA – 1, VTOTAL – VAREA – 1).**
**A buffer under-run is likely to occur if the period between these two points is too short. Keep the vertical blanking interval as long as possible.**

### 4.1.5 Horizontal synchronization signal

The horizontal synchronization signal is defined by using the lower right coordinate system, in LCD clock units. A pulse is generated by each line in the square specified with the origin point (0, 0) and (HTOTAL – 1, VTOTAL – 1).

The first edge of a horizontal synchronization signal is set by the HEDGE1 bit of the horizontal synchronization edge 1 register, and the second edge is set by the HEDGE2 bit of the horizontal synchronization edge 2 register. To control the polarity of a horizontal synchronization signal, use the HPOL bit of the control register (LCD_CONTROL) (0: a signal level changes from high to low at the first edge, and changes from low to high at the second edge, 1: opposite setting to 0).

**Figure 4-3. Horizontal Synchronizing Signal**




Define the parameters by setting the following bits of the relevant LCD controller registers.

**Table 4-2. Parameters Related to Horizontal Synchronization**

| Register | Setting Bits |
|---|---|
| Horizontal synchronization edge 1 register | HEDGE1[12:0] |
| Horizontal synchronization edge 2 register | HEDGE2[12:0] |

**Caution Set the parameters so as to satisfy the following expression.**

**HTOTAL > HEDGE2 > HEDGE1 ≥ 0**

### 4.1.6 Vertical synchronization signal

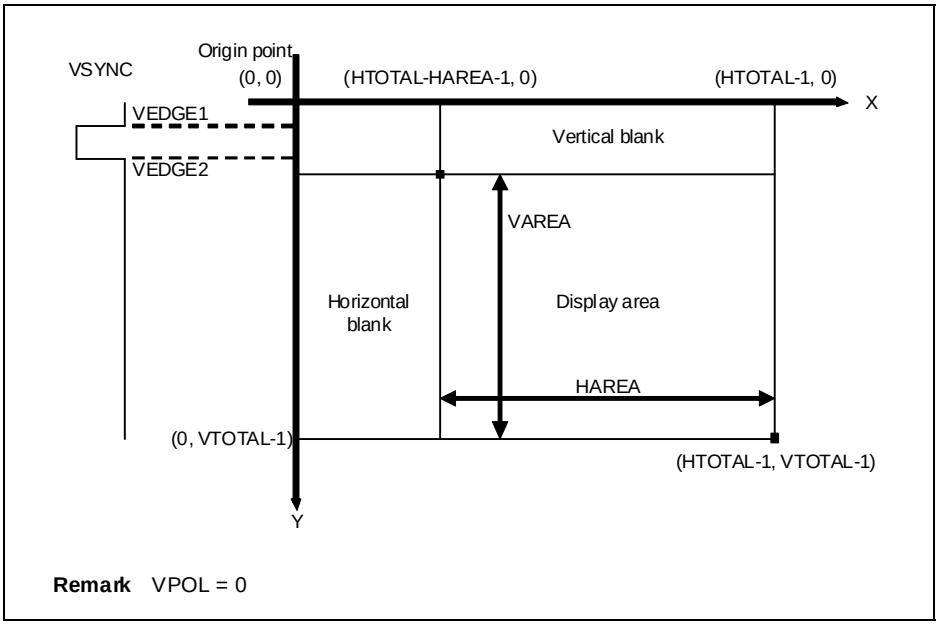
The vertical synchronization signal is defined by using the lower right coordinate system, in LCD clock units. A pulse is generated in the square.

The first edge of a vertical synchronization signal is set by the VEDGE1 bit of the vertical synchronization edge 1 register, and the second edge is set by the VEDGE2 bit of the vertical synchronization edge 2 register.

To control the polarity of a vertical synchronization signal, use the VPOL bit of the control register (LCD_CONTROL) (0: a signal level changes from high to low at the first edge, and changes from low to high at the second edge, 1: reverse of setting to 1).

The VSYNC level changes at the horizontal coordinate 0 in the following figure, regardless of the EDGE setting for HSYNC. When HEDGE is set to a value other than 0, therefore, note that VSYNC and HSYNC do not change at the same time.

**Figure 4-4. Vertical Synchronization Signal**




Define the parameters by setting the following bits of the relevant LCD controller registers.

**Table 4-3. Parameters Related to Vertical Synchronization**

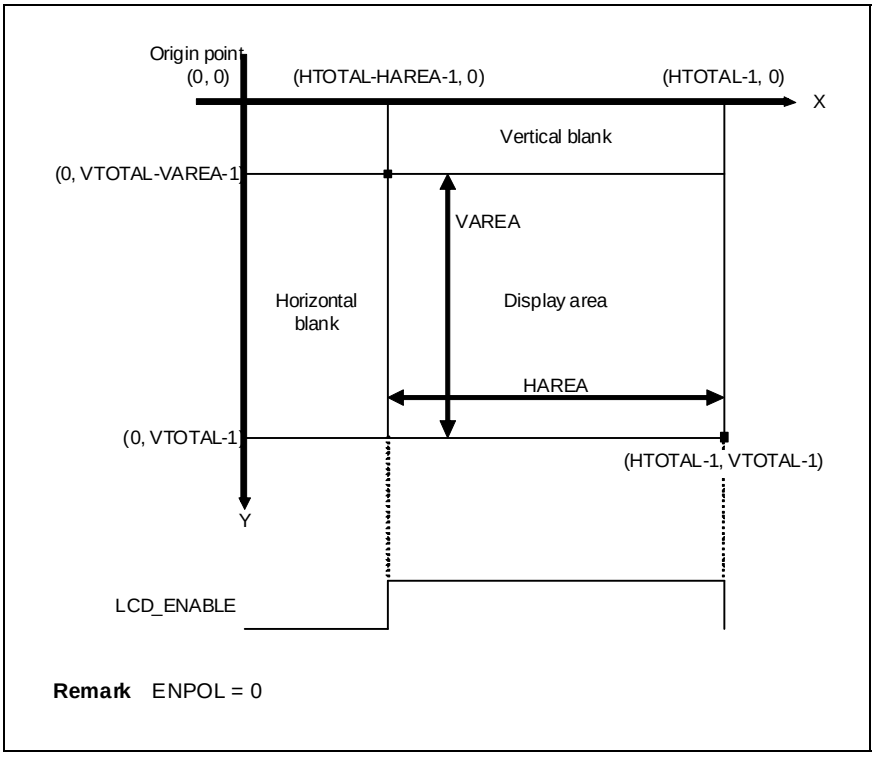
| Register | Setting Bits |
|---|---|
| Vertical synchronization edge 1 register | VEDGE1[12:0] |
| Vertical synchronization edge 2 register | VEDGE2[12:0] |

**Caution Set the parameters so as to satisfy the following expression.**
**VTOTAL > VEDGE2 > VEDGE1 ≥ 0**

### 4.1.7 Enable signal

An enable signal is set by the HAREA and VAREA bits of the horizontal/vertical direction display area registers, and goes into the active level in the display area. The active level can be controlled by the ENPOL bit of the control register (LCD_CONTROL) (0: active level of enable signal = high, 1: active level of enable signal = low).

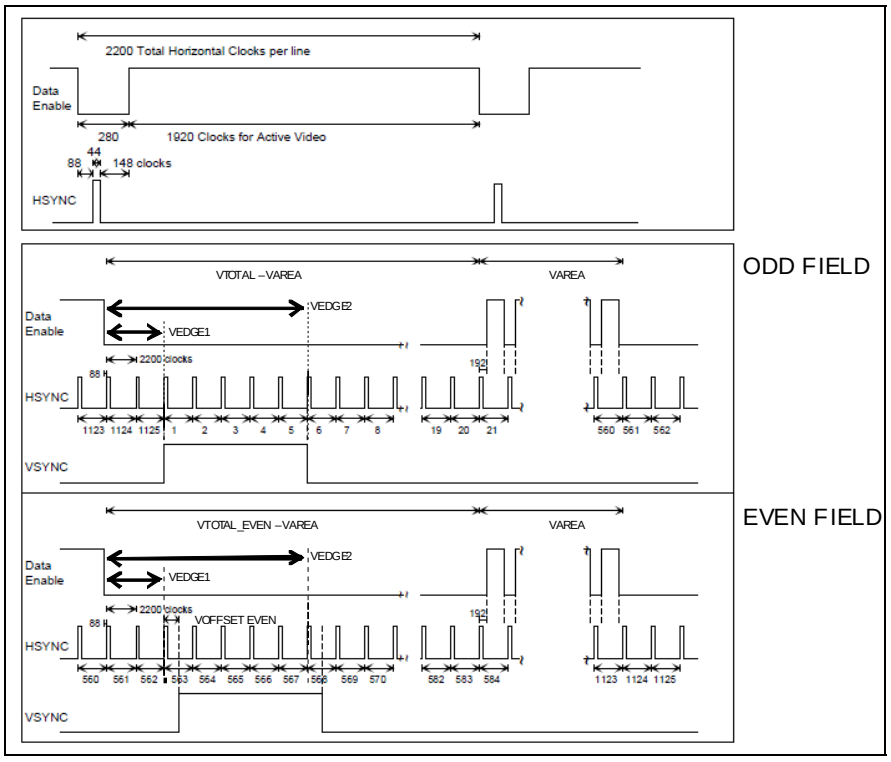
**Figure 4-5. Enable Signal**



### 4.1.8 Interlace output mode

The occasions of this mode are ODD_FIELD and EVEN_FIELD, and it's possible to set the vertical direction location of the vertical sync signal and the horizontal direction location separately.
The parameters which can be established separately are VTOTAL (the vertical direction total), VEDGE1/2 (the vertical sync signal vertical direction synchronous location) and VOFFSET (the vertical sync signal horizontal direction synchronous location).
A vertical synchronizing signal control registers can establish each function effectively/invalidly.

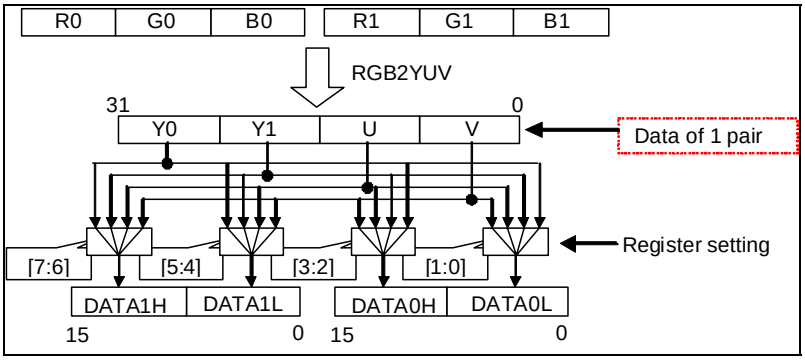
**Figure 4-6. Vertical sync signal setting (When the value is 1980×1080.)**



## 4.2 YUV output interface

### 4.2.1 Image data

It corresponds to YUV422 (Interlace) Format.
YUV change is performed by the 2Pixel unit, and data of RGB888 Format is output.
(It doesn't correspond to input of RGB565/666 Format.)
It's possible to output by the 1CLK 16bit unit and change the output order of Y0, Y1, U, V by register setting.




### 4.2.2 Format transform

RGB-> YUV transform is calculated by the following equation.

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} Y0 & Y1 & Y2 & Y3 \\ U0 & U1 & U2 & U3 \\ V0 & V1 & V2 & V3 \end{pmatrix} * \begin{pmatrix} R \\ G \\ B \\ 1 \end{pmatrix} * \frac{1}{256}$$

12 calculation coefficients, with all signs, it's possible to establish it at 11 bits of value (-1024-1023).
Further an original value of each coefficient is the following. (ITU-BT601 is based.)

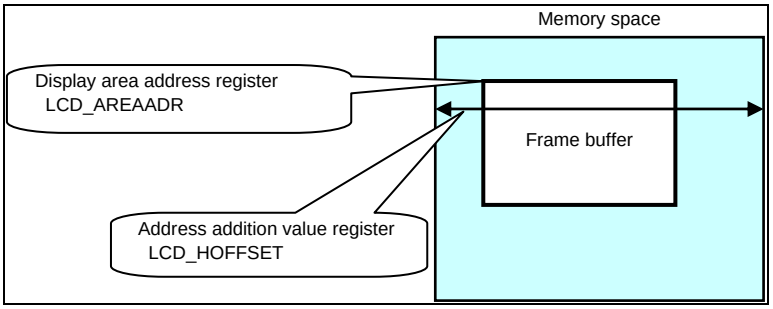
$$\begin{pmatrix} Y0 & Y1 & Y2 & Y3 \\ U0 & U1 & U2 & U3 \\ V0 & V1 & V2 & V3 \end{pmatrix} = \begin{pmatrix} 66 & 129 & 25 & 16 \\ -38 & -74 & 112 & 128 \\ 112 & -94 & -18 & 128 \end{pmatrix}$$

## 4.3 Frame buffer and Data buffer

### 4.3.1 Frame buffer

Buffer that stores a screen of image data is collectively called frame buffer. The LCD controller can set one screen. The display area address register is used to set the start address of the frame buffer, by any address. In addition, the horizontal direction size of the frame buffer area can be specified by the address addition value register (word boundary). Using this, the rectangle area cropped from the frame buffer mapped larger than the displayed image size can be output to an LCD panel.

**Figure 4-7. Frame Buffer**




The number of horizontal pixels of image data to be stored in the frame buffer can be set with the HAREA bit. The volume of image data varies depending on the data format.

- When the input format is RGB565: 32 bytes with 16 pixels (8 words)
- When the input format is RGB666: 36 bytes with 16 pixels (9 words)
- When the input format is RGB888: 48 bytes with 16 pixels (12 words)

Therefore, set a value that satisfies the following conditions in the address addition value register.

- When the input format is RGB565: LCD_HOFFSET ≥ HAREA / 16 × 32
- When the input format is RGB666: LCD_HOFFSET ≥ HAREA / 16 × 36
- When the input format is RGB888: LCD_HOFFSET ≥ HAREA / 16 × 48

**Caution In the LCD controller in EM/EV, the value that can be specified as the number of horizontal pixels is defined as a multiple of 2. Consequently, the condition for setting the HOFFSET value changes as follows.**

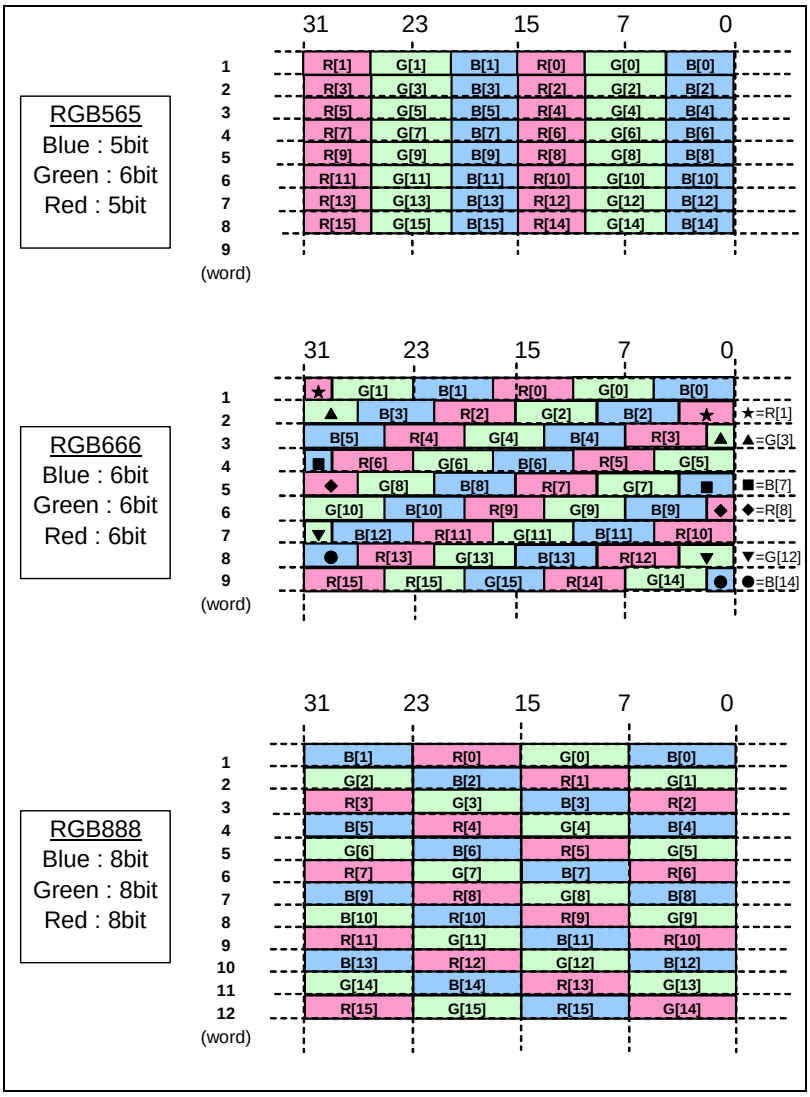
**When the input format is RGB565: LCD_HOFFSET ≥HAREA / 2 × 4**

**In the case of RGB666, one pixel is composed of 18 bits. That is, data amount in one line is HAREA × 18 (bits). Divide the data amount in word units (32 bits), round up the fractional part, and set to HOFFSET a value of the obtained result or larger.**

### 4.3.2 Frame buffer storage format

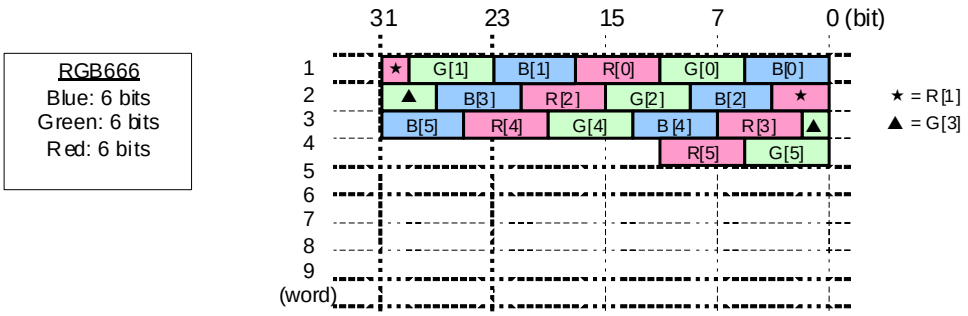
When storing image data in the frame buffer, fill the last word of the line thoroughly. Store RGB666 data in 36 bytes, store RGB565 data in 32 bytes and store RGB888 data in 48 bytes, in 16-pixel units.

**Figure 4-8. Frame Buffer Storage Format**



The following shows an example of memory storage when the number of horizontal pixels is not a multiple of 16.

**Figure 4-9. When Number of Horizontal Pixels Is Not a Multiple of 16**




For example, when the input format is RGB666 and the number of horizontal pixels is 6, data of 6 pixels can be stored as long as there is a space of at least 4 words, as shown in the above figure. Therefore, set a value equivalent to 4 words or more as the address addition value.
In the fifth word, store the first pixel in the second line, starting from the LSB.
(Bits 13 to 31 of the forth word are not used.)

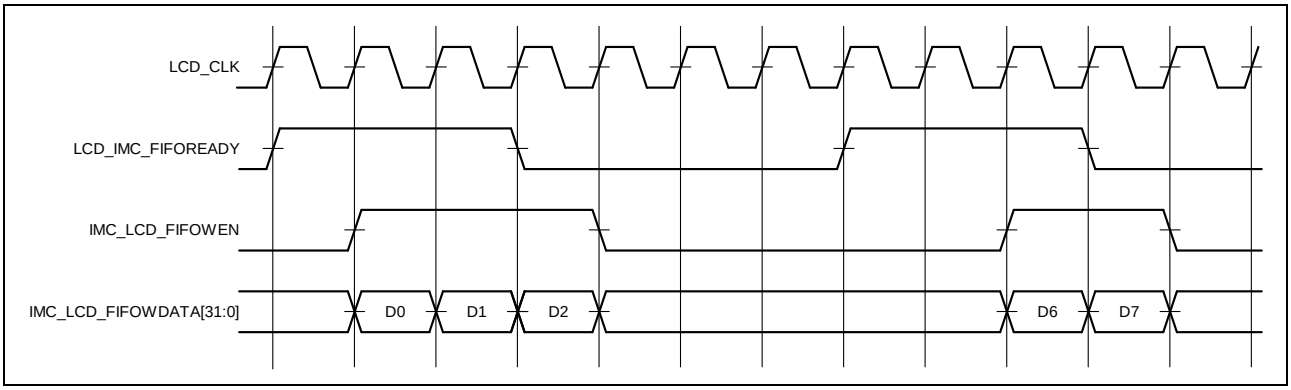
### 4.3.3 Access of frame buffer

DirectPath access between the MEMC-LCD and Local Bus access between the IMC-LCD is performed selectively by the BUSSEL set value. The respective interface rules are indicated in the following.

Local Bus

- When there is 8 words of space in FIFO, LCD sets FIFOREADY.
- Set FIFOREADY never drops it until the bit of FIFOWEN rises.
- LCD has to receive the data when being FIFOWEN=1, certainly.
- IMC checks the rising of FIFOREADY, sets the data which has finished synthetic processing in FIFOWDATA and also sets FIFOWEN at the same time.

**Figure 4-10. Frame buffer read (Local Bus)**




DirectPath

- Transfer to DPMODE between the MEMC-LCD in async at the time of DP mode use by the frame unit..
- LCD sets CLKREQ for MEMC 1 cycle before of CS issue.
- After that the address (26-or 3 bits) and CSZ which correspond to 8 words are issued. -> By REQEN and a CS simultaneous activist, CS release.
- It'll be once's address set and be 8 burst read.
- 32 bit data in an active period of VALID (8 cycles which continued) are taken in.
- Unnecessary data is read and thrown away.

**Figure 4-11. DirectPath mode transfer**

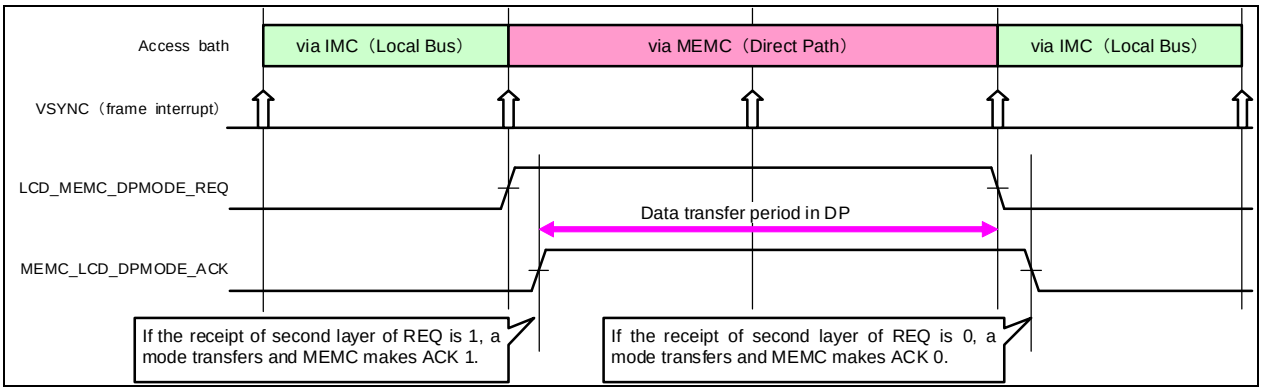
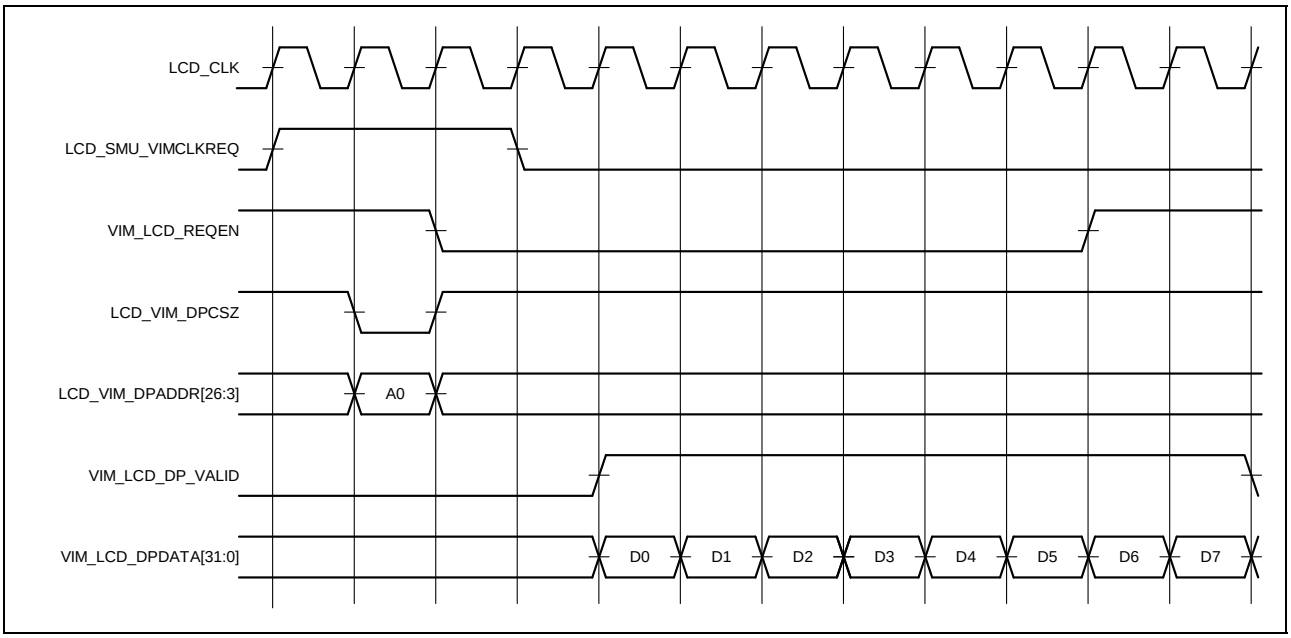


**Figure 4-12. Frame buffer read (DirectPath)**

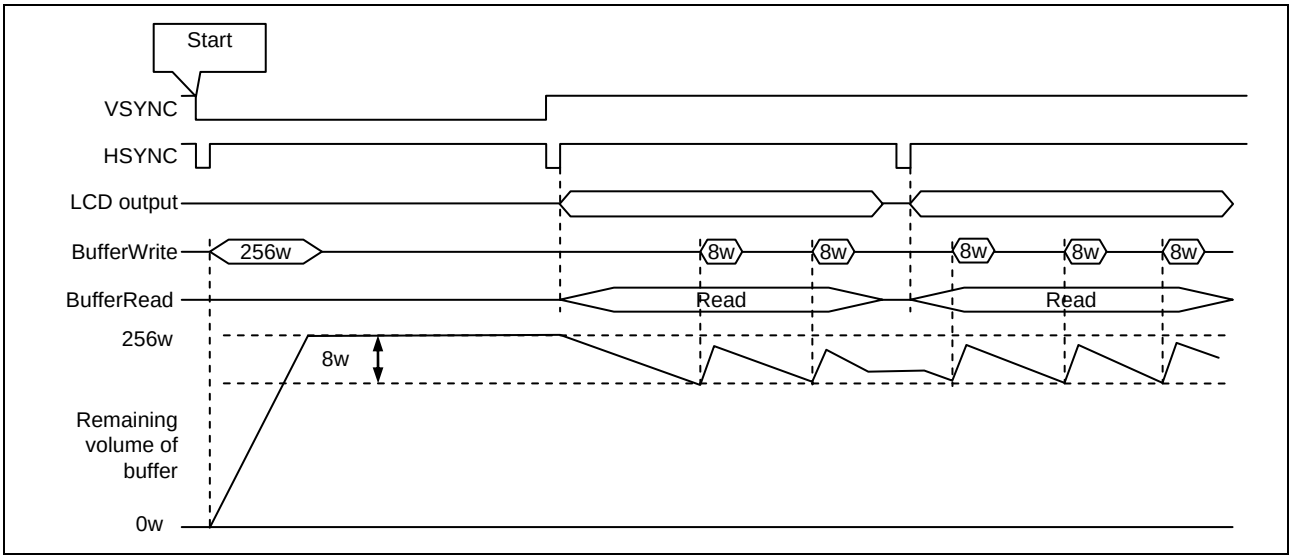


### 4.3.4 Data buffer

The data buffer is incorporated in the LCD controller and captures the image data read from the frame buffer. The data buffer consists of two ports (one read port and one write port) of 32 bits × 128 words and is used as a FIFO. When there is an available space of 8 words in the data buffer (when DATAREQ bit = 0 (initial value)), data is written to the write port of the data buffer via the frame buffer interface. The read port is used for reading out (LCD displaying) data from the LCD interface.

Figure 4-13 shows accessing the data buffer. First, image data is written to the data buffer via the frame buffer interface. Next, the LCD interface reads the image data from the area to which data was written via the frame buffer interface, and performs LCD display. After that, image data is written to the data buffer via the frame buffer interface when the data buffer has an available space of 8 words (when DATAREQ bit = 0 (initial value)). If the data buffer read speed via the LCD interface is faster than the buffer write speed, an under-run interrupt is generated.

**Figure 4-13. Data Buffer Access**




**Caution If the data transfer rate is not fast enough in comparison with the LCD panel image refresh rate, the image data amount is insufficient, which results in a fatal image deterioration.**
**To avoid this, determine the clock cycle so that the following expression is sufficiently met.**
**1. Pixel clock frequency <<< Main clock frequency**
**(EM/EV specification: PIXCLK = 6 to 50 MHz, LCD_CLK = 166 MHz)**

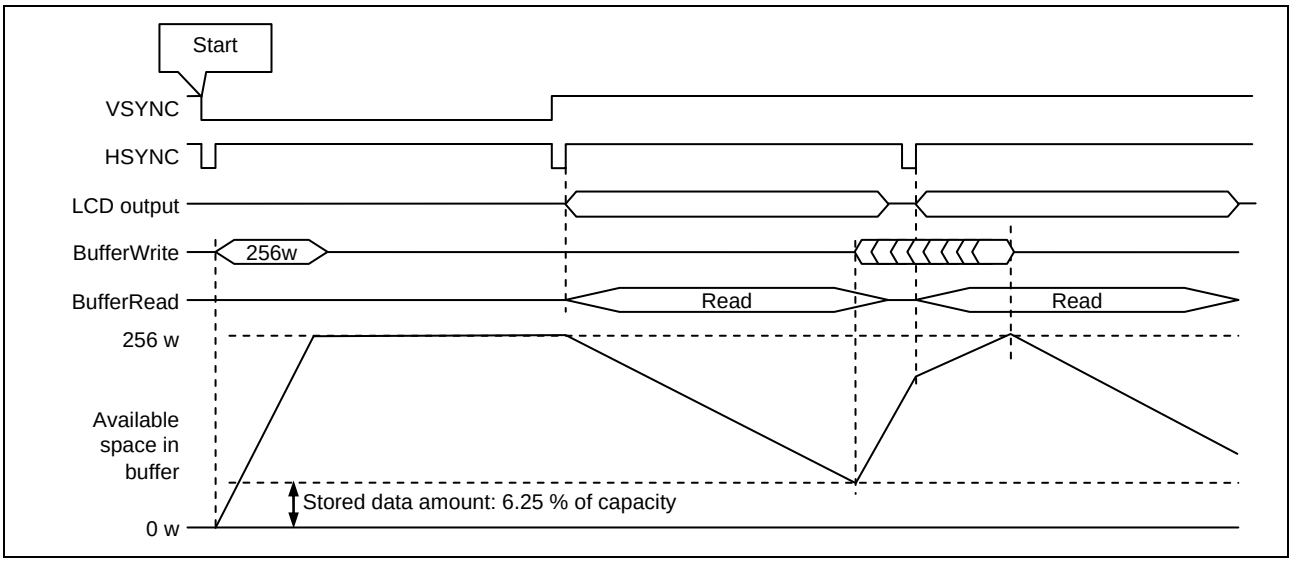
### 4.3.5 Data request cycle setting

When the frame buffer is mapped on SDRAM, usually data is written to the data buffer if there is an available space of 8 words. Due to this, SDRAM is frequently accessed and thus effective power management is disturbed. The data request cycle register (LCD_DATAREQ) can be used to concentrate issuance of SDRAM access requests in a specific period.

For example, when 6.25% is set in the LCD_DATAREQ register and the data stored in the data buffer decreases to 6.25% or less, data is continuously read from the frame buffer until the data buffer becomes full. Refer to the following figure for the operation. Note that a buffer under-run is more likely to occur if reading of data is stalled due to a certain cause.

**Caution The following figure just shows a concept of operation and values in the figure does not necessarily match those of the actual operation.**

**Figure 4-14. Access When 6.25% Is Set in Data Request Cycle Setting Register**



## 4.4 Hardware cursor function

Overlays is equipped with the possible hardware cursor function in an indication output picture in this macro in a LCD controller.

The explanation about the hardware cursor function is mentioned at this chapter.

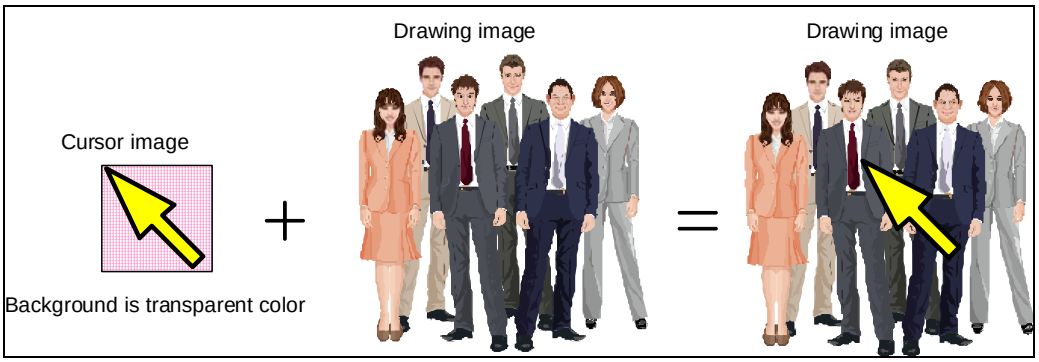
### 4.4.1 Details of specification

Without using the outside macro for the interior with a memory for cursor storing data, it's possible of this LCD controller to indicate a hardware cursor. Cursor data is 64×64 pixel, and 1 pixel is established by 256 colors of color palette value. 1 color palette consists of RGB565 form (16 bits). Palette number 0 is dealt with as transparent color. It isn't possible to make the transparent color function invalidity.

It'll be at most 128 × 128 pixel as the drawing size with the simple double resizing function.

A memory for cursor storing data consists of a memory of 2, A side and a B side. 1 of cursor data can be stocked in each, and it's chosen, and it indicates whether it's both in LCD. During indicating one cursor data, the double buffer control by which other data is rewritten is possible.

**Figure 4-15. Cursor function**



### 4.4.2 RAM access for cursors

There is 2 kinds of memory for cursors. To establish 1 color palette, another is to establish cursor data.
A memory for color palette is a memory of 16bit×256word. 1 palette is 16bit data of RGB565 form and consists of 256 tables.
There are 2, A side and a B side, and a memory for cursor data is a memory of 32bit×1024word respectively. 1 cursor of 64×64 pixel is stocked in 1 memory.
Both memories do data writing in through APB.
It's possible to access all memories in the state which made during an indication stop or the cursor function invalidity.
Whole LCD indication output after we assumed that the cursor function was effective, can't access a memory for color palette through APB. Only the side where a memory for cursor data isn't chosen by LCD indication is accessible.
A list of APB access pros and cons is indicated in table 4-4.

**Table 4-4. APB access to a memory for cursors**

| | During a LCD indication stop | Cursor function invalidity | During A side cursor indication | During B side cursor indication |
|---|---|---|---|---|
| Memory for color palette | ○ | ○ | × | × |
| Memory for cursor data A side | ○ | ○ | × | ○ |
| Memory for cursor data B side | ○ | ○ | ○ | × |

(○ : APB is accessible, × : APB access prohibited)

Access to a memory for color palette is achieved by "cursor palette RAM register". This consists of 256 of entry (the address), and the value of RGB565 form (16 bit) which answered to 1 palette number with 1 of entry is established. It's possible to read data of a memory by reading from a register continuously twice because it's possible to refer to memory data by reading this register, but it isn't possible to read data by the 1st time of lead access by latency of a synchronous SRAM.

Access to a memory for cursor data is achieved by "0 registers of cursor data RAM" and "a register of cursor data RAM". RAM0 register accesses a memory in A side, and RAM1 register accesses a memory in a B side. It consists of 1024 of entry (the address) respectively, and the palette number for 4 pixel of level is established in the 1st address. The 2nd time can read the price of the memory by reading a register continuously twice about the reading of the memory data.

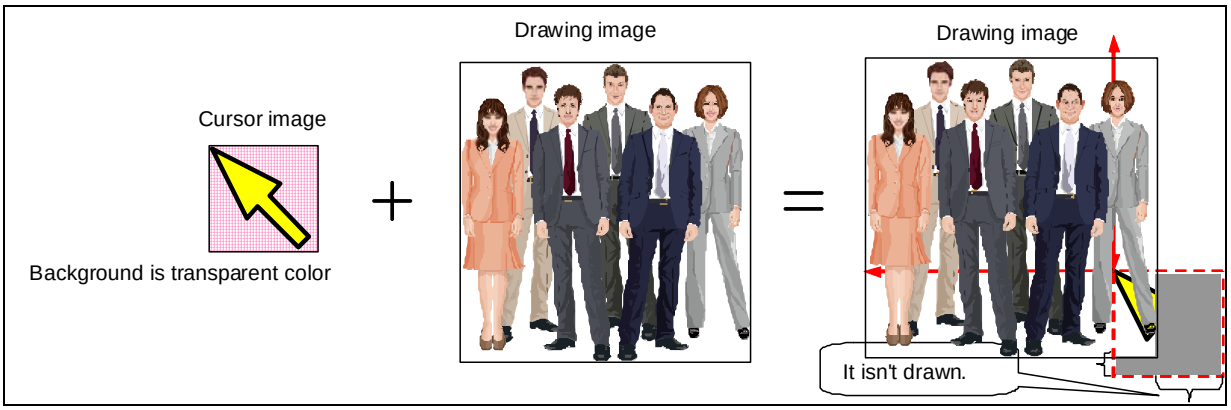
### 4.4.3 The indication point of view and the simple resizing function

The indication location of the cursor is set by "cursor horizontal register" and "cursor vertical register". When the starting point location of the cursor is set as the CURPOSH bit and the CURPOSV bit, a pixel in the upper left of a cursor picture is drawn by the set location.

It's possible to establish setting in the indication location at the reach to the available picture area size (horizontal direction indication territory register and vertical way indication territory register). Setting beyond the screen size isn't also prohibited, but nothing appears in an indication screen.

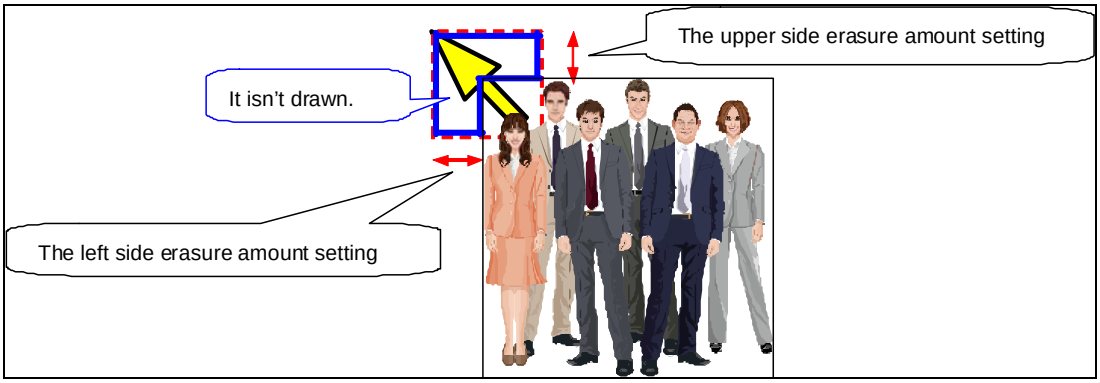
The setting out of which a cursor indication area sticks in the right side and the underside to available picture area territory is also possible. The cursor picture until I stick up is indicated in this case.

**Figure 4-16. Cursor indication possible territory**



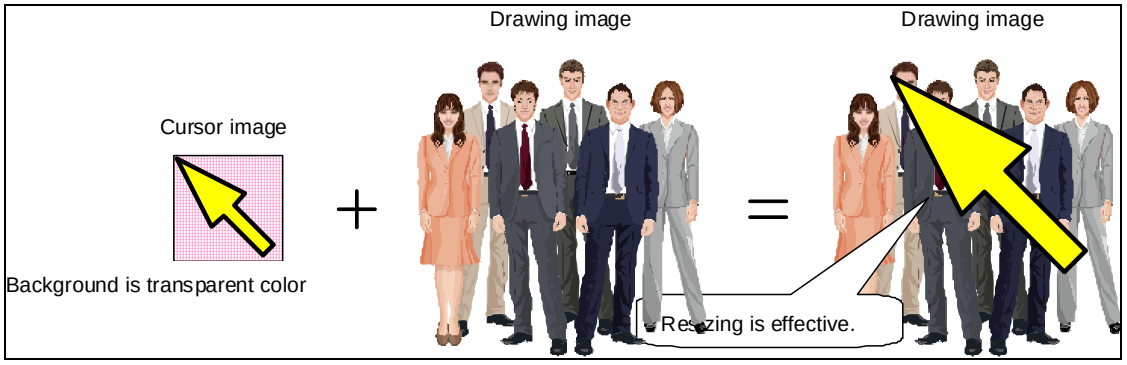

The cursor indication location erases the left side or the upper side of the cursor picture respectively in case of the left end (CURPOSH=0) or upper end (CURPOSV=0), and can indicate it. The left erasure amount is set as the LEFT_ERASE bit and the erasure amount of the upper side is set as the UP_ERASE bit. It's possible to indicate it as a cursor stuck out in the left side and the upper side seemingly by this setting. The setting reach of the erasure amount setting is 63 from 0. Therefore at least 1 pixel is shown to a screen.

**Figure 4-17. The erasure amount setting of a cursor**




The cursor simple resizing function becomes effective by making bit 0 of "cursor operation register" (CURRSZ) 1. When the simple resizing function becomes effective, it's possible to magnify cursor drawing territory with 1 pixel of a cursor picture is stretched in 2×2 pixel of horizontal/verticalness, and drawing it in 128×128 pixel. The erasure range by the erasure amount setting of a cursor also becomes double, and erasure territory will be 0-126 picture element (2 pixel unit) to set value 0-63.

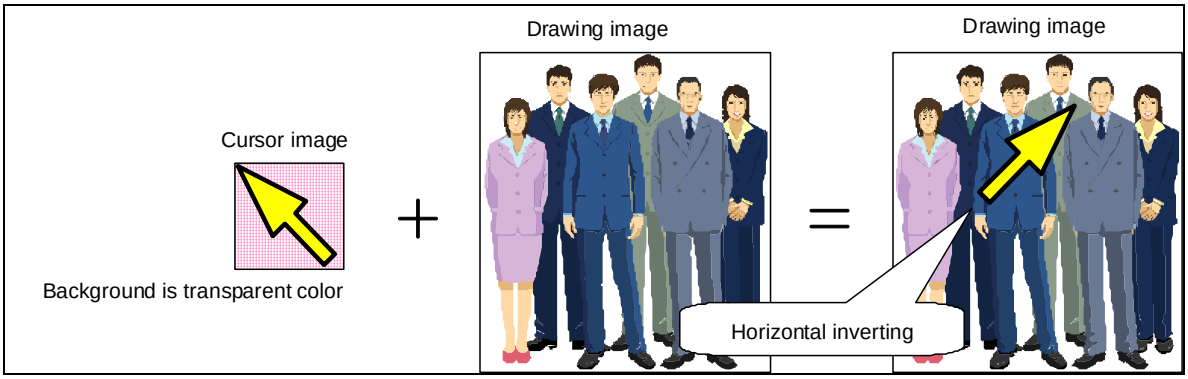
**Figure 4-18. Simple resizing of a cursor**




### 4.4.4 Cursor inverting function

It's reversed in horizontal direction or the verticalness direction, and it's possible to indicate a cursor without rewriting a cursor picture when cursor inverting function is used.

**Figure 4-19. Cursor inverting function**



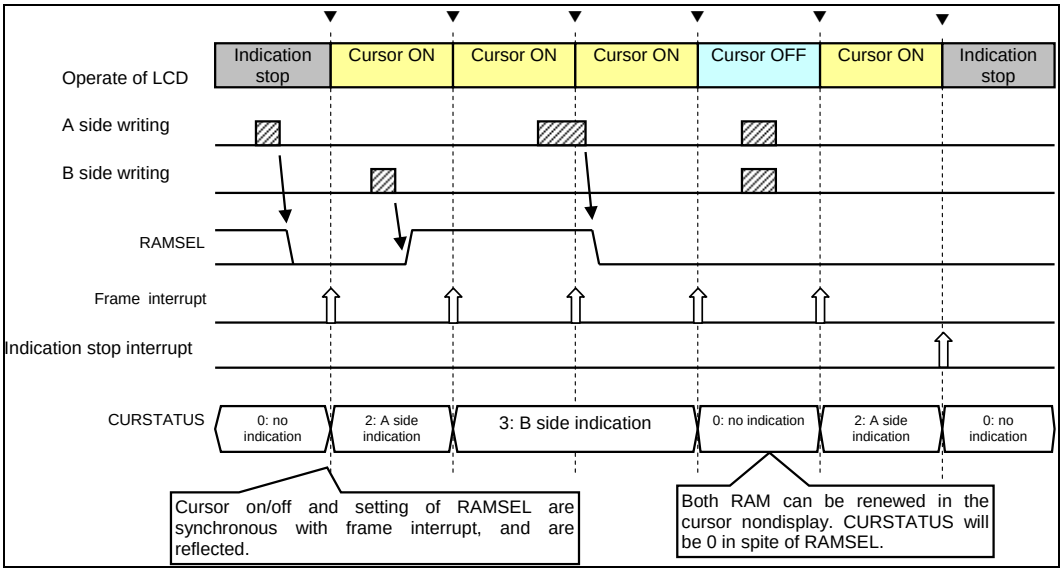
### 4.4.5 Double buffer control

LCD controller changes a cursor picture in A side and a B side, and is to show to LCD, and it's synchronous with a frame, and it's possible to change a cursor picture.

The cursor data shown to LCD is chosen by the RAMSEL bit of "cursor data RAM choice register". The established value is synchronous with frame interrupt, and is taken in the interior, and is reflected by LCD indication. It's possible to read the reflected price by the CURSTATUS bit of "cursor status register".

Flow of control is indicated on figure 4-20.
Cursor data is written in during a LCD indication stop, in the cursor nondisplay and an optional cursor data RAM. If writing in is completed, it's set in the RAM side where the RAMSEL bit has been completed and the cursor indication function is turned on.
The CURSTATUS bit synchronizes with frame interrupt after that, and is renewed, so it write, judge possible RAM and rewrite it while checking the operating state of LCD controller.
For example it's a B side when it's that itI doesn't make for change and is during B side indication then when CURSTATUS after frame interruption isn't in the state of the A side indication after setting RAMSEL in A side, it can't be rewritten.
When CURSTATUS makes for change, and is A side indication, they're accessible to RAM in a B side until RAMSEL is made a B side personally.

**Figure 4-20. Double buffer control**



# 5. Usage Procedures

The following shows general procedures for using the LCD controller.

## 5.1 Starting LCD display

An example of the setting procedure to start LCD display is described below.

<1> Set the control register (LCD_CONTROL) according to the specifications of the LCD panel connected.
<2> In the same manner, set the parameters related to SYNC and effective pixels, using the following registers.
Horizontal direction total register (LCD_HTOTAL)
Horizontal direction display area register (LCD_HAREA)
Horizontal synchronization edge 1 register (LCD_HEDGE1)
Horizontal synchronization edge 2 register (LCD_HEDGE2)
Vertical direction total register (LCD_VTOTAL)
Vertical direction display area register (LCD_VAREA)
Vertical synchronization edge 1 register (LCD_VEDGE1)
Vertical synchronization edge 2 register (LCD_VEDGE2)
<3> Perform settings related to frame buffers according to the usage rule of internal memory.
Display area address register (LCD_AREAADR)
Address addition value register (LCD_HOFFSET)
Input format register (LCD_IFORMAT)
Simple resize register (LCD_RESIZE)
<4> Set the access bus select register (LCD_BUSSEL).
When accessing the local bus via the IMC is set, separately set parameters for the IMC.
<5> Set “1” in the display register (LCD_LCDOUT) to start LCD display.

**Remark** The setting of <1> to <4> is not in particular order.
Only the settings of <3> and <4> can be changed during operation in frame units.
Change of settings of <1> and <2> during operation is not supported, so be sure to stop operating before change the settings.

## 5.2 Stopping LCD Display

When the LCDOUT bit of the display register (LCD_LCDOUT) is set to “0”, the LCD controller stops operation after transfer of the frame currently being output to display is completed.
Whether the LCD display has stopped can be checked by detecting a display stop interrupt or by polling the STATUS bit of the status register (LCD_STATUS).

## 5.3 Mode Change During Operation (BUSSEL)

Local bus access via the IMC and fixed-value output mode can be switched by setting the access bus select register (LCD_BUSSEL). The setting change takes effect after an LCD frame interrupt occurs after the register values are rewritten. The actual operating mode can be checked by reading the status register immediately after occurrence of an LCD frame interrupt.

## 5.4 VGA Standby Mode Use Procedure

### 5.4.1 When data to be displayed has been stored in frame cache memory

If the LCDOUT bit of the display register (LCD_LCDOUT) is set to 1 while the BUSSEL bit of the access bus select register (LCD_BUSSEL) is set to 1, the LCD controller operates VGA standby mode by using the DirectPath between the MEMC and LCD controller.

### 5.4.2 When data to be displayed has not been stored in frame cache memory

If the LCDOUT bit of the display register (LCD_LCDOUT) is set to 1 while the BUSSEL bit of the access bus select register (LCD_BUSSEL) is set to 2 or 3, the LCD controller operates in the ordinary image synthesis display mode by using the local bus between the IMC and LCD controller and requests WB to the IMC macro. Completion of WB can be checked by detecting a WB end interrupt issued by the IMC macro.
When BUSSEL = 010, the LCD controller automatically enters the ordinary image synthesis display mode after completion of WB. In this case, manually switch to the cache display (BUSSEL = 1).
In either automatic transition, the transition status can be checked by reading the MODESTATUS bit of the status register (LCD_STATUS) immediately after a vertical synchronization interrupt occurs.

Caution When access to DirectPath isn't performed by the address which continued, in (in case of twice's linear reading by using the case when the large address addition amount setting was done relatively and simple double resizing to the level number of pixels), in Pre Fetch territory on the MEMC side, Hit, latency in DirectPath will be large, and a possibility that an under-run occurs becomes high. Meet following condition, and use to secure the most suitable performance.

Condition1 When not using the resizing function, GAP between the line (finite difference of the level effective pixel amount of data and the address addition amount register value) is made 0. (When the amount of data is incomplete, we assume that it's smallest.)
Condition2 When using the resizing function, the amount of data of 1 line is made 1KByte below.

Note When under-run occurred at the time of DirectPath use, more image data of the next frame is disordered by the frame which is being indicated as of it and genesis timing. When Bus congestion is dissolved, even if everything doesn't set the LCD side in particular, I return to normal movement.

## 5.5 Use of the cursor function

Before beginning LCD indication when using the cursor function, it's necessary to write data in a memory for cursor palettes and a memory for cursor data.

<1> Preparations of memory data

Data necessary to following address is written in.

1st to 255th address of “Cursor palette RAM register” .

Everything in 1024th address in one of “ Cursor data RAM0 register (A side) ” or “ Cursor data RAM1 register (B side) “.

Data writing in to a memory can be written in by once's register light.

(Reading is read twice and it's needed.)

The side where cursor data was stocked in "cursor data RAM choice register" is set.

<2> Use of usually

Indication of a cursor and nondisplay are established by "cursor enable register". It becomes effective after the frame interrupt occurrence by which setting is after that.

Reverse and simple resizing are established by "cursor operation register" and setting in the indication location is changed by "cursor horizontal register" and "cursor vertical register". The set value becomes effective after frame interrupt occurrence.

<3> Change of cursor picture

The cursor picture is changed as follows.

(1) It wait for occurrence of frame interrupt and read "cursor status register".

(2) It's confirmed that a cursor in the side chosen by "cursor data RAM choice register" becomes during indication. When it isn't during indication, setting isn't reflected yet by the frame, so I wait for the next frame interrupt.

(3) New cursor picture is written in the side which isn't used for indication (when the occasions during A side indication are the B side and during B side indication, A side).

(4) The side where data was written in "cursor data RAM choice register" newly is set. Setting is synchronous with frame interrupt, and is reflected.

When changing the picture once again, it's repeated from the reading of the first "cursor status register" of a procedure.

| REVISION HISTORY | EMMA Mobile EV2 User's Manual: LCD Controller |
|---|---|

| Rev. | Date | Description | |
|---|---|---|---|
| | | Page | Summary |
| 1.00 | Jan 29, 2010 | — | 1st revision release |
| 2.00 | Jun 7, 2010 | — | Incremental update from comments to the 1.0. |
| 3.00 | Jun 30, 2010 | — | Incremental update from comments to the 2.0.<br>(A change part from the old revision is "★" marked in the page left end.) |
| 4.00 | Sep 30, 2010 | — | Incremental update from comments to the 3.0.<br>(A change part from the old revision is "★" marked in the page left end.) |
| 5.00 | May 31, 2011 | — | Incremental update from comments to the 4.0. |
| 6.00 | Sep 30, 2011 | — | Incremental update from comments to the 5.0. |
| | | 25 | Table 3-1 corrected. (FIELD interrupt added.) |
| | | 26 | Chapter 3.2.20 (1) corrected. |
| | | 27 | Chapter 3.2.20 (2) corrected. |
| | | 28 | Chapter 3.2.20 (3) corrected. |
| | | 30 | Chapter 3.2.20 (4) corrected. |
| | | 31 | Chapter 3.2.20 (5) corrected. |
| 7.00 | Dec 21, 2011 | — | Incremental update from comments to the 6.0. |
| | | 9 | Chapter 3.2.1 corrected. (bit[1] and [0] Function) |

EMMA Mobile EV2 User's Manual: LCD Controller

Publication Date: Rev.1.00 Jan 29, 2010
Rev.7.00 Dec 21, 2011

Published by: Renesas Electronics Corporation

**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.**
7F, 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 1.0

# LCD Controller

EMMA Mobile EV2

Renesas Electronics Corporation

R19UH0044EJ0700