Please note that Cypress is an Infineon Technologies Company.
The document following this cover page is marked as “Cypress” document as this is the company that originally developed the product. Please note that Infineon will continue to offer the product to new and existing customers as part of the Infineon product portfolio.

Continuity of document content
The fact that Infineon offers the following product as part of the Infineon product portfolio does not lead to any changes to this document. Future revisions will occur when appropriate, and any changes will be set out on the document history page.

Continuity of ordering part numbers
Infineon continues to support existing part numbers. Please continue to use the ordering part numbers listed in the datasheet for ordering.

CY91590 Series

FR81S Family GDC Manual

Document Number: 002-05580 Rev. *B

Cypress Semiconductor
198 Champion Court
San Jose, CA 95134-1709
http://www.cypress.com

© Cypress Semiconductor Corporation, 2013-2019. This document is the property of Cypress Semiconductor Corporation and its subsidiaries (“Cypress”). This document, including any software or firmware included or referenced in this document (“Software”), is owned by Cypress under the intellectual property laws and treaties of the United States and other countries worldwide. Cypress reserves all rights under such laws and treaties and does not, except as specifically stated in this paragraph, grant any license under its patents, copyrights, trademarks, or other intellectual property rights. If the Software is not accompanied by a license agreement and you do not otherwise have a written agreement with Cypress governing the use of the Software, then Cypress hereby grants you a personal, non-exclusive, nontransferable license (without the right to sublicense) (1) under its copyright rights in the Software (a) for Software provided in source code form, to modify and reproduce the Software solely for use with Cypress hardware products, only internally within your organization, and (b) to distribute the Software in binary code form externally to end users (either directly or indirectly through resellers and distributors), solely for use on Cypress hardware product units, and (2) under those claims of Cypress’s patents that are infringed by the Software (as provided by Cypress, unmodified) to make, use, distribute, and import the Software solely for use with Cypress hardware products. Any other use, reproduction, modification, translation, or compilation of the Software is prohibited.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CYPRESS MAKES NO WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, WITH REGARD TO THIS DOCUMENT OR ANY SOFTWARE OR ACCOMPANYING HARDWARE, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. No computing device can be absolutely secure. Therefore, despite security measures implemented in Cypress hardware or software products, Cypress shall have no liability arising out of any security breach, such as unauthorized access to or use of a Cypress product. CYPRESS DOES NOT REPRESENT, WARRANT, OR GUARANTEE THAT CYPRESS PRODUCTS, OR SYSTEMS CREATED USING CYPRESS PRODUCTS, WILL BE FREE FROM CORRUPTION, ATTACK, VIRUSES, INTERFERENCE, HACKING, DATA LOSS OR THEFT, OR OTHER SECURITY INTRUSION (collectively, “Security Breach”). Cypress disclaims any liability relating to any Security Breach, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any Security Breach. In addition, the products described in these materials may contain design defects or errors known as errata which may cause the product to deviate from published specifications. To the extent permitted by applicable law, Cypress reserves the right to make changes to this document without further notice. Cypress does not assume any liability arising out of the application or use of any product or circuit described in this document. Any information provided in this document, including any sample design information or programming code, is provided only for reference purposes. It is the responsibility of the user of this document to properly design, program, and test the functionality and safety of any application made of this information and any resulting product. “High-Risk Device” means any device or system whose failure could cause personal injury, death, or property damage. Examples of High-Risk Devices are weapons, nuclear installations, surgical implants, and other medical devices. “Critical Component” means any component of a High-Risk Device whose failure to perform can be reasonably expected to cause, directly or indirectly, the failure of the High-Risk Device, or to affect its safety or effectiveness. Cypress is not liable, in whole or in part, and you shall and hereby do release Cypress from any claim, damage, or other liability arising from any use of a Cypress product as a Critical Component in a High-Risk Device. You shall indemnify and hold Cypress, its directors, officers, employees, agents, affiliates, distributors, and assigns harmless from and against all claims, costs, damages, and expenses, arising out of any claim, including claims for product liability, personal injury or death, or property damage arising from any use of a Cypress product as a Critical Component in a High-Risk Device. Cypress products are not intended or authorized for use as a Critical Component in any High-Risk Device except to the limited extent that (i) Cypress’s published data sheet for the product explicitly states Cypress has qualified the product for use in a specific High-Risk Device, or (ii) Cypress has given you advance written authorization to use the product as a Critical Component in the specific High-Risk Device and you have signed a separate indemnification agreement.

Cypress, the Cypress logo, Spansion, the Spansion logo, and combinations thereof, WICED, PSoC, CapSense, EZ-USB, F-RAM, and Traveo are trademarks or registered trademarks of Cypress in the United States and other countries. For a more complete list of Cypress trademarks, visit cypress.com.

Preface

Thank you for your continued use of Cypress semiconductor products.

Read this manual and "Datasheet" thoroughly before using products in the CY91590 series.

How to Use This Manual

Finding a function

The following methods can be used to search for the explanation of a desired function in this manual:

- Search from the table of the contents

  The table of the contents lists the manual contents in the order of description.

- Search from the register

  The register list for this device has been described. You can look up the name of a desired register on the list to find the address of its location or the page that explains it.

- Search from the index

  You can look up the keyword such as the name of a peripheral function in the index to find the explanation of the function.

About the chapters

Basically, this manual explains 1 function per chapter.

Terminology

This manual uses the following terminology.

Term	Explanation
Word	Indicates access in units of 32 bits.
Half word	Indicates access in units of 16 bits.
Byte	Indicates access in units of 8 bits.

Access unit and Bit position

One example is shown about the register used in this manual.

(Example) 1.22.6.3 HS-SPI Peripheral Communication Configuration Registers (HSSPIn_PCC0 to 3)

These registers are used for making various settings regarding serial communication of slave selects 0 to 3.

Bit	7	6	5	4	3	2	1	0
Field	SDIR	SS2CD[1]	SS2CD[0]	SSPOL	RTM	ACES	CPOL	CPHA
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial value	0	0	0	0	0	0	0	0

Meaning of Bit Attribute Symbols

R : Read enabled

W : Write enabled

RM : Reading operation during read-modify-write (RMW) operation

"/" (slash) R/W : Read and write enabled. (The read value is the written value.)

"," (comma) R, W : The read and written values differ from each other. (The read value is different from the written value.)

R0 : The read value is "0".

R1 : The read value is "1".

W0 : This bit must always be written to "0".

W1 : This bit must always be written to "1".

(RM0) : "0" is read by read-modify-write (RMW) operation.

(RM1) : "1" is read by read-modify-write (RMW) operation.

RX : The read value is undefined. (A reserved bit or an undefined bit)

WX : Writing does not affect on the operation. (Undefined bit)

R/W writing examples

- R/W : Read and write enabled (The read value is the written value.)
- R,W : Read and write enabled (The read value is different from the written value.)
- R,RM/W : Read and write enabled (The read value is different from the written value. The written value is read by read-modify-write (RMW) instruction.) An example is a port data register.
- R(RM1),W : Read and write enabled (The read value is different from the written value. For read-modify-write (RMW) instructions, "1" will be read out.) An example is an interrupt request flag.
- R,WX : Read only (Read enabled. Writing has no effect on operation.)
- R1,W : Write only (Write enabled. The read value is "1".)
- R0,W : Write only (Write enabled. The read value is "0".)
- RX,W : Write only (Write enabled. The read value is undefined.)
- R0,W0 : Reserved bit (The written value is "0". The read value is the written value.)
- R0,W0 : Reserved bit (The written value is "0". The read value is "0".)
- R1,W0 : Reserved bit (The written value is "0". The read value is "1".)
- RX,W0 : Reserved bit (The written value is "0". The read value is undefined.)

- R/W1 : Reserved bit (The written value is "1". The read value is the written value.)
- R1,W1 : Reserved bit (The written value is "1". The read value is "1".)
- R0,W1 : Reserved bit (The written value is "1". The read value is "0".)
- RX,W1 : Reserved bit (The written value is "1". The read value is undefined.)
- RX,WX : Undefined bit (The read value is undefined. Writing has no effect on operation.)
- R0,WX : Undefined bit (The read value is "0". Writing has no effect on operation.)

Contents

1. Graphic Display Controller 10

1.1 Overview 11
1.2 Features 12
1.3 Block Diagram 14
1.3.1 CY91F591/2/4/6/7/9 15
1.3.2 CY91F59A/B 16
1.4 Memory Maps 17
1.4.1 CY91F592/4/7/9 18
1.4.2 CY91F591/6 20
1.4.3 CY91F59A/B 22
1.5 Explanation of GDC Macro Operation 24
1.5.1 Basic Settings 24
1.6 Display Controller (DISPLAY) 44
1.6.1 Overview 44
1.6.2 Features 44
1.6.3 Configuration 45
1.6.4 Registers 46
1.6.5 Explanation of Operation 95
1.7 Capture Controller (CAPTURE) 121
1.7.1 Overview 121
1.7.2 Features 121
1.7.3 Positioning in the Overall GDC Macro Configuration 122
1.7.4 Registers 123
1.7.5 Explanation of Operation 154
1.8 Draw Engine (DRAW) 181
1.8.1 Overview 181
1.8.2 Features 181
1.8.3 Configuration 181
1.8.4 Registers 182
1.8.5 Explanation of Operation 205
1.9 Sprite Engine (SPE) 222
1.9.1 Overview 222
1.9.2 Features 222
1.9.3 Configuration 223
1.9.4 Registers 224
1.9.5 Explanation of Operation 251
1.9.6 Interrupts 271
1.9.7 Notes 271
1.10 Graphics Memory (VRAM) 272

1.10.1 Overview 272
1.10.2 Configuration 273
1.10.3 Explanation of Operation 274
1.11 Command Sequencer (CMDSEQ) 276
1.11.1 Overview 276
1.11.2 Features 276
1.11.3 Configuration 277
1.11.4 Registers 278
1.11.5 Explanation of Operation 293
1.11.6 Notes 315
1.11.7 Examples 316
1.12 Run-Length Decompression (RLD) 321
1.12.1 Overview 321
1.12.2 Features 321
1.12.3 Restrictions 321
1.12.4 Configuration 321
1.12.5 Registers 322
1.12.6 Explanation of Operation 335
1.13 Signature Generator (SIG) 340
1.13.1 Overview 340
1.13.2 Features 340
1.13.3 Configuration 340
1.13.4 Registers 341
1.13.5 Explanation of Operation 357
1.14 DMA Controller (DMAC) 363
1.14.1 Overview 363
1.14.2 Features 363
1.14.3 Configuration 363
1.14.4 Registers 364
1.14.5 Explanation of Operation 376
1.14.6 Examples 381
1.15 Memory Controller (MEMC) 382
1.15.1 Overview 382
1.15.2 Configuration 384
1.15.3 Registers 387
1.15.4 Explanation of Operation 395
1.15.5 Connection Examples 415
1.16 SPI Controller (SPICNT) 419
1.16.1 Overview 419
1.16.2 Features 419
1.16.3 Configuration 420
1.16.4 Registers 421
1.16.5 Explanation of Operation 433
1.16.6 Notes 447
1.17 NTSC Decoder 448
1.17.1 Overview 448
1.17.2 Features 448

1.17.3 Configuration 449
1.17.4 Registers 450
1.17.5 Explanation of Operation 463
1.18 Command RAM (CMDRAM) 467
1.18.1 Overview 467
1.18.2 Features 467
1.18.3 Configuration 467
1.18.4 Registers 468
1.18.5 Explanation of Operation 472
1.19 Module Controller (MCNT) 478
1.19.1 Overview 478
1.19.2 Features 478
1.19.3 Configuration 479
1.19.4 Registers 480
1.19.5 Explanation of Operation 504
1.20 Local Bus Bridge 511
1.20.1 Overview 511
1.20.2 Features 511
1.20.3 Configuration 511
1.20.4 Explanation of Operation 512
1.21 GDC Bus Bridge 513
1.21.1 Overview 513
1.21.2 Features 513
1.21.3 Configuration 513
1.21.4 Registers 514
1.22 Hi-speed SPI controller (HS-SPICNT) 516
1.22.1 Overview 516
1.22.2 Features 516
1.22.3 Operation of the Hi-Speed SPI Controller 518
1.22.4 Direct Mode 529
1.22.5 Command Sequencer Mode 534
1.22.6 Register Set of the Hi-Speed SPI Controller 543
1.22.7 Notes on Using the Hi-Speed SPI Controller 631

Revision History 650

Document Revision History 650

1. Graphic Display Controller

This chapter explains Graphic Display Controller (GDC) function.

1.1 Overview

1.2 Features

1.3 Block Diagram

1.4 Memory Maps

1.5 Explanation of GDC Macro Operation

1.6 Display Controller (DISPLAY)

1.7 Capture Controller (CAPTURE)

1.8 Draw Engine (DRAW)

1.9 Sprite Engine (SPE)

1.10 Graphics Memory (VRAM)

1.11 Command Sequencer (CMDSEQ)

1.12 Run-Length Decompression (RLD)

1.13 Signature Generator (SIG)

1.14 DMA Controller (DMAC)

1.15 Memory Controller (MEMC)

1.16 SPI Controller (SPICNT)

1.17 NTSC Decoder

1.18 Command RAM (CMDRAM)

1.19 Module Controller (MCNT)

1.20 Local Bus Bridge

1.21 GDC Bus Bridge

1.22 Hi-speed SPI controller (HS-SPICNT)

1.1 Overview

The Graphic Display Controller macro (referred to below as GDC macro) supports graphic engines (draw and sprite), image input, display output, automatic GDC macro execution using a command list, memory interfaces, data decompression, DMA, etc.

Note: The GDC macro does not support the master function and operates as a slave.

1.2 Features

The GDC macro has the following features.

- Maximum operating frequency
  81 MHz
- GDC AHB BUS
  Round robin support
- Draw engine
  - ☐ Line drawing
  - ☐ BitBlt function
  - ☐ Execution from a display list
  - ☐ 8 bpp indirect color
  - ☐ ARGB1555 direct color
  - ☐ Alpha-blending and anti-aliasing
- Sprite engine
  - ☐ Maximum display of 512 sprites
  - ☐ 32 special sprites that enable automatic animation
  - ☐ 1 bpp, 2 bpp, 4 bpp, and 8 bpp indirect color
  - ☐ ARGB1555, RGB565, ARGB8888 direct color
  - ☐ Can set a color format per sprite
  - ☐ Horizontal inversion and vertical inversion
  - ☐ Alpha-blending
- Image input
  - ☐ Analog video input (NTSC/PAL)
  - ☐ Digital video input (RGB666/555)
  - ☐ ITU-R BT.656 input
  - ☐ Video image expansion/reduction/rotation function support
- Display output
  - ☐ Maximum resolution of 800 × 480
  - ☐ Overlay display (window) of up to 5 layers at a time, including 1 sprite-dedicated layer
  - ☐ Size change for resolutions that can be supported according to the color format
  - ☐ RGB digital output (6 bits × 3 / 8 bits × 3)
- Command list
  - ☐ Automatic command execution by a GDC macro reset start

    Automatic command execution by a reset start using the Hi-speed SPI Controller installed on the CY91F59A/B and SPI Controller installed on the CY91F591/2/4/6/7/9 are not supported.
  - ☐ Automatic command execution by individual triggers
  - ☐ Automatic command execution by a register setting

- Memory interfaces
  - ☐ Built-in memory
    1. Graphics RAM (260KB) ----- (CY91F591/6)
    2. Graphics RAM (800KB) ----- (CY91F59/2/4/7/9)
    3. Graphics RAM (1.8MB) ----- (CY91F59A/B)
    4. Command RAM (8KB: Can select to enable/disable ECC)
  - ☐ External memory

    The following interface controllers are installed.

    (**Note:** Each interface controller is exclusively controlled.)
    1. Memory Controller (referred to below as MEMC)

       Interface: 8-bit/16-bit NOR Flash (SRAM)

       Capacity: 64MB (maximum)
    2. SPI Controller (referred to below as SPICNT)

       Interface: Serial Flash

       Capacity: 16MB (maximum)

       Speed: 20 MHz (maximum)

       Mode: Support for 0
    3. Hi-Speed SPI Controller (referred to below as HS-SPICNT) ----- (CY91F59A/B)

       Interface: Serial/Dual/Quad Flash

       Capacity: 256MB (maximum)

       Speed: 40 MHz (maximum)

       Mode: Support for 0, 1, 2, and 3
- Data decompression
  Run-Length Decompression support
- DMA
  2-channel DMA installed

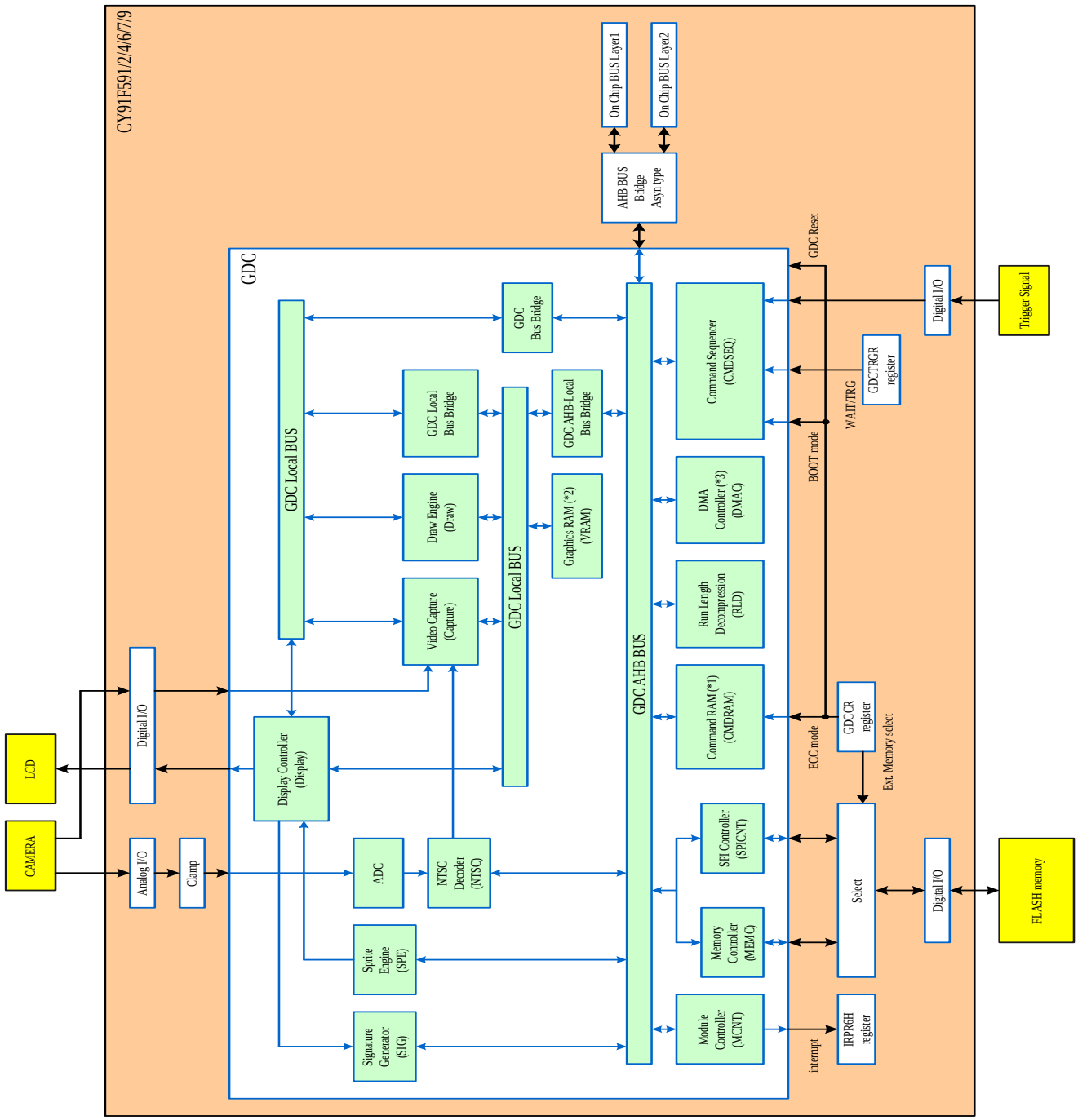
1.3 Block Diagram

The following sections show block diagrams of the GDC macro of the CY91590 series.

Note: The descriptions in this and subsequent sections use the abbreviations written in parentheses in modules in the block diagrams.

1.3.1 CY91F591/2/4/6/7/9

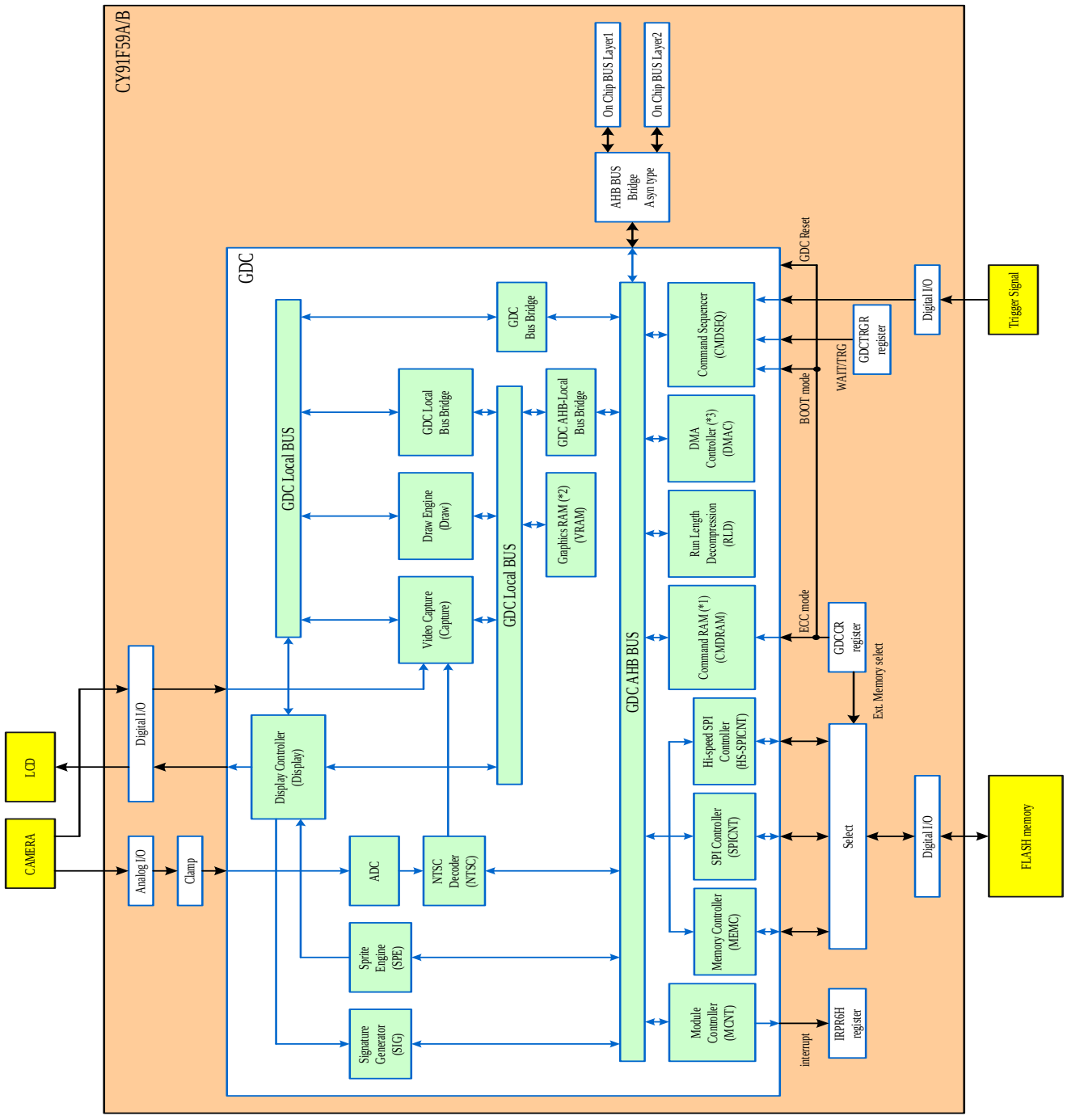
Figure 1-1. Block Diagram (CY91F591/2/4/6/7/9)



(*1) Command RAM	(CMDRAM)	: 8KB of built-in RAM
(*2) Graphics RAM	(VRAM)	: 800KB of built-in RAM (CY91F592/4/7/9)
		: 260KB of built-in RAM (CY91F591/6)
(*3) DMA Controller	(DMAC)	: 2 built-in channels

1.3.2 CY91F59A/B

Figure 1-2. Block Diagram (CY91F59A/B)



(*1) Command RAM (CMDRAM) : 8KB of built-in RAM

(*2) Graphics RAM (VRAM) : 1.8MB of built-in RAM

(*3) DMA Controller (DMAC) : 2 built-in channels

1.4 Memory Maps

This section shows memory maps of the GDC macro of the CY91590 series.

The 2 kinds of addresses written in the memory maps are outlined below.

- GDC address

  This address indicates an address in the GDC macro.
  Use this address for access from each master module in the GDC macro.

  Such addresses in this GDC manual are expressed as shown in the following example.
  Example: 0000_0000_H

- On Chip BUS Layer 1/2 address

  This address indicates an address from the CPU side.

  The address is obtained by adding 40_0000H to a GDC address.

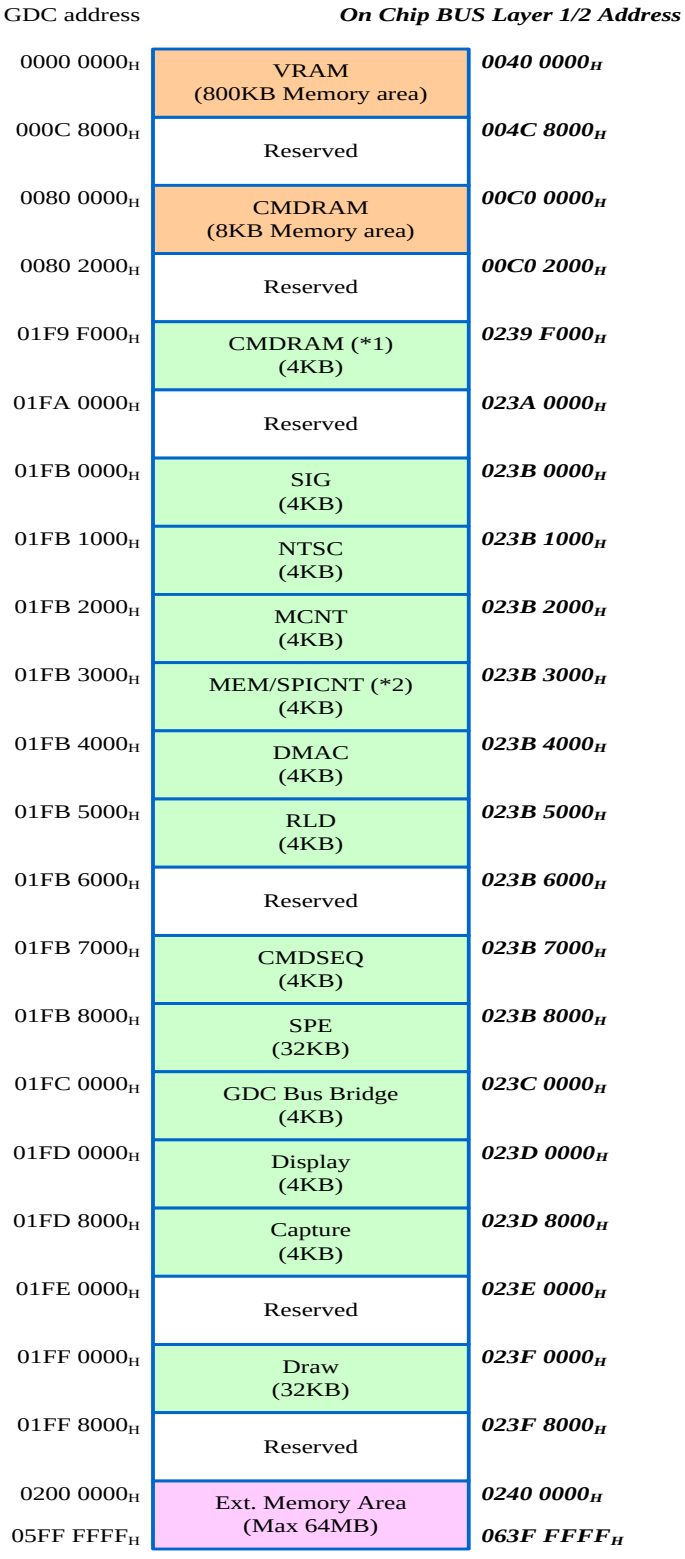
  Use this address for access from the CPU to the GDC macro.

  For an On Chip BUS Layer 1/2 address memory map, see the CY91590 Series Hardware Manual.

  Such addresses in this GDC manual are expressed as shown in the following example.
  Example: 0000_0000_H

1.4.1 CY91F592/4/7/9

Figure 1-3. Memory Map (CY91F592/4/7/9)



(*1) CMDRAM : If the ECC function is not used, CMDRAM becomes a reserved area.

For details on mode settings, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.

For details on functions, see "1.18 Command RAM (CMDRAM)" in this hardware manual.

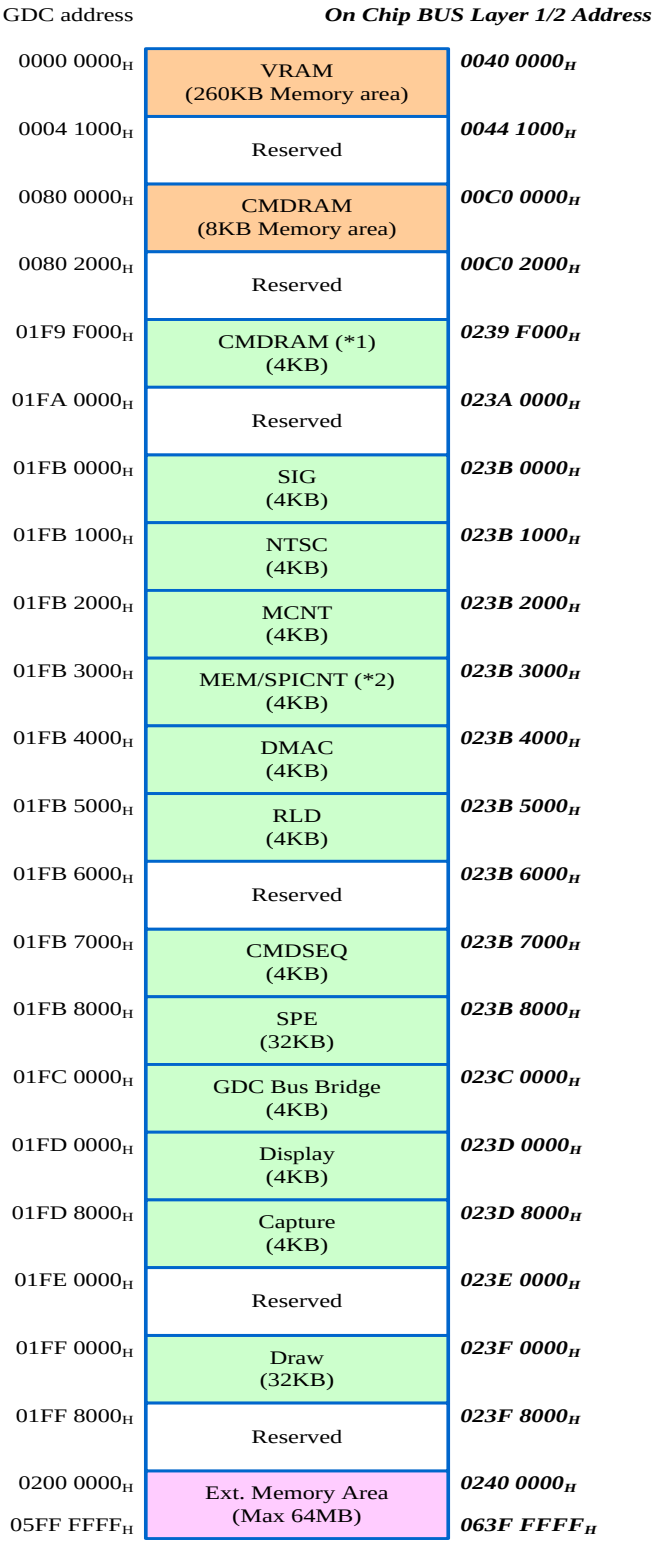
(*2) MEMC/SPICNT : The register contents vary depending on the selected external memory interface.

For details on selecting memory, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.

For details on the registers, see "1.15 Memory Controller (MEMC)" and "1.16 SPI Controller (SPICNT)" in this hardware manual.

1.4.2 CY91F591/6

Figure 1-4. Memory Map (CY91F591/6)



(*1) CMDRAM : If the ECC function is not used, CMDRAM becomes a reserved area.

For details on mode settings, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.

For details on functions, see "1.18 Command RAM (CMDRAM)" in this hardware manual.

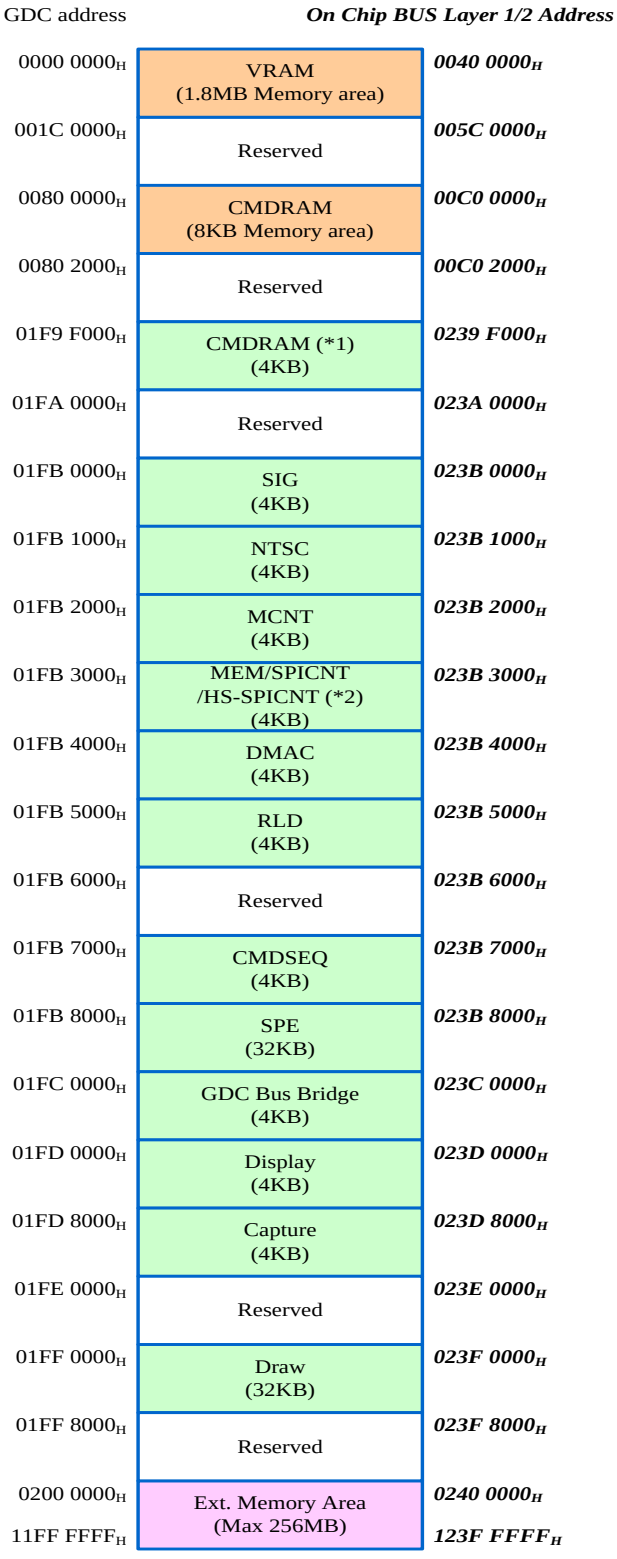
(*2) MEMC/SPICNT : The register contents vary depending on the selected external memory interface.

For details on selecting memory, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.

For details on the registers, see "1.15 Memory Controller (MEMC)" and "1.16 SPI Controller (SPICNT)" in this hardware manual.

1.4.3 CY91F59A/B

Figure 1-5. Memory Map (CY91F59A/B)



(*1) CMDRAM : If the ECC function is not used, CMDRAM becomes a reserved area.

For details on mode settings, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.

For details on functions, see "1.18 Command RAM (CMDRAM)" in this hardware manual.

(*2) MEMC/SPICNT/HS-SPICNT : The register contents vary depending on the selected external memory interface.

For details on selecting memory, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.

For details on the registers, see "1.15 Memory Controller (MEMC)", "1.16 SPI Controller (SPICNT)", and "1.22 Hi-speed SPI controller (HS-SPICNT)" in this hardware manual.

1.5 Explanation of GDC Macro Operation

1.5.1 Basic Settings

1.5.1.1 Startup Sequence

After the GDC startup sequence is processed, perform the operations described below in "GDC Access Sequence."

(For details on the GCC startup sequence, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.

GDC Access Sequence

Make the following required settings for using the GDC macro in periods where the GRST bit in the GDCCR register (*0000_0F65H*) is "1" (reset period).

- GDC macro operating mode settings

  Set the various operating modes of the GDC macro.

  For details on how to set them, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.

Immediately after "0" is written to the GRST bit in the GDCCR register (*0000_0F65H*) (when a reset is released), a command list is automatically executed[1] by a reset start.

Use a 16-bit device to automatically execute it through a reset start with NOR Flash/SRAM.

Note: It is not possible to automatically execute it through a reset start using HS-SPICNT installed on the CY91F59A/B and SPI Controller installed on the CY91F591/2/4/6/7/9.

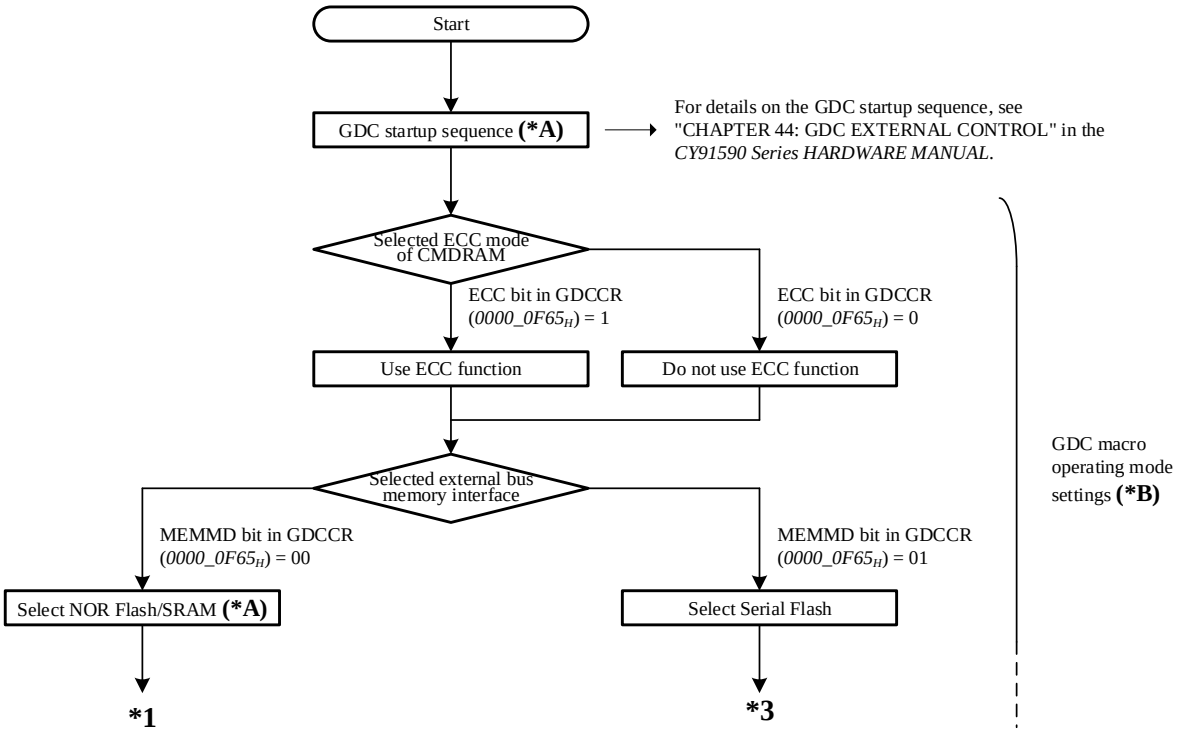
[1]: The BOOT bit setting in the GDCCR register (*0000_0F65H*) can be the setting of "whether to use/not use a reset start" for automatic command list execution.

Note: After "0" is written to the GRST bit in the GDCCR register (*0000_0F65H*) (after a reset is released) and an interrupt at the GDC bit in the IRPR6H register (*0000_0424H*) is confirmed, it is recommended to access the GDC.

The next sections describe the standard flowcharts of the GDC access sequence.

CY91F591/2/4/6/7/9

Figure 1-6. GDC Access Sequence (CY91F591/2/4/6/7/9)



(*A): Display digital output pins and external bus data input/output (external bus interface) pins are exclusively controlled at P016-011 of CY91590 series.

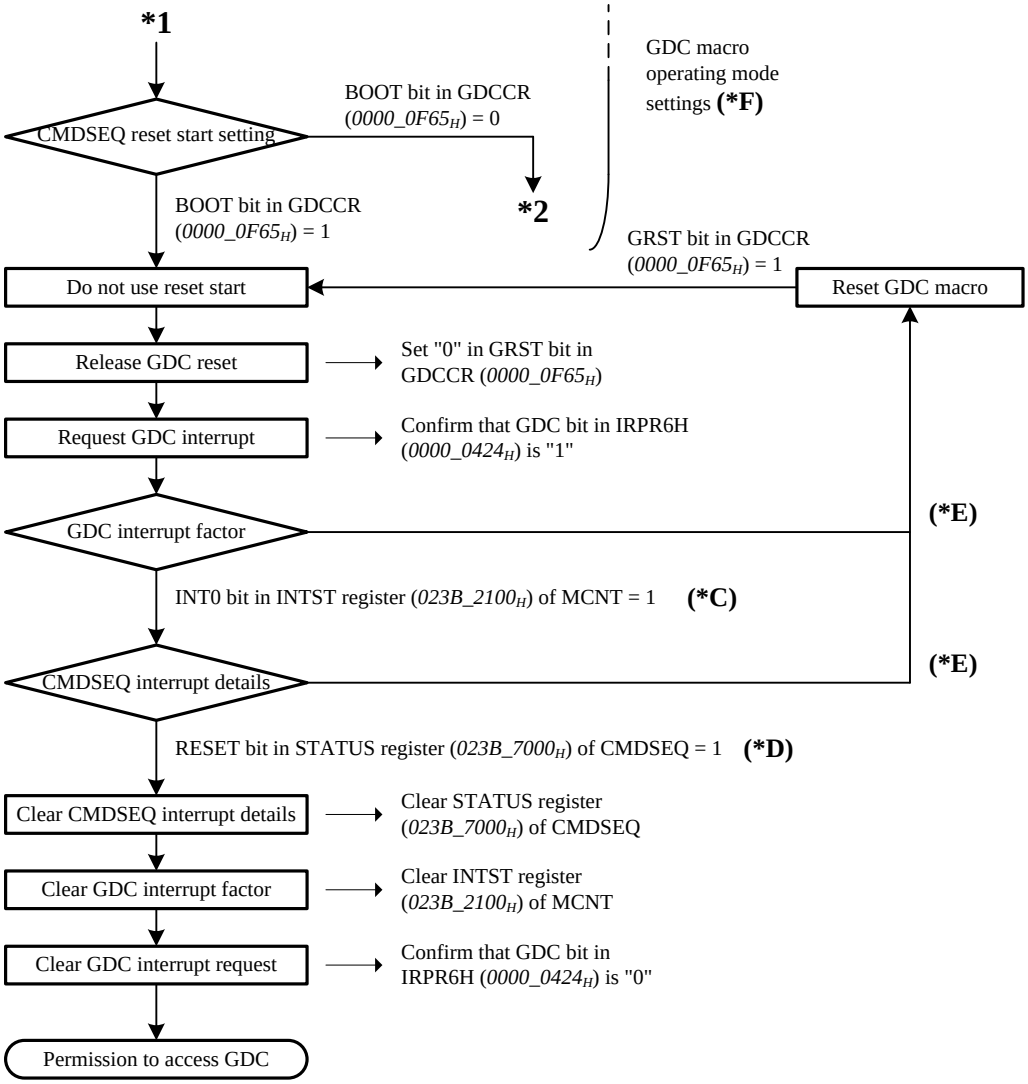
Using external bus interface and selecting NOR Flash/SRAM, note the following.

For display digital output 8-bit × 3, 8-bit NOR Flash/SRAM can be only used with external bus interface.

(*B): This is an example sequence of GDC macro operation mode setting.

Any sequence of setting between "Selected ECC mode of CMDRAM",

"Selected external bus memory interface", and "CMDSEQ reset start setting" can be applied.



(*C): If there is no NTSC input, the INT20 bit becomes "1". If NTSC input is not used, bypass this part.

(*D): If the CMDSEQ reset start is not used, the EXTNB bit becomes "1".

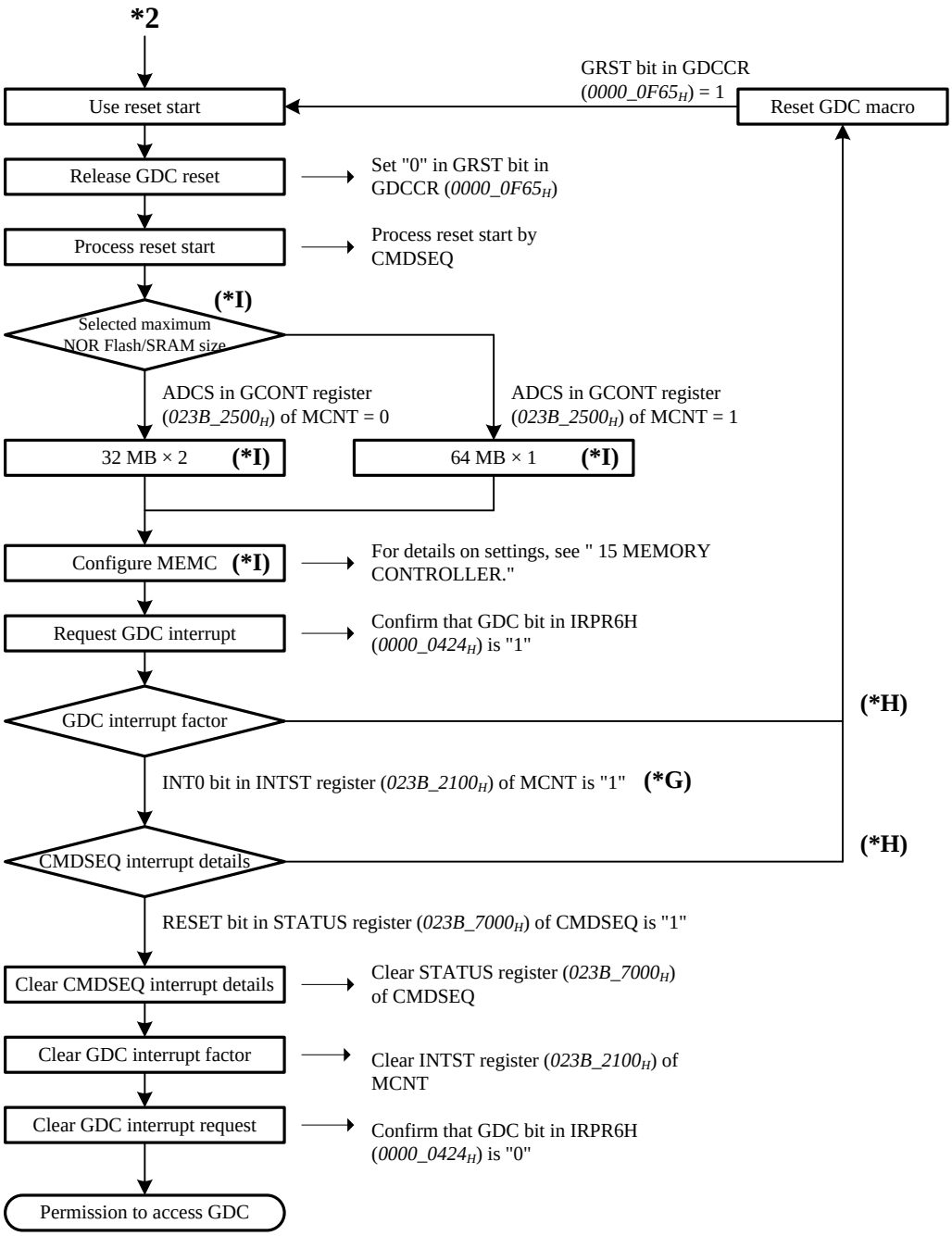
If the reset start is not used, bypass this part.

(*E): If the CMDSEQ reset start is not used, no interrupt with a normal end is generated except at (*A) and (*B) above. If any other interrupt is generated, the state is abnormal, so reset the GDC macro (GRST bit in the GDCCR register).

(*F): This is an example sequence of GDC macro operation mode setting.

Any sequence of setting between "Selected ECC mode of CMDRAM",

"Selected external bus memory interface", and "CMDSEQ reset start setting" can be applied.

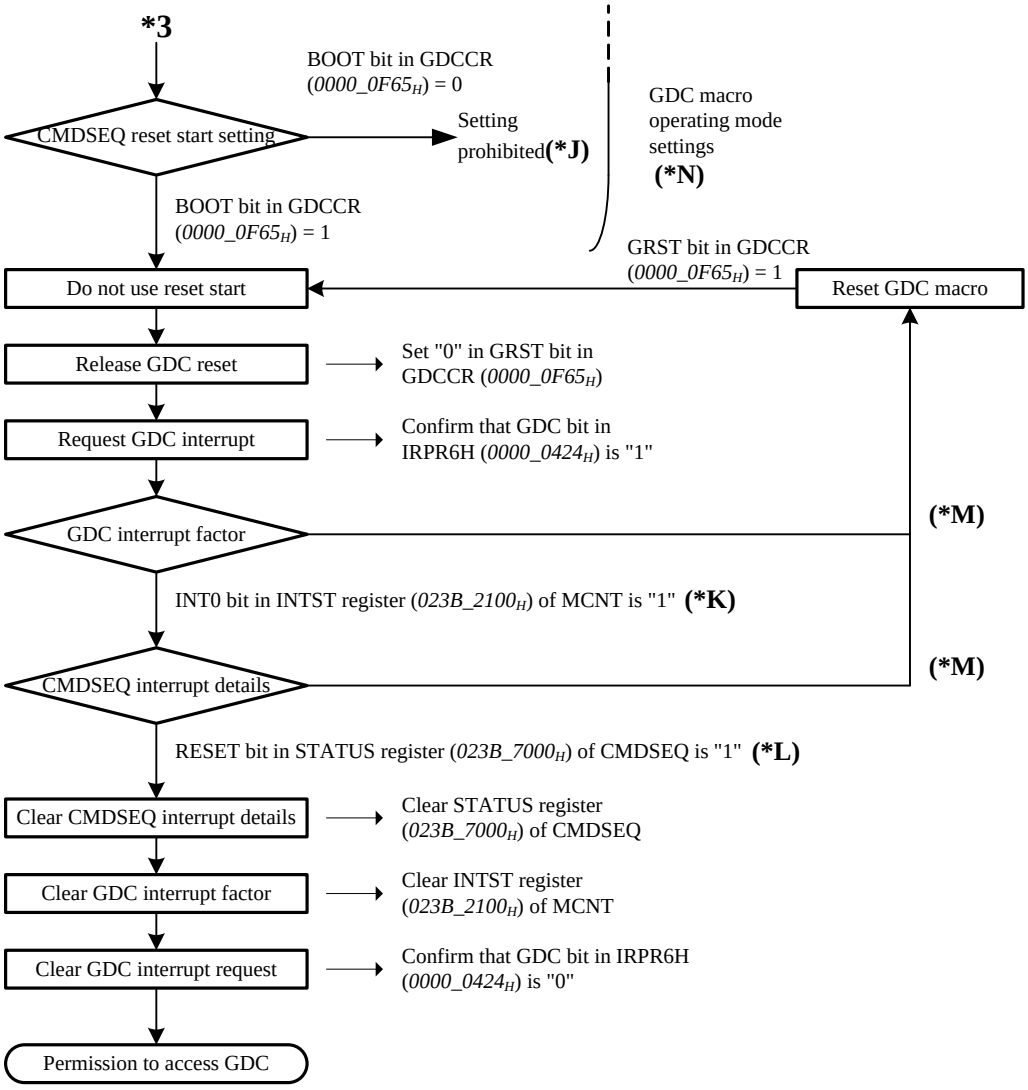


(*G): If there is no NTSC input, the INT20 bit becomes "1". If NTSC input is not used, bypass this part.

(*H): If the CMDSEQ reset start is used, the interrupt that is generated varies depending on the contents of the command list used with the reset start. If an error interrupt is generated, the state is abnormal, so reset the GDC (GRST bit in the GDCCR register).

(*I): Execute CMDLIST to configure MCNT and MEMC.

(For details, see "Command List Arrangement for a Reset Start") Be sure to use a 16-bit device when using NOR Flash/SRAM in a reset start.



(*J): If SPI controller is used, CMDSEQ reset start cannot be used.

If the reset start is used, the operation of the GDC macro is not guaranteed.

(*K): If there is no NTSC input, the INT20 bit becomes "1". If NTSC input is not used, bypass this part.

(*L): If the CMDSEQ reset start is not used, the EXTNB bit becomes "1".

If the reset start is not used, bypass this part.

(*M): If the CMDSEQ reset start is not used, no interrupt with a normal end is generated except at (*A) and (*B) above. If any other interrupt is generated, the state is abnormal, so reset the GDC macro (GRST bit in the GDCCR register).

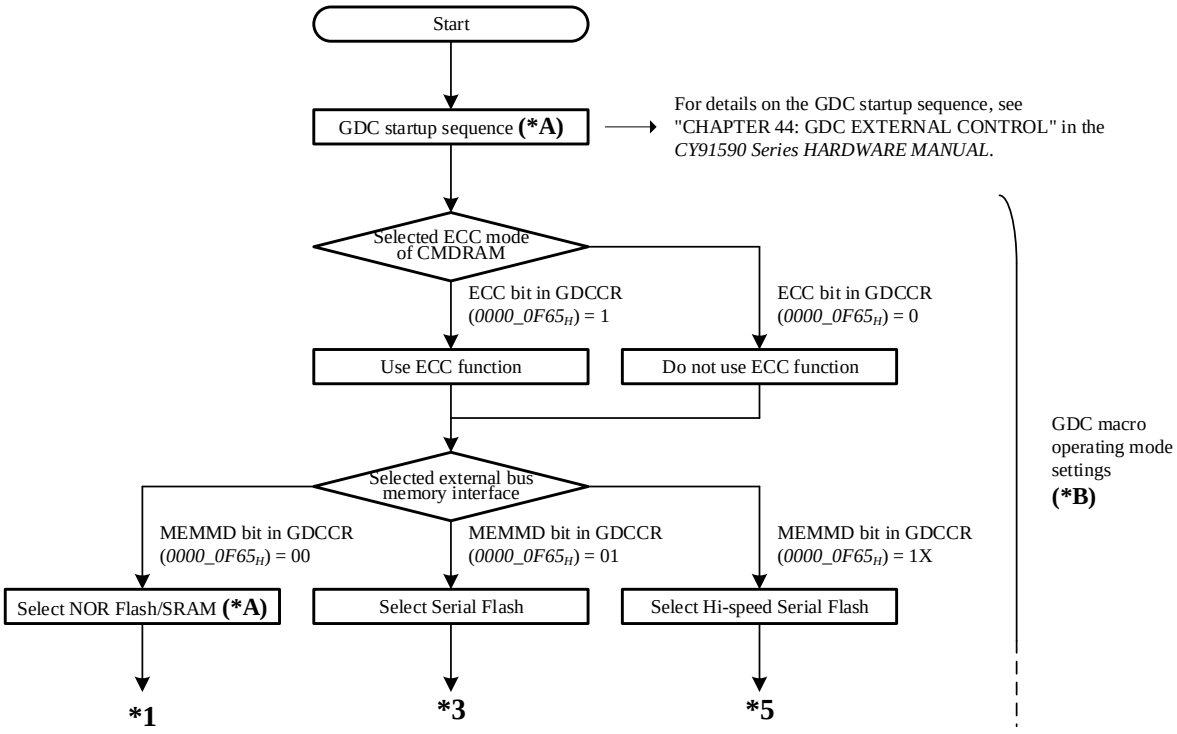
(*N): This is an example sequence of GDC macro operation mode setting.

Any sequence of setting between "Selected ECC mode of CMDRAM",

"Selected external bus memory interface", and "CMDSEQ reset start setting" can be applied.

CY91F59A/B

Figure 1-7. GDC Access Sequence (CY91F59A/B)



(*A): Display digital output pins and external bus data input/output (external bus interface) pins are exclusively controlled at P016-011 of CY91590 series.

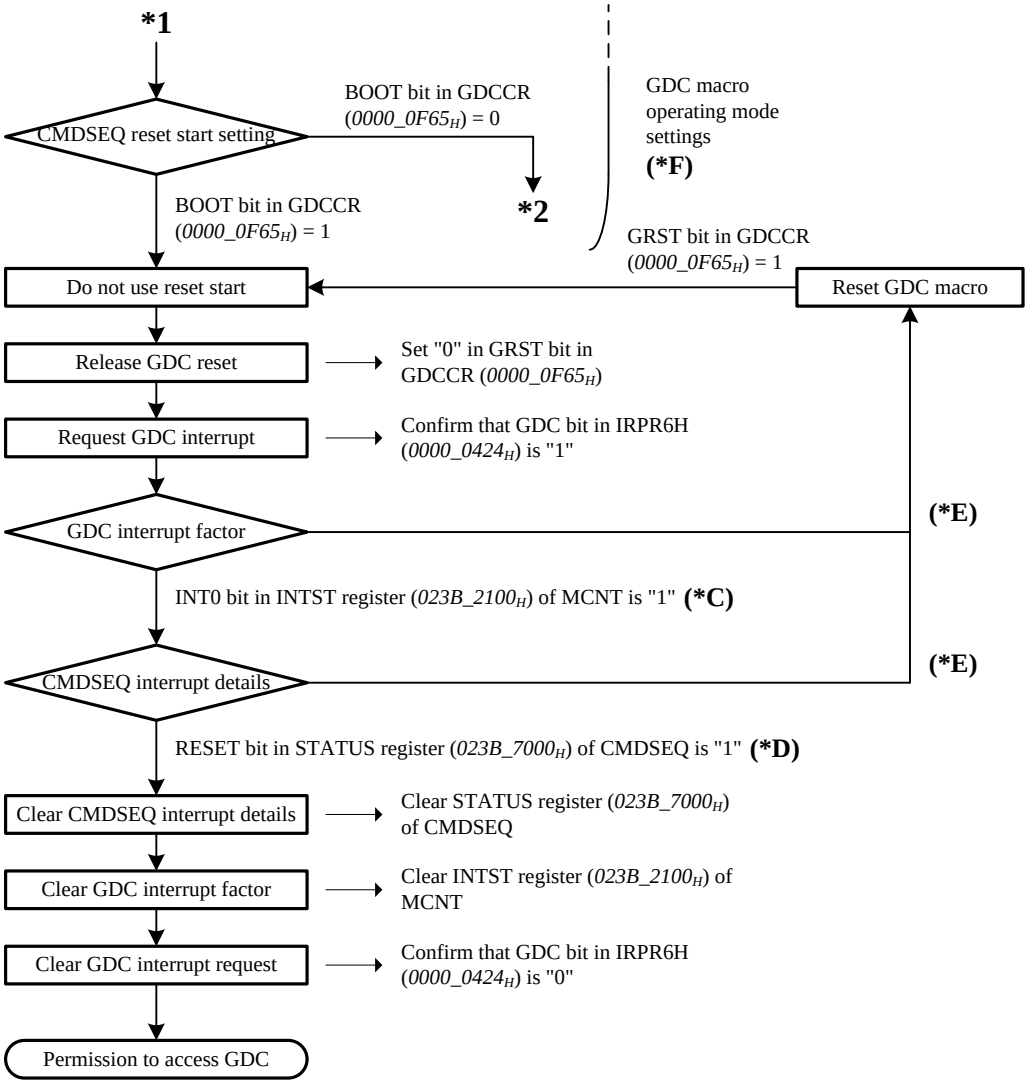
Using external bus interface and selecting NOR Flash/SRAM, note the following.

For display digital output 8-bit × 3, 8-bit NOR Flash/SRAM can be only used with external bus interface.

(*B): This is an example sequence of GDC macro operation mode setting.

Any sequence of setting between "Selected ECC mode of CMDRAM",

"Selected external bus memory interface", and "CMDSEQ reset start setting" can be applied.



(*C): If there is no NTSC input, the INT20 bit becomes "1". If NTSC input is not used, bypass this part.

(*D): If the CMDSEQ reset start is not used, the EXTNB bit becomes "1".

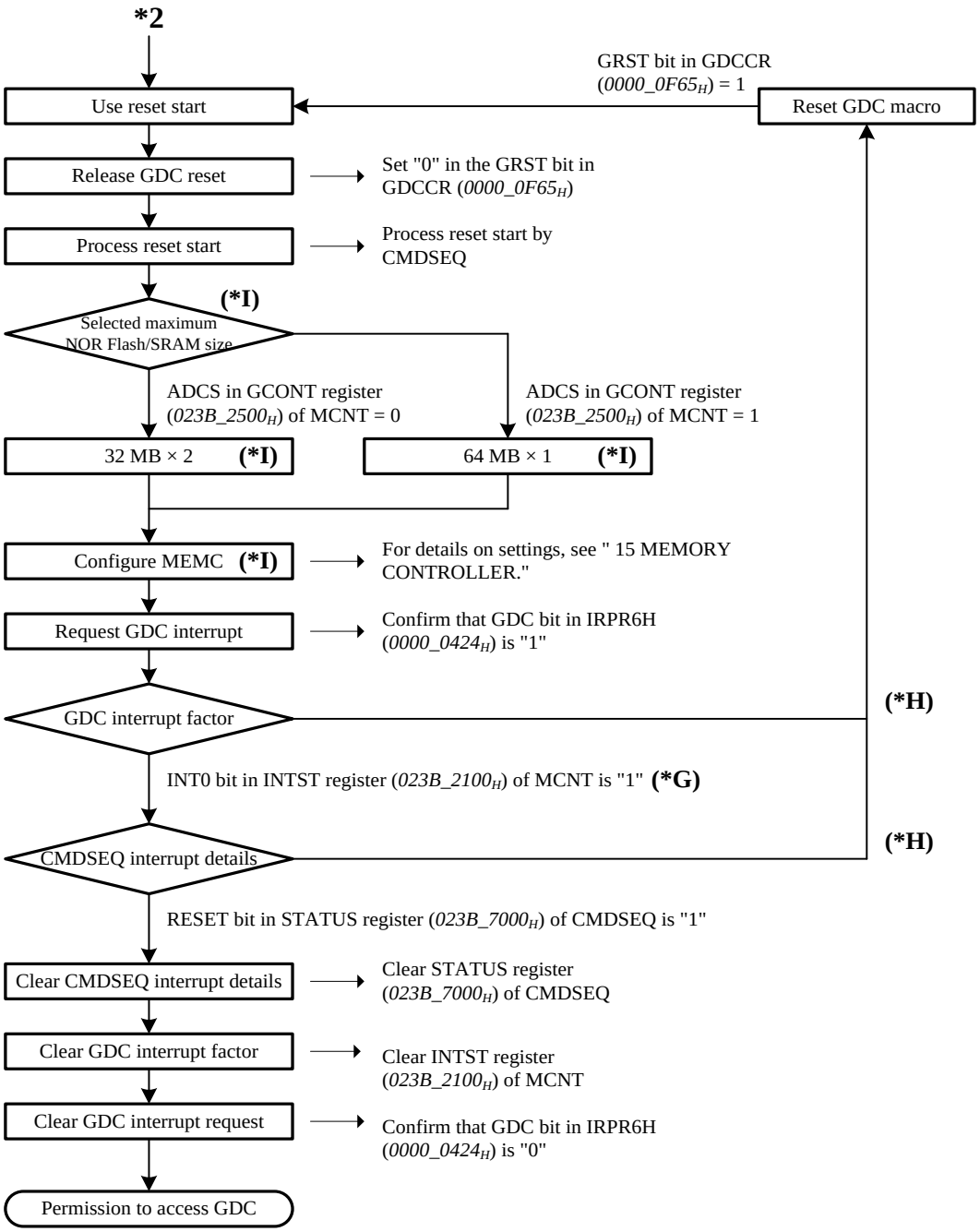
If the reset start is not used, bypass this part.

(*E): If the CMDSEQ reset start is not used, no interrupt with a normal end is generated except at (*A) and (*B) above. If any other interrupt is generated, the state is abnormal, so reset the GDC macro (GRST bit in the GDCCR register).

(*F): This is an example sequence of GDC macro operation mode setting.

Any sequence of setting between "Selected ECC mode of CMDRAM",

"Selected external bus memory interface", and "CMDSEQ reset start setting" can be applied.

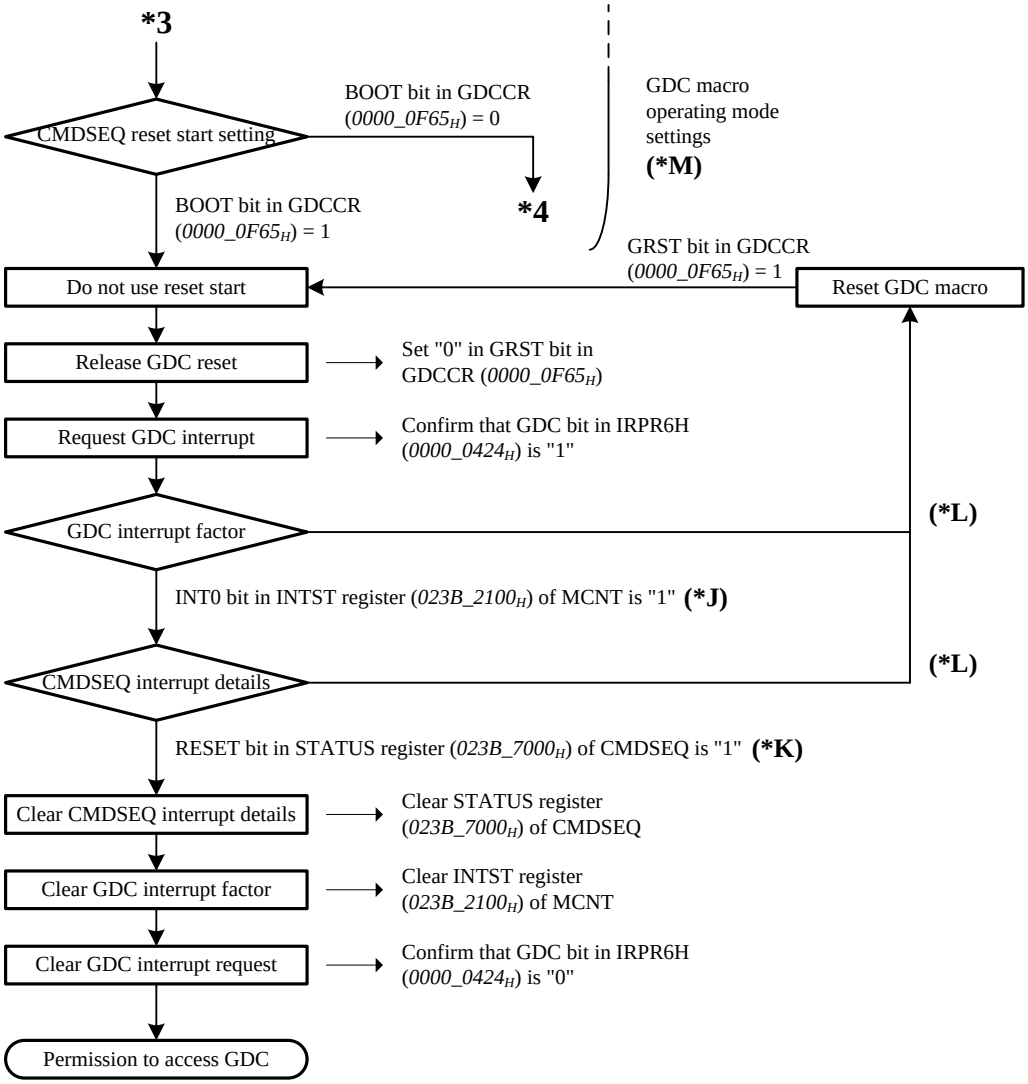


(*G): If there is no NTSC input, the INT20 bit becomes "1". If NTSC input is not used, bypass this part.

(*H): If the CMDSEQ reset start is used, the interrupt that is generated varies depending on the contents of the command list used with the reset start. If an error interrupt is generated, the state is abnormal, so reset the GDC (GRST bit in the GDCCR register).

(*I): Execute CMDLIST to configure MCNT and MEMC.

(For details, see "Command List Arrangement for a Reset Start") Be sure to use a 16-bit device when using NOR Flash/SRAM in a reset start.



(*J): If there is no NTSC input, the INT20 bit becomes "1". If NTSC input is not used, bypass this part.

(*K): If the CMDSEQ reset start is not used, the EXTNB bit becomes "1".

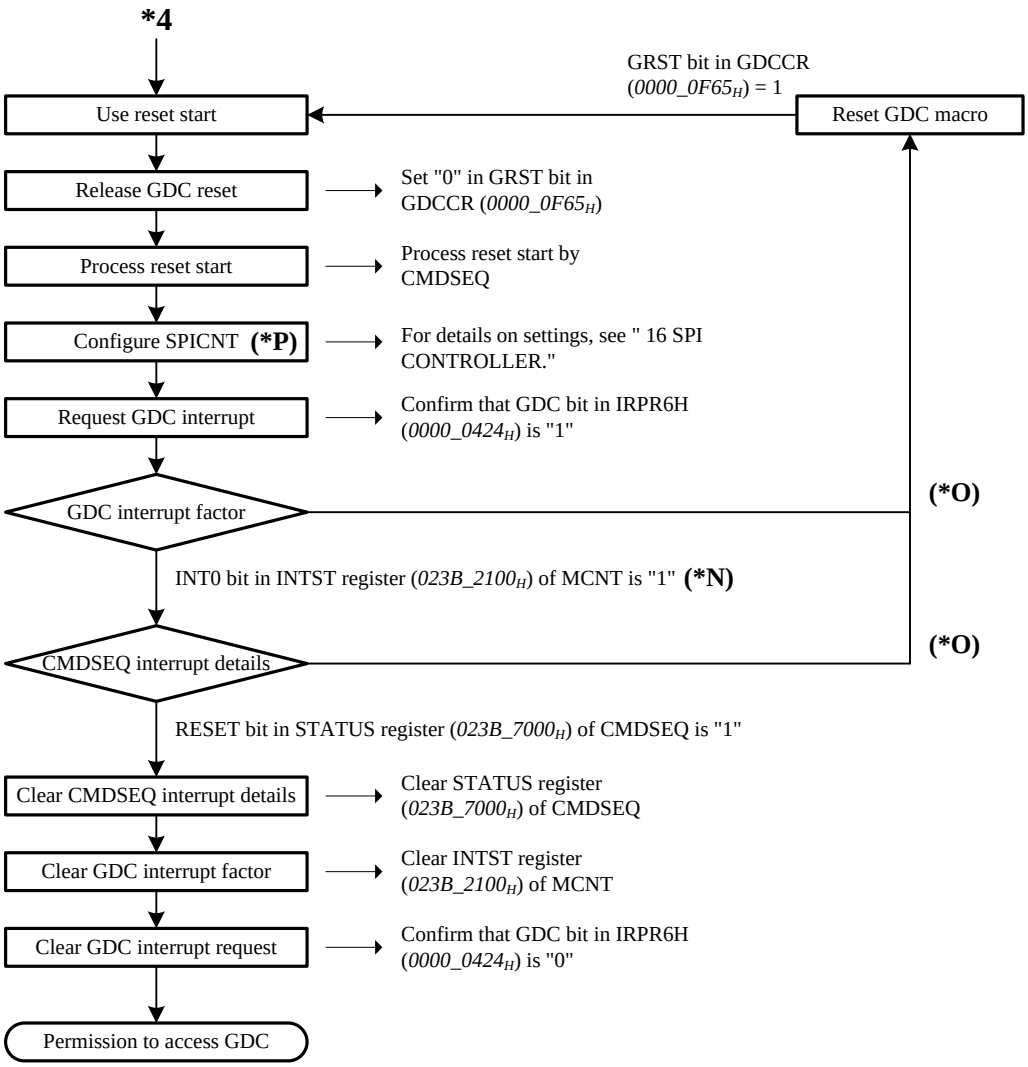
If the reset start is not used, bypass this part.

(*L): If the CMDSEQ reset start is not used, no interrupt with a normal end is generated except at (*A) and (*B) above. If any other interrupt is generated, the state is abnormal, so reset the GDC macro (GRST bit in the GDCCR register).

(*M): This is an example sequence of GDC macro operation mode setting.

Any sequence of setting between "Selected ECC mode of CMDRAM",

"Selected external bus memory interface", and "CMDSEQ reset start setting" can be applied.

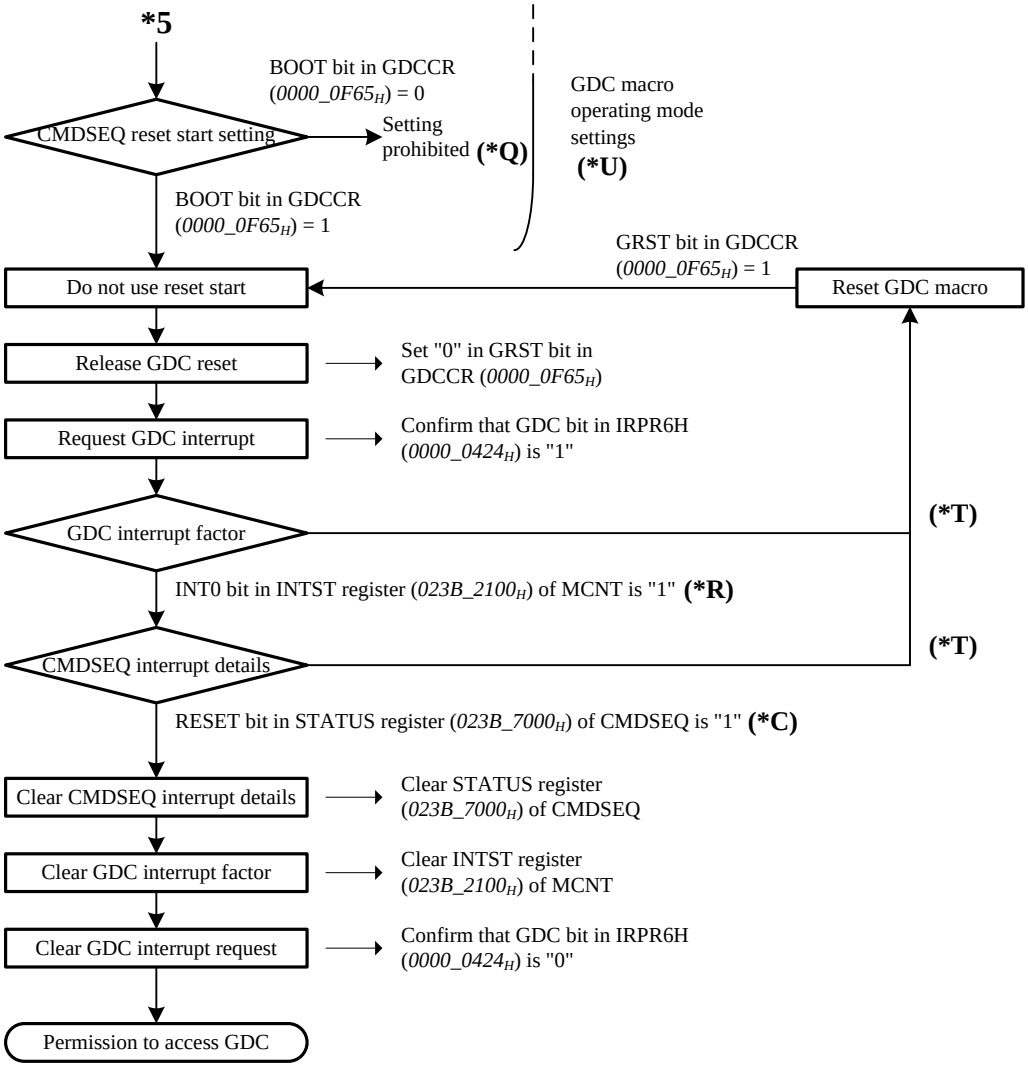


(*N): If there is not NTSC input, the INT20 bit becomes "1". If NTSC input is not used, bypass this part.

(*O): If the CMDSEQ reset start is used, the interrupt that is generated varies depending on the contents of the command list used with the reset start. If an error interrupt is generated, the state is abnormal, so reset the GDC (GRST bit in the GDCCR register).

(*P): Execute CMDLIST to configure SPICNT.

(For details, see "Command List Arrangement for a Reset Start")



(*Q): If HS-SPICNT is used, the CMDSEQ reset start cannot be used.

If the reset start is used, the operation of the GDC macro is not guaranteed.

(*R): If there is no NTSC input, the INT20 bit becomes "1". If NTSC input is not used, bypass this part.

(*S): If the CMDSEQ reset start is not used, the EXTNB bit becomes "1".

If the reset start is not used, bypass this part.

(*T): If the CMDSEQ reset start is not used, no interrupt with a normal end is generated except at(*A) and (*B) above. If any other interrupt is generated, the state is abnormal, so reset the GDC macro (GRST bit in the GDCCR register).

(*U): This is an example sequence of GDC macro operation mode setting.

Any sequence of setting between "Selected ECC mode of CMDRAM",

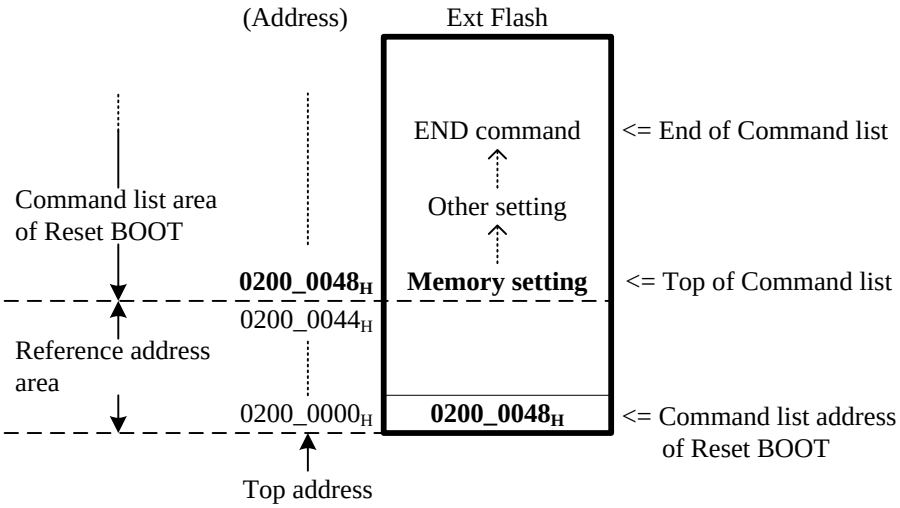
"Selected external bus memory interface", and "CMDSEQ reset start setting" can be applied.

Command List Arrangement for a Reset Start

In the GDC macro, a reset start can automatically execute a command list, but the command list must be located at the specified address in external Flash.

For details on the location in external Flash, see the Figure 1-8.

Figure 1-8. Command List Arrangement



Restrictions:

- The command list for a reset start must have the Memory Controller configured for use as the initial command process at "Top of Command list (Memory setting)" shown in Figure 1-8.
- Unless the Memory Controller is first configured for use, the command list may not work normally.
- Use a 16-bit device for automatic execution by a reset start with NOR Flash/SRAM.
- Automatic execution by a reset start using HS-SPICNT installed on the CY91F59A/B and SPI Controller installed on the CY91F591/2/4/6/7/9 are not available.

1.5.1.2 Interrupts

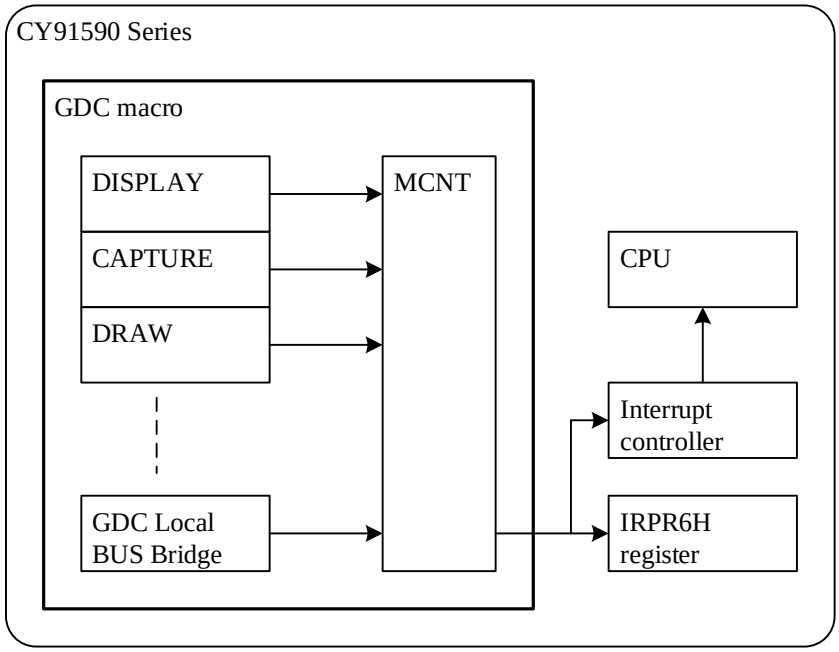
Overview

As the GDC macro interrupts, interrupts from each module in the GDC macro are logically added in MCNT temporarily, and output to the IRPR6H register and the interrupt controller.

For details on the GDC macro interrupt architecture, see the following figure.

(For a list of interrupts in the GDC macro, see "Input Interrupt Signal".)

Figure 1-9. GDC Macro Interrupt Architecture



Interrupt Setting Procedure

The interrupt setting procedure is as follows:

1. After performing the operations in "GDC Access Sequence", configure the interrupts to be logically added and output to the IRPR6H register and the interrupt controller in the INTEN register (see "INTEN Register") of MCNT.
2. Configure the interrupt output of each module.[1]

[1]: The following modules support output settings for interrupts to MCNT.

CY91F591/2/4/6/7/9	:	CMDSEQ, Capture, SIG, DMAC, RLD, CMDRAM, MEMC, SPICNT
CY91F59A/B	:	CMDSEQ, Capture, SIG, DMAC, RLD, CMDRAM, MEMC, SPICNT, HS-SPICNT

For modules other than the above, the occurrence of an interrupt factor causes output to MCNT.

Interrupt Clearing Procedure

There are 3 procedures for clearing GDC macro interrupts.

The following describe each procedure.

Clearing Procedure for Edge Detection

The following procedure clears an interrupt in edge detection mode.

(For a list of interrupt factors, see "Input Interrupt Signal").

1. Check the interrupt factor that occurred in the INTST register of MCNT.
2. Write "1" in the bit in the INTST register of MCNT to clear the interrupt factor.
3. Check the interrupt factor that occurred in the status register of the module where the interrupt was detected.
4. Clear the status register of the module where the interrupt was detected.

The above steps 3 and 4 are not required for modules that do not have a status register.

Clearing Procedure for Level Detection

The following procedure clears an interrupt in level detection mode.

(For a list of interrupt factors, see "Input Interrupt Signal").

1. Check the interrupt factor that occurred in the INTST register of MCNT.
2. Check the interrupt factor that occurred in the status register of the module where the interrupt was detected.
3. Clear the status register of the module where the interrupt was detected.
4. Write "1" in the bit in the INTST register of MCNT to clear the interrupt factor.

Interrupt Clearing Procedure as Required for a GDC Macro Reset

The following procedure clears an interrupt that requires a GDC macro reset (GRST bit in the GDCCR register).

(For a list of interrupt factors, see "Input Interrupt Signal").

1. Check the interrupt factor that occurred in the INTST register of MCNT.
2. Execute a GDC macro reset.
   (For details on a GDC macro reset, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.)
3. Release the GDC macro reset.
   (For details on a GDC macro reset, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.)
4. Perform the operations from "Request GDC interrupt" in "Figure 1-6" in "CY91F591/2/4/6/7/9" or in "Figure 1-7" in "CY91F59A/B".
5. Perform the operations in "Interrupt Setting Procedure".

1.5.1.3 Bus Access

The GDC macro does not support the master function and operates as a slave.

Endian

The GDC macro supports Little Endian.

All of these written macro specifications are for Little Endian.

Note: Write data in Little Endian on devices connected to the external bus memory interfaces (for data used in the GDC macro).

Access Size

Register access to the GDC macro is limited to 4-byte access (word).

The following is prohibited: 2-byte access (half word) and 1-byte access (byte).

Note that only the following functions change the access size.

NOR Flash Writing

The following access sizes apply to NOR Flash writing.

- 16-bit NOR Flash
  Only 2-byte access is supported.
- 8-bit NOR Flash
  Only 1-byte access is supported.

Data cannot be written correctly to NOR Flash unless it is written in a prescribed size.

Note: Data can be read normally to NOR Flash in reading with any access size.

Direct Mode of the Hi-Speed SPI Controller (CY91F59A/B)

Follow the HSSPIn_FIFOCFG register settings for access to the HSSPIn_TXFIFO register and the HSSPIn_RXFIFO register in the direct mode of the Hi-Speed SPI Controller.

For details, see "1.22 Hi-speed SPI controller (HS-SPICNT)".

Lock Transfer of the AHB Bus

The GDC macro does not support Lock transfer.

If Lock transfer to the GDC macro is executed, operation cannot be guaranteed.

Address Boundary

Burst access exceeding the 1KB address boundary is not possible for the GDC macro.

1.5.1.4 External Bus Memory Interfaces

Maximum Size of GDC External Memory

The maximum size of memory that can be connected to an external bus memory interface is as described below.

Set 0200_0000_H for the CS0 address area setting of MEMC.

- When SPICNT is selected
  Maximum of 16MB
- When MEMC is selected and "CS1" is selected in the GCONT register of the MCNT module
  Maximum of 32MB x 2
- When MEMC is selected and "Address25" is selected in the GCONT register of the MCNT module
  Maximum of 64MB x 1

  Note:

  When selecting Address25, set the same value for both CS0 and CS1, except in the MFCAREA 0/1 register of MEMC.

  1. Set addresses from 0200_0000H to 03FF_FFFFH for the CS0 address area setting.
  2. Set addresses from 0400_0000H to 05FF_FFFFH for the CS1 address area setting.

  For details, see "1.15 Memory Controller (MEMC)".
- When HS-SPICNT is selected (CY91F59A/B)
  Maximum of 256MB

External Bus Memory Interface Mode Setting

The external bus memory interfaces built into the GDC macro are as follows.

- CY91F591/2/4/6/7/9: MEMC and SPICNT
- CY91F59A/B: MEMC, SPICNT, and HS-SPICNT

Each interface operates exclusively according to the set values shown in "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.

Restrictions:

- To switch an external bus memory interface, do so during a GDC macro reset. (For details, see "1.5.1.1 Startup Sequence.")
- If the external bus memory interface is not switched not during the reset, the operation of the external bus memory interface cannot be guaranteed.

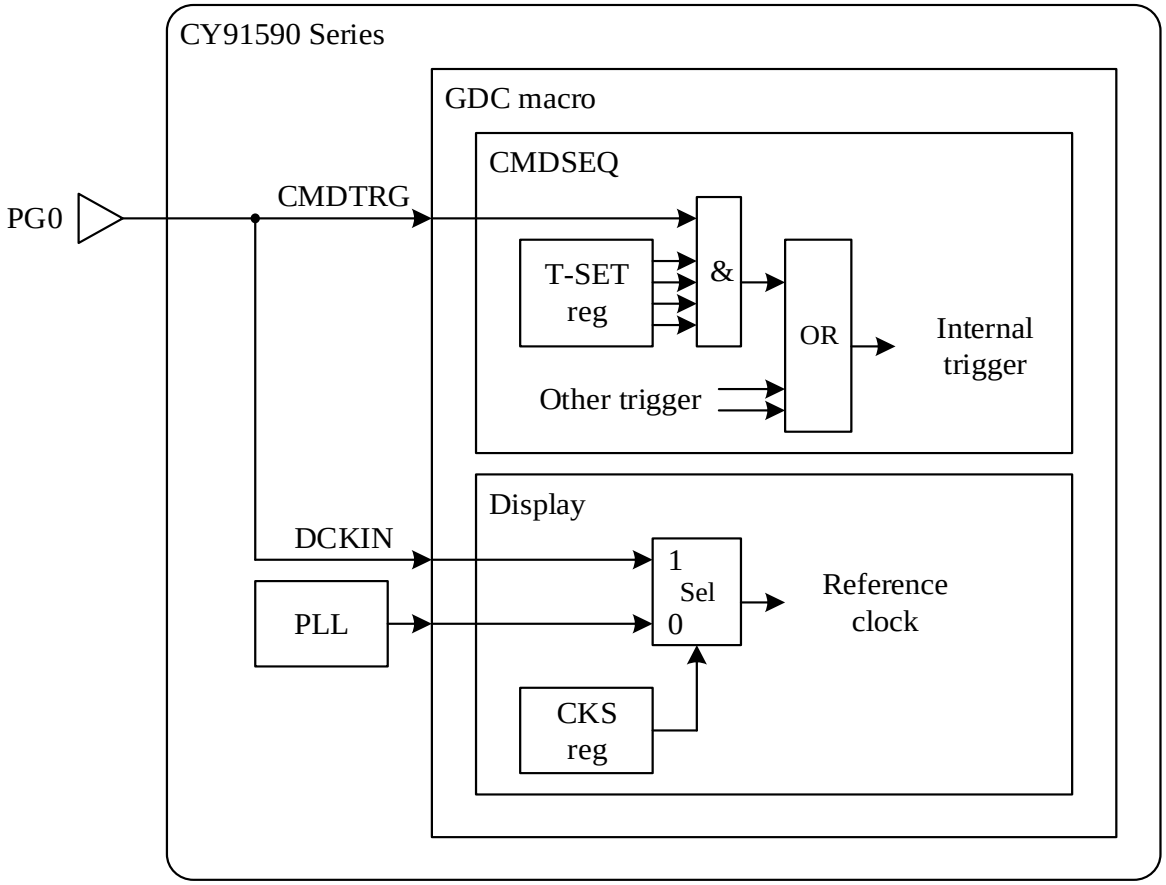
1.5.1.5 Notes on Use

Shared External Pin

The following figure shows that the PG0 pin (pin number: 200) of the CY91F590 series is used for both reference clock input (DCLKIN) by Display and trigger input (CMDTRG) by CMDSEQ to start a trigger. Therefore, either of these uses requires some caution as described below.

- The pin is used as the reference clock of Display.
  1. Set a value other than F_H in the T-SET register (bit19 to bit16 in 01FB_7018$_H$) of CMDSEQ.
  2. Set 1_H in the CKS register (bit15 in 01FD_0000$_H$) of Display.
- The pin is used as the trigger input of CMDSEQ.
  1. Set F_H in the T-SET register (bit19 to bit16 in 01FB_7018$_{H)}$ of CMDSEQ.
  2. Set 0_H in the CKS register (bit15 in 01FD_0000$_H$) of Display.

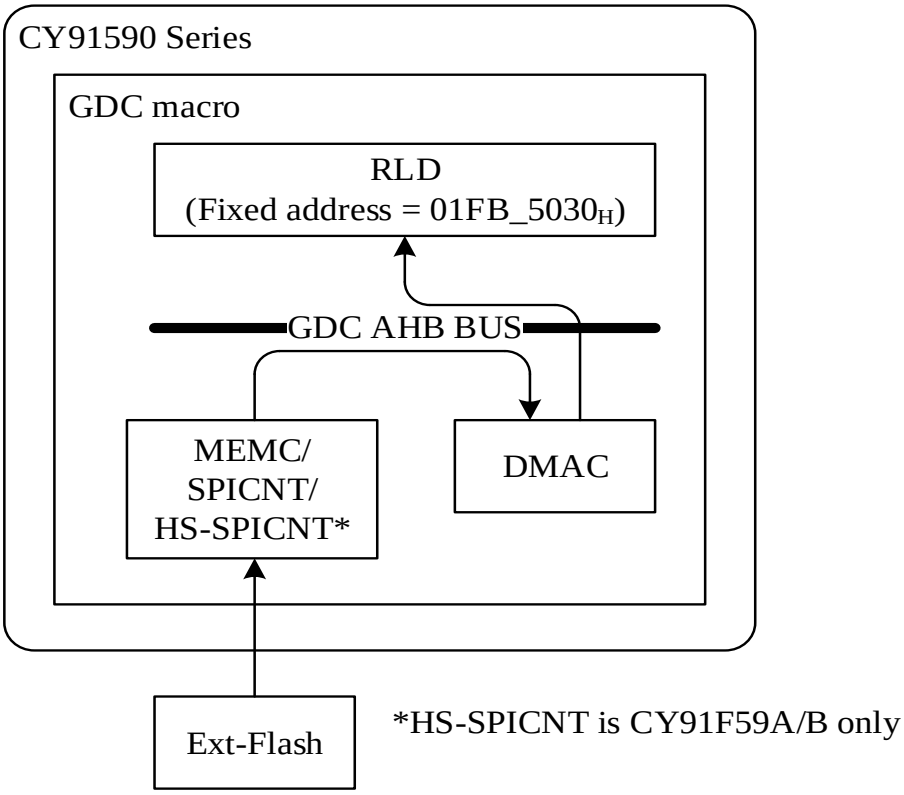
Figure 1-10. Shared External Pin



Data Transfer to Run-Length Decompression

RLD needs to have data transferred using DMAC because of a requirement to input compressed data to a static address. (For details, see "1.12 Run-Length Decompression (RLD)".)

Figure 1-11. Data Transfer to RLD



Note:

Use of SPICNT requires some caution as described below.

The "master transfer mode" installed in SPICNT cannot be used for data transfer to RLD because it does not support static addressing.

Therefore, even data transfer from SPICNT needs to have data transferred using DMAC as shown above.

Port Exclusive control

At P016-P011 of CY91F590 series, display digital output pins and external bus data input/output pins are controlled exclusively.

Depending on the setting for pins or selection of external memory interface, some of specification is limited for example that CMDSEQ reset start cannot be used. See Table 1-1 and Table 1-2.

For details, see the CY91590 Series Hardware Manual.

Table 1-1. Pin/Function Combination List (CY91F591/2/4/6/7/9)

External pin	FR81S(GDCCR Register Setting)		Specification Limited for GDC Mode	
P016-P011	MEM[1:0] (External bus memory interface selection)	CMDBOOT (CMDSEQ Reset BOOT mode)	External bus memory interface	CMDSEQ Reset BOOT mode
When selecting Display digital output pin	00[1]	0[3]	Limited[5]	Limited[6]
		1[4]		-
	01[2]	0[3]	-	Limited[7]
		1[4]		-
When selecting external bus memory input/output pin	00[1]	0[3]	-	Limited[8]
		1[4]		-
	01[2]	0[3]	-	Limited[7]
		1[4]		-

[1]: MEMC is selected as external bus memory interface.

[2]: SPICNT is selected as external bus memory interface.

[3]: CMDSEQ reset start is used.

[4]: CMDSEQ reset start is not used.

[5]: Only 8-bit can be used in MEMC when P016-P011 pins are used as display digital output because they are not used as external bus data input/output.

[6]: CMDSEQ reset start cannot be used when P016-P011 pins are used as display digital output and as external bus memory interface of MEMC.

[7]: CMDSEQ reset start cannot be used when SPICNT is selected as external bus memory interface.

[8]: Use 16-bit devise when P016-P011 pins are used as external bus memory interface by reset start.

Table 1-2. Pin/Function Combination List (CY91F59A/B)

External Pin	FR81S(GDCCR Register Setting)		Specification Limited for GDC Mode	
P016-P011	MEM[1:0] (External bus memory interface selection)	CMDBOOT (CMDSEQ Reset BOOT mode)	External bus memory interface	CMDSEQ Reset BOOT mode
When selecting Display digital output pin	00[1]	0[4]	Limited[6]	Limited[7]
		1[5]		-
	01[2]	0[4]	-	-
		1[5]		-
	1X[3]	0[4]	-	Limited[8]
		1[5]		-
When selecting external bus memory input/output pin	00[1]	0[4]	-	Limited[9]
		1[5]		-
	01[2]	0[4]	-	-
		1[5]		-
	1X[3]	0[4]	-	Limited[8]
		1[5]		-

[1]: MEMC is selected as external bus memory interface.

[2]: SPICNT is selected as external bus memory interface.

[3]: HS-SPICNT is selected as external bus memory interface.

[4]: CMDSEQ reset start is used.

[5]: CMDSEQ reset start is not used.

[6]: Only 8-bit can be used in MEMC when P016-P011 pins are used as display digital output because they are not used as external bus data input/output.

[7]: CMDSEQ reset start cannot be used when P016-P011 pins are used as display digital output and as external bus memory interface of MEMC.

[8]: CMDSEQ reset start cannot be used when SPICNT is selected as external bus memory interface.

[9]: Use 16-bit devise when P016-P011 pins are used as external bus memory interface by reset start.

1.6 Display Controller (DISPLAY)

1.6.1 Overview

The Display Controller (Display) is a controller for displaying image data with a layer structure in VRAM on a display device such as an LCD panel.

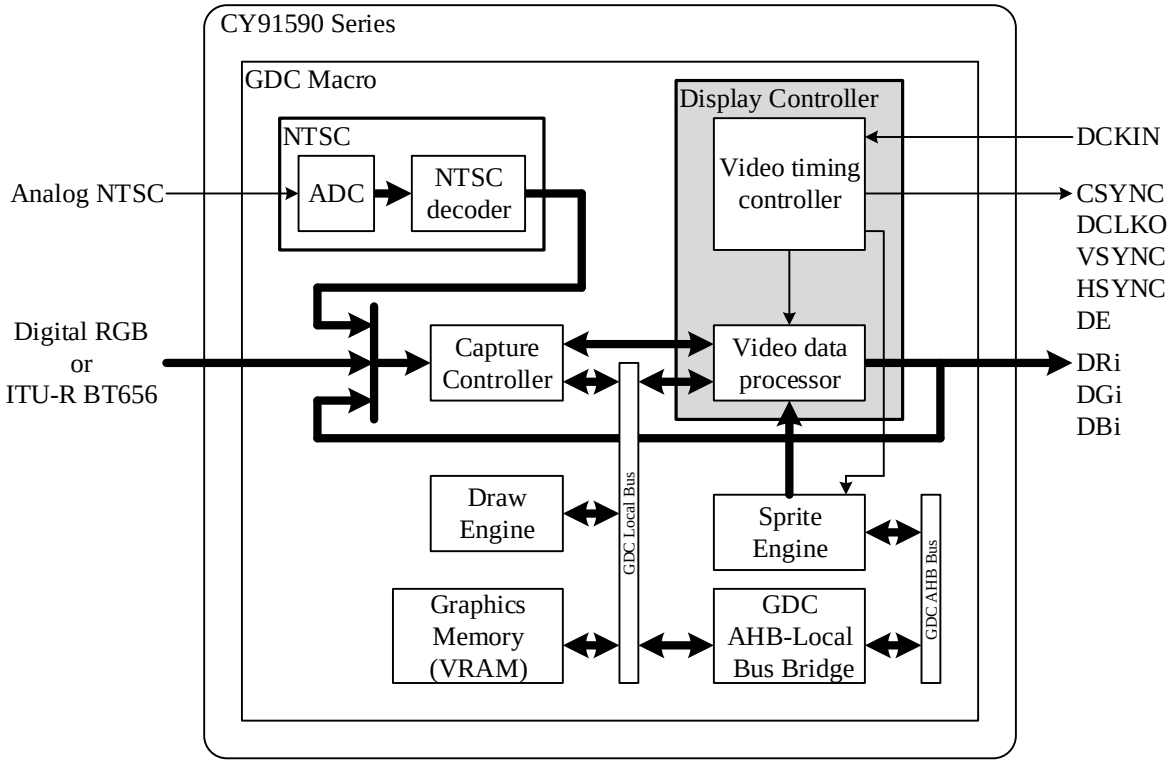
1.6.2 Features

- Display control
  The controller can display a window with 5 layers, including 1 sprite-dedicated layer. Screen scrolling and other processes are also possible.
- Downward compatibility
  Compatibility with the 4 higher layers in Cypress's LSI MB86276 is maintained. The lower layers are not supported.
- Dual display using a single display controller
  A single display controller can output 2 different screens.
  Arbitrary layers can be selectively output to the 2 screens.
- Video timing generation circuit
  The generated video timing corresponds to screen resolutions from 320 x 240 to 800 x 480.
- Color look-up
  For the frame display of indirect color mode (8 bits/pixel), 3 built-in color look-up tables using palette RAM are used.

1.6.3 Configuration

1.6.3.1 Positioning in the Overall GDC Macro Configuration

Figure 1-12. Display Block Diagram



Notes:

- Analog NTSC corresponds to the external pin VIN.
- Digital RGB corresponds to the external pins PA2 to PA7, PB2 to PB7, and PC2 to PC7.
- ITU-R.BT656 corresponds to the external pins PA2 to PA7, PB2, and PB3.
- DCLKI corresponds to the external pin DCKIN.
- CSYNC corresponds to the external pin PG3.
- DCLKO corresponds to the external pin PG4.
- VSYNC corresponds to the external pin PG5.
- HSYNC corresponds to the external pin PG6.
- DE corresponds to the external pin PG7.
- DR*i* corresponds to the external pins P011, P012, and PD2 to PD7.
- DG*i* corresponds to the external pins P013, P014, and PE2 to PE7.
- DB*i* corresponds to the external pins P015, P016, and PF2 to PF7.

1.6.4 Registers

1.6.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (0040_0000_H) is added for access from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.

  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  RX: The read value is always undefined.
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.6.4.2 Register List

DisplayBaseAddress = 01FD_0000$_H$

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x7ff8	VCCC (Video/Capture Common Control)																															
																			Csel											C0sr		V0sr
0x7ffc	VCSR (Video/Capture Soft Reset)																															

DisplayBaseAddress = 01FD_0000$_H$

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x000	DCM0 (Display Control Mode 0)																															
	DEN												L4E	L23E	L1E	L0E	CKS	LCS		SC					EEQ	SCSY			SF	ESY	SYNC	
0x100	DCM1 (Display Control Mode 1)																															
	DEN											L4E	L3E	L2E	L1E	L0E	CKS	LCS	SC					EEQ	SCSY			SF	ESY	SYNC	
0x104	DCM2 (Display Control Mode 2)																															
																															RUF	RUM0
0x108	DCM3 (Display Control Mode 3)																															
						SELLAT					RGBrv	RGBsh											DCKed	DCKinv				DCKD				
0x004					HTP (H Total Pixels)																											
0x008					HDB (H Display Boundary)																HDP (H Display Period)											
0x00C	VSWH		VSW						HSW												HSP (H Sync pulse Position)											
0x010					VTR (V Total Rasters)																											
0x014					VDP (V Display Period)																VSP (V Sync pulse Position)											
0x018					L1WY (L1 Window Y)																L1WX (L1 Window X)											
0x01C					L1WH (L1 Window Height)																L1WW (L1 Window Width)											

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x020	L0M (L0 Mode)																															
	L0C								L0W (L0 width)												L0H (L0 Height)											
0x024	L0OA (L0 Origin Address)																															
0x028	L0DA (L0 Display Address)																															
0x02C					L0D Y (L0 Display Y)																L0DX (L0 Display X)											
0x110	L0EM (L0 Extend Mode)																															
	L0EC								L0PB																							L0WP
0x114					L0WY (L0 Window Y)																L0WX (L0 Window X)											
0x118					L0WH (L0 Window Height)																L0WW (L0 Window Width)											
0x030	L1M (L1 Mode)																															
	L1C	L1YC	L1CS	L1IM					L1W (L1 width)																							
0x034	L1DA (L1 layer Display Address) / CBDA0 (Capture Buffer Display Address 0)																															
0x038	CBDA1 (Capture Buffer Display Address 1)																															
0x120	L1EM (L1 Extend Mode)																															
	L1EC						VMAG		L1PB							L1LCE					CAOFS (Capture Address Offset)											
0x124					L1WY (L1 Window Y)																L1WX (L1 Window X)											
0x128					L1WH (L1 Window Height)																L1WW (L1 Window Width)											
0x040	L2M (L2 Mode)																															
	L2C	L2FLP							L2W (L2 width)												L2H (L2 Height)											
0x044	L2OA0 (L2 Origin Address 0)																															
0x048	L2DA0 (L2 Display Address 0)																															
0x04C	L2OA1 (L2 Origin Address 1)																															
0x050	L2DA1 (L2 Display Address 1)																															
0x054					L2DY (L2 Display Y)																L2DX (L2 Display X)											
0x130	L2EM (L2 Extend Mode)																															
	L2EC								L2PB																						L2OM	L2WP
0x134					L2WY (L2 Window Y)																L2WX (L2 Window X)											
0x138					L2WH (L2 Window Height)																L2WW (L2 Window Width)											
0x058	L3M (L3 Mode)																															
	L3C	L3FLP							L3W (L3 width)												L3H (L3 Height)											
0x05C	L3OA0 (L3 Origin Address 0)																															
0x060	L3DA0 (L3 Display Address 0)																															
0x064	L3OA1 (L3 Origin Address 1)																															
0x068	L3DA1 (L3 Display Address 1)																															

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x06C					L3DY (L3 Display Y)																L3DX (L3 Display X)												
0x140	L3EM (L3 Extend Mode)																															
	L3EC								L3PB																						L3OM	L3WP
0x144					L3WY (L3 Window Y)																L3WX (L3 Window X)											
0x148					L3WH (L3 Window Height)																L3WW (L3 Window Width)											
0x170	MDC (Multi Display Control)																															
	MDEN																SC1EN								SC0EN							
0x180	DLS (Display Layer Select)																															
													DLS4				DLS3				DLS2				DLS1				DLS0			
0x184									DBGC (Display Back Ground Color)																							
0x0B4	L0BLD (L0 Blend)																															
																L0BE	L0BS	L0BI	L0BP	L0ID	L0AF				L0BR							
0x188	L1BLD (L1 Blend)																															
																L1BE	L1BS	L1BI	L1BP	L1ID	L1AF				L1BR							
0x18C	L2BLD (L2 Blend)																															
																L2BE	L2BS	L2BI	L2BP	L2ID	L2AF				L2BR							
0x190	L3BLD (L3 Blend)																															
																L3BE	L3BS	L3BI	L3BP	L3ID	L3AF				L3BR							
0x194	L4BLD (L4 Blend)																															
																L4BE	L4BS	L4BI	L4BP	L4ID	L4AF				L4BR							

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x0B8	CKC (Chroma Key Control)																															
																	KCS	KYEN	KEYC													
0x0BC																	L0TC (L0 Transparent Control)															
																	L0ZT	L0TC (L0 Transparent Color)														
0x0C0	L2TC (L2 Transparent Control)																L3TC (L3 Transparent Control)															
	L2ZT	L2TC (L2 Transparent Color)															L3ZT	L3TC (L3 Transparent Color)														
0x1A0	L0ETC (L0 Extend Transparency Control)																															
	L0EZT								L0ETC (L0 Extend Transparent Color)																							
0x1A4	L1ETC (L1 Transparent Extend Control)																															
	L1EZT								L1ETC (L1 Extend Transparent Color)																							
0x1A8	L2ETC (L2 Transparent Extend Control)																															
	L2EZT								L2ETC (L2 Extend Transparent Color)																							
0x1AC	L3ETC (L3 Transparent Extend Control)																															
	L3EZT								L3ETC (L3 Extend Transparent Color)																							
0x1B0	L4ETC (L4 Extend Transparent Control)																															
	L4EZT								L4ETC (L4 Extend Transparent Color)																							
0x1E0	L1YCR0 (L1 YC to Red Coefficient 0)																															
						a12																a11										
0x1E4	L1YCR1 (L1 YC to Red Coefficient 1)																															
								b1														a13										
0x1E8	L1YCG0 (L1 YC to Green Coefficient 0)																															
						a22																a21										
0x1EC	L1YCG1 (L1 YC to Green Coefficient 1)																															
								b2														a23										
0x1F0	L1YCB0 (L1 YC to Blue Coefficient 0)																															
						a32																a31										
0x1F4	L1YCB1 (L1 YC to Blue Coefficient 1)																															
								b3														a33										

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0x400	L0PAL0																															
	A								R								G								B							
0x404	L0PAL1																															
:	:																															
0x7FC	L0PAL255																															
0x800	L1PAL0																															
	A								R								G								B							
0x804	L1PAL1																															
:	:																															
0xBFC	L1PAL255																															
0x1000	L2PAL0																															
	A								R								G								B							
0x1004	L2PAL1																															
:	:																															
0x13FC	L2PAL255																															
0x2000	L1LCT0																															
			R								G										B											
0x2004	L1LCT1																															
:	:																															
0x23FC	L1LCT255																															

1.6.4.3 Register Details

Common Control Registers

VCCC (Video Display/Capture Common Control)

VCCC

Register Address	01FD_7FF8$_H$																											
Bit Number	31	30	29	28	27	26							16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-															Csel		-									C0sr	-	V0sr
R/W	R0W0															RW		R0W0									RW	RW	RW
Initial Value	0_0000_0000_0000_0000															00		0_0000_0000									0	0	0

This register fully controls Display and Capture.

[bit0] V0sr (Vdisp 0 software reset)

This bit specifies whether to perform the Display software reset. Writing the VCSR register starts this reset operation. Only the reset target selection is written here.

0 Do not use the software reset.

1 Use the software reset.

[bit2] C0sr (Capture 0 software reset)

This bit specifies whether to perform the Capture software reset. Writing the VCSR register starts this reset operation. Only the reset target selection is written here.

0 Do not use the software reset.

1 Use the software reset.

[bit13, bit12] Csel (Capture select)

These bits select the Capture input.

00 External pin

01 Built-in NTSC decoder

10 Display output

11 Reserved

VCSR (Video Display/Capture Software Reset)

VCSR

Register Address	01FD_7FFC$_H$																											
Bit Number	31	30	29	28	27	26	25	24						15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	VCSR																												
R/W	R0W																												
Initial Value	0000_0000_0000_0000_0000_0000_0000_0000																												

This register executes a software reset. Write data is ignored. The writing operation generates a reset pulse for a single shot. The VCCC register is used to select the register for the software reset.

Display Controller Registers

DCM0 (Display Control Mode 0), DCM1 (Display Control Mode 1)

DCM0

Register Address	$01FD_0000_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	DEN	-											L4E	L23E	L1E	L0E	CKS	LCS	-	SC					EEQ	SCSY	-		SF	ESY	SYNC	
R/W	RW	RW0	R0					RXW0	R0		RW0		RW				RW	RW	R0	RW					RW	RW	RW0		RW	RW	RW	
Initial Value	0	0	0_0000					X	00		000						1	0	0	1_1110					0	0	00		0	1	00	

DCM1

Register Address	$01FD_0100_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	DEN	-											L4E	L3E	L2E	L1E	L0E	CKS	LCS	SC					EEQ	SCSY	-		SF	ESY	SYNC	
R/W	RW	RW0	R0		RW0								RW					RW	RW	RW					RW	RW	RW0		RW	RW	RW	
Initial Value	0	0	00		000_0000								0_0000					1	0	11_1101					0	0	00		0	1	00	

These registers set the display control mode. Although the DCM0 and DCM1 registers are in fact a single register, they are separated partly by the difference in their bit field formats. Different values between the 2 registers are not maintained. These registers are not initialized by a software reset.

[bit1, bit0] SYNC (Synchronize)

These bits set the synchronous mode.

00 Non-interlace mode

01 Interlace mode

11 Interlaced video mode

[bit2] ESY (External Synchronize)

This bit sets external synchronous mode.

0 External synchronization disabled

1 External synchronization enabled

[bit3] SF (Synchronize signal Format)

This bit sets the synchronous signal format (VSYNC, HSYNC).

0 Negative logic

1 Positive logic

[bit6] SCSY (Select CSYNC output)

This bit sets what is output to the CSYNC output pin.

0 Output a CSYNC signal to the CCYNC output pin.
See "Composite Synchronous Signal".

1 Output a GV signal to the CCYNC output pin.
See "1.6.5.5 External Synchronization".

[bit7] EEQ (Enable Equalizing pulse)

This bit sets the CCYNC signal mode.

0 Do not insert an equalizing pulse into the CCYNC signal.

1 Insert an equalizing pulse into the CCYNC signal.

[bit13 to bit8] SC (Scaling)

These bits divide the display reference clock by the preset ratio to generate a dot clock.

DCM0	
x00000	Frequency not divided
x00001	Frequency division = 1/4
x00010	Frequency division = 1/6
x00011	Frequency division = 1/8
:	:
x11111	Frequency division = 1/64

DCM1	
000000	Frequency not divided
000001	Frequency division = 1/2
000010	Frequency division = 1/3
000011	Frequency division = 1/4
:	:
111111	Frequency division = 1/64

If n is set in DCM0, the frequency division rate is 1/(2n+2).

If m is set in DCM1, the frequency division rate is 1/(m+1).

Basically, both cases are setting parameters with the same function, leading to the following expression: 2n+2=m+1. Accordingly, the relation of m=2n+1 is established. Therefore, if n is set for the SC field of DCM0, 2n+1 is reflected in DCM1.

[bit14] LCS (Lower frequency Clock Select)

This bit selects the frequency of the built-in PLL clock.

0 Standard frequency GDCPLLCLK (108 MHz or less)

1 Slightly low-frequency GDCSSCGCLK (81 MHz or less)

[bit15] CKS (Clock Source)

This bit selects the reference clock.

0 Set the reference clock to built-in PLL output.

1 Set the reference clock to DCLKI signal input.

[bit16] L0E (L0-layer Enable)

This bit enables L0 layer display.

0 Do not display the L0 layer.

1 Display the L0 layer.

[bit17] L1E (L1-layer Enable)

This bit enables L1 layer display.

0 Do not display the L1 layer.

1 Display the L1 layer.

[bit18] L23E (L2 & L3-layer Enable) -------- DCM0

This bit enables the L2 and L3 layers simultaneously. The layers correspond to the M layer of existing products.

0 Do not display the L2 layer and L3 layer.

1 Display the L2 layer and L3 layer.

L2E(L2-layer Enable) ------- DCM1

This bit enables L2 layer display.

0 Do not display the L2 layer.

1 Display the L2 layer.

[bit19] L4E (L4 layer Enable) -------- DCM0

This bit enables the L4 layer. The layer corresponds to the BL layer of existing products. It is used only for sprites.

0 Do not display the L4 layer.

1 Display the L4 layer.

L3E (L3-layer Enable) ------- DCM1

This bit enables L3 layer display.

0 Do not display the L3 layer.

1 Display the L3 layer.

[bit20] L4E (L4-layer Enable)

This bit enables L4 layer display.

0 Do not display the L4 layer.

1 Display the L4 layer.

[bit31] DEN (Display Enable)

This bit enables display. When the bit is read, a value through the DCLK clock is read.

Notes:
Note the following points about updating this bit when using the sprite display function. To update the bit when it is set (0 written) to not output a display signal, do so during a vertical synchronous period. To detect a vertical synchronous event, use a vertical synchronous interrupt or the Wait Trigger function of a command list. If the bit is not updated during a vertical synchronous period, sprites may not be appropriately displayed upon a restart of display output.

0 Do not output a display signal.

1 Output a display signal.

DCM2 (Display Control Mode 2)

Register Address	$01FD_0104_H$															
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-														RUF	RUM
R/W	R0														RW	RW
Initial Value	00_0000_0000_0000														0	0

[bit0] RUM (Register Update Mode)

This bit selects the mode for reflecting register values synchronized for vertical synchronization.

0 Reflect register updates in real time to the internal control circuit. Updating it during the display period will interfere with display.

1 Send register values synchronized for vertical synchronization to the internal control circuit. The following RUF flag controls the simultaneity.

[bit1] RUF (Register Update Flag)

Writing "1" in this flag gives an instruction to update values at the next vertical synchronization. This bit becomes "0" when the update is finished.

0 Initial state, or update finished

1 Waiting for vertical synchronous

DCM3 (Display Control Mode 3)

Register Address	$01FD_0108_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19		12	11	10	9	8	7	6	5	4	3	2	1	0	
Bit Field Name	-					SELLAT	-				RGBrv	RGBsh	-					DCKed	DCKinv	-			DCKD					
R/W	RW0					RW	RW0				RW	RW	RW0			R0	RW 0	RW	RW	R0			RW					
Initial Value	00000					0	0000				0	0	00 .. 0			0	0	0	0	000			00000					

[bit4 to bit0] DCKD (Display Clock Delay)

These bits define an additional delay with the internal PLL clock.

00000	No additional delay
00010	+2 PLL clocks
00100	+3 PLL clocks
00110	+4 PLL clocks
:	:
11110	+17 PLL clocks
xxxx1	Reserved

[bit8] DCKinv (Display Clock inversion)

0 DCLKO output signal not inverted

1 DCLKO output signal inverted

[bit9] DCKed (Display clock edge)

This bit defines which edge is used.

0 Single-edge mode. The rising edge is used for RGB output.

1 Both-edges mode. The positive and negative edges are used for RGB output.

[bit20] RGBsh (RGB shift)

0 Normal output

1 RGB output shifted by only 2 bits to the LSB side

[bit21] RGB rv (RGB reverse)

0 Normal output

1 Output with RGB bit positions reversed

[bit26] SELLAT (Select latency)

This bit selects the RGB output latency of the horizontal synchronous signal reference.

0 14 display clocks

1 1 display clock

[Restrictions]

If the SELLAT setting is "0", HSP must satisfy the following expression:

HSP > HDP+14

1 –1 display clock

(Continued on the next page)

Restrictions:

- When the SELLAT setting is "1", the horizontal front porch can be set to 3 or more.
- If the SELLAT setting is "1", HSP must satisfy the following expression: HSP>HDP+1
- Use under the following conditions is allowed even with the setting of HSP=HDP+1.
- The VSYNC output position should be only 1 raster behind, and the vertical front porch should be 2 rasters or more ahead.[1]
- The synchronous mode (DCM1/SYNC field setting) should be non-interlace mode (00).

[1]: There is no problem with the above delay of the VSYNC output signal as long as it is within the acceptable range of the display panel on the receiver side.

HTP (Horizontal Total Pixels)

Register Address	$01FD_0004_H$															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Field Name	-				HTP											
R/W	R0				RW											
Initial Value	0000				X											

This register specifies the horizontal total pixel count. The set value + 1 is the total pixel count.

HDP (Horizontal Display Period)

Register Address	$01FD_0008_H$															
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				HDP											
R/W	R0				RW											
Initial Value	0000				X											

This register specifies the horizontal display period in units of pixel clocks. The set value + 1 is the pixel count of the display period.

HDB (Horizontal Display Boundary)

Register Address	$01FD_0008_H$															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Field Name	-				HDB											
R/W	R0				RW											
Initial Value	0000				X											

This register specifies the left-screen display period in units of pixel clocks. The set value + 1 is the pixel count of the left-screen display period. If right and left split display is not specified, the set value is equal to HDP.

HSP (Horizontal Synchronize Pulse Position)

Register Address	$01FD_000C_H$															
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				HSP											
R/W	R0				RW											
Initial Value	0000				X											

This register specifies the pulse position of the horizontal synchronous signal in units of pixel clocks. The horizontal synchronous signal is asserted when the clock count from the start of the display period (including an offset equal to 14 clocks) reaches the set value + 1.

HSW (Horizontal Synchronize Pulse Width)

Register Address	01FD_000C$_H$							
Bit Number	23	22	21	20	19	18	17	16
Bit Field Name	HSW							
R/W	RW							
Initial Value	X							

This register specifies the pulse width of the horizontal synchronous signal in units of pixel clocks. The set value + 1 is the clock count of the pulse width.

VSW (Vertical Synchronize Pulse Width)

Register Address	01FD_000C$_H$							
Bit Number	31	30	29	28	27	26	25	24
Bit Field Name	VSWH	-	VSW					
R/W	RW	R0	RW					
Initial Value	0	0	X					

[bit29 to bit24] VSW (Vertical Synchronize pulse Width)

These bits specify the pulse width of the vertical synchronous signal in units of rasters. The set value + 1 is the pulse width raster count.

[bit31] VSWH (VSW Half)

This bit extends the pulse width of the vertical synchronization signal by the width of half a raster.

If the bit is "1", the width is extended by half of the H circle. This is the assumed function for PAL interlaced display. For LCD panel display, use "0" in VSWH because the display is non-interlaced.

VTR (Vertical Total Rasters)

Register Address	01FD_0010$_H$															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Field Name	-				VTR											
R/W	R0				RW											
Initial Value	0000				X											

This register specifies the vertical total raster count. The set value + 1 is the total raster count. For interlaced display, the set value + 1.5 is the total raster count in 1 field, and 2 * set value + 3 is the total raster count in 1 frame (see "Interlaced Display").

VSP (Vertical Synchronize Pulse Position)

Register Address	01FD_0014H															
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				VSP											
R/W	R0				RW											
Initial Value	0000				X											

This register specifies the pulse position of the vertical synchronous signal in units of rasters. A vertical synchronous pulse is asserted from the set value + first raster counted from the raster starting the display.

VDP (Vertical Display Period)

Register Address	01FD_0014H															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Field Name	-				VDP											
R/W	R0				RW											
Initial Value	0000				X											

This register specifies the vertical display period in units of rasters. The set value + 1 is the display raster count.

L0M (L0-layer Mode)

Register Address	01FD_0020H																											
Bit Number	31	30	29	28			25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L0C	-							L0W								-				L0H											
R/W	RW	R0							RW								R0				RW											
Initial Value	0	000_0000							X								0000				X											

[bit11 to bit0] L0H (L0-layer Height)

These bits specify the logical frame height of the L0 layer in units of pixels. The set value + 1 is the height.

[bit23 to bit16] L0W (L0-layer memory Width)

These bits set the logical frame memory width (stride) of the L0 layer in units of 64 bytes.

[bit31] L0C (L0-layer Color mode)

This bit sets the color mode of the L0 layer.

0 Indirect color (8 bits/pixel) mode

1 Direct color (16 bits/pixel) ARGB mode

L0EM (L0-layer Extended Mode)

Register Address	$01FD_0110_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	-----	4	3	2	1	0				
Bit Field Name	L0EC				-				L0PB				-															L0WP				
R/W	RW				R0				RW				R0															RW				
Initial Value	000				0_0000				0000				000_0000_0000_0000_0000															0				

[bit0] L0WP (L0-layer Window Position enable)

This bit selects the display position of the L0 layer. Display compatibility with the old model (MB86290-292) can be selected.

0 Compatibility-mode display (C layer supported)

1 Window display

[bit23 to bit20] L0PB (L0-layer Palette Base)

These bits indicate the value added to the index when subtracting the palette of the L0 layer. The added value is 16 times the set value.

[bit31 to bit29] L0EC (L0-layer Extended Color mode)

These bits set the extended color mode of the L0 layer.

000 Mode determined by L0C

010 Direct color (24 bits/pixel) mode

011 Direct color (16 bits/pixel) RGB565 mode

100 Direct color (16 bits/pixel) RGBA mode

110 Direct color (24 bits/pixel) RGBA mode

other Reserved

L0OA (L0-layer Origin Address)

Register Address	$01FD_0024_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L0OA																															
R/W	RW																												RW0			
Initial Value	X																															

This register sets the origin address of the logical frame of the L0 layer. Since the lower 4 bits are fixed at "0", the address is 16-byte aligned.

L0DA (L0-layer Display Address)

Register Address	$01FD_0028_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L0DA																															
R/W	RW																															
Initial Value	X																															

This register sets the display origin address of the L0 layer. For direct color mode (16 bits/pixel), the lower 1 bit is "0", and the address is treated as being aligned in units of 2 bytes.

L0DP (L0-layer Display Position)

Register Address	$01FD_002C_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L0DY												-				L0DX											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the display start position coordinates (DX, DY) of the L0 layer, based on the origin of the logical frame, in units of pixels.

[bit11 to bit0] L0DX (L0-layer Display Position X)

These bits specify the X coordinate.

[bit27 to bit16] L0DY (L0-layer Display Position Y)

These bits specify the Y coordinate.

L0WP (L0-layer Window Position)

Register Address	$01FD_0114_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L0WY												-				L0WX											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the display position coordinates (WX, WY) of the L0 layer window. The origin is at the top left of the display window.

[bit11 to bit0] L0WX (L0-layer Window Position X)

These bits specify the X coordinate.

[bit27 to bit16] L0WY (L0-layer Window Position Y)

These bits specify the Y coordinate.

L0WS (L0-layer Window Size)

Register Address	$01FD_0118_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L0WH												-				L0WW											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the size of the L0 layer window.

[bit11 to bit0] L0WW (L0-layer Window Width X)

These bits specify the width in units of pixels. Do not set 0.

[bit27 to bit16] L0WH (L0-layer Window Height Y)

These bits specify the height. The set value + 1 is the height.

L1M (L1-layer Mode)

Register Address	$01FD_0030_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	-----		5	4	3	2	1	0
Bit Field Name	L1C	L1YC	L1CS	L1IM	-				L1W								-											
R/W	RW	RW	RW	RW	R0				RW								R0											
Initial Value	0	0	0	0	0000				X								0000_0000_0000_0000											

[bit23 to bit16] L1W (L1-layer memory Width)

These bits set the logical frame memory width (stride) of the L1 layer in units of 64 bytes.

[bit28] L1IM (L1-layer Interlace Mode)

This bit sets the video capture operation mode when L1CS is in capture mode.

0 Normal mode

1 For non-interlaced display, display is in WEAVE mode. For interlaced and video display, buffers are managed in units of frames (pairs of odd and even fields).

[bit29] L1CS (L1-layer Capture Synchronize)

This bit sets whether to use the L1 layer as the normal display layer or as video capture.

0 Normal mode

1 Capture mode

[bit30] L1YC (L1-layer YC mode)

This bit sets the color format of the L1 layer.

0 RGB mode

1 YC mode

[bit31] L1C (L1-layer Color mode)

This bit sets the color mode of the L1 layer.

0 Indirect color (8 bits/pixel) mode

1 Direct color (16 bits/pixel) RGB mode

L1EM (L1-layer Extended Mode)

Register Address	$01FD_0120_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	...	4	3	2	1	0
Bit Field Name	L1EC			-			VMAG		L1PB				-			L1LCE	-			CAOFS										
R/W	RW			RW0	R0		RW		RW				R0			RW	RW0			RW										
Initial Value	000			0	00		00		0000				000			0				0_0000_0000_0000										

[bit12 to bit0] CAOFS (Capture Offset)

These bits fine-tune the capture image display position. Specify 0 when not using it.

[bit16] L1LCE (L1-layer level conversion enable)

This bit specifies color element level conversion of the L1 layer. The conversion table is defined in L1LCT0 to 255.

0 Color element level conversion not used

1 Color element level conversion used

[bit23 to bit20] L1PB (L1-layer Palette Base)

These bits indicate the value added to the index when pulling the palette of the L1 layer. The added value is 16 times the set value.

[bit25, bit24] VMAG (Video Magnify)

These bits specify expansion of the capture image.

00 Expansion function not used

01 Reserved

10 Expansion function used

11 Reserved

[bit31 to bit29] L1EC (L1-layer Extended Color mode)

These bits set the extended color mode of the L1 layer.

000 Determined by L1C

010 Direct color (24 bits/pixel) mode

011 Direct color (16 bits/pixel) RGB565 mode

100 Direct color (16 bits/pixel) RGBA mode

110 Direct color (24 bits/pixel) RGBA mode

other Reserved

L1DA (L1-layer Display Address)

Register Address	$01FD_0034_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L1DA																															
R/W	RW																															
Initial Value	X																															

This register sets the display origin address of the L1 layer. For direct color mode (16 bits/pixel), the lower 1 bit is "0", and the address is treated as being aligned in 2 bytes.

Since wraparound processing is not executed for the L1 layer, no XY coordinates are specified for the frame origin linear address and display position.

The register is assigned to the same address as that of the CBDA0 register described below. The L1CS bit in the L1M register determines which of the 2 registers is enabled.

L1CS=0 --- L1DA register enabled

L1CS=1 --- CBDA0 register enabled

CBDA0 (Capture Buffer Display Address 0)

Register Address	$01FD_0034_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	CBDA0																															
R/W	R																															
Initial Value	X																															

This register is a read-only register that can be accessed only when the L1CS bit in the L1M register is "1". It indicates the start address of the displayed capture image. If the L1CS bit is "1" and the L1IM bit is also "1", it indicates the start address of odd fields of the capture window.

CBDA1 (Capture Buffer Display Address 1)

Register Address	$01FD_0038_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	CBDA1																															
R/W	R																															
Initial Value	X																															

This register is a read-only register that is enabled only when the L1CS bit is "1" and the L1IM bit is also "1". It indicates the start address of even fields of the capture window.

L1WP (L1-layer Window Position)

Register Address	$01FD_0124_H$ ($01FD_0018_H$)																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L1WY												-				L1WX											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the display position coordinates (WX, WY) of the L1 layer window. The origin is at the top left of the display window. It is mapped to 2 addresses.

[bit11 to bit0] L1WX (L1-layer Window Position X)

These bits specify the X coordinate.

[bit27 to bit16] L1WY (L1-layer Window Position Y)

These bits specify the Y coordinate.

L1WS (L1-layer Window Size)

Register Address	$01FD_0128_H$ ($01FD_001C_H$)																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L1WH												-				L1WW											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the size of the L1 layer window. It is mapped to 2 addresses.

[bit11 to bit0] L1WW (L1-layer Window Width X)

These bits specify the width in units of pixels. Do not set 0.

[bit27 to bit16] L1WH (L1-layer Window Height Y)

These bits specify the height. The set value + 1 is the height.

L2M (L2-layer Mode)

Register Address	$01FD_0040_H$																								
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	--	4	3	2	1	0
Bit Field Name	L2C	L2FLP		-					L2W								-				L2H								
R/W	RW	RW		RW0	R0				RW								R0				RW								
Initial Value	0	00		0	0000				X								0000				X								

[bit11 to bit0] L2H (L2-layer Height)

These bits specify the logical frame height of the L2 layer in units of pixels. The set value + 1 is the height.

[bit23 to bit16] L2W (L2-layer memory Width)

These bits set the logical frame memory width (stride) of the L2 layer in units of 64 bytes.

[bit30, bit29] L2FLP (L2-layer Flip mode)

These bits set the flipping mode of the L2 layer.

00 Side 0 displayed

01 Side 1 displayed

10 Sides 0 and 1 displayed alternately for every frame

11 Reserved

[bit31] L2C (L2-layer Color mode)

This bit sets the color mode of the L2 layer.

0 Indirect color (8 bits/pixel) mode

1 Direct color (16 bits/pixel) ARGB mode

L2EM (L2-layer Extended Mode)

Register Address	$01FD_0130_H$																									
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	4	3	2	1	0
Bit Field Name	L2EC			-					L2PB				-													L2OM	L2WP
R/W	RW			R0					RW				R0													RW	RW
Initial Value	000			0_0000					0000				00_0000_0000_0000_0000													0	0

[bit0] L2WP (L2-layer Window Position enable)

This bit selects the display position of the L2 layer. Display compatibility with the old model (MB86290-292) can be selected.

0 Compatibility-mode display (ML layer supported)

1 Window display

[bit1] L2OM (L2-layer Overlay Mode)

This bit selects the overlay mode of the L2 layer. Display compatibility with the old model (MB86290-292) can be selected.

0 Compatibility mode

1 Extended mode

[bit23 to bit20] L2PB (L2-layer Palette Base)

These bits indicate the value added to the index when subtracting the palette of the L2 layer. The added value is 16 times the set value.

[bit31 to bit29] L2EC (L2-layer Extended Color mode)

These bits set the extended color mode of the L2 layer.

000 Determined by L2C

010 Direct color (24 bits/pixel) mode

011 Direct color (16 bits/pixel) RGB565 mode

100 Direct color (16 bits/pixel) RGBA mode

110 Direct color (24 bits/pixel) RGBA mode

other Reserved

L2OA0 (L2-layer Origin Address 0)

Register Address	$01FD_0044_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L2OA0																															
R/W	RW																												RW0			
Initial Value	X																															

This register sets the origin address of the logical frame of the L2 layer and side 0. Since the lower 4 bits are fixed at "0", the address is 16-byte aligned.

L2DA0 (L2-layer Display Address 0)

Register Address	$01FD_0048_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L2DA0																															
R/W	RW																															
Initial Value	X																															

This register sets the display origin address of the L2 layer and side 0. For direct color mode (16 bits/pixel), the lower 1 bit is "0", and the address is treated as being aligned in 2 bytes.

L2OA1 (L2-layer Origin Address 1)

Register Address	$01FD_004C_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L2OA1																															
R/W	RW																												RW0			
Initial Value	X																															

This register sets the origin address of the logical frame of the L2 layer and side 1. Since the lower 4 bits are fixed at "0", the address is 16-byte aligned.

L2DA1 (L2-layer Display Address 1)

Register Address	$01FD_0050_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L2DA0																															
R/W	RW																															
Initial Value	X																															

This register sets the display origin address of the L2 layer and side 1. For direct color mode (16 bits/pixel), the lower 1 bit is "0", and the address is treated as being aligned in 2 bytes.

L2DP (L2-layer Display Position)

Register Address	$01FD_0054_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L2DY												-				L2DX											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the display start position coordinates (DX, DY) of the L2 layer, based on the origin of the logical frame, in units of pixels.

[bit11 to bit0] L2DX (L2-layer Display Position X)

These bits specify the X coordinate.

[bit27 to bit16] L2DY (L2-layer Display Position Y)

These bits specify the Y coordinate.

L2WP (L2-layer Window Position)

Register Address	$01FD_0134_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L2WY												-				L2WX											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the display position coordinates (WX, WY) of the L2 layer window. The origin is at the top left of the display window.

[bit11 to bit0] L2WX (L2-layer Window Position X)

These bits specify the X coordinate.

[bit27 to bit16] L2WY (L2-layer Window Position Y)

These bits specify the Y coordinate.

L2WS (L2-layer Window Size)

Register Address	$01FD_0138_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L2WH												-				L2WW											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the size of the L2 layer window.

[bit11 to bit0] L2WW (L2-layer Window Width X)

These bits specify the width in units of pixels. Do not set 0.

[bit27 to bit16] L2WH (L2-layer Window Height Y)

These bits specify the height. The set value + 1 is the height.

L3M (L3-layer Mode)

Register Address	$01FD_0058_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	--	4	3	2	1	0
Bit Field Name	L3C	L3FLP		-					L3W								-				L3H								
R/W	RW	RW		RW0	R0				RW								R0				RW								
Initial Value	0	00		0	0000				X								0000				X								

[bit11 to bit0] L3H (L3-layer Height)

These bits specify the logical frame height of the L3 layer in units of pixels. The set value + 1 is the height.

[bit23 to bit16] L3W (L3-layer memory Width)

These bits set the logical frame memory width (stride) of the L3 layer in units of 64 bytes.

[bit30, bit29] L3FLP (L3-layer Flip mode)

These bits set the flipping mode of the L3 layer.

00 Side 0 displayed

01 Side 1 displayed

10 Sides 0 and 1 displayed alternately for every frame

11 Reserved

[bit31] L3C (L3-layer Color mode)

This bit sets the color mode of the L3 layer.

0 Indirect color (8 bits/pixel) mode

1 Direct color (16 bits/pixel) ARGB mode

L3EM (L3-layer Extended Mode)

Register Address	$01FD_0140_H$																									
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	4	3	2	1	0
Bit Field Name	L3EC			-					L3PB				-													L3OM	L3WP
R/W	RW			R0					RW				R0													RW	RW
Initial Value	000			0_0000					0000				00_0000_0000_0000_0000													0	0

[bit0] L3 WP (L3-layer Window Position enable)

This bit selects the display position of the L3 layer. Display compatibility with the old model (MB86290-292) can be selected.

0 Compatibility-mode display (ML layer supported)

1 Window display

[bit1] L3OM (L3-layer Overlay Mode)

This bit selects the overlay mode of the L3 layer. Display compatibility with the old model (MB86290-292) can be selected.

0 Compatibility mode

1 Extended mode

[bit23 to bit20] L3PB (L3-layer Palette Base)

These bits indicate the value added to the index when subtracting the palette of the L3 layer. The added value is 16 times the set value.

[bit31 to bit29] L3EC (L3-layer Extended Color mode)

These bits set the extended color mode of the L3 layer.

000 Determined by L3C

010 Direct color (24 bits/pixel) ARGB mode

011 Direct color (16 bits/pixel) RGB565 mode

100 Direct color (16 bits/pixel) RGBA mode

110 Direct color (24 bits/pixel) RGBA mode

other Reserved

L3OA0 (L3-layer Origin Address 0)

Register Address	$01FD_005C_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L3OA0																															
R/W	RW																												RW0			
Initial Value	X																															

This register sets the origin address of the logical frame of the L3 layer and side 0. Since the lower 4 bits are fixed at "0", the address is 16-byte aligned.

L3DA0 (L3-layer Display Address 0)

Register Address	$01FD_0060_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L3DA0																															
R/W	RW																															
Initial Value	X																															

This register sets the display origin address of the L3 layer and side 0. For direct color mode (16 bits/pixel), the lower 1 bit is "0", and the address is treated as being aligned in 2 bytes.

L3OA1 (L3-layer Origin Address 1)

Register Address	$01FD_0064_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L3OA1																															
R/W	RW																												RW0			
Initial Value	X																															

This register sets the origin address of the logical frame of the L3 layer and side 1. Since the lower 4 bits are fixed at "0", the address is 16-byte aligned.

L3DA1 (L3-layer Display Address 1)

Register Address	$01FD_0068_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L3DA1																															
R/W	RW																															
Initial Value	X																															

This register sets the display origin address of the L3 layer and side 1. For direct color mode (16 bits/pixel), the lower 1 bit is "0", and the address is treated as being aligned in 2 bytes.

L3DP (L3-layer Display Position)

Register Address	$01FD_006C_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L3DY												-				L3DX											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the display start position coordinates (DX, DY) of the L3 layer, based on the origin of the logical frame, in units of pixels.

[bit11 to bit0] L3DX (L3-layer Display Position X)

These bits specify the X coordinate.

[bit27 to bit16] L3DY (L3-layer Display Position Y)

These bits specify the Y coordinate.

L3WP (L3-layer Window Position)

Register Address	$01FD_0144_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L3WY												-				L3WX											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the display position coordinates (WX, WY) of the L3 layer window. The origin is at the top left of the display window.

[bit11 to bit0] L3WX (L3-layer Window Position X)

These bits specify the X coordinate.

[bit27 to bit16] L3WY (L3-layer Window Position Y)

These bits specify the Y coordinate.

L3WS (L3-layer Window Size)

Register Address	$01FD_0148_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				L3WH												-				L3WW											
R/W	R0				RW												R0				RW											
Initial Value	0000				X												0000				X											

This register sets the size of the L3 layer window.

[bit11 to bit0] L3WW (L3-layer Window Width X)

These bits specify the width in units of pixels. Do not set 0.

[bit27 to bit16] L3WH (L3-layer Window Height Y)

These bits specify the height. The set value + 1 is the height.

DLS (Display Layer Select)

Register Address	$01FD_0180_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-												DLS4				DLS3				DLS2				DLS1				DLS0			
R/W	RW												RW				RW				RW				RW				RW			
Initial Value	0111_0110_0101												0100				0011				0010				0001				0000			

This register defines the sequence of overlaying layers.

[bit3 to bit0] DLS0 (Display Layer Select 0)

These bits select the topmost layer.

0000 L0 layer
0001 L1 layer
: :
0111 L7 layer
1000 Reserved
: :
1110 Reserved
1111 None selected

[bit7 to bit4] DLS1 (Display Layer Select 1)

These bits select the second layer from the top. The values have the same meaning as those of DSL0.

[bit11 to bit8] DLS2 (Display Layer Select 2)

These bits select the third layer from the top. The values have the same meaning as those of DSL0.

[bit15 to bit12] DLS3 (Display Layer Select 3)

These bits select the fourth layer from the top. The values have the same meaning as those of DSL0.

[bit19 to bit16] DLS4 (Display Layer Select 4)

These bits select the fifth layer from the top. The values have the same meaning as those of DSL0.

MDC (Multi Display Control)

Register Address	$01FD_0170_H$																													
Bit Number	31	30	29					25	24	23	22					19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	MDen	-								-										SC1en								SC0en							
R/W	RW	R0								R0W0										RW								RW							
Initial Value	0	000_0000								X										X								X							

This register controls dual display. The register enables or disables it and sets the relationship between the display screen and the display layer. (See "1.6.5.8 Dual Display.")

[bit0] SC0en0 (screen 0 enable 0)

0 L0 not included in screen 0

1 L0 included in screen 0

[bit1] SC0en1 (screen 0 enable 1)

0 L1 not included in screen 0

1 L1 included in screen 0

[bit4] SC0en4 (screen 0 enable 4)

0 L4 not included in screen 0

1 L4 included in screen 0

[bit8] SC1en0 (screen 1 enable 0)

0 L0 not included in screen 1

1 L0 included in screen 1

[bit9] SC1en1 (screen 1 enable 1)

0 L1 not included in screen 1

1 L1 included in screen 1

[bit12] SC1en4 (screen 1 enable 4)

0 L4 not included in screen 1

1 L4 included in screen 1

[bit31] MDen (multi display enable)

This bit enables dual (multi) display mode.

0 Single-display mode

1 Dual-display mode

DBGC (Display Background Color)

Register Address	$01FD_0184_H$																													
Bit Number	31	30	29	-----	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-						DBGR								DBGG								DBGB							
R/W	RW0	R0					RW								RW								RW							
Initial Value	0	000_0000					0000_0000								0000_0000								0000_0000							

This register specifies the color displayed in areas outside the display area of each layer in the window.

[bit7 to bit0] DBGB (Display Background Blue)

These bits specify the blue level of the background color.

[bit15 to bit8] DBGG (Display Background Green)

These bits specify the green level of the background color.

[bit23 to bit16] DBGR (Display Background Red)

These bits specify the red level of the background color.

L0BLD (L0 Blend)

Register Address	01FD_00B4$_H$																											
Bit Number	31	30								18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-											L0BE	L0BS	L0BI	L0BP	L0ID	L0AF		-		L0BR							
R/W	R0											RW	RW	RW	RW	RW	RW		R0W0	R0	RW							
Initial Value	000_0000_0000_0000											0	0	0	0	0	0		0	0	0000_0000							

This register specifies the blend parameters of the L0 layer. The register corresponds to BRATIO or BMODE of the old model (MB86290-292).

[bit7 to bit0] L0BR (L0-layer Blend Ratio)

These bits set the blend ratio. Basically, the blend ratio is the set value/256.

[bit11, bit10] L0AF (L0-layer Alpha Field)

These bits select the bit field of the alpha layer.

00 bit7 to bit0 are treated as the alpha layer.
01 bit15 to bit8 are treated as the alpha layer.
10 bit23 to bit16 are treated as the alpha layer.
11 Reserved

[bit12] L0ID (L0-layer Ignore Data)

This bit specifies whether field A of the display data is valid.

0 Blend only if field A of the display data is "1".
1 Ignore field A of the display data.

[bit13] L0BP (L0-layer Blend Plane)

This bit selects whether to use a constant value or the alpha layer for the blend ratio.

0 Use the L0BR value as the blend ratio.
1 Use the L3 layer pixels as the blend ratio.

[bit14] L0BI (L0-layer Blend Increment)

This bit selects whether to add 1/256 when the blend ratio is not 0.

0 Calculate the blend ratio as is.
1 Add 1/256 when the blend ratio is not 0.

[bit15] L0BS (L0-layer Blend Select)

This bit selects the blend calculation expression.

0 Upper image * blend ratio + lower image * (1 – blend ratio)
1 Upper image * (1 – blend ratio) + lower image * blend ratio

[bit16] L0BE (L0-layer Blend Enable)

This bit enables blending.

0 Overlay using a transparent color

1 Overlay using blending

For blending, the blend mode must be specified for L0BE, and field A in the L0 layer display data must be "1". If L0ID is "1", field A of the display data is ignored.

L1BLD (L1 Blend)

Register Address	$01FD_0188_H$																			
Bit Number	31	30		18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-					L1BE	L1BS	L1BI	L1BP	L1ID	L1AF		-		L1BR							
R/W	R0					RW	RW	RW	RW	RW	RW		R0W0	R0	RW							
Initial Value	000_0000_0000_0000					0	0	0	0	0	0		0	0	0000_0000							

This register specifies the blend parameters of the L1 layer

[bit7 to bit0] L1BR (L1-layer Blend Ratio)

These bits set the blend ratio. Basically, the blend ratio is the set value/256.

[bit11, bit10] L1AF (L1-layer Alpha Field)

These bits select the bit field of the alpha layer.

00 bit7 to bit0 are treated as the alpha layer.

01 bit15 to bit8 are treated as the alpha layer.

10 bit23 to bit16 are treated as the alpha layer.

11 Reserved

[bit12] L1ID (L1-layer Ignore Data)

This bit specifies whether field A of the display data is valid.

0 Blend only if field A of the display data is "1".

1 Ignore field A of the display data.

[bit13] L1BP (L1-layer Blend Plane)

This bit selects whether to use a constant value or the alpha layer for the blend ratio.

0 Use the L1BR value as the blend ratio.

1 Use the L3 layer pixels as the blend ratio.

[bit14] L1BI (L1-layer Blend Increment)

This bit selects whether to add 1/256 when the blend ratio is not 0.

0 Calculate the blend ratio as is.

1 Add 1/256 when the blend ratio is not 0.

[bit15] L1BS (L1-layer Blend Select)

This bit selects the blend calculation expression.

0 Upper image * blend ratio + lower image * (1 – blend ratio)

1 Upper image * (1 – blend ratio) + lower image * blend ratio

[bit16] L1BE (L1-layer Blend Enable)

This bit enables blending.

0 Overlay using a transparent color

1 Overlay using blending

For blending, the blend mode must be specified for L1BE, and field A in the L1 layer display data must be "1". If L1ID is "1", field A of the display data is ignored.

L2BLD (L2 Blend)

Register Address	01FD_018C$_H$																			
Bit Number	31	30		18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-					L2BE	L2BS	L2BI	L2BP	L2ID	L2AF		-		L2BR							
R/W	R0					RW	RW	RW	RW	RW	RW		R0W0	R0	RW							
Initial Value	000_0000_0000_0000					0	0	0	0	0	0		0	0	0000_0000							

This register specifies the blend parameters of the L2 layer.

[bit7 to bit0] L2BR (L2-layer Blend Ratio)

These bits set the blend ratio. Basically, the blend ratio is the set value/256.

[bit11, bit10] L2AF (L2-layer Alpha Field)

These bits select the bit field of the alpha layer.

00 bit7 to bit0 are treated as the alpha layer.
01 bit15 to bit8 are treated as the alpha layer.
10 bit23 to bit16 are treated as the alpha layer.
11 Reserved

[bit12] L2ID (L2-layer Ignore Data)

This bit specifies whether field A of the display data is valid.

0 Blend only if field A of the display data is "1".
1 Ignore field A of the display data.

[bit13] L2BP (L2-layer Blend Plane)

This bit selects whether to use a constant value or the alpha layer for the blend ratio.

0 Use the L2BR value as the blend ratio.
1 Use the L3 layer pixels as the blend ratio.

[bit14] L2BI (L2-layer Blend Increment)

This bit selects whether to add 1/256 when the blend ratio is not 0.

0 Calculate the blend ratio as is.
1 Add 1/256 when the blend ratio is not 0.

[bit15] L2BS (L2-layer Blend Select)

This bit selects the blend calculation expression.

0 Upper image * blend ratio + lower image * (1 – blend ratio)
1 Upper image * (1 – blend ratio) + lower image * blend ratio

[bit16] L2BE (L2-layer Blend Enable)

This bit enables blending.

0 Overlay using a transparent color
1 Overlay using blending

For blending, the blend mode must be specified for L2BE, and field A in the L2 layer display data must be "1". If L2ID is "1", field A of the display data is ignored.

L3BLD (L3 Blend)

Register Address	01FD_0190$_H$																			
Bit Number	31	30		18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-					L3BE	L3BS	L3BI	L3BP	L3ID	L3AF		-		L3BR							
R/W	R0					RW	RW	RW	RW	RW	RW		R0W0	R0	RW							
Initial Value	000_0000_0000_0000					0	0	0	0	0	0		0	0	0000_0000							

This register specifies the blend parameters of the L3 layer.

[bit7 to bit0] L3BR (L3-layer Blend Ratio)

These bits set the blend ratio. Basically, the blend ratio is the set value/256.

[bit11, bit10] L3AF (L3-layer Alpha Field)

These bits select the bit field of the alpha layer. Since the L3 layer refers to itself as an alpha value, these values do not apply.

- 00 bit7 to bit0 are treated as the alpha layer.
- 01 bit15 to bit8 are treated as the alpha layer.
- 10 bit23 to bit16 are treated as the alpha layer.
- 11 Reserved

[bit12] L3ID (L3-layer Ignore Data)

This bit specifies whether field A of the display data is valid.

- 0 Blend only if field A of the display data is "1".
- 1 Ignore field A of the display data.

[bit13] L3BP (L3-layer Blend Plane)

This bit selects whether to use a constant value or the alpha layer for the blend ratio.

- 0 Use the L3BR value as the blend ratio.
- 1 Use the L3 layer pixels as the blend ratio.

[bit14] L3BI (L3-layer Blend Increment)

This bit selects whether to add 1/256 when the blend ratio is not 0.

- 0 Calculate the blend ratio as is.
- 1 Add 1/256 when the blend ratio is not 0.

[bit15] L3BS (L3-layer Blend Select)

This bit selects the blend calculation expression.

- 0 Upper image * blend ratio + lower image * (1 – blend ratio)
- 1 Upper image * (1 – blend ratio) + lower image * blend ratio

[bit16] L3BE (L3-layer Blend Enable)

This bit enables blending.

- 0 Overlay using a transparent color
- 1 Overlay using blending

For blending, the blend mode must be specified for L3BE, and field A in the L3 layer display data must be "1". If L3ID is "1", field A of the display data is ignored.

If L3BP is "1" when used, the L3 layer refers to itself as an alpha value. The display data has meaning only if L3AF is "00" with 32 bits/pixel in RGBA format. In such cases, L3BI must be "1". Although L3AF can have settings other than "00", the result would be meaningless to image display because color elements are also used as the blend coefficient.

L4BLD (L4 Blend)

Register Address	$01FD_0194_H$																			
Bit Number	31	30	…	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-					L4BE	L4BS	L4BI	L4BP	L4ID	L4AF		-		L4BR							
R/W	R0					RW	RW	RW	RW	RW	RW		R0W0	R0	RW							
Initial Value	000_0000_0000_0000					0	0	0	0	0	0		0	0	0000_0000							

This register specifies the blend parameters of the L4 layer.

[bit7 to bit0] L4BR (L4-layer Blend Ratio)

These bits set the blend ratio. Basically, the blend ratio is the set value/256.

[bit11 to bit10] L4AF (L4-layer Alpha Field)

These bits select the bit field of the alpha layer.

00 bit7 to bit0 are treated as the alpha layer.
01 bit15 to bit8 are treated as the alpha layer.
10 bit23 to bit16 are treated as the alpha layer.
11 Reserved

[bit12] L4ID (L4-layer Ignore Data)

This bit specifies whether field A of the display data is valid. Use "1" in L4ID because sprite images have no field A.

0 Blend only if field A of the display data is "1".
1 Field A of the display data is ignored. (Recommended value)

[bit13] L4BP (L4-layer Blend Plane)

This bit selects whether to use a constant value or the alpha layer for the blend ratio.

0 Use the L4BR value as the blend ratio.
1 Use the L3 layer pixels as the blend ratio.

[bit14] L4BI (L4-layer Blend Increment)

This bit selects whether to add 1/256 when the blend ratio is not 0.

0 Calculate the blend ratio as is.
1 Add 1/256 when the blend ratio is not 0.

[bit15] L4BS (L4-layer Blend Select)

This bit selects the blend calculation expression.

0 Upper image * blend ratio + lower image * (1 – blend ratio)
1 Upper image * (1 – blend ratio) + lower image * blend ratio

[bit16] L4BE (L4-layer Blend Enable)

This bit enables blending.

0 Overlay using a transparent color

1 Overlay using blending

For blending, the blend mode must be specified for L4BE, and field A in the L4 layer display data must be "1". If L4ID is "1", field A of the display data is ignored.

L0TC (L0-layer Transparency Control)

Register Address	$01FD_00BC_H$															
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L0ZT	L0TC														
R/W	RW	RW														
Initial Value	0	000_0000_0000_0000														

This register sets the transparent color of the L0 layer. In blend mode, the color set by this register is transparent. If L0TC is "0" and L0ZT is "0", color 0 is displayed as black (opaque). The register corresponds to the CTC register of the old model (MB86290-292).

[bit14 to bit0] L0TC (L0-layer Transparent Color)

These bits set the transparent color value (code) of the L0 layer. In indirect color mode (8 bits/pixel), bit7 to bit0 are used.

[bit15] L0ZT (L0-layer Zero Transparency)

This bit sets how color value (code) 0 of the C layer is treated.

0 Code 0 not treated as transparent

1 Code 0 treated as transparent

L2TC (L2-layer Transparency Control)

Register Address	$01FD_00C2_H$															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Bit Field Name	L2ZT	L2TC														
R/W	RW	RW														
Initial Value	0	000_0000_0000_0000														

This register sets the transparent color of the L2 layer. If L2TC is "0" and L2ZT is "0", color 0 is displayed as black (opaque). The register corresponds to the MLTC register of the old model (MB86290-292).

[bit30 to bit16] L2TC (L2-layer Transparent Color)

These bits set the transparent color value (code) of the L2 layer. In indirect color mode (8 bits/pixel), bit7 to bit0 are used.

[bit31] L2ZT (L2-layer Zero Transparency)

This bit sets how color value (code) 0 of the L2 layer is treated.

0 Code 0 not treated as transparent

1 Code 0 treated as transparent

L3TC (L3-layer Transparency Control)

Register Address	$01FD_00C0_H$															
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L3ZT	L3TC														
R/W	RW	RW														
Initial Value	0	000_0000_0000_0000														

This register sets the transparent color of the L3 layer. If L3TC is "0" and L3ZT is "0", color 0 is displayed as black (opaque). The register corresponds to the MRTC register of the old model (MB86290-292).

[bit14 to bit0] L3TC (L3-layer Transparent Color)

These bits set the transparent color value (code) of the L3 layer. In indirect color mode (8 bits/pixel), bit7 to bit0 are used.

[bit15] L3ZT (L3-layer Zero Transparency)

This bit sets how color value (code) 0 of the L3 layer is treated.

0 Code 0 not treated as transparent

1 Code 0 treated as transparent

L0ETC (L0-layer Extend Transparency Control)

Register Address	$01FD_01A0_H$																													
Bit Number	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L0EZT	-					L0TEC																							
R/W	RW	R0					RW																							
Initial Value	0	000_0000					0000_0000_0000_0000_0000_0000																							

This register sets the transparent color of the L0 layer. The register uses a transparent color of 24 bits/pixel. The lower 15 bits in this register are physically the same as those of L0TC. Also, L0ETZ is physically the same as L0TZ. If L0ETC is "0" and L0EZT is "0", color 0 is displayed as black (opaque).

[bit23 to bit0] L0ETC (L0-layer Extend Transparent Color)

These bits set the transparent color value (code) of the L0 layer. In indirect color mode (8 bits/pixel), bit7 to bit0 are used.

[bit31] L0EZT (L0-layer Extend Zero Transparency)

This bit sets how color value (code) 0 of the L0 layer is treated.

0 Code 0 not treated as transparent

1 Code 0 treated as transparent

L1ETC (L1-layer Extend Transparency Control)

Register Address	$01FD_01A4_H$																													
Bit Number	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L1EZT	-					L1ETC																							
R/W	RW	R0					RW																							
Initial Value	0	000_0000					0000_0000_0000_0000_0000_0000																							

This register sets the transparent color of the L1 layer. If L1ETC is "0" and L1EZT is "0", color 0 is displayed as black (opaque). For YCbCr display, no transparent color check is performed, and the color is always processed as non-transparent.

[bit23 to bit0] L1ETC (L1-layer Extend Transparent Color)

These bits set the transparent color value (code) of the L1 layer. In indirect color mode (8 bits/pixel), bit7 to bit0 are used.

[bit31] L1EZT (L1-layer Extend Zero Transparency)

This bit sets how color value (code) 0 of the L1 layer is treated.

0 Code 0 not treated as transparent

1 Code 0 treated as transparent

L2ETC (L2-layer Extend Transparency Control)

Register Address	$01FD_01A8_H$																													
Bit Number	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L2EZT	-					L2ETC																							
R/W	RW	R0					RW																							
Initial Value	0	000_0000					0000_0000_0000_0000_0000_0000																							

This register sets the transparent color of the L2 layer. The register uses a transparent color of 24 bits/pixel. The lower 15 bits in this register are physically the same as those of the L2TC. Also, L2ETZ is physically the same as L2TZ. If L2ETC is "0" and L2EZT is "0", color 0 is displayed as black (opaque).

[bit23 to bit0] L2ETC (L2-layer Extend Transparent Color)

These bits set the transparent color value (code) of the L2 layer. In indirect color mode (8 bits/pixel), bit7 to bit0 are used.

[bit31] L2EZT (L2-layer Extend Zero Transparency)

This bit sets how color value (code) 0 of the L2 layer is treated.

0 Code 0 not treated as transparent

1 Code 0 treated as transparent

L3ETC (L3-layer Extend Transparency Control)

Register Address	$01FD_01AC_H$																													
Bit Number	31	30	29	28	---	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	L3EZT	-					L3ETC																							
R/W	RW	R0					RW																							
Initial Value	0	000_0000					0000_0000_0000_0000_0000_0000																							

This register sets the transparent color of the L3 layer. The register uses a transparent color of 24 bits/pixel. The lower 15 bits in this register are physically the same as those of L3TC. Also, L3ETZ is physically the same as L3TZ. If L3ETC is "0" and L3EZT is "0", color 0 is displayed as black (opaque).

[bit23 to bit0] L3ETC (L3-layer Extend Transparent Color)

These bits set the transparent color value (code) of the L3 layer. In indirect color mode (8 bits/pixel), bit7 to bit0 are used.

[bit31] L3EZT (L3-layer Extend Zero Transparency)

This bit sets how color value (code) 0 of the L3 layer is treated.

0 Code 0 not treated as transparent

1 Code 0 treated as transparent

L4ETC (L4-layer Extend Transparency Control)

Register Address	01FD_01B0$_H$		
Bit Number	31 | 30 29 28 --- 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8 | 7 6 5 4 3 2 1 0
Bit Field Name	L4EZT	-	L4ETC
R/W	RW	R0	RW
Initial Value	0	000_0000	0000_0000_0000_0000_0000_0000

This register sets the transparent color of the L4 layer. If L4ETC is "0" and L4EZT is "0", color 0 is displayed as black (opaque).

[bit23 to bit0] L4ETC (L4-layer Extend Transparent Color)

These bits set the transparent color value of the L4 layer in 24-bit RGB format. The same color value as that of the background color in the sprite window is the general setting.

[bit31] L4EZT (L4-layer Extend Zero Transparency)

This bit sets how color value (code) 0 of the L4 layer is treated.

0 Code 0 not treated as transparent

1 Code 0 treated as transparent

CKC (Chroma Key Control)

Register Address	01FD_00B8$_H$			
Bit Number	31 30 29 28 27 22	21 20 19 18 17 16	15	14 13 12 11 10 9 8 | 7 6 5 4 3 2 1 0
Bit Field Name	-	KCS	KYEN	KYC
R/W	R0	RW	RW	RW
Initial Value	000_0000_0000_0000	0	0	X

[bit14 to bit0] KYC (Key Color)

These bits set the key color used to execute chroma key processing. If indirect color mode (8 bits/pixel) and chroma key mode are set for the L0 layer color, bit7 to bit0 are used.

[bit15] KYEN (chroma-Key Enable)

This bit sets whether to execute chroma key processing.

0 Chroma key processing not executed (The GV pin always outputs H.)

1 Chroma key processing executed

[bit16] KCS (Key Color Select)

This bit selects the display color or L0 layer color as the key color used to execute chroma key processing.

0 Display color used as the key color

1 L0 layer color used as the key color

For details on the chroma key function and GV pin, see "External Synchronization".

L1YCR0 (L1 Layer YC to Red Coefficient 0)

Register Address	$01FD_01E0_H$			
Bit Number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Bit Field Name	-	a12	-	a11
R/W	R0	RW	R0	RW
Bit Number	0_0000	000_0000_0000	0_0000	001_0010_1011

This register defines red component-related parameters for YCbCr/RGB conversion.

[bit10 to bit0] a11

11-bit signed fixed-point value. The lower 8 bits follow the decimal point. The value is a 2's complement representation.

[bit26 to bit16] a12

11-bit signed fixed-point value. The lower 8 bits follow the decimal point. The value is a 2's complement representation.

L1YCR1 (L1 Layer YC to Red Coefficient 1)

Register Address	$01FD_01E4_H$			
Bit Number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Bit Field Name	-	b1	-	a13
R/W	R0	RW	R0	RW
Bit Number	000_0000	1_1111_0000	0_0000	001_1001_1000

This register defines red component-related parameters for YCbCr/RGB conversion.

[bit10 to bit0] a13

11-bit signed fixed-point value. The lower 8 bits follow the decimal point. The value is a 2's complement representation.

[bit24 to bit16] b1

9-bit signed integer. The value is a 2's complement representation.

L1YCG0 (L1 Layer YC to Green Coefficient 0)

Register Address	$01FD_01E8_H$			
Bit Number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Bit Field Name	-	a22	-	a21
R/W	R0	RW	R0	RW
Bit Number	0_0000	111_1001_1100	0_0000	001_0010\1011

This register defines green component-related parameters for YCbCr/RGB conversion.

[bit10 to bit0] a21

11-bit signed fixed-point value. The lower 8 bits follow the decimal point. The value is a 2's complement representation.

[bit26 to bit16] a22

11-bit signed fixed-point value. The lower 8 bits follow the decimal point. The value is a 2's complement representation.

L1YCG1 (L1 Layer YC to Green Coefficient 1)

Register Address	$01FD_01EC_H$			
Bit Number	31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0	
Bit Field Name	-	b2	-	a23
R/W	R0	RW	R0	RW
Bit Number	000_0000	1_1111_0000	0_0000	111_0010_1111

This register defines green component-related parameters for YCbCr/RGB conversion.

[bit10 to bit0] a23

11-bit signed fixed-point value. The lower 8 bits follow the decimal point. The value is a 2's complement representation.

[bit24 to bit16] b2

9-bit signed integer. The value is a 2's complement representation.

L1YCB0 (L1 Layer YC to Blue Coefficient 0)

Register Address	$01FD_01F0_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-					a32											-					a31										
R/W	R0					RW											R0					RW										
Bit Number	0_0000					010_0000_0100											0_0000					001_0010_1011										

This register defines blue component-related parameters for YCbCr/RGB conversion.

[bit10 to bit0] a31

11-bit signed fixed-point value. The lower 8 bits follow the decimal point. The value is a 2's complement representation.

[bit26 to bit16] a32

11-bit signed fixed-point value. The lower 8 bits follow the decimal point. The value is a 2's complement representation.

L1YCB1 (L1 Layer YC to Blue Coefficient 1)

Register Address	$01FD_01F4_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-							b3									-					a33										
R/W	R0							RW									R0					RW										
Bit Number	000_0000							1_1111_0000									0_0000					000_0000_0000										

This register defines blue component-related parameters for YCbCr/RGB conversion.

[bit10 to bit0] a33

11-bit signed fixed-point value. The lower 8 bits follow the decimal point. The value is a 2's complement representation.

[bit24 to bit16] b3

9-bit signed integer. The value is a 2's complement representation.

L0PAL0-255 (L0-layer Palette 0-255)

Register Address	$01FD_0400_H$ to $01FD_07FC_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	A	-							R						-		G						-		B						-	
R/W	RW	R0							RW						R0		RW						R0		RW						R0	
Initial Value	X	000_0000							X						00		X						00		X						00	

This register is the color palette register of the L0 layer. In indirect color mode, the color code in the display frame is used as the palette register number, and the set colors in the register are applied as pixel display colors. This register corresponds to CPALn of the old model (MB86290-292).

[bit7 to bit2] B (Blue)

These bits set the blue color component.

[bit15 to bit10] G (Green)

These bits set the green color component.

[bit23 to bit18] R (Red)

These bits set the red color component.

[bit31] A (Alpha)

This bit specifies whether to blend the layer with lower layers when blending mode is enabled.

0 Do not blend even when blending mode is enabled. Overlay by using a transparent color.

1 Blend.

L1PAL0-255 (L1-layer Palette 0-255)

Register Address	$01FD_0800_H$ to $01FD_0BFC_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	A	-							R						-		G						-		B						-	
R/W	RW	R0							RW						R0		RW						R0		RW						R0	
Initial Value	X	000_0000							X						00		X						00		X						00	

This register is the color palette register of the L1 layer. In indirect color mode, the color code in the display frame is used as the palette register number, and the set colors in the register are applied as pixel display colors. This register corresponds to MBPALn of the old model (MB86290-292).

[bit7 to bit2] B (Blue)

These bits set the blue color component.

[bit15 to bit10] G (Green)

These bits set the green color component.

[bit23 to bit18] R (Red)

These bits set the red color component.

[bit31] A (Alpha)

This bit specifies whether to blend the layer with lower layers when blending mode is enabled.

0 Do not blend even when blending mode is enabled. Overlay by using a transparent color.

1 Blend.

L2PAL0-255 (L2-layer Palette 0-255)

Register Address	$01FD_1000_H$ to $01FD_13FC_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	A	-							R						-		G						-		B						-	
R/W	RW	R0							RW						R0		RW						R0		RW						R0	
Initial Value	X	000_0000							X						00		X						00		X						00	

This register is the color palette register of the L2 layer. In indirect color mode, the color code in the display frame is used as the palette register number, and the set colors in the register are applied as pixel display colors.

[bit7 to bit2] B (Blue)

These bits set the blue color component.

[bit15 to bit10] G (Green)

These bits set the green color component.

[bit23 to bit18] R (Red)

These bits set the red color component.

[bit31] A (Alpha)

This bit specifies whether to blend the layer with lower layers when blending mode is enabled.

0 Do not blend even when blending mode is enabled. Overlay by using a transparent color.

1 Blend.

L1LCT0-255 (L1 Level Conversion Table 0-255)

Register Address	$01FD_2000_H$ to $01FD_23FC_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name			R										G										B									
R/W	R0		RW								R0		RW								R0		RW								R0	
Initial Value	00		X								00		X								00		X								00	

This register converts the color element level of the L1 layer. The conversion is from 256 levels to 256 levels.

[bit9 to bit2] B (Blue)

With the value x set in field B of L1LCT*n*, these bits convert level n of element B to x.

[bit19 to bit12] G (Green)

With the value y set in field G of L1LCT*n*, these bits convert level n of element G to y.

[bit29 to bit22] R (Red)

With the value z set in field R of L1LCT*n*, these bits convert level n of element R to z.

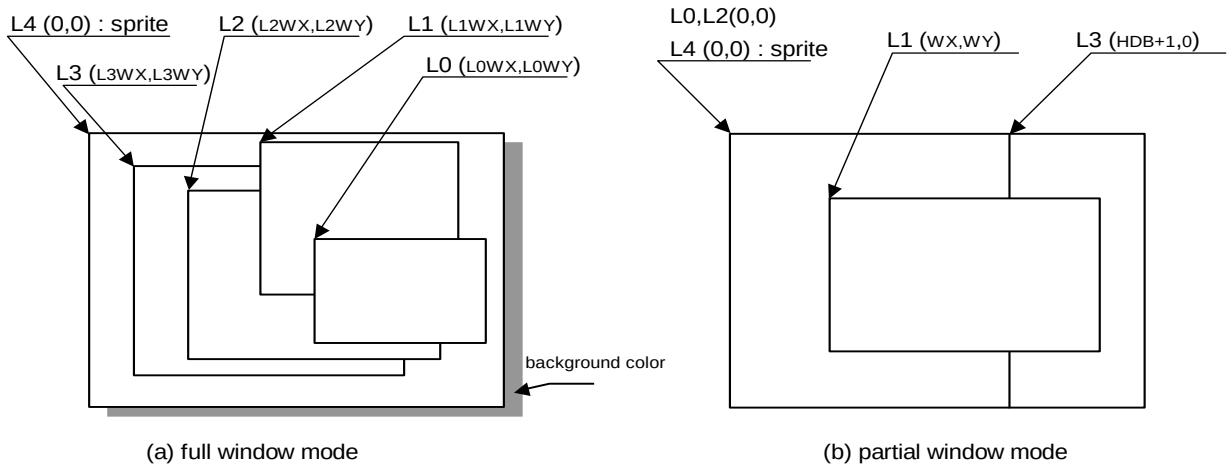
1.6.5 Explanation of Operation

1.6.5.1 Display Function

Window Structure

The window display consists of 4 tiers. An arbitrary sequence of overlaying layers can be set. Compared with the old model (MB86290-296), the 4 higher layers are compatible. The window display consists of the sprite window overlaid on the windows of the old model. The sprite window is treated as L4.

Figure 1-13. Display Window Configuration



(a) full window mode (b) partial window mode

The display layers correspond to those of the old model (MB86290-296) as follows.

Layer in Old Model		GDC Macro	Start Point Coordinates		Width and Height	
			(a) Window	(b) Compatibility	(a) Window	(b) Compatibility
L0 L1 L2 L3	C W ML MR	L0 L1 L2 L3 (alpha)	(L0WX,L0WY) (L1WX,L1WY) (L2WX,L2WY) (L3WX,L3WY)	(0,0) (WX,WY) (0,0) (HDB,0)	(L0WW,L0WH+1) (L1WW,L1WH+1) (L2WW,L2WH+1) (L3WW,L3WH+1)	(HDP+1,VDP+1) (WW,WH+1) (HDB+1,VDP+1) (HDP-HDB,VDP+1)
L4	BL	L4 sprite	(0,0)		(HDP+1,VDP+1)	
L5 (alpha)	BR	-----	N/A			
L6 L7	--- ---	----- -----				

Here, C, W, ML, MR, BL, and BR mean the layers in the old model (MB86290-292). For each of the layers, full/partial window mode can be selected. The new and old display modes are not completely separate but instead coexist, enabling new functions through minor changes of the programs used on the old model.

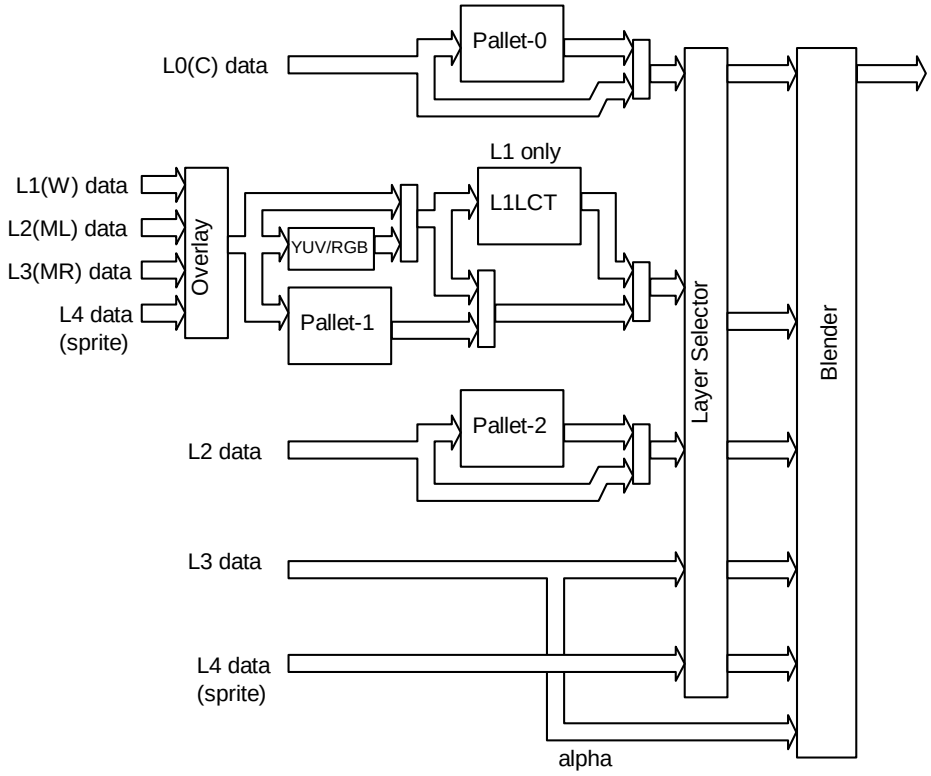
For display at higher resolutions, however, the number of layers and volume of pixel data that can be simultaneously displayed may be limited depending on the data supply capability of VRAM.

Overlaying

Overview

The image data of 4 layers (L0 to L3) and sprites are processed as follows.

Figure 1-14. Display Layer Structure



The fundamental process flow is from the palette to the layer selector to the blender. The palette converts 8-bit color data to RGB format. The layer selector changes the overlay sequence to an arbitrary one. The blender either blends by using the blend coefficient defined for each layer or overlays in accordance with transparent color definitions.

The L0 layer corresponds to the C layer of the old model (MB86290-292).

The L1 layer corresponds to the W layer of the old model (MB86290-292). To ensure downward compatibility with the old model, the layer is overlaid with lower layers before blend calculation. Also, color element level conversion such as gamma correction is available for the L1 layer.

The L2 to L4 layers have 2 processing routes: independent input to the blender, and input to the blender through overlays with L1. The former is selected for processing in extended mode, and the latter is processed in the same way as in the old model (MB86290-292). A route can be selected for each layer.

Overlay Mode

There are 2 modes for overlaying image layers. One is simple priority mode, and the other is blend mode.

In simple priority mode, processing is executed according to the transparent color defined for each layer. If the color is transparent, the value for lower layers is the image value for the next stage. If the color is not transparent, the value of the layer itself is the image value for the next stage.

$D_{view} = D_{new}$ (where D_{new} does not match the transparent color)

$= D_{lower}$ (where D_{new} matches the transparent color)

Note: If the L1 layer is in YCbCr mode, no transparent color check is performed, and the color is always processed as non-transparent.

Blend ratio r, which is defined for each layer, is specified with 8-bit tolerance in blend mode, leading to the following calculation:

$$D_{view} = D_{new} * r + D_{lower} * (1 – r)$$

Blending is enabled on each layer by a mode setting. Furthermore, field A (alpha) of pixels must be "1". The bit to specify to enable blending is the MSB of palette RAM data in the case of 8 bits/pixel, the MSB of one's own data in the case of 16 bits/pixel, and the MSB of the word in the case of 24 bits/pixel. This bit position is that of the ARGB format, and its position would be different in the RGBA format (see "Data Formats").

A new mode has been added with the GDC macro, where field A on the pixel side is ignored and blending is always enabled. The mode is selected using LnID (Ln Ignore bit of blended Data).

LnID = 0 Field A of the data to be blended is enabled.

LnID = 1 Field A of the data to be blended is ignored.

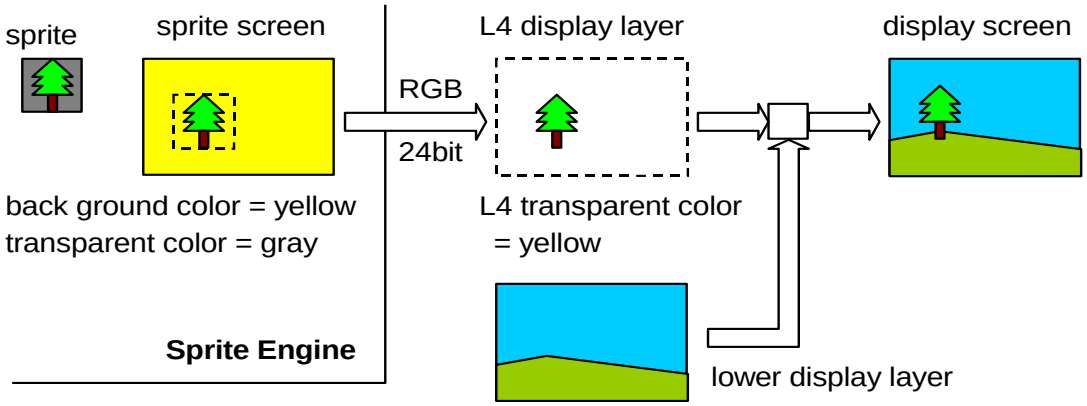
Blend Coefficient Layer

Although the blend coefficient is fixed by layer in normal blend mode, the L3 layer can be used as the blend coefficient layer. In this mode, the blend coefficient can be specified for each pixel, which in turn enables gradation, for example. To use this function, set the L3 layer to the widow display mode of 8 bits/pixel and extended overlay mode.

If the L3 layer is set to 32 bits/pixel, it can be used as an alpha region with 4 sides.

Sprite Synthesis

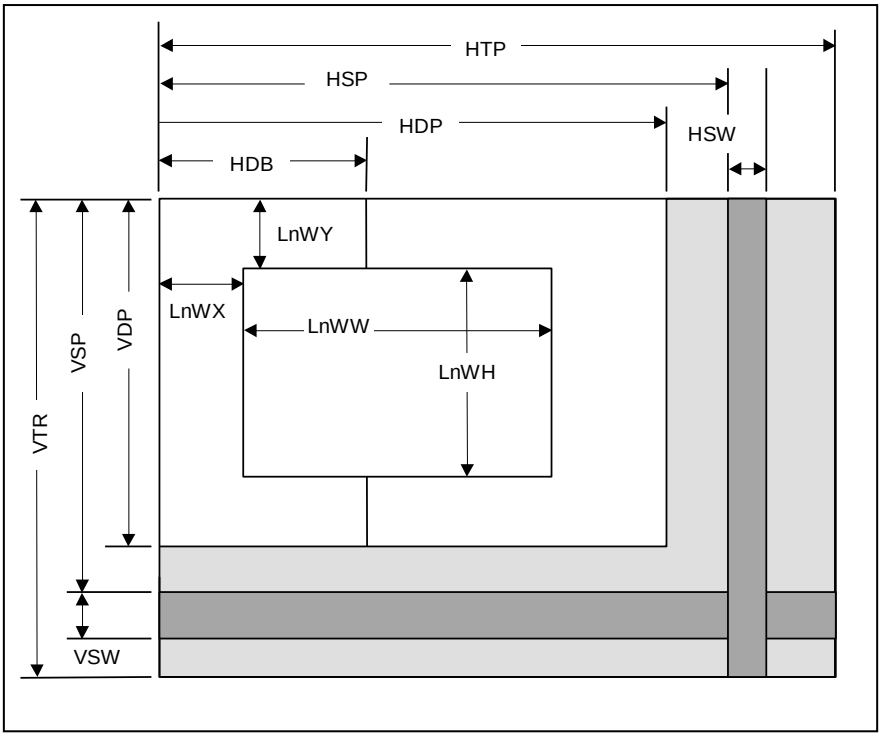
RGB images from the sprite engine are input for the L4 layer. The data structure in a sprite is not individually recognized at the overlay stage in the display controller. It is processed as a single-layer image with RGB = 8:8:8. To selectively synthesize it with a lower layer, set the background color of the sprite as the 24-bit transparent color of the L4 layer. The background color area displays the lower layer. The transparent color of the L4 layer can be specified as a set register value and also be 0x000000 as a transparent color at the same time.



Display Parameters

The parameters shown below define the display area. Each parameter sets the value for each register.

Figure 1-15. Display Parameters



Notes:

The display parameter settings actually differ slightly from the above. For details, see "1.6.5.2 Display Timing".

HTP	Horizontal Total Pixels	Horizontal total pixel count
HSP	Horizontal Synchronize pulse Position	Horizontal synchronous pulse position
HSW	Horizontal Synchronize pulse Width	Horizontal synchronous pulse width
HDP	Horizontal Display Period	Horizontal display period
HDB	Horizontal Display Boundary	Horizontally divided display boundary position
VTR	Vertical Total Raster	Vertical total raster count
VSP	Vertical Synchronize pulse Position	Vertical synchronous pulse position
VSW	Vertical Synchronize pulse Width	Vertical synchronous pulse width
VDP	Vertical Display Period	Vertical display period
LnWX	Layer n Window position X	X coordinate of Ln layer window
LnWY	Layer n Window position Y	Y coordinate of Ln layer window
LnWW	Layer n Window Width	Ln layer window width
LnWH	Layer n Window Height	Ln layer window height

If the window is not divided, HDP equals HDB in the settings, so only the left side is displayed. The set values must have the following magnitude relationships:

$0 < HDB \leq HDP < HSP < HSP+HSW+1 < HTP$

$0 < VDP < VSP < VSP+VSW+1 < VTR$

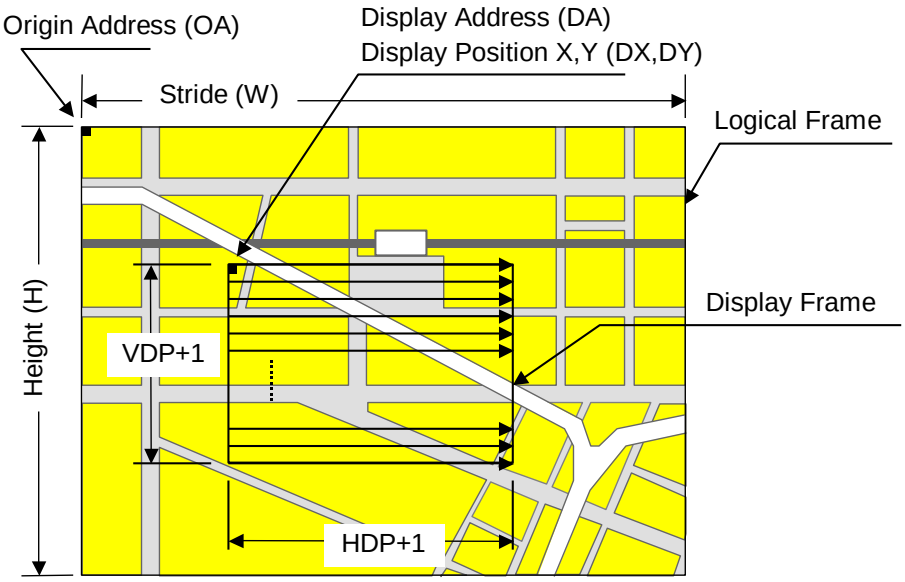
Control of Display Positions

Image data for display is located in a 2-dimensional logical coordinate space (logical image space) in VRAM. The following 4 logical image spaces are the space where images are maintained:

L0 layer, L1 layer, L2 layer, and L3 layer

This section defines the relationship between a logical image space and display positions.

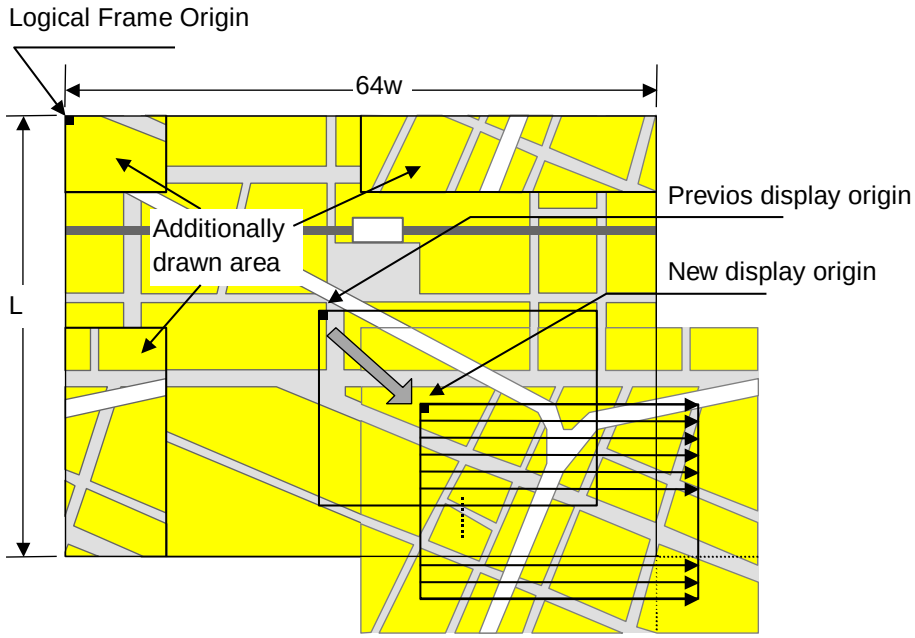
Figure 1-16. Parameters for Setting Display Positions



OA	Origin Address	Logical space origin address Pixel memory address as logical frame origin (top left)
W	Stride	Logical space (memory) width Logical frame width in units of 64 bytes
H	Height	Logical space height Raster (pixel) count of logical space
DA	Display Address	Display origin address Address of display frame origin (top left)
DX DY	Display Position	Display origin coordinates Logical frame space coordinates of display frame origin

Display scans for a logical space are performed as though connected cyclically in both the vertical and horizontal directions. When displaying an image outside the boundaries of a logical space, this function enables smooth scrolling of the image by additionally drawing extended graphics at the part outside the boundaries.

Figure 1-17. Wraparound Processing for Display



The relational expression of the linear addresses (in bytes) corresponding to x and y coordinates in the frame is as follows:

$A(x,y) = x * bpp / 8 + 64wy$ (bpp=8, 16, 32)

The frame must have the display coordinate origin inside it. Specifically, the parameters have the following setting restrictions:

$0 \le DX < w * 64 * 8 / bpp$ (bpp=8, 16, 32)

$0 \le DY < H$

DA indicates the same point as DX and DY in the frame. Thus, the following relationship must be established:

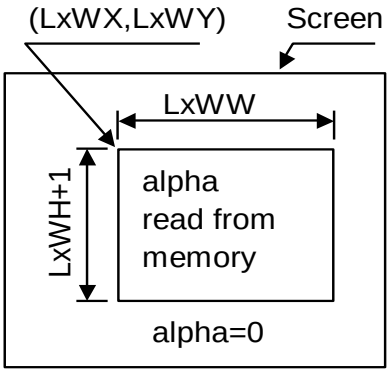
$DA = OA + DX * bpp / 8 + 64w * DY$ (bpp=8, 16, 32)

Note: The L1 layer has no wraparound function.

The L3 layer can be used as an alpha coefficient when display is blended. The alpha value corresponds to the layer at an absolute position on the display screen, and it is used for the calculation of each pixel.

If a window matching the display window has been defined, the alpha value read from memory is applied to the pixels of the whole area of the display window.

If a window smaller than the display window has been defined, a value read from VRAM is applied to the alpha value in the window, while 0 is applied to the alpha value.



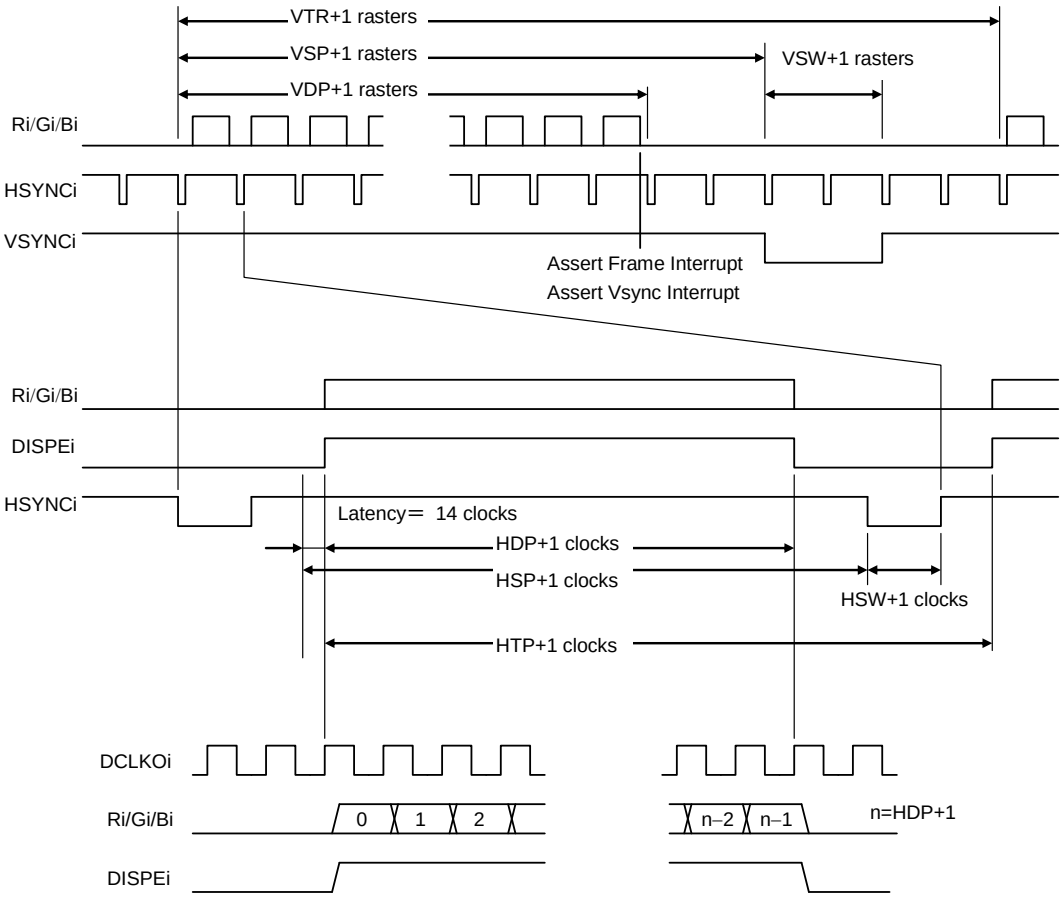
Restrictions

The display digital output pin of Display (ROUT[1:0], GOUT[1:0], GOUT[1:0]) and the data bus input/output pin of MEMC (MEM_ED[14:9]) are exclusively controlled.
For this reason, if the external bus memory interface is selected for MEMC when 16-bit NOR Flash/SRAM is used, RGB digital output (8 bits × 3) cannot be used.

(For details, see the CY91590 Series Hardware Manual.)

1.6.5.2 Display Timing

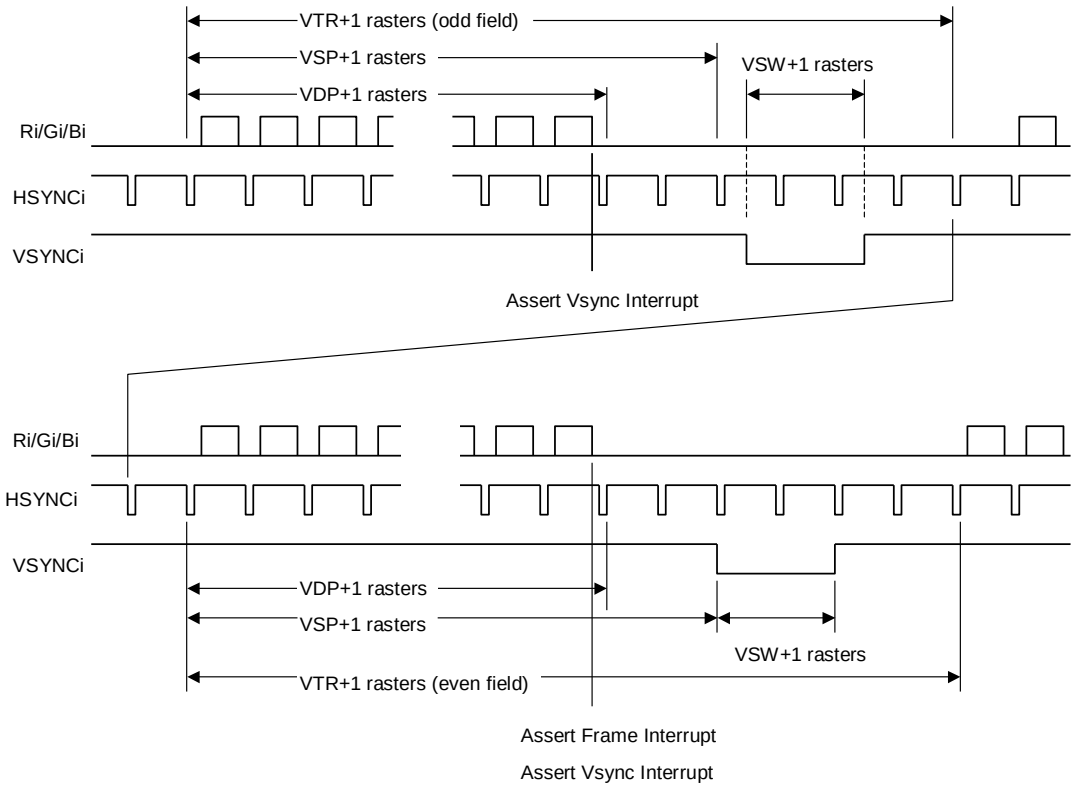
Non-interlaced



In the above figure, VTR, HDP, etc. denote the set values of the corresponding registers.

A VSYNC/frame interrupt is asserted when the display of the last raster ends. When updating display parameters, synchronization with the frame interrupt does not cause any display interference. Since the calculation for the next frame begins immediately after the vertical synchronous pulse is asserted, the parameters must be updated by the time that the calculation begins.

Interlaced Video

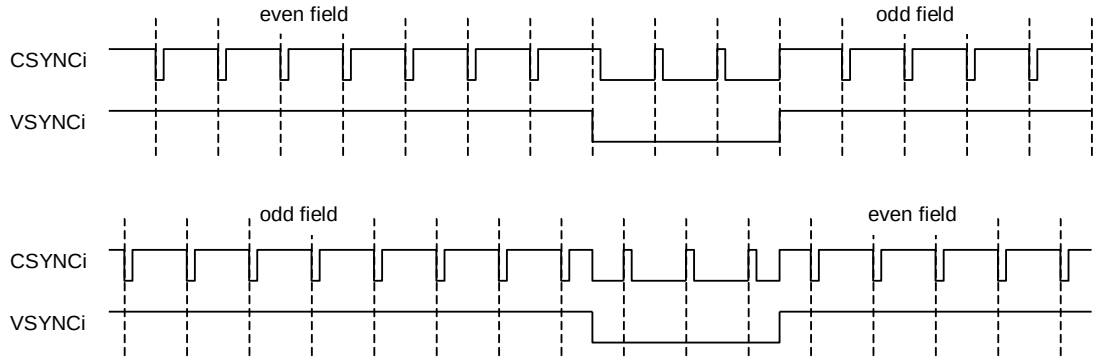


In the above figure, VTR, HDP, etc. denote the set values of the corresponding registers.

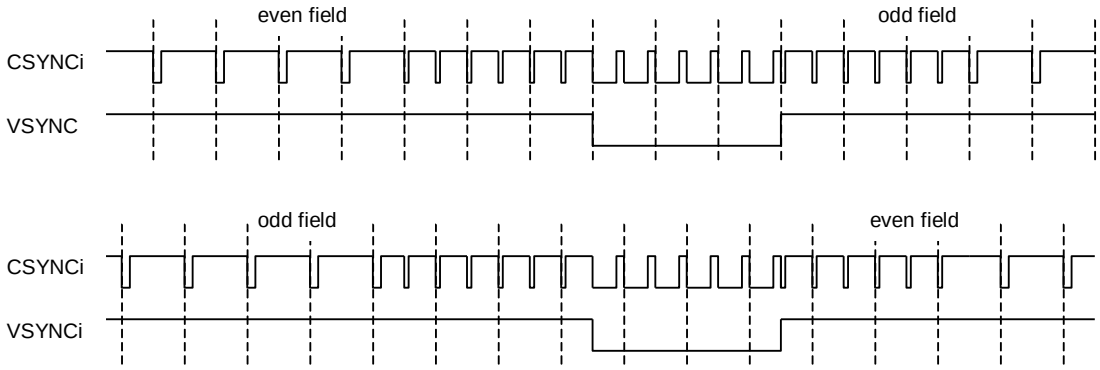
Interlace mode also has the same timing of operations. Only the output image data is different.

Composite Synchronous Signal

The following waveform is produced by CSYNC signal output when the EEQ bit in the DCM register is "0".



The following waveform is produced by the insertion of an equalizing pulse into the CSYNC signal when the EEQ bit in the DCM register is "1".



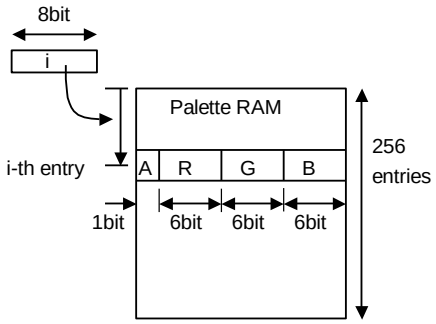
The equalizing pulse is inserted at the beginning of the vertical blanking period. The pulse is also inserted 3 times after the vertical synchronous period has ended.

1.6.5.3 Data Formats

Display handles the data formats described below.

Indirect Color (8 Bits/Pixel)

This is the index of palette RAM. Palette RAM is used for conversion to the image data of RGB, each of which is 6 bits, and display of the data. The available palette differs depending on the layer.



If the pixel value is i, the i-th entry of the palette determines the RGB output value.

The resolution of each color element of the palette is 6 bits. Display output has the basic resolution of RGB, each of which is 8 bits, and color elements of the palette are shifted 2 bits to the MSB side for the display output.

Direct Color (16 Bits/Pixel)

The RGB level is expressed with 5 bits or 6 bits. Display output has the basic resolution of RGB, each of which is 8 bits, and each color element value is shifted to the MSB side for the display output.

There are 3 formats.

Format	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARGB	A	R					G					B				
RGBA	R					G					B					A
RGB565	R					G						B				

The A bit determines whether to enable or disable display blending.

Only G has a resolution high enough for RGB565. Field A is not included, and A is always "1".

Direct Color (24 Bits/Pixel)

The RGB level is expressed with 8 bits each. 1 pixel is actually 32 bits.

There are 2 formats: ARGB and RGBA.

Format	31	30	...	25	24	23	22	...	17	16	15	14	...	9	8	7	6	...	1	0
ARGB	A					R					G					B				
RGBA	R					G					B					A				

In the ARGB format, the MSB of field A determines whether to enable or disable blending.

In the RGBA format, the 8-bit integer value of A=0 or A≠08 determines whether to enable or disable blending.

YCbCr Color (16 Bits/Pixel)

This is image data of the YcbCr=4:2:2 format. The displayed data has been converted by the calculation circuit to image data of RGB, each of which is 8 bits. 2 pixels of RGB, which is 24 bits/pixel, are expressed with 32 bits. Accordingly, the image data is treated as 16 bits/pixel.

Format	31	30	...	25	24	23	22	...	17	16	15	14	...	9	8	7	6	...	1	0
YCbCr	Y					Cr					Y					Cb				

Alpha Coefficient (8 Bits/Pixel)

The coefficient of display blending is maintained. If the value is t, t/256 is expressed as a binary fraction. Display blending performs the following calculation for each color element of each pixel:

$$c' = c0 * t / 256 + c1 (1–t/256)$$

Alpha Coefficient (16 Bits/Pixel)

The coefficient of display blending is maintained in the packed format of an alpha value for 2 regions.

Format	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Format	A1								A0							

The LnAF parameter is used to select the side.

Alpha Coefficient (24 Bits/Pixel)

The coefficient of display blending is maintained in the packed format of an alpha value for 3 regions.

Format	31	30	...	25	24	23	22	...	17	16	15	14	...	9	8	7	6	...	1	0
Alpha						A2					A1					A0				

The LnAF parameter is used to select the side.

Layer Dependency

The following table lists the display colors available in each layer.

Layer	Compatibility Mode	Extended Mode
L0	Direct (16, 24), Indirect (P0)	Direct (16, 24), Indirect (P0)
L1	Direct (16, 24), Indirect (P1), YCbCr	Direct (16, 24), Indirect (P1), YCbCr
L2	Direct (16, 24), Indirect (P1)	Direct (16, 24), Indirect (P2)
L3	Direct (16, 24), Indirect (P1)	Direct (16, 24), Alpha (8, 32)
L4	Sprite (24)	

In the above table, Pn means the corresponding palette RAM. The 3 palettes used are as follows:

- Palette 0 (P0): Palette corresponding to the C layer palette of the old model (MB86290-292). It is the palette of L0.
- Palette 1 (P1): Palette corresponding to the MB layer palette of the old model (MB86290-292). In compatibility mode, it is a palette common to L1 to L3. In extended mode, it is a palette dedicated to L1.
- Palette 2 (P2): Palette dedicated to L2. It is available only in extended mode.

1.6.5.4 Display Scan Control

Corresponding Display

This section shows typical usable display resolutions and their synchronous signal frequencies. The setting of the division rate of the display reference clock determines the pixel clock frequency. Here, the display reference clock is either the built-in PLL-based clock (108 MHz or 81 MHz) or the clock supplied to the DCLKI input pin. The following table lists various setting examples.

Table 1-3. Resolutions and Display Frequencies

Resolution	Reference Frequency/Division Rate = Pixel Frequency	Horizontal Total Pixel Count	Horizontal Frequency	Vertical Total Raster Count	Vertical Frequency
320 × 240	108/17 = 6.35 MHz	403	15.75 kHz	263	59.9 Hz
400 × 240	108/13 = 8.31 MHz	538	15.73 kHz	263	59.8 Hz
480 × 240	108/11 = 9.82 MHz	624	15.76 kHz	263	59.8 Hz
640 × 480	25 MHz (DCLKI)	800	31.25 kHz	525	59.5 Hz
640 × 480	108/4 = 27 MHz	858	31.47 kHz	525	59.5 Hz
800 × 480	108/3 = 36 MHz	1143	31.5 kHz	525	60.0 Hz

Pixel frequency = reference frequency / division rate = reference frequency / (SC+1)

Horizontal frequency = pixel frequency / horizontal total pixel count = pixel frequency / (HTP+1)

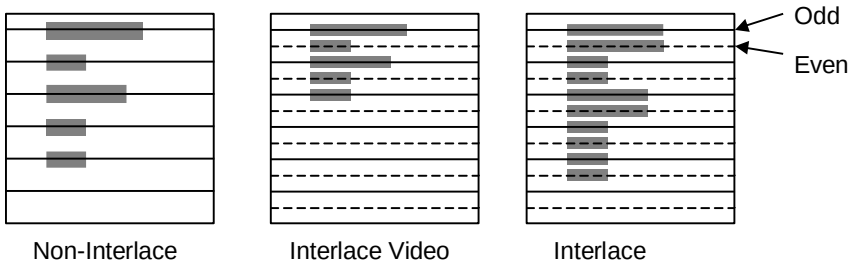
Vertical frequency = horizontal frequency / vertical total raster count = horizontal frequency / (VTR+1)

Interlaced Display

The GDC macro is capable of not only non-interlaced display but also interlaced display.

If the synchronous mode setting of the DCM0 register is interlaced video mode (11), images in VRAM are output alternately in odd and even rasters to each field, and 1 screen is displayed by 1 frame (odd + even fields). If the synchronous mode setting of the DCM0 register is interlace mode (01), images in VRAM are output in raster order. The same image data is output to odd and even fields. Thus, the number of rasters on the screen is half of that of interlaced video mode. However, unlike non-interlace mode, there is a distinction between odd and even fields, depending on the phase relationship between the horizontal and vertical synchronous signals.

Figure 1-18. Display Differences among Synchronous Modes

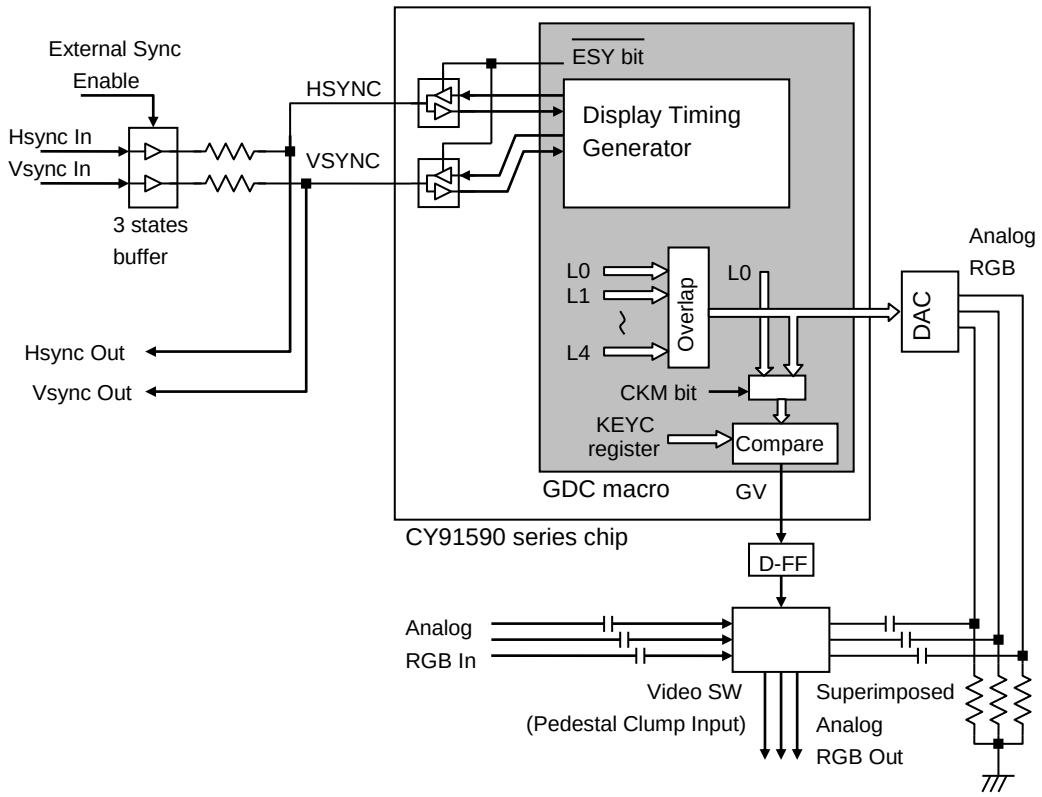


1.6.5.5 External Synchronization

A display scan can be performed by synchronizing horizontal/vertical synchronous signals from the outside.

Once external synchronous mode is selected using the registers, the GDC macro samples the HSYNC signal to display the synchronized external video signal. The built-in PLL clock or DCLKI signal input can be selected as the sampling clock. Superimposing can be performed in chroma key processing. The following diagram shows an example of an external synchronous circuit.

Figure 1-19. Example of External Synchronization



The ESY bit in the DCM0 (or DCM1) register sets external synchronous mode. Once external synchronous mode is set, the HSYNC and VSYNC pins of the GDC macro enter input mode. After that, use the 3-state buffer to supply the synchronous signal from the outside. To clear external synchronous mode, terminate the synchronous input from the outside, and then set the internal ESY bit in the GDC macro to OFF.

Be sure not to set the external synchronous signal buffer to ON with the synchronous output of the GDC macro being left ON. Follow the above-mentioned procedure for their control to prevent them from being ON at the same time.

For external synchronization with the display clock based on the built-in PLL, the GDC macro extends the clock period and fits the clock phase with the horizontal synchronous signal phase immediately after horizontal synchronous pulse input. Caution is necessary at this time.

If an LVDS or other high-speed serial transmitter is connected to the digital RGB output, the PLL built into the high-speed serial transmitter temporarily becomes unstable. Therefore, do not perform any built-in PLL-based operation of external synchronization with a high-speed serial transmitter.

To perform superimposing, switch the external video signal with the GDC output by using the GV output signal. The circuit example in Figure 1-19 shows the switching circuit at the lower right. The GV output determines the output value by using the chroma key function. The CKC register controls the chroma key function.

Chroma key disabled: GV=1

Chroma key enabled in a blanking period: GV=0

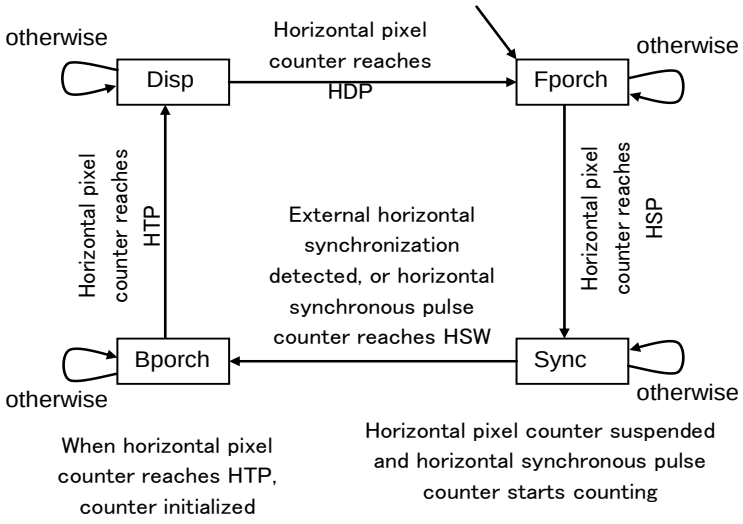
Chroma key enabled, and output pixels do not match the chroma key in the display period: GV=1

Chroma key enabled, and output pixels match the chroma key in the display period: GV=0

Superimposing when GV=0 is done through control of the video switch so that the external video signal is selected.

The following figure shows horizontal synchronization controlled by state transitions.

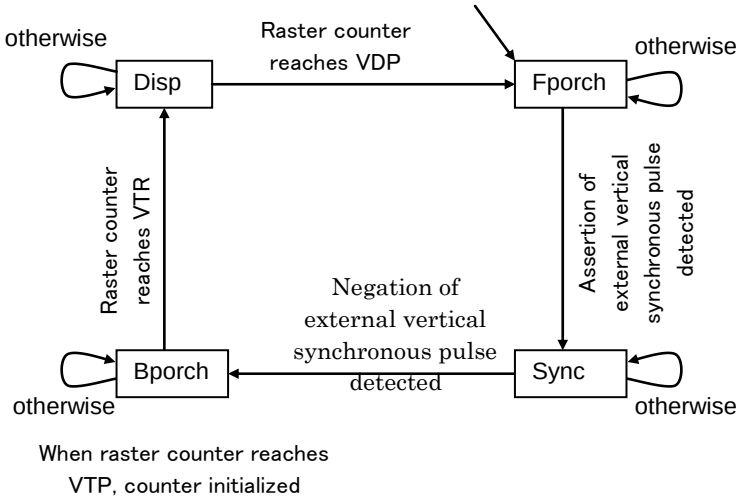
Figure 1-20. State Transition Diagram of Horizontal Synchronization



State transitions are mainly controlled with the count values of the horizontal pixel counter. The display period is equivalent to the Disp state. The display period ends when the horizontal pixel counter value reaches the set value of the HDP register, resulting in a transition from the Disp state to the Fporch state (front porch). In the Fporch state, when the count value of the horizontal pixel counter reaches the set value of the HSP register, the circuit enters the Sync state. In this state, it waits for the external horizontal synchronous signal. Upon detecting the negating edge of the external horizontal synchronous pulse, the GDC macro performs synchronization. There is a transition to the Bporch state (back porch) when the external horizontal synchronous signal is detected. In the Sync state, the horizontal pixel counter is stopped. In its place, the horizontal synchronous pulse counter starts incrementing from 0. There is a transition to the Bporch state when this count value reaches the set value of the HSW register without the detection of the external horizontal synchronous signal. In the Bporch state, when the horizontal pixel counter value reaches the set value of the HTP register, the horizontal pixel counter is reset, which is followed by a transition to the Disp state, starting the display of the next raster.

The following figure shows vertical synchronization controlled by state transitions too.

Figure 1-21. State Transition Diagram of Vertical Synchronization



Basically, state transitions for vertical synchronization are controlled with the count values of the raster counter. The display period is equivalent to the Disp state. The display period ends when the raster counter value reaches the set value of the VDP register, resulting in a transition from the Disp state to the Fporch state (front porch). In the Fporch state, the circuit waits for the external vertical synchronous pulse to be asserted. There is a transition to the Sync state when the assertion of the external vertical synchronous pulse is detected. In this state, the circuit waits for the negation of the external vertical synchronous signal. There is a transition to the Bporch state (back porch) when the negation is detected. In the Bporch state, when the raster counter value reaches the set value of the VTR register, the raster counter is reset, which is followed by a transition to the Disp state, starting the display of the next raster.

1.6.5.6 Variable Parameters for L1 Layer YCbCr/RGB Conversion

Data in the YCbCr format on the L1 layer can be converted to RGB for display, and the conversion parameters are variable parameters. YCbCr data conversion uses the following expressions:

$R = a_{11} * Y + a_{12} * (Cb-128) + a_{13} * (Cr-128) + b_1$

$G = a_{21} * Y + a_{22} * (Cb-128) + a_{23} * (Cr-128) + b_2$

$B = a_{31} * Y + a_{32} * (Cb-128) + a_{33} * (Cr-128) + b_3$

a_{ij} ---- 11-bit signed real (lower 8 bits are fraction, 2's complement)

b_i ----- 9-bit signed integer (2's complement)

They can be expressed as the following matrix calculation:

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \mathbf{A} \begin{pmatrix} Y \\ Cb-128 \\ Cr-128 \end{pmatrix} + \boldsymbol{b} \quad \text{where} \quad \mathbf{A} = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}, \boldsymbol{b} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \end{pmatrix}$$

These parameters are set to the following registers:

L1YCR0 (a_{12}, a_{11}), L1YCR1 (b_1, a_{13})

L1YCG0 (a_{22}, a_{21}), L1YCG1 (b_2, a_{23})

L1YCB0 (a_{32}, a_{31}), L1YCB1 (b_3, a_{33})

As with the old model (MB86290-296), this conversion uses the initial values set in the registers immediately after a reset. The register values immediately after a reset are as follows:

a_{11} = 0x12b (299/256), a_{12} = 0x0, a_{13} = 0x198 (408/256)

a_{21} = 0x12b (299/256), a_{22} = 0x79c (-100/256), a_{23} = 0x72f (-209/256)

a_{31} = 0x12b (299/256), a_{32} = 0x204 (516/256), a_{33} = 0x0

b_1= b_2= b_3= 0x1f0 (-16)

The brightness, contrast, hue, and color saturation can be controlled with changes to these conversion parameters.

- Adding a constant value to $\boldsymbol{b}$ increases the brightness.
- Multiplying a scalar constant that is larger than 1 by $\mathbf{A}$ increases the contrast.
- A 2-dimensional rotation of Cb-128 and Cr-128 changes the hue.
- The color saturation level is the color intensity relative to the brightness.

The following expressions are new conversion coefficients reflecting these variations.

$$\mathbf{A} = c_1 \mathbf{A}_0 \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(t) & \sin(t) \\ 0 & -\sin(t) & \cos(t) \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & c_2 & 0 \\ 0 & 0 & c_2 \end{pmatrix} = \mathbf{A}_0 \begin{pmatrix} c_1 & 0 & 0 \\ 0 & \cos(t)c_1c_2 & \sin(t)c_1c_2 \\ 0 & -\sin(t)c_1c_2 & \cos(t)c_1c_2 \end{pmatrix}$$

$$\boldsymbol{b} = \boldsymbol{b}_0 + \begin{pmatrix} c_3 \\ c_3 \\ c_3 \end{pmatrix}$$

$\mathbf{A}_0$, $\boldsymbol{b}_0$: Initial values

c_1 : Contrast parameter. The standard value is 1. For example, 1.2 means slightly strong contrast.

c_2 : Color saturation level parameter. The standard value is 1, and 0 means monochrome images.

c_3 : Brightness parameter. The standard value is 0.

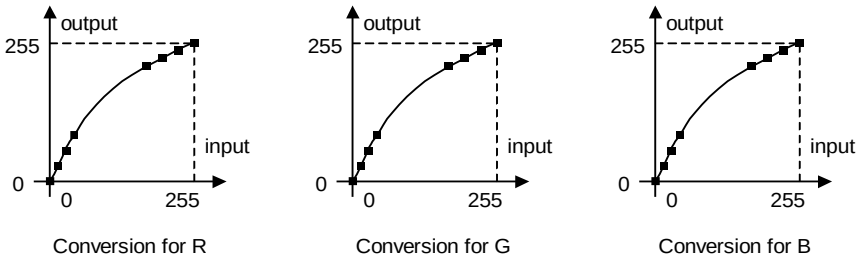
t : Hue rotation parameter. The standard value is the 0 level.

Note: Clip newly calculated a_{ij} and b_i values so that they are in the valid numerical value range for the corresponding registers and then set them.

1.6.5.7 L1 Color Element Level Conversion

Level conversion for the color elements RGB is available for an L1 layer image. This conversion can adjust the brightness (gamma correction) of the entire image.

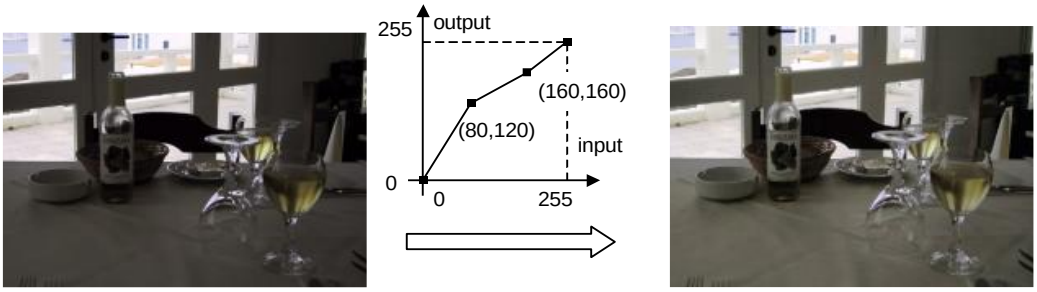
For each element, data in the RGB format is converted from 256 levels to 256 levels, using the tables defined in RAM. Data in the YCbCr format is first converted to the RGB format before level conversion is performed on it.



To enable level conversion, set the L1LCE bit in the L1EM register to "1".

The conversion table settings are L1LCT0 to 255 (0x2000 to 0x23fc).

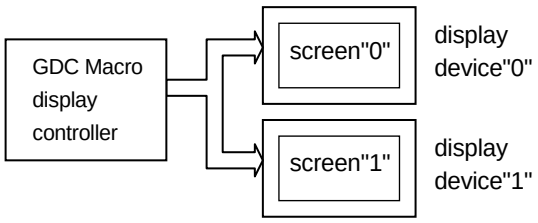
The following example shows areas converted from dark to slightly bright in an image.



1.6.5.8 Dual Display

Overview

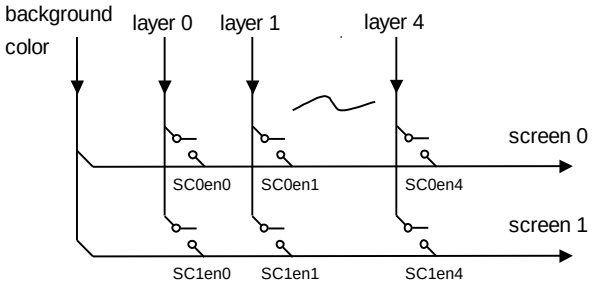
Compatible with the MB86276, the dual display function enables the display of different contents on 2 display devices. The function multiplexes and outputs 2 screens. It can control which layer is output to which screen. In the following diagram, screen 0 is assumed to be output to display device 0, and screen 1 is assumed to be output to display device 1.



Layer Destination Control

Any layer can be included in 2 screens or just 1 screen. The layers that are not included in any screen are treated as transparent. The background color is displayed when all outputs are OFF.

This destination control can be considered in the following virtual cross-point switch.



The MDen (multi display enable) bit is in the MDC (multi display control) register, and it enables this dual display operation.

The SC0en (screen "0" enable) field in the MDC register specifies which layer is output to screen 0.

The SC1en (screen "1" enable) field in the MDC register specifies which layer is output to screen 1.

bit0 ---- L0 is included

bit1 ---- L1 is included

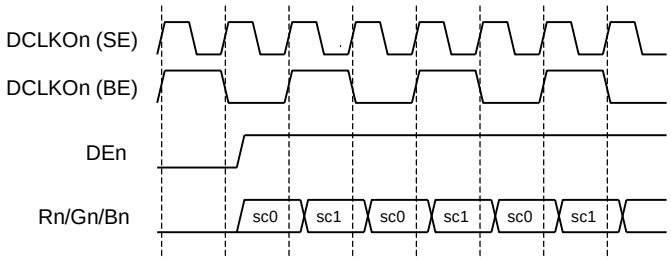
bit2 ---- L2 is included

bit3 ---- L3 is included

bit4 ---- L4 is included

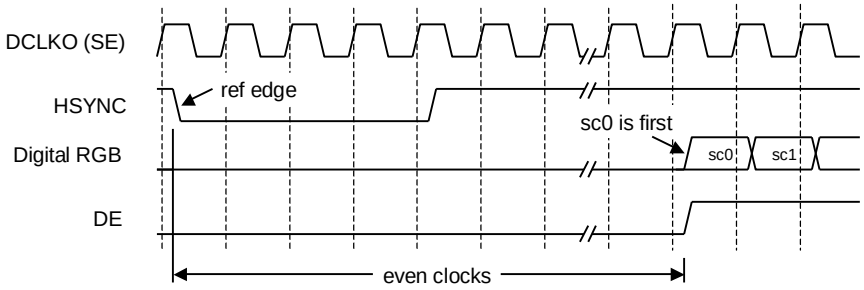
Output Signal Control

The 2 output modes are parallel mode and multiplex mode, but the GDC macro uses only multiplex mode. 2 screens are multiplexed and output to the RGB output.



There are 2 modes for clock output.

- In BE (bi-edge) DCLKO mode, 2 output phases can be identified by both edges.
- In SE (single-edge) DCLKO mode, 2 output phases are identified by HSYNCn or DEn.

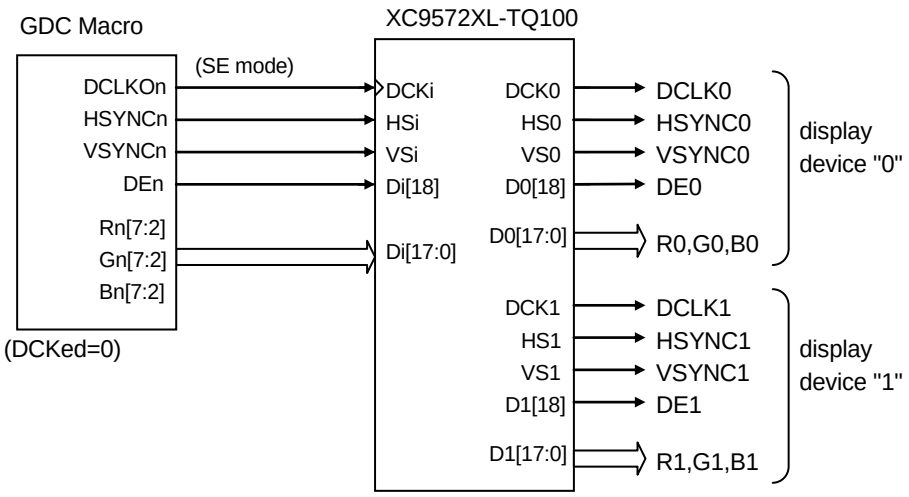


The DCKed (clock edge) bit in the DCM3 register selects the DCLKO mode. SE (single edge) mode is used when DCKed is "0", and BE (bi-edge) mode is used when DCKed is "1".

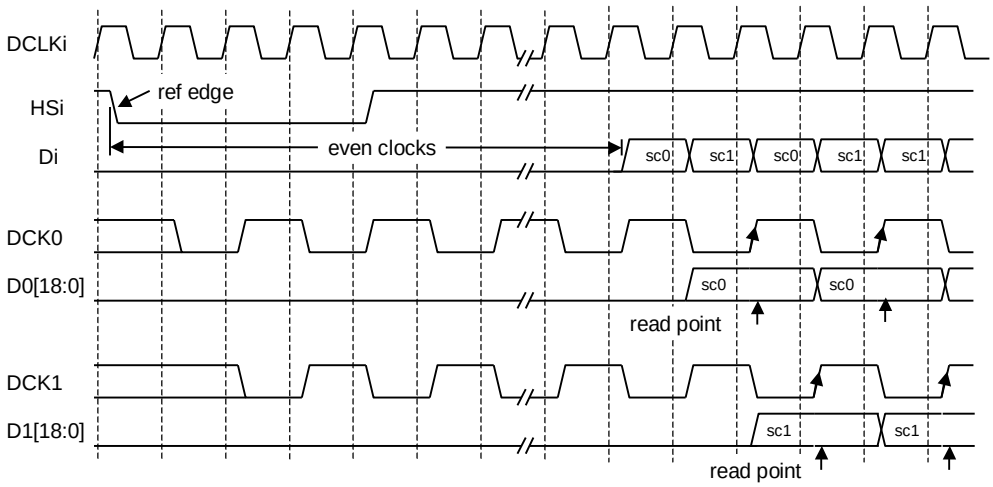
Examples of Output Circuits

1. SE mode

   In this example, a lower-price CPLD separates the DCLKOn clock and DEn output used in SE mode.

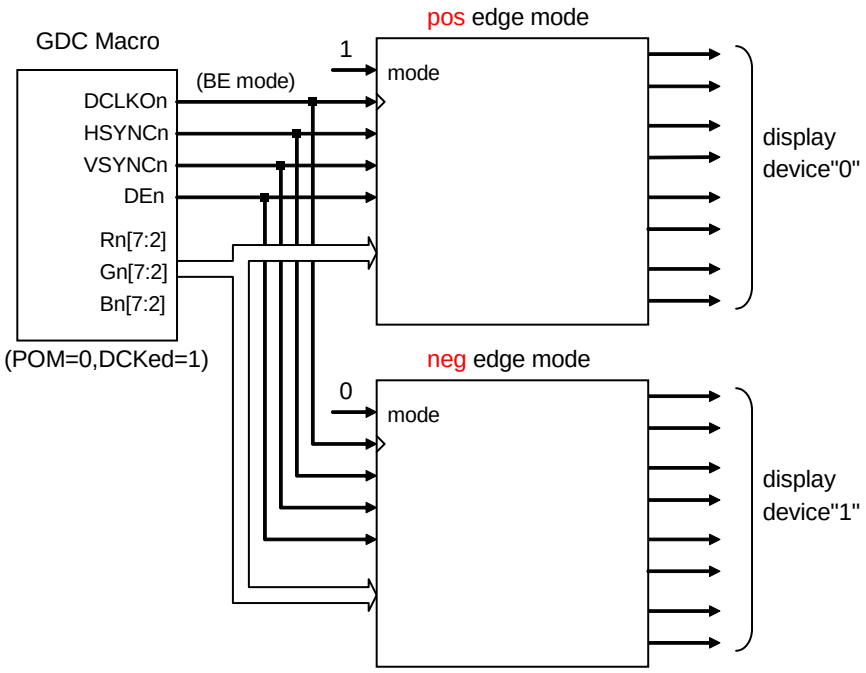



```
module XC9572XL ( DCKi, HSi,VSi,Di,    DCK0,HS0,VS0,D0,    DCK1,HS1,VS1,D1 );
      input DCKi,HSi,VSi;
      input[18:0] Di;
      output DCK0,HS0,VS0, DCK1,HS1,VS1;
      output[18:0] D0,D1;
      reg HS0,HS1, VS0,VS1, DCK0,DCK1;
      reg[18:0] D0,D1;

      always @(posedge DCKi) begin
            HS0 <= HSi; HS1 <= HS0;
            VS0 <= VSi; VS1 <= VS0;
            DCK0 <= (HS0&!HSi)? 0: !DCK0; // sync to ref edge : flip
            DCK1 <= DCK0;
            if(DCK0) D0 <= Di;
            if(DCK1) D1 <= Di;
      end

endmodule
```



2. BE mode

For the device where a positive edge or negative edge can be selected as the active edge of the clock, the DCLKO clock in BE mode enables data for 2 screens to be individually received .



Display Clock and Timing

Dual display requires the supplied frequency to be twice that of normal. A display clock of 25 MHz is used for VGA display, but dual display mode requires use of a display clock of 50 MHz. The same applies to the timing setting of the HTP and other parameters except SC (scaling ratio). Determine the maximum resolution that can be used with the maximum display clock frequency. The SVGA (800 x 600) resolution can be used when the display clock is 80 MHz.

Restrictions

- The 2 display devices must have the same scan speed and resolution with synchronous signals common to them.
- Dual display and external synchronization cannot be used simultaneously.
- BE mode cannot be used when DCKIN clock input is used as the display clock.

1.6.5.9 Interrupts

The following interrupt events are generated in the display controller. They have status flags corresponding to the INTST register of MCNT, which can manage the synchronization with each event.

Event		MCNT/INTST support
VSYNC	Occurs at the beginning of the vertical blanking period.	INT2 (bit2)
FSYNC	Occurs at the same time as VSYNC, during non-interlaced display. During interlaced display or interlaced video display, this event occurs at the beginning of the vertical blanking period immediately after the end of an even-number field.	INT3 (bit3)
SYNCERR	Occurs when HSYNC is disrupted in external synchronous mode.	INT4 (bit4)
RUPDATE	Occurs at the completion of an update using register synchronous update mode.	INT5 (bit5)

1.7 Capture Controller (CAPTURE)

1.7.1 Overview

The capture controller captures video signals from the outside as image data. The image data can be synthesized with other image data and then displayed. The image data can also be used as static images for the target of processing by the host CPU.

1.7.2 Features

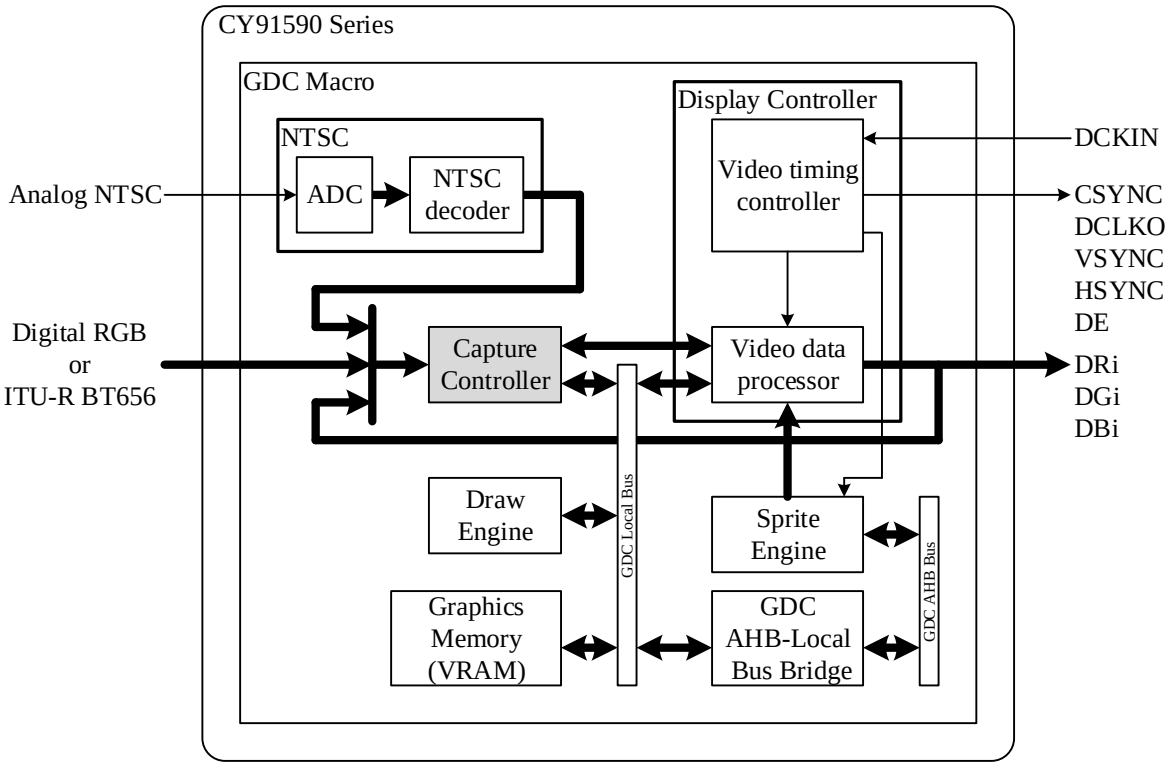
Video input

- The input format is ITU RBT656 or RGB666.
- After being stored in graphics memory (VRAM), the video format is synchronized with a display scan and then displayed on the screen.

Scaling

- Magnification factors of 1 to 2 can be used. PAL or NTSC images can be displayed in wide screen.
- Shrink factors of 1 to 1/32 can be used.
- For picture-in-picture, a drawn picture and a video image can be displayed on the same screen.

1.7.3 Positioning in the Overall GDC Macro Configuration



Notes:

- Analog NTSC corresponds to the external pin VIN.
- Digital RGB corresponds to the external pins PA2 to PA7, PB2 to PB7, and PC2 to PC7.
- ITU-R.BT656 corresponds to the external pins PA2 to PA7, PB2, and PB3.
- DCLKI corresponds to the external pin DCKIN.
- CSYNC corresponds to the external pin PG3.
- DCLKO corresponds to the external pin PG4.
- VSYNC corresponds to the external pin PG5.
- HSYNC corresponds to the external pin PG6.
- DE corresponds to the external pin PG7.
- DR*i* corresponds to the external pins P011, P012, and PD2 to PD7.
- DG*i* corresponds to the external pins P013, P014, and PE2 to PE7.
- DB*i* corresponds to the external pins P015, P016, and PF2 to PF7.

1.7.4 Registers

1.7.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (0040_0000H) is added for access from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  RX: The read value is always undefined.
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.7.4.2 Register List

CaptureBaseAddress = 01FD_8000$_H$

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
000$_H$	VCM (Video Capture Mode)																															
	VIE	VIS		VICE			CM					VI																		NRGB	VS	
004$_H$	CSC (Capture SCale)																															
	VSCI				VSCF												HSCI				HSCF											
008$_H$	VCS (Video Capture Status 0)																															
																										CE0						
010$_H$	CBM0 (Capture Buffer Mode 0)																															
	OO	SBUF	CRGB						CBW (stride)								BLEN	C24	RGBA	CSW												CBST
320$_H$	CBM1 (Capture Buffer Mode 1)																															
																				C565						Hrev	Vrev	Rrot				TBUF
014$_H$	CBA0 (Capture Buffer Address 0)																															
018$_H$	CBA1 (Capture Buffer Address 1)																															
324$_H$	CBA2 (Capture Buffer Address 2)																															
328$_H$	CBOFS (Capture Buffer Offset)																															
01C$_H$	CISTR (Capture Image Start)																															
					CIVSTR																CIHSTR											
020$_H$	CIEND (Capture Image End)																															
					CIVEND																CIHEND											

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
028_H	CHP (Capture Horizontal Pixel)																															
																							CHP								
040_H	CLPF (Capture Low Pass Filter)																															
					CVLPF								CHLPF																			
048_H	CMSS (Capture Magnify Source Size)																															
					CMSHP																CMSVL											
$04C_H$	CMDS (Capture Magnify Display Size)																															
					CMDHP																CMDVL											
080_H	RGBHC (RGB input HSYNC Cycle)																															
																			RGBHC												
084_H	RGBHEN (RGB input Horizontal Enable Area)																															
					RGBHST															RGBHEN												
088_H	RGBVEN (RGB input Vertical Enable Area)																															
			RGBVST_T					RGBVST												RGBVEN												
$08C_H$	VIN_VSAMP																															
																							VJITFLT								FLDREV
090_H	RGBS (RGB input SYNC)																															
																RM														HP	VP
$0C0_H$	RGBCMY (RGB Color convert Matrix Y coefficient)																															
	a11												a12										a13									
$0C4_H$	RGBCMCb (RGB Color convert Matrix Cb coefficient)																															
	a21												a22										a23									
$0C8_H$	RGBCMCr (RGB Color convert Matrix Cr coefficient)																															
	a31												a32										a33									
$0CC_H$	RGBCMb (RGB Color convert Matrix b coefficient)																															
		b1											b2											b3								

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
178_H	CINT																															
																															VS	
$17c_H$	CIMSK																															
																															VS	
180_H	SYNC_err																															
							VSL_err	VSS_err								VS_err							HSL_err	HSS_err								HS_err
184_H	SYNC_error_MSK																															
							MVSL_err	MVSS_err								MVS_err							MHSL_err	MHSS_err								MHS_err
300_H																					CVCNT											
4000_H	CDCN (Capture Data Count for NTSC)																															
				BDCN																VDCN												
4004_H	CDCP (Capture Data Count for PAL)																															
				BDCP																VDCP												
4014_H	MDS																															
																											YCMIM					
$404C_H$	VINLC																															
																			VIN_LINE_NO_kep													
4054_H	VHSLS (Video Input HSYNC Long/Short)																															
			VIN_HS_LONG																VIN_HS_SHORT														
4058_H	VHSDC (Video HSYNC Down Count)																															
																									VIN_HDOWN_CNT							
4060_H	VVSLS (Video VSYNC long/short)																															
				VIN_HS_LONG																VIN_HS_SHORT													
4064_H	VVSDC (Video VSYNC Down Count)																															
																									VIN_VDOWN_CNT							

1.7.4.3 Register Details

General

VCM (Video Capture Mode)

Register Address	01FD_8000$_H$																			
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	12	11	10	4	3	2	1	0
Bit Field Name	VIE	VIS	-	VICE	-	CM			-			VI	-						NRGB	VS	-
R/W	RW	RW	RX	RW	RX	RW			RX			RW	RX		RW0	RX			RW	RW	RX
Initial Value	0	0	X	0	X	00			X			0	X		0	X			0	0	X

This register sets the video capture mode.

[bit1] VS (Video Select)

This bit selects NTSC or PAL to detect code errors.
(Only for R.BT656 input)

0 NTSC

1 PAL

[bit2] NRGB (Native RGB input on)

This bit sets the native RGB mode.

0 YUV 4:2:2

1 Native RGB

[bit20] VI (Vertical Interpolation)

This bit sets the interpolation processing in the vertical direction.

0 Interpolate in the vertical direction.
The image is magnified by twice the size in the vertical direction.

1 Do not interpolate in the vertical direction.

[bit25, bit24] CM (Capture Mode)

These bits set the mode for video capture. To capture video, set 11.

00 Initial value

01 Reserved

10 Reserved

11 Capture used

[bit28] VICE (Video Input Clock Enable)

This bit enables the capture clock.

0 Enable

1 Disable

[bit30] VIS (Video Input Select)

0 RBT656

1 RGB

[bit31] VIE (Video Input Enable)

This bit enables the video capture function.

0 Do not use video capture.

1 Use video capture.

- Procedure for stopping the video capture clock -
  1. Write "0" to bit31 (VIE) in the VCM register to disable the video capture function.
  2. Write "1" to bit28 (VICE) in the VCM register to stop the video capture clock.

- Procedure for starting the video capture clock -
  1. Write "0" to bit28 (VICE) in the VCM register to enable the video capture clock.
  2. Write "1" to bit31 (VIE) in the VCM register to enable the video capture function.

CSC (Capture Scale)

Register Address	$01FD_8004_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	VSCI					VSCF											HSCI					HSCF										
R/W	RW					RW											RW					RW										
Initial Value	0_0001					000_0000_0000											0_0001					000_0000_0000										

This register sets the scaling factor of video capture.

[bit10 to bit0] HSCF (Horizontal SCale Fraction)

These bits set the fraction part of the scaling factor in the horizontal direction.

[bit15 to bit11] HSCI (Horizontal Scale Integer)

These bits set the integer part of the scaling factor in the horizontal direction.

[bit26 to bit16] VSCF (Vertical SCale Fraction)

These bits set the fraction part of the scaling factor in the vertical direction.

[bit31 to bit27] VSCI (Vertical SCale Integer)

These bits set the integer part of the scaling factor in the vertical direction.

CBM0 (Video Capture Buffer Mode 0)

Register Address	$01FD_8010_H$																											
Bit Number	31	30	29	28	27	...	24	23	22	...	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	OO	SBUF	CRGB	-				CBW					BLEN	C24	RGBA	CSW	-											CBST
R/W	RW	RW	RW	RW0	R0			RW					RW	RW	RW	RW	R0				RW0				R0			RW
Initial Value	0	0	0	0	0000			X					0	0	0	0	0000				0000				000			0

[bit0] CBST (Capture Burst)

This bit specifies the burst length at the capture write time. The recommended setting is "1" because the long burst improves access efficiency.

0 Standard burst write (4word)

1 Long burst write (8word)

[bit12] CSW (Color Swap)

This bit swaps the byte positions of color components for writing in the YCbCr format.

0 Swap.

1 Do not swap.

[bit13] RGBA (RGBA format)

This bit selects the ARGB/RGBA format when writing in the RGB format.

0 ARGB format

1 RGBA format

[bit14] C24 (Color 24bit/pixel)

This bit selects 24 bits/pixel or 16 bits/pixel for capturing RGB input.

This function is enabled with the Native RGB capture (NRGB = 1) or converted RGB capture (CRGB = 1) setting.

0 16 bits/pixel

1 24 bits/pixel

[bit15] BLEN (Blend Enable)

This bit specifies the field A (blend bit) value when writing in the RGB format.

0 A = 0

1 A = 1

[bit23 to bit16] CBW (Capture Buffer memory Width)

These bits set the capture buffer memory width (stride) in units of 64 bytes.

[bit29] CRGB (Capture RGB write)

This bit specifies that data in the YCbCr format is converted to RGB = 5:5:5 (16 bits/pixel) before being written.

0 YCbCr format (No conversion)

1 RGB format

[bit30] SBUF (Single Buffer)

This bit specifies single buffer mode for capture buffer management.

0 Normal mode (Ring buffer)

1 Single buffer mode

[bit31] OO (Odd Only mode)

This bit specifies that only odd-number fields are captured.

0 Normal mode

1 Odd-numbers-only mode

CBM1 (Video Capture Buffer Mode 1)

Register Address	$01FD_8320_H$																			
Bit Number	31	30	…	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-								C565	-					Hrev	Vrev	Rrot	-			Tbuf
R/W	R0								RW	R0				RW0	RW	RW	RW	R0			RW
Initial Value	000_0000_0000_0000_0000								0	0000				0	0	0	0	000			0

[bit0] Tbuf (Triple buffer mode)

This bit specifies that the capture buffer is used for a triple buffer.

0 Do not use a triple buffer.

1 Use a triple buffer.

[bit4] Rrot (Rectangular rotate)

This bit specifies rotation by 90 degrees.

0 Do not rotate.

1 Rotate.

[bit5] Vrev (Vertical reverse mode)

This bit specifies inversion in the vertical direction.

0 Do not invert in the vertical direction.

1 Invert in the vertical direction.

[bit6] Hrev (Horizontal reverse mode)

This bit specifies inversion in the horizontal direction.

0 Do not invert in the horizontal direction.

1 Invert in the horizontal direction.

[bit12] C565 (Color 565)

This bit specifies the use of the RGB = 5:6:5 format when using 16 bits/pixel in the RGB format.

0 Use a format other than RGB = 5:6:5.

1 Use the RGB = 5:6:5 format.

CBA0 (Video Capture Buffer Address 0)

Register Address	$01FD_8014_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	CBA0																															
R/W	RW																												R0			
Initial Value	X																												0000			

This register specifies the first address of the video capture buffer. It corresponds to CBOA of the old model (MB86296).

CBA1 (Video Capture Buffer Address 1)

Register Address	$01FD_8018_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	CBA1																															
R/W	RW																												R0			
Initial Value	X																												0000			

This register specifies the last address of the video capture buffer. It corresponds to CBLA of the old model (MB86296). Be sure to set the address such that CBA1>CBA0.

CBA2 (Video Capture Buffer Address 2)

Register Address	$01FD_8324_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	CBA2																															
R/W	RW																												R0			
Initial Value	X																												0000			

This register specifies the first address of the third buffer when a triple buffer is used.

CBOFS (Video Capture Offset)

Register Address	$01FD_8328_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				CBOFS																											-
R/W	R0				RW																											R0
Initial Value	0_0000				00_0000_0000_0000_0000_0000_0000																													0

This register specifies the start point for frame writing with an address in bytes to write an image to the capture buffer.

CISTR (Capture Image Start)

Register Address	$01FD_801C_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				CIVSTR												-				CIHSTR											
R/W	RX				RW												RX				RW											
Initial Value	X				X												X				X											

This register sets the area of an image to write to the video capture buffer. The top left coordinates (CIHSTR, CIVSTR) of that write area are specified as a base point for the top left of the image. If the image is to be shrunk, this applies to the coordinates of the shrunken image.

[bit11 to bit0] CIHSTR (Capture Image Horizontal STaRt)

These bits specify the X coordinate.

[bit27 to bit16] CIVSTR (Capture Image Vertical STaRt)

These bits specify the Y coordinate.

CIEND (Capture Image End)

Register Address	$01FD_8020_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				CIVEND												-				CIHEND											
R/W	RX				RW												RX				RW											
Initial Value	X				X												X				X											

This register sets the area of an image to write to the video capture buffer. The bottom right coordinates (CIHEND, CIVEND) of that write area are specified as a base point for the top left of the image. If the image is to be shrunk, this applies to the coordinates of the shrunken image.

If the number of rasters of the input image is smaller than the set area, only the input image size part is written.

[bit11 to bit0] CIHEND (Capture Image Horizontal END)

These bits specify the X coordinate.

[bit27 to bit16] CIVEND (Capture Image Vertical END)

These bits specify the Y coordinate.

CVCNT (Capture Vertical Count)

Register Address	$01FD_8300_H$															
Bit Number	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				CVCNT											
R/W	R0				R											
Initial Value	0000				X											

This register indicates the Y coordinate of the currently captured raster. It is read-only data.

CHP (Capture Horizontal Pixel)

Register Address	$01FD_8028_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																						CHP									
R/W	RX																						RW									
Initial Value	X																						0x168 (360)									

This register sets the number of horizontal pixels of the image to output to the capture buffer. The value is specified in units of 2 pixels. The maximum value is 840 pixels (set value: 0x1A4).

CLPF (Capture Low Pass Filter)

Register Address	01FD_8040$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				CVLPF				-				CHLPF				-															
R/W	RX				R/W				RX				R/W				RX															
Initial Value	0000				0000				0000				0000				X															

This register sets the low-pass filter coefficients. The vertical low-pass filter is configured with a FIR filter of 3 taps, and the horizontal low-pass filter is configured with a FIR filter of 5 taps. Coefficient codes of 2 bits are specified independently for the brightness signal (Y) and color difference signal (C). With the setting of "00" for the coefficient code, the low-pass filter is OFF (through).

[bit17, bit16] CHLPF_C (Capture Horizontal LPF coefficient C)

CHLPF_C	K0	K1	K2	K3	K4
00	0	0	1	0	0
01	0	1/4	2/4	1/4	0
10	0	3/16	10/16	3/16	0
11	3/32	8/32	10/32	8/32	3/32

[bit19, bit18] CHLPF_Y(Capture Horizontal LPF coefficient Y)

CHLPF_Y	K0	K1	K2	K3	K4
00	0	0	1	0	0
01	0	1/4	2/4	1/4	0
10	0	3/16	10/16	3/16	0
11	3/32	8/32	10/32	8/32	3/32

[bit25, bit24] CVLPF_C (Capture Vertical LPF coefficient C)

CVLPF_C	K0	K1	K2
00	0	1	0
01	1/4	2/4	1/4
10	3/16	10/16	3/16
11	Setting prohibited	Setting prohibited	Setting prohibited

[bit27, bit26] CVLPF_Y (Capture Vertical LPF coefficient Y)

CVLPF_Y	K0	K1	K2
00	0	1	0
01	1/4	2/4	1/4
10	3/16	10/16	3/16
11	Setting prohibited	Setting prohibited	Setting prohibited

CMSS (Capture Magnify Source Size)

Register Address	$01FD_8048_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-								CMSHP								-						CMSVL									
R/W	RX								RW								RX						RW									
Initial Value	X								X								X						X									

[bit9 to bit0] CMSVL (Capture Magnify Source Vertical Line)

These bits set the number of vertical lines of an image before magnify processing.

[bit25 to bit16] CMSHP (Capture Magnify Source Horizontal Pixel)

These bits set the number of horizontal pixels of an image before magnify processing. The value is specified in units of 2 pixels.

CMDS (Capture Magnify Display Size)

Register Address	$01FD_804C_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-						CMDHP										-						CMDVL									
R/W	RX						RW										RX						RW									
Initial Value	X						X										X						X									

[bit9 to bit0] CMDVL (Capture Magnify Display Vertical Line)

These bits set the number of vertical lines of an image after magnify processing.

[bit26 to bit16] CMDHP (Capture Magnify Display Horizontal Pixel)

These bits set the number of horizontal pixels of an image after magnify processing. The value is specified in units of 2 pixels.

RGBHC (RGB Input Hsync Cycle)

Register Address	$01FD_8080_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																		RGBHC													
R/W	RX																		RW													
Initial Value	X																		X													

[bit13 to bit0] RGBHC

These bits input the horizontal cycle at the video input time. This is used for VSYNC sampling as set in the VIN_VLSAMP register and is used at the RGB/R.BT656 input time when HYSNC is not detected. The set value + 1 is the horizontal cycle.

RGBHEN (RGB Input Horizontal Enable Area)

Register Address	$01FD_8084_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-				RGBHST												-		RGBHEN													
R/W	RX				RW												RX		RW													
Initial Value	X				X												X		X													

This register is the parameter that determines the effective pixel data for the horizontal direction.

[bit12 to bit0] RGBHEN

These bits set the effective pixel data size in units of 2 pixels.

[bit27 to bit16] RGBHST

These bits set the start position of effective pixel data. The set value + 4 is the start position.

RGBVEN (RGB Input Vertical Enable Area)

Register Address	$01FD_8088_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-		RGBVST_T		-			RGBVST									-			RGBVEN												
R/W	RX		RW		RX			RW									RX			RW												
Initial Value	X		01		X			X									X			X												

This register is the parameter that determines the effective pixel data for the vertical direction. It is used with the RGB input format.

[bit12 to bit0] RGBVEN (RGB input Vertical Enable area Size)

These bits set the effective line size in the vertical direction.

[bit24 to bit16] RGBVST (RGB input Vertical Enable area Start position)

These bits set the start position of effective line data. The set value + 1 is the start position.

[bit29, bit28] RGBVST_T (RGB input Vertical Enable area Start position for Top field)

These bits are a 2-bit signed integer.

The start position of the effective line is RGBVST_T + RGBVST_BOTTOM. Since this capture controller supports only progressive RGB input, use it with RGBVST_O = 0.

VIN_VSAMP (Video Input Vsync Sampling Mode)

Register Address	01FD_808C$_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20		14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																		VJITFLT		-							FLDREV
R/W	RX																		RW		RX							RW
Initial Value	X																		00		X							0

[bit0] FLDREV (FieLD REVerse)

This bit reverses the field identification.

0 Do not reverse.

1 Reverse.

[bit9, bit8] VJITFLT (Vsync JITter FiLTer)

These bits set the sampling method for vertical synchronous signals. Select "00".

00 Sampling with HSYNC

01 Reserved

10 Reserved

11 Reserved

RGBS (RGB Input Sync)

Register Address	$01FD_8090_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-															RM	-														HP	VP
R/W	RX															RW	RX														RW	RW
Initial Value	X															1	X														0	0

This register sets edge detection for synchronous signals. It is used for RGB input.

[bit0] VP (VSYNC Polarity)

0 The falling edge of the VSIN signal is VSYNC.

1 The rising edge of the VSIN signal is VSYNC.

[bit1] HP (HSYNC Polarity)

0 The falling edge of the HSIN signal is HSYNC.

1 The rising edge of the HSIN signal is HSYNC.

[bit16] RM (RGB input Mode select)

This bit sets the RGB666 direct input mode.

0 Reserved

1 RGB666 direct input mode

Color Space Conversion

The following matrix formula is used to convert RGB input data to the YCbCr format:

Y = a11 * R + a12 * G + a13 * B + b1

Cb = a21 * R + a22 * G + a23 * B + b2 aij 10-bit signed real (lower 8 bits are fraction)

Cr = a31 * R + a32 * G + a33 * B + b3 bi 8-bit unsigned integer

The registers described below set each of the coefficients.

The color difference signals Cb and Cr undergo 2-4 conversion filter processing before color space conversion, after which they are in the YCbCr 4:2:2 format.

RGBCMY (RGB Color Convert Matrix Y Coefficient)

Register Address	01FD_80C0$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	a11										-	a12										-	a13									
R/W	RW										RX	RW										RX	RW									
Initial Value	00_0100_0010										X	00_1000_0000										X	00_0001_1001									

This register sets the coefficient of the Y signal calculation part of RGB/YCbCr conversion.

[bit9 to bit0] a13

10-bit signed real (lower 8 bits are fraction)

[bit20 to bit11] a12

10-bit signed real (lower 8 bits are fraction)

[bit31 to bit22] a11

10-bit signed real (lower 8 bits are fraction)

RGBCMCb (RGB Color Convert Matrix Cb Coefficient)

Register Address	01FD_80C4$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	a21										-	a22										-	a23								
R/W	RW										RX	RW										RX	RW								
Initial Value	11_1101_1010										X	11_1011_0110										X	00_0111_0000								

This register sets the coefficient of the Cb signal calculation part of RGB-to-YCbCr conversion.

[bit9 to bit0] a23

10-bit signed real (lower 8 bits are fraction)

[bit20 to bit11] a22

10-bit signed real (lower 8 bits are fraction)

[bit31 to bit22] a21

10-bit signed real (lower 8 bits are fraction)

RGBCMCr (RGB Color Convert Matrix Cr Coefficient)

Register Address	01FD_80C8$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	a31										-	a32										-	a33								
R/W	RW										RX	RW										RX	RW								
Initial Value	00_0111_0000										X	11_1010_0010										X	11_1110_1110								

This register sets the coefficient of the Cr signal calculation part of RGB-to-YCbCr conversion.

[bit9 to bit0] a33

10-bit signed real (lower 8 bits are fraction)

[bit20 to bit11] a32

10-bit signed real (lower 8 bits are fraction)

[bit31 to bit22] a31

10-bit signed real (lower 8 bits are fraction)

RGBCMb (RGB Color Convert Matrix b Coefficient)

Register Address	01FD_80CC$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-	b1									-		b2									-		b3								
R/W	RX	RW									RX		RW									RX		RW								
Initial Value	X	0_0001_0000									X		0_1000_0000									X		0_1000_0000								

This register sets the numerical value of an additional term for RGB/YCbCr conversion.

[bit8 to bit0] b3

9-bit unsigned integer

[bit19 to bit11] b2

9-bit unsigned integer

[bit19 to bit22] b1

9-bit unsigned integer

MDS (Mode Select)

Register Address	01FD_C014$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																										YCMIM		-			
R/W	RX																										RW		RX			
Initial Value	X																										00		X			

[bit5, bit4] YCMIM (YC Multiplex video Input Mode)

Use as the default value.

00 Fixed

01 Reserved

10 Reserved

11 Setting prohibited

Synchronization Error Detection

<Common to RGB/R.BT656>

VINLC (Video Input Line Count)

Register Address	01FD_C04C$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																			VIN_LINE_NO_kep												
R/W	RX																			R												
Initial Value	X																			0_0000_0000_0000												

[bit12 to bit0] VIN_LINE_NO_kep

These bits indicate the number of lines for 1 frame (field), including the blanking period. The displayed value + 1 is the number of lines.

VHSLS (Video Input HSYNC Long/Short)

Register Address	01FD_C054$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-		VIN_HS_LONG														-		VIN_HS_SHORT													
R/W	RX		RW														RX		RW													
Initial Value	X		X														X		X													

[bit13 to bit0] VIN_HS_SHORT

These bits set HSYNC monitoring with a shorter interval for video input.

If the input HSIN signal is shorter than the set interval, HSS_err is 1. The set value + 1 is the cycle.

[bit29 to bit16] VIN_HS_LONG

These bits set HSYNC monitoring with a longer interval for video input.

If the input HSIN signal is longer than the set interval, HSL_err is 1. The set value + 1 is the cycle.

VHSDC (Video Input HSync Down Count)

Register Address	$01FD_C058_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22		18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																						VIN_HDOWN_CNT							
R/W	RW0		RX		RW0	RX																	RW							
Initial Value	0		X		0	X																	0000_0000							

[bit7 to bit0] VIN_HDOWN_CNT

These bits set the monitoring cycle for no HSYNC video input.
If the period of no HSYNC exceeds the set cycle, HS_err is 1. The set value + 1 is the cycle.

VVSLS (Video Input VSync Long/Short)

Register Address	$01FD_C060_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-			VIN_VS_LONG													-			VIN_VS_SHORT											
R/W	RX			RW													RX			RW											
Initial Value	X			X													X			X											

[bit12 to bit0] VIN_VS_SHORT

These bits set VSYNC monitoring with a shorter interval for video input.
If the input VSIN signal is shorter than the set interval, VSS_err is 1. The set value + 1 is the cycle.

[bit28 to bit16] VIN_VS_LONG

These bits set VSYNC monitoring with a longer interval for video input.
If the input VSIN signal is longer than the set interval, VSL_err is 1. The set value + 1 is the cycle.

VVSDC (Video Input VSync Down Count)

Register Address	$01FD_C064_H$																															
Bit Number	31	30	29		26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																							VIN_VDOWN_CNT							
R/W	RW0	R																						RW							
Initial Value	0	000_0000_0000_0000_0000_0000																						0000_0000							

[bit7 to bit0] VIN_VDOWN_CNT

These bits set the monitoring cycle for no VSYNC video input.

If the period of no VSYNC exceeds the set cycle, VS_err is 1. The set value + 1 is the cycle.

CINT (Capture Interrupt)

Register Address	$01FD_8178_H$																															
Bit Number	31	30	29	28	27	26																									1	0
Bit Field Name	-																														VS	-
R/W	R0																														RW0	R0
Initial Value	00_0000_0000_0000_0000_0000_0000_0000																															

This register is the interrupt status register for video synchronous signals. Writing "0" clears the register value.

[bit1] VS (VSYNC)

1: VSYNC occurrence

0: No occurrence

CIMSK (Capture Interrupt Mask)

Register Address	$01FD_817C_H$																															
Bit Number	31	30	29	28	27	26																									1	0
Bit Field Name	-																														VS	-
R/W	RW0																														RW	RW0
Initial Value	00_0000_0000_0000_0000_0000_0000_0000																															

This register masks the interrupts of video synchronous signals.

[bit1] VS (VSYNC)

1: No masking

0: Masking

SYNC_err (SYNC Error)

Register Address	01FD_8180$_H$																													
Bit Number	31	30	29			26	25		24		23	21			18	17	16	15	14			11	10	9		8	7			3	2	1	0
Bit Field Name	-						VSL_err		VSS_err		-						VS_err	-						HSL_err		HSS_err	-						HS_err
R/W	RX						RW0		RW0		RX						RW0	RX						RW0		RW0	RX						RW0
Initial Value	X						0		0		X						0	X						0		0	X						0

This register is the interrupt status register for error-related video synchronous signals. Writing "0" clears the register value.

[bit0] HS_err (Hsync error)

1: Video input no-HSYNC error

0: No error

[bit8] HSS_err (HSync Short error)

1: Video input short-interval HSYNC error

0: No error

[bit9] HSL_err (Hsync Long error)

1: Video input long-interval HSYNC error

0: No error

[bit16] VS_err (Vsync down error)

1: Video input VSYNC stop error

0: No error

[bit24] VSS_err (Vsync Short error)

1: Video input short-interval VSYNC error

0: No error

[bit25] VSL_err (Vsync Long error)

1: Video input long-interval VSYNC error

0: No error

SYNC_err_MSK (SYNC Error Mask)

Register Address	$01FD_8184_H$																											
Bit Number	31	30	29			26	25	24	23	21			18	17	16	15	14			11	10	9	8	7			3	2	1	0
Bit Field Name	-						MVSL_err	MVSS_err	-						MVS_err	-						MHSL_err	MHSS_err	-						MHS_err
R/W	RX						RW	RW	RX						RW	RX						RW	RW	RX						RW
Initial Value	X						0	0	X						0	X						0	0	X						0

This register masks the interrupts of video synchronous signals.

[bit0] MHS_err (Mask HSync error)

1: No masking

0: Masking

[bit8] MHSS_err (Mask HSync Short error)

1: No masking

0: Masking

[bit9] MHSL_err (Mask HSync Long error)

1: No masking

0: Masking

[bit16] MVS_err (Mask VSync error)

1: No masking

0: Masking

[bit24] MVSS_err (Mask VSync Short error)

1: No masking

0: Masking

[bit25] MVSL_err (Mask VSync Long error)

1: No masking

0: Masking

Code Error Detection

<R.BT656 format input only>

CDCN (Capture Data Count for NTSC)

Register Address	01FD_C000$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-			BDCN													-			VDCN											
R/W	RX			RW													RX			RW											
Initial Value	X			0x10F(271)													X			0x5A3(1443)											

This register sets the data count for input video streams when the NTSC format is used.

(Valid only with the R.BT656 format)

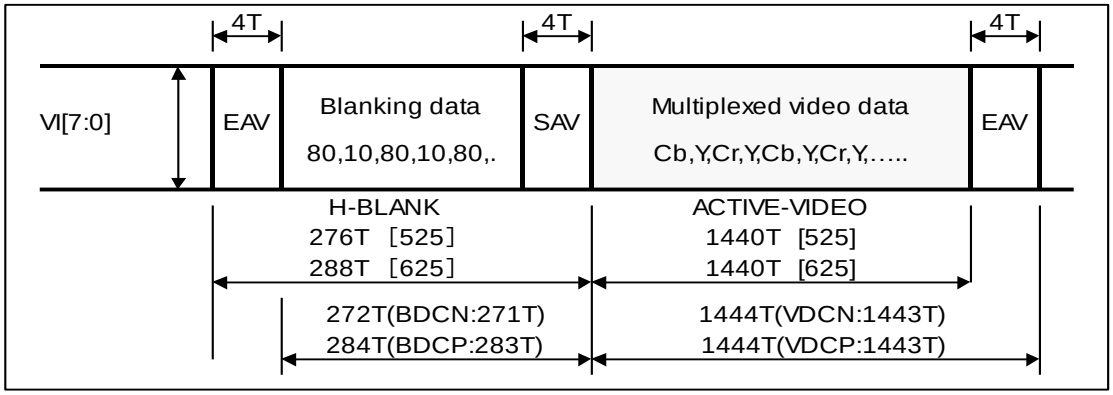
[bit12 to bit0] VDCN (Valid Data Count for NTSC)

These bits set the data count during the valid period when the NTSC format is used. The set value + 1 is the data count.

[bit28 to bit16] BDCN (Blanking Data Count for NTSC)

These bits set the data count during the blanking period when the NTSC format is used. The set value + 1 is the data count.

The following figure shows the VDCN and BDCN areas.



SAV: Start of active video timing reference code

EAV: End of active video timing reference code

T: Clock period 37 ns nom

CDCP (Capture Data Count for PAL)

Register Address	01FD_C004$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-			BDCP													-			VDCP											
R/W	RX			RW													RX			RW											
Initial Value	X			0x11B(283)													X			0x5A3(1443)											

This register sets the data count for input video streams when the PAL format is used.

(Valid only with the R.BT656 format)

[bit12 to bit0] VDCP (Valid Data Count for PAL)

These bits set the data count during the valid period when the PAL format is used. The set value + 1 is the data count.

[bit28 to bit16] BDCP (Blanking Data Count for PAL)

These bits set the data count during the blanking period when the PAL format is used. The set value + 1 is the data count.

CDCNS (Capture Data Count for NTSC Short)

Register Address	01FD_C018$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-			BDCN_S													-			VDCN_S											
R/W	RX			RW													RX			RW											
Initial Value	X			0x10F(271)													X			0x5A3(1443)											

This register sets the data count for input video streams with a shorter interval when the NTSC format is used.

(Valid only with the R.BT656 format)

[bit12 to bit0] VDCN_S (Valid Data Count for NTSC Short)

These bits set the data count during the valid period when the NTSC format is used. The set value + 1 is the data count.

[bit28 to bit16] BDCN_S (Blanking Data Count for NTSC Short)

These bits set the data count during the blanking period when the NTSC format is used. The set value + 1 is the data count.

CDCPS (Capture Data Count for PAL Short)

Register Address	01FD_C01C$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-			BDCP_S													-			VDCP_S										
R/W	RX			RW													RX			RW										
Initial Value	X			0x11B(283)													X			0x5A3(1443)										

This register sets the data count for input video streams with a shorter interval when the PAL format is used.

(Valid only with the R.BT656 format)

[bit12 to bit0] VDCP_S (Valid Data Count for PAL Short)

These bits set the data count during the valid period when the PAL format is used. The set value + 1 is the data count.

[bit28 to bit16] BDCP_S (Blanking Data Count for PAL Short)

These bits set the data count during the blanking period when the PAL format is used. The set value + 1 is the data count.

VCS (Video Capture Status)

Register Address	01FD_8008$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																									CE0						
R/W	RX																									RW0						
Initial Value	X																									000_0000						

This register is a status register indicating whether there was an error in input RBT.656 code. To detect errors in the code, NTSC/PAL must be set in the VS bit in the VCM register. If NTSC is set, the controller looks up the data count in a capture data count register (CDCN, CDCN_S). If PAL is set, it looks up the data count in a capture data count register (CDCP, CDCP_S). If the data count does not match the stream data or an undefined code is detected in the Fourth word of SAV/EAV, bit6 to bit0 in the Video Capture Status, VCS, register is as shown below. (For the code definitions, see "R.BT656 YUV422 Input Format".)

[bit0] CE0

1: RBT.656 Undefined code error (Code bit7)

0: No error

[bit1] CE0

1: RBT.656 Undefined code error (Code bit7 to bit4)

0: No error

[bit2] CE0

1: RBT.656 Undefined code error (Code bit7 to bit0)

0: No error

[bit3] CE0

1: RBT.656 Long interval H code error (SAV)

0: No error

[bit4] CE0
1: RBT.656 Long interval H code error (EAV)
0: No error

[bit5] CE0
1: RBT.656 Short interval H code error (SAV)
0: No error

[bit6] CE0
1: RBT.656 Short interval H code error (EAV)
0: No error

VCS_MSK (Video Capture Status Mask)

Register Address	01FD_800C$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																									MSK_CE0						
R/W	RX																									RW						
Initial Value	X																									000_0000						

This register specifies whether to mask each interrupt event. If an unmasked interrupt event occurs, interrupt output to the host CPU is generated. If a masked event occurs, no interrupt output to the host CPU is generated.

[bit0] MSK_CE0
RBT.656 Undefined code error masking
0: Masking
1: No masking

[bit1] MSK_CE0
RBT.656 Undefined code error (Code bit7 to bit4) masking
0: Masking
1: No masking

[bit2] MSK_CE0
RBT.656 Undefined code error (Code bit7 to bit0) masking
0: Masking
1: No masking

[bit3] MSK_CE0
RBT.656 Long interval H code error (SAV) masking
0: Masking
1: No masking

[bit4] MSK_CE0
RBT.656 Long interval H code error (EAV) masking
0: Masking
1: No masking

[bit5] MSK_CE0
RBT.656 Short interval H code error (SAV) masking

0: Masking
1: No masking

[bit6] MSK_CE0
RBT.656 Short interval H code error (EAV) masking
0: Masking
1: No masking

1.7.5 Explanation of Operation

1.7.5.1 Capture Controller Functions

Input Data Format

The input data format conforms to the ITU RBT656 format. (For details, see "1.7.5.6 External Video Signal Input"). The signal transmission system supports NTSC and PAL.

The system also supports digital RGB666 input.

Video Signal Capture

The capture controller is enabled when VIE in the Video Capture Mode (VCM) register is "1", capturing video stream data from the video data input pin by synchronizing with the CCLK clock.

Non-interlaced Conversion (Progressive Conversion)

The captured video images can be displayed in non-interlace mode. The 2 modes that can be selected for non-interlaced conversion are BOB mode and WEAVE mode.

BOB Mode

In the odd-number field, 1 frame is created with an average of the rasters of even-number fields and added. Likewise, in the even-number field, 1 frame is created with an average of the rasters of odd-number fields and added.

To select BOB mode, vertical interpolation must be enabled in the VI bit in the VCM (Video Capture Mode) register, and also the L1IM bit in the L1M (L1-layer Mode) register must be set to "0".

WEAVE Mode

In this mode, the odd-number fields and even-number fields are merged in the video capture buffer to form a frame. The vertical resolution is higher than that of BOB, but some misalignment of rasters may be visible in an image with more motion.



Stop Moving

To select WEAVE mode, vertical interpolation must be disabled in the VI bit in the VCM (Video Capture Mode) register, and also the L1IM bit in the L1M (L1-layer Mode) register must be set to "1".

Progressive Mode

The mode where non-interlaced conversion is used is called progressive mode. Progressive mode is used with VCM at "0" and L1IM at "0".

1.7.5.2 Video Buffer

Data Formats

Basically, data is stored in the capture buffer in 16 bits/pixel. The 2 color components (Cb, Cr) have half the resolution in the horizontal direction compared with brightness data (Y component), and as result, they are expressed with 16 bits/pixel. The data is converted to the RGB format on the L1 layer for display.

Format	31	30	···	25	24	23	22	···	17	16	15	14	···	9	8	7	6	···	1	0
YCbCr 16 bits/pixel	Y					Cr					Y					Cb				

The data can also be stored in the RGB format. The following formats are available.

Format	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARGB 16 bits/pixel	A	R					G					B				
RGBA 16 bits/pixel	R					G					B					A
RGB565 16 bits/pixel	R					G						B				

Format	31	30	···	25	24	23	22	···	17	16	15	14	···	9	8	7	6	···	1	0
ARGB 24 bits/pixel	A					R					G					B				
RGBA 24 bits/pixel	R					G					B					A				

The following table shows the relationship with the data format control bits in registers.

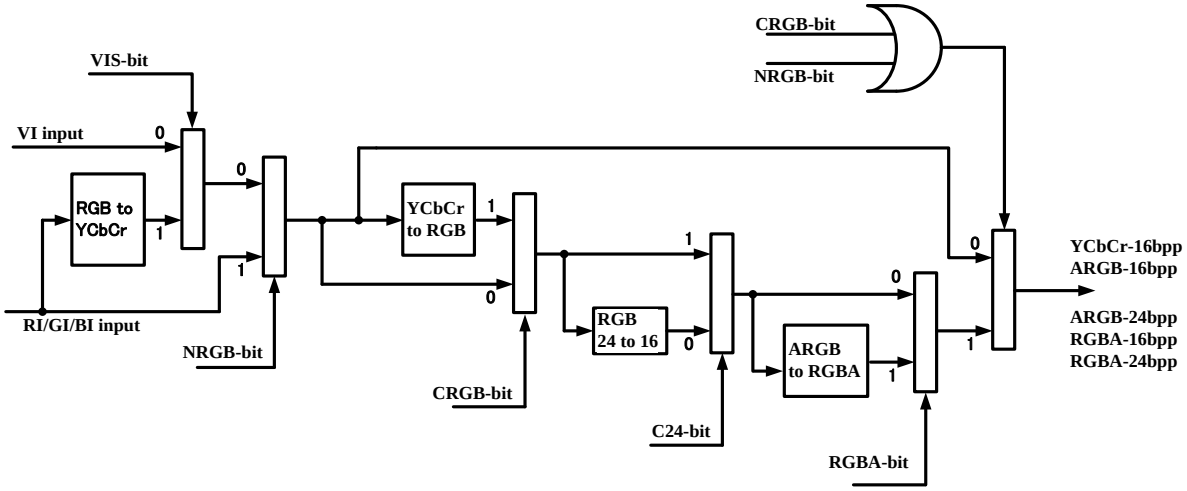
NRGB	CRGB	C24	RGBA	C565	Capture Data Format	Magnified Display
0	0	0	0	0	YCbCr 16 bits/pixel	Possible
0	1	0	0	0	ARGB 16 bits/pixel (YCbCr-to-RGB conversion)	
0	1	0	0	1	RGB565 16 bits/pixel (YCbCr-to-RGB conversion)	
0	1	0	1	0	RGBA 16 bits/pixel (YCbCr-to-RGB conversion)	
0	1	1	0	0	ARGB 24 bits/pixel (YCbCr-to-RGB conversion)	
0	1	1	1	0	RGBA 24 bits/pixel (YCbCr-to-RGB conversion)	
1	0	0	0	0	ARGB 16 bits/pixel	Possible
1	0	0	0	1	RGB565 16 bits/pixel	Possible
1	0	0	1	0	RGBA 16 bits/pixel	Possible
1	0	1	0	0	ARGB 24 bits/pixel	Possible
1	0	1	1	0	RGBA 24 bits/pixel	Possible

The magnified display of captured images is possible only with the formats that have "Possible" in the rightmost column.

Under any settings other than above, the written data has no meaning as image data.

The NRGB bit is bit2 in the VCM register. The CBM register has bit values other than those for this bit.

The following diagram shows the selection of a data format for captured images.



Note: The BLEN bit in the CBM register can be used for the ARGB or RGBA format to select whether the blend bit is "1" or "0" at the pixel write time.

BLEN=0 Blend bit=0 (MSB in ARGB, LSB in RGBA)

BLEN=1 Blend bit=1 (MSB in ARGB, LSB in RGBA)

Frame Management

There are 3 management modes

1. Single buffer mode

   Both display read and capture write can be executed, without synchronization, using an area size of 1 frame. As a result, the n-th frame and (n+1)-th frame are vertically mixed when displayed. The border of the frame may be momentarily visible when the image contents have a greater horizontal motion.



Stop



Moving

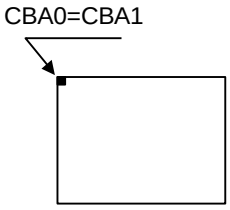
   Though the display quality in scenes with more motion is not good, the required area size is smaller.

   The area secured for the buffer area has the size of 1 frame + 1 vertical raster each, and the first address is set in the CBA0/1 register. Set the same value in these 2 registers.

   To capture n rasters of video, the following area is necessary for the video capture buffer:

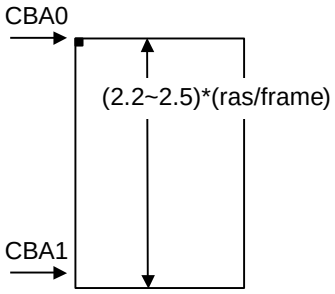
   "CBA0 – (CBW * 64)" to "CBA0 + (CBW * 64) * (n+1 rasters) – 1"

   As shown in the above formula, the area used for the buffer is not only an area from CBA0 up to as much as n rasters but also an area of 1 vertical raster.



2. Ring buffer mode

This mode has secured an area equivalent in size to 2.2 frames to 2.5 frames and a ring buffer configured in units of rasters. The border of frames is never visible because the displayed frame is 1 frame that has completed the capture operation.
If the capture speed of frames is higher than the display speed, a frame is skipped. Conversely, if the capture speed of frames is slower than the display speed, the same frame is displayed twice.



The first address is set in CBA0. The upper limit of the area is set in CBA1.

To secure an area of n rasters, set the value of the following formula:

CBA1 = CBA0 + (n – 2) * (CBW * 64)

The area used by hardware as the capture buffer has the following range:

"CBA0" to "CBA1 + (CBW * 64) * 2 rasters –1"

Here, note that CBA1 is not the upper limit address of the capture buffer itself.

To shrink display, secure a buffer area with the size of the shrunken frame.

3. Triple buffer mode

The area secured for each of the buffer areas has the size of 3 frames + 1 vertical raster each, and these 3 areas are used one after another cyclically. For video capture of n rasters, the necessary area for the video capture buffer is as follows:

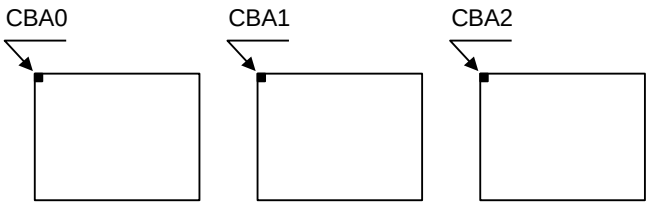
"CBA0 – (CBW * 64)" to "CBA0 + (CBW * 64) * (n+1 rasters) – 1"

"CBA1 – (CBW * 64)" to "CBA1 + (CBW * 64) * (n+1 rasters) – 1"

"CBA2 – (CBW * 64)" to "CBA2 + (CBW * 64) * (n+1 rasters) – 1"

As shown in the above formula, the area used for each buffer is not only an area from CBA0/1/2 up to as much as n rasters but also an area of 1 vertical raster.

This mode is used for capture accompanied by the 90-degree rotations described below. The operation when there is a relative difference between the capture speed and display speed is the same as in 2).
Each register, CBA0/1/2, specifies the first address of each area.



Capture including the 90-degree rotations described below uses the areas.

The following bits are selected in 1 to 3.

	CBM0/Sbuf-bit	CBM1/Tbuf-bit
Single Buffer	1	0
Ring Buffer	0	0
Triple Buffer	1	1

Window Display

The captured video images are displayed with the use of the L1 layer. All or part of a captured image can be displayed on the entire screen or as a window.

For capture display, set the L1 layer to the capture synchronous mode (L1CS=1). In this mode, the L1 layer displays the latest frame within the video capture buffer. The display addresses used in normal mode are ignored.

The stride of the L1 layer must match the stride of the video capture buffer. If they do not match, the displayed image is obliquely distorted.

The display size of the L1 layer must be the same as the image size of the shrunken video captured images. If the set display size of the L1 layer is larger than the captured image size, invalid data is displayed.

Though RGB display and YCbCr display can be selected for the L1 layer, select the YCbCr format (L1YC=1) when using video capture.

Interlaced Display

WEAVE mode enables the display of images captured in the video capture buffer in interlace mode. To configure that, enable WEAVE mode, and select interlaced and video display for a display scan.

However, if the display scan is in asynchronous mode, there is flickering in scenes with more motion. To prevent this from happening, set "1" in the OO (Odd Only) bit in CBM0 (Capture Buffer Mode 0).

1.7.5.3 Rotation Write for Captured Images

Overview

Images being written to the capture buffer can be rotated in the vertical and horizontal directions by 90 degrees. 7 conversion types excluding identical transformation are available.

	No Inversion	Right/Left Inversion	Up/Down Inversion	Right/Left and Up/Down Inversion
Coordinates not Converted				
Coordinates Converted				

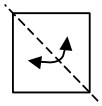
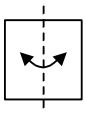
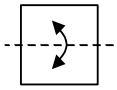
Note the following restrictions.

1. Use it with a single buffer or triple buffer. It cannot be used with a ring buffer.
2. Use it in BOB or progressive mode. It cannot be used in WEAVE mode.
3. Magnification cannot be applied to each axis when it is with rotational operation. If magnification is not applied, there is no limitation.
4. Stop the capture operation first to switch parameters. If a parameter is switched during the operation, data in another data area in VRAM may be overwritten.

The RRot bit, Hrev bit, and Vrev bit described below specify this rotation. If all of these bits are "0", operation is exactly the same as with the old GDC (MB86296).

Various Inversion Modes

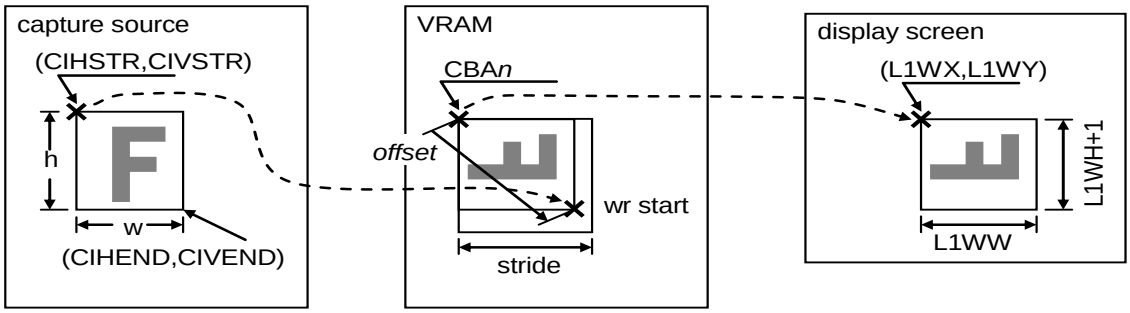
Specify the modes with the following parameters.

	y=x Symmetric inversion (xy coordinates swapped) Use the RRot bit in the CBM1 register to select this mode.
	Horizontal inversion Use the Hrev bit in the CBM1 register to select this mode.
	Vertical inversion Use the Vrev bit in the CBM1 register to select this mode.

Writing Start Point Offset

To start writing a rotated image from the proper location in the capture buffer, it is necessary to specify a start point offset. A captured input image is input in the order of a raster scan, with its top left corner as the start point. The writing start point offset specifies a corresponding write destination for this input start point.

The top left is the origin. The origin for this write destination point is the origin of the frame in the single buffer or triple buffer. In VRAM, the CBA0/1/2 register specifies the point. This point is set with a linear address as the base point in the CBOFS (capture buffer offset) register. CBOFS has the corresponding address in bytes. For identical transformation, the offset is 0.

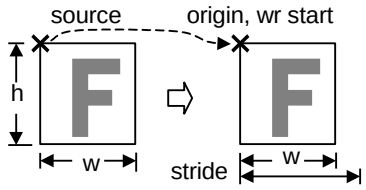
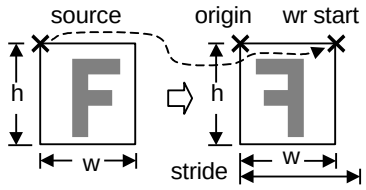
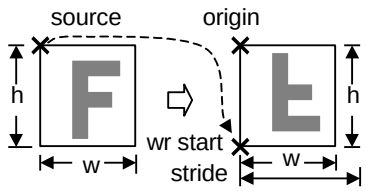


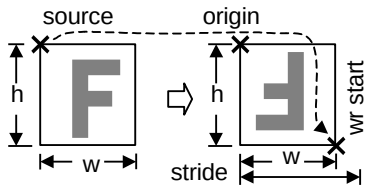
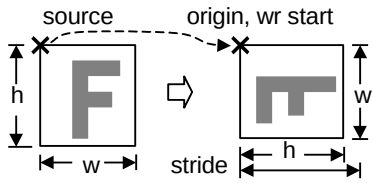
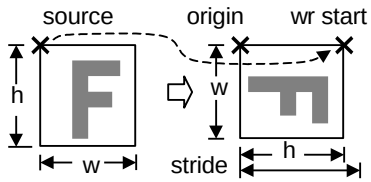
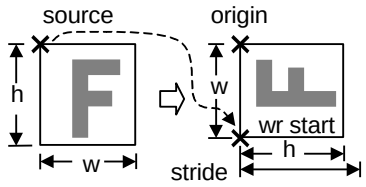
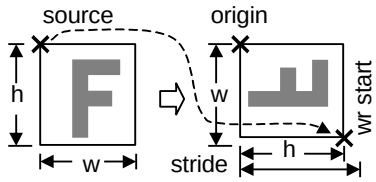
The source image frame becomes the frame clipped from (CIHSTR, CIVSTR) to (CIHEND, CIVEND). Therefore, the height h and the width w of the area have the following relationship:

w = CIHEND–CIHSTR+1

h = CIVEND–CIVSTR+1

The following table shows the writing start point offset for each inversion mode.

	RectRot=0, Hrev=0, Vrev=0 CBOFS=0 Actually, the set values in the register are ignored and forcibly handled as 0.
	RectRot=0, Hrev=1, Vrev=0 CBOFS= (w–1)*bpp/8 (bpp=16,32)
	RectRot=0, Hrev=0, Vrev=1 CBOFS=(h–1)*stride (stride=64*CBW)

	RectRot=0, Hrev=1, Vrev=1 CBOFS= (w–1)*bpp/8+(h–1)*stride (bpp=16,32; stride=64*CBW)
	RectRot=1, Hrev=0, Vrev=0 CBOFS=0
	RectRot=1, Hrev=1, Vrev=0 CBOFS = (h–1)*bpp/8 (bpp=16,32)
	RectRot=1, Hrev=0, Vrev=1 CBOFS= (w–1)*stride (stride=64*CBW)
	RectRot=1, Hrev=1, Vrev=1 CBOFS= (h–1)*bpp/8+(w–1)*stride (bpp=16,32; stride=64*CBW)

Display Position Fine Adjustment

The set value for the writing start point offset (CBOFS) has the following restriction.

Suppose RectRot=0 and Hrev=1:

- For 32 bits/pixel, the lower 3 bits in CBOFS are "100".
- For 16 bits/pixel, the lower 3 bits in CBOFS are "110".

Under this restriction, excess pixels may be displayed to the left of the L1 display window.

To prevent this from happening, configure the CAOFS field in the L1EM register on the display side.

To ignore n pixels of the excess pixels on the left side, set CAOFS = n*bpp/8 (bpp=16, 32).

1.7.5.4 Scaling

Shrink Function for Video

The capture controller shrinks video images when CM in the Video Capture Mode (VCM) register is "11". Shrinking can have independent vertical and horizontal settings, in units of 1 line in the vertical direction and in units of 2 pixels in the horizontal direction. The set value for scaling is defined by the input/output value as a 16-bit fixed-point value expressed with 5 bits for the integer part and 11 bits for the fraction part. The valid set values range from 0x0800 to 0xFFFF. Set a value for the vertical direction in bit31 to bit16 in the capture scale register (CSC), and set a value for the horizontal direction in bit15 to bit0. The initial value of this register is 0x08000800 (same size). The following example shows setting calculation expressions.

Shrinking in the vertical direction	576 -> 490 lines	576/490 = 1.176
	1.176*2048=2408 -> 0x0968	
Shrinking in the horizontal direction	720 -> 648 pixels	720/648 = 1.111
	1.111*2048=2275 -> 0x08E3	

The calculations show that the setting for CSC is 0x096808E3.

Note: The Capture Horizontal Pixel (CHP) register restricts the number of pixels during scaling processing and does not have the set value for scaling itself. Clamp processing is applied to video streaming data that exceeds the set value in CHP. Normally, it is possible to use this register with the initial value unchanged.

Magnify Function for Video

The capture controller can magnify video images in the horizontal and vertical directions, independently in these 2 directions. This function can be used, for example, for full-screen display of a video stream input that does not necessarily meet the resolution for full-screen display. It can also be used to magnify (zoom) part of a video stream input. The initial setting procedure is described below:

1. Set the magnify flag in the L1-layer Mode register of the Display Controller.
2. Set the unmagnified source image size in CMSHP and CMSVL.
3. Set the magnified output image size in CMDHP and CMDVL.

As an example, suppose the following case:

- Input image size: 480 x 360 pixels
- Output image size: 640 x 480 pixels

Then, each of the register settings would be as follows:

HSCALE = (480/640) * 2048 = 0x0600
VSCALE = (360/480) * 2048 = 0x0600

CMSHP = 0x00f0
CMSVL = 0x0168
CMDHP = 0x0140
CMDVL = 0x01e0
L1WW = 0x0280
L1WH = 0x01df

Notes:

- When magnification and shrinking have different specifications for each direction, the parameters for the direction of shrinking are set as follows:

  1. Magnifying vertically and shrinking horizontally
     Set CMSHP and CMDHP as follows.
     CMDHP = Display window width/2
     CMSHP = CMDHP

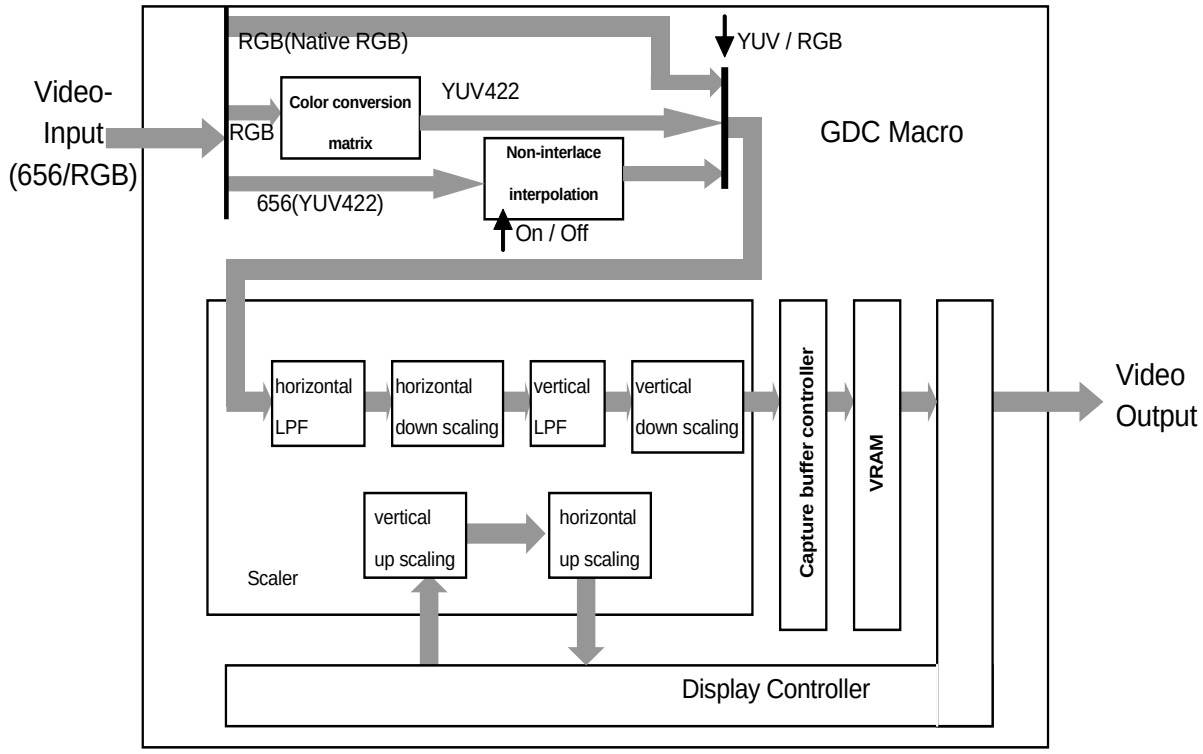
  2. Magnifying horizontally and shrinking vertically
     CMDVP = Number of display rasters
     CMSVP = CMDVP

- To switch from magnifying to shrinking because of a display size change, place a setting of the same size (CSC=0x08000800) in between them for at least 2V periods (vertical synchronization of display). This restriction is due to the fact that an interpolating filter is commonly used for magnifying and shrinking. There is no restriction on switching from shrinking to magnifying.
- Shrunken or magnified images are slightly distorted when the scaling factor is continuously changed for the respective scale of display. This is due to the scheme of the capture controller functions.
- Magnified display cannot be used when the external synchronization function is used. The internal control circuit does not support such use, which consequently can cause unstable display.

Image Processing Flow

The following figure shows the flow of image processing for the captured image displayed in the L1 layer window.

Figure 1-22. Image Processing Flow



Non-interlaced Interpolation Processing

The interlaced screen is vertically interpolated by using the data inside the same field, when VI in the Video Capture Mode (VCM) register is "0". The screen is vertically magnified by 2 times. There is no vertical interpolation when VI is "1".

Horizontal Low-pass Filter Processing

A low-pass filter in the horizontal direction can be applied as preprocessing for shrinking an image in the horizontal direction. The horizontal low-pass filter (LPF) can be set ON/OFF regardless of whether the image is being magnified or shrunk.

The horizontal low-pass filter is configured with a FIR filter of 5 taps. The following register specifies the coefficients.

CHLPF_Y	Horizontal LPF brightness component coefficient code
CHLPF_C	Horizontal LPF color difference component coefficient code

Coefficient codes of 2 bits independently specify the coefficients for the brightness (Y) signal and color difference (C) signal. The coefficients are left-right symmetric.

CHLPF_x	K0	K1	K2	K3	K4
00	0	0	1	0	0
01	0	1/4	2/4	1/4	0
10	0	3/16	10/16	3/16	0
11	3/32	8/32	10/32	10/32	3/32

With the setting of "00" for the coefficient code, the horizontal LPF is OFF (through).

Note: In native RGB mode (NRGB=1), only the coefficient code of CHLPF_Y is valid.

Horizontal Shrink and Magnify Processing

To shrink and magnify in the horizontal direction, set bit15 to bit0 in the capture scale register (CSC).
Images are shrunken in the horizontal direction before being written to VRAM. In contrast, images are magnified in the horizontal direction after being read from VRAM.
The interpolation filter processing of the brightness (Y) signal is executed using the Cubic Interpolation method, and the interpolation filter processing of the color difference (C) signal is executed using the Bilinear Interpolation method.

Vertical Low-pass Filter Processing

A low-pass filter in the vertical direction can be applied as preprocessing for shrinking an image in the vertical direction. The vertical low-pass filter (LPF) can be set ON regardless of whether the image is being magnified or shrunk in the vertical direction.
The vertical low-pass filter is configured with a FIR filter of 3 taps. The following register specifies the coefficients.

CVLPF_Y	Vertical LPF brightness component coefficient code
CVLPF_C	Vertical LPF color difference component coefficient code

Coefficient codes of 2 bits independently specify the coefficients for the brightness (Y) signal and color difference (C) signal. The coefficients are left-right symmetric.

CVLPF_x	K0	K1	K2
00	0	1	0
01	1/4	2/4	1/4
10	3/16	10/16	3/16
11	Setting prohibited	Setting prohibited	Setting prohibited

With the setting of "00" for the coefficient code, the vertical LPF is OFF (through).

Note: In native RGB mode (NRGB=1), only the coefficient code of CVLPF_Y is valid.

Vertical Shrink and Magnify Processing

To shrink and magnify in the vertical direction, set bit31 to bit16 in the capture scale register (CSC).
Images are shrunken in the vertical direction before being written to VRAM. In contrast, images are magnified in the vertical direction after being read from VRAM.
The interpolation filter processing of the brightness (Y) signal is executed using the Cubic Interpolation method, and the interpolation filter processing of the color difference (C) signal is executed using the Bilinear Interpolation method.

1.7.5.5 Interrupts

Overview

The following interrupt events are generated from the capture controller.

- Error detection
  This event occurs when video input has an error.
- Capture VSYNC
  This event occurs in sync with the vertical synchronous signal of video input.

Interrupt Status

Interrupts in the capture controller are transmitted to MCNT.

Error Detection

If an expected control code or synchronous signal is not detected in input video data, an error occurs. The corresponding status register in the capture controller is the SYNC_err register.

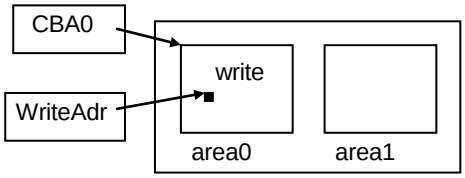
Capture VSYNC Interrupt

This interrupt can be informative with regard to the capture VSYNC timing. The corresponding status register in the capture controller is the CINT register.

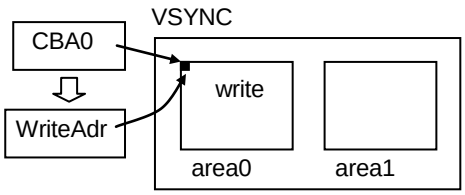
The expected uses are as follows:

1. Detection of valid image input
2. Frame management by the host CPU

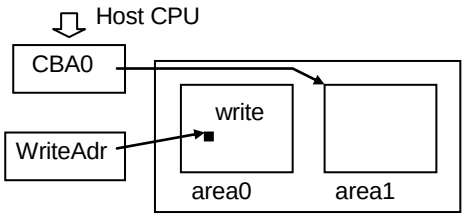
The single buffer mode is used for frame management by the host CPU. The CBA0 and CBA1 register, which specify the first address of the buffer, can be rewritten at the VSYNC interrupt generation time.
The frame immediately after the interrupt is written to the pre-update address. The new address applies to the next and subsequent frames. This means that the capture image at the old address can be used as a still image.
The status of writing to area 0 is indicated. The CBA0 register retains the first address of area 0.



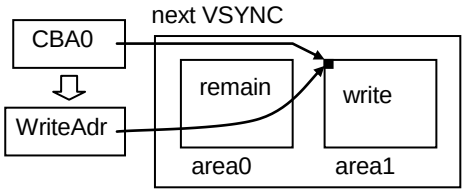
In sync with VSYNC, the write address moves to the first address of area 0. A VSYNC interrupt is generated.



After the VSYNC interrupt, the CPU changes CBA0 to the first address of area 1. There is no impact on the write operation until the next VSYNC occurrence.



After the next VSYNC, writing to area 1 begins, and the image in area 0 is saved.



Note: For single-buffer operation, set CBA0=CBA1.

1.7.5.6 External Video Signal Input

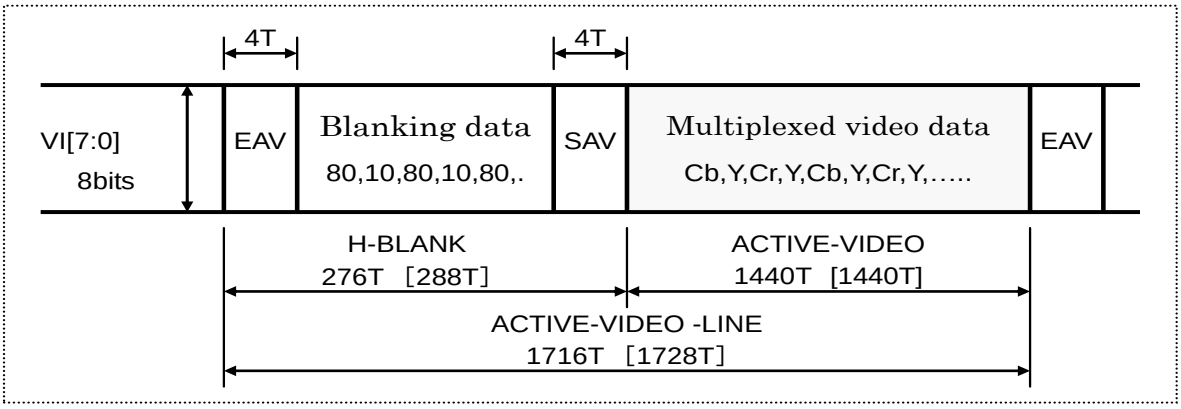
R.BT656 YUV422 Input Format

The ITU R.BT-656 format is widely used for digital NTSC and PAL transmission. The format supports YUV422. The capture controller enables capture operations where the input interlaced video signals are converted to non-interlaced video signals through linear interpolation.

The capture controller is capture-enabled when VIE in the Video Capture Mode register (VCM) is "1", capturing video stream data from the 8-bit VI pin in sync with the CCLK clock. The format must be Y,Cb,Cr 4:2:2 with a timing reference code added because the video streams that can be processed are limited to the digital video streams conforming to ITU RBT656. Even though the capture controller can adapt automatically to NTSC and PAL since it employs the timing reference code for stream capturing, set NTSC or PAL in VS in VCM to detect code errors. The data count in the capture data count register (CDCN) is referenced when NTSC is set, and that in the capture data count register (CDCP) is referenced when PAL is set. If the count does not match the stream data, bit4 to bit0 in the video capture status register (VCS) is any value other than "00000".

R.BT656 Input Format

In sync with the 27 MHz clock, synchronous code and image data (Cb, Y, Cr, Y) are input as multiplexed 8-bit data, and the active pixels are transmitted sandwiched between 2 synchronous codes, SAV and EAV.



SAV: Start code of active video data (4 bytes)
EAV: End code of active video data (4 bytes)
T: 27 MHz
[ ]: 625/50 series (PAL)

BLANKING PERIOD			TIMING REF-CODE				720 PIXELS YUV4:2:2 DATA									TIMING REF-CODE				BLANKING PERIOD		
…	80	10	FF	00	00	SAV	Cb0	Y0	Cr0	Y1	Cb2	Y2	…	Cr718	Y719	FF	00	00	EAV	80	10	…

R.BT656 Synchronous Code (4 Bytes) Format

Bit \ Word	SYNC Code (Fixed)			EAV/SAV
	1st	2nd	3rd	4th
7	1	0	0	1 (Fixed)
6	1	0	0	F 0: 1st field 1: 2nd field
5	1	0	0	V 0: ACTIVE-VIDEO 1: VBI
4	1	0	0	H 0: SAV 1: EAV
3	1	0	0	P3 Protected bit
2	1	0	0	P2 Protected bit
1	1	0	0	P1 Protected bit
0	1	0	0	P0 Protected bit

SAV/EAV Timing Reference Signal

Bit	7	6	5	4	3	2	1	0
Function	Fixed	F	V	H	P3	P2	P1	P0
80	1	0	0	0	0	0	0	0
9D	1	0	0	1	1	1	0	1
AB	1	0	1	0	1	0	1	1
B6	1	0	1	1	0	1	1	0
C7	1	1	0	0	0	1	1	1
DA	1	1	0	1	1	0	1	0
EC	1	1	1	0	1	1	0	0
F1	1	1	1	1	0	0	0	1

80: SAV code of 1st-field active pixel period (Active-video)

9D: EAV code of 1st-field active pixel period (Active-video)

AB: SAV code of 1st-field vertical blanking period

B6: EAV code of 1st-field vertical blanking period

C7: SAV code of 2nd-field active pixel period (Active-video)

DA: EAV code of 2nd-field active pixel period (Active-video)

EC: SAV code of 2nd-field vertical blanking period

F1: EAV code of 2nd-field vertical blanking period

R.BT656 Synchronous Code (EAV) Timing (525/60 Series)

line-no, 1st field	522	523	524	525	1	2	3	4	5	6	7	8	9
	Active video				equalizing pulse			vertical sync			equalizing pulse		
EAV	DA	DA	DA	DA	F1	F1	F1	B6	B6	B6	B6	B6	B6
F	1	1	1	1	1	1	1	0	0	0	0	0	0
V	0	0	0	0	1	1	1	1	1	1	1	1	1
line-no, 2nd field	260	261	262	263	264	265	266	267	268	269	270	271	272
	Active video			equalizing pulse			vertical sync			equalizing pulse			
EAV	9D	9D	9D	9D	B6	B6	F1	F1	F1	F1	F1	F1	F1
F	1	1	1	1	1	1	1	0	0	0	0	0	0
V	0	0	0	0	1	1	1	1	1	1	1	1	1

line-no, 1st field	10	11	12	13	14	15	16	17	18	19	20	21	22
	VBI-lines 1st field											Act-video	
EAV	B6	B6	B6	B6	B6	B6	B6	B6	B6	B6	9D	9D	9D
F	0	0	0	0	0	0	0	0	0	0	0	0	0
V	1	1	1	1	1	1	1	1	1	1	0	0	0
line-no, 2nd field	273	274	275	276	277	278	279	280	281	282	283	284	285
	VBI-lines 2nd field											Act-video	
EAV	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	DA	DA	DA
F	1	1	1	1	1	1	1	1	1	1	1	1	1
V	1	1	1	1	1	1	1	1	1	1	0	0	0

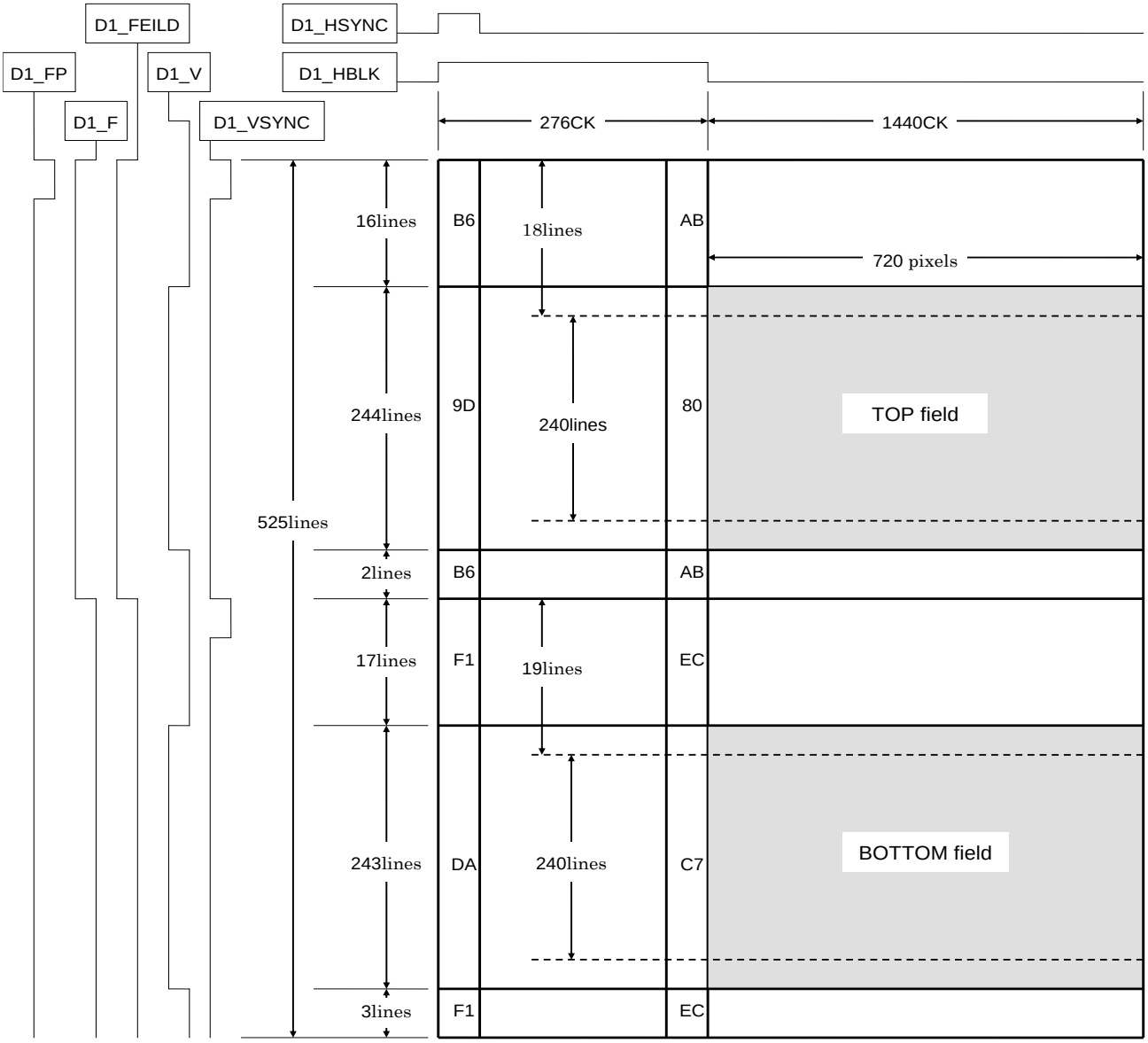
R.BT656 Synchronous Code (EAV) Timing (625/50 Series)

line-no, 1st field	620	621	622	623	624	625	1	2	3	4	5	6	7	8	9
	Active video			equalizing pulse			vertical sync			equalizing pulse		VBI-lines 1st field			
EAV	DA	DA	DA	DA	F1	F1	B6	B6	B6	B6	B6	B6	B6	B6	B6
F	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0
V	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
line-no, 2nd field	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337
	Active video			equalizing pulse			vertical sync		equalizing pulse			VBI-lines 2nd field			
EAV	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	DA	DA
F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
V	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

line-no, 1st field	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
	VBI-lines 1st field													Act-video	
EAV	B6	B6	B6	B6	B6	B6	B6	B6	B6	B6	B6	B6	B6	9D	9D
F	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
V	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
line-no, 2nd field	323	324	325	326	327	328	329	330	331	332	333	334	335	336	337
	VBI-lines 2nd field												Active-video		
EAV	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	F1	DA	DA
F	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
V	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0

R.BT656 Frame Format

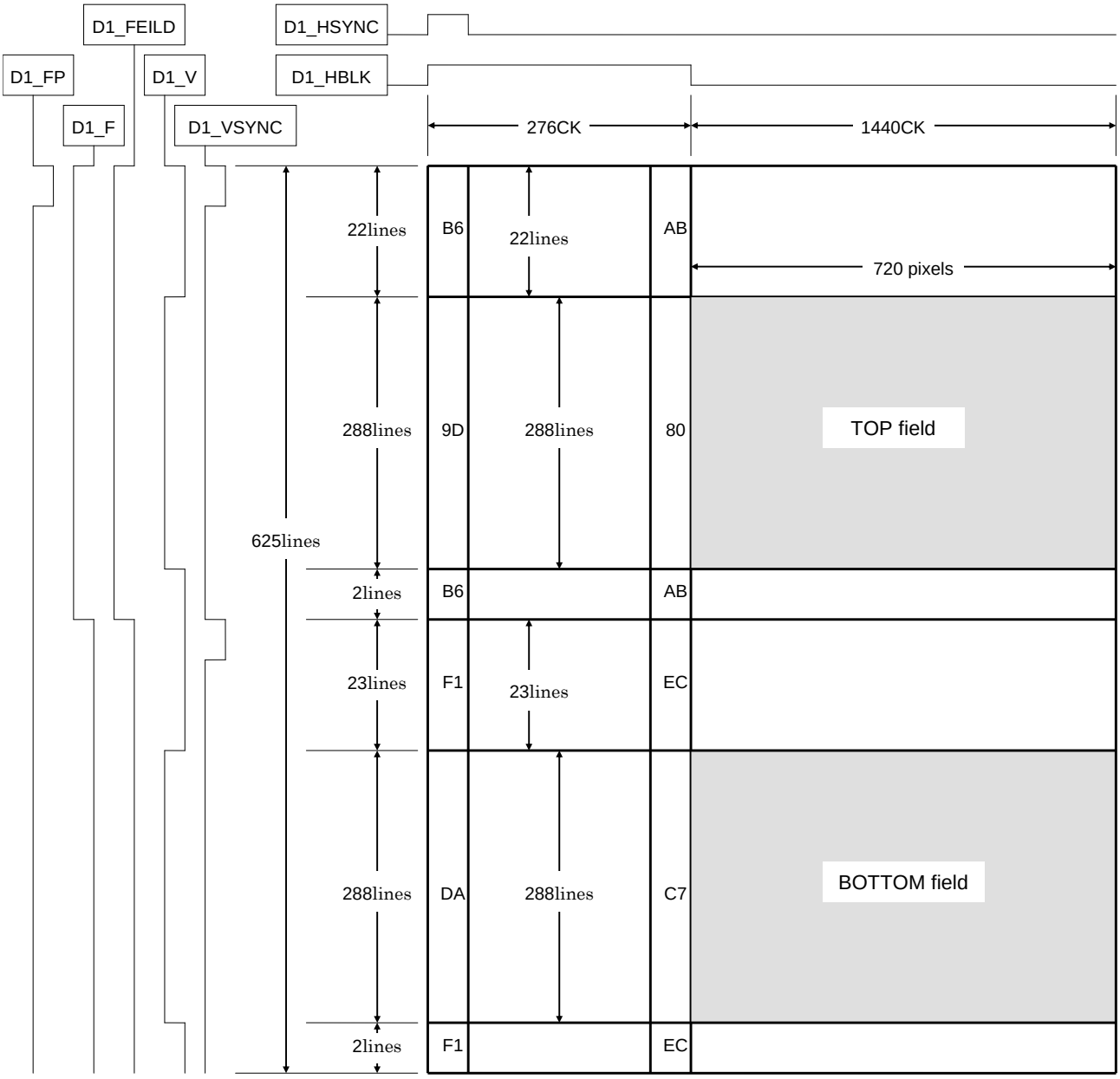
R.BT656 Format Input [525/60 Series]



Notes:

- CCLK_RGB = 27 MHz
- SAV and EAV as denoted are actually 4 bytes and "FF 00 00 XX".
- D1_FP, D1_F, D1_FEILD, D1_V, D1_VSYNC, D1_HSYNC, and D1HBLK are illustrations of internal signals generated by SAV and EAV.

R.BT656 Format Input [625/50 Series]



Notes:

- CCLK_RGB = 27 MHz
- SAV and EAV as denoted are actually 4 bytes and "FF 00 00 XX".
- D1_FP, D1_F, D1_FEILD, D1_V, D1_VSYNC, D1_HSYNC, and D1HBLK are illustrations of internal signals generated by SAV and EAV.

RGB Input Format

There are 2 data processing modes for the RGB input video capture function. One is the mode of processing as Native RGB, and the other is the mode of processing where RGB is converted to YUV422 by the internal RGB preprocessor.

The RGB input function supports progressive video input. It does not support an interlaced-to-progressive conversion function. It also supports input of a maximum 66 Mpixels/second. The RGB component data consists of 6 bits.

Note: To set Native RGB mode, set "1" in NRGB.

RGB Input Signal

Name	I/O	Function
CCLK_RGB	Input	Clock for RGB input
CAP0R0-5	Input	Red component value
CAP0G0-5	Input	Green component value
CAP0BI0-5	Input	Blue component value
VSIN	Input	Vertical sync for RGB capture
HSIN	Input	Horizontal sync for RGB capture

Note: Use the VIS bit in the VCM (Video Capture Mode) register to select YUV422 input (R.BT656) or RGB input.

Capture Area Settings

Registers are configured with the RGB input function as follows.

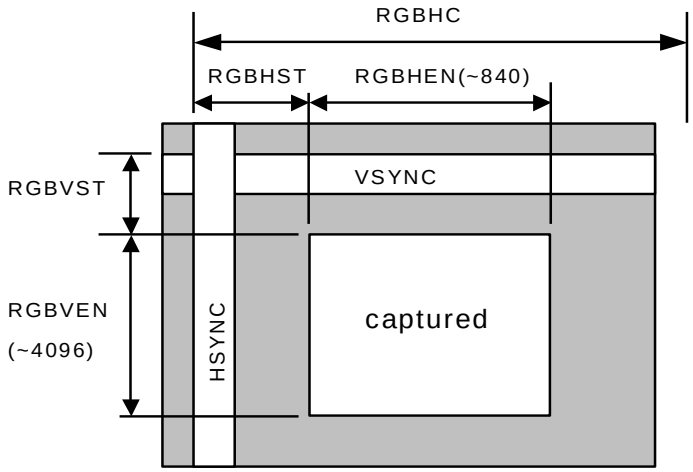
- RGB input mode setting:
  1. Set the RGB666 input flag (VIS) in the VCM register.
  2. Set the NRGB bit in the VCM register to "1" in Native RGB mode.
- HSYNC cycle setting

  Set the HSYNC cycle in RGBHC.
  1. Horizontal active pixel area setting

     Set the start position for the active pixels and the active pixel size in RGBHST and RGBHEN, respectively.
  2. Vertical active pixel area setting

     Set the start position for the active pixels and the active pixel size in RGBVST and RGBVEN, respectively.

The following figure defines the video capture area.



RGBHC	RGB input Hsync Cycle
RGBHST	RGB input Horizontal enable area STart position
RGBHEN	RGB input Horizontal enable area size
RGBVST	RGB input Vertical ENable area STart position
RGBVEN	RGB input Vertical ENable area size

Notes:

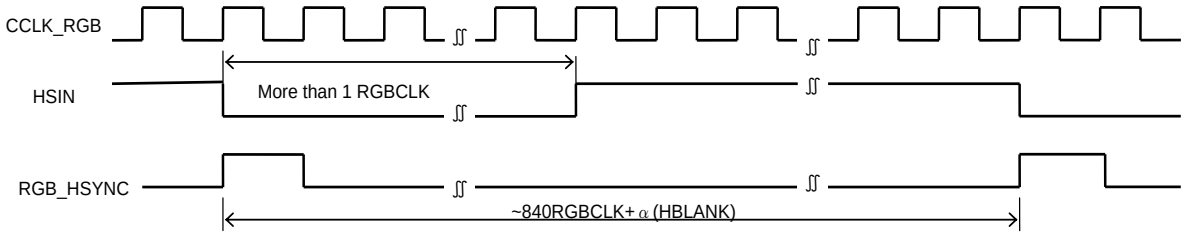
- The actual settings of the displayed parameters differ slightly from those above. For details, see "1.7.4.3 Register Details."
- The horizontal active pixel setting (RGBHEN) is a value up to 840, maximum. This restriction is due to the line buffer size in the video capture module.

RGB Input Timing

The data at the RGB input time is synchronized by VSIN and HSIN that are input together with the data RI, GI, and BI.

- HSIN input rule

  According to the HP register setting, the rising edge or falling edge of the HSIN signal is used for horizontal



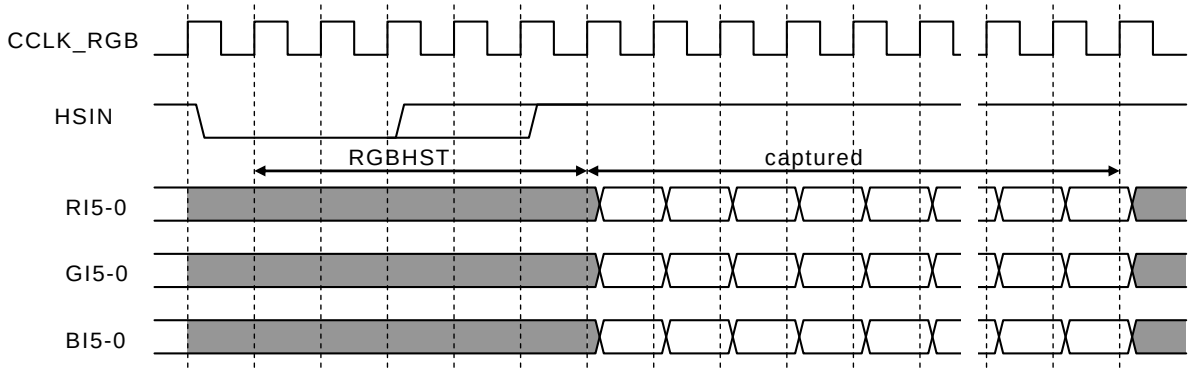
  synchronization. The input signal is 1 clock (CCLK_RGB) length or more.

  Notes:

- The horizontal active pixel setting (RGBHEN) is a value up to 840, maximum. This restriction is due to the line buffer size in the video capture module.
- RGB_HSYNC is the internal signal generated by the HSIN signal.

  Active pixel data input rule pertaining to HSYNC

  The active pixel data input rule pertaining to HSYNC is shown here.
  Input data is input in sync with HSYNC for each line.
  The distance from HSYNC to the top of the active pixels of the data can be changed with the RGBHST register setting.

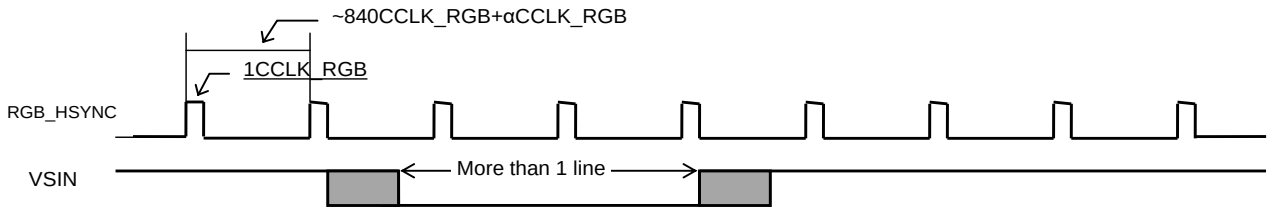


- VSYNC input rule

  According to register settings, the VSIN signal is sampled by HSYNC and handled as an internal VSYNC signal.

  In this case, the width of the VSIN signal input from the outside should be at least 1 line.

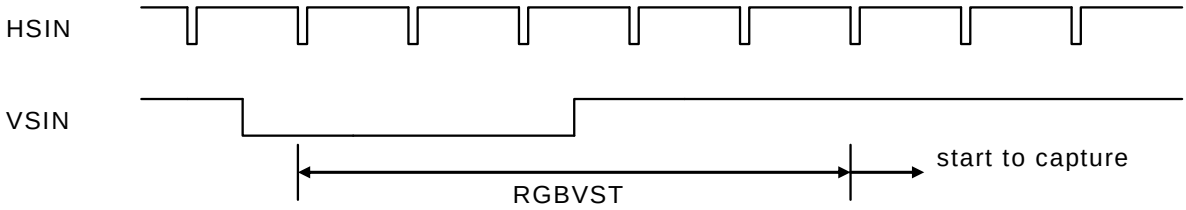
  According to the VP register setting, the rising edge or falling edge of the VSIN signal is used as the VSYNC signal.



Note: RGB_HSYNC is the internal signal image generated by the HSIN signal.

Active pixel data input rule pertaining to VSYNC

The active pixel data input rule pertaining to VSYNC is shown below.



Note: Input of interlaced video is not supported in RGB input mode. Set RGBVST_T_O to "00".

Color Space Conversion

The following matrix formula is used to convert RGB to YCbCr:

Y = a11*R + a12*G + a13*B + b1

Cb = a21*R + a22*G + a23*B + b2 aij: 10-bit signed real (lower 8-bits are fraction)

Cr = a31*R + a32*G + a33*B + b3 bi: 8-bit unsigned integer

Configure RGBCMY, RGBCMCb, RGBCMCr and RGBCMb for the coefficients of the color conversion matrix.

Notes:

- Each coefficient can be defined with a register setting.
- A converted YCbCr signal is further converted to the 4:2:2 format and undergoes internal image processing.

Procedure for Starting the Capture Operation

The capture operation is in the stopped state immediately after the hardware reset. To start the capture operation, follow the procedure below.

1. Software reset for capture

   Select capture as the target of the software reset by using the VCCC register, and execute the software reset with the VCSR register. VCCC is a control register commonly used for display and capture, and it is allocated in the display space. More specifically, configure the registers in the following steps.

STEP 1	Write "1" in [bit2] C0sr (Capture 0 software reset) in the VCCC register. Set "1" in bit2 with a read-modify-write instruction to save [bit13, bit12] Csel (Capture select). To exclude the display part from the reset target, set "0" in V0sr (bit0). Be sure to always select the software reset target module (display part and capture part) immediately before executing the software reset.
STEP 2	Write 0x00000000 in the VCSR register. Immediately after this write operation, execute the software reset for capture.
STEP 3	Execute a dummy read of the VCSR register twice to wait for the completion of software reset.

2. Various parameter settings

   Specify items such as the buffer address and the scaling factor.

3. Starting the capture operation

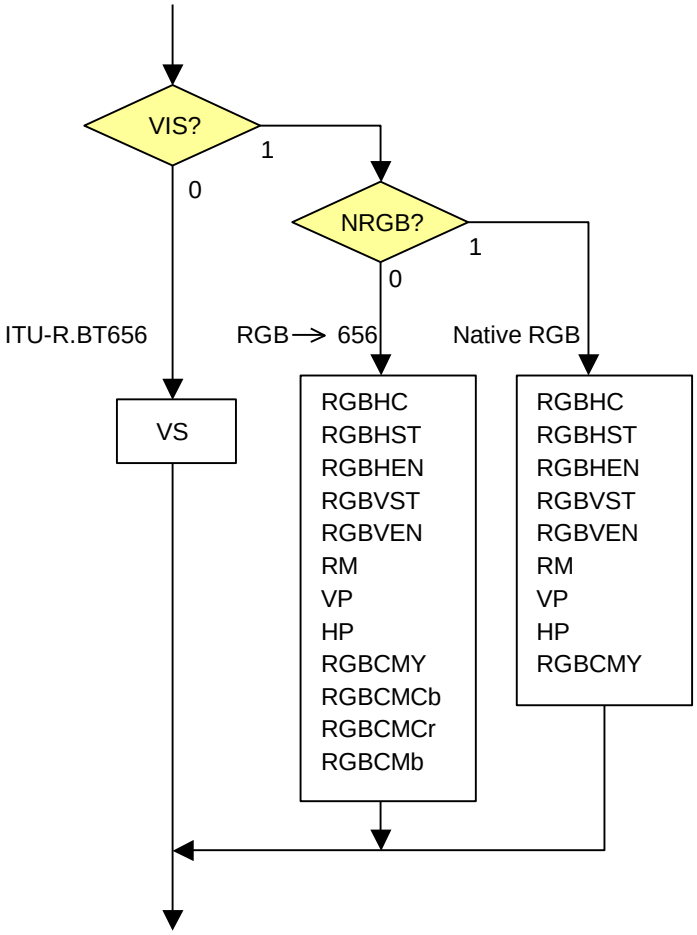
   Leaving the values as is in all VCM register bits except the VIE bit, write "1" in the VIE bit.

When modifying parameter settings, first stop capture by writing "0" in the VIE bit in the VCM register, and then follow the above procedure.

RGB Video Input Parameter Setting Chart

The registers that need parameter settings vary depending on the video input mode. For details, see the following chart.

Register Setting Chart for Each RGB Video Input Mode



1.8 Draw Engine (DRAW)

1.8.1 Overview

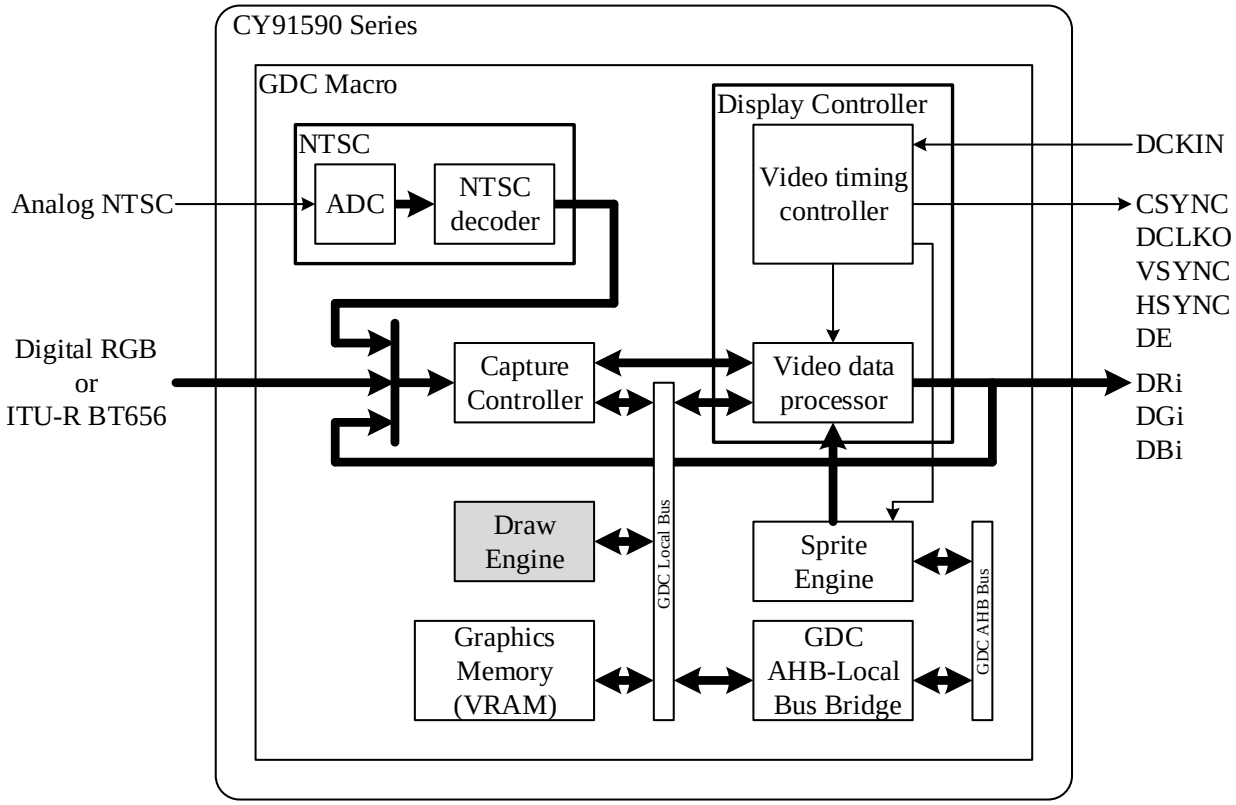
The GDC macro has a line drawing function that is compatible with Cypress's LSI MB86276 and the BitBlt function.

1.8.2 Features

- Line drawing support (line with a width from 1 to 32 pixels, broken line, anti-aliased line)
- BitBlt function support (rectangle fill, copy, character pattern)
- Display list execution support
- 8-bpp indirect color
- ARGB1555 direct color
- Binary bitmap (only for the BitBlt function)
- Alpha-blending function support

1.8.3 Configuration

The following figure shows the location of the draw engine on the system. The draw engine includes 3 blocks as sub-blocks: setup (preprocessing), DDA (middle processing), and pixel engine (post processing).



1.8.4 Registers

1.8.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (0040_0000_H) is added for access from the FR81S (CPU). Use the absolute address shown in the parentheses to write data with the display list command SetRegister.
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field value immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.8.4.2 Register List

The values in the offset field enclosed in parentheses indicate the absolute address used by the SetRegister command.

DrawBase = $01FF_0000_H$

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
140_H (050_H)	LPN																															
	0	0	0	0	Int												0															
144_H (051_H)	LXs																															
	S	S	S	S	Int												Frac															
148_H (052_H)	LXde																															
	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	Int	Frac															
$14C_H$ (053_H)	LYs																															
	S	S	S	S	Int												Frac															
150_H (054_H)	LYde																															
	S	S	S	S	S	S	S	S	S	S	S	S	S	S	S	Int	Frac															

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
200_H (080_H)	RXs																															
	0	0	0	0	Int												0															
204_H (081_H)	RYs																															
	0	0	0	0	Int												0															
208_H (082_H)	RsizeX																															
	0	0	0	0	Int												0															
$20C_H$ (083_H)	RsizeY																															
	0	0	0	0	Int												0															
240_H (090_H)	SADDR																															
	0	0	0	0	0	0	0	Address																								
244_H (092_H)	SStride																															
	0	0	0	0	Int												0															
248_H (092_H)	SRXs																															
	0	0	0	0	Int												0															
$24C_H$ (093_H)	SRYs																															
	0	0	0	0	Int												0															
250_H (094_H)	DADDR																															
	0	0	0	0	0	0	0	Address																								
254_H (095_H)	DStride																															
	0	0	0	0	Int												0															
258_H (096_H)	DRXs																															
	0	0	0	0	Int												0															
$25C_H$ (097_H)	DRYs																															
	0	0	0	0	Int												0															
260_H (098_H)	BRsizeX																															
	0	0	0	0	Int												0															
264_H (099_H)	BRsizeY																															
	0	0	0	0	Int												0															
280_H ($0A0_H$)	TColor																															
	0																Color															

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
$3E0_H$ ($0F8_H$)	BLPO																															
																												BCR				
400_H ($100_{H)}$	CTR																															
								FD	CE				FCNT					NF	FF	FE			SS				DS			PS		
404_H (-)	IFSR																															
																														NF	FF	FE
408_H (-)	IFCNT																															
																												FCNT				
$40C_H$ (-)	SST																															
																															SS	
410_H (-)	DS																															
																															DS	
414_H (-)	PST																															
																															PS	
418_H (-)	EST																															
																														FD	CE	
420_H (108_H)	MDR0																															
												ZP				CF							CY	CX					BSV		BSH	
424_H (109_H)	MDR1																															
				LW								BP	BL							LOG				BM								
430_H ($10C_H$)	MDR4																															
																				LOG				BM						TE	

Offset	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
440_H (110_H)	FBR																															
							FBASE																									
444_H (111_H)	XRES																															
																					XRES											
454_H (115_H)	CXMIN																															
																					CLIPXMIN											
458_H (116_H)	CXMAX																															
																					CLIPXMAX											
$45C_H$ (117_H)	CYMIN																															
																					CLIPYMIN											
460_H (118_H)	CYMAX																															
																					CLIPYMAX											
480_H (120_H)	FC																															
																	FGC8/16															
484_H (121_H)	BC																															
																	BGC8/16															
488_H (122_H)	ALF																															
																									A							
$48C_H$ (123_H)	BLP																															
$4A0_H$ (-)	DFIFO																															
																	BC8/16															
540_H (150_H)	LX0dc																															
	0	0	0	0	Int												0															
544_H (151_H)	LY0dc																															
	0	0	0	0	Int												0															
548_H (152_H)	LX1dc																															
	0	0	0	0	Int												0															
$54C_H$ (153_H)	LY1dc																															
	0	0	0	0	Int												0															

1.8.4.3 Drawing Control Registers

CTR (Control Register)

Register Address	01FF_0400$_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-							FO	CE				FCNT					NF	FF	FE	-		SS		-		DS		-		PS	
R/W								RW	RW				R					R	R	R			R				R				R	
Initial Value								0	00				10000					0	0	1			00				00				00	

This register reflects the flag and status information for the drawing part. The flags from bit24 to bit22 are not cleared until "0" is written to them.

[bit24] FO (FIFO Overflow)

This bit indicates an FIFO overflow occurrence.

0 Normal

1 Overflow detected

[bit23, bit22] CE (Display List Command Error)

These bits indicate a command error occurrence. The bits report the error during decoding. To implement recovery, the GDC macro initialization sequence must be executed.

00 Normal

01 Reserved

10 Reserved

11 Error detected

[bit19 to bit15] FCNT (FIFO Counter)

These bits indicate empty layers of the Display List FIFO (0 to 10000b).

[bit14] NF (FIFO Near Full)

This bit indicates whether less than half of the Display List FIFO is empty.

0 At least half of the FIFO is empty.

1 Less than half of the FIFO is empty.

[bit13] FF (FIFO Full)

This bit indicates whether the Display List FIFO is full.

0 Not full

1 Full

[bit12] FE (FIFO Empty)

This bit indicates whether the Display List FIFO contains no data.

0 Contains data

1 Contains no data

[bit9, bit8] SS (Setup Status)

These bits indicate the status of the setup unit, which is a sub-block.

00 Idle

01 Processing

10 Reserved

11 Reserved

[bit5, bit4] DS (DDA Status)

These bits indicate the status of the DDA unit, which is a sub-block.

00 Idle

01 Processing

10 Processing

11 Reserved

[bit1, bit0] PS (Pixel engine Status)

These bits indicate the status of the pixel engine unit, which is a sub-block.

00 Idle

01 Processing

10 Reserved

11 Reserved

IFSR (Input FIFO Status Register)

Register Address	$01FF_0404_H$																														
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																													NF	FF	FE
R/W																														R	R	R
Initial Value																														0	0	1

This register is a mirror register of bit14 to bit12 in CTR.

IFCNT (Input FIFO Counter)

Register Address	$01FF_0408_H$																														
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																											FCNT				
R/W																												R				
Initial Value																												10000				

This register is a mirror register of bit19 to bit15 in CTR.

SST (Setup Engine Status)

Register Address	$01FF_040C_H$																														
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																														SS	
R/W																															R	
Initial Value																															00	

This register is a mirror register of bit9 and bit8 in CTR.

DST (DDA Status)

Register Address	$01FF_0410_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																														DS	
R/W																															R	
Initial Value																															00	

This register is a mirror register of bit5 and bit4 in CTR.

PST (Pixel Engine Status)

Register Address	$01FF_0414_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																														PS	
R/W																															R	
Initial Value																															00	

This register is a mirror register of bit1 and bit0 in CTR.

EST (Error Status)

Register Address	$01FF_0418_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																													FO	CE	
R/W																														RW	RW	
Initial Value																														0	00	

This register is a mirror register of bit24 to bit22 in CTR.

1.8.4.4 Drawing Mode Registers

Use the SetRegister command to write value in these registers. The registers cannot be accessed from the CPU.

MDR0 (Mode Register for Miscellaneous)

Register Address	$01FF_0420_H$																											
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-											ZP	-			CF		-					CY	CX	-				BSV		BSH	
R/W												RW				RW							RW	RW					RW		RW	
Initial Value	X											0	X			00		X					0	0	X				00		00	

[bit20] ZP (Z Precision)

This bit sets the precision of the Z value used in hidden surface removal. (It has no meaning in this macro.)

0 16 bits/pixel

1 8 bits/pixel

[bit16, bit15] CF (Color Format)

These bits set the format of the drawing color.

00 Indirect color mode (8 bits/pixel)

01 Direct color mode (16 bits/pixel)

10-11 Reserved

[bit9] CY (Clip Y enable)

This bit sets the clip processing in the Y direction.

0 No clipping

1 Clipping

[bit8] CX (Clip X enable)

This bit sets the clip processing in the X direction.

0 No clipping

1 Clipping

[bit3, bit2] BSV (Bitmap Scale Vertical)

These bits set the multiplying factor in the vertical direction in bitmap drawing.

00 Once

01 Twice

10 Half

11 Reserved

[bit1, bit0] BSH (Bitmap Scale Horizontal)

These bits set the multiplying factor in the horizontal direction in bitmap drawing.

00 Once

01 Twice

10 Half

11 Reserved

MDR1 (Mode Register for LINE)

Register Address	$01FF_0424_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-			LW					-			BP	BL	-						LOG				BM		-						
R/W				RW								RW	RW							RW				RW								
Initial Value	X			00000					X			0	0	X						0011				00		X						

This register specifies the line and dot drawing modes.

[bit28 to bit24] LW (Line Width)

These bits set the line width at the line drawing time.

00000 1-pixel width

00001 2-pixel width

: :

11111 32-pixel width

[bit20] BP (Broken line Period)

This bit selects the cycle of the broken line pattern.

0 32 bits

1 24 bits

[bit19] BL (Broken Line)

This bit selects either a solid line or a broken line.

0 Solid line

1 Broken line

[bit12 to bit9] LOG (Logical operation)

These bits set the logical operation mode at the drawing time.

0000 CLEAR
0001 AND
0010 AND REVERSE
0011 COPY
0100 AND INVERTED
0101 NOP
0110 XOR
0111 OR
1000 NOR
1001 EQUIV
1010 INVERT
1011 OR REVERSE
1100 COPY INVERTED
1101 OR INVERTED
1110 NAND
1111 SET

[bit8, bit7] BM (Blend Mode)

These bits set the blend mode.

00 Normal (source copy)
01 Alpha-blending
10 Drawing with logical operations
11 Reserved

MDR4 (Mode Register for BLT)

Register Address	$01FF_0430_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																			LOG				BM		-					TE	-
R/W																				RW				RW							RW	
Initial Value																				0011				00							0	

This register sets the BLT mode.

[bit12 to bit9] LOG (Logical operation)

These bits set the logical operation mode at the drawing time.

0000 CLEAR
0001 AND
0010 AND REVERSE
0011 COPY
0100 AND INVERTED
0101 NOP
0110 XOR
0111 OR
1000 NOR
1001 EQUIV
1010 INVERT
1011 OR REVERSE
1100 COPY INVERTED
1101 OR INVERTED
1110 NAND
1111 SET

[bit8, bit7] BM (Blend Mode)

These bits set the blend mode.

00 Normal (source copy)

01 Reserved

10 Drawing with logical operations

11 Reserved

[bit1] TE (Transparent Enable)

This bit sets the transparent mode.

0 No transparent processing

1 Do not draw pixels matching the set transparent color (transparent copy) at BLT.
(Note) Specify "normal" for the blend mode (BM).

1.8.4.5 Drawing Configuration Registers

FBR (Frame Buffer Base)

Register Address	$01FF_0440_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-						FBASE																									
R/W							RW																								R0	
Initial Value	X						X																								0	

This register stores the base address of the drawing frame.

[bit25 to bit0] FBASE (Frame Base Address)

This field indicates the frame base address compatible with the MB86276.

XRES (X Resolution)

Register Address	$01FF_0444_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																				XRES											
R/W																					RW											
Initial Value																					X											

This register sets the resolution in the X-axis direction on the drawing screen.

CXMIN (Clip X Minimum)

Register Address	$01FF_0454_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																				CLIPXMIN											
R/W																					RW											
Initial Value																					X											

This register sets the minimum X value of the clip frame.

CXMAX (Clip X Maximum)

Register Address	$01FF_0458_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																				CLIPXMAX											
R/W																					RW											
Initial Value																					X											

This register sets the maximum X value of the clip frame.

CYMIN (Clip Y Minimum)

Register Address	01FF_045C$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																				CLIPYMIN											
R/W																					RW											
Initial Value																					X											

This register sets the minimum Y value of the clip frame.

CYMAX (Clip Y Maximum)

Register Address	01FF_0460$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																				CLIPYMAX											
R/W																					RW											
Initial Value																					X											

This register sets the maximum Y value of the clip frame.

FC (Foreground Color)

Register Address	01FF_0480$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	FGC8/16																															
R/W	RW																															
Initial Value	0																															

This register sets the foreground color for drawing.
This color is the object color for flat shading or the foreground color at the bitmap drawing or broken line drawing time.
Pixels with "1" set are drawn in the color set in this register.

- 8-bit color mode:

[bit31 to bit8] Unused bits

[bit7 to bit0] FGC8 (Foreground 8 bit Color)

These bits set indirect color (color index code) for the foreground color.

- 16-bit color mode:

[bit31 to bit16] Unused bits

[bit15 to bit0] FGC16 (Foreground 16 bit Color)

These bits set the 16-bit direct color of the foreground color.
The value set in bit15 is reflected unchanged to memory.

BC (Background Color)

Register Address	01FF_0484$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	BGC8/16																															
R/W	RW																															
Initial Value	0																															

This register sets the background color for drawing.
This setting is for the background color at the broken line drawing time. Pixels with "0" set are drawn in the color set in the register. The background color can be made transparent (no background drawn). The setting can be made with the BT bit in the register.

- 8-bit color mode:

[bit31 to bit16] Unused bits

[bit15] BT (Background Transparency)

This bit sets the transparent mode for the background color.

0 Draw in the set BGC color as the background color.

1 Do not draw the background. (Transparent)

[bit14 to bit8] Unused bits

[bit7 to bit0] BGC8 (Background 8 bit Color)

These bits set indirect color (color index code) for the background.

- 16-bit color mode:

[bit31 to bit16] Unused bits

[bit15] BT (Background Transparency)

This bit sets the transparent mode for the background color.

0 Draw in the set BGC color as the background color.

1 Do not draw the background. (Transparent)

[bit14 to bit0] BGC16 (Background 16 bit Color)

These bits set 16-bit direct color RGB for the background color.

ALF (Alpha Factor)

Register Address	$01FF_0488_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																								A							
R/W																									RW							
Initial Value																									0							

This register sets the alpha-blending coefficient.

BLP (Broken Line Pattern)

Register Address	$01FF_048C_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	BLP																															
R/W	RW																															
Initial Value	0																															

This register sets the broken line pattern.
In the broken pattern, bits with "1" are drawn as the foreground, and bits with "0" are drawn as the background. The broken line pattern corresponding to 1 pixel of a line is assigned to the direction from the MSB to LSB. This pattern returns to the MSB after reaching the LSB. The BLPO register controls the bit number of the broken pattern. Broken pattern repetition can be selected with the BP register in the MDR1 register, and 32 bits or 24 bits can be selected. 24-bit repetition uses bit23 to bit0 in the BLP register.

BLPO (Broken Line Pattern Offset)

Register Address	$01FF_03E0_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-																											BCR				
R/W																												RW				
Initial Value																												11111				

This register stores the bit number of the broken line pattern set in the BLP register at the broken line drawing time. The register is decremented by drawing 1 pixel. Set any value in the register to start a drawing broken pattern from any start position. If no value is written, the broken pattern position is maintained.

1.8.4.6 Rectangular Drawing Registers

The drawing command uses these registers. Access from the CPU and the SetRegister command are not available.

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXs	0200_H	0	0	0	0	Int												0															
RYs	0204_H	0	0	0	0	Int												0															
RsizeX	0208_H	0	0	0	0	Int												0															
RsizeY	$020C_H$	0	0	0	0	Int												0															

Address : Offset value from DrawBaseAddress

S : Sign and sign extension

0 : Unused, or 0 extension

Int : Integer or the integer part of the fixed-point format

Frac : Fraction part of the fixed-point format

These registers set coordinate parameters in rectangular drawings. The foreground color is applied as the color.

RXs	Specifies the X coordinate of the top left of the rectangle to be filled.
RYs	Specifies the Y coordinate of the top left of the rectangle to be filled.
RsizeX	Specifies the size in the X direction of the rectangle to be filled.
RsizeY	Specifies the size in the Y direction of the rectangle to be filled.

1.8.4.7 Blt Registers

The parameters of each register are set as follows.

- Use the SetRegister command to configure the Tcolor register.
  Note that access from the CPU and the drawing command cannot set values in the register.
- Execute the drawing command to configure registers other than the Tcolor register.
  Note that access from the CPU and the SetRegister command cannot set values in these registers.

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR	0240_H	0	0	0	0	0	0	0	Address																								
SStride	0244_H	0	0	0	0	Int												0															
SRXs	0248_H	0	0	0	0	Int												0															
SRYs	$024C_H$	0	0	0	0	Int												0															
DADDR	0250_H	0	0	0	0	0	0	0	Address																								
DStride	0254_H	0	0	0	0	Int												0															
DRXs	0258_H	0	0	0	0	Int												0															
DRYs	$025C_H$	0	0	0	0	Int												0															
BRsizeX	0260_H	0	0	0	0	Int												0															
BRsizeY	0264_H	0	0	0	0	Int												0															
TColor	0280_H	0																Color															

Address : Offset values from DrawBaseAddress

S : Sign and sign extension

0 : Unused, or 0 extension

Int : Integer or the integer part of the fixed-point format

Frac : Fraction part of the fixed-point format

These registers specify parameters for Blt processing.

SADDR	Specifies the start address of the rectangular area of the source. The address is specified with a byte address.
SStride	Specifies the stride of the source.
SRXs	Specifies the start X coordinate of the rectangular area of the source.
SRYs	Specifies the start Y coordinate of the rectangular area of the source.
DADDR	Specifies the start address of the rectangular area of the destination. The address is specified with a byte address.
DStride	Specifies the stride of the destination.
DRXs	Specifies the start X coordinate of the rectangular area of the destination.
DRYs	Specifies the start Y coordinate of the rectangular area of the destination.
BRsizeX	Specifies the X size of the rectangle.
BRsizeY	Specifies the Y size of the rectangle.
TColor	Sets the transparent color. For indirect color, a pallet code is set in the lower 8 bits.

1.8.4.8 Hi-speed 2D Line Drawing Registers

The drawing command uses these registers. Access from the CPU is not available.

Register	Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LX0dc	0540_H	0	0	0	0	Int												0															
LY0dc	0544_H	0	0	0	0	Int												0															
LX1dc	0548_H	0	0	0	0	Int												0															
LY1dc	$054c_H$	0	0	0	0	Int												0															

Address : Offset value from DrawBaseAddress

S : Sign and sign extension

0 : Unused, or 0 extension

Int : Integer or the integer part of the fixed-point format

Frac : Fraction part of the fixed-point format

When a high-speed 2D line drawing is executed, the top coordinates (V0, V1) of the line end point are set.

LX0dc	Sets the X coordinate of the top V0.
LY0dc	Sets the Y coordinate of the top V0.
LX1dc	Sets the X coordinate of the top V1.
LY1dc	Sets the Y coordinate of the top V1.

1.8.4.9 Display List FIFO

DFIFO (Display List FIFO)

Register Address	$01FF_04A0_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	DFIFO																															
R/W	W																															
Initial Value	X																															

This is the FIFO register for display list transfer.

GDC Bus Bridge itself becomes a master with a function for performing DMA transfer of data from the GDC memory (VRAM, CMDRAM) to the Display List FIFO.
For details on the above-mentioned DMA transfer, see "1.21 GDC Bus Bridge".

1.8.5 Explanation of Operation

1.8.5.1 Coordinate System

Drawing Coordinates

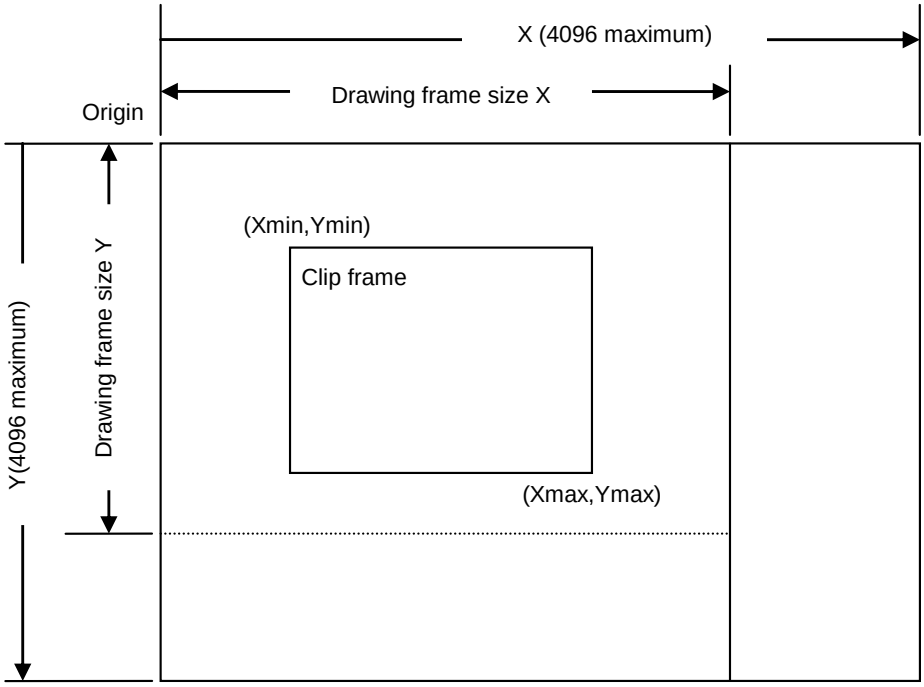
The draw engine draws in the drawing frame in VRAM using drawing coordinates (device coordinates).

The drawing frame is handled as 2-dimensional coordinates as shown below, with the top left point as the origin. It can handle a maximum of 4096 x 4096 coordinates.

The drawing frame is located in VRAM by specifying the origin address of the drawing frame and the resolution (size) in the X direction in the register. It is not necessary to specify the size in the Y direction, but the maximum Y coordinate must not overlap another area at the drawing time.

Also, the drawing can be kept from going outside the clip frame during drawing.

Specify the top left and the bottom right coordinates of the frame for the clip frame.



1.8.5.2 Figure Drawing

Drawing Primitive

The draw engine supports line drawing compatible with Cypress's LSI MB86276 graphics controller and rectangular drawing (BitBlt function).

It can draw 2-dimensional figures in VRAM (drawing screen) in direct color mode and indirect color mode.

It can also draw wide heavy lines and broken lines. It also can draw a smooth diagonal stroke with anti-aliasing.

Table 1-4. 2-dimensional Figure Drawing Primitive

Primitive	Explanation
Line	Draws a line
Rectangle	Draws a rectangle

Drawing Color

The draw engine supports 8-bit indirect color and 16-bit direct color as drawing input data.

Anti-aliasing Function

The draw engine can make jaggies invisible and smooth out lines with anti-aliasing in line drawing. It processes the boundary of lines in units of sub-pixels and blends pixel colors before drawing in color, so that the jaggies on diagonal strokes appear smooth. This can be specified as an attribute of line and high-speed 2-dimensional line drawing. (It is available only for the drawing frame in direct color mode.)

Alpha-blending Function

Alpha-blending describes a transparent effect from blending the colors of 2 images. The transparent coefficient is specified in the register. The specified transparent coefficient is used during the drawing of a primitive.

Blending is not performed on transparent colors, and transparency is maintained. (Alpha-blending in drawing is valid only for the drawing frame in direct color mode.)

Bitmap Processing

BitBlt

BitBlt can transfer a rectangle in units of pixels. The following types of transfer are available:

- Transfer from the FR81S to the drawing frame
- Transfer between VRAM

BitBlt can perform binary logical operations between the source data and the destination data and write the results. It can also draw while passing through particular pixels by specifying the transparent color. If the source data and destination data overlap, the transfer start point must be specified.

Pattern Format

The types of bitmap data handled by the GDC macro include indirect color mode (8 bits/pixel), direct color mode (16 bits/pixel), and binary bitmap (1 bit/pixel).

By using a binary bitmap, character patterns are drawn in the foreground color if the pixel in the bitmap is "1", and in the background color (the background color can be made transparent via a setting) if the bit is "0".

Logical Operations

These are specified to perform a logical operation between pixels to be drawn and pixels already drawn in the frame buffer and to draw the operation results. Alpha-blending cannot be specified through such specification.

Table 1-5. Logical Operations

Operation Format	ID	Behavior	Operation Format	ID	Behavior
CLEAR	0000	0	AND	0001	S & D
COPY	0011	S	OR	0111	S | D
NOP	0101	D	NAND	1110	!(S & D)
SET	1111	1	NOR	1000	!(S | D)
COPY INVERTED	1100	!S	XOR	0110	S xor D
INVERT	1010	!D	EQUIV	1001	!(S xor D)
AND REVERSE	0010	S & !D	AND INVERTED	0100	!S & D
OR REVERSE	1011	S | !D	OR INVERTED	1101	!S | D

1.8.5.3 Figure Drawing Attributes

Line Drawing Attributes

The following attributes can be specified in line drawings.

Table 1-6. Line Drawing Attributes

Drawing Attribute	Function
Line width	A line width between 1 and 32 pixels can be specified. The start point and end point of a heavy line are perpendicular to the principal axis.
Broken line processing	A broken line pattern can be specified. Specify a 32-bit broken line pattern. The broken line pattern is perpendicular to the principal axis.
Anti-alias processing	When anti-alias is enabled, it is possible to draw a smooth line with anti-aliasing.

BitBlt Attributes

The following attributes can be specified for drawing when BLT processing is executed.

Table 1-7. BitBlt Attributes

Drawing Attribute	Function
Logical operation mode	Specifies the binary logical operation mode.
Transparent mode	Specifies the transparent copy mode and the transparent color.

Character Pattern Drawing Attributes

Table 1-8. Character Pattern Drawing and Attributes

Drawing Attribute	Function
Character pattern enlarging/shrinking	Can specify 2 times larger vertically and horizontally, 2 times larger horizontally, 1/2 smaller vertically and horizontally, and 1/2 smaller horizontally.
Character pattern color	Specifies the character color (foreground color) and the background color.
Transparency	Specifies whether the background color is transparent or non-transparent.

1.8.5.4 Display List

A collection of display list commands, parameters, and pattern data is called a display list. Display list commands stored in FIFO are executed and processed sequentially.

The display list is transferred to the Display List FIFO. Any of the following transfer methods is used:

- Writing to FIFO by the FR81S.
- Transfer from FR81S memory to FIFO using the FR81S DMA controller.
- Transfer from VRAM to FIFO using the register setting for local transfer in the GDC macro.

Table 1-9. Display List

Displaylist Command-1
Data1-1
Data1-2
Data1-3
Displaylist Command-2
Data2-1
Data2-2
Data2-3
...

Header Format

This section shows the format of the display list headers.

Format list

Format	31 24	23 16	15 0
Format 1	Type	Reserved	Reserved
Format 2	Type	Count	Address
Format 5	Type	Command	Reserved
Format 7	Type	Command	Reserved / Vertex
Format 9	Type	Reserved	Reserved / Flag

Table 1-10. Explanation of Fields

Type	Display List Type
Command	Command
Count	Parameter data count excluding the header part
Address	Address at data transfer time, etc.
Vertex	Vertex number
Flag	Attribute flag specific to display list command

Table 1-11. Vertex Numbers Specified in the Vertex Code

Vertex	Vertex Number (Line)	Vertex Number (Triangle)
00	V0	V0
01	V1	V1
10	Setting prohibited	V2
11	Setting prohibited	Setting prohibited

Parameter Format

The parameters of the rendering commands apply the data formats compatible with the MB86276.

Rendering Commands

Command List

The following shows the rendering commands and command codes in the draw engine.

Type	Command	Explanation
Nop	-	No operation
Interrupt	-	Generation of an interrupt to MCNT
Sync	-	Synchronization with an event
SetRegister	-	Data set to a register
Draw	Flush_FB/Z	Flushing of the drawing pipeline
DrawLine	Xvector	Line drawing (principal axis X)
	Yvector	Line drawing (principal axis Y)
	AntiXvector	Line drawing with anti-aliasing (principal axis X)
	AntiYvector	Line drawing with anti-aliasing (principal axis Y)
DrawLine2i	ZeroVector	High-speed 2D line drawing (start point of vertex 0)
DrawLine2iP	OneVector	High-speed 2D line drawing (start point of vertex 1)
DrawRectP	BltFill	Rectangular drawing in a single color
DrawBitmapP	BltDraw	Blt drawing (with 16 bits specified for the number of parameters)
	Bitmap	Binary bitmap (character) drawing
BltCopyP BltCopy- AlternateP	TopLeft	Blt transfer from the top left coordinates
	TopRight	Blt transfer from the top right coordinates
	BottomLeft	Blt transfer from the bottom left coordinates
	BottomRight	Blt transfer from the bottom right coordinates

Type code table

Type	Code
DrawLine	0000_0010
DrawLine2i	0000_0011
DrawLine2iP	0000_0100
DrawRectP	0000_1001
DrawBitmapP	0000_1011
BitCopyP	0000_1101
BitCopyAlternateP	0000_1111
Draw	1111_0000
SetRegister	1111_0001
Sync	1111_1100
Interrupt	1111_1101
Nop	1111_1111

Command code table (1)

Command	Code
Xvector	001_00000
Yvector	001_00001
XvectorNoEnd	001_00010
YvectorNoEnd	001_00011
XvectorBlpClear	001_00100
YvectorBlpClear	001_00101
XvectorNoEndBlpClear	001_00110
YvectorNoEndBlpClear	001_00111
AntiXvector	001_01000
AntiYvector	001_01001
AntiXvectorNoEnd	001_01010
AntiYvectorNoEnd	001_01011
AntiXvectorBlpClear	001_01100
AntiYvectorBlpClear	001_01101
AntiXvectorNoEndBlpClear	001_01110
AntiYvectorNoEndBlpClear	001_01111
ZeroVector	001_10000
Onevector	001_10001
ZeroVectorNoEnd	001_10010
OnevectorNoEnd	001_10011
ZeroVectorBlpClear	001_10100
OnevectorBlpClear	001_10101
ZeroVectorNoEndBlpClear	001_10110
OnevectorNoEndBlpClear	001_10111

Command	Code
AntiZeroVector	001_11000
AntiOnevector	001_11001
AntiZeroVectorNoEnd	001_11010
AntiOnevectorNoEnd	001_11011
AntiZeroVectorBlpClear	001_11100
AntiOnevectorBlpClear	001_11101
AntiZeroVectorNoEndBlpClear	001_11110
AntiOnevectorNoEndBlpClear	001_11111

Command code table (2)

Command	Code
BltFill	010_00001
Flush_FB	110_00001

Explanation of Rendering Commands

Parameters belonging to these command are stored in the corresponding registers. For the meaning of the parameters, see the explanation of the registers.

Nop (Format 1)

31 24	23 16	15 0
Nop	Reserved	Reserved

There is no operation.

Interrupt (Format 1)

31 24	23 16	15 0
Interrupt	Reserved	Reserved

An interrupt to MCNT is generated.

Sync (Format 9)

31 24	23 16	15 4	0
Sync	Reserved	Reserved	Flag

The subsequent display list processing halts until the event specified in the flag is detected.

Flag:

Bit Number	4	3	2	1	0
Bit Field Name	Reserved	Reserved	Reserved	Reserved	VBLANK

[bit0] VBLANK

VBLANK synchronization

0 No operation

1 Processing halts until the vertical blank specified in the FS field in the MDR0 register is detected.

SetRegister (Format 2)

31 24	23 16	15 0
SetRegister	Count	Address
(Val 0)		
(Val 1)		
...		
(Val n)		

Data is set to consecutive registers.

Count : Data count (in units of double words)

Address : Set the address value for SetRegister in the register address register list.
Set the first register address when transferring two or more data items.

Draw (Format 5)

31 24	23 16	15 0
Draw	Command	Reserved

The drawing command is executed. The parameters required for executing the command need to be specified by SetRegister, etc.

Command:

Flush_FB	Flushes drawing data in the drawing pipeline to VRAM. Insert this command at the end of the display list.

DrawLine (Format 5)

31 24	23 16	15 0
DrawLine	Command	Reserved
LPN		
LXs		
LXde		
LYs		
LYde		

These commands draw lines. Drawing starts after parameters are set in the register for line drawing.

Command:

Xvector	Draws a line. (Principal axis X)
Yvector	Draws a line. (Principal axis Y)
XvectorNoEnd	Draws a line. (Principal axis Y. No end point is drawn.)
YvectorNoEnd	Draws a line. (Principal axis Y. No end point is drawn.)
XvectorBlpClear	Draws a line. (Principal axis X. The reference position of the broken line pattern is cleared before drawing begins.)
YvectorBlpClear	Draws a line. (Principal axis Y. The reference position of the broken line pattern is cleared before drawing begins.)
XvectorNoEndBlpClear	Draws a line. (Principal axis X. No end point is drawn. The reference position of the broken line pattern is cleared before drawing begins.)
YvectorNoEndBlpClear	Draws a line (Principal axis Y. No end point is drawn. The reference position of the broken line pattern is cleared before drawing begins.)
AntiXvector	Draws a line with anti-aliasing. (Principal axis X)
AntiYvector	Draws a line with anti-aliasing. (Principal axis Y)
AntiXvectorNoEnd	Draws a line with anti-aliasing. (Principal axis X. No end point is drawn.)
AntiYvectorNoEnd	Draws a line with anti-aliasing. (Principal axis Y. No end point is drawn.)
AntiXvectorBlpClear	Draws a line with anti-aliasing. (Principal axis X. The reference position of the broken line pattern is cleared before drawing begins.)
AntiYvectorBlpClear	Draws a line with anti-aliasing. (Principal axis Y. The reference position of the broken line pattern is cleared before drawing begins.)
AntiXvectorNoEndBlpClear	Draws a line with anti-aliasing. (Principal axis X. No end point is drawn. The reference position of the broken line pattern is cleared before drawing begins.)
AntiYvectorNoEndBlpClear	Draws a line with anti-aliasing. (Principal axis Y. No end point is drawn. The reference position of the broken line pattern is cleared before drawing begins.)

DrawLine2i (Format 7)

31 24	23 16	15 0	
DrawLine2i	Command	Reserved	Vertex
LFXs		0	
LFYs		0	

These commands perform high-speed 2D line drawing. Drawing begins after parameters are set in the Hi-speed 2D Line Drawing register. Only integers are available for coordinates.

Command:

ZeroVector	Draws a line from vertex 0 to vertex 1.
OneVector	Draws a line from vertex 1 to vertex 0.
ZeroVectorNoEnd	Draws a line from vertex 0 to vertex 1 (No end point is drawn.)
OneVectorNoEnd	Draws a line from vertex 1 to vertex 0 (No end point is drawn.)
ZeroVectorBlpClear	Draws a line from vertex 0 to vertex 1. (The reference position of the broken line pattern is cleared before drawing begins.)
OneVectorBlpClear	Draws a line from vertex 1 to vertex 0. (The reference position of the broken line pattern is cleared before drawing begins.)
ZeroVectorNoEndBlpClear	Draws a line from vertex 0 to vertex 1. (No end point drawn. The reference position of the broken line pattern is cleared before drawing begins.)
OneVectorNoEndBlpClear	Draws a line from vertex 1 to vertex 0 (No end point drawn. The reference position of the broken line pattern is cleared before drawing begins.)
AntiZeroVector	Draws a line with anti-aliasing from vertex 0 to vertex 1.
AntiOneVector	Draws a line with anti-aliasing from vertex 1 to vertex 0.
AntiZeroVectorNoEnd	Draws a line with anti-aliasing from vertex 0 to vertex 1 (No end point is drawn.)
AntiOneVectorNoEnd	Draws a line with anti-aliasing from vertex 1 to vertex 0 (No end point is drawn.)
AntiZeroVectorBlpClear	Draws a line with anti-aliasing from vertex 0 to vertex 1. (The reference position of the broken line pattern is cleared before drawing begins.)
AntiOneVectorBlpClear	Draws a line with anti-aliasing from vertex 1 to vertex 0. (The reference position of the broken line pattern is cleared before drawing begins)
AntiZeroVectorNoEndBlpClear	Draws a line with anti-aliasing from vertex 0 to vertex 1. (No end point drawn. The reference position of the broken line pattern is cleared before drawing begins.)
AntiOneVectorNoEndBlpClear	Draws a line with anti-aliasing from vertex 1 to vertex 0. (No end point drawn. The reference position of the broken line pattern is cleared before drawing begins.)

DrawLine2iP (Format 7)

31 24	23 16	15 0	
DrawLine2iP	Command	Reserved	Vertex
LFYs		LFXs	

These commands perform high-speed 2-dimensional line drawing. Drawing begins after parameters are set in the Hi-speed 2D Line Drawing register. Only integers (in packed format) are available for coordinates.

Command:

ZeroVector	Draws a line from vertex 0 to vertex 1.
OneVector	Draws a line from vertex 1 to vertex 0.
ZeroVectorNoEnd	Draws a line from vertex 0 to vertex 1 (No end point is drawn.)
OneVectorNoEnd	Draws a line from vertex 1 to vertex 0 (No end point is drawn.)
ZeroVectorBlpClear	Draws a line from vertex 0 to vertex 1. (The reference position of the broken line pattern is cleared before drawing begins.)
OneVectorBlpClear	Draws a line from vertex 1 to vertex 0. (The reference position of the broken line pattern is cleared before drawing begins.)
ZeroVectorNoEndBlpClear	Draws a line from vertex 0 to vertex 1. (No end point drawn. The reference position of the broken line pattern is cleared before drawing begins.)
OneVectorNoEndBlpClear	Draws a line from vertex 1 to vertex 0 (No end point drawn. The reference position of the broken line pattern is cleared before drawing begins.)
AntiZeroVector	Draws a line with anti-aliasing from vertex 0 to vertex 1.
AntiOneVector	Draws a line with anti-aliasing from vertex 1 to vertex 0.
AntiZeroVectorNoEnd	Draws a line with anti-aliasing from vertex 0 to vertex 1. (No end point drawn.)
AntiOneVectorNoEnd	Draws a line with anti-aliasing from vertex 1 to vertex 0. (No end point drawn.)
AntiZeroVectorBlpClear	Draws a line with anti-aliasing from vertex 0 to vertex 1. (The reference position of the broken line pattern is cleared before drawing begins.)
AntiOneVectorBlpClear	Draws a line with anti-aliasing from vertex 1 to vertex 0. (The reference position of the broken line pattern is cleared before drawing begins.)
AntiZeroVectorNoEndBlpClear	Draws a line with anti-aliasing from vertex 0 to vertex 1. (No end point drawn. The reference position of the broken line pattern is cleared before drawing begins.)
AntiOneVectorNoEndBlpClear	Draws a line with anti-aliasing from vertex 1 to vertex 0. (No end point drawn. The reference position of the broken line pattern is cleared before drawing begins.)

DrawRectP (Format 5)

31 24	23 16	15 0
DrawRectP	Command	Reserved
RYs		RXs
RsizeY		RsizeX

This command performs rectangle drawing. The rectangular area is filled with the current color after parameters are set in the register.

Specify an 8-byte aligned value in XRES (X resolution) when this command is executed.

Command:

BltFill	Fills the rectangular area with the current color (a single color).

DrawBitmapP (Format6)

31 24	23 16	15 0
DrawBitmapP	Command	Count
RYs		RXs
RsizeY		RsizeX
(Pattern 0)		
(Pattern 1)		
...		
(Pattern n)		

These commands perform rectangular pattern drawing.

Specify an 8-byte aligned value in XRES when this command is executed.

Command:

BltDraw : This command draws a pattern with 8-bit or 16-bit pixels.

DrawBitmap : This command draws a binary bitmap character pattern.
A rectangular pattern is drawn in a transparent color or the color set in the background color register when "0" is set.
Alternatively, it is drawn in the color set in the foreground color register when "1" is set.

Parameter:

RXs	Specifies the top left X coordinate of the rectangle to be filled. The specifiable value range is 0 to 4095.
RYs	Specifies the top left Y coordinate of the rectangle to be filled. The specifiable value range is 0 to 4095.
RsizeX	Specifies the size in the X direction of the rectangle to be filled. <BSH in the MDR0 register> <effective range> 00 1 time The specifiable value range is 1 to 4095. 01 2 times The specifiable value range is 1 to 2047. 10 1/2 times The specifiable value range is 2 to 4095.
RsizeY	Specifies the size in the Y direction of the rectangular to be filled. <BSV in the MDR0 register><effective range> 00 1 time The specifiable value range is 1 to 4095. 01 2 times The specifiable value range is 1 to 2047. 10 1/2 times The specifiable value range is 2 to 4095.

BltCopyP (Format5)

31 24	23 16	15 0
BltCopyP	Command	Reserved
SRYs		SRXs
DRYs		DRXs
BRsizeY		BRsizeX

These commands copy a rectangular pattern in the drawing frame.

Specify an 8-byte aligned value in XRES when this command is executed.

Command:

TopLeft : This command starts a BitBlt transfer from the top left coordinates.
TopRight : This command starts a BitBlt transfer from the top right coordinates.
BottomLeft : This command starts a BitBlt transfer from the bottom left coordinates.
BottomRight : This command starts a BitBlt transfer from the bottom right coordinates.

Parameter:

SRXs	Specifies the start X coordinate of the rectangular area of the source. The specifiable value range is 0 to 4095.
SRYs	Specifies the start Y coordinate of the rectangular area of the source. The specifiable value range is 0 to 4095.
DRXs	Specifies the start X coordinate of the rectangular area of the destination. The specifiable value range is 0 to 4095.
DRYs	Specifies the start Y coordinate of the rectangular area of the destination. The specifiable value range is 0 to 4095.
BRsizeX	Specifies the X size of the rectangle. The specifiable value range is 1 to 4096.
BRsizeY	Specifies the Y size of the rectangle. The specifiable value range is 1 to 4096.

BltCopyAlternateP (Format5)

31 24	23 16	15 0
BltCopyAlternateP	Command	Reserved
SADDR		
SStride		
SRYs		SRXs
DADDR		
DStride		
DRYs		DRXs
BRsizeY		BRsizeX

This command copies a rectangular pattern between different drawing frames.

Specify an 8-byte aligned value in XRES when this command is executed.

Specify 8-byte aligned values for SStride and DStride values.

Command:

TopLeft : This command starts a BitBlt transfer from the top left coordinates.
The command does not execute a drawing clip.

Parameter:

SADDR	Specifies the start address of the rectangular area of the source. The address is specified with a byte address.
SStride	Specifies the stride of the source.
SRXs	Specifies the start X coordinate of the rectangular area of the source. The specifiable value range is 0 to 4095.
SRYs	Specifies the start Y coordinate of the rectangular area of the source. The specifiable value range is 0 to 4095.
DADDR	Specifies the start address of the rectangular area of the destination. The address is specified with a byte address.
DStride	Specifies the stride of the destination.
DRXs	Specifies the start X coordinate of the rectangular area of the destination. The specifiable value range is 0 to 4095.
DRYs	Specifies the start Y coordinate of the rectangular area of the destination. The specifiable value range is 0 to 4095.
BRsizeX	Specifies the X size of the rectangle. The specifiable value range is 1 to 4096.
BRsizeY	Specifies the Y size of the rectangle. The specifiable value range is 1 to 4096.

1.8.5.5 Interrupts

The following interrupt events are generated in the draw engine. They have status flags corresponding to the INTST register of MCNT, which can manage synchronization with each event.

Event		MCNT/INTST support
Command error	This event is generated when an error is detected during display list decoding. Operation cannot be guaranteed when an error occurs. The DrawEngine block needs to be reset to be restored.	INT11 (bit11)
INT command execution	This event is generated when the display list command Interrupt is executed.	INT12 (bit12)
DMA command execution	This interrupt is reserved for extended implementation. This event is not generated as long as the draw engine is used correctly.	INT13 (bit13)
BUS access error	This event is generated when a BUS protocol error occurs during access from the draw engine to GDC Local BUS.	INT14 (bit14)

1.9 Sprite Engine (SPE)

This section describes the sprite engine (SPE) of the GDC macro.

1.9.1 Overview

The sprite engine (SPE) is used to create images with the GDC macro.
Many kinds of image processing methods can be used according to various register settings by software.

1.9.2 Features

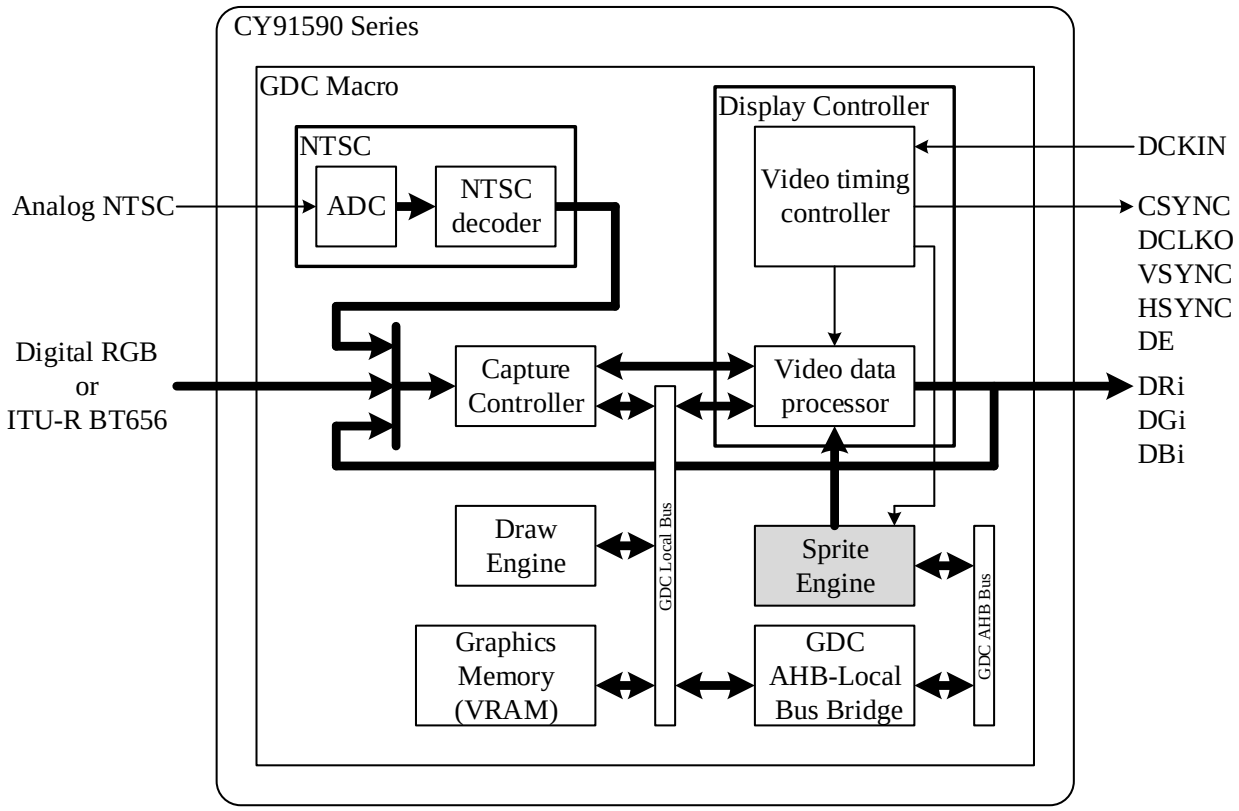
- Maximum of 512 sprites
- Single-color sprite (rectangular drawing) and font mode sprite
- 32 special sprites that enable automatic animation
- Maximum of 16,777,216 colors
- Individual setting possible for 2-color palette modes:
- 16 bpp (ARGB4444/ARGB1555/RGB565), 32 bpp (ARGB8888)
- 8 bpp color palette tables (ARGB4444/ARGB1555/RGB565), maximum of 8
- 1 bpp, 2 bpp, 4 bpp, and 8 bpp indirect color
- ARGB1555, RGB565, ARGB8888 direct color
- Can set a color format per sprite
- Alpha-blending (4/8 bpp alpha)
- Image reverse function (vertical/horizontal)

1.9.3 Configuration

1.9.3.1 Block Diagram

Figure 1-23 is a block diagram showing the SPE.

Figure 1-23. SPE Block Diagram



Notes:

- Analog NTSC corresponds to external pin VIN.
- Digital RGB corresponds to external pins PA2 to PA7, PB2 to PB7, and PC2 to PC7.
- ITU-R.BT656 corresponds to external pins PA2 to PA7, PB2, and PB3.
- DCLKI corresponds to external pin DCKIN.
- CSYNC corresponds to external pin PG3.
- DCLKO corresponds to external pin PG4.
- VSYNC corresponds to external pin PG5.
- HSYNC corresponds to external pin PG6.
- DE corresponds to external pin PG7.
- DR*i* corresponds to external pins P011, P012, and PD2 to PD7.
- DG*i* corresponds to external pins P013, P014, and PE2 to PE7.
- DB*i* corresponds to external pins P015, P016, and PF2 to PF7.

1.9.4 Registers

1.9.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (0040_0000_H) is added for access from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field value immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.9.4.2 Register List

Table 1-12. SPE Register List

Address	Register Name	Description
Function Register		
$01FB_8000_H$	SPEBG (S)	Background color definition register
$01FB_8004_H$	SPETP0 (S)	Transparent color definition register 0 (indirect)
$01FB_8008_H$	SPETP1 (S)	Transparent color definition register 1 (direct)
$01FB_800C_H$	SPEABC (S)	Alpha blend single-color definition register
$01FB_8010_H$	SPECTL	Sprite Engine control register
$01FB_8014_H$	SPEST	Status of the sprite engine register
$01FB_8018_H$	SPEILN	Interrupt line (0 to 4095) specify register
$01FB_801C_H$	SPESDN (S)	Sprite number range for display
$01FB_8020_H$	SPEDETC (S)	Sprite display enable table clear register
$01FB_8024_H$	SPEDPAR	Display area setting register
$01FB_8028_H$	SPEPBA	Pattern memory base address register
$01FB_802C_H$	SPERSC	Resource FIFO enable register
SDET (Sprite Display Enable Table)		
$01FB_8100_H$	SPESDE0 (S)	Sprite display enable (SPR0 to SPR31)
$01FB_8104_H$	SPESDE1 (S)	Sprite display enable (SPR32 to SPR63)
...	...	...
$01FB_813C_H$	SPESDE15 (S)	Sprite display enable (SPR480 to SPR511)
SSCR (Special Sprite Configure Register)		
$01FB_8200_H$	SPESS0CR3	Special Sprite 0 control register 3
$01FB_8204_H$	SPESS0CR4	Special Sprite 0 control register 4
$01FB_8208_H$	SPESS0CR5	Special Sprite 0 control register 5
...	...	...
$01FB_8374_H$	SPESS31CR3	Special Sprite 31 control register 3
$01FB_8378_H$	SPESS31CR4	Special Sprite 31 control register 4
$01FB_837C_H$	SPESS31CR5	Special Sprite 31 control register 5
$01FB_8400_H$	SPESSN0	Special sprite specify register 0
$01FB_8404_H$	SPESSN1	Special sprite specify register 1
...	...	...
$01FB_847C_H$	SPESSN31	Special sprite specify register 31

Address	Register Name	Description
SPT (Sprite Priority Table)		
$01FB_9000_H$	SPESPRI0	Priority 0 set register
$01FB_9004_H$	SPESPRI1	Priority 1 set register
$01FB_9008_H$	SPESPRI2	Priority 2 set register
...	...	...
$01FB_97FC_H$	SPESPRI511	Priority 511 set register
SAT (Sprite Attribute Table)		
$01FB_A000_H$	SPES0CR0	Sprite 0 control register 0
$01FB_A004_H$	SPES0CR1	Sprite 0 control register 1
$01FB_A008_H$	SPES0CR2	Sprite 0 control register 2
...	...	...
$01FB_ABF0_H$	SPES254CR2	Sprite 254 control register 2
$01FB_ABF4_H$	SPES255CR0	Sprite 255 control register 0
$01FB_ABF8_H$	SPES255CR1	Sprite 255 control register 1
$01FB_ABFC_H$	SPES255CR2	Sprite 255 control register 2
...	...	...
$01FB_B7F4_H$	SPES511CR0	Sprite 511 control register 0
$01FB_B7F8_H$	SPES511CR1	Sprite 511 control register 1
$01FB_B7FC_H$	SPES511CR2	Sprite 511 control register 2
LUT (Look-up Table)		
$01FB_E000_H$	SPELUTS0	LUT Setting register 0
$01FB_E004_H$	SPELUTS1	LUT Setting register 1
$01FB_E008_H$	SPELUTS2	LUT Setting register 2
...	...	...
$01FB_EFFC_H$	SPELUTS1023	LUT Setting register 511

Notes:

- **(S)** written next to a register name indicates a shadow register.
- The shadow registers are loaded at the next VSYNC timing.
- All of the reserved or unused address locations cannot be accessed.

1.9.4.3 Register Details

Background Color Register (SPEBG)

Address	01FB_8000$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-								RVAL[7:0]							
R/W	R/W								R/W							
Initial	00$_H$								00$_H$							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GVAL[7:0]								BVAL[7:0]							
R/W	R/W								R/W							
Initial	00$_H$								00$_H$							

Bit Field		**Explanation**
No.	**Name**	
7-0	BVAL[7:0]	Blue color definition of background
15-8	GVAL[7:0]	Green color definition of background
23-16	RVAL[7:0]	Red color definition of background

Notes:

- This register is activated only if SPECTL:BGM is "0".
- The register is a shadow register.
- Shadow registers are loaded at the timing of the next VSYNC after SRE=1 in the SPECTL register and SRL=1 in the SPECTL register are set.

Transparent Color Register 0 (SPETP0) (Indirect Color Mode)

Address	$01FB_8004_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-	TP1B	TP2B[1:0]		TP4B[3:0]				TP8B[7:0]							
R/W	R/W	R/W	R/W		R/W				R/W							
Initial	0_B	0_B	0_H		0_H				00_H							

Bit Field		Explanation
No.	Name	
7-0	TP8B[7:0]	Transparent color code definition in 8-bit color mode
11-8	TP4B[3:0]	Transparent color code definition in 4-bit color mode
13-12	TP2B[1:0]	Transparent color code definition in 2-bit color mode
14	TP1B	Transparent color code definition in 1-bit color mode

Notes:

- This register is a shadow register.
- Shadow registers are loaded at the timing of the next VSYNC after SRE = 1 in the SPECTL register and SRL=1 in the SPECTL register are set.

Transparent Color Register 1 (SPETP1) (Direct Color Mode)

Address	01FB_8008$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-								RVAL[7:0]							
R/W	R/W								R/W							
Initial	00_H								00_H							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GVAL[7:0]								BVAL[7:0]							
R/W	R/W								R/W							
Initial	00_H								00_H							

Bit Field		Explanation
No.	**Name**	
7-0	BVAL[7:0]	Blue color definition of transparent color in direct color mode
15-8	GVAL[7:0]	Green color definition of transparent color in direct color mode
23-16	RVAL[7:0]	Red color definition of transparent color in direct color mode

Notes:

- This register is a shadow register.
- Shadow registers are loaded at the timing of the next VSYNC after SRE=1 in the SPECTL register and SRL=1 in the SPECTL register are set.

Alpha-Blending Color Register (SPEABC)

Address	01FB_800C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-								RVAL[7:0]							
R/W	R/W								R/W							
Initial	00$_H$								00$_H$							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GVAL[7:0]								BVAL[7:0]							
R/W	R/W								R/W							
Initial	00$_H$								00$_H$							

Bit Field		Explanation
No.	Name	
7-0	BVAL[7:0]	Blue color definition of single-color alpha-blending
15-8	GVAL[7:0]	Green color definition of single-color alpha-blending
23-16	RVAL[7:0]	Red color definition of single-color alpha-blending

Notes:

- This register is a shadow register.
- Shadow registers are loaded at the timing of the next VSYNC after SRE=1 in the SPECTL register and SRL=1 in the SPECTL register are set.
- This register is used when alpha-blending mode is set to a mode allowing blending with single color.

Control Register (SPECTL)

Address	01FB_8010$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W															
Initial	0000$_H$															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-			EBCE	BERE	PERE	ILNE	LBKE	-		SRL	SSPE	BGM	SRE	PRI	SEN
R/W	R/W			R/W	R/W	R/W	R/W	R/W	R/W		R/W	R/W	R/W	R/W	R/W	R/W
Initial	0$_H$			0$_B$	0$_B$	0$_B$	0$_B$	0$_B$	0$_H$		0$_B$	0$_B$	0$_B$	0$_B$	0$_B$	0$_B$

Bit Field		Explanation
No.	**Name**	
0	SEN	Enables SPE processing. 1: Enable 0: Disable
1	PRI	Selects the priority mode of sprite display. 1: SPT mode 0: Fixed mode Fixed with a sprite number: SPR0 < SPR1 < ... < SPR511
2	SRE	Enables shadow register updates. 1: Enable 0: Disable
3	BGM	Selects the background mode. 1: Fixed value 000001$_H$ 0: Data set in the SPEBG register
4	SSPE	Enables the special sprite function. 1: Enable 0: Disable
5	SRL	Activates shadow register loading. 1: Load the shadow register at the next VSYNC. 0: Do not load the shadow register.
8	LBKE	Enables line blank interrupts. 1: Enable 0: Disable
9	ILNE	Enables line interrupts. 1: Enable 0: Disable
10	PERE	Enables the processing error interrupt flag. 1: Enable 0: Disable
11	BERE	Enables the bus error interrupt flag. 1: Enable 0: Disable
12	EBCE	Enables "Enable Bits Changed" interrupts. 1: Enable 0: Disable

Notes:

- Set **SEN** when the initial setting of other function registers is completed so that the pattern data is ready.
- If SRE is set to "1", all shadow registers only are updated at the next VSYNC after **SRL** is set to "1".
- The "Enable Bits Changed" interrupt is the timing flag for updating the SPEDE0 to SPEDE15 registers by using shadow registers.

Status Register (SPEST)

Address	$01FB_8014_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-														SSUP	BSY
R/W	R														R	R
Initial	0000_H														0_B	0_B

Bit Field		**Explanation**
No.	**Name**	
0	BSY	Busy flag of sprite engine 1: The SPE is busy. 0: The SPE is idle.
1	SSUP	Whether there is period for updating special sprite attributes with SPE 1: Update. 0: Do not update.

Interrupt Line Register (SPEILN)

Address	$01FB_8018_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-				ILIN[11:0]											
R/W	R/W				R/W											
Initial	0_H				000_H											

Bit Field		Explanation
No.	Name	
11-0	ILIN[11:0]	Interrupt line number (0 to 4095)

Sprite Display Number (SPESDN)

Address	01FB_801C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-							ESN[8:0]								
R/W	R/W							R/W								
Initial	00$_H$							000$_H$								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-							BSN[8:0]								
R/W	R/W							R/W								
Initial	00$_H$							000$_H$								

Bit Field		Explanation
No.	Name	
8-0	BSN[8:0]	SPECTL.PRI=1: SPT start address (32-bit word address, 0 to 511) SPECTL.PRI=0: Start sprite number for display (SPR0 to SPR511)
24-16	ESN[8:0]	SPECTL.PRI=1: SPT end address (32-bit word address, 0 to 511) SPECTL.PRI=0: End sprite number for display (SPR0 to SPR511)

Notes:

- The function of this register is different when the **SPECTL** register is set.
- For **ESN**, set a value larger than that for **BSN**.
- This register is a shadow register.
- Shadow registers are loaded at the timing of the next VSYNC after SRE = 1 in the SPECTL register and SRL=1 in the SPECTL register are set.

Sprite Display Enable Table Clear Register (SPEDETC)

Address	$01FB_8020_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-							ESN[8:0]								
R/W	R/W							R/W								
Initial	00_H							000_H								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GO	DEV	-					BSN[8:0]								
R/W	R/W	R/W	R/W					R/W								
Initial	0_B	0_B	00_H					000_H								

Bit Field		Explanation
No.	Name	
8-0	BSN[8:0]	Start sprite number (SPR0 to SPR511)
14	DEV	SDET (SpriteDisplayEnableTable) clear value 1: Display. 0: Do not display.
15	GO	Whether to start clearing SDET (SpriteDisplayEnableTable) 1: Execute (This bit automatically changes to 0 after execution.) 0: There is no operation.
24-16	ESN[8:0]	End sprite number (SPR0 to SPR511)

Notes:

- This register is a shadow register.
- Shadow registers are loaded at the timing of the next VSYNC after SRE=1 in the SPECTL register and SRL=1 in the SPECTL register are set.

Display Area Register (SPEDPAR)

Address	$01FB_8024_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W															
Initial	$0000_0000_0000_0000_B$															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-					DPWD[10:0]										
R/W	R/W					R/W										
Initial	0_0000_B					$000_0000_0000_B$										

Bit Field		Explanation
No.	Name	
10-0	DPWD[10:0]	Display width: (1 to 50) * 16 pixels – 1 DPWD[3:0] is fixed at "0".

Pattern Memory Base Address, SPEPBA

Address	$01FB_8028_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								PBA[7:0]							
R/W	R/W								R/W							
Initial	00_H								00_H							

Bit Field		Explanation
No.	Name	
7-0	PBA[7:0]	Base address for pattern data: PBA[7:0] x 16 MB **Note:** For the CY91590 series, the maximum size for connected external FLASH is 64MB, which amounts to less than 128MB with another GDC-related area added. Therefore, the base address of the pattern data does not need to be changed. This function is available for future extension.

Resource FIFO Enable Register (SPERSC)

Address	01FB_802C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W															
Initial	0000$_H$															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-													SPTE	SATE	LUTE
R/W	R/W													R/W	R/W	R/W
Initial	0000$_H$													0$_B$	0$_B$	0$_B$

Bit Field		Explanation
No.	Name	
0	LUTE	Resource FIFO for LUT setting 1: Enable 0: Disable
1	SATE	Resource FIFO for SAT setting 1: Enable 0: Disable
2	SPTE	Resource FIFO for SPT setting 1: Enable 0: Disable

Notes:

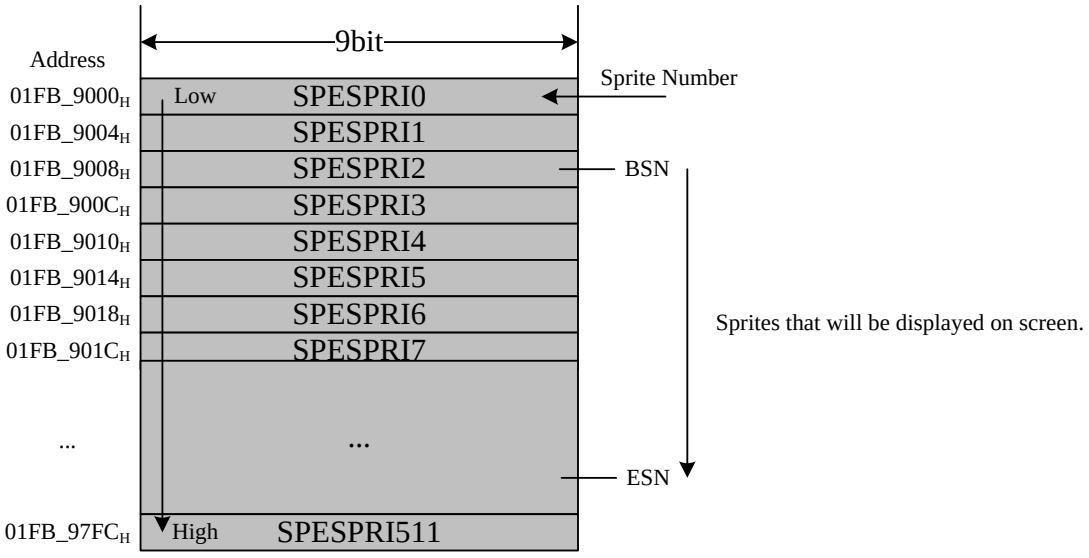
- If the corresponding register of each field is enabled, the resource FIFO can temporarily retain the SPT, SAT, and LUT setting information when the SPE draws the frame.
- Outside the SPE processing time, the corresponding register field is set via the resource FIFO.
- Within the SPE processing time, setting information for all 512 registers only can be retained in the resource FIFO.

SPT (Sprite Priority Table)

SPT allows for the programming of the display priority order of a maximum of 512 sprites. SPT is a 512 x 9 bit register table for setting 512 levels of the display priority order. This table is enabled only when the priority level mode is SPT mode. With SPT, a lower address refers to a lower priority. Write a sprite with a number that is higher than that of the other sprites, in a higher SPT address.

A series of sprite switching functions can be realized with ease through the setting of the **BSN** and **ESN** (**SPESDN**) registers.

Figure 1-24. SPT Structure



Sprite Priority Register (0 to 511), SPESPRI (0 to 511)

Address	$01FB_9000_H$ + $(n<<2)_H$, n=0 - 511															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-							SPN[8:0]								
R/W	R/W							R/W								
Initial	00_H							X								

Bit Field		**Explanation**
No.	**Name**	
8-0	SPN[8:0]	Sprite number (0 to 511)

SAT (Sprite Attribute Table)

Each sprite has an attribute register group. Therefore, 512 sprite attribute register groups constitute the sprite attribute table of the GDC macro.

Table 1-13. Basic Sprite Attribute Table

Name	Width	Attribute
DY	12	Y coordinate of display area
DX	12	X coordinate of display area
PA	25	Pattern address used by sprite (32-bit word address)
SH	9	Y size of source pattern
SW	7	X size of source pattern
VR	1	Enabling vertical reverse
HR	1	Enabling horizontal reverse
CF	3	Selecting color format
CP	10	Selecting color palette
CPF	2	Selecting color palette format
TE	1	Enabling transparency
ABM	1	Selecting alpha-blending mode
ATE	1	Enabling alpha table
ATS	1	Selecting alpha table
AF	1	Alpha data format (Only alpha table data of pattern memory)
AV	8	Alpha-blending value
DE	1	Enabling sprite display

Table 1-14. Special Sprite Attribute Table

Name	Width	Attribute
MY	8	Moving pixel in Y direction
MX	8	Moving pixel in X direction
YMV	3	Method of moving in Y direction
XMV	3	Method of moving in X direction
PAAL	1	Enabling automatic pattern address load function
BLINK	1	Enabling blink function
AMV	1	Enabling automatic movement function
REPEAT	1	Repeating action
VCNT0	8	VSYNC counter 0
VCNT1/CNT1	8	VSYNC counter 1 or CNT1
CNT0	8	Animation function execution count

SDET (Sprite Display Enable Table)

This register is the SPESDE0 to 15 (Sprite display enable) registers.

Address	$01FB_8100_H$ + $(n<<2)_H$, n=0 - 15															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	$DE[m_{31}:m_{16}]$															
R/W	R/W															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	$DE[m_{15}:m_0]$															
R/W	R/W															
Initial	0000_H															

Bit Field		**Explanation**
No.	**Name**	
31-0	$DE[m_{31}:m_0]$	$DE[m_{31}:m_0]$ has the relationships given below. SPESDE0 ($01FB_8100_H$) : DE[31:0] SPESDE1 ($01FB_8104_H$) : DE[63:32] SPESDE2 ($01FB_8108_H$) : DE[95::64] SPESDE3 ($01FB_810C_H$) : DE[127:96] SPESDE4 ($01FB_8110_H$) : DE[159:128] SPESDE5 ($01FB_8114_H$) : DE[191:160] SPESDE6 ($01FB_8118_H$) : DE[223:192] SPESDE7 ($01FB_811C_H$) : DE[255:224] SPESDE8 ($01FB_8120_H$) : DE[287:256] SPESDE9 ($01FB_8124_H$) : DE[319:288] SPESDE10 ($01FB_8128_H$) : DE[351:320] SPESDE11 ($01FB_812C_H$) : DE[383:352] SPESDE12 ($01FB_8130_H$) : DE[415:384] SPESDE13 ($01FB_8134_H$) : DE[447:416] SPESDE14 ($01FB_8138_H$) : DE[479:448] SPESDE15 ($01FB_813C_H$) : DE[511:480]

Notes:

- This register is a shadow register.
- Shadow registers are loaded at the timing of the next VSYNC after SRE=1 in the SPECTL register and SRL=1 in the SPECTL register are set.
- The SPE has a high-speed clear function. This table is configured immediately with the setting of the **SPEDETC** register.

Sprite Configuration Register 0 (SPES (0 to 511) CR0)

Address	$01FB_A000_H + (n \times 3 << 2)_H$, n=0 - 511															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CF[2:0]			AF	TE	CPF[1:0]		PA[24:16]								
				FCF[1:0]		FAF[1:0]										
R/W	R/W			R/W	R/W	R/W	R/W	R/W								
Initial	X			X	X	X	X	X								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PA[15:0]															
R/W	R/W															
Initial	X															

Bit Field		Explanation
No.	**Name**	
24-0	PA[24:0]	Pattern data address excluding the single-color sprite format (32-bit word address). If the sprite is a single-color sprite, PA[23:0] is used as 24-bit color.
26-25	FAF[1:0]	Sets the transparency rate for a font mode sprite (enabled only in font mode). 00_B: 1 (Not transparent) 01_B: 1/2 10_B: 1/4 11_B: 1/8
26-25	CPF[1:0]	Sets the color palette format. 00_B: 32 bpp (ARGB8888) 01_B: 16 bpp (ARGB4444) 10_B: 16 bpp (RGB565) 11_B: 16 bpp (ARGB1555)
28-27	FCF[1:0]	Sets the color format for a font mode sprite (enabled only in font mode). 00_B: 1 bit 01_B: 2 bits 10_B: 4 bits 11_B: 8 bits
27	TE	Enables transparency. 0: Disable 1: Enable (Do not display pixels of the defined transparent color.)
28	AF	Sets the alpha data format (enabled only for the alpha table in the pattern memory). 0: 4 bits 1: 8 bits
31-29	CF[2:0]	Sets the color format. 000_B: 1 bit 001_B: 2 bits 010_B: 4 bits 011_B: 8 bits 100_B: 16 bits (RGB565) 101_B: 16 bits (ARGB1555) 110_B: 32 bits (ARGB8888) 111_B: Single-color sprite (rectangular drawing), 24 bits (RGB888) Note: If CF=111_B and ATE=1, a font mode sprite is set.

Sprite Configuration Register 1 (SPES (0 to 511) CR1)

Address	$01FB_A004_H + (n \times 3 << 2)_H$, n=0 - 511															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CP[7:0]								AV[7:0]							
	FCV[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SW[6:0]							SH[8:0]								
R/W	R/W							R/W								
Initial	X							X								

Bit Field		Explanation
No.	Name	
8-0	SH[8:0]	Sprite height (1 to 512) pixels: (SH+1)
15-9	SW[6:0]	Sprite width (1 to 128) x 4 pixels: (SW+1)*4
31-16	FCV[15:0]	Sets a font mode sprite color value (enabled only in font mode). Only 16-bit RGB565 format is supported. FCV[15:11]: Red FCV[10: 5]: Green FCV[4:0]: Blue
23-16	AV[7:0]	Sets the alpha value for a sprite.
31-24	CP[7:0]	Selects the color palette (address for LUT indirect color mode): 7 bits to 0 bits

Sprite Configuration Register 2 (SPES (0 to 511) CR2)

Address	$01FB_A008_H + (n \times 3 << 2)_H$, n=0 - 511															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VR	CP[9:8]		-	DY[11:0]											
R/W	R/W	R/W		R/W	R/W											
Initial	X	X		X	X											
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	HR	ATE	ABM	ATS	DX[11:0]											
R/W	R/W	R/W	R/W	R/W	R/W											
Initial	X	X	X	X	X											

Bit Field		**Explanation**
No.	**Name**	
11-0	DX[11:0]	Displays the X coordinate (-2048 to 2047).
12	ATS	Selects an alpha table. 0: Memory 1: LUT (only in indirect color mode)
13	ABM	Sets the alpha-blending mode. 0: Alpha-blending with an already drawn Pre-layer 1: Alpha-blending with a single color (SPEABC)
14	ATE	Enables the alpha table. 0: Disable Alpha-blending mode of sprites 1: Enable Alpha-blending mode of pixels Note: If $CF=111_B$ and ATE=1, a font mode sprite is set.
15	HR	Horizontal reverse 0: Disable 1: Enable
27-16	DY[11:0]	Displays the Y coordinate (-2048 to 2047).
30-29	CP[9:8]	Selects a color palette (address for LUT indirect color mode): 9 bits to 8 bits
31	VR	Vertical reverse 0: Disable 1: Enable

Sprite Configuration Register 3 (SPESS (0 to 31) CR3) (for Special Sprites)

Address	$01FB_8200_H + (n \times 3 << 2)_H$, n=0 - 31															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	VCNT1[7:0] CNT1[7:0]								VCNT0[7:0]							
R/W	R/W								R/W							
Initial	00_H								00_H							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	PAAL	BLINK	AMV	REPEAT	-				CNT0[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W				R/W							
Initial	0_B	0_B	0_B	0_B	0_H				00_H							

Bit Field		**Explanation**
No.	**Name**	
7-0	CNT0	Auto animation function execution count (1 to 256) TIMES=CNT0+1
12	REPEAT	Always repeats auto animation. The CNT0 setting is not referenced. 1: Enable 0: Disable
13	AMV	Enables the automatic movement function. 1: Enable 0: Disable
14	BLINK	Enables the blink function. 1: Enable 0: Disable
15	PAAL	Enables the automatic pattern address load function. 1: Enable 0: Disable
23-16	VCNT0[7:0]	VSYNC counter 0 (1 to 256)
31-24	VCNT1[7:0]/ CNT1[7:0]	These bits define different parameters for each animation mode. VSYNC1: VSYNC Counter 1 (1 to 256) for blink mode CNT1: Definition of the movement count for animation function mode or the switching count for image switching mode (1 to 256)

Sprite Configuration Register 4 (SPESS (0 to 31) CR4) (for Special Sprites)

Address	$01FB_8204_H + (n \times 3 << 2)_H$, n=0 - 31															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	YMV[2:0]			-					MY[7:0]							
R/W	R/W			R/W					R/W							
Initial	0_H			00_H					00_H							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	XMV[2:0]			-					MX[7:0]							
R/W	R/W			R/W					R/W							
Initial	0_H			00_H					00_H							

Bit Field		**Explanation**
No.	**Name**	
7-0	MX[7:0]	Pixels to move special sprite in X direction (1 to 256 pixels)
15-13	XMV[2:0]	Movement mode in X direction 000_B: Stop in X direction 001_B: Left 010_B: Right 011_B: Reserved 100_B: Left -> Right -> Left 101_B: Left -> Return -> Left 110_B: Right -> Left -> Right 111_B: Right -> Return -> Right
23-16	MY[7:0]	Pixels to move special sprite in Y direction (1 to 256 pixels)
31-29	YMV[2:0]	Movement mode in Y direction 000_B: Stop in Y direction 001_B: Up 010_B: Down 011_B: Reserved 100_B: Up -> Down -> Up 101_B: Up -> Return -> Up 110_B: Down -> Up -> Down 111_B: Down -> Return -> Down

Note: Even in automatic movement mode with a special sprite setting, be sure that the sprites do not cross the sprite display area of both X (-2048 to 2047) and Y (-2048 to 2047).

Sprite Configuration Register 5 (SPESS (0 to 31) CR5) (for Special Sprites)

Address	$01FB_8208_H$ + $(n \times 3 << 2)_H$, n=0 - 31															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-							ADTA[24:16]								
R/W	R/W							R/W								
Initial	00_H							000_H								
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADTA[15:0]															
R/W	R/W															
Initial	0000_H															

Bit Field		Explanation
No.	Name	
24-0	ADTA[24:0]	Base address setting for pattern data address table (32-bit word address)

Special Sprite Number Specify Register (SPESSN (0 to 31)) (for Special Sprites)

Address	$01FB_8400_H$ + $(n << 2)_H$, n=0 - 31															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-							SPN[8:0]								
R/W	R/W							R/W								
Initial	00_H							000_H								

Bit Field		Explanation
No.	Name	
8-0	SPN[8:0]	Sprite number (0 to 511). Special sprite N (0 to 31) is specified.

Notes:

- The same value cannot be set in these registers.
- Set these registers before using a special sprite.

LUT (Look-up Table)

The LUT registers are used to write and read alpha data/color data from LUT. These registers are specified as SPELUTS (0 to 1023) and mapped to 000_H to FFC_H.

LUT Setting Register (0 to 1023), SPELUTS (0 to 1023)

- 32-bit color palette mode (ARGB8888)

Address	$01FB_E000_H$ + $(n<<2)_H$, n=0-1023															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AVAL[7:0]								RVAL[7:0]							
R/W	R/W								R/W							
Initial	X								X							
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	GVAL[7:0]								BVAL[7:0]							
R/W	R/W								R/W							
Initial	X								X							

Bit Field		**Explanation**
No.	**Name**	
7-0	BVAL[7:0]	Blue color definition
15-8	GVAL[7:0]	Green color definition
23-16	RVAL[7:0]	Red color definition
31-24	AVAL[7:0]	Alpha value specification

■ 16-bit color palette mode (ARGB4444)

Address	$01FB_E000_H + (n<<2)_H$, n=0-1023															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AVAL[3:0]				RVAL[3:0]				GVAL[3:0]				BVAL[3:0]			
R/W	R/W				R/W				R/W				R/W			
Initial	X				X				X				X			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AVAL[3:0]				RVAL[3:0]				GVAL[3:0]				BVAL[3:0]			
R/W	R/W				R/W				R/W				R/W			
Initial	X				X				X				X			

Bit Field		**Explanation**
No.	**Name**	
3-0	BVAL[3:0]	Blue color definition
7-4	GVAL[3:0]	Green color definition
11-8	RVAL[3:0]	Red color definition
15-12	AVAL[3:0]	Alpha value specification
19-16	BVAL[3:0]	Blue color definition
23-20	GVAL[3:0]	Green color definition
27-24	RVAL[3:0]	Red color definition
31-28	AVAL[3:0]	Alpha value specification

■ 16-bit color palette mode (RGB565)

Address	$01FB_E000_H$ + $(n<<2)_H$, n=0-1023															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	RVAL[4:0]					GVAL[5:0]						BVAL[4:0]				
R/W	R/W					R/W						R/W				
Initial	X					X						X				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RVAL[4:0]					GVAL[5:0]						BVAL[4:0]				
R/W	R/W					R/W						R/W				
Initial	X					X						X				

Bit Field		**Explanation**
No.	**Name**	
4-0	BVAL[4:0]	Blue color definition
10-5	GVAL[5:0]	Green color definition
15-11	RVAL[4:0]	Red color definition
20-16	BVAL[4:0]	Blue color definition
26-21	GVAL[5:0]	Green color definition
31-27	RVAL[4:0]	Red color definition

■ 16-bit color palette mode (ARGB1555)

Address	$01FB_E000_H$ + $(n<<2)_H$, n=0-1023															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AVAL	RVAL[4:0]					GVAL[4:0]					BVAL[4:0]				
R/W	R/W	R/W					R/W					R/W				
Initial	X	X					X					X				
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AVAL	RVAL[4:0]					GVAL[4:0]					BVAL[4:0]				
R/W	R/W	R/W					R/W					R/W				
Initial	X	X					X					X				

Bit Field		**Explanation**
No.	**Name**	
4-0	BVAL[4:0]	Blue color definition
9-5	GVAL[4:0]	Green color definition
14-10	RVAL[4:0]	Red color definition
15	AVAL	Alpha value specification
20-16	BVAL[4:0]	Blue color definition
25-21	GVAL[4:0]	Green color definition
30-26	RVAL[4:0]	Red color definition
31	AVAL	Alpha value specification

1.9.5 Explanation of Operation

1.9.5.1 Processing Overview

The SPE is the core image processing unit of the GDC macro. The SPE functions include shift, reverse, and alpha-blending. All of these functions can be controlled with the corresponding register settings by software.

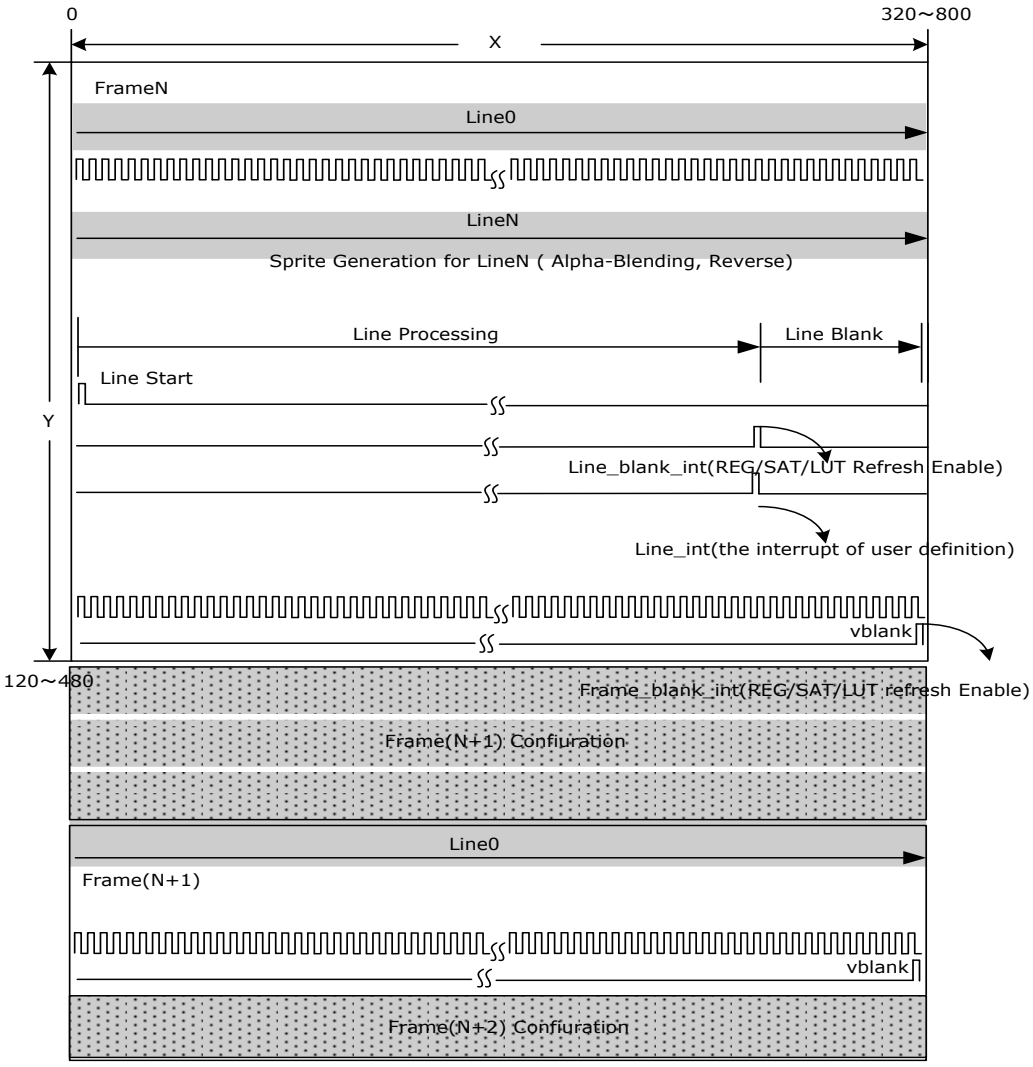
Sprite Generation Flow

The SPE is equipped with scan line rendering hardware.

The SPE analyzes SAT (Sprite Attribute Table) and other registers at each scan line and detects the pixel position of the highest-priority sprite. Moreover, it reads the pattern data including the alpha value and executes image processing to write the results of the final color data to LineBuffer.

Frame Generation Flow

Figure 1-25. Overview of Frame Processing

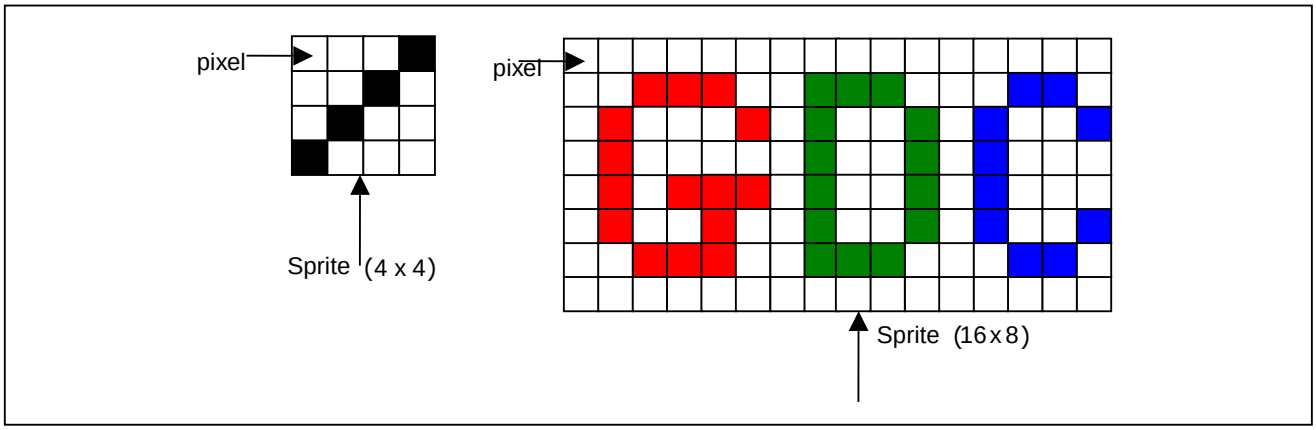


SAT (Sprite Attribute Table) and other registers are refreshed during the vertical blanking period or the specified line blanking period. However, some shadow registers can always be set.

1.9.5.2 Sprites

A sprite displays the source image on the screen. Sprites can display arbitrary patterns stored in the pattern memory. Processing is executed according to each sprite attribute such as reverse. The sprite position is defined from the top left corner. A sprite can be moved per pixel according to the SAT (Sprite Attribute Table) setting.

Figure 1-26. Sprite Definition



In the GDC macro, sprite sizes are from 4 x 1 pixels to 512 x 512 pixels. The size can be specified in the X and Y directions separately. (X is in steps of 4 pixels, and Y is in steps of 1 pixel.)
Figure 1-26 shows sprites of 4 x 4 pixels and 16 x 8 pixels.

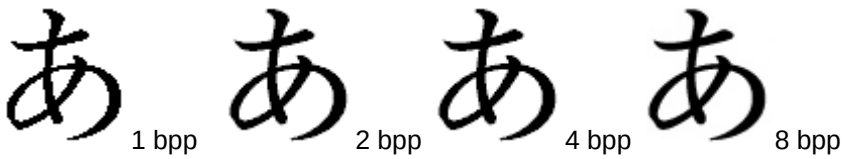
1.9.5.3 Single-color Sprite (Rectangular Drawing)

The sprite engine can easily generate single-color rectangular graphics with a size from 4 x 1 to 512 x 512 pixels. The color format is 24 bpp (RGB888). Pattern data is not required.

1.9.5.4 Font Mode Sprite

The sprite engine has a font mode sprite function for depicting fonts with a smaller amount of pattern data. A font mode sprite performs alpha-blending with lower layers with the alpha value of a different pixel of the same color to generate a new color value. It is called a font mode sprite since the function is especially suitable for font realization. Unlike ordinary sprites, font mode sprites do not have color values but alpha factors only for pattern data. Therefore, there is no need to refer to a color palette.

The font mode sprite "color (alpha)" format supports 1 bpp, 2 bpp, 4 bpp, and 8 bpp.



The above sizes indicate the effects of each format of a font mode sprite with a size of 64 x 70.

1.9.5.5 Sprite Numbers

The serial number of each sprite is fixed, ranging from 0 to 511, with the respective names of SPR0, SPR1,..., SPR511.

1.9.5.6 Priority

The sprite layer has a maximum of 512 sprites. They may overlap since the positions of sprites can be defined anywhere at the user's discretion.

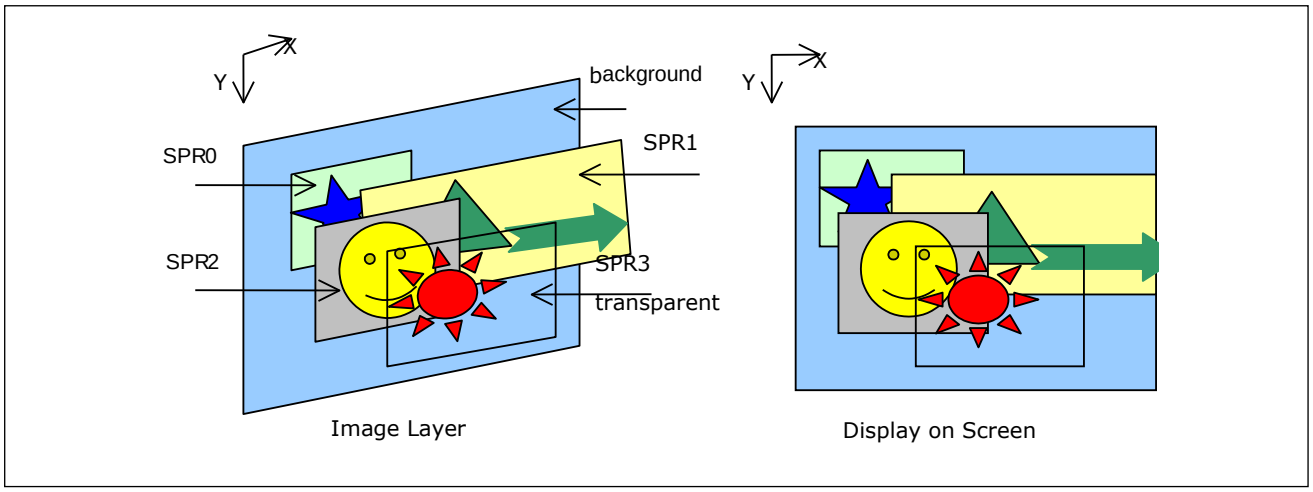
Sprites have 2 modes, fixed mode and SPT (Sprite Priority Table) mode, to determine the priority order of 512 sprites. The mode to be used depends on the PRI register (SPECTL bit1) setting.

Fixed Mode

In fixed mode, the sprite priority order is determined by the serial number (e.g., 0 to 511). The priority order of each sprite is fixed in the initial settings. The larger the serial number, the higher the priority. The higher the priority, the closer the sprite is positioned to the viewer. Each sprite layer has a higher priority than the background layer.

Background < SPR0 < SPR1 < ... < SPR31 < ... < SPR511

Figure 1-27. Sprite Priority (Fixed Mode)



shows 4 sprites displayed on the screen at the same time. At places where they overlap, the degree of visibility is determined by their priority order. High-priority pixels are visible except in cases where the pixel color is equal to the transparent color (when the transparency function is enabled).

SPT Mode

In SPT mode, the priority order of 512 sprites is determined based on the SPT setting, not on the sprite numbers. SPT is a 512 x 9 bit register table for storing sprite numbers. The SPE determines the priority order according to the sprite addresses. The lower the address, the lower the priority.

When using this mode, initialize the SPT before executing SPE processing.

Figure 1-28. Sprite Priority (SPT Mode)

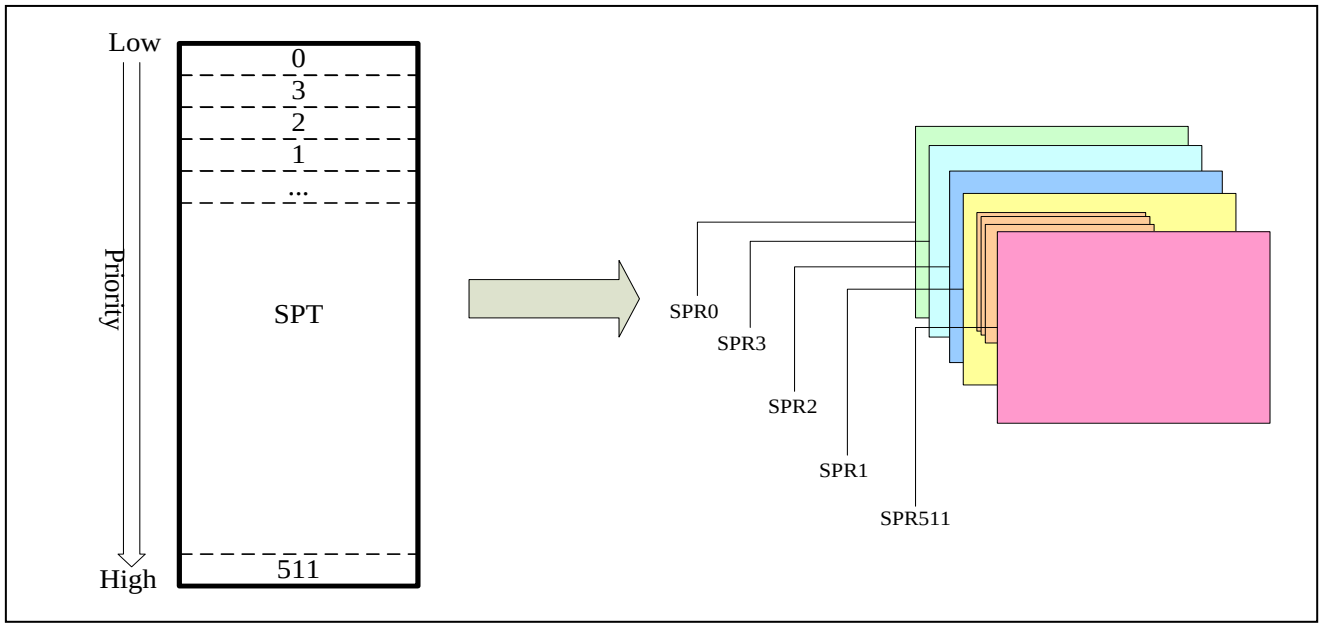


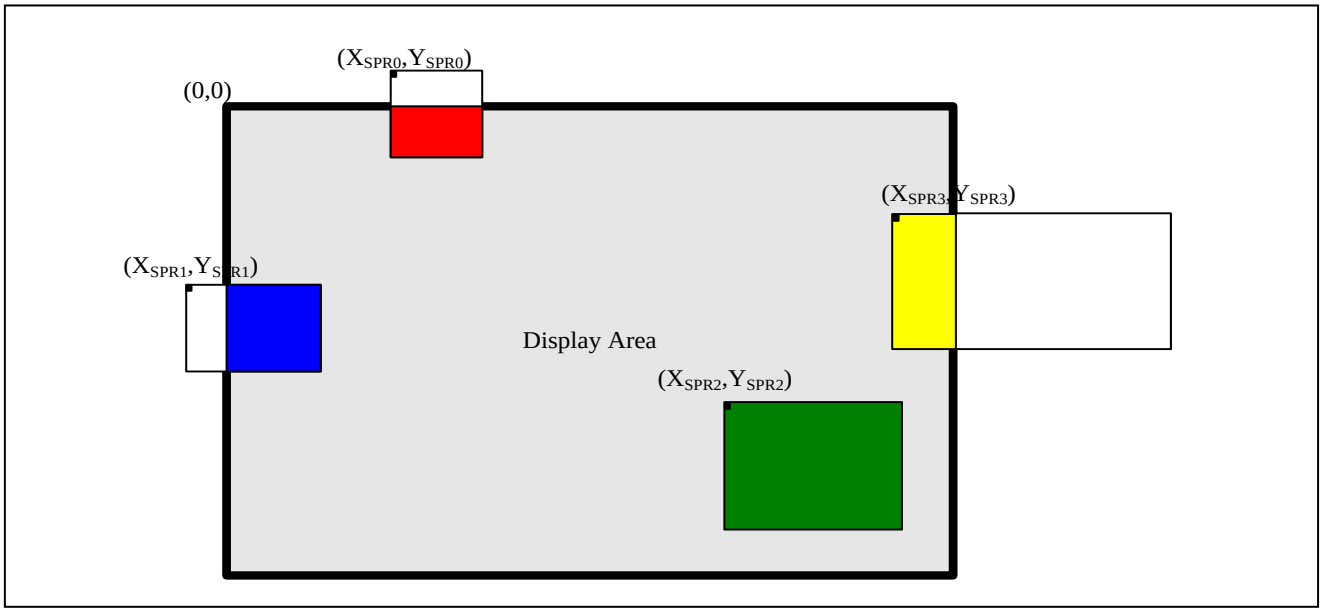
Figure 1-28 shows the sprite priority order in SPT mode. The numbers written in SPT are from lower addresses to higher addresses, "0, 3, 2, 1,..., 511." The sprite display order on the screen is from back to front, "SPR0, SPR3, SPR2, SPR1,..., SPR511."

1.9.5.7 Sprite Display Area

All sprites can be displayed in the display area. Sprite coordinates are related to the coordinates of the display area. The part of a sprite beyond the display area cannot be displayed on the screen.

It is possible to define 320 to 800 pixels in the X direction and 120 to 600 pixels in the Y direction in the display area. Therefore, the SPE can accept up to the WVGA format.

Figure 1-29. Sprite Display Area



The display coordinates of **SPRn** can be derived from DX and DY of the *SAT* register *SPESnCR2* (n=0 to 511).

DX and **DY** are 12-bit signed registers where a value from -2048 to +2047 can be defined.

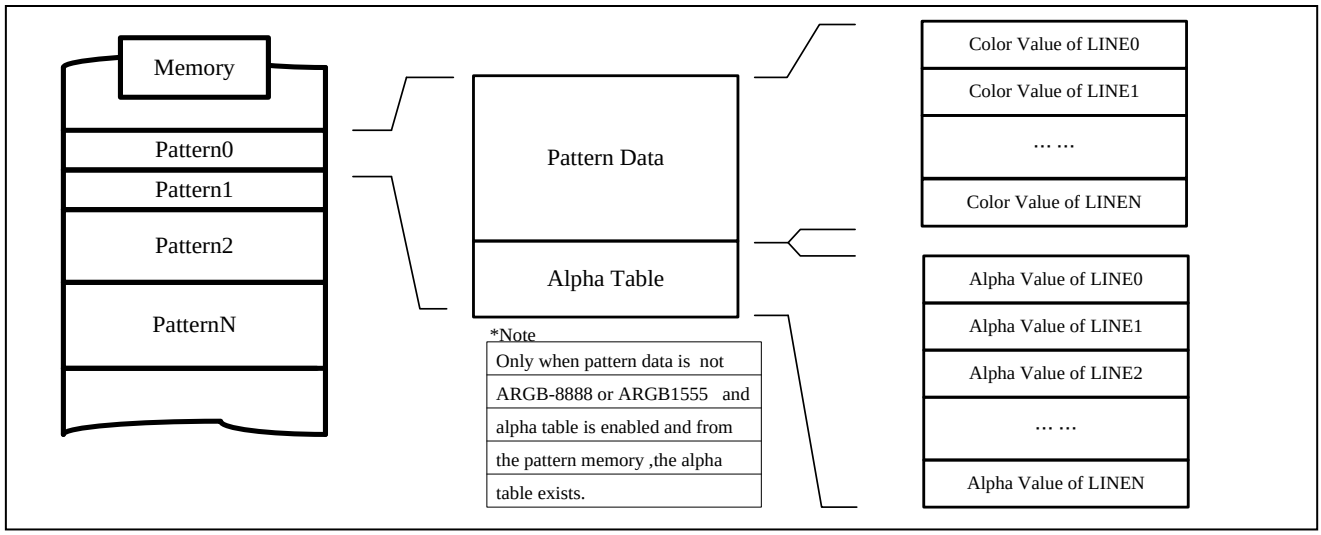
1.9.5.8 Pattern Data Format

Pattern data is the source image data of sprites. It is stored in memory in a different color data format and is located on a per-line basis to enhance read efficiency with LineBuffer.

Furthermore, when *ATE* (SAT) of the alpha table is enabled, the alpha table exists in *ATS* (SAT) in the pattern memory unless the color format of the pattern is ARGB8888 or ARGB1555. It is located under the pattern data as the alpha table containing the alpha data of all the pattern pixels. As for the pattern data, the alpha data is located in memory on a per-line basis.

The start addresses of the pattern data and alpha data table are 32-bit addresses. By setting SPEPBA, the base address of the pattern memory area can be set optionally. Generally, it can be used as the default setting (initial value) (which is 0). When changing values, specify the base address of the pattern memory area before access processing is executed.

Figure 1-30. Pattern Memory Structure



The line data structure containing color and alpha data is in "Little Endian." That is, a smaller pixel of the X coordinate in the same line has the smaller address and bit number in memory. Moreover, the color data or alpha data of all pixels must be continuous without any gaps. A font mode sprite consists of only the alpha table and does not require pattern data.

Figure 1-31. Pattern Data Format

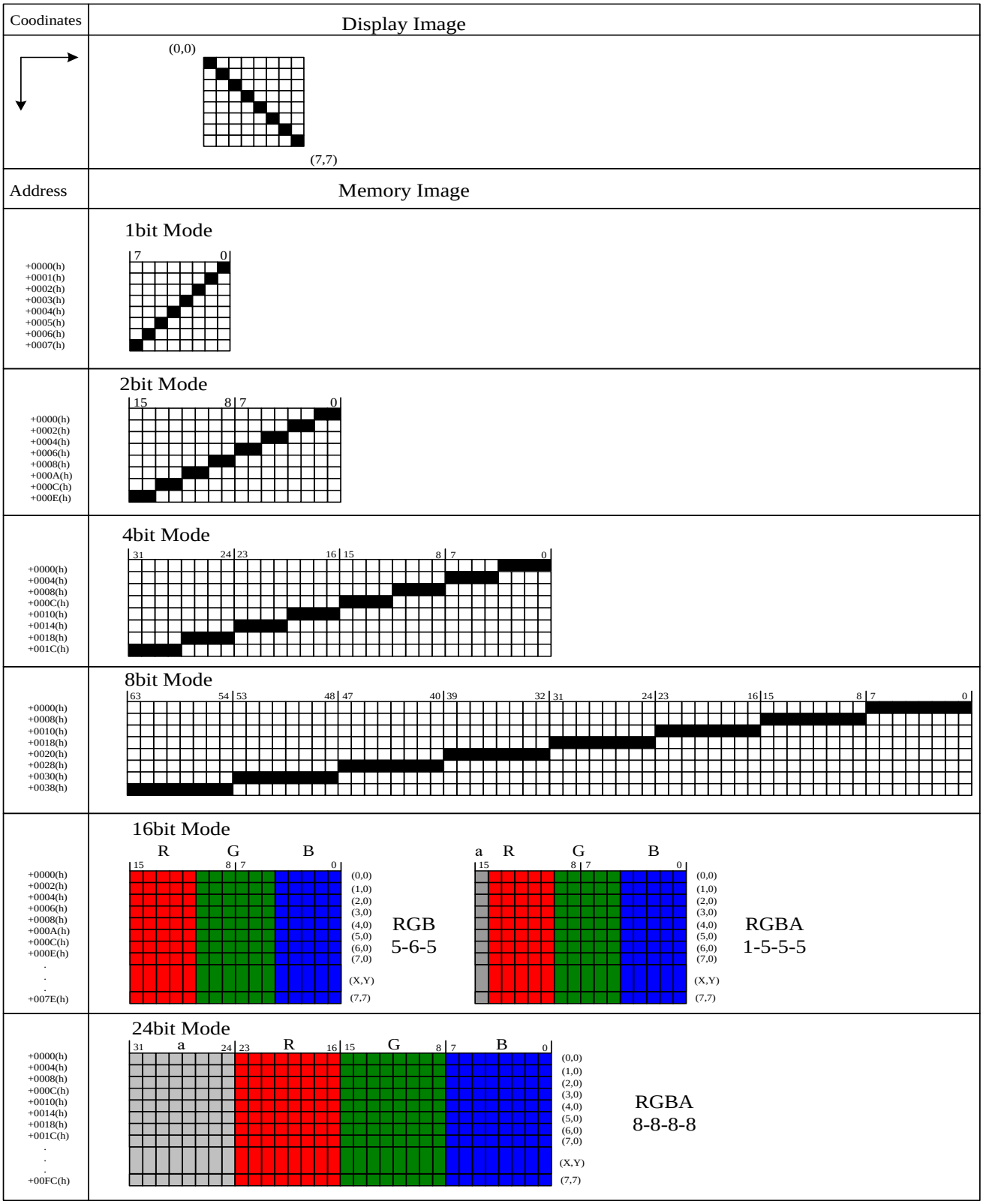
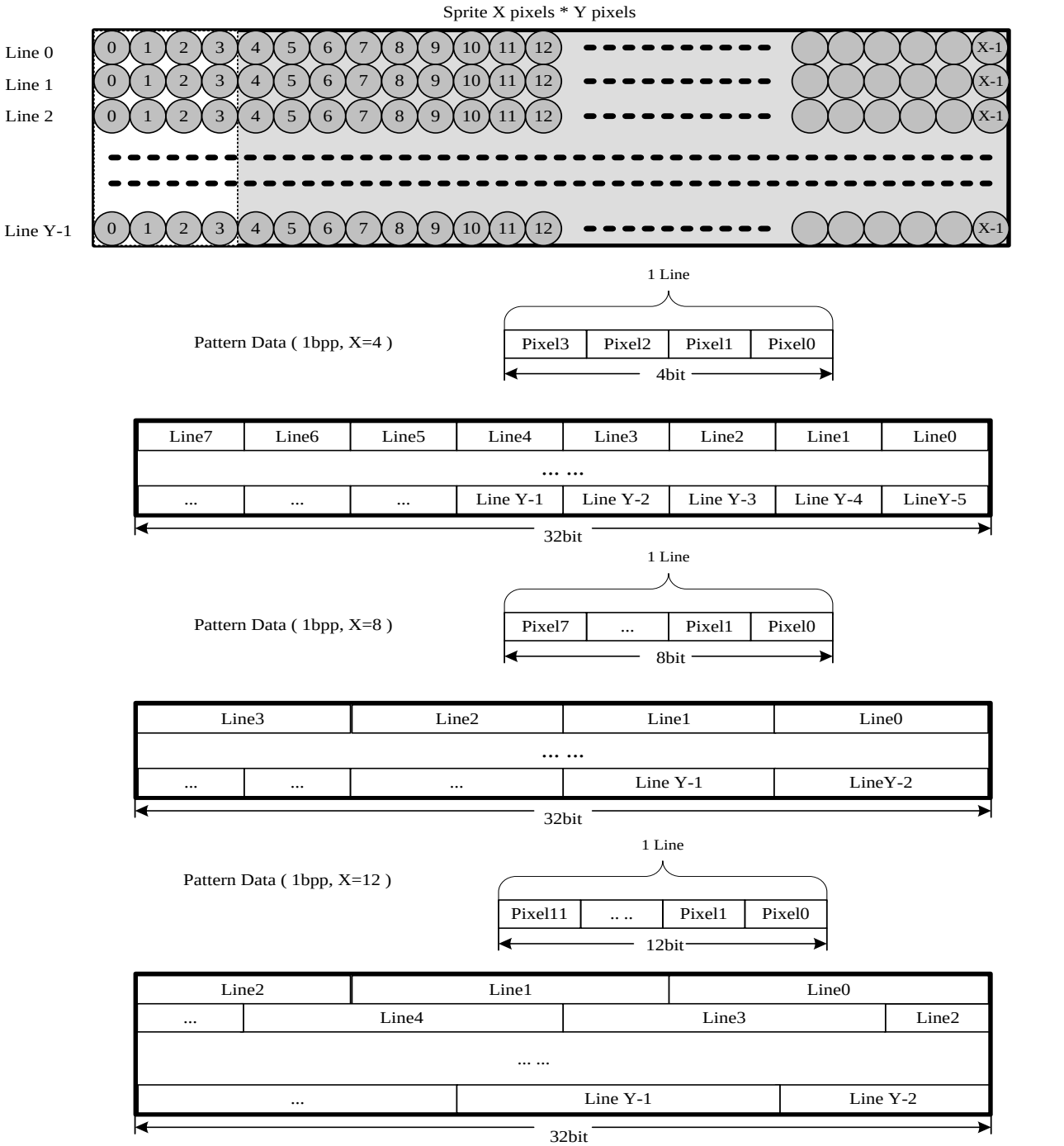


Figure 1-31 shows sprite pattern data of 8 x 8 pixels in memory with a different color data format. The alpha table has 2 modes, 4 bpp and 8 bpp, which are determined by a register corresponding to SAT.

The alpha table structure is the same as the color value of the 4 bpp and 8 bpp color formats. The data structure of the font mode color format is the same as that of the 1 bpp, 2 bpp, 4 bpp, and 8 bpp of ordinary sprites.

Data of 1 line is arranged consecutively between lines without any gaps even when the data is less than 1 word (4 bytes). For example, a 1 bpp sprite has the following configuration.



1.9.5.9 Transparent Color Function

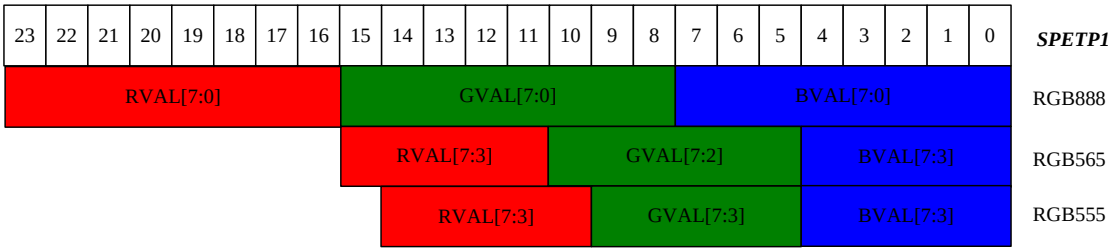
Transparent color can be set in the color palette table. The pixels of a transparent color sprite are not displayed on the screen.

Direct color mode has different setting registers from indirect color mode (*SPETP0* register and *SPETP1* register, respectively).

In indirect color mode, the *SPETP0* register defines the transparent color of 4 color formats (1 bpp, 2 bpp, 4 bpp, and 8 bpp).

The GDC macro supports 3 direct color formats of RGB (565), ARGB (1555), and ARGB (8888). As shown in Figure 1-32, the GDC macro determines the transparent color of each color format.

Figure 1-32. Transparent Color Format of Direct Color Mode



1.9.5.10 Background Color Function

The background mode is set by the BGM (SPECTL) register.

In ordinary background mode (**BGM = "0"**), the background color is displayed on the screen at the pixels with no sprite or with a transparent color. The background color is in 24 bpp of a single color (RGB888) and can be set by using the corresponding SPEBG register.

Moreover, the SPE supports other background modes so that the Display Controller can optimize overlaps with other layers. If the SPE is in the mode where BGM is "1", all transparent pixels are treated as fixed-color data ("000001_H"), and real-color data ("000001_H") is converted to "000000_H".

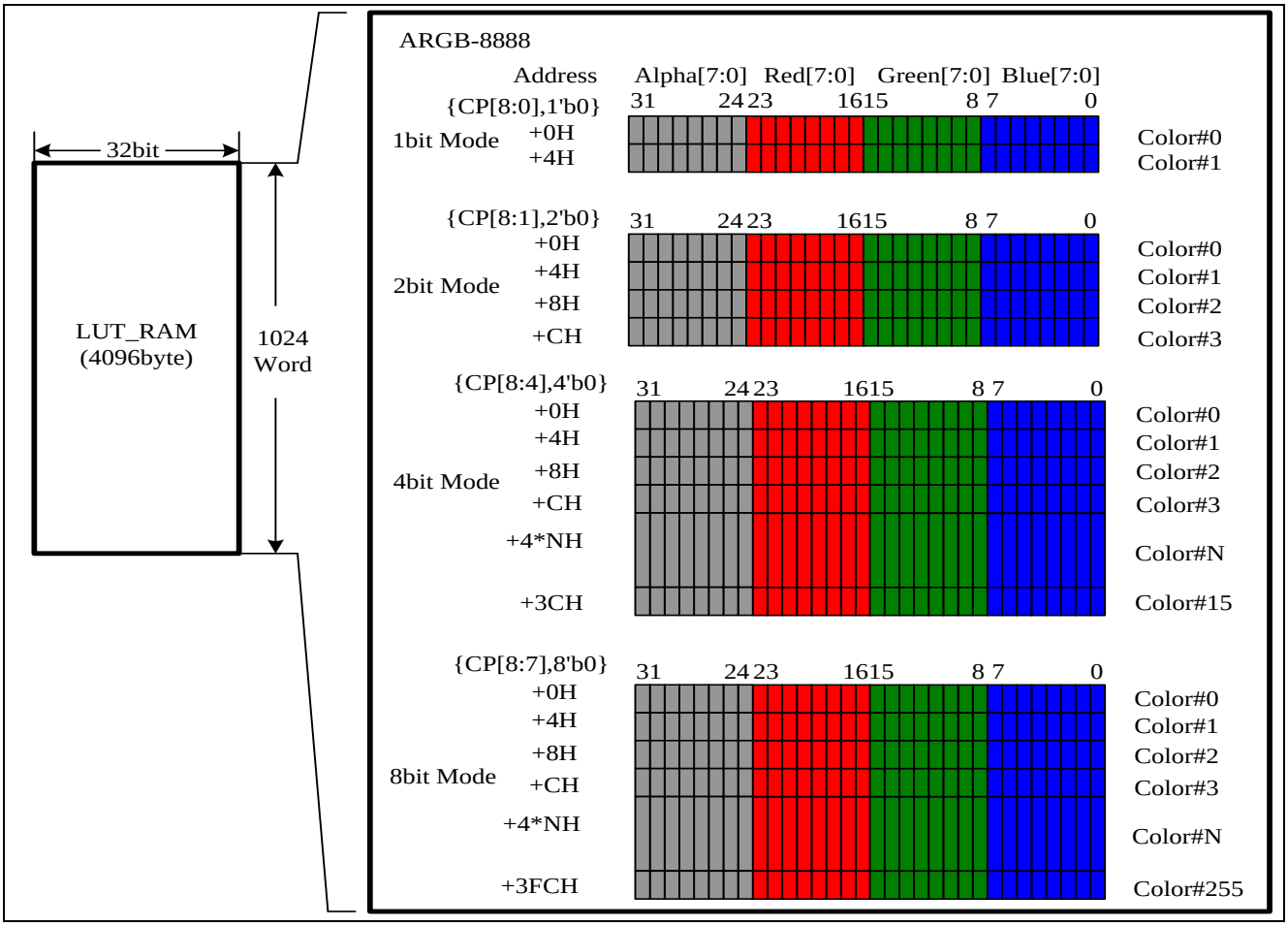
1.9.5.11 Color Palette Tables

The **color palette table (LUT_RAM)** for the SPE has a 4096-byte RAM (1024w x 32 bits).

RAM per 32-bit word shows 1 color in ARGB8888 format or 2 colors in 16-bit color in ARGB4444/ARGB1555/RGB565. In ARGB8888, 1-bit mode consists of 512 palettes, 2-bit mode 256 palettes, 4-bit mode 64 palettes, and 8-bit mode 4 palettes. In ARGB4444/ARGB1555/RGB565, 1-bit mode consists of twice the number of palettes as those in ARGB8888. That is, 1-bit mode consists of 1024 palettes, 2-bit mode 512 palettes, 4-bit mode 128 palettes, and 8-bit mode 8 palettes.

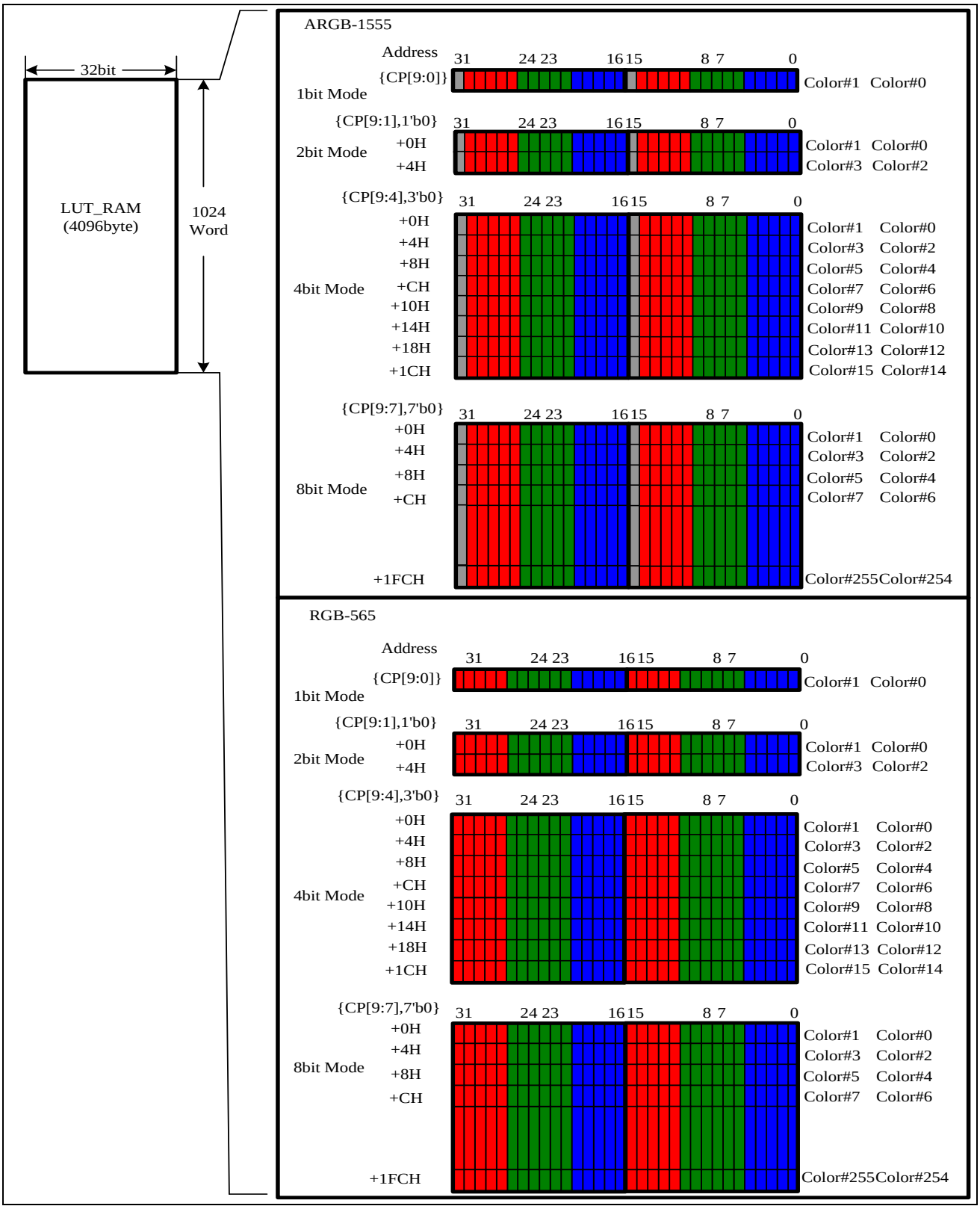
The color palette table values can be set directly from CMDSEQ or an external CPU, or can be transferred from external memory by DMAC.

Figure 1-33. Color Palette Table (ARGB8888)



The RAM address of the color palette used by sprites is determined by the CP[7:0] field of *SPECnCR1* (n=0 to 511) in the SAT register. The color palette type is determined by the CPF[1:0] field of *SPECnCR0* (n=0 to 511) in the SAT register.

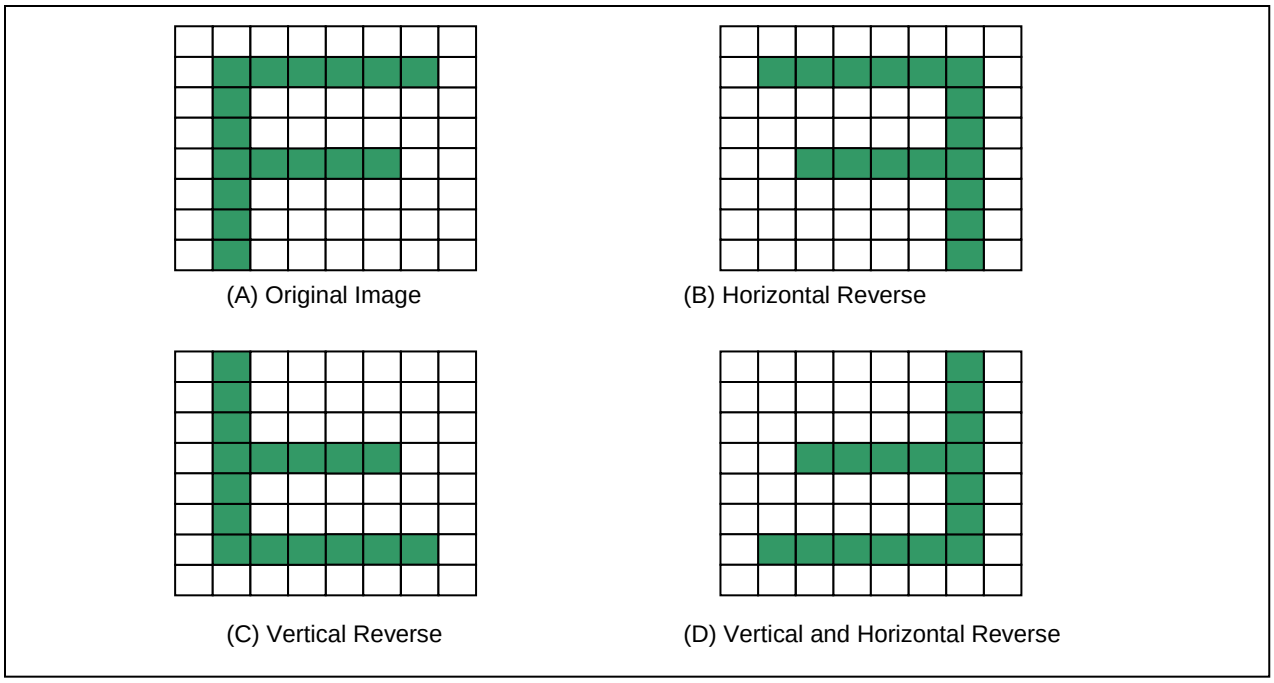
Figure 1-34. Color Palette Table (ARGB1555/RGB565)



1.9.5.12 Reverse Function

The SPE supports the reverse function in both the vertical and horizontal directions. Whether to reverse a sprite when displaying it on the screen is determined by the corresponding register of the sprite attribute table (**HR, VR**). The reverse function can be enabled for vertical and horizontal directions at the same time.

Figure 1-35. Reverse Function



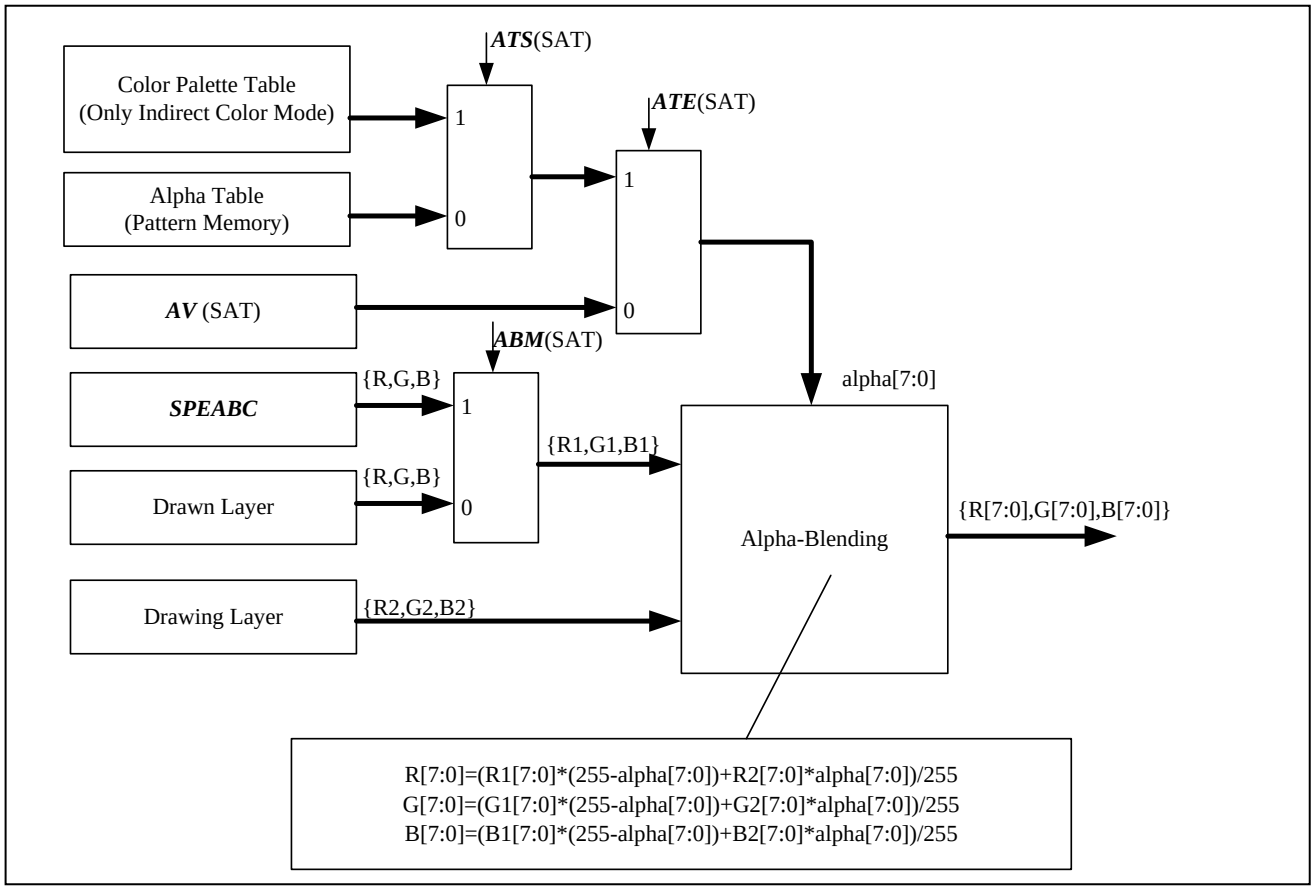
1.9.5.13 Alpha-blending

The SPE (sprite engine) supports the alpha-blending function.

If the corresponding register is enabled by the alpha-blending mode setting, alpha-blending is performed using already drawn image data of a sprite or a single color (RGB888) to write the image data of the sprite drawing in LineBuffer.

The alpha value data can be obtained from the alpha table under the pattern data of the pattern memory. The alpha data can be obtained from the alpha channel of LUT_RAM in pixel alpha-blending mode or from the sprite attribute table in sprite alpha-blending mode.

Figure 1-36. Alpha-blending Function



According to the calculation formula, if alpha[7:0] is "00_H", only the lower layer (single color or drawing layer) remains as the blending result. That is, the drawing layer becomes transparent. If the alpha value is "ff_H", the drawing layer is not transparent.

1.9.5.14 Special Sprites

The GDC macro has 32 special sprites. All 512 sprites can be defined as special sprites with the SPESSN0 to SPESSN31 register settings. These sprites can be used to automatically realize many 2D animation functions such as blink, automatic movement, image switching, etc. All of these functions are controlled by the *SPESS (0 to 31) CR3*, *SPESS (0 to 31) CR4*, and *SPESS (0 to 31) CR5* registers.

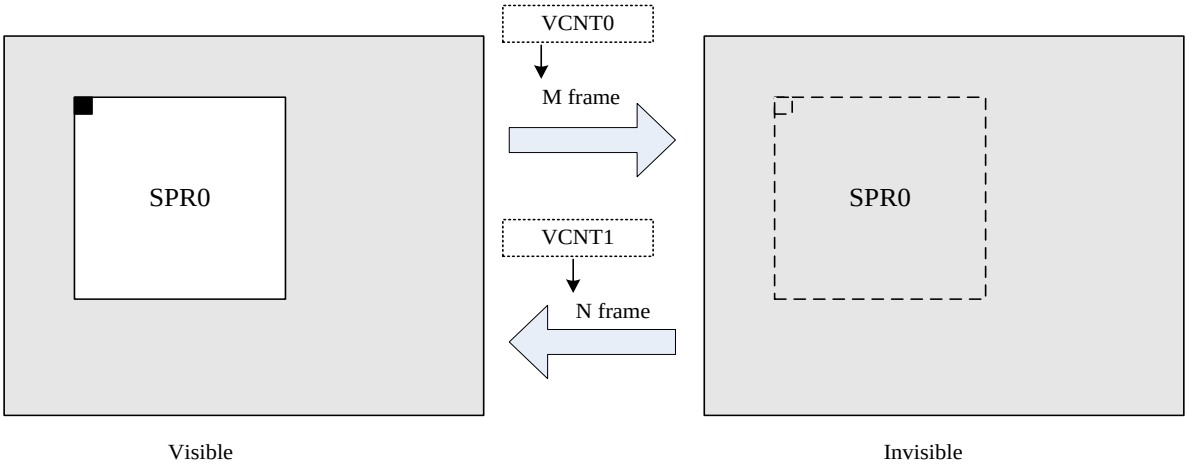
The blink function cannot be used concurrently with the movement function and the image switching function for special sprites. However, the automatic movement and image switching functions can be executed concurrently.

Blink Function

The blink function of special sprites is executed with control of the sprite display and non-display times. The display period is set in *VCNT0 (SPESSnCR3)*, and the non-display period in *VCNT1 (SPESSnCR3)*. Use the **BLINK (SPESSnCR3)** bit to enable/disable this function. Moreover, set the execution frequency of the blink function with *CNT0 (SPESSnCR3)*.

Before using the blink function of special sprites, set *SDET* to set the initial status (displayed/not displayed).

Figure 1-37. Blink Function



Automatic Movement Function

Special sprites support many movement methods. Setting the corresponding SPESS (0 to 31) CR4 registers enables complicated movements in addition to simple movements such as vertical and horizontal movements.

Set the time interval of each movement with the VCNT0 (SPESSnCR3) register.
Automatic movement mode and a single movement for all pixels can be set separately in the X and Y directions.

Pixel movement in the X direction is set with the MX (SPESSnCR4) register, and pixel movement in the Y direction is set with the MY (SPESSnCR4) register. A value from 1 to 256 pixels is allowed in both directions.

Automatic movement mode in the X direction is set with the XMV (SPESSnCR4) register, and automatic movement mode in the Y direction is set with the YMV (SPESSnCR4) register.
In automatic movement mode, the CNT1 (SPESSnCR3) register setting shows the number of times that a sprite performs a series of movements. However, the actual number of times required for the sprite to perform the series of movements differs according to the automatic movement mode setting.

Figure 1-38. Basic Movement in the X Direction

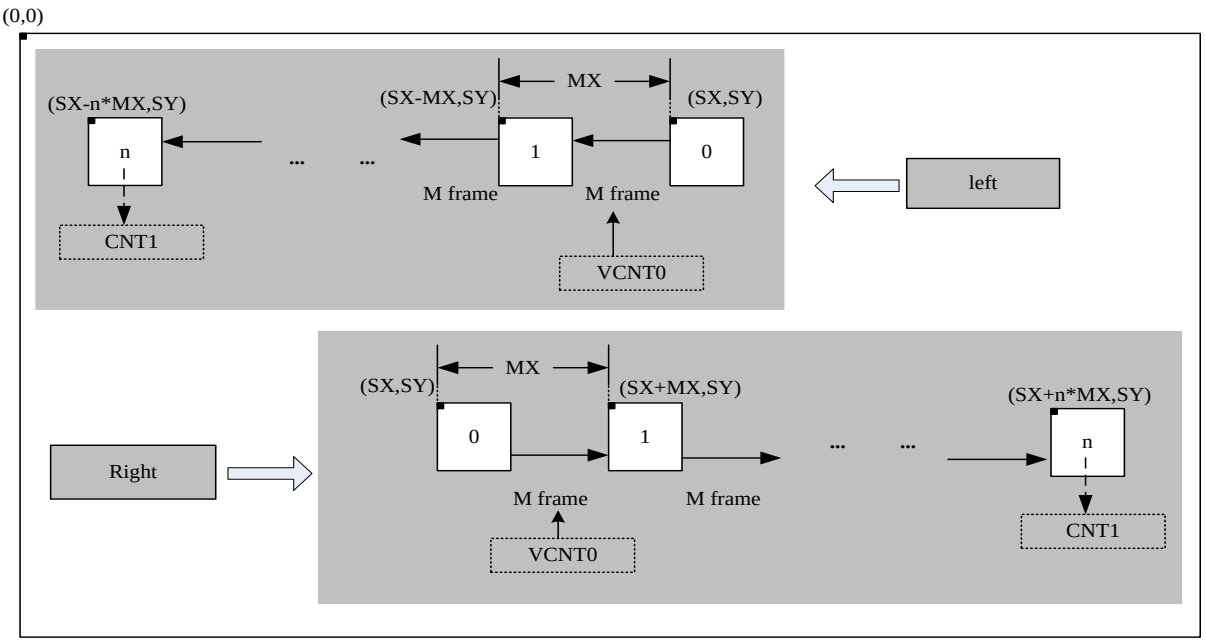


Figure 1-39. Basic Movement in the Y Direction

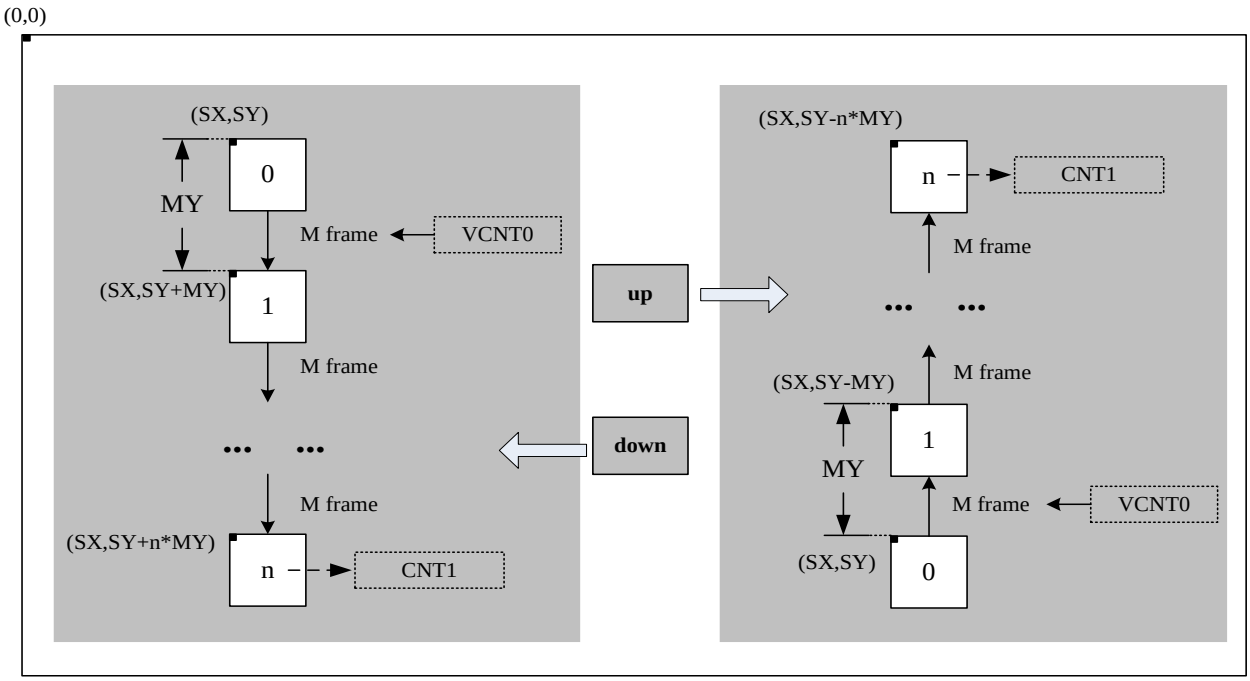


Figure 1-38 and Figure 1-39 show the basic movement method of a special sprite in the X and Y directions. In this mode, the sprite moves n (n=CNT1+1) times (automatic movement unit) in a specific direction.

Figure 1-40. Specific Movement in the X Direction

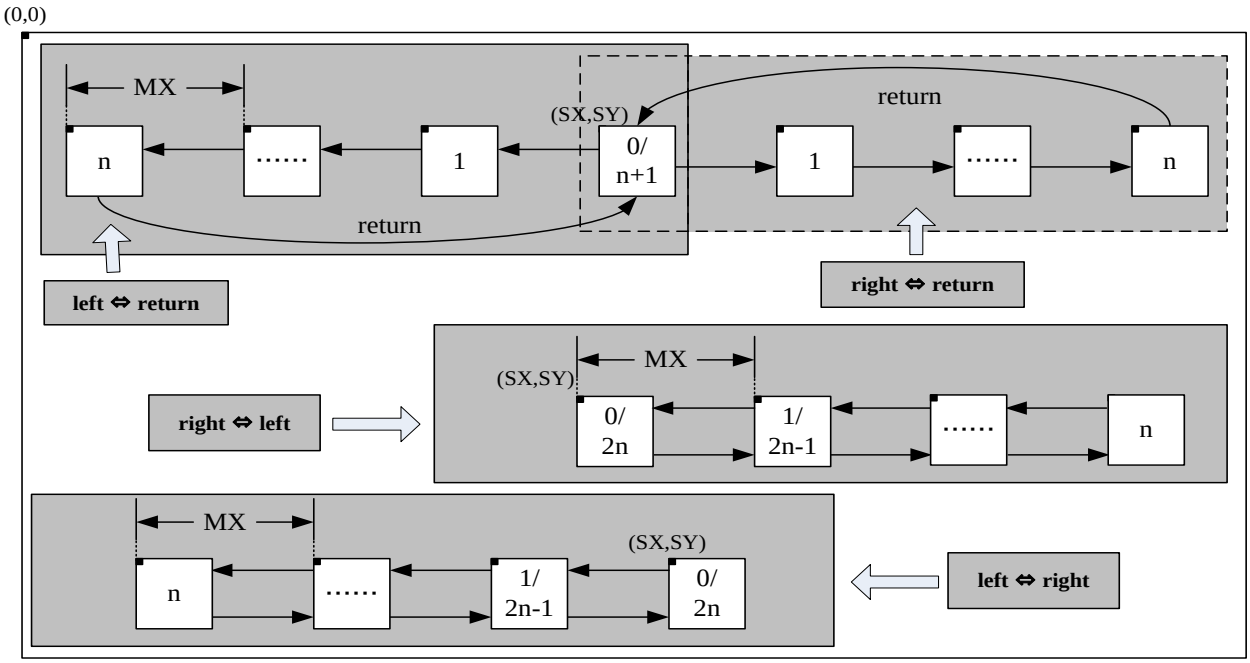


Figure 1-41. Specific Movement in the Y Direction

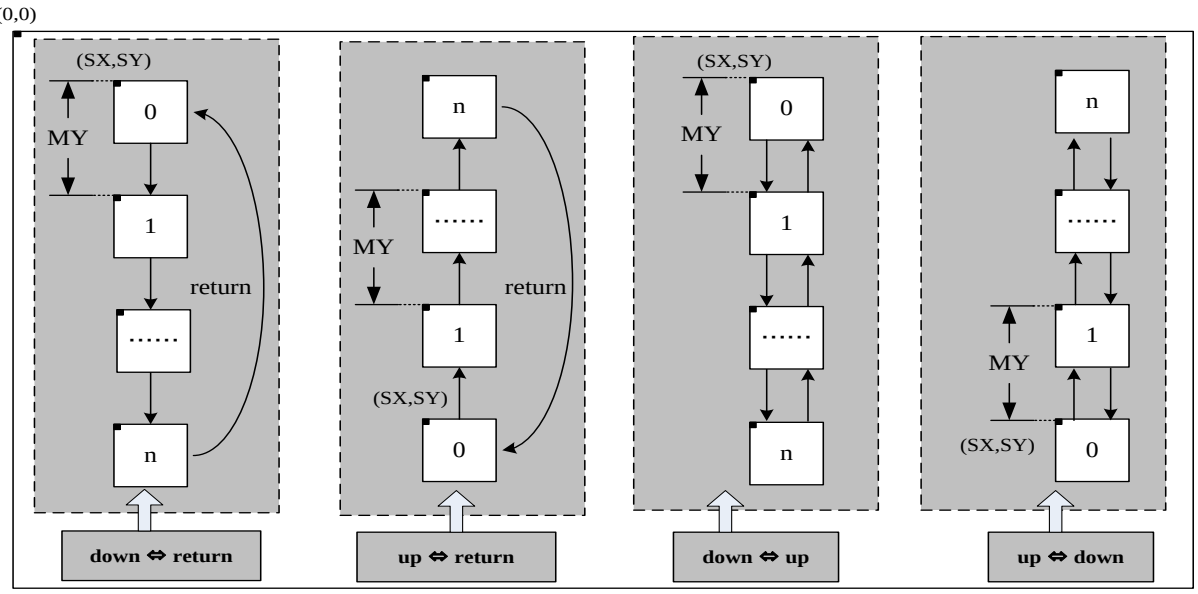


Figure 1-40 and Figure 1-41 show 2 specific movement methods of a special sprite in the X and Y directions. Both methods enable a sprite to move n times (n = ***CNT1***+1) and return to the initial position. The difference between the 2 methods is that the number of times of movement in which sprite returns to the initial position is 1 time or n times, respectively. These movements are therefore called STR (Single Time Return) and NTR (N Times Return). Moreover, the automatic movement unit of the 2 methods is also different, (n+1) times and 2n times, respectively.

Since the X and Y directions can be set separately in an automatic movement method, the automatic movement method can be realized by combining the two of them.

Several examples are shown below. The automatic movement unit differs depending on the combination.

Figure 1-42. Combined Movement in the X and Y Directions (1)

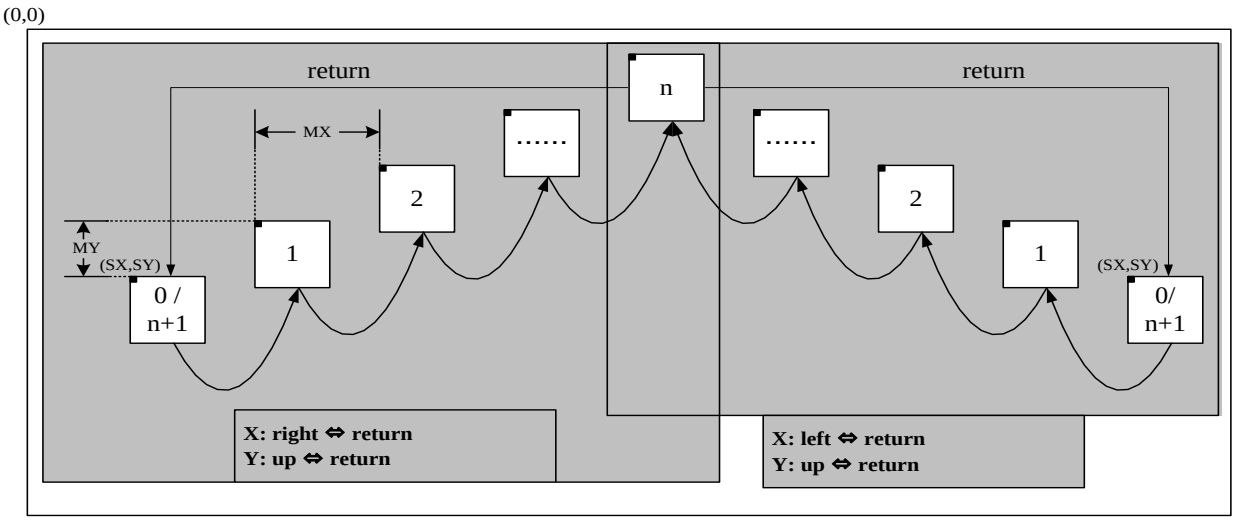


Figure 1-42 shows the movement method with STR in the Y direction, STR in the X direction, and n+1 times for the automatic movement unit.

Figure 1-43. Combined Movement in the X and Y Directions (2)

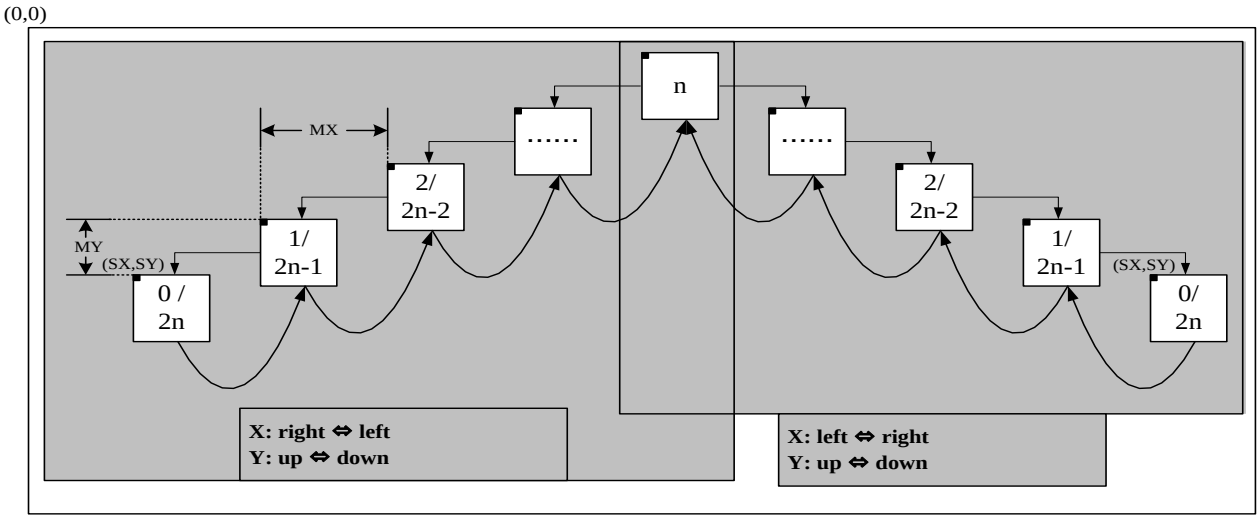


Figure 1-43 shows the movement method with NTR in the Y direction, NTR in the X direction, and 2n times for the automatic movement unit.

Figure 1-44. Combined Movement in the X and Y Directions (3)

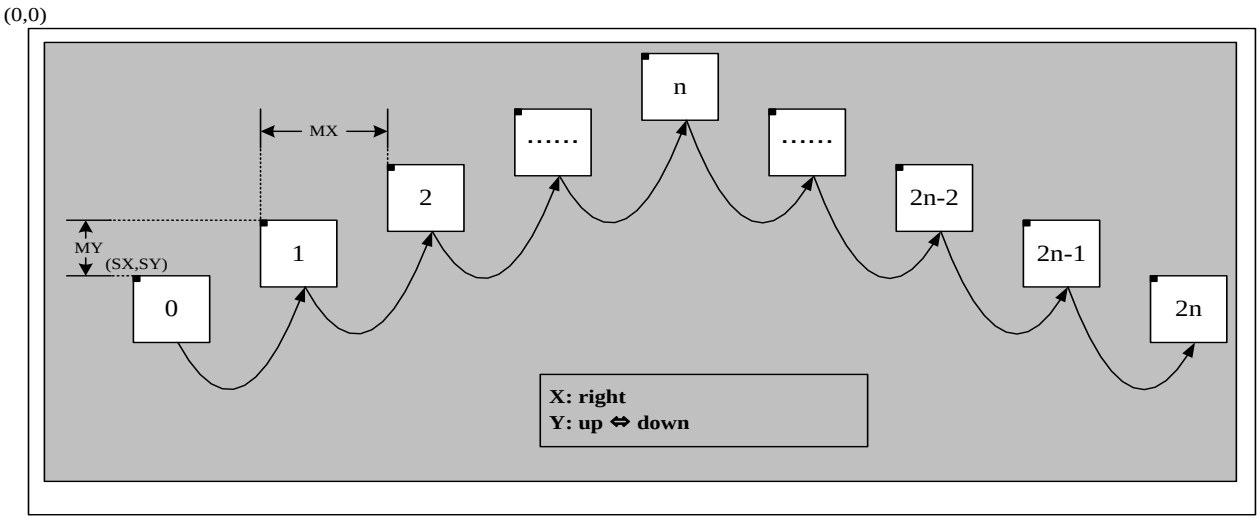


Figure 1-44 shows the movement method with unidirectional movement in the Y direction, unidirectional movement in the X direction, and 2n times for the automatic movement unit.

Figure 1-45. Combined Movement in the X and Y Directions (4)

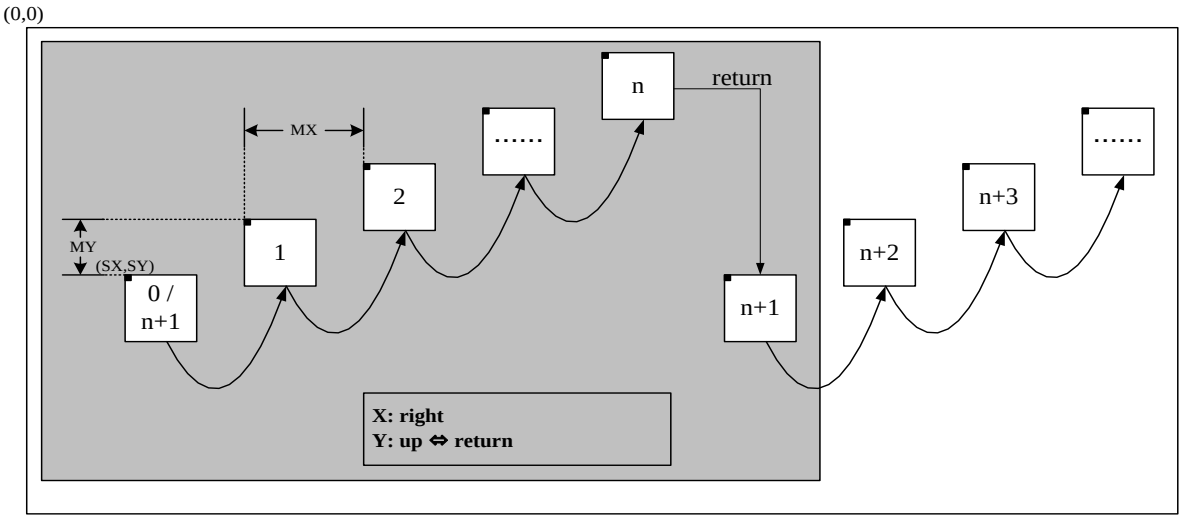


Figure 1-45 shows the movement method with unidirectional movement in the X direction, STR in the Y direction, and n+1 times for the automatic movement unit.

Figure 1-46. Combined Movement in the X and Y Directions (5)

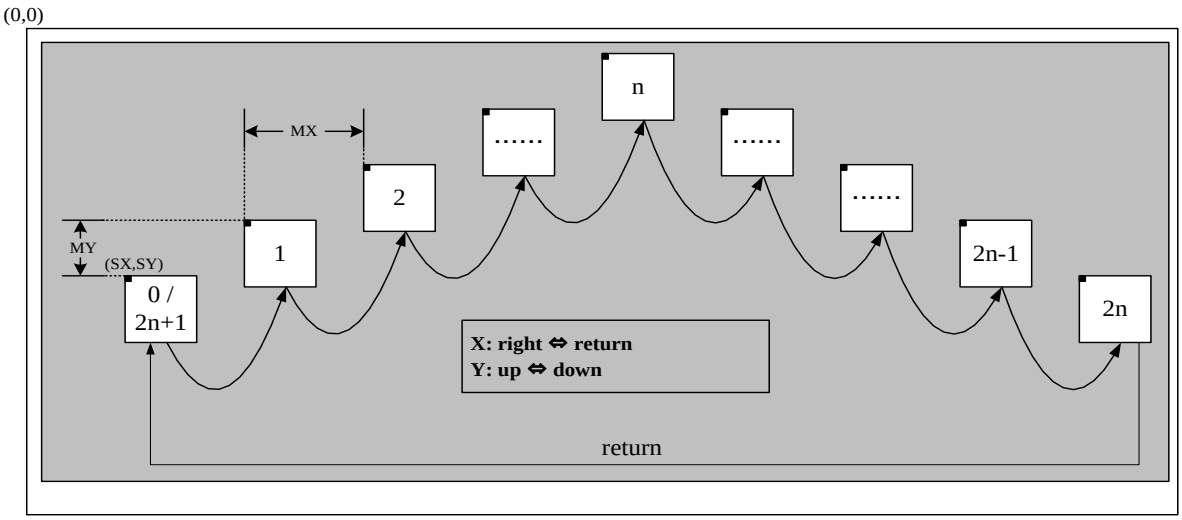


Figure 1-46 shows the movement method with STR in the X direction and NTR in the Y direction (special combined movement method). The automatic movement unit is **2n+1** times. Note that the same Y coordinate is used for **2n+1** and **2n** of the sprite position.

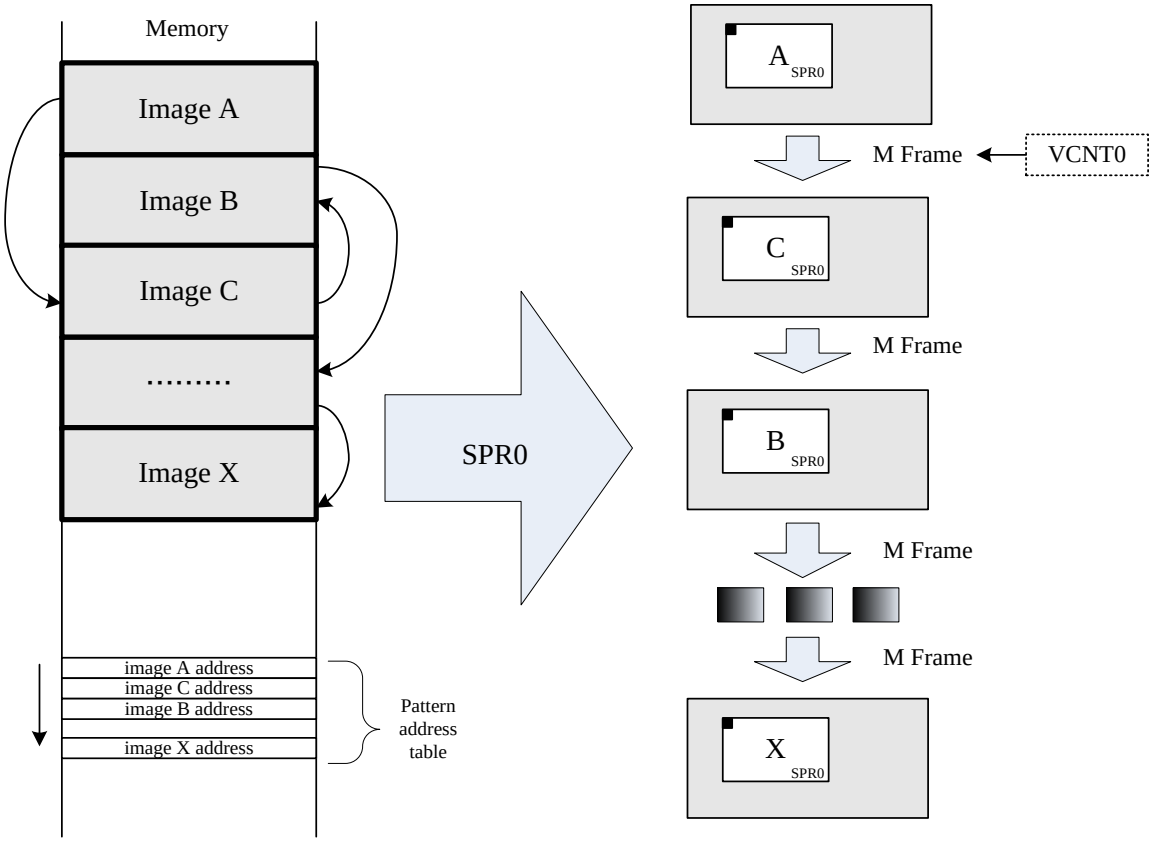
Image Switching Function

When **PAAL** is set to "1", a special sprite can automatically load the following pattern data address from the address table in the pattern memory. Set the initial address of the address table used by special sprites with the ADTA (SPESSnCR5) register. Set the time interval for image switching with the VCNT0 (SPESSnCR3) register. Also, set a series of image numbers used by special sprites with the CNT1 (SPESSnCR3) register. Set the function execution count with the CNT0 (SPESSnCR3) register. When 1 set of the image switching function execution ends, the image in the initial address of the address table is loaded.

Since the initial image of a sprite is automatically loaded by a special sprite before processing frame 0, it is not necessary to set PA (SPESnCR0) for the sprite.

When using the image switching function with the automatic movement function for special sprites, the series of image numbers equals the automatic movement unit. That is, it depends on the setting of the automatic movement method.

Figure 1-47. Image Switching Function



1.9.6 Interrupts

The following interrupt events are generated in the SPE. They have status flags corresponding to the INTST register of MCNT, which can manage synchronization with each event.

Event		MCNT/INTST support
ILN	Line interrupt: The processing of all sprites specified in SPEILN has completed.	INT19 (bit19)
LBK	Line blank interrupt: The processing of all sprites in each line has completed.	INT18 (bit18)
PER	Processing error interrupt: A processing error occurred.	INT17 (bit17)
BER	Bus error interrupt: A response is made to a bus error during reading of pattern data with an internal bus.	INT16 (bit16)
EBC	"Enable bits change" interrupt: ShadowRegister has completed an automatic update.	INT15 (bit15)

1.9.7 Notes

1.9.7.1 Restrictions

The blink function cannot be used concurrently with the movement function and image switching function for special sprites.

To not output a display signal when using the sprite display function, update Display Enable during a vertical synchronous period. (See "1.6.5.2. Display Timing")

1.10 Graphics Memory (VRAM)

This section describes the GDC macro graphics memory.

1.10.1 Overview

The graphics memory (VRAM) is the following built-in RAM:

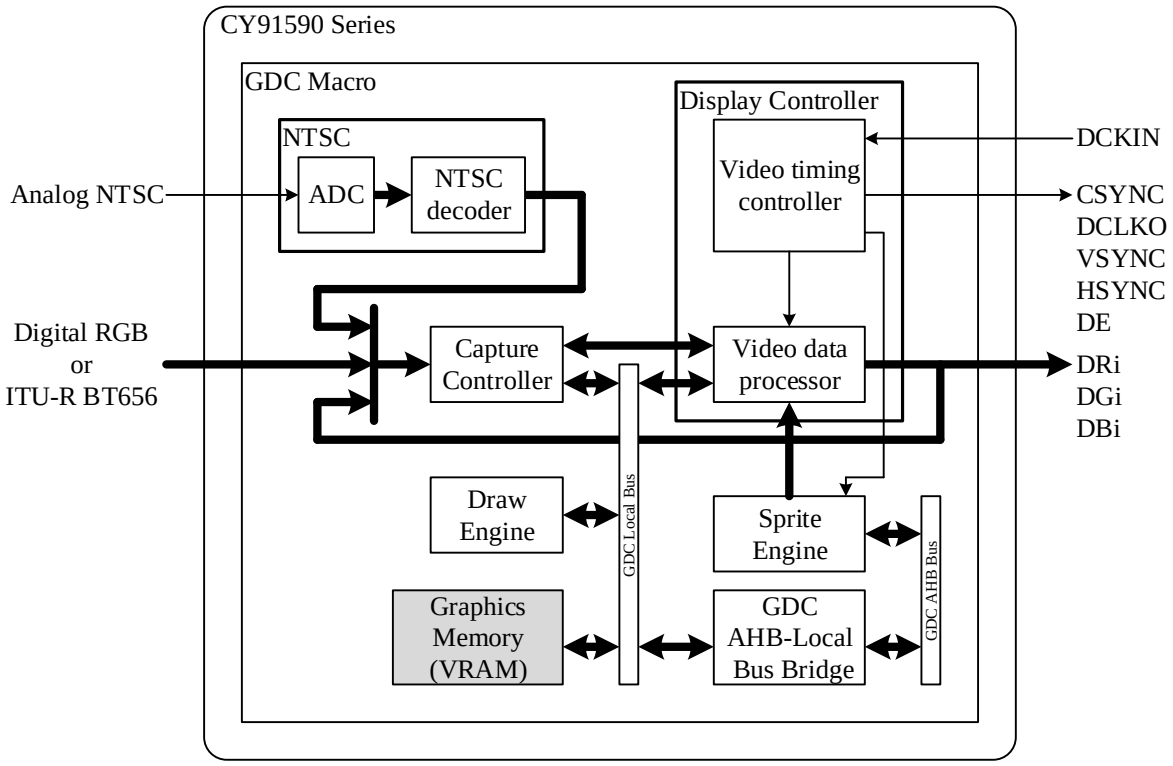
- CY91F591/6
  260KB RAM
- CY91F592/4/7/9
  800KB RAM
- CY91F59A/B
  1.8MB RAM

1.10.2 Configuration

1.10.2.1 Block Diagram

Figure 1-48 is a block diagram showing VRAM and peripherals.

Figure 1-48. Block Diagram of VRAM and Peripherals



Notes:

- Analog NTSC corresponds to the external pin VIN.
- Digital RGB corresponds to the external pins PA2 to PA7, PB2 to PB7, and PC2 to PC7.
- ITU-R.BT656 corresponds to the external pins PA2 to PA7, PB2, and PB3.
- DCLKI corresponds to the external pin DCKIN.
- CSYNC corresponds to the external pin PG3.
- DCLKO corresponds to the external pin PG4.
- VSYNC corresponds to the external pin PG5.
- HSYNC corresponds to the external pin PG6.
- DE corresponds to the external pin PG7.
- DR*i* corresponds to the external pins P011, P012, and PD2 to PD7.
- DG*i* corresponds to the external pins P013, P014, and PE2 to PE7.
- DB*i* corresponds to the external pins P015, P016, and PF2 to PF7.

1.10.3 Explanation of Operation

1.10.3.1 Memory Configuration

The graphics memory is used to manage display by the draw engine, display controller, and capture controller. The following sections describe the graphics memory configuration.

Drawing Frame

The drawing frame is a rectangular image data area for making 2D drawings. Not only can multiple drawing frames be used but also a drawing frame can be made larger than the display frame to display part of a drawing. The drawing frame can be configured with maximum X and Y resolutions of up to 4096 pixels each, within the range of the graphics memory capacity. The data format uses indirect color (8 bits/pixel) or direct color (16 bits/pixel).

Display Frame

The display frame is a rectangular image data area used for display. The frame can overlay and display up to 4 screens of graphics. Also, 1 layer can be used as the coefficient frame of alpha blending.

1.10.3.2 Types of Data

The draw engine handles the data described below. The display list described here can also be allocated to the Command RAM. The display list can also be used to set a texture tiling pattern. The programmable mask window can exclude the receiving pixels from the signature calculation.

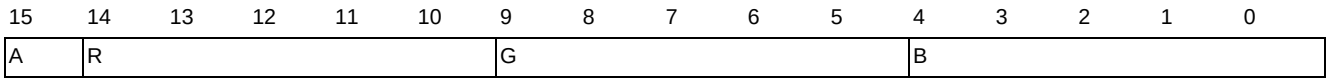
Display list buffer

This is a series of commands and parameters for processing by the draw engine.

1.10.3.3 Data Formats

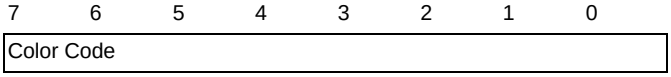
Direct Color (16 Bits/Pixel)

This data format is expressed with RGB, each of which is 5 bits. bit15 is used as an alpha bit to display semi-transparent layers. For other layers, set bit15 to "0".



Indirect Color (8 Bits/Pixel)

This is the color index codes of a look-up table (palette).



Video Image Data

The 4:2:2 YCbCr format uses 16 bits per pixel to store data in memory.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Y0								Cb							

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Y1								Cr							

The RGB666 format uses 24 bits per pixel to store data in memory. The upper 8 bits of bit31 to bit24 are undefined values because the data in memory is aligned in 32 bits.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved								R							

Direct Color (24 Bits/Pixel)

This data format is expressed with RGB, each of which is 8 bits. bit31 is used as an alpha bit to display semi-transparent layers. Note that the 24 bits/pixel color modes are not equipped with a drawing function, so use the CPU to, for example, write directly to graphics memory and draw.

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

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
A	Reserved							R							

1.10.3.4 Frame Control

Single Buffer

All or part of the drawing frame acts as the display frame. Scrolling of the screen can be done with the position of the display frame being moved every frame. If the display frame crosses the boundary of the drawing frame, the frame is displayed as connected to the respective top, bottom, left, or right side of the drawing frame.

Drawing data can be transferred in the blanking period without affecting display.

Double Buffer

After 2 drawing frames are created, drawing processing is executed in 1 frame while the other frame is displayed. Flicker-free animation can be displayed with the display frame and drawing frame being alternately switched every frame. The display frame is switched in the blanking period. The 2 switching modes are the mode for switching at every frame and the user-controlled switching mode.

1.11 Command Sequencer (CMDSEQ)

1.11.1 Overview

The Command Sequencer (referred to below as CMDSEQ) interprets the command list in the specified area and processes it sequentially. This module can be started by the cancellation of a reset, by a trigger signal, or by a register setting.

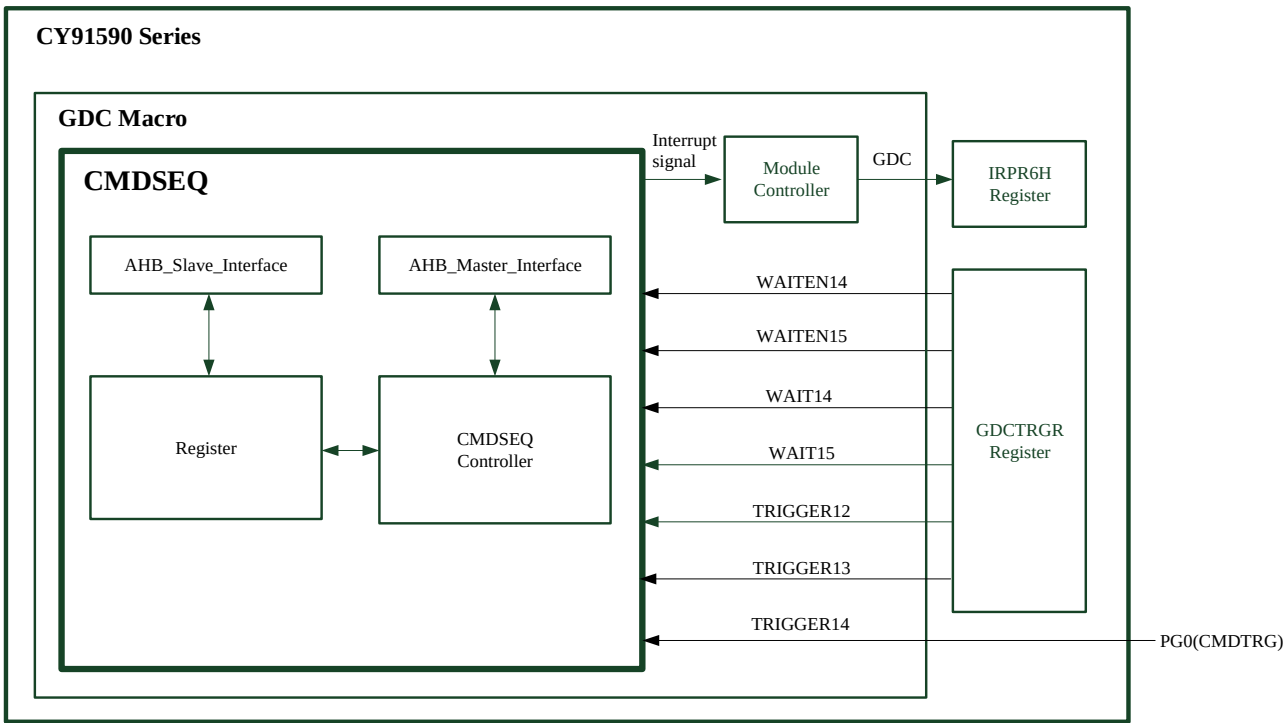
1.11.2 Features

- The processing of unique command lists enables the following operations:
  - ☐ Writing data to a register or memory
  - ☐ Comparing a value in a register or memory to the expected value
  - ☐ Wait processing based on the specified signal
- The module can be started in the following ways:
  - ☐ Start caused by the cancellation of a reset
  - ☐ Start caused by a trigger signal
  - ☐ Start caused by a register setting
- The module can be forcibly terminated while processing a command list.

  Command list processing ends when any of the following errors is detected.
  - ☐ The COMPREG or COMPREG2 command detects a value that does not match the expected value.
  - ☐ A slave module access error occurs.
- Interrupts to various processes can be reported to the IRPR6H register of the FR81S via the Module Controller (referred to below as MCNT).
- This module can access only locations within the address area in the GDC macro. To process a command list, configure the address area in the GDC macro.

1.11.3 Configuration

Figure 1-49. CMDSEQ Block Diagram



Note: PG0 (CMDTRG) is an external input signal.

1.11.4 Registers

1.11.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  Add the base address (0040_0000H) for access from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.11.4.2 Register List

Address	Register Name	Description
$01FB_7000_H$	Status	CMDSEQ status register
$01FB_7004_H$	Status enable	Enable/Disable of CMDSEQ interrupt signal
$01FB_7008_H$	Start	Start of CMDSEQ
$01FB_700C_H$	CMDSEQ IDLE status	Status confirmation of IDLE
$01FB_7010_H$	REGISTER start address	Addressing of register start setting
$01FB_7014_H$	Forced termination	Forced termination
$01FB_7018_H$	TRIGGER start enable	Enable/Disable TRIGGER start signal
$01FB_701C_H$	Transfer status 1	Execution status of various starts
$01FB_7020_H$	Transfer status 2	Value of header part
$01FB_7024_H$	Transfer status 3	Value of the first address or base address
$01FB_7028_H$	COMPREG error address hold	Address of comparison error is held
$01FB_702C_H$	Slave access error address hold	Address of slave access error address is held

1.11.4.3 Register Details

Status Register

Address	01FB_7000$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-												BLANE	CMDR	CMDE	-
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W1	R/W1	R/W1	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REGI	TRIGI	PRI3I	RSTI	ENDE	WAITE	EXTF	EXTNB	CMPE	SERR	FTERM	REG	TRIG	PRI3	TWDP1	RESET
R/W	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register indicates the status of this module.

When the status enable register bit corresponding to a bit (RESET, TWDP1, PRI3, TRIG, REG, FTERM, SERR, CMPE, EXTNB, EXTF, WAITE, ENDE, RSTI, PRI3I, TRIGI, REGI, CMDE, CMDR, or BLANE) in this register is enabled, the information for the respective bit is logically added and output as a CMDSEQ interrupt signal to the INTST register of MCNT.

Bit0	RESET (RESET start)
	This bit displays "1" when any of the following factors occurs during a reset start. ■ The BOOT bit in the GDCCR register (**0000_0F65$_H$**) is "1". ■ The BOOT bit in the GDCCR register (**0000_0F65$_H$**) is "0", and the set data value in 0200_0000$_H$ (reference address area) is FFFF_FFFF$_H$. ■ The END command is executed. ■ An expected-value error occurs. ■ A slave module access error is detected. ■ Processing is forcibly terminated. ■ A WAIT command error is detected. To add the information for this bit as an interrupt factor, set "1" in the RESETE bit in the status enable register. This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit1	TWDP1 (Trigger start by Priority1 signal)
	This bit displays "1" when any of the following factors occurs during a trigger start caused by a Priority1 signal (TRG_PRI1). ■ The set data value in 0200_0004H (reference address area) is FFFF_FFFFH. ■ The END command is executed. ■ An expected-value error occurs. ■ A slave module access error is detected. ■ Processing is forcibly terminated. ■ A WAIT command error is detected. To add the information for this bit as an interrupt factor, set the TWDTSE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt

Bit2	PRI3 (trigger start by PRIority3 signal)
	This bit displays "1" when any of the following factors occurs during a trigger start caused by a Priority3 signal (TRG_PRI2). ■ The set data value in 0200_0008H (reference address area) is FFFF_FFFFH. ■ The END command is executed. ■ An expected-value error occurs. ■ A slave module access error is detected. ■ Processing is forcibly terminated. ■ A WAIT command error is detected. To add the information for this bit as an interrupt factor, set the PRI3E bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit3	TRIG (trigger start by various TRIGger)
	This bit displays "1" when any of the following factors occurs during a trigger start caused by a miscellaneous trigger signal (TRIGGER0 to TRIGGER14). ■ The set data value in 0200_000CH to 0200_005CH (reference address area) is FFFF_FFFFH. ■ The END command is executed. ■ An expected-value error occurs. ■ A slave module access error is detected. ■ Processing is forcibly terminated. ■ A WAIT command error is detected. To add the information for this bit as an interrupt factor, set the TRIGE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit4	REG (REGister start interrupt)
	This bit displays "1" when any of the following factors occurs during a register start. ■ The END command is executed. ■ An expected-value error occurs. ■ A slave module access error is detected. ■ Processing is forcibly terminated. ■ A WAIT command error is detected. To add the information for this bit as an interrupt factor, set the REGE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit5	FTERM (Forced TERMination interrupt)
	This bit displays "1" when the FTERM bit in the forced termination register is set to "1". To add the information for this bit as an interrupt factor, set the FTERME bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit6	SERR (module Slave access ERRor interrupt)
	This bit displays "1" when this module receives an error response from the slave module being accessed. To add the information for this bit as an interrupt factor, set the SERRE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt

	1: Interrupt
Bit7	CMPE (CoMParison Error interrupt)
	This bit displays "1" when a comparison with the expected value detects a mismatch during processing of the COMPREG or COMPREG2 command list. To add the information for this bit as an interrupt factor, set the CMPEE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit8	EXTNB (EXT_NonBoot interrupt)
	This bit displays "1" when the BOOT bit in the GDCCR register (**0000_0F65$_H$**) is "1" at the reset start time. To add the information for this bit as an interrupt factor, set the EXTNBE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit9	EXTF (the data of EXTernal Flash-if are all "F")
	This bit displays "1" when any of the following factors occurs. ■ The BOOT bit in the GDCCR register (**0000_0F65$_H$**) is "0" at the reset start time, and the set data value in 0200_0000$_H$ (reference address area) consists of only "F". ■ The set data value in 0200_0004H to 0200_0044H (reference address area), which is accessed at the trigger start time, consists of only "F". To add the information for this bit as an interrupt factor, set the EXTFE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit10	WAITE (WAIT trigger Error interrupt)
	This bit displays "1" when the WAIT Trigger signal disabled by the WAIT9 command list or by the WAITEN14 or WAITEN15 bit in the GDCTRGR register (***0000_0F66$_H$***) is selected from the command list. To add the information for this bit as an interrupt factor, set the WAITE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit11	ENDE (END command Error)
	This bit displays "1" if the END command result does not consist of only "F". If the bit displays "1", check for problems in the processed command list. To add the information for this bit as an interrupt factor, set the ENDEE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit12	RSTI (ReSeT start Interrupt)
	This bit displays "1" when Priority2 reset start processing is forcibly terminated because another start request with a higher priority (TRG_PRI1) was issued during the processing. To add the information for this bit as an interrupt factor, set the RSTIE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt

Bit13	PRI3I (PRIority3 signal start Interrupt)
	This bit displays "1" when Priority3 (TRG_PRI2) trigger start processing is forcibly terminated because another start request with a higher priority (TRG_PRI1) was issued during the processing. To add the information for this bit as an interrupt factor, set the PRI3IE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit14	TRIGI (various TRIGger start Interrupt)
	This bit displays "1" when Priority4 (TRIGGER0 to TRIGGER14) trigger start processing is forcibly terminated because another start request with a higher priority (TRG_PRI1 or TRG_PRI2) was issued during the processing. To add the information for this bit as an interrupt factor, set the TRIGIE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit15	REGI (REGister start Interrupt)
	This bit displays "1" when Priority5 register start processing is forcibly terminated because another start request with a higher priority (trigger start) was issued during the processing. To add the information for this bit as an interrupt factor, set the REGIE bit in the status enable register to "1". This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit17	CMDE (CMDseq Error)
	This bit becomes "1" when the internal state machine of this module makes an unexpected transition due to noise occurrence. Operation within the GDC module is not guaranteed when the bit is "1". If the bit is "1", set the GRST bit in the GDCCR register (0000_0F65H) to "1" to initialize the GDC module. This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit18	CMDR (CMDseq individual Reset completion)
	This bit specifies that a reset be automatically issued only to this module when a mismatch in the number of internal transfers is due to noise occurrence and self-recovery is judged as not possible. The bit becomes "1" when the module is automatically reset and initialized. Operation within the GDC module is not guaranteed when the bit is "1". If the bit is "1", set the GRST bit in the GDCCR register (0000_0F65H) to "1" to initialize the GDC module. This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt
Bit19	BLANE (Byte LANE of compreg2 was all "0")
	This bit is "1" if the byte lane of the COMPREG2 command consists of only "0". If the byte lane of the COMPREG2 command consists of only "0", confirm that there is no problem. This bit is cleared when "1" is written to it.
	0: No interrupt
	1: Interrupt

Status Enable Register

Address	01FB_7004$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-												BLANEE	CMDRE	CMDEE	-
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REGIE	TRIGIE	PRI3IE	RSTIE	ENDEE	WAITEE	EXTFE	EXTNBE	CMPEE	SERRE	FTERME	REGE	TRIGE	PRI3E	TWDP1E	RESETE
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	1	1	0	1	0	0	0	0	0	1

This register controls whether to output the information for the corresponding bits in the status register as interrupt signals.

When each bit (RESETE, TWDP1E, PRI3E, TRIGE, REGE, FTERME, SERRE, CMPEE, EXTNBE, EXTFE, WAITEE, ENDEE, RSTIE, PRI3IE, TRIGIE, REGIE, CMDE, CMDR, or BLANEE) in this register is enabled, the information for the corresponding bit in the status register is logically added and output as a CMDSEQ interrupt signal to MCNT.

Bit0	RESETE (RESET start interrupt Enable)
	Writing "1" in this bit outputs the information for the RESET bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the RESET bit in the status register from being output as an interrupt signal.
	0: RESET bit information disabled
	1: RESET bit information enabled
Bit1	TWDP1E (interrupt enable of Trigger start by Priority1 signal)
	Writing "1" in this bit outputs the information for the TWDP1 bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the TWDP1 bit in the status register from being output as an interrupt signal.
	0: TWDP1 bit information disabled
	1: TWDP1 bit information enabled
Bit2	PRI3E (interrupt enable of trigger start by PRIority3 signal)
	Writing "1" in this bit outputs the information for the PRI3 bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the PRI3 bit in the status register from being output as an interrupt signal.
	0: PRI3 bit information disabled
	1: PRI3 bit information enabled
Bit3	TRIGE (interrupt enable of trigger start by various TRIGger)
	Writing "1" in this bit outputs the information for the TRIG bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the TRIG bit in the status register from being output as an interrupt signal.
	0: TRIG bit information disabled
	1: TRIG bit information enabled

Bit4	REGE (REGister start interrupt Enable)
	Writing "1" in this bit outputs the information for the REG bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the REG bit in the status register from being output as an interrupt signal.
	0: REG bit information disabled
	1: REG bit information enabled
Bit5	FTERME (Forced TERMination interrupt Enable)
	Writing "1" in this bit outputs the information for the FTERM bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the FTERM bit in the status register from being output as an interrupt signal.
	0: FTERM bit information disabled
	1: FTERM bit information enabled
Bit6	SERRE (module Slave access ERRor interrupt Enable)
	Writing "1" in this bit outputs the information for the SERR bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the SERR bit in the status register from being output as an interrupt signal.
	0: SERR bit information disabled
	1: SERR bit information enabled
Bit7	CMPEE (CoMParison Error interrupt Enable)
	Writing "1" in this bit outputs the information for the CMPE bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the CMPE bit in the status register from being output as an interrupt signal.
	0: CMPE bit information disabled
	1: CMPE bit information enabled
Bit8	EXTNBE (EXT_NonBoot interrupt Enable)
	Writing "1" in this bit outputs the information for the EXTNB bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the EXTNB bit in the status register from being output as an interrupt signal.
	0: EXTNB bit information disabled
	1: EXTNB bit information enabled
Bit9	EXTFE (the data of EXTenal Flash-if are all "F" Enable)
	Writing "1" in this bit outputs the information for the EXTF bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the EXTF bit in the status register from being output as an interrupt signal.
	0: EXTF bit information disabled
	1: EXTF bit information enabled
Bit10	WAITEE (WAIT trigger Error interrupt Enable)
	Writing "1" in this bit outputs the information for the WAITE bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the WAITE bit in the status register from being output as an interrupt signal.
	0: WAITE bit information disabled
	1: WAITE bit information enabled
Bit11	ENDEE (END command Error interrupt Enable)
	Writing "1" in this bit outputs the information for the ENDE bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the ENDE bit in the status register from being output as an interrupt signal.
	0: ENDE bit information disabled
	1: ENDE bit information enabled
Bit12	RSTIE (ReSeT start Interrupt Enable)
	Writing "1" in this bit outputs the information for the RSTI bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the RSTI bit in the status register from being output as an interrupt signal.
	0: RSTI bit information disabled
	1: RSTI bit information enabled
Bit13	PRI3IE (PRIority3 signal start Interrupt Enable)

	Writing "1" in this bit outputs the information for the PRI3I bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the PRI3I bit in the status register from being output as an interrupt signal.
	0: PRI3I bit information disabled
	1: PRI3I bit information enabled
Bit14	TRIGIE (various TRIGger start Interrupt Enable)
	Writing "1" in this bit outputs the information for the TRIGI bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the TRIGI bit in the status register from being output as an interrupt signal.
	0: TRIGI bit information disabled
	1: TRIGI bit information enabled
Bit15	REGIE (REGister start Interrupt Enable)
	Writing "1" in this bit outputs the information for the REGI bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the REGI bit in the status register from being output as an interrupt signal.
	0: REGI bit information disabled
	1: REGI bit information enabled
Bit17	CMDEE (CMDseq Error Enable)
	Writing "1" in this bit outputs the information for the CMDE bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the CMDE bit in the status register from being output as an interrupt signal.
	0: CMDE bit information disabled
	1: CMDE bit information enabled
Bit18	CMDRE (CMDseq individual Reset completion Enable)
	Writing "1" in this bit outputs the information for the CMDR bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the CMDR bit in the status register from being output as an interrupt signal.
	0: CMDR bit information disabled
	1: CMDR bit information enabled
Bit19	BLANEE (Byte LANE of compreg2 was all "0" Enable)
	Writing "1" in this bit outputs the information for the BLANE bit in the status register as an interrupt signal. Writing "0" in this bit prevents the information for the BLANE bit in the status register from being output as an interrupt signal.
	0: BLANE bit information disabled
	1: BLANE bit information enabled

Start Register

Address	$01FB_7008_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															START
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can configure the beginning of a register start.	
Bit0	START (register START)
	With a setting of "1" in this bit, this module begins command list processing with a register start. Note that any attempt to configure this register during register start processing is ignored. "0" is automatically set in the bit when any of the following factors occurs. ■ The END command is executed during a register start. ■ Interrupts due to a trigger start (due to a Priority1, Priority3, or Priority4 signal) are enabled, and a register start is forcibly terminated. ■ A slave module access error is detected. ■ An expected-value error occurs. ■ Processing is forcibly terminated. Before setting the bit to "1", set the REGADR bit in the REGISTER start address register.
	0: No request for the processing caused by a register start, or end of the processing
	1: Waiting for the processing caused by a register start, or currently executing the processing

CMDSEQ IDLE Status Register

Address	01FB_700C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															IDLE
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register indicates whether this module is in the IDLE state.
The register can only be read.

Bit0	IDLE (state of IDLE)
	This bit displays "0" if this module is in the IDLE state. The module is in the IDLE state when it has not been started using any of the start methods (reset start, trigger start, or register start). The bit can only be read.
	0: IDLE state
	1: WORK state

REGISTER Start Address Register

Address	01FB_7010$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	REGADR[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	REGADR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can set the address for executing a register start.

Bit31 to Bit0	REGADR (REGister start ADdRess)
	These bits specify the address at which to begin a register start. The lower 2 bits of this register are fixed at "0" because only word access is supported for command list reading. These bits cannot be rewritten while "1" is the setting of the START bit in the start register. Even after register start processing, the settings of these bits are not cleared.

Forced Termination Register

Address	01FB_7014$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															FCTERM
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W1
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register can configure the forced termination of a command list being executed.

Bit0	FCTERM (ForCed TERMination)		
	With a setting of "1" in this bit, this module terminates processing after the end of the command list being executed. After processing is forcibly terminated, the bit is automatically reset to "0".		
	R	0:	No request for forced termination, or forced termination completed
		1:	Forced termination in progress
	W	0:	Don't care
		1:	Processing forcibly terminated

TRIGGER Start Enable Register

Address	01FB_7018$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-												TSET			
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-													PRI3ET	PRI1ET	-
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R/W	R/W	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register enables the individual signals of Priority1, Priority3, and Priority4 for trigger start.

Bit1	PRI1ET (PRIority1 trigger start Enable by TRG_PRI1)	
	With this bit enabled, this module executes a trigger start upon detection of a TRG_PRI1 rising edge. After the end of the Priority1 start caused by TRG_PRI1, the bit is not cleared. For details on connected trigger signals, see Table 1-17 in "Trigger Start."	
	0:	Disable trigger start by TRG_PRI1.
	1:	Enable trigger start by TRG_PRI1.

Bit2	PRI3ET (PRIority3 trigger start Enable by TRG_PRI2)
	With this bit enabled, this module executes a trigger start upon detection of a TRG_PRI2 rising edge. After the end of the Priority3 trigger start, the bit is not cleared. For details on connected trigger signals, see Table 1-17 in "Trigger Start."
	0: Disable trigger start by TRG_PRI2.
	1: Enable trigger start by TRG_PRI2.
Bit19 to Bit16	TSET (Trigger Start Enable by TRIGGER0-14)
	These bits enable only one of TRIGGER0 to TRIGGER14 to be the Priority4 signal used for a trigger start. With these bits enabled, this module executes a trigger start upon detection of a rising edge of the selected trigger signal. After the end of the Priority4 trigger start, these bits are not cleared. If you want to change these bits during a Priority4 trigger start, output a start completion interrupt, and then set the bits. For details on connected trigger signals, see Table 1-17 in "Trigger Start."
	0: Disable the Priority4 signal.
	1: Enable TRIGGER0 as the Priority4 signal.
	2: Enable TRIGGER1 as the Priority4 signal.
	3: Enable TRIGGER2 as the Priority4 signal.
	4: Enable TRIGGER3 as the Priority4 signal.
	5: Enable TRIGGER4 as the Priority4 signal.
	6: Enable TRIGGER5 as the Priority4 signal.
	7: Enable TRIGGER6 as the Priority4 signal.
	8: Enable TRIGGER7 as the Priority4 signal.
	9: Enable TRIGGER8 as the Priority4 signal.
	10: Enable TRIGGER9 as the Priority4 signal.
	11: Enable TRIGGER10 as the Priority4 signal.
	12: Enable TRIGGER11 as the Priority4 signal.
	13: Enable TRIGGER12 as the Priority4 signal.
	14: Enable TRIGGER13 as the Priority4 signal.
	15: Enable TRIGGER14 as the Priority4 signal.

Transfer Status Register 1 (Confirmation of Start)

Address	01FB_701C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-											REGS	TTRIG	TPRI3	WDTP1	RESETS
R/W	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R0	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register shows which of the start methods is used for start processing. If all the bits in the register are "0", this module is in the IDLE state.	
Bit0	RESETS (RESET Status)
	This bit enables a check of whether a reset start is in progress. The bit can only be read.
	0: No reset start
	1: Reset start in progress
Bit1	WDTP1 (trigger start by TRG_PRI1)
	This bit enables a check of whether a trigger start caused by TRG_PRI1 is in progress. The bit can only be read.
	0: No trigger start caused by TRG_PRI1
	1: Trigger start caused by TRG_PRI1 in progress
Bit2	TPRI3 (Trigger start by TRG_PRI2)
	This bit enables a check of whether a trigger start caused by TRG_PRI2 is in progress. The bit can only be read.
	0: No trigger start caused by TRG_PRI2
	1: Trigger start caused by TRG_PRI2 in progress
Bit3	TTRIG (Trigger start by TRIGGER0-14)
	This bit enables a check of whether a trigger start caused by TRIGGER0 to TRIGGER14 is in progress. The bit can only be read.
	0: No trigger start caused by TRIGGER0 to TRIGGER14
	1: Trigger start caused by TRIGGER0 to TRIGGER14 in progress
Bit4	REGS (REGister Status)
	This bit enables a check of whether a register start is in progress. The bit can only be read.
	0: No register start
	1: Register start in progress

Transfer Status Register 2 (Command Header Part Value)

Address	01FB_7020$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CMDLIST[31:16]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMDLIST[15:0]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register enables a check of the value of the header part of the command list being executed.	
Bit31 to Bit0	CMDLIST (CoMmanD LIST header part value)
	These bits enable a check of the value of the header part of the command list being executed. Only a reset (the GRST bit in the GDCCR register (**0000_0F65$_H$**) is "1") can clear the bits to their initial values. The bits can only be read.

Transfer Status Register 3 (Access Address)

Address	01FB_7024$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CMDADDR[31:16]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMDADDR[15:0]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register displays the first address or base address that is set after the header part for the SETREG, OSETREG, COMPREG, or COMPREG2 command being executed.	
Bit31 to Bit0	CMDADDR (CoMmanD list ADDRess)
	These bits enable a check of the set address for the command list being executed. The bits are cleared when the executed command list ends normally or when there is a reset. The bits can only be read.

COMPREG Error Address Hold Register

Address	01FB_7028$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CMPEADDR[31:16]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CMPEADDR[15:0]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register retains the address where the COMPREG command encountered an expected-value error.	
Bit31 to Bit0	CMPEADDR (CoMParison Error ADDRess)
	These bits retain the address where the COMPREG or COMPREG2 command encountered an expected-value error. This register retains only the address of the first error in a start method. Only a reset (the GRST bit in the GDCCR register (**0000_0F65$_H$**) is "1") can clear the bits to their initial values. The bits can only be read.

Slave Access Error Address Hold Register

Address	01FB_702C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	SEADDR[31:16]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEADDR[15:0]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

This register retains the error address for an error response returned in access of the slave module.	
Bit31 to Bit0	SEADDR (Slave access Error ADDRess)
	These bits retain the error address for an error response returned in access of the slave module. This register retains only the address of the first error. If the error response is returned again after the SERR bit in the status register is cleared, this register retains the new error address. Only a reset (the GRST bit in the GDCCR register (**0000_0F65$_H$**) is "1") can clear the bits to their initial values. The bits can only be read.

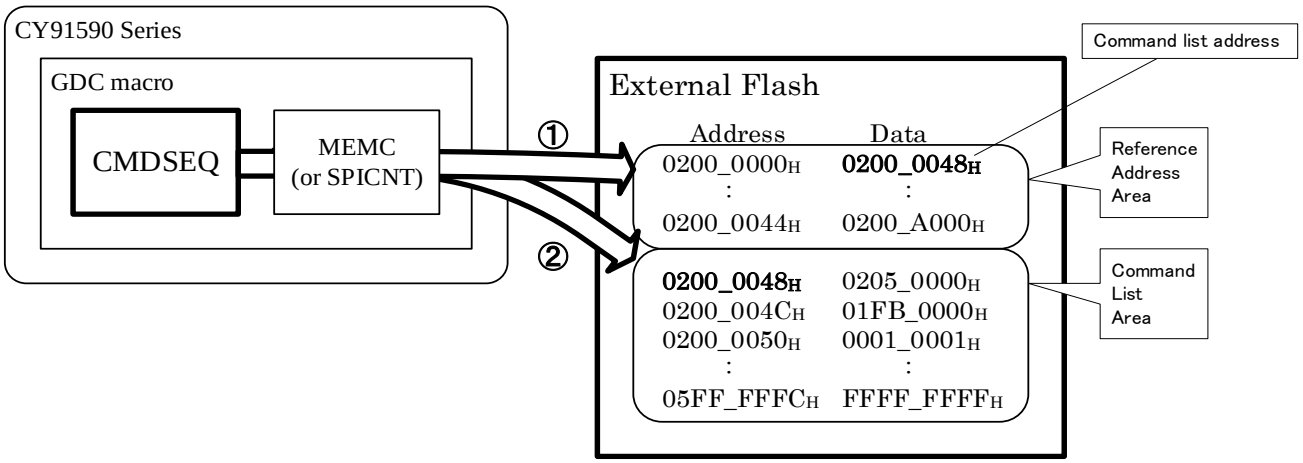
1.11.5 Explanation of Operation

1.11.5.1 Overview of Start Processing

This section explains operation as divided into 2 types of processing: one requires external Flash memory, and the other does not.

Processing that requires external flash memory

Figure 1-50. Example of Processing That Requires External Flash Memory



① After the start, a predetermined address (reference address area) in the external Flash memory is read-accessed via MEMC (or SPICNT).

② The data value (command list address) read at ① is recognized as the next address to be accessed. The command list at this address is processed sequentially.

The following start methods are processing that requires external Flash memory:

- Reset start
- Trigger start

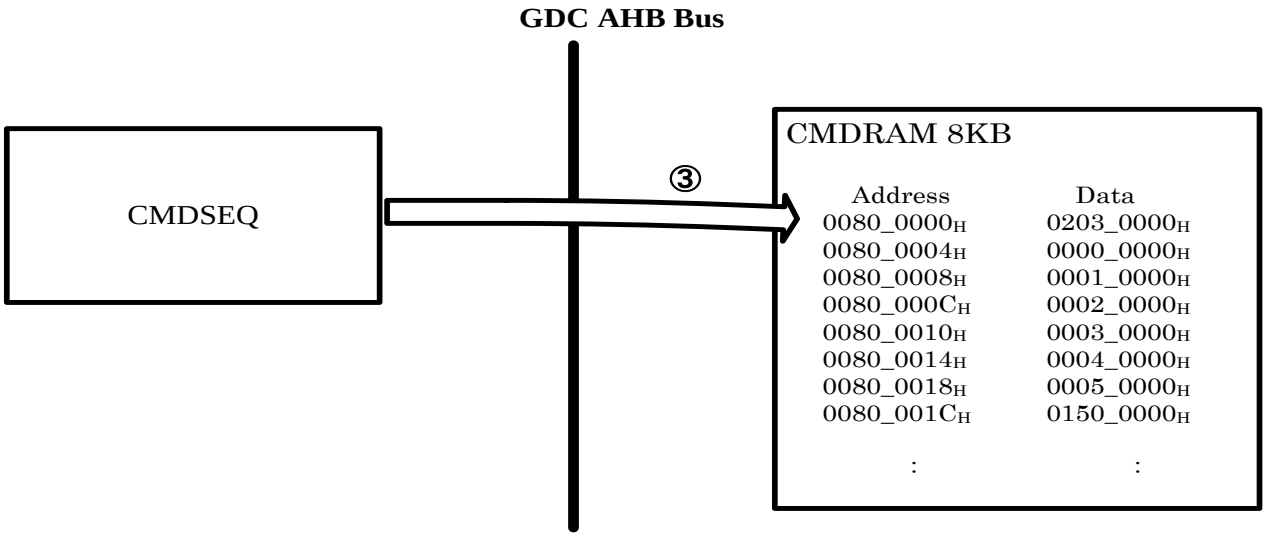
The command list area for a trigger start can be placed not only in external Flash memory but also in the VRAM or CMDRAM inside the GDC macro.

Notes:

- The reference address area is from 0200_0000_H to 0200_0044_H.
- Use a GDC memory map address for a command list address stored in the reference address area. If the base address (**0040_0000_H**) from the FR81S is added, CMDSEQ cannot execute the command list normally.
- If a reset start is used, the command list area is in external Flash memory.
- If NOR Flash memory is used, the command list area is in CS0.
- If the command list area is in memory other than the above, a reset start cannot be executed normally.

Processing that does not require external flash memory

Figure 1-51. Example of Processing That Does Not Require External Flash Memory



③ After starting, this module sequentially processes the command list at the specified address.

The following start method is processing that does not require external Flash memory:

- Register start

1.11.5.2 Start Methods

The start methods for this module are reset start, trigger start, and register start, and each has a different method of execution. The execution triggers for the start methods are listed below.

- Reset start

  Start when bit0 in the GDCCR register (**0000_0F65$_H$**) is set to "1". (Priority2)

- Trigger start
  - ☐ Start when the Priority1 signal (TRG_PRI1) is enabled and the enable register (TRIGGER start enable register) is enabled. (Priority1)
  - ☐ Start when the Priority3 signal (TRG_PRI2) is enabled and the enable register (TRIGGER start enable register) is enabled. (Priority3)
  - ☐ Start when the Priority4 signal (TRIGGER0 to TRIGGER14) is enabled and the enable register is enabled. (Priority4)
- Register start

  Start according to the start register of this module. (Priority5)

The following sections describe each kind of start.

Reset Start

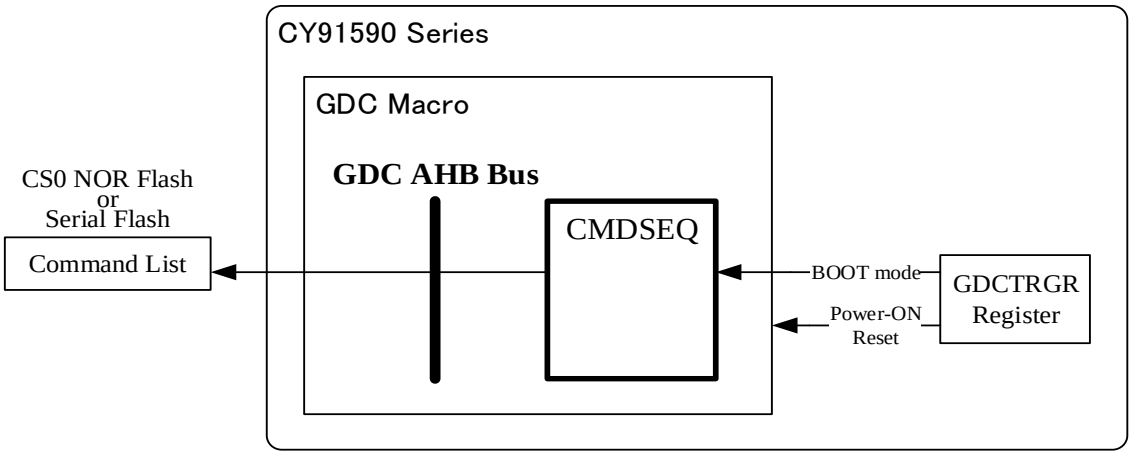
This kind of start begins processing a command list when the GDC macro reset is released (bit0 in the GDCCR register (**0000_0F65$_H$**) is set to "0").

If BOOT mode is enabled (bit1 in the GDCCR register (**0000_0F65$_H$**) is "0"), the set data in 0200_0000$_H$ (reference address area) is read via the AHB Bus. The read data value is the address of the location of the command list. This module sequentially processes the command list at that address.

If BOOT mode is disabled (bit1 in the GDCCR register (**0000_0F65$_H$**) is "1"), this module terminates operation with a completion notification retained in the status register (RESET=1).

After the end of processing with this kind of start, the INTST register (bit0) of MCNT is notified, with a completion notification retained in the status register (RESET=1).

Figure 1-52. Overview of Reset Start



The factors causing the output of a completion notification for this kind of start are listed below. The following factors also set the RESET bit in the status register to "1".

- BOOT mode is disabled (the BOOT bit (bit1) in the GDCCR register (**0000_0F65$_H$**) is "1").
- The set data value in 0200_0000$_H$ (reference address area) is FFFF_FFFF$_H$.
- The END command is executed.
- An expected-value error occurs.
- A slave module access error is detected.
- Processing is forcibly terminated.
- A WAIT command error is detected.

If an operation request based on a Priority1 start request (TRG_PRI1 is "1") is issued during the processing of this function, this processing is forcibly terminated to accept the Priority1 request. If an operation request other than Priority1 is issued during the processing of this function, the request is accepted after all command list processing is completed.

Trigger Start

This kind of start begins command list processing when a trigger signal is enabled. The trigger signal types are as follows.

- Start when the enable register (TRIGGER start enable register) is enabled and a rising edge of the Priority1 signal (TRG_PRI1) is detected. (Priority1)
- Start when the enable register (TRIGGER start enable register) is enabled and a rising edge of the Priority3 signal (TRG_PRI2) is detected. (Priority3)
- Start when the enable register (TRIGGER start enable register) is enabled and a rising edge of the Priority4 signal (TRIGGER0 to TRIGGER14), which users can freely set in the register, is detected. (Priority4)

To enable this kind of start, the enable bit in a register in this module must be enabled before the corresponding trigger signal is enabled. If the enable bit is disabled, the trigger signal cannot be detected. The 2 types of registers for the enable setting have different functions.

Table 1-15. TRIGGER Start Enable Register and Function

Register Name	Operation When Enable Register Is Made Effective
TRIGGER start enable register	CMDSEQ starts and processes the command list when the trigger becomes effective.

Note: If a trigger start request occurs while another trigger start request is being handled, the new request is discarded.

With this kind of start enabled, this module reads the set data in the reference address area in external Flash memory. The read data value is the address of the location of the command list. This module sequentially processes the command list at that address. The following table indicates the first accessed address in the reference address area in the external Flash memory, when this kind of start is enabled by each trigger.

Table 1-16. Access Address List for External Flash

External Flash Access Address	Start Factor
0200_0004_H	TRG_PRI1 (Priority1 signal)
0200_0008_H	TRG_PRI2 (Priority3 signal)
0200_000C_H	TRIGGER0 (Priority4 signal)
0200_0010_H	TRIGGER1 (Priority4 signal)
0200_0014_H	TRIGGER2 (Priority4 signal)
0200_0018_H	TRIGGER3 (Priority4 signal)
0200_001C_H	TRIGGER4 (Priority4 signal)
0200_0020_H	TRIGGER5 (Priority4 signal)
0200_0024_H	TRIGGER6 (Priority4 signal)
0200_0028_H	TRIGGER7 (Priority4 signal)
0200_002C_H	TRIGGER8 (Priority4 signal)
0200_0030_H	TRIGGER9 (Priority4 signal)
0200_0034_H	TRIGGER10 (Priority4 signal)
0200_0038_H	TRIGGER11 (Priority4 signal)
0200_003C_H	TRIGGER12 (Priority4 signal)
0200_0040_H	TRIGGER13 (Priority4 signal)
0200_0044_H	TRIGGER14 (Priority4 signal)

Note: Do not set addresses other than GDC macro addresses in the reference address area.
This module cannot access any location outside the GDC macro address area.

The following table lists the TRIGGER signals for connections.

Table 1-17. TRIGGER Signal Connection List

Trigger Signal Name	Output Module	Trigger Condition
TRG_PRI1	SIG	SIG error count reached interrupt
TRG_PRI2	NTSC	No signal detection
TRIGGER0	SIG	SIG error count reached interrupt
TRIGGER1	Display	FSYNC interrupt
TRIGGER2	Display	VSYNC interrupt
TRIGGER3	SPE	Specified line processing over
TRIGGER4	SPE	Line blank
TRIGGER5	RLD	Byte count achieved, AHB Slave Error, Input FIFO empty, Input FIFO full
TRIGGER6	DMAC	DMAC ch0 interrupt
TRIGGER7	DMAC	DMAC ch1 interrupt
TRIGGER8	Capture	Frame sync
TRIGGER9	Capture	656 stream error
TRIGGER10	Draw	DMA command execution
TRIGGER11	Draw	INT command execution
TRIGGER12	GDCTRGR register in FR81s	TRG12 bit (bit0)
TRIGGER13	GDCTRGR register in FR81s	TRG13 bit (bit1)
TRIGGER14	External pin (PG0:CMDTRG)	External trigger[1]

[1]: For TRIGGER14 for an external trigger when a rising edge is detected, enter a High pulse of 160 ns or more.

After the end of processing with this kind of start, a completion notification is retained in the status register. The status register bit indicating the completion notification varies depending on the start trigger. The following table outlines the relationship between the start triggers and status register bits.

Table 1-18. Relationship between Start Factors and Status Register Bits

Bit Name of Status Register	Kind of Trigger Signal
TWDP1	TRG_PRI1 (Priority1 signal)
TPRI3	TRG_PRI2 (Priority3 signal)
TRIG	TRIGGER0 to TRIGGER14 (Priority4 signal)

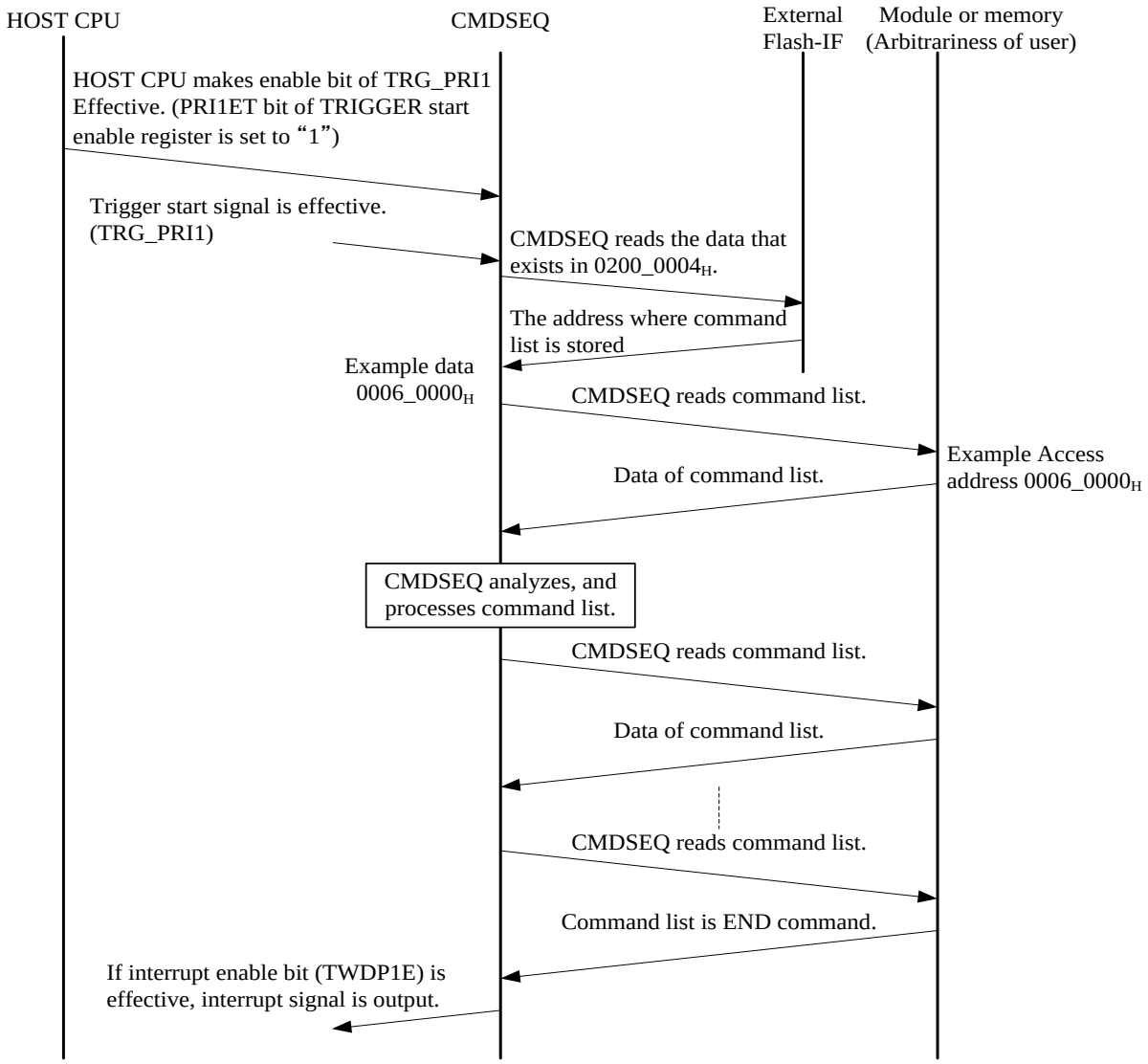
The factors causing the output of a completion notification for this kind of start are listed below. The following factors also set the respective bits in the status register to "1".

- The set data value in 0200_0004_H to 0200_0044_H (reference address area), which is accessed at the start time, is $FFFF_FFFF_H$.
- The END command is executed.
- An expected-value error occurs.
- A slave module access error is detected.
- Processing is forcibly terminated.
- A WAIT command error is detected.

The following figure shows a sample flowchart of a trigger start.

To enable the trigger start caused by the Priority1 signal (TRG_PRI1), the host CPU enables the enable bit (the PRI1ET bit in the TRIGGER start enable register). CMDSEQ reads the value in 0200_0004$_H$ (reference address area) when the Priority1 signal (TRG_PRI1) becomes "1", to recognize the address where the command list is located. After sequential processing of the command list, if the interrupt enable bit (the TWDP1E bit in the status enable register) is enabled, this module outputs an interrupt signal.

Figure 1-53. Trigger Start Flow



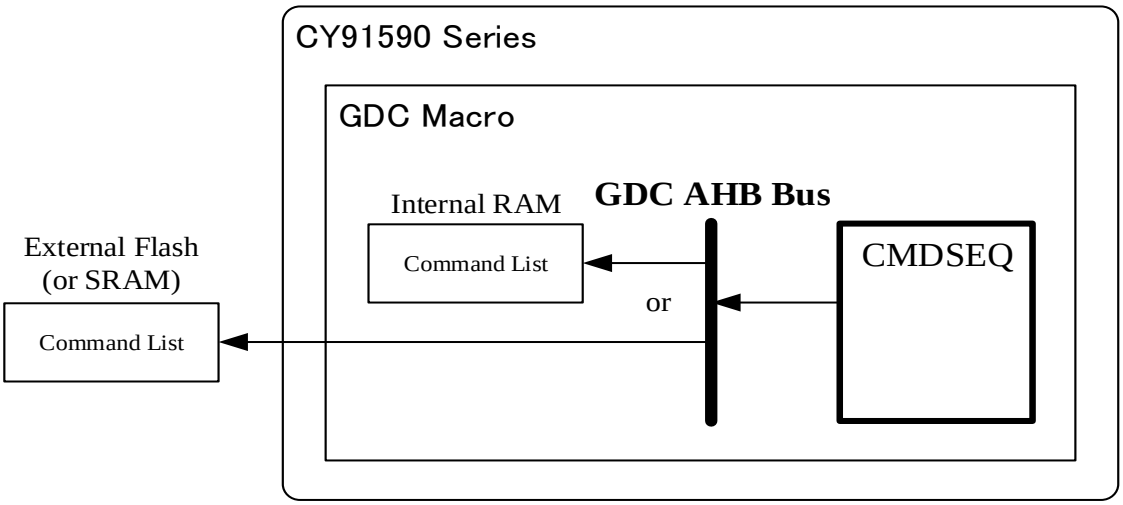
Register Start

This module executes the command list at the set address according to the start address register setting (REGISTER start address register). The setting of "1" in the start register starts this function.

Until the end of command list processing, a new attempt for this kind of start is not possible.

After the end of processing with this kind of start, a completion notification is retained in the status register (REG=1).

Figure 1-54. Overview of Register Start



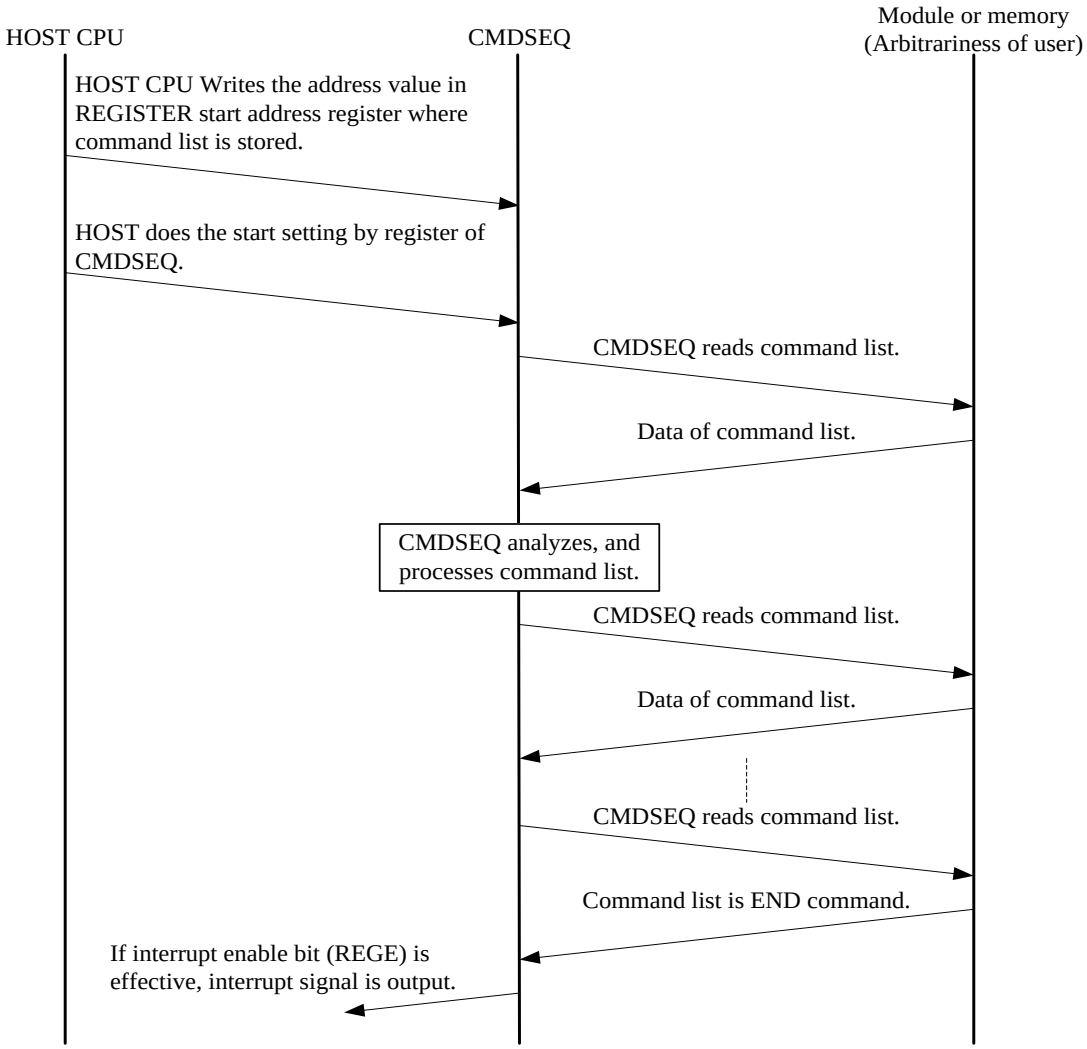
The factors causing the output of a completion notification for this kind of start are listed below. The following factors also set the REG bit in the status register to "1".

- The END command is executed.
- An expected-value error occurs.
- A slave module access error is detected.
- Processing is forcibly terminated.
- A WAIT command error is detected.

The following figure shows a sample flowchart of a register start.

The host CPU sets the address of the location of the command list in the start address register (REGISTER start address register) and sets "1" in the start register. CMDSEQ processes the command list. After sequential processing of the command list, if the interrupt enable bit (the REGE bit in the status enable register) is enabled, this module outputs an interrupt signal.

Figure 1-55. Register Start Flow



1.11.5.3 Priorities

This module executes processing based on the following priorities, where a start with a higher priority is done earlier.

Priority1 > Priority2 > Priority3 > Priority4 > Priority5

Table 1-19. Priorities

Priority	Signal Name or Demand
Priority1	TRG_PRI1 (Trigger start)
Priority2	Reset start
Priority3	TRG_PRI2 (Trigger start)
Priority4	TRIGGER0 to TRIGGER14 (Trigger start)
Priority5	Register start

The following table indicates how the start methods handle an operation request.

Figure 1-56. How the Start Methods Handle an Operation Request

☆ Processing is interrupted and the demand is accepted.
△ After operated, the demand is accepted.
□ The demand is disregarded.

Operation demand \ It's operating	RESET START	TRIGGER START(Pri1)	TRIGGER START(Pri3)	TRIGGER START(Pri4)	REGISTER START
RESET START					
TRIGGER START(Pri1)	☆	□	☆	☆	☆
TRIGGER START(Pri2)	△	△	□	☆	☆
TRIGGER START(Pri3)	△	△	△	□	☆
REGISTER START	△	△	△	△	

1.11.5.4 Command List

A collection of parameter settings and pattern data is called a command list. Execution of a command list enables setting and expected-value comparisons for a register, and wait operations based on various trigger signal counts.

A command list consists of the header part, which contains the packet code, and the parameter parts, which follow the header part.

Header part [31:0]
Parameter parts [31:0]

Header Part

The format of the header part is shown below. The packet code appears in the upper 8 bits in the header part.

The Other [23:0] section varies depending on the command. For details, see the descriptions of the individual command lists.

Packet Code [31:24]	Other [23:0]

Parameter Parts

Only SETREG, OSETREG, COMPREG, and COMPREG2 have the parameter parts. For details on the parameter parts, see the descriptions of the individual command lists of SETREG, OSETREG, COMPREG, and COMPREG2.

Packet Code

The following table lists packet codes. This module identifies a command list by the packet code contained in the header part. For details on command lists, see the next section.

Table 1-20. Command Lists and Packet Codes

Packet Code	Packet Name	Description
0000_0001_B	WAIT Trigger	WAIT by trigger
0000_0010_B	SETREG	Set Register
0000_0011_B	OSETREG	Offset Address Set Register
0000_0100_B	COMPREG	Comparison Register
0000_0101_B	COMPREG2	Comparison Register Part 2
1111_1111_B	END	Command End

Note: If a packet code other than the above is detected, it is considered to be the END command.

Various Command Lists

This section describes each of the following command lists: WAIT Trigger, SETREG, OSETREG, COMPREG, COMPREG2, and END.

WAIT Trigger

This function counts as many signals as the set number in Count. The function does not execute other processing during this time. The signal used as the WAIT Trigger signal and the Count number are specified in the header part of this function.

This section shows the format and setting values of the WAIT Trigger command list.

	31 24	23 16	15 4	3 0
Header part	WAIT=01_H	Count	Reserved	Trigger

Trigger [3:0] These bits select the display trigger signal to be synchronized.
- 0000_B: WAIT0
- 0001_B: WAIT1
- 0010_B: WAIT2
- ... ...
- 1111_B: WAIT15

Count [23:16] These bits detect the edge of the trigger signal as many times as specified by the count value.
- 0_H: Detect the trigger signal 256 times before processing the next command list.
- 1_H: Detect the trigger signal once before processing the next command list.
- 2_H: Detect the trigger signal twice before processing the next command list.
- ... ...
- FF_H: Detect the trigger signal 255 times before processing the next command list.

The following table lists the WAIT Trigger signals that are connected.

Table 1-21. WAIT Trigger Signal Connection List

WAIT Trigger Signal Name	Output Module	WAIT Trigger Condition
WAIT0	-	GSSCGCLK
WAIT1	SIG	SIG error count reached interrupt
WAIT2	Display	FSYNC interrupt
WAIT3	Display	VSYNC interrupt
WAIT4	SPE	Specified line processing over
WAIT5	SPE	Line blank
WAIT6	RLD	Byte count achieved, AHB Slave Error, Input FIFO empty, Input FIFO full
WAIT7	DMAC	DMAC ch0 interrupt
WAIT8	DMAC	DMAC ch1 interrupt
WAIT9	-	Reserved
WAIT10	Capture	Frame sync
WAIT11	Capture	656 stream error
WAIT12	Draw	DMA command execution
WAIT13	Draw	INT command execution
WAIT14	GDCTRGR Register in FR81s	WAIT14 bit (bit4)
WAIT15	GDCTRGR Register in FR81s	WAIT15 bit (bit5)

- WAIT Trigger Enable Signal and Error Detection

To execute the WAIT Trigger command using WAIT14 or WAIT15, set "1" in the WAITEN14 (bit6) or WAITEN15 (bit7) bit in the GDCTRGR register (**0000_0F66$_H$**). Even if the WAITEN14 or WAITEN15 bit is not set, once this module recognizes the WAIT14 or WAIT15 command list (Trigger=E_H or Trigger=F_H), it considers that a WAIT Trigger error has occurred.

The module also considers that a WAIT Trigger error has occurred when it recognizes the WAIT9 command list (Trigger=9_H).

Upon detecting a WAIT Trigger error, this module forcibly terminates command list processing and sets the WAITE bit in the status register to "1".

SETREG

This function sets data in a register or memory area allocated in the memory map. The setting of a count value enables data to be written in consecutive addresses.

The function is used to set data in registers and memory.

This section shows the format and setting values of the SETREG command list.

	31 – 24	23 – 16	15 – 0
Header part	SETREG=02$_H$	Count [23:16]	-
Parameter parts	Address		
	(Data0)		
	(Data1)		
	:		
	(Data255)		

Count [23:16] These bits specify the number of times to set data in a register or memory.

- 0_H: Set data 256 times.
- 1_H: Set data once.
- 2_H: Set data twice.
- ... ...
- FF_H: Set data 255 times.

Address [31:0] These bits specify the first address of the register or memory area at which to start setting data.

Data [31:0] These bits specify the data to set in a register or memory. Up to 256 data items can be set.

The following example shows the SETREG command format.

SETREG Count = 5	Address for setting data	Set data
Address = 1000_0000$_H$	Address	Data
Data0 = 0000_0001$_H$	1000_0000$_H$	0000_0001$_H$
Data1 = 0000_0002$_H$	1000_0004$_H$	0000_0002$_H$
Data2 = 0000_0003$_H$	1000_0008$_H$	0000_0003$_H$
Data3 = 0000_0004$_H$	1000_000C$_H$	0000_0004$_H$
Data4 = 0000_0005$_H$	1000_0010$_H$	0000_0005$_H$

Notes:

- Do not set addresses other than GDC macro addresses in Address [31:0].
- This module cannot access any location outside the GDC macro address area.

OSETREG

This function sets data in a register or memory area allocated in the memory map. The transfer destination address is obtained from the addition of the base address to the offset address. Data can be set in nonconsecutive addresses by the setting of offset addresses with data values. Setting a count value enables continuous data setting. The transfer size can also be set.

This function is used to set data in registers and memory.

This section shows the formats and setting values of the OSETREG command list.

If the sum of the base address and offset address values exceeds 32 bits, a carry-out occurs in the calculation of the transfer destination address.

- 8-bit transfer format

	31 – 24	23 – 16	15 – 8	7 – 2	1 – 0
Header part	OSETREG=03$_H$	Count [23:16]	-		Size=00
	Base Address [31:0]				
	Offset Address0			(Data0)	
Parameter parts	Offset Address1			(Data1)	
	:				
	Offset Address255			(Data255)	

Note: The transfer destination address in 8-bit transfer is as follows.

Base Address [31:0] + Offset Address [31:8]

- 16-bit transfer format

	31 24	23 17	16	15 8	7 2	1 0
Header part	OSETREG=03_H	Count [23:16]		-		Size=01
	Base Address [31:1]					-
	Offset Address0		-	(Data0)		
Parameter parts	Offset Address1		-	(Data1)		
	:					
	Offset Address255		-	(Data255)		

Note: The transfer destination address in 16-bit transfer is as follows.

The lower 1 bit of the transfer destination address is fixed at "0".

Base Address [31:1] + Offset Address [31:17]

- 24-bit transfer format

	31 26	25 24	23 16	15 8	7 2	1 0
Header part	OSETREG=03_H		Count [23:16]	-		Size=10,11
	Base Address [31:2]					-
	Offset Address0	-	(Data0)			
Parameter parts	Offset Address1	-	(Data1)			
	:					
	Offset Address255	-	(Data255)			

Notes:

- The transfer destination address in 24-bit transfer is as follows.

  The lower 2 bits of the transfer destination address are fixed at "0".

  Base Address [31:2] + Offset Address [31:26]

- The upper 8 bits of the transfer data are fixed at "0".

Size [1:0]	These bits select the transfer data size. 00_B: 8-bit transfer (only 1-byte transfer supported) 01_B: 16-bit transfer (only 2-byte transfer supported) 10_B: 24-bit transfer (only 4-byte transfer supported) 11_B: 24-bit transfer (only 4-byte transfer supported)
Count [23:16]	These bits specify the number of times to set data in a register or memory. 0_H: Set data 256 times. 1_H: Set data once. 2_H: Set data twice. ...: ... FF_H: Set data 255 times.
Base Address	These bits specify the base address of the register or memory area at which to set data.
Offset Address	These bits specify the offset address of the register or memory area at which to set data.
Data	These bits specify the data to set in a register or memory. In 24-bit transfer, "0" is always set in the upper bits [31:24]. Up to 256 data items can be set.

The following example shows the OSETREG command format.

OSETREG Count = 7 Size = 0		Address for setting data	Set data
Base Address = 1000_0000_H		Address	Data
Offset Address0 = 00_0001_H	Data0 = 1_H	1000_0001_H	01_H
Offset Address1 = 00_0003_H	Data1 = 2_H	1000_0003_H	02_H
Offset Address2 = 02_0000_H	Data2 = 3_H	1002_0000_H	03_H
Offset Address3 = 24_0001_H	Data3 = 4_H	1024_0001_H	04_H
Offset Address4 = 00_2500_H	Data4 = 5_H	1000_2500_H	05_H
Offset Address5 = 00_c000_H	Data5 = 6_H	1000_c000_H	06_H
Offset Address6 = $c4_0000_H$	Data6 = 7_H	$10c4_0000_H$	07_H

Note: Do not set an address other than a GDC macro address as the base address or offset address. This module cannot access any location outside the GDC macro address area.

COMPREG

This function reads a register allocated in the memory map and compares the read value to set values in the command list. The comparison, in units of 32 bits, determines whether the read value matches the expected value.

This function can be used to, for example, compare a register or memory area value to the expected value and then, if the two values match, process the next command list.

This section shows the format and setting values of the COMPREG command list.

	31 – 24	23 – 16	15 – 0
Header part	COMPREG=04_H	Count [23:16]	-
	Address		
	(Data0)		
Parameter parts	(Data1)		
	...		
	(Data255)		

Count [23:16] These bits compare a data value to register values.

- 0_H: Compare the data value 256 times.
- 1_H: Compare the data value once.
- 2_H: Compare the data value twice.
- ... ...
- FF_H: Compare the data value 255 times.

Address [31:0] These bits specify the first address of the register or memory area at which to start an expected-value comparison.

Data [31:0] These bits specify the expected-value data. Up to 256 data items can be set.

The following example shows the COMPREG command format.

COMPREG Count = 5	Start address for expected-value comparison	Expected-value data
Address = 1000_0000_H	Address	Data
Data0 = 1111_1111_H	1000_0000_H	1111_1111_H
Data1 = 2222_2222_H	1000_0004_H	2222_2222_H

Data2 = 3333_3333$_H$	1000_0008$_H$	3333_3333$_H$
Data3 = 4444_4444$_H$	1000_000C$_H$	4444_4444$_H$
Data4 = 5555_5555$_H$	1000_0010$_H$	5555_5555$_H$

Note: Do not set addresses other than GDC macro addresses in Address [31:0].
This module cannot access any location outside the GDC macro address area.

COMPREG2

This function reads a register allocated in the memory map and compares the read value to set values in the command list. The comparison, in units of bytes, determines whether the read value matches the expected value.

This function can be used to, for example, compare a register or memory area value in units of bytes to the expected value and then, if the two values match, process the next command list.

This section shows the format and setting values of the COMPREG2 command list.

	31 – 24	23 – 16	15 – 4	3 – 0
Header part	COMPREG2=05$_H$	Count [23:16]	-	Byte lane[3:0]
	Address			
	(Data0)			
Parameter parts	(Data1)			
	...			
	(Data255)			

Count [23:16] These bits compare a data value to register values.

- 0_H: Compare the data value 256 times.
- 1_H: Compare the data value once.
- 2_H: Compare the data value twice.
- ... ...
- FF_H: Compare the data value 255 times.

Byte lane [3:0]	These bits specify the byte lane where expected-value comparison is enabled. If "1" is set, there is a comparison of the value in the specified byte lane. The mapping between byte lanes and data is as follows: Byte lane[3] -> Data[31:24] Byte lane[2] -> Data[23:16] Byte lane[1] -> Data[15:8] Byte lane[0] -> Data[7:0]
Address [31:0]	These bits specify the first address of the register or memory area at which to start an expected-value comparison.
Data [31:0]	These bits specify the expected-value data. Up to 256 data items can be set. If "0" is set in a byte lane in Byte lane [3:0], the value is "Don't care."

The following example shows the COMPREG2 command format. The letter X means "Don't care."

COMPREG2 Count = 5 Byte lane = B	**Start address for expected-value comparison**	**Expected-value data**
Address = 1000_0000_H	**Address**	**Data**
Data0 = $11XX_1111_H$	1000_0000_H	$11XX_1111_H$
Data1 = $22XX_2222_H$	1000_0004_H	$22XX_2222_H$
Data2 = $33XX_3333_H$	1000_0008_H	$33XX_3333_H$
Data3 = $44XX_4444_H$	1000_000C_H	$44XX_4444_H$
Data4 = $55XX_5555_H$	1000_0010_H	$55XX_5555_H$

Note: Do not set addresses other than GDC macro addresses in Address [31:0].
This module cannot access any location outside the GDC macro address area.

END

This function terminates command list processing.

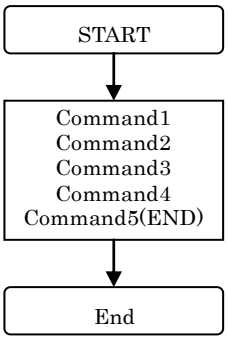
After processing the END command, this module retains a completion notification in the status register.

This section shows the format and setting values of the END command list.

	31	0
Header part	END = FFFF_FFFF$_H$	

Note: If a packet code other than WAIT Trigger, SETREG, OSETREG, COMPREG, or COMPREG2 is detected, it is also considered to be the END command.

The following example shows the END command flowchart.



- END Command Error

  If the END command result does not consist of only "F", "1" is set in the ENDE bit in the status register. If the ENDEE bit in the status enable register is enabled, this module outputs the information for the ENDE bit to an interrupt signal.

1.11.5.5 Slave Module Access Error Response

For any error response for the slave module being accessed, this module retains error information (SERR=1) in the status register and the error address in the slave access error address hold register.

If the SERRE bit in the status enable register is enabled, this module outputs the information for the SERR bit in the status register to an interrupt signal.

For any error response from the slave module, this module terminates start processing without waiting for the completion of the transaction.

1.11.5.6 Forced Termination

A register setting enables forced termination of command list processing in progress due to a reset start, trigger start, or register start. Once this function is enabled, transfer is stopped after the completion of the transaction. To enable the function, set the FTERM bit in the forced termination register to "1".

After the end of processing by this function, a completion notification is retained in the status register (FTERM=1).

If the FTERME bit in the status enable register is enabled, this module outputs the information for the FTERM bit in the status register to an interrupt signal.

1.11.5.7 Transfer Status

The start method (reset start, trigger start, or register start) currently used by this module can be confirmed from first address or offset address in transfer status registers 1 to 3. The address is the next setting after the header part of the command list. For details, see Register Details.

Header part [31:0]	-> Can check in transfer status register 2
Parameter parts_0 [31:0]	-> Can check in transfer status register 3
Parameter parts_1 [31:0]	
Parameter parts_2 [31:0]	
:	

1.11.5.8 Interrupts

This module can logically add and output interrupt signals to MCNT.

The following table lists the factors for logically adding and outputting interrupt signals to MCNT.

Interrupt Factor	Status Register Bit Name	Description
Termination of reset start	RESET	Start processed is terminated and the RESET bit displays "1" when any of the following factors occurs during a reset start. ■ The BOOT bit in the GDCCR register (**0000_0F65H**) is "1". ■ The BOOT bit in the GDCCR register (**0000_0F65H**) is "0", and the set data value in 0200_0000H (reference address area) is FFFF_FFFFH. ■ The END command is executed. ■ An expected-value error occurs. ■ A slave module access error is detected. ■ Processing is forcibly terminated. ■ A WAIT command error is detected.
Termination of Priority1 trigger start	TWDP1	Start processed is terminated and the TWDP1 bit displays "1" when any of the following factors occurs during a trigger start caused by a Priority1 signal (TRG_PRI1). ■ The set data value in 0200_0004H (reference address area) is FFFF_FFFFH. ■ The END command is executed. ■ An expected-value error occurs. ■ A slave module access error is detected. ■ Processing is forcibly terminated. ■ A WAIT command error is detected.
Termination of Priority3 trigger start	PRI3	Start processed is terminated and the PRI3 bit displays "1" when any of the following factors occurs during a trigger start caused by a Priority3 signal (TRG_PRI2). ■ The set data value in 0200_0008H (reference address area) is FFFF_FFFFH. ■ The END command is executed. ■ An expected-value error occurs. ■ A slave module access error is detected. ■ Processing is forcibly terminated. ■ A WAIT command error is detected.

Interrupt Factor	Status Register Bit Name	Description
Termination of Priority4 trigger start	TRIG	Start processed is terminated and the TRIG bit displays "1" when any of the following factors occurs during a trigger start caused by a miscellaneous trigger (TRIGGER0 to TRIGGER14) signal. ■ The set data value in 0200_000C$_H$ to 0200_005C$_H$ (reference address area) is FFFF_FFFF$_H$. ■ The END command is executed. ■ An expected-value error occurs. ■ A slave module access error is detected. ■ Processing is forcibly terminated. ■ A WAIT command error is detected.
Termination of register start	REG	Start processed is terminated and the REG bit displays "1" when any of the following factors occurs during a register start. ■ The END command is executed. ■ An expected-value error occurs. ■ A slave module access error is detected. ■ Processing is forcibly terminated. ■ A WAIT command error is detected.
Completion of forced termination	FTERM	The FTERM bit displays "1" when the FCTERM bit in the forced termination register is set to "1".
Slave error response	SERR	The SERR bit displays "1" when the slave module being accessed by this module returns an error response.
Expected-value error	CMPE	The CMPE bit displays "1" when an expected-value comparison detects a mismatch during COMPREG or COMPREG2 command list processing.
GDC macro reset release in BOOT start mode (reset start)	EXTNB	The EXTNB bit displays "1" when the GDC macro reset is released (bit0 in the GDCCR register (**0000_0F65$_H$**) is set to "0") while BOOT start mode is enabled (bit1 in the GDCCR register (**0000_0F65$_H$**) is "0").
Reference Address = FFFF_FFFF$_H$	EXTF	This bit, EXTF, displays "1" when any of the following factors occurs. ■ BOOT start mode is enabled (bit1 in the GDCCR register (**0000_0F65$_H$**) is "0") when the GDC macro reset is released, and the set data value in 0200_0000H (reference address area) consists of only "F". ■ The set data value in 0200_0004$_H$ to 0200_0044$_H$ (reference address area), which is accessed at the trigger start time, consists of only "F".
WAIT Trigger command error	WAITE	This bit, WAITE, bit displays "1" when any of the following factors occurs. ■ The WAIT9 command list is processed. ■ The WAIT14 command list is processed when WAITEN14 (bit6) in the GDCTRGR register (0000_0F66H) is "0". ■ The WAIT15 command list is processed when WAITEN15 (bit7) in the GDCTRGR register (0000_0F66H) is "0".
END command error	ENDE	The ENDE bit displays "1" if the END command result does not consist of only "F".
Forced termination of reset start caused by interrupt request	RSTI	The RST1 bit displays "1" when a Priority2 reset start is forcibly terminated because another start request with a higher priority (TRG_PRI1) was issued.
Forced termination of Priority3 trigger start caused by interrupt request	PRI3I	The PRI3I bit displays "1" when a Priority3 (TRG_PRI) trigger start is forcibly terminated because another start request with a higher priority (TRG_PRI1) was issued.

Interrupt Factor	Status Register Bit Name	Description
Forced termination of Priority4 trigger start caused by interrupt request	PRI4I	The PRI4I bit displays "1" when a Priority4 (TRIGGER0 to TRIGGER14) trigger start is forcibly terminated because another start request with a higher priority (TRG_PRI1 or TRG_PRI2 trigger start request) was issued.
Forced termination of register start caused by interrupt request	REGI	This bit displays "1" when a Priority5 register start is forcibly terminated because another start (trigger start) request with a higher priority was issued.
Unexpected transition made by state machine	CMDE	This bit displays "1" when the internal state machine of this module makes an unexpected transition due to noise occurrence. Initialize the GDC module (the GRST bit in the GDCCR register (0000_0F65H) is "1") because operation within the GDC module is not guaranteed.
Mismatch in number of internal transfers	CMDR	This bit displays "1" when an automatic reset is issued only to this module because a mismatch in the number of internal transfers is due to noise occurrence and self-recovery is judged as not possible. Initialize the GDC module (the GRST bit in the GDCCR register (0000_0F65H) is "1") because operation within the GDC module is not guaranteed.
Only "0" in byte lane specification for COMPREG2 command	BLANE	This bit displays "1" when this module recognizes the COMPREG2 command but does not find the byte lane specification (Byte lane[3:0]=4'h0) required for expected-value comparison. Check for problems in the specified byte lane.

For status register information, interrupt signals can be logically added and output to MCNT when the corresponding bits in the status enable register are enabled ("1").

1.11.5.9 Interrupt Enable

For status register information, interrupt signals can be logically added and output to MCNT when the corresponding bits in the status enable register are enabled ("1").

1.11.5.10 Interrupt Clear

Setting "1" in a status register bit clears the corresponding interrupt signal to "0".

1.11.6 Notes

1.11.6.1 Restrictions

CY91F591/2/4/6/7/9

- This module can access only the address area within the GDC macro. Any address area that is set for command list processing must be within the GDC macro. If the specified address area is outside the GDC macro, operation cannot be guaranteed.
- If SPICNT is selected, use of reset start is prohibited.
  For details, see "GDC Access Sequence".

CY91F59A/B

- This module can only access the address area within the GDC macro. Any address area that is set for command list processing must be within the GDC macro. If the specified address area is outside the GDC macro, operation cannot be guaranteed.
- If HS-SPICNT is selected, use of reset start is prohibited.
  For details, see "GDC Access Sequence".

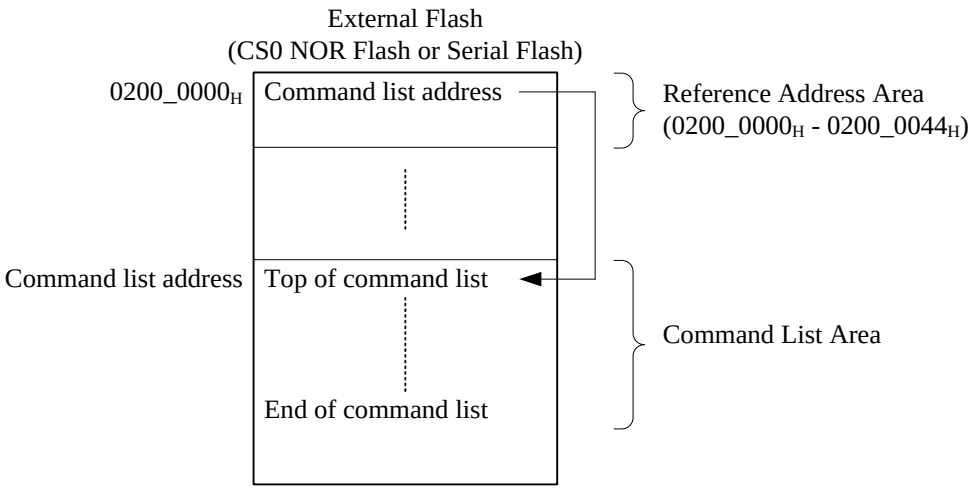
1.11.7 Examples

1.11.7.1 Reset Start

The reset start procedure is as follows:

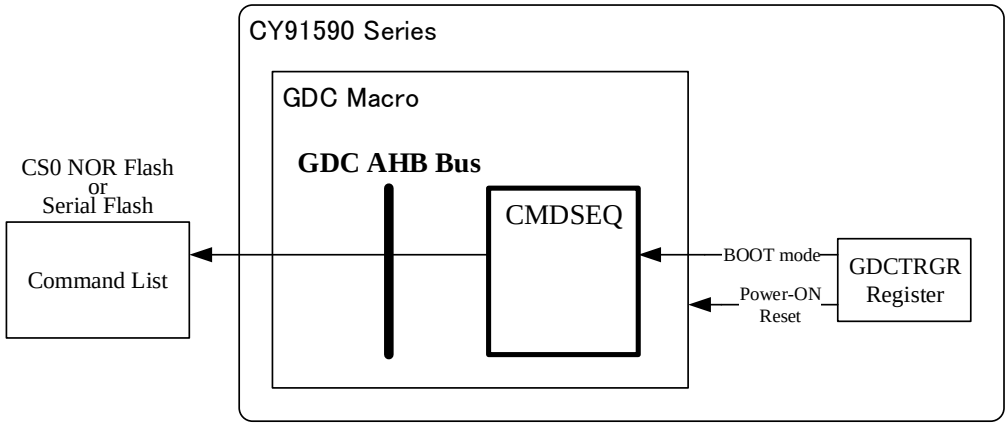
1. Write the command list address and command list to external Flash (CS0 NOR Flash or Serial Flash) memory.
   To output interrupt signals from this module, write the setting that enables the status enable register, in the command list.

Figure 1-57. Conceptual Image of External Flash



2. Enable BOOT mode (set bit1 in the GDCCR register (**0000_0F65H**) to "0").
3. Release the GDC macro reset (set bit0 in the GDCCR register (**0000_0F65H**) to "0").
4. After the GDC macro reset is released, this module sequentially processes the command list at the storage address of the command list (command list address).

Figure 1-58. Conceptual Image of CMDSEQ Start



5. After the end of command list processing, status information appears in the status register. If the status enable register is enabled, the corresponding status register bit information is output as an interrupt signal to the INTST register.

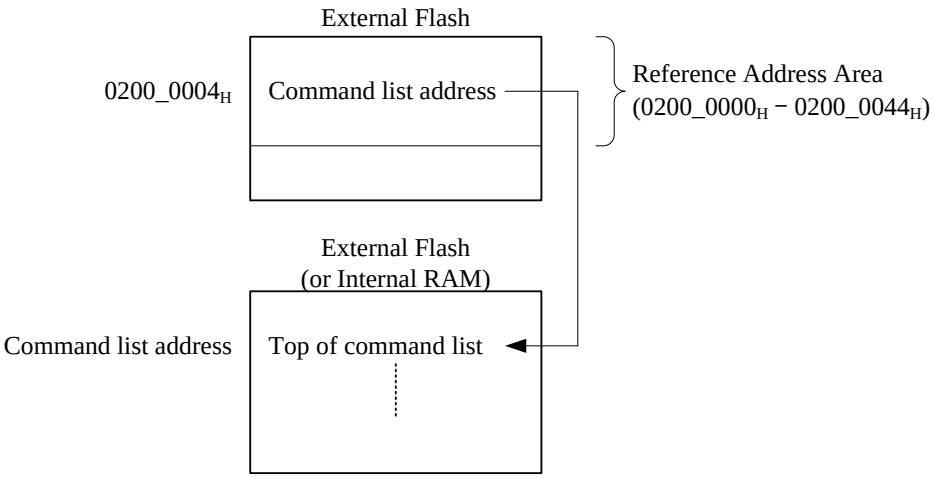
1.11.7.2 Trigger Start

Start by Priority1

The procedure for a trigger start caused by a Priority1 signal (TRG_PRI1) is as follows:

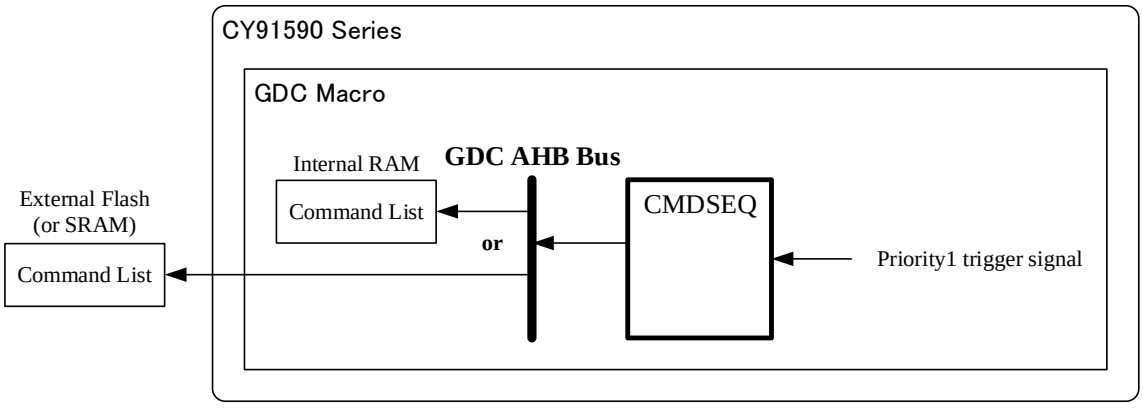
1. Store both the value of the storage address for the command list (command list address) and the command list beginning at the command list address (in external Flash memory or internal RAM) in advance in 0200_0004_H (reference address area).
   To output interrupt signals from this module, write the setting that enables the status enable register, in the command list.

Figure 1-59. Conceptual Image of Data Storage



2. Set "1" in the PRI1ET bit in the TRIGGER start enable register.
3. This module sequentially processes the command list at the storage address of the command list (command list address) when the TRG_PRI1 value changes to "1".

Figure 1-60. Conceptual Image of CMDSEQ Start



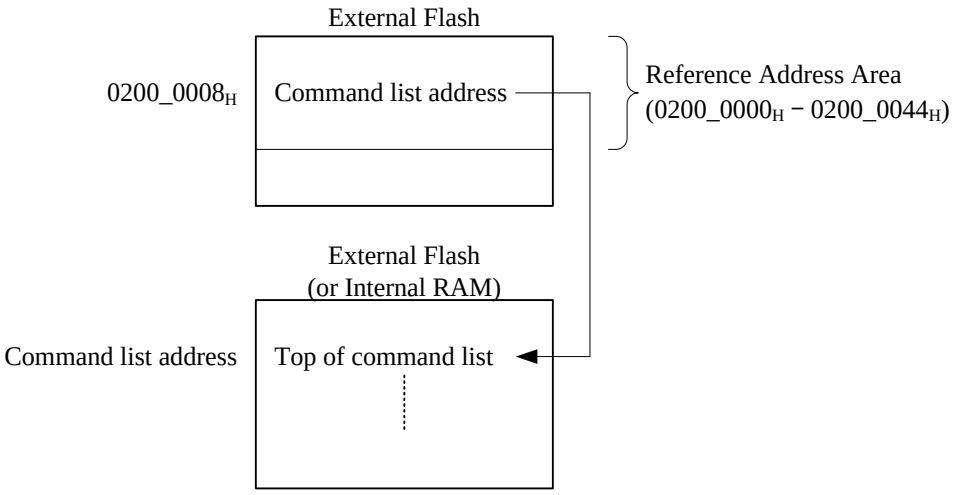
4. After the end of command list processing, status information appears in the status register. If the status enable register is enabled, the corresponding status register bit information is output as an interrupt signal to the INTST register.

Start by Priority3

The procedure for a trigger start caused by a Priority3 signal (TRG_PRI2) is as follows:

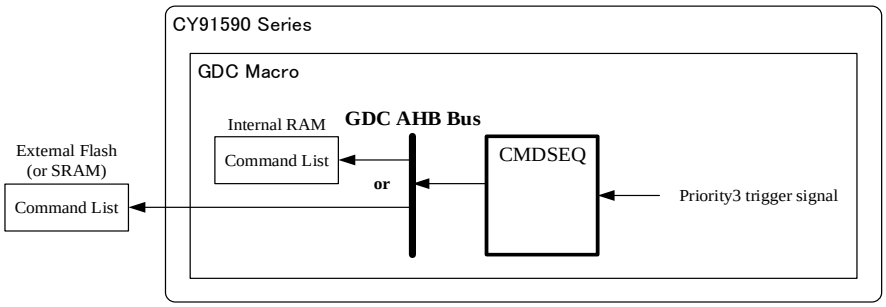
1. Write both the value of the storage address for the command list (command list address) and the command list beginning at the command list address in advance in 0200_0008$_H$.

Figure 1-61. Conceptual Image of Data Storage



2. Set "1" in the PRI3ET bit in the TRIGGER start enable register.
   To output interrupt signals from this module, enable the status enable register.
3. This module sequentially processes the command list at the storage address of the command list (command list address) when the Priority3 signal (TRG_PRI2) changes to "1".

Figure 1-62. Conceptual Image of CMDSEQ Start



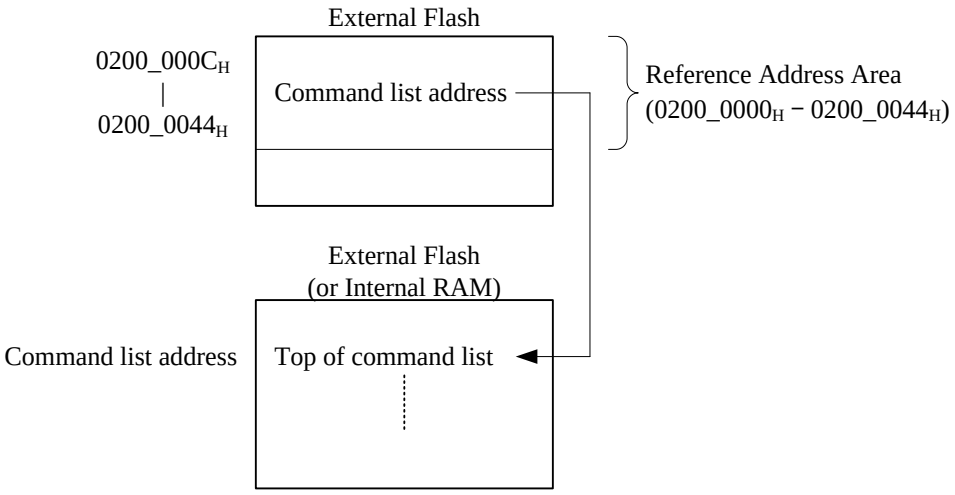
4. After the end of command list processing, status information appears in the status register. If the status enable register is enabled, the corresponding status register bit information is output as an interrupt signal to the INTST register.

Start by Priority4

The procedure for a trigger start caused by a Priority4 signal (TRIGGER0 to TRIGGER14) is as follows:

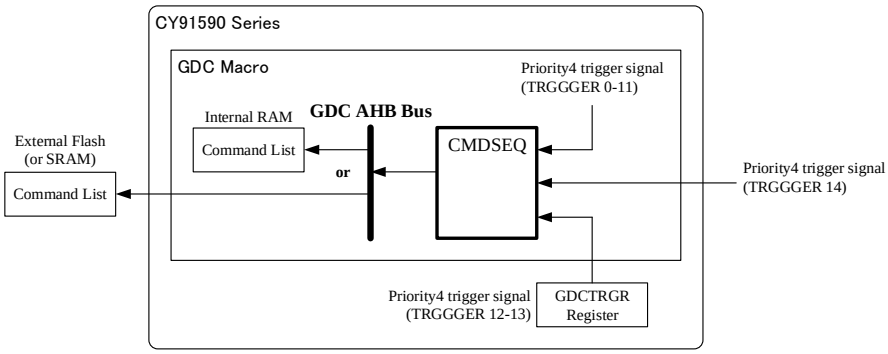
1. Write both the value of the storage address for the command list (command list address) and the command list beginning at the command list address in advance in 0200_000C$_H$ to 0200_0044$_H$.

Figure 1-63. Conceptual Image of Data Storage



2. Set the TSET bit in the TRIGGER start enable register.
   To output interrupt signals from this module, enable the status enable register.
3. This module sequentially processes the command list at the storage address of the command list (command list address) when the signal selected by the TSET bit in the TRIGGER start enable register changes to "1".

Figure 1-64. Conceptual Image of CMDSEQ Start



4. After the end of command list processing, status information appears in the status register. If the status enable register is enabled, the corresponding status register bit information is output as an interrupt signal to the INTST register.

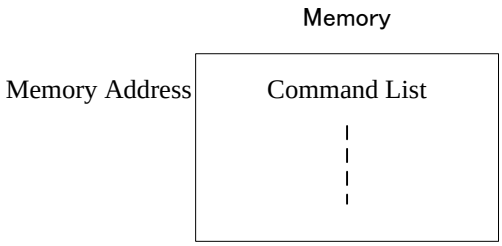
Note: During a trigger start caused by TRIGGER0 to TRIGGER14, the TSET bit value in the TRIGGER start enable register can be changed only after output of an interrupt signal indicating trigger start completion.

1.11.7.3 Register Start

The register start procedure is as follows:

1. Store the command list in memory, and set the memory address of the stored command list in the REGISTER start address register.
   To output interrupt signals from this module, enable the status enable register.

Figure 1-65. Conceptual Image of Memory



2. Set "1" in the start register.
3. This module sequentially processes the command list at the set address in the REGISTER start address register when the start register changes to "1".
4. After the end of command list processing, status information appears in the status register. If the status enable register is enabled, the corresponding status register bit information is output as an interrupt signal to the INTST register.

1.12 Run-Length Decompression (RLD)

This chapter describes the Run-Length Decompression (RLD) unit of the GDC macro.

1.12.1 Overview

Run-Length Decompression (RLD) decompresses run-length compressed data on pure hardware that has no interaction with the processor. Run-length encoding of pictograms, corporate images (logos), and other simple graphic content is highly useful for saving bandwidth when using an external bus system, especially during setup.

1.12.2 Features

- Run-length compression formats supported
- 1, 2, 4, 8, 16, 24, and 32 bits per pixel supported

1.12.3 Restrictions

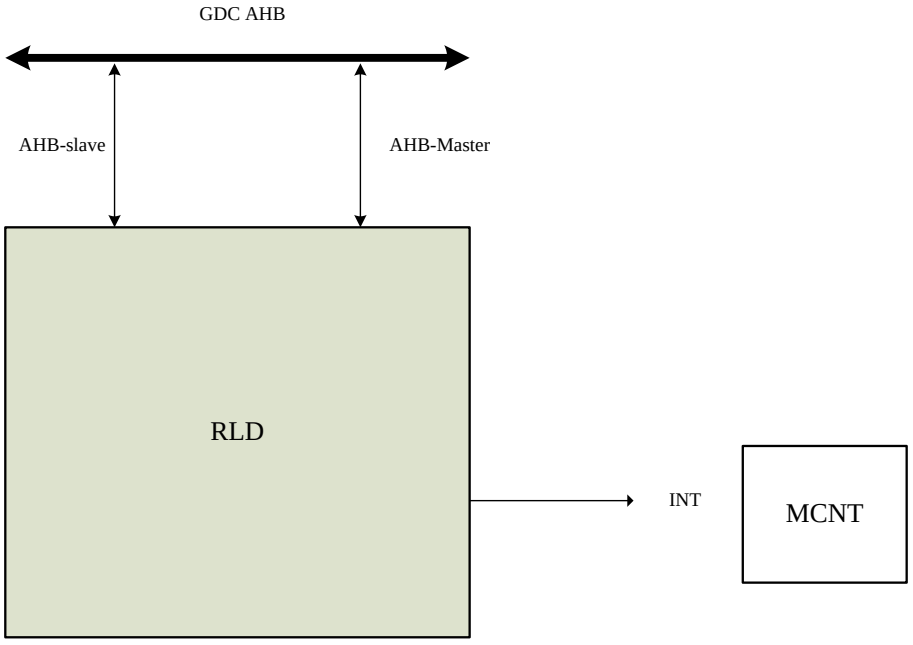
- It is necessary to specify VRAM or CMDRAM as the destination area of decompressed data transfer. The RLD operation is unpredictable if another area is specified.

1.12.4 Configuration

1.12.4.1 Block Diagram

Figure 1-66 is a block diagram of RLD.

Figure 1-66. RLD block diagram



1.12.5 Registers

1.12.5.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (0040_0000H) is added when seen from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field in a register is shown.
  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.12.5.2 Register List

Address	Register Name	Description
$01FB_5000_H$	SWReset	Software reset register
$01FB_5004_H$	RldCfg	Configuration register for the entire module
$01FB_5008_H$	StrideCfg0	Configuration register for general strides
$01FB_500C_H$	StrideCfg1	Configuration register for line and stride lengths
$01FB_5010_H$	BYTECNT	Configuration register for the target number of decompressed bytes
$01FB_5014_H$	OFIFO	Output FIFO control register
$01FB_5018_H$	DestAddress	Configuration register for the master transfer destination address
$01FB_501C_H$	AHBMCtrl	Configuration control register for master transfer
$01FB_5020_H$	RLDCtrl	Register for overall module control
$01FB_5024_H$	IEN	Interrupt enable register
$01FB_5028_H$	ISTS	Interrupt status register
$01FB_502C_H$	Status	Status register
$01FB_5030_H$	SAHBData	Configuration register for input data
$01FB_5034_H$	TransferCount	Register for counting master transfer bytes
$01FB_5038_H$	CurAddress	Register for checking the current master transfer address

1.12.5.3 Register Details

SWReset

This register is a software reset register.

Address	$01FB_5000_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W																
Initial																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															SW Reset
R/W																R/W
Initial																0_B

Bit Field		Explanation
No.	**Name**	
0	SWReset	This bit is used to suspend run-length processing. When "1" is written in this field, a software reset is issued (to initialize the run-length data). **Note:** To release the software reset, write "0" again in this field. Once this reset comes into effect, the run-length processing data is no longer guaranteed. Therefore, after releasing the reset, configure the registers again.

RldCfg

This register is a configuration register for the entire module.

Address	01FB_5004$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W																
Initial																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-							Align	-					BPP[3:0]		
R/W								R/W						R/W		
Initial								0_B						0_H		

Bit Field		**Explanation**
No.	**Name**	
2-0	BPP[3:0]	Bits per pixel 0_H: 1 1_H: 2 2_H: 4 3_H: 8 4_H: 16 5_H: 24 6_H: 32 7_H: Reserved
8	Align	Output data format 0_B: Bit alignment output 1_B: Word (32-bit) alignment output

StrideCfg0

This register is a configuration register for general strides.

Address	01FB_5008$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W																
Initial																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															StrideEN
R/W																R/W
Initial																0_B

Bit Field		Explanation
No.	Name	
0	StrideEn	Specifies whether to enable output stride alignment. 0_B: Disable output stride alignment (disable the StrideCfg1 register). 1_B: Enable output stride alignment.

StrideCfg1

This register is a configuration register for line and stride lengths.

Address	01FB_500C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	Stride[15:0]															
R/W	R/W															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-		LineLength[13:0]													
R/W			R/W													
Initial			0000_H													

Bit Field		Explanation
No.	Name	
13-0	LineLength [13:0]	Number of bytes per line – 1
31-16	Stride[15:0]	Number of bytes in the stride – 1

BYTECNT

This register is a configuration register for the target number of decompressed bytes.

Address	$01FB_5010_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	ByteCnt[31:16]															
R/W	R/W															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ByteCnt[15:0]															
R/W	R/W															
Initial	0000_H															

Bit Field		Explanation
No.	Name	
31-0	ByteCnt [31:0]	Target number of decompressed bytes

OFIFO

This register is the output FIFO control register.

Address	$01FB_5014_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W																
Initial																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-												WriteThreshold[3:0]			
R/W													R/W			
Initial													0_H			

Bit Field		Explanation
No.	Name	
3-0	WriteThreshold [3:0]	Transfer begins after data of the specified value + 4 bytes is accumulated. Note: If WriteThreshold=0_H is set, transfer begins after 8 bytes of data are accumulated.

DestAddress

This register is a configuration register for the master transfer destination address.

Address	$01FB_5018_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AHBMDA[31:16]															
R/W	R/W															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AHBMDA[15:0]															
R/W	R/W															
Initial	0000_H															

Bit Field		**Explanation**
No.	**Name**	
31-0	AHBMDA [31:0]	Destination address for starting master transfer **Note:** The value assigned to AHBMDA must be an address within the GDC macro.

AHBMCtrl

This register is a configuration control register for master transfer.

Address	01FB_501C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W																
Initial																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-						AHBM Transfer Width[1:0]		-							AHBM Fixed Dest
R/W							R/W									R/W
Initial							0_H									0_B

Bit Field		**Explanation**
No.	**Name**	
0	AHBM Fixed Dest	0_B: Increment the destination address. 1_B: Fix the destination address.
9-8	AHBM Transfer Width[1:0]	0_H: 1 byte 1_H: 2 bytes 2_H: 4 bytes 3_H: Reserved

RLDCtrl

This register is a register for overall module control.

Address	$01FB_5020_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W																
Initial																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															Accept Data
R/W																R/W
Initial																0_B

Bit Field		Explanation
No.	Name	
0	AcceptData	Enables acceptance of compressed data. This field is reset by hardware after processing is completed.

IEN

This register is an interrupt enable register.

Address	01FB_5024															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W																
Initial																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-												IEnIFfull	IEnIFempty	-	IEnComplete
R/W													R/W	R/W	R	R/W
Initial													0_B	0_B	X	0_B

Bit Field		Explanation
No.	**Name**	
0	IEnComplete	0_B: Do not output the IStsComplete bit information in ISTS as an interrupt signal. 1_B: Output the IStsComplete bit information in ISTS as an interrupt signal.
2	IEnIFempty	0_B: Do not output the IStsError bit information in ISTS as an interrupt signal. 1_B: Output the IStsError bit information in ISTS as an interrupt signal.
3	IEnIFfull	0_B: Do not output the IStsError bit information in ISTS as an interrupt signal. 1_B: Output the IStsError bit information in ISTS as an interrupt signal.

ISTS

This register is an interrupt status register.

Address	$01FB_5028_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W																
Initial																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-												ISts IF full	ISts IF empty	-	ISts Complete
R/W													R/W	R/W	R	R/W
Initial													0_B	0_B	X	0_B

Bit Field		Explanation
No.	**Name**	
0	IStsComplete	Displays "1" when the RLD completion condition is satisfied (the target number of bytes is reached). This bit is cleared when "1" is written to it. 0_B: No interrupt 1_B: Interrupt
2	IStsIFempty	Displays "1" when the input FIFO is empty. This bit is cleared when "1" is written to it. 0_B: No interrupt 1_B: Interrupt
3	IStsIFfull	Displays "1" when the input FIFO is full. This bit is cleared when "1" is written to it. 0_B: No interrupt 1_B: Interrupt

Status

This register is a status register.

Address	01FB_502C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W																
Initial																
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-												IFIFO empty	OFIFO full	IFIFO full	Busy
R/W													R	R	R	R
Initial													1_B	0_B	0_B	0_B

Bit Field		Explanation
No.	**Name**	
0	Busy	This bit indicates whether the RLD unit is busy. 0_B: The RLD unit is not busy. 1_B: The RLD unit is busy.
1	IFIFOfull	This bit indicates whether the input FIFO is full. 0_B: The input FIFO is not full. 1_B: The input FIFO is full.
2	OFIFOfull	This bit indicates whether the output FIFO is full. 0_B: The output FIFO is not full. 1_B: The output FIFO is full.
3	IFIFOempty	This bit indicates whether the input FIFO is empty. 0_B: The input FIFO is not empty. 1_B: The input FIFO is empty.

SAHBData

This register is a configuration register for input data.

Address	$01FB_5030_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	InData[31:16]															
R/W	R/W															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	InData[15:0]															
R/W	R/W															
Initial	0000_H															

Bit Field		Explanation
No.	Name	
31-0	InData[31:0]	RLD input data (The data written at this address is latched to IFIFO of RLD.) Access the SAHBData register after configuring various registers.

TransferCount

This register is a register for counting master transfer bytes.

Address	$01FB_5034_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AHBMTransferCount[31:16]															
R/W	R															
Initial	0000_H															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AHBMTransferCount[15:0]															
R/W	R															
Initial	0000_H															

Bit Field		Explanation
No.	Name	
31-0	AHBMTransferCount [31:0]	Number of bytes yet to be transferred during a transaction (The counter is decremented.)

CurAddress

This register is a register for checking the current master transfer address.

Address	$01FB_5038_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AHBMCA[31:16]															
R/W	R															
Initial	X															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AHBMCA[15:0]															
R/W	R															
Initial	X															

Bit Field		Explanation
No.	**Name**	
31-0	AHBMCA [31:0]	Current destination address

1.12.6 Explanation of Operation

1.12.6.1 Data Formats

Input Data Formats

Table 1-22 lists the supported data formats.

Table 1-22. Supported Data Formats

Format [Bits/Pixel]	Mode	Command Byte <MSB LSB>	Color Byte <MSB LSB>
1 bpp	Compressed	<1NNN NNNN>	Color bit <C> (bit-aligned)
	Decompressed	<0NNN NNNN>	(NNN_NNNN +1) bits with color data <C...C> (bit-aligned)
2 bpp	Compressed	<1NNN NNNN>	Color bit <CC> (bit-aligned)
	Decompressed	<0NNN NNNN>	(NNN_NNNN+1) x 2 bits with color data <CC...CC> (bit-aligned)
4 bpp	Compressed	<1NNN NNNN>	Color bits <CCCC> (bit-aligned)
	Decompressed	<0NNN NNNN>	(NNN_NNNN+1) x 4 bits with color data <CCCC...CCCC> (bit-aligned)
8 bpp	Compressed	<1NNN NNNN>	1 color byte <CCCC CCCC>
	Decompressed	<0NNN NNNN>	(NNN_NNNN +1) bytes with color data <CCCC CCCC>
16 bpp[1]	Compressed	<1NNN NNNN>	2 bytes color data (<CCCC CCCC>,<CCCC CCCC>)
	Decompressed	<0NNN NNNN>	(NNN_NNNN +1) x 2 bytes with color data (<CCCC CCCC>,<CCCC CCCC>)
24 bpp	Compressed	<1NNN NNNN>	3 bytes color data (<CCCC CCCC>,<CCCC CCCC>, <CCCC CCCC>)
	Decompressed	<0NNN NNNN>	(NNN_NNNN +1) x 3 bytes with color data (<CCCC CCCC>,<CCCC CCCC>, <CCCC CCCC>)
32 bpp	Compressed	<1NNN NNNN>	4 bytes color data (<CCCC CCCC>,<CCCC CCCC>, <CCCC CCCC>,<CCCC CCCC>)
	Decompressed	<0NNN NNNN>	(NNN_NNNN +1) x 4 bytes with color data (<CCCC CCCC>,<CCCC CCCC>, <CCCC CCCC>,<CCCC CCCC>)

[1]: Used for RGB555 or RGB565

The following statement applies in any mode:

Compressed data "<1NNN NNNN> <pixel>" is decompressed to (<NNN NNNN> + 1) pixels.

Figure 1-67. RGB888 Decompression Data Input Example

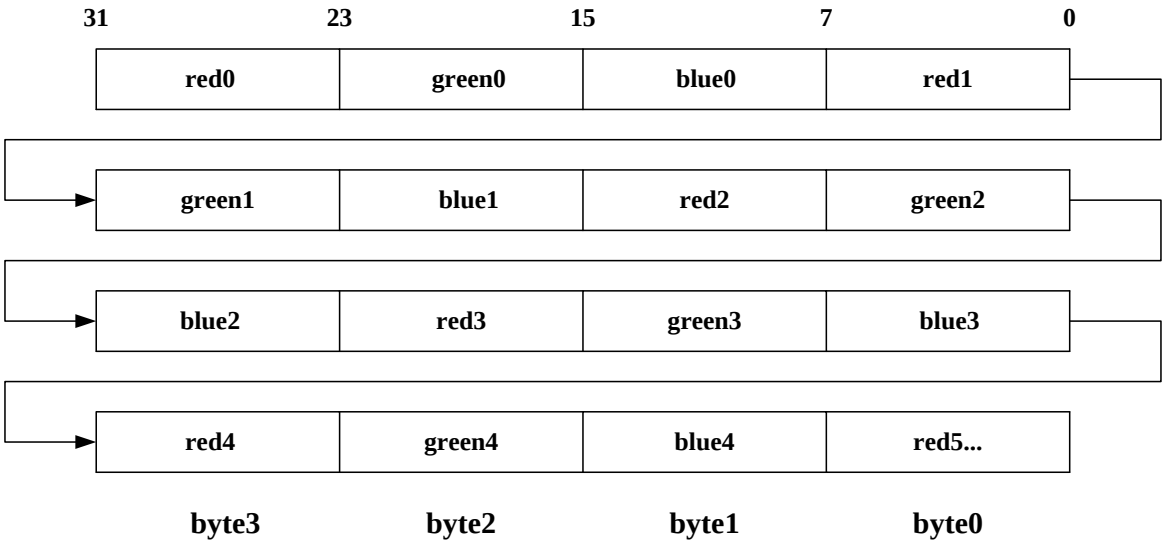
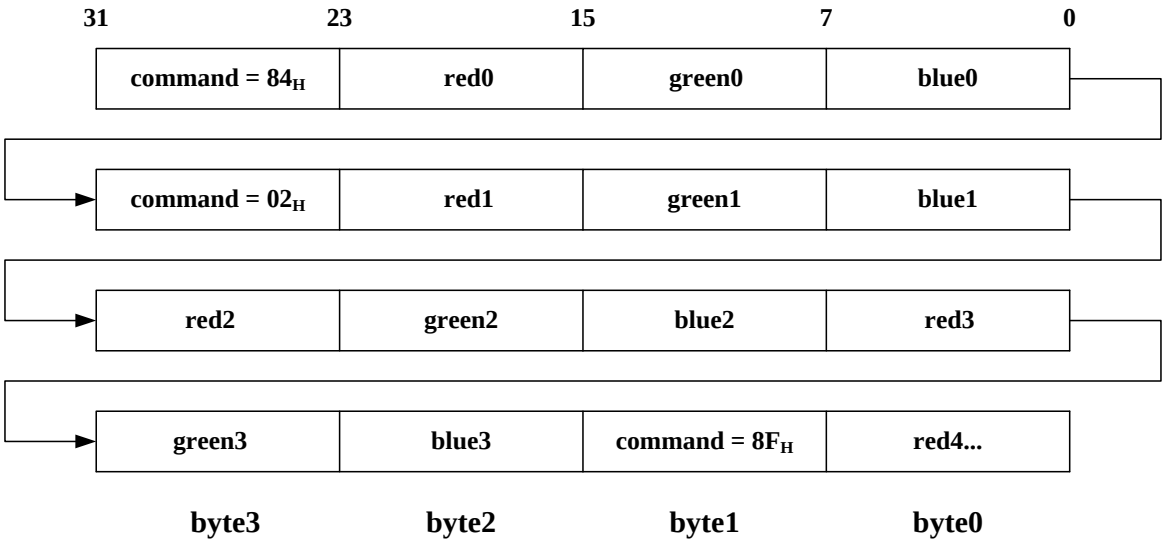


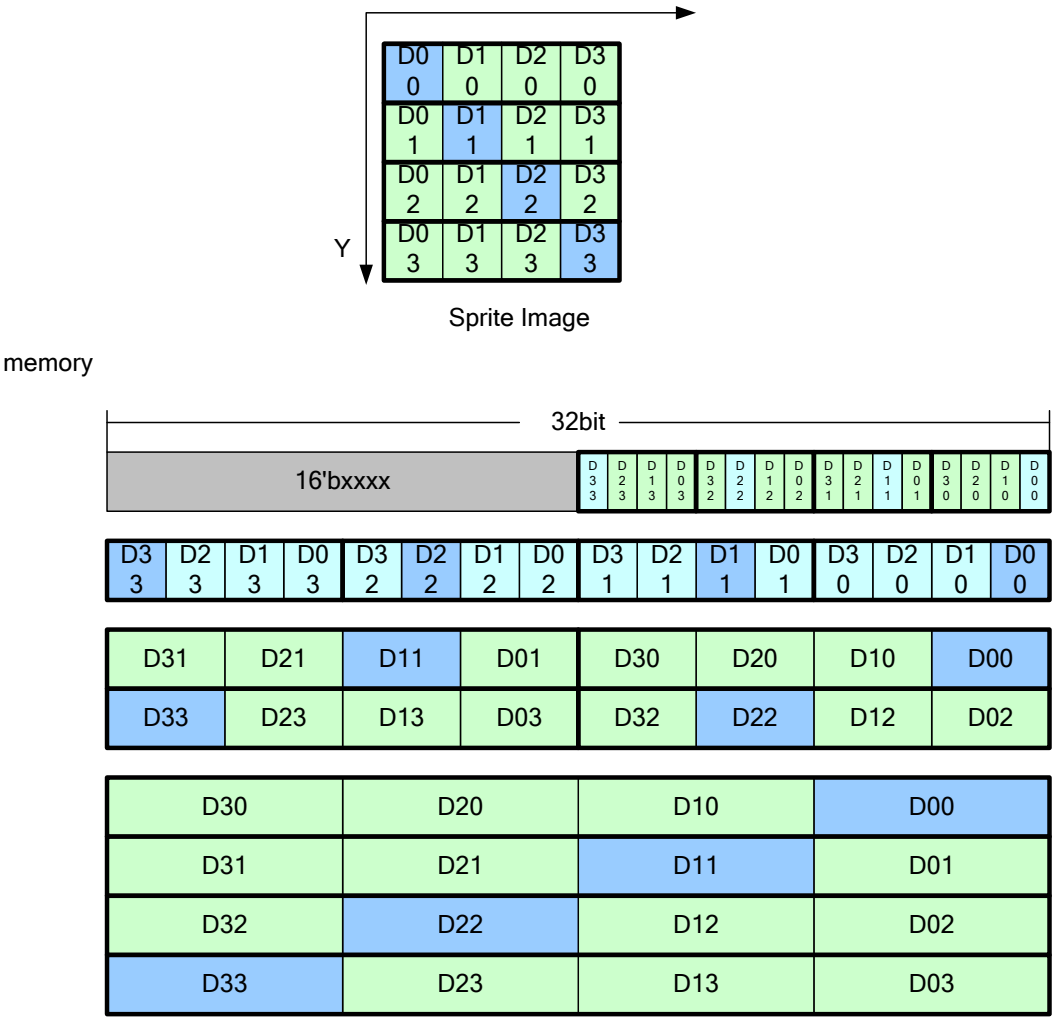
Figure 1-68. RGB888 Compression Data Input Example



Output Data Formats

The output data format depends on the selected BPP (Bits Per Pixel) value. In addition, the hardware supports calculation of memory strides for bit or word alignment.
Output data is in Little Endian.

Figure 1-69. Memory Layout for the Sprite Engine

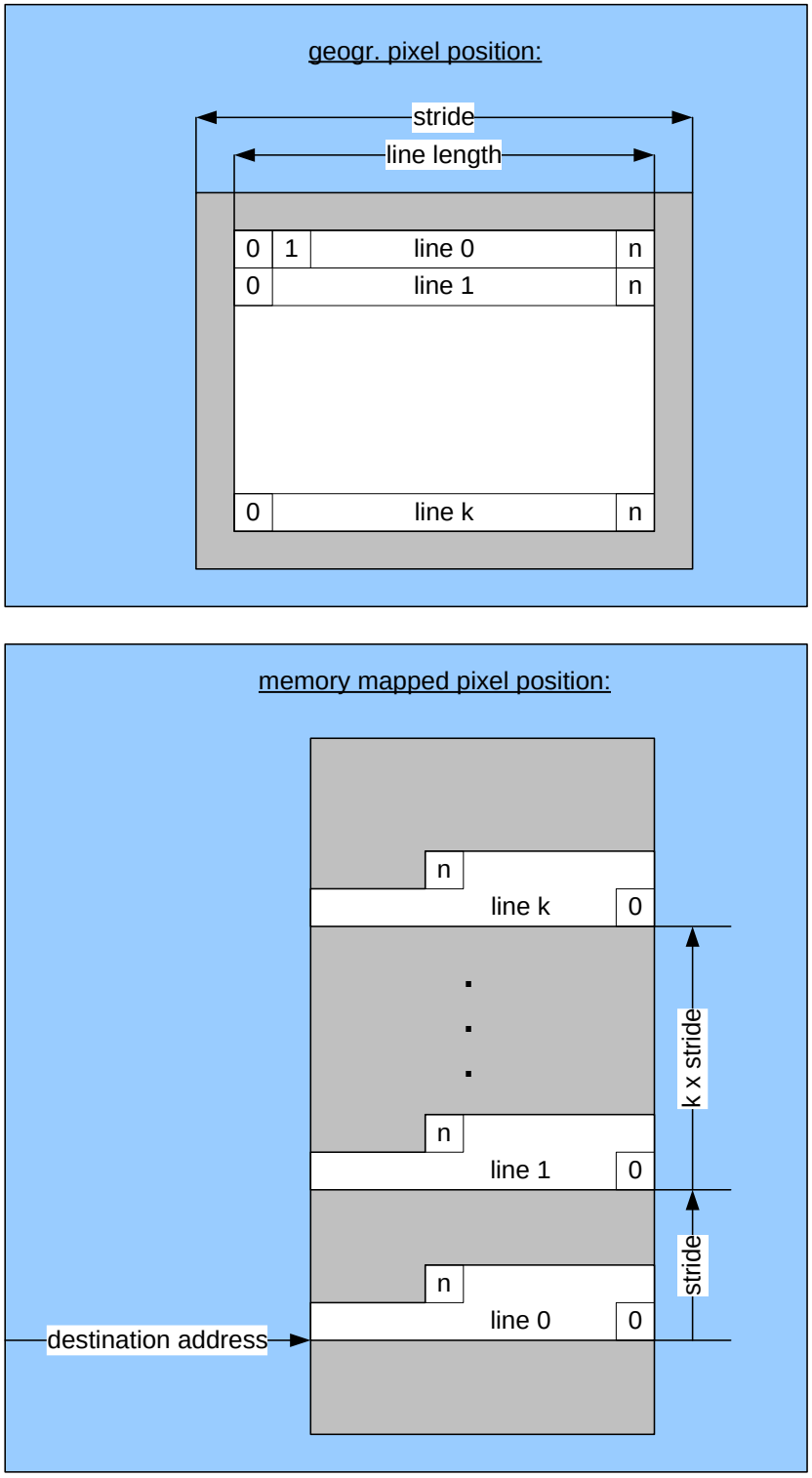


Configure the BYTECNT register (address: 01FB_5010$_H$) to have consistency with the data to be decompressed.

Memory Strides

For details on stride calculation performed by hardware, see Figure 1-70.

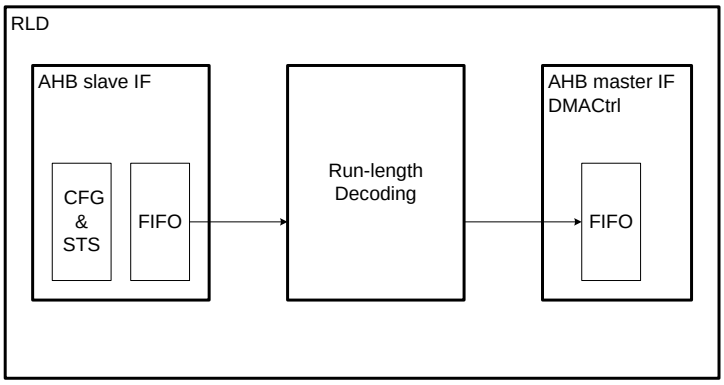
Figure 1-70. Memory Strides



1.12.6.2 Processing Flow

Figure 1-71 shows the RLD processing flow.

Figure 1-71. Processing Flowchart



1.12.6.3 Control Flow

Control Flow Example

- RLD settings
  1. Reset OFIFO and IFIFO.
  2. Set the BPP value.
  3. Set the target number of bytes.
  4. Set the AHB master IF.

     Example: Video memory destination address
  5. Enable RLD.
- Transmission of compressed data to the RLD AHB slave
  - ☐ Case A: The compressed data source is the AHB slave.
    1. Configure DMA transfer.
       a. Configure the source (e.g., flash).
       b. DMA destination setting: RLD
    2. Start DMA transfer.
    3. Generate an interrupt after processing is completed.
  - ☐ Case B: The compressed data source is the AHB master.
    1. Write active data to the RLD AHB slave.
    2. Generate an interrupt after processing is completed.

1.13 Signature Generator (SIG)

This chapter describes the signature generator of the GDC macro.

1.13.1 Overview

The Signature Generator unit (SIG) calculates different types of checksums for input data. The application generates a checksum for pixel stream data for a user-defined evaluation window. (The entire size and position can be programmed.)

The system microcontroller can use such signatures, for example, to determine whether the displayed image is identical (or almost identical) to the original image data. This is necessary for critical safety displays and helps to fulfill the requirements of safety standards (e.g., Automotive Safety Integrity Level (ASIL)).

1.13.2 Features

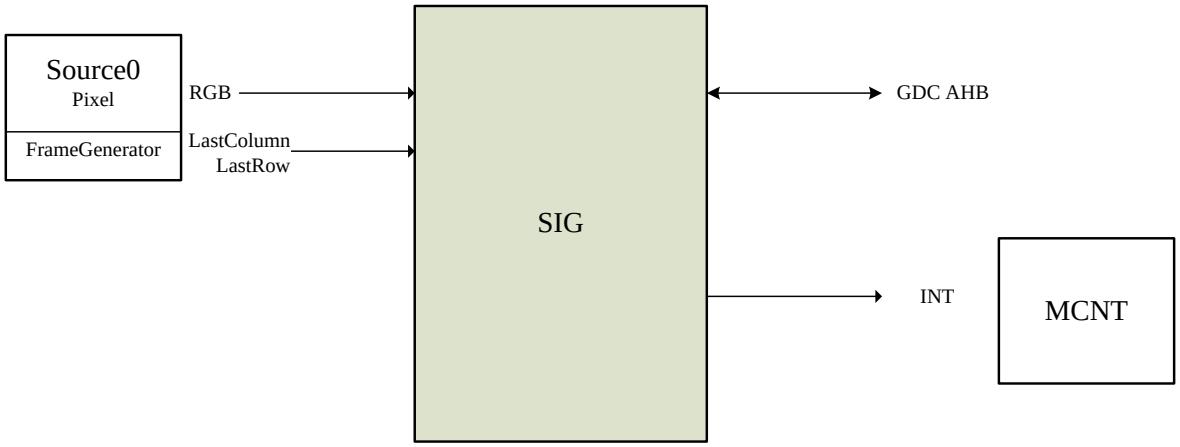
- Generation of 2 different image signatures for each color channel
  - ☐ Summation of color values
  - ☐ CRC-32 for color values
- Programmable evaluation window position and size
- Programmable evaluation window mask
- Automatic monitoring using reference signature registers
- Interrupt generation
- Programmable image source
- Self-restoring error counter

1.13.3 Configuration

1.13.3.1 Block Diagram

Figure 1-72 shows a block diagram of SIG.

Figure 1-72. Block Position in the Whole LSI



1.13.4 Registers

1.13.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (**0040_0000**$_H$) is added for access registers from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

Unused register fields are shown with a gray background.

If there is no description in particular, the bit vector is an unsigned integer.

1.13.4.2 Register List

Base address

The base address of this module is 01FB_0000$_H$.

Address	Register Name	Description
01FB_0000$_H$	SigSWreset	SIG software reset
01FB_0004$_H$	SigCtrl	SIG general configuration register
01FB_0008$_H$	MaskHorizontalUpperLeft	Horizontal mask point (upper-left) setting
01FB_000C$_H$	MaskHorizontalLowerRight	Horizontal mask point (lower-right) setting
01FB_0010$_H$	MaskVerticalUpperLeft	Vertical mask point (upper-left) setting
01FB_0014$_H$	MaskVerticalLowerRight	Vertical mask point (lower-right) setting
01FB_0018$_H$	HorizontalUpperLeftW0	Horizontal evaluation window (upper-left) setting
01FB_001C$_H$	HorizontalLowerRightW0	Horizontal evaluation window (lower-right) setting
01FB_0020$_H$	VerticalUpperLeftW0	Vertical evaluation window (upper-left) setting
01FB_0024$_H$	VerticalLowerRightW0	Vertical evaluation window (lower-right) setting
01FB_0028$_H$	SignAReferenceRW0	Reference value setting on Red in Signature A
01FB_002C$_H$	SignAReferenceGW0	Reference value setting on Green in Signature A
01FB_0030$_H$	SignAReferenceBW0	Reference value setting on Blue in Signature A
01FB_0034$_H$	SignBReferenceRW0	Reference value setting on Red in Signature B
01FB_0038$_H$	SignBReferenceGW0	Reference value setting on Green in Signature B
01FB_003C$_H$	SignBReferenceBW0	Reference value setting on Blue in Signature B
01FB_0040$_H$	ThrBRW0	Threshold setting on Red in Signature B
01FB_0044$_H$	ThrBGW0	Threshold setting on Green in Signature B
01FB_0048$_H$	ThrBBW0	Threshold setting on Blue in Signature B
01FB_004C$_H$	ErrorThreshold	Error counter threshold
01FB_0050$_H$	CtrlCfgW0	Evaluation window control/configuration register
01FB_0054$_H$	TriggerW0	Trigger register
01FB_0058$_H$	IENW0	Interrupt enable register
01FB_005C$_H$	InterruptStatusW0	Interrupt status register
01FB_0060$_H$	StatusW0	Status register
01FB_0064$_H$	Signature error	Amount of video frames with signature errors
01FB_0068$_H$	SignatureARW0	Signature A calculation result on Red
01FB_006C$_H$	SignatureAGW0	Signature A calculation result on Green
01FB_0070$_H$	SignatureABW0	Signature A calculation result on Blue
01FB_0074$_H$	SignatureBRW0	Signature B calculation result on Red
01FB_0078$_H$	SignatureBGW0	Signature B calculation result on Green
01FB_007C$_H$	SignatureBBW0	Signature B calculation result on Blue

1.13.4.3 Register Details

SigSWreset

Register Address	$01FB_0000_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																																SWRes
R/W																																RW
Reset Value																																0

SIG software reset
[bit0] SWReset
Software reset

SigCtrl

Register Address	$01FB_0004_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name															Reserved								Vmask_mode							Hmask_mode	
R/W															RW								RW							RW	
Reset Value															0	0							0	0						0	0

SIG general configuration register
[bit17, bit16] Reserved
Restrictions: Do not output a value other than "0" in bit17 and bit16.

[bit9, bit8] Vmask_mode
00 = No mask
01 = Mask inside vertical coordinates
10 = Mask outside vertical coordinates
11 = Reserved

[bit1, bit0] Hmask_mode
00 = No mask
01 = Mask inside horizontal coordinates
10 = Mask outside horizontal coordinates
11 = Reserved

MaskHorizontalUpperLeft

Register Address	01FB_0008$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																					MaskHorizontalUpperLeft											
R/W																					RW											
Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0

Register for horizontal mask point (upper-left) setting
[bit11 to bit0] MaskHorizontalUpperLeft
These bits set a horizontal mask point (upper-left).

MaskHorizontalLowerRight

Register Address	01FB_000C$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																					MaskHorizontalLowerRight											
R/W																					RW											
Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0

Register for horizontal mask point (lower-right) setting
[bit11 to bit0] MaskHorizontalLowerRight
These bits set a horizontal mask point (lower-right).

MaskVerticalUpperLeft

Register Address	01FB_0010$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																					MaskVerticalUpperLeft											
R/W																					RW											
Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0

Register for vertical mask point (upper-left) setting
[bit11 to bit0] MaskVerticalUpperLeft
These bits set a vertical mask point (upper-left).

MaskVerticalLowerRight

Register Address	01FB_0014$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																					MaskVerticalLowerRight										
R/W																					RW											
Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0

Register for vertical mask point (lower-right) setting
[bit11 to bit0] MaskVerticalLowerRight
These bits set a vertical mask point (lower-right).

HorizontalUpperLeftW0

Register Address	01FB_0018$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																					HorizontalUpperLeftW0										
R/W																					RW											
Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0

Register for horizontal evaluation window (upper-left) setting
[bit11 to bit0] HorizontalUpperLeftW0
These bits set a horizontal evaluation window (upper-left).

HorizontalLowerRightW0

Register Address	01FB_001C$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																					HorizontalLowerRightW0										
R/W																					RW											
Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0

Register for horizontal evaluation window (lower-right) setting
[bit11 to bit0] HorizontalLowerRightW0
These bits set a horizontal evaluation window (lower-right).

VerticalUpperLeftW0

Register Address	$01FB_0020_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																					VerticalUpperLeftW0											
R/W																					RW											
Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0

Register for vertical evaluation window (upper-left) setting
[bit11 to bit0] VerticalUpperLeftW0
These bits set a vertical evaluation window (upper-left).

VerticalLowerRightW0

Register Address	$01FB_0024_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																					VerticalLowerRightW0											
R/W																					RW											
Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0

Register for vertical evaluation window (lower-right) setting
[bit11 to bit0] VerticalLowerRightW0
These bits set a vertical evaluation window (lower-right).

SignAReferenceRW0

Register Address	$01FB_0028_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignAReferenceRW0																															
R/W	RW																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for reference value setting on Red in Signature A mode
[bit31 to bit0] SignAReferenceRW0
These bits set the reference value on Red in Signature A mode.

SignAReferenceGW0

Register Address	01FB_002C$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignAReferenceGW0																															
R/W	RW																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for reference value setting on Green in Signature A mode

[bit31 to bit0] SignAReferenceGW0

These bits set the reference value on Green in Signature A mode.

SignAReferenceBW0

Register Address	01FB_0030$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignAReferenceBW0																															
R/W	RW																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for reference value setting on Blue in Signature A mode

[bit31 to bit0] SignAReferenceBW0

These bits set the reference value on Blue in Signature A mode.

SignBReferenceRW0

Register Address	01FB_0034$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignBReferenceRW0																															
R/W	RW																															
Reset Value	0_H																															

Register for reference value setting on Red in Signature B mode

[bit31 to bit0] SignBReferenceRW0

These bits set the reference value on Red in Signature B mode.

SignBReferenceGW0

Register Address	01FB_0038$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignBReferenceGW0																															
R/W	RW																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for reference value setting on Green in Signature B mode
[bit31 to bit0] SignBReferenceGW0
These bits set the reference value on Green in Signature B mode.

SignBReferenceBW0

Register Address	01FB_003C$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignBReferenceBW0																															
R/W	RW																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for reference value setting on Blue in Signature B mode
[bit31 to bit0] SignBReferenceBW0
These bits set the reference value on Blue in Signature B mode.

ThrBRW0

Register Address	01FB_0040$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	ThrBRW0																															
R/W	RW																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for threshold setting on Red in Signature B mode
[bit31 to bit0] ThrBRW0
These bits set the Red threshold in Signature B mode.

ThrBGW0

Register Address	$01FB_0044_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	ThrBGW0																															
R/W	RW																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for threshold setting on Green in Signature B mode
[bit31 to bit0] ThrBGW0
These bits set the Green threshold in Signature B mode.

ThrBBW0

Register Address	$01FB_0048_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	ThrBBW0																															
R/W	RW																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for threshold setting on Blue in Signature B mode
[bit31 to bit0] ThrBBW0
These bits set the Blue threshold in Signature B mode.

ErrorThreshold

Register Address	01FB_004C$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name									ErrThresReset																ErrThres							
R/W									RW																RW							
Reset Value									0	0	0	0	1	0	0	0									0	0	0	0	0	0	0	1

Error counter threshold value

[bit23 to bit16] ErrThresReset

Number of consecutive error-free video frames, as the cause of an error_count reset

0x00 = No reset

0x01 = 1

:

0xFF = 255

[bit7 to bit0] ErrThres

Error counter threshold value

0x00 = 256

0x01 = 1

:

0xFF = 255

If error_counter≥"ErrThres," an interrupt is generated.

CtrlCfgW0

Register Address	01FB_0050$_H$																													
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																EnCoordW0								EnSignB							EnSignA
R/W																RW								RW							RW
Reset Value																0								0							0

Evaluation window control/configuration register

[bit16] EnCoordW0

This bit enables window 0 coordinates.

[bit8] EnSignB

This bit enables Signature calculation B.

[bit0] EnSignA

This bit enables Signature calculation A.

TriggerW0

Register Address	$01FB_0054_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																							TrigMode									Trigger
R/W																							RW									W
Reset Value																							0	0								0

Trigger register

[bit9, bit8] TrigMode

00 = Start of 1 trigger generation, and cancellation of the cyclic trigger

01 = Start of cyclic trigger generation

10 = Reserved

11 = Reserved

[bit0] Trigger

This bit generates a trigger for signature generation.

For details on the trigger mode, see TrigMode.

IENW0

Register Address	$01FB_0058_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																														IEnResVal	IEnCfgCop	IEnDiff
R/W																														RW	RW	RW
Reset Value																														0	0	0

Evaluation window control/configuration register

[bit2] IEnResVal

This bit enables an interrupt.

[bit1] IEnCfgCop

This bit enables an interrupt.

[bit0] IEnDiff

This bit enables a difference detection interrupt.

InterruptStatusW0

Register Address	01FB_005C$_H$																														
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																														IStsResVal	IStsCfgCop	IStsDiff
R/W																														RW	RW	RW
Reset Value																														0	0	0

Interrupt status register

[bit2] IStsResVal

Interrupt status flag

"1" indicates that the corresponding interrupt status occurred. (This applies even if disabled is the interrupt setting.)

Writing "1" clears the flag.

Condition: The Result register is valid.

[bit1] IStsCfgCop

Interrupt status flag

"1" indicates that the corresponding interrupt status occurred. (This applies even if disabled is the interrupt setting.)

Writing "1" clears the flag.

Condition: The configuration registers are copied to the shadow registers.

[bit0] IStsDiff

Interrupt status flag

"1" indicates that the corresponding interrupt status occurred. (This applies even if disabled is the interrupt setting.)

Writing "1" clears the flag.

Condition: 1 of the active signature results is different from the corresponding reference value.

StatusW0

Register Address	01FB_0060$_H$																														
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name														Diff_B_B	Diff_B_G	Diff_B_R						Diff_A_B	Diff_A_G	Diff_A_R	Reserved2						Active	Pending
R/W														R	R	R						R	R	R	RW						R	R
Reset Value														0	0	0						0	0	0	0	0	0				0	0

Status register

[bit18] Diff_B_B

Signature B: A valid B result is compared with the reference value.

0 = Equal

1 = Different

[bit17] Diff_B_G

Signature B: A valid G result is compared with the reference value.

0 = Equal

1 = Different

[bit16] Diff_B_R

Signature B: A valid R result is compared with the reference value.

0 = Equal

1 = Different

[bit10] Diff_A_B

Signature A: A valid B result is compared with the reference value.

0 = Equal

1 = Different

[bit9] Diff_A_G

Signature A: A valid G result is compared with the reference value.

0 = Equal

1 = Different

[bit8] Diff_A_R

Signature A: A valid R result is compared with the reference value.

0 = Equal

1 = Different

[bit7 to bit5] Reserved 2

[bit1] Active

The generation task is active.

[bit0] Pending

The generation task is pending.

Signature_error

Register Address	01FB_0064																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name																					Sig_error_count											
R/W																					R											
Reset Value																					0	0	0	0	0	0	0	0	0	0	0	0

Number of video frames with signature errors
[bit11 to bit0] Sig_error_count
Number of video frames with signature errors
All triggers (see TriggerW0) reset Signature_error to "0".

SignatureARW0

Register Address	01FB_0068$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignatureARW0																															
R/W	R																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for the Signature A calculation result on Red
[bit31 to bit0] SignatureARW0
Calculation result of Signature A:Red

SignatureAGW0

Register Address	$01FB_006C_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignatureAGW0																															
R/W	R																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for the Signature A calculation result on Green
[bit31 to bit0] SignatureAGW0
Calculation result of Signature A:Green

SignatureABW0

Register Address	$01FB_0070_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignatureABW0																															
R/W	R																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for the Signature A calculation result on Blue
[bit31 to bit0] SignatureABW0
Calculation result of Signature A:Blue

SignatureBRW0

Register Address	$01FB_0074_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignatureBRW0																															
R/W	R																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for the Signature B calculation result on Red
[bit31 to bit0] SignatureBRW0
Calculation result of Signature B:Red

SignatureBGW0

Register Address	01FB_0078$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignatureBGW0																															
R/W	R																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for the Signature B calculation result on Green
[bit31 to bit0] SignatureBGW0
Calculation result of Signature B:Green

SignatureBBW0

Register Address	01FB_007C$_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Field Name	SignatureBBW0																															
R/W	R																															
Reset Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Register for the Signature B calculation result on Blue
[bit31 to bit0] SignatureBBW0
Calculation result of Signature B:Blue

1.13.5 Explanation of Operation

1.13.5.1 Signature A: CRC-32 Signature

For each color channel in the evaluation window area, a CRC-32 checksum is generated.

The applied algorithm is the same as that in the Ethernet Standards (CRC-32-IEEE 802.3) except for Step e, the last step. This means that the resulting bit sequence is not complemented at the end.

1.13.5.2 Signature B: Summation Signature

The sum of the pixel color values for each color channel (R, G, B) in the evaluation window area is calculated.

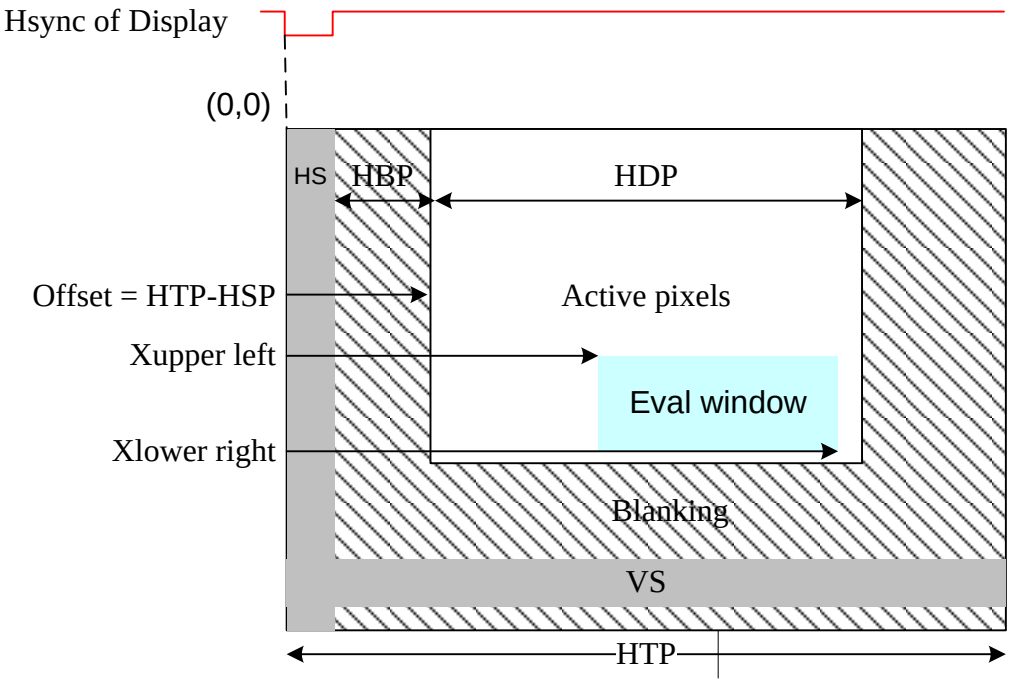
1.13.5.3 Programmable Evaluation Window (Position and Size)

The position and size (upper-left and lower-right coordinates) of the evaluation window are programmable.

The window coordinates follow the coordinate system shown in Figure 1-73.

The coordinate reference point is determined by the falling edge of a horizontal sync signal of the video display controller module and the first active pixel line.

Figure 1-73. Coordinate System



1.13.5.4 Programmable Evaluation Window Mask

A programmable mask window allows the exclusion of incoming pixels for the signature calculation.

This is applied in the coordinate system shown in Figure 1-73.

1.13.5.5 Automatic Monitoring and Interrupt

A set of reference signature registers enables the monitoring of the calculated signatures.

An interrupt can be generated upon the detection of a difference between the calculated signature and the reference value.

For Signature B, in order to limit the interrupt load for the microcontroller, the filtering threshold is different.

1.13.5.6 Self-restoring Error Counter

A counter is incremented if 1 of the active signature results differs from the corresponding reference value. If the counter reaches the programmable error counter threshold, an interrupt may be generated. If the programmable number of consecutive video frames with correct signature values is received, the programmable error counter is reset to zero.

1.13.5.7 Interrupts for Control Flow

An interrupt can be generated for both the start and end of a signature calculation.

The start interrupt (CfgCop) indicates that the configuration parameters (for example, window coordinates) have been copied from the Shadow registers and are now active for the current signature calculation. This enables the next configuration parameters to be loaded into the Shadow registers without impeding the current calculation.

The end interrupt (ResVal) indicates that a signature calculation has been completed and the result data can be read from the Result shadow registers.

These interrupts help control signature calculations for every incoming frame with different evaluation window coordinates.

1.13.5.8 Restrictions

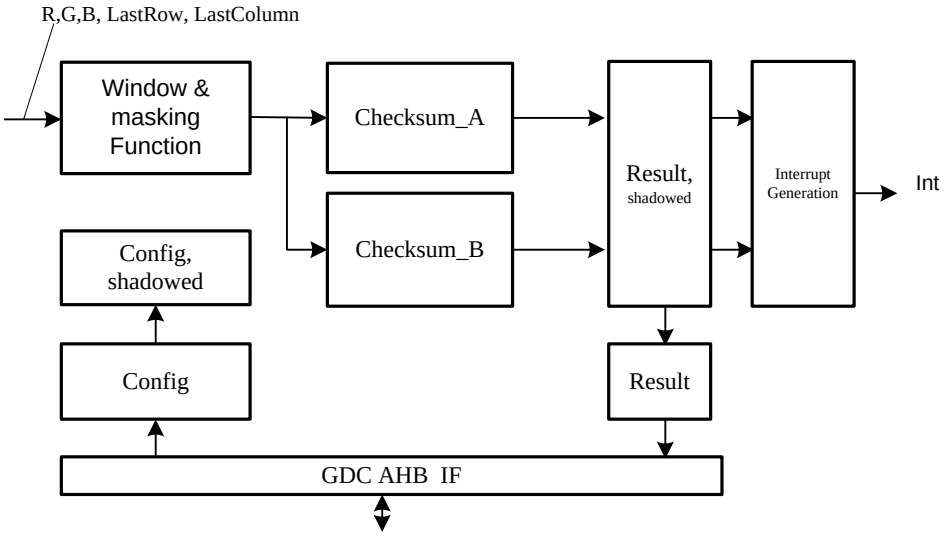
- The maximum resolution for image sources and windows is 4096 x 4096 pixels.
- The evaluation window position must be completely inside the image source frame.
- The source selection must be configured before the evaluation window coordinates and may not be changed during operation.
- There is no support for interlaced sources.
- Also for de-interlaced sources, no special processing is supported. This means that reference values must be calculated by software by taking the right de-interlace algorithm for such material into consideration.

1.13.5.9 Processing Mode

Processing Flow

Figure 1-74 shows the SIG processing flowchart.

Figure 1-74. Processing Flowchart



Algorithm Processing

See 1.13.2 Features.

Checksum generation is possible for each incoming pixel frame. When checksum generation is triggered after each incoming pixel frame, a set of signature checksum results is enabled.

1.13.5.10 Control Flow

Signature Generation with Every Incoming Frame

- General configuration phase (Most registers not shadowed):
  1. Enable mask mode.
  2. Write mask window coordinates.
  3. Enable signature types.
  4. Enable interrupts.
- Configuration phase for calculation 0:
  1. Write Window 0 coordinates.
  2. Set trigger mode for single-trigger generation.
  3. Generate 1 trigger by writing "1" to the trigger field.
- Wait for an interrupt, or poll IstsCfgCop and IstsResVal.
  - ☐ IstsCfgCop:
    1. Configuration phase for calculation n
    2. Write Window n coordinates.
    3. Set trigger mode for single-trigger generation.
    4. Generate 1 trigger by writing "1" to the trigger field.
  - ☐ IstsResVal:
    1. Read Signature A (B) to Result registers.
    2. Process results.

Cyclic Signature Generation with Every Incoming Frame

Cyclic monitoring of 1 window

- General configuration phase (Most registers not shadowed):
  1. Enable mask mode.
  2. Write mask window coordinates.
  3. Enable signature types.
  4. Enable interrupts.
- Configuration phase for calculation 0:
  1. Write Window 0 coordinates.
  2. Set the trigger mode to cyclic trigger mode.
  3. Generate a cyclic trigger by writing "1" to the trigger field.
- Wait for an interrupt, or poll IstsResVal.
  - IStsResVal:
    1. Read Signature A (B) to Result registers.
    2. Process results.

Cyclic Signature Generation with Every Incoming Frame, Limiting Read Access

Cyclic trigger for 1 window, with interrupt monitoring

- General configuration phase (Most registers not shadowed):
  1. Enable mask mode.
  2. Write mask window coordinates.
  3. Enable signature types.
  4. Enable interrupts.
- Configuration phase for calculation 0
  1. Write Window 0 coordinates.
  2. Write reference values for Signature A. (Write the reference value for Signature B, and set the threshold for B.)
  3. Set the trigger mode to cyclic trigger mode.
  4. Generate a cyclic trigger by writing "1" to the trigger field.
- Wait for an interrupt, or poll IStsDiff.
  - ☐ IStsDiff:
    1. Read Signature A (B) to Result registers.
    2. Process results.
- Before the displayed content of the evaluation window changes:
  - ☐ Cancel a cyclic trigger, and reprogram reference values.

1.14 DMA Controller (DMAC)

This section describes the functions and operations of the DMA controller (DMAC) of the GDC macro.

1.14.1 Overview

DMAC is equipped with 2 DMA channels and supports DMA transfer within the GDC macro.

Transfer to outside the GDC macro is not possible.

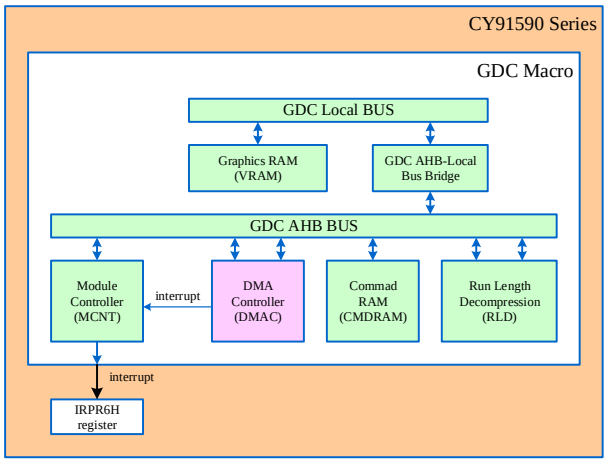
1.14.2 Features

- 2 DMA channels
- Support of software requests as DMA transfer triggers
- Support of the following 2 transfer modes:
  - Block transfer
  - Burst transfer
- Increment addressing/fixed addressing supported for both the source and destination addresses

1.14.3 Configuration

1.14.3.1 Block Diagram

Figure 1-75. DMAC Block Diagram



1.14.4 Registers

This section describes the DMAC registers.

1.14.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (***0040_0000H***) is added for access from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.14.4.2 Register List

Module	Address	Register Name	Function
DMAC common	$01FB_4000_H$	DMACR	DMAC configuration register
DMAC ch.0	$01FB_4010_H$	DMACA0	DMAC0 configuration A register
	$01FB_4014_H$	DMACB0	DMAC0 configuration B register
	$01FB_4018_H$	DMACSA0	DMAC0 source address register
	$01FB_401C_H$	DMACDA0	DMAC0 destination address register
DMAC ch.1	$01FB_4020_H$	DMACA1	DMAC1 configuration A register
	$01FB_4024_H$	DMACB1	DMAC1 configuration B register
	$01FB_4028_H$	DMACSA1	DMAC1 source address register
	$01FB_402C_H$	DMACDA1	DMAC1 destination address register

1.14.4.3 Register Details

DMA Configuration Register (DMACR)

Address	01FB_4000$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DE	DS	-	PR	DH[3:0]				-							
R/W	R/W	R/W0	R	R/W	R/W	R/W	R/W	R/W	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>31</td><td>DE
(DMA Enable)</td><td>This bit controls all DMA channel transfers.

<table><tr><td>0</td><td>Disable all DMA channels, and do not perform a DMA transfer until "1" is set in this bit.
If the bit is set to "0" during DMA transfer, the channel that is in the middle of the transfer stops DMA with a transfer gap.</td></tr><tr><td>1</td><td>Enable all DMA channels.
The start of DMA transfer depends on the settings of each channel.</td></tr></table>
[Transfer gap]
The transfer gap depends on the transfer mode.
Block transfer: Transfer gap at BC="0" (after transfer in BC units is completed)
Burst transfer: No transfer gap

A transfer gap means that DMAC deasserts a bus request to the arbiter for about 4 clocks in the middle of a DMA transfer.
This bit is useful for reconfiguring the Configuration registers of all channels during DMA transfer.</td></tr>
<tr><td>30</td><td>DS
(DMA Stop)</td><td>This bit indicates that the DMA transfers of all channels have stopped.

<table><tr><td>0</td><td>The disable/halt setting has been cleared. (Initial value)</td></tr><tr><td>1</td><td>The DMA transfers of all channels have stopped because of the disable/halt setting.</td></tr></table>
This bit is set to "1" when either of the following conditions is satisfied during DMA transfer.
"0" is set in the DE bit in the DMACR register. (All channels disabled)
A value other than "0" is set in the DH bit in the DMACR register. (All channels halted)

DMAC stops transfer according to the disable/halt settings of all channels. This bit is useful for confirming that the DMA transfers of all channels have stopped.
Except for register initialization ("0" setting), writing "0" or "1" to this bit is ignored.</td></tr>
<tr><td>29</td><td>(Reserved)</td><td>Reserved
Writing is ignored. The value read from this bit is always "0".</td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>28</td><td>PR
(Priority Rotation)</td><td>This bit controls DMA channel priority. However, there is no difference in the behavior of this DMAC because it supports 2ch DMA only.

<table><tr><td>0</td><td>"Fixed"
Priorities: ch.0 > ch.1</td></tr><tr><td>1</td><td>"Rotation"
The order of priority is rotated.</td></tr></table></td></tr>
<tr><td>27-24</td><td>DH[3:0]
(DMA Halt)</td><td>These bits control the stopped state of all DMA channels.
If a value other than "0" is set in the bits, all DMA channels are stopped, and no DMA transfer is performed until "0" is set.
If a value other than "0" is set in the bits during DMA transfer, the channel that is in the middle of the transfer stops DMA with a transfer gap.
For details on the transfer gap, see the DE bit description.

These bits are useful for stopping DMA transfer without reconfiguring the Configuration registers of all channels.
If these bits and the DE bit are set at the same time, the DE bit has priority.

<table><tr><td>0</td><td>Initial value</td></tr><tr><td>Other</td><td>Stop all channels.</td></tr></table></td></tr>
</table>

DMA Configuration A Register (DMACAx)

Address	ch.0: 01FB_4010$_H$ ch.1: 01FB_4020$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	EB0 EB1	PB0 PB1	ST0 ST1	-					BT0[3:0] BT1[3:0]				BC0[3:0] BC1[3:0]			
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	TC0[15:0] TC1[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Field		Description
No.	**Name**	
31	ch.0: EB0 ch.1: EB1 (Enable Bit)	This bit is used to control DMA channel transfer. If "1" is set in the bit, the channel waits for a trigger to start DMA transfer. (The DE bit in the DMACR register must be set to "1" beforehand.) After the DMA transfer is completed, DMAC sets this bit to "0". If "0" is set in the bit, this channel is disabled, and no DMA transfer is performed until "1" is set in the bit. If "0" is set in the bit during DMA transfer, DMA stops with a transfer gap. This is regarded as a forced stop. For details on the transfer gap, see the description about the DE bit in the DMACR register. This bit is useful for reconfiguring the Configuration registers of channels during DMA transfer. 0: Disable this channel. (Initial value) 1: Enable this channel.
30	ch.0: PB0 ch.1: PB1 (Pause Bit)	This bit is used to control pauses in DMA channel transfer. If "1" is set in the bit, this channel stops the transfer, and no DMA transfer is performed until the bit is cleared. If the bit is set to "1" during DMA transfer, DMA stops with a transfer gap. For details on the transfer gap, see the description about the DE bit in the DMACR register. If this bit is set to "0" during a pause in DMA transfer, the pause state is cleared, and DMAC waits for a new transfer request. This bit is useful for stopping DMA transfer without reconfiguring the Configuration registers of each channel. 0: Initial value 1: Stop the channel.

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>29</td><td>ch.0: ST0
ch.1: ST1
(Software Trigger)</td><td>This bit is used to generate software triggers.

If this bit is set to "1", a software request starts DMA transfer. After this DMA transfer is completed, DMAC sets the bit to "0".
If the bit is set to "0" during DMA transfer due to a software request, the DMA transfer is stopped with a transfer gap.

<table><tr><td>0</td><td>Initial value</td></tr><tr><td>1</td><td>Software request</td></tr></table></td></tr>
<tr><td>23-20</td><td>ch.0: BT0[3:0]
ch.1: BT1[3:0]
(Beat Type)</td><td>These bits are used to select beat transfer on GDC AHB.

If these bits are set to NORMAL or SINGLE, access alternates between single source access and single destination access.
If the bits are set to INCR* or WRAP*, access alternates between continuous source access and continuous destination access.
For details on INCR* and WRAP*, see the AMBA specification (v2.0).
If INCR (undefined-length burst) is set, the BC bits specify the burst length.

<table><tr><th>BT[3:0]</th><th>Function</th></tr><tr><td>0_H</td><td>NORMAL (same as SINGLE) (Initial value)</td></tr><tr><td>$1_H – 7_H$</td><td>Invalid</td></tr><tr><td>8_H</td><td>SINGLE (same as NORMAL)</td></tr><tr><td>9_H</td><td>INCR</td></tr><tr><td>A_H</td><td>WRAP4</td></tr><tr><td>B_H</td><td>INCR4</td></tr><tr><td>C_H</td><td>WRAP8</td></tr><tr><td>D_H</td><td>INCR8</td></tr><tr><td>E_H</td><td>WRAP16</td></tr><tr><td>F_H</td><td>INCR16</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>19-16</td><td>ch.0: BC0[3:0]
ch.1: BC1[3:0]
(Block Count)</td><td>These bits are used to specify the number of blocks for block/burst transfer. The maximum number of blocks is 16 ("F_H").
The bits are valid when the beat type (BT) is NORMAL, SINGLE, or INCR.
If another beat type (fixed-length burst and wrap) is set, the bits are ignored.
The bits can be read during DMA transfer.
Normally, the BC bits are decremented by 1 when single source access or single destination access is completed successfully.

Note:
These bits can be set even when the beat type (BT) bit is INCR.
However, after DMA transfer begins, the read data of BC in the DMA transfer of INCR is always "0x0". Therefore, BC monitoring during the DMA transfer of INCR is not necessary.

After DMA transfer is completed successfully, DMAC sets these bits to "0".

<table><tr><th>BC[3:0]</th><th>Function</th></tr><tr><td>X</td><td>Block count (Initial value: 0)</td></tr></table></td></tr>
<tr><td>15-0</td><td>ch.0: TC0[15:0]
ch.1: TC1[15:0]
(Transfer Count)</td><td>These bits are used to specify the number of transfers for block/burst transfer.
The maximum number of transfers is 65536 ($FFFF_H$). The bits are valid when any beat type is set in BT.

These bits can be read during DMA transfer. Normally, in NORMAL mode or SINGLE mode (BT is NORMAL or SINGLE), the TC bits are decremented by 1 upon the successful completion of a DMA transfer of BC="0".
In other beat transfer modes (INCR, INCR*, WRAP*), the TC bits are decremented by 1 after continuous source/destination access is completed once. For example, in the case of INCR4, the TC bits are decremented by 1 when 4 continuous source accesses and 4 continuous destination accesses are completed.
After DMA transfer is completed successfully, DMAC sets these bits to "0000_H".

<table><tr><th>TC[15:0]</th><th>Function</th></tr><tr><td>X</td><td>Transfer count (Initial value: 0000_H)</td></tr></table></td></tr>
</table>

DMA Configuration B Register (DMACBx)

Address	ch.0: 01FB_4014$_H$ ch.1: 01FB_4024$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	TT0[1:0] TT1[1:0]		MS0[1:0] MS1[1:0]		TW0[1:0] TW1[1:0]		FS0 FS1	FD0 FD1	RC0 RC1	RS0 RS1	RD0 RD1	EI0 EI1	CI0 CI1	SS0[2:0] SS1[2:0]		
R/W	R/W	R/W	R/W0	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W0	R/W0	R/W0
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

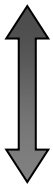
<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>31, 30</td><td>ch.0: TT0[1:0]
ch.1: TT1[1:0]
(Transfer Type)</td><td>These bits are used to specify the transfer type.
DMAC supports 2-cycle transfer mode only.
<table><tr><th>TT[1:0]</th><th>Function</th></tr><tr><td>0</td><td>2-cycle transfer (Initial value)</td></tr><tr><td>Other</td><td>Reserved</td></tr></table></td></tr>
<tr><td>29, 28</td><td>ch.0: MS0[1:0]
ch.1: MS1[1:0]
(Mode Select)</td><td>These bits are used to select a transfer mode.
<table><tr><th>MS[1:0]</th><th>Function</th></tr><tr><td>0_H</td><td>Block transfer mode (Initial value)</td></tr><tr><td>1_H</td><td>Burst transfer mode</td></tr><tr><td>2_H</td><td>Reserved</td></tr><tr><td>3_H</td><td>Reserved</td></tr></table></td></tr>
<tr><td>27, 26</td><td>ch.0: TW0[1:0]
ch.1: TW1[1:0]
(Transfer Width)</td><td>These bits are used to specify the transfer data width.
<table><tr><th>TW[1:0]</th><th>Function</th></tr><tr><td>0_H</td><td>1 byte (Initial value)</td></tr><tr><td>1_H</td><td>2 bytes</td></tr><tr><td>2_H</td><td>4 bytes</td></tr><tr><td>3_H</td><td>Reserved</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>25</td><td>ch.0: FS0
ch.1: FS1
(Fixed Source)</td><td>This bit is used to determine the source address. If the source address must be incremented after each transfer, be sure to set "0" in the bit.
<table><tr><th>FS</th><th>Function</th></tr><tr><td>0</td><td>Increment the source address. (Initial value)</td></tr><tr><td>1</td><td>Fix the source address.</td></tr></table></td></tr>
<tr><td>24</td><td>ch.0: FD0
ch.1: FD1
(Fixed Destination)</td><td>This bit is used to determine the destination address. If the destination address must be incremented after each transfer, be sure to set "0" in the bit.
<table><tr><th>FD</th><th>Function</th></tr><tr><td>0</td><td>Increment the destination address. (Initial value)</td></tr><tr><td>1</td><td>Fix the destination address.</td></tr></table></td></tr>
<tr><td>23</td><td>ch.0: RC0
ch.1: RC1
(Reload Count)</td><td>This bit is used to control the reload function for the block count (BC in the DMACA register) and the transfer count (TC in the DMACA register).
If this bit is set to "1", BC in the DMACA register and TC in the DMACA register are set to their initial values after DMA transfer is completed.
<table><tr><th>RC</th><th>Function</th></tr><tr><td>0</td><td>Disable the reload function for the transfer count. (Initial value)</td></tr><tr><td>1</td><td>Enable the reload function for the transfer count.</td></tr></table></td></tr>
<tr><td>22</td><td>ch.0: RS0
ch.1: RS1
(Reload Source)</td><td>This bit is used to control the reload function for the source address (DMACSA register).
If the bit is set to "1", the DMACSA register is set to its initial value after DMA transfer is completed.
If the bit is set to "0", DMAC sets the next source address in the DMACSA register after DMA transfer is completed.
<table><tr><th>RS</th><th>Function</th></tr><tr><td>0</td><td>Disable the reload function for the source address. (Initial value)</td></tr><tr><td>1</td><td>Enable the reload function for the source address.</td></tr></table></td></tr>
<tr><td>21</td><td>ch.0: RD0
ch.1: RD1
(Reload Destination)</td><td>This bit is used to control the reload function for the destination address (DMACDA register).
If the bit is set to "1", the DMACDA register is set to its initial value after DMA transfer is completed.
If the bit is set to "0", DMAC sets the next destination address in the DMACDA register after DMA transfer is completed.
<table><tr><th>RD</th><th>Function</th></tr><tr><td>0</td><td>Disable the reload function for the destination address. (Initial value)</td></tr><tr><td>1</td><td>Enable the reload function for the destination address.</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>20</td><td>ch.0: EI0
ch.1: EI1
(Error Interrupt)</td><td>This bit controls error interrupts issued to MCNT.
With "1" set in the bit, the logical add results of the following transfer errors are issued as error interrupts:
■ Address overflow
■ Transfer disabled by the EB and DE bits
■ Source access error
■ Destination access error
<table><tr><th>EI</th><th>Function</th></tr><tr><td>0</td><td>Disable error interrupt issuance. (Initial value)</td></tr><tr><td>1</td><td>Enable error interrupt issuance.</td></tr></table></td></tr>
<tr><td>19</td><td>ch.0: CI0
ch.1: CI1
(Completion Interrupt)</td><td>This bit controls transfer completion interrupts issued to MCNT.
With "1" set in the bit, transfer completion interrupts are issued.
<table><tr><th>CI</th><th>Function</th></tr><tr><td>0</td><td>Disable completion interrupt issuance. (Initial value)</td></tr><tr><td>1</td><td>Enable completion interrupt issuance.</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>18-16</td><td>ch.0: SS0[2:0]
ch.1: SS1[2:0]
(Stop Status)</td><td>
These bits are used to indicate the DMA transfer end code.
The following table lists the end codes.
The bits are also used to clear error/transfer completion interrupts issued to MCNT.
If such an interrupt is issued because of an error or a normal end, writing "000_B" to the bits clears the interrupt.

<table>
<tr><th>SS</th><th>Function</th><th>Status type</th></tr>
<tr><td>0_H</td><td>Initial value</td><td>None</td></tr>
<tr><td>1_H</td><td>Address overflow</td><td>Error</td></tr>
<tr><td>2_H</td><td>Transfer stop request</td><td>Error</td></tr>
<tr><td>3_H</td><td>Source access error</td><td>Error</td></tr>
<tr><td>4_H</td><td>Destination access error</td><td>Error</td></tr>
<tr><td>5_H</td><td>Normal end</td><td>End</td></tr>
<tr><td>6_H</td><td>Reserved</td><td>-</td></tr>
<tr><td>7_H</td><td>DMA pause</td><td>None</td></tr>
</table>

The order of priority for displaying the end codes of various errors occurring at the same time is as follows.
High priority

Reset
Clear by writing 000_B
Address overflow
Stop request
Source access error
Destination access error
Low priority
</td></tr>
</table>

DMAC Source Address Register (DMACSAx)

Address	ch.0: 01FB_4018$_H$ ch.1: 01FB_4028$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DMACSA0[31:16] DMACSA1[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DMACSA0[15:0] DMACSA1[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>31-0</td><td>ch.0: DMACSA0[31:0]
ch.1: DMACSA1[31:0]
(DMAC Source Address)</td><td>These bits are used to specify a source address for starting DMA transfer.
The bits can be read during DMA transfer.
After source access is performed successfully with the fixed address function (FS bit in the DMACB register) disabled, the bits are incremented according to the transfer width (TW bits in the DMACB register).

DMAC sets these bits to the next source address when DMA transfer is completed.

Note:
■ Do not set the address of this module in DMACSA.
■ The set value in this register is used as is for bus access. If it is not consistent with the TW bit setting in the DMACB register, a source access error interrupt is generated, so take note of this fact.

<table>
<tr><th>DMACSA</th><th>Function</th></tr>
<tr><td>X</td><td>Source address for starting DMA transfer
(Initial value: 00000000$_H$)</td></tr>
</table>
</td></tr>
</table>

DMAC Destination Address Register (DMACDAx)

Address	ch.0: 01FB_401C$_H$ ch.1: 01FB_402C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DMACDA0[31:16] DMACDA1[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DMACDA0[15:0] DMACDA1[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Description</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>31-0</td><td>ch.0: DMACDA0[31:0]
ch.1: DMACDA1[31:0]
(DMAC Destination Address)</td><td>These bits are used to specify a destination address for starting DMA transfer.
The bits can be read during DMA transfer.
After destination access is performed successfully with the fixed address function (FD bit in the DMACB register) disabled, the bits are incremented according to the transfer width (TW bits in the DMACB register).

DMAC sets these bits to the next destination address when DMA transfer is completed.

Note:
■ Do not set the address of this module in DMACDA.
■ The set value in this register is used as is for bus access. If it is not consistent with the TW bit setting in the DMACB register, a destination access error interrupt is generated, so take note of this fact.

<table>
<tr><th>DMACDA</th><th>Function</th></tr>
<tr><td>X</td><td>Destination address for starting DMA transfer
(Initial value: 00000000$_H$)</td></tr>
</table>
</td></tr>
</table>

1.14.5 Explanation of Operation

This section describes DMAC operation.

1.14.5.1 Transfer Modes

DMAC has 2 different transfer modes. The MS bits in the DMACB register determine the transfer mode.

Block Transfer

Explanation of Operation

In block transfer mode, the DMA transfer specified in the block count (BC in the DMACA register) is performed with 1 transfer request. If the transfer count (TC in the DMACA register) is set to a value other than "0", TC is decremented by 1 after the DMA transfer of BC is completed.

After the last transfer (BC is "0" and TC is "0000_H") has been performed, DMA transfer ends.

Transfer Gap

In block transfer mode, DMAC negates the bus request to the arbiter once after the transfer of the BC bit is completed. This operation prevents DMAC from monopolizing the bus.

A transfer gap is useful for reflecting register settings (e.g., disable/pause setting) to DMAC during DMA transfer.

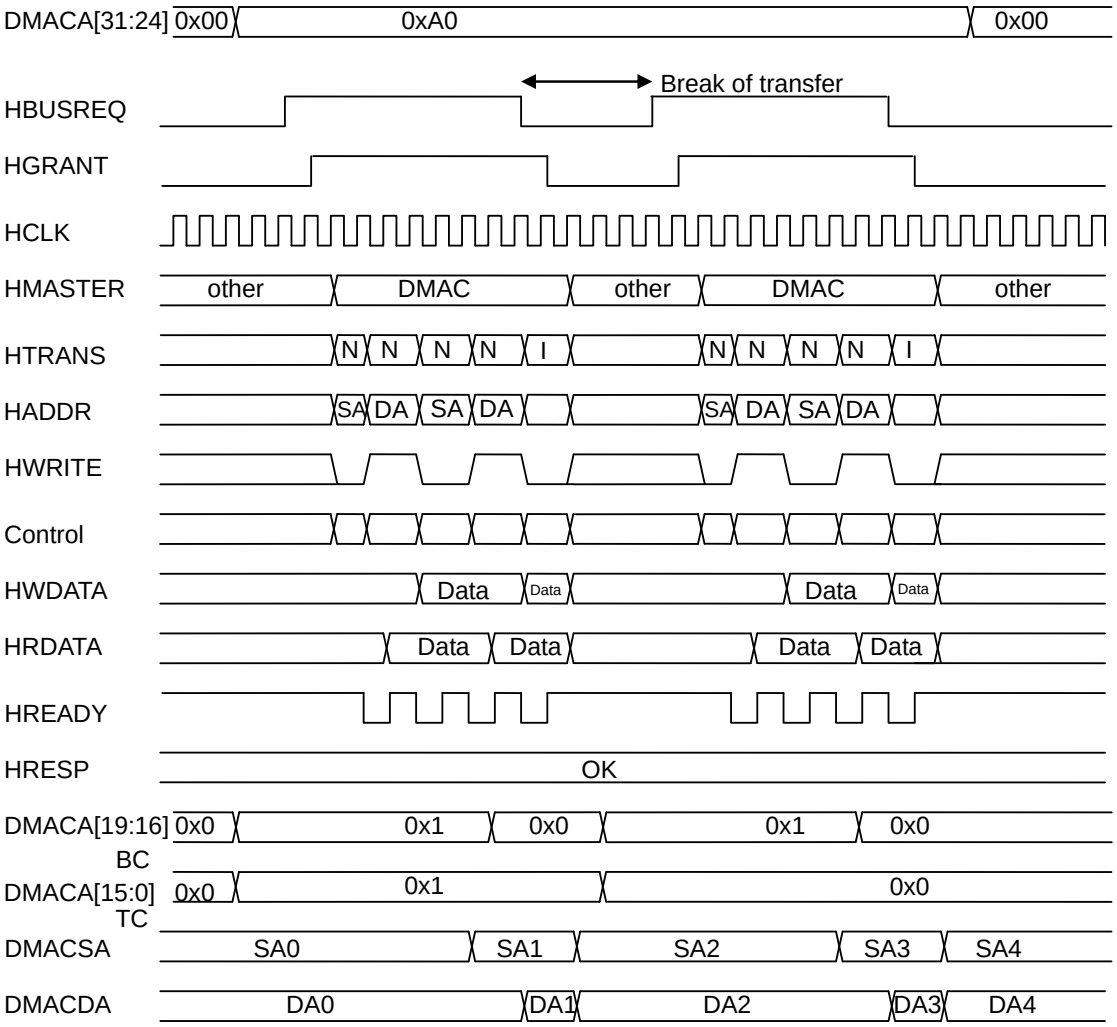
Transfer Request

The ST setting in the DMACA register is "1".

Timing Chart

Figure 1-76 shows a block transfer timing chart.

Figure 1-76. Block Transfer (BC="1" and TC="1")



Burst Transfer

Explanation of Operation

In burst transfer mode, a DMA transfer where the block count is multiplied by the transfer count (BC in the DMACA register x TC in the DMACA register) is performed with 1 transfer request. If the transfer count (TC in the DMACA register) is set to a value other than "0", TC is decremented by 1 after the DMA transfer of BC is completed.

After the last transfer (BC is "0" and TC is "0000_H") has been performed, DMA transfer ends.

Transfer Gap

In burst transfer mode, DMAC negates the bus request to the arbiter after DMA transfer is completed. As a result, the burst transfer does not have a transfer gap.

Changes made to register settings (e.g., disable/pause setting) during DMA transfer are reflected after the DMA transfer is completed.

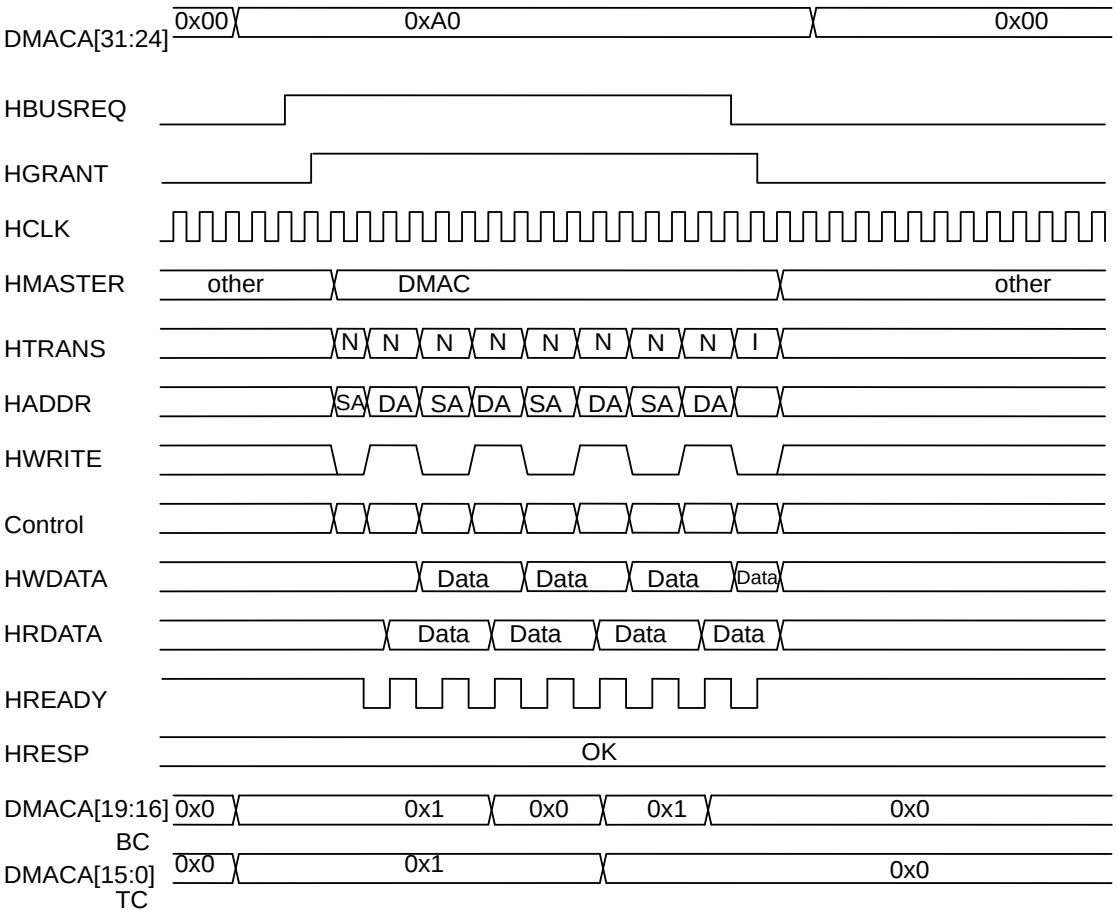
Transfer Request

The ST setting in the DMACA register is "1".

Timing Chart

Figure 1-77 shows a burst transfer timing chart.

Figure 1-77. Burst Transfer (BC="1" and TC="1")



1.14.5.2 Beat Transfer

DMAC supports beat transfer. The term "beat transfer" means incrementing/wrapping a burst as described in the AMBA specification.

DMAC has a 64-byte FIFO that is shared by all channels, thereby enabling continuous source access and destination access.

The BT bits in the DMACA register set the beat transfer type.

The following table shows the relationship between the BT bits in the DMACA register and HBURST of AHB.

Table 1-23. DMACA/BT and HBURST

DMACA/BT	Beat Transfer Type	HBURST	DMACA/MS (Mode Select)	
			Block	Burst
0000_B	NORMAL	SINGLE	OK	OK
1000_B	SINGLE	SINGLE	OK	OK
1001_B	INCR	INCR	OK	OK
1010_B	WRAP4	WRAP4	OK	OK
1011_B	INCR4	INCR4	OK	OK
1100_B	WRAP8	WRAP8	OK	OK
1101_B	INCR8	INCR8	OK	OK
1110_B	WRAP16	WRAP16	OK	OK
1111_B	INCR16	INCR16	OK	OK

1.14.5.3 Error

DMAC supports error response from AHB slaves.

AHB Slave Error

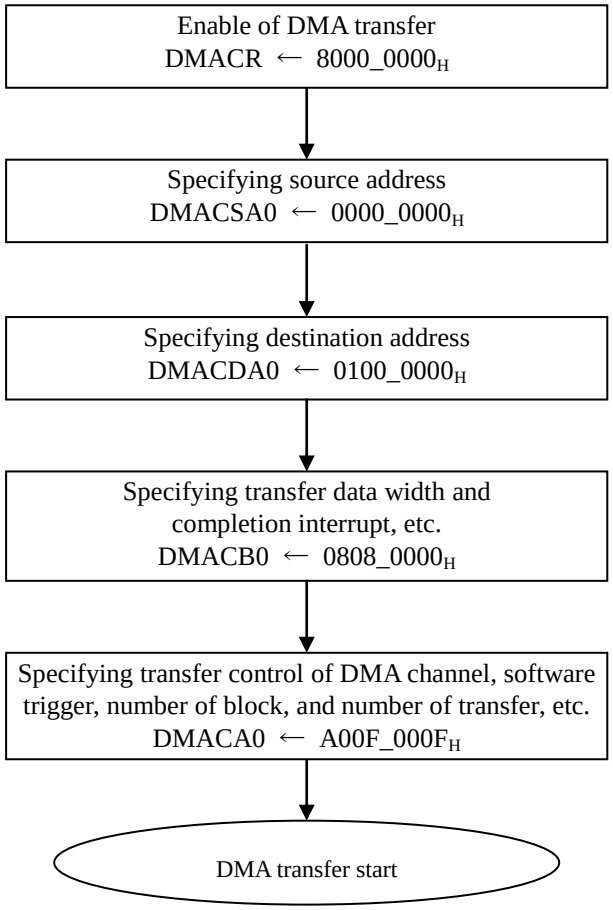
If DMAC receives an error response from an AHB slave during DMA transfer, DMAC immediately stops transfer even though the DMA transfer is in progress.

In such cases, the block/transfer count register and the source/destination address register are not updated.

1.14.6 Examples

1.14.6.1 DMA Start with a Single Channel

Example of block transfer by software request.



In cases of software requests, configure the DMACA register at the end.

1.15 Memory Controller (MEMC)

This section describes the Memory Controller (referred to below as MEMC) of the GDC macro.

1.15.1 Overview

The GDC macro is equipped with external bus memory interfaces that can connect to SRAM or NOR Flash. These interfaces are controlled exclusively with other external bus memory interfaces. (For details on the exclusive control, see "1.5.1.4 External Bus Memory Interfaces".)

1.15.1.1 Features

The MEMC module has the following features:

- Up to 64MB supported
  - Chip select 0 (MEM_XCS0) = 64MB, or
  - chip select 0 (MEM_XCS0) = 32MB and chip select 1 (MEM_XCS1) = 32MB
- 8-bit (or 16-bit) wide SRAM and NOR Flash supported
- External bus memory interfaces equipped with 2 chip selects (MEM_XCS(0/1))
- Each chip select (MEM_XCS(0/1)) individually configurable
- NOR Flash page access supported

1.15.1.2 Restrictions

CY91F591/2/4/6/7/9

- In cases of 64MB support with MEM_XCS0, the ADCS bit (bit0) in the GCONT register of the MCNT module must have a setting.
  (For details, see "GCONT Register" and "Switching Control Signals of External Bus Memory Interfaces" about the MCNT module.)
- In cases of 64MB support with MEM_XCS0, MEM_XCS1 cannot be used.
- 64MB support with MEM_XCS1 is not possible.
- To use the MEMC module, be sure to set the MEM_ED pins to PULL-UP or PULL-DWN in the PPCR01 and PPER01 registers of the FR81S.
  (For details on the setting pins, see the CY91590 Series Hardware Manual.)
- Access beyond the external memory capacity is not permitted.
- Only SPICNT can use MEMC exclusively.
- The data bus input/output pin of MEMC (MEM_ED[14:9]) and the display digital output pin of Display (ROUT[1:0], GOUT[1:0], GOUT[1:0]) are exclusively controlled.
  When RGB digital output (8 bits × 3) is used, only 8-bit wide SRAM and NOR Flash can be used.
  (For details on the setting pins, see the CY91590 Series Hardware Manual.)

CY91F59A/B

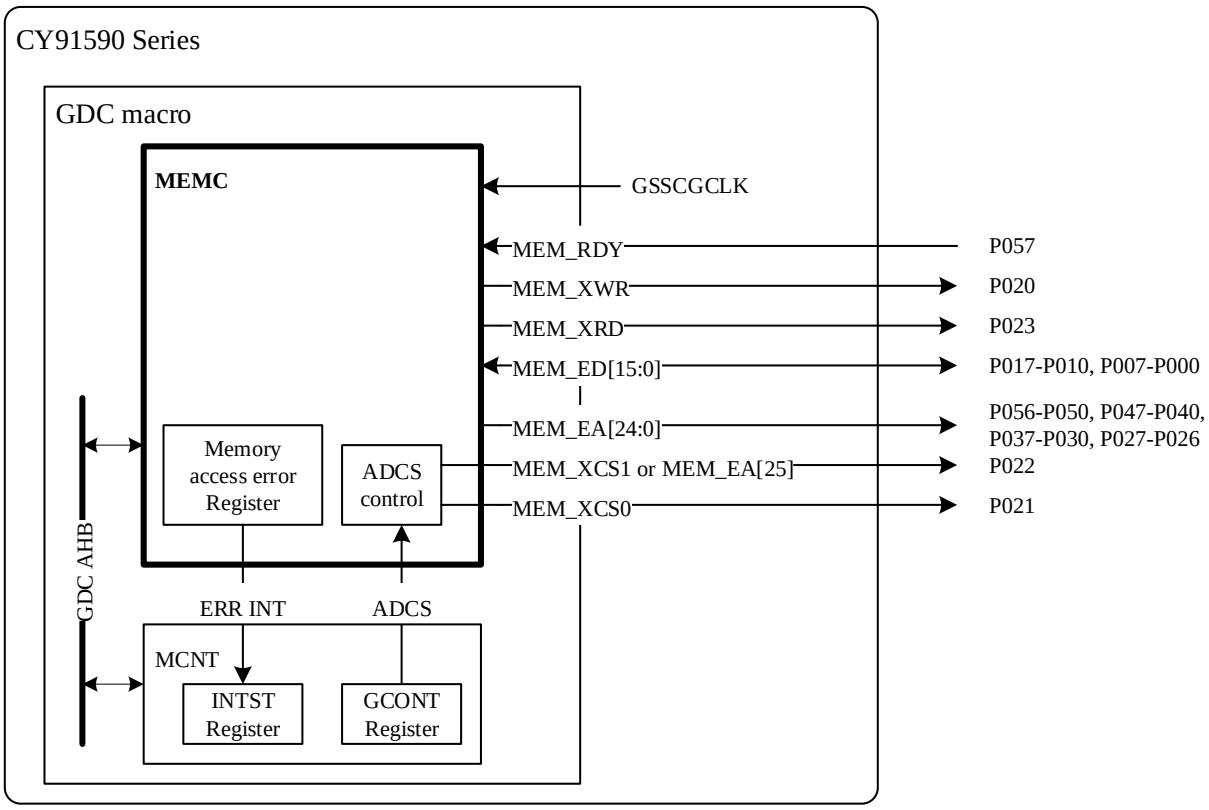
- In cases of 64MB support with MEM_XCS0, the ADCS bit (bit0) in the GCONT register of the MCNT module must have a setting.
  (For details, see "GCONT Register" and "Switching Control Signals of External Bus Memory Interfaces" about the MCNT module.)
- In cases of 64MB support with MEM_XCS0, MEM_XCS1 cannot be used.
- 64MB support with MEM_XCS1 is not possible.
- In cases of 64MB support with MEM_XCS0, the ADCS bit (bit0) in the GCONT register of the MCNT module must have a setting.
  (For details, see "GCONT Register" and "Switching Control Signals of External Bus Memory Interfaces" about the MCNT module.)
- In cases of 64MB support with MEM_XCS0, MEM_XCS1 cannot be used.
- 64MB support with MEM_XCS1 is not possible.
- To use the MEMC module, be sure to set the MEM_ED pins to PULL-UP or PULL-DWN in the PPCR01 and PPER01 registers of the FR81S.
  (For details on the setting pins, see the CY91590 Series Hardware Manual.)
- Access beyond the external memory capacity is not permitted.
- Only SPICNT and HS-SPICNT can use MEMC exclusively.
- The data bus input/output pin of MEMC (MEM_ED[14:9]) and the display digital output pin of Display (ROUT[1:0], GOUT[1:0], GOUT[1:0]) are exclusively controlled.
  When RGB digital output (8 bits × 3) is used, only 8-bit wide SRAM and NOR Flash can be used.
  (For details on the setting pins, see the CY91590 Series Hardware Manual.)

1.15.2 Configuration

1.15.2.1 Block Diagram

The following figure is a block diagram showing the MEMC module.

Figure 1-78. MEMC Module Block Diagram



1.15.2.2 External Pins

Table 1-24. External Interface Pins

LSI External Pin Name	Pin Name	I/O	Function
P056	MEM_EA[24]	O	Address bus
P055	MEM_EA[23]		
P054	MEM_EA[22]		
P053	MEM_EA[21]		
P052	MEM_EA[20]		
P051	MEM_EA[19]		
P050	MEM_EA[18]		
P047	MEM_EA[17]		
P046	MEM_EA[16]		
P045	MEM_EA[15]		
P044	MEM_EA[14]		
P043	MEM_EA[13]		
P042	MEM_EA[12]		
P041	MEM_EA[11]	O	Address bus
P040	MEM_EA[10]		
P047	MEM_EA[9]		
P036	MEM_EA[8]		
P035	MEM_EA[7]		
P034	MEM_EA[6]		
P033	MEM_EA[5]		
P032	MEM_EA[4]		
P031	MEM_EA[3]		
P030	MEM_EA[2]		
P027	MEM_EA[1]		
P026	MEM_EA[0]		
P020	MEM_XWR	O	Write enabled
P023	MEM_XRD	O	Read enabled
P021	MEM_XCS0	O	Chip select 0
P022	MEM_XCS1/ MEM_EA[25]	O	Chip select 1/ address bus[1]

LSI External Pin Name	Pin Name	I/O	Function
P017	MEM_ED[15]	IO	Data bus[2][4]
P016	MEM_ED[14]		
P015	MEM_ED[13]		
P014	MEM_ED[12]		
P013	MEM_ED[11]		
P012	MEM_ED[10]		
P011	MEM_ED[9]		
P010	MEM_ED[8]		
P007	MEM_ED[7]		
P006	MEM_ED[6]		
P005	MEM_ED[5]		
P004	MEM_ED[4]		
P003	MEM_ED[3]		
P002	MEM_ED[2]		
P001	MEM_ED[1]		
P000	MEM_ED[0]		
P057	MEM_RDY	I	Ready input for low-speed device[3]

[1]: The MEM_XCS1 pin can be used as the MEM_EA[25] pin. As the MEM_EA[25] pin, it can connect with SRAM or NOR Flash of up to 64MB.
To use it as the MEM_EA[25] pin, set the ADCS bit in the GCONT register ($01FB_2500_H$) of the MCNT module to "1".

[2]: Be sure to set the MEM_ED pins to PULL-UP or PULL-DWN in the PPCR01 and PPER01 registers of the FR81S. (For details on the setting pins, see the CY91590 Series Hardware Manual.)

[3]: When not using it, connect it to "H."

[4]: The data bus input/output pin of MEMC (MEM_ED[14:9]) and the display digital output pin of Display (ROUT[1:0], GOUT[1:0], GOUT[1:0]) are exclusively controlled.
(For details on the setting pins, see the CY91590 Series Hardware Manual.)

1.15.3 Registers

1.15.3.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (***0040_0000H***) is added for access from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.15.3.2 Register List

Table 1-25. Register List

Address	Register Name	Description
$01FB_3000_H$	MCFMODE0	Mode register of SRAM and NOR Flash in MEM_XCS0
$01FB_3004_H$	MCFMODE1	Mode register of SRAM and NOR Flash in MEM_XCS1
$01FB_3020_H$	MCFTIM0	Timing register of SRAM and NOR Flash in MEM_XCS0
$01FB_3024_H$	MCFTIM1	Timing register of SRAM and NOR Flash in MEM_XCS1
$01FB_3040_H$	MCFAREA0	Access area register of SRAM and NOR Flash in MEM_XCS0
$01FB_3044_H$	MCFAREA1	Access area register of SRAM and NOR Flash in MEM_XCS1
$01FB_3200_H$	MCERR	Interrupt error in MEMC

1.15.3.3 MCFMODE0/1 (Memory Controller Flash MODE 0/1)

Address	01FB_3000$_H$ (MEM_XCS0) 01FB_3004$_H$ (MEM_XCS1)															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W0															
Initial	X															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-									RDY	PAGE	-				WDTH
R/W	R/W0									R/W	R/W	R/W0				R/W
Initial	X									0	0	X				1

Bit Field		Explanation
No.	**Name**	
0	WDTH (data WiDTH)	This bit specifies the data bus width of connected devices. **WDTH** / **Data Width of External BUS Memory** 0 – 8 bits 1 – 16 bits
5	PAGE (PAGE access mode)	This bit controls the page access mode of NOR Flash. In NOR Flash page access mode, the first address cycle is issued according to the FRADC bit setting in the MCFTIM0/1 register. Then, the access continues until it reaches the 16-byte boundary, according to the RACC bit setting in the MCFTIM0/1 register. To select this mode, also set the RADC bits in the MCFTIM0/1 register to "0". **PAGE** / **Page Access Mode of NOR Flash** 0 – OFF 1 – ON
6	RDY (ReaDY mode)	For handshaking with a low-speed device using the MEM_RDY signal, set this bit to "1". Assert the MEM_RDY signal to "L" within 1 cycle (GSSCGCLK) from the falling of a chip select signal (MEM_XCS(0/1)). (For details, see "Ready Mode".) Like with SRAM, when not using the MEM_RDY signal, set this bit to "0". **RDY** / **Ready Mode** 0 – OFF 1 – ON

Note: Be sure to write "0" in each Reserved bit ("-"). If "1" is written, operation is not guaranteed.

1.15.3.4 MCFTIM0/1
(Memory Controller Flash TIMing 0/1)

Address	01FB_3020$_H$ (MEM_XCS0) 01FB_3024$_H$ (MEM_XCS1)															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	WIDLC[3:0]				WEC[3:0]				WADC[3:0]				WACC[3:0]			
R/W	R/W				R/W				R/W				R/W			
Initial	0_H				5_H				5_H				F_H			
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	RIDLC[3:0]				FRADC[3:0]				RADC[3:0]				RACC[3:0]			
R/W	R/W				R/W				R/W				R/W			
Initial	F_H				0_H				0_H				F_H			

Bit Field		Explanation
No.	**Name**	
3-0	RACC[3:0] (Read ACcess Cycle)	These bits set the number of cycles (GSSCGCLK) required for read access. Even though the address does not change during the cycles specified by the bits, the data is fetched in the last cycle. (For details, see "Read Access Cycle".) **RACC** / **Cycle (GSSCGCLK)** 0_H / 1 cycle | / | F_H / 16 cycles
7-4	RADC[3:0] (Read ADdress setup Cycle)	These bits set the number of read address setup cycles (GSSCGCLK). The value specified here must be within the range of the number of read access cycles. When using NOR Flash page access mode (PAGE bit in the MCFMODE0/1 register), set these bits to "0". When setting these bits to a value other than "0", specify "0" in the FRADC bits. (For details, see "Read Address Setup Cycle".) **RADC** / **Cycle (GSSCGCLK)** 0_H / 0 cycles | / | F_H / 15 cycles
11-8	FRADC[3:0] (First Read ADdress Cycle)	These bits are used exclusively for the NOR Flash setting supporting page mode access. The bits set the initial latency for an address in NOR Flash read access. When not using NOR Flash page access mode (PAGE bit in the MCFMODE0/1 register), set these bits to "0". When setting these bits to a value other than "0", specify "0" in the RADC bits. (For details, see "First Read Address Cycle".) **FRADC** / **Cycle (GSSCGCLK)** 0_H / 0 cycles | / | F_H / 15 cycles

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>15-12</td><td>RIDLC[3:0]
(Read IDLe Cycle)</td><td>These bits set the number of idle cycles after read access.
(For details, see "Read Idle Cycle".)
<table><tr><th>RIDLC</th><th>Cycle (GSSCGCLK)</th></tr><tr><td>0_H</td><td>1 cycle</td></tr><tr><td>|</td><td>|</td></tr><tr><td>F_H</td><td>16 cycles</td></tr></table></td></tr>
<tr><td>19-16</td><td>WACC[3:0]
(Write ACcess Cycle)</td><td>These bits specify the number of cycles required for write access.
This value must be larger than the sum of the WADC bits and WEC bits.
(For details, see "Write Access Cycle".)
<table><tr><th>WACC</th><th>Cycle (GSSCGCLK)</th></tr><tr><td>0_H</td><td>Reserved</td></tr><tr><td>1_H</td><td>Reserved</td></tr><tr><td>2_H</td><td>3 cycles</td></tr><tr><td>|</td><td>|</td></tr><tr><td>F_H</td><td>16 cycles</td></tr></table></td></tr>
<tr><td>23-20</td><td>WADC[3:0]
(Write ADdress setup Cycle)</td><td>These bits set the number of write access setup cycles.
(For details, see "Write Address Setup Cycle".)
<table><tr><th>WADC</th><th>Cycle (GSSCGCLK)</th></tr><tr><td>0_H</td><td>1 cycle</td></tr><tr><td>|</td><td>|</td></tr><tr><td>E_H</td><td>15 cycles</td></tr><tr><td>F_H</td><td>Reserved</td></tr></table></td></tr>
<tr><td>27-24</td><td>WEC[3:0]
(Write Enable Cycle)</td><td>These bits set the number of write enable cycles. This bit setting also affects MEM_XWR.
(For details, see "Write Enable Cycle".)
<table><tr><th>WEC</th><th>Cycle (GSSCGCLK)</th></tr><tr><td>0_H</td><td>1 cycle</td></tr><tr><td>|</td><td>|</td></tr><tr><td>E_H</td><td>15 cycles</td></tr><tr><td>F_H</td><td>Reserved</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>31-28</td><td>WIDLC[3:0]
(Write IDLe Cycle)</td><td>These bits set the number of idle cycles after write access. When setting the RDY bit to "1", specify a value of "2" or more.
(For details, see "Write Idle Cycle".)

<table><tr><th>WIDLC</th><th>Cycle (GSSCGCLK)</th></tr><tr><td>0_H</td><td>1 cycle</td></tr><tr><td>|</td><td>|</td></tr><tr><td>F_H</td><td>16 cycles</td></tr></table></td></tr>
</table>

1.15.3.5 MCFAREA0/1 (Memory Controller Flash AREA 0/1)

Address	01FB_3040$_H$ (MEM_XCS0)															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-									MASK[6:0]						
R/W	R/W0									R/W						
Initial	X									1F$_H$						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								ADDR[7:0]							
R/W	R/W0								R/W							
Initial	X								20$_H$							

Address	01FB_3044$_H$ (MEM_XCS1)															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-									MASK[6:0]						
R/W	R/W0									R/W						
Initial	X									1F$_H$						
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								ADDR[7:0]							
R/W	R/W0								R/W							
Initial	X								40$_H$							

Bit Field		**Explanation**
No.	**Name**	
7-0	ADDR[7:0] (ADDRess)	These bits specify the start address of the corresponding chip select area. (For details, see "Start Address".) [Restrictions] Be sure to set 20$_H$ for the start address of MEM_XCS0 (01FB_3040$_H$). If the set address is not 20$_H$, the trigger start of the CMDSEQ module does not operate normally, so operation is not guaranteed. However, if the trigger start of the CMDSEQ module is not used, the address does not need to be fixed at 20$_H$.
22-16	MASK[6:0] (address MASK)	These bits set the memory size (area) for the set address in the ADDR bit. (For details, see "Memory Size" and "Chip Select Area Settings".) The following table shows the relationship between the MASK bits and memory size. **MASK** / **Memory Size** 1F$_H$ – 32MB 0F$_H$ – 16MB 07$_H$ – 8MB 03$_H$ – 4MB 01$_H$ – 2MB 00$_H$ – 1MB Other – Prohibition [Restrictions] If the MASK bit setting is Other (setting prohibited), external bus memory is not normally accessible, so operation is not guaranteed.

Notes:

- Be sure to write "0" in each Reserved bit ("-"). If "1" is written, the operation of this module is not guaranteed.
- Configure the areas of the 2 chip selects (MEM_XCS(0/1)) set by the ADDR bits and MASK bits, such that the areas do not overlap. If these chip select areas overlap, external bus memory is not normally accessible, so operation is not guaranteed.

1.15.3.6 MCERR (Memory Controller ERRor)

Address	01FB_3200$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W0															
Initial	X															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-													SFION	-	SFER
R/W	R/W0													R/W	R	R/W
Initial	X													0	0	0

Bit Field		Explanation
No.	**Name**	
0	SFER (Sram nor Flash access ERror)	This bit indicates that an area not mapped to external bus memory was accessed. Upon detecting the error, this module notifies the MCNT module of an error interrupt (Error INT) and sets Error ("1") in this bit. Writing "1" can clear this bit error. **SFER** / **Access Error** 0 – No error 1 – Error
2	SFION (Sram nor Flash access error Interrupt ON)	This bit controls the interrupt information that is output to the INT26 bit in the INTST register of the MCNT module. **SFION** / **Interrupt Output Control** 0 – OFF 1 – ON

Note: Be sure to write "0" in each Reserved bit ("-"). If "1" is written, the operation of this module is not guaranteed.

1.15.4 Explanation of Operation

The MEMC module has areas of up to 64MB with 2 chip select signals (MEM_XCS(0/1)). The supported devices are 8-bit (or 16-bit) SRAM and NOR Flash.

The following sections describe each function.

1.15.4.1 Endian

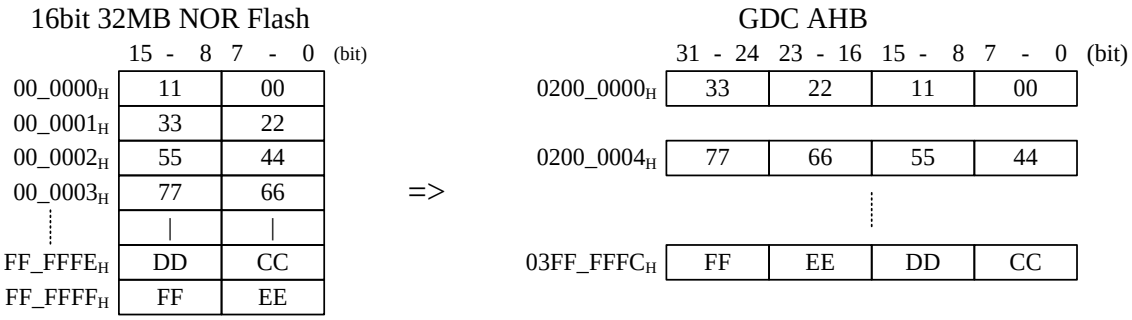
The GDC macro supports the MEMC module registers, external bus memory, and Little Endian (not Big Endian).

Therefore, be sure to write in Little Endian when writing with a ROM writer, etc. to external bus memory (NOR Flash). If the writing uses Big Endian, the operation of the GDC macro is not guaranteed.

The following figure is a conceptual diagram that shows writing of 16-bit 32MB NOR Flash and reading of the NOR Flash from GDC AHB.

In this conceptual diagram, the MEM_XCS0 memory area is from 0200_0000_H to $03FF_FFFF_H$.

Figure 1-79. Endian of NOR Flash



1.15.4.2 External Bus Memory Mode

External bus memory mode controls the data bus width of the device connected to external bus memory, NOR Flash page access, and low-speed devices. This control can be configured with the MCFMODE0/1 register (MEM_XCS0=01FB_3000$_H$ and MEM_XCS1=01FB_3004$_H$).

Be sure to make these settings before accessing external bus memory.

(For details on the setting timing, see "GDC Access Sequence".)

Data Bus Width

The device connected to each chip select of an external bus memory interface has a specified data bus width (8 bits or 16 bits). This setting is made with the WDTH bit in the MCFMODE0/1 register.

Table 1-26. Data Width of External Bus Memory

MCFMODE0/1 Register WDTH (bit0)	Data Width of External Bus Memory
0	8 bits
1	16-bit (Default)

Page Access Mode

Page access mode controls NOR Flash page access.

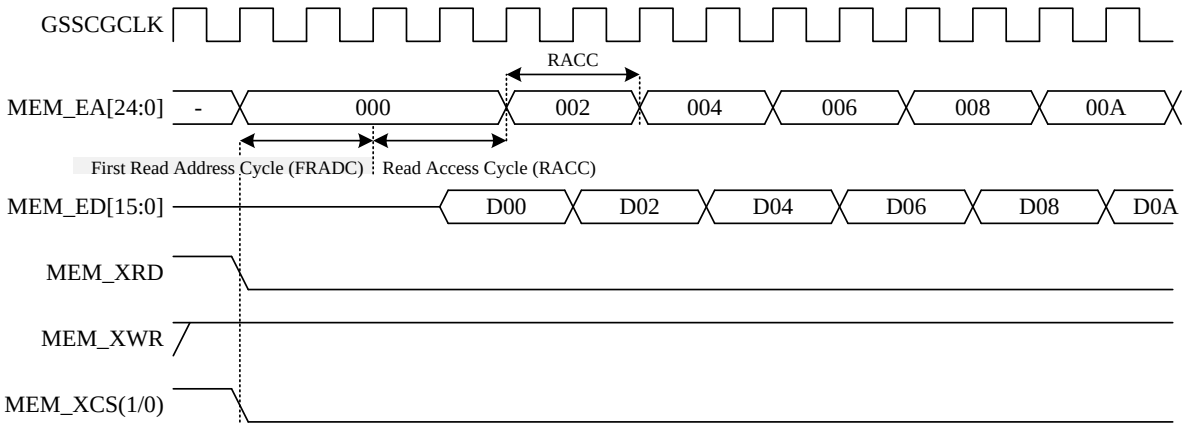
In NOR Flash page access mode, the first address cycle is issued according to the FRADC bit setting in the MCFTIM0/1 register. Then, the access continues until it reaches the 16-byte boundary, according to the RACC bit setting in the MCFTIM0/1 register. To select this mode, also set the RADC bits in the MCFTIM0/1 register to "0".

This setting is made with the PAGE bit in the MCFMODE0/1 register.

Table 1-27. Page Access Mode of NOR Flash

MCFMODE0/1 Register PAGE (bit1)	Page Access Mode of NOR Flash
0	OFF (Default)
1	ON

Figure 1-80. Page Read of 16-bit NOR Flash



Ready Mode

The MEMC module has an interface function using a low-speed device and the MEM_RDY pin. The RDY signal of the low-speed device is connected to the MEM_RDY pin of this LSI to use this function. This setting is made with the RDY bit in the MCFMODE0/1 register.

Note the following points given for using ready mode.

This function cannot be applied to the RDY/BUSY signal of NOR Flash.

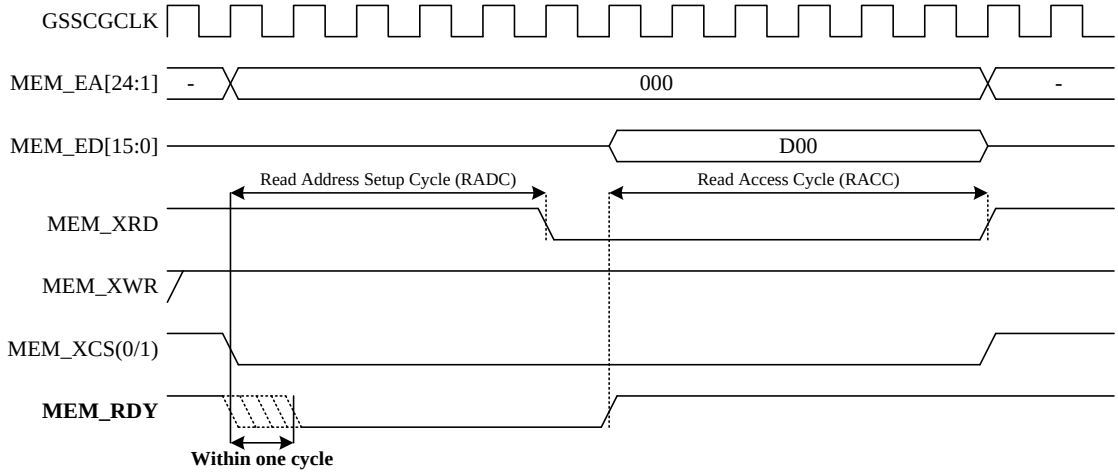
- When using 2 low-speed devices, an external AND circuit must handle the processing because there is only 1 MEM_RDY pin.
  (For a connection example, see "1.15.5.4 Low-speed Peripherals.")
- The RDY signals of low-speed devices must be set to Wait (which is "L") and Ready (which is "H").
- Access may exceed the data bus width of a low-speed device (e.g., word (32-bit) access to a 16-bit device). If so, the low-speed device access continues through a series of "read -> read and write -> write" operations until all the excess bits have been covered. In this case, the MEM_XCS(0/1) signal is not negated during the access in the "read -> read and write -> write" operations regardless of settings.
- To set the MEM_RDY signal to "L", the signal must be asserted to "L" within 1 cycle (GSSCGCLK) from the falling of the MEM_XCS(0/1) signal.
- For devices (e.g., SRAM memory) not using the RDY function, be sure to set the RDY bit of the corresponding chip select to "0".
- Operation cannot be guaranteed when RDY becomes the "L" or "H" pulse during an access cycle.
- Access with a device using the negation of the MEM_XCS(0/1) signal must have a width within the target width.
- Access where the RDY signal is in the "H" state at the beginning of the access is done in the same way as normal SRAM access.

Table 1-28. Ready Mode

MCFMODE0/1 Register RDY (bit2)	Ready Mode
0	OFF (Default)
1	ON

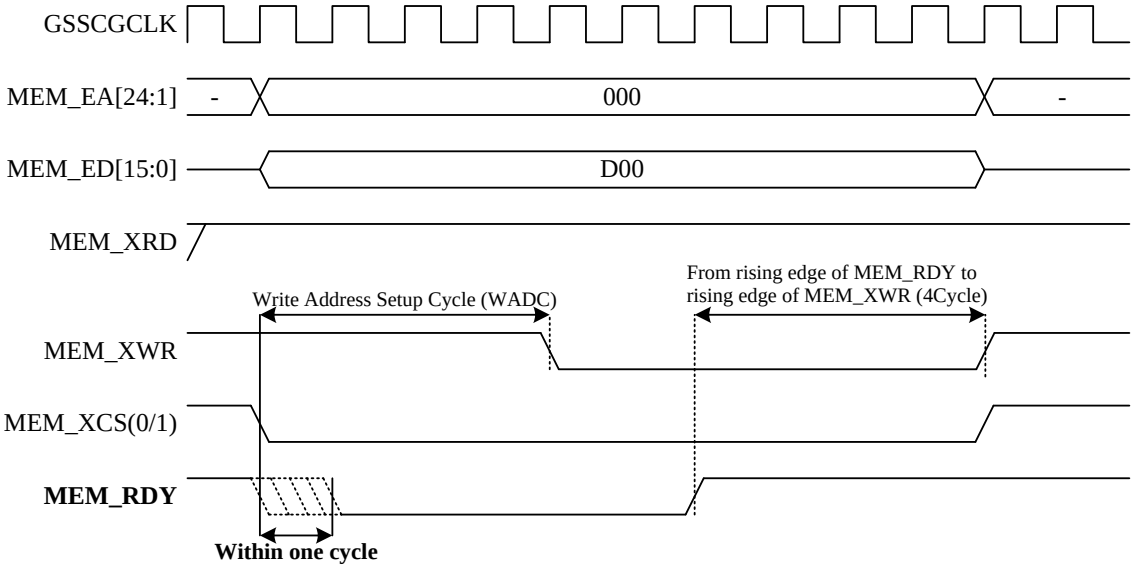
The following figure shows a waveform read from a low-speed device.

Figure 1-81. Read Access from a Low-speed Device



The following figure shows a waveform written to a low-speed device.

Figure 1-82. Write Access to a Low-speed Device



1.15.4.3 External Bus Memory Timing Control

External bus memory timing control is the control of the read and write times of individual control signals of devices connected to external bus memory.

This control can be configured with the MCFTIM0/1 register (MEM_XCS0=01FB_3020$_H$ and MEM_XCS1=01FB_3024$_H$).

Be sure to make these settings before accessing external bus memory.

(For details on the setting timing, see "GDC Access Sequence".)

Read Timing Control

Read timing control is control using the RACC bits, RADC bits, FRADC bits, and RIDLC bits in the MCFTIM0/1 register.

Read Access Cycle

The read access cycle (RACC) sets the number of cycles (GSSCGCLK) required for read access.

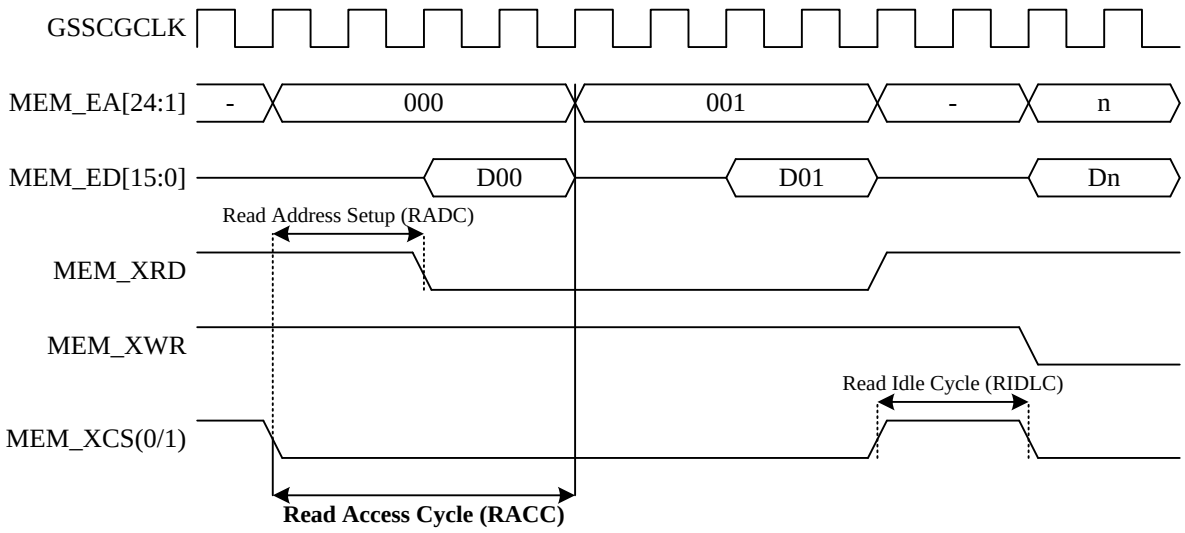
This control is configured with the RACC bits in the MCFTIM0/1 register.

Table 1-29. Read Access Cycle

MCFTIM0/1 Register RACC (bit3 to bit0)	Cycle (GSSCGCLK)
0_H	1 cycle
|	|
F_H	16 cycles (Default)

The following figure shows a waveform of the read access cycle (RACC).

Figure 1-83. Read Access Cycle (RACC)



Read Address Setup Cycle

The read address setup cycle (RADC) sets the number of read address setup cycles (GSSCGCLK) of MEM_XRD. After the number of cycles that is set in the RADC bits, MEM_XRD is asserted based on the falling of MEM_XCS(0/1).

This control is configured with the RADC bits in the MCFTIM0/1 register.

Note the following points about using RADC.

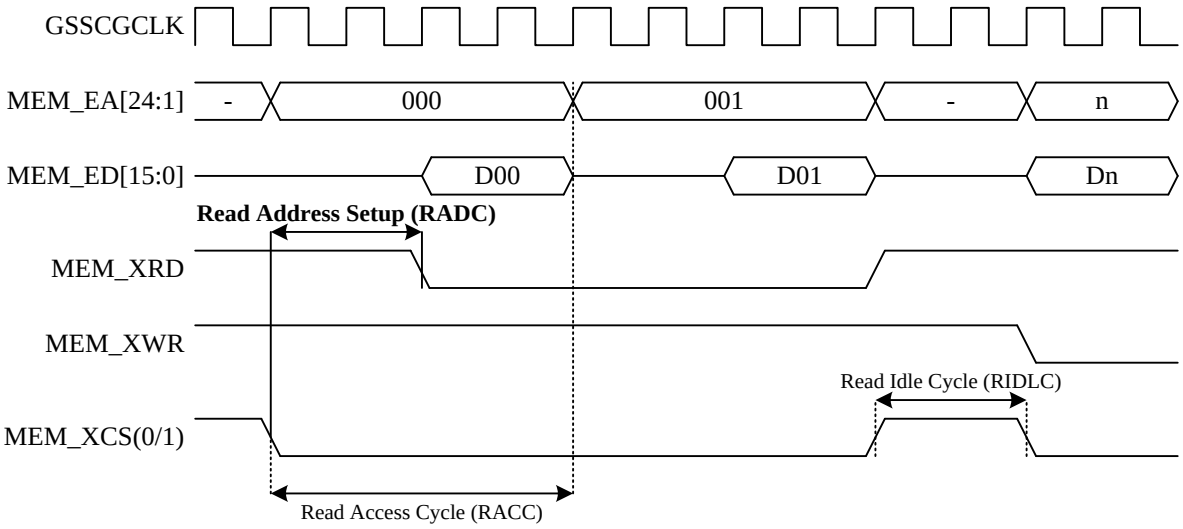
- The value specified in the RADC bits must be within the range of the number of read access cycles (RACC).
- When using NOR Flash page access mode (PAGE bit in the SRAM/Flash mode 0/1 register), set these bits to "0".

Table 1-30. Read Address Setup Cycle

MCFTIM0/1 Register RADC (bit7 to bit4)	Cycle (GSSCGCLK)
0_H	0 cycles (Default)
|	|
F_H	15 cycles

The following figure shows a waveform of the read address setup cycle (RADC).

Figure 1-84. Read Address Setup Cycle (RADC)



First Read Address Cycle

The first read address cycle (FRADC) sets the initial latency for an address (MEM_EA) in NOR Flash page access. Only at the time of first access, the address is maintained in the number of cycles (GSSCGCLK) specified here. Subsequent access is done according to the set number of cycles in the RACC bits. MEM_XCS(0/1) and MEM_XRD are simultaneously asserted in page access. When setting these bits to a value other than "0", specify "0" in the RADC bits.

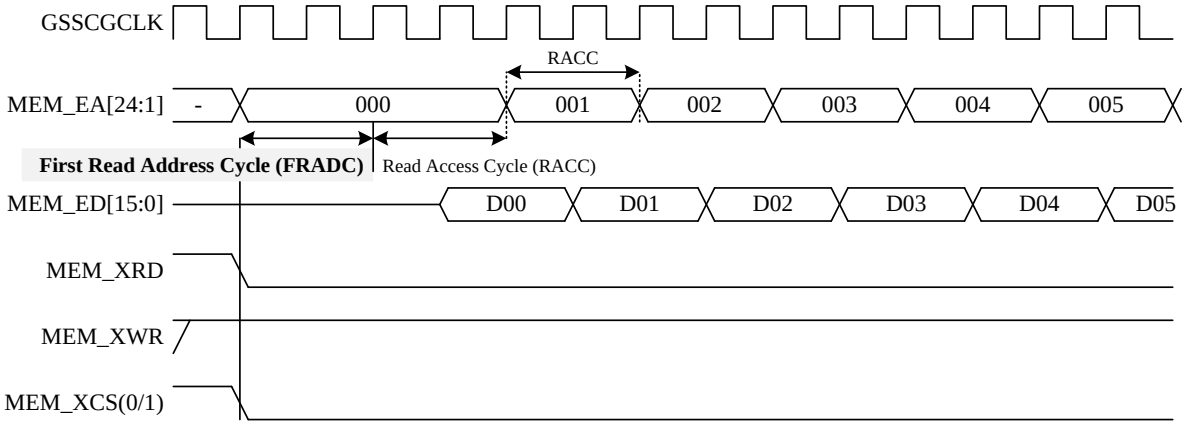
This control is configured with the FRADC bits in the MCFTIM0/1 register.

Table 1-31. First Read Address Cycle

MCFTIM0/1 Register FRADC (bit11 to bit8)	Cycle (GSSCGCLK)
0_H	0 cycles (Default)
|	|
F_H	15 cycles

The following figure shows a waveform of the first read address cycle (FRADC).

Figure 1-85. First Read Address Cycle (FRADC)



Read Idle Cycle

The read idle cycle (RIDLC) sets the number of idle cycles (GSSCGCLK) after read access. These bits are used to prevent the data collisions caused by write access done immediately after read access.

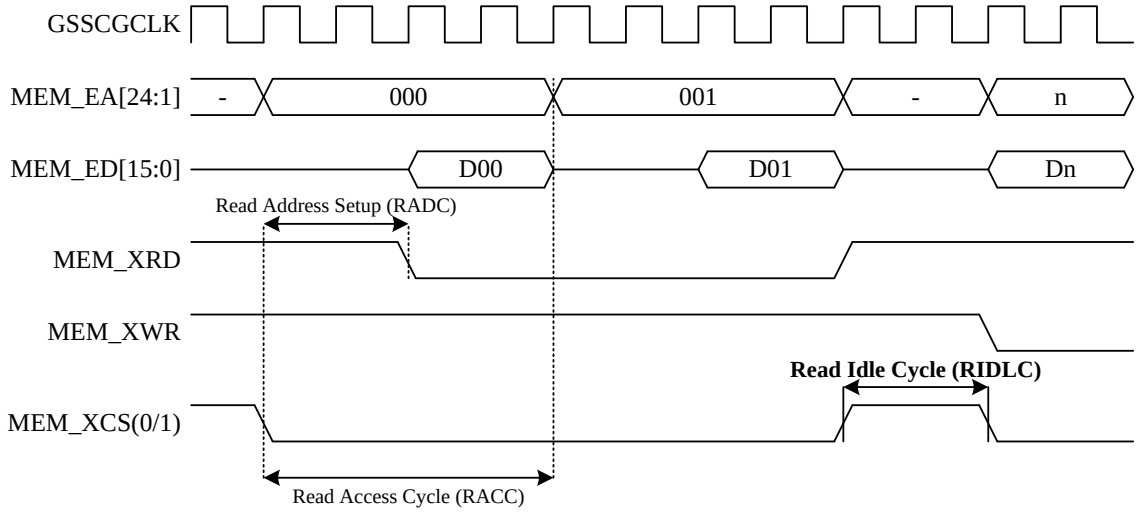
This control is configured with the RIDLC bits in the MCFTIM0/1 register.

Table 1-32. Read Idle Cycle

MCFTIM0/1 Register RIDLC (bit15 to bit12)	Cycle (GSSCGCLK)
0_H	1 cycle
|	|
F_H	16 cycles (Default)

The following figure shows a waveform of the read idle cycle.

Figure 1-86. Read Idle Cycle (RIDLC)



Write Timing Control

Write timing control is the control using the WACC bits, WADC bits, WEC bits, and WIDLC bits in the MCFTIM0/1 register.

Write Access Cycle

The write access cycle (WACC) sets the number of cycles (GSSCGCLK) required for write access.

The address does not change during the cycles specified by these bits.

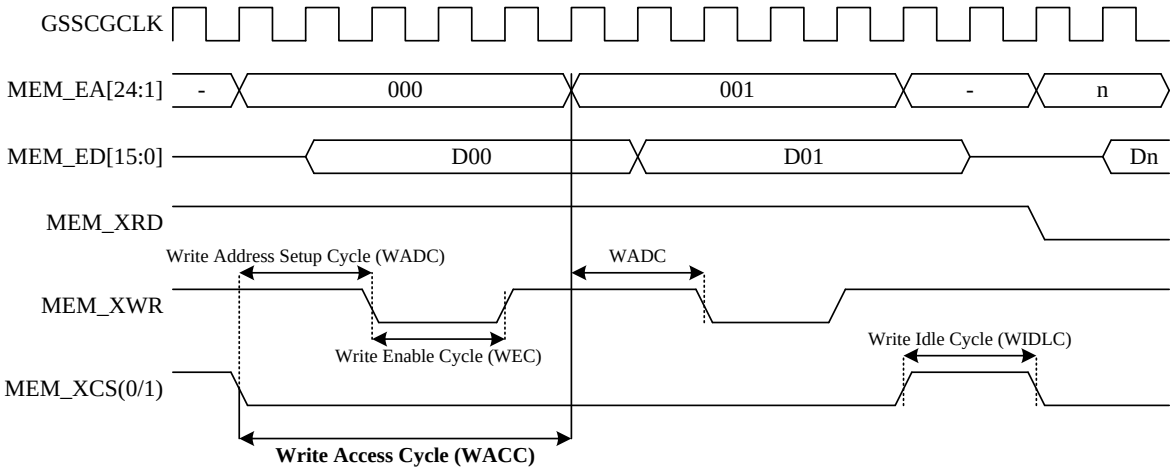
This control is configured with the WACC bits in the MCFTIM0/1 register.

Table 1-33. Write Access Cycle

MCFTIM0/1 Register WACC (bit19 to bit16)	Cycle (GSSCGCLK)
0_H	Reserved
1_H	Reserved
2_H	3 cycles
|	|
F_H	16 cycles (Default)

The following figure shows a waveform of the write access cycle.

Figure 1-87. Write Access Cycle (WACC)



Write Address Setup Cycle

The write address setup cycle (WADC) sets the number of WADC cycles (GSSCGCLK). After the number of cycles that is set in the WADC bits, the MEM_XWR signal is asserted based on the falling of the MEMXCS(0/1) signal.

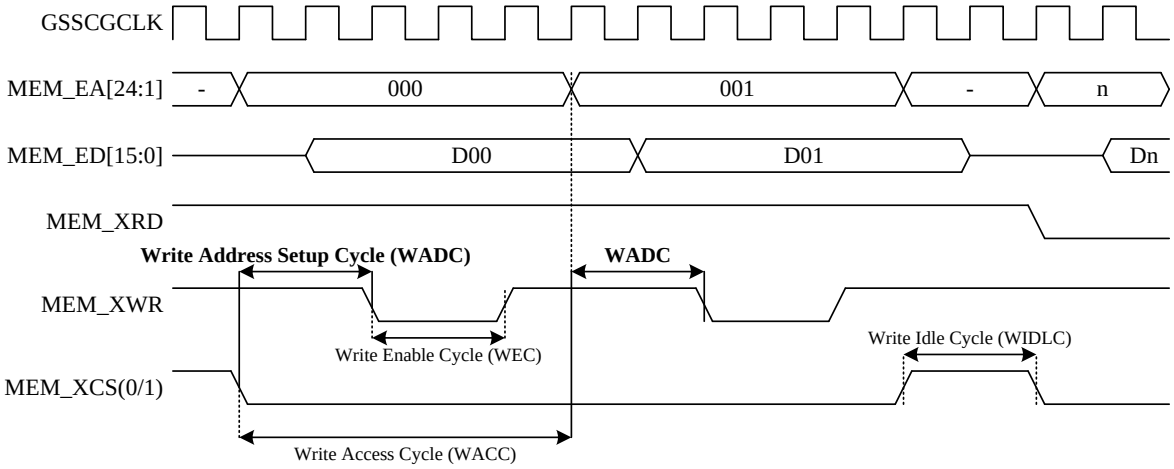
This control is configured with the WADC bits in the MCFTIM0/1 register.

Table 1-34. Write Address Setup Cycle

MCFTIM0/1 Register WADC (bit23 to bit20)	Cycle (GSSCGCLK)
0_H	1 cycle
|	|
5_H	6 cycles (Default)
|	|
E_H	15 cycles
F_H	Reserved

The following figure shows a waveform of the write address setup cycle.

Figure 1-88. Write Address Setup Cycle (WADC)



Write Enable Cycle

The write enable cycle (WEC) sets the number of enable cycles (GSSCGCLK) of the MEM_XWR signal.

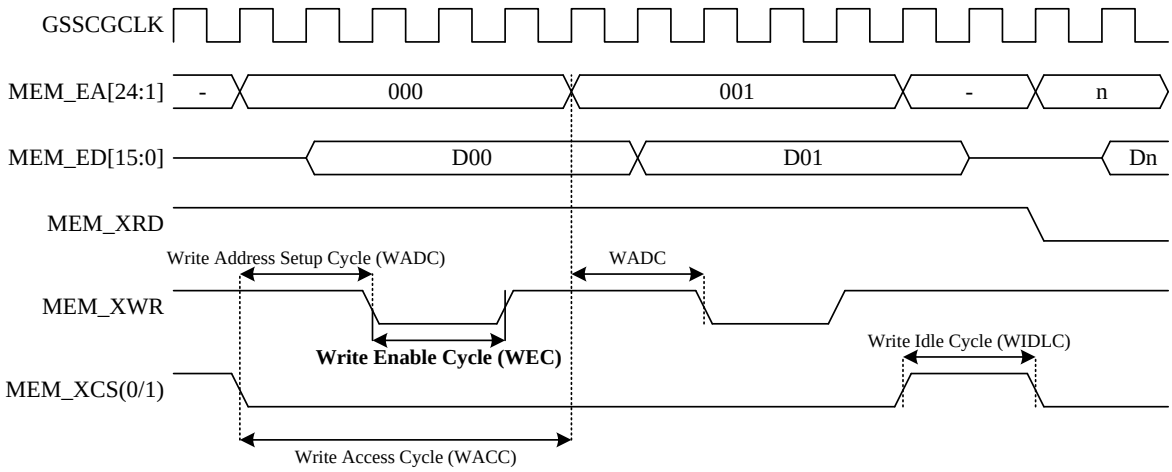
This control is configured with the WEC bits in the MCFTIM0/1 register.

Table 1-35. Write Enable Cycle

MCFTIM0/1 Register WEC (bit27 to bit24)	Cycle (GSSCGCLK)
0_H	1 cycle
|	|
5_H	6 cycles (Default)
|	|
E_H	15 cycles
F_H	Reserved

The following figure shows a waveform of the write enable cycle.

Figure 1-89. Write Enable Cycle (WEC)



Write Idle Cycle

The write idle cycle (WIDLC) sets the number of idle cycles (GSSCGCLK) after write access. When setting the RDY bit in the MCFMODE0/1 register to "1", specify a value of 3 cycles or more.

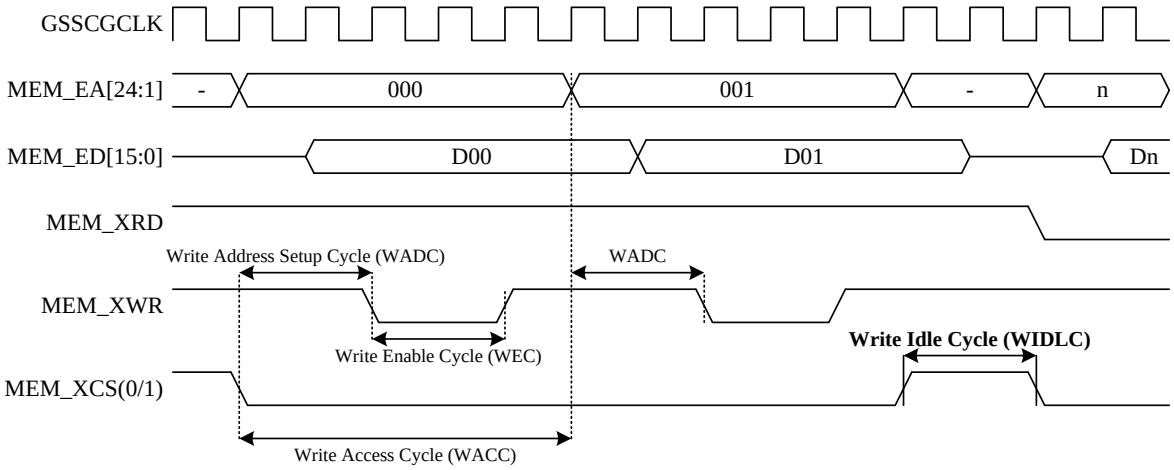
This control is configured with the WIDLC bits in the MCFTIM0/1 register.

Table 1-36. Write Idle Cycle

WCFTIM0/1 Register WIDLC (bit31 to bit28)	Cycle (GSSCGCLK)
0_H	1 cycle (Default)
|	|
F_H	16 cycles

The following figure shows a waveform of the write idle cycle.

Figure 1-90. Write Idle Cycle (WIDLC)



1.15.4.4 External Bus Memory Area

The MEMC module is equipped with an address space of up to 64MB with 2 chip selects (MEM_XCS(0/1)).

(Only 64MB support with chip select 0 (MEM_XCS0) is available using the ADCS bit in the GCONT register of the MCNT module. However, in this case, the chip select 1 (MEM_XCS1) area cannot be used.)

This control can be configured with the MCFAREA0/1 register (MEM_XCS0=01FB_3040$_H$ and MEM_XCS1=01FB_3044$_H$).

The ADDR bits and MASK bits determine the area of each chip select (MEM_XCS(0/1)).

(For details, see "Chip Select Area Settings".)

Be sure to make these settings before accessing external bus memory.

(For details on the setting timing, see "GDC Access Sequence".)

Start Address

This address specifies the start address (ADDR) of each chip select (MEM_XCS(0/1)).

The specified external bus memory area of the GDC macro is from 0200_0000$_H$ to 05FF_FFFF$_H$.

(For a memory map, see "1.4 Memory Maps.")

The ADDR bits are limited to set values according to the MASK bits, with 20$_H$ as the base.

(For details, see Table 1-37.)

The start address setting is made with the ADDR bits in the MCFAREA0/1 register.

Table 1-37. Setting Top Address of the MEM_XCS Area

MCFAREA0/1 Register		Setting Top Address of MEM_XCS Area
MASK (Bit22 to Bit16)	ADDR (Bit7 to Bit0)	
1F$_H$ (32MB)	40$_H$	0400_0000$_H$ (MEM_XCS1: Default)
	20$_H$	0200_0000$_H$ (MEM_XCS0: Default)
0F$_H$ (16MB)	50$_H$	0500_0000$_H$
	40$_H$	0400_0000$_H$
	30$_H$	0300_0000$_H$
	20$_H$	0200_0000$_H$ (MEM_XCS0: Default)
07$_H$ (8MB)	58$_H$	0580_0000$_H$
	50$_H$	0500_0000$_H$
	48$_H$	0480_0000$_H$
	40$_H$	0400_0000$_H$ (MEM_XCS1: Default)
	38$_H$	0380_0000$_H$
	30$_H$	0300_0000$_H$
	28$_H$	0280_0000$_H$
	20$_H$	0200_0000$_H$ (MEM_XCS0: Default)

MCFAREA0/1 Register		Setting Top Address of MEM_XCS Area
MASK (Bit22 to Bit16)	ADDR (Bit7 to Bit0)	
03_H (4MB)	$5C_H$	$05C0_0000_H$
	58_H	0580_0000_H
	54_H	0540_0000_H
	50_H	0500_0000_H
	|	|
	$2C_H$	$02C0_0000_H$
	28_H	0280_0000_H
	24_H	0240_0000_H
	20_H	0200_0000_H (MEM_XCS0: Default)
01_H (2MB)	$5E_H$	$05E0_0000_H$
	$5C_H$	$05C0_0000_H$
	$5A_H$	$05A0_0000_H$
	|	|
	$2E_H$	$02E0_0000_H$
	$2C_H$	$02C0_0000_H$
	$2A_H$	$02A0_0000_H$
	28_H	0280_0000_H
	26_H	0260_0000_H
	24_H	0240_0000_H
	22_H	0220_0000_H
	20_H	0200_0000_H (MEM_XCS0: Default)

MCFAREA0/1 Register		Setting Top Address of MEM_XCS Area
MASK (bit22 to bit16)	ADDR (bit7 to bit0)	
00_H (1MB)	$5F_H$	$05F0_0000_H$
	$5E_H$	$05E0_0000_H$
	|	|
	$2F_H$	$02F0_0000_H$
	$2E_H$	$02E0_0000_H$
	$2D_H$	$02D0_0000_H$
	$2C_H$	$02C0_0000_H$
	$2B_H$	$02B0_0000_H$
	$2A_H$	$02A0_0000_H$
	29_H	0290_0000_H
	28_H	0280_0000_H
	27_H	0270_0000_H
	26_H	0260_0000_H
	25_H	0250_0000_H
	24_H	0240_0000_H
	23_H	0230_0000_H
	22_H	0220_0000_H
	21_H	0210_0000_H
	20_H	0200_0000_H (MEM_XCS0: Default)

Notes:

- If the ADDR bits are set to any value other those in Table 1-37, external bus memory is not normally accessible, so operation is not guaranteed.
- Be sure to set the start address of MEM_XCS0 ($01FB_3040_H$) to 20_H when using the trigger start of the CMDSEQ module.
  If the set address is not 20_H, the trigger start of the CMDSEQ module does not operate normally, so operation is not guaranteed. However, if the trigger start of the CMDSEQ module is not used, the address does not need to be fixed at 20_H.
  (For details on trigger start, see "Trigger Start" of the CMDSEQ module.)
- Be sure to set the start address to 20_H when using MEM_XCS0 in a 64 MB area. If the set start address is not 20_H, external bus memory is not normally accessible, so operation is not guaranteed.
- Configure the areas of the 2 chip selects used and set in each MEM_XCS by the ADDR bits and MASK bits, such that the areas do not overlap. If these chip select areas overlap, external bus memory is not normally accessible, so operation is not guaranteed.

Memory Size

Memory size settings (MASK) correspond to the ADDR bits of the respective chip selects (MEM_XCS(0/1)) .

The MEMC module starts access from the master to external bus memory after determining the access to the corresponding chip select area that is set by the ADDR bits and MASK bits.

(For details, see "Chip Select Area Settings".)

Memory size settings are made with the MASK bits in the MCFAREA0/1 register.

Table 1-38. Memory Size

MCFAREA0/1 Register MASK (bit22 to bit16)	Memory Size
$1F_H$	32 MB (Default)
$0F_H$	16 MB
07_H	8 MB
03_H	4 MB
01_H	2 MB
00_H	1 MB
Other	Prohibition

Notes:

- If the MASK bit setting is Other, external bus memory is not normally accessible, so operation is not guaranteed.
- Be sure to set the MASK bits of both MEM_XCS0 and MEM_XCS1 to 32MB ($1F_H$) when using 64MB MEM_XCS0. If the set value is not 32MB, external bus memory is not normally accessible, so operation is not guaranteed.
- Configure the areas of the 2 chip selects used and set in each MEM_XCS by the ADDR bits and MASK bits, such that the areas do not overlap. If these chip select areas overlap, external bus memory is not normally accessible, so operation is not guaranteed.

Chip Select Area Settings

The ADDR[7:0] bits and MASK[6:0] bits in the MCFAREA0 register (01FB_3040 $_H$) and MCFAREA1 register (01FB_3044 $_H$) set chip select areas of external bus memory.

The ADDR[7:0] bits specify the start address of each chip select (MEM_XCS0, MEM_XCS1) of external bus memory. The MASK[6:0] bits specify the area (memory size) corresponding to the start address specified by the ADDR[7:0] bits. The MEMC module generates internal bus mask addresses by using the ADDR[7:0] bits and MASK[6:0] bits. It also generates external bus mask addresses by using the address accessed from each master (Access address) and the MASK[6:0] bits. After a comparison of these bus mask addresses, access to MEM_XCS(0/1) begins where each of the bus mask addresses matches the other.

The following examples show uses of these settings (where bus mask addresses match or do not match).

The external bus memory area of MEM_XCS0 in these examples is from 0200_0000_H to $03FF_FFFF_H$ (32MB).

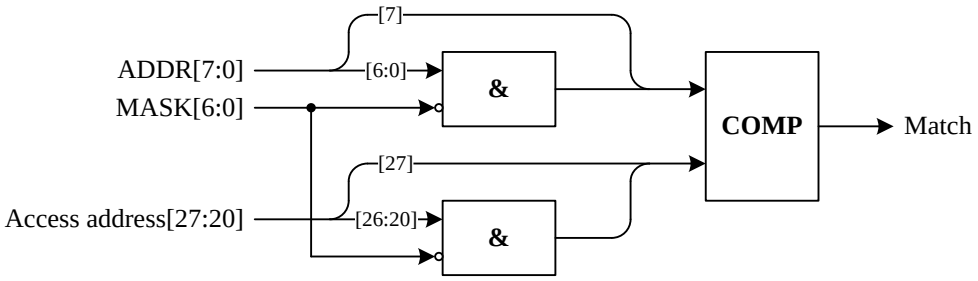
(Example 1) Bus mask addresses match

<MCFAREA0 (01FB_3040$_H$ setting value>

ADDR[7:0] bits = 0010_0000$_B$ (Setting of 0200_0000$_H$)
MASK[6:0] bits = 001_1111$_B$ (Setting of 32 MB)

<Address where the master accesses external bus memory>

Access address[31:0] = 0200_0004$_H$ (Access request of 0200_0004$_H$)



Internal bus mask address = 0010_0000$_B$ is compared with

External bus mask address = 0010_0000$_B$,

and they are found to match, so access begins at the address of 0200_0004$_H$ of external bus memory.

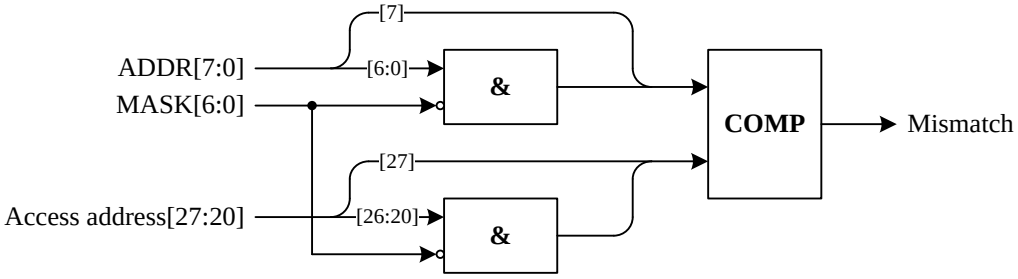
(Example 2) Bus mask addresses do not match

<MCFAREA0/1 setting value>

ADDR[7:0] bit = 0010_0000$_B$ (Setting of 0200_0000$_H$)
MASK[6:0] bits = 001_1111$_B$ (Setting of 32 MB)

<Address where master accesses external bus memory>

Access address[31:0] = 0400_0000$_H$ (Access request of 0400_0000$_H$)



Internal bus mask address = 0010_0000$_B$ is compared with

External bus mask address = 0010_0000$_B$,

and they are found not to match, so external bus memory is not accessed.

1.15.4.5 Error Interrupts

The MEMC module features error interrupts for indicating that areas not mapped to external bus memory were accessed.

Access Error

The MEMC module is equipped with a notification function for SRAM/NOR Flash access errors. The function indicates that an area not mapped to external bus memory was accessed.

The SFER bit in the MCERR register reports such an error.

However, writing "1" can clear the error reported by this bit.

Table 1-39. Access Error

MCERR Register SFER (bit0)	Access Error
0	No error
1	Error

Interrupt Output Control

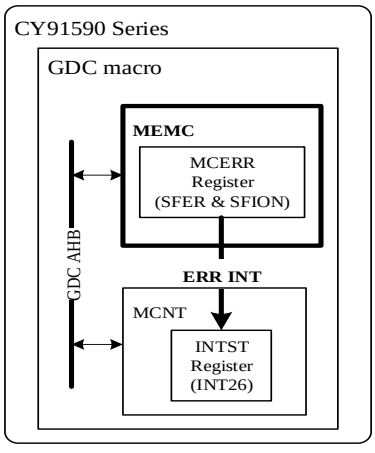
The MEMC module is equipped with a function for notifying the MCNT module of the above-mentioned access error information.

This interrupt output control can be configured with the SFION bit in the MCERR register.

Table 1-40. Interrupt Output Control

MCERR Register SFION (bit2)	Interrupt Output Control
0	OFF
1	ON

Figure 1-91. Interrupt Output to the MCNT Module



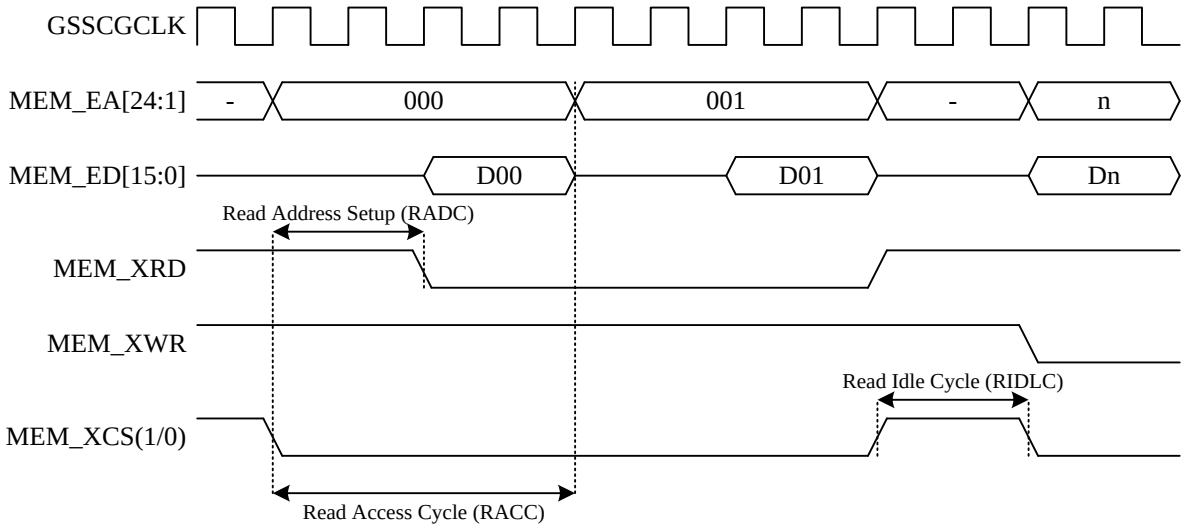
1.15.4.6 External Bus Memory Access

In external bus memory access, MEM_XCS(0/1) is selected with 1-time access. If access with a wider data bus width than that of the target device is requested, the access is converted to continuous access. In continuous access, the MEM_XCS(0/1) value is fixed at "L", and the address is changed. If 32-bit read access is requested from the internal bus for the target device that has a 16-bit width, for example, the address transitions from 0 to 2 with MEM_XCS(0/1) fixed at "L". Then, data is continuously fetched from MEM_ED[15:0] according to the transition time.

If access with a narrower data bus width than that of the target device is requested (e.g., byte access to a 16-bit target device) during a write operation, the byte access is done under the control of the MEM_XWR signal. (The MEMC module outputs only the necessary data.)

Note: The above-mentioned procedure applies to SRAM reading/writing and NOR Flash reading.
Access for writing to NOR Flash uses the same data bus width as that of the target device.
NOR Flash cannot be normally written during access with a different data bus width than that of the target device.

Figure 1-92. External Bus Memory Access



1.15.4.7 Trigger Start of the GDC Macro

The GDC macro can use a trigger start with the CMDSEQ module.

For a trigger start, the command list location address (start address) must be written to the designated address of NOR Flash (MEM_XCS0).

To use a reset start, which is a trigger start function, the MEMC module must be set at the beginning of the command list. (If the initial value of MEMC has no problem, the setting is not necessary.)

Notes:

The following points for using this function.

- Set the ADDR[7:0] bit (MCFAREA0 register) of MEM_XCS0 to 20_H.
  If the set value is not 20_H, trigger start cannot operate normally.
  (The default value is 20_H.)
- To use a reset start, place the command list for the reset start within MEM_XCS0. To extend it to MEM_XCS1, place it such that there is no empty space in the address space with MEM_XCS0.
  (For details, see "Command List Arrangement for a Reset Start".)

The following example shows the setting of 16-bit 64MB NOR Flash.

1. Configure control signal switching of external bus memory of the MEMC module.
   Set the ADCS bit in the GCONT register ($01FB_2500_H$) of the MEMC module to "1".
2. Specify the data bus width of the device connected to MEM_XCS0.
   Set the WDTH bit in the MCFMODE0 register ($01FB_3000_H$) of the MEMC module to "1".
3. Specify the set address of the chip select area corresponding to MEM_XCS0.
   Set the ADDR bits and MASK bits in the MCFMODE0 register ($01FB_3040_H$) of the MEMC module to 20_H and $1F_H$, respectively.
4. Specify the data bus width of the device connected to MEM_XCS1.
   Set the WDTH bit in the MCFMODE1 register ($01FB_3004_H$) of the MEMC module to "1".
5. Specify the set address of the chip select area corresponding to MEM_XCS1.
   Set the ADDR bits and MASK bits in the MCFMODE1 register ($01FB_3044_H$) of the MEMC module to 40_H and $1F_H$, respectively.

Notes:

- Be sure to set the set address of the chip select area of MEM_XCS0 to 0200_0000_H.
- Place the CMDLIST of the reset start such that it does not overlap the MEM_XCS0 area.

1.15.5 Connection Examples

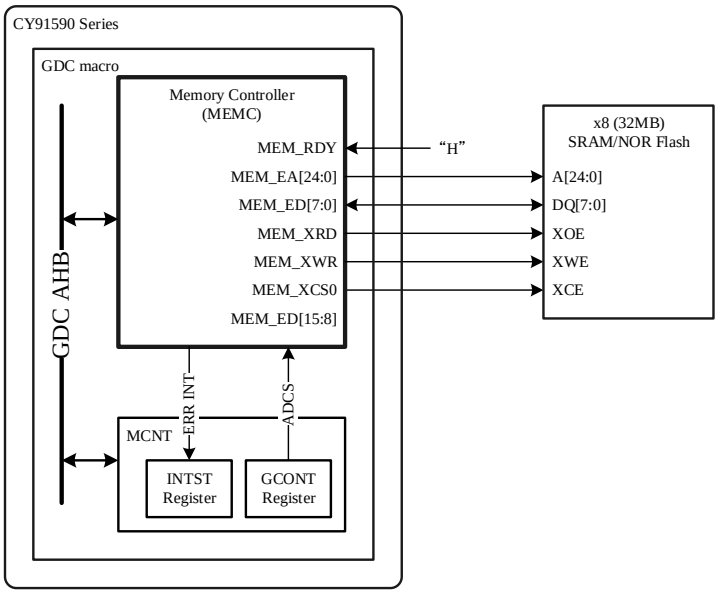
1.15.5.1 8-bit 32MB SRAM/NOR Flash

This section shows an example of the connections of an 8-bit 32MB SRAM/NOR Flash device.

Be sure to note the following points about connecting a single 8-bit (or 16-bit) 32MB SRAM or NOR Flash device.

- Be sure to set the MEM_ED pins to PULL-UP or PULL-DWN in the PPCR01 and PPER01 registers of the FR81S. (For details on the setting pins, see the CY91590 Series Hardware Manual.)
- Connect the MEM_RDY pin to "H."
- Use MEM_XCS0 for the chip select when using the trigger start of the CMDSEQ module.
- Be sure to set the ADDR bits in the MCFAREA0 register (01FB_3040$_H$) to 20$_H$ to use the trigger start of the CMDSEQ module.
- Be sure to set the ADDR bits in the MCFAREA1 register (01FB_3044$_H$) to 1F$_H$ to use the trigger start of the CMDSEQ module. Also, set the MASK bits to 0$_H$.
  (For details on the trigger start of the CMDSEQ module, see "Trigger Start.")

Figure 1-93. Connection Diagram of 8-bit 32MB SRAM/NOR Flash



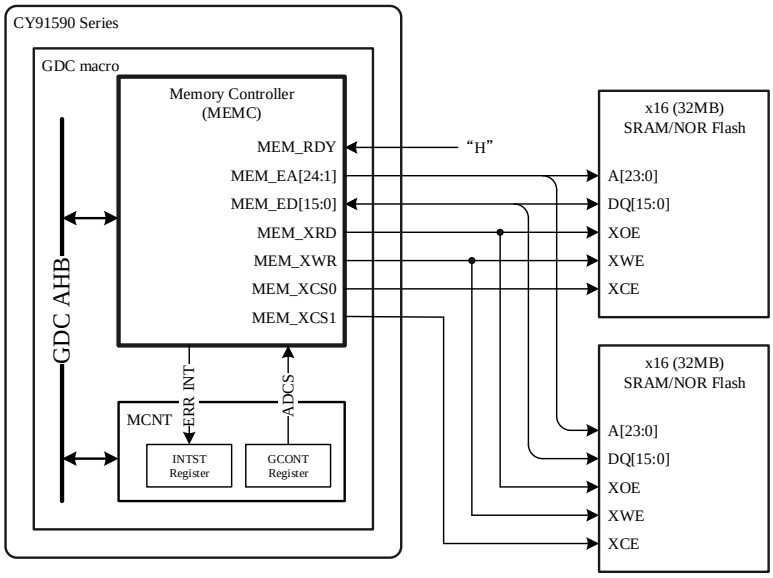
1.15.5.2 Dual 16-bit 32-MB SRAM/NOR Flash Devices

This section shows an example of the connections of 16-bit 32MB SRAM/NOR Flash devices.

Be sure to note the following points about connecting dual 16-bit 32MB SRAM or NOR Flash devices.

- Be sure to set the MEM_ED pins to PULL-UP or PULL-DWN in the PPCR01 and PPER01 registers of the FR81S. (For details on the setting pins, see the CY91590 Series Hardware Manual.)
- Be sure to set the ADDR bits in the MCFAREA0 register (01FB_3040$_H$) to 20$_H$.
- Be sure to set the ADDR bits in the MCFAREA1 register (01FB_3044$_H$) to 40$_H$.
- Be sure to set the ADDR bits in the MCFAREA0 register (01FB_3040$_H$) to 20$_H$ to use the trigger start of the CMDSEQ module with the dual 8-bit 32MB SRAM or NOR Flash devices connected.

Figure 1-94. Connection Diagram of Dual 16-bit 32 MB SRAM/NOR Flash Devices

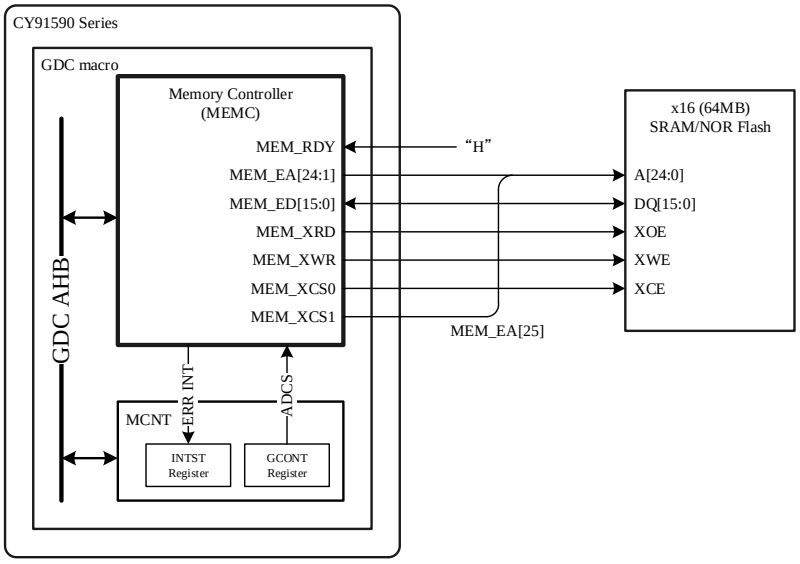


1.15.5.3 16-bit 64-MB SRAM/NOR Flash

Note the following points about connecting a 16-bit (or 8-bit) 64MB SRAM or NOR Flash device.

- Be sure to set the MEM_ED pins to PULL-UP or PULL-DWN in the PPCR01 and PPER01 registers of the FR81S. (For details on the setting pins, see the CY91590 Series Hardware Manual.)
- Be sure to connect MEM_XCS0 for the chip select.
- Be sure to connect MEM_XCS1 to address bit25 of the external device because it is used as MEM_EA[25].
- Be sure to set the ADCS bit in the GCONT ($01FB_2500_H$) register of the MCNT module to 1_B.
- Be sure to set the ADDR bits in the MCFAREA0 ($01FB_3040_H$) register to 20_H.
  Also be sure to set the MASK bits to $1F_H$.
- Be sure to set the ADDR bits in the MCFAREA1 ($01FB_3044_H$) register to 40_H.
  Also be sure to set the MASK bits to $1F_H$.
- Set the same values for both MEM_XCS0 and MEM_XCS1 in each register of the other MEMC module.

Figure 1-95. Connection Diagram of 16-bit 64 MB SRAM/NOR Flash

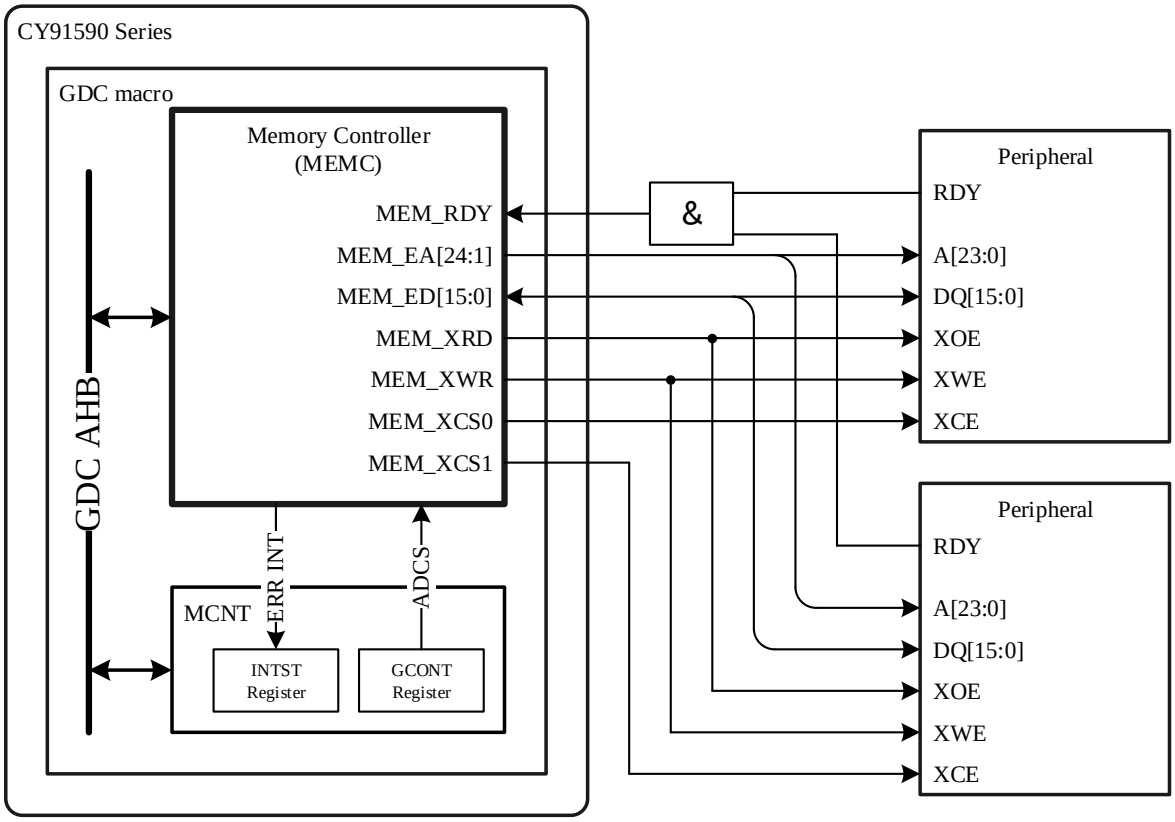


1.15.5.4 Low-speed Peripherals

Note the following points about connecting low-speed peripherals.

- Be sure to set the MEM_ED pins to PULL-UP or PULL-DWN in the PPCR01 and PPER01 registers of the FR81S. (For details on the setting pins, see the CY91590 Series Hardware Manual.)
- Connect MEM_RDY to the low-speed peripherals.
- To connect 2 low-speed peripherals, connect the RDY signals through an AND circuit as shown in the following figure.

Figure 1-96. Connection Diagram of Low-speed Peripherals



1.16 SPI Controller (SPICNT)

This section describes the SPI Controller (referred to below as SPICNT).

1.16.1 Overview

This module has an SPI interface that is used as a Serial Flash memory interface. This interface is controlled exclusively with other external bus memory interfaces. (For details on the exclusive control, see "1.5.1.4 External Bus Memory Interfaces".)

1.16.2 Features

- Slave read mode

  Read access from the GDC macro to the Serial Flash memory area is converted to the read protocol of Serial Flash memory.

- Manual mode

  Commands, addresses, and data are individually set in the Command/Address register and Data registers of this module. Serial Flash memory is directly controlled.

- Master transfer mode
  - ☐ Read

    This module transfers Serial Flash memory data to the GDC macro.

  - ☐ Write

    This module transfers data in memory on the GDC macro to Serial Flash memory.

- SPI FLASH size

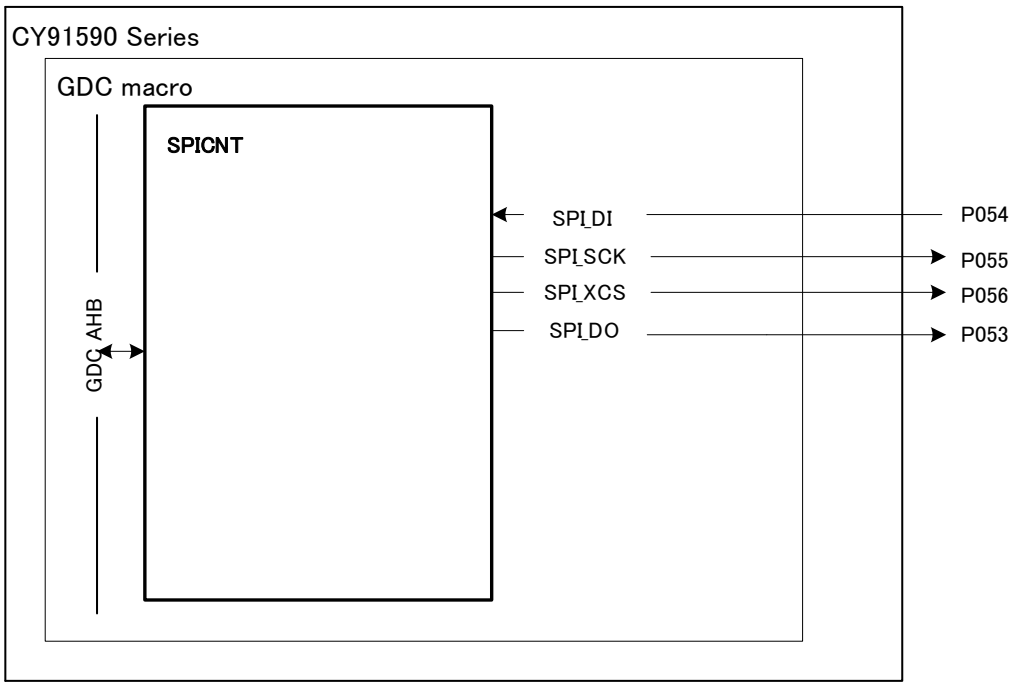
  The module supports Serial Flash memory of up to 128Mbit (16MB).

1.16.3 Configuration

1.16.3.1 Block Diagram

Figure 1-97 is a block diagram showing SPICNT.

Figure 1-97. SPICNT Block Diagram



1.16.4 Registers

1.16.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (**0040_0000$_H$**) is added for access from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.16.4.2 Register List

Address	Register Name	Description
01FB_3000$_H$	Interrupt Status register	Interrupt status register
01FB_3004$_H$	Interrupt Permission register	Interrupt enable/mask setting register
01FB_3008$_H$	Internal Status register	Internal status register
01FB_300C$_H$	Control register	Control register
01FB_3010$_H$	Command/Address register	Command/Address register
01FB_3014$_H$	Data1 register	Data1 register
01FB_3018$_H$	Data2 register	Data2 register
01FB_301C$_H$	Manual Transfer Control register	Manual transfer control register
01FB_3020$_H$	AHB Address register	AHB address register
01FB_3024$_H$	SPI FLASH Address register	Serial Flash address register
01FB_3028$_H$	Master Transfer Control register	Master transfer control register
01FB_302C$_H$	Master Transfer Finish Data Number register	Master transfer completion data number register
01FB_3030$_H$	Master Transfer Guard Release register	Master transfer guard release register

1.16.4.3 Register Details

Interrupt Status Register

This register indicates the interrupt factor status.

Address	01FB_3000$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-											ISR_ MSTGW	ISR_ SRSTISS	ISR_ MNLEND	ISR_ ERRRESP	ISR_ MSTEND
R/W	R											R/W	R/W	R/W	R/W	R/W
Initial	0											0	0	0	0	0

Bit Field		Explanation
No.	**Name**	
0	ISR_MSTEND	Indicates the completion of transfer in master transfer mode. Writing "1" in this bit clears interrupt information. 0: Not interrupted 1: Interrupted
1	ISR_ERRRESP	Indicates an internal transfer error during transfer in master transfer mode. Writing "1" in this bit clears interrupt information. Transfer stops when an error response is received. The necessary processing is not executed when an error response occurs. 0: Not interrupted 1: Interrupted
2	ISR_MNLEND	Indicates the completion of transfer in manual mode. Writing "1" in this bit clears interrupt information. 0: Not interrupted 1: Interrupted
3	ISR_SRSTISS	Indicates the issuance of a soft reset. Writing "1" in this bit clears interrupt information. 0: Not interrupted 1: Interrupted

Bit Field		Explanation
No.	Name	
4	ISR_MSTGW	Indicates that master transfer mode (write) was executed without the guard release code being set. 0: Not interrupted 1: Interrupted

Interrupt Permission Register

This register sets interrupt enable/mask for the Interrupt Status register.

The start of an interrupt with the interrupt mask set by this register causes the mask to stop logical addition interrupt output to the MCNT module. However, the interrupt factor shown in the Interrupt Status register remains.

Address	01FB_3004$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-											IPR_ MSTGW	IPR_ SRSTISS	IPR_ MNLEND	IPR_ ERRRESP	IPR_ MSTEND
R/W	R											R/W	R/W	R/W	R/W	R/W
Initial	0											0	0	0	0	0

Bit Field		Explanation
No.	Name	
0	IPR_MSTEND	Controls to enable logical addition interrupt output of bit0 in the Interrupt Status register (ISR_MSTEN) to the MCNT module. 0: Mask 1: Enable
1	IPR_ERRRESP	Controls to enable logical addition interrupt output of bit1 in the Interrupt Status register (ISR_ERRRESP) to the MCNT module. 0: Mask 1: Enable

Bit Field		Explanation
No.	Name	
2	IPR_MNLEND	Controls to enable logical addition interrupt output of bit2 in the Interrupt Status register (ISR_MNLEND) to the MCNT module. 0 Mask 1 Enable
3	IPR_SRSTISS	Controls to enable logical addition interrupt output of bit3 in the Interrupt Status register (ISR_SRSTISS) to the MCNT module. 0 Mask 1 Enable
4	IPR_MSTGW	Controls to enable logical addition interrupt output of bit4 in the Interrupt Status register (ISR_MSTGW) to the MCNT module. 0 Mask 1 Enable

Internal Status Register

This register indicates the transfer status of SPICNT.

Address	01FB_3008$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-											MNL BUSY	-			MST BUSY
R/W	R											R	R			R
Initial	0											0	0			0

Bit Field		Explanation
No.	**Name**	
0	MSTBUSY	Indicates the status of transfer in master transfer mode. 0: Transfer in master transfer mode not in progress 1: Transfer in master transfer mode in progress
4	MNLBUSY	Indicates the status of transfer to Serial Flash memory in manual mode. 0: Transfer in manual mode not in progress 1: Transfer in manual mode in progress

Control Register

This register controls the Serial Flash I/F, soft reset issuance, and byte swaps.

Address	01FB_300C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-			BYTE SWAP	-			SRST	-			CSHT[4:0]				
R/W	R			R/W	R			R0/W	R			R/W	R/W	R/W	R/W	R/W
Initial	0			0	0			0	0			1	1	1	1	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	CLK_PR[3:0]				SF_CDIV[3:0]				-						TEST[1:0]	
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R						R/W0	R/W0
Initial	1	1	1	1	0	0	[1]	0	0						0	0

[1]: For the CY91F591/2/4/6/7/9, the initial value is "0".
For the CY91F59A/B, the initial value is "1".

Bit Field		Explanation
No.	**Name**	
1-0	TEST[1:0]	Bit for TEST Set this bit field to "00".
11-8	SF_CDIV[3:0]	Sets the input clock division ratio for Serial Flash memory. SPI_SCK[MHz] = GDCSSCGCLK[MHz]/(SF_CDIV+2) Setting the frequency division ratio to an odd number results in a long High period of SPI_SCK. 0_H / 1_H: Prohibited 2_H: Division 1/4 : F_H: Division 1/17 Restrictions: Operation with SF_CDIV[3:0] = 0_H or 1_H cannot be guaranteed, because it is not normal access to external bus memory.
15-12	CLK_PR[3:0]	Sets the period (in units of GDCSSCGCLK) from assertion of SPI_XCS signal output to Serial Flash memory to SPI_SCK operation start. The period is CLK_PR+1.
20-16	CSHT[4:0]	Sets the High retention period (in units of GDCSSCGCLK), when the SPI_XCS signal enters the negate state from the assert state. The period is CSHT+1.
24	SRST	Issues a soft reset. Data transfer in manual mode or master transfer mode stops when a soft reset is issued. Use this bit to discontinue transfer during transfer in manual mode or master transfer mode. This bit is automatically cleared, and "0" is always read. 0: Disable 1: Issue soft reset

Bit Field		Explanation
No.	Name	
28	BYTESWAP	Sets data replacement (byte swap). 0: Setting to replace data with lower byte as on MSB side 1: Setting to replace data with lower byte as on LSB side

Command/Address Register

This register determines the command and address to transfer to Serial Flash memory.

Set the command and address during transfer in manual mode.

Address	$01FB_3010_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	CMD[7:0]								ADDR[23:16]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ADDR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Field		Explanation
No.	Name	
23-0	ADDR[23:0]	Sets the address to transfer to Serial Flash memory.
31-24	CMD[7:0]	Sets the command to transfer to Serial Flash memory.

Data1 Register

This register sets the write data for Serial Flash memory in manual mode. It also stores the data output by Serial Flash memory in manual mode.

Address	$01FB_3014_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA1[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA1[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Field		Explanation
No.	**Name**	
31-0	DATA1[31:0]	Sets the data to write to Serial Flash memory. The register also stores the data output by Serial Flash memory.

Data2 Register

This register stores the data output by Serial Flash memory in manual mode.

Address	$01FB_3018_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DATA2[31:16]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	DATA2[15:0]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Field		Explanation
No.	**Name**	
31-0	DATA2[31:0]	Stores the data output by Serial Flash memory.

Manual Transfer Control Register

This register sets the address transfer size and the data to transfer to Serial Flash memory when the manual command is issued. It also sets the start of transfer in manual mode.

Address	01FB_301C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															MNL_ START
R/W	R															R0/W
Initial	0															0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-										ADDR_SIZE[1:0]		-	DATA_SIZE[2:0]		
R/W	R										R/W	R/W	R	R/W	R/W	R/W
Initial	0										0	0	0	0	0	0

Bit Field		**Explanation**
No.	**Name**	
2-0	DATA_SIZE[2:0]	Sets the size of the data (in units of bytes) to transfer to Serial Flash memory when the manual command is issued. These bits set the size of data transfer (in units of bytes) when data is read from Serial Flash memory in manual mode. The maximum valid value is 101_B (5 bytes).
5-4	ADDR_SIZE[1:0]	Sets the transfer size of the address (in units of bytes) to transfer to Serial Flash memory when the manual command is issued.
16	MNL_START	Starts transfer in manual mode. This bit is automatically cleared, and "0" is always read. 0: Disable 1: Start transfer in manual mode

AHB Address Register

This register specifies the address, in the GDC macro, to access in master transfer mode.

Access in master transfer mode begins at the address specified here.

Address	01FB_3020$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	AHB_ADDR[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	AHB_ADDR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Field		**Explanation**
No.	**Name**	
31-0	AHB_ADDR[31:0]	Sets the address to transfer to the GDC macro. The set value for the address value is always in units of 4 bytes. (It cannot be written in AHB_ADDR[1:0].)

SPI FLASH Address Register

This register specifies the Serial Flash memory address to access in master transfer mode.

Access in master transfer mode begins at the address specified here.

Address	01FB_3024$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-								SPI FLASH_ADDR[23:16]							
R/W	R								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0								0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SPI FLASH_ADDR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Field		**Explanation**
No.	**Name**	
23-0	SPI FLASH_ ADDR[23:0]	Sets the address to transfer to Serial Flash memory.

Master Transfer Control Register

This register determines the transfer size of master transfer mode. It also sets the transfer direction (GDC macro -> Serial Flash memory, or Serial Flash memory -> GDC macro) in master transfer mode.

Transfer begins in master transfer mode.

Address	$01FB_3028_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-	MST_ START	-	WXR	-			SIZE[24:16]								
R/W	R	R0/W	R	R/W	R			R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0			0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SIZE[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Field		**Explanation**
No.	**Name**	
24-0	SIZE[24:0]	Sets the transfer amount (in units of bytes) in master transfer mode. The set transfer amount is always in units of 4 bytes. (It cannot be written in SIZE[1:0].)
28	WXR	Sets the transfer direction. 0 : Read direction (Serial Flash memory -> GDC macro) 1 : Write direction (GDC macro -> Serial Flash memory)
30	MST_START	Starts transfer in master transfer mode. If the guard release code is not set in the Master Transfer Guard Release register, the transfer direction becomes the Write direction, and this bit does not start transfer. For transfer in the Write direction, set the guard release code without error in the Master Transfer Guard Release register in advance. Transfer in the Read direction is possible even when the guard release code is not set. If transfer in the Write direction begins without the guard release code being set, an interrupt is generated. This bit is automatically cleared, and "0" is always read. 0 : Disable 1 : Start transfer

Master Transfer Finish Data Number Register

This register shows the transfer completion data count in master transfer mode.

Address	01FB_302C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-							FINDATA[24:16]								
R/W	R							R	R	R	R	R	R	R	R	R
Initial	0							0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	FINDATA[15:0]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Field		Explanation
No.	**Name**	
24-0	FINDATA[24:0]	Shows the transfer completion data count (in units of bytes) in master transfer mode. This bit is cleared at the start of transfer by master transfer.

Master Transfer Guard Release Register

This register sets the guard release of master transfer mode. Unless a specific guard release code is input to the register, no writing can be done in master transfer mode.

Address	01FB_3030$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	MSTGR[31:16]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	MSTGR[15:0]															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit Field		Explanation
No.	**Name**	
31-0	MSTGR[31:0]	Sets the guard release code[1]. [1]: The guard release code is 5752_454E$_H$. This bit is cleared in any of the following cases: When transfer by master transfer (Write) is completed When a soft reset is issued during transfer by master transfer (Write) When an internal transfer error is received during transfer by master transfer (Write)

1.16.5 Explanation of Operation

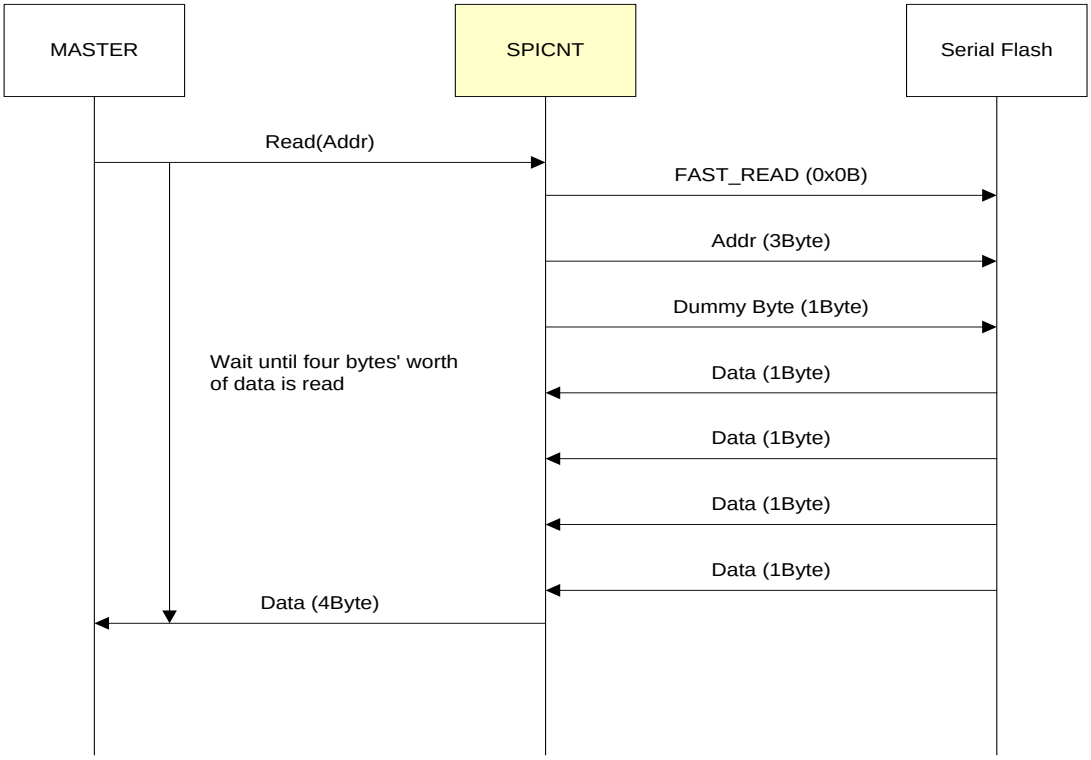
1.16.5.1 Slave Read Mode

Notes:

- If data is transferred using slave read mode during a period of data transfer using master transfer mode, the data transfer using master transfer mode is completed before data is transferred using slave read mode.
- If data is transferred using slave read mode during a period of data transfer using manual mode, the data transfer using manual mode is completed before data is transferred using slave read mode.

The following figure shows the slave read mode flow.

Figure 1-98. Slave Read Mode Flow



The control flow of slave read mode is as follows.

- MASTER executes read access to SPICNT.
- To read data from Serial Flash memory, SPICNT issues the FAST_READ command, an address, and a dummy byte.
- SPICNT allows GDC AHB to wait until 4 bytes of data are read from Serial Flash memory.

1.16.5.2 Manual Mode

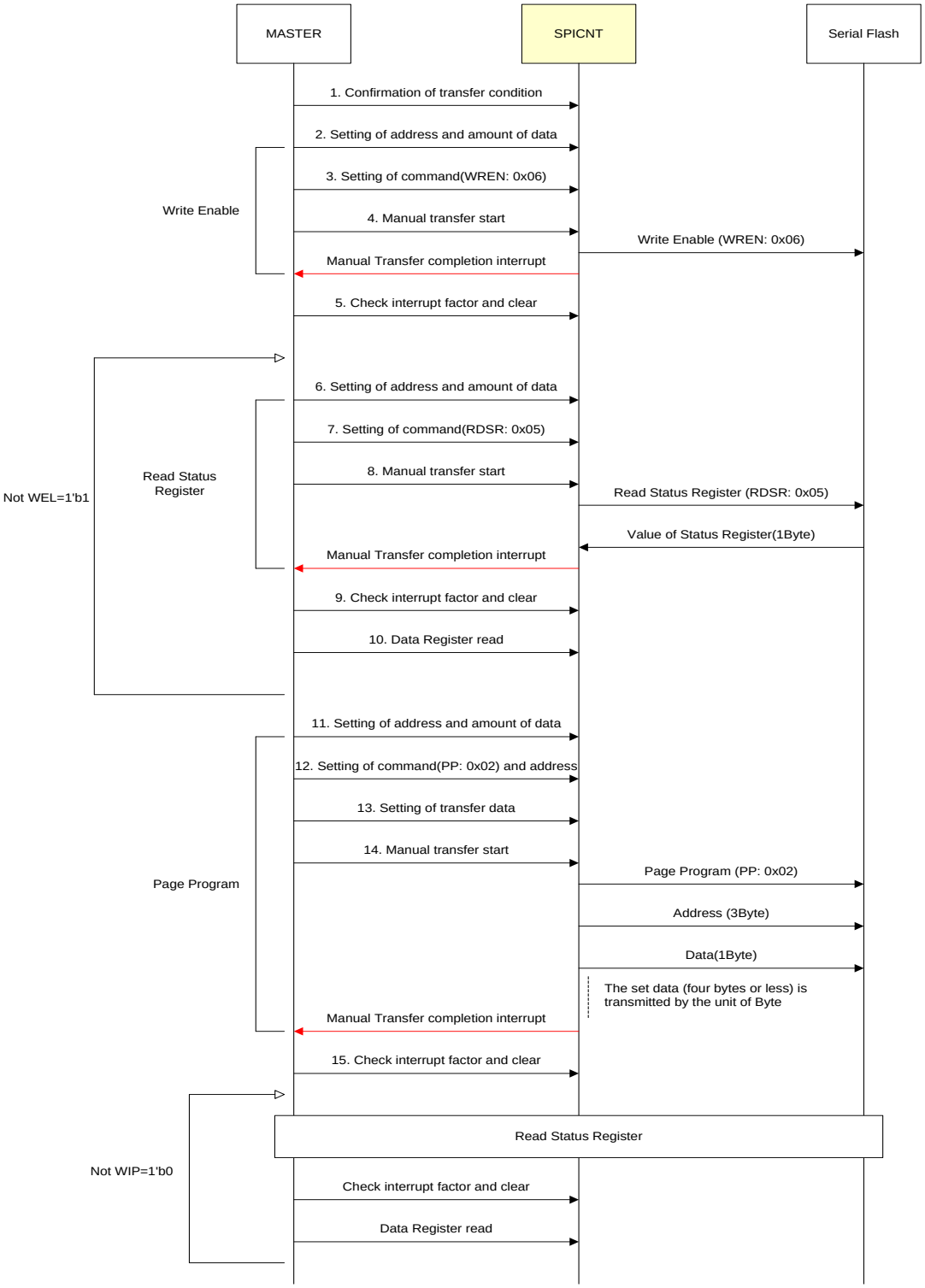
Manual mode sets commands, addresses, and data in the Command/Address register (01FB_300C$_H$) and Data1 register (01FB_3014$_H$), respectively, of this module to directly control Serial Flash memory.

Note: Do not change any register setting other than the SRST bit in the Control register (01FB_300C$_H$) during transfer here.

Write

The following figure shows the flow for manually writing in Serial Flash memory.

Figure 1-99. Manual Write Flow



The control flow of manual writing is as follows:

1. Confirm with the Internal Status register (01FB_3008$_H$) that manual or master transfer mode is not being executed.
2. Set the sizes of the address and data to transfer in the Manual Transfer Control register (01FB_301C$_H$).[1]
   [1]: Set each transfer size to 0bytes because the WREN command does not require address and data transfer.
3. Set the WREN command in the Command/Address register (01FB_3010$_H$).
4. The Manual Transfer Control register (01FB_301C$_H$) starts manual transfer.
   A manual transfer completion interrupt is issued when transfer of the designated size is completed.
5. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
   The target of the check is bit2 in the Interrupt Status register.
6. Set the sizes of the address and data to transfer in the Manual Transfer Control register (01FB_301C$_H$).[2]
   [2]: The RDSR command does not require address transfer, and it reads 1byte of data from Serial Flash memory.
   Therefore, set the address transfer size to 0 bytes and the data transfer size to 1byte.
7. Set the RDSR command in the Command/Address register (01FB_3010$_H$).
8. The Manual Transfer Control register (01FB_301C$_H$) starts manual transfer.
   A manual transfer completion interrupt is issued when transfer of the designated size is completed.
9. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
   The target of the check is bit2 in the Interrupt Status register.
10. With the reading of data by the RDSR command, data is stored in the Data1 register (01FB_3014$_H$).
    Check whether the WEL bit is "1". If the WEL bit is not "1", repeat steps 5 to 8.
11. Set the sizes of the address and data to transfer in the Manual Transfer Control register (01FB_301C$_H$).
12. Set the PP command and address in the Command/Address register (01FB_3010$_H$).
13. Set the data written to Serial Flash memory in the Data1 register (01FB_3014$_H$).
14. The Manual Transfer Control register (01FB_301C$_H$) starts manual transfer.
    A manual transfer completion interrupt is issued when transfer of the designated size is completed.
15. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
    The target of the check is bit2 in the Interrupt Status register.
16. Read the Serial Flash memory status by issuing the RDSR command. Check whether the WIP bit is "0". If the WIP bit is not "0", repeat this step.

Read

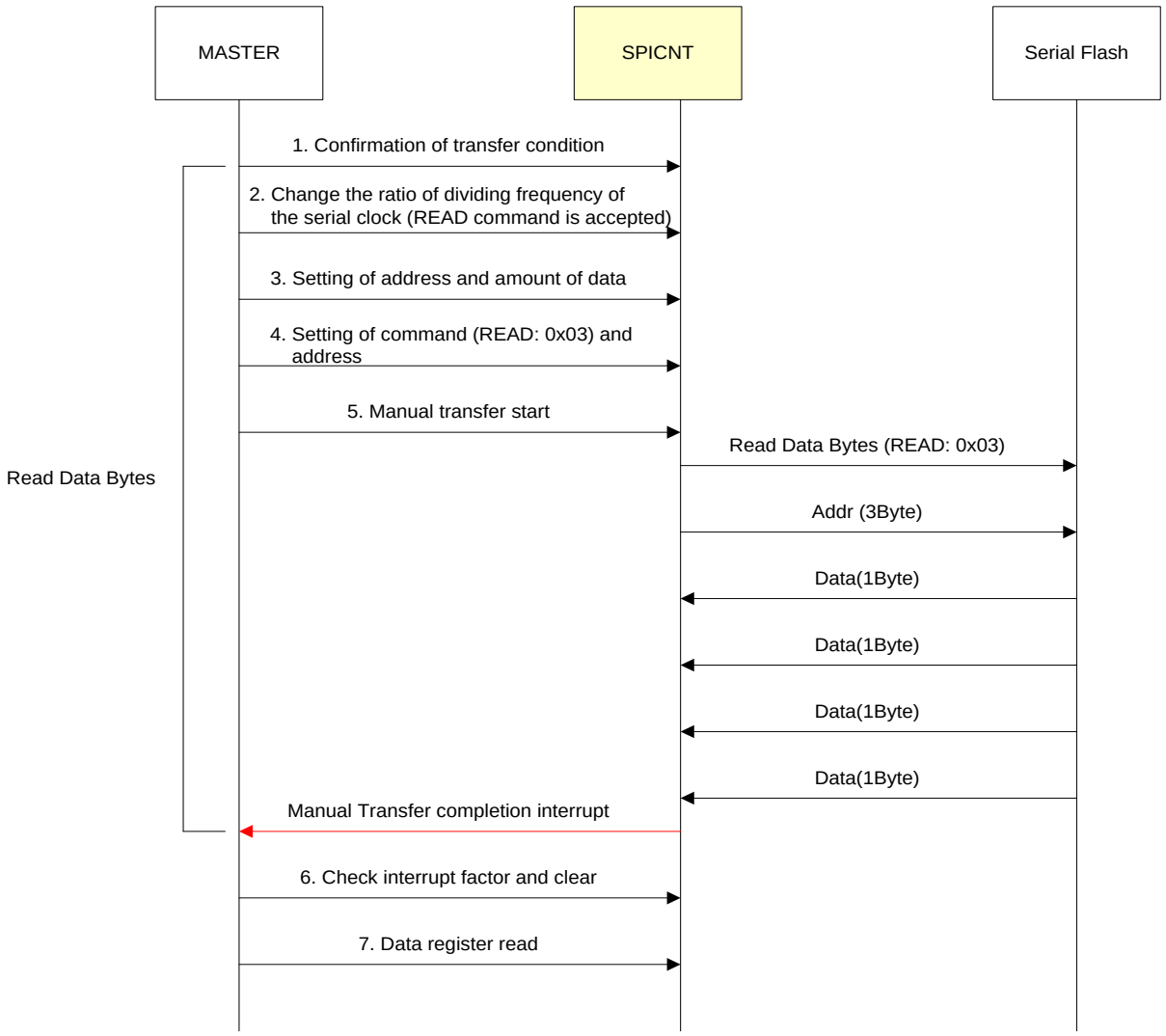
This mode includes only processing in units of 1 word. Use master mode for processing of 2 words or more.

Notes:

- The FAST_READ command cannot be issued in manual reading. That is executed by the READ command.
- Change the division ratio of the serial clock frequency setting so that the READ command is accepted.

The following figure shows the flow for manually reading out Serial Flash memory.

Figure 1-100. Manual Read Flow



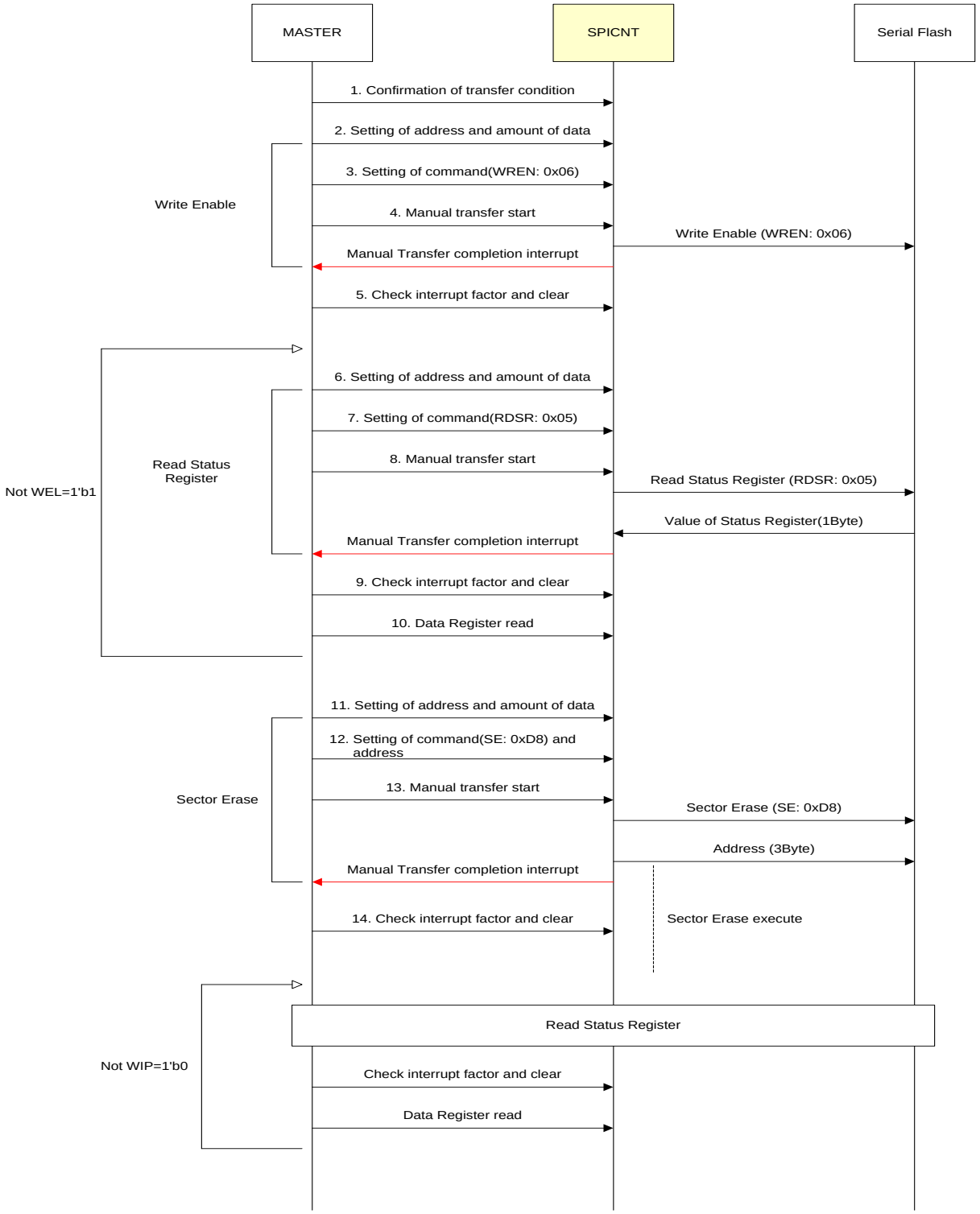
The control flow of manual reading is as follows:

1. Confirm with the Internal Status register (01FB_3008$_H$) that manual or master mode is not being executed.
2. Set the frequency division ratio of the serial clock to be acceptable to the READ command, in the Control register (01FB_300C$_H$).
3. Set the sizes of the address and data to transfer in the Manual Transfer Control register (01FB_301C$_H$).[1]
   [1]: Set the address transfer size to 3bytes and the data transfer size to 4 bytes because the READ command reads 4bytes of data from Serial Flash memory.
4. Set the READ command and address in the Command/Address register (01FB_3010$_H$).
5. The Manual Transfer Control register (01FB_301C$_H$) starts manual transfer.
   A manual transfer completion interrupt is issued when transfer of the designated size is completed.
6. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
   The target of the check is bit2 in the Interrupt Status register.
7. With the reading of data by the READ command, data is saved in the Data register (01FB_3014$_H$). Check the contents.

Sector Erase

The following figure shows the flow for manually erasing a sector of Serial Flash memory.

Figure 1-101. Manual Sector Erase Flow



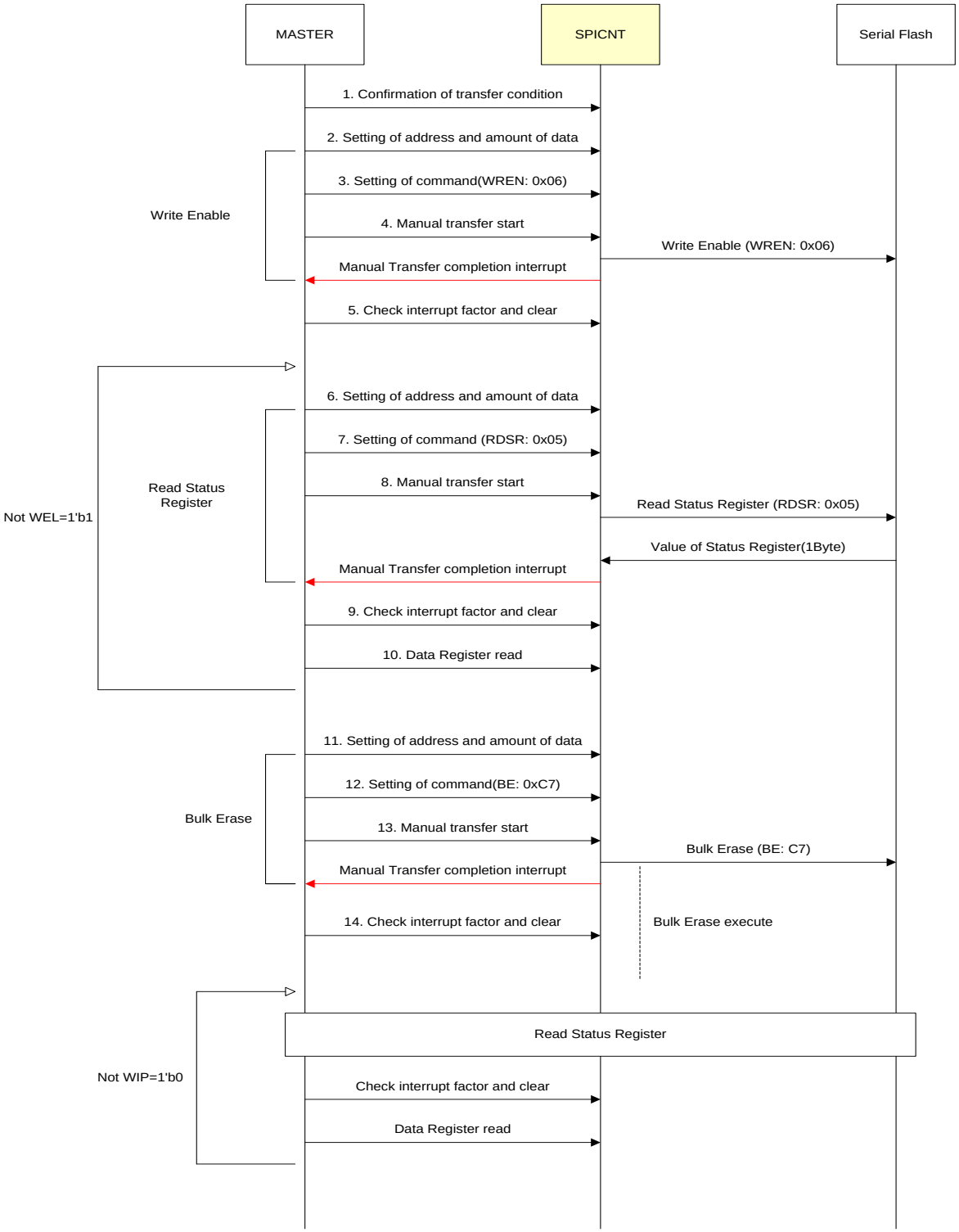
The control flow of manually erasing a sector is as follows:

1. Confirm with the Internal Status register (01FB_3008$_H$) that manual or master mode is not being executed.
2. Set the sizes of the address and data to transfer in the Manual Transfer Control register (01FB_301C$_H$).[1]
   [1]: Set each transfer size to 0bytes because the WREN command does not require address and data transfer.
3. Set the WREN command in the Command/Address register (01FB_3010$_H$).
4. The Manual Transfer Control register (01FB_301C$_H$) starts manual transfer.
   A manual transfer completion interrupt is issued when transfer of the designated size is completed.
5. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
   The target of the check is bit2 in the Interrupt Status register.
6. Set the sizes of the address and data to transfer in the Manual Transfer Control register (01FB_301C$_H$).[2]
   [2]: The RDSR command does not require address transfer, and it reads 1byte of data from Serial Flash memory. Therefore, set the address transfer size to 0bytes and the data transfer size to 1byte.
7. Set the RDSR command in the Command/Address register (01FB_3010$_H$).
8. The Manual Transfer Control register (01FB_301C$_H$) starts manual transfer.
   A manual transfer completion interrupt is issued when transfer of the designated size is completed.
9. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
   The target of the check is bit2 in the Interrupt Status register.
10. With the reading of data by the RDSR command, data is saved in the Data register (01FB_3014$_H$). Check whether the WEL bit is "1". If the WEL bit is not "1", repeat steps 5 to 8.
11. Set the sizes of the address and data to transfer in the Manual Transfer Control register (01FB_301C$_H$).[3]
    [3]: Set the address transfer size to 3bytes and the data transfer size to 0 bytes because the SE command does not require data transfer.
12. Set the SE command and address in the Command/Address register (01FB_3010$_H$).
13. The Manual Transfer Control register (01FB_301C$_H$) starts manual transfer.
    A manual transfer completion interrupt is issued when transfer of the designated size is completed.
    The sector is erased.
14. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
    The target of the check is bit2 in the Interrupt Status register.
15. Read the Serial Flash memory status by issuing the RDSR command. Check whether the WIP bit is "0". If the WIP bit is not "0", repeat this step.

Bulk Erase

The following figure shows the flow for manually bulk erasing sectors of Serial Flash memory.

Figure 1-102. Manual Bulk Erase Flow



The control flow of manual bulk erasing is as follows:

1. Confirm with the Internal Status register (01FB_3008$_H$) that manual or master mode is not being executed.
2. Set the sizes of the address and data to transfer in the Manual Transfer Control register (01FB_301C$_H$).[1]
   [1]: Set each transfer size to 0bytes because the WREN command does not require address and data transfer.
3. Set the WREN command in the Command/Address register (01FB_3010$_H$).
4. The Manual Transfer Control register (01FB_301C$_H$) starts manual transfer.
   A manual transfer completion interrupt is issued when transfer of the designated size is completed.
5. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
   The target of the check is bit2 in the Interrupt Status register.
6. Set the sizes of the address and data to transfer in the Manual Transfer Control register (01FB_3001C$_H$). [2]
   [2]: The RDSR command does not require address transfer, and it reads 1byte of data from Serial Flash memory.
   Therefore, set the address transfer size to 0bytes and the data transfer size to 1byte.
7. Set the RDSR command in the Command/Address register (01FB_3010$_H$).
8. The Manual Transfer Control register (01FB_301C$_H$) starts manual transfer.
   A manual transfer completion interrupt is issued when transfer of the designated size is completed.
9. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
   The target of the check is bit2 in the Interrupt Status register.
10. With the reading of data by the RDSR command, data is saved in the Data register (01FB_3014$_H$). Check whether the WEL bit is "1". If the WEL bit is not "1", repeat steps 5 to 8.
11. Set the sizes of the address and data to transfer in the Manual Transfer Control register (01FB_301C$_H$). [3]
    [3]: Set the address and data transfer sizes to 0 bytes because the BE command does not require data transfer.
12. Set the BE command in the Command/Address register (01FB_3010$_H$).
13. The Manual Transfer Control register (01FB_301C$_H$) starts manual transfer.
    A manual transfer completion interrupt is issued when transfer of the designated size is completed.
    Bulk erase is executed.
14. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
    The target of the check is bit2 in the Interrupt Status register.
15. Read the Serial Flash memory status by issuing the RDSR command. Check whether the WIP bit is "0". If the WIP bit is not "0", repeat this step.

1.16.5.3 Master Transfer Mode

Write

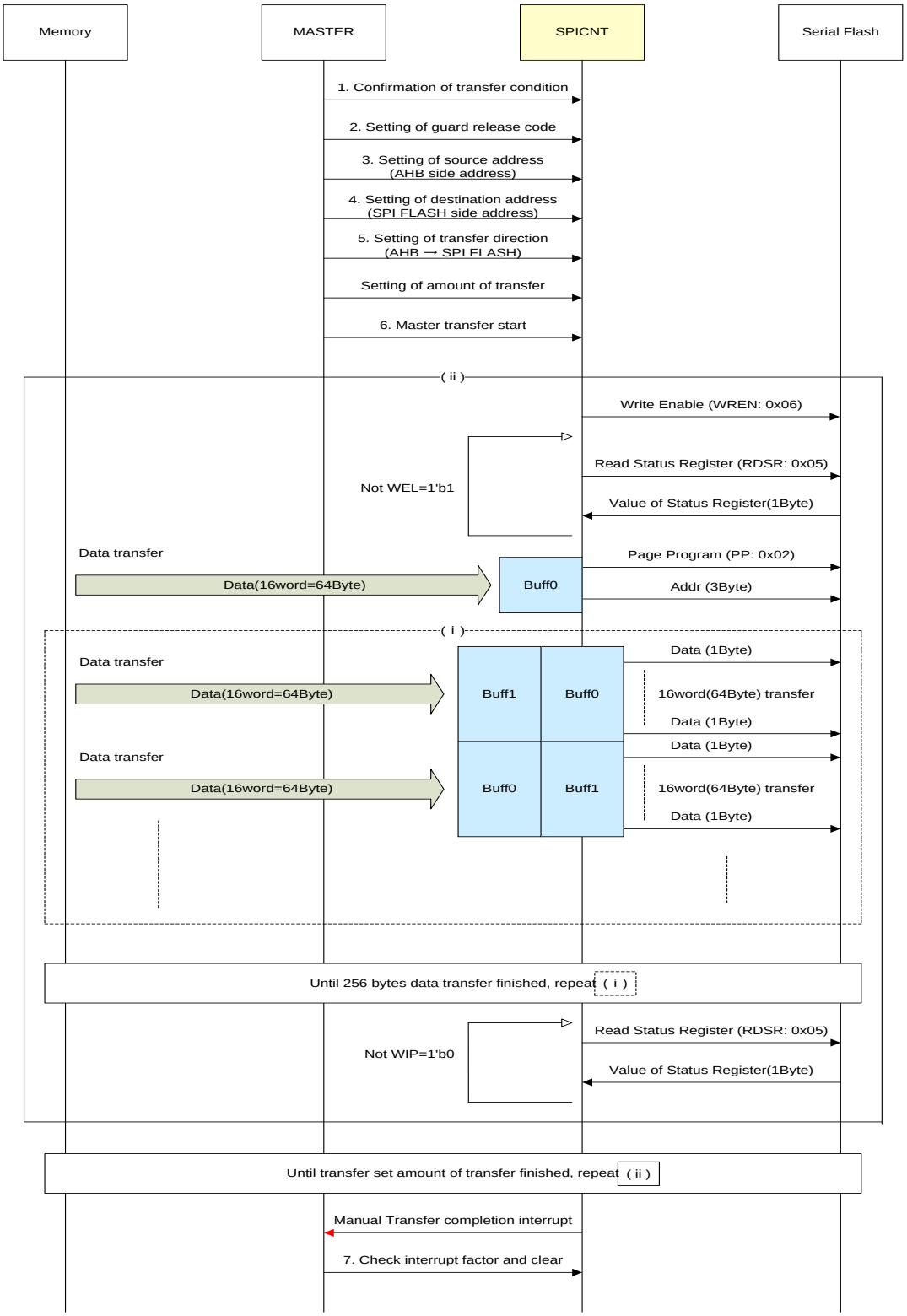
After the source address (GDC macro side), destination address (Serial Flash memory side), transfer direction, and transfer size are set, the start bit starts transfer. A master transfer completion interrupt is issued when the transfer is completed.

Notes:

- Do not change any register setting other than the SRST bit in the Control register (01FB_300C$_H$) during this transfer.
- If a soft reset is issued during transfer in master transfer mode (write), Serial Flash memory may be processing the PP command. In this case, check the status of Serial Flash memory to confirm that it is not in the BUSY status, and then execute access as follows.

The following figure shows the master write flow.

Figure 1-103. Master Write Flow



The control flow of master writing is as follows.

1. Confirm with the Internal Status register (01FB_3008$_H$) that manual or master mode is not being executed.
2. Set the guard release code in the Master Transfer Guard Release register (01FB_3030$_H$).
3. Set the source address in the AHB Address register (01FB_3020$_H$).
4. Set the destination address in the SPI FLASH memory address register (01FB_3024$_H$).
5. Set the master data transfer direction (write: GDC macro -> Serial Flash memory) and transfer size in the Master Transfer Control register (01FB_3028$_H$).
6. The Master Transfer Control register (01FB_3028$_H$) starts master write transfer.
    a. Transfer the WREN command to Serial Flash memory.
    b. Transfer the RDSR command to Serial Flash memory, and check the read data contents (whether the WEL bit is "1"). If WEL is not "1", repeat this step.
    c. Transfer the PP command and destination address that was set in step 4.
    At the same time, transfer 16 words (or 64bytes) of data to the internal buffer (Buff0) from the source address that was set in step 3.
    d. The data stored in Buff0 is sent to Serial Flash memory when data transfer to Buff0 is completed. At the same time, data is transferred from the AHB side to Buff1.
    e. The data stored in Buff1 is sent to Serial Flash memory when data transfer to Buff1 is completed. At the same time, data is transferred from the AHB side to Buff0.
    f. Repeat steps 1 to 5 until the completion of the transfer of 256bytes of data from the GDC macro side to Serial Flash memory.
    g. Transfer the RDSR command to Serial Flash memory, and check the read data contents (whether the WIP bit is "0"). If WIP is not "0", repeat this step.
    h. Repeat steps 1 to 7 until the completion of the transfer that was set in step 5.
    i. A master transfer completion interrupt is issued when transfer of the designated size is completed.
7. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
    The target of the check is bit0 in the Interrupt Status register.

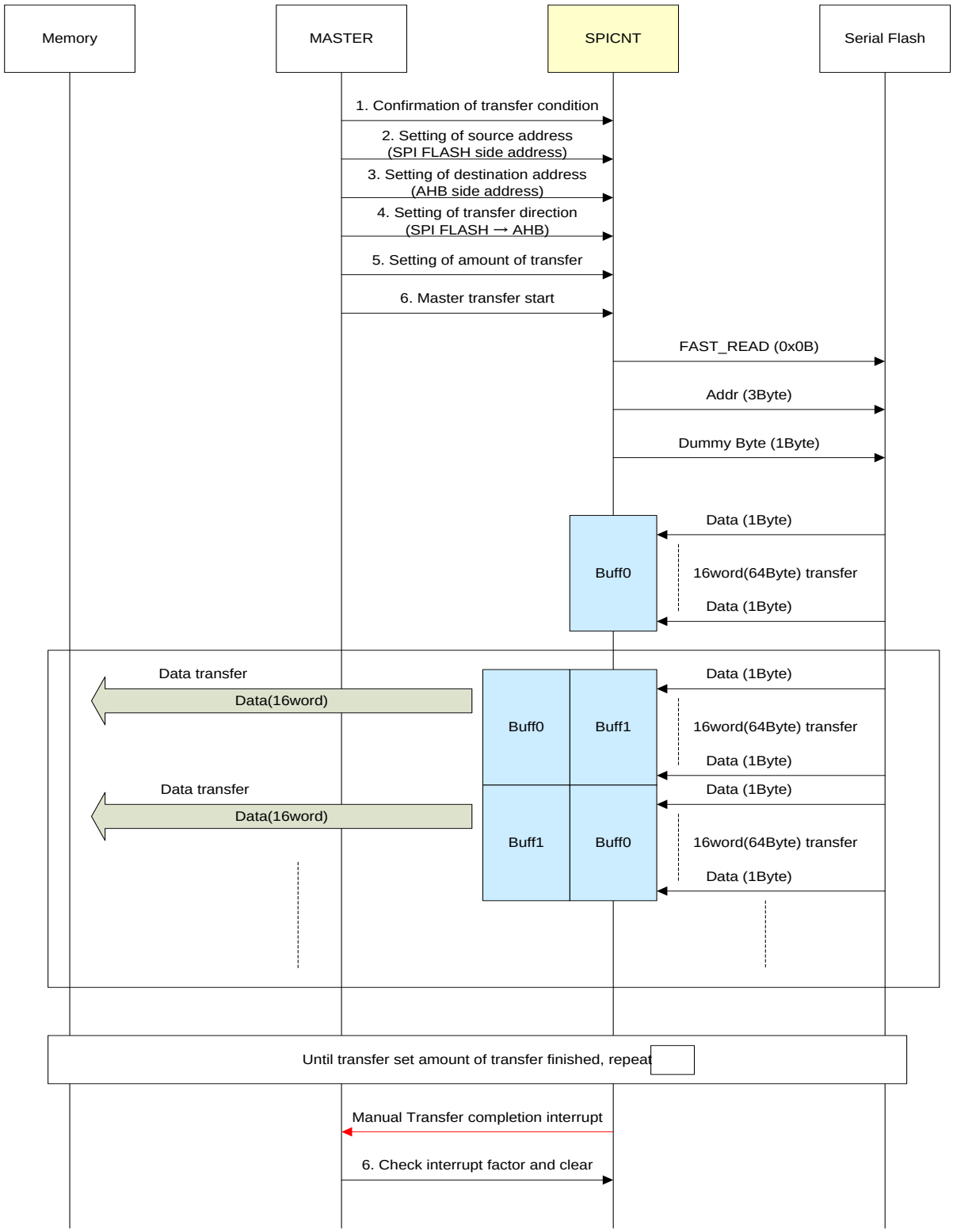
Read

After the source address (Serial Flash memory side), destination address (GDC macro side), transfer direction, and transfer size are set, the start bit starts transfer. A master transfer completion interrupt is issued when transfer is completed.

Note: Do not change any register setting other than the SRST bit in the Control register (01FB_300C$_H$) during this transfer.

The following figure shows the master read flow.

Figure 1-104. Master Read Flow



The control flow of master reading is as follows.

1. Confirm with the Internal Status register (01FB_3008$_H$) that manual or master mode is not being executed.
2. Set the source address in the SPI FLASH memory address register (01FB_3024$_H$).
3. Set the destination address in the AHB Address register (01FB_3020$_H$).
4. Set the master data transfer direction (read: Serial Flash memory -> GDC macro) and transfer size in the Master Transfer Control register (01FB_3028$_H$).
5. The Master Transfer Control register (01FB_3028$_H$) starts master read transfer.
    a. Transfer the FAST_READ command, the source address that was set in step 2, and a dummy byte to Serial Flash memory.
    b. Store the data read from Serial Flash memory in Buff0.
    c. Store the data read from Serial Flash memory in Buff1 when data transfer to Buff0 is completed. At the same time, the data stored in Buff0 is sent to the GDC macro side.
    d. Store the data read from Serial Flash memory in Buff0 when data transfer to Buff1 is completed. At the same time, data stored in Buff1 is sent to the GDC macro side.
    e. Repeat steps 1 to 4 until the completion of the transfer that was set in step 4.
    f. A master transfer completion interrupt is issued when transfer of the designated size is completed.
6. Check the interrupt factor and clear it with the Interrupt Status register (01FB_3000$_H$).
   The target of the check is bit0 in the Interrupt Status register.

1.16.6 Notes

1.16.6.1 Restrictions

CY91F591/2/4/6/7/9

- SPICNT corresponds to Mode0 only.
- SPICNT supports the Serial Flash memory corresponding to Mode0.
- Access beyond the external memory capacity is not permitted.
- Only MEMC can use SPICNT exclusively.
- CMDSEQ reset start is prohibited.
  (For details, see "GDC Access Sequence".)

CY91F59A/B

- SPICNT corresponds to Mode0 only.
- SPICNT supports the Serial Flash memory corresponding to Mode0.
- Access beyond the external memory capacity is not permitted.
- Only MEMC and HS-SPICNT can use SPICNT exclusively.

1.17 NTSC Decoder

This section describes the NTSC decoder (referred to below as NTSC) of the GDC macro.

1.17.1 Overview

The decoder demodulates NTSC/PAL analog video signals into Y/Cb/Cr digital data.

The demodulated data is sent to CAPTURE.

1.17.2 Features

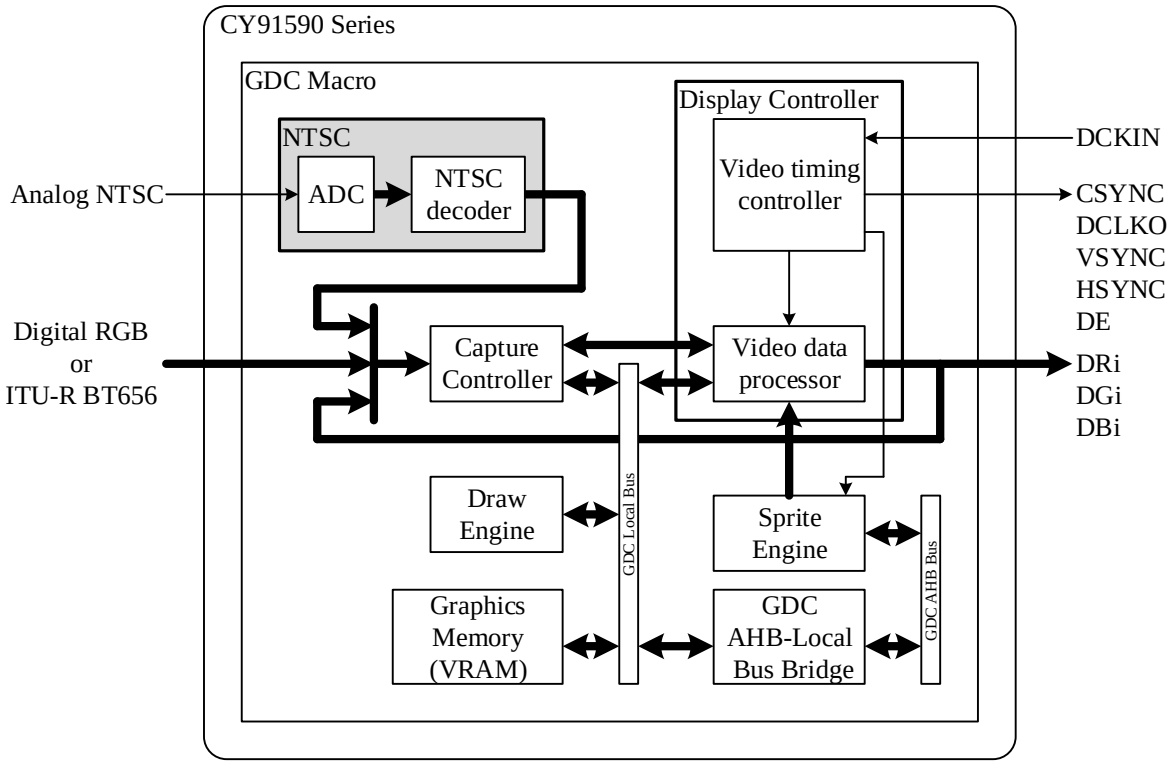
The decoder has the following features:

- Dedicated built-in ADC
- NTSC, PAL, PAL-M, and PAL-N support
- Automatic detection of no signal

1.17.3 Configuration

Figure 1-105 is a block diagram showing NTSC and peripherals.

Figure 1-105. Block Diagram of NTSC and Peripherals



Notes:

- Analog NTSC corresponds to the external pin VIN.
- Digital RGB corresponds to the external pins PA2 to PA7, PB2 to PB7, and PC2 to PC7.
- ITU-R.BT656 corresponds to the external pins PA2 to PA7, PB2, and PB3.
- DCLKI corresponds to the external pin DCKIN.
- CSYNC corresponds to the external pin PG3.
- DCLKO corresponds to the external pin PG4.
- VSYNC corresponds to the external pin PG5.
- HSYNC corresponds to the external pin PG6.
- DE corresponds to the external pin PG7.
- DR*i* corresponds to the external pins P011, P012, and PD2 to PD7.
- DG*i* corresponds to the external pins P013, P014, and PE2 to PE7.
- DB*i* corresponds to the external pins P015, P016, and PF2 to PF7.

1.17.4 Registers

1.17.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (**0040_0000$_H$**) is added for access from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0."
  R1: The read value is always "1."
  W0: The write value is always "0." If "1" is written, operation is not guaranteed.
  W1: The write value is always "1." If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0."
  1: The value is "1."
  X: The value is undefined.

1.17.4.2 Register List

Address	Register Name	Description
01FB_1000$_H$	NTSC_CFG	General module setting register
01FB_1004$_H$	NTSC_KILL	Color killer level adjustment
01FB_1008$_H$	NTSC_PIC	High-frequency compensation and edge enhancement
01FB_100C$_H$	NTSC_TINT	Tint adjustment for NTSC
01FB_1010$_H$	NTSC_TINT_P	Tint adjustment for PAL
01FB_1014$_H$	NTSC_COL_CR	Cr gain adjustment
01FB_1018$_H$	NTSC_COL_CB	Cb gain adjustment
01FB_101C$_H$	NTSC_CONT	Contrast adjustment (Y gain adjustment)
01FB_1020$_H$	NTSC_BRT	Brightness adjustment (Y offset adjustment)
01FB_1024$_H$	NTSC_RELATE	Correlation coefficient of comb filters
01FB_1028$_H$	NTSC_HDEL	Delay adjustment for internal timing signals
01FB_102C$_H$	NTSC_HSDLY	Output video position adjustment
01FB_1030$_H$	NTSC_THRES	Threshold setting for synchronization separation
01FB_1034$_H$	NTSC_NO_SIG	Detection of no signal
01FB_103C$_H$	NTSC_PWRDWN	AD converter power-down control

1.17.4.3 Register Details

NTSC_CFG

Address	01FB_1000$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-											MUTE	MV	AGC	FORMAT[1:0]	
R/W	R0											R/W	R/W	R/W	R/W	R/W
Initial	0											0	0	0	0	0

General module setting register

Bit Field		**Explanation**
No.	**Name**	
1-0	FORMAT[1:0]	Switches to NTSC/PAL/PAL-M/PAL-N. 00: NTSC signal input 01: PAL signal input 10: PAL-M signal input 11: PAL-N signal input
2	AGC	Switches to AGC ON/OFF. 0: OFF 1: ON
3	MV	Enables only macrovision mode NTSC. 0: Non-macrovision signal input 1: Macrovision signal input To prevent macrovision signal color stripes (color burst signal inversion), move the color burst detection point backward by 600 ns from the set value in HDEL[7:0].
4	MUTE	Turns mute ON/OFF. 0: OFF 1: ON A black level of Y=16 and Cb=Cr=128 is output when mute is ON.

NTSC_KILL

Address	$01FB_1004_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-										KILL[5:0]					
R/W	R0										R/W	R/W	R/W	R/W	R/W	R/W
Initial	0										0	0	0	1	0	0

Color killer level

Bit Field		Explanation
No.	**Name**	
5-0	KILL[5:0]	KILL[5:0] = 00_H: The color killer is OFF. KILL[5:0] at higher values: The color killer more easily runs. (If KILL[5:0] is $3F_H$, the color killer is not necessarily ON.)

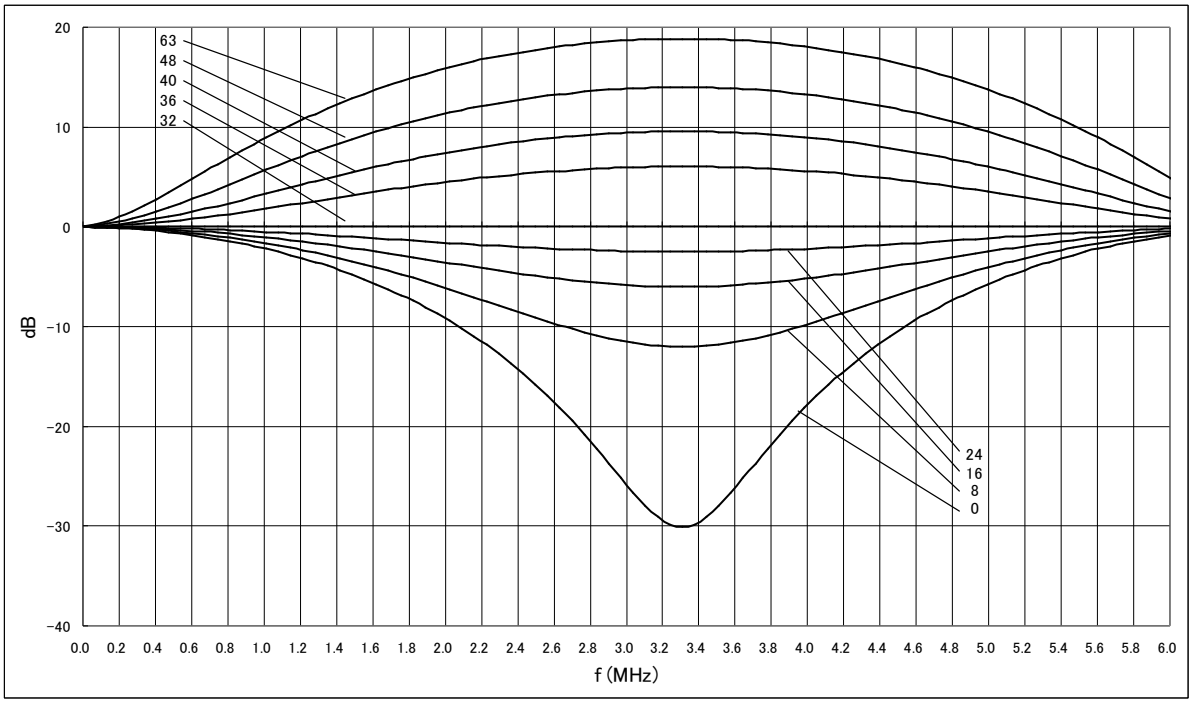
NTSC_PIC

Address	01FB_1008$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name											PIC[5:0]					
R/W	R0										R/W	R/W	R/W	R/W	R/W	R/W
Initial	0										1	0	0	0	0	0

High-frequency compensation and edge enhancement

Bit Field		Explanation
No.	**Name**	
5-0	PIC[5:0]	PIC[5:0] < 32(20$_H$): Attenuated PIC[5:0] = 32(20$_H$): Flat PIC[5:0] > 32(20$_H$): Enhanced The following figure shows frequency characteristics with an arbitrary PIC value.

Figure 1-106. Frequency Characteristics of High-frequency Compensation and Edge Enhancement



NTSC_TINT

Address	01FB_100C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								TINT[7:0]							
R/W	R0								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0								1	0	0	0	0	0	0	0

Tint adjustment for NTSC

Bit Field		Explanation
No.	**Name**	
7-0	TINT[7:0]	TINT[7:0] = 80$_H$: Flat The tint adjustment calculation formula is as follows: Degree = (TINT[7:0]–128)*0.35°

NTSC_TINT_P

Address	01FB_1010$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								TINT_P[7:0]							
R/W	R0								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0								1	0	0	0	0	0	0	0

Tint adjustment for PAL

Bit Field		Explanation
No.	**Name**	
7-0	TINT_P[7:0]	TINT_P[7:0] = 80$_H$: Flat The tint adjustment calculation formula is as follows: Degree = (TINT_P[7:0]–128)*0.35°

NTSC_COL_CR

Address	01FB_1014$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								COL_CR[7:0]							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0								1	0	0	0	0	0	0	0

Cr color adjustment (from 0 times to approximately 2 times greater)

Bit Field		Explanation
No.	Name	
7-0	COL_CR[7:0]	COL_CR[7:0] = 80$_H$: 1 time The color adjustment calculation formula is as follows: Cr = (R–Y)*COL_CR[7:0]/128 If Cr is 254 (FE$_H$) or greater, the value is clipped at 254. If it is 1 (01$_H$) or less, the value is clipped at 1.

NTSC_COL_CB

Address	01FB_1018$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								COL_CB[7:0]							
R/W	R0								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0								1	0	0	0	0	0	0	0

Cb color adjustment (from 0 times to approximately 2 times greater)

Bit Field		Explanation
No.	**Name**	
7-0	COL_CB[7:0]	COL_CB[7:0] = 80$_H$: 1 time The color adjustment calculation formula is as follows: Cb = (B–Y)*COL_CB[7:0]/128 If Cb is 254 (FE$_H$) or greater, the value is clipped at 254. If it is 1 (01$_H$) or less, the value is clipped at 1.

NTSC_CONT

Address	01FB_101C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								CONT[7:0]							
R/W	R0								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0

Contrast adjustment (from 0 times to 4 times greater)

Bit Field		Explanation
No.	Name	
7-0	CONT[7:0]	CONT[7:0] = 40$_H$: 1 time The contrast adjustment calculation formula is as follows: Y' = Y*CONT[7:0]/64 If Y is 254 (FE$_H$) or greater, the value is clipped at 254. If it is 16 (10$_H$) or less, the value is clipped at 16.

NTSC_BRT

Address	01FB_1020$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								BRT[7:0]							
R/W	R0								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0								1	0	0	1	0	0	0	0

Brightness adjustment (-128 to +128)

Bit Field		Explanation
No.	Name	
7-0	BRT[7:0]	The brightness adjustment calculation formula is as follows: Y'' = Y'+(BRT[7:0]–128) If Y is 254 (FE$_H$) or greater, the value is clipped at 254. If it is 16 (10$_H$) or less, the value is clipped at 16.

NTSC_RELATE

Address	01FB_1024$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								RELATE[7:0]							
R/W	R0								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0								0	1	0	0	0	0	0	0

Correlation coefficient of comb filters

Bit Field		Explanation
No.	Name	
7-0	RELATE[7:0]	RELATE = 00$_H$: BPF only (Comb filters OFF) RELATE = FF$_H$: Most comb filters run

NTSC_HDEL

Address	01FB_1028$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								HDEL[7:0]							
R/W	R0								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0								0	1	1	0	0	0	0	0

Color burst detection point adjustment

Bit Field		Explanation
No.	Name	
7-0	HDEL[7:0]	Adjusts the timing from internal H.SYNC to a color burst.

NTSC_HSDLY

Address	01FB_102C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-										HSDLY[5:0]					
R/W	R0										R/W	R/W	R/W	R/W	R/W	R/W
Initial	0										0	1	0	0	1	0

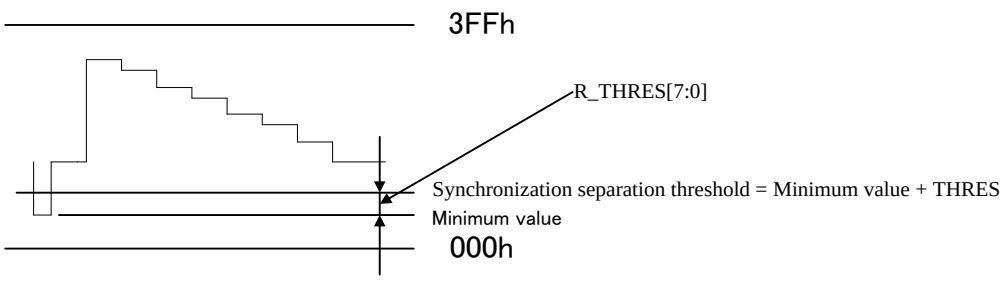
Output video position adjustment in the H direction

Bit Field		Explanation
No.	**Name**	
5-0	HSDLY[5:0]	HSDLY = 00$_H$: Video moves to the right. HSDLY = 3F$_H$: Video moves to the left.

NTSC_THRES

Address	01FB_1030$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-								THRES[7:0]							
R/W	R								R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0								0	0	0	1	0	0	1	0

Threshold for synchronization separation

Bit Field		Explanation
No.	**Name**	
7-0	THRES[7:0]	THRES = 00$_H$: Synchronization separation becomes impossible. THRES = FF$_H$: Synchronization separation becomes impossible. For synchronization separation, the value considered to be SYNC is the minimum video value + THRES, or less. 

NTSC_NO_SIG

Address	$01FB_1034_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															NO_SIG
R/W	R0															R
Initial	0															0

Signal when no signal is detected

Bit Field		Explanation
No.	**Name**	
0	NO_SIG	NO_SIG = 0: Video NO_SIG = 1: No signal Video output becomes a black image ($Y=10_H$, $Cb=Cr=80_H$) at the time of no signal.

NTSC_PWRDWN

Address	01FB_103C$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															PWRDWN
R/W	R0															R/W
Initial	0															0

Power-down

Bit Field		Explanation
No.	**Name**	
0	PWRDWN	PWRDWN = 0: AD converter power-down disabled PWRDWN = 1: AD converter power-down enabled **Note:** NTSC decoder functions are not available by default because AD converter power-down is disabled by default. Be sure to write "1" to the NTSC_PWRDWN register before using NTSC decoder functions.

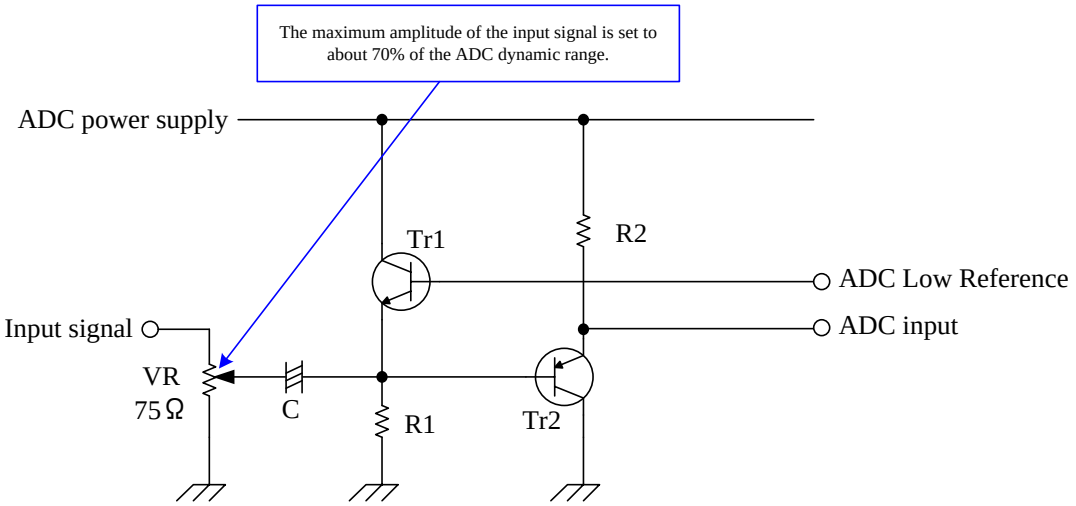
1.17.5 Explanation of Operation

1.17.5.1 Analog Clamping Circuit of the Internal ADC

The Sync level of the ADC input signal must be clamped at the ADC Low Reference level.

The following figure shows an example of the circuit.

Figure 1-107. Analog Clamping Circuit Example



VR is inserted into the input signal to align the maximum amplitude of the input signal (100% White - Sync) to approximately 70% of the ADC dynamic range (High Reference - Low Reference). However, this does not mean that the VR is necessary.

The numerical value of 70% does not have a special meaning, but the range generally used with the ADC is approximately 50% to 75%, depending on the integrity of the input signal.

If no signal exceeds 100% White, or if it does not matter that a signal exceeding 100% White becomes saturated, the ADC can be used at 75%, which improves the S/N ratio of the ADC.

If it is used at 75% or more, 100% Yellow and Cyan color components are broken down, so 75% must not be exceeded.

The input signal in the above figure terminated at 75 ohms. However, this is a general diagram where the received signal was sent through 75 ohms of resistance, and there is no problem as long as the impedance of the signal is low enough.

A desirable design has Tr1 (NPN) and Tr2 (PNP) complementary to each other or, if possible, provides them in a single package because they also operate for temperature compensation.

1.17.5.2 Y/C Separation

An adaptive 2D comb filter separates Y/C.

1.17.5.3 ACC

The ACC automatically adjusts the color level according to the burst level. (1 to 8 times)

If the burst level drops below the set value in the relevant register, the forcible color killer runs.
(The ACC function cannot be turned OFF.)

1.17.5.4 AGC

The AGC automatically adjusts the brightness level according to the Sync level. (1 to 4 times)
The AGC can be turned off from the relevant register. (If the signal level is accurate in the video period, like in output from the latest video equipment, better results can be obtained with the AGC OFF. This is because sync breakdown for a macrovision insertion signal or sync breakdown due to an analog circuit failure can be prevented.)

1.17.5.5 Macrovision Support

Pseudo Sync added to a macrovision signal is always eliminated.
The setting of macrovision mode by the relevant register moves the burst signal detection point backward by approximately 600 nsec.

1.17.5.6 Automatic Judgment Function (Detection of No Signal)

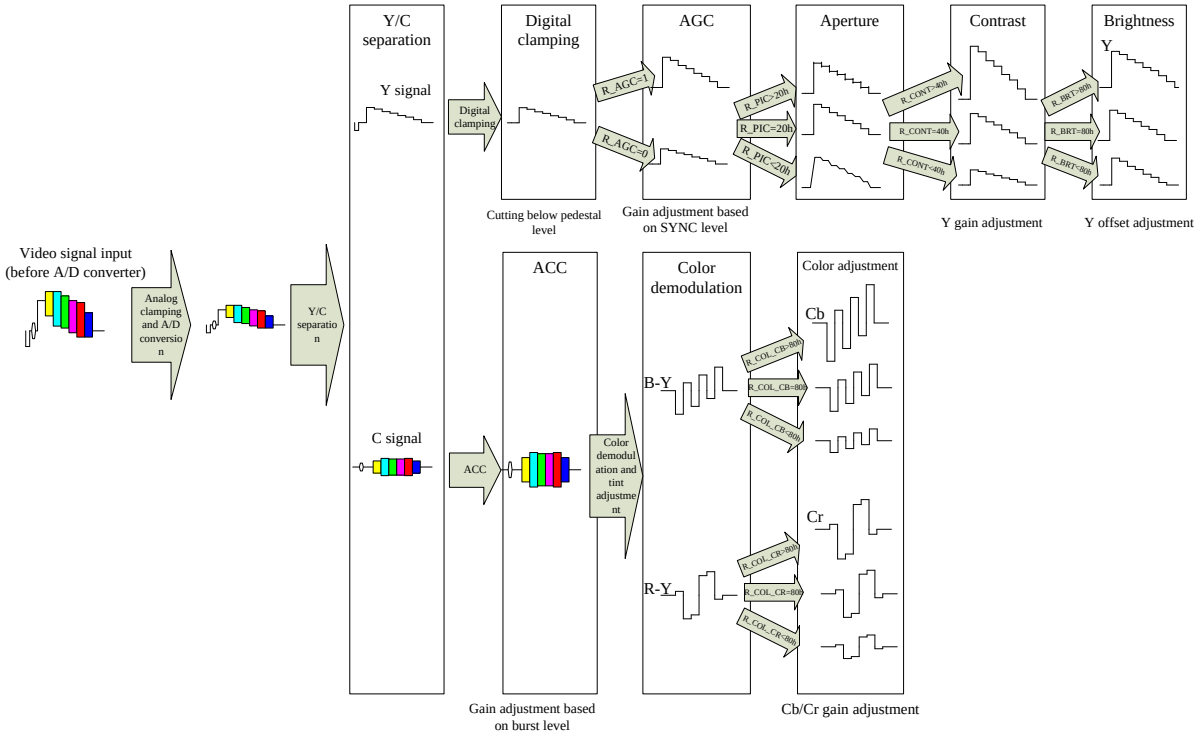
The output of black video when there is no input signal uses free running H.Sync and V.Sync as the video signals. The signal when no signal is detected is also output at the same time.

1.17.5.7 Other Adjustments

Various adjustments can be made with registers. For details on the adjustments, see "Register List".

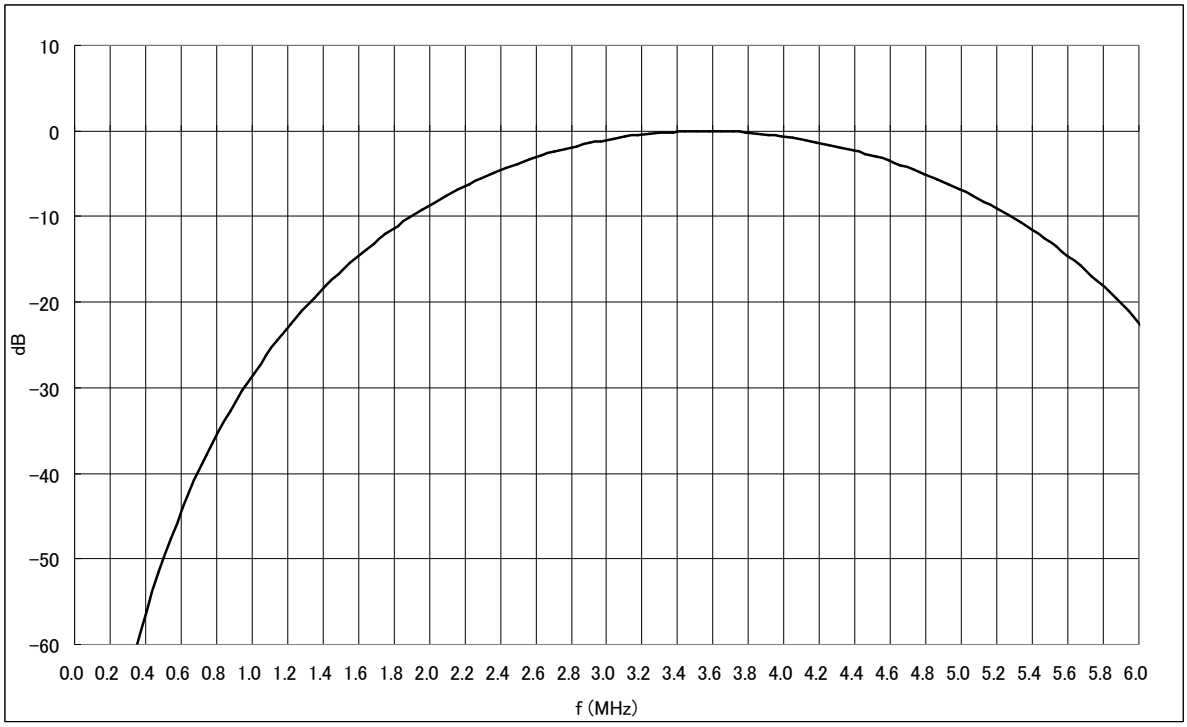
1.17.5.8 Registers and Data Flow Diagram

Figure 1-108. Data Flow

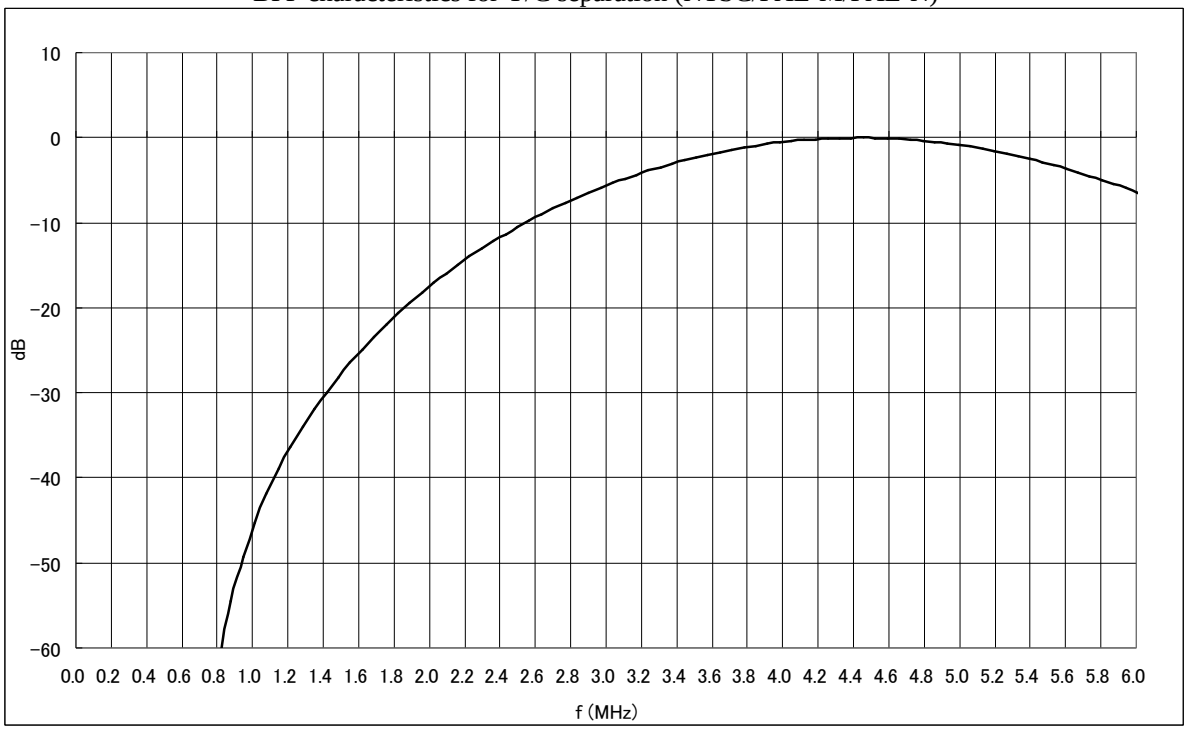


1.17.5.9 BPF Characteristics for Y/C Separation

Figure 1-109. BPF Frequency Characteristics for Y/C Separation



BPF characteristics for Y/C separation (NTSC/PAL-M/PAL-N)



BPF characteristics for Y/C separation (PAL)

1.18 Command RAM (CMDRAM)

1.18.1 Overview

The Command RAM (referred to below as CMDRAM) contains an 8-KB memory area to store user data. It also has data check and error correction functions using Error Correcting Code (referred to below as ECC) for user data in units of 1 byte. The ECC functions can be enabled/disabled with register settings.

1.18.2 Features

- 8-KB RAM area installed
- ECC mode settings

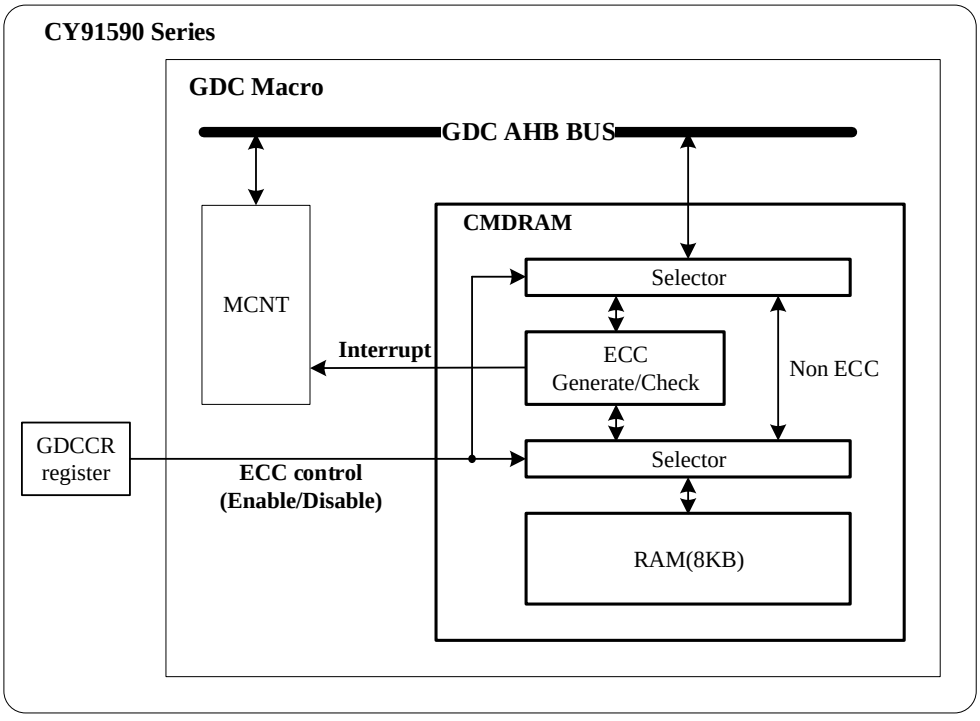
  ECC mode can be enabled/disabled.
- ECC generation and inspection functions
  - ☐ 1-bit error correction in units of 1 byte
  - ☐ 2-bit error detection in units of 1 byte (If an error of 3 bits or more occurs, operation is not guaranteed.)

1.18.3 Configuration

You can enable/disable ECC mode by setting bit2 in the GDCCR register (**0000_0F65$_H$**) of the FR81S.

(For details on ECC mode settings, see "GDC Control Register: GDCCR" in "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.)

Figure 1-110. CMDRAM Module Block Diagram



1.18.4 Registers

1.18.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (***0040_0000H***) is added for access from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.18.4.2 Register List

The registers described below are accessible only when ECC mode is enabled.

They are Reserved when ECC mode is disabled, with writing being ignored and reading returning a value of all "0"s.

Table 1-41. Register List

Address	Register Name	Description
$01F9_F000_H$	EEAR	Error address register of ECC
$01F9_F004_H$	EECSR	Error control register of ECC

1.18.4.3 Register Details

EEAR

EEAR (Ecc Error Address Register) is used when ECC mode is enabled. This register retains generated addresses for 1-bit error correction and 2-bit error verification at the ECC inspection time.

Address	01F9_F000$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	DEEAR[7:0]								-					DEEAR[10:8]		
	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0						ADR10	ADR9	ADR8
R/W	R	R	R	R	R	R	R	R	R0					R	R	R
Initial	0	0	0	0	0	0	0	0	0					0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	SEEAR[7:0]								-					SEEAR[10:8]		
	ADR7	ADR6	ADR5	ADR4	ADR3	ADR2	ADR1	ADR0						ADR10	ADR9	ADR8
R/W	R	R	R	R	R	R	R	R	R0					R	R	R
Initial	0	0	0	0	0	0	0	0	0					0	0	0

Bit Field		**Explanation**
No.	**Name**	
2-0, 15-8	SEEAR[10:0] (Single bit Ecc Error AddRess)	The generated address for 1-bit error correction (the SEI bit in the EECSR register is "1") at the ECC inspection time is set in these bits. These bits are retained until 1-bit error correction is performed again after the writing of "0" to clear the SEI bit. This register stores the address in units of words from bit12 to bit2 inside the CMDRAM memory area (0080_0000$_H$ to 0080_1FFF$_H$). Ex. Single bit error address (0080_0010$_H$) SEEAR[10:0](ADR10-ADR0)=004$_H$
18-16, 31-24	DEEAR[10:0] (Double bit Ecc Error AddRess)	The generated address for when a 2-bit error is detected (the DEI bit in the EECSR register is "1") at the ECC inspection time is set in these bits. These bits are retained until 2-bit error detection is performed again after the writing of "0" to clear the DEI bit. This register stores the address in units of words from bit12 to bit2 inside the CMDRAM memory area (0080_0000$_H$ to 0080_1FFF$_H$). Ex. Double bit error address (0080_0140$_H$) DEEAR[10:0](ADR10-ADR0)=050$_H$

Note: This register is Reserved when ECC mode is disabled, returning "writing disabled" and "read values of all "0"s."

EECSR (Ecc Error Control and Status Register)

EECSR (Ecc Error Control and Status Register) is used when ECC mode is enabled. This register can control interrupt information and the logical addition and output of interrupts for 1-bit error correction and 2-bit error verification at the ECC inspection time.

Address	$01F9_F004_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R0															
Initial	0															
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-												DEIE	DEI	SEIE	SEI
R/W	R1								R0				R/W	R/W	R/W	R/W
Initial	1								0				0	0	0	0

Bit Field		**Explanation**
No.	**Name**	
0	SEI (Single bit ecc Error Interrupt)	This bit indicates whether 1-bit error correction has occurred at the ECC inspection time. Writing "0" to the bit clears the interrupt information. If this bit becomes "1" during 1-bit error correction, the address generated at that time is set in the SEEAR bit in the EEAR register. **SEI** / **1-bit Error Correction** 0 – No 1 – Yes
1	SEIE (Single bit ecc Error Interrupt Enable)	This bit controls interrupts of the SEI bit to logically add and output them to bit31 in the INTST register ($01FB_2100_H$) of the MCNT module. **SEIE** / **SEI Logical Addition and Output Control** 0 – Disable (Logical-addition-and-output disabled) 1 – Enable (Logical-addition-and-output enabled)
2	DEI (Double bit ecc Error Interrupt)	This bit indicates whether 2-bit error detection has occurred at the ECC inspection time. Writing "0" to the bit clears the interrupt information. If this bit becomes "1" when a 2-bit error is detected, the address generated at that time is set in the DEEAR bit in the EEAR register. **DEI** / **2-bit Error Detection** 0 – No 1 – Yes

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>3</td><td>DEIE
(Double bit ecc Error Interrupt Enable)</td><td>This bit controls interrupts of the DEI bit to logically add and output them to bit31 in the INTST register ($01FB_2100_H$) of the MCNT module.

<table><tr><th>DEIE</th><th colspan="2">DEI Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(Logical-addition-and-output disabled)</td></tr><tr><td>1</td><td>Enable</td><td>(Logical-addition-and-output enabled)</td></tr></table></td></tr>
</table>

Note: This register is Reserved when ECC mode is disabled, returning "writing disabled" and "read values of all "0"s."

1.18.5 Explanation of Operation

1.18.5.1 ECC Mode Switching

The bit2 setting in the GDCCR register (**0000_0F65$_H$**) of the FR81S can enable/disable ECC mode as follows. For details on this switching, see Table 1-42. The functional operation of this module varies depending on the ECC mode switching.

For details on the functional operation in each mode, see "1.18.5.2 ECC Mode (Enabled)" and "1.18.5.3 ECC Mode (Disabled)."

(For details on the ECC mode switching timing, see "GDC Access Sequence.")

(For details on the GDCCR register, see "GDC Control Register: GDCCR" in "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.)

Table 1-42. ECC Mode Switching

Register Setting Value	ECC Mode
bit2 in GDCCR register = 0	Disabled
bit2 in GDCCR register = 1	Enabled

Restrictions

- Perform the switching in bit2 in the GDCCR register while the GDC macro is resetting.
- The GDC macro reset can be controlled with bit0 in the GDCCR register.
- In ECC mode switching at the GDC macro reset time, the data before the reset remains in RAM. For this reason, initializing the memory at the same time is recommended.

1.18.5.2 ECC Mode (Enabled)

1-bit error correction and 2-bit error detection when ECC mode is enabled is done in units of bytes. (If an error of 3 bits or more occurs, operation is not guaranteed.)

The following sections describe each of the functions when ECC mode is enabled.

Interrupts

With ECC mode enabled, this module detects interrupts and outputs them externally (the INTST register of the MCNT module).

Interrupt Factors

This module has the following interrupt factors:

- ECC single bit error interrupt (EECSR register (01F9_F004$_H$, bit0 (SEI)))
- ECC double bit error interrupt (EECSR register (01F9_F004$_H$, bit2 (DEI)))

Writing "0" to the corresponding bit (bit0/bit2 in the EECSR register) can clear the above-described interrupt information.

For details on the EECSR register, see "EECSR (Ecc Error Control and Status Register)."

Interrupt Output

This module logically adds and outputs the interrupt information for any interrupt on the module to bit31 in the INTST register (01FB_2100$_H$) of the MCNT module.

Logical addition and output can be controlled with bit1 (SEIE) and bit3 (DEIE) in the EECSR register.

For details on the logical addition and output control, see "EECSR (Ecc Error Control and Status Register)."

Reserved Area

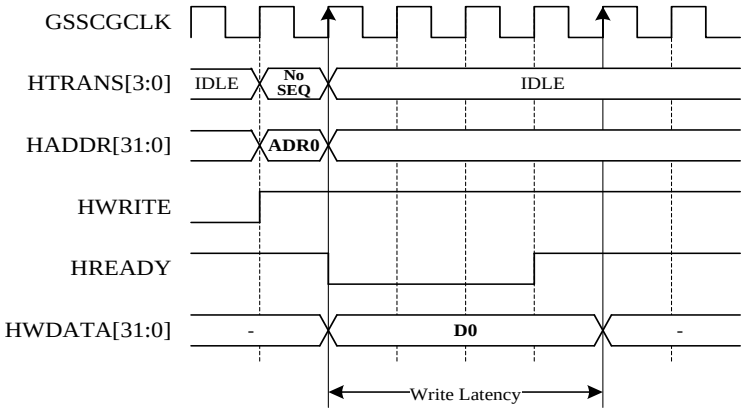
In the Reserved part (01F9_F08$_H$ to 01F9_FFFF$_H$) of the register area, writing is ignored and reading returns a value of all "1"s.

Memory Access

The following sections show the latency of single and burst accesses when ECC mode is enabled.

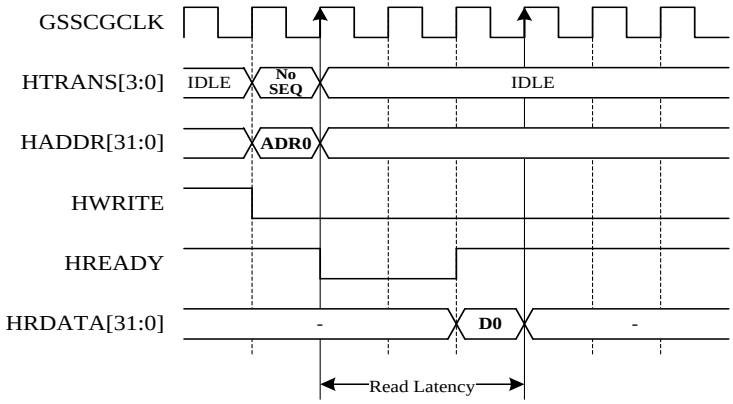
Single Access

- Single Write Access
  - ☐ Write latency = 4 cycles (GSSCGCLK)



[Note]
HTRANS : Transfer type
HADDR : AHB Address
HWRITE : Write/Read (Write=1, Read=0)
HREADY : CMDRAM Ready (Ready=1, Not Ready=0)
HWDATA : Write Data

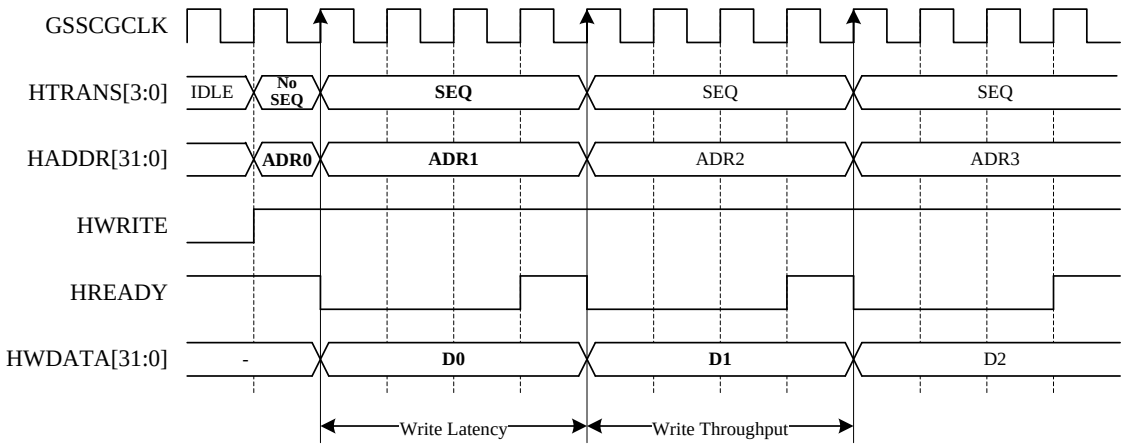
- Single Read Access
  - ☐ Read latency = 3 cycles (GSSCGCLK)



[Note]
HTRANS : Transfer type
HADDR : AHB Address
HWRITE : Write/Read (Write=1, Read=0)
HREADY : CMDRAM Ready (Ready=1, Not Ready=0)
HRDATA : Read Data

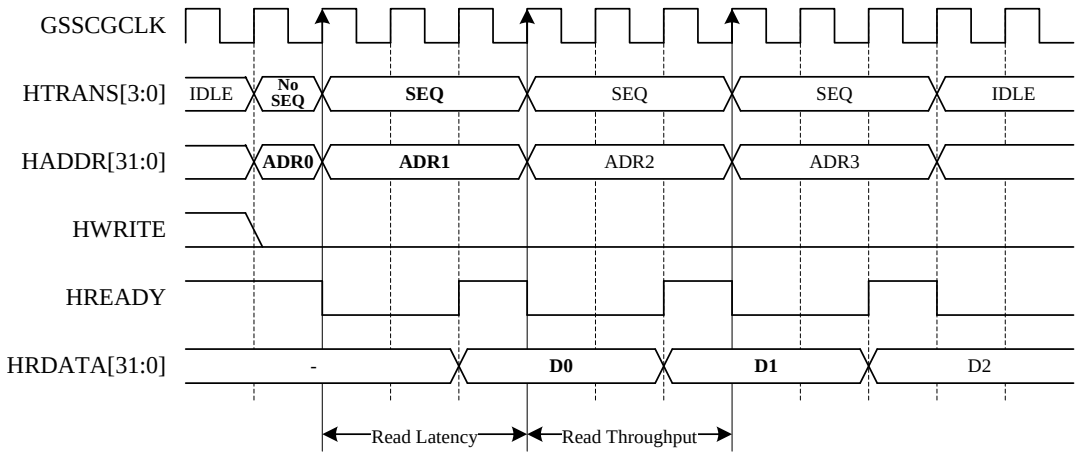
Burst Access

- Burst Write Access
  - Write latency = 4 cycles (GSSCGCLK)
  - Write throughput = 4 cycles (GSSCGCLK)



[Note]
HTRANS : Transfer type
HADDR : AHB Address
HWRITE : Write/Read (Write=1, Read=0)
HREADY : CMDRAM Ready (Ready=1, Not Ready=0)
HWDATA : Write Data

- Burst Read Access
  - Read latency = 3 cycles (GSSCGCLK)
  - Read throughput = 3 cycles (GSSCGCLK)



[Note]
HTRANS : Transfer type
HADDR : AHB Address
HWRITE : Write/Read (Write=1, Read=0)
HREADY : CMDRAM Ready (Ready=1, Not Ready=0)
HRDATA : Read Data

1.18.5.3 ECC Mode (Disabled)

The inspection and correction that use ECC mode are not performed.

Also, all the registers listed in "1.18.4.2 Register List" enter the Reserved state.

Interrupts

No interrupt is generated when ECC mode is disabled.

Reserved Area

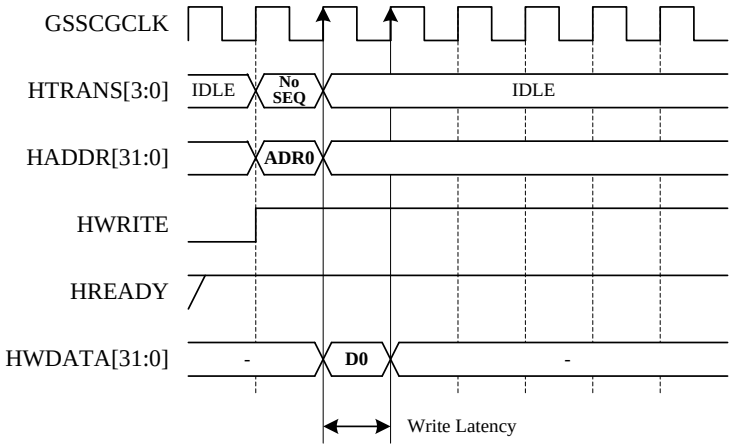
In the Reserved part ($01F9_F000_H$ to $01F9_FFFF_H$) of the register area, writing is ignored and reading returns a value of all "0"s.

Memory Access

The following sections show the latency of single and burst accesses when ECC mode is disabled.

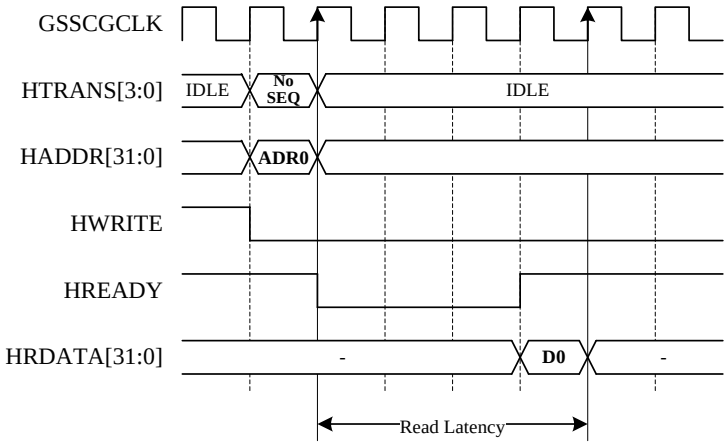
Single Access

- Single Write Latency
  - ☐ Write latency = 1 cycle (GSSCGCLK)



[Note]
HTRANS : Transfer type
HADDR : AHB Address
HWRITE : Write/Read (Write=1, Read=0)
HREADY : CMDRAM Ready (Ready=1, Not Ready=0)
HWDATA : Write Data

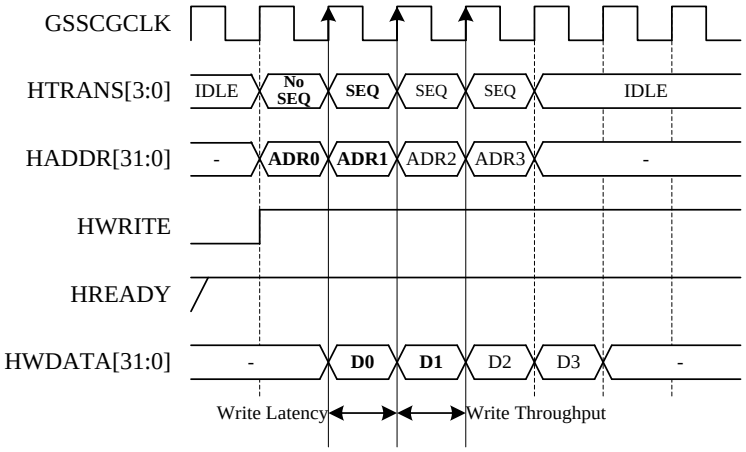
- Single Read Latency
  - ☐ Read latency = 4 cycles (GSSCGCLK)



[Note]
HTRANS : Transfer type
HADDR : AHB Address
HWRITE : Write/Read (Write=1, Read=0)
HREADY : CMDRAM Ready (Ready=1, Not Ready=0)
HRDATA : Read Data

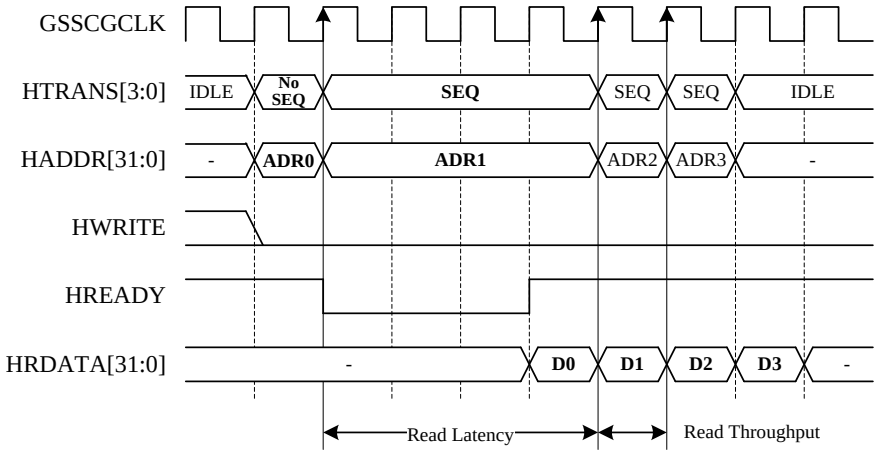
Burst Access

- Burst Write Latency
  - ☐ Write latency = 1 cycle (GSSCGCLK)
  - ☐ Write throughput = 1 cycle (GSSCGCLK)



[Note]
HTRANS : GDC Transfer type
HADDR : GDC AHB Address
HWRITE : GDC Write/Read (Write=1, Read=0)
HREADY : CMDRAM Ready (Ready=1, Not Ready=0)
HWDATA : Write Data

- Burst Read Latency
  - ☐ Read latency = 4 cycles (GSSCGCLK)
  - ☐ Read throughput = 1 cycle (GSSCGCLK)



[Note]
HTRANS : GDC Transfer type
HADDR : GDC AHB Address
HWRITE : GDC Write/Read (Write=1, Read=0)
HREADY : CMDRAM Ready (Ready=1, Not Ready=0)
HRDATA : Read Data

1.19 Module Controller (MCNT)

1.19.1 Overview

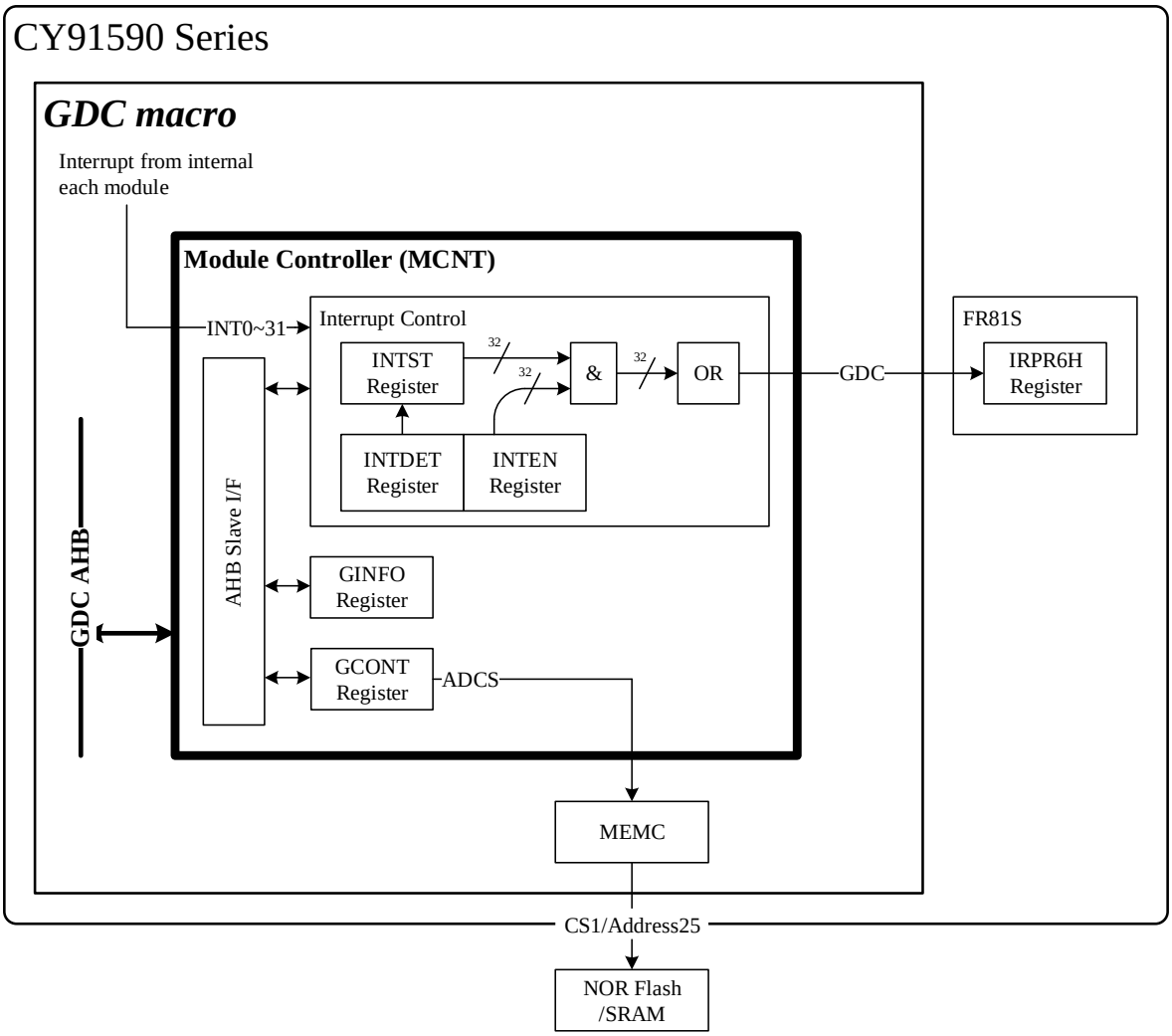
The Module Controller (referred to below as MCNT) reads Graphic Display Controller (referred to below as GDC macro) information and switches the control signals of external bus memory interfaces.

1.19.2 Features

- GDC macro information
  You can check the version, year of manufacture, and ID of the GDC macro.
- Interrupt controller
  This controller detects each interrupt signal inside the GDC macro and logically adds and outputs it to the interrupt controller (IRPR6H register) of the FR81S.
- Switching control signals of external bus memory interfaces
  This feature changes CS1 to the upper bit of the address (Address bit25) when using an external bus memory interface (NOR Flash/SRAM). The change to Address bit25 results in the maximum capacity of 1 external bus memory (CS0) changing from 32MB to 64MB.

1.19.3 Configuration

Figure 1-111. MCNT Block



1.19.4 Registers

1.19.4.1 Format of Register Descriptions

- Endian
  The registers of this module support Little Endian.
- Base address
  The base address (**0040_0000$_H$**) is added for access from the FR81S (CPU).
- Bit
  A bit number in a register is shown.
- Name
  A bit field name in a register is shown.
  "-" indicates Reserved.
- R/W
  The read/write attribute of each bit field is shown.
  R0: The read value is always "0".
  R1: The read value is always "1".
  W0: Be sure to write "0". If "1" is written, operation is not guaranteed.
  W1: Be sure to write "1". If "0" is written, operation is not guaranteed.
  R: Read
  W: Write
- Initial value
  The value of each bit field immediately after a reset is shown.
  0: The value is "0".
  1: The value is "1".
  X: The value is undefined.

1.19.4.2 Register List

Table 1-43. Register List

Address	Register Name	Description
01FB_2000$_H$	GINFO	GDC information
01FB_2100$_H$	INTST	Status of interrupt
01FB_2110$_H$	INTDET	Interrupt detection control
01FB_2120$_H$	INTEN	Output control of external interrupt
01FB_2500$_H$	GCONT	GDC control

1.19.4.3 Register Details

GINFO Register

The GINFO (GDC information) register can check the version, ID, and year of manufacture of the GDC macro.

CY91F591/2/4/6/7/9

Address	$01FB_2000_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	YEAR[15:0]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ID[7:0]								VER[7:0]							
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bit Field		**Explanation**
No.	**Name**	
7-0	VER[7:0] (VERsion)	Indicates the GDC macro version information. The version number is 3 (VER[7:0]=02_H).
15-8	ID[7:0] (ID)	Indicates the GDC macro ID information. The ID is 0 (ID[7:0]=00_H).
31-16	YEAR[15:0] (YEAR)	Indicates information on the year of manufacture of the GDC macro. The year of manufacture is 2011 (YEAD[15:0]=2011_H).

CY91F59A/B

Address	01FB_2000$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	YEAR[15:0]															
R/W	R	R	R	R	R	R	R	R	R	R	R	R:	R	R	R	R
Initial	0	0	1	0	0	0	0	0	0	0	0	1	0	0	1	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	ID[7:0]								VER[7:0]							
R/W	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R
Initial	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit Field		**Explanation**
No.	**Name**	
7-0	VER[7:0] (VERsion)	Indicates the GDC macro version information. The version number is 1 (VER[7:0]=00$_H$).
15-8	ID[7:0] (ID)	Indicates the GDC macro ID information. The ID is 1 (ID[7:0]=01$_H$).
31-16	YEAR[15:0] (YEAR)	Indicates information on the year of manufacture of the GDC macro. The year of manufacture is 2012 (YEAD[15:0]=2012$_H$).

INTST Register

The INTST (interrupt status) register indicates interrupt information from each of the modules inside the GDC macro. This register also uses INTEN register control to logically add and output the interrupt information to bit7 (GDC) in the interrupt controller (IRPR6H register (**0000_0424$_H$**)) of the FR81S. Writing "1" to this register can clear the interrupt information.

Address	01FB_2100$_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	INT 31	INT 30	INT 29	INT 28	INT 27	INT 26	INT 25	INT 24	INT 23	INT 22	INT 21	INT 20	INT 19	INT 18	INT 17	INT 16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	INT 15	INT 14	INT 13	INT 12	INT 11	INT 10	INT 9	INT 8	INT 7	INT 6	INT 5	INT 4	INT 3	INT 2	INT 1	INT 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>0</td><td>INT0
(INTerrupt0)</td><td>Indicates the interrupt status of CMDSEQ.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT0</th><th>CMDSEQ (Interrupt) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>1</td><td>INT1
(INTerrupt1)</td><td>Reserved</td></tr>
<tr><td>2</td><td>INT2
(INTerrupt2)</td><td>Indicates the VSYNC status of DISPLAY.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT2</th><th>DISPLAY (VSYNC) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>3</td><td>INT3
(INTerrupt3)</td><td>Indicates the FSYNC status of DISPLAY.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT3</th><th>DISPLAY (FSYNC) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>4</td><td>INT4
(INTerrupt4)</td><td>Indicates the SYNCERR status of DISPLAY.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT4</th><th>DISPLAY (SYNCERR) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>5</td><td>INT5
(INTerrupt5)</td><td>Indicates the RUPDATE status of DISPLAY.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT5</th><th>DISPLAY (RUPDATE) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>6</td><td>INT6
(INTerrupt6)</td><td>Indicates the BUS access error status of DISPLAY.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT6</th><th>DISPLAY (BUS access error) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>7</td><td>INT7
(INTerrupt7)</td><td>Indicates the interrupt status of CAPTURE.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT7</th><th>CAPTURE (Interrupt) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>10-8</td><td>INT10-8
(INTerrupt10-8)</td><td>Reserved</td></tr>
<tr><td>11</td><td>INT11
(INTerrupt11)</td><td>Indicates the command error status of DRAW.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT11</th><th>CAPTURE (Command error) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>12</td><td>INT12
(INTerrupt12)</td><td>Indicates the INT command execution status of DRAW.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT12</th><th>DRAW (INT command execution) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>13</td><td>INT13
(INTerrupt13)</td><td>Indicates the DMA command execution status of DRAW.
For details, see "0Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT13</th><th>DRAW (DMA command execution) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>14</td><td>INT14
(INTerrupt14)</td><td>Indicates the BUS access error status of DRAW.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT14</th><th>DRAW (BUS access error) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>15</td><td>INT15
(INTerrupt15)</td><td>Indicates the Enable bits change status of SPE.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT15</th><th>SPE (Enable bits change) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>16</td><td>INT16
(INTerrupt16)</td><td>Indicates the BUS error status of SPE.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT16</th><th>SPE (BUS error) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>17</td><td>INT17
(INTerrupt17)</td><td>Indicates the processing error status of SPE.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT17</th><th>SPE (Processing error) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>18</td><td>INT18
(INTerrupt18)</td><td>Indicates the line blank status of SPE.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT18</th><th>SPE (Line blank) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>19</td><td>INT19
(INTerrupt19)</td><td>Indicates the Specified line processing over status of SPE.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT19</th><th>SPE (Specified line processing) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>20</td><td>INT20
(INTerrupt20)</td><td>Indicates the No signal detection status of NTSC.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT20</th><th>NTSC (No signal detection) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>21</td><td>INT21
(INTerrupt21)</td><td>Indicates the interrupt status of SIG.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT21</th><th>SIG (Interrupt) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>22</td><td>INT22
(INTerrupt22)</td><td>Indicates the ch.0 interrupt status of DMAC.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.

<table><tr><th>INT22</th><th>DMAC (ch.0 interrupt) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>23</td><td>INT23
(INTerrupt23)</td><td>Indicates the ch.1 interrupt status of DMAC.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.

<table><tr><th>INT23</th><th>DMAC (ch.1 interrupt) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>24</td><td>INT24
(INTerrupt24)</td><td>Indicates the interrupt status of RLD.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.
<table><tr><th>INT24</th><th>RLD (Interrupt) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>25</td><td>INT25
(INTerrupt25)</td><td>Reserved</td></tr>
<tr><td>26</td><td>INT26
(INTerrupt26)</td><td>Indicates the interrupt status of MEMC.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.

Note:
In the CY91F591/2/4/6/7/9, MEMC control excludes SPICNT control. In the CY91F59A/B, MEMC control excludes SPICNT and HS-SPICNT control.
Consequently, if MEMC is not used, this interrupt is not generated.
For details on the exclusive control, see "External Bus Memory Interface Mode Setting".

<table><tr><th>INT26</th><th>MEMC (Interrupt) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>27</td><td>INT27
(INTerrupt27)</td><td>Indicates the interrupt status of SPICNT.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.

Note:
In the CY91F591/2/4/6/7/9, SPICNT control excludes MEMC control. In the CY91F59A/B, SPICNT control excludes MEMC and HS-SPICNT control.
Consequently, if SPICNT is not used, this interrupt is not generated.
For details on the exclusive control, see "External Bus Memory Interface Mode Setting".

<table><tr><th>INT27</th><th>SPICNT (Interrupt) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>28</td><td>INT28
(INTerrupt28)</td><td>[CY91F591/2/4/6/7/9]
Reserved

[CY91F59A/B]
Indicates the interrupt status of HS-SPICNT.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.

Note:
HS-SPICNT control excludes MEMC and SPICNT control.
Consequently, if HS-SPICNT is not used, this interrupt is not generated.
For details on the exclusive control, see "External Bus Memory Interface Mode Setting".

<table><tr><th>INT28</th><th>HS-SPI (Interrupt) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>29</td><td>INT29
(INTerrupt29)</td><td>Indicates the BUS protocol error status of the GDC Local BUS Bridge.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.

<table><tr><th>INT29</th><th>GDC Local BUS Bridge (BUS protocol error) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
<tr><td>30</td><td>INT30
(INTerrupt30)</td><td>Indicates the protocol error status of the GDC AHB-Local BUS Bridge.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.

<table><tr><th>INT30</th><th>GDC AHB-Local BUS Bridge (Protocol error) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>31</td><td>INT31
(INTerrupt31)</td><td>Indicates the error status of CMDRAM.
For details, see "Input Interrupt Signal".
Writing "1" to this bit clears the interrupt information.

<table><tr><th>INT31</th><th>CMDRAM (Error) Interrupt</th></tr><tr><td>0</td><td>No</td></tr><tr><td>1</td><td>Yes</td></tr></table></td></tr>
</table>

INTDET Register

The INTDET (interrupt detection) register can select an interrupt detection method (edge detection or level detection) of the INTST register. Edge detection uses rising edges, and level detection uses high levels.

Address	$01FB_2110_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IDET 31	IDET 30	IDET 29	IDET 28	IDET 27	IDET 26	IDET 25	IDET 24	IDET 23	IDET 22	IDET 21	IDET 20	IDET 19	IDET 18	IDET 17	IDET 16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	1	1	0	0	1	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IDET 15	IDET 14	IDET 13	IDET 12	IDET 11	IDET 10	IDET 9	IDET 8	IDET 7	IDET 6	IDET 5	IDET 4	IDET 3	IDET 2	IDET 1	IDET 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>0</td><td>IDET0
(Interrupt DETection 0)</td><td>Sets an interrupt detection method in bit0 (Interrupt of CMDSEQ) in the INTST register.

<table><tr><th>IDET0</th><th>INT0 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>1</td><td>IDET1
(Interrupt DETection 1)</td><td>Reserved</td></tr>
<tr><td>2</td><td>IDET2
(Interrupt DETection 2)</td><td>Sets an interrupt detection method in bit2 (VSYNC of DISPLAY) in the INTST register.

<table><tr><th>IDET2</th><th>INT2 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>3</td><td>IDET3
(Interrupt DETection 3)</td><td>Sets an interrupt detection method in bit3 (FSYNC of DISPLAY) in the INTST register.

<table><tr><th>IDET3</th><th>INT3 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>4</td><td>IDET4
(Interrupt DETection 4)</td><td>Sets interrupt detection in bit4 (SYNCERR of DISPLAY) in the INTST register.

<table><tr><th>IDET4</th><th>INT4 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>5</td><td>IDET5
(Interrupt DETection 5)</td><td>Sets interrupt detection in bit5 (RUPDATE of DISPLAY) in the INTST register.
<table><tr><th>IDET5</th><th>INT5 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>6</td><td>IDET6
(Interrupt DETection 6)</td><td>Sets interrupt detection in bit6 (BUS access error of DISPLAY) in the INTST register.
<table><tr><th>IDET6</th><th>INT6 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>7</td><td>IDET7
(Interrupt DETection 7)</td><td>Sets an interrupt detection method in bit7 (Interrupt of CAPTURE) in the INTST register.
<table><tr><th>IDET7</th><th>INT7 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>10-8</td><td>IDET10-8
(Interrupt DETection 10-8)</td><td>Reserved</td></tr>
<tr><td>11</td><td>IDET11
(Interrupt DETection 11)</td><td>Sets an interrupt detection method in bit11 (Command error of DRAW) in the INTST register.
<table><tr><th>IDET11</th><th>INT11 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>12</td><td>IDET12
(Interrupt DETection 12)</td><td>Sets an interrupt detection method in bit12 (INT command execution of DRAW) in the INTST register.
<table><tr><th>IDET12</th><th>INT12 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>13</td><td>IDET13
(Interrupt DETection 13)</td><td>Sets interrupt detection in bit13 (DMA command execution of DRAW) in the INTST register.
<table><tr><th>IDET13</th><th>INT13 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>14</td><td>IDET14
(Interrupt DETection 14)</td><td>Sets interrupt detection in bit14 (BUS access error of DRAW) in the INTST register.
<table><tr><th>IDET14</th><th>INT14 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>15</td><td>IDET15
(Interrupt DETection 15)</td><td>Sets interrupt detection in bit15 (Enable bits change of SPE) in the INTST register.
<table><tr><th>IDET15</th><th>INT15 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>16</td><td>IDET16
(Interrupt DETection 16)</td><td>Sets interrupt detection in bit16 (BUS error of SPE) in the INTST register.
<table><tr><th>IDET16</th><th>INT16 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>17</td><td>IDET17
(Interrupt DETection 17)</td><td>Sets interrupt detection in bit17 (Processing error of SPE) in the INTST register.
<table><tr><th>IDET17</th><th>INT17 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>18</td><td>IDET18
(Interrupt DETection 18)</td><td>Sets an interrupt detection method in bit18 (Line blank of SPE) in the INTST register.
<table><tr><th>IDET18</th><th>INT18 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>19</td><td>IDET19
(Interrupt DETection 19)</td><td>Sets an interrupt detection method in bit19 (Specified line processing over of SPE) in the INTST register.
<table><tr><th>IDET19</th><th>INT19 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>20</td><td>IDET20
(Interrupt DETection 20)</td><td>Sets an interrupt detection method in bit20 (No signal detection of NTSC) in the INTST register.
<table><tr><th>IDET20</th><th>INT20 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>21</td><td>IDET21
(Interrupt DETection 21)</td><td>Sets interrupt detection in bit21 (Interrupt of SIG) in the INTST register.

<table><tr><th>IDET21</th><th>INT21 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>22</td><td>IDET22
(Interrupt DETection 22)</td><td>Sets interrupt detection in bit22 (ch.0 interrupt of DMAC) in the INTST register.

<table><tr><th>IDET22</th><th>INT22 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>23</td><td>IDET23
(Interrupt DETection 23)</td><td>Sets interrupt detection in bit23 (ch.1 interrupt of DMAC) in the INTST register.

<table><tr><th>IDET23</th><th>INT23 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>24</td><td>IDET24
(Interrupt DETection 24)</td><td>Sets interrupt detection in bit24 (Interrupt of RLD) in the INTST register.

<table><tr><th>IDET24</th><th>INT24 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>25</td><td>IDET25
(Interrupt DETection 25)</td><td>Reserved</td></tr>
<tr><td>26</td><td>IDET26
(Interrupt DETection 26)</td><td>Sets interrupt detection in bit26 (Interrupt of MEMC) in the INTST register.

Note:
In the CY91F591/2/4/6/7/9, MEMC control excludes SPICNT control. In the CY91F59A/B, MEMC control excludes SPICNT and HS-SPICNT control.
Consequently, if MEMC is not used, this detection is disabled even when set.
For details on the exclusive control, see "External Bus Memory Interface Mode Setting".

<table><tr><th>IDET26</th><th>INT26 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>27</td><td>IDET27
(Interrupt DETection 27)</td><td>Sets interrupt detection in bit27 (Interrupt of SPICNT) in the INTST register.

Note:
In the CY91F591/2/4/6/7/9, SPICNT control excludes MEMC control. In the CY91F59A/B, SPICNT control excludes MEMC and HS-SPICNT control.
Consequently, if SPICNT is not used, this detection is disabled even when set.
For details on the exclusive control, see "External Bus Memory Interface Mode Setting".

<table><tr><th>IDET27</th><th>INT27 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>28</td><td>IDET28
(Interrupt DETection 28)</td><td>[CY91F591/2/4/6/7/9]
Reserved

[CY91F59A/B]
Sets interrupt detection in bit28 (Interrupt of HS-SPINCT) in the INTST register.

Note:
HS-SPICNT control excludes MEMC and SPICNT control.
Consequently, if HS-SPICNT is not used, this detection is disabled even when set.
For details on the exclusive control, see "External Bus Memory Interface Mode Setting".

<table><tr><th>IDET28</th><th>INT28 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>29</td><td>IDET29
(Interrupt DETection 29)</td><td>Sets an interrupt detection method in bit29 (BUS protocol error of GDC Local BUS Bridge) in the INTST register.

<table><tr><th>IDET29</th><th>INT29 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
<tr><td>30</td><td>IDET30
(Interrupt DETection 30)</td><td>Sets an interrupt detection method in bit30 (Protocol error of GDC AHB-Local BUS Bridge) in the INTST register.

<table><tr><th>IDET30</th><th>INT30 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>31</td><td>IDET31
(Interrupt DETection 31)</td><td>Sets interrupt detection in bit31 (Error of CMDRAM) in the INTST register.

<table><tr><th>IDET31</th><th>INT31 Detection Method</th></tr><tr><td>0</td><td>Edge detection</td></tr><tr><td>1</td><td>Level detection</td></tr></table></td></tr>
</table>

INTEN Register

The INTEN (interrupt enable) register controls the INTST register information to logically add and output to bit7(GDC) in the interrupt controller (IRPR6H register (**0000_0424H**)) of the FR81S.

Address	01FB_2120H															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	IEN 31	IEN 30	IEN 29	IEN 28	IEN 27	IEN 26	IEN 25	IEN 24	IEN 23	IEN 22	IEN 21	IEN 20	IEN 19	IEN 18	IEN 17	IEN 16
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	1	1	[1]	1	1	1	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	IEN 15	IEN 14	IEN 13	IEN 12	IEN 11	IEN 10	IEN 9	IEN 8	IEN 7	IEN 6	IEN 5	IEN 4	IEN 3	IEN 2	IEN 1	IEN 0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

[1]: For the CY91F591/2/4/6/7/9, the initial value is "0".
For the CY91F59A/B, the initial value is "1".

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>0</td><td>IEN0
(Interrupt ENable 0)</td><td>Controls bit0 (Interrupt of CMDSEQ) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table>
<tr><th>IEN0</th><th colspan="2">INT0 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table></td></tr>
<tr><td>1</td><td>IEN1
(Interrupt ENable 1)</td><td>Reserved</td></tr>
<tr><td>2</td><td>IEN2
(Interrupt ENable 2)</td><td>Controls bit2 (VSYNC of DISPLAY) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table>
<tr><th>IEN2</th><th colspan="2">INT2 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>3</td><td>IEN3
(Interrupt ENable 3)</td><td>Controls bit3 (FSYNC of DISPLAY) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN3</th><th colspan="2">INT3 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>4</td><td>IEN4
(Interrupt ENable 4)</td><td>Controls bit4 (SYNCERR of DISPLAY) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN4</th><th colspan="2">INT4 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>5</td><td>IEN5
(Interrupt ENable 5)</td><td>Controls bit5 (RUPDATE of DISPLAY) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN5</th><th colspan="2">INT5 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>6</td><td>IEN6
(Interrupt ENable 6)</td><td>Controls bit6 (BUS access error of DISPLAY) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN6</th><th colspan="2">INT6 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>7</td><td>IEN7
(Interrupt ENable 7)</td><td>Controls bit7 (Interrupt of CAPTURE) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN7</th><th colspan="2">INT7 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>10-8</td><td>IEN10-8
(Interrupt ENable 10-8)</td><td>Reserved</td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>11</td><td>IEN11
(Interrupt ENable 11)</td><td>Controls bit11 (Command error of DRAW) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table>
<tr><th>IEN11</th><th colspan="2">INT11 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table></td></tr>
<tr><td>12</td><td>IEN12
(Interrupt ENable 12)</td><td>Controls bit12 (INT command execution of DRAW) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table>
<tr><th>IEN12</th><th colspan="2">INT12 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table></td></tr>
<tr><td>13</td><td>IEN13
(Interrupt ENable 13)</td><td>Controls bit13 (DMA command execution of DRAW) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table>
<tr><th>IEN13</th><th colspan="2">INT13 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table></td></tr>
<tr><td>14</td><td>IEN14
(Interrupt ENable 14)</td><td>Controls bit14 (BUS access error of DRAW) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table>
<tr><th>IEN14</th><th colspan="2">INT14 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table></td></tr>
<tr><td>15</td><td>IEN15
(Interrupt ENable 15)</td><td>Controls bit15 (Enable bits change of SPE) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table>
<tr><th>IEN15</th><th colspan="2">INT15 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>16</td><td>IEN16
(Interrupt ENable 16)</td><td>Controls bit16 (BUS error of SPE) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN16</th><th colspan="2">INT16 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>17</td><td>IEN17
(Interrupt ENable 17)</td><td>Controls bit17 (Processing error of SPE) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN17</th><th colspan="2">INT17 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>18</td><td>IEN18
(Interrupt ENable 18)</td><td>Controls bit18 (Line blank of SPE) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN18</th><th colspan="2">INT18 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>19</td><td>IEN19
(Interrupt ENable 19)</td><td>Controls bit19 (Specified line processing over of SPE) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN19</th><th colspan="2">INT19 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>20</td><td>IEN20
(Interrupt ENable 20)</td><td>Controls bit20 (No signal detection of NTSC) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN20</th><th colspan="2">INT20 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>21</td><td>IEN21
(Interrupt ENable 21)</td><td>Controls bit21 (Interrupt of SIG) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN21</th><th colspan="2">INT21 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>22</td><td>IEN22
(Interrupt ENable 22)</td><td>Controls bit22 (Ch0 interrupt of DMAC) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN22</th><th colspan="2">INT22 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>23</td><td>IEN23
(Interrupt ENable 23)</td><td>Controls bit23 (Ch1 interrupt of DMAC) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN23</th><th colspan="2">INT23 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>24</td><td>IEN24
(Interrupt ENable 24)</td><td>Controls bit24 (Interrupt of RLD) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.
<table><tr><th>IEN24</th><th colspan="2">INT24 Logical Addition and Output Control</th></tr><tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr><tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr></table></td></tr>
<tr><td>25</td><td>IEN25
(Interrupt ENable 25)</td><td>Reserved</td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>26</td><td>IEN26
(Interrupt ENable 26)</td><td>Controls bit26 (Interrupt of MEMC) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.

Note:
In the CY91F591/2/4/6/7/9, MEMC control excludes SPICNT control. In the CY91F59A/B, MEMC control excludes SPICNT and HS-SPICNT control.
Consequently, if MEMC is not used, this output control is disabled even when set to Enable.
For details on the exclusive control, see "External Bus Memory Interface Mode Setting".

<table>
<tr><th>IEN26</th><th colspan="2">INT26 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table>
</td></tr>
<tr><td>27</td><td>IEN27
(Interrupt ENable 27)</td><td>Controls bit27 (Interrupt of SPICNT) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.

Note:
In the CY91F591/2/4/6/7/9, SPICNT control excludes MEMC control. In the CY91F59A/B, SPICNT control excludes MEMC and HS-SPICNT control.
Consequently, if SPICNT is not used, this output control is disabled even when set to Enable.
For details on the exclusive control, see "External Bus Memory Interface Mode Setting".

<table>
<tr><th>IEN27</th><th colspan="2">INT27 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table>
</td></tr>
<tr><td>28</td><td>IEN28
(Interrupt ENable 28)</td><td>[CY91F591/2/4/6/7/9]
Reserved

[CY91F59A/B]
Controls bit28 (Interrupt of HS-SPICNT) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.

Note:
HS-SPICNT control excludes MEMC and SPICNT control.
Consequently, if HS-SPICNT is not used, this output control is disabled even when set to Enable.
For details on the exclusive control, see "External Bus Memory Interface Mode Setting".

<table>
<tr><th>IEN28</th><th colspan="2">INT28 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>29</td><td>IEN29
(Interrupt ENable 29)</td><td>Controls bit29 (BUS protocol error of GDC Local BUS Bridge) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.

<table>
<tr><th>IEN29</th><th colspan="2">INT29 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table>
</td></tr>
<tr><td>30</td><td>IEN30
(Interrupt ENable 30)</td><td>Controls bit30 (Protocol error of GDC AHB-Local BUS Bridge) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.

<table>
<tr><th>IEN30</th><th colspan="2">INT30 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table>
</td></tr>
<tr><td>31</td><td>IEN31
(Interrupt ENable 31)</td><td>Controls bit31 (Error of CMDRAM) in the INTST register for logical addition and output to the interrupt controller (IRPR6H register) of the FR81S.

<table>
<tr><th>IEN31</th><th colspan="2">INT31 Logical Addition and Output Control</th></tr>
<tr><td>0</td><td>Disable</td><td>(There is no logical addition and output to the IRPR6H register.)</td></tr>
<tr><td>1</td><td>Enable</td><td>(There is logical addition and output to the IRPR6H register.)</td></tr>
</table>
</td></tr>
</table>

GCONT Register

The GCONT (GDC control) register controls the MEMC module.

Address	$01FB_2500_H$															
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Name	-															
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Name	-															ADSC
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

<table>
<tr><th colspan="2">Bit Field</th><th rowspan="2">Explanation</th></tr>
<tr><th>No.</th><th>Name</th></tr>
<tr><td>0</td><td>ADCS
(ADdress/Cs Select)</td><td>Switches the control signals of external bus memory interfaces of MEMC.
For details, see "Switching Control Signals of External Bus Memory Interfaces".
<table>
<tr><th>ADCS</th><th colspan="2">Switching Control Signals of External Bus Memory Interfaces</th></tr>
<tr><td>0</td><td>CS1</td><td>Switch the control signal to CS1.</td></tr>
<tr><td>1</td><td>Addr25</td><td>Switch the control signal to Address bit25.</td></tr>
</table>
</td></tr>
</table>

1.19.5 Explanation of Operation

1.19.5.1 GDC Information (GINFO)

You can check the version information, ID, and year of manufacture of the GDC macro in the GDC information.

Version Information

This is the GDC macro version information.

(For details, see "GINFO Register".)

ID

This is the GDC macro ID information.

(For details, see "GINFO Register".)

Year of Manufacture

This is information on the year of manufacture of the GDC macro.

(For details, see "GINFO Register".)

1.19.5.2 Interrupt Control

Interrupt signals from each of the modules inside the GDC macro are stored in the INTST register by interrupt control. This control also uses INTEN Register control to logically add and output the interrupt information to bit7 (GDC) in the interrupt controller and the IRPR6H register (**0000_0424$_H$**) of the FR81S.

(For details on interrupt processing, see "1.5.1.2 Interrupts".)

Interrupt Detection

Interrupt detection uses the set contents (edge detection or level detection) of the INTDET register to detect interrupt signals from each of the modules and store them in the INTST register.

Edge Detection

This detects an interrupt signal at a rising edge.

(For details on the detection mode of each macro, see "Input Interrupt Signal".)

Level Detection

This detects an interrupt signal at a "H" level.

(For details on the detection mode of each macro, see "Input Interrupt Signal".)

Interrupt Clearing

The interrupt information is cleared when "1" is written to the corresponding bit in the INTST register.

(For details on the clearing procedure of the GDC macro, see "Interrupt Clearing Procedure".)

External Interrupt Output

This outputs an external interrupt to bit7 (GDC) in the interrupt controller and the IRPR6H register (**0000_0424$_H$**) of the FR81S.

This external interrupt uses INTEN Register control to logically add and output INTST register information.

Interrupts disabled in the INTEN Register are neither logically added nor output as external interrupts.

(For details, see "INTEN Register".)

Input Interrupt Signal

The following table lists the interrupts inside the GDC macro.

For details on interrupt generation conditions, see the chapter of each module.

For details on the interrupt setting procedure and the clearing procedure of the GDC macro, see "1.5.1.2 Interrupts".

Notes:

- "Interrupt Number" in the table corresponds to the numbers listed in "INTST Register".
- "Interrupt Detection Mode" in the table corresponds to that listed in "INTDET Register".
- The modules listed with (#) under "Detection Source Module" in the table contain the whole control function for interrupt output to MCNT. The modules listed with (#*) under "Detection Source Module" in the table contain part of the control function for interrupt output to MCNT.

Table 1-44. Interrupt List

Detection Source Module (Interrupt Number)	Initial Value of Interrupt Detection Mode (Recommended)	Interrupt Factor	Referred Register	Interrupt Clearing Procedure
CMDSEQ (#) (INT0)	Edge (Edge)	End of reset start	$01FB_7000_H$	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”
		End of priority1 trigger start		
		End of priority3 trigger start		
		End of various trigger start		
		End of register start		
		Forced ending of commandlist processing		
		Error of AHB slave access		
		Error of comparison		
		Non boot mode set by GDCCR register		
		The data of External Flash are all "f"		
		Error of WAIT trigger command		
		Error of END command		
		Interrupt start by priority (Reset start is starting)		
		Interrupt start by priority (Priority3 start is starting)		
		Interrupt start by priority (Various trigger start is starting)		
		Interrupt start by priority (Register start is starting)		
		Byte lane of CMPREG2 command is all "0"		
		Error of CMDSEQ	$01FB_7000_H$	For details on GDC macro reset, see “Interrupt Clearing Procedure as Required for a GDC Macro Reset” (For detailed interrupt factors, check in the CMDSEQ register.)
		Individual Reset completion		
Reserved (INT1)	- (-)	-	-	-
Display (INT2)	Edge (Edge)	Vertical SYNC	-	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”. (The clearing procedure continues up to MCNT because these interrupts have no status register.)
Display (INT3)	Edge (Edge)	Frame SYNC	-	
Display (INT4)	Edge (Edge)	SYNC error	-	
Display (INT5)	Edge (Edge)	Register update	-	

Detection Source Module (Interrupt Number)	Initial Value of Interrupt Detection Mode (Recommended)	Interrupt Factor	Referred Register	Interrupt Clearing Procedure
Display (INT6)	Edge (Edge)	BUS access error (Master) detection	-	For details on GDC macro reset, see “Interrupt Clearing Procedure as Required for a GDC Macro Reset”.
Capture (#) (INT7)	Edge (Level)	Frame SYNC	$01FD_8178_H$	For the level clearing procedure, see “Clearing Procedure for Level Detection”.
		656 stream error	$01FD_8180_H$	
Reserved (INT8-10)	- (-)	-	-	-
Draw (INT11)	Edge (Edge)	Command error	$01FF_0400_H$	For details on GDC macro reset, see “Interrupt Clearing Procedure as Required for a GDC Macro Reset”.
Draw (INT12)	Edge (Edge)	INT command execution	-	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”. (The clearing procedure continues up to MCNT because these interrupts have no status register.)
Draw (INT13)	Edge (Edge)	DMA command execution	-	
Draw (INT14)	Edge (Edge)	BUS access error (Master) detection	-	For details on GDC macro reset, see “Interrupt Clearing Procedure as Required for a GDC Macro Reset”.
SPE (INT15)	Edge (Edge)	Enable bits change	-	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”. (The clearing procedure continues up to MCNT because these interrupts have no status register.)
SPE (INT16)	Edge (Edge)	BUS error	-	
SPE (INT17)	Edge (Edge)	Processing error	-	
SPE (INT18)	Edge (Edge)	Line blank	-	
SPE (INT19)	Edge (Edge)	Specified line processing over	-	
NTSC (INT20)	Edge (Edge)	No signal detection	$01FB_1034_H$	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”. (**Note:** The clearing procedure continues up to MCNT because a status register cannot be cleared.)
SIG (#) (INT21)	Edge (Edge)	Unequal sum	$01FB_005C_H$	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”
		Unequal CRC		
		Shadow took over register		
		AHB Slave error		
		Input FiFo empty		
		Input FiFo full		

Detection Source Module (Interrupt Number)	Initial Value of Interrupt Detection Mode (Recommended)	Interrupt Factor	Referred Register	Interrupt Clearing Procedure
DMAC ch.0 (#) (INT22)	Edge (Edge)	Address overflow error Transfer stop request error Source access error Destination access error Normal end	01FB_4014$_H$	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”
DMAC ch.1 (#) (INT23)	Edge (Edge)	Address overflow error Transfer stop request error Source access error Destination access error Normal end	01FB_4024$_H$	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”.
RLD (#) (INT24)	Edge (Edge)	Byte count achieved AHB Slave error Input FiFo empty Input FiFo full	01FB_5028$_H$	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”.
Reserved (INT25)	Edge (Edge)	-	-	-
MEMC (#) (INT26)[1]	Edge (Edge)	Access error outside area	01FB_3200$_H$	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”.
SPICNT (#) (INT27)[1]	Edge (Edge)	Transfer complete in Master Transfer Mode Error response of AHB during Manual Mode transfer complete Soft reset Without setting the guard release code when Master Transfer Mode (Write) is executed	01FB_3000$_H$	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”0.
Reserved (INT28)[3]	- (-)	-	-	-
HS-SPICNT (#*) (INT28)[1][4]	Level (Level)	TX-FIFO Full Interrupt TX-FIFO and Shift Register are Empty Interrupt TX-FIFO overrun Interrupt TX-FIFO Underrun Interrupt TX-FIFO Fill Level is Less than or Equal to Threshold Interrupt TX-FIFO Fill Level is More than Threshold Interrupt Slave Select Released Interrupt	01FB_3014$_H$	For the level clearing procedure, see “Clearing Procedure for Level Detection”.

Detection Source Module (Interrupt Number)	Initial Value of Interrupt Detection Mode (Recommended)	Interrupt Factor	Referred Register	Interrupt Clearing Procedure
HS-SPICNT (#*) (INT28)[1][4]	Level (Level)	RX-FIFO Full Interrupt	01FB_3020$_H$	For the level clearing procedure, see “Clearing Procedure for Level Detection”.
		RX-FIFO is Empty Interrupt		
		RX-FIFO Overrun Interrupt		
		RX-FIFO Underrun Interrupt		
		RX-FIFO Fill Level is Less than or Equal to Threshold Interrupt		
		RX-FIFO Fill Level is More than Threshold Interrupt		
		Slave Select Released Interrupt		
		Unmapped Memory Access Fault	01FB_302C$_H$	For the level clearing procedure, see “Clearing Procedure for Level Detection”0.
		Write Access Fault		
		Protection Violation Fault		
GDC Local BUS Bridge (INT29)	Edge (None)	Local BUS protocol error	-	For details on GDC macro reset, see “Interrupt Clearing Procedure as Required for a GDC Macro Reset”.
GDC AHB-Local BUS Bridge (INT30)	Edge (Edge)	Protocol error	-	For details on GDC macro reset, see “Interrupt Clearing Procedure as Required for a GDC Macro Reset”.
CMDRAM (#) (INT31)[2]	Edge (Edge)	Single-bit error	01F9_F004$_H$	For the edge clearing procedure, see “Clearing Procedure for Edge Detection”.
		Double-bit error		

[1]: In the CY91F591/2/4/6/7/9, interrupts corresponding to INT26 and interrupts corresponding to INT27 (MEMC/SPICNT) are mutually exclusive.
In the CY91F59A/B, interrupts corresponding to INT26, interrupts corresponding to INT27, and interrupts corresponding to INT28 (MEMC/SPICNT/HS-SPICNT) are mutually exclusive.
To switch exclusivity, use the GDCCR register *(**0000_0F65$_H$**)* of the FR81S.
(For details on switching exclusivity, see "External Bus Memory Interface Mode Setting".)

[2]: Interrupts with INT31 are enabled only when CMDRAM is in ECC mode.
(For details on ECC mode switching, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.)

[3]: In the CY91F591/2/4/6/7/9, INT28 is reserved.

[4]: In the CY91F59A/B, INT28 is HS-SPICNT.

1.19.5.3 GDC Control

GDC control is the control of the MEMC module as described in "19.4.3.5. GCONT Register."

Switching Control Signals of External Bus Memory Interfaces

This control (ADCS bit) switches the control signals (address upper bit/CS) of the external bus memory interfaces (NOR Flash/SRAM) when MEMC is used.[1]

[1]: Exclusive switching between MEMC (NOR Flash/SRAM control) and SPICNT (serial Flash control) is possible with the GDCCR register (**0000_0F65$_H$**) of the FR81S.
(For details on the GDCCR register, see "Chapter 44: GDC External Control" in the CY91590 Series Hardware Manual.)

The following table lists the correspondent relationship and restrictions of the external bus memory control signals controlled by ADCS.

ADCS (bit0)	External Bus Memory Interface Control Signal	Restrictions	
		Maximum Capacity of External Bus Memory	Maximum Number of External Bus Memory Controls
0	CS1	32MB	2
1	Address bit25	64MB	1

1.19.5.4 Restrictions

If the external bus memory interface control signal is switched to Address bit25 (ADCS is "1"), 1 external bus memory (CS0) can be controlled. If it is set to CS1 (ADCS is "0"), the maximum capacity of 1 external bus memory is 32MB.

1.20 Local Bus Bridge

This section explains the local bus bridges in the GDC macro.

1.20.1 Overview

The GDC macro includes 2 types of local bus bridge between a GDC Local BUS and GDC AHB BUS. These bus bridges convert bus access between a GDC Local BUS and another GDC Local BUS and between a GDC Local BUS and GDC AHB BUS. They also monitor whether transactions occurring on the buses are being processed correctly.

- GDC Local BUS-GDC Local BUS bridge (GDC Local Bus Bridge)
- GDC Local BUS-GDC AHB BUS bridge (GDC AHB-Local Bus Bridge)

Note: This section does not cover the GDC Bus Bridge.

For details on the GDC Bus Bridge, see "1.21 GDC Bus Bridge."

1.20.2 Features

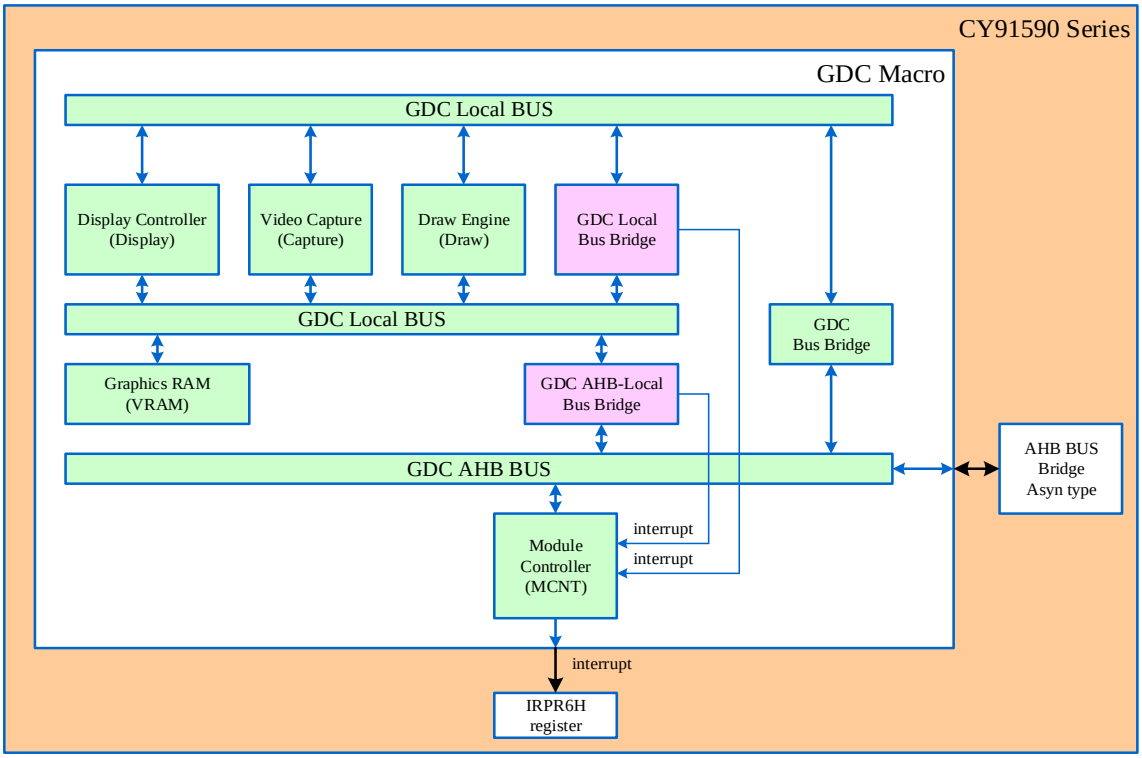
If an access error occurs, the local bus bridge issues an interrupt.

1.20.3 Configuration

1.20.3.1 Block Diagram

Figure 1-112 is a block diagram showing the local bridges and peripherals.

Figure 1-112. Block Diagram of the Local Bridges and Peripherals



1.20.4 Explanation of Operation

1.20.4.1 GDC Local Bus Bridge

Memory in the GDC macro is accessed via this bridge when the display list DMA function in the GDC Bus Bridge module transfers a display list from the memory to the draw engine.

This module issues an interrupt when an access error response occurs on the GDC Local BUS during the above-described access.

Initialize the GDC macro when the module issues the interrupt.

The interrupt information issued by this bridge can be checked with bit29 in the INTST register ($01FB_2100_H$) of the MCNT module.

1.20.4.2 GDC AHB-Local Bus Bridge

The GDC AHB-Local Bus Bridge converts bus access from the master module on a GDC Local BUS to a slave module on a GDC AHB BUS. It also converts bus access from the master module on a GDC AHB BUS to a slave module on a GDC Local BUS.

Regarding access from the master module on a GDC Local BUS to a slave module on a GDC AHB BUS, if the slave module on the GDC AHB BUS generates an access error response, this bridge issues an interrupt.

If this module issues an interrupt, retry the relevant access.

The interrupt information issued by this bridge can be checked with bit30 in the INTST register ($01FB_2100_H$) of the MCNT module.

1.21 GDC Bus Bridge

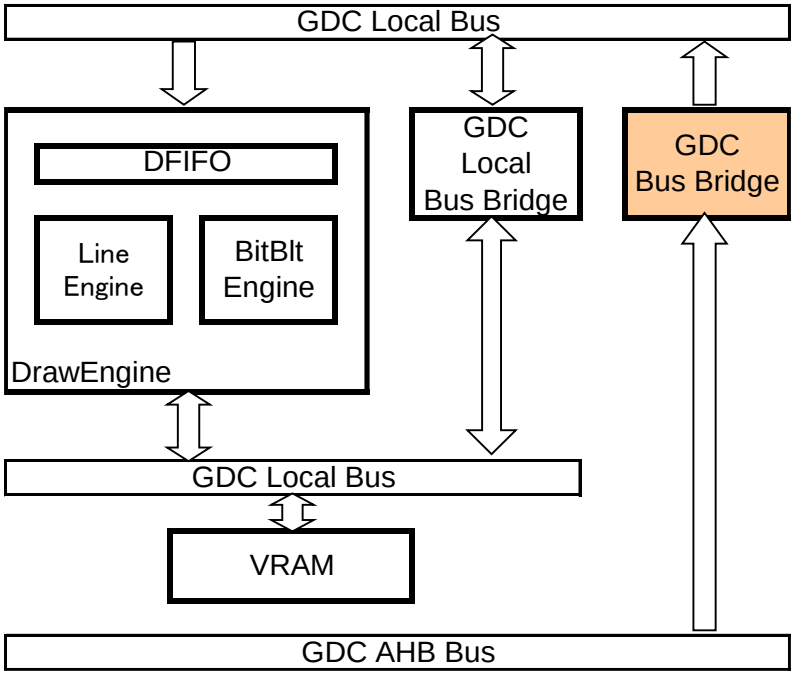
1.21.1 Overview

This module has the DMA transfer function for display lists and registers that retain interrupt factors of the draw engine.

1.21.2 Features

The GDC macro has a DMA controller dedicated for display lists. This module can become a master to transfer display lists placed in memory to the display list FIFO of the draw engine.

1.21.3 Configuration



1.21.4 Registers

1.21.4.1 Display List DMA-related Registers

Address	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
01FC_0008H																	LTS															
																								LTS								
01FC_0010H	LSTA																															
																																LSTA
01FC_0040H	LSA																															
								LSA																								
01FC_0044H	LCO																															
									LCO																							
01FC_0048H																									LREQ							
																																LREQ

LTS (displayList Transfer Stop)

Register Address	01FC_0009H							
Bit Number	7	6	5	4	3	2	1	0
Bit Field Name	-							LTS
R/W	R0							RW
Initial Value	0							0

This register stops the transfer of a display list.
Writing "1" in the register stops the display list transfer currently in progress.

LSTA (displayList transfer STAtus)

Register Address	01FC_0010H							
Bit Number	7	6	5	4	3	2	1	0
Bit Field Name	-							LSTA
R/W	R0							R
Initial Value	0							0

This register indicates the display list transfer status of a display list transfer from VRAM. The setting of the register is "1" when transfer is in progress. The register is cleared to "0" when the transfer is completed.

LSA (displayList Source Address)

Register Address	$01FC_0040_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-							LSA																								
R/W	R0							RW																								R0
Initial Value	0							X																								0

This register sets the source address of display list transfer.
The start address for transferring a display list stored in VRAM is set at the display list transfer time. Since the lower 2 bits of the register are always taken as "0", place display lists on a 4-byte boundary. The value of the register remains unchanged during and after transfer.

LCO (displayList Count)

Register Address	$01FC_0044_H$																															
Bit Number	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bit Field Name	-							LCO																								
R/W	R0							RW																								
Initial Value	0							X																								

This register sets the display list transfer count.

The number of times of transfer in units of long words is set at the display list transfer time. If the set number of times of transfer is "1(h)", transfer occurs once. If the setting is "0(h)", it is taken as the maximum value, and transfer occurs 16M (16777216) times. The value of the register remains unchanged during and after transfer.

LREQ (displayList transfer REQuest)

Register Address	$01FC_0048_H$							
Bit Number	7	6	5	4	3	2	1	0
Bit Field Name	-							LREQ
R/W	R0							RW1
Initial Value	0							0

This register starts transfer of a display list.

Writing "1" in the register transfers a display list in VRAM to the display list FIFO.

1.22 Hi-speed SPI controller (HS-SPICNT)

This section describes the HS-SPI (High-Speed Serial Parallel Interface) controller.

This module is targeted at the CY91F59A/B.

1.22.1 Overview

HS-SPICNT supports a myriad range of SPI devices (Serial Peripheral devices). These devices are those that use the de-facto standard SPI protocol. This controller also supports the dual-bit and quad-bit SPI protocols, which are new specifications.

The interfaces of the controller are controlled exclusively with other external bus memory interfaces. (For details on the exclusive control, see "1.5.1.4 External Bus Memory Interfaces".)

1.22.2 Features

- Built-in direct mode and command sequencer mode
  - In "direct mode," HS-SPICNT operates as an SPI through the FIFO interface.
    "Command sequencer mode" can be used to assign external serial memory (FLASH, SRAM, etc.) to the system address space of the CPU.
  - In "command sequencer mode," memory access by the CPU and other AHB masters is automatically converted into signals for accessing external serial memory through this HS-SPICNT.

- Support of single-bit, dual-bit, and quad-bit SPIs
  - Single-bit: Access to 1 serial memory device uses a 1-bit wide data line.
  - Dual-bit: Access to 1 serial memory device uses a 2-bit wide data line.
  - Quad-bit: Access to 1 serial memory device uses a 4-bit wide data line.

- Support for up to 4 slave devices
  In "command sequencer mode," up to 4 serial memory devices can be assigned as slave devices in a 256 MB area in the external Flash space.
  For details on areas in the external Flash space, see "1.4.2 CY91F59A/B".
  - For 4 slave devices:
    The 256MB area in the external Flash space is divided into quarters for slaves 0 to 3.
  - For 2 slave devices:
    The 256MB area in the external Flash space is divided into halves for slaves 0 and 1.
  - For 1 slave device:
    Slave 0 is assigned to the 256MB area in the external Flash space.

- Availability of the following settings for each slave
  - Transfer speed
  - Slave select polarity
  - Serial clock polarity and phase

Warning:

"Command sequencer mode" is not necessarily available for all devices. To see whether this mode is supported by a device to be used, check the data sheet of the device.

1.22.2.1 Error Interrupts

This section describes the interrupts for the occurrences of errors.

An error interrupt is generated when even one of the status bits (UMAFS, WAFS, PVFS, DWCBSFS, and DRCBSFS) in the HSSPIn_FAULTF register has been set.
For details, see "1.22.6.10 HS-SPI Fault Interrupt Factor Register (HSSPIn_FAULTF)" and "1.22.6.11 HS-SPI Fault Interrupt Clear Register (HSSPIn_FAULTC)".

1.22.2.2 In Direct Mode

If access in direct mode causes an error interrupt (referred to below as "FAULT violation") that arises from FIFO access, the FIFO access that caused the violation is disabled. This FAULT violation does not directly stop serial transfer operations; instead, operation continues according to the FIFO state.

Section "1.22.4 Direct Mode", has a functional description that shows the relationship between FIFO states and serial transfer operations.

Also, the occurrence of the FAULT violation indirectly results in inadequate FIFO data transfer for as many cycles as the FIFO access is disabled, and the serial transfer data decreases in amount accordingly.

1.22.2.3 In Command Sequencer Mode

If access on the AHB interface in command sequencer mode affects sequential serial transfer, a FAULT violation occurs, causing the relevant AHB transactions to become invalid. As a result, memory access during the cycle that caused the FAULT violation on AHB does not reach the serial interface, and the serial transfer corresponding to this disabled memory access does not occur.

1.22.3 Operation of the Hi-Speed SPI Controller

This section describes HS-SPICNT operation.

1.22.3.1 Operating Modes

HS-SPICNT is set to either direct mode or command sequencer mode. Direct mode enables the following operations.

Transmission data can be written directly to the TX-FIFO.

Data can be read directly from the RX-FIFO and shift register.

At this time, the SPI core sends and receives data in the FIFO via the serial interface. Section "1.22.4 Direct Mode", provides details on direct mode.

In command sequencer mode, HS-SPICNT assigns external serial memory to the system address space of the CPU. In this way, up to 4 external serial memory devices are assigned inside the system with 4 slave selects sent to the individual serial memory devices. When the system bus master accesses the assigned area of any of the serial memory devices, HS-SPICNT performs the write or read operation for the corresponding external serial memory. Transactions on AHB are subject to wait control during the period of this serial memory access. Section "1.22.5 Command Sequencer Mode", provides details on command sequencer mode.

1.22.3.2 Clocking Modes

With the setting of each of the HSSPIn_PCC0 to PCC3:ACES, CPOL, and CPHA control bits, 4 clocking modes can be set for peripheral devices (external serial memory) independently from one another. The settings of these bits and the RTM bit in the same register determines the timing relationship between the following in HS-SPICNT:

- Serial data input-output
- SPI clock input-output

Table 1-45 shows this relationship.

Table 1-45. Clocking Modes

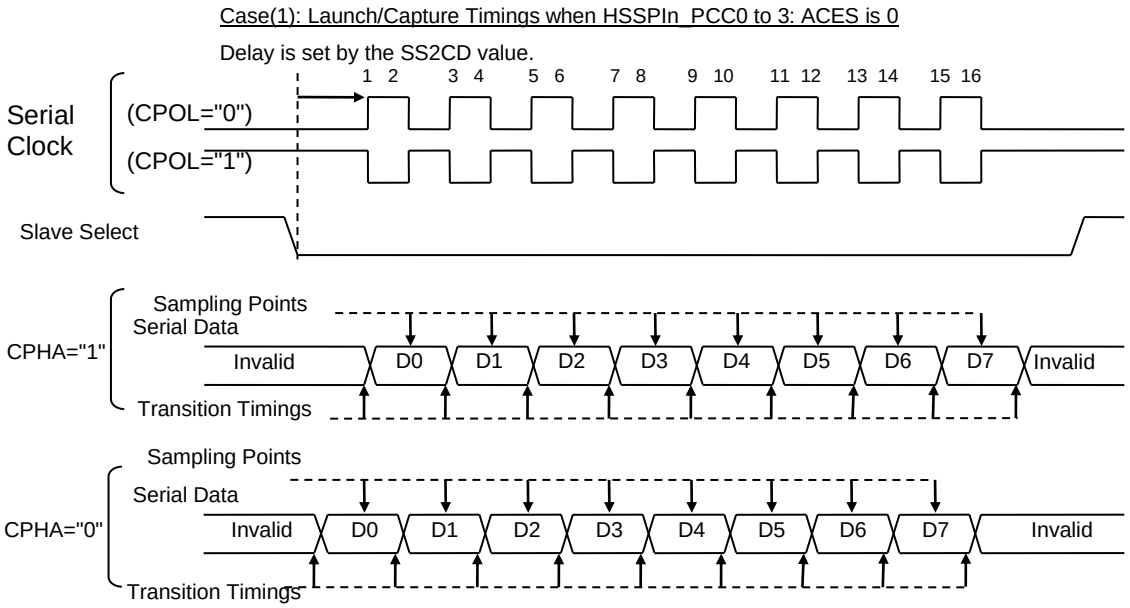
Mode	ACES (Active Clock Edges Are Same)	CPOL (Clock Polarity)	CPHA (Clock Phase)	Description
Mode 0	0	0	0	The output data changes a half cycle before the first rising edge of the serial clock, and then changes at the falling edge of this serial clock.
				The input data is captured at the rising edge of the serial clock.
Mode 1		0	1	The output data changes at the rising edge of the serial clock.
				The input data is captured at the falling edge of the serial clock.
Mode 2		1	0	The output data changes a half cycle before the first falling edge of the serial clock, and then changes at the rising edge of this serial clock.
				The input data is captured at the falling edge of the serial clock.
Mode 3		1	1	The output data changes at the falling edge of the serial clock.
				The input data is captured at the rising edge of the serial clock.
Settings other than above				Don't care

Figure 1-113 shows the settings of each of the ACES, CPOL, and CPHA control bits, and the corresponding relationships with serial data and the serial clock.

Warning:

In command sequencer mode, apply the same settings to all of the HSSPIn_PCC0 to PCC3 registers.

Figure 1-113. Serial Interface Clock Operating Modes



The HSSPIn_PCC0 to PCC3:SS2CD[1:0] value controls the periods shown in the above figure. The following table shows the relationship between setting values and delay amounts.

SS2CD[1:0]	CPHA	Setup Time from Slave Select Assertion to Serial Clock [Number of Serial Clock Cycles]
0	0	For CDSR[6:0]=0, 1.5 [cycles] of serial clock (1.5 [cycles] of GSSCGCLK) For CDSR[6:0]>0, 1.5 [cycles] of serial clock (CDRS[6:0] x 3 [cycles] of GSSCGCLK)
	1	For CDSR[6:0]=0, 1.0 [cycles] of serial clock (1.0 [cycles] of GSSCGCLK) For CDSR[6:0]>0, 1.0 [cycles] of serial clock (CDRS[6:0] x 2 [cycles] of GSSCGCLK)
1	0	For CDSR[6:0]=0, 2.5 [cycles] of serial clock (2.5 [cycles] of GSSCGCLK) For CDSR[6:0]>0, 2.5 [cycles] of serial clock (CDRS[6:0] x 5 [cycles] of GSSCGCLK)
	1	For CDSR[6:0]=0, 2.0 [cycles] of serial clock (2.0 [cycles] of GSSCGCLK) For CDSR[6:0]>0, 2.0 [cycles] of serial clock (CDRS[6:0] x 4 [cycles] of GSSCGCLK)
2	0	For CDSR[6:0]=0, 3.5 [cycles] of serial clock (3.5 [cycles] of GSSCGCLK) For CDSR[6:0]>0, 3.5 [cycles] of serial clock (CDRS[6:0] x 7 [cycles] of GSSCGCLK)
	1	For CDSR[6:0]=0, 3.0 [cycles] of serial clock (3.0 [cycles] of GSSCGCLK) For CDSR[6:0]>0, 3.0 [cycles] of serial clock (CDRS[6:0] x 6 [cycles] of GSSCGCLK)
3	0	For CDSR[6:0]=0, 4.5 [cycles] of serial clock (4.5 [cycles] of GSSCGCLK) For CDSR[6:0]>0, 4.5 [cycles] of serial clock (CDRS[6:0] x 9 [cycles] of GSSCGCLK)
	1	For CDSR[6:0]=0, 4.0 [cycles] of serial clock (4.0 [cycles] of GSSCGCLK) For CDSR[6:0]>0, 4.0 [cycles] of serial clock (CDRS[6:0] x 8 [cycles] of GSSCGCLK)

1.22.3.3 Deselect Time

Deselect time means the time period from slave select deassertion to slave select assertion. The deselect time is secured according to the set values of the HSSPIn_PCC0 to PCC3:RDDSEL[1:0] bits or the HSSPIn_PCC0 to PCC3:WRDSEL[3:0] bits. The HSSPIn_PCC0 to PCC3:WRDSEL[3:0] bits are used under the following conditions, and the HSSPIn_PCC0 to PCC3:RDDSEL[1:0] bits are used under other conditions.

Use conditions of the HSSPIn_PCC0 to PCC3:WRDSEL[3:0] bits

- For a change made to the HSSPIn_RDCSDC0 to RDCSDC7:DEC bit or RDCSDATA[7:0] bits
- For a change made to the HSSPIn_WRCSDC0 to WRCSDC7:DEC bit or WRCSDATA[7:0] bits
- To use the WRCSDC0 to WRCSDC7 registers after using the HSSPIn_RDCSDC0 to RDCSDC7 registers (For write access after read access)
- To use the RDCSDC0 to RDCSDC7 registers after using the WRCSDC0 to WRCSDC7 registers (For read access after write access)
- To use the WRCSDC0 to WRCSDC7 registers (Write access)
  For a deselect occurrence due to the idle timer

1.22.3.4 SPI Protocols

HS-SPICNT supports the following SPI protocols:

- Single-bit SPI
- Dual-bit SPI
- Quad-bit SPI

In direct mode, 1 of these 3 modes is selected according to the TRP[1:0] bit setting in the HSSPIn_DMTRP register. Similarly, in command sequencer mode, one of them is selected according to the MBM bit setting in the HSSPIn_CSCFG register.

Single-bit SPI

The single-bit SPI is a communication protocol for a full-duplex line. When configured to use the single-bit SPI, HS-SPICNT receives 1-bit wide serial data from QSPI_SIO1 and sends such data from QSPI_SIO0 at the same time. The upper data bits in QSPI_SIO3 and QSPI_SIO2, which are not used, stay in the Hi-Z state during the use of the single-bit SPI.

In direct mode, the HSSPIn_DMTRP:TRP[3:0] bits are set to any of the following:

- TX-and-RX single-bit mode
- TX-Only single-bit mode
- RX-Only single-bit mode

Dual-bit SPI

Operation in this mode becomes that for a half-duplex line, with 2-bit wide data lines (QSPI_SIO1 and QSPI_SIO0) being used. Data transmission/reception operations are performed exclusively. In this mode, the upper data bits in QSPI_SIO3 and QSPI_SIO2, which are not used, stay in the Hi-Z state.

In direct mode, the HSSPIn_DMTRP:TRP[3:0] bits are set to any of the following:

- TX-Only dual-bit mode
- RX-Only dual-bit mode

Quad-bit SPI

Operation in this mode becomes that for a half-duplex line, with 4-bit wide data lines (QSPI_SIO3, QSPI_SIO2, QSPI_SIO1, and QSPI_SIO0) being used. Data transmission/reception operations are performed exclusively.

In direct mode, the HSSPIn_DMTRP:TRP[3:0] bits are set to any of the following:

- TX-Only quad-bit mode
- RX-Only quad-bit mode

1.22.3.5 Shift Direction

The SDIR bit in the HS-SPICNT peripheral communication setting registers (HSSPIn_PCC0 to PCC3) determines the shift direction of the shift register used for data transfer. If the SDIR bit is "0", data transfer starts with the highest bit and ends with the lowest bit to send the bits sequentially in the order of highest bit to lowest bit. If these bits are arranged left to right from the highest bit to lowest bit, the shift is in the left direction. If the SDIR bit is "1" and the bits are arranged in the same way, the shift is in the right direction. Regardless of the shift direction, write/read access to the data register always places the lowest bit at the location of bit0.

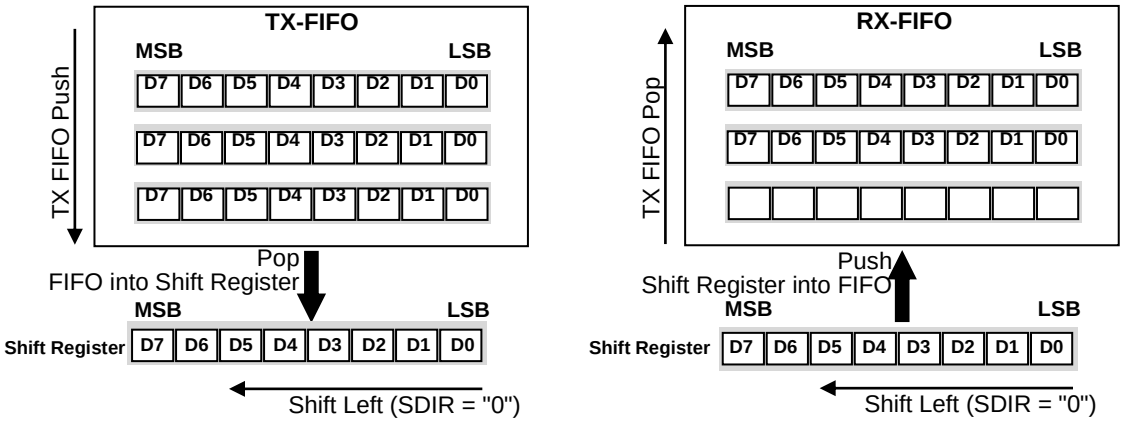
Figure 1-114 and Figure 1-115 show shift register operations for each of the single-bit SPI, dual-bit SPI, and quad-bit SPI. The waveforms of these operations are based on the following register settings:

- HSSPIn_PCC0 to PCC3:CPOL bit = "0"
- HSSPIn_PCC0 to PCC3:CPHA bit = "0"
- HSSPIn_FIFOCFG:FWIDTH bits = "00"

Figure 1-114 and Figure 1-115 show flows where transmission data is set in the shift register via the TX-FIFO. Here, you need to also note that transmission data may be set from other registers too (the following 2 registers):

- HSSPIn_RDCSDC0 to RDCSDC7:RDCSDATA[7:0]
- HSSPIn_WRCSDC0 to WRCSDC7:WRCSDATA[7:0]

Figure 1-114. Shift Direction (for the Settings of CPOL="0", CPHA="0", SDIR="0", and FWIDTH="00")



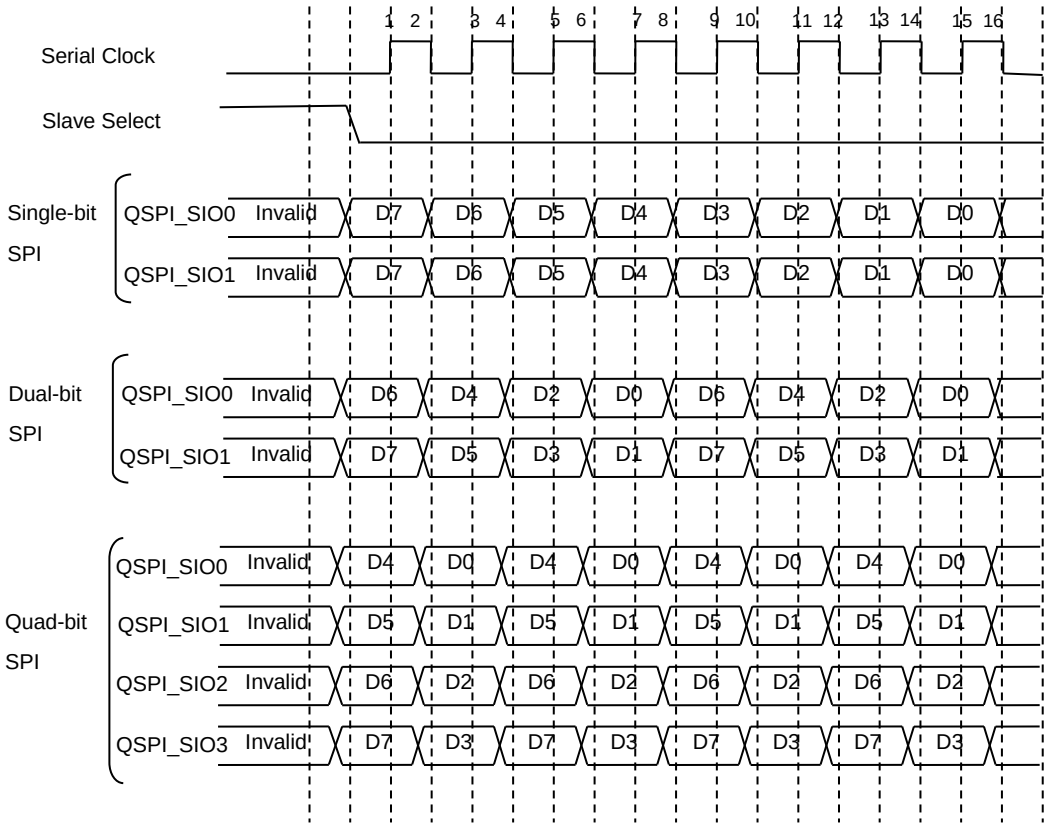
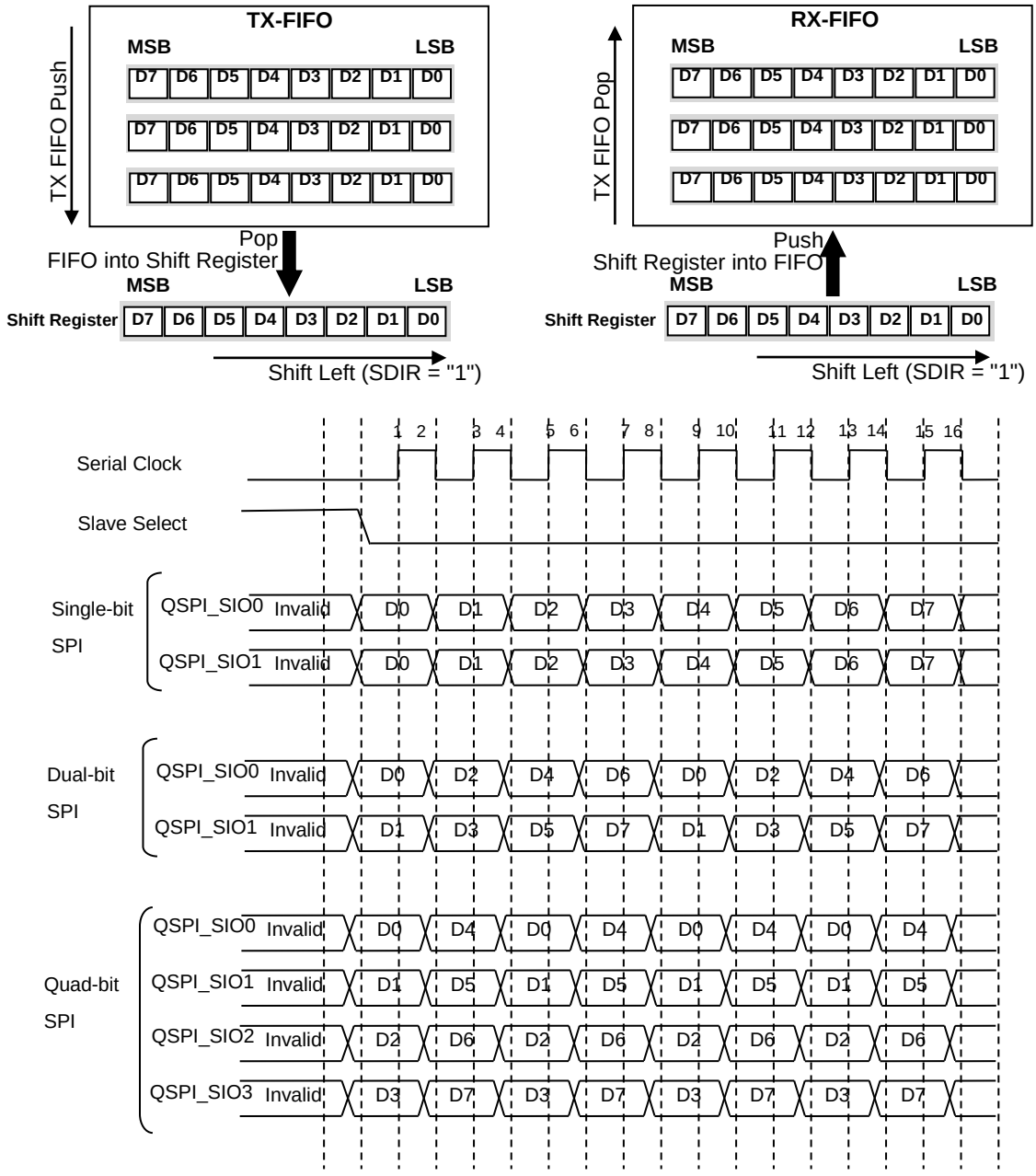


Figure 1-115. Shift Direction (for the Settings of CPOL="0", CPHA="0", SDIR="1", and FWIDTH="00")



1.22.3.6 Resynchronization

The cases where data is not correctly synchronized in serial transfer because of a bottleneck due to resynchronization processing must be considered. The HSSPIn_PCC0 to PCC3:SAFESYNC bit setting of "1" can prevent such a condition.

The conditions in which the SAFESYNC bit will be set to "0" or "1" are determined by:

- The transfer protocol type,
- The bit width of the shift register,
- The frequency ratio of the AHB clock to serial clock (integral multiple).

HS-SPICNT temporarily stops the serial transfer in progress when "1" is set in the SAFESYNC bit. This operation is necessary for resynchronizing data, and it occurs when both of the following conditions are satisfied.

- The HSSPIn_PCC0 to PCC3:SAFESYNC bit is set to "1".
- "The shift register is 8 bits long and the serial interface is set to dual-bit or quad-bit mode," or "the shift register is 16 bits long and the serial interface is set to quad-bit mode."

Therefore, to determine whether to insert a wait for resynchronization when the HSSPIn_PCC0 to PCC3:SAFESYNC bit is set to "1", HS-SPICNT checks the shift register length after a block of data is transferred. Also, this SAFESYNC bit setting of "1" does not always mean inserting a wait in data transfer. That only occurs under a specific condition that is in addition to the combination of the shift register length and SPI protocol type (single/dual/quad). This operation does not greatly affect the bandwidth of the serial transfer.

The following is supplementary information on the relationship between the shift register length and the SAFESYNC bit.

In direct mode, the shift register length is determined by the bit width of the transmission/reception FIFO. The bit width of this transmission/reception FIFO is set in HSSPIn_FIFOCFG:FWIDTH. However, note the following points about this setting.

- Data may remain in the TX-FIFO, waiting to be transferred. The shift register length used to transfer all of this data is at least 8 bits. HS-SPICNT checks this shift register length to determine whether resynchronization must be stabilized.
- The smallest shift register length among those ever used in data transfer is used, especially when the following 2 conditions are satisfied.
  - ☐ Counter mode (using the HSSPIn_DMBCC and HSSPIn_DMBCS registers) is used to stop serial transfer.
  - ☐ The amount of data to transfer (setting in the HSSPIn_DMBCC register) is not divisible by the FIFO bit width (HSSPIn_FIFOCFG:FWIDTH).

  As an example, think of a case where the following 2 conditions are satisfied.

  - ☐ HSSPIn_DMBCC:BCC is set to 10 bytes.
  - ☐ HSSPIn_FIFOCFG:FWIDTH is set to 4 bytes.

  In this case, HS-SPICNT operates as follows:

  1. The shift register length becomes 4 bytes, and data is transferred twice in units of 4 bytes.
  2. The remaining 2 fractional bytes are transferred in the one time of the last transfer. At this point in time, the TX-FIFO does not contain any remaining data waiting to be transferred.

  Therefore, the minimum value of the shift register length during the above data transfer is 2 bytes. HS-SPICNT determines whether it needs to stabilize resynchronization by checking this minimum value of the shift register length.

Table 1-46 lists conditions in which HS-SPICNT sets "1" in the SAFESYNC bit in direct mode.

Table 1-46. SAFESYNC Bit Setting Conditions in Direct Mode

Mode	Shift Register Length	Protocol	Condition in Which SAFESYNC Bit Setting Must Be "1"	Maximum Frequency of Serial Clock
Direct mode	8 bits	Single-bit	SAFESYNC unnecessary	Fsclk = Fhclk / 2
		Dual-bit	Fsclk = Fhclk / 2	
		Quad-bit	Serial clock is GSSCGCLK divided into 5 parts or less	
	16 bits	Single-bit	SAFESYNC unnecessary	
		Dual-bit		
		Quad-bit	Fsclk = Fhclk / 2	
	24 bits	Single-bit	SAFESYNC unnecessary	
		Dual-bit		
		Quad-bit		
	32 bits	Single-bit	SAFESYNC unnecessary	
		Dual-bit		
		Quad-bit		

(Fsclk: Serial clock; Fhclk: GSSCGCLK)

In command sequencer mode, the shift register length remains 8 bits when a command sequence is sent. The shift register length remains the same as the AHB Bus width during a data exchange with external serial memory.

Warning:

- The following clock frequency restriction between clocks is applicable:
  GSSCGCLK frequency divided by 2 is greater than or equal to serial clock
- This relationship must always be established regardless of the shift register length, SPI protocol (bit width), SAFESYNC setting value, and clock division ratio (CDRS setting value).

Table 1-47 lists conditions in which HS-SPICNT sets "1" in the SAFESYNC bit in command sequencer mode.

Table 1-47. SAFESYNC Bit Setting Conditions in Command Sequencer Mode

Mode	AHB Transfer Size Assigned to Memory	Protocol	Condition in Which SAFESYNC Bit Setting Must Be "1"	Maximum Frequency of Serial Clock
Command sequencer mode	8 bits	Single-bit	SAFESYNC unnecessary	Fsclk = Fhclk / 2
		Dual-bit	SAFESYNC unnecessary	
		Quad-bit	Fsclk = Fhclk / 2	
	16 bits	Single-bit	SAFESYNC unnecessary	
		Dual-bit		
		Quad-bit	SAFESYNC unnecessary	
	32 bits	Single-bit	SAFESYNC unnecessary	
		Dual-bit		
		Quad-bit		

(Fsclk: Serial clock; Fhclk: GSSCGCLK)

Warning:

- The following clock frequency restriction between clocks is applicable:
  GSSCGCLK frequency divided by 2 is greater than or equal to serial clock
- This relationship must always be established regardless of the AHB transfer size, SPI protocol (bit width), SAFESYNC setting value, and clock division ratio (CDRS setting value).

1.22.4 Direct Mode

In direct mode, the CPU directly controls data transfer operations on the serial interface. Setting "0" in the CSEN bit in the HSSPIn_MCTRL register makes direct mode available. In this mode, the transmission/reception FIFO is used as an intermediate buffer for serial transfer data.

This section describes this direct mode.

1.22.4.1 FIFO

HS-SPICNT has 2 FIFOs that temporarily store data. Of the 2 FIFOs, one is for transmission and the other is for reception.

Any of the following modes can be set in HSSPIn_DMTRP:TRP:

- TX-Only (Only the TX-FIFO is available.)
- RX-Only (Only the RX-FIFO is available.)
- TX-and-RX (Both the TX-FIFO and RX-FIFO are available.)

FIFO Size

Each TX/RX-FIFO has a depth of 16 stages, and each stage has a data width of 32 bits. The HSSPIn_FIFOCFG:FWIDTH setting can adjust this data width.

The shift register connected to this FIFO has a width of 32 bits. However, the width of the shift register becomes the same as that of the FIFO according to the above FWIDTH setting.

In addition to the data width of 32 bits, the TX-FIFO contains another upper bit, which serves as a flag bit. The data in each stage in the TX-FIFO individually contains this flag bit. This flag bit is called the TXCTRL bit. According to this flag information, HS-SPICNT determines whether to output data from the TX-FIFO on the SPI or to set the Hi-Z state for data output on the SPI.

If the TXCTRL bit is "1", HS-SPICNT decodes the lowest bit (bit [0]) of the relevant data in the TX-FIFO. Table 1-48 lists possible combinations of the TXCTRL bit and the lowest bit of the relevant data in the TX-FIFO.

Table 1-48. Tristate Control during Data Output

TXCTRL	TX-FIFO Data Bit [0]	Description
0	Don't care	Serial data output does not enter the Hi-Z state during the data output period (and stays in the driven state).
1	0	Serial data output is in the Hi-Z state for the duration of 1-byte data. The relevant data is not transmitted to the TX-FIFO.
1	1	Data is transferred as follows. 1. Bits [7:4] in the relevant data are transmitted to the TX-FIFO. The transmission order of these bits depends on the HSSPIn_PCC0 to PCC3:SDIR bit setting. 2. Then, serial data (SDATA) output stays in the Hi-Z state for the duration of 4-bit data.

For practical reasons, data with the TXCTRL bit setting of "1" in the TX-FIFO is 1 byte wide. With the TXCTRL bit associated with the data in the TX-FIFO, software can issue commands in direct mode. In order for some of the currently popular memory devices with quad-bit support to issue commands, serial data (SDATA) must be set in the Hi-Z state. This Hi-Z period is defined as follows:

- A cycle period to be inserted as a "dummy"
- A period of the lower 4 bits of "mode bits"

Warning:

If "1" is set in the TXCTRL bit, written data is handled as 8 bits regardless of the FIFO bit width. The LSB in the effective bit range set as the FWIDTH value corresponds to the above-mentioned "TX-FIFO Data Bit [0]."

FIFO Access

The FIFOs are accessible only with the same bus width as the transmission/reception FIFO bit width (HSSPIn_FIFOCFG:FWIDTH). However, if the bit width of the transmission/reception FIFO is 24 bits, 32-bit width access is possible.

The targets of this access are the HSSPIn_RXFIFO0 to RXFIFO15 registers and the HSSPIn_TXFIFO0 to TXFIFO15 registers.

When the HSSPIn_RXFIFO0 to RXFIFO15 registers are read, the data in the RX-FIFO comes out as is, which is 1-word (32-bit) wide data. If the set width of the RX-FIFO is 8 bits, "0" appears in bits [31:8] of the read data value. Similarly, "0" appears in bits [31:16] for a 16-bit width, and in bits [31:24] for a 24-bit width.

When data is written to the HSSPIn_TXFIFO0 to TXFIFO15 registers, the HSSPIn_FIFOCFG:TXCTRL bit value is added as the highest bit of the data. Then, it is written together with the data to the TX-FIFO. In response, the HS-SPI outputs the effective range of the data written in the TX-FIFO (LSB side in a word) to the serial interface. This effective range is determined in units of bytes by the bit width of the TX-FIFO (HSSPIn_FIFOCFG:FWIDTH value).

1.22.4.2 Service Requests

In operation in direct mode, interrupts to the CPU are generated according to the amount of data accumulated in the TX-FIFO or RX-FIFO. The threshold values that are conditions for generating these interrupts can be set in registers.

Interrupt Service Request Originating from the FIFO Level

One possible way to reference the amount (referred to below as "level") of data remaining in a FIFO is to read a register. An interrupt originating from the FIFO is generated from the relationship between the FIFO level and the threshold value that is set separately.

In the transmission operation, the interrupt flag is set when "the TX-FIFO level becomes equal to or less than the threshold value." The following bits have this control:

- TX-FIFO level: HSSPIn_DMSTATUS:TXFLEVEL
- TX-FIFO threshold value: HSSPIn_FIFOCFG:TXFTH

In the reception operation, the interrupt flag is set when "the RX-FIFO level exceeds the threshold value." The following bits have this control:

- RX-FIFO level: HSSPIn_DMSTATUS:RXFLEVEL
- RX-FIFO threshold value: HSSPIn_RXFIFOCFG:RXFTH

If HS-SPICNT is set to "TX-Only mode," only the TX-FIFO is used. If HS-SPICNT is set to "RX-Only mode," only the RX-FIFO is used.

Service Request Originating from Transfer Completion

During operation in direct mode, HS-SPICNT asserts an interrupt signal when the slave select is deasserted. The following interrupt flags become "1" for events that deassert the slave select:

- HSSPIn_TXF:TSSRS bit
- HSSPIn_RXF:RSSRS bit

These interrupt flags have an "interrupt clear" bit and "interrupt enable" bit, respectively.

1.22.4.3 SPI Transfer

HS-SPICNT, as the SPI master, controls the SPI slave select. The HSSPIn_DMPSEL:PSEL bit is used for that selection.

Communication Settings

There are several settings for the HS-SPICNT serial interface. They include settings for the frequency, polarity, and phase of the serial interface clock. The slave select polarity is also included. These serial interface communication-related settings depend on the connected SPI device. Settings can be made individually for up to 4 SPI devices when HS-SPICNT operates in direct mode.

These settings use the HSSPIn_PCC0 to PCC3 registers.

Start of Serial Transfer

If HS-SPICNT satisfies all of the following conditions, writing "1" in HSSPIn_DMSTART:START starts HS-SPICNT serial transfer.

- HS-SPICNT is in the enabled state (HSSPIn_MCTRL:MEN is "1").
- HS-SPICNT is in direct mode (HSSPIn_MCTRL:CSEN is "0").

If all of the following conditions are satisfied, writing "1" in HSSPIn_DMSTART:START delays the start of serial transfer by HS-SPICNT until software writes data in the TX-FIFO.

- The HSSPIn_DMTRP:TRP setting enables transmission (TX-Only or TX-and-RX).
- The TX-FIFO is empty.

The following occurs while HS-SPICNT is delayed from starting serial transfer (until software writes data in the TX-FIFO).

1. When software writes "0" in HSSPIn_MCTRL:MEN, the next transfer is canceled, and HS-SPICNT enters the disabled state.
2. If the transfer length is controlled in counter mode, the HSSPIn_DMBCS register is initialized to the value in the HSSPIn_DMBCC register. This initialization occurs when HS-SPICNT starts serial transfer. The HSSPIn_DMBCS register is held at "0" until this transfer begins.

Once HSSPIn_DMSTART:START is set to "1", this bit cannot be cleared by the writing of any value to it. HSSPIn_DMSTART:START is automatically cleared to "0" when HS-SPICNT starts serial transfer. Also, note the following points.

- Writing "1" in this bit when HSSPIn_DMSTART:START is "1" does not result in any operation.
- Writing "1" in this bit when HSSPIn_DMSTART:START is "0" and serial transfer is in progress does not affect the transfer operation. After the current transfer processing is finished, a new serial transfer can be submitted.

Stop Due to Insufficient Transmission Data or Insufficient Free Space in the RX-FIFO

During serial interface operation, a transfer in progress can be temporarily stopped. To do so, temporarily stop the serial clock while leaving the slave select enabled.

HS-SPICNT automatically stops the serial clock while waiting for data to be written to the TX-FIFO or be read from the RX-FIFO. The corresponding slave select is kept asserted during this period. This means that the transfer for the slave has not ended yet. This stopped state is automatically canceled when the FIFO becomes available again, and the transfer operation resumes.

HS-SPICNT uses different methods to stop serial transfer, depending on the data transfer direction (TX-Only/RX-Only/TX-and-RX).

- TX-Only mode:

  Serial transfer stops when the TX-FIFO and shift register become empty.
- RX-Only mode:

  Serial transfer stops when the shift register becomes full.
- TX-and-RX mode:

  Serial transfer stops under the following condition (1) or (2).

  1. The TX-FIFO and shift register become empty.
  2. The RX-FIFO and shift register become full and stop.

Transfer Length Control

There are 2 methods for data transfer length control (using the slave select):

- Counter mode
- Software flow control mode

HSSPIn_DMCFG:SSDC can set these modes.

In counter mode, the number of data bytes to transfer to the serial interface is set in HSSPIn_DMBCC:BCC by CPU access. When serial transfer begins, HS-SPICNT starts counting the number of data bytes to transfer. HS-SPICNT deasserts the slave select when the transfer amount reaches the amount set as the above BCC field value.

In software flow control mode, CPU access uses HSSPIn_DMSTOP:STOP to control the transfer amount. The method of cancelling the slave select varies depending on the mode (TX-Only/RX-Only/TX-and-RX) that HS-SPICNT is operating in.

- TX-Only mode:
  Transfer stops when HSSPIn_DMSTOP:STOP is "1" and all the data in the TX-FIFO has been transmitted.
- RX-Only mode:
  Transfer stops when HSSPIn_DMSTOP:STOP is "1" and the shift register shifts up to the data width of the RX-FIFO.
- TX-and-RX mode:
  Transfer stops when HSSPIn_DMSTOP:STOP is "1" and all the data in the TX-FIFO has been transmitted.

1.22.5 Command Sequencer Mode

In command sequencer mode, HS-SPICNT operates as the SPI master and controls serial memory devices. It uses 4 slave selects to assign serial memory devices of the same type to the system memory. When the CPU or another bus master accesses the system area to which these serial memory devices are assigned, HS-SPICNT automatically converts the access into a read/write operation for serial memory.

This section describes command sequencer mode.

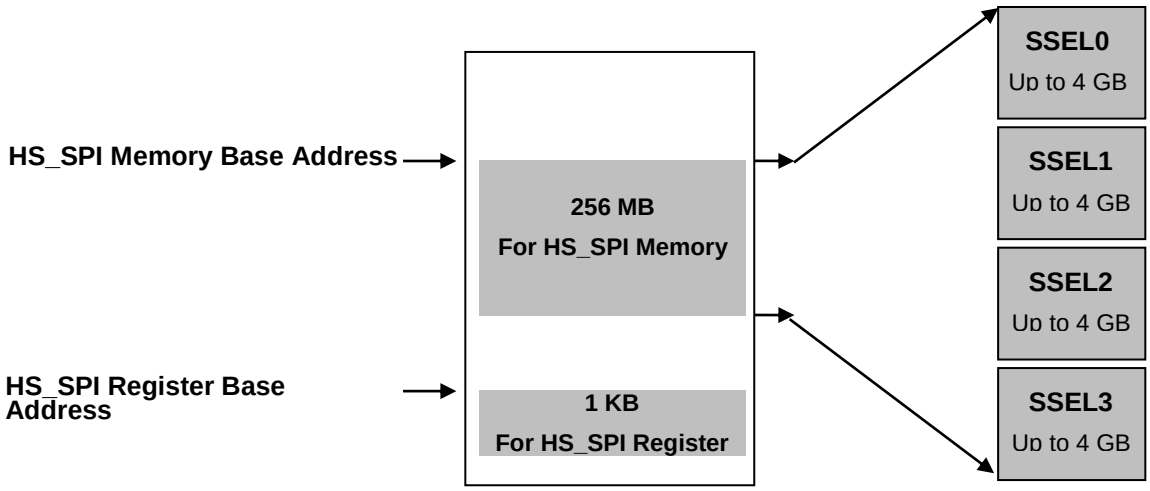
1.22.5.1 Assignment to Memory

In command sequencer mode, up to 4 serial memory devices can be assigned to the external Flash space. The 4 slave select control signals output from HS-SPICNT are connected to these serial memory devices. These memory devices, up to 4 of which can be connected, must be of the same type.

In command sequencer mode, these memory devices are assigned to a 256MB memory area in the external Flash space. The address extension mechanism can implement an address space of theoretically up to 4GB for each slave select. Each of the extended address values is generated as a 32-bit address value of a slave from a combination of the highest bits in the address extension register (HSSPIn_CSAEXT[18:0]) as higher bits and several bits from an AHB address value. This address value is used to select and access each memory device.

The 256MB address area in the external Flash space is thus virtually assigned to a 16GB memory space. Figure 1-116 shows this concept.

Figure 1-116. Memory Device Assignment for Each Slave Select



1.22.5.2 Slave Selection

HSSPIn_CSCFG:MSEL indicates the range of serial memory assigned to the external Flash space. The command sequencer of HS-SPICNT asserts 1 of the 4 slave selects based on the following:

- HSSPIn_CSCFG:MSEL
- AHB address output from the CPU or another AHB master

Table 1-49 shows their combinations in detail.

For example, if HSSPIn_CSCFG:MSEL is set so that each slave select has an 8 KB area in the system space, slaves are selected as follows.

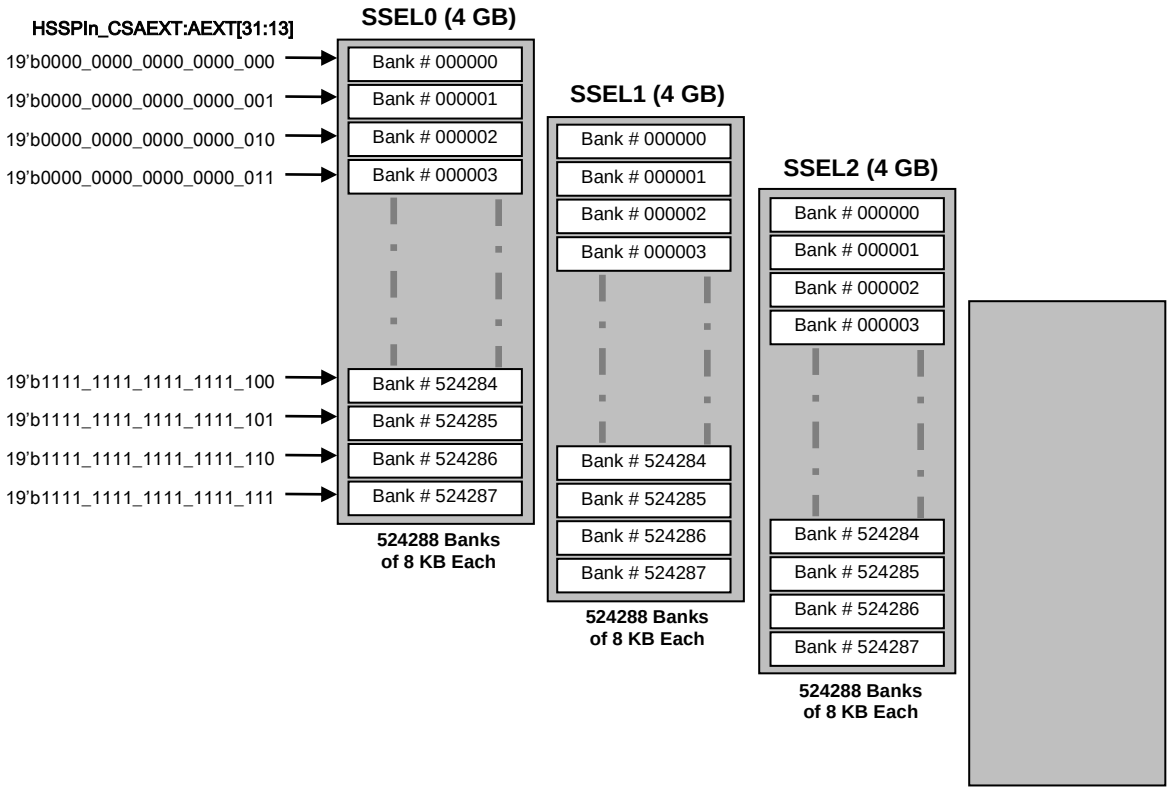
- For access by AHB to an address in the 8KB range starting at the "HS-SPI Memory Base Address" in the system space, slave select 0 is asserted. If this address is in the 8KB range starting at the "HS-SPI Memory Base Address + 8KB," slave select 1 is asserted. Similarly, slave selects 2 and 3 are their extensions. If the address is in a range starting at "HS-SPI Memory Base Address + 32KB," HSSPIn_FAULT:UMAFS is set to "1" as an interrupt flag because the address is outside the specified range.

1.22.5.3 32-bit Memory Address

A 256MB area in the external Flash space can be assigned to an area of up to 4GB for each slave select line. The address extension mechanism is used at this time. Each serial memory device configures memory banks so that it appears in the external Flash space. HSSPIn_CSCFG:MSEL sets the size of each bank. The HSSPIn_CSAEXT register switches the range of each memory bank to enable access to another area in serial memory. In this way, individual memory spaces of up to 4GB are accessible.

Figure 1-117 shows this concept. In this figure, a 4GB memory space is divided into 524,288 banks, and HSSPIn_CSCFG:MSEL is set to "0000".

Figure 1-117. Assignment of a 4GB Address Space Using Slave Select/Internal Slave Bank Switching



The lower bits of the AHB address for accessing HS-SPICNT are used as an address offset value in the selected memory bank. The selection of a memory bank uses the address extension bits.

An appropriate combination of the address extension register and AHB address provides a 32-bit wide address space on the serial interface. Table 1-49 shows these combinations.

Table 1-49. Assignment to Addresses in the External Flash Space

HSSPIn_CSCFG: MSEL[3:0]	Memory Bank Size of Each Slave Select, or AHB Address Range Associated with Each Slave Select	Slave Select Line Used	Bit Range of HSSPIn_CSAEXT Register Used to Select Memory Bank	Bit Range of AHB Address Used to Specify Address in Memory Bank
0000	8KB	SSEL0, SSEL1, SSEL2, and SSEL3	AEXT[31:13]	Address[12:0]
0001	16KB		AEXT[31:14]	Address[13:0]
0010	32KB		AEXT[31:15]	Address[14:0]
0011	64KB		AEXT[31:16]	Address[15:0]
0100	128KB		AEXT[31:17]	Address[16:0]
0101	256KB		AEXT[31:18]	Address[17:0]
0110	512KB		AEXT[31:19]	Address[18:0]
0111	1MB		AEXT[31:20]	Address[19:0]
1000	2MB		AEXT[31:21]	Address[20:0]
1001	4MB		AEXT[31:22]	Address[21:0]
1010	8MB		AEXT[31:23]	Address[22:0]
1011	16MB		AEXT[31:24]	Address[23:0]
1100	32MB		AEXT[31:25]	Address[24:0]
1101	64MB		AEXT[31:26]	Address[25:0]
1110	128MB	Only SSEL0 and SSEL1	AEXT[31:27]	Address[26:0]
1111	256MB	Only SSEL0	AEXT[31:28]	Address[27:0]

The rightmost 2 columns in Table 1-49 show the relationship between HSSPIn_CSAEXT:AEXT and AHB addresses. The final 32-bit address spaces on the serial interface are implemented based on this relationship.

The final generated address is 32 bits wide, and software can determine the number of data bytes to transfer to serial memory. This selection is made in the address phase of the memory read/write command.

Table 1-50. Slave Selection Using AHB Address Values

HSSPIn_CSCFG: MSEL[3:0]	Memory Bank Size of Each Slave Select, or AHB Address Range Associated with Each Slave Select	Bit Range of Addresses Involved in Slave Select		Slave Select Used to Make Selection
		Bit Range	Bit Value	
0000	8 KB	Address[14:13]	0	0 (SSEL0)
			1	1 (SSEL1)
			2	2 (SSEL2)
			3	3 (SSEL3)
0001	16 KB	Address[15:14]	0	0 (SSEL0)
			1	1 (SSEL1)
			2	2 (SSEL2)
			3	3 (SSEL3)
1101	64 MB	Address[27:26]	0	0 (SSEL0)
			1	1 (SSEL1)
			2	2 (SSEL2)
			3	3 (SSEL3)
1110	128 MB	Address[27]	0	0 (SSEL0)
			1	1 (SSEL1)
1111	256 MB	-	-	0 (SSEL0)

The upper bits of bit1 to bit2 of an AHB address shown in Table 1-49 are used for the slave select of the serial interface. Table 1-50 has supplementary information on that relationship. For example, if HSSPIn_CSCFG:MSEL[3:0] is "0000" (each slave select is assigned with a size of 8KB in the external Flash space), address[12:0] is output to the serial interface, and address[14:13] is used for slave selection (SSEL0 to SSEL3). If HSSPIn_CSCFG:MSEL is "1101" (13 in decimal notation) or less, the highest 2 bits of the address range in the above table are used for slave selection. If HSSPIn_CSCFG:MSEL is "1110" (14 in decimal notation), the highest bit of the address is used. If HSSPIn_CSCFG:MSEL is "1111" (15 in decimal notation), no upper bits of the address are used (only SSEL0 is used).

1.22.5.4 Command Sequence Settings

The command sequencer is capable of memory read access. In addition, setting HSSPIn_CSCFG:SRAM to "1" when writable serial memory is assigned to the external Flash space also enables write access by the command sequencer.

A flow of command phases (instruction phase, address phase, and data phase) is generated in command sequencer mode of HS-SPICNT. Each phase that software sets in HSSPIn_RDCSDC0 to HSSPIn_RDCSDC7 or HSSPIn_WRCSDC0 to HSSPIn_WRCSDC7 in the register area is output on SDATA.

Issuing a Memory Read Command Sequence

In the memory read operation, command phases are executed in the sequence in which they were set in 8 registers (HSSPIn_RDCSDC0 to HSSPIn_RDCSDC7), to read data from memory. These registers are referenced one by one in order from HSSPIn_RDCSDC0 to HSSPIn_RDCSDC7. Figure 1-118 shows details of this operation.

The DEC bit in each register indicates whether the data type must be decoded. Table 1-51 shows the relationship between the DEC bit and data type bit field (RDCSDATA[2:0]). RDCSDATA[7:0] is handled as data as is when the DEC bit is "0".

Figure 1-118. Memory Read Command Sequence List

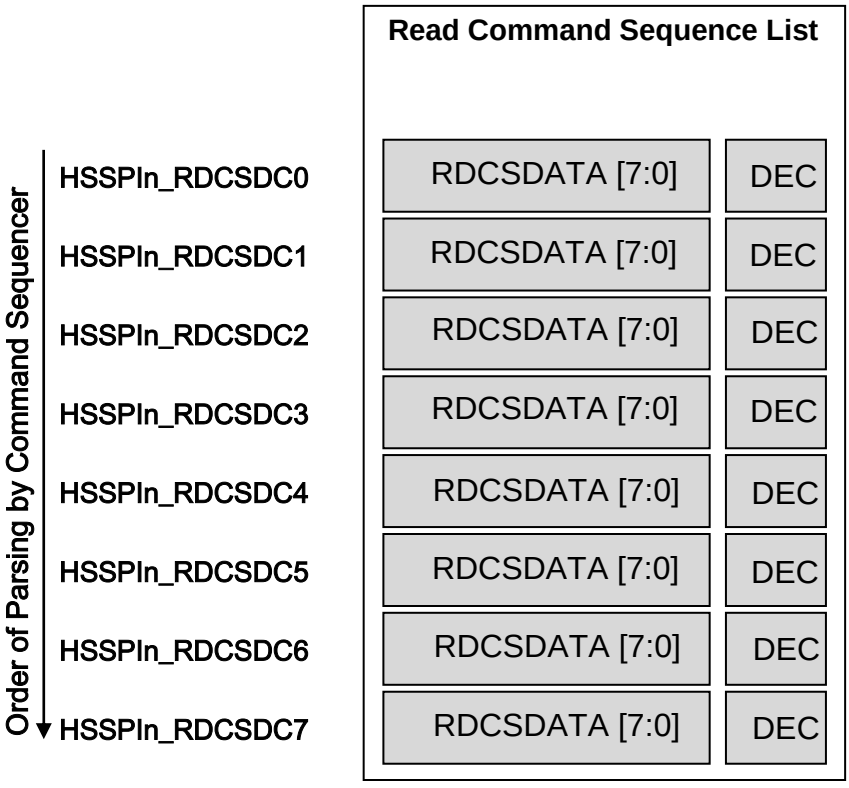


Table 1-51. Read Command Sequence List

DEC	RDCSDATA [2:0]	Description
0	Don't care	Transmits RDCSDATA[7:0] as is.
1	000	Transmits address[7:0] to access serial memory.
1	001	Transmits address[15:8] to access serial memory.
1	010	Transmits address[23:16] to access serial memory.
1	011	Transmits address[31:24] to access serial memory.
1	100	Sets SDATA to the Hi-Z state for the duration of 1-byte data.
1	101	Transmits 4 bits of 1-byte data and then keeps SDATA in the Hi-Z state for the duration of the remaining 4-bit data. The operation is as follows. 1) RDCSDATA[7:4] is transmitted as is. The transmission sequence of these data bits complies with the HSSPIn_PCC0 to PCC3:SDIR setting. 2) SDATA is kept in the Hi-Z state for the duration of 4-bit data.
1	111	Indicates the end of the list.

The command sequencer makes a transition to the data reading phase when any of the following conditions is satisfied.

- The end of the list is detected.
- Reading of the HSSPIn_RDCSDC7 register has finished.

In the data reading phase, HS-SPICNT captures serial data to assemble parallel data, which is then output as the data corresponding to AHB read access.

Warning:

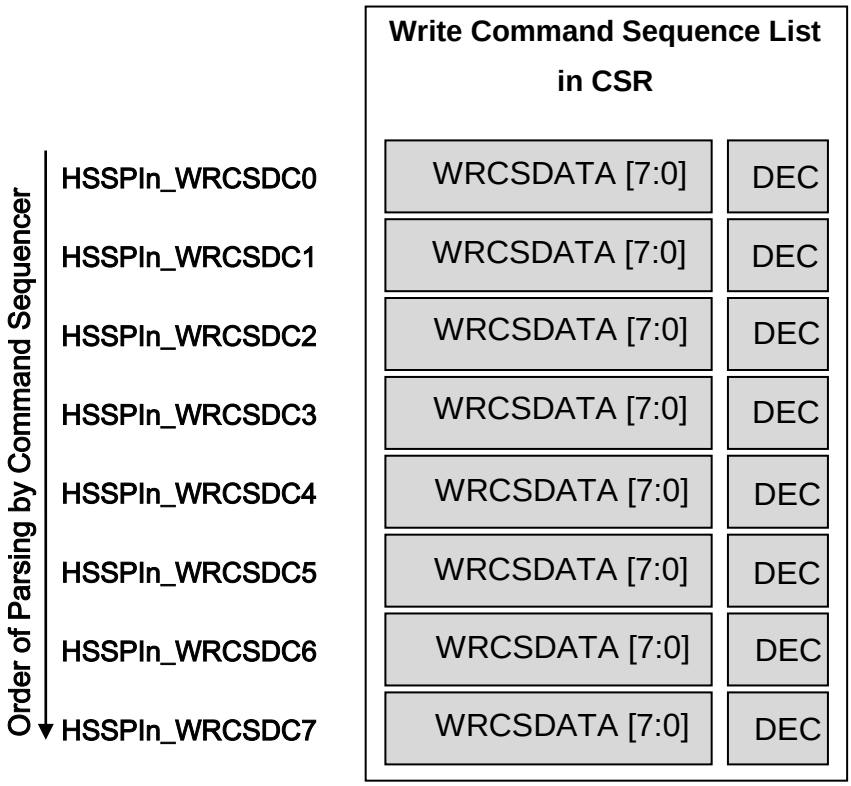
- After finishing reading the HSSPIn_RDCSDC7 register and making a transition to the data reading phase, the command sequencer uses the TRP[1:0] bits and the CONT bit in the HSSPIn_RDCSDC7 register.
- The "end of the list" setting is prohibited in the HSSPIn_RDCSDC0 register.

Issuing a Memory Write Command Sequence

The command sequencer issues a memory write command sequence only when HSSPIn_CSCFG:SRAM is set to "1".

In the memory write operation, command phases are executed in the sequence in which they were set in 8 registers (HSSPIn_WRCSDC0 to HSSPIn_WRCSDC7), to write data to memory. These registers are referenced one by one in order from HSSPIn_WRCSDC0 to HSSPIn_WRCSDC7. Figure 1-119 shows details of this operation.

Figure 1-119. Memory Write Command Sequence List



The DEC bit in each register indicates whether the data type must be decoded. Table 1-52 shows the relationship between the DEC bit and data type bit field (WRCSDATA[2:0]). WRCSDATA[7:0] is handled as data as is when the DEC bit is "0".

Table 1-52. Write Command Sequence List

DEC	WRCSDATA [2:0]	Description
0	Don't care	Transmits WRCSDATA[7:0] as is.
1	000	Transmits address[7:0] to access serial memory.
1	001	Transmits address[15:8] to access serial memory.
1	010	Transmits address[23:16] to access serial memory.
1	011	Transmits address[31:24] to access serial memory.
1	100	Sets SDATA to the Hi-Z state for the duration of 1-byte data.
1	101	Transmits 4 bits of 1-byte data and then sets SDATA to the Hi-Z state for the duration of the remaining 4-bit data. The operation is as follows. 1) WRCSDATA[7:4] is transmitted as is. The bit transmission sequence of these data bits complies with the HSSPIn_PCC0 to PCC3:SDIR setting. 2) SDATA is kept in the Hi-Z state for the duration of 4-bit data.
1	111	Indicates the end of the list.

The command sequencer makes a transition to the data writing phase when any of the following conditions is satisfied.

- The end of the list is detected.
- Reading of the HSSPIn_WRCSDC7 register has finished.

In the data writing phase, data written from AHB is transmitted to SDATA as serial data. Serial conversion of the data conforms to the already set SPI protocol.

Warning:

After finishing reading the HSSPIn_WRCSDC7 register and making a transition to the data writing phase, the command sequencer uses the TRP[1:0] bits and the CONT bit in the HSSPIn_WRCSDC7 register.

Return from a Continuous Instruction

With "1" set in the CONT bit in the HSSPIn_RDCSDC0 to RDCSDC7 registers, the set information in these HSSPIn_RDCSDC0 to RDCSDC7 registers is completely processed at the first processing time. However, from the second time, the HSSPIn_RDCSDC0 to RDCSDC7 registers with "1" set in the CONT bit are omitted from processing.

To return from a continuous instruction to a non-continuous instruction, execute the following processing.

For reading:

1. Cancel the toggle of the HSSPIn_RDCSDC0 to RDCSDC7 registers, and perform a read operation.
2. Then, set "0" in all the CONT bits in the HSSPIn_RDCSDC0 to RDCSDC7 registers.
3. Then, perform another read operation.

The above processing enables a return from a continuous instruction.

Warning:

- Among all the CONT bits in the HSSPIn_RDCSDC0 to RDCSDC7 registers, writing "1" to any CONT bit in the HSSPIn_RDCSDC1 to RDCSDC7 registers is prohibited.
- Set "0" for the CONT bit in the HSSPIn_WRCSDC0 to WRCSDC7 registers.

1.22.6 Register Set of the Hi-Speed SPI Controller

HS-SPICNT has various registers for its operation settings. These registers are mainly used to read and write HS-SPI states, HS-SPI settings, and the data to transfer to the serial interface.

This section describes all the registers in HS-SPICNT.

1.22.6.1 Register Map

The following table lists register addresses.

Table 1-53. Register Map

Address	+ 3	+ 2	+ 1	+ 0
$01FB_3000_H$	HSSPIn_MCTRL			
$01FB_3004_H$	HSSPIn_PCC0			
$01FB_3008_H$	HSSPIn_PCC1			
$01FB_300C_H$	HSSPIn_PCC2			
$01FB_3010_H$	HSSPIn_PCC3			
$01FB_3014_H$	HSSPIn_TXF			
$01FB_3018_H$	HSSPIn_TXE			
$01FB_301C_H$	HSSPIn_TXC			
$01FB_3020_H$	HSSPIn_RXF			
$01FB_3024_H$	HSSPIn_RXE			
$01FB_3028_H$	HSSPIn_RXC			
$01FB_302C_H$	HSSPIn_FAULTF			
$01FB_3030_H$	HSSPIn_FAULTC			
$01FB_3034_H$	read0			HSPIn_ DMCFG
$01FB_3038_H$	HSSPIn_ DMTRP	HSSPIn_ DMPSEL	HSSPIn_ DMSTOP	HSSPIn_ DMSTART
$01FB_303C_H$	HSSPIn_DMBCS		HSSPIn_DMBCC	
$01FB_3040_H$	HSSPIn_DMSTATUS			
$01FB_3044_H$	read0		HSSPIn_ RXBITCNT	HSSPIn_ TXBITCNT
$01FB_3048_H$	-			
$01FB_304C_H$	HSSPIn_FIFOCFG			
$01FB_3050_H$	HSSPIn_TXFIFO0			
$01FB_3054_H$	HSSPIn_TXFIFO1			
$01FB_3058_H$	HSSPIn_TXFIFO2			
$01FB_305C_H$	HSSPIn_TXFIFO3			
$01FB_3060_H$	HSSPIn_TXFIFO4			
$01FB_3064_H$	HSSPIn_TXFIFO5			
$01FB_3068_H$	HSSPIn_TXFIFO6			
$01FB_306C_H$	HSSPIn_TXFIFO7			
$01FB_3070_H$	HSSPIn_TXFIFO8			
$01FB_3074_H$	HSSPIn_TXFIFO9			

Address	+ 3	+ 2	+ 1	+ 0
01FB_3078$_H$	HSSPIn_TXFIFO10			
01FB_307C$_H$	HSSPIn_TXFIFO11			
01FB_3080$_H$	HSSPIn_TXFIFO12			
01FB_3084$_H$	HSSPIn_TXFIFO13			
01FB_3088$_H$	HSSPIn_TXFIFO14			
01FB_308C$_H$	HSSPIn_TXFIFO15			
01FB_3090$_H$	HSSPIn_RXFIFO0			
01FB_3094$_H$	HSSPIn_RXFIFO1			
01FB_3098$_H$	HSSPIn_RXFIFO2			
01FB_309C$_H$	HSSPIn_RXFIFO3			
01FB_30A0$_H$	HSSPIn_RXFIFO4			
01FB_30A4$_H$	HSSPIn_RXFIFO5			
01FB_30A8$_H$	HSSPIn_RXFIFO6			
01FB_30AC$_H$	HSSPIn_RXFIFO7			
01FB_30B0$_H$	HSSPIn_RXFIFO8			
01FB_30B4$_H$	HSSPIn_RXFIFO9			
01FB_30B8$_H$	HSSPIn_RXFIFO10			
01FB_30BC$_H$	HSSPIn_RXFIFO11			
01FB_30C0$_H$	HSSPIn_RXFIFO12			
01FB_30C4$_H$	HSSPIn_RXFIFO13			
01FB_30C8$_H$	HSSPIn_RXFIFO14			
01FB_30CC$_H$	HSSPIn_RXFIFO15			
01FB_30D0$_H$	HSSPIn_CSCFG			
01FB_30D4$_H$	HSSPIn_CSITIME			
01FB_30D8$_H$	HSSPIn_CSAEXT			
01FB_30DC$_H$	HSSPIn_RDCSDC1		HSSPIn_RDCSDC0	
01FB_30E0$_H$	HSSPIn_RDCSDC3		HSSPIn_RDCSDC2	
01FB_30E4$_H$	HSSPIn_RDCSDC5		HSSPIn_RDCSDC4	
01FB_30E8$_H$	HSSPIn_RDCSDC7		HSSPIn_RDCSDC6	
01FB_30EC$_H$	HSSPIn_WRCSDC1		HSSPIn_WRCSDC0	
01FB_30F0$_H$	HSSPIn_WRCSDC3		HSSPIn_WRCSDC2	
01FB_30F4$_H$	HSSPIn_WRCSDC5		HSSPIn_WRCSDC4	
01FB_30F8$_H$	HSSPIn_WRCSDC7		HSSPIn_WRCSDC6	
01FB_30FC$_H$	HSSPIn_MID			

General Control Register List

Abbreviated Register Name	Register Name	Reference
HSSPIn_MCTRL	HS-SPI control register	1.22.6.2

Serial Interface Control Register List

Abbreviated Register Name	Register Name	Reference
HSSPIn_PCC0	HS-SPI peripheral 0 communication setting register	1.22.6.3
HSSPIn_PCC1	HS-SPI peripheral 1 communication setting register	1.22.6.3
HSSPIn_PCC2	HS-SPI peripheral 2 communication setting register	1.22.6.3
HSSPIn_PCC3	HS-SPI peripheral 3 communication setting register	1.22.6.3

FIFO Control Register List (Common to TX/RX)

Abbreviated Register Name	Register Name	Reference
HSSPIn_FIFOCFG	HS-SPI FIFO setting register	1.22.6.24

TX-FIFO Control Register List

Abbreviated Register Name	Register Name	Reference
HSSPIn_TXF	HS-SPI transmission interrupt factor register	1.22.6.4
HSSPIn_TXE	HS-SPI transmission interrupt enable register	1.22.6.5
HSSPIn_TXC	HS-SPI transmission interrupt clear register	1.22.6.6
HSSPIn_TXBITCNT	HS-SPI remaining transmission bit count register	1.22.6.20
HSSPIn_TXFIFO0	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO1	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO2	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO3	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO4	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO5	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO6	HS-SPI TX-FIFO register	1.22.6.22

Abbreviated Register Name	Register Name	Reference
HSSPIn_TXFIFO7	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO8	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO9	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO10	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO11	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO12	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO13	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO14	HS-SPI TX-FIFO register	1.22.6.22
HSSPIn_TXFIFO15	HS-SPI TX-FIFO register	1.22.6.22

RX-FIFO Control Register List

Abbreviated Register Name	Register Name	Reference
HSSPIn_RXF	HS-SPI reception interrupt factor register	1.22.6.7
HSSPIn_RXE	HS-SPI reception interrupt enable register	1.22.6.8
HSSPIn_RXC	HS-SPI reception interrupt clear register	1.22.6.9
HSSPIn_RXBITCNT	HS-SPI remaining reception bit count register	1.22.6.21
HSSPIn_RXFIFO0	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO1	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO2	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO3	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO4	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO5	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO6	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO7	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO8	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO9	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO10	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO11	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO12	HS-SPI RX-FIFO register	1.22.6.23

Abbreviated Register Name	Register Name	Reference
HSSPIn_RXFIFO13	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO14	HS-SPI RX-FIFO register	1.22.6.23
HSSPIn_RXFIFO15	HS-SPI RX-FIFO register	1.22.6.23

Fault Register List

Abbreviated Register Name	Register Name	Reference
HSSPIn_FAULTF	HS-SPI fault interrupt factor register	1.22.6.10
HSSPIn_FAULTC	HS-SPI fault interrupt clear register	1.22.6.11

Direct Mode Control Register List

Abbreviated Register Name	Register Name	Reference
HSSPIn_DMCFG	HS-SPI direct mode setting register	1.22.6.12
HSSPIn_DMSTART	HS-SPI direct mode transfer start control register	1.22.6.13
HSSPIn_DMSTOP	HS-SPI direct mode transfer stop control register	1.22.6.14
HSSPIn_DMPSEL	HS-SPI direct mode slave select register	1.22.6.15
HSSPIn_DMTRP	HS-SPI direct mode transfer protocol setting register	1.22.6.16
HSSPIn_DMBCC	HS-SPI direct mode transfer byte count setting register	1.22.6.17
HSSPIn_DMBCS	HS-SPI direct mode transfer remaining count register	1.22.6.18
HSSPIn_DMSTATUS	HS-SPI direct mode status register	1.22.6.19

Command Sequencer Control Register List (Common to Writing/Reading)

Abbreviated Register Name	Register Name	Reference
HSSPIn_CSCFG	HS-SPI command sequencer setting register	1.22.6.25
HSSPIn_CSITIME	HS-SPI command sequencer idle timer setting register	1.22.6.26
HSSPIn_CSAEXT	HS-SPI command sequencer address extension register	1.22.6.27

Command Sequence Setting Register List (Write Command)

Abbreviated Register Name	Register Name	Reference
HSSPIn_WRCSDC0	HS-SPI write command sequence data/control register	1.22.6.29
HSSPIn_WRCSDC1	HS-SPI write command sequence data/control register	1.22.6.29
HSSPIn_WRCSDC2	HS-SPI write command sequence data/control register	1.22.6.29
HSSPIn_WRCSDC3	HS-SPI write command sequence data/control register	1.22.6.29
HSSPIn_WRCSDC4	HS-SPI write command sequence data/control register	1.22.6.29
HSSPIn_WRCSDC5	HS-SPI write command sequence data/control register	1.22.6.29
HSSPIn_WRCSDC6	HS-SPI write command sequence data/control register	1.22.6.29
HSSPIn_WRCSDC7	HS-SPI write command sequence data/control register	1.22.6.29

Command Sequence Setting Register List (Read Command)

Abbreviated Register Name	Register Name	Reference
HSSPIn_RDCSDC0	HS-SPI read command sequence data/control register	1.22.6.28
HSSPIn_RDCSDC1	HS-SPI read command sequence data/control register	1.22.6.28
HSSPIn_RDCSDC2	HS-SPI read command sequence data/control register	1.22.6.28
HSSPIn_RDCSDC3	HS-SPI read command sequence data/control register	1.22.6.28
HSSPIn_RDCSDC4	HS-SPI read command sequence data/control register	1.22.6.28
HSSPIn_RDCSDC5	HS-SPI read command sequence data/control register	1.22.6.28
HSSPIn_RDCSDC6	HS-SPI read command sequence data/control register	1.22.6.28
HSSPIn_RDCSDC7	HS-SPI read command sequence data/control register	1.22.6.28

Module Identification Register List

Abbreviated Register Name	Register Name	Reference
HSSPIn_MID	HS-SPI module identification register	1.22.6.30

Explanatory Notes

The following explanatory notes apply to the detailed descriptions on subsequent pages.

- R means read-only in privileged mode.
- R0 means that the read value is always "0" in privileged mode.
- R1 means that the read value is always "1" in privileged mode.
- W means write-only in privileged mode.
- W1 means that "1" is always the only value that can be written in privileged mode.
- X means that the read value or written value is undefined.

1.22.6.2 HS-SPI Control Register (HSSPIn_MCTRL)

The purpose of this register is to control HS-SPICNT. The register contains the module enable bit, the command sequencer enable bit, and other bits that significantly change the operating mode.

Software can use this register to switch the operation of this controller between the enabled and disabled states.

Register Configuration

- HS-SPI control register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI control register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI control register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI control register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	SYNCON	MES	Reserved	Reserved	CSEN	MEN
Attribute	R0	R0	R/W	R	R0	R0	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:6] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit5] SYNCON: "Synchronizer ON": Synchronizer circuit operation bit

Bit Value	Function
0	Operate the synchronizer circuit. [Initial value]
1	Do not operate the synchronizer circuit. The latency between AHB and SPI is reduced in command sequencer mode.

Warning:

- This bit is enabled only in command sequencer mode. In direct mode, "0" is read.
- To change this bit, do so when the MES bit is "0".

[bit4] MES: "Module Enable Status": Module enable status bit

Bit Value	Function
0	This indicates the state where module operation is fully stopped. [Initial value]
1	Module operation is enabled.

[bit3:2] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit1] CSEN: "Command Sequencer Enable": Command sequencer mode enable bit

Bit Value	Function
0	Enable direct mode. [Initial value]
1	Enable command sequencer mode.

Warning:

- To switch from command sequencer mode to direct mode, the SPI transaction must have ended (SPI slave is deselected).
  1. Clear the CSEN bit to switch to direct mode.
  2. Detect the deselected slave of the slave select by checking whether HSSPIn_TXF:TSSRS is "1" or HSSPIn_RXF:RSSRS is "1".
  3. Disable HS-SPICNT operation (MEN is "0").
  4. Wait for HS-SPICNT operation to be disabled (MES is "0").
  5. Configure various registers for direct mode.
  6. Enable HS-SPICNT operation (MEN is "1").
  7. Wait for HS-SPICNT operation to be enabled (MES is "1").
  8. Operation in direct mode becomes possible. After this, operation of the HSSPIn_DMSTART register is enabled.

- Likewise, to switch from direct mode to command sequencer mode, the SPI transaction must have ended (SPI slave is deselected).
  1. Detect the deselected slave of the slave select by checking whether HSSPIn_TXF:TSSRS is "1" or HSSPIn_RXF:RSSRS is "1".
  2. Disable HS-SPICNT operation (MEN is "0").
  3. Wait for HS-SPICNT operation to be disabled (MES is "0").
  4. Set the CSEN bit to switch to command sequencer mode.
  5. Configure various registers for command sequencer mode.
  6. Enable HS-SPICNT operation (MEN is "1").
  7. Wait for HS-SPICNT operation to be enabled (MES is "1").
  8. Memory access in command sequencer mode becomes possible.

[bit0] MEN: "Module Enable": Module enable bit

Bit Value	Function
0	Set HS-SPICNT to the disabled state. All bidirectional pins enter the Hi-Z state. [Initial value]
1	Set HS-SPICNT to the enabled state. Software sets "1" in this bit after configuring various HS-SPICNT registers. As a result, HS-SPICNT can operate.

Warning:

To change this bit from "1" to "0", do so after the deassertion of the slave select in direct mode is detected (HSSPIn_TXF:TSSRS is "1" or HSSPIn_RXF:RSSRS is "1"). If HS-SPICNT is in command sequencer mode, perform the same operation after clearing the CSEN bit and switching to direct mode.

1.22.6.3 HS-SPI Peripheral Communication Setting Registers (HSSPIn_PCC0 to PCC3)

The purpose of these registers is various serial communication-related settings for slave selects 0 to 3. Software uses these registers to make settings that match the specifications of the serial devices corresponding to the slave selects (0 to 3). In command sequencer mode, use these registers with the same settings.

These registers 0 to 3 have the same bit field configuration, so this section provides a detailed description of just the HSSPIn_PCC0 register.

Register Configuration

■ HS-SPI peripheral communication setting register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI peripheral communication setting register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	RDDSEL[1]	RDDSEL[0]	WRDSEL[3]	WRDSEL[2]	WRDSEL[1]	WRDSEL[0]	SAFE SYNC
Attribute	R0	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	1	1	1	1	1	1	1

■ HS-SPI peripheral communication setting register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	CDRS [6]	CDRS [5]	CDRS [4]	CDRS [3]	CDRS [2]	CDRS [1]	CDRS [0]	SENDIAN
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI peripheral communication setting register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	SDIR	SS2CD [1]	SS2CD [0]	SSPOL	RTM	ACES	CPOL	CPHA
Attribute	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:23] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit22:21] RDDSEL[1:0]: "Read Deselect Time": Read deselect time setting bits

Bit Value	Function
0x0 to 0x3	Use as the deselect time. 0: 1-serial clock time 1: 2-serial clock time 2: 3-serial clock time 3: 4-serial clock time [Initial value]

These bits are used for the deselect time of the slave select at the read time.

Warning:

To change these bits, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").

[bit20:17] WRDSEL[3:0]: "Write/different command Deselect Time": Write or different command deselect time setting bits

Bit Value	Function
0x0 to 0xF	Use as the deselect time. 0: 1-serial clock time 1: 2-serial clock time 2: 3-serial clock time 3: 4-serial clock time 4: 5-serial clock time 5: 6-serial clock time 6: 7-serial clock time 7: 8-serial clock time 8: 9-serial clock time 9: 10-serial clock time 10: 11-serial clock time 11: 12-serial clock time 12: 13-serial clock time 13: 14-serial clock time 14: 15-serial clock time 15: 16-serial clock time [Initial value]

These bits are used for the deselect time of the slave select at the write time or execution time of a different command.

Warning:

To change these bits, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").

[bit16] SAFESYNC: "Safe Synchronization": Safe synchronization bit

Bit Value	Function
0	HS-SPICNT operates normally. Do not add the delay that is set in advance to ensure safe resynchronization during the serial communication period.
1	Add the delay that is set in advance to ensure safe resynchronization of serial communication. [Initial value]

If the serial clock frequency is equal to or greater than half of the AHB clock frequency, set this bit to "1" to ensure serial communication of HS-SPICNT.

Warning:

- To change this bit, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").
- If the SYNCON setting is "0" in command sequencer mode, set "1" in this bit regardless of the frequency ratio or the CDRS value.

[bit15:9] CDRS[6:0]: "Clock Division Ratio Select": Clock division ratio setting bits

Bit Value	Function
0x00	Reserved [Initial value]
0x01 to 0x7F	The bit values have the following meanings, shown with decimal numbers. 1: Divide by 2. 2: Divide by 4. 3: Divide by 6. ... 127: Divide by 254. The clock frequency is divided by twice the set value of the CDRS bits. This relationship is represented by the following expression, where Fi represents the clock frequency before division, and Fo represents the clock frequency after division: Fo = Fi / (2 x CDRS)

Serial communication with the CDRS bits set to 0x00 is prohibited. Set 0x01 or greater.

Warning:

- To change these bits, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").
- The HSSPIn_PCC0 to PCC3:CDRS setting of 0x00 and serial communication are prohibited. Also, for the HSSPIn_PCC0 to PCC3:CDRS setting, specify a value that is equal to or less than the maximum frequency of the serial clock defined in each clocking mode.

[bit8] SENDIAN: "SPI Endian": Endian setting bit

Bit Value	Function
0	Use Big Endian. [Initial value] Byte data on the serial interface is arranged in the order from higher to lower position in word data.
1	Use Little Endian. Byte data on the serial interface is arranged in the order from lower to higher position in word data.

This bit has the function of controlling the arrangement of byte data in 1-word data. Word data corresponds to the parallel interface side of the shift register, and byte data corresponds to the serial interface side.

Warning:

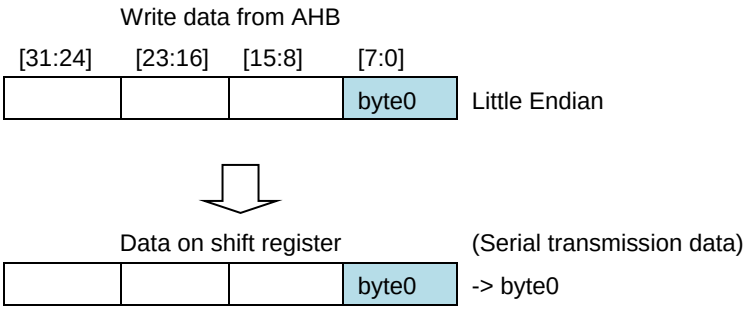
To change this bit, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").

The following figures show the arrangement of byte data between the AHB interface and serial interface.

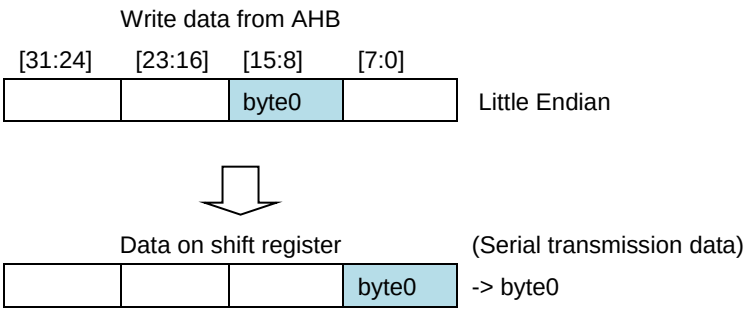
- Transmission operation (write operation)

  8-bit access (FWIDTH = "00" in a case of direct mode)

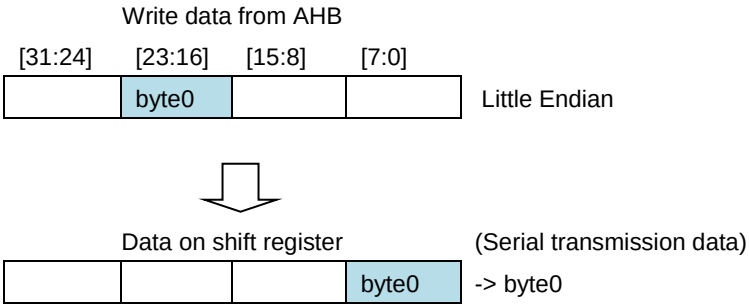
Address[1:0] = "LL"



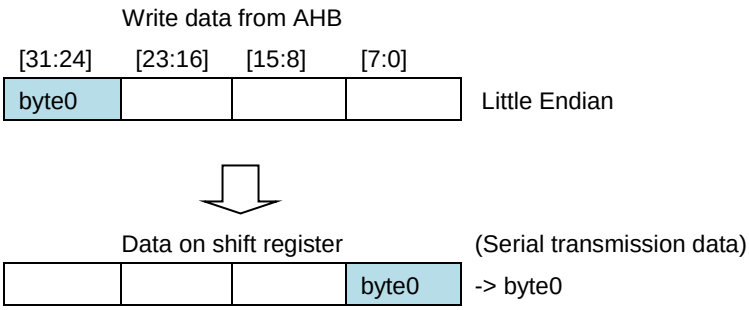
Address[1:0]="LH"



Address[1:0]="HL"

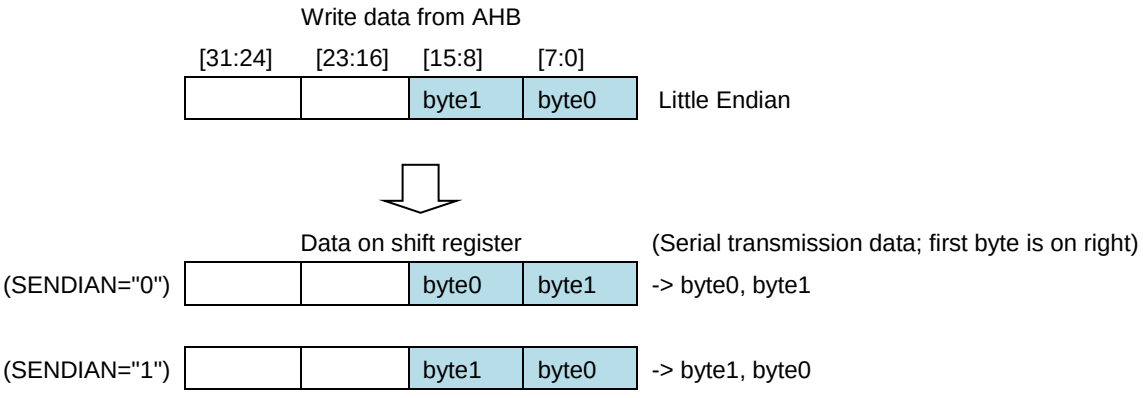


Address[1:0]="HH"

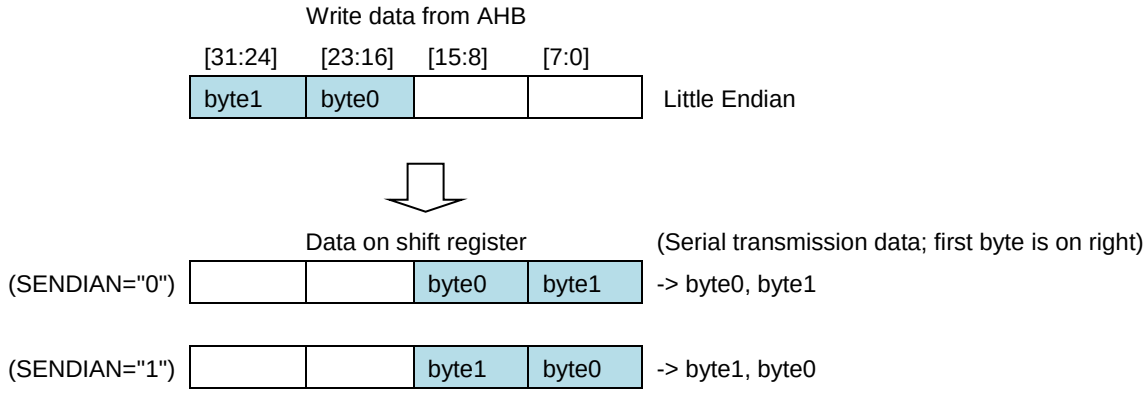


16-bit access (FWIDTH="01" in a case of direct mode)

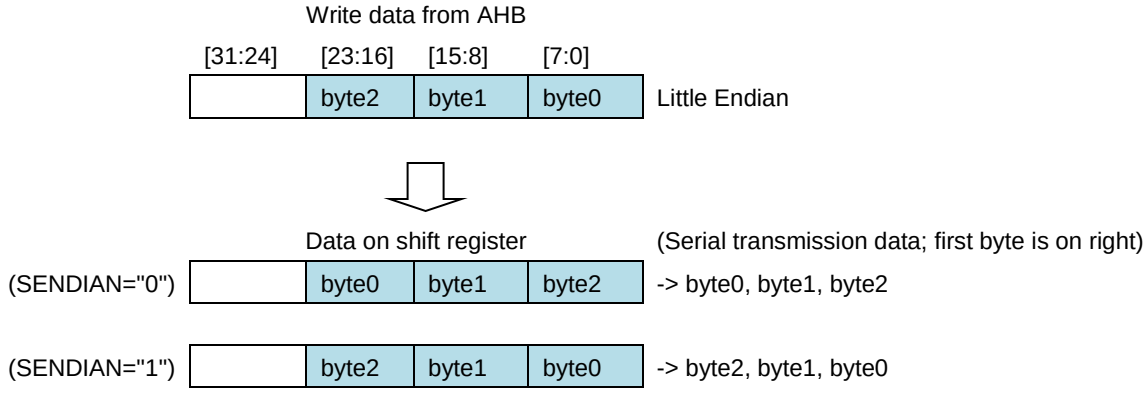
Address[1:0]="LL"



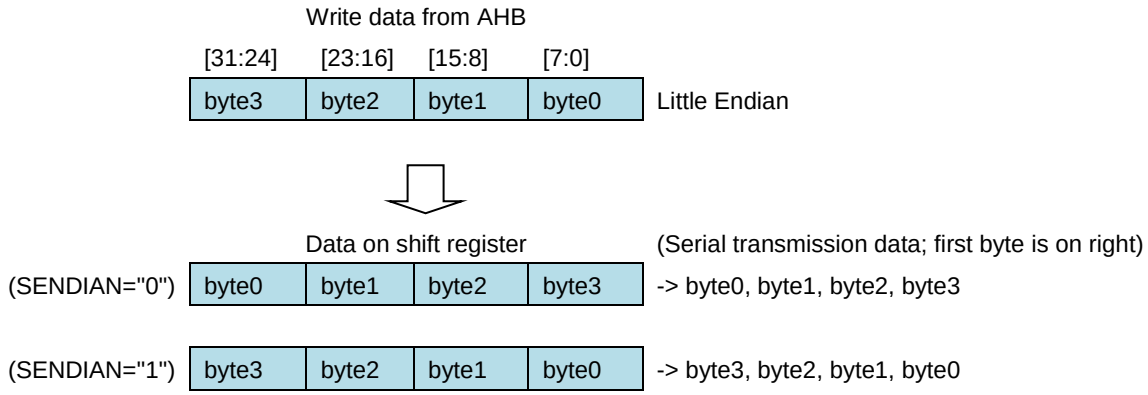
Address[1:0]="HL"



32-bit access (FWIDTH="10" in a case of direct mode)



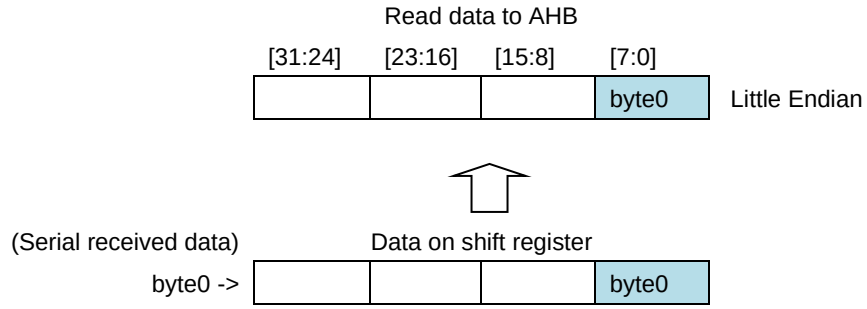
32-bit access (FWIDTH="11" in a case of direct mode)



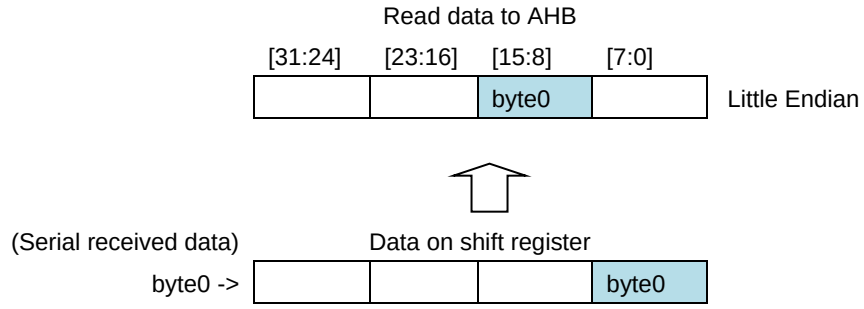
- Reception operation (read operation)

8-bit access (FWIDTH="00" in a case of direct mode)

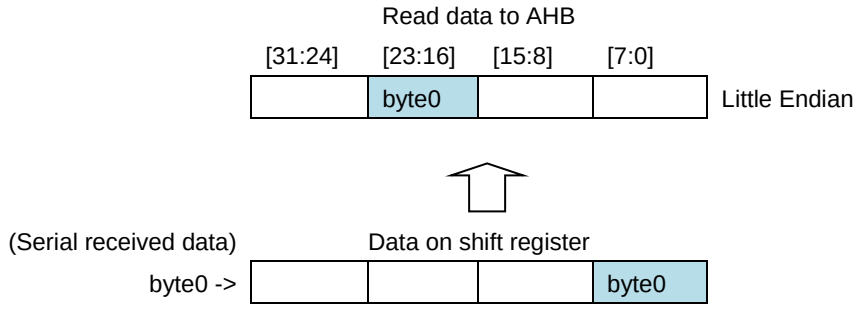
Address[1:0]="LL"



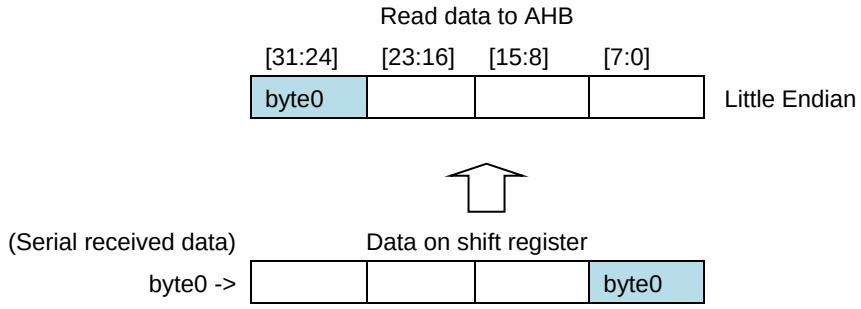
Address[1:0]="LH"



Address[1:0]="HL"

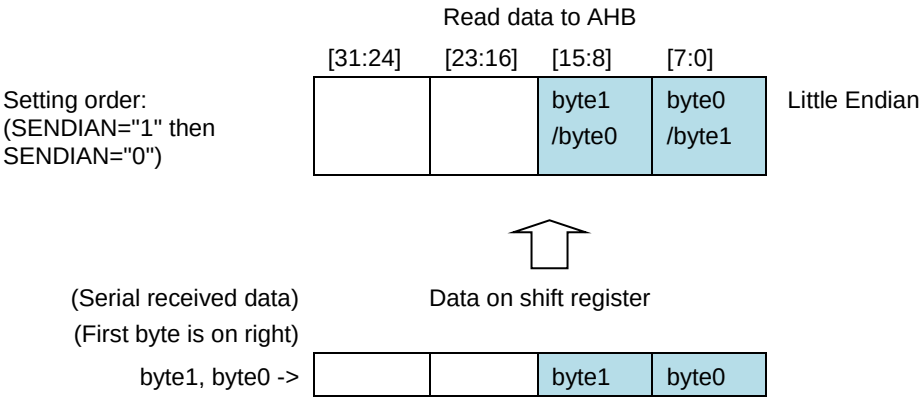


Address[1:0]="HH"

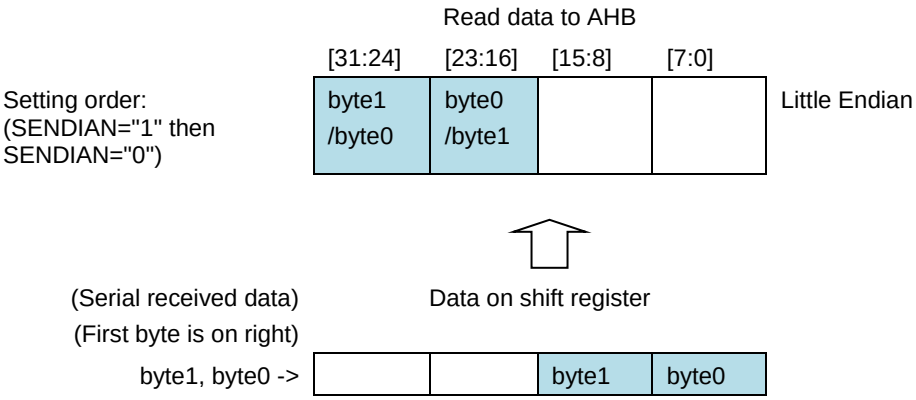


16-bit access (FWIDTH="01" in a case of direct mode)

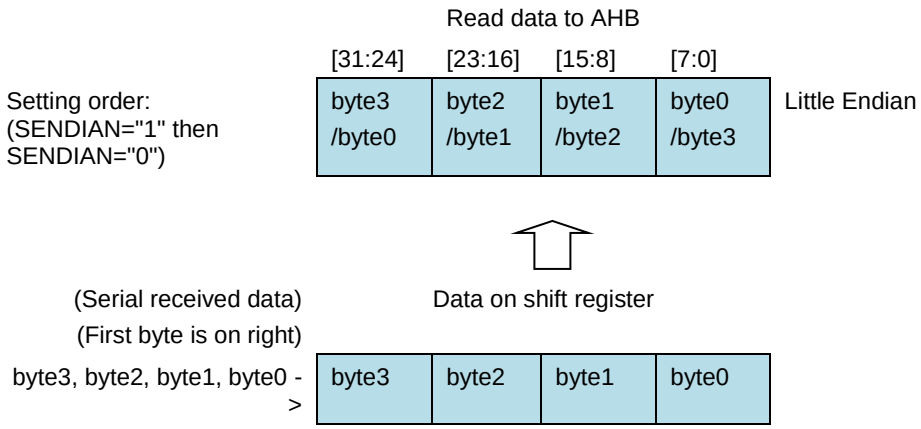
Address[1:0]="LL"



Address[1:0]="HL"



32-bit access (FWIDTH="11" in a case of direct mode)



Note: The descriptions of the SDIR bit functions show data arranged in units of bits.

[bit7] SDIR: "Shift Direction": Shift direction setting bit

Bit Value	Function
0	Set the bit order in byte data from the highest bit to the lowest bit. [Initial value]
1	Set the bit order in byte data from the lowest bit to the highest bit.

This bit determines the order (i.e., shift direction) of bit transfer in a field. Its setting does not change the positions of the highest bit and lowest bit in the other registers. In the read/write operation of the data register, the lowest bit is always at the location of bit0.

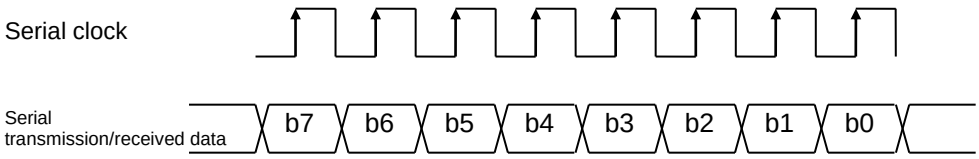
The SDIR bit setting does not affect the byte order in the shift register. The SDIR bit manipulates only the bit order in each byte. Manipulation of the byte order requires manipulation of the SENDIAN bit.

Warning:

To change this bit, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").

The following figure shows an arrangement where bit data flowing through the serial interface corresponds to byte data according to the bit width.

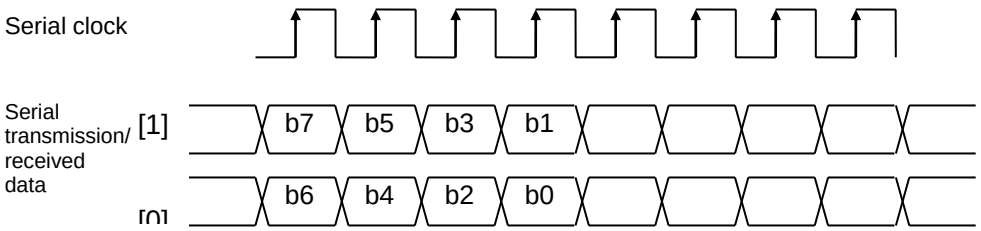
1. Single-bit mode



The bit order in 1-byte data is as follows according to the SDIR setting.

SDIR	Reception Bit Count	Bit Storage Location in Byte							
		7	6	5	4	3	2	1	0
0	8	b7	b6	b5	b4	b3	b2	b1	b0
	7		b7	b6	b5	b4	b3	b2	b1
	6			b7	b6	b5	b4	b3	b2
	5				b7	b6	b5	b4	b3
	4					b7	b6	b5	b4
	3						b7	b6	b5
	2							b7	b6
	1								b7
1	8	b0	b1	b2	b3	b4	b5	b6	b7
	7	b1	b2	b3	b4	b5	b6	b7	
	6	b2	b3	b4	b5	b6	b7		
	5	b3	b4	b5	b6	b7			
	4	b4	b5	b6	b7				
	3	b5	b6	b7					
	2	b6	b7						
	1	b7							

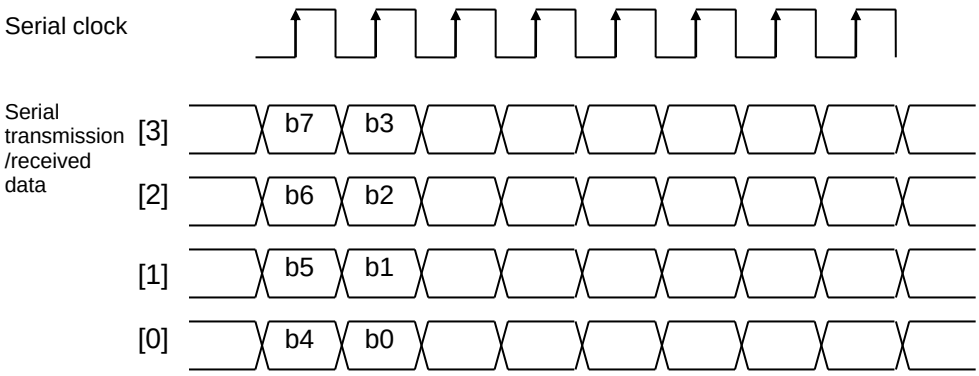
2. Dual-bit mode



The bit order in 1-byte data is as follows according to the SDIR setting.

SDIR Value	Reception Bit Count	Bit Storage Location in Byte							
		7	6	5	4	3	2	1	0
0	8	b7	b6	b5	b4	b3	b2	b1	b0
	6			b7	b6	b5	b4	b3	b2
	4					b7	b6	b5	b4
	2							b7	b6
1	8	b1	b0	b3	b2	b5	b4	b7	b6
	6	b3	b2	b5	b4	b7	b6		
	4	b5	b4	b7	b6				
	2	b7	b6						

3. Quad-bit mode



The bit order in 1-byte data is as follows according to the SDIR setting.

SDIR Value	Reception Bit Count	Bit Storage Location in Byte							
		7	6	5	4	3	2	1	0
0	8	b7	b6	b5	b4	b3	b2	b1	b0
	4					b7	b6	b5	b4
1	8	b3	b2	b1	b0	b7	b6	b5	b4
	4	b7	b6	b5	b4				

Note: The descriptions of the SENDIAN bit functions show data arranged in units of bytes.

[bit6:5] SS2CD[1:0]: "Slave-Select to Clock Delay"

Slave-select-to-clock-start delay time setting bit

By delaying the change point of the serial clock, HS-SPICNT delays the time of data transmission from the change point of slave selection in units of serial clock cycles.

If HSSPIn_PCC0 to PCC3:CPHA is "0", the delay from the slave select assertion point to the first serial clock change is "SS2CD + 1.5" serial clock cycles.

If HSSPIn_PCC0 to PCC3:CPHA is "1", the delay from the slave select assertion point to the first serial clock change is "SS2CD + 1.0" serial clock cycles.

Figure 1-113 shows the use of SS2CD.

If the slave select is asserted, send data must be prepared on the slave side within the time period specified by the SS2CD bits.

Warning:

To change these bits, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").

[bit4] SSPOL: "Slave Select Polarity": Slave select polarity setting bit

Bit Value	Function
0	Use negative logic ("L" means active). [Initial value]
1	Use positive logic ("H" means active).

This bit is used to determine the slave select polarity.

Figure 1-113 shows operation when SSPOL is "0". "Slave Select" in the figure shows reverse polarity when SSPOL is "1".

Warning:

To change this bit, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").

[bit3] RTM: "Use retimed clock for Capturing the data": Timing compensation setting bit

Bit Value	Function
0	Capture received serial data without using a timing-compensated clock. [Initial value]
1	Capture received serial data by using a timing-compensated clock.

This bit specifies whether to use a timing-compensated serial clock. If there is not a sufficient margin for the setup hold time of serial received data, set this bit to "1".

In coordination with one another, CPHA, CPOL, ACES, and RTM determine the clock operating mode of the HS-SPI serial interface.

Warning:

To change this bit, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").

[bit2] ACES: "Active Clock Edges are Same": Serial data transmission/reception timing setting bit

Bit Value	Function
0	Use different clock edges for data transmission and reception (rising edge for one, and falling edge for the other). [Initial value]
1	Reserved

Figure 1-113 shows the use of ACES.

In coordination with one another, CPHA, CPOL, ACES, and RTM determine the clock operating mode of the HS-SPI serial interface.

Warning:

The setting of "1" is prohibited for this bit.

[bit1] CPOL: "Clock Polarity": Serial clock polarity setting bit

Bit Value	Function
0	Specify the "L" level. [Initial value]
1	Specify the "H" level.

This bit specifies a serial clock level in a period during which there is no serial transfer.

Figure 1-113 shows the use of CPOL.

In coordination with one another, CPHA, CPOL, ACES, and RTM determine the clock operating mode of the HS-SPI serial interface.

Warning:

To change this bit, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").

[bit0] CPHA: "Clock Phase": Clock phase setting bit

Bit Value	Function
0	Capture at an odd-numbered edge. [Initial value]
1	Capture at an even-numbered edge.

This bit selects the serial clock edge at which to capture received data. Both the rising edges and falling edges can be set as the clock edges used. If both edges are numbered in sequence from the beginning (with the first edge counted as edge 1), odd-numbered and even-numbered edges are numbered as discussed below.

Figure 1-113 shows the use of CPHA.

In coordination with one another, CPHA, CPOL, ACES, and RTM determine the clock operating mode of the HS-SPI serial interface.

Warning:

To change this bit, do so when HS-SPICNT operation is disabled (HSSPIn_MCTRL:MES is "0").

1.22.6.4 HS-SPI Transmission Interrupt Factor Register (HSSPIn_TXF)

This register indicates the states of transmission interrupt flags. The flag bits contained in this register operate only in direct mode.

After enabling the interrupt enable bit for each of these interrupt flags, software can either wait for an interrupt or poll the flag and wait for it to change to learn the interrupt flag state.

Register Configuration

- HS-SPI transmission interrupt factor register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI transmission interrupt factor register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI transmission interrupt factor register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI transmission interrupt factor register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	TSSRS	TFMTS	TFLETS	TFUS	TFOS	TFES	TFFS
Attribute	R0	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:7] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit6] TSSRS: "Slave Select Released Interrupt": Slave select released detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that the slave select has been released by the SPI master. The shift register state has no impact on the bit.

This flag generates a transmission interrupt when HSSPIn_TXE:TSSRE is set to "1".

[bit5] TFMTS: "TX-FIFO Fill Level is More than Threshold Interrupt"

TX-FIFO-exceeded-threshold detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that the TX-FIFO level exceeds the threshold. "H" is set in each GDSSCGCLK cycle when conditions are satisfied. The shift register state has no impact on the bit.

The phrase "exceeds the threshold" specifically means that "HSSPIn_DMSTATUS:TXFLEVEL" exceeds "HSSPIn_FIFOCFG:TXFTH." This flag generates a transmission interrupt when HSSPIn_TXE:TFMTE is set to "1".

[bit4] TFLETS: "TX-FIFO Fill Level is Less than or Equal to Threshold Interrupt"

TX-FIFO-less-than-or-equal-to-threshold detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that the TX-FIFO level (1 is subtracted when data is transferred to the shift register) is less than or equal to the threshold. "1" is set in each AHB clock cycle when conditions are satisfied. The shift register state has no impact on the bit.

The phrase "is less than or equal to the threshold" specifically means that "HSSPIn_DMSTATUS:TXFLEVEL" is less than or equal to "HSSPIn_FIFOCFG:TXFTH." This flag generates a transmission interrupt when HSSPIn_TXE:TFLETE is set to "1".

[bit3] TFUS: "TX-FIFO Underrun Interrupt"

TX-FIFO underrun detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that a TX-FIFO underrun has occurred.

This underrun occurs when the TX-FIFO is empty in HS-SPICNT and data transfer from the TX-FIFO to the shift register is attempted.

This flag generates a transmission interrupt when HSSPIn_TXE:TFUE is set to "1".

Warning:

If this flag is set to "1", data is not transmitted normally. Also, other flags do not operate normally either.

[bit2] TFOS: "TX-FIFO Overrun Interrupt"

TX-FIFO overrun detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that a TX-FIFO overrun has occurred. The shift register state has no impact on the bit.

This overrun occurs when the TX-FIFO is full in HS-SPICNT and software writes data to it.

This flag generates a transmission interrupt when HSSPIn_TXE:TFOE is set to "1".

Warning:

If this flag is set to "1", data is not transmitted normally. Also, other flags do not operate normally either.

[bit1] TFES: "TX-FIFO and Shift Register are Empty Interrupt"

TX-FIFO and shift register empty detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that the TX-FIFO and the TX shift register are empty. "1" is set in each GSSCGCLK cycle when conditions are satisfied. The shift register state has an impact only on this bit.

This flag generates a transmission interrupt when HSSPIn_TXE:TFEE is set to "1".

[bit0] TFFS: "TX-FIFO Full Interrupt": TX-FIFO full detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that the TX-FIFO is full. "1" is set in each GSSCGCLK cycle when conditions are satisfied. The shift register state has no impact on the bit. This flag generates a transmission interrupt when HSSPIn_TXE:TFFE is set to "1".

The conditions for setting each TX-FIFO flag are as follows:

- TFMTS: The bit is set depending on "write pointer = read pointer + 16" or "write pointer > read pointer + TFTH value."
- TFMTS: The bit is set depending on "write pointer <= read pointer + TFTH value."
- TFLETS: The bit is set depending on "write pointer <= read pointer + TFTH value."
- TFUS: The bit is set depending on "write pointer = read pointer" and "read pulse."
- TFOS: The bit is set depending on "write pointer = read pointer + 16" and "write pulse."
- TFES: The bit is set depending on "write pointer = read pointer" and "bit count (subtraction) value of transmission shift register is 0."
- TFFS: The bit is set depending on "write pointer = read pointer + 16" (TX-FIFO is full).

The above "write pulse" and "read pulse" mean signals that detected CPU access to the TX-FIFO. The "write pointer" is incremented by 1 each time that 1-word data is written to the TX-FIFO. The "read pointer" is incremented by 1 each time that 1-word data is read from the TX-FIFO. These pointers are cleared when "1" is written to TXFLSH.

1.22.6.5 HS-SPI Transmission Interrupt Enable Register (HSSPIn_TXE)

This register sets whether to notify MCNT of transmission interrupts according to the HSSPIn_TXF register state. Software needs to enable the respective bits in this register to use these interrupts.

Register Configuration

- HS-SPI transmission interrupt enable register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI transmission interrupt enable register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI transmission interrupt enable register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI transmission interrupt enable register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	TSSRE	TFMTE	TFLETE	TFUE	TFOE	TFEE	TFFE
Attribute	R0	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:7] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit6] TSSRE: "Slave Select Released Interrupt Enable"

Slave select released detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_TXF:TSSRS generates a transmission interrupt.

[bit5] TFMTE: "TX-FIFO Fill Level is More than Threshold Interrupt Enable"

TX-FIFO-exceeded-threshold detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_TXF:TFMTS generates a transmission interrupt.

[bit4] TFLETE: "TX-FIFO Fill Level is Less than or Equal to Threshold Interrupt Enable"

TX-FIFO-less-than-or-equal-to-threshold detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_TXF:TFLETS generates a transmission interrupt.

[bit3] TFUE: "TX-FIFO Underrun Interrupt Enable"

TX-FIFO underrun detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_TXF:TFUS generates a transmission interrupt.

[bit2] TFOE: "TX-FIFO Overrun Interrupt Enable"

TX-FIFO overrun detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_TXF:TFOS generates a transmission interrupt.

[bit1] TFEE: "TX-FIFO and Shift Register are Empty Interrupt Enable"

TX-FIFO and shift register empty detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_TXF:TFES generates a transmission interrupt.

[bit0] TFFE: "TX-FIFO Full Interrupt Enable": TX-FIFO full detection interrupt enable bit

Bit Value	Function
0	Read value always "0" [Initial value]
1	Write-only

This bit sets whether HSSPIn_TXF:TFFS generates a transmission interrupt.

1.22.6.6 HS-SPI Transmission Interrupt Clear Register (HSSPIn_TXC)

This register clears the HSSPIn_TXF register state to "0".

Software can write "1" to a bit in this register to clear the corresponding interrupt factor flag in the HSSPIn_TXF register.

Register Configuration

- HS-SPI transmission interrupt clear register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI transmission interrupt clear register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI transmission interrupt clear register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI transmission interrupt clear register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	TSSRC	TFMTC	TFLETC	TFUC	TFOC	TFEC	TFFC
Attribute	R0	W1/R0	W1/R0	W1/R0	W1/R0	W1/R0	W1/R0	W1/R0
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:7] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit6] TSSRC: "Slave Select Released Interrupt Clear": Slave select released detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only is enabled.

This bit is a control bit that clears HSSPIn_TXF:TSSRS.

[bit5] TFMTC:"TX-FIFO Fill Level is More than Threshold Interrupt Clear"

TX-FIFO-exceeded-threshold detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value]
1	Write-only

This bit is a control bit that clears HSSPIn_TXF:TFMTS.

[bit4] TFLETC: "TX-FIFO Fill Level is Less than or Equal to Threshold Interrupt"

TX-FIFO-less-than-or-equal-to-threshold detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value]
1	Write-only

This bit is a control bit that clears HSSPIn_TXF:TFLETS.

[bit3] TFUC: "TX-FIFO Underrun Interrupt Clear"

TX-FIFO underrun detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value]
1	Write-only

This bit is a control bit that clears HSSPIn_TXF:TFUS.

[bit2] TFOC: "TX-FIFO Overrun Interrupt Clear": TX-FIFO overrun detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value]
1	Write-only

This bit is a control bit that clears HSSPIn_TXF:TFOS.

[bit1] TFEC: "TX-FIFO and Shift Register are Empty Interrupt Clear"

TX-FIFO and shift register empty detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value]
1	Write-only

This bit is a control bit that clears HSSPIn_TXF:TFES.

[bit0] TFFC: "TX-FIFO Full Interrupt Clear": TX-FIFO full detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value]
1	Write-only

This bit is a control bit that clears HSSPIn_TXF:TFFS.

1.22.6.7 HS-SPI Reception Interrupt Factor Register (HSSPIn_RXF)

This register indicates the states of reception interrupt flags. The flag bits contained in this register operate only in direct mode.

After enabling the interrupt enable bit for 1 of these interrupt flags, software can either wait for an interrupt or poll the flag and wait for it to change to learn the interrupt flag state.

Register Configuration

- HS-SPI reception interrupt factor register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI reception interrupt factor register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI reception interrupt factor register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI reception interrupt factor register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	RSSRS	RFMTS	RFLETS	RFUS	RFOS	RFES	RFFS
Attribute	R0	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:7] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit6] RSSRS: "Slave Select Released Interrupt"

Slave select released detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that the slave select has been released by the SPI master. The shift register state has no impact on the bit.

This flag generates a reception interrupt when HSSPIn_RXE:RSSRE is set to "1".

[bit5] RFMTS: "RX-FIFO Fill Level is More than Threshold Interrupt"

RX-FIFO-exceeded-threshold detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that the RX-FIFO level (1 is added when data is transferred from the shift register) exceeds the threshold. "1" is set in each GSSCGCLK cycle when conditions are satisfied. The shift register state has no impact on the bit.

The phrase "exceeds the threshold" specifically means that "HSSPIn_DMSTATUS:RXFLEVEL" exceeds "HSSPIn_FIFOCFG:RXFTH." This flag generates a reception interrupt when HSSPIn_RXE:RFMTE is set to "1".

[bit4] RFLETS: "RX-FIFO Fill Level is Less than or Equal to Threshold Interrupt"

RX-FIFO-less-than-or-equal-to-threshold detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that the RX-FIFO level is less than or equal to the threshold. "1" is set in each GSSCGCLK cycle when conditions are satisfied. The shift register state has no impact on the bit.

The phrase "is less than or equal to the threshold" specifically means that "HSSPIn_DMSTATUS:RXFLEVEL" is less than or equal to "HSSPIn_FIFOCFG:RXFTH." This flag generates a reception interrupt when HSSPIn_RXE:RFLETE is set to "1".

[bit3] RFUS: "RX-FIFO Underrun Interrupt"

RX-FIFO underrun detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that an RX-FIFO underrun has occurred.

This underrun occurs when the RX-FIFO is empty in HS-SPICNT and the AHB master attempts to read data from it. The shift register state has no impact on the bit.

This flag generates a reception interrupt when HSSPIn_RXE:RFUE is set to "1".

[bit2] RFOS: "RX-FIFO Overrun Interrupt"

RX-FIFO overrun detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that an RX-FIFO overrun has occurred.

This overrun occurs when the RX-FIFO is full in HS-SPICNT and the engine in the controller writes data to it. The data that has already been written to the RX-FIFO when the RX-FIFO overrun occurs is protected from being overwritten. Consequently, the subsequent overflow data is discarded.

This controller can retain received data of up to 17 words in total by using the RX-FIFO and the shift register. If received data exceeds the limit by 1 or more bits, an overrun is considered to have occurred, and RFOS is set accordingly. The shift register state has an impact only on this bit.

This flag generates a reception interrupt when HSSPIn_RXE:RFOE is set to "1".

[bit1] RFES: "RX-FIFO is Empty Interrupt"

RX-FIFO empty detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that the RX-FIFO is empty. "1" is set in each GSSCGCLK cycle when conditions are satisfied. The shift register state has no impact on the bit.

This flag generates a reception interrupt when HSSPIn_RXE:RFEE is set to "1".

[bit0] RFFS: "RX-FIFO Full Interrupt": RX-FIFO full detection bit

Bit Value	Function
0	Clear the interrupt factor.
1	Detect the interrupt factor.

This bit is an interrupt factor flag indicating that the RX-FIFO is full. "1" is set in each GSSCGCLK cycle when conditions are satisfied. The shift register state has no impact on the bit.

This flag generates a transmission interrupt when HSSPIn_RXE:RFFE is set to "1".

The conditions for setting each RX-FIFO flag are as follows:

- RFMTS: The bit is set depending on "write pointer = read pointer + 16" or "write pointer > read pointer + RFTH value."
- RFLETS: The bit is set depending on "write pointer <= read pointer + RFTH value."
- RFUS: The bit is set depending on "write pointer = read pointer" and "read pulse."
- RFOS: The bit is set depending on "write pointer = read pointer + 16" and "write pulse."
- RFES: The bit is set depending on "write pointer = read pointer."
- RFFS: The bit is set depending on "write pointer = read pointer + 16" (RX-FIFO is full).

The above "write pulse" and "read pulse" mean signals that detected CPU access to the RX-FIFO. The "write pointer" is incremented by 1 each time that 1-word data is written to the RX-FIFO. The "read pointer" is incremented by 1 each time that 1-word data is read from the RX-FIFO. These pointers are cleared when "1" is written to RXFLSH.

1.22.6.8 HS-SPI Reception Interrupt Enable Register (HSSPIn_RXE)

This register sets whether to notify MCNT of reception interrupts according to the HSSPIn_RXF register state. Software needs to enable the respective bits in this register to use these interrupts.

Register Configuration

- HS-SPI reception interrupt enable register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI reception interrupt enable register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI reception interrupt enable register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI reception interrupt enable register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	RSSRE	RFMTE	RFLETE	RFUE	RFOE	RFEE	RFFE
Attribute	R0	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:7] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit6] RSSRE: "Slave Select Released Interrupt Enable"

Slave select released detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_RXF:RSSRS generates a reception interrupt.

[bit5] RFMTE: "RX-FIFO Fill Level is More than Threshold Interrupt Enable"

RX-FIFO-exceeded-threshold detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_RXF:RFMTS generates a reception interrupt.

[bit4] RFLETE: "RX-FIFO Fill Level is Less than or Equal to Threshold Interrupt Enable"

RX-FIFO-less-than-or-equal-to-threshold detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_RXF:RFLETS generates a reception interrupt.

[bit3] RFUE: "RX-FIFO Underrun Interrupt Enable"

RX-FIFO underrun detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_RXF:RFUS generates a reception interrupt.

[bit2] RFOE: "RX-FIFO Overrun Interrupt Enable"

RX-FIFO overrun detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_RXF:RFOS generates a reception interrupt.

[bit1] RFEE: "RX-FIFO and Shift Register are Empty Interrupt Enable"

RX-FIFO and shift register empty-state detection interrupt enable bit

Bit Value	Function
0	Disable interrupts. [Initial value]
1	Enable interrupts.

This bit sets whether HSSPIn_RXF:RFES generates a reception interrupt.

[bit0] RFFE: "RX-FIFO Full Interrupt Enable": RX-FIFO full detection interrupt enable bit

Bit Value	Function
0	Read value always "0" [Initial value]
1	Write-only

This bit sets whether HSSPIn_RXF:RFFS generates a reception interrupt.

1.22.6.9 HS-SPI Reception Interrupt Clear Register (HSSPIn_RXC)

This register clears the HSSPIn_RXF register state to "0".

Software can write "1" to a bit in this register to clear the corresponding interrupt factor flag in the HSSPIn_RXF register.

Register Configuration

■ HS-SPI reception interrupt clear register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI reception interrupt clear register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI reception interrupt clear register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI reception interrupt clear register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	RSSRC	RFMTC	RFLETC	RFUC	RFOC	RFEC	RFFC
Attribute	R0	W1/R0	W1/R0	W1/R0	W1/R0	W1/R0	W1/R0	W1/R0
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:7] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit6] RSSRC: "Slave Select Released Interrupt Clear"

Slave select released detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only

This bit is a control bit that clears HSSPIn_RXF:RSSRS.

[bit5] RFMTC: "RX-FIFO Fill Level is More than Threshold Interrupt Clear"

RX-FIFO-exceeded-threshold detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only

This bit is a control bit that clears HSSPIn_RXF:RFMTS.

[bit4] RFLETC: "RX-FIFO Fill Level is Less than or Equal to Threshold Interrupt"

RX-FIFO-less-than-or-equal-to-threshold detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only

This bit is a control bit that clears HSSPIn_RXF:RFLETS.

[bit3] RFUC: "RX-FIFO Underrun Interrupt Clear"

RX-FIFO underrun detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only

This bit is a control bit that clears HSSPIn_RXF:RFUS.

[bit2] RFOC: "RX-FIFO Overrun Interrupt Clear"

RX-FIFO overrun detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only

This bit is a control bit that clears HSSPIn_RXF:RFOS.

[bit1] RFEC: "RX-FIFO and Shift Register are Empty Interrupt Clear"

RX-FIFO and shift register empty-state detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only

This bit is a control bit that clears HSSPIn_RXF:RFES.

[bit0] RFFC: "RX-FIFO Full Interrupt Clear"

RX-FIFO full detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only

This bit is a control bit that clears HSSPIn_RXF:RFFS.

1.22.6.10 HS-SPI Fault Interrupt Factor Register (HSSPIn_FAULTF)

This register indicates the states of FAULT interrupt flags. HS-SPICNT does not allow masking of fault interrupts and notifies MCNT of the interrupts. For these fault interrupts that occur, software needs to determine the factors and perform recovery operations.

Register Configuration

■ HS-SPI fault interrupt factor register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI fault interrupt factor register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI fault interrupt factor register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI fault interrupt factor register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	PVFS	WAFS	UMAFS
Attribute	R0	R0	R0	R0	R0	R	R	R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:3] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit2] PVFS: "Protection Violation Fault"

Protection violation fault detection bit

Bit Value	Function
0	No protection violation occurred. [Initial value]
1	A protection violation occurred.

This bit is an interrupt factor flag indicating that a protection violation has occurred. This flag is set when any of the following conditions is satisfied.

- Read access is attempted to a register area where there is no register.
- There has been unauthorized access to a register.
  The width of access in the HSSPIn_TXFIFO0 to TXFIFO15 and HSSPIn_RXFIFO0 to RXFIFO15 registers does not match the FWIDTH value in the HSSPIn_FIFOCFG register.
- Write access is attempted to a register area that does not permit write access.
  This is determined in units of bytes.

No interrupt factor from this bit can be masked.

[bit1] WAFS: "Write Access Fault"

Write access fault detection bit

Bit Value	Function
0	No write access violation occurred. [Initial value]
1	A write access violation occurred.

This bit is an interrupt factor flag indicating that a write access violation has occurred. Also, no interrupt factor from this bit can be masked.

The bit is set in command sequencer mode because of write access to an area for which HSSPIn_CSCFG:SRAM has set "0" and the AHB master has assigned external serial memory on the system memory.

[bit0] UMAFS: "Unmapped Memory Access Fault"

Unmapped memory access fault detection bit

Bit Value	Function
0	No UMA violation occurred. [Initial value]
1	A UMA violation occurred.

This bit is an interrupt factor flag indicating an access violation of an unmapped memory area. No interrupt factor from this bit can be masked.

The bit is set when any of the following conditions is satisfied:

- The AHB master accessed a 256MB external Flash space in direct mode (HSSPIn_MCTRL:CSEN is "0").
- The AHB master accessed an external serial memory area not enabled in command sequencer mode (HSSPIn_MCTRL:CSEN is "1").
  The state indicated by the HSSPIn_CSCFG:SSEL0EN to SSEL3EN bits is not the enabled state.
- The AHB master accesses an area to which no external serial memory is assigned in command sequencer mode (HSSPIn_MCTRL:CSEN is "1"). This assignment is indicated by HSSPIn_CFGFG:MSEL.
- The AHB master accessed a 256MB external Flash space when HS-SPICNT is disabled (HSSPIn_MCTRL:MEN is "0").

1.22.6.11 HS-SPI Fault Interrupt Clear Register (HSSPIn_FAULTC)

This register clears an interrupt factor flag in the HSSPIn_FAULTF register.

The relevant interrupt factor flag in the HSSPIn_FAULTF register is cleared to "0" when "1" is written to a bit in this register.

Register Configuration

■ HS-SPI fault interrupt clear register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI fault interrupt clear register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI fault interrupt clear register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI fault interrupt clear register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	PVFC	WAFC	UMAFC
Attribute	R0	R0	R0	R0	R0	W1/R0	W1/R0	W1/R0
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:3] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit2] PVFC: "Protection Violation Fault Interrupt Clear"

Protection violation fault detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only

This bit is a control bit that clears HSSPIn_FATLTF:PVFS.

[bit1] WAFC: "Write Access Fault Interrupt Clear"

Write access fault detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only

This bit is a control bit that clears the HSSPIn_FATLTF:WAFS bit.

[bit0] UMAFC: "Unmapped Memory Access Fault Interrupt Clear"

Unmapped memory access fault detection clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the interrupt flag. Write-only

This bit is a control bit that clears HSSPIn_FATLTF:UMAFS.

1.22.6.12 HS-SPI Direct Mode Setting Register (HSSPIn_DMCFG)

This register configures the following operation of HS-SPICNT:

- Whether deassertion of the slave select should be software-controlled or performed in byte counter mode.

The register is used only when HS-SPICNT is in direct mode.

Register Configuration

- HS-SPI direct mode setting register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	SSDC	Reserved
Attribute	R0	R0	R0	R0	R0	R0	W/R	R1
Initial Value	0	0	0	0	0	0	0	1

Register Functions

[bit7:2] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit1] SSDC: "Slave Select Disassertion Control"

Slave select deassertion setting bit

Bit Value	Function
0	Software control The slave select is deasserted by HSSPIn_DMSTOP:STOP. [Initial value]
1	Byte counter mode The slave select is deasserted through HSSPIn_DMBCS:BCS.

The function of this bit is to select the slave select deassertion method.

[bit0] Reserved bit

The value read from the reserved bit is "1". Set "1" when writing to this bit.

1.22.6.13 HS-SPI Direct Mode Transfer Start Control Register (HSSPIn_DMSTART)

This register sets the start of serial transfer. The bit that controls this transfer start is set by writing of the register from the CPU. However, it is also set when the following condition is satisfied:

- HSSPIn_DMCFG:MSTARTEN is set to "1".

The register is used only in direct mode.

Register Configuration

- HS-SPI direct mode transfer start control register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	START
Attribute	R0	R0	R0	R0	R0	R0	R0	W1/R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit7:1] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit0] START: Transfer start bit

Bit Value	Function
0	No transfer start request [Initial value] Writing "0" is invalid.
1	Transfer start request

This bit is valid only with "1" written in it. Writing "0" is ignored. Also, writing "1" is ignored when the bit is already "1".

Serial transfer begins after "1" is written to this bit, which is then automatically returned to "0". Though the bit can be set during the serial transfer, there are the following restrictions.

- Transmission operation:

  The state when data is written to the TX-FIFO as many as "HSSPIn_FIFOCFG:TXFTH+1" stages after the previous serial transfer ends (the TSSRS bit is set) is a trigger for the start of the next serial transmission operation. After transfer begins, this bit is automatically cleared as described above.

- Reception operation:

  The state when all data is read from the RX-FIFO after the previous serial transfer ends (the RSSRS bit is set) is a trigger for the start of the next serial reception operation. After transfer begins, this bit is automatically cleared as described above.

1.22.6.14 HS-SPI Direct Mode Transfer Stop Control Register (HSSPIn_DMSTOP)

This register stops serial transfer when HS-SPICNT is in direct mode. To use this stop control, the serial interface must be set to software control mode (HSSPIn_DMCFG:SSDC is "0") beforehand.

Once STOP is set in this register, software must not clear the STOP until the controller stops the current serial transfer.

Register Configuration

- HS-SPI direct mode transfer stop control register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	STOP
Attribute	R0	R0	R0	R0	R0	R0	R0	W/R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit7:1] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit0] STOP: Transfer stop bit

Bit Value	Function
0	No transfer stop request [Initial value]
1	Transfer stop request

This bit is used only when software control (HSSPIn_DMCFG:SSDC is "0") is set for deassertion of the slave select. Otherwise, writing to this bit is ignored.

When "1" is written to this bit, operation of the slave select varies depending on the following conditions.

- TX-Only mode:

  With this bit set to "1", the transmission operation ends when all data in the TX-FIFO is transmitted.

- RX-Only mode:

  With this bit set to "1", the reception operation stops upon completion of the reception operation receiving the data currently being transferred to the shift register. For example, when the bit width of the FIFO is 32 bits (HSSPIn_FIFOCFG:FWIDTH[1:0] is "11") and the reception shift register has all of the data from the 32 bits, the data is transferred to the RX-FIFO and the operation stops.

- TX-and-RX mode:

  With this bit set to "1", the transmission/reception operation stops when all data in the TX-FIFO is transmitted or when the RX-FIFO becomes full.

When "0" is written to this bit, operation of the slave select varies depending on the following conditions.

- TX-Only mode:

  The writing of "0" is immediately reflected in the bit value but has no impact on the transmission operation.

- RX-Only mode:

  The writing of "0" is immediately reflected in the bit value. However, do not write "0" unless HSSPIn_RXF:RSSRS is "1". Otherwise, the reception operation cannot be guaranteed to stop, in which case the reception operation may continue until the RX-FIFO becomes full.

- TX-and-RX mode:

  The writing of "0" is immediately reflected in the bit value but has no impact on the transmission/reception operation.

Once set, this bit stays in the state of "1" until cleared by a software operation. To start the subsequent data transfer, an operation to clear the bit is required first.

1.22.6.15 HS-SPI Direct Mode Slave Select Register (HSSPIn_DMPSEL)

This register enables serial transfer by HS-SPICNT with 1 signal selected from 4 slave select signals.

The register is used only in direct mode.

Register Configuration

- HS-SPI direct mode slave select register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	PSEL[1]	PSEL[0]
Attribute	R0	R0	R0	R0	R0	R0	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit7:2] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit1:0] PSEL "Peripheral Select": Peripheral select bits

Bit Value	Function
00	Select slave 0. [Initial value]
01	Select slave 1.
10	Select slave 2.
11	Select slave 3.

These bits select and enable 1 of 4 slave selects.

1.22.6.16 HS-SPI Direct Mode Transfer Protocol Setting Register (HSSPIn_DMTRP)

This register sets the protocol for serial transfer.

The register is used only when HS-SPICNT is in direct mode.

Register Configuration

- HS-SPI direct mode transfer protocol setting register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	TRP[3]	TRP[2]	TRP[1]	TRP[0]
Attribute	R0	R0	R0	R0	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit7:4] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit3:2] TRP "Transfer Protocol": Transfer protocol setting bits (transfer direction)

Bit Value	Function
00	Set the protocol for TX-and-RX. [Initial value] In this case, TRP[1:0] must be set to "00".
01	Set the protocol for RX-Only.
10	Set the protocol for TX-Only.
11	Reserved

[bit1:0] TRP "Transfer Protocol": Transfer protocol setting bits (transfer bit width)

Bit Value	Function
00	Single-bit mode (serial) [Initial value]
01	Dual-bit mode (2-bit width) In this case, TRP[3:2] must be set to "01" or "10".
10	Quad-bit mode (4-bit width) In this case, TRP[3:2] must be set to "01" or "10".
11	Reserved

Warning:

- Transmission operation
  With this register used in the default state, operation is in TX-and-RX mode and single-bit mode. If only the transmission operation is performed in this state, the RX-FIFO retains the same amount of data as transmitted. The transmission of data of the maximum size of the TX-FIFO causes the RX-FIFO to become full. To be precise, the shift register on the RX side also retains data until full (HSSPIn_RXF:RFFS is "1"), so the transmission operation stops when a second transmission operation transmits data of the same size as the shift register (and that varies depending on the HSSPIn_FIFOCFG:FWIDTH setting). To continue the transmission operation, the received data must be read from the RX-FIFO. Therefore, processing of the transmission operation only is as follows.
  - ☐ In TX-and-RX mode, while transmission data is written to the TX-FIFO, received data must be read from the RX-FIFO (regardless of whether the received data is valid or invalid).
  - ☐ In TX-Only mode, the transmission operation can continue with only a write operation to the TX-FIFO.

- Operation of this register

  Operation of this register is prohibited during serial transfer. Check the HSSPIn_TXF or HSSPIn_RXF register to confirm that FIFO data transfer is stopped, before operating the HSSPIn_DMTRP register.

1.22.6.17 HS-SPI Direct Mode Transfer Byte Count Setting Register (HSSPIn_DMBCC)

This register sets the amount of data transferred by serial transfer in units of bytes. To make this setting, flow control must be in byte counter mode ("1" is set in HSSPIn_DMCFG:SSDC).

The register is used only when HS-SPICNT is in direct mode.

Register Configuration

■ HS-SPI direct mode transfer byte count setting register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	BCC[15]	BCC[14]	BCC[13]	BCC[12]	BCC[11]	BCC[10]	BCC[9]	BCC[8]
Attribute	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI direct mode transfer byte count setting register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	BCC[7]	BCC[6]	BCC[5]	BCC[4]	BCC[3]	BCC[2]	BCC[1]	BCC[0]
Attribute	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit15:0] BCC "Byte Count Control": Transferred byte count setting value

These bits set the amount of data transferred by serial transfer in units of bytes. BCC is used only when HS-SPICNT is in direct mode. Before transferring data, software needs to set a data amount for any of {transmission/reception/both} in units of bytes.

The subtraction counter captures this BCC when serial transfer begins, and 1 is subtracted from the counter each time that serial transfer of 1-byte data is completed. HS-SPICNT finishes the transfer and deasserts the slave select when the subtraction count reaches 0.

Note that BCC may not be a multiple of FWIDTH, so also see the related description of byte order in the sections about the HSSPIn_TXFIFO0 to TXFIFO15 registers and the HSSPIn_RXFIFO0 to RXFIFO15 registers.

Warning:

Set a value of 1 or higher as the BCC setting.

1.22.6.18 HS-SPI Direct Mode Transfer Remaining Count Register (HSSPIn_DMBCS)

This register is read-only and used by software to check the currently remaining bytes to be transferred. The register is enabled when the HS-SPICNT settings satisfy all of the following conditions:

- Direct mode
- Flow control in byte counter mode (HSSPIn_DMCFG:SSDC is "1")

Register Configuration

- HS-SPI direct mode transfer remaining count register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	BCS[15]	BCS[14]	BCS[13]	BCS[12]	BCS[11]	BCS[10]	BCS[9]	BCS[8]
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI direct mode transfer remaining count register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	BCS[7]	BCS[6]	BCS[5]	BCS[4]	BCS[3]	BCS[2]	BCS[1]	BCS[0]
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit15:0] BCS "Byte Count Status": Number of remaining bytes to transfer

These bits are read-only. They indicate the remaining data in the current serial transfer in units of bytes. BCS is used only when all of the following conditions are satisfied.

- HS-SPICNT is in direct mode.
- The HSSPIn_DMCFG:SSDC setting is "1" (byte counter mode).

If BCC[15:0] is set when serial transfer begins, BCC is passed as the initial value of BCS.

Then, each time that data is transferred between the TX/RX-FIFO and the shift register, "FWIDTH[1:0]+1" is subtracted from BCS. This register is set to "0" when the serial transfer ends normally.

1.22.6.19 HS-SPI Direct Mode Status Register (HSSPIn_DMSTATUS)

This register contains status bits and indicates the transmission/reception route in operation and how much data is contained in the current transmission/reception FIFO.

The register is used only when HS-SPICNT is in direct mode.

Register Configuration

- HS-SPI direct mode status register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI direct mode status register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	TXFLEVEL [4]	TXFLEVEL [3]	TXFLEVEL [2]	TXFLEVEL [1]	TXFLEVEL [0]
Attribute	R0	R0	R0	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI direct mode status register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	RXFLEVEL [4]	RXFLEVEL [3]	RXFLEVEL [2]	RXFLEVEL [1]	RXFLEVEL [0]
Attribute	R0	R0	R0	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI direct mode status register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	TXACTIVE	RXACTIVE
Attribute	R0	R0	R0	R0	R0	R0	R	R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:21] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit20:16] TXFLEVEL[4:0]: "Current fill level of TX-FIFO"

Remaining TX-FIFO data indication bits

These bits indicate the amount of data currently remaining in the TX-FIFO, in the set unit of FWIDTH.

[bit15:13] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit12:8] RXFLEVEL[4:0]: "Current fill level of RX-FIFO"

Remaining RX-FIFO data indication bits

These bits indicate the amount of data currently remaining in the RX-FIFO by using the FWIDTH setting.

[bit1] TXACTIVE: Transmission status bit

Bit Value	Function
0	Serial transmission not in progress [Initial value]
1	Serial transmission in progress

This bit indicates that a serial data transmission operation is in progress.

HSSPIn_DMSTART:START is set in order to set the start point from which a signal is transmitted to the serial interface. It is cleared at the end of transmission of the last bit of the transmission data.

[bit0] RXACTIVE: Reception status bit

Bit Value	Function
0	Serial reception not in progress [Initial value]
1	Serial reception in progress

This bit indicates that a serial data reception operation is in progress.

HSSPIn_DMSTART:START is set in order to set the start point from which a signal is received by the serial interface. It is cleared at the end of reception of the last bit of the data.

1.22.6.20 HS-SPI Remaining Transmission Bit Count Register (HSSPIn_TXBITCNT)

This register indicates how many bits remain in the transmission shift register after serial transfer ends or stops.

The register is used only when HS-SPICNT is in direct mode.

Register Configuration

- HS-SPI remaining transmission bit count register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	TXBIT CNT[5]	TXBIT CNT[4]	TXBIT CNT[3]	TXBIT CNT[2]	TXBIT CNT[1]	TXBIT CNT[0]
Attribute	R0	R0	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit7:6] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit5:0] TXBITCNT[5:0] "Transmit Bit Count": Transmit bit count value

This bit field indicates how many untransmitted bits remain in the transmission shift register. It can have the following values.

0: No untransmitted bit remains.
1: 1 untransmitted bit remains.
...
31: 31 untransmitted bits remain.
32: 32 untransmitted bits remain.

TXBITCNT is updated when:

- HS-SPICNT stops the data transmission operation.

1.22.6.21 HS-SPI Remaining Reception Bit Count Register (HSSPIn_RXBITCNT)

This register indicates how many bits remain in the reception shift register after serial transfer ends or stops. Remaining bits are bits that are in a stage before transfer to the RX-FIFO.

Register Configuration

- HS-SPI remaining reception bit count register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	RXBIT CNT[5]	RXBIT CNT[4]	RXBIT CNT[3]	RXBIT CNT[2]	RXBIT CNT[1]	RXBIT CNT[0]
Attribute	R0	R0	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit7:6] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit5:0] RXBITCNT[5:0] "Receive Bit Count": Receive bit count value

This bit field indicates how many bits of data remain in the reception shift register. It can have the following values.

0: No valid bit remains. This means that the reception shift register has no "data that has not been transmitted to the RX-FIFO."
1: 1 valid bit remains.
...
31: 31 valid bits remain.
32: 32 valid bits remain.

1.22.6.22 HS-SPI TX-FIFO Registers (HSSPIn_TXFIFO0 to TXFIFO15)

This set of registers is used to write data to the TX-FIFO. These 16 registers for writing data to the TX-FIFO are at consecutive locations on the register map. They all have the same functions. This configuration is employed because the AHB protocol does not support burst access to the same address. This section describes HSSPIn_TXFIFO0, 1 of these 16 registers, as a representative.

These registers are accessible only with a bus width that is the same as the FIFO width. Access with an incorrect bus width causes a protection violation.

Once written to these registers, data has a width of 33 bits and is written to the TX-FIFO. The highest bit, the 33rd bit, is called the TXCTRL bit and has the value of HSSPIn_FIFOCFG:TXCTRL.

Register Configuration

- HS-SPI TX-FIFO0 data write register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	TXDATA [31]	TXDATA [30]	TXDATA [29]	TXDATA [28]	TXDATA [27]	TXDATA [26]	TXDATA [25]	TXDATA [24]
Attribute	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI TX-FIFO0 data write register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	TXDATA [23]	TXDATA [22]	TXDATA [21]	TXDATA [20]	TXDATA [19]	TXDATA [18]	TXDATA [17]	TXDATA [16]
Attribute	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI TX-FIFO0 data write register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	TXDATA [15]	TXDATA [14]	TXDATA [13]	TXDATA [12]	TXDATA [11]	TXDATA [10]	TXDATA [9]	TXDATA [8]
Attribute	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI TX-FIFO0 data write register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	TXDATA [7]	TXDATA [6]	TXDATA [5]	TXDATA [4]	TXDATA [3]	TXDATA [2]	TXDATA [1]	TXDATA [0]
Attribute	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0	W/R0
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:0] TXDATA[31:0] "TX-FIFO Register 0": TX-FIFO0 write data

This bit field is used to write transmission data to the TX-FIFO. When data is written to this 32-bit wide register, the write position moves in the forward direction in the TX-FIFO. This write operation does not depend on the order of the HSSPIn_TXFIFO0 to HSSPIn_TXFIFO15 registers.

Each time data is written to TXDATA[31:0], 1 bit is added at a high order of the 32-bit wide data. The bit reflects information from HSSPIn_FIFOCFG:TXCTRL. To include output Hi-Z in the data to be written to the TX-FIFO, "1" must be written to HSSPIn_FIFOCFG:TXCTRL before the data is written, and then TXDATA[0] is set according to the Hi-Z duration.

TXCTRL	TXDATA[0]	
0	--	An effective range of TXDATA[31:0] is output as is to SDATA as data.
1	0	1-byte data of serial data output enters the Hi-Z state. Data in TXDATA[31:0] is not transmitted.
1	1	Data in the lower 8 bits in TXDATA[31:0] is used. Among these 8 bits, the higher 4 bits are transmitted, and SDATA output enters the Hi-Z state for the subsequent 4-bit duration.

Warning:

If "1" is set in the TXCTRL bit, written data is handled as 8 bits regardless of the FIFO bit width. The LSB in the effective bit range set by the FWIDTH value corresponds to the above-mentioned "TX-FIFO Data Bit [0]."

The following restrictions on the bus width used when writing data to TXDATA[31:0] depend on the FIFO width (FWIDTH[1:0]). For any access that violates these restrictions, the write operation through the access becomes invalid, and an interrupt (originating from PVFS) is asserted.

FWIDTH[1:0]	Access to TXDATA[31:0]
0 (8-bit)	8-bit access
1 (16-bit)	16-bit access
2 (24-bit)	32-bit access
3 (32-bit)	32-bit access

For details on the order in which data written to TXDATA[31:0] is transmitted to the serial interface, see the description of the SENDIAN and SDIR control bits in the HSSPIn_PCC0 to PCC3 registers.

If the TX-FIFO is full and software writes new data, a TX-FIFO overrun occurs. The authenticity of the transmission data is not guaranteed when this overrun occurs. To prevent the overrun from occurring, software needs to confirm that the TX-FIFO is not full before writing data to the TX-FIFO.

1.22.6.23 HS-SPI RX-FIFO Registers (HSSPIn_RXFIFO0 to RXFIFO15)

This set of registers is used to read data from the RX-FIFO. These 16 registers for reading data from the RX-FIFO are at consecutive locations on the register map. They all have the same functions. This configuration is employed because the AHB protocol does not support burst access to the same address. This section describes HSSPIn_RXFIFO0, 1 of these 16 registers, as a representative.

These registers are accessible only with a bus width that is the same as the FIFO width. Access with an incorrect bus width causes a protection violation.

Received data in the RX-FIFO can be read through these registers.

Register Configuration

- HS-SPI RX-FIFO0 data read register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	RXDATA [31]	RXDATA [30]	RXDATA [29]	RXDATA [28]	RXDATA [27]	RXDATA [26]	RXDATA [25]	RXDATA [24]
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI RX-FIFO0 data read register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	RXDATA [23]	RXDATA [22]	RXDATA [21]	RXDATA [20]	RXDATA [19]	RXDATA [18]	RXDATA [17]	RXDATA [16]
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI RX-FIFO0 data read register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	RXDATA [15]	RXDATA [14]	RXDATA [13]	RXDATA [12]	RXDATA [11]	RXDATA [10]	RXDATA [9]	RXDATA [8]
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI RX-FIFO0 data read register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	RXDATA [7]	RXDATA [6]	RXDATA [5]	RXDATA [4]	RXDATA [3]	RXDATA [2]	RXDATA [1]	RXDATA [0]
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:0] RXDATA[31:0] "RX-FIFO Register 0": RX-FIFO0 read data

This bit field is used to read received data from the RX-FIFO. When data is read from this 32-bit wide register, the read position moves in the forward direction in the RX-FIFO. This read operation does not depend on the order of the HSSPIn_RXFIFO0 to HSSPIn_RXFIFO15 registers.

The following restrictions on the bus width used when reading data from RXDATA[31:0] depend on the FIFO width (FWIDTH[1:0]). For any access that violates these restrictions, the read operation through the access becomes invalid, and an interrupt (originating from PVFS) is asserted.

FWIDTH[1:0]	Access to RXDATA[31:0] (data width on AHB)
0 (8-bit)	8-bit access
1 (16-bit)	16-bit access
2 (24-bit)	32-bit access
3 (32-bit)	32-bit access

For details on the bit field range in which data received from the serial interface is written to RXDATA[31:0], see the description of the SENDIAN and SDIR control bits in the HSSPIn_PCC0 to PCC3 registers.

If the RX-FIFO is empty and software attempts to read new data, an RX-FIFO underrun occurs. An interrupt originating from the HSSPIn_RXF:RFUS bit is generated when this underrun occurs.

1.22.6.24 HS-SPI FIFO Setting Register (HSSPIn_FIFOCFG)

This register is used to configure TX-FIFO and RX-FIFO operations. Software can set the FIFO threshold and the FIFO data bit width.

Software can also empty the transmission and reception FIFOs by using the TXFLSH and RXFLSH bit functions.

Register Configuration

- HS-SPI FIFO setting register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI FIFO setting register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI FIFO setting register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	TXFLSH	RXFLSH	TXCTRL	FWIDTH [1]	FWIDTH [0]
Attribute	R0	R0	R0	W1/R0	W1/R0	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI FIFO setting register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	TXFTH [3]	TXFTH [2]	TXFTH [1]	TXFTH [0]	RXFTH [3]	RXFTH [2]	RXFTH [1]	RXFTH [0]
Attribute	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	0	1	1	1	0	1	1	1

Register Functions

[bit31:13] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit12] TXFLSH "TX-FIFO Flush": TX-FIFO clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the TX-FIFO. Write-only is enabled. Transmission of the data currently being transmitted from the shift register continues to the end, and the shift register does not capture the subsequent data. Also, the HSSPIn_DMBCS register stops at the current value. The HSSPIn_TXF register becomes as follows: TFLETS: "1" TFES: "1" if no untransmitted bits remain in the transmission shift register

Warning:

- The function of this bit is to initialize the TX-FIFO internal state. It does not clear data.
- Operations that set the TXFLSH bit are prohibited during serial data transfer.
  If such an operation is performed, the data to transfer cannot be guaranteed.

[bit11] RXFLSH "RX-FIFO Flush": RX-FIFO clear bit

Bit Value	Function
0	Read value always "0" [Initial value] Writing "0" is invalid.
1	Clear the RX-FIFO. Write-only is enabled. The data currently being received and data in the RX-FIFO become invalid, but data in the reception shift register is not affected. Therefore, after an operation is performed on this bit, the RX-FIFO accumulates the subsequently received data as new data. The HSSPIn_RXF register becomes as follows: RFLETS: "1" RFES: "1"

Warning:

- The function of this bit is to initialize the RX-FIFO internal state. It does not clear data.
- Operations that set the RXFLSH bit are prohibited during serial data transfer.
  If such an operation is performed, the data to transfer cannot be guaranteed.

[bit10] TXCTRL "TX-FIFO Control": TX-FIFO transmission data control bit

Bit Value	Function
0	Data written to the TX-FIFO (HSSPIn_TXFIFO0 to TXFIFO15) is transmitted as data as is.[Initial value]
1	Data written to the TX-FIFO (HSSPIn_TXFIFO0 to TXFIFO15) becomes transmission control as follows depending on the lowest bit in the effective bit range. 0: 1-byte data of serial data output enters the Hi-Z state. The relevant data in the TX-FIFO is not transmitted. 1: The relevant data in the lower 8 bits in the TX-FIFO is used. Among these 8 bits, the higher 4 bits are transmitted, and SDATA output enters the Hi-Z state for the subsequent 4-bit duration.

This bit is reflected by the highest bit (33rd bit) of the data written to the TX-FIFO. That bit is used when software writes data to the HSSPIn_TXFIFO0 to TXFIFO15 registers.

Software needs to set a value in this bit before writing data to the HSSPIn_TXFIFO0 to TXFIFO15 registers. The software determines whether the data in the TX-FIFO is data that should be transmitted, and sets this bit to "1" or "0" accordingly.

Warning:

If "1" is set in the TXCTRL bit, written data is handled as 8 bits regardless of the FIFO bit width. The LSB in the effective bit range set by the FWIDTH value corresponds to the above-mentioned "TX-FIFO Data Bit [0]."

[bit9:8] FWIDTH[1:0] "FIFO WIdth": FIFO bit width setting value

Bit Value	Function
00	8-bit width for all of the TX-FIFO, RX-FIFO, and shift registers
01	16-bit width for all of the TX-FIFO, RX-FIFO, and shift registers
10	24-bit width for all of the TX-FIFO, RX-FIFO, and shift registers
11	32-bit width for all of the TX-FIFO, RX-FIFO, and shift registers

This bit field indicates the effective bit width in the FIFO.

[bit7:4] TXFTH[3:0] "TX-FIFO Threshold Level": TX-FIFO threshold

This bit field indicates the TX-FIFO threshold. This value has an impact on the HSSPIn_TXF:TFMTS and TFLETS bit operations.

[bit3:0] RXFTH[3:0] "RX-FIFO Threshold Level": RX-FIFO threshold

This bit field indicates the RX-FIFO threshold. This value has an impact on the HSSPIn_RXF:RFMTS and RFLETS bit operations.

1.22.6.25 HS-SPI Command Sequencer Setting Register (HSSPIn_CSCFG)

This register sets the command sequencer mode of HS-SPICNT. This register must be set before command sequencer mode is enabled. The settings include the type of transfer protocol, enabling/disabling writing to memory, and the capacity of the memory device connected to HS-SPICNT.

Register Configuration

- HS-SPI command sequencer setting register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI command sequencer setting register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	MSEL [3]	MSEL [2]	MSEL [1]	MSEL [0]
Attribute	R0	R0	R0	R0	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI command sequencer setting register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	Reserved	Reserved	Reserved	Reserved	SSEL3EN	SSEL2EN	SSEL1EN	SSEL0EN
Attribute	R0	R0	R0	R0	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI command sequencer setting register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	MBM [1]	MBM [0]	SRAM
Attribute	R0	R0	R0	R0	R0	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:20] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit19:16] MSEL[3:0] "Memory Device Selection bits": Memory device selection bits

This bit field represents an address space on AHB associated with each slave select.

It also represents the size of each memory bank in the selected memory device. These bit values are used with the command sequencer and have the following 2 meanings.

1. They select which 1 of the 4 slave selects to assert.
2. They select the memory bank size in the selected memory device.

For details, see the "1.22.5 Command Sequencer Mode" section.

[bit15:12] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit11] SSEL3EN "Slave Select 3 Enable": Slave select 3 enable bit

Bit Value	Function
0	Disable access to the area to which the slave select 3 memory is assigned. Access to this area causes UMAF. [Initial value]
1	Enable access to the area to which the slave select 3 memory is assigned.

This bit is a control bit that indicates whether slave select 3 is enabled.

[bit10] SSEL2EN "Slave Select 2 Enable": Slave select 2 enable bit

Bit Value	Function
0	Disable access to the area to which the slave select 2 memory is assigned. Access to this area causes UMAF. [Initial value]
1	Enable access to the area to which the slave select 2 memory is assigned.

This bit is a control bit that indicates whether slave select 2 is enabled.

[bit9] SSEL1EN "Slave Select 1 Enable": Slave select 1 enable bit

Bit Value	Function
0	Disable access to the area to which the slave select 1 memory is assigned. Access to this area causes UMAF. [Initial value]
1	Enable access to the area to which the slave select 1 memory is assigned.

This bit is a control bit that indicates whether slave select 1 is enabled.

[bit8] SSEL0EN "Slave Select 0 Enable": Slave select 0 enable bit

Bit Value	Function
0	Disable access to the area to which the slave select 0 memory is assigned. Access to this area causes UMAF. [Initial value]
1	Enable access to the area to which the slave select 0 memory is assigned.

This bit is a control bit that indicates whether slave select 0 is enabled.

[bit7:6] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit2:1] MBM[1:0] "Multi Bit Mode": SPI data width setting bits

Bit Value	Function
00	Access to the memory device in command sequencer mode uses single-bit mode. The data read from serial memory is captured from QSPI_SIO0. Memory instruction, address, and other control codes are transmitted from QSPI_SIO1. QSPI_SIO3 and QSPI_SIO2 output is fixed at the Hi-Z state. [Initial value]
01	Access to the memory device in command sequencer mode is a half-duplex line, and it uses dual-bit mode. The data read from serial memory is captured from QSPI_SIO1 and QSPI_SIO0. Memory instruction, address, and other control codes are transmitted from QSPI_SIO1 and QSPI_SIO0.
10	Access to the memory device in command sequencer mode uses quad-bit mode. The data read from serial memory is captured from QSPI_SIO3, QSPI_SIO2, QSPI_SIO1, and QSPI_SIO0. Memory instruction, address, and other control codes are transmitted from QSPI_SIO3, QSPI_SIO2, QSPI_SIO1, and QSPI_SIO0.
11	Reserved

This bit field is control bits that define the widths used by QSPI_SIO3, QSPI_SIO2, QSPI_SIO1, and QSPI_SIO0. These widths directly determine the type of SPI protocol.

[bit0] SRAM "Serial SRAM /Serial FlashMemory Type Select"

"Readable/Writable"/"Read only" selection bit

Bit Value	Function
0	Disable write access. With this setting, an operation to write to the serial memory area causes WAFS. [Initial value]
1	Enable write access.

Even if read-only memory is connected, set this bit to "1" to configure the registers of the memory (e.g., bit width and address width).

Warning:

After updating this HSSPIn_CSCFG register, read the register values again to confirm that settings are reflected before accessing memory.

1.22.6.26 HS-SPI Command Sequencer Idle Timer Setting Register (HSSPIn_CSITIME)

This register is used only in command sequencer mode to set the timer that measures the AHB interface idle time. The transaction of the serial interface ends when the timer reaches the cycle count value in this register.

Software needs to set this register before enabling command sequencer mode.

Register Configuration

■ HS-SPI command sequencer idle timer setting register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI command sequencer idle timer setting register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

■ HS-SPI command sequencer idle timer setting register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	ITIME [15]	ITIME [14]	ITIME [13]	ITIME [12]	ITIME [11]	ITIME [10]	ITIME [9]	ITIME [8]
Attribute	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	1	1	1	1	1	1	1	1

■ HS-SPI command sequencer idle timer setting register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	ITIME [7]	ITIME [6]	ITIME [5]	ITIME [4]	ITIME [3]	ITIME [2]	ITIME [1]	ITIME [0]
Attribute	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	1	1	1	1	1	1	1	1

Register Functions

[bit31:16] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit15:0] ITIME[15:0] "Idle Timer": Idle timer setting value

This bit field is used only in command sequencer mode to specify the time to wait during the AHB interface idle time.

HS-SPICNT keeps the slave select asserted after completing reading/writing of the requested amount of data on the serial interface. After that, the slave select is deasserted when the AHB interface has no memory access for the cycle specified by these bits. This can improve performance when all of the following conditions are satisfied.

- Accesses to serial memory are of the same type (read-only or write-only).
- The access range is a range of continuous addresses.
  - ☐ Single access on AHB:

    4byte access with address [31:0] incremented by +4
  - ☐ Burst access on AHB:

    Support of any of 1-byte access, 2-byte access, and 4-byte access
- The next access occurs within the period specified by these bits.

In such cases, the set command sequence in HSSPIn_RDCSDC0 to RDCSDC7 and HSSPIn_WRCSDC0 to WRCSDC7 is issued, in the middle of its course, only 1 time at the beginning. It is omitted for the subsequent period of continuous operation. The serial clock is driven as needed for data transfer during this period of command sequence omission. Once this set of continuous accesses ends, the slave select is deasserted. Then, the next transaction on the serial interface is issued based on the set command sequence in HSSPIn_RDCSDC0 to RDCSDC7 or HSSPIn_WRCSDC0 to WRCSDC7.

Warning:

After updating this HSSPIn_CSITIME register, read the register values again to confirm that settings are reflected before accessing memory.

1.22.6.27 HS-SPI Command Sequencer Address Extension Register (HSSPIn_CSAEXT)

This register extends a space allocated on memory in command sequencer mode.

Software needs to configure this register to use the address extension function. This function provides virtual access to up to 16 GB of serial memory. When not using this address extension function, software must set all bits in this register to "0".

Register Configuration

- HS-SPI command sequencer address extension register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	AEXT [18]	AEXT [17]	AEXT [16]	AEXT [15]	AEXT [14]	AEXT [13]	AEXT [12]	AEXT [11]
Attribute	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI command sequencer address extension register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	AEXT [10]	AEXT [9]	AEXT [8]	AEXT [7]	AEXT [6]	AEXT [5]	AEXT [4]	AEXT [3]
Attribute	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI command sequencer address extension register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	AEXT [2]	AEXT [1]	AEXT [0]	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	W/R	W/R	W/R	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI command sequencer address extension register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved	Reserved
Attribute	R0	R0	R0	R0	R0	R0	R0	R0
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit31:13] AEXT[18:0] "Address Extension Bits": Address extension bits

This bit field is used only in command sequencer mode to extend address values. The command sequencer uses the set values in this field as the bit range [31:13] of a memory address. In command sequencer mode, the address generated for each slave select is a combination of the values in these bits and the AHB address value.

These bits must be set to 0x00000000 when no address extension is used.

[bit12:0] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

Warning:

After updating this HSSPIn_CSAEXT register, read the register values again to confirm that settings are reflected before accessing memory.

1.22.6.28 HS-SPI Read Command Sequence Data/Control Registers (HSSPIn_RDCSDC0 to RDCSDC7)

This set of registers consists of 8 groups of data/control registers. They set the phase of serial transfer to be generated by the command sequencer during a memory read operation. These registers are used only in command sequencer mode.

These 8 groups of registers have the same structure, so this section describes the HSSPIn_RDCSDC0 register as a representative.

Register Configuration

- HS-SPI read command sequence data/control register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	RDCS DATA[7]	RDCS DATA[6]	RDCS DATA[5]	RDCS DATA[4]	RDCS DATA[3]	RDCS DATA[2]	RDCS DATA[1]	RDCS DATA[0]
Attribute	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI read command sequencer data/control register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	CONT	TRP[1]	TRP[0]	DEC
Attribute	R0	R0	R0	R0	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit15:8] RDCSDATA[7:0] "Command Sequencer Data/Control Byte for Memory-Readtransactions"

Read command sequencer data/control setting values

This bit field is a register area for writing the data and control codes to be provided to the command sequencer. DEC is a bit that controls whether to transmit RDCSDATA[7:0] as is or to decode and then transmit RDCSDATA[7:0]. The following table shows the relationship between DEC and RDCSDATA[7:0].

DEC	RDCSDATA [2:0]	Description
0	Don't care	Transmits RDCSDATA[7:0] as is.
1	000	Transmits address[7:0] to access serial memory.
1	001	Transmits address[15:8] to access serial memory.
1	010	Transmits address[23:16] to access serial memory.
1	011	Transmits address[31:24] to access serial memory.
1	100	Sets SDATA output to the Hi-Z state for the duration of 1-byte data.
1	101	Transmits data as follows regardless of the HSSPIn_PCC0 to PCC3:SDIR setting. 1. The RDCSDATA[7:4] value is transmitted as is. The transmission sequence of these data bits complies with the HSSPIn_PCC0 to PCC3:SDIR setting. 2. SDATA output stays in the Hi-Z state for the duration of 4-bit data.
1	111	Indicates the end of the list.
1	Others	Reserved

Warning:

The "end of the list" setting (DEC is "1" and RDCSDATA[2:0] is "111") is prohibited in the HSSPIn_RDCSDC0 register.

[bit7:4] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit3] CONT "Continuous": Continuous instruction setting bit

Bit Value	Function
0	Do not omit the list. [Initial value]
1	Omit the list.

This bit is a control bit that omits command transmission for the second and subsequent times. To see whether a command can be omitted for the second and subsequent times when using this bit, check the command set on the data sheet of the serial memory used.

Warning:

- To cancel omission of the list, cancel the toggle of the RDCSDC0 to RDCSDC7 registers and perform a read operation. Then, set "0" in all the CONT bits in the RDCSDC0 to RDCSDC7 registers. As a result, from the next access time, there is no omission of the list.
- If the end of the list is not found in any of RDCSDC0 to RDCSDC7, the set value of the CONT bit in RDCSDC7 is used.
- When setting the end of the list in any of the RDCSDC0 to RDCSDC7 registers, set the CONT bit in that register to "0".
- When not setting the end of the list in the RDCSDC0 to RDCSDC7 registers, set the CONT bit in the RDCSDC7 register to "0".
- Among all the CONT bits in the HSSPIn_RDCSDC0 to RDCSDC7 registers, the setting of "1" in any CONT bit in the HSSPIn_RDCSDC1 to RDCSDC7 registers is prohibited.

[bit2:1] TRP "Transfer Protocol": Serial interface width control bits

Bit Value	Function
00	Serial interface bit width complies with HSSPIn_CSCFG:MBM[1:0] setting [Initial value]
01	Dual-bit mode
10	Quad-bit mode
11	Single-bit mode (serial)

These bit values are control bits that set the bit width used for transmitting RDCSDATA[7:0] to or receiving RDCSDATA[7:0] from the serial interface.

Warning:

If the end of the list is not found in any of RDCSDC0 to RDCSDC7, the set value of the TRP bit in RDCSDC7 is used.

[bit0] DEC "Decode": Decode control bit

Bit Value	Function
0	Transmit RDCSDATA[7:0] as 1-byte data as is. [Initial value]
1	Decode RDCSDATA[2:0] as a control code.

This bit is a control bit that determines whether RDCSDATA[7:0] is data or a control code.

Warning:

After updating any of these HSSPIn_RDCSDC0 to RDCSDC7 registers, read the register values again to confirm that settings are reflected before accessing memory.

1.22.6.29 HS-SPI Write Command Sequence Data/Control Registers (HSSPIn_WRCSDC0 to WRCSDC7)

This set of registers consists of 8 groups of data/control registers. They set the phase of serial transfer to be generated by the command sequencer during a memory write operation. These registers are used only in command sequencer mode.

These 8 sets of registers have the same structure, so this section describes the HSSPIn_WRCSDC0 register as a representative.

Register Configuration

- HS-SPI write command sequence data/control register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	WRCS DATA[7]	WRCS DATA[6]	WRCS DATA[5]	WRCS DATA[4]	WRCS DATA[3]	WRCS DATA[2]	WRCS DATA[1]	WRCS DATA[0]
Attribute	W/R	W/R	W/R	W/R	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI write command sequencer data/control register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	Reserved	Reserved	Reserved	Reserved	CONT	TRP[1]	TRP[0]	DEC
Attribute	R0	R0	R0	R0	W/R	W/R	W/R	W/R
Initial Value	0	0	0	0	0	0	0	0

Register Functions

[bit15:8] WRCSDATA[7:0] "Command Sequencer Data/Control Byte for Memory-Write transactions"

Write command sequencer data/control setting values

This bit field is a register area for writing the data and control codes to be provided to the command sequencer. DEC is a bit that controls whether to transmit WRCSDATA[7:0] as is or to decode and then transmit WRCSDATA[7:0]. The following table shows the relationship between DEC and WRCSDATA[7:0].

DEC	WRCSDATA [2:0]	Description
0	Don't care	Transmits WRCSDATA[7:0] as is.
1	000	Transmits address[7:0] to access serial memory.
1	001	Transmits address[15:8] to access serial memory.
1	010	Transmits address[23:16] to access serial memory.
1	011	Transmits address[31:24] to access serial memory.
1	100	Sets SDATA output to the Hi-Z state for the duration of 1-byte data.
1	101	Transmits data as follows regardless of the HSSPIn_PCC0 to PCC3:SDIR setting. 1) WRCSDATA[7:4] is transmitted as is. The transmission sequence of these data bits complies with the HSSPIn_PCC0 to PCC3:SDIR setting. 2) SDATA output stays in the Hi-Z state for the duration of 4-bit data.
1	111	Indicates the end of the list.
1	Others	Reserved

[bit7:4] Reserved bits

The value read from a reserved bit is "0". Set "0" when writing to a reserved bit.

[bit3] CONT "Continuous": Continuous instruction setting bit

Bit Value	Function
0	Do not omit the list. [Initial value]
1	Omit the list.

Warning:

Set "0" in the CONT bit in the WRCSDC0 to WRCSDC7 registers.

[bit2:1] TRP "Transfer Protocol": Serial interface width control bits

Bit Value	Function
00	Serial interface bit width complies with HSSPIn_CSCFG:MBM[1:0] setting [Initial value]
01	Dual-bit mode
10	Quad-bit mode
11	Single-bit mode (serial)

These bits are control bits that set the bit width used for transmitting WRCSDATA[7:0] to the serial interface.

Warning:

If the end of the list is not found in any of WRCSDC0 to WRCSDC7, the set value of the TRP bit in WRCSDC7 is used.

[bit0] DEC "Decode": Decode control bit

Bit Value	Function
0	Transmit WRCSDATA[7:0] as 1-byte data as is. [Initial value]
1	Decode WRCSDATA[2:0] as a control code.

This bit is a control bit that determines whether WRCSDATA[7:0] is data or a control code.

Warning:

After updating any of these HSSPIn_WRCSDC0 to WRCSDC7 registers, read the register values again to confirm that settings are reflected before accessing memory.

1.22.6.30 HS-SPI Module Identification Register (HSSPIn_MID)

This register represents an identification number specific to HS-SPICNT. The identification number means the HS-SPICNT version used within the CPU.

Register Configuration

- HS-SPI module identification register [31:24]

Bit	31	30	29	28	27	26	25	24
Field	MID [31]	MID [30]	MID [29]	MID [28]	MID [27]	MID [26]	MID [25]	MID [24]
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI module identification register [23:16]

Bit	23	22	21	20	19	18	17	16
Field	MID [23]	MID [22]	MID [21]	MID [20]	MID [19]	MID [18]	MID [17]	MID [16]
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	0	0	0

- HS-SPI module identification register [15:8]

Bit	15	14	13	12	11	10	9	8
Field	MID [15]	MID [14]	MID [13]	MID [12]	MID [11]	MID [10]	MID [9]	MID [8]
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	0	0	0	1	1	0

- HS-SPI module identification register [7:0]

Bit	7	6	5	4	3	2	1	0
Field	MID [7]	MID [6]	MID [5]	MID [4]	MID [3]	MID [2]	MID [1]	MID [0]
Attribute	R	R	R	R	R	R	R	R
Initial Value	0	0	1	1	0	0	0	0

Register Functions

[bit31:0] MID[31:0] "Module ID": Module identification information bits

This bit field is read-only and represents a module identification number specific to HS-SPICNT.

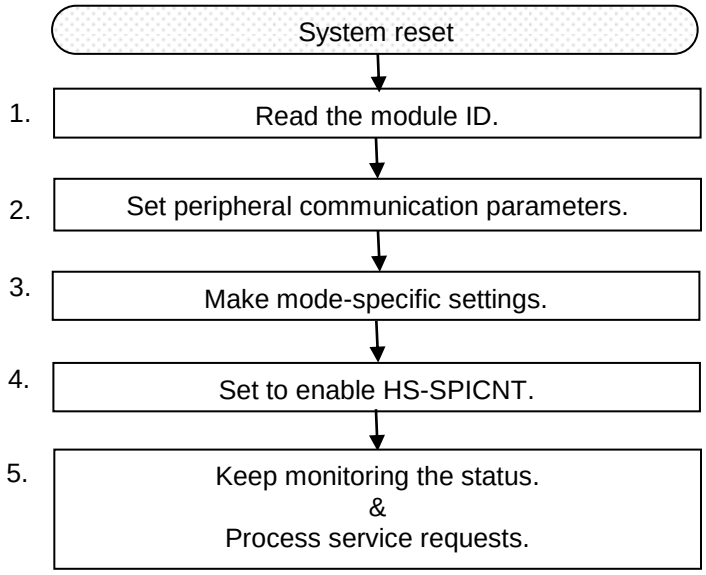
1.22.7 Notes on Using the Hi-Speed SPI Controller

This section is a "Programmer's Guide" that describes all the points you should note about creating a program to control HS-SPICNT. Before creating an HS-SPICNT control program, you need to read the guidelines shown here.

1.22.7.1 General Usage Notes

- Do not make any changes to the specified parameters and control bits for a serial transfer while the transfer is in progress. These changes include selecting a setting in the HSSPIn_PCC0 to PCC3 registers and switching between direct mode and command sequencer mode. Before any such change is done, the current serial transfer must have finished (HSSPIn_TXF:TSSRS is "1" or HSSPIn_RXF:RSSRS is "1") and then HS-SPICNT must be set to disabled (HSSPIn_MCTRL:MEN is "0"). Software can refer to HSSPIn_DMSTATUS:TXACTIVE and RXACTIVE in the same register to confirm that HS-SPICNT has finished all serial transfers.
- To imitate the transfer protocols of some Serial Flash memory devices (Winbond products, etc.) in direct mode, access may have to begin with a single-bit SPI protocol. In this case, you need to change settings in the HSSPIn_DMTRP register during serial transfer. Specifically, the change must be made while the slave select is asserted. If the HSSPIn_DMTRP register must be reset while software is performing the serial transfer, you need to do so after the current ongoing serial transfer has ended. See the "Stop Due to Insufficient Transmission Data or Insufficient Free Space in the RX-FIFO" section.
- However, do not switch from "TX-Only single-bit mode" to "RX-Only single-bit mode" (or vice versa) on the HSSPIn_DMTRP register while the serial transfer is stopped. Instead of stopping the serial transfer to switch the modes, you need to set "TX-and-RX single-bit mode" before starting the transfer. In this case, ignore the received data for not accepting reception operations, or transmit dummy data for not accepting transmission operations.
- In direct mode, software needs to ensure that the FIFOs in HS-SPICNT do not cause an overrun or underrun. If an overrun or underrun occurs, the software is required to flash (discard internal data all at once) the FIFO because the FIFO level has lost its meaning.
- Before starting a serial transfer, you need to use HSSPIn_FIFOCFG:RXFLSH and TXFLSH to clear the FIFOs. This operation prevents the status of the previous finished transfer from affecting the next transfer. In direct mode, the contents of the reception shift register are transferred to the RX-FIFO when the serial transfer ends (if the RX-FIFO has free space).
  - ☐ If the RX-FIFO is already full when the serial transfer ends, data remains in the reception shift register while the serial transfer stops and the serial interface of HS-SPICNT stops. The serial interface of HS-SPICNT resumes operation as soon as the RX-FIFO acquires free space, and it transfers the contents of the reception shift register to the RX-FIFO before releasing the slave select. Even if the reception shift register is not completely full, the same operation occurs.
  - ☐ If the RX-FIFO is already full when the serial transfer ends, data remains in the reception shift register while the serial transfer stops and the serial interface of HS-SPICNT stops. The serial interface of HS-SPICNT resumes operation as soon as the RX-FIFO acquires free space, and it transfers the contents of the reception shift register to the RX-FIFO before releasing the slave select. Even if the reception shift register is not completely full, the same operation occurs.

1.22.7.2 Software Control Procedures of the Hi-Speed SPI Controller



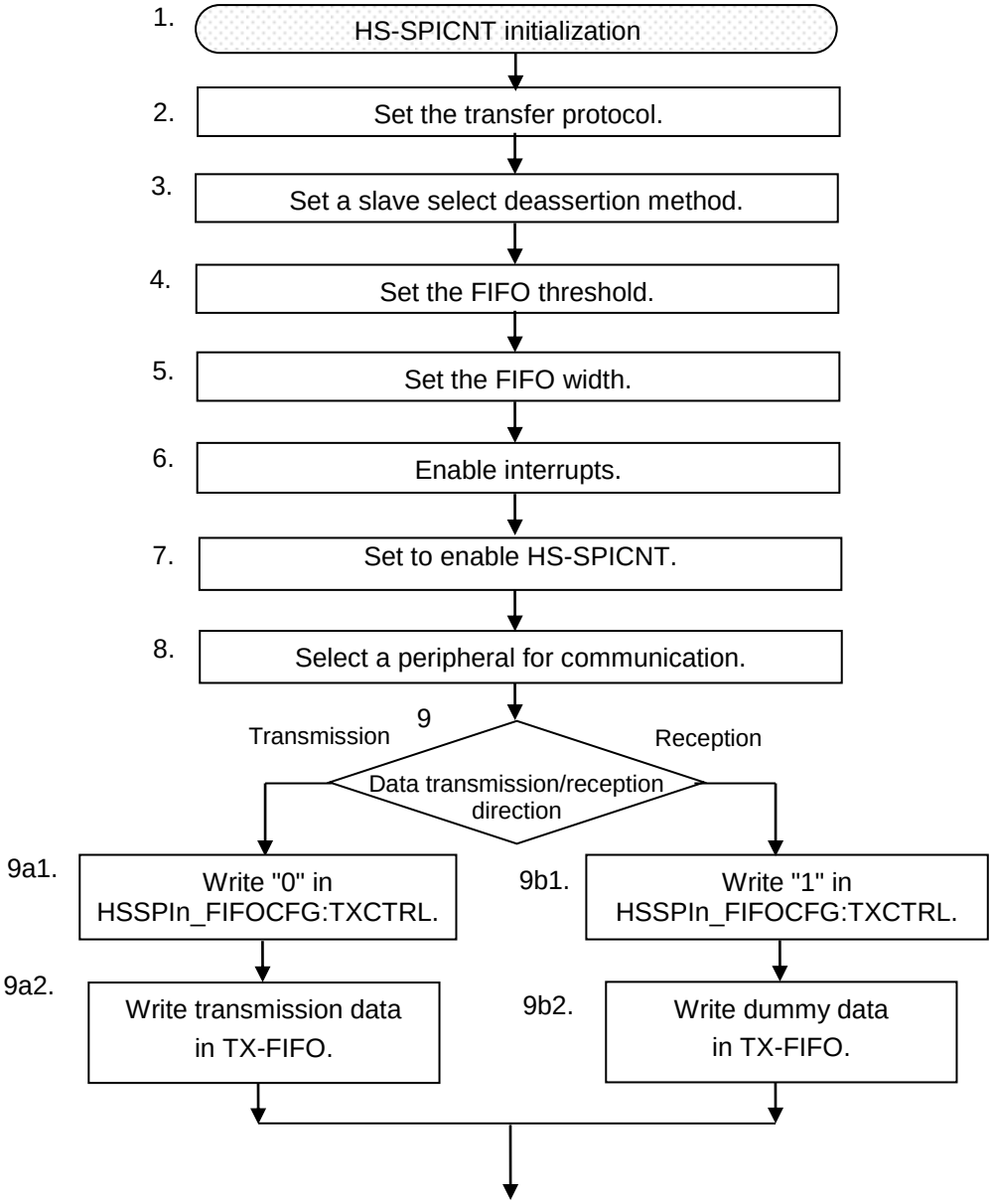
1. After a system reset, the software reads the HSSPIn_MID register of HS-SPICNT to acquire the module ID. The software can identify the HS-SPICNT settings and performance based on this information.
2. Next, the software configures peripheral communication with the serial devices to be connected. HS-SPICNT can connect with up to 4 serial devices. The software uses the HSSPIn_PCC0, HSSPIn_PCC1, HSSPIn_PCC2, and HSSPIn_PCC3 registers to make serial communication settings, including clock polarity, clock phase, transfer frequency (clock division ratio), and slave select polarity. These settings must conform to the specifications of the actually connected serial devices. Also, these settings must not be changed during serial communication. To make any changes, the software must first disable HS-SPICNT and wait for the current serial transfer to finish.
3. HS-SPICNT can be set in either direct mode or command sequencer mode (with the HSSPIn_MCTRL:CSEN setting). The registers for direct mode are HSSPIn_TXF, HSSPIn_TXE, HSSPIn_TXC, HSSPIn_RXF, HSSPIn_RXE, HSSPIn_RXC, HSSPIn_DMCFG, HSSPIn_DMSTATUS, HSSPIn_TXFIFO0 to TXFIFO15, HSSPIn_RXFIFO0 to RXFIFO15, and HSSPIn_FIFOCFG. The registers for command sequencer mode are HSSPIn_CSCFG, HSSPIn_CSITIME, HSSPIn_CSAEXT, HSSPIn_RDCSDC0 to RDCSDC7, and HSSPIn_WRCSDC0 to WRCSDC7.
4. After the necessary settings are made on HS-SPICNT, the controller can be enabled. Here, this means writing "1" in HSSPIn_MCTRL:MEN.
5. Once HS-SPICNT is enabled, its normal operation begins. The software uses various status bits to monitor the status of the controller. The software processes these requests appropriately and maintains normal HS-SPI operation.

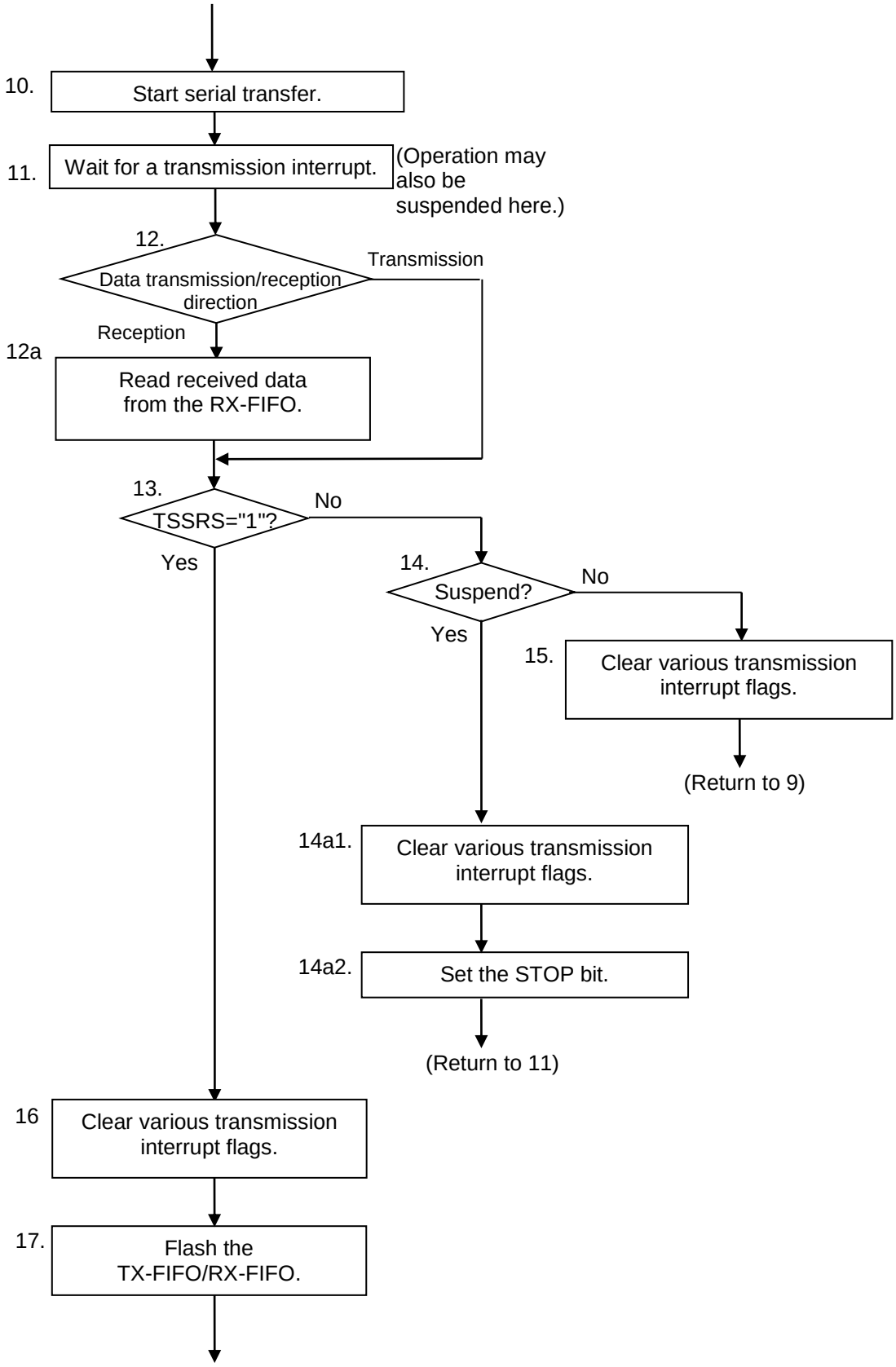
Software Operations for Use in Direct Mode

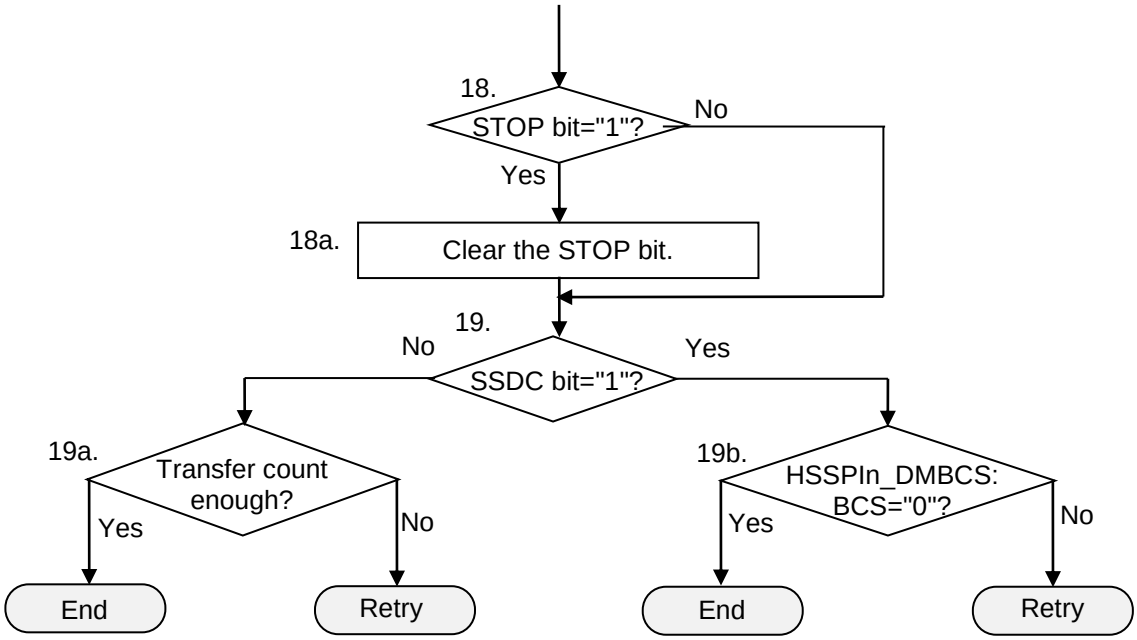
The following figures show typical procedures for using HS-SPICNT in direct mode. This section describes the flow for each of the TX-and-RX, TX-Only, and RX-Only modes.

- TX-and-RX mode

  The transfer direction is set to TX-and-RX mode when HS-SPICNT is in single-bit mode (TRP[1:0] is "00").




10.
Start serial transfer.
11.
Wait for a transmission interrupt.
(Operation may also be suspended here.)
12.
Data transmission/reception direction
Transmission
Reception
12a
Read received data from the RX-FIFO.
13.
TSSRS="1"?
No
Yes
14.
Suspend?
No
Yes
15.
Clear various transmission interrupt flags.
(Return to 9)
14a1.
Clear various transmission interrupt flags.
14a2.
Set the STOP bit.
(Return to 11)
16
Clear various transmission interrupt flags.
17.
Flash the TX-FIFO/RX-FIFO.



1. After a system reset, the software reads the HSSPIn_MID register of HS-SPICNT and configures peripheral communication in the HSSPIn_PCC0 to PCC3 registers. It is important that those settings conform to the specifications of the devices connected on the serial interface. Also, the software confirms that the HSSPIn_MCTRL:CSEN bit is initialized to "0".
2. The software sets the transfer protocol to single-bit mode and the transfer direction to TX-and-RX. These settings are made in HSSPIn_DMTRP:TRP.
3. The software sets HSSPIn_DMCFG:SSDC. This determines the slave select deassertion method. If it is set to byte counter mode, this value must be provided to HSSPIn_DMBCC:BCC as the number of data bytes for serial transfer. If it is set to software flow control, the software needs to set HSSPIn_DMSTOP:STOP for a suspend operation after the intended data is transmitted/received. If the data transfer amount has been determined in advance, the above-described byte counter mode is a possible option.
4. The software sets the FIFO threshold in the HSSPIn_FIFOCFG register. Setting this threshold enables the generation of service requests.
5. The software sets the FIFO bit width in HSSPIn_FIFOCFG:FWIDTH.
6. The software configures service requests. HS-SPICNT handles interrupt requests. It reads data from and writes data to FIFOs as normal operations.

   ☐ Interrupt requests:

   In TX-and-RX mode, the amount of received data is equal to the amount of transmitted data. Therefore, data transfer on the transmission side also controls data transfer on the reception side. Because of this, reception operations also use interrupts with HSSPIn_TXF:TFES (by setting HSSPIn_TXE:TFEE to "1"). Instead of TFES, TFLETS can also be used for flow control (the HSSPIn_TXE:TFLETE setting is "1" in that case).
   To enable a single serial interface access to complete a set of transfer operations including both transmission and reception, all operation cycles occurring on the serial interface must be programmed on the transmission side. This programming means "normal transmission data write" and "dummy transmission data write while expecting a reception operation."

7. Now, the initial settings to run HS-SPICNT in direct mode are complete.

The software sets HSSPIn_MCTRL:MEN to enable transfer operations.

8. The software sets HSSPIn_DMPSEL:PSEL to select the peripheral that transfers data.

9. The procedure branches to either 9a1 or 9a2 below, depending on the data transfer direction.
   Note: Each of the operations described below falls in the serial transfer preparation phase. That is, the actual data transfer has not yet started.

[To transmit data]

9a1) Transmission data write process. All the data written to the TX-FIFO becomes valid transmission data. For this reason, the software writes "0" in HSSPIn_FIFOCFG:TXCTRL. Then, the data written to the TX-FIFO becomes valid data to be transmitted.

9a2) The software writes the data to transmit in any of the HSSPIn_TXFIFO0 to TXFIFO15 registers. The data can be written to any of registers 0 to 15 but must be written in the order of transmission.

[To receive data]

9b1) To receive data, the software needs the TX-FIFO function for driving a reception operation. After that, to perform a dummy transmission operation in order to receive data into the RX-FIFO, the software writes "1" in HSSPIn_FIFOCFG:TXCTRL. Writing "1" causes Hi-Z to be transmitted on the serial interface for the duration of the 1-byte time. The received data for the duration of the 1-byte time becomes valid data that is captured in the RX-FIFO.

9b2) The software writes dummy transmission data in any of the HSSPIn_TXFIFO0 to TXFIFO15 registers. The data can be written to any of registers 0 to 15, but the amount of written data must be the same as that of the data to be received. QSPI_SIO0 output enters the Hi-Z state outside the chip during the transmission of the data. For that period of time, QSPI_SIO1 input is captured as received data. The data undergoes serial/parallel conversion and is transferred to the RX-FIFO.

[To transmit and receive data]

To switch between transmission from the TX-FIFO and reception into the RX-FIFO for each unit of data, HSSPIn_FIFOCFG:TXCTRL must be set appropriately for each data write operation in HSSPIn_TXFIFO0 to TXFIFO15. To set a dummy data area on the serial interface by 4 bits instead of by 1 byte, bit [0] of the data to be written to the HSSPIn_TXFIFO0 to TXFIFO15 registers must be set to "1". In that case, [7:4] of the data written in the HSSPIn_TXFIFO0 to TXFIFO15 registers becomes valid data, and [3:0] is the 4-bit dummy data area.

10. The software sets HSSPIn_DMSTART:START to start a serial transfer, only when starting serial interface communication. After that, the software does not have to operate the bit until SSELm (m=0 to 3) is deasserted. For the subsequent data transfers, data is transmitted sequentially each time that the data is written to the TX-FIFO. The RX-FIFO captures received data at the same time.

Once the serial transfer is started, HS-SPICNT transfers the data in the TX-FIFO to the transmission shift register if HSSPIn_DMTRP:TRP is TX-Only or TX-and-RX. The shift register transmits the serial data by shifting to upper bits or lower bits depending on HSSPIn_PCC0 to PCC3:SDIR. When the reception operation of HS-SPICNT is enabled (with HSSPIn_DMTRP:TRP), HS-SPICNT collects and parallelizes received data on the reception shift register and transfers it to the RX-FIFO.

11. In TX-and-RX mode, data is received according to the amount of transmitted data. Therefore, when a transfer completion interrupt is generated on the transmission side, the data transfer on the reception side also ends. The software waits for an interrupt from HSSPIn_TXF:TFES. Alternatively, it can use TFLETS.

To suspend the data transfer, the software performs the following register operations while waiting for the interrupt. After these operations, the software waits for an interrupt to occur.

11a1) If HSSPIn_DMCFG:SSDC is "1", the software writes "0" to the bit.

11a2) The software writes "1" in HSSPIn_DMSTOP:STOP.

12. The subsequent operations differ depending on whether the current data transfer is transmission or reception.

If it is a reception operation, the software reads data from the RX-FIFO, in the amount indicated by HSSPIn_DMSTATUS:RXLEVEL[4:0].

13. The software determines whether an interrupt has been generated with a serial interface enable signal deasserted. If HSSPIn_TXF:TSSRS is "1", it means the end of the set of transmission/reception operations.

14. The software determines whether to suspend the data transfer. When suspending it, the software performs the following operations 14a1 to 14a2.

14a1) The software clears various interrupts originating from the TX-FIFO/RX-FIFO.

14a2) The software sets the HSSPIn_DMSTOP:STOP bit. If HSSPIn_DMCFG:SSDC is set to "1", the software first clears HSSPIn_DMCFG:SSDC and then performs the HSSPIn_DMSTOP:STOP operation. Next, the software returns to operation 11 to wait for an HSSPIn_TXF:TSSRS interrupt.

HSSPIn_DMSTOP:STOP is not cleared automatically, so the software needs to have an operation to clear it later in this flow.

15. This position in the flow indicates that the set of data transfers has not completed within 1 TX-FIFO operation. In this case, to continue the subsequent data transfer, the software needs to write the data to the TX-FIFO. The software clears interrupts originating from the TX-FIFO and returns to branch point 9 in this flow.

16. The set of data transfer operations has completed (reached the end or been suspended). The TX-FIFO and RX-FIFO operations have already finished. Therefore, the software operates the HSSPIn_TXC and HSSPIn_RXC registers to clear the interrupt factors of HSSPIn_TXF and HSSPIn_RXF.

17. The software sets HSSPIn_FIFOCFG:TXFLSH and RXFLSH to initialize the TX-FIFO/RX-FIFO internal state. This operation initializes the pointer in the FIFO and clears some FIFO interrupt events accompanying it. However, the HSSPIn_TXF/HSSPIn_RXF register values are not cleared until the software operates the HSSPIn_TXC/HSSPIn_RXC register.

18. If HSSPIn_DMSTOP:STOP is "1", the software clears it.

18a) The software clears HSSPIn_DMSTOP:STOP.

19. The software branches to either 19a or 19b depending on the HSSPIn_DMCFG:SSDC value.

19a) If SSDC is "0", it represents a mode where the software controls SSELm (m=0 to 3) deassertion. In this case, the software is responsible for controlling the data transfer amount.

19b) If SSDC is "1", it represents a mode where SSELm (m=0 to 3) is automatically deasserted. In this case, the HSSPIn_DMBCC and HSSPIn_DMBCS registers control the data transfer amount.

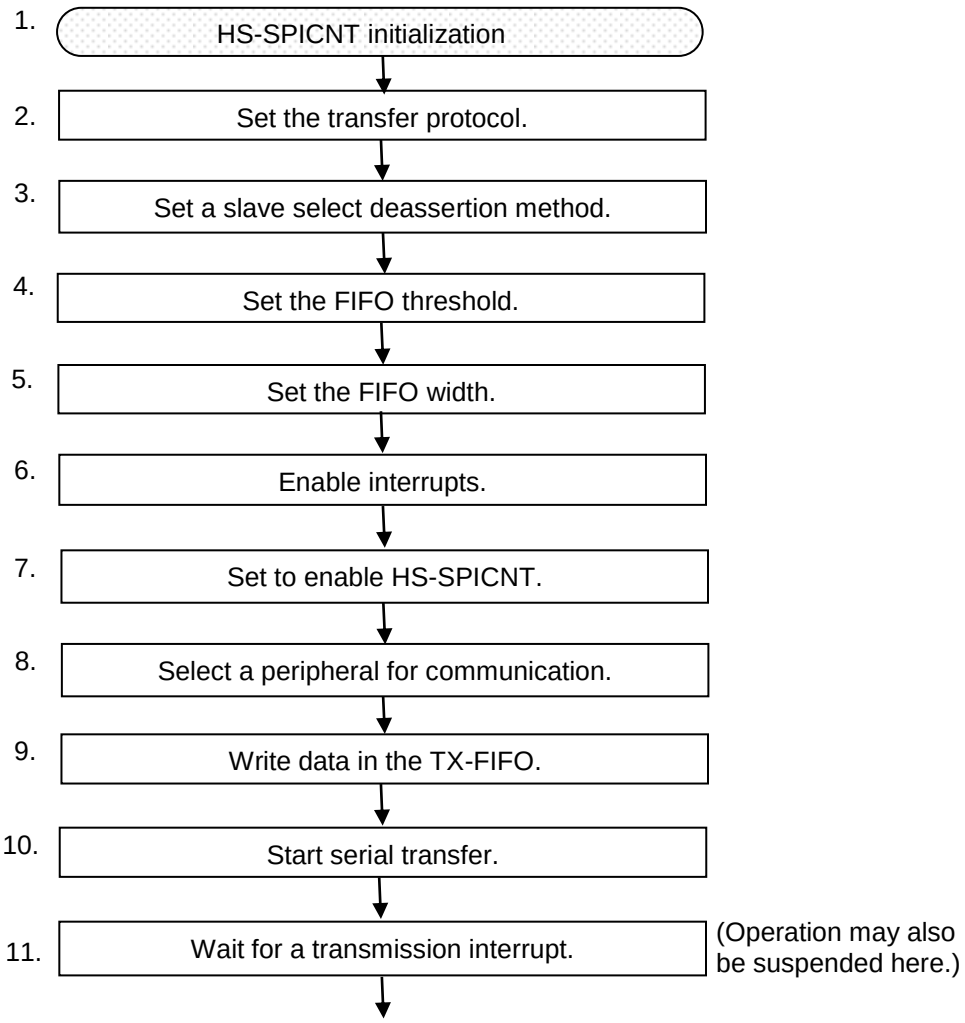
If the amount of data actually transferred matches the programmed amount, a normal end occurs. If they do not match, the set of transfer operations has not completed successfully, and this flow must be processed again from the beginning.

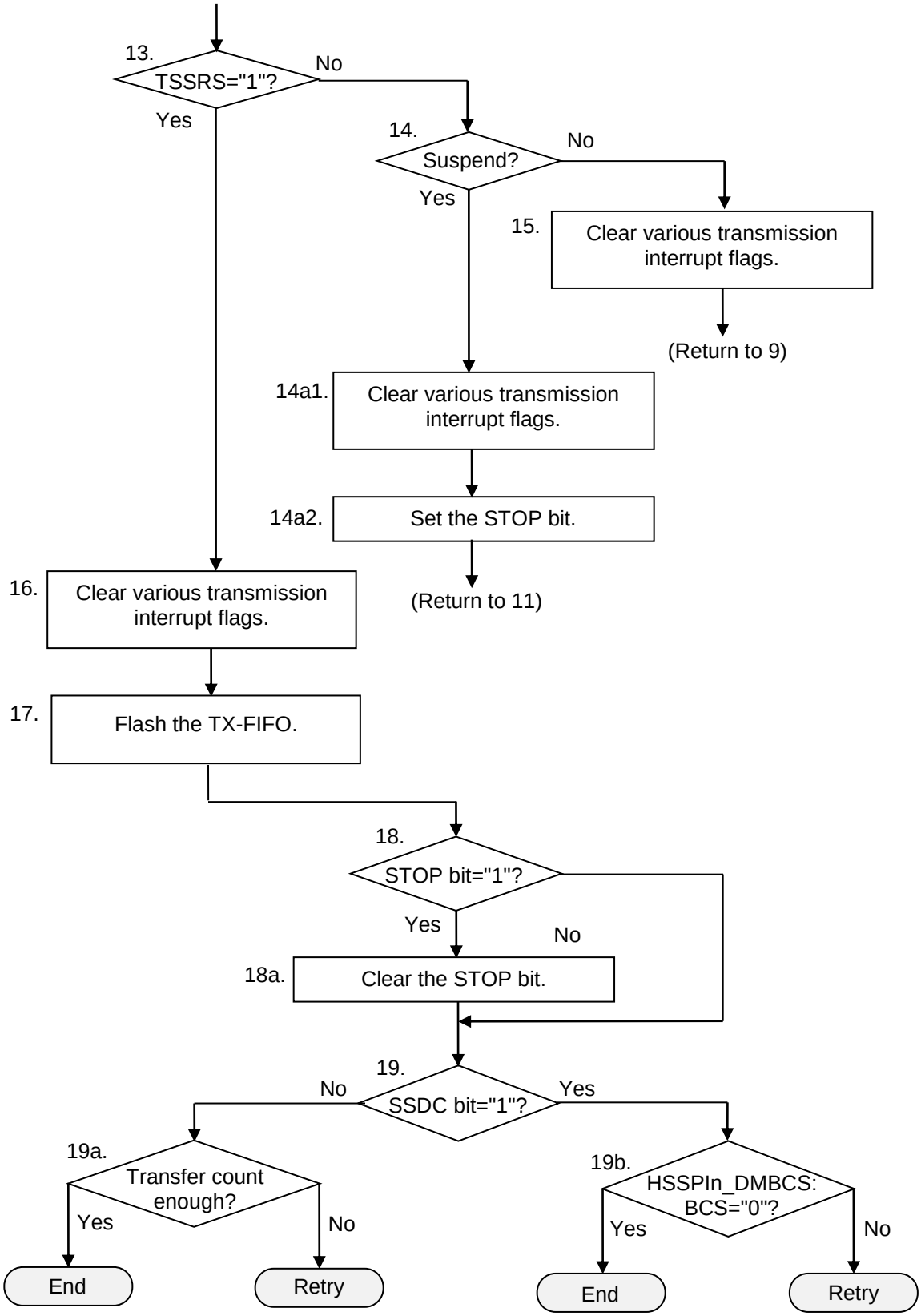
The following flows apply to each of the TX-and-RX, TX-Only, and RX-Only modes.

To reset serial transfer-related parameters, the software needs to stop the current transfer first and then disable HSSPICNT. These settings must be made after all currently running operations are stopped. The stopped status of transfer operations can be confirmed with HSSPIn_DMSTATUS:TXACTIVE/RXACTIVE.

- TX-Only mode

  The transfer direction can be set to TX-Only mode when HS-SPICNT is in any of single-bit mode (TRP[1:0] is "00"), dual-bit mode (TRP[1:0] is "01"), or quad-bit mode (TRP[1:0] is "10").




13.
TSSRS="1"?
No
Yes
14.
Suspend?
No
Yes
15.
Clear various transmission interrupt flags.
(Return to 9)
14a1.
Clear various transmission interrupt flags.
14a2.
Set the STOP bit.
(Return to 11)
16.
Clear various transmission interrupt flags.
17.
Flash the TX-FIFO.
18.
STOP bit="1"?
Yes
No
18a.
Clear the STOP bit.
19.
SSDC bit="1"?
No
Yes
19a.
Transfer count enough?
Yes
No
End
Retry
19b.
HSSPIn_DMBCS: BCS="0"?
Yes
No
End
Retry

1. Same as for TX-and-RX mode
2. The software sets the transfer protocol to any of single-bit mode, dual-bit mode, or quad-bit mode, and the transfer direction to TX-Only. HSSPIn_DMTRP:TRP is used for these settings.
3. Same as for TX-and-RX mode
4. Same as for TX-and-RX mode
5. Same as for TX-and-RX mode
6. The software configures service requests. HS-SPICNT handles interrupt requests. The software reads data from and writes data to the internal FIFO as normal operations.

   ☐ Cases using interrupt requests:

   TX-Only mode uses interrupts with HSSPIn_TXF:TFES (by setting HSSPIn_TXE:TFEE to "1"). Instead of the TFES bit, the TFLETS bit can also be used for flow control (the HSSPIn_TXE:TFLETE setting is "1" in that case).
7. Same as for TX-and-RX mode
8. Same as for TX-and-RX mode
9. Transmission data write process

   The software writes the data to transmit in any of the HSSPIn_TXFIFO0 to TXFIFO15 registers. The data can be written to any of registers 0 to 15 but must be written in the order of transmission.

   Each of the operations described below falls in the serial transfer preparation phase. That is, the actual data transfer has not yet started.
10. The software sets HSSPIn_DMSTART:START to start a serial transfer, only when starting serial interface communication. After that, the software does not need to operate the bit until SSELm (m=0 to 3) is deasserted. For the subsequent data transfers, data is transmitted sequentially each time that the data is written to the TX-FIFO.

    Once the serial transfer is started, HS-SPICNT transfers the data in the TX-FIFO to the transmission shift register. The shift register transmits the serial data by shifting to upper bits or lower bits depending on HSSPIn_PCC0 to PCC3:SDIR.
11. The software waits for an interrupt with the HSSPIn_TXF:TFES bit. Alternatively, it can use TFLETS.

    To suspend the data transfer, the software performs the following register operations while waiting for the interrupt. After these operations, the software waits for an interrupt to occur.

    11a1) If HSSPIn_DMCFG:SSDC is "1", the software writes "0" to the bit.

    11a2) The software writes "1" in HSSPIn_DMSTOP:STOP.
12. N/A
13. The software determines whether an interrupt has been generated with a serial interface enable signal deasserted. If HSSPIn_TXF:TSSRS is "1", it means the set of transmission operations has ended.

    14. The software determines whether to suspend the data transfer. To finish, the software performs the following operations 14a1 to 14a2.

    14a1) The software clears various interrupts originating from the TX-FIFO.

    14a2) The software sets HSSPIn_DMSTOP:STOP. If HSSPIn_DMCFG:SSDC is set to "1", the software first clears SSDC and then performs the STOP operation. Then, the software returns to operation 11 to wait for an HSSPIn_TXF:TSSRS interrupt.

    STOP is not cleared automatically, so the software needs to have an operation to clear it later in this flow.

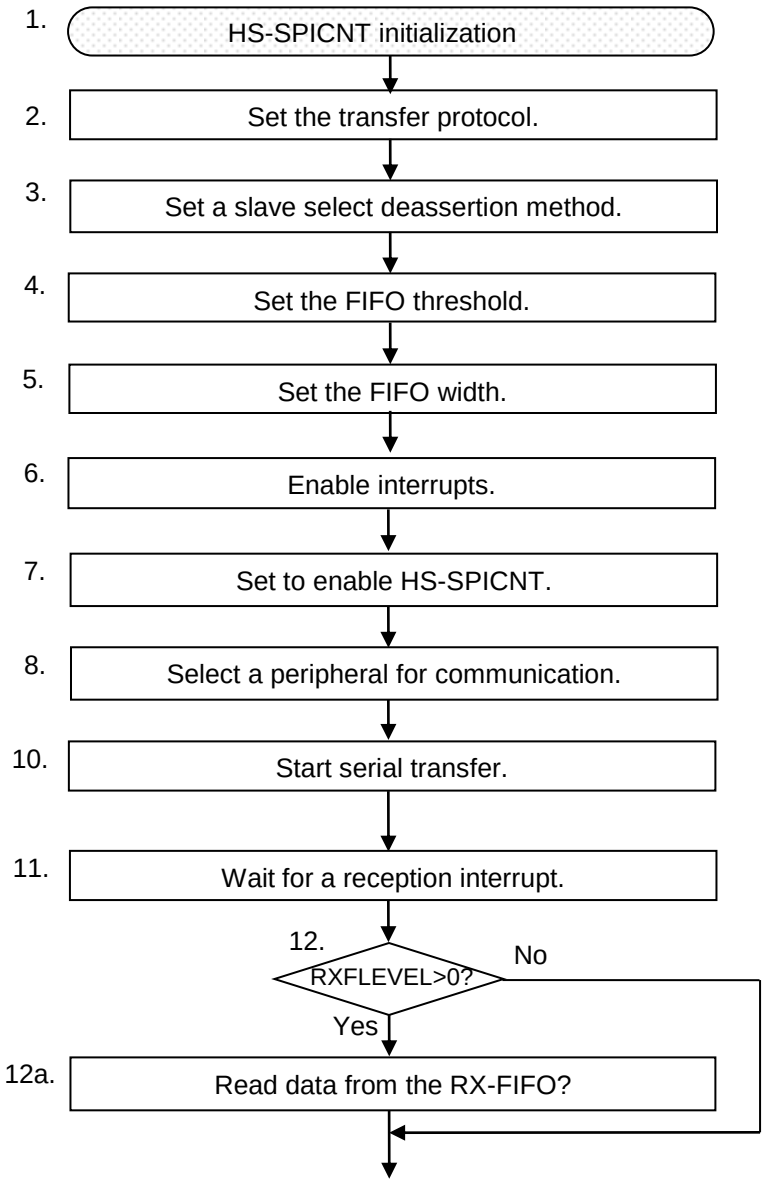
    15. This position in the flow indicates that the set of data transfers has not completed within 1 TX-FIFO operation. In this case, to continue the subsequent data transfer, the software needs to write the data to the TX-FIFO. The software clears interrupts originating from the TX-FIFO and returns to operation 9 in this flow.

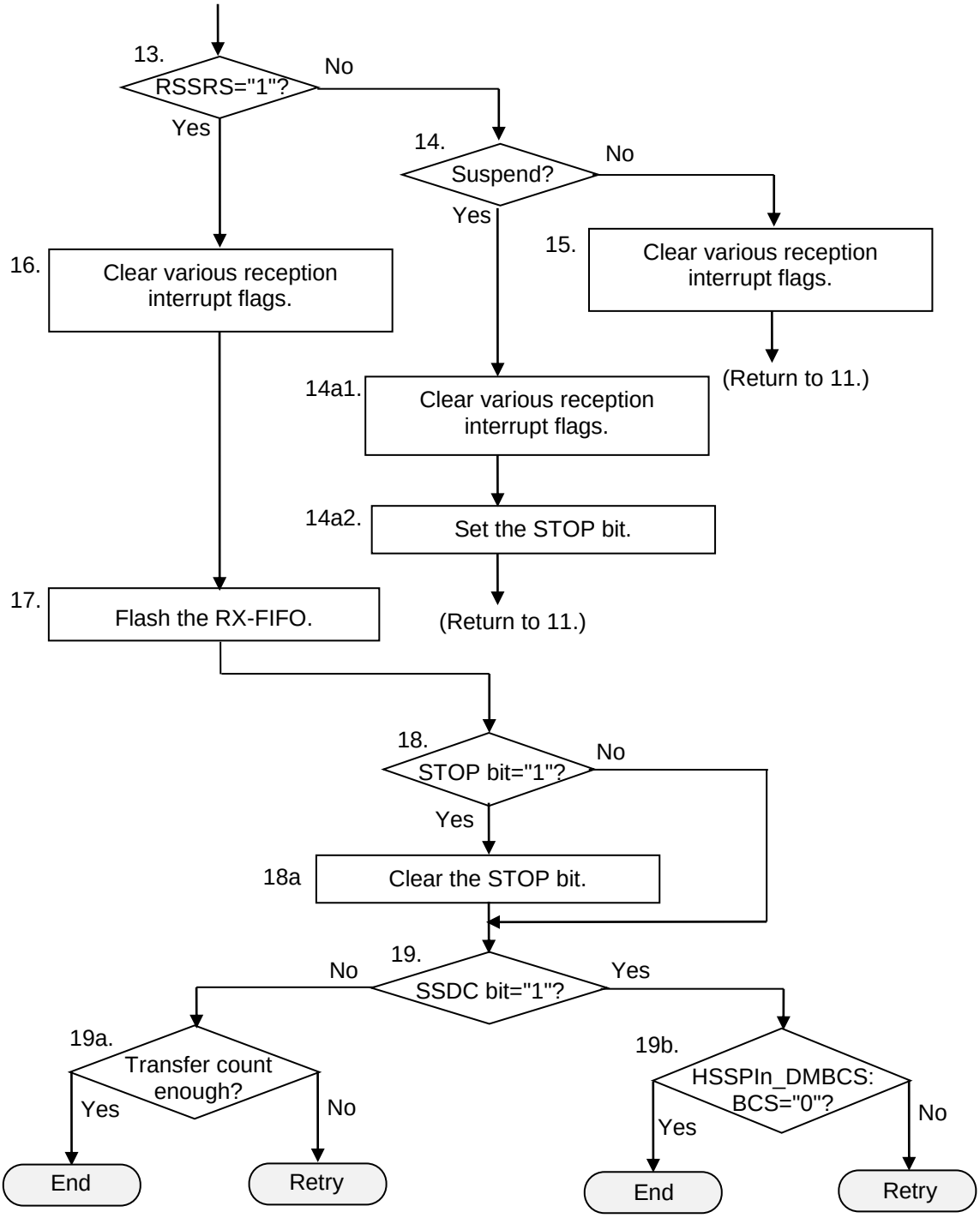
    16. The set of data transfer operations has completed or been suspended. The TX-FIFO operation has already finished. Therefore, the software operates the HSSPIn_TXC register to clear the interrupt factors of HSSPIn_TXF.

17. The software sets HSSPIn_FIFOCFG:TXFLSH to initialize the TX-FIFO internal state. This operation initializes the pointer in the FIFO and clears some FIFO interrupt events accompanying it. However, the HSSPIn_TXF register values are not cleared until the software operates the HSSPIn_TXC register.
18. Same as for TX-and-RX mode
19. Same as for TX-and-RX mode

- RX-Only mode

The transfer direction can be set to RX-Only mode when HS-SPICNT is in any of single-bit mode (TRP[1:0] is "00"), dual-bit mode (TRP[1:0] is "01"), or quad-bit mode (TRP[1:0] is "10").




13.
RSSRS="1"?
No
Yes
14.
Suspend?
No
Yes
15.
Clear various reception interrupt flags.
(Return to 11.)
16.
Clear various reception interrupt flags.
14a1.
Clear various reception interrupt flags.
14a2.
Set the STOP bit.
(Return to 11.)
17.
Flash the RX-FIFO.
18.
STOP bit="1"?
No
Yes
18a
Clear the STOP bit.
19.
SSDC bit="1"?
No
Yes
19a.
Transfer count enough?
Yes
No
End
Retry
19b.
HSSPIn_DMBCS: BCS="0"?
Yes
No
End
Retry

1. Same as for TX-and-RX mode
2. The software sets the transfer protocol to single-bit mode and the transfer direction to RX-Only. These settings are made in HSSPIn_DMTRP:TRP.
3. Same as for TX-and-RX mode
4. Same as for TX-and-RX mode
5. Same as for TX-and-RX mode
6. The software configures service requests. HS-SPICNT handles interrupt requests. It reads data from and writes data to the FIFO as normal operations.

   ☐ Cases using interrupt requests:

   RX-Only mode uses interrupts with HSSPIn_RXF:RFFS (by setting HSSPIn_RXE:RFFE to "1"). Instead of RFFS, RFLETS can also be used for flow control (the HSSPIn_RXE:RFLETE setting is "1" in that case).

   To enable a single serial interface access to complete a set of transfer operations including both transmission and reception, all operation cycles occurring on the serial interface must be programmed on the transmission side. This programming means "normal transmission data write" and "dummy transmission data write while expecting a reception operation."
7. Same as for TX-and-RX mode
8. Same as for TX-and-RX mode
9. N/A
10. The software sets HSSPIn_DMSTART:START to start a serial transfer. After that, the software does not need to operate the bit until SSELm (m=0 to 3) is deasserted. The RX-FIFO captures received data sequentially.

    Once the serial transfer is started, HS-SPICNT collects and parallelizes received data on the reception shift register and transfers it to the RX-FIFO. The shift register captures received data by shifting to upper bits or lower bits depending on HSSPIn_PCC0 to PCC3:SDIR. When the data in this shift register reaches the bit width specified by FWIDTH, it is transferred to the RX-FIFO as 1-word data.
11. The software waits for an interrupt from HSSPIn_RXF:RFFS. Alternatively, it can use RFMTS.

    To suspend the data transfer, the software performs the following register operations while waiting for the interrupt. After these operations, the software waits for an interrupt to occur.

    11a1) If HSSPIn_DMCFG:SSDC is "1", the software writes "0" to the bit.

    11a2) The software writes "1" in HSSPIn_DMSTOP:STOP.
12. The software reads data from the RX-FIFO, in the amount indicated by HSSPIn_DMSTATUS:RXLEVEL[4:0].
13. The software determines whether an interrupt has been generated with a serial interface enable signal deasserted. If HSSPIn_RXF:RSSRS is "1", it means the end of the set of transmission/reception operations.
14. The software determines whether to suspend the data transfer. To finish, the software performs the following operations 14a1 to 14a2.
    14a1) The software clears various interrupts originating from the RX-FIFO.

    14a2) The software sets HSSPIn_DMSTOP:STOP. If HSSPIn_DMCFG:SSDC is set to "1", the software first clears SSDC and then performs the STOP operation. Next, the software returns to operation 11 to wait for an HSSPIn_RXF:RSSRS interrupt.

    STOP is not cleared automatically, so the software needs to have an operation to clear it later in this flow.
15. This position in the flow indicates that the set of data transfers has not completed within 1 RX-FIFO operation. In this case, to continue the subsequent data transfer, the RX-FIFO must receive the subsequent data. The software clears interrupts originating from the RX-FIFO and returns to operation 11 in this flow.

16. The set of data transfer operations has completed or been suspended. The RX-FIFO operation has already finished. Therefore, the software operates the HSSPIn_RXC register to clear the interrupt factors of HSSPIn_RXF.
17. The software sets HSSPIn_FIFOCFG:RXFLSH to initialize the RX-FIFO. This operation initializes the pointer in the FIFO and clears some FIFO interrupt events accompanying it. However, the HSSPIn_RXF register values are not cleared until the software operates the HSSPIn_ RXC register.
18. Same as for TX-and-RX mode
19. Same as for TX-and-RX mode

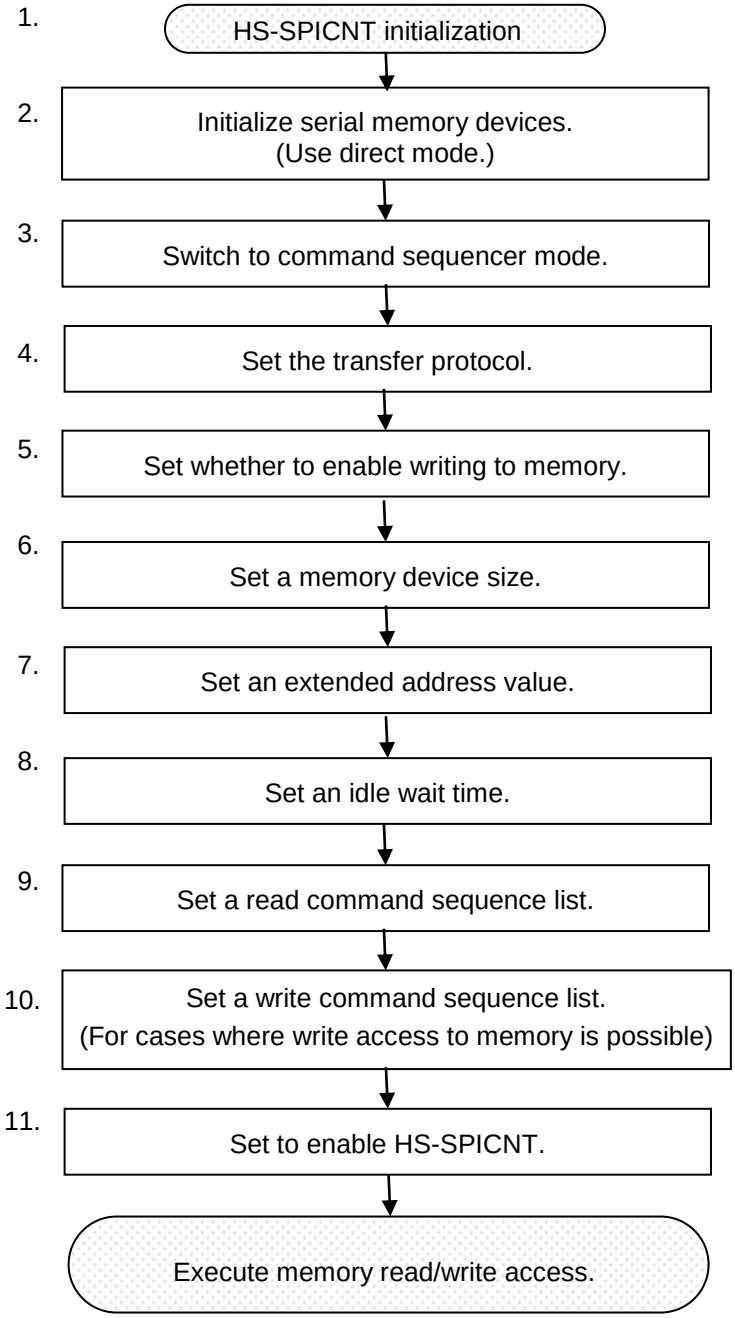
1.22.7.3 Software Operations for Use in Command Sequencer Mode

To use serial memory in command sequencer mode, observe the following notes.

Usage Rules and Notes

- In command sequencer mode, all serial memory devices connected to HS-SPICNT must be of the same type. Avoid a mix of different types.
- To determine HSSPIn_CSCFG:MSEL for memory devices of the same type but different memory sizes, you need to consider the maximum size of the memory devices. In this case, keep the following points in mind. The number of bytes that are read from the final memory address (physical address) is set in a command sequence list (on a set of the HSSPIn_RDCSDC0 to RDCSDC7 registers and HSSPIn_WRCSDC0 to WRCSDC7 registers) and passed from HS-SPICNT to serial memory devices. Memory devices that have 32-bit addresses cannot be mixed with ones that have 24-bit addresses. This is because 24-bit address devices have a 3-cycle address phase while 32-bit address devices have a different one.
- If a memory device supports only 21-bit addresses and HS-SPICNT is configured to use 24-bit addresses, the upper bit range [23:21] is filled with zeros. The HSSPIn_CSAEXT register is used for this zero value. If the HSSPIn_CSAEXT register is not zero, the address value passed to the serial memory is wrapped. This may result in unexpected behavior.
- Serial SRAM devices enable burst transfer by setting the appropriate mode. However, command sequencer mode expects SRAM devices to always be in burst mode. Before enabling command sequencer mode of HS-SPICNT, the software needs to set serial SRAM devices to burst mode. To set this mode, use direct mode.
- Command sequencer mode does not allow switching of the SPI protocol to single-, dual-, or quad-bit once a transfer has started. For this reason, some newer Serial Flash devices, such as Winbond memory devices, can be set to "continuous read mode" and then used in command sequencer mode. In this case, set the memory devices to "continuous read mode" in direct mode, and then switch to command sequencer mode.

Software Operation Flowchart



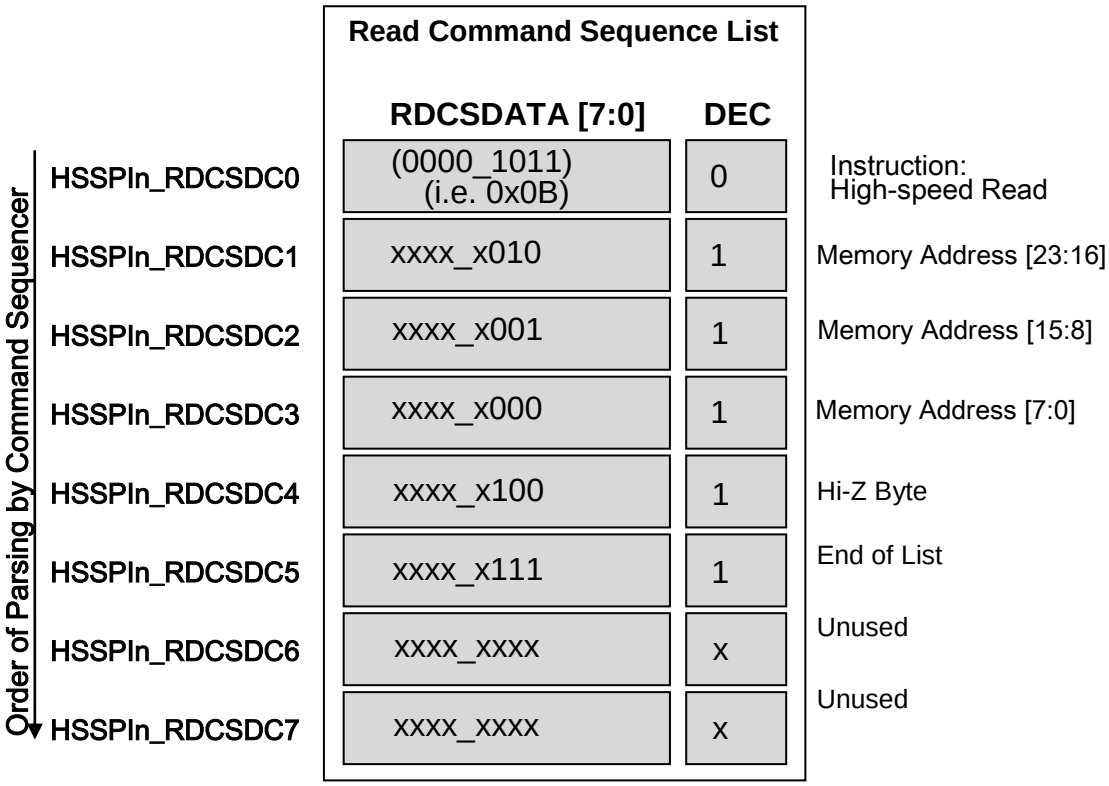
1. After a system reset, the software needs to initialize HS-SPICNT, configure the HSSPIn_PCC0 to PCC3 registers, and configure related peripheral communications. It is important that those settings conform to the specifications of the devices actually connected. To allocate serial memory devices on the system space in command sequencer mode, all of the serial memory devices used must be of the same type. Therefore, all the HSSPIn_PCC0 to PCC3 registers have the same settings.
2. Next, the software initializes the memory devices to be allocated to the system space. The contents of the initialization are specific to each device. For some devices, the initialization may involve configuring the device control/status registers. For example, when using a serial memory device that supports quad-bit SPI, you can use it in the quad-bit mode that provides the best performance. For serial interface protocol specifications, see the data sheet of the device. The software uses the direct mode of HS-SPICNT to perform an initialization operation for a serial memory device.
3. After the serial devices are initialized, the software resets HS-SPICNT to enter command sequencer mode. To switch the mode, the software first stops currently running serial transfers, if any. Then, it disables HS-SPICNT. After confirmation with HSSPIn_DMSTATUS:TXACTIVE/RXACTIVE that HS-SPICNT has finished all serial transfers, the software can be set to command sequencer mode.
4. Next, the software sets the transfer protocol. HS-SPICNT is set to any of the SPI protocols, single-, dual-, or quad-bit, depending on the HSSPIn_CSCFG:MBM bit value.
5. Write access must also be considered when a serial SRAM device is connected. In this case, HSSPIn_CSCFG:SRAM is used. Also, set the same SRAM bit as needed when a Serial Flash device is connected.
6. The software sets HSSPIn_CSCFG:MSEL to specify the size of allocation on the system for each serial memory device. For details, see "1.22.5.2 Slave Selection".
7. If a serial memory address allocated on the system space is a value virtually extended to a 16-GB space, the value must be set in the HSSPIn_CSAEXT register.
8. Setting HSSPIn_CSITIME:ITIME can improve response performance when the following conditions are satisfied.
    3. There are multiple accesses to external serial memory allocated on the system space.
    4. The address consists of a continuous address range.
9. The software writes a set of commands and parameters used for a read command sequence in the HSSPIn_RDCSDC0 to RDCSDC7 registers. The commands and parameters must be set according to the specifications of the memory device used.
10. If writing to memory is also enabled, the software sets HSSPIn_CSCFG:SRAM and writes a write command sequence in the HSSPIn_WRCSDC0 to WRCSDC7 registers. Similar to a read command sequence, the commands and parameters must be set according to the specifications of the memory device used.
11. Once the above operations are completed, the preparations for HS-SPICNT to access memory devices are complete. Then, the software sets HSSPIn_MCTRL:MEN to enable HS-SPICNT.

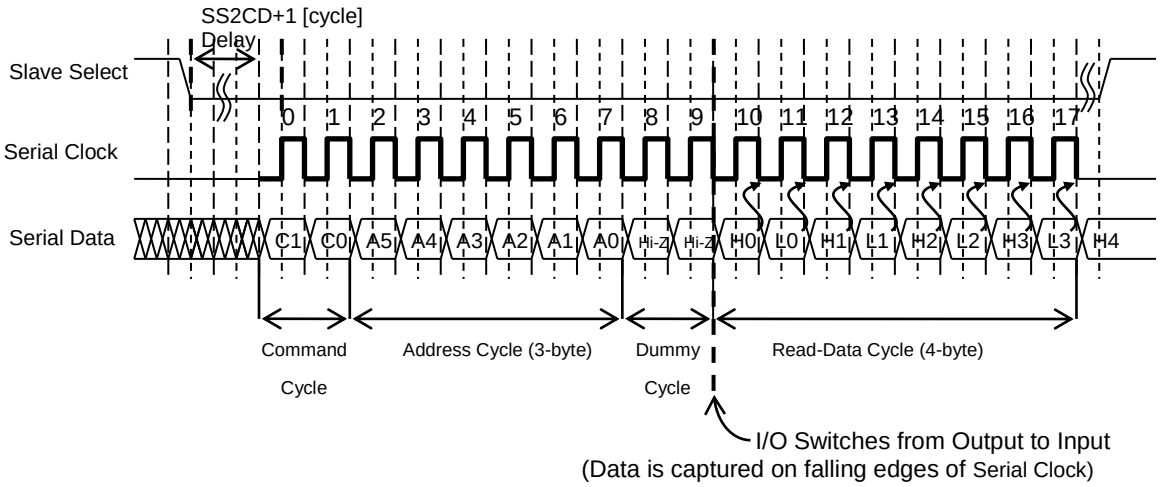
Diagrams of Command Sequencer Operations

The following figure shows an example illustrating how the command sequencer issues a memory read sequence. For example, suppose a read command sequence is programmed as a list in the HSSPIn_RDCSDC0 to RDCSDC5 registers. Then, the command sequencer would interpret the list starting with HSSPIn_RDCSDC0 and execute the command as described in the "1.22.5 Command Sequencer Mode" section.

In addition to the figure, an operation timing diagram for this read command sequence is provided. The clocking mode is "Mode 0" (see Table 1-45).

Illustration of a read command sequence (clocking mode "Mode 0")





Also, you can use the TRP control bit in the list to control the (single, dual, or quad) bit width of the serial interface for each byte to transmit.

To do so, add the setting of the TRP[1:0] bit value to the above register setting example.

HSSPIn_RDCSDC0:TRP=3 (single): Instruction High-speed Read

HSSPIn_RDCSDC1:TRP=2 (quad): Memory Address [23:16]

HSSPIn_RDCSDC2:TRP=2 (quad): Memory Address [15:8]

HSSPIn_RDCSDC3:TRP=2 (quad): Memory Address [7:0]

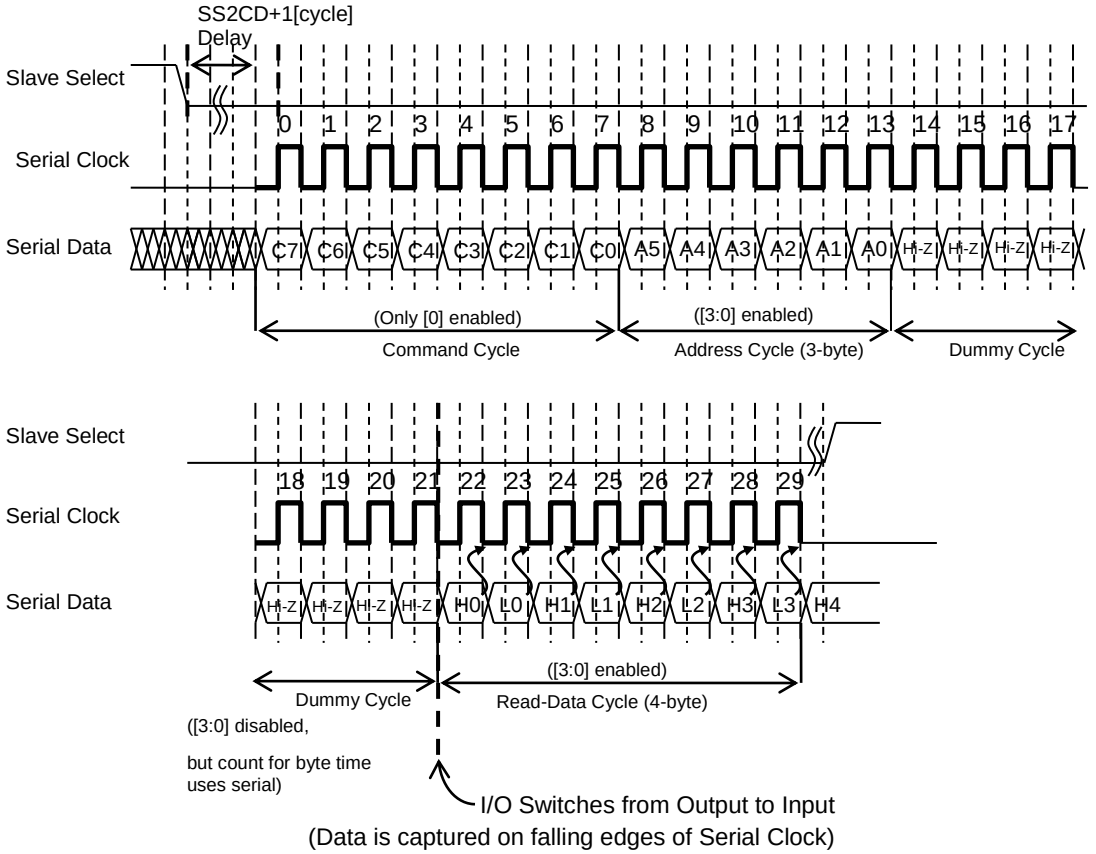
HSSPIn_RDCSDC4:TRP=3 (single): Hi-Z Byte (Hi-Z for 1-byte data)

HSSPIn_RDCSDC5:TRP=0 (depends on MBM[1:0]): End of List (bit width control disabled)

HSSPIn_RDCSDC6:TRP=0 (depends on MBM[1:0]): Unused (bit width control disabled)

HSSPIn_RDCSDC7:TRP=0 (depends on MBM[1:0]): Unused (bit width control disabled)

The operation with these settings is as follows.


SS2CD+1[cycle]
Delay
Slave Select
Serial Clock
Serial Data
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17
C7 C6 C5 C4 C3 C2 C1 C0 A5 A4 A3 A2 A1 A0 Hi-Z Hi-Z Hi-Z Hi-Z
(Only [0] enabled)
Command Cycle
([3:0] enabled)
Address Cycle (3-byte)
Dummy Cycle
Slave Select
Serial Clock
Serial Data
18 19 20 21 22 23 24 25 26 27 28 29
Hi-Z Hi-Z Hi-Z Hi-Z H0 L0 H1 L1 H2 L2 H3 L3 H4
Dummy Cycle
([3:0] disabled,
but count for byte time
uses serial)
([3:0] enabled)
Read-Data Cycle (4-byte)
I/O Switches from Output to Input
(Data is captured on falling edges of Serial Clock)

Revision History

Document Revision History

Document Title: CY91590 Series FR81S Family GDC Manual Document Number: 002-05580				
Revision	**Issue Date**	**ECN No.**	**Origin of Change**	**Description of Change**
**	07/05/2013	-	NNAS	Initial release
*A	08/26/2016	5362010	NNAS	Spansion document “MN705-00022-1v0-E” migrated to Cypress template.
*B	05/29/2019	6578065	TORS	Changed to new Cypress logo. Changed product name. MB91590 -> CY91590