PanaXSeries

The One to Watch for Constant Innovation-Making the Future Come Alive

MICROCOMPUTER MN101D00

MN101D02D/02E/02F/ 02G/02H

LSI User's Manual

Pub.No.21502-020E

Panasonic

Request for your special attention and precautions in using the technical information and semiconductors described in this book

(1) An export permit needs to be obtained from the competent authorities of the Japanese Government if any of the products or technologies described in this book and controlled under the "Foreign Exchange and Foreign Trade Control Law" is to be exported or taken out of Japan.

(2) The contents of this book are subject to change without notice in matters of improved function. When finalizing your design,therefore,ask for the most up-to-date version in advance in order to check for any changes.

(3) We are not liable for any damage arising out of the use of the contents of this book, or for any infringement of patents or any other rights owned by a third party.

(4) No part of this book may be reprinted or reproduced by any means without written permission from our company.

(5) This book deals with standard specifications. Ask for the latest individual Product Standards or Specifications in advance for more detailed information required for your design,purchasing and applications.

If you have any inquiries or questions about this book or our semiconductors, please contact one of our sales offices listed at the back of this book or Matsushita Electronics Corporation's Sales Department.

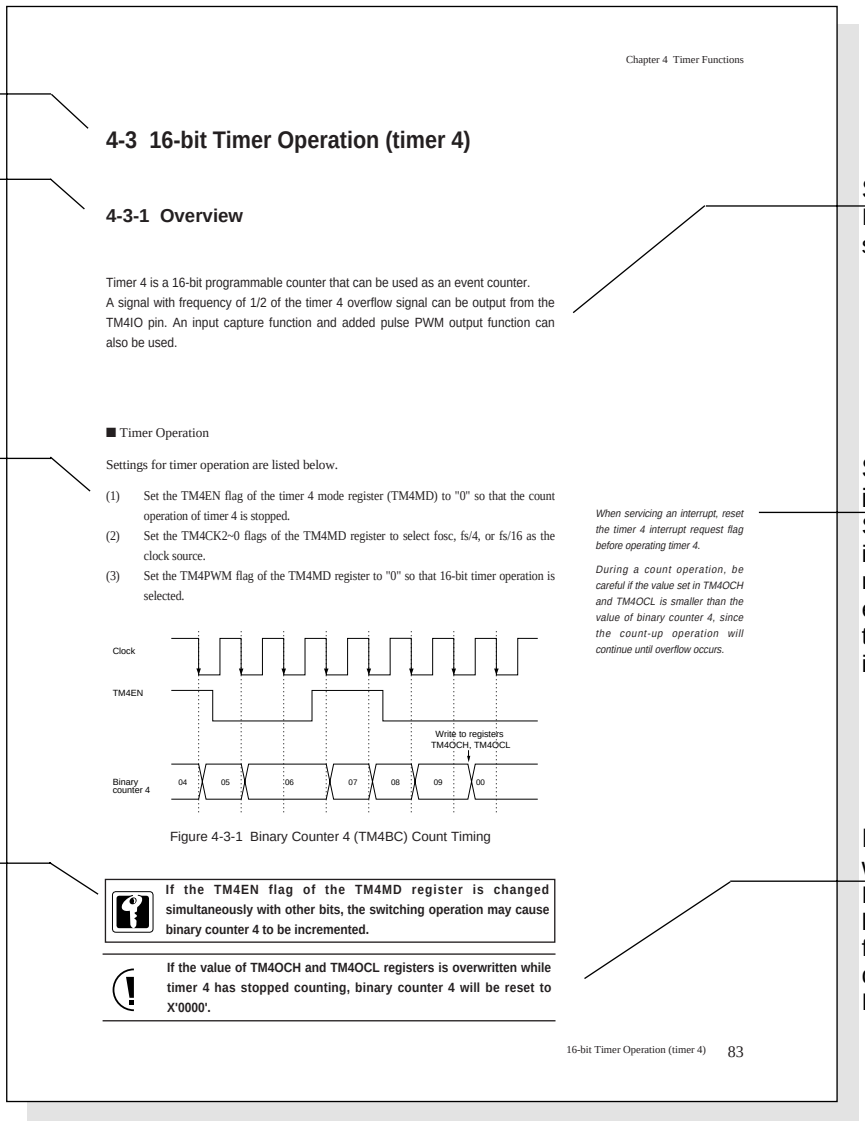
How to Read This Manual

■ Organization

In this LSI manual, the MN101D02X functions are presented in the following order: overview, CPU basic functions, port functions, timer functions, serial functions, and other peripheral hardware functions.

■ Manual Configuration

Each section of this manual consists of a title, summary, main text, supplemental information, precautions and warnings. The layout and definition of each section are shown below.



Subtitle

Sub-subtitle
The smallest block in this manual.

Main text

Key information
Important information from the text.

Summary
Introduction to the section.

Supplementary information
Supplementary information for the main text. An explanation of terminology is also included.

Precautions and warnings
Precautions are listed in case of lost functionality or damage.
Be sure to read.

■ Finding Desired Information

This manual provides four methods for finding desired information quickly and easily.

(1) Consult the index at the front of the manual to locate the beginning of each section.
(2) Consult the table of contents at the front of the manual to locate desired titles.
(3) Consult the list of figures at the front of the manual to locate illustrations and charts by title name.
(4) Chapter names are located at the top outer corner of each page, and section titles are located at the bottom outer corner of each page.

■ Related Manuals

The following manuals are also available from Panasonic as part of the MN101D00 series.

MN101D00 Series LSI Manual
<Device Hardware Description>
MN101D00 Series Command Manual
<Command Descriptions>
MN101D00 Series Cross Assembler User's Manual
<Assembler Syntax and Entry Methods>
MN101D00 Series C Compiler User's Manual Operation
<C Compiler Installation, Startup, Option Descriptions>
MN101D00 Series C Compiler User's Manual Language
<C Language Syntax Description>
MN101D00 Series C Compiler User's Manual Library
<C Compiler Standard Library Description>
MN101D00 Series C Source Code Debugger User's Manual
<C Source Code Debugger Usage Methods>
MN101D00 Series PanaX Series Installation Manual
<Installation of C Compiler, Cross Assembler, C Source Code Debugger; In-circuit Emulator>

■ Where to Send Inquires

Please send any inquires or questions concerning the contents of this manual to the Panasonic semiconductor design center closest to you. A list of addresses is provided at the end of this manual for your convenience.

Contents

Chapter 1 Overview

Chapter 2 Description of Internal Functions

Chapter 3 Port Functions

Chapter 4 Interrupt Functions

Chapter 5 Timer Functions

Chapter 6 Serial Functions

Chapter 7 A/D Converter and D/A Converter

Chapter 8 Synchronous Output Functions

Chapter 9 Servo Functions

Chapter 10 Standby, Clock Operation/Halt, and Reset Functions

Chapter 11 OSD and XDS Functions

Chapter 12 Appendix

Contents

0

CONTENTS

Chapter 1 Overview

1-1 Product Overview 2
1-1-1 Overview 2
1-1-2 Features 2
1-1-3 Product Line 2
1-2 Hardware Functions 3
1-3 Pin Functions 12
1-3-1 Pin Connection Diagram 12
1-3-2 Pin Function Summary 13

Chapter 2 Description of Internal Functions

2-1 Memory Map 18
2-2 System Clock Definition 19

Chapter 3 Port Functions

3-1 Port Allocation 22
3-2 Port Configuration 24
3-2-1 Port 0 Configuration 24
3-2-2 Port 1 Configuration 28
3-2-3 Port 2 Configuration 33
3-2-4 Port 4 Configuration 39
3-2-5 Port 5 Configuration 43
3-2-6 Port 6 Configuration 44
3-2-7 Port 7 Configuration 48
3-2-8 Port 8 Configuration 49
3-2-9 Port 9 Configuration 50
3-2-10 Port A Configuration 51
3-2-11 Port B Configuration 54
3-2-12 Port C Configuration 59

Chapter 4 Interrupt Functions

4-1 Interrupt Causes and Vector Addresses 62
4-1-1 Overview 62
4-1-2 Interrupt Control Register 70
4-1-3 Interrupt Level 72
4-1-4 Interrupt Processing Operations 75
4-1-5 Interrupt Return Operations 76
4-1-6 Nested Interrupts 77

4-2 Method of Using Interrupts 79
4-2-1 Reset Interrupt 79
4-2-2 Non-maskable Interrupts (WD, undefined instruction execution) 80
4-2-3 External Pin Interrupts (IRQ0, 1, 2, 3, 4) 81
4-2-4 KEY Interrupt 84
4-2-5 Cylinder FG Interrupt 87
4-2-6 Capstan FG Interrupt 89
4-2-7 Timer 0 Interrupt 92
4-2-8 Timer 1 Interrupt 94
4-2-9 Timer 2 Interrupt 96
4-2-10 Timer 3,4 Interrupts 99
4-2-11 Timer 6 Interrupt 101
4-2-12 Control Interrupt 103
4-2-13 HSW Both Edge Interrupt 106
4-2-14 Servo VSYNC Interrupt 109
4-2-15 SSPG Interrupt, FOCR0, 1, 2 Interrupts 112
4-2-16 OSD Interrupt 115
4-2-17 XDS Interrupt 118
4-2-18 Serial 0 Interrupt 120
4-2-19 Serial 1 Interrupt 123
4-2-20 Serial 2 Interrupt 126
4-2-21 AD/PWM14 Interrupt 127
4-2-22 OSD VSYNC Interrupt 129
4-2-23 Undefined instruction Interrupt (during ICE operation) 132
4-3 Task Control Section 133
4-3-1 Overview 133
4-3-2 Initial Processing of Tasks 133
4-3-3 Method of Starting the Second Task 136
4-3-4 Method of Switching Tasks 138
4-3-5 Method of Stopping a Task 140
4-3-6 Task Manipulating Instructions and Operation during Conflict with the Task Timer 142
4-3-7 Interrupt Acceptance Operation during Task Switching 143
4-3-8 Task Timer Register (TSKBC) Operation 144
4-3-9 Task Starting during Interrupt Processing 146

Chapter 5 Timer Functions

5-1 Timer 0 Functions 148
5-1-1 Timer 0 Settings 148
5-1-2 How to Use Timer 0 148
5-1-3 Timer 0 Interrupt Processing Settings 148

5-2 Timer 1 Functions 150
5-2-1 Timer 1 Settings 150
5-2-2 How to Use Timer 1 150
5-2-3 Timer 1 Interrupt Processing Settings 150
5-3 Timer 2 Functions 152
5-3-1 Timer 2 Settings 152
5-3-2 How to Use Timer 2 (Timer Function) 152
5-3-3 Timer 2 Interrupt Processing Settings 153
5-3-4 How to Use Timer 2 (Input Capture Function) 153
5-3-5 How to Use Timer 2 (Duty Judgment Function) 154
5-4 Timer 3 Functions 156
5-4-1 Timer 3 Settings 156
5-4-2 How to Use Timer 3 (Timer Function) 156
5-4-3 How to Use Timer 3 (Serial Transfer Header Function) 156
5-4-4 Timer 3 Interrupt Processing Settings 157
5-5 Timer 4 Functions 159
5-5-1 Timer 4 Settings 159
5-5-2 How to Use Timer 4 (Timer Function) 159
5-5-3 Serial Transfer Clock Generation Function 160
5-5-4 Timer 4 Interrupt Processing Settings 160
5-6 Timer 5 Operation 162
5-6-1 Timer 5 Functions and Common Settings 162
5-6-2 Setting and Operation of the Watchdog Timer Function 162
5-6-3 Oscillation Stabilization Wait Function 163
5-7 Timer 6 Functions 165
5-7-1 Timer 6 Settings 165
5-7-2 Timer 6 Operation 165
5-7-3 Timer 6 Interrupt Processing Settings 165
5-8 Buzzer Output Function 167
5-8-1 Buzzer Output Setting 167
5-8-2 Clock Output Function 171

Chapter 6 Serial Functions

6-1 Serial 0 Functions 174
6-1-1 Serial 0 Configuration 174
6-1-2 Synchronous Serial Interface Transmission Settings for Serial 0 175
6-1-3 Synchronous Serial Interface 0 Transmission Timing 177
6-1-4 Synchronous Serial Interface Reception Settings for Serial 0 178
6-1-5 Synchronous Serial Interface Reception Timing 180
6-1-6 UART Serial Interface Operation 181
6-1-7 UART Serial Interface Transfer Timing Diagram 184

6-2 Serial 1 186
6-2-1 Serial 1 Configuration 186
6-2-2 Serial Interface 1 (SIF1) Operation 189
6-2-3 Serial Interface 1 (SIF1) Transmission Timing 190
6-2-4 Serial Interface 1 (SIF1) Reception Timing 194
6-2-5 Serial Interface 1 (SIF1) Registers 198
6-2-6 Remote Control Output Function 200
6-2-7 Serial Transfer Header 208
6-3 Serial 2 210
6-3-1 Overview 210
6-3-1-1 I²C Bus 210
6-3-1-2 Operating Modes 212
6-3-1-3 Data Transfer Sequence 213
6-3-1-4 Data Format 214
6-3-1-5 Block Diagram 215
6-3-2 I²C Serial Interface Control Registers 216
6-3-2-1 Data Shift Register (I²CR0) 217
6-3-2-2 Address Register (I²CR0') 217
6-3-2-3 Status Register (I²CR1) 218
6-3-2-4 Clock Control Register (I²CR2) 222
6-3-2-5 I²C System Control Register (I²CCN) 223
6-3-2-6 I²C Direction Control Register (I²CDIR) 224
6-3-3 Transmission Control 225
6-3-3-1 Recognition Sequence in Acknowledgment Mode 225
6-3-3-2 Arbitration Sequence 227
6-3-3-3 Clock Synchronization Characteristics 228
6-4 I²C Serial Transfer Settings 229
6-4-1 Master Transmission Procedure 229
6-4-2 Master Reception Procedure 230
6-4-3 Slave Transmission Procedure 231
6-4-4 Slave Reception Procedure 233

Chapter 7 A/D Converter and D/A Converter

7-1 A/D Converter 236
7-1-1 A/D Converter Configuration 236
7-1-2 A/D Conversion Operation and Register Configuration 237
7-1-3 Interrupt Processing Settings 237
7-1-4 Warnings when Using the A/D Converter 238
7-2 D/A Converter 240
7-2-1 D/A Converter Configuration 240
7-2-2 How to Use the D/A Converter 241

Chapter 8 Synchronous Output Functions

8-1 Synchronous Output Configuration 244
8-2 Synchronous Output Control Registers 245
8-3 Synchronous Output Function 1 (SPG Function) 250
8-3-1 Synchronous Output Pin Configuration 250
8-3-2 How to Use Synchronous Output 1 250
8-4 Synchronous Output Function 2 (HOCR Function) 253
8-4-1 Synchronous Output Pin Configuration 253
8-4-2 How to Use Synchronous Output 2 254
8-5 D/A Conversion Synchronous Output 259
8-5-1 D/A Conversion Output Pin Configuration 259
8-5-2 How to Use the D/A Conversion Synchronous Output 259
8-6 Synchronous Output Interrupts 262
8-7 RCTLD Signal Generation 264

Chapter 9 Servo Functions

9-1 Analog Input Signal Processing Unit 268
9-1-1 Analog Input Signal Processing Circuits 268
9-2 Servo Signal Processing Unit 280
9-2-1 Servo Signal Processing Circuit 280
9-2-2 VSYNC Weak Electric Field Backup Function (NF, N8) 292
9-3 Input Capture Unit 295
9-3-1 Input Capture Unit Configuration 295
9-3-2 Control Registers of the Input Capture Unit 295
9-3-3 Input Capture Interrupt 300
9-3-4 Cylinder and Capstan Phase Control 301
9-4 PWM Unit 305
9-4-1 PWM Unit Configuration 305
9-4-2 PWM Unit Output Control 307
9-4-3 PWM14 Settings 310
9-4-4 Method of Adding Bits 311
9-5 Pseudo-V Signal Generating Unit 318
9-5-1 Pseudo-V Signal Control Registers 319
9-5-2 Pseudo-H Signal Superimposition Function 322
9-6 Head Amp and Rotary Control Unit 323
9-6-1 Dedicated Output Functions for Head Amp and Rotary Control 323
9-7 Digital PGMM 325
9-7-1 Digital PGMM Configuration 325
9-7-2 Digital PGMM Operation 326

Chapter 10 Standby, Clock Operation/Halt, and Reset Functions

10-1 Standby Functions 328
10-1-1 Entering Standby Mode 328
10-1-2 CPU Mode Control Register 331
10-1-3 Transition between Operating Modes 333
10-1-4 Transition between SLOW and NORMAL Modes 340
10-1-5 Entering Standby Mode 342
10-1-6 Return from Standby Mode 344
10-2 Reset Function 345
10-3 Power Supply Control 346

Chapter 11 OSD and XDS Functions

11-1 OSD Function 348
11-1-1 OSD Function Summary 348
11-1-2 Pin Configuration 350
11-1-3 Control Registers 351
11-1-3-1 OSDCNT1 OSD Operation Control Register 1 352
11-1-3-2 OSDCNT2 OSD Operation Control Register 2 355
11-1-3-3 OSDCNT3 OSD Operation Control Register 3 357
11-1-3-4 BKALCL Screen Background Color Control Register 359
11-1-3-5 VP Vertical Display Start Position Control Register 361
11-1-3-6 HP Horizontal Display Start Position Control Register 362
11-1-3-7 VOFS Display Line Interval Control Register 363
11-1-3-8 VLIN Display Control Register 364
11-1-3-9 VCOCNT VCO Control Register 366
11-1-3-10 CLCNT Video Signal Control Register 368
11-1-3-11 DOTCNT Dot Clock Control Register 370
11-1-3-12 PCSTA Horizontal Sync AFC Control Register 372
11-1-3-13 CMPLIN OSD Interrupt Display Line Control Register 374
11-1-3-14 ROMEND CGROM Last Address Register 375
11-1-4 V-RAM 376
11-1-4-1 Line Control VRAM CN2 378
11-1-4-2 Line Control VRAM CN1 380
11-1-4-3 Line Control VRAM CN0 381
11-1-4-4 Character Control VRAM CCn (n=1 to 24) 382
11-1-4-5 Character Control VRAM CDn (n=1 to 24) 384
11-1-5 CGROM 385
Internal Synchronization Mode Screen Display 391
External Synchronization Mode Screen Display 394

11-1-6 OSD Peripheral Circuits 397
11-1-6-1 CVIN2 Input Filter Setting 397
11-1-6-2 AFC Filter Setting 398
11-1-6-3 Precautions on the FSCI Pin Input 398
11-2 XDS Data Slice Function 399
11-2-1 XDSLIN1, XDSLIN2 Settings 400
11-2-2 Sampling Clock Phase Adjustment 402
11-2-3 Data Slice Levels 403
11-2-4 XDS Interrupt Control 404
11-2-5 XDS Data Register 405
11-3 Synchronization Judgment 407
11-3-1 Synchronization Judgment Circuit Configuration 407
11-3-2 Method of Using Sync Judgment 411
11-4 Synchronization Signal Separation Function 414
11-4-1 Synchronization Signal Separation Function Summary 414
11-4-2 Horizontal Synchronization Signal Separation Function 415
11-4-3 Vertical Synchronization Signal Separation Function 416
11-4-4 Field Judgment Function 419
11-4-5 Method of Using Sync Separation Function 420

Chapter 12 Appendix

RAM Address Map 422
Register Description 425
Cylinder Speed Controller 566
Configuration of the Cylinder Speed Controller 566
Cylinder Speed Control Registers 567
Cylinder Speed Control Reference Value (Preset Value) Setting 571
Capstan Speed Controller 573
Configuration of Capstan Speed Controller 573
Capstan Speed Control Register 574
Capstan Speed Control Reference Value (Preset Value) Setting 577
List of Instructions 579
Addressing Modes 580
Internal Registers 581
Address Registers 581
Operation Registers 583
Processor Status Word 584
The Instruction Set 587
Instruction Map 593
Precautions in Manipulating the Interrupt Flags 595

Chapter 1 Overview

1

1-1 Product Overview

1-1-1 Overview

This LSI microcontroller has built-in hardware optimized for high-speed, real-time multi-tasking.

Equipped with a highly efficient instruction set, a powerful CPU core, and a wide variety of highly functional peripheral functions, VCR device control, servo control, timer functions and OSD display functions can be implemented on a single chip.

OSD:
Abbreviation of On Screen Display.

Two oscillation circuits (high-speed oscillation/low-speed oscillation) are built-in to allow switching between high and low-speed system clocks.

1-1-2 Features

- Digital servo and system control functions for VCRs
- OSD display function (variable character types, 512 characters maximum)
- Built-in ROM correction function
- Built-in XDS (US closed caption data slicer)
- C language supported (assembler ratio 1.0)
- Servo analog interface
- CPU multi-tasking (4 tasks maximum)
- Abundance of peripheral devices (timer, serial, A/D, etc.)
- IIC serial interface function
- 16-bit arithmetic processing (high-speed processing function)
- Variety of ROM versions (72 Kbytes, 96 Kbytes, 128 Kbytes, 160 Kbytes)

XDS:
Abbreviation of eXtended Data Service.

1-1-3 Product Line

Table 1-1-1 Product Line

Model	ROM Capacity	RAM Capacity	Classification
MN101D02D	72Kbytes	2048 bytes	Mask ROM version
MN101D02F	96Kbytes	3072 bytes	Mask ROM version
MN101D02G	128Kbytes	4096 bytes	Mask ROM version
MN101D02H	160Kbytes	5120 bytes	Mask ROM version
MN101DP02JAF	192Kbytes	5120 bytes	EPROM version (OTP)
MN101DP02JAC	192Kbytes	5120 bytes	EPROM version (ATP)

OTP:
Abbreviation of One Time Programmable.
OTP Microcomputers are sealed in plastic packages and a program can be written in them only once.
ATP:
Abbreviation of Any Time Programmable.
Although ATP microcomputers are sealed in a ceramic package, they have a glass window that enables writing a program again in them after erasing the previous program by exposing the microcomputer to ultraviolet light.

1-2 Hardware Functions

ROM 72 Kbytes, 96 Kbytes, 128 Kbytes, 160 Kbytes

- Used for programming and also shared with the OSD function CG-ROM
- The OSD function CG-ROM area can be changed from 0 to 18 Kbytes corresponding to the number of character types (approximately 9 Kbytes are used for 256 character types).

RAM 2048 bytes, 3072 bytes, 4096 bytes, 5120 bytes

- Used for data area, stack area, and OSD function V-RAM
- OSD function V-RAM area

When the number of character types is less than or equal to 128 characters:

0 to 336 bytes can be set in 54 byte units corresponding to the display screen configuration (270 bytes are used when there are 10 lines in one screen).

When the number of character types is greater to or equal to 128 characters:

0 to 624 bytes can be set in 102 byte units corresponding to the display screen configuration (510 bytes are used when there are 10 lines in one screen).

Minimum instruction execution time

When using main clock	139.7 ns (@14.32 MHz)
	112.8 ns (@17.73 MHz)
When using sub clock	61 µs (@32.768 kHz)

Interrupts RESET, watchdog, timer-counters 0 to 4, timer-counter 6, Serial 0 to Serial 2, A/D conversion complete (PWM14), external pins 0 to 4, key interrupts, cylinder FG, capstan FG, control, HSW both edge, servo VSYNC, synchronous output, OSD, XDS, undefined instruction execution, OSD VSYNC

Timer-counters Timer-counter 0:16 bits × 1

Function summary	Timer function, linear timer counter function
Clock source	1/2, 1/4, 1/8, 1/16 of system clock, timer-counter 6 overflow XI oscillation clock, or 1/512 of OSC oscillation clock
Interrupt source	Overflow of timer-counter 0

Timer-counter 1: 16 bits × 1

- Function summary ..... Timer function, serial time counter function
- Clock source ..... 1/2, 1/4, 1/8, 1/16 of system clock, CTL signal
- Interrupt source ..... Overflow of timer-counter 1

Timer-counter 2: 16 bits × 1

- Function summary ..... Timer function, input capture function, duty determination function (VISS/VASS signal detection)
- Clock source ..... 1/2, 1/4, 1/8, 1/12, 1/16, 1/24 of system clock
- Interrupt source ..... Overflow of timer-counter 2, designated edge of DCTL signal, underflow of the 4-bit counter of the timer 2 shift register, matching of timer 2 shift register and compare register
- Duty detection ..... 40%, 50%, 60% settings are possible

Timer-counter 3: 16 bits × 1

- Function summary ..... Timer function, serial transfer header function
- Remote control output .... Carrier period generation
- Clock source ..... 1/2, 1/4, 1/8, 1/16 of system clock
- Interrupt source ..... Overflow of timer-counter 3

Timer-counter 4: 16 bits × 1

- Function summary ..... Timer function, event counter function, serial transfer clock generation function
- Clock source ..... 1/8, 1/16 of system clock, TC4I (Timer 4 event input pin)
- Interrupt source ..... Overflow of timer-counter 4

Timer-counter 5: 19 bits × 1

Function summary ..... Watchdog function, oscillation stabilization wait function

Clock source ..... System clock

Interrupt source ..... $1/2^{16}$, $1/2^{19}$ Outputs of timer-counter 5

Oscillation stability release ..... $1/2^{8}$ (during reset release), $1/2^{14}$, $1/2^{17}$ of timer-counter 5

Timer-counter 6: 16 bits × 1

Function summary ..... Clock function

Clock source ..... 1/4, 1/8, 1/64, 1/128 of system clock, 1/512 of OSC oscillation clock, XI oscillation clock

Interrupt source ..... $1/2^{13}$, $1/2^{14}$, $1/2^{15}$ of timer-counter 6, overflow of timer-counter 6

Serial Interface

Serial 0: 8 bits × 1 (synchronous/start-stop synchronization)

Function summary ..... Transfer direction is selectable as MSB or LSB first.

Clock source ..... 1/4, 1/8, 1/16, 1/32, 1/64, 1/128, 1/256 of system clock, timer 4 output frequency divided by 2, NSBT0 pin output

UART clock ..... The above clock frequencies are divided by 8.

Serial 1: 8 bits × 1 (synchronous, remote control transmission, simple remote control reception)

Function overview ..... Transfer direction is selectable as MSB or LSB first, start condition function

Clock source ..... 1/8, 1/16, 1/32, 1/64, 1/128, 1/256 of system clock, timer 4 output frequency divided by 2, NSBT0 pin output

Remote control clock ... Timer 4 output frequency divided by 2 (synchronous/I^2C)

Serial 2: 8 bits × 1 (I^2C)
- Function overview ..... Master transmit/receive signals, slave transmit/receive signals
- Clock source ..... 1/72, 1/80, 1/84, 1/96, 1/102, 1/112, 1/120, 1/128, 1/144, 1/160, 1/168, 1/192, 1/224, 1/256, 1/320 of system clock, SCK pin input

I/O ports

73 I/O ports
(15 ports with internal pull-up resistors)
4 input ports
Dedicated servo ports 13
Others 10

A/D conversion

8-bit A/D converter × 12 channels
- Conversion frequency ..... 7.8 µs (@14.32 MHz)

PWM

13-bit PWM × 2 channels
- Resolution ..... OSC oscillation clock cycle
- Bit modulation ..... 5 bits

10-bit PWM × 2 channels
- Resolution ..... OSC oscillation clock cycle
- Bit modulation ..... 4 bits

14-bit PWM × 1 channel
- Resolution ..... OSC oscillation clock cycle
- Bit modulation ..... 6 bits

Servo input signal

Control amp

Analog interface for processing

FG 1st stage amp
FG Schmitt amp
PG/FG3-value input
PG analog monostable multivibrator

Servo signal processing

HSW generation function
- Cylinder FG frequency division × 1
- Digital PG monostable multivibrator × 1
- Resolution ..... 16 times the OSC clock cycle
- Maximum cycle setting ..... 73.2 ms (@14.32 MHz)
- At the nth FG after PG, starting with the monostable multivibrator is possible

2H compensation function
Delay value setting .... 2032 times or 2048 times the OSC clock cycle
High-speed switching signal (HAMP/ROTA) generation function
Pseudo V-sync generation function
Resolution ..... 128 times the OSC clock cycle
Maximum cycle setting
..... 2.29 ms (@14.32 ms)
Capstan FG frequency division × 1
Control frequency division × 1
Capstan FG mask timer × 1
Resolution ..... 128 times the OSC clock cycle
Maximum cycle setting
..... 2.29 ms (@14.32 MHz)

FRC
26-bit free running counter × 1
Clock source ..... 1/4, 1/8 of OSC oscillation clock, upper 24 bits can be read

ICR
18-bit input capture register × 6
Capture factors ..... Designated edge of capstan FG signal, designated edge of cylinder FG signal, designated edge of control signal, designated edge of HSW signal (falling edge or both edges), falling edge of recording control signal, rising edge of V-sync signal
Resolution ..... 4 times or 8 times the OSC oscillation clock cycle
Maximum measurement cycle
..... 73.2 ms (@14.32 MHz)

Synchronous output function
16-bit output compare register × 4
Function summary .. Synchronous output function, N8/NF generation function, control overwrite function, recording control signal generation function
Resolution ..... 16 times or 32 times the OSC oscillation clock cycle
Maximum cycle setting
..... 73.2 ms (@14.32 MHz)
Synchronous output pin
..... Synchronous output: 7 pins
3-value synchronous output: 4 pins

OSD function

Compatible with menu display or superimposed display

Compatible broadcast formats:
NTSC, PAL, PAL-M, PAL-N

Screen configuration:	24 characters × 2n lines (where n=1 to 6)
Character types:	512 character types maximum (variable)
Character size:	12 dot × 18 dot
Character enlargement:	Settings of 2×, 3×, 4× are possible in both horizontal and vertical directions
Character interpolation:	None
Line background color:	8 colors (during menu display, can be set for each row)
Line background brightness:	8 level adjustments, can be set for each row
Screen background color:	8 colors
Character color:	White
Character brightness:	8 level adjustments, can be set for each row
Border function:	1 dot border in 4 or 8 directions
Border brightness:	4 level adjustments, can be set for each row
BOX shadow function:	Can be set for each character (only when there are 128 or more character types during composite output)
Blinking:	None (Blinking can be implemented in software)
Highlighted characters:	Can be set for each character
Halftone:	Can be set for each row, 2 brightness level adjustments (set for each row)
Input:	Composite video signal input (output signal level 1 V_{PP}/2 V_{PP})
Clamp method:	Sync chip clamp, 4 clamp levels
Output:	Composite video signal output Digital output (6 pins) During digital output, the color and background of each character can be selected from 8 colors
Vibration resistant display:	Built-in AFC circuit

Sync signal detection function

Detection function for horizontal sync signal and vertical sync signal (integration method)

Interpolation function for horizontal sync signal

ROM correction function
Correction address specification: up to 2 addresses can be specified
Correction method: correction program stored in internal RAM

XDS function Built-in US closed caption data slicer
An arbitrary 2 lines of data can be extracted

Other Built-in synchronous evaluation circuit

Standby modes STOP and HALT modes

Error prevention functions
Undefined code fetch
Overflow of watchdog timer

Operating Voltage Range
4.0 V to 5.5 V (for 14.32 MHz)
2.5 V to 5.5 V
(During operation at 32.768kHz and 1/1024th of 14.32MHz)

Package 100 QFP (18 mm square, 0.65 mm lead pitch)

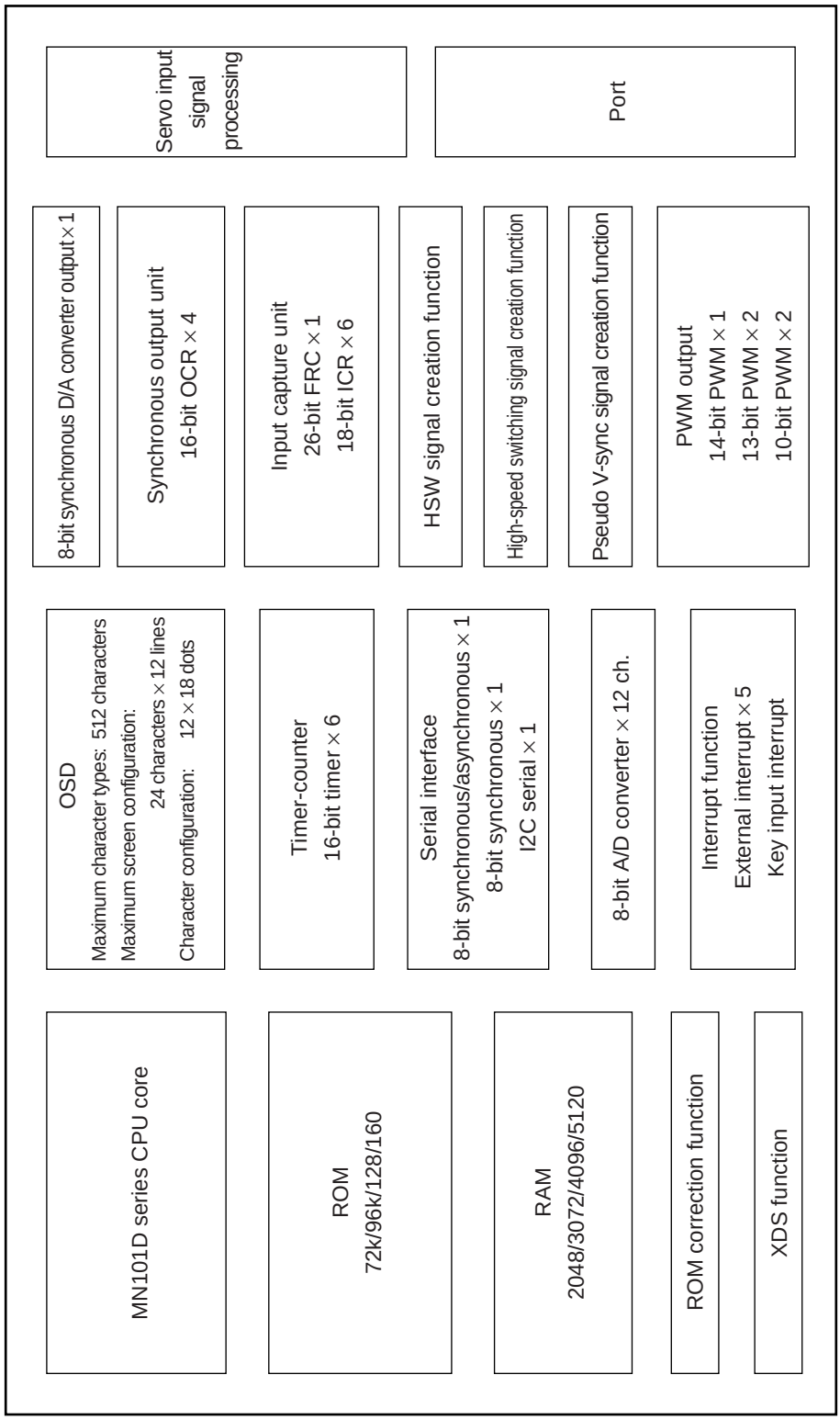


Figure 1-2-1 Block Diagram

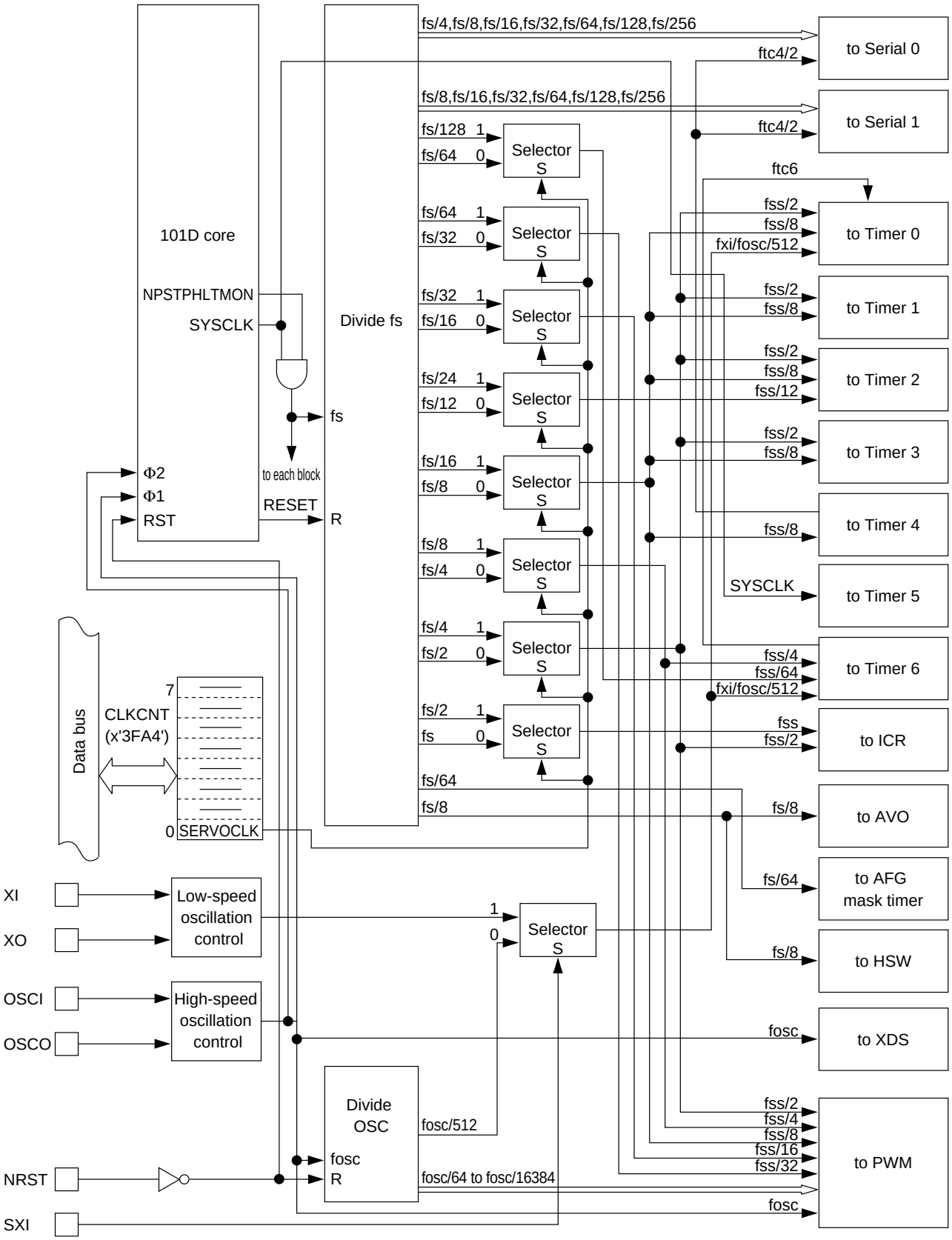


Figure 1-2-2 Block Diagram of Clock Control

1-3 Pin Functions

1-3-1 Pin Connection Diagram

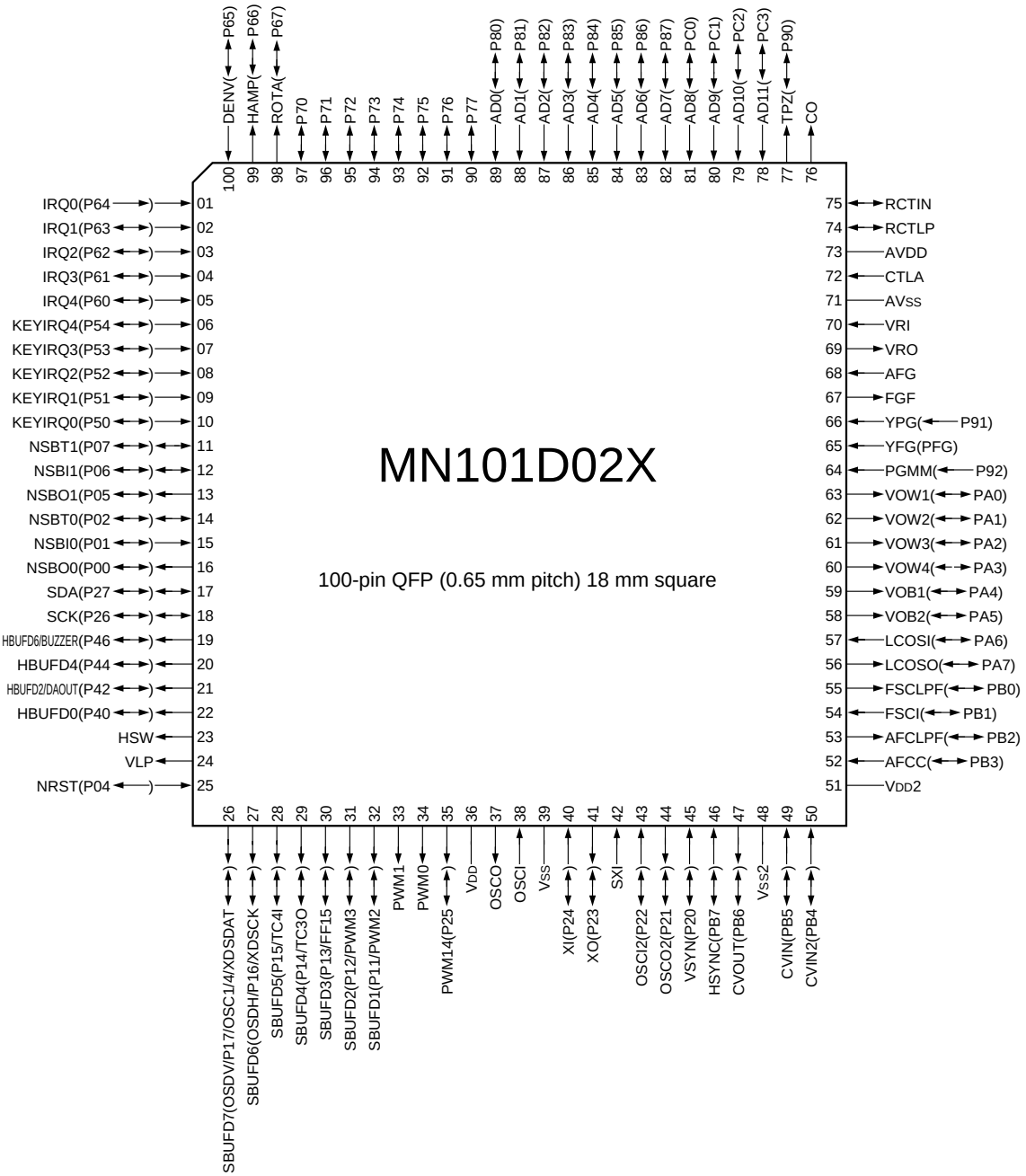


Figure 1-3-1 MN101D02X Pin Diagram

1-3-2 Pin Function Summary

Table 1-3-1 Pin Function Summary (1/4)

Pin Name	Type	Function	Description
VDD	—	Power supply	Connects to 5 V. (Typ)
VSS	—	Power supply	Connects to 0 V (GND).
VDD2	—	OSD power supply	Power supply for OSD. Connects to 5 V. (Typ)
VSS2	—	OSD power supply	Power source for OSD. Connects to 0 V (GND).
AVDD	—	Analog power supply	Analog power supply for A/D and servo. Connects to 5 V. (Typ)
AVSS	—	Analog power supply	Analog power supply for A/D and servo. Connects to 0 V (GND).
OSCI	Input	High-speed oscillation input pin	Connects to a crystal oscillator. There is an internal feedback resistor between OSCI and OSCO. If an external clock is to be used, couple this pin with capacitors and input the clock signal to OSCI, leaving OSCO unconnected. This signal is halted during reset.
OSCO	Output	High-speed oscillation output pin	This is an output pin for the system clock oscillation circuit. If used with self-excited oscillation, connect a crystal oscillator between OSCI and OSCO. Oscillation stops if the reset pin goes to a low-level.
OSCI2 (P22)	Input (I/O)	Oscillation input pin for OSD	Connects to a crystal oscillator. There is an internal feedback resistor between OSCI2 and OSCO2. If an external clock is to be used, couple this pin with capacitors and input the clock signal to OSCI2, leaving OSCO2 unconnected. This pin can also be used as general purpose I/O pin P22.
OSCO2 (P21)	Output (I/O)	Oscillation output pin for OSD	This is the output pin for the OSD clock oscillation circuit. If used with self-excited oscillation, connect a crystal oscillator between OSCI2 and OSCO2. This pin can also be used as general purpose I/O pin P21.
NRST (P04)	Input (Output)	Reset pin (output pin)	This pin resets the microcontroller. Setting the NRST pin to a low-level will activate the reset. A reset can be implemented through software by writing '0' to pin P04. The NRST pin has an internal pull-up resistor and an n-channel transistor output.
SXI	Input	Operating mode selection pin	After reset is released, the high or low-level of the SXI pin determines whether there is XI or XO oscillation for the operating mode. The oscillation stabilization wait interval immediately after reset is 2^{18}/fosc, regardless of the level of the SXI pin. The initial operating mode is the normal mode with the OSC frequency divided by 1024. [☞ Chapter 10 "Standby, Clock Operation/Halt, and Reset Functions"]
XI (P24)	Input (I/O)	Low-speed oscillation input pin	Connects to a crystal oscillator. There is an internal feedback resistor between XI and XO. If an external clock is to be used, couple this pin with capacitors and input the clock signal to XI, leaving XO unconnected. During reset, this pin is set as XI. However, it may also be configured by register settings for use as I/O pin P24. If connected, a crystal oscillator will oscillate during the reset interval.
XO (P23)	Output (I/O)	Low-speed oscillation output pin	This is an output pin for the system clock oscillation circuit. If used with self-excited oscillation, connect a crystal oscillator between XI and XO. During reset, this pin is set as XO. However, it may also be configured by register settings for use as I/O pin P23.

Table 1-3-1 Pin Function Summary (2/4)

Pin Name	Type	Function	Description
NSBO0 (P00) NSBO1 (P05)	Output (I/O)	Serial data transmission pins	These are data transmission pins for the serial interface. Serial data is output as negative logic. Output data changes at the falling edge of the serial clock. If used as P00 or P05, these are I/O pins.
NSBI0 (P01) NSBI1 (P06)	Input (I/O)	Serial data reception pins	These are data reception pins for the serial interface. Serial data is input as negative logic. Input data changes at the rising edge of the serial clock. If used as P01 or P06, these are I/O pins.
NSBT0 (P02) NSBT1 (P07)	I/O (I/O)	Serial clock I/O pins	These I/O pins are for the serial interface clock. When the internal clock is set, these are output pins. When an external clock is set, these are input pins. If used as P02 or P07, these are I/O pins. P07 can be used as a general purpose I/O during remote control output.
SDA (P27)	I/O (I/O)	Serial data I/O pin	This is the I/O pin for the IIC serial interface. If used as P27, this is an I/O pin.
SCK (P26)	I/O (I/O)	Serial clock I/O pin	This is the I/O pin for the IIC serial interface clock. If used as P26, this is an I/O pin.
IRQ0 to IRQ4 (P64 to P60)	Input (I/O)	External interrupt input pins	Interrupts are generated when the signal level at the external interrupt input pins rises or falls. The enabled edge can be set by a control register. If used as P60 to P63, these are I/O pins. P64 is an input-only pin.
AD0 to AD11 (P80 to P87) PC0 to PC3	Input (I/O)	A/D conversion input pins	These are 8-bit A/D conversion input pins. AD0 to AD11 can also function as P80 to PC3, for use as I/O pins.
DENV (P65) HAMP (P66) ROTA (P67)	Input (I/O) Output (I/O)	DENV input pin HAMP/ROTA switching output pins	The DENV input pin is used to generate and output HAMP/ROTA signals. If used as P65, this is an I/O pin. These output pins are dedicated to HAMP/ROTA switching. Setting P66 and P67 as input pins allows the levels at the HAMP and ROTA output pins to be read. If used as P66 and P67, these are I/O pins.
P70 to P77	I/O	General purpose I/O pin	This is a general purpose input/output pin. It is possible to specify the presence or absence of a pull-up resistor for every bit.
KEYIRQ0 to KEYIRQ4 (P50 to P54)	Input (I/O)	KEY input interrupt pins	These are the KEY input interrupt pins. An interrupt is generated when any of these pins goes to low-level. If used as P50 to P54, these are I/O pins.
SBUFD1 to SBUFD7 (P11 to P17, PWM2, 3, TC3O, TC4I, OSCDIV, OSDV, OSDH, XDSDAT, ZDSCK)	Output (I/O)	Synchronous output pins	These are the synchronous output pins. When the contents of the SPGTIM register match those of the free running counter, data set in the SPGDAT register is latched into the SPGBUF register and then output to SBUFD1 through SBUFD7. When not used for synchronous output, these pins can be configured as general purpose I/O ports with the direction for each bit specified in the SPGDIR register. Further, SBUFD7 outputs 1/2 frequency signal of OSC (2Vpp under no load) or 1/4 frequency signal (1Vpp under no load) (OSDV or XDSDAT), SBUFD1,2 output the 10-bit PWM2,3, SBUFD4 outputs TC3, SBUFD5 is the TC4 clock input pin, and SBUFD6 is the output pin for OSDH or XDSCK. A pull-up resistor can be specified between presence or absence for each pin.

Table 1-3-1 Pin Function Summary (3/4)

Pin Name	Type	Function	Description
HBUFD0, 2, 4, 6 (P40, P42, P44, P46, BUZZER) DAOUT	Output (I/O)	Synchronous output pins	A pull-up resistor can be enabled or disabled for each bit. When the contents of the free running counter match those of the output compare register, data set in the HOCRDA0 register is latched into HOCRBUF0 and output to the pins. When not used for synchronous output, setting the odd numbered bits in HOCRBUF0 will allow these pins to be configured as general purpose I/O ports with input or output specified for each bit. HBUFD6 is also the BUZZER output and HBUFD2 is also the D/A converter output.
HSYNC (PB7)	Input (I/O)	Horizontal sync signal input pin	This is the horizontal sync signal input pin. When not used as HSYNC, this is a general purpose pin.
HSW	Output	Head SW output pin	Setting the YFG frequency division in the YFGDIV register, causes the frequency of the YFG signal to be divided. After shifting during the interval that begins at the YPG signal timing and is set by the PG monostable multivibrator (PGMM pin or digital PGMM function), the HSW signal is generated and output.
VLP	Output	Pseudo-VSYNC output pin	Setting the AVPR0, AVPR1, and AVOCR registers causes a pseudo-VSYNC signal, synchronized with the HSW signal, to be output. Different output timings are possible at the rising edge and falling edge of the HSW signal.
VSYNC	Input	VSYNC input pin	This is the vertical sync signal input pin. When not used as VSYNC, this is a general purpose pin. TTL and CMOS input levels are selectable.
PWM 0,1	Output	PWM output pins	PWM0 and PWM1 are 13-bit PWM output pins. A sign bit and overflow bit are appended to the PWM data. If '0' is set in the corresponding PWM register, a signal with a 50% duty cycle will be output. The resolution of each PWM is 1/fosc and 5 bits are modulated. A PWM signal is output to each pin in accordance with data set in the corresponding PWM register. Since this pin is in the High-Z state immediately after reset, it is necessary to enable output when outputting PWM.
PWM14 (P25)	Output (I/O)	14-bit PWM output pin	PWM14 is the 14-bit PWM output pin. If not used as PWM14, this pin can be used as a general purpose port (P25).
PGMM (P92)	Input (Input)	PG analog monostable multivibrator (general purpose input pin)	Connecting a capacitor and resistor to this pin will delay output of the HSW signal that will be output by the YFG and YPG signals. If not used as the PGMM pin (when using digital PGMM), this pin can be used as a general purpose port (P92). If used as the PGMM pin, disable digital inputs and make sure there is no current flow.
FGF	Output	AFG limiter amp output pin	This is the limiter amp output pin for the CAP FG signal. This pin outputs the AFG signal after being amplified by the CAP FG amp.
AFG	Input	AFG limiter amp input pin	This is the limiter amp input pin for the CAP FG signal. The CAP FG signal is input. After being amplified by the amp, the signal is output from the FGF pin.
YPG (P91)	Input (Input)	YPG Schmitt amp input pin	This is the Schmitt amp input pin for the CYL PG signal. If the ANACNT register has selected 3-valued YFG and YPG signal input for the YFG pin, this pin can be used as a general purpose port (P91). When this pin is used as YPG, the digital input is disabled so that no conduction current flows.

Table 1-3-1 Pin Function Summary (4/4)

Pin Name	Type	Function	Description
YFG (PFG)	Input	YFG Schmitt amp input pin (YFG/YPG 3-valued input pin)	This is the Schmitt amp input pin for the CYL FG signal. 3-valued input of YFG and YPG signals is also possible. Switching between 3-valued input and the YFG amp input is performed by the ANACNT register.
VRO	Output	Servo reference voltage output pin	This is the amp output pin for the VRI (servo reference voltage input) pin level.
VRI	Input	Servo reference voltage input pin	This is the input pin for the servo reference voltage. An internal resistor divider network eliminates the necessity to attach external resistors. This pin is fixed at a low-level when the analog power source is on standby.
TPZ (P90)	Output (Input)	Trapezoidal wave generating pin with attached capacitor (timer event counter input pin)	This pin with an attached capacitor generates a trapezoidal waveform, a CTL overwrite signal. Connecting a capacitor to this pin will generate a trapezoidal waveform of the RCTLP/RCTLN output. If used as the TPZ pin, disable digital inputs and make sure there is no current flow. If not used as the TPZ pin, this pin can be used as a general purpose input port (P90). This is the CTL signal I/O pin. During recording, the REC CTL signal is output; during CTL overwrite, a trapezoidal waveform is output; during playback, this pin is the CTL signal input.
RCTLP RCTLN	I/O I/O	CTL head pin (+) CTL head pin (–)	These are the CTL signal I/O pins. During recording, a signal is output with reverse polarity of the RCTLP pin output. During CTL overwrite, a trapezoidal waveform is output. During playback, these are CTL signal input pins.
CTLA	Input	CTL amp AC ground pin	This is the AC ground pin for the CTL amp.
CO	Output	CTL amp output pin	This is the amp output pin for the CTL signal. This pin outputs the CTL signal that has been amplified by the CTL amp.
CVIN (PB5)	Input (Input)	Composite video signal input	This video input pin is for superimposed and XDS video signals. This pin can also be used as a general purpose I/O pin (PB5).
CVIN2 (PB4)	Input	Composite video signal input2	This is the video input pin for separating sync signals. This pin can also be used as a general purpose I/O pin (PB4).
LCOSI (PA6)	Input (Input)	LC transmit input circuit	This input pin is for use with an LC transmission circuit. This pin can also be used as a general purpose I/O pin (PA6).
LCOSO (PA7)	Output (I/O)	LC transmit output circuit	This output pin is for use with an LC transmission circuit. This pin can also be used as a general purpose I/O pin (PA7).
AFCC (PB3)	Input (I/O)	AFC low-pass filter input pin	This pin is for the VCO low-pass filter built-in to the AFC circuit. This pin can also be used as a general purpose I/O pin (PB3).
AFCLPF (PB2)	Output (Input)	AFC low-pass filter output pin	This pin is for the VCO low-pass filter built-in to the AFC circuit. This pin can also be used as a general purpose I/O pin (PB2).
FSCI (PB1)	Input (I/O)	4X low-pass filter input pin	This pin is for the low-pass filter of the 4X multiplying circuit. This pin can also be used as a general purpose I/O pin (PB1).
FSCLPF (PB0)	Output (Input)	4X low-pass filter output pin	This pin is for the low-pass filter of the 4X multiplying circuit. This pin can also be used as a general purpose I/O pin (PB0).
VOW1 to VOW4 (PA0 to PA3)	Output (I/O)	OSD character digital output pins	These are digital data output pins for OSD characters. The four outputs correspond to VOW1 (B), VOW2 (G), VOW3 (R), and VOW4 (character), respectively. These pins can also be used as general purpose I/O pins (PA0 to PA3).
VOB1, VOB2 (PA4, PA5)	Output (I/O)	OSD Framing digital output pins	These are output pins for the digital framing data of OSD characters. These pins can also be used as general purpose I/O pins (PA4 and PA5).

Chapter 2 Description of Internal Functions

2

2-1 Memory Map

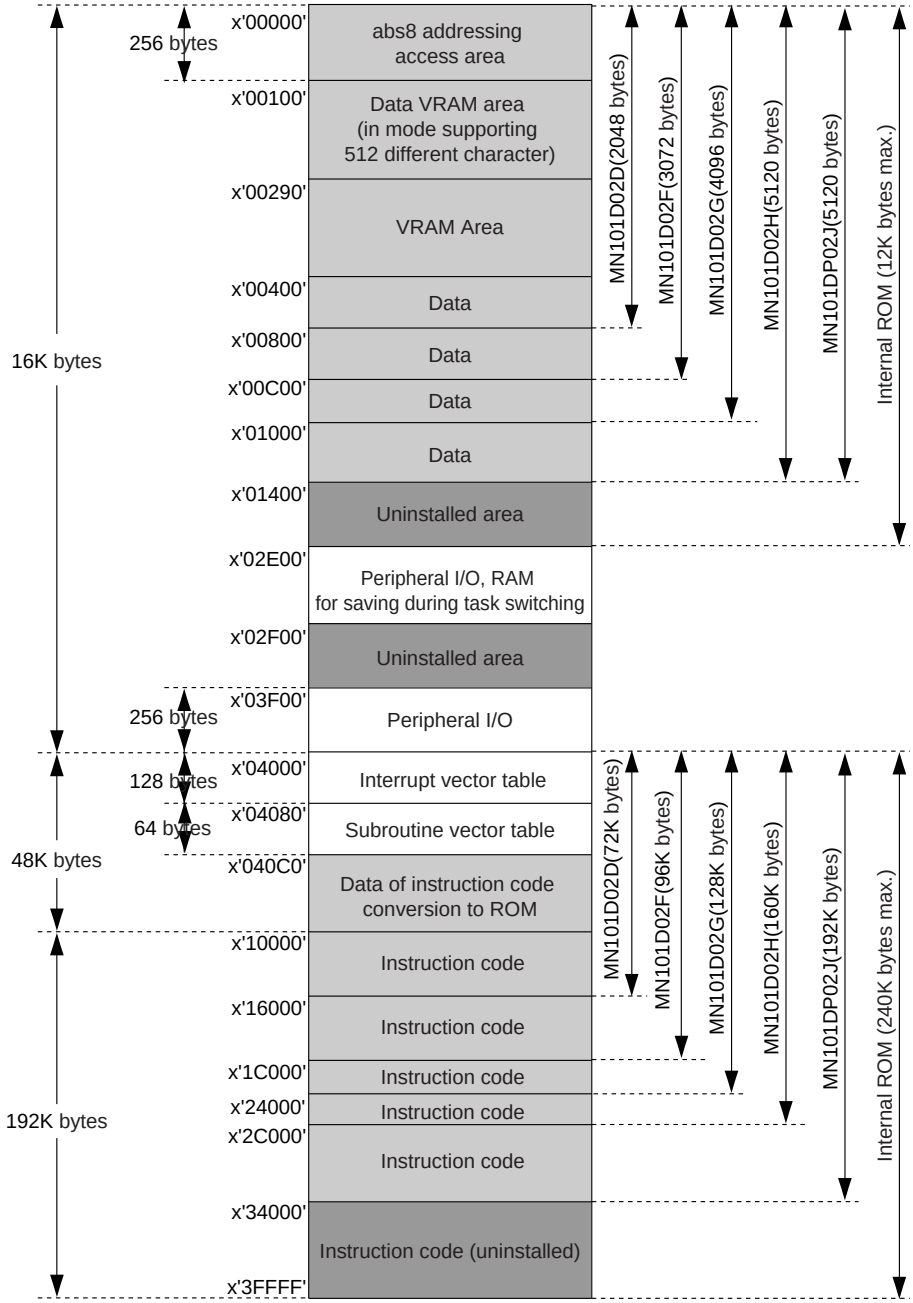


Figure 2-1-1 Memory Map

2-2 System Clock Definition

- Source oscillations (fosc, fxi)

  High speed ..... fosc

  Low speed ..... fxi (≅32kHz: can be used only when the SXI pin is High)

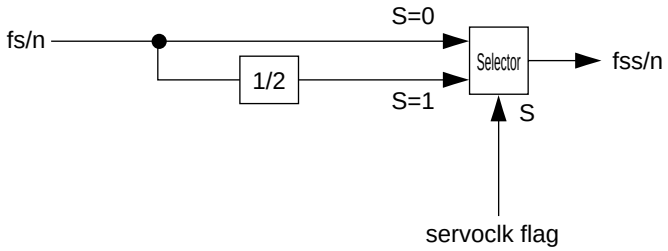
- System clock (fs)

  NORMAL Mode .... fs = fosc/2, fosc/1024

  SLOW Mode ..... fs = fxi/2

  * Can be selected by the CPUM register.

- Clock for servo and peripherals (fss)



The relation between fs and fss is shown in the above figure.

Q:	What type of signal is fss/8? (In the NORMAL mode of fosc/2, and when fosc = 14.32MHz)
A:	Since fs = fosc/2 and fosc = 14.32MHz, (1) When servoclk = 0: fss/8 = fs/8 = fosc/16 = 895kHz (2) When servoclk = 1: fss/8 = 1/ × fs/8 = fosc/32 = 447kHz

Chapter 3 Port Functions

3

3-1 Port Allocation

The standard unit for each of these I/O ports is 8 bits and Port 0 through Port C contain a total of 78 pins. The type column in the I/O Port Allocation table below indicates whether input and/or output is possible when used as a general-purpose port.

Table 3-1-1 I/O Port Allocation Table (1/2)

Port Name	Pin Name	Dual Function	Type	R/W	Dual Function Switching	I/O Switching
P0	P00	NSBO0 output	I/O	R/W	SC0MD3 x'3F9A'	P0DIR x'3F2E' P0OUT x'3F20' P0IN x'3F21'
	P01	NSBI0 input	I/O	R/W		
	P02	NSBT0 I/O	I/O	R/W	SC0MD3 x'3F9A'	
	P04	NRST input	O	W		
	P05	NSBO1 output	I/O	R/W	SIC1 x'3F9E'	
	P06	NSBI1 input	I/O	R/W		
	P07	NSBT1 I/O	I/O	R/W	SIC1 x'3F9E' P0DRV x'3F3C'	
P1	P11	SBUFD1 output PWM2 output	I/O	R/W	SPGCNT x'3F7F' P1SEL x'3F38'	P1DIR x'3F2F' P1 x'3F22' Pull up control P1PUP x'3F3D'
	P12	SBUFD2 output PWM3 output	I/O	R/W	SPGCNT x'3F7F' P1SEL x'3F38'	
	P13	SBUFD3 output FF15 output	I/O	R/W	SPGCNT x'3F7F' TRCNT x'3F85'	
	P14	SBUFD4 output TC3O output	I/O	R/W	SPGCNT x'3F7F' P1SEL x'3F38'	
	P15	SBUFD5 output TC4I input	I/O	R/W	SPGCNT x'3F7F'	
	P16	SBUFD6 output OSDH output XDSCK output	I/O	R/W	SPGCNT x'3F7F' PCSTA x'3FD3' P1SEL x'3F38'	
	P17	SBUFD7 output OSCDIV output OSDV output XDSDAT output	I/O	R/W	PCSTA x'3FD3' SPGCNT x'3F7F' P1SEL x'3F38'	
P2	P20	VSYNC input	I/O	R/W		P2DIR x'3F30' P2 x'3F23'
	P21 P22 P23 P24 P25 P26 P27	OSCO2 output OSCI2 input XO output XI input PWM14 output SCK I/O SDA I/O	I/O I/O I/O I/O I/O I/O I/O	R/W R/W R/W R/W R/W R/W R/W	P2SEL x'3F39' P2SEL x'3F39' P2SEL x'3F39' P2SEL x'3F39' P2SEL x'3F39' P2SEL x'3F39' P2SEL x'3F39'	
P4	P40	HBUFD0 output	I/O	R/W	HOCRCNT x'3F7C'	P4OUT x'3F24' P4IN x'3F25'
	P42	HBUFD2 output	I/O	R/W	HOCRCNT x'3F7C'	
	P44	HBUFD4 output	I/O	R/W	HOCRCNT x'3F7C'	
	P46	HBUFD6 output	I/O	R/W	HOCRCNT x'3F7C' BUZCNT x'3F3F'	

Table 3-1-1 I/O Port Allocation Table (2/2)

Port Name	Pin Name	Dual Function	Type	R/W	Dual Function Switching	I/O Switching
P5	P50 P51 P52 P53 P54	KEYIRQ0 input KEYIRQ1 input KEYIRQ2 input KEYIRQ3 input KEYIRQ4 input	I/O I/O I/O I/O I/O	R/W R/W R/W R/W R/W		P5DIR x'3F31' P5 x'3F26'
P6	P 60 P 61 P 62 P 63 P 64 P 65	IRQ4 input IRQ3 input IRQ2 input IRQ1 input IRQ0 input DENV input	I/O I/O I/O I/O I I/O	R/W R/W R/W R/W R R/W		P6DIR x'3F32' P6 x'3F27'
	P 66	HAMP output	I/O	R/W	P6SEL x'3F3A'	
	P 67	ROTA output	I/O	R/W	P6SEL x'3F3A'	
P7	P 70 P 71 P 72 P 73 P 74 P 75 P 76 P 77	None	I/O I/O I/O I/O I/O I/O I/O I/O	R/W R/W R/W R/W R/W R/W R/W R/W		P7DIR x'3F33' P7 x'3F28' Pull-up control P7PUP x'3F3E'
P8	P 80 P 81 P 82 P 83 P 84 P 85 P 86 P 87	AD0 input AD1 input AD2 input AD3 input AD4 input AD5 input AD6 input AD7 input	I/O I/O I/O I/O I/O I/O I/O I/O	R/W R/W R/W R/W R/W R/W R/W R/W		P8DIR x'3F34' P8 x'3F29'
P9	P 90 P 91 P 92	TPZ output YPG input PGMM input	I I I	R R R	ANACNT x'3F81'	P9 x'3F2A'
PA	P A0 P A1 P A2 P A3 P A4 P A5 P A6 P A7	VOW1 output (B) VOW2 output (G) VOW3 output (R) VOW4 output (Character) VOB1 output (Frame) VOB2 output (Frame) LCOSI input LCOSO output	I/O I/O I/O I/O I/O I/O I/O I/O	R/W R/W R/W R/W R/W R/W R/W R/W	PASEL x'3F3B'	PADIR x'3F35' PA x'3F2B'
PB	P B0 P B1	FSCLPF FSCI	I/O I/O	R/W R/W		PBDIR x'3F36' PB x'3F2C'
	P B2 P B3	AFCLPF AFCC	I/O I/O	R/W R/W	PCSTA x'3FD3'	
	P B4 P B5 P B6	CVIN2 input CVIN input CVOUT output	I/O I/O I/O	R/W R/W R/W	CLCNT x'3FD1'	
	P B7	HSYNC input	I/O	R/W	PASEL x'3F3B'	
PC	P C0 P C1 P C2 P C3	AD8 input AD9 input AD10 input AD11 input	I/O I/O I/O I/O	R/W R/W R/W R/W		PCDIR x'3F37' PC x'3F2D'

3-2 Port Configuration

3-2-1 Port 0 Configuration

Data is output to pins P00 through P07 by writing output data to the Port 0 Data Output Register (P0OUT: x'3F20', only the W part of R/W). Data is input to pins P00 through P07 by reading the data from the Port 0 Data Input Register (P0IN: x'3F21', R).

(1) I/O direction control for Port 0 is performed individually for each bit by the Port 0 Direction Control Register (P0DIR: x'3F2E'). A data setting of '0' specifies input, and '1' specifies output.

(2) The I/O circuitry of pins P00 to P02 and P05 to P07 is constructed from CMOS, with a Schmitt circuit at the inputs. P04 is a special pin used to reset the CPU (NRST), has an n-channel open-drain configuration with a built-in pull-up resistor at the output. A Schmitt circuit is built into the P04 input.

(3) Dual functions of the P00 to P07 pins are listed in the table below.

Table3-2-1

Port Name	Dual Function	Register that Switches Pin
P00	NSBO0	Bit 2 of SC0MD3
P01	NSBI0	—
P02	NSBT0	Bit 0 of SC0MD3
P04	NRST	—
P05	NSBO1	Bit 0 of SIC1
P06	NSBI1	—
P07	NSBT1	Bit 0 of SIC1, Bit 3 of P0DRV

Refer to Figure 10-2-2 for the configuration of the reset pin (NRST/P04).

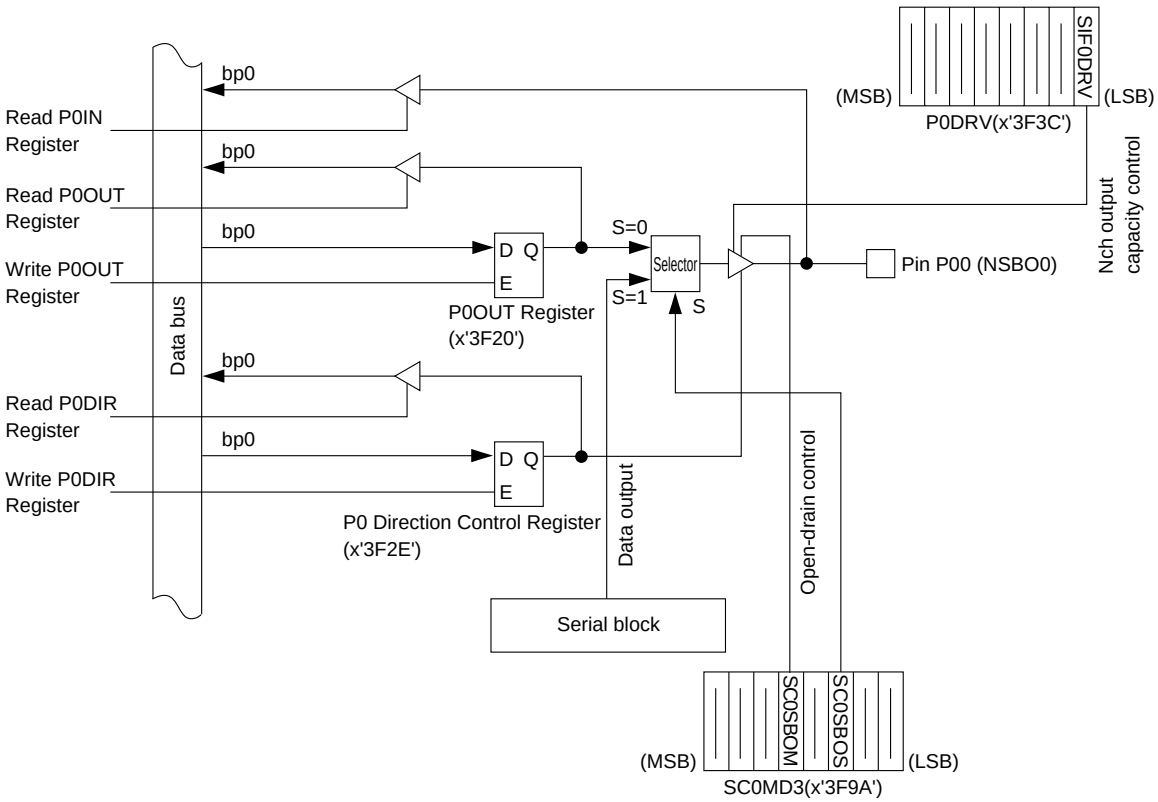


Figure 3-2-1 P00 Configuration

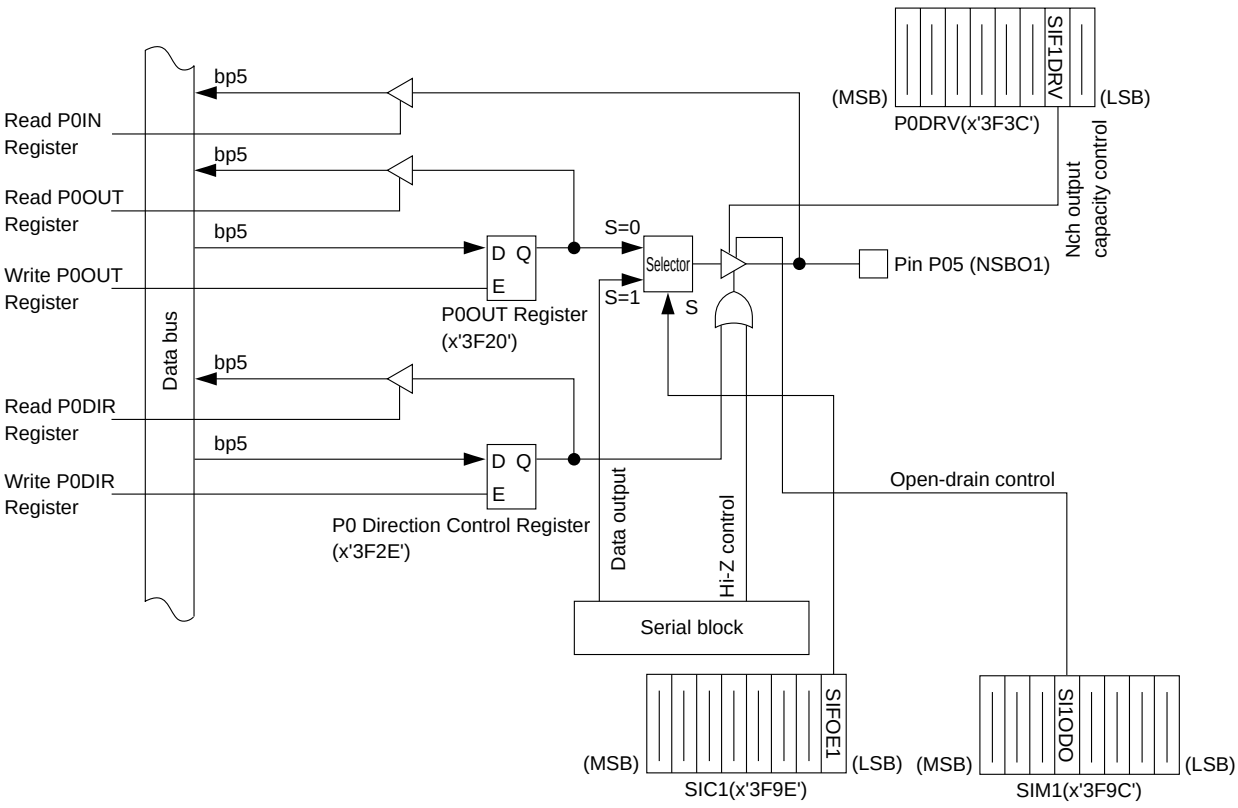


Figure 3-2-2 P05 Configuration

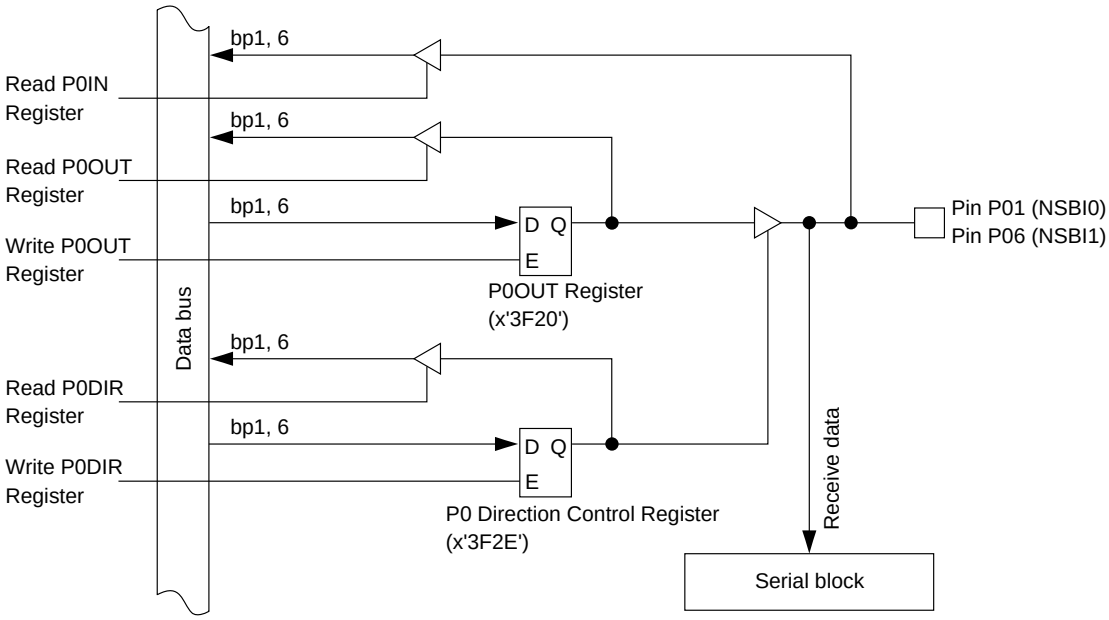


Figure 3-2-3 P01 and P06 Configuration

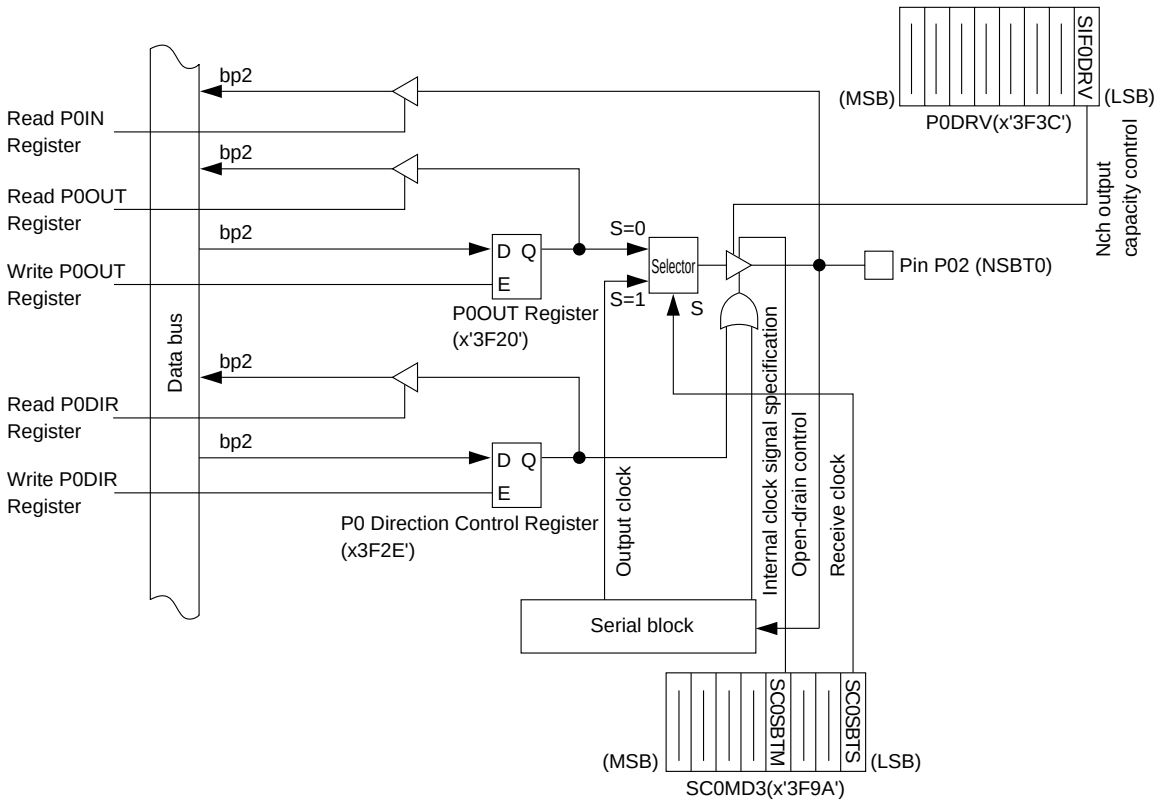


Figure 3-2-4 P02 Configuration

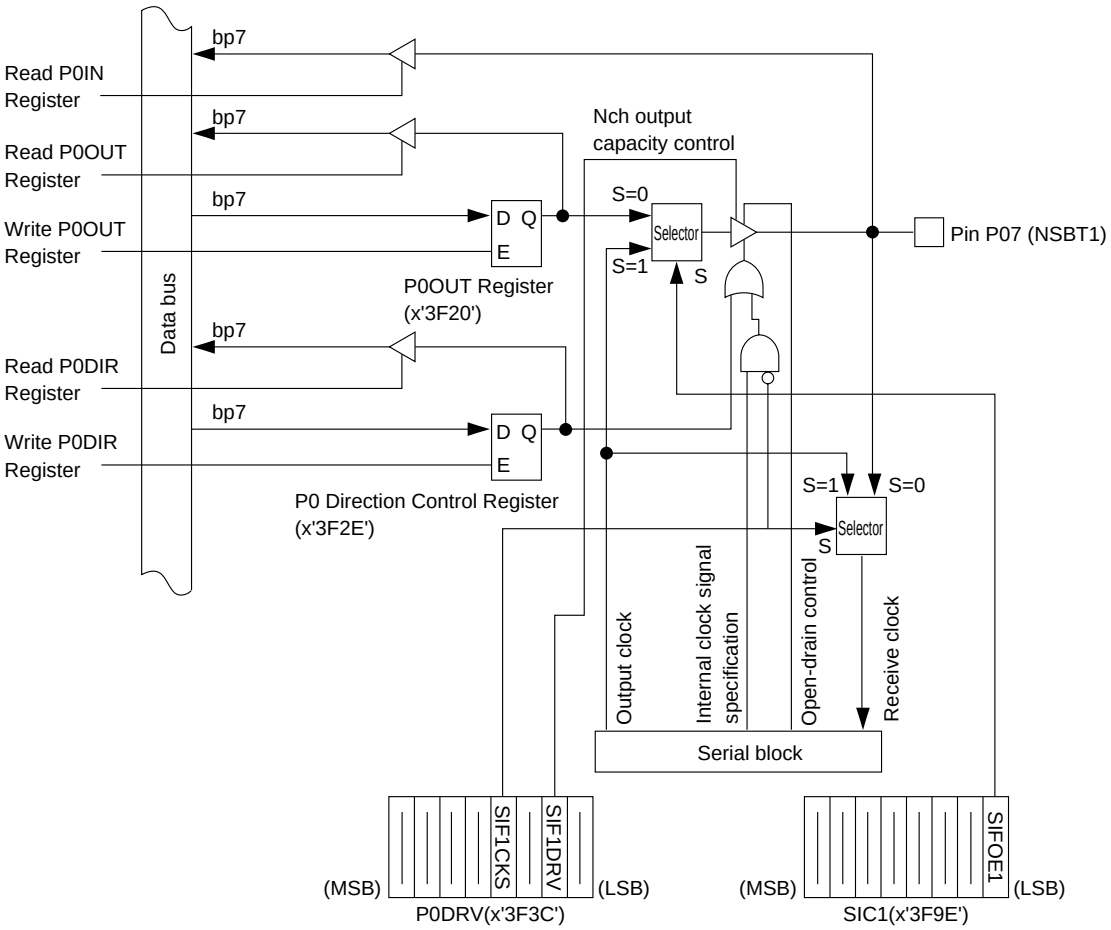


Figure 3-2-5 P07 Configuration

[☞ See Fig. 10-2-2 for the construction of P04.]

3-2-2 Port 1 Configuration

Pins P11 through P17 also function as Synchronous Output Pins (SPG output) SBUFD1 through SBUFD7.
Data is output to pins P11 through P17 by writing output data to the Port 1 Data (Synchronous Output Buffer) Register (P1(SPGBUF): x'3F22', R/W). Data is input to pins P11 through P17 by reading the data from the Synchronous Output Buffer Register.

(1) If P11 through P17 are used as general-purpose I/O pins, each bit can be individually switched for use as an SPG output by the SPG Control Register (SPGCNT: x'3F7F', R/W). A data setting of '0' specifies input, and '1' specifies output.

I/O direction control for Port 1 is performed individually for each bit by the Port 1 (SPG) Direction Control Register (P1(SPG)DIR: x'3F2F', R/W). A data setting of '0' specifies input, and '1' specifies output.

(2) The I/O circuitry of pins P11 to P17 is constructed from CMOS, with a Schmitt circuit at the inputs.

(3) Dual functions of the P11 to P17 pins are listed in the table below.

Table3-2-2

Port Name	Dual Function	Register that Switches Pin
P11	PWM2 SBUF1	Bit 1 of P1SEL SPGCNT
P12	PWM3 SBUF2	Bit 2 of P1SEL SPGCNT
P13	FF15 SBUF3	Bit 1 of TRCNT SPGCNT
P14	TC3O SBUF4	Bit 4 of P1SEL SPGCNT
P15	TC4I SBUF5	— SPGCNT
P16	SBUF6 OSDH XDSCK	SPGCNT Bit 7 of PCSTA Bit 6 of P1SEL
P17	OSCDIV SBUF7 OSDV XDSDAT	— SPGCNT Bit 7 of PCSTA Bit 6 of P1SEL

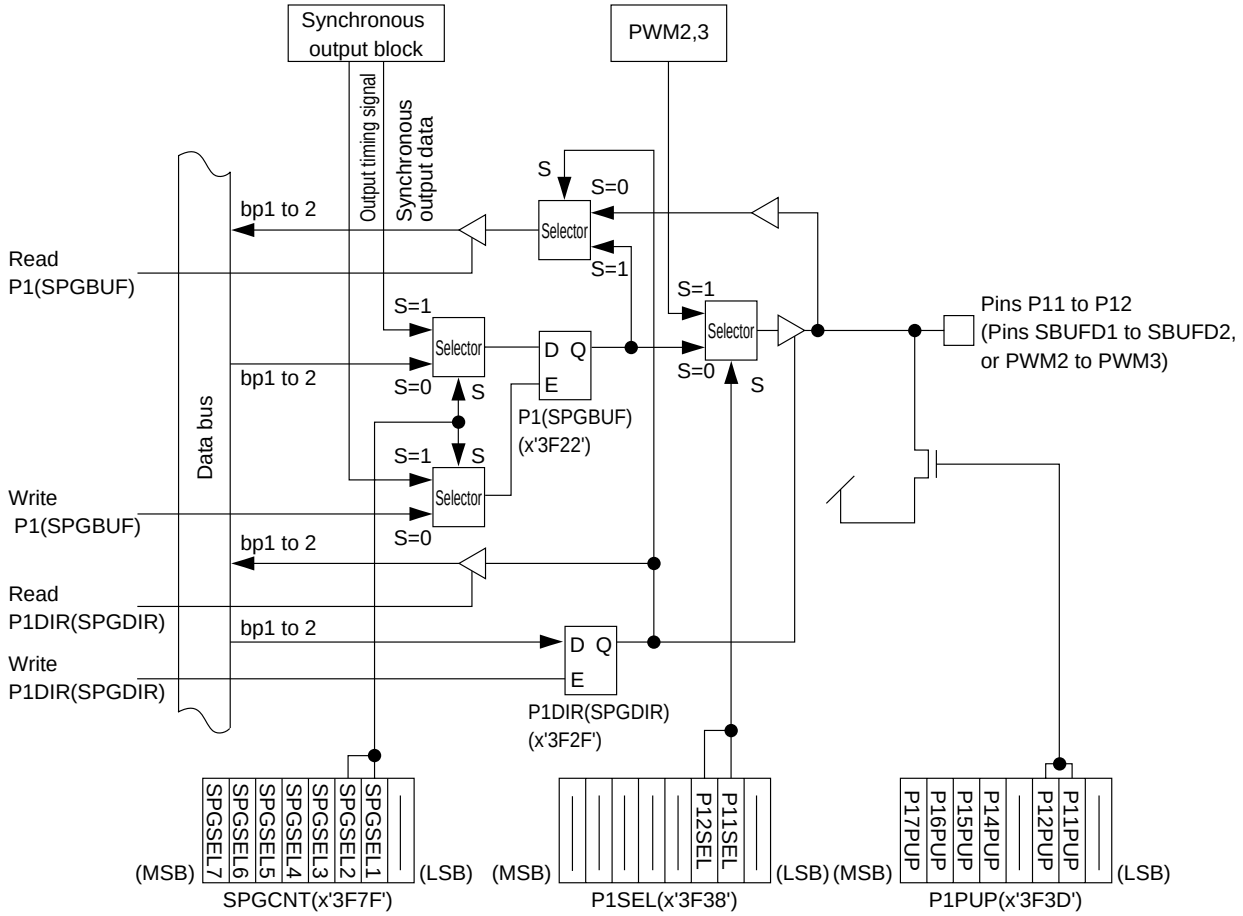


Figure 3-2-6 P11 to P12 Configuration

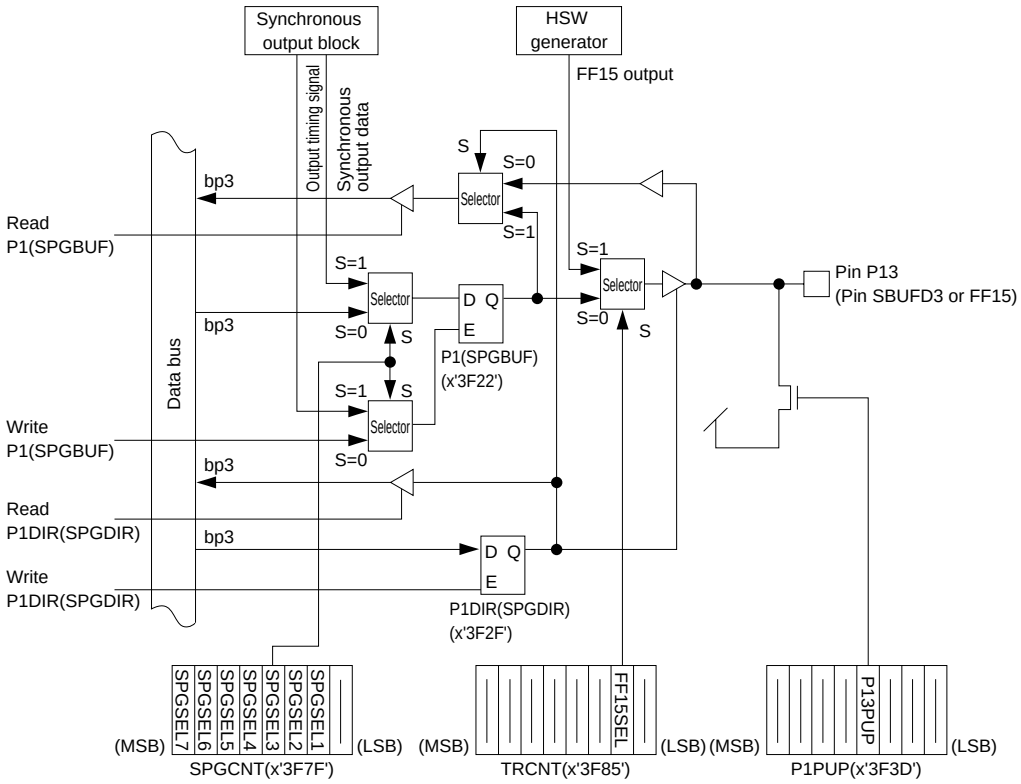


Figure 3-2-7 P13 Configuration

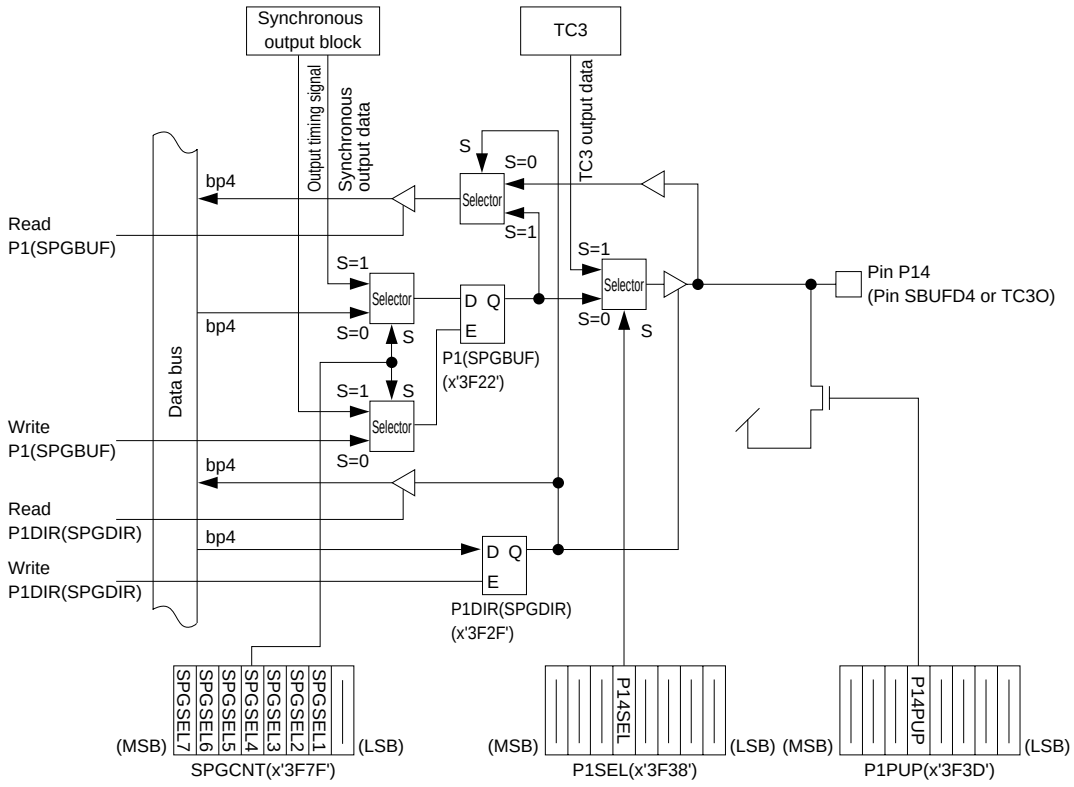


Figure 3-2-8 P14 Configuration

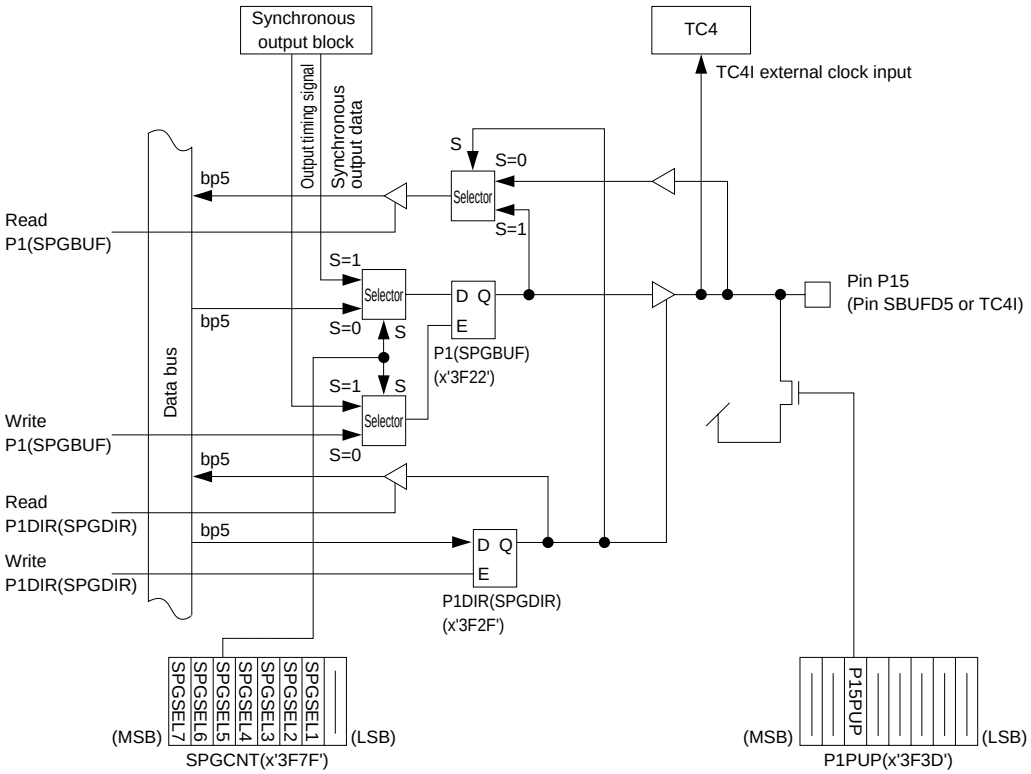
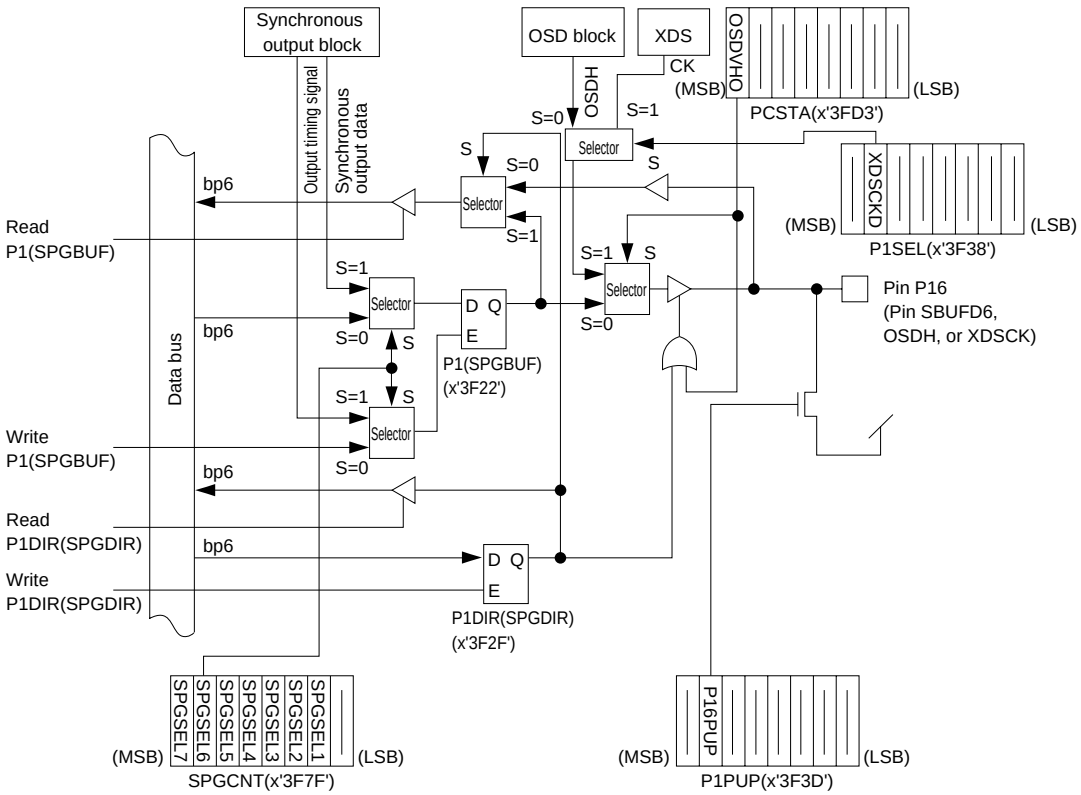


Figure 3-2-9 P15 Configuration



* See the list of port 16 settings in the next page.

Figure 3-2-10 P16 Configuration

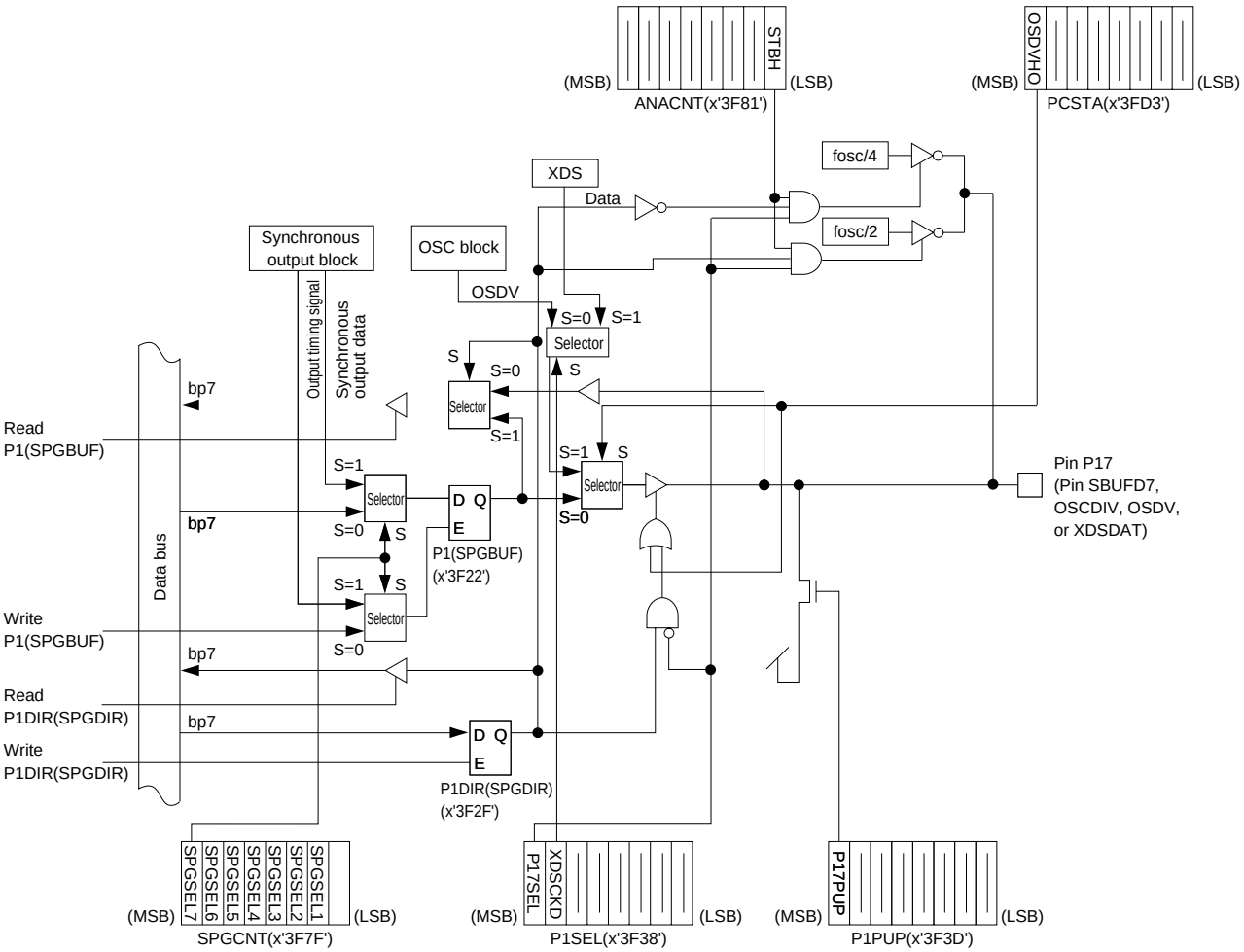


Figure 3-2-11 P17 Configuration

List of P17 settings

PCSTA x'3FD3'bp7 OSDVHO	P1SEL x'3F38' bp7 P17SEL	P1SEL x'3F38' bp6 XDSCKD	SPGCNT x'3F7F'bp7 SPGSEL7	P1DIR x'3F2F'bp7 P1DIRT	P1 x'3F22'bp7 P17	Function of P17	Comments
1	-	0	-	-	-	OSDV output	
	-	1	-	-	-	XDSDAT output	
0	1	-	-	1	-	fosc/2 (2Vpp) output	Only when STBH='1'
		-	-	0	-	fosc/4 (1Vpp) output	
0	0	-	0	1	0/1	Normal data output	
		-	1	1	-	Synchronous data output	
0	0	-	-	0	-	Port 17 data input	

List of P16 settings

PCSTA x'3FD3'bp7 OSDVHO	P1SEL x'3F38'bp6 XDSCKD	SPGCNT x'3F7F'bp6 SPGSEL7	P1DIR x'3F2F'bp6 P1DIR6	P1 x'3F22'bp6 P16	Function of P16	Comments
1	0	-	-	-	OSDH output	
	1	-	-	-	XDSDCK output	
0	-	0	1	0/1	Normal data output	
	-	1	1	-	Synchronous data output	
0	-	-	0	-	Port 16 data input	

3-2-3 Port 2 Configuration

Port 2 is an 8-bit general-purpose I/O port. Data is output to pins P20 through P27 by writing output data to the Port 2 Data Register (P2: x'3F23', R/W). Data is input to pins P20 through P27 by reading the data from the Port 2 Data Register.

(1) I/O direction control for Port 2 is performed individually for each bit by the Port 2 Direction Control Register (P2DIR: x'3F30', R/W). A data setting of '0' specifies input, and '1' specifies output.

(2) The I/O circuitry of pins P20 to P27 is constructed from CMOS. It is possible to select between the CMOS and TTL levels for terminal P20. Further, a Schmitt circuit is present in the input circuit.

(3) Dual functions of the P20 to P27 pins are listed in the table below.

Table3-2-3

Port Name	Dual Function	Register that Switches Pin
P20	VSYNC	—
P21	OSCO2	Bit 1 of P2SEL
P22	OSCI2	Bit 1 of P2SEL
P23	XO	Bit 3 of P2SEL
P24	XI	Bit 3 of P2SEL
P25	PWM14	Bit 5 of P2SEL
P26	SCK	Bit 6 of P2SEL
P27	SDA	Bit 7 of P2SEL

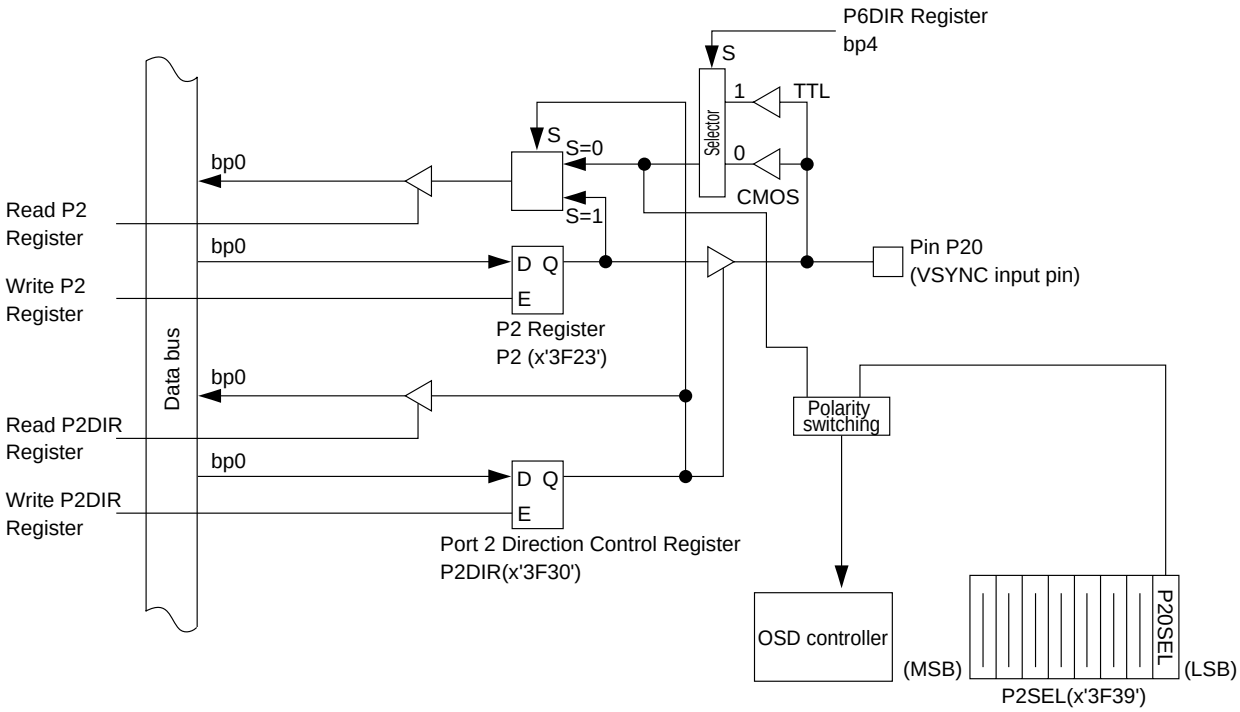


Figure 3-2-12 P20 Configuration

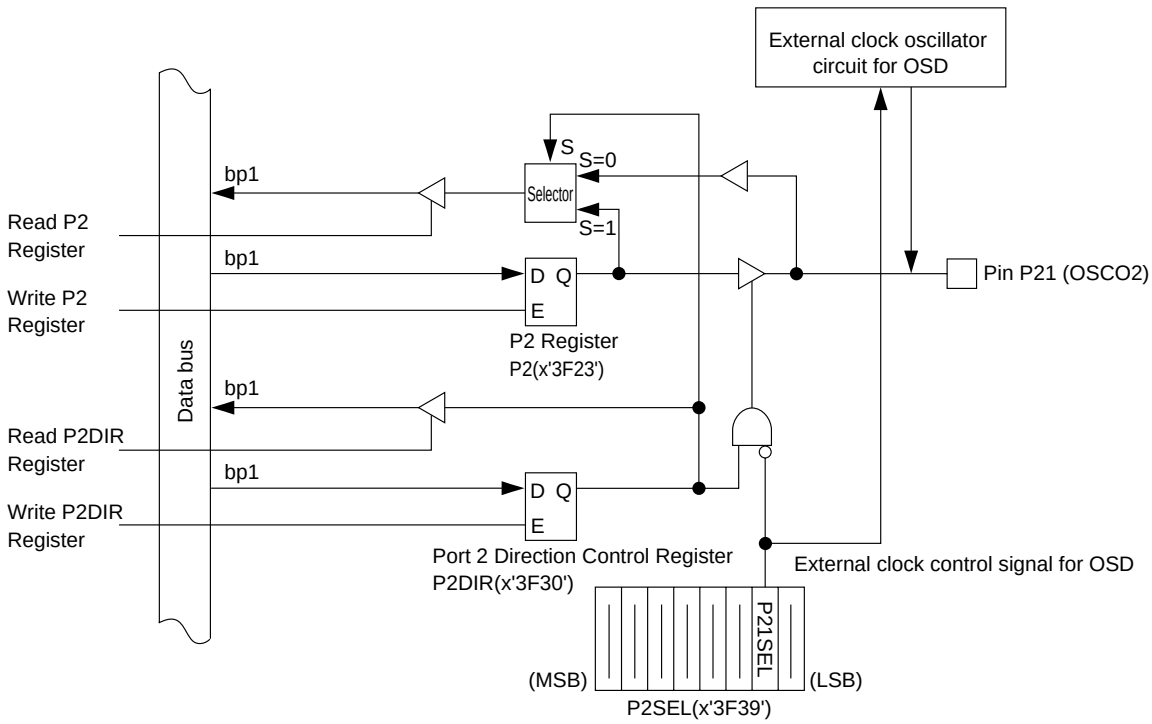


Figure 3-2-13 P21 Configuration

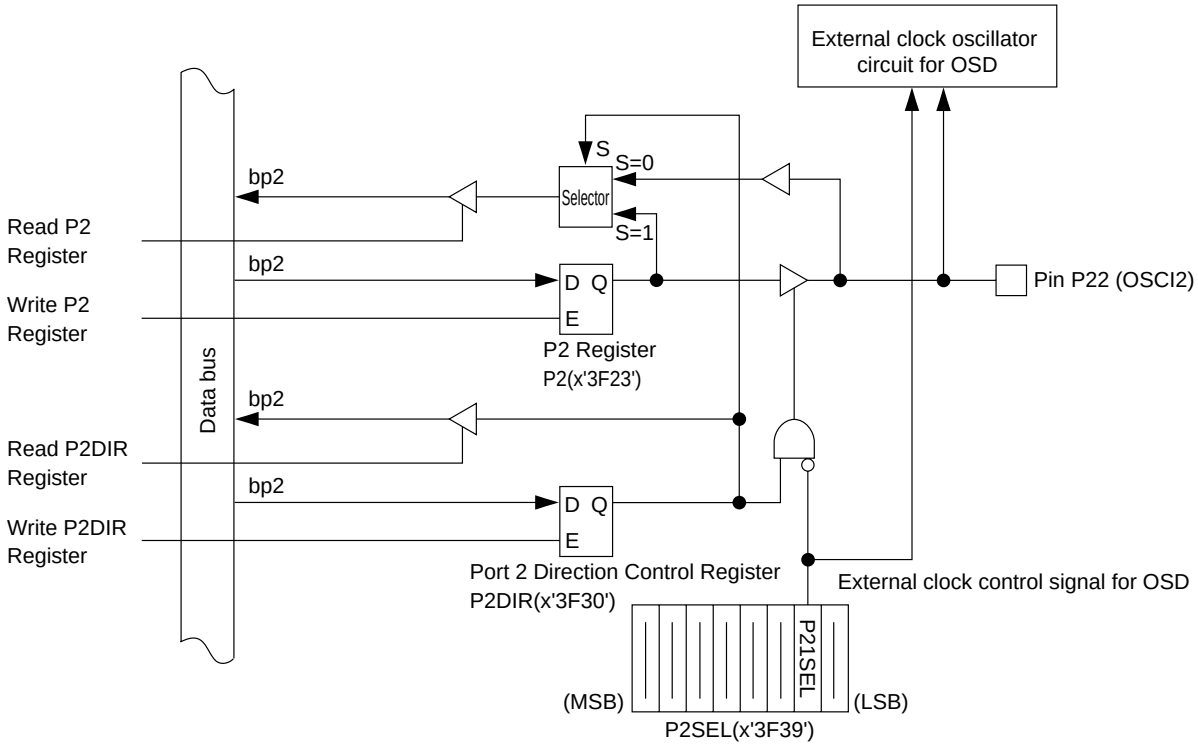


Figure 3-2-14 P22 Configuration

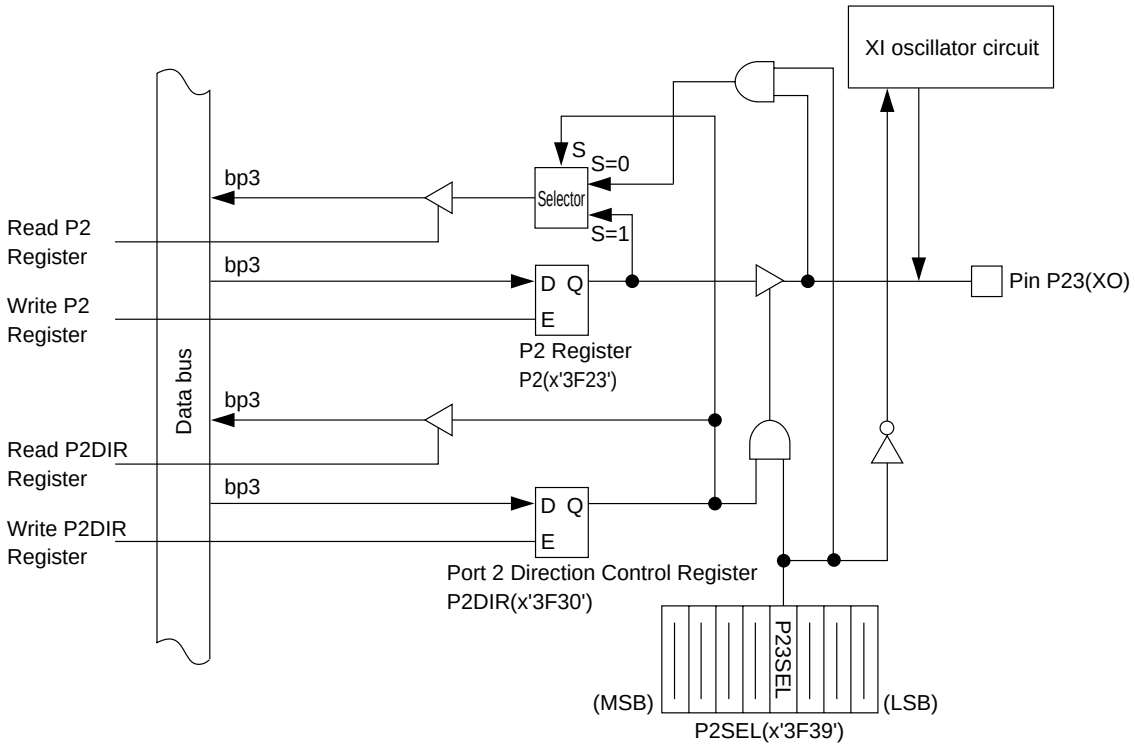


Figure 3-2-15 P23 Configuration

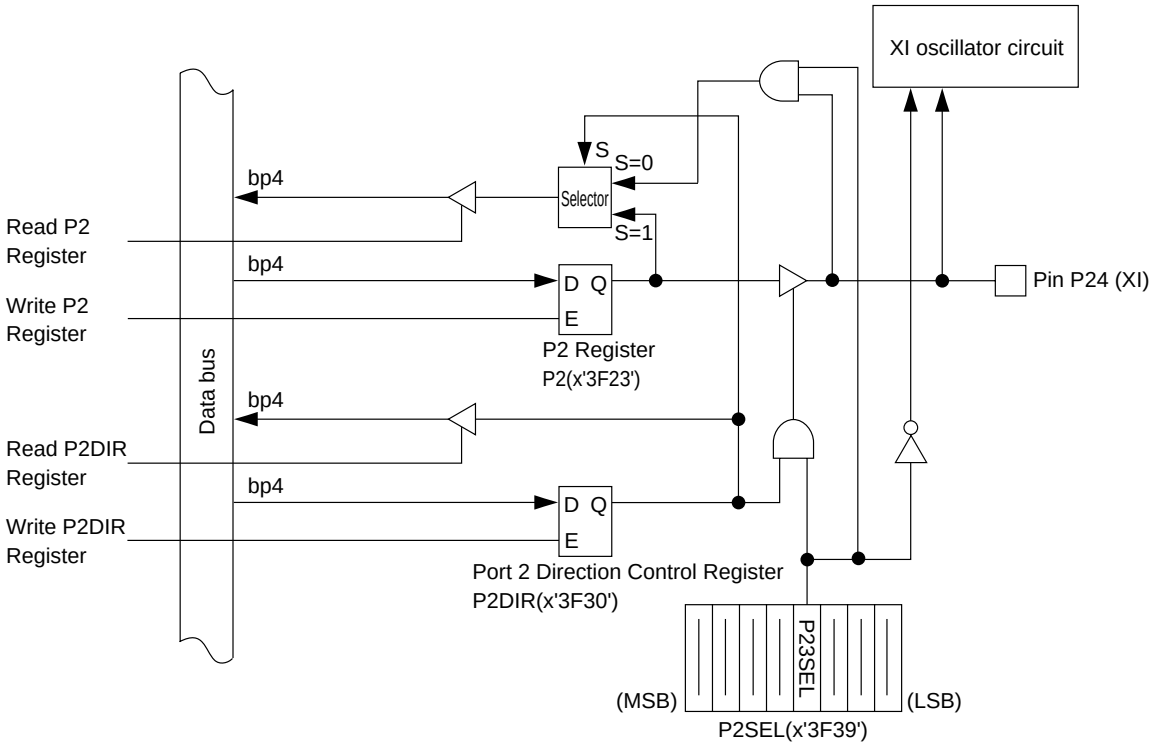


Figure 3-2-16 P24 Configuration

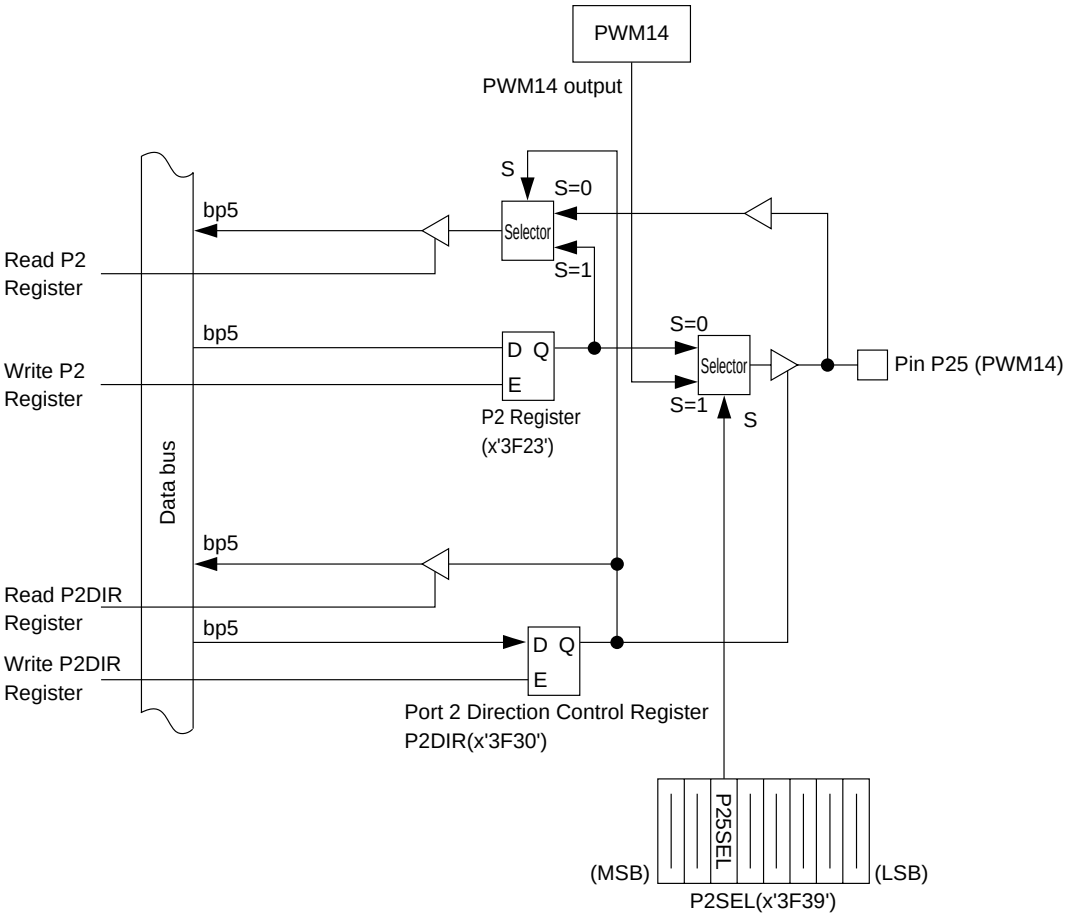


Figure 3-2-17 P25 Configuration

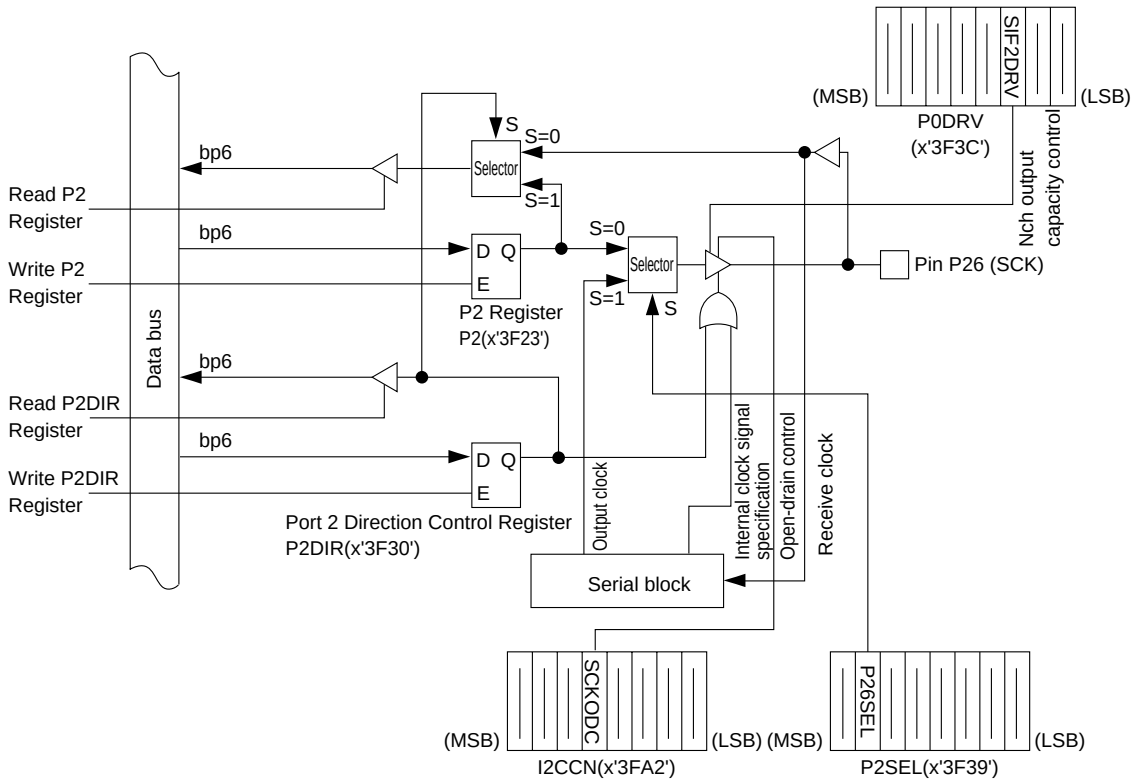


Figure 3-2-18 P26 Configuration

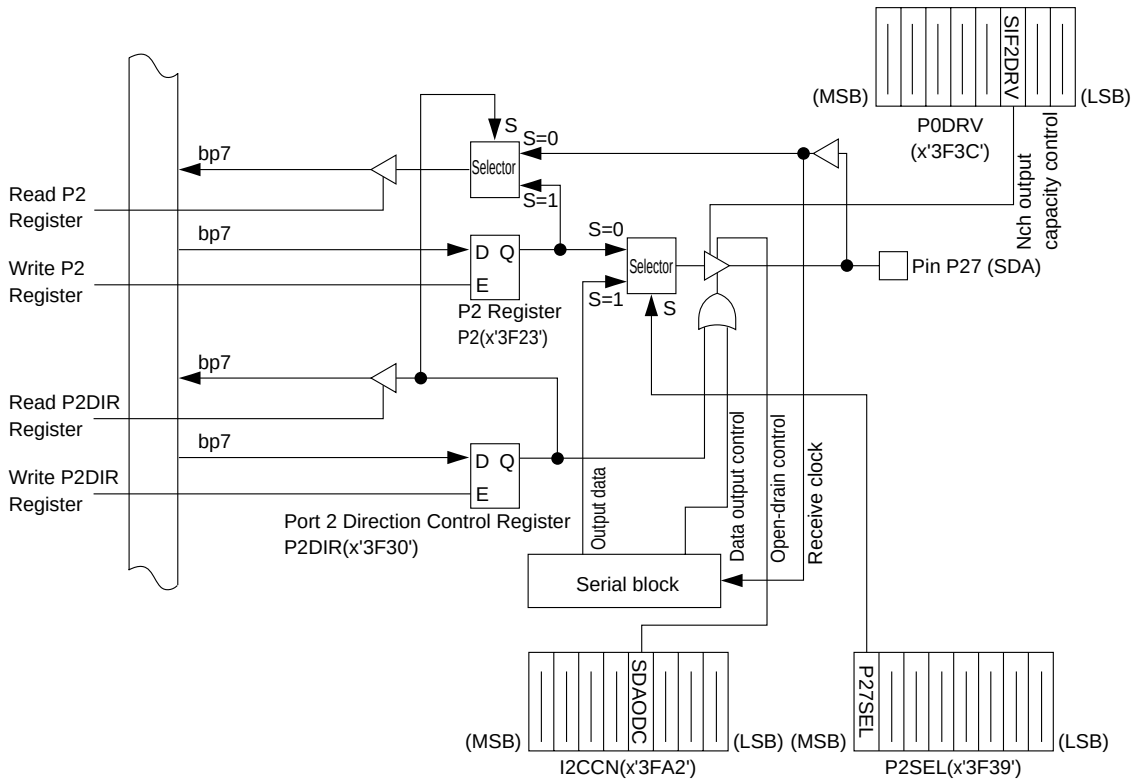


Figure 3-2-19 P27 Configuration

3-2-4 Port 4 Configuration

Pins P40, P44 and P46 also function as pins HBUFD0, HBUFD4 and HBUFD6 of the Synchronous Output (HOCR output). Pin P42 also functions as pin HBUFD2 of the Synchronous Output and as a D/A output pin. Data is output to pins P40, P42, P44 and P46 by writing output data to the Port 4 Data Output Register (Synchronous Output Buffer Register 0) (P4OUT(HOCRBUF0): x'3F24', R/W). Data is input to pins P40, P42, P44 and P46 by reading the data from the Port 4 Data Input Register (Synchronous Output Buffer Register) (P4IN: x'3F25', R).

(1) The HOCRSEL0 flag (bit 1) of the HOCR Control Register (HOCRCNT: x'3F7C', R/W) specifies whether pins P40, P42, P44 and P46 are used as general-purpose I/O pins or as HOCR output pins. If the data setting is '0', general-purpose I/O is specified, and if '1', HOCR output is specified.
However, when the DAEN flag (bit 5) of the HOCR Control Register (HOCRCNT: x'3F7C', R/W) is '1', regardless of the above settings, pin P42 acts as a D/A conversion output pin.
I/O direction control for Port 4 is performed by the Port 4 Data Output Register (Synchronous Output Buffer Register 0) (P4OUT(HOCRBUF0): x'3F24', R/W). Odd bits control the output direction for the corresponding even bits. A data setting of '0' specifies input, and '1' specifies output.

(2) The I/O circuitry for pins P40, P42, P44 and P46 is constructed from CMOS, with a Schmitt circuit at the inputs.

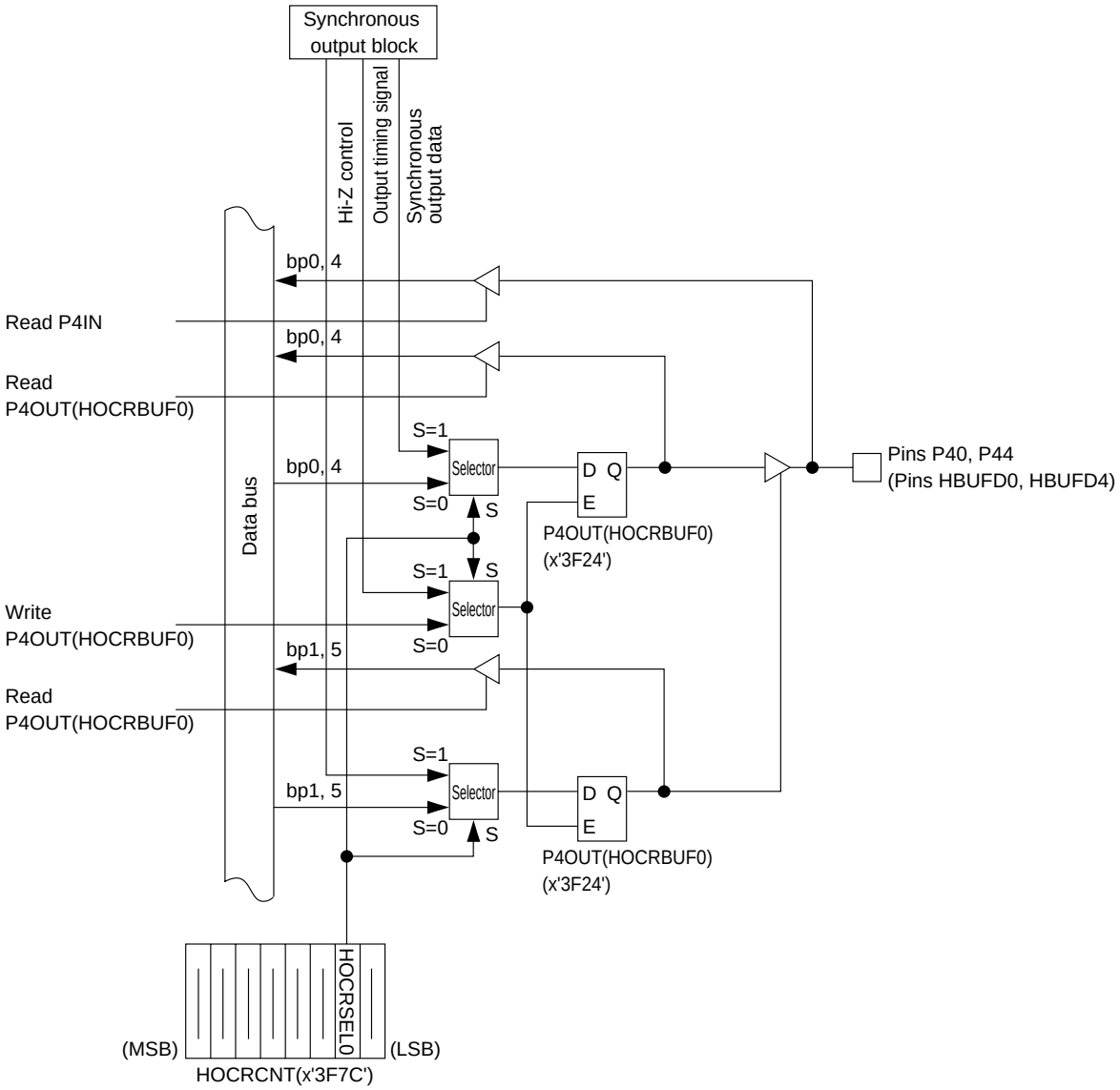


Figure 3-2-20 P40 and P44 Configuration

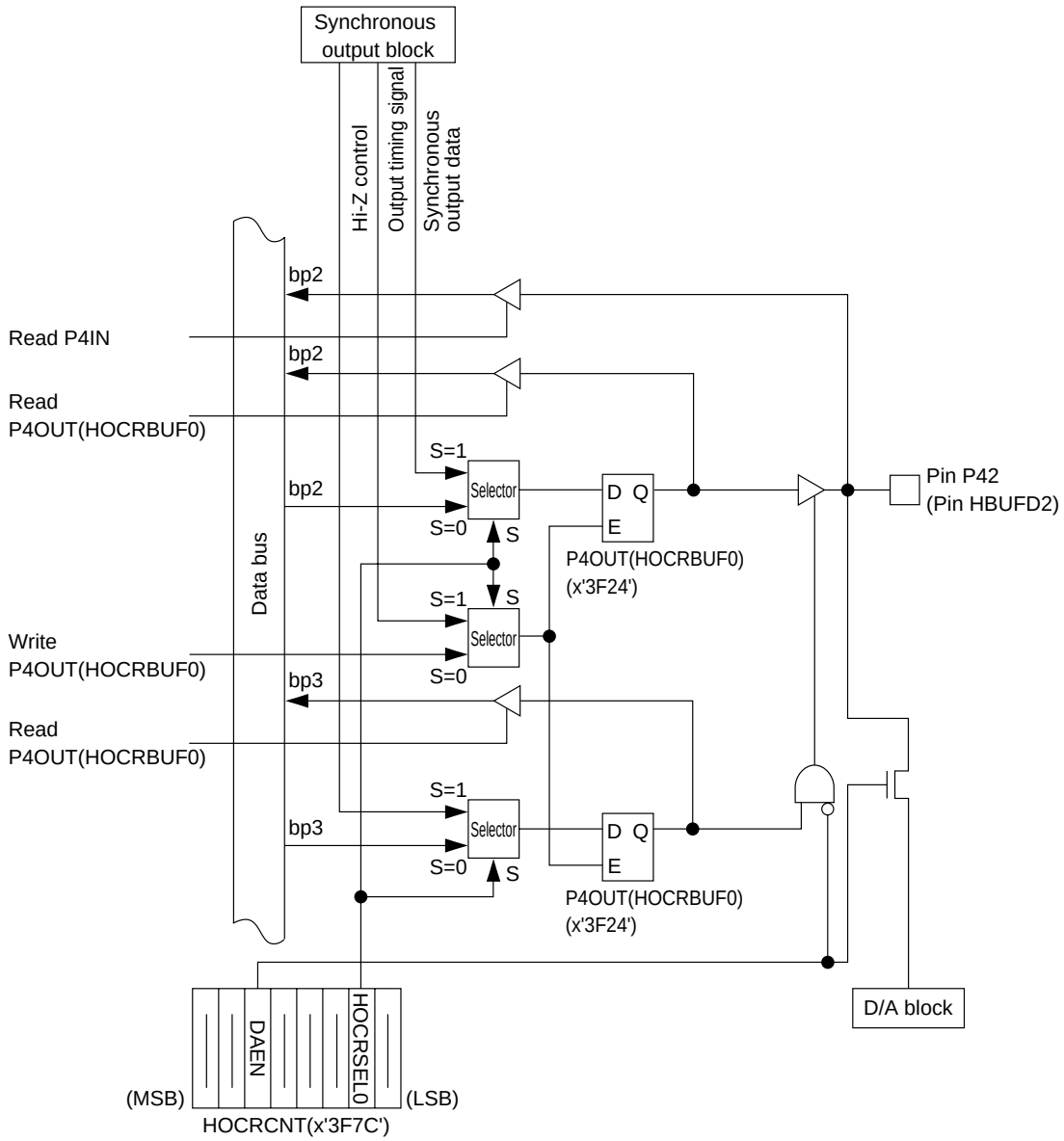


Figure 3-2-21 P42 Configuration

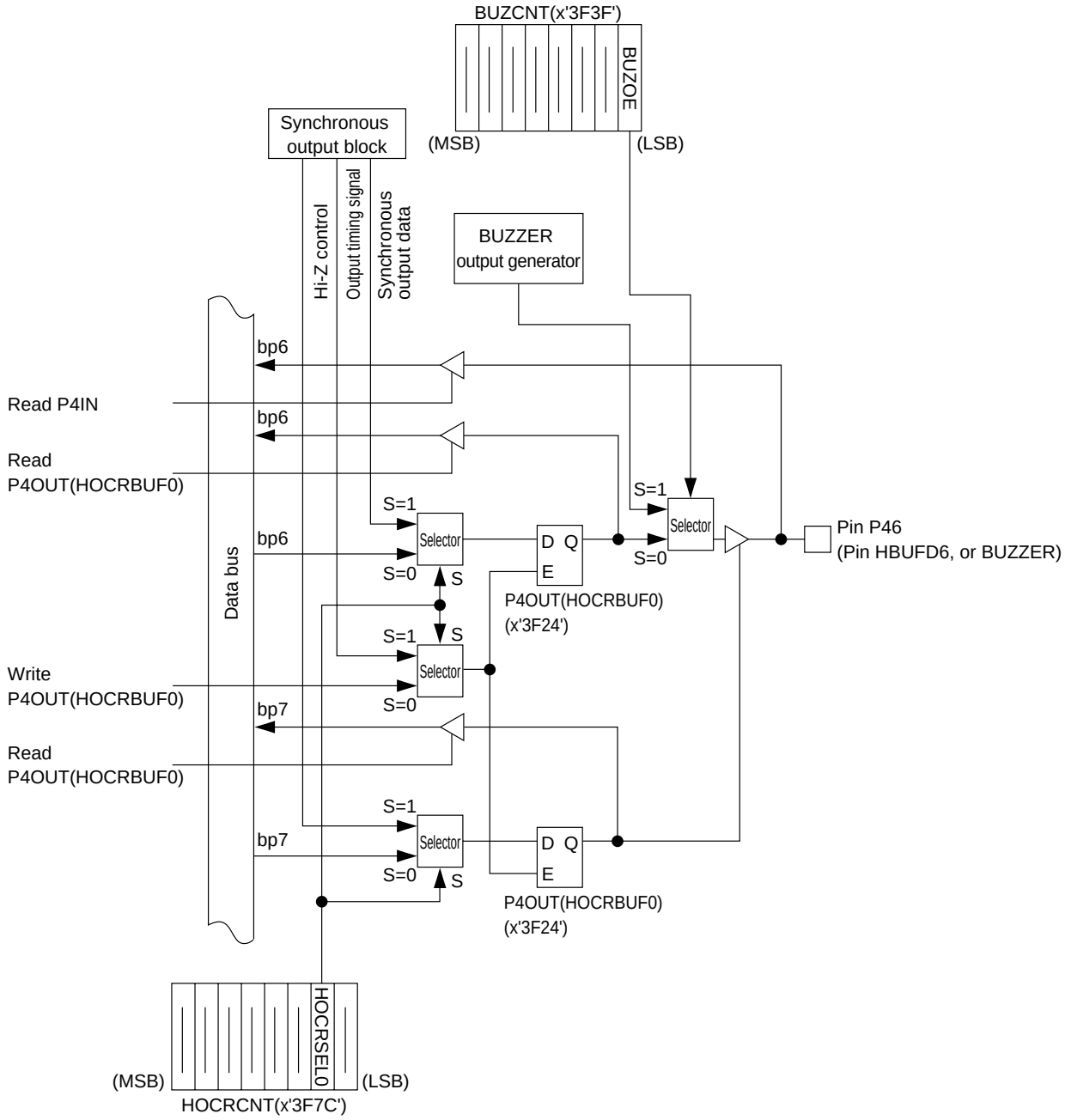


Figure 3-2-22 P46 Configuration

3-2-5 Port 5 Configuration

Port 5 is a 5-bit general-purpose I/O port. Data is output to pins P50 through P54 by writing output data to the Port 5 Data Register (P5: x'3F26', R/W). Data is input to pins P50 through P54 by reading the data from the Port 5 Data Register. Port 5 also functions as the input pins for KEYIRQ0 to KEYIRQ4.

(1) I/O direction control for Port 5 is performed individually for each bit by the Port 5 Direction Control Register (P5DIR: x'3F31', R/W). A data setting of '0' specifies input, and '1' specifies output.

(2) The I/O circuitry of pins P50 to P54 is constructed from CMOS, with a Schmitt circuit at the inputs.

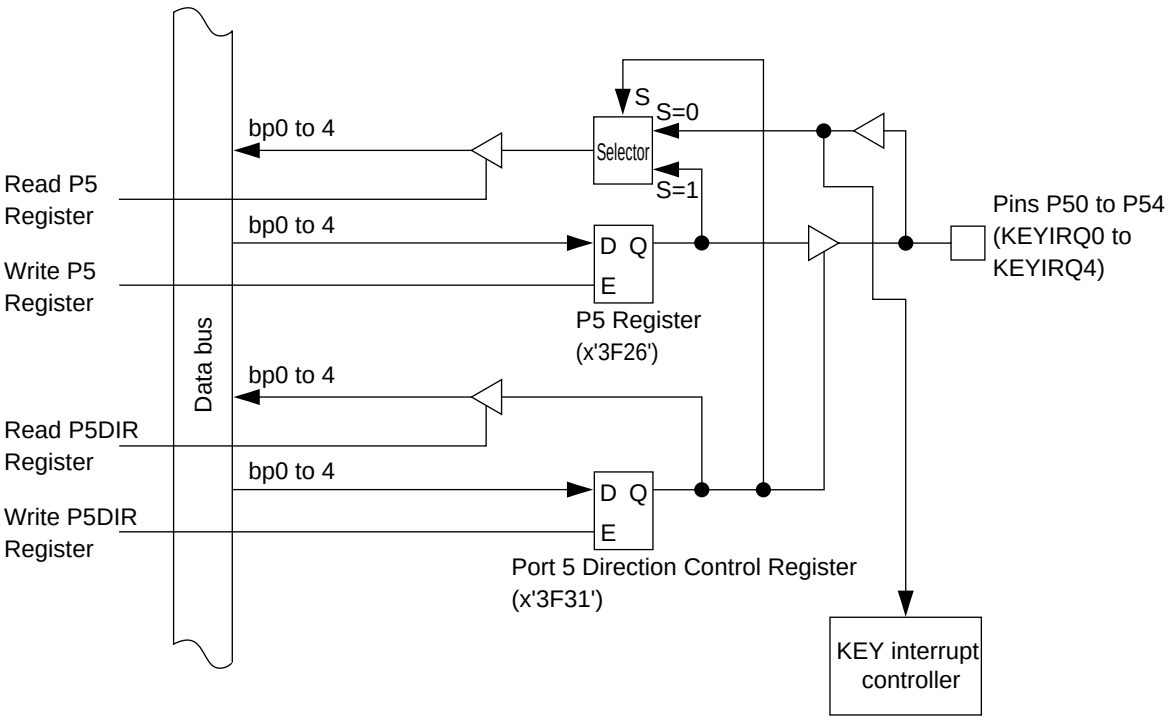


Figure 3-2-23 P50 to P54 Configuration

3-2-6 Port 6 Configuration

Data is output to pins P60 to P67 by writing output data to the Port 6 Data Register (P6: x'3F27', R/W). Data is input to pins P60 to P67 by reading the data from the Port 6 Data Register.

(1) I/O direction control for Port 6 is performed individually for each bit by the Port 6 Direction Control Register (P6DIR: x'3F32', R/W). A data setting of '0' specifies input, and '1' specifies output.

(2) The I/O circuitry of pins P60 to P67 is constructed from CMOS. The P64 pin is an input-only pin. It is possible to select between the CMOS and TTL levels for terminals P60 to P63. Further, a Schmitt circuit is present in the input circuit.

(3) It is possible to input a voltage in the range of 0V to 7V at terminal P64.

(4) Dual functions of the Port 6 pins are listed in the table below.

Table3-2-4

Port Name	Dual Function	Register that Switches Pin
P60	IRQ4	—
P61	IRQ3	—
P62	IRQ2	—
P63	IRQ1	—
P64	IRQ0	—
P65	DENV	—
P66	HAMP	Bit 6 of P6SEL
P67	ROTA	Bit 7 of P6SEL

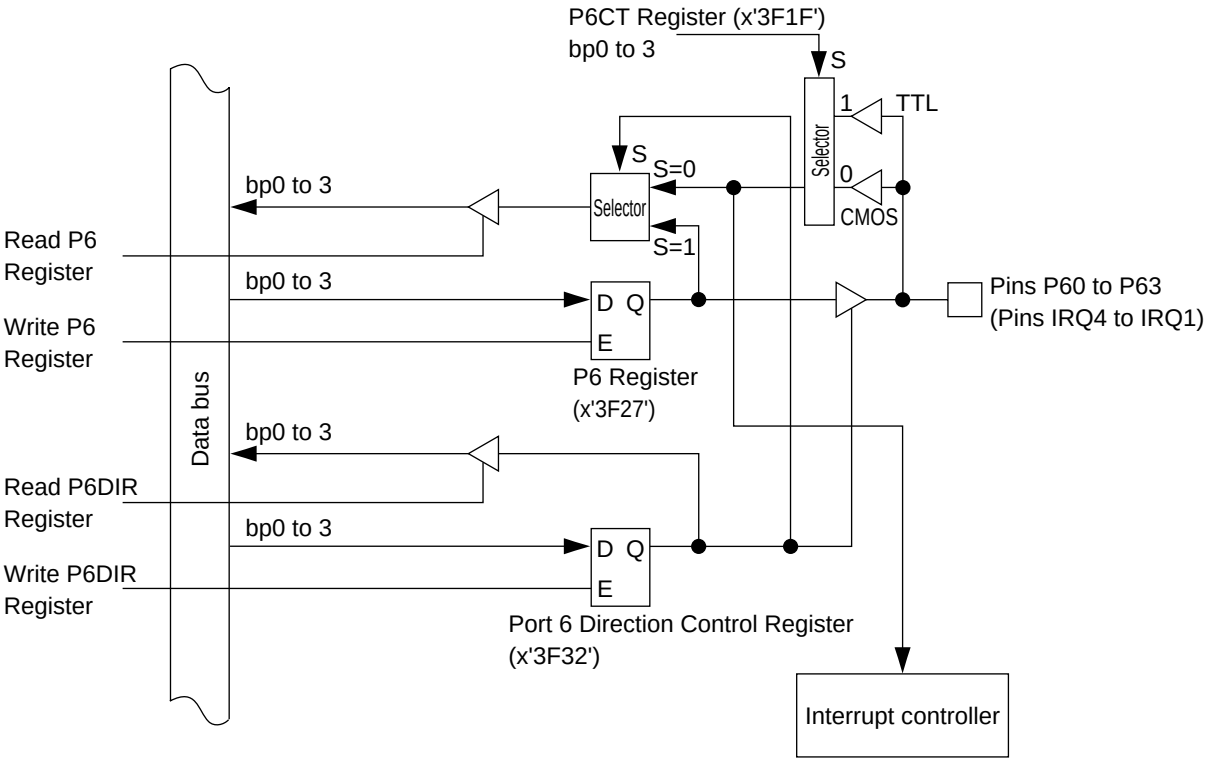


Figure 3-2-24 P60 to P63 Configuration

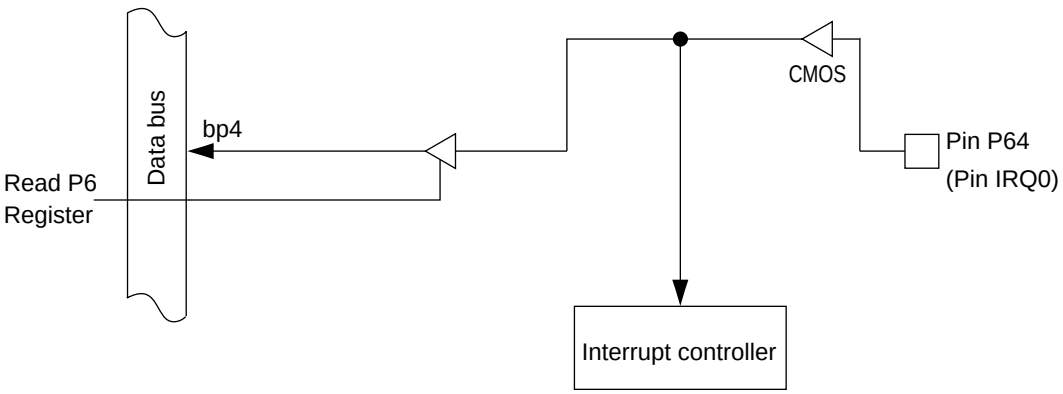


Figure 3-2-25 P64 Configuration

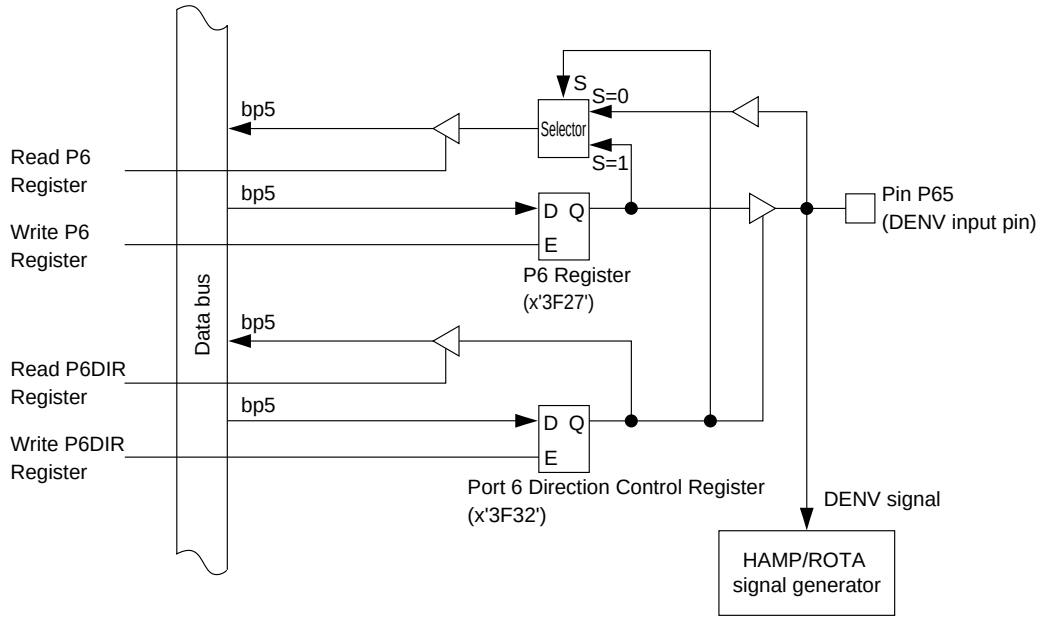


Figure 3-2-26 P65 Configuration

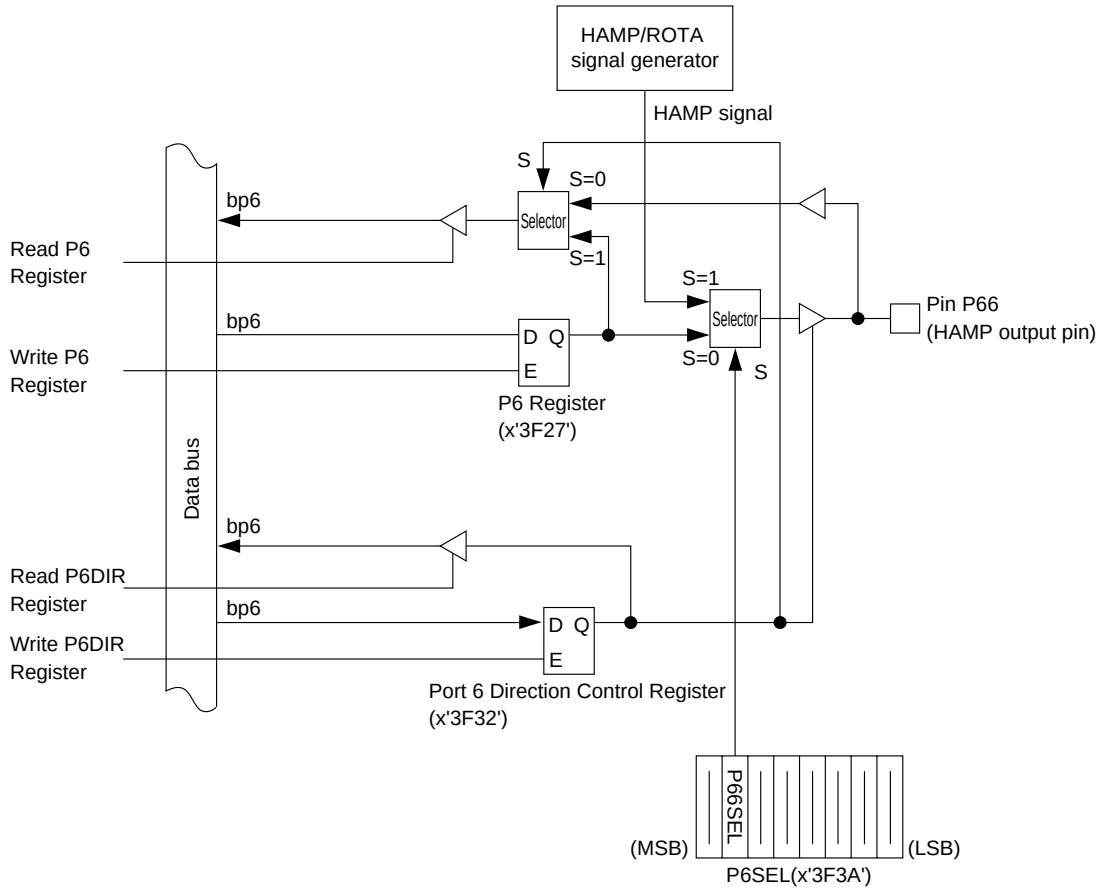


Figure 3-2-27 P66 Configuration

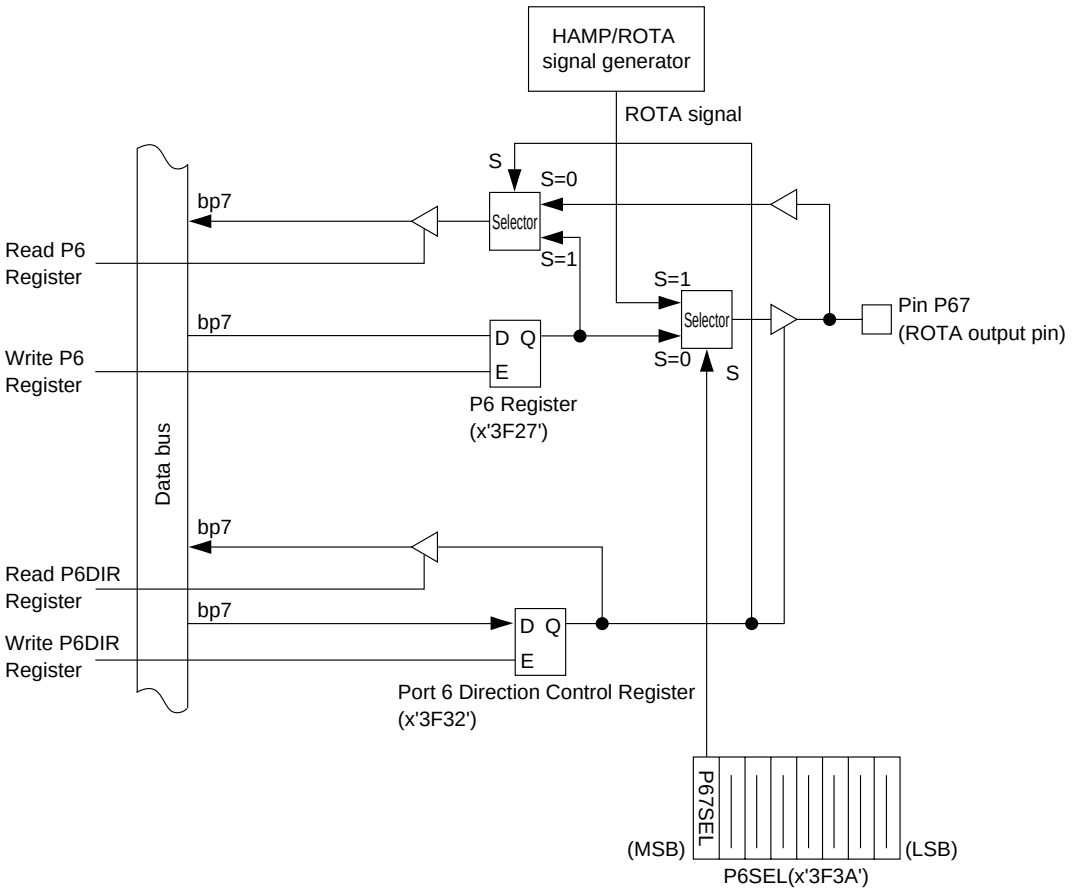


Figure 3-2-28 P67 Configuration

3-2-7 Port 7 Configuration

Data is output to pins P70 to P77 by writing output data to the Port 7 Data Register (P7: x'3F28', R/W). Data is input to pins P70 to P77 by reading the data from the Port 7 Data Register.

(1) I/O direction control for Port 7 is performed individually for each bit by the Port 7 Direction Control Register (P7DIR: x'3F33', R/W). A data setting of '0' specifies input, and '1' specifies output.

(2) The I/O circuitry of pins P70 to P77 is constructed from CMOS, with a Schmitt circuit at the inputs.

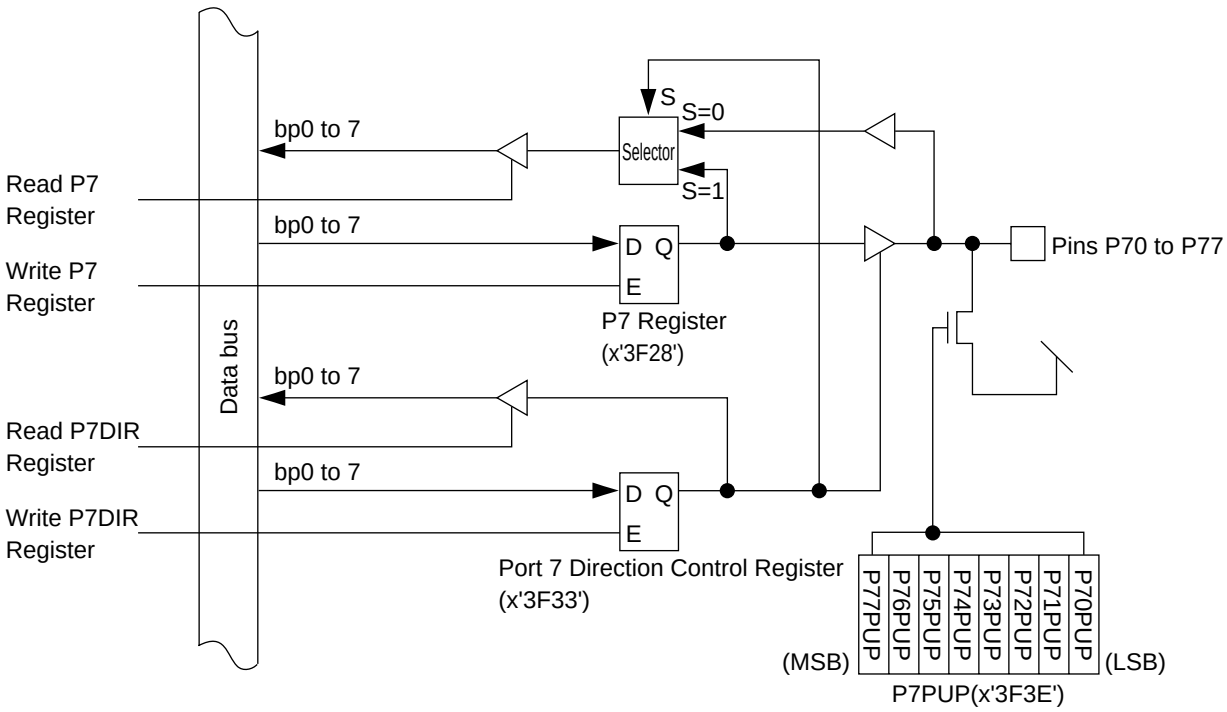


Figure 3-2-29 P70 to P77 Configuration

3-2-8 Port 8 Configuration

Data is output to pins P80 to P87 by writing output data to the Port 8 Data Register (P8: x'3F29', R/W). Data is input to pins P80 to P87 by reading the data from the Port 8 Data Register.

(1) I/O direction control for Port 8 is performed individually for each bit by the Port 8 Direction Control Register (P8DIR: x'3F34', R/W). A data setting of '0' specifies input, and '1' specifies output.

(2) The I/O circuitry of pins P80 to P87 is constructed from CMOS, with a Schmitt circuit at the inputs.

(3) Pins P80 to P87 also function as A/D Conversion input pins.

(4) The input is enabled only during reading out the data register of port 8 (P8:x'3F29'). Therefore, since the input is disabled at all times other than during a read out of the register, no conduction current will be generated.

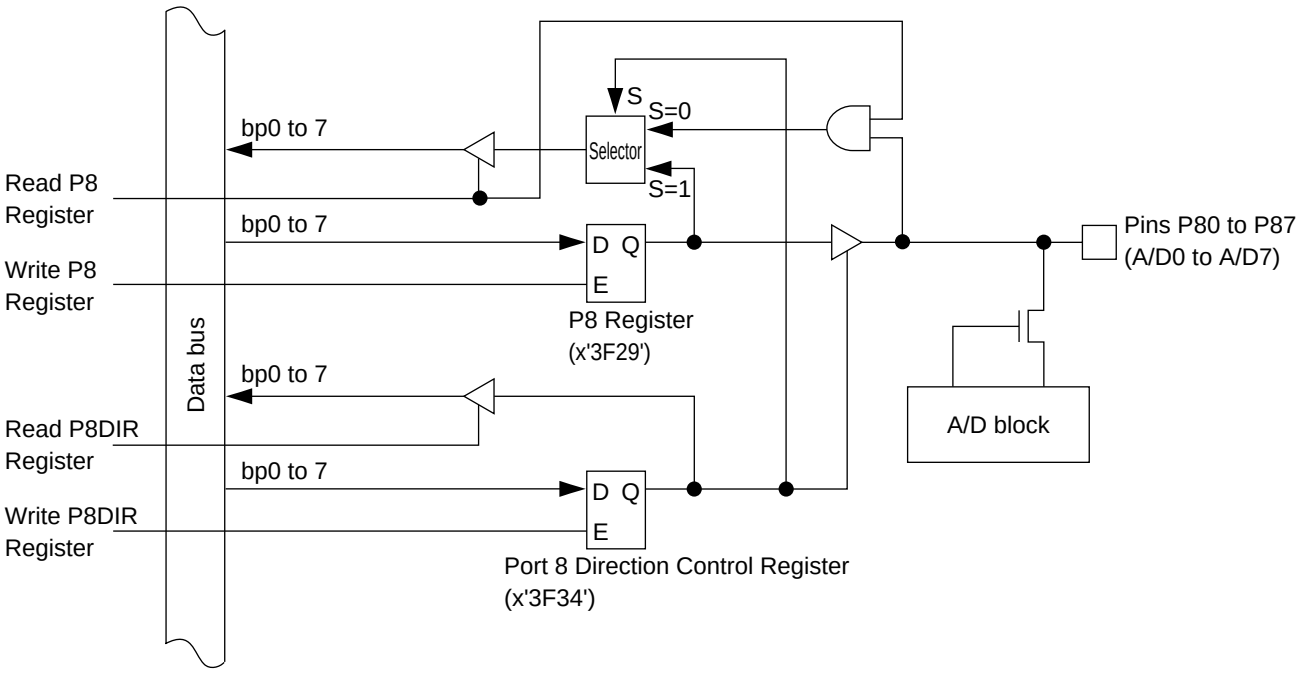


Figure 3-2-30 P80 to P87 Configuration

3-2-9 Port 9 Configuration

Port 9 contains 3 general-purpose input pins. Pins P90, P91 and P92 also function as analog pins TPZ, YPG and PGMM respectively. The Analog Control Register (ANACNT: x'3F81', R/W) specifies whether these pins are used as general-purpose input pins or analog pins.
Data is input to pins P90, P91, and P92 by reading the data in the Port 9 Data Register (P9: x'3F2A', R) in the input enable setting state.

The I/O circuitry of pins P90, P91 and P92 is constructed from CMOS, with a Schmitt circuit at the inputs.

Unlike port 8, port B, or port C, the control of input enable/disable will have to be implemented through the enable control flag of the ANACNT register.

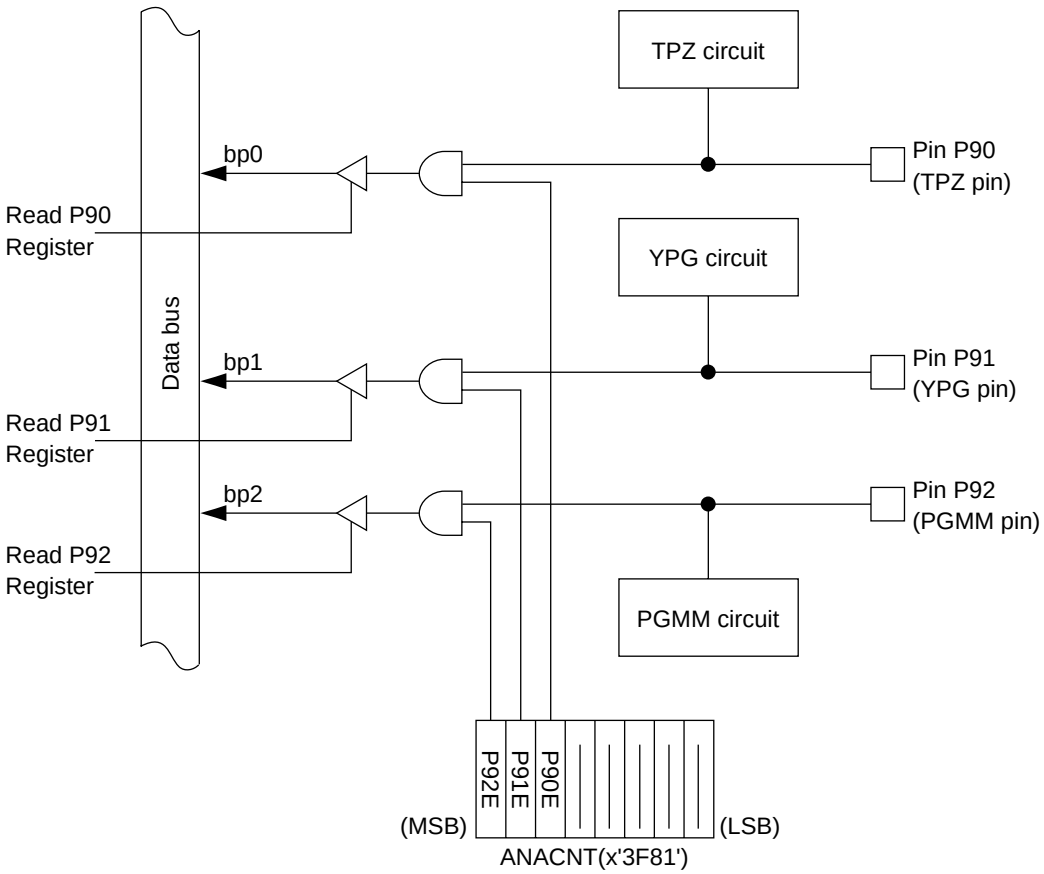


Figure 3-2-31 P90, P91 and P92 Configuration

3-2-10 Port A Configuration

Data is output to pins PA0 to PA7 by writing output data to the Port A Data Register (PA: x'3F2B', R/W). Data is input to pins PA0 to PA7 by reading the data from the Port A Data Register.

(1) I/O direction control for Port A is performed individually for each bit by the Port A Direction Control Register (PADIR: x'3F35', R/W). A data setting of '0' specifies input, and '1' specifies output.

(2) The I/O circuitry of pins PA0 to PA7 is constructed from CMOS, with a Schmitt circuit at the inputs.

(3) Pins PA0 to PA7 also function as OSD pins. Dual functions of the Port A pins are listed in the table below.

Table 3-2-5

Port Name	Dual Function	Register that Switches Pin
PA0	VOW1 (B)	Bit 0 of PASEL
PA1	VOW2 (G)	Bit 1 of PASEL
PA2	VOW3 (R)	Bit 2 of PASEL
PA3	VOW4 (Character)	Bit 3 of PASEL
PA4	VOB1 (Frame)	Bit 4 of PASEL
PA5	VOB2 (Frame)	Bit 5 of PASEL
PA6	LCOSI	—
PA7	LCOSO	—

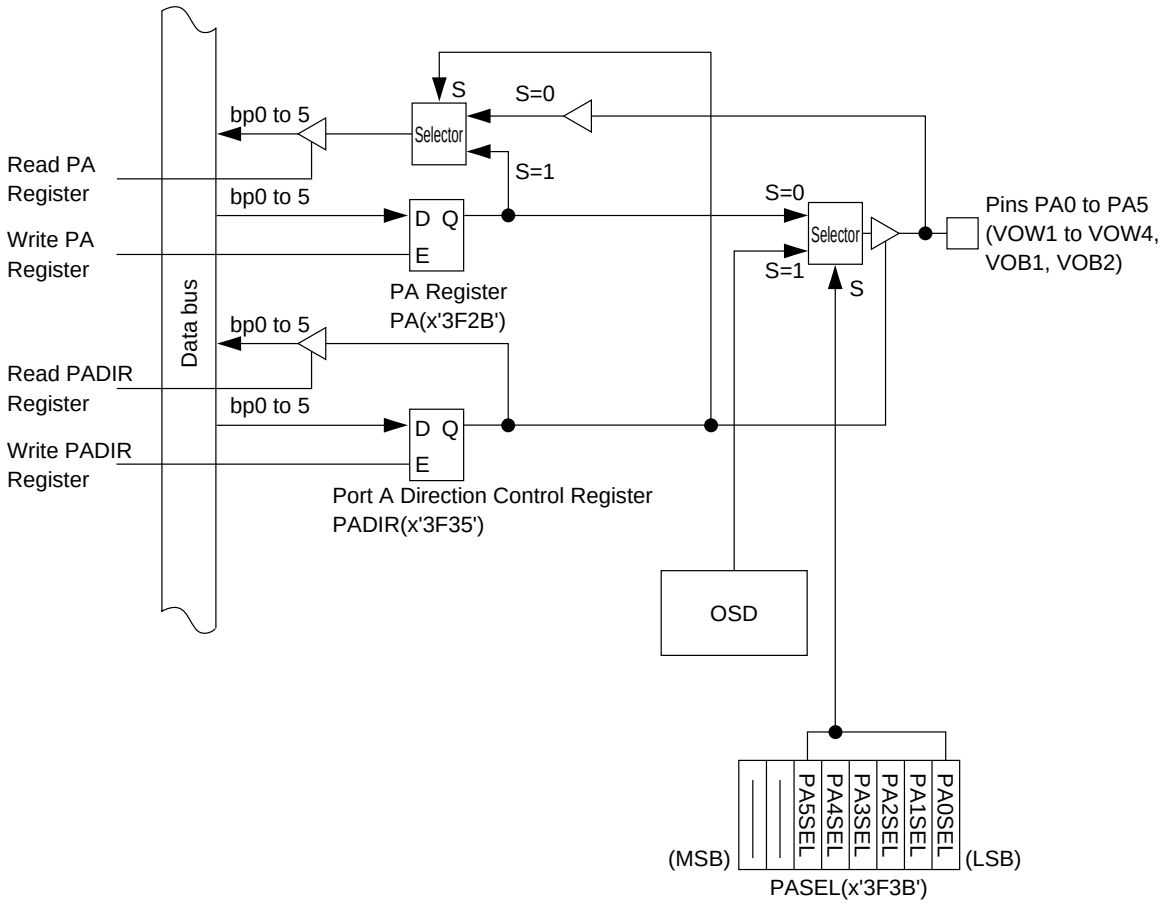


Figure 3-2-32 PA0 to PA5 Configuration

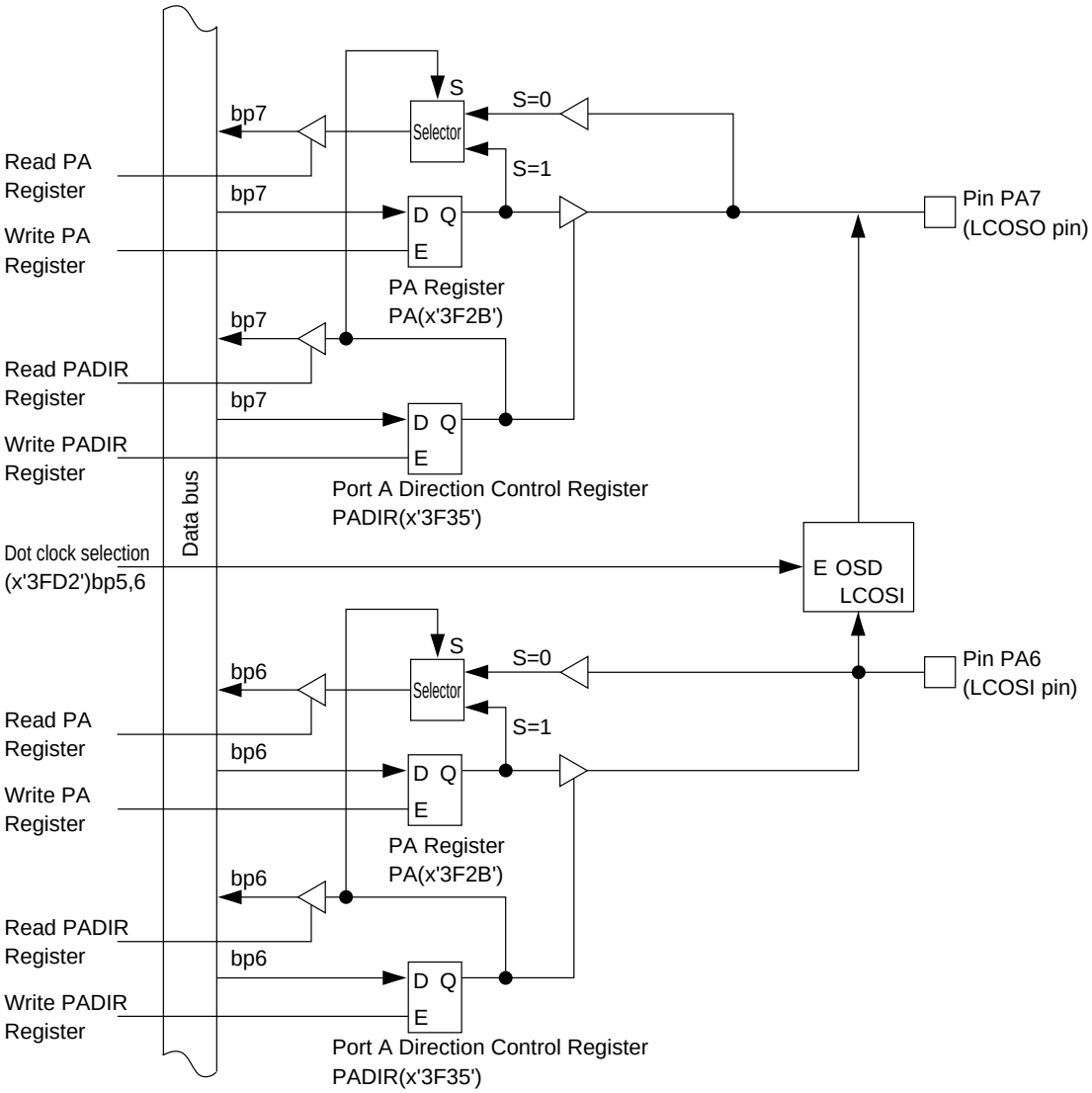


Figure 3-2-33 PA6 and PA7 Configuration

3-2-11 Port B Configuration

Data is output to pins PB0 to PB7 by writing output data to the Port B Data Register (PB: x'3F2C', R/W). Data is input to pins PB0 to PB7 by reading the data from the Port B Data Register.

(1) I/O direction control for Port B is performed individually for each bit by the Port B Direction Control Register (PBDIR: x'3F36', R/W). A data setting of '0' specifies input, and '1' specifies output.

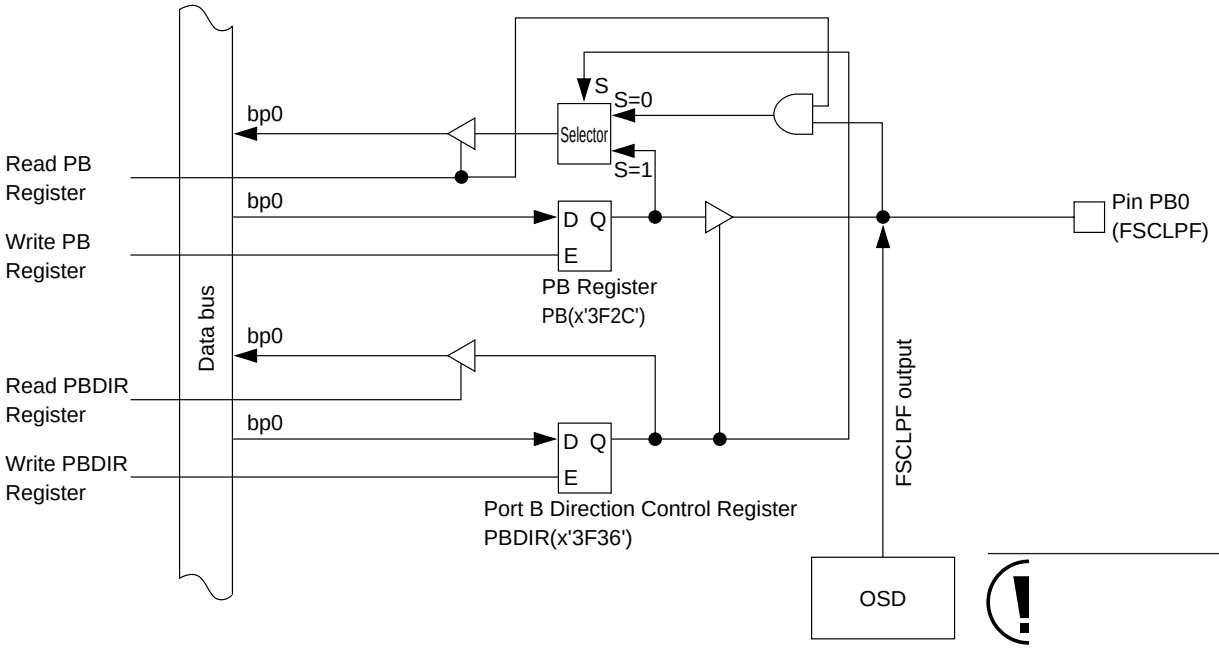
(2) The I/O circuitry of pins PB0 to PB7 is constructed from CMOS, with a Schmitt circuit at the inputs.

(3) Pins PB0 to PB7 also function as OSD pins. Dual functions of the Port B pins are listed in the table below.

(4) The input is enabled only during reading out the Port B data register (PB:x'3F2C'). Therefore, since the input is disabled at all times other than during a read out, no conduction current will be generated. However, input are always enabled at pins PB4 (CVIN2 input) and PB7 (HSYNC input).

Table 3-2-6

Port Name	Dual Function
PB0	FSCLPF
PB1	FSCI
PB2	AFCLPF
PB3	AFCC
PB4	CVIN2
PB5	CVIN
PB6	CVOUT
PB7	HSYNC



When using the FSCLPF output, clear bp0 of PBDIR (x'3FB6').

Figure 3-2-34 PB0 Configuration

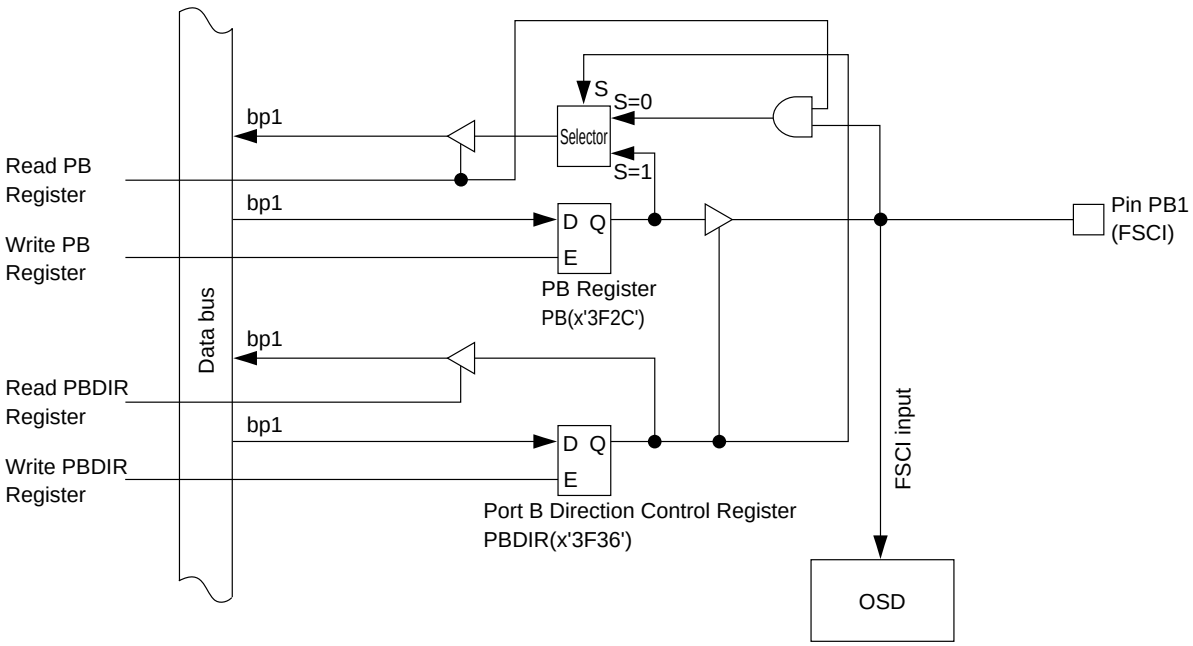


Figure 3-2-35 PB1 Configuration

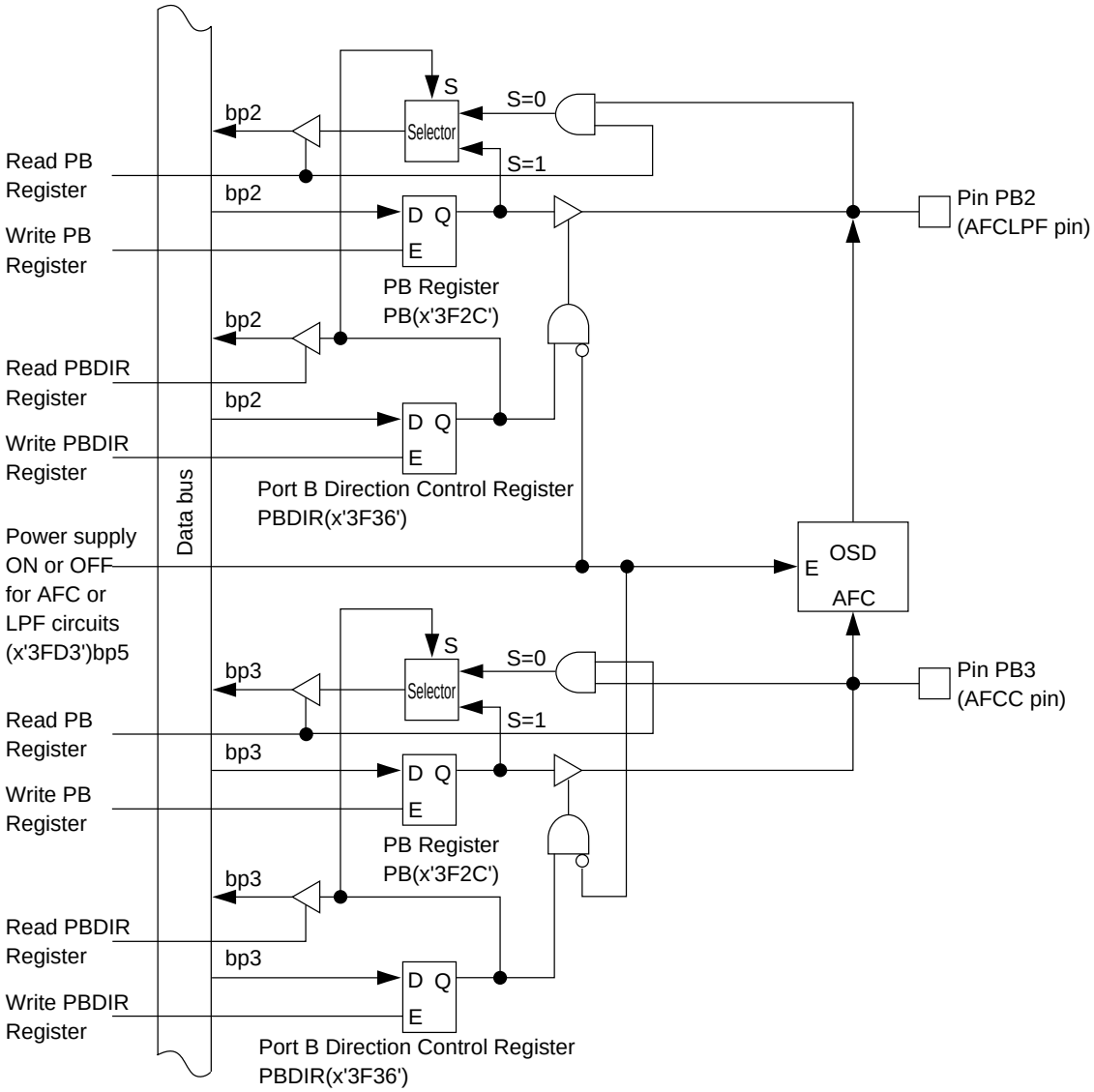


Figure 3-2-36 PB2 and PB3 Configuration

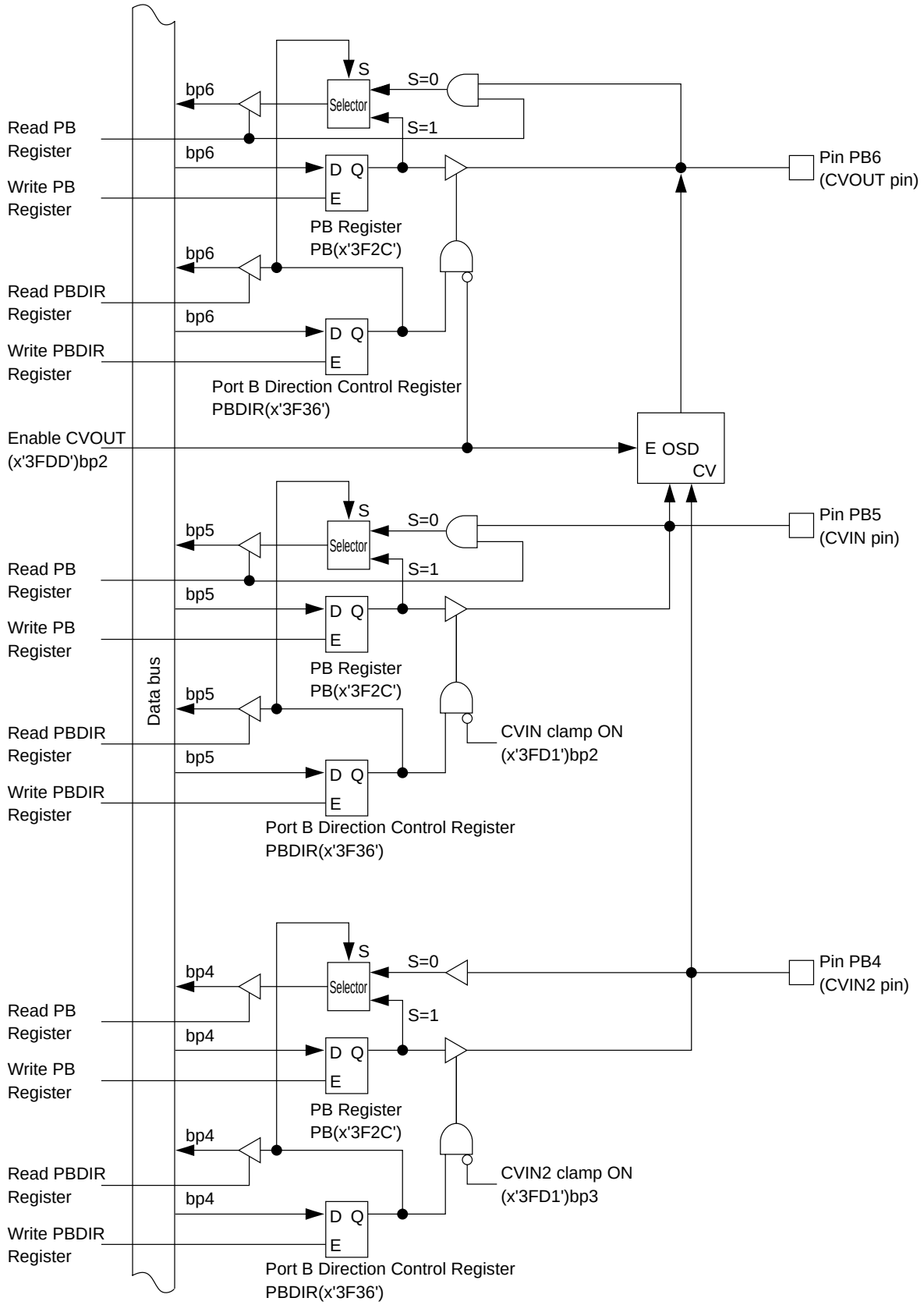


Figure 3-2-37 PB4, PB5 and PB6 Configuration

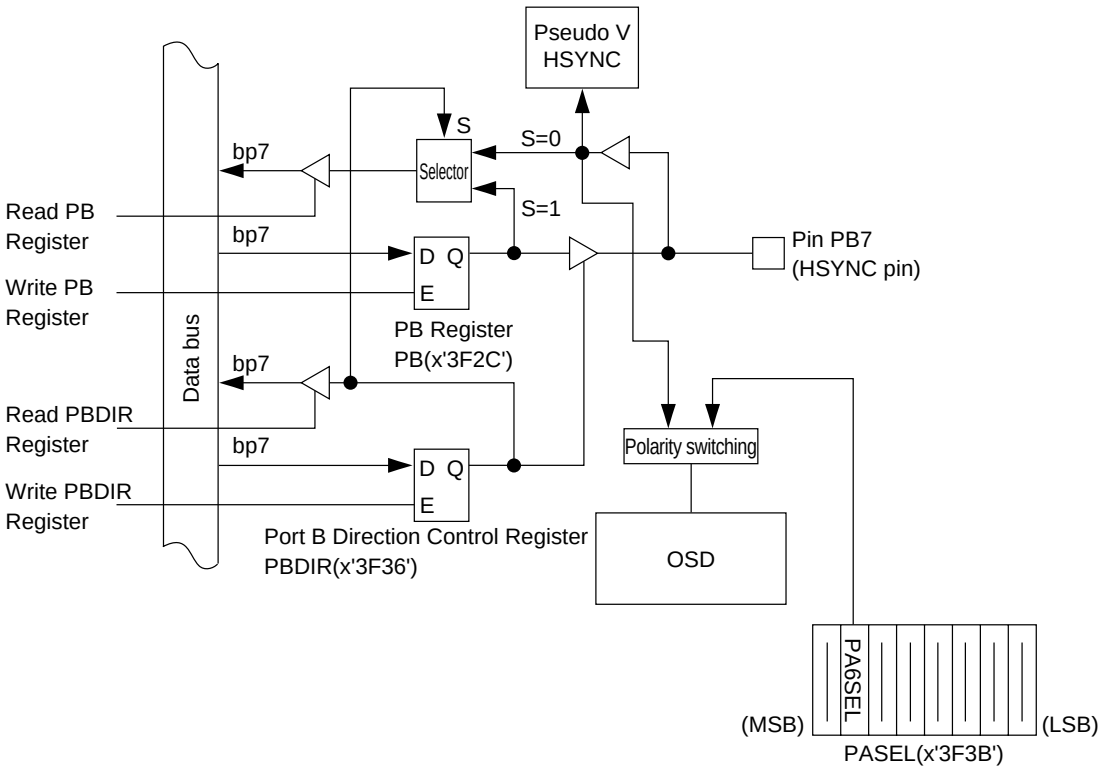


Figure 3-2-38 PB7 Configuration

3-2-12 Port C Configuration

Data is output to pins PC0 to PC3 by writing output data to the Port C Data Register (PC: x'3F2D', R/W). Data is input to pins PC0 to PC7 by reading the data from the Port C Data Register.

(1) I/O direction control for Port C is performed individually for each bit by the Port C Direction Control Register (PCDIR: x'3F37', R/W). A data setting of '0' specifies input, and '1' specifies output.

(2) The I/O circuitry of pins PC0 to PC3 is constructed from CMOS, with a Schmitt circuit at the inputs.

(3) Pins PC0 to PC3 also function as A/D Conversion Input Pins (AD8 to AD11).

(4) The input is enabled only during reading out the Port C data register (PC:x'3F2D'). Therefore, since the input is disabled at all times other than during a read out, no conduction current will be generated.

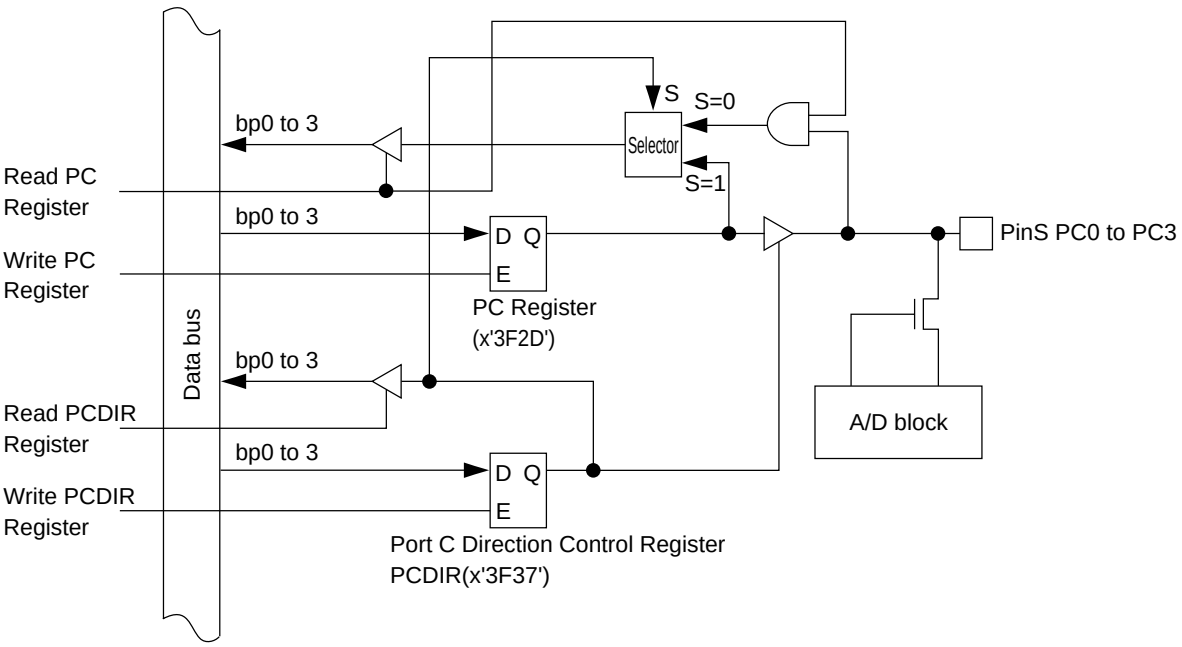


Figure 3-2-39 PC0 to PC3 Configuration

Chapter 4 Interrupt Functions

4

4-1 Interrupt Causes and Vector Addresses

4-1-1 Overview

[☞ For details, see Section 2-5, Interrupt Control Section, in the MN101D00 Series LSI Manual.]

There are thirty types of interrupt vectors other than reset that indicate the starting addresses of the interrupt service routines.

The vector addresses in the table at right are the addresses when the IVBM flag in register MEMCTR has been set to '0'.

Table 4-1-1 Interrupt Control Register

Vector No.	Interrupt cause	Control register (address)	Vector address
0	Reset	–	X'04000'
1	Non-Maskable interrupt (undefined interrupts, watch-dog timer interrupt)	NMICR (x'3FE1')	X'04004'
2	External pin (IRQ0) interrupt	IRQ0ICR (x'3FE2')	X'04008'
3	External pin (IRQ1) interrupt	IRQ1ICR (x'3FE3')	X'0400C'
4	External pin (IRQ2) interrupt	IRQ2ICR (x'3FE4')	X'04010'
5	External pin (IRQ3) interrupt	IRQ3ICR (x'3FE5')	X'04014'
6	External pin (IRQ4) interrupt	IRQ4ICR (x'3FE6')	X'04018'
7	Key interrupt	KEYICR (x'3FE7')	X'0401C'
8	Cylinder FG interrupt	YFGICR (x'3FE8')	X'04020'
9	Capstan FG interrupt	AFGICR (x'3FE9')	X'04024'
10	Timer 0 interrupt	TC0ICR (x'3FEA')	X'04028'
11	Timer 1 interrupt	TC1ICR (x'3FEB')	X'0402C'
12	Timer 2 interrupt	TC2ICR (x'3FEC')	X'04030'
13	Timer 3 interrupt	TC3ICR (x'3FED')	X'04034'
14	Timer 4 interrupt	TC4ICR (x'3FEE')	X'04038'
15	Timer 6 interrupt	TC6ICR (x'3FEF')	X'0403C'
16	Control interrupt	CTLICR (x'3FF0')	X'04040'
17	HSW Rising and falling edge interrupt	HSWICR (x'3FF1')	X'04044'
18	Servo VSYNC interrupt	VSYICR (x'3FF2')	X'04048'
19	SPG Interrupt	SPGICR (x'3FF3')	X'0404C'
20	FOCR0 Interrupt	FOCR0ICR (x'3FF4')	X'04050'
21	FOCR1 Interrupt	FOCR1ICR (x'3FF5')	X'04054'
22	FOCR2 Interrupt	FOCR2ICR (x'3FF6')	X'04058'
23	OSD Interrupt	OSDICR (x'3FF7')	X'0405C'
24	XDS Interrupt	XDSICR (x'3FF8')	X'04060'
25	SIF0 Interrupt	SIF0ICR (x'3FF9')	X'04064'
26	SIF1 Interrupt	SIF1ICR (x'3FFA')	X'04068'
27	SIF2 Interrupt	SIF2ICR (x'3FFB')	X'0406C'
28	A/D Interrupt (PWM14 interrupt)	ADICR (x'3FFC')	X'04070'
29	OSD Vsync interrupt	OSDVICR (x'3FFD')	X'04074'
31	ICE Interrupt	DCR (x'3FE0')	X'0407C'

Interrupt type	Interrupt level	Interrupt cause	Generation operation	Acceptance period	Starting address	PSW State after acceptance	Interrupt generation condition
Reset	—	External input at the NRST pin	Input directly to the CPU core	At any time	Address pointed to by the vector address x'4000'	All the flags are cleared to '0'.	When '0' is input at the NRST pin (P04).
Non-maska-ble interrupt		Watchdog timer interrupt When an undefined instruction is executed	Input to the CPU core from the non-maskable interrupt control register (NMICR)		Address pointed to by the vector address x'4004'	The interrupt mask level of PSW is cleared to '00', and the stack pointer flag for interrupt is set to '1'.	Overflow of the watchdog timer (Timer 5) When an undefined instruction x'FF' is executed (In the ICE, the vector x'407C' is accessed when an undefined instruction code is executed.)
External pin interrupt (IRQ0)	Can be set by the program at any level from 0 to 2	Input at pin IRQ0 (P64)	Interrupt request level set in the interrupt level flag of register IRQ0ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register IRQ0ICR.	Address pointed to by the vector address x'4008'	The content of the interrupt level flag is set to the interrupt mask level of PSW. The stack pointer flag for interrupts is set to '1'.	An interrupt is generated when the edge specified by bit 0 of register IRQCNT (x'3FFD') is input at the IRQ0 pin (P64).
External pin interrupt (IRQ1)		Input at pin IRQ1 (P63)	Interrupt request level set in the interrupt level flag of register IRQ1ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register IRQ1ICR.	Address pointed to by the vector address x'400C'		An interrupt is generated when the edge specified by bit 1 of register IRQCNT (x'3FFD') is input at the IRQ1 pin (P63).
External pin interrupt (IRQ2)		Input at pin IRQ2 (P62)	Interrupt request level set in the interrupt level flag of register IRQ2ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register IRQ2ICR.	Address pointed to by the vector address x'4010'		An interrupt is generated when the edge specified by bit 2 of register IRQCNT (x'3FFD') is input at the IRQ2 pin (P62).
External pin interrupt (IRQ3)		Input at pin IRQ3 (P61)	Interrupt request level set in the interrupt level flag of register IRQ3ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register IRQ03ICR.	Address pointed to by the vector address x'4014'		An interrupt is generated when the edge specified by bit 3 of register IRQCNT (x'3FFD') is input at the IRQ3 pin (P61).
External pin interrupt (IRQ4)		Input at pin IRQ4 (P60)	Interrupt request level set in the interrupt level flag of register IRQ4ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register IRQ4ICR.	Address pointed to by the vector address x'4018'		An interrupt is generated when the edge specified by bit 4 of register IRQCNT (x'3FFD') is input at the IRQ4 pin (P60).
Key interrupt		The input at one of pins KEYICR0 to 4 specified by bits 0 to 4 of register KEYCNT	Interrupt request level set in the interrupt level flag of register KEYICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register KEYICR.	Address pointed to by the vector address x'401C'		An interrupt is generated when a falling edge is input at one of pins KEYICR0 to 4 specified by bits 0 to 4 of register KEYCNT (x'3FFE').
Cylinder FG interrupt		Input at the YFG pin	Interrupt request level set in the interrupt level flag of register YFGICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register YFGICR.	Address pointed to by the vector address x'4020'		The polarity specified by bits 1 and 3 of the OPTION register (x'3F83') is selected for the input signal at the YFG pin specified by bit 4 of register ANACNT (x'3F81') and an interrupt is generated if this output is a rising edge.

Interrupt type	Interrupt level	Interrupt cause	Generation operation	Acceptance period	Starting address	PSW State after acceptance	Interrupt generation condition
Capstan FG interrupt	Can be set by the program at any level from 0 to 2	Input at the AFG pin	Interrupt request level set in the interrupt level flag of register AFGICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register AFGICR.	Address pointed to by the vector address x'4024'	The content of the interrupt level flag is set to the PSW interrupt mask level. The stack pointer flag for interrupt is set to '1'.	The interrupt is generated at the rising edge of the signal selected by bit 3 of register TRCNT.
Timer 0 interrupt		Timer 0 overflow	Interrupt request level set in the interrupt level flag of register TC0ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register TC0ICR.	Address pointed to by the vector address x'4028'		The interrupt is generated when the count in Timer 0 overflows.
Timer 1 interrupt		Timer 1 overflow	Interrupt request level set in the interrupt level flag of register TC1ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register TC1ICR.	Address pointed to by the vector address x'402C'		The interrupt is generated when the count in Timer 1 overflows.
Timer 2 interrupt		- Timer 2 overflow - Match between the shift register value and the compare register value - Capture enable edge - Underflow of SRBC2	Interrupt request level set in the interrupt level flag of register TC2ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register TC2ICR.	Address pointed to by the vector address x'4030'		The interrupt selected by bits 5 and 6 of the TM2 register (x'3F5C') is generated. BC2 Overflow if these bits are '00'. Capture enable edge of the ICR2 register if these bits are '01'. SRBC2 Underflow if these bits are '10'. Match between the values of the CMPR2 register and the SR2F register if these bits are '11'.
Timer 3 interrupt		Timer 3 overflow	Interrupt request level set in the interrupt level flag of register TC3ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register TC3ICR.	Address pointed to by the vector address x'4034'		The interrupt is generated when the count in Timer 3 overflows.
Timer 4 interrupt		Timer 4 overflow	Interrupt request level set in the interrupt level flag of register TC4ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register TC4ICR.	Address pointed to by the vector address x'4038'		The interrupt is generated when the count in Timer 4 overflows.
Timer 6 interrupt		- Timer 6 overflow (2^{16}) - 2^{15} Count - 2^{14} Count - 2^{13} Count	Interrupt request level set in the interrupt level flag of register TC6ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register TC6ICR.	Address pointed to by the vector address x'403C'		The interrupt selected by bits 1 and 2 of register TC6CNT (x'3F6B') is generated. BC6 Overflow if these bits are '00'. 2^{15} Count if '01'. 2^{14} Count if '10'. 2^{13} Count if '11'.
Control interrupt		Change in the control signal	Interrupt request level set in the interrupt level flag of register CTLICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register CTLICR.	Address pointed to by the vector address x'4040'		The interrupt is generated at the rising edge of the signal specified by bit 0 of register AFGPR (x'3F86').
HSW Interrupt		Change in the HSW signal	Interrupt request level set in the interrupt level flag of register HSWICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register HSWICR.	Address pointed to by the vector address x'4044'		The interrupt is generated at both edges of the HSW signal output at the HSW pin.
Servo VSYNC Interrupt		Change in the VSYNC signal for servo	Interrupt request level set in the interrupt level flag of register VSYICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register VSYICR.	Address pointed to by the vector address x'4048'		The interrupt is generated at the rising edge of the VSYNC signal for servo that is detected by register VDETS (x'3FDE').
SPG Interrupt		Match between the values in the free running counter and register SPGTIM.	Interrupt request level set in the interrupt level flag of register SPGICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register SPGICR.	Address pointed to by the vector address x'404C'		The interrupt is generated when bit 3 to 18 of the free running counter becomes equal to the contents of register SPGTIM (x'2E1C' to x'2E1D').

Interrupt type	Interrupt level	Interrupt cause	Generation operation	Acceptance period	Starting address	PSW State after acceptance	Interrupt generation condition
FOCR0 Interrupt	Can be set by the program at any level from 0 to 2	Match between the free running counter and register FOCR0 contents.	Interrupt request level set in the interrupt level flag of register FOCR0ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register FOCR0ICR.	Address pointed to by the vector address x'4050'	The interrupt level flag content is set to the PSW interrupt mask level. The stack pointer flag for interrupt is set to '1'.	The interrupt is generated when the value of bits 3 to 18 of the free running counter becomes equal to register FOCR0 (x'2E16' to x'2E17').
FOCR1 Interrupt		Match between the free running counter and register FOCR1 contents.	Interrupt request level set in the interrupt level flag of register FOCR1ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register FOCR1ICR.	Address pointed to by the vector address x'4054'		The interrupt is generated when the value of bits 3 to 18 of the free running counter becomes equal to the FOCR1 register (x'2E18' to x'2E19').
FOCR2 Interrupt		Match between the free running counter and register FOCR2 contents.	Interrupt request level set in the interrupt level flag of register FOCR2ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register FOCR2ICR.	Address pointed to by the vector address x'4058'		The interrupt is generated when the value of bits 3 to 18 of the free running counter becomes equal to the FOCR2 register (x'2E1A' to x'2E1B').
OSD Interrupt		Specified display line	Interrupt request level set in the interrupt level flag of register OSDICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register OSDICR.	Address pointed to by the vector address x'405C'		The interrupt is generated at the interrupt generation line set in bits 3 to 0 of the CMPLIN register (x'3FDC').
XDS Interrupt		Match between the vertical display counter and the contents of register XDSLIN2.	Interrupt request level set in the interrupt level flag of register XDSICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register XDSICR.	Address pointed to by the vector address x'4060'		The interrupt is generated when the values of the XDSLIN2 register becomes equal to the vertical display counter.
SIF0 Interrupt		After Serial 0 data transfer	Interrupt request level set in the interrupt level flag of register SIF0ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register SIF0ICR.	Address pointed to by the vector address x'4064'		The interrupt is generated after the data transfer for Serial 0.
SIF1 Interrupt		After Serial 1 data transfer	Interrupt request level set in the interrupt level flag of register SIF1ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register SIF1ICR.	Address pointed to by the vector address x'4068'		The interrupt is generated after the data transfer for Serial 1.
SIF2 Interrupt		After Serial 2 data transfer	Interrupt request level set in the interrupt level flag of register SIF2ICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register SIF2ICR.	Address pointed to by the vector address x'406C'		The interrupt is generated after the data transfer for Serial 2.
AD Interrupt PWM14 Interrupt		After AD conversion Change in the PWM14 output	Interrupt request level set in the interrupt level flag of register ADICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register ADICR.	Address pointed to by the vector address x'4070'		- The interrupt is generated at the end of A/D conversion. - The interrupt is generated at the falling edge of the basic cycle of PWM14 (when the polarity of the output at the PWM14 pin is positive).
OSD VSYNC Interrupt		Change in the VSYNC signal for OSD	Interrupt request level set in the interrupt level flag of register OSDVICR is input to the CPU core.	The acceptance is determined by the interrupt mask level in the PSW and the interrupt control set in register OSDVICR.	Address pointed to by the vector address x'4074'		The interrupt is generated at the rising edge of the TVSSG VSYNC signal during internal synchronization. The interrupt is generated at the rising edge of the OSD VSYNC signal that is detected by register OSDVDET (x'3FBA') during external synchronization.

An example of entries in an actual interrupt vector table using an assembler is shown below.

```
org     x'0000'
        da  A(start)
org     x'0004'
        da  A(nmicr)
org     x'0008'
        da  A(irg0icr)
org     x'000C'
        da  A(irg1icr)
org     x'0010'
        da  A(irg2icr)
org     x'0014'
        da  A(irg3icr)
org     x'0018'
        da  A(irg4icr)
org     x'001C'
        da  A(keyicr)
org     x'0020'
        da  A(yfgicr)
org     x'0024'
        da  A(afgicr)
org     x'0028'
        da  A(tc0icr)
org     x'002C'
        da  A(tc1icr)
org     x'0030'
        da  A(tc2icr)
org     x'0034'
        da  A(tc3icr)
org     x'0038'
        da  A(tc4icr)
org     x'003C'
        da  A(tc6icr)
org     x'0040'
        da  A(ctlicr)
org     x'0044'
        da  A(hswicr)
org     x'0048'
        da  A(vsyicr)
org     x'004C'
        da  A(spgicr)
org     x'0050'
        da  A(focr0icr)
```

```
org     x'0054'
        da  A(focr1icr)
org     x'0058'
        da  A(focr2icr)
org     x'005C'
        da  A(osdicr)
org     x'0060'
        da  A(xdsicr)
org     x'0064'
        da  A(sif0icr)
org     x'0068'
        da  A(sif1icr)
org     x'006C'
        da  A(sif2icr)
org     x'0070'
        da  A(adicr)
org     x'0074'
        da  A(osdicr)
org     x'007C'
        da  A(dcr)

start  equ  *
```

The CPU processing program starts below.

■ Interrupt Mask Level Setting

The user program is used to set the interrupt level for all vectors other than vector 0 (reset), vector 1 and vector 31 (dedicated to non-maskable interrupts). The interrupt level can be set to any of levels 0 to 3. The priority order of several interrupts that have been set at the same level is from lower to higher vector number. (For example, if a vector 3 interrupt and a vector 4 interrupt both set at level 1 are requested at the same time, the vector 3 interrupt is accepted first.)

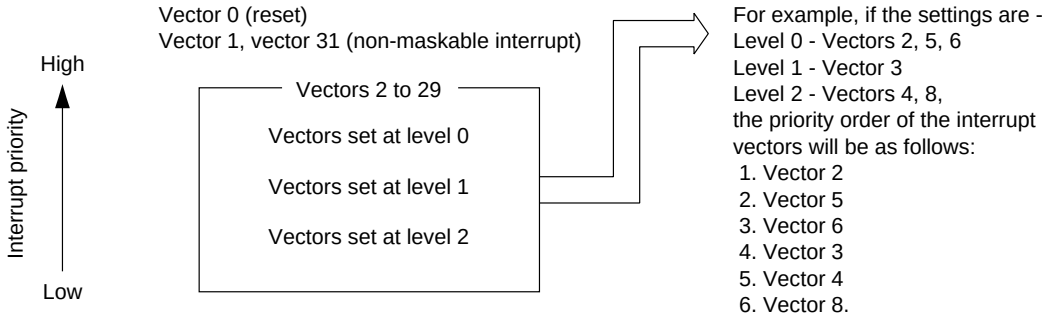


Figure 4-1-1 Concept of Interrupt Priority Order

■ Interrupt Acceptance Decision

For the maskable interrupts (vector 2 to 29), the interrupt level corresponding to the cause of the interrupt that has been generated is input to the CPU core. This level is the information of the interrupt level flags (xxxLV1 to 0) of the maskable interrupt control register (xxxICR). The interrupts are accepted when maskable interrupt enable flag (MIE) of the processor status word (PSW) of the CPU core is '1', the priority of the interrupt is higher than the interrupt mask level (IM1, IM0), and also the interrupt enable flag (xxxIE) permits that interrupt. As a result of servicing the interrupt, the IM1 and IM0 flags of the PSW are updated to the level of the accepted interrupt.

However, the reset input and the non-maskable interrupts are always accepted irrespective of the mask level or the status of the MIE flag.

■ Sequence of Interrupt Processing

The processing sequence of all interrupts other than reset input consists of interrupt request, interrupt acceptance, hardware processing, etc. The hardware operations made after accepting the interrupt consist of pushing the program counter and the processor status word, etc., onto the stack, and branching to the address specified by the interrupt vector. Upon completion of the interrupt service routine, register values pushed to the stack when the interrupt was accepted are restored, and the program counter and processor status word are popped back from the stack by hardware operation.

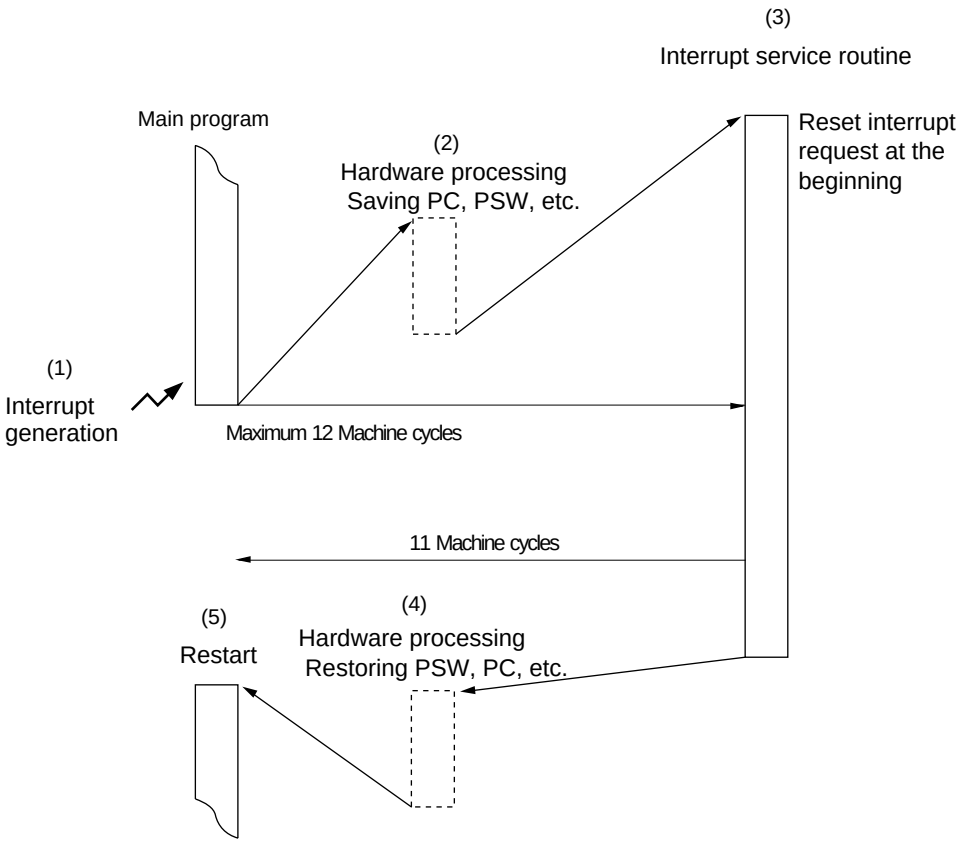


Figure 4-1-2 Sequences of Interrupt Processing (Maskable Interrupts)

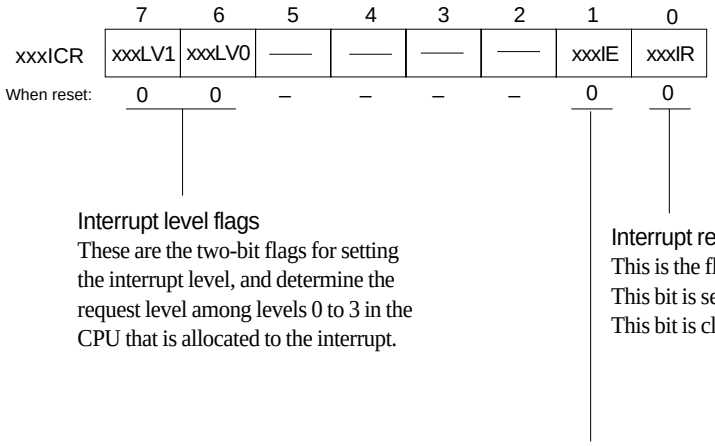
4-1-2 Interrupt Control Register

The interrupt control registers consist of the control registers of maskable interrupt control registers (xxxICR) and the non-maskable control register (NMICR).

If xxxLVn is set to '11' (level 3), the interrupt of that vector is disabled irrespective of the contents of the interrupt enable flag and the interrupt request flag.

■ Maskable interrupt control registers (xxxICR) (interrupts of vectors 2 to 29)

The maskable interrupt control registers (xxxICR) are allocated for each maskable interrupt other than vector 1 and respectively control the interrupt allocated for that vector number. These registers are composed of the interrupt level flags (xxxLV1,0), the interrupt enable flag (xxxIE), and the interrupt request flag (xxxIR).



Interrupt level flags
These are the two-bit flags for setting the interrupt level, and determine the request level among levels 0 to 3 in the CPU that is allocated to the interrupt.

Interrupt request flag
This is the flag for registering the interrupt request.
This bit is set to '1' when there is an interrupt request.
This bit is cleared automatically when the interrupt is accepted.

Interrupt enable flag
The interrupt of the corresponding peripheral function is enabled or disabled by this flag. The interrupt is enabled when this flag is set to '1'.
If the interrupt flag (xxxIR) has been set to '1', the interrupt request is output to the CPU when the corresponding IE bit is changed from '0' to '1'.

Set a higher level (smaller set value) in the interrupt level flags (xxxLV1,0) for the interrupt causes that have a high urgency or have a short allowed processing time.

Figure 4-1-3 Maskable Interrupt Control Register (xxxICR)

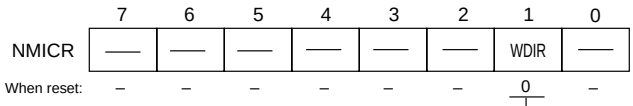
When manipulating the interrupt request flags (xxxIR) by software, such as in the initialization program, etc., it is necessary to first set the interrupt request enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, reset IRWE to '0' at the end of such operations.

Table 4-1-3 Relationship between Interrupt Mask Level (IM) and Interrupt Level (IL)

Interrupt mask level (PSW)	Processed interrupt level	Order
0	Non-maskable interrupt (NMI) only	High
1	Level 0, NMI	↑
2	Levels 0 to 1, NMI	
3	Levels 0 to 2, NMI	Low

■ Non-maskable interrupt control register (NMICR address: x'3FE1')

The non-maskable interrupt control register (NMICR) is allocated for vector 1, and stores the interrupt cause when a non-maskable interrupt is generated. When a non-maskable interrupt is generated, the interrupt is accepted irrespective of the interrupt mask level (IM) in the PSW and the program execution branches to the address written at location x'4004' of the interrupt vector table.



WDIR	Watchdog timer interrupt request
0	No interrupt request
1	The interrupt request is generated

Figure 4-1-4 Non-maskable Interrupt Control Register (NMICR)

Watchdog timer overflow interrupt request flag (WDIR)

The watchdog timer overflow interrupt request flag is set to '1' when the watchdog timer overflows.

The state of the watchdog timer interrupt request (WDIR) is retained even after the interrupt is accepted. This flag should be cleared by the non-maskable interrupt service routine.

4-1-3 Interrupt Level

The user program sets the interrupt priority level for each interrupt group. Whether or not to accept the maskable interrupt request is determined by the maskable interrupt enable flag (MIE) set in the processor status word (PSW) inside the CPU, the interrupt mask level (IM1, IM0), and the status of the interrupt enable flag (xxxIE) in the maskable interrupt control register (xxxICR).

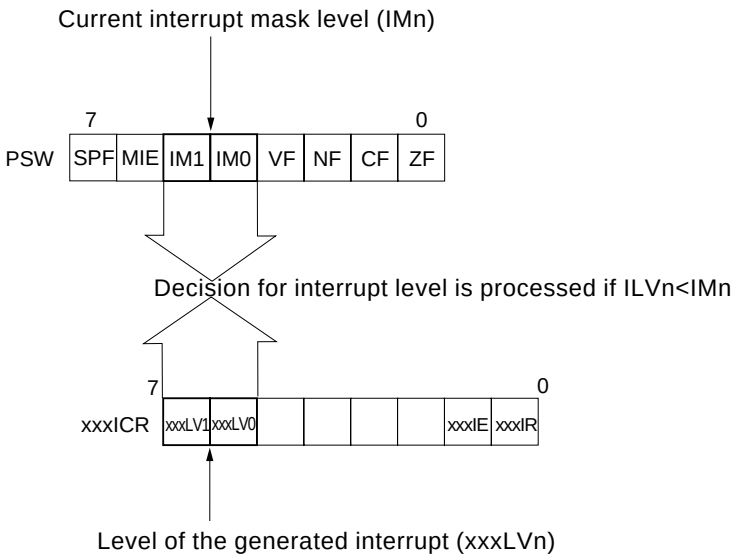


Figure 4-1-5 Decision of Interrupt Acceptance

The sequence from the occurrence of an interrupt cause up to the acceptance of the interrupt is described below.

(1) When an interrupt cause occurs, the interrupt request flag (xxxIR) of the maskable interrupt control register (xxxICR) corresponding to the interrupt cause becomes '1'.

(2) If the interrupt enable flag (xxxIE) corresponding to the interrupt request flag is '1', the interrupt request is output to the CPU.

(3) The interrupt priority level for the interrupt request is that in the corresponding interrupt level flags (xxxLV1,0).

(4) The interrupt is accepted if the interrupt request signal that is output has a level higher than the level set in the interrupt mask level (IM1,0) of the processor status word (PSW) and also the interrupt enable flag (MIE) in the PSW is '1' (enable).

(5) The interrupt request flag (xxxIR) is cleared after the interrupt is accepted.

The interrupt enable flag (xxxIE) is cleared after the interrupt is accepted.

MIE becomes '0' and interrupts are disabled in the following cases.

- When the program writes '0' to the MIE flag in the PSW.
- When a reset is input to the processor.

MIE becomes '1' and interrupts are enabled in the following case.

- When the program writes '1' to the MIE flag in the PSW.

The value of the interrupt mask level is changed under the following conditions.

- When a changed value is written in IM1 and IM0 of the PSW by the program.
- When a reset input is received. In this case IM1,0 become '00'.
- When a maskable interrupt is accepted, the interrupt mask level is the level of that interrupt.
- At the end of the interrupt service routine, when the RTI instruction is executed, the mask level is reset to the value before that interrupt was accepted.

When a maskable interrupt and a non-maskable interrupt are generated simultaneously, the non-maskable interrupt takes priority.

Upon acceptance of the interrupt, MN101DXX does not reset MIE of PSW to '0'. Care should be taken because this is different from other microcomputers (MN10200 Series, MN1860 Series, MN1870 Series, MN1880 Series, etc.)

When an interrupt with a lower mask level is generated during processing of an interrupt: (When Interrupt (1): LV1,0='00', interrupt (2): LV1,0='10')

The processing enclosed within parentheses () is carried out by the hardware.

****1***
When an interrupt is generated within the interrupt service routine (1), the interrupt is accepted as a nested interrupt if IL < IM. The interrupt is not accepted if IL ≥ IM.

****2***
When the interrupt (2) that occurred during the execution of the interrupt service routine (1) is not accepted because IL ≥ IM, it is accepted at the end of execution of the interrupt service routine (1).

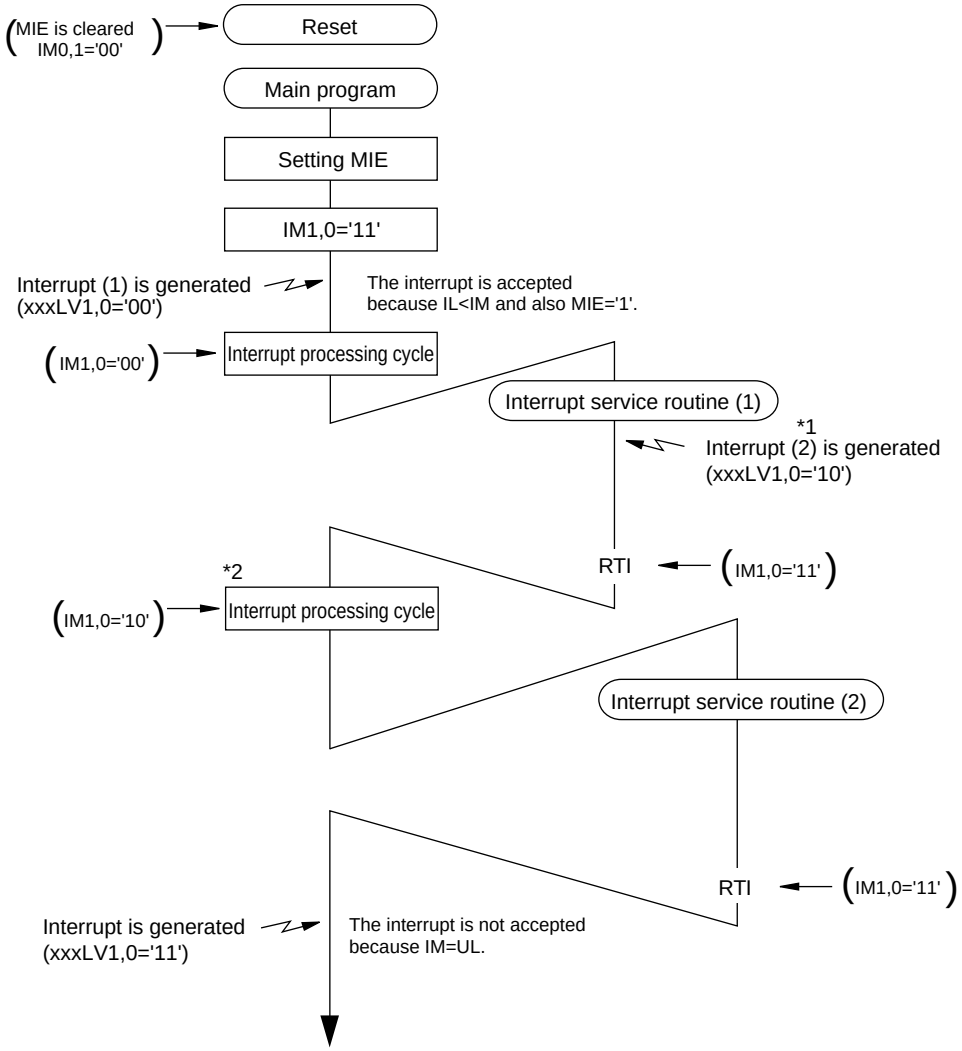


Figure 4-1-6 Processing Sequence of a Maskable Interrupt

4-1-4 Interrupt Processing Operations

In the MN101D02 Series, when an interrupt is accepted, the return address of the program, the PSW, etc., are saved onto the stack area by the hardware, and the program execution branches to the starting address of the interrupt service routine pointed to by the interrupt vector table.

The following is the sequence of operations performed by the hardware when an interrupt is accepted.

1. The content of the stack pointer is updated (SPI-6 → SPI).
2. The handy address register (HA) contents are saved onto the stack.
   HA Upper byte → (SPI+5)
   HA Lower byte → (SPI+4)
3. The contents of the program counter (PC = return address) are saved onto the stack.
   PC (Bit 18 - bit 17, bit 0) → (SPI+3)
   PC (Bit 16 - bit 9) → (SPI+2)
   PC (Bit 8 - bit 1) → (SPI+1)
4. The PSW is saved onto the stack.
   PSW → (SPI)
5. The xxxLVn of the accepted interrupt is copied to IM of PSW.
   Interrupt level (xxxLVn) → IM
6. The execution branches to the address pointed to in the vector table.

The handy address register is an internal register whose information is saved by the hardware so that the handy addressing function is not affected by the interrupt.

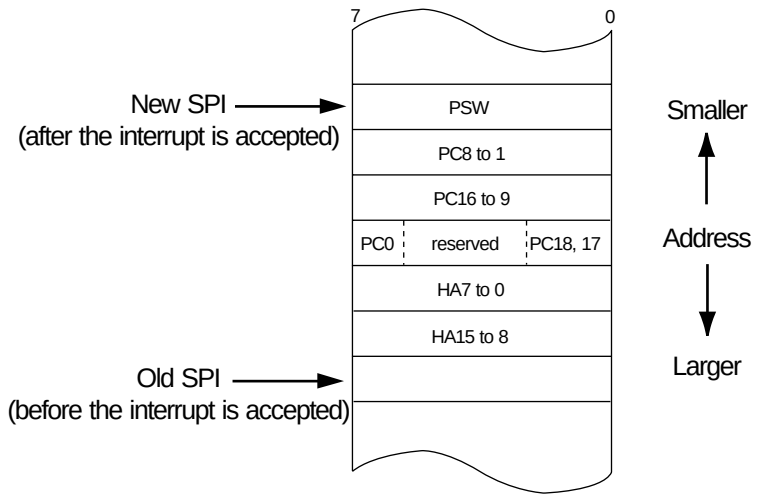


Figure 4-1-7 Condition of Stack during an Interrupt

Bit 6 to bit 2 in (SPI+3) are the reserved area of the hardware and are used for saving the hardware information. Do not modify these bits in the program.

4-1-5 Interrupt Return Operations

Returning from interrupt processing is done by first restoring within the program the contents of registers, etc., that were saved onto the stack at the beginning of interrupt processing (using the POP instruction), and then executing an RTI instruction to return to the program that was being executed at the time the interrupt was accepted.

The following is the sequence of operations of the return from interrupt (RTI) instruction.

1. The contents of the processor status word (PSW) saved in the stack (SPI) are restored.
2. The contents of the program counter (PC = return address) saved in the stack (SPI+1 to 3) are restored.
3. The contents of the handy address register (HA) are restored from the stack (SPI+4,5).
4. The value of SPI is updated. (SPI+6 → SPI)
5. The execution branches to the address indicated by the program counter (PC).

4-1-6 Nested Interrupts

After accepting an interrupt, the MN102D02 Series automatically disables the processing of interrupts of a lower level than that of the accepted interrupt.

During the processing of an interrupt, the values of xxxLV1,0 of the accepted interrupt are copied to IM1,0 of the processor status word (PSW). Therefore, following acceptance of an interrupt, interrupts with a level higher than that of the accepted interrupt are accepted as nested interrupts although all interrupts with a level lower than that of the accepted interrupt are automatically disabled. Basically, the operations are made in the priority order of the interrupt levels irrespective of the interrupt being processed, but it is also possible to control nested interrupts using the following procedure.

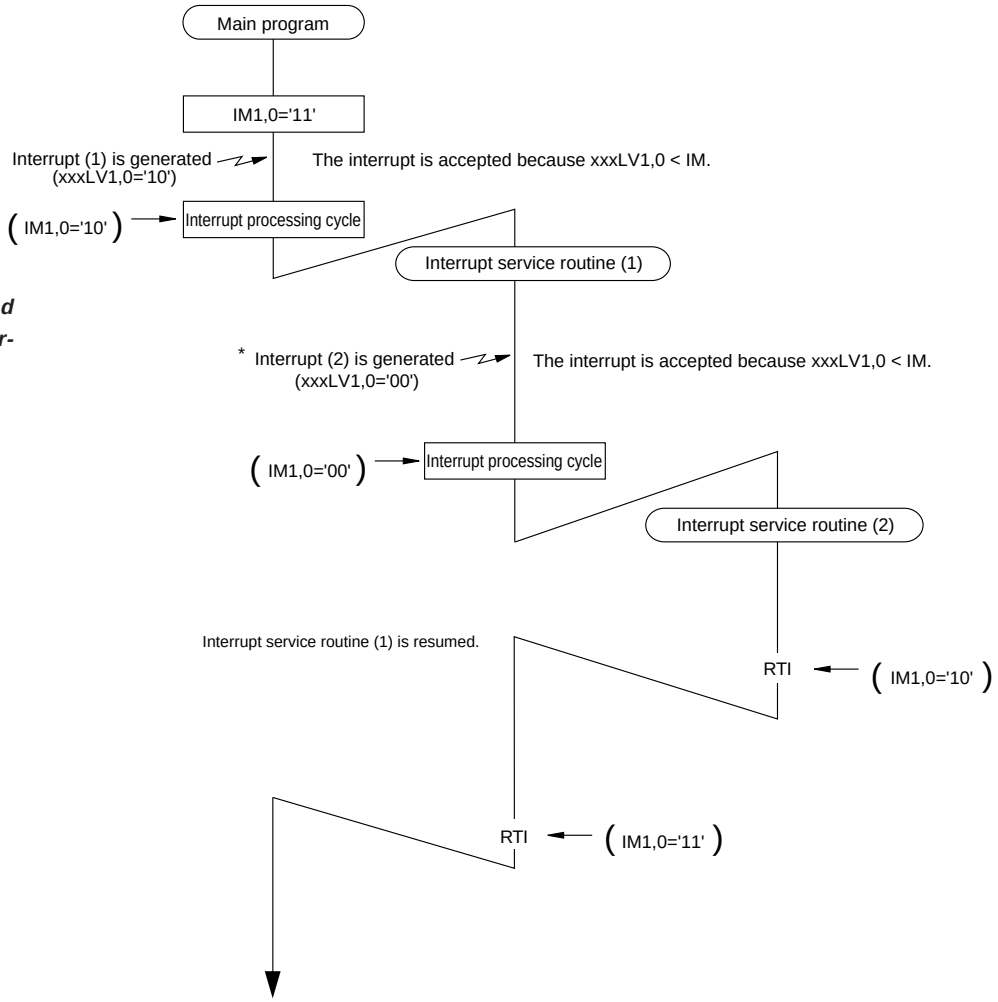
1. When disabling nested interrupts: One of the following two actions can be taken -
   - Clearing MIE of the PSW to '0', or
   - Rewriting IM1 and IM0 of the PSW to a higher mask level.
2. When enabling interrupts with a level lower than the accepted interrupt:
   - Lower the mask level by rewriting IM1 and IM0 of the PSW.

Nested interrupts are enabled only for interrupts of a level higher than the interrupt mask level (IM) in the PSW.

Although it is possible to accept interrupts of a lower level than the interrupt being currently processed by forcibly rewriting IM, care will have to be taken in such cases about overflowing of the stack due to nested interrupts.

Do not manipulate the maskable interrupt control register (xxxICR) when nested interrupts are enabled. When such an operation is necessary, temporarily disable interrupts by clearing the MIE flag of the PSW.

The sequence of operations for nested interrupts is shown below. (When interrupt (1) is xxxLV1,0='10' and interrupt (2) is xxxLV1,0='00')



The processing enclosed within parentheses () is carried out by the hardware.

Figure 4-1-8 Processing Sequence for Enabling Nested Interrupts

4-2 Method of Using Interrupts

The methods of using the different interrupts are described below.

4-2-1 Reset Interrupt

- Interrupt vector	x'04000'
- Interrupt level	Highest priority
- Interrupt causes	The interrupt is generated when an 'L' pulse is input at the NRST pin for more than two machine cycles. It is also generated when the P0OUT4 flag (bp4) of the P0OUT register (x'3F20') is cleared to '0'.
- Interrupt source selection	None
- Interrupt control register	None
- Interrupt processing	Always
- Related item	Section 10-2 Reset Functions

4-2-2 Non-maskable Interrupts (WD, undefined instruction execution)

- Interrupt vector — x'04004'

- Interrupt level — Lower than reset interrupt, but higher than maskable interrupts

- Interrupt causes — Overflow of the watchdog timer, execution of an undefined instruction code, or attempt to perform a write operation to an uninstalled ROM area.

☞ **The uninstalled ROM area is the area beyond the internal ROM area indicated for each type in the memory map of Figure 2-1-1 and up to the location x'3FFFF'.**

Microcomputer code (Note 1)	Watchdog timer period setting (Note 2)	Watchdog timer period (seconds)	Remarks
NORMAL SLOW IDLE	wds=0	2^{16}/fs	The watchdog timer interrupt is generated.
	wds=1	2^{19}/fs	
HALT STOP	—	Halted	The watchdog timer interrupt is not generated.

Note 1: The microcomputer mode is set in the CPUM register (x'3F00').
Note 2: The watchdog timer period setting is made by the WDS flag (bp1) of register WDCNT (x'3F1D').

- Interrupt control register (WD)

Watchdog timer counting operation The WDEN flag (bp0) of register WDCNT (x'3F1D')	
0	Watchdog timer counting stopped
1	Watchdog timer counting started

- Interrupt processing — When the above WDEN flag is '1', the interrupt is accepted irrespective of the interrupt mask level given by the IM1,0 flags of the processor status word (PSW) when either the watchdog timer overflows or when an undefined instruction code is executed, and the program execution branches to the address pointed to by the location x'4004' of the interrupt vector table.

When activating the watchdog timer, always make sure to clear beforehand the watchdog timer (WDCLR flag =1). In particular, care should be taken immediately after restarting from the HALT mode. Further, after the WDCLR flag has been set, it is cleared by the hardware after one machine cycle.

- Outline of the method of use (WD) — The watchdog timer is cleared at intervals shorter than the watchdog timer period, thereby ensuring that normally the watchdog timer interrupt is not generated. The watchdog timer is cleared by using the WDCLR flag of register WDCNT (x'3F1D').

- Return from interrupt (WD) — Return from the interrupt only after first clearing the WDCLR flag (bp2) of register WDCNT (x'3F1D').

- Related items — Section 5-6 Timer 5 Operation, Table 5-8-2 List of Timer Clock Sources.

4-2-3 External Pin Interrupts (IRQ0, 1, 2, 3, 4)

- Interrupt vectors

IRQ0 : x'04008'
IRQ1 : x'0400C'
IRQ2 : x'04010'
IRQ3 : x'04014'
IRQ4 : x'04018'

- Interrupt levels — Maskable interrupts
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled).

- Interrupt causes — When a signal of the specified edge is input at pins IRQ0 to IRQ4 (independent interrupt acceptance is permitted for the five pins).

- Interrupt source selection — Determined by the combination of the microcomputer mode, both edge/single edge selection, and edge polarity selection.

Microcomputer mode	Specified by register CPUM (x'3F00').
Both edge/single edge selection	Specified by the IRQ1 to IRQ3 EGCNT flag of register KEYCNT (x'3FB3').
Edge polarity	Specified by the IRQ0 to IRQ4 PO flag of register IRQCNT (x'3FB2').

Note: The content of control differs in IRO0, IRQ1-3, and IRQ4. See Figure 4-2-1 Block Diagram of External Interrupts IRQ0 to IRQ4.

- Interrupt control registers

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

IRQxLV1	IRQxLV0	Interrupt request level setting (bp7 and 6 of register IRQxICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (IRQx interrupts are disabled)

IRQxIE	IRQx pin interrupt enable (bp1 of register IRQxICR)
0	Interrupt disabled
1	Interrupt enabled

IRQxIR	IRQx pin interrupt request (bp0 of register IRQxICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (IRQ0ICR) ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (IRQ0ICR) 1      ; Enables the IRQ0 interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

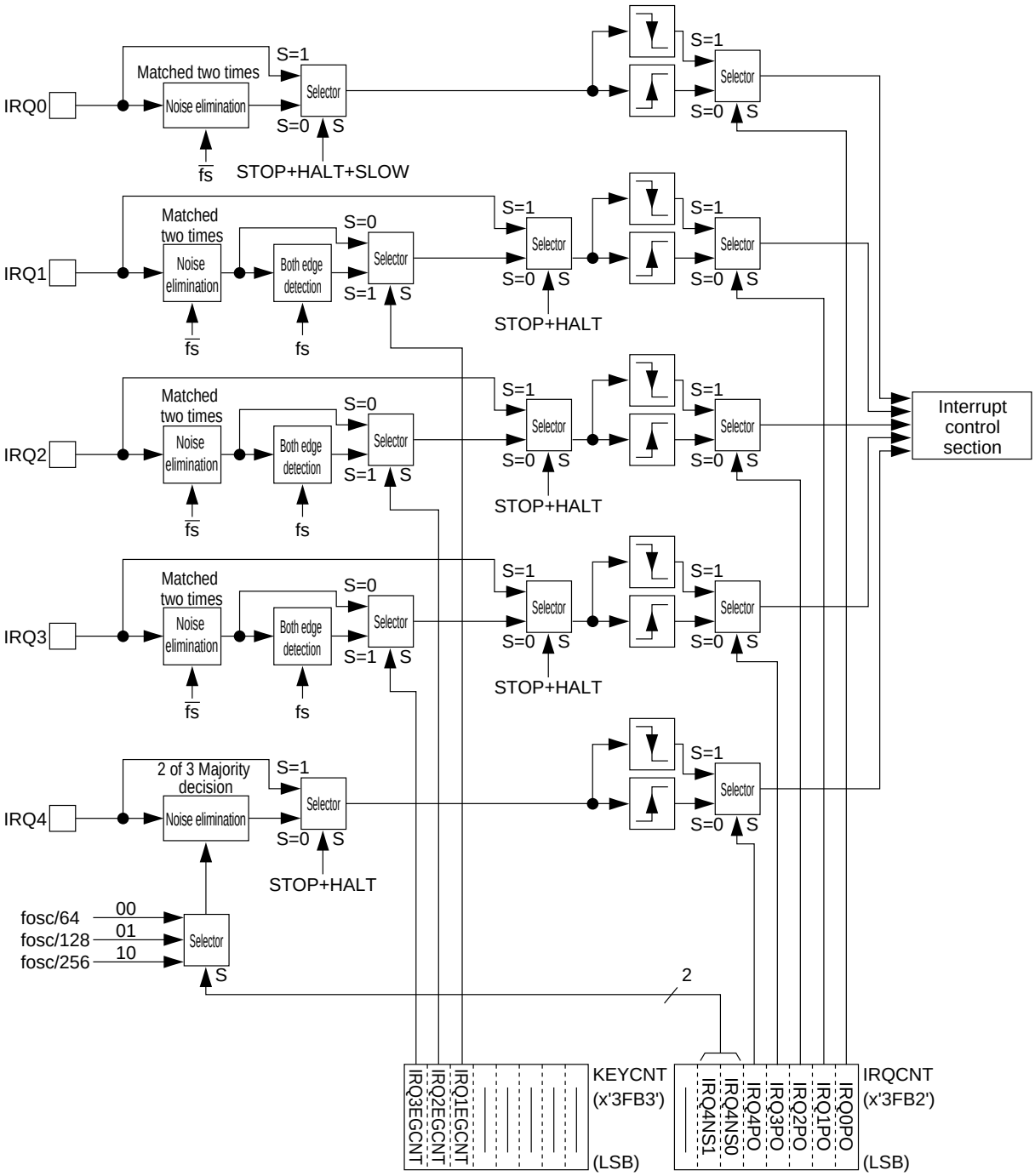


Figure 4-2-1 Block Diagram of External Interrupts IRQ0 to IRQ4

4-2-4 KEY Interrupt

- Interrupt vector x'0401C'
- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled)
- Interrupt causes When any one of the specified signals among pins KEYIRQ0 to KEYIRQ4 goes to the 'L' level.
- Interrupt source selection Determined by the microcomputer mode and the KEYIRQ pin input enable specification.

Microcomputer mode	Specified by register CPUM (x'3F00').
KEYIRQ pin input enable specification	Specified by the KEY0 to 4 SEL flags of register KEYCNT (x'3FB3').

Note: The microcomputer mode affects only the presence/absence of noise filter. See Figure 4-2-2 Block Diagram of Key Interrupts KEYIRQ0 to KEYIRQ4.

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

KEYLV1	KEYLV0	Interrupt request level setting (bp7 and 6 of register KEYICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (Key interrupts are disabled)

KEYIE	Key interrupt enable (bp1 of register KEYICR)
0	Interrupt disabled
1	Interrupt enabled

KEYIR	Key interrupt request (bp0 of register KEYICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (KEYICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (KEYICR) 1       ; Enables the key interrupts
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

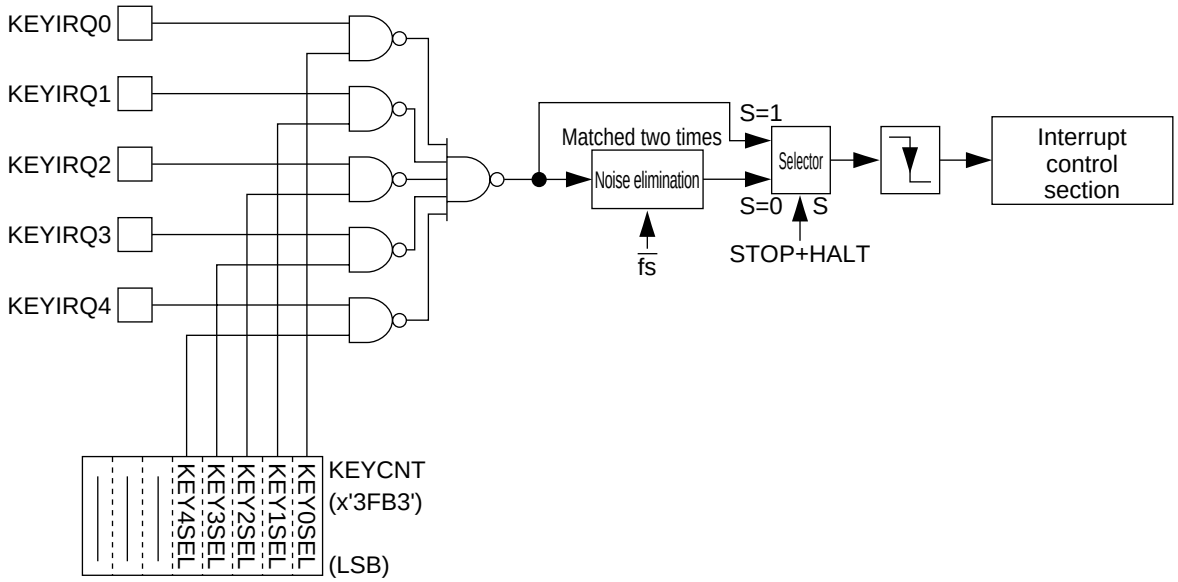


Figure 4-2-2 Block Diagram of Key Interrupts KEYIRQ0 to KEYIRQ4

4-2-5 Cylinder FG Interrupt

- Interrupt vector x'04020'

- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled)

- Interrupt cause The signal input to the YFG pin is either input to the YFG amplifier or is separated in the PFG 3-value separator. The interrupt is generated at the timing of the specified edge of the cylinder FG signal thus extracted.

- Interrupt source selection

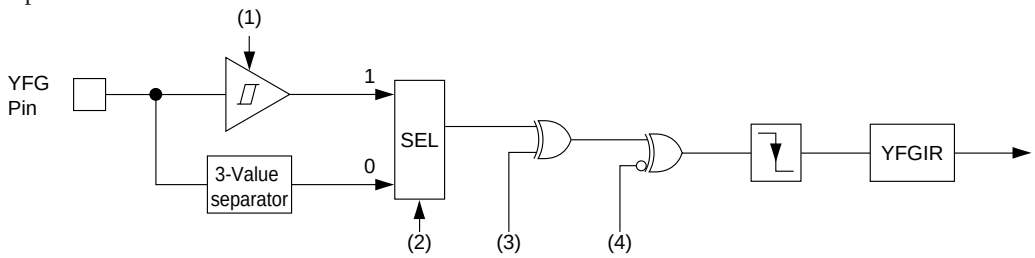


Figure 4-2-3 Block Diagram of Cylinder FG Interrupt Source

(1) YFG Amplifier input sensitivity setting
..... Set by the FGIVT flag of the OPTION register (x'3F83').
(0 : 50mVpp, 1 : 100mVpp)

(2) Selection of YFG amplifier or PFG 3-value input
..... Selected by the PFGS flag of register ANACNT (x'3F81').
(0 : PFG 1 : YFG)

(3) YFG Signal polarity 0
..... Selected by the YFGPO0 flag of the OPTION register (x'3F83') (linked to the reference edge for generating HSW).

(4) YFG Signal polarity 1
..... Selected by the YFGPO1 flag of the OPTION register (x'3F83') (linked to the input capture timing and YSP timing).

Note: This cannot be used if the analog power supply has not been made ON. (STBH Flag of register ANACNT)

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

YFGLV1	YFGLV0	Interrupt request level setting (bp7 and 6 of register YFGICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The cylinder FG interrupt is disabled)

YFGIE	Cylinder FG interrupt enable (bp1 of register YFGICR)
0	Interrupt disabled
1	Interrupt enabled

YFGIR	Cylinder FG interrupt request (bp0 of register YFGICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (YFGICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (YFGICR) 1       ; Enables the YFG interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

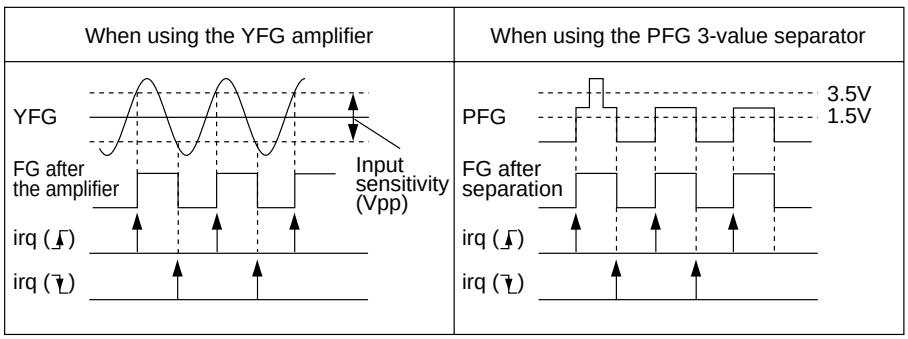


Figure 4-2-4 Timing Diagram of Cylinder FG Interrupt

4-2-6 Capstan FG Interrupt

- Interrupt vector x'04024'
- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled)
- Interrupt cause This interrupt is generated at the rising edge of the valid capstan FG signal output after the signal input to the FGF pin is subjected to the specified signal processing.
- Interrupt source selection

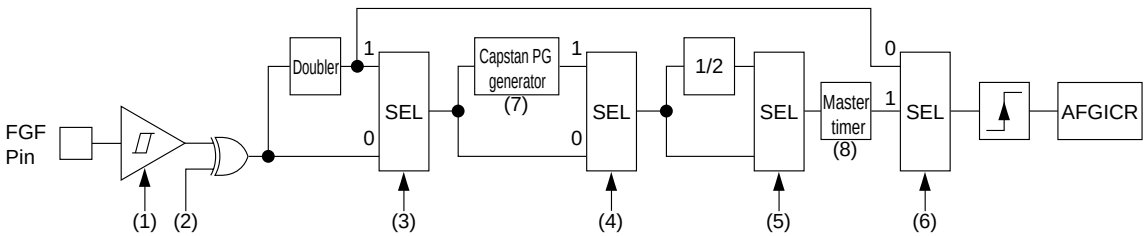


Figure 4-2-5 Block Diagram of Capstan FG Interrupt Source

(1) C-FG Amplifier input sensitivity setting ..... Set by the FGIVT flag of the OPTION register (x'3F83'). (0 : 50mVpp, 1 : 100mVpp)

(2) C-FG Signal polarity ..... Specified by the FGIPOL flag of the OPTION register (x'3F83').

(3) Prescaler selection (1) ..... Specified by the AFGD0 flag of register AFGPR (x'3F86').

(4) AFG Frequency divider control ..... Selected by the AFGS flag of register AFGDIV (x'3F87').

(5) Prescaler selection (2) ..... Specified by the AFGD1 flag of register AFGPR (x'3F86').

(6) Interrupt signal selection ..... Specified by the AFGSEL flag of register TRCNT (x'3F85').

(7) AFG Frequency divider ratio setting ..... Set by the flags AFGDIV6 to AFGDIV0 of register AFGDIV (x'3F87').

(8) Master timer setting of the capstan FG signal .... Set in register AMSKTM (x'3F8D').

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

AFGLV1	AFGLV0	Interrupt request level setting (bp7 and 6 of register AFGICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The capstan FG interrupt is disabled)

AFGIE	Capstan FG interrupt enable (bp1 of register AFGICR)
0	Interrupt disabled
1	Interrupt enabled

AFGIR	Capstan FG interrupt request (bp0 of register AFGICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (AFGICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (AFGICR) 1       ; Enables the AFG interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

- Related items Figure 9-1-3, Figure 9-2-2

- Reference

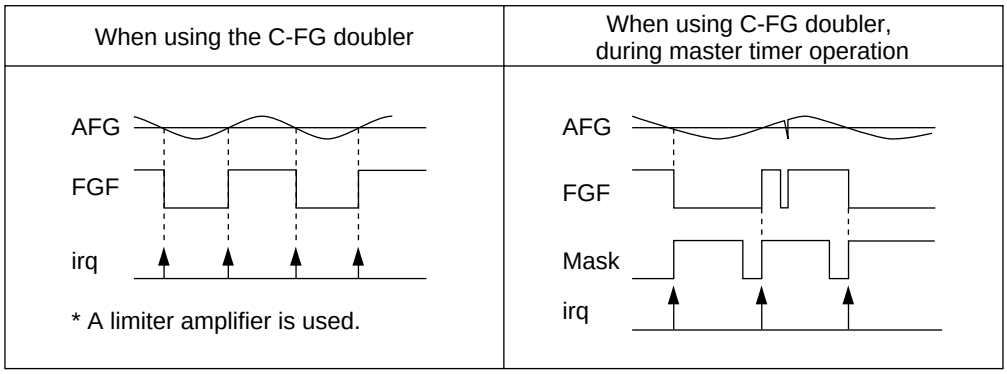


Figure 4-2-6 Timing Diagram of Capstan FG Interrupt

4-2-7 Timer 0 Interrupt

- Interrupt vector x'04028'
- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled)
- Interrupt cause Timer 0 overflow. The Timer 0 interrupt can be used as a cause for recovering from the HALT mode.
- Interrupt source selection

Timer 0 clock source specification				
Register TM1 (x'3F5B')		SXI Pin	The servoclk flag (bp0) of register CLKCNT (x'3FA4	
TC0M1 (bp1)	TC0M0 (bp0)		0	1
0	0	L	fosc/512	
		H	fxi	
0	1	—	ftc6	
1	0	—	fs/2	fs/4
1	1	—	fs/8	fs/16

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

TC0LV1	TC0LV0	Interrupt request level setting (bp7 and 6 of register TC0ICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The Timer 0 interrupt is disabled)

TC0IE	Timer 0 interrupt enable (bp1 of register TC0ICR)
0	Interrupt disabled
1	Interrupt enabled

TC0IR	Timer 0 interrupt request (bp0 of register TC0ICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (TC0ICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (TC0ICR) 1       ; Enables the Timer 0 interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

- Related items Section 5-1 Timer 0 Functions, Figure 5-1-1, Table 5-8-2

4-2-8 Timer 1 Interrupt

- Interrupt vector x'0402C'
- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled)
- Interrupt cause Timer 1 overflow.
- Interrupt source selection

Timer 1 clock source specification						Remarks
Register TM1 (x'3F5B')		Register ANACNT (x'3F81')	Register AFGPR (x'3F86')	Register CLKCNT (x'3FA4')		
TC1M1 (bp4)	TC1M0 (bp3)	REVH (bp2)	DREC (bp0)	servoclk(bp0)		
				0	1	
0	0	—	—	fs/8	fs/16	
0	1	—	—	fs/2	fs/4	
1	0	0 (Forward)	0 (Playback)	CTL Input positive pulse (_/‾_)		PCTL Signal
		1 (Reverse)		CTL Input negative pulse (‾_/‾)		
		—	1 (Record)	The output bp0 (RCTLD signal) of register RCTLBUF (x'3F7D')		

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

TC1LV1	TC1LV0	Interrupt request level setting (bp7 and 6 of register TC1ICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The Timer 1 interrupt is disabled)

TC1IE	Timer 1 interrupt enable (bp1 of register TC1ICR)
0	Interrupt disabled
1	Interrupt enabled

TC1IR	Timer 1 interrupt request (bp0 of register TC1ICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (TC1ICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (TC1ICR) 1       ; Enables the Timer 1 interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

- Related items Section 5-2 Timer 1 Functions, Figure 5-2-1, Table 5-8-2

4-2-9 Timer 2 Interrupt

- Interrupt vector x'04030'

- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled)

- Interrupt cause

TC2SEL1 (bp6)	TC2SEL0 (bp5)	Timer 2 interrupt cause specification: Register TM2 (x'3F5C')
0	0	Binary counter (BC2) overflow
0	1	Capture timing signal of ICR2
1	0	Underflow of down counter (SRBC2)
1	1	Match between the compare register (CMPR2) and the shift register (SR2F)

- Interrupt source selection

Binary counter (BC2) source clock selection						
Register TM2 (x'3F5C')			Register CLKCNT (x'3FA4')			
TC2M2 (bp7)	TC2M1 (bp4)	TC2M0 (bp0)	servoclk flag (bp0)			
			0		1	
0	0	0	fs/2		fs/4	
0	1	1	fs/12		fs/24	
1	1	1	fs/8		fs/16	
			VISSCTL Input = '0'	VISSCTL Input = '1'	VISSCTL Input = '0'	VISSCTL Input = '1'
0	0	1	fs/12	fs/8	fs/24	fs/16
1	0	1	fs/8	fs/12	fs/16	fs/24

Valid edge of Timer 2 capture operation (ICR2), shift register operation (SR2F)		
Register TM2 (x'3F5C') TC2EG(bp1)	Valid edge of the capture operation (ICR2)	Valid edge of shift register operation (SR2F)
0	Falling edge	Rising edge
1	Rising edge	Falling edge

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

TC2LV1	TC2LV0	Interrupt request level setting (bp7 and 6 of register TC2ICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The Timer 2 interrupt is disabled)

TC2IE	Timer 2 interrupt enable (bp1 of register TC2ICR)
0	Interrupt disabled
1	Interrupt enabled

TC2IR	Timer 2 interrupt request (bp0 of register TC2ICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing — The above control register is set as shown in the following sample program.

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (TC2ICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (TC2ICR) 1       ; Enables the Timer 2 interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

- Related items — Section 5-3 Timer 2 Functions, Figure 5-3-1, Table 5-8-2

4-2-10 Timer 3,4 Interrupts

- Interrupt vectors x'04034' (Timer 3), x'04038' (Timer 4)
- Interrupt level Maskable interrupt
- Interrupt cause Timer 3 or 4 overflow.
- Interrupt source selection

Timer 3 clock source specification		
Register TM1 (x'3F5B') TC3M (bp5)	The servoclk flag (bp0) of register CLKCNT (x'3FA4')	
	0	1
0	fs/8	fs/16
1	fs/2	fs/4

Timer 4 clock source specification		
Register TM1 (x'3F5B') TC4M (bp7)	The servoclk flag (bp0) of register CLKCNT (x'3FA4')	
	0	1
0	fs/8	fs/16
1	TC4I (Input through pin P15)	

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

TC3LV1	TC3LV0	Interrupt request level setting (bp7 and 6 of register TC3ICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The Timer 3 interrupt is disabled)

TC3IE	Timer 3 interrupt enable (bp1 of register TC3ICR)
0	Interrupt disabled
1	Interrupt enabled

TC3IR	Timer 3 interrupt request (bp0 of register TC3ICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (TC3ICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (TC3ICR) 1       ; Enables the Timer 3 interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

- Related items Section 5-4 Timer 3 Functions, Figure 5-4-1, Table 5-8-2
Section 5-5 Timer 5 Functions, Figure 5-5-1, Table 5-8-2

4-2-11 Timer 6 Interrupt

- Interrupt vector x'0403C'

- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled)

- Interrupt cause

TC6IRQ1 (bp2)	TC6IRQ0 (bp1)	Timer 6 interrupt cause specification: Register TC6CNT (x'3F6B')
0	0	2^{16} Count (overflow)
0	1	2^{15} Count
1	0	2^{14} Count
1	1	2^{13} Count

The Timer 6 interrupt can be used as a cause for recovering from the HALT mode.

- Interrupt source selection

Timer 6 clock source specification				
Register TC6CNT (x'3F6B')		SXI Pin	The servoclk flag (bp0) of register CLKCNT (x'3FA4')	
TC6CK1 (bp6)	TC6CK0 (bp5)		0	1
0	0	L	fosc/512	
		H	fxi	
1	0	—	fs/4	fs/8
1	1	—	fs/64	fs/128

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

TC6LV1	TC6LV0	Interrupt request level setting (bp7 and 6 of register TC6ICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The Timer 6 interrupt is disabled)

TC6IE	Timer 6 interrupt enable (bp1 of register TC6ICR)
0	Interrupt disabled
1	Interrupt enabled

TC6IR	Timer 6 interrupt request (bp0 of register TC6ICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (TC6ICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (TC6ICR) 1       ; Enables the Timer 6 interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

- Related items Section 5-7 Timer 6 Functions, Figure 5-7-1, Table 5-8-2, Table 10-1-6

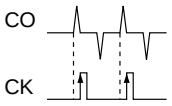
4-2-12 Control Interrupt

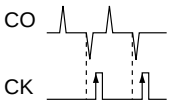
- Interrupt vector x'04040'

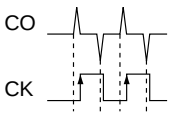
- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled)

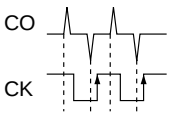
- Interrupt causes
  - During the playback mode (DREC flag =0), an interrupt is generated when the CLK frequency divider counter becomes equal to register CTLDIV.
  - During the record mode (DREC flag =1), an interrupt is generated when the AFG frequency divider counter becomes equal to register AFGDIV.

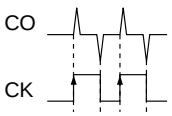
- Interrupt source selection

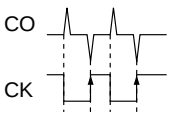
<table>
<tr><td rowspan="2">DREC Flag (bp0) of register AFGPR (x'3F86')</td><td colspan="2">Register TPGCNT x'3F84'</td><td colspan="2" rowspan="2">REVH Flag (bp2) of register ANACNT (x'3F81')</td><td rowspan="2">Frequency division ratio setting register before interrupt</td></tr>
<tr><td>CTLSEL Flag (bp3)</td><td>CTLS Flag (bp5)</td></tr>
<tr><td></td><td></td><td></td><td>Forward 0</td><td>Reverse 1</td><td></td></tr>
<tr><td rowspan="3">Playback mode 0</td><td>0</td><td>—</td><td> 

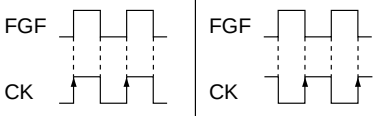
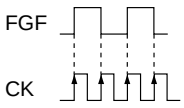
Noise elimination present</td><td> 

Noise elimination present</td><td rowspan="3">Register CTLDIV (x'3F89) bits 6 to 0</td></tr>
<tr><td>1</td><td>0</td><td> 

Noise elimination present</td><td> 

Noise elimination present</td></tr>
<tr><td>1</td><td>1</td><td> 

Noise elimination absent</td><td> 

Noise elimination absent</td></tr>
<tr><td rowspan="3">Record mode 1</td><td colspan="2" rowspan="2">AFGD0 Flag (bp2) of register AFGPR (x'3F86')</td><td colspan="2">FGIPOL Flag (bp5) of the OPTION register (x'3F83')</td><td rowspan="3">Register AFGDIV (x'3F87) bits 6 to 0</td></tr>
<tr><td>Positive polarity 0</td><td>Negative polarity 1</td></tr>
<tr><td>0
(x1)</td><td colspan="2"> 
</td></tr>
<tr><td colspan="2">1
(Doubling)</td><td colspan="2"> 
</td></tr>
</table>

- Capstan FG input sensitivity setting

  Set by the FGIVT flag (bp4) of the OPTION register (x'3F83')

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

CTLLV1	CTLLV0	Interrupt request level setting (bp7 and 6 of register CTLICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The control interrupt is disabled)

CTLIE	Control interrupt enable (bp1 of register CTLICR)
0	Interrupt disabled
1	Interrupt enabled

CTLIR	Control interrupt request (bp0 of register CTLICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

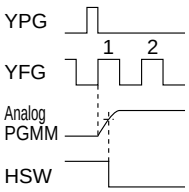
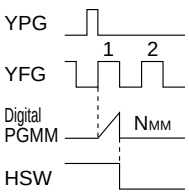
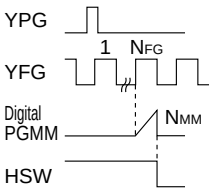
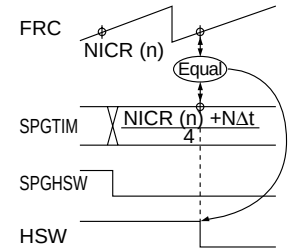
```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (CTLICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (CTLICR) 1       ; Enables the control interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSWÅB
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

- Related items
  - Figure 9-1-5 Block Diagram of CTL Amplifier
  - Figure 9-2-7 CTL Signal Processing Section
  - Figure 9-1-3 Capstan FG Amplifier Configuration
  - Figure 9-2-2 Capstan FG Signal Processing Section
  - Figure 9-2-3 Capstan FG Mask Timer Section

4-2-13 HSW Both Edge Interrupt

- Interrupt vector x'04044'
- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled)
- Interrupt cause Interrupts are generated at both edges of the HSW signal.
- Interrupt source selection

<table>
<tr><th rowspan="2">HSWSEL Flag (bp0) of register TRCNT (x'3F85')</th><th colspan="2">Register DPGMCNT (x'3F8A')</th><th rowspan="2">HSW Generation timing</th><th rowspan="2">PGMM Start timing</th><th rowspan="2">HSW Change timing</th></tr>
<tr><th>DPGMS Flag (bp7)</th><th>PGMS Flag (bp6)</th></tr>
<tr><td rowspan="3">0</td><td>0</td><td>0</td><td>(Analog PGMM)
 
</td><td rowspan="2">Started YFG counter = '1' after YPG, or the next time YFG counter becomes '1' after the YFGDIV register (x'3F88') becomes equal to the YFG counter.</td><td>The charging waveform at the PGMM pin starts at the PGMM start timing, and changes when the pin voltage exceeds the input threshold level.</td></tr>
<tr><td>0</td><td>1</td><td>(Digital PGMM)
 
</td><td rowspan="2">The digital PGMM starts counting up at the PGMM start timing, and changes when the set values (N_{MM}) becomes equal to registers PGMMTIM (x'3F8B', x'3F8C')</td></tr>
<tr><td>1</td><td>1</td><td>(Digital PGMM)
 
</td><td>Started at the timing (N_{FG}) specified by DPGMC5 to 0 for the YFG counter after YPG.</td></tr>
<tr><td>1</td><td>—</td><td>—</td><td>(FRC becomes equal to SPGTIM)
 
</td><td colspan="2">When FRC becomes equal to SPGTIM, the value set in the SPGHSW flag is output to HSW.</td></tr>
</table>

The values YPG and YFG in the above table are specified by the settings in the following registers.

(1) Selection between the YFG amplifier and the PFG 3-value input

..... Selected by the setting of the PFGS flag in register ANACNT (x'3F81')

(0 : PFG, 1 : YFG)

PFGS Flag	0	1
YPG YFG Input timing	PFG Pin 3-Value separator YPG 3-Value separator YPG	YPG Pin YFG Pin YPG after the amplifier YFG after the amplifier Input sensitivity (Vpp) Input sensitivity (Vpp)

* The 3-value separator YPG/YFG and the YPG/YFG after the amplifier in the above figures are the fundamental signals.

(2) YPF Amplifier input sensitivity selection

..... Selected by the YPGVT flag (bp7) of the OPTION register (x'3F83').

(0: 50mVop, 1: 100mVop)

(3) YPG Input polarity selection

..... Selected by the YPGPOL flag (bp6) of the OPTION register (x'3F83').

(0: Positive, 1: negative)

(4) HSW Signal polarity after PG

..... Selected by the HSWPOL0 flag of the OPTION register (x'3F83').

(0: HSW Rising edge after PG, 1: falling edge of HSW after PG)

* Set also the contents described in the interrupt source selection in Section 4-2-5 "Cylinder FG Interrupt".

Note: Although the HSW interrupt is always generated at both edges, the edge of HSW capture is selected by the HSWS flag (bp5) of register TRCNT (x'3F85'). (0: Rising edge of the HSW signal, 1: falling edge of the HSW signal)

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

HSWLV1	HSWLV0	Interrupt request level setting (bp7 and 6 of register HSWICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The HSW interrupt is disabled)

HSWIE	HSW Interrupt enable (bp1 of register HSWICR)
0	Interrupt disabled
1	Interrupt enabled

HSWIR	HSW Interrupt request (bp0 of register HSWICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (HSWICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (HSWICR) 1       ; Enables the HSW interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

- Related items Section 9-2-1 Servo Signal Processing
Figure 9-2-1 Cylinder PG/FG Signal Processing Section
Figure 9-2-4 Block Diagram of HSW Generation Section
Figure 8-3-2 Synchronization Output Section 1 (SPG)

4-2-14 Servo VSYNC Interrupt

- Interrupt vector — x'04048'
- Interrupt level — Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), level 2 (normal), or level 3 (disabled)
- Interrupt cause — The interrupt is generated at the timing of the rising edge of the VSYNC signal that is obtained by carrying out (or also, not carrying out) the VSYNC mask processing and VSYNC escape interpolation processing for the VSYNC signal selected between the vertical synchronization component separated from the signal input at the VSYNC pin or from the signal input at the CVIN2 pin. This is used as the VSYNC for the servo.

- Interrupt source selection

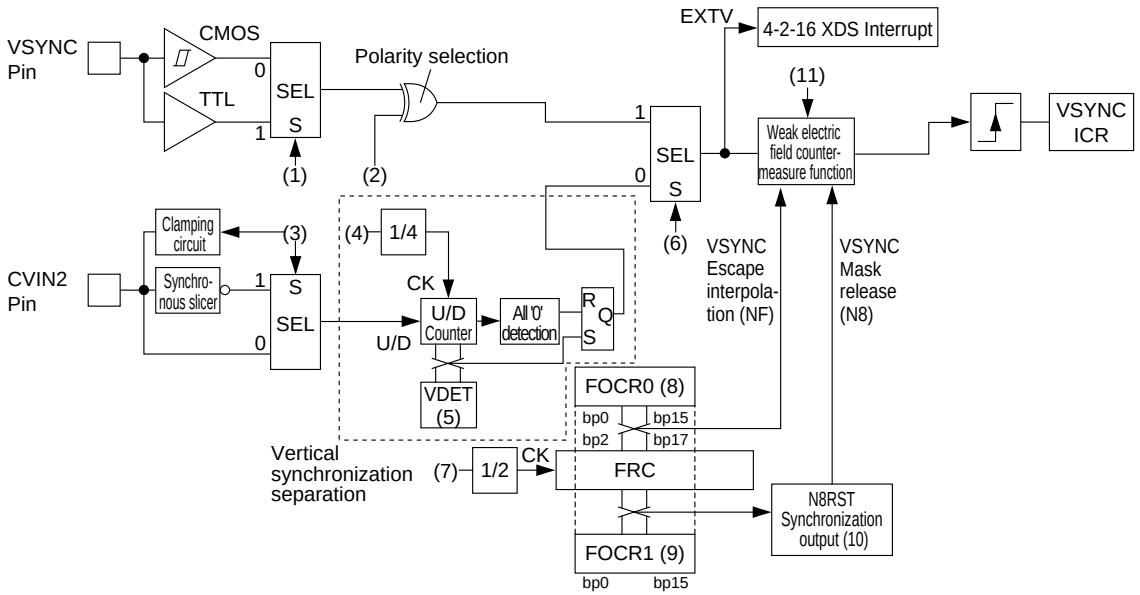


Figure 4-2-7 Block Diagram of Servo VSYNC Interrupt Source

(1) VSYNC Input level setting
..... Set by the P6DIR4 flag (bp4) of register P6DIR (x'3F32').
(0: CMOS Input (with Schmitt circuit), 1: TTL Input (without Schmitt circuit)

(2) VSYNC Input polarity setting
..... Set by the P20SEL flag (bp0) of register P2SEL (x'3F39').
(0: Positive polarity, 1: Negative polarity)

(3) CVIN2 Pin input signal setting
..... Set by the CSCLPW flag (bp3) of register CLCNT (x'3FD1').

0: Specified when the clamp power is OFF, and the digital signal after synchronization slicing is input
1: Specified when the clamp power is ON and the synchronization slicer is used, and when the video signal is input. (PB4 Output is also OFF)

(4) OSD Section operating clock (4fsc) setting
..... Set by bp7 and bp6 of register OSDCNT1 (x'3FC8').
(00: fosc, 01: 4-times clock (fsc), 10: fosc2)
(5) VSYNC Separator circuit VSYNC detection sensitivity setting
..... Register VDET (x'3FDE')
(6) External synchronization signal selection
..... EXSSEL Flag (bp5) of register OSDCNT1 (x'3FC8')
(0: Synchronization separator output, 1: VSYNC/HSYNC Pin input)
(7) Servo section operating clock (fss) setting
..... The servoclk flag (bp0) of register CLKCNT (x'3FA4')
(0: fs, 1: fs/2)
(8) Setting the VSYNC escape interpolation timing (NF)
..... Set by register FOCR0 (x'2E17, 16')
(9) Setting the VSYNC mask release timing (N8)
..... Set by register FOCR1 (x'2E19, 18')
(10) N8RST signal and synchronization signal output setting
- Synchronization output buffer
..... N8RST Flag (bp4) of register HOCRBUF1 (x'3F7B')
- Synchronization output data
..... HOCRDAT14 Flag (bp4) of register HOCRDAT1 (x'3F79')
- Setting the operation of register HOCRBUF1
..... Set by the HOCRSEL1 flag (bp2) of register HOCRCNT (x'3F7C')
(0: Normal register operation, 1: Synchronization output operation)
(11) Specification of the presence/absence of the VSYNC weak electric field countermeasure function
..... N8SEL Flag (bp1) of register SPEC (x'3FDF')
(0: Absent, 1: Present)

Note: The VSYNC signal for servo processing and the VSYNC signal for the OSD display are different. See Figure 11-1-1 OSD Section Block Diagram, for details.

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

VSYLV1	VSYLV0	Interrupt request level setting (bp7 and 6 of register VSYICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The servo VSYNC interrupt is disabled)

VSYIE	Servo VSYNC interrupt enable (bp1 of register VSYICR)
0	Interrupt disabled
1	Interrupt enabled

VSYIR	Servo VSYNC interrupt request (bp0 of register VSYICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (VSYICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (VSYICR) 1       ; Enables the servo VSYNC interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

- Related items
Figure 3-2-12 P20 (VSYNC) Configuration
Figure 3-2-37 PB4 (CVIN2) Configuration
Figure 11-1-1 OSD Section Block Diagram
Figure 9-2-8 VSYNC Input Processing Section
Figure 8-4-3 Synchronization Output Section 2
Figure 11-4-1 Synchronization Separator Block Diagram

4-2-15 SSPG Interrupt, FOCR0, 1, 2 Interrupts

- Interrupt vectors

SPG : x'0404C'
FOCR0 : x'04050'
FOCR1 : x'04054'
FOCR2 : x'04058'

- Interrupt level — Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), or level 2 (normal)

- Interrupt cause — Generated at the timing of a match of bp2 to bp17 of the free running counter (FRC) with each of register SPGTIM, register FOCR0, register FOCR1, and register FOCR2.

- Interrupt source selection

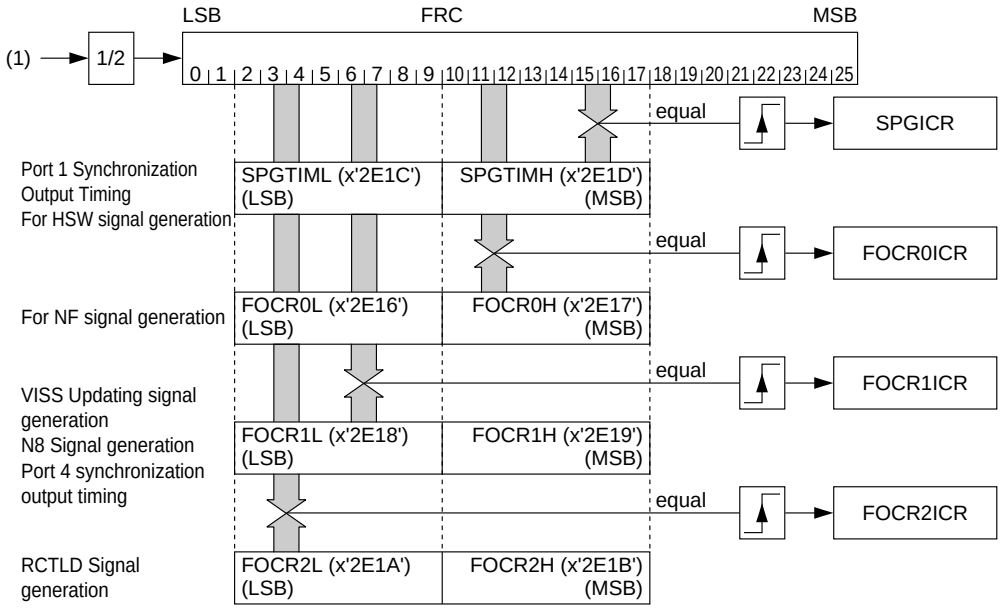


Figure 4-2-8 Block diagram of Synchronization Interrupt Source

(1) Servo section operating clock (fss) setting
..... The servoclk flag (bp0) of register CLKCNT (x'3FA4')
(0 : fs, 1 : fs/2)

Resolution of synchronization output timing, maximum setting time (@fosc = 14.32MHz)		
servoclk flag	Resolution	Maximum setting time
0	1.117µs	73.224ms
1	2.234µs	146.449ms

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

SPGLV1	SPGLV0	Interrupt request level setting (bp7 and 6 of register SPGICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The SPG interrupt is disabled)

SPGIE	SPG interrupt enable (bp1 of register SPGICR)
0	Interrupt disabled
1	Interrupt enabled

SPGIR	SPG interrupt request (bp0 of register SPGICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

* The FOCR0 interrupt is controlled by register FOCR0ICR,
the FOCR1 interrupt by register FOCR1ICR, and
the FOCR2 interrupt by register FOCR2ICR.

- Interrupt processing

The above control register is set as shown in the following sample program.

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (SPGICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (SPGICR) 1       ; Enables the SPG interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

4-2-16 OSD Interrupt

- Interrupt vector x'0405C'

- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), or level 2 (normal)

- Interrupt causes (1) When there is no inter-line setting (register VOFS = 0), the interrupt is generated at the timing of the start of the horizontal synchronization signal at the end of display of the specified display line.

  (2) When there is an inter-line setting (register VOFS ≥1), the interrupt is generated at the timing of the start of the next horizontal synchronization signal after the inter-line display is completed following the display of the specified display line.

- Interrupt source selection

* See Section 11-1, OSD Functions, for how to generate the OSDV and OSDH signals.

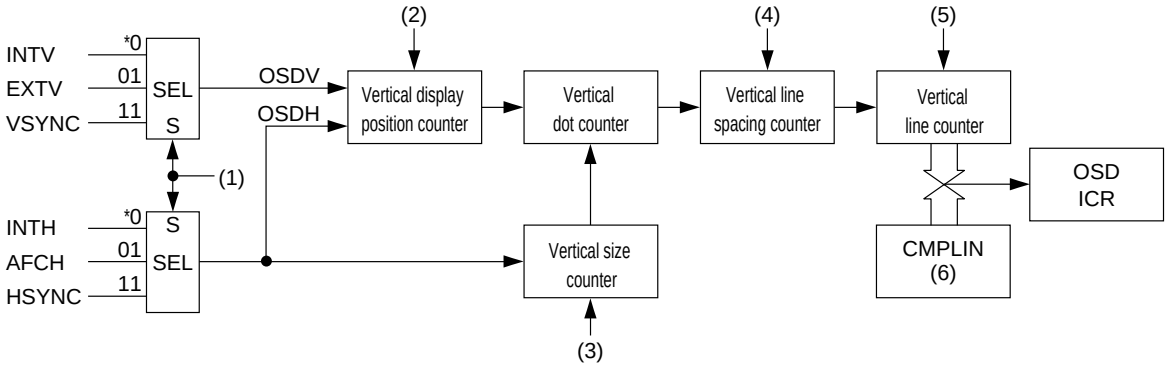


Figure 4-2-9 Block diagram of OSD Interrupt Source

(1) Synchronization mode setting
..... Set by the EXSSEL flag (bp5) and the SYNMOD flag (bp4) of register OSDCNT1 (x'3FC8').

(*0: Internal synchronization mode (selects INTV/INTH)
01: External synchronization mode (selects the sync separator section output EXTV/AFCH)
11: External synchronization mode (selects the VSYNC/HSYNC pin inputs))

(2) Vertical display start position setting
..... Register VP (x'3FD8')

(3) Vertical direction character size setting
..... The VS1 and VS0 flags (bp7 and bp6) of the line control VRAM0 (CN0)
(00: Normal size, 10: triple size, 01: double size, 11: quadruple size)

(4) Display inter-line setting
..... Set by register VOFS (x'3FDA).
(5) Number of display lines specification
..... VLN2 to VLN0 (bp2 to bp0) of register VLIN (x'3FDB')
The value (number of screen display lines/2) is set here.
(6) OSD Interrupt display line setting
..... Set by register CMPLIN (x'3FDC').

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

OSDLV1	OSDLV0	Interrupt request level setting (bp7 and 6 of register OSDICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The OSD interrupt is disabled)

OSDIE	OSD interrupt enable (bp1 of register OSDICR)
0	Interrupt disabled
1	Interrupt enabled

OSDIR	OSD interrupt request (bp0 of register OSDICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (OSDICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (OSDICR) 1       ; Enables the OSD interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

- Related item Section 11-1 OSD Functions

4-2-17 XDS Interrupt

- Interrupt vector x'04060'
- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), or level 2 (normal)
- Interrupt cause The interrupt is generated when the horizontal line counter that counts the HSYNC pulses after VSYNC becomes equal to register XDSLIN2.
- Interrupt source selection

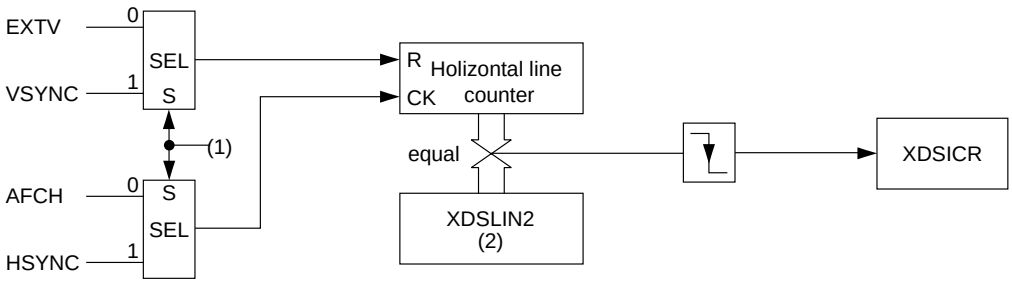


Figure 4-2-10 Block Diagram of XDS Interrupt Source

(1) External synchronization mode setting
..... Set by the EXSSEL flag (bp5) of register OSDCNT1 (x'3FC8').
(0: Selects the sync separator section outputs EXTV/AFCH, 1: selects the VSYNC/HSYNC pin inputs)

(2) XDS Interrupt display line setting
..... Set by register XDSLIN2 (x'3FC7').

* See Section 11-1, OSD Functions, for how to generate the EXTV/VSYNC and AFCH/HSYNC signals.

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

XDSLV1	XDSLV0	Interrupt request level setting (bp7 and 6 of register XDSICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The XDS interrupt is disabled)

XDSIE	XDS interrupt enable (bp1 of register XDSICR)
0	Interrupt disabled
1	Interrupt enabled

XDSIR	XDS interrupt request (bp0 of register XDSICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (XDSICR)  ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (XDSICR) 1       ; Enables the XDS interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

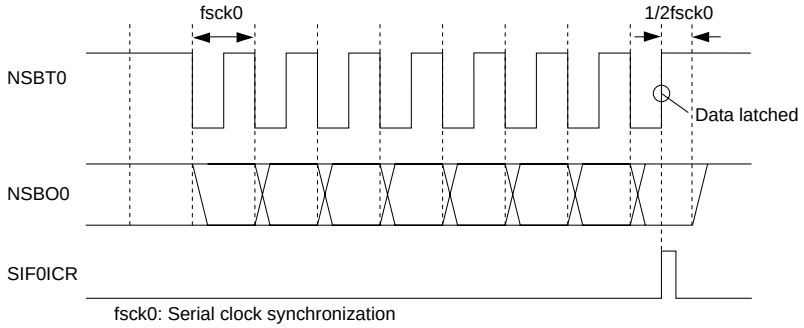
- Related item Section 11-2 XDS Data Slice Function

4-2-18 Serial 0 Interrupt

- Interrupt vector x'04064'

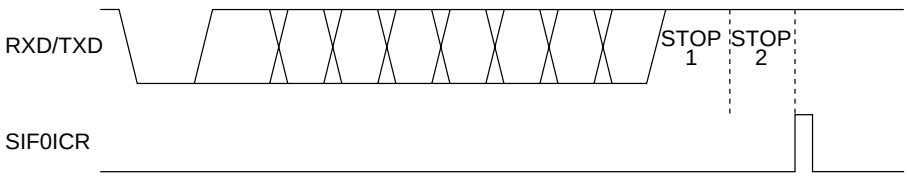
- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), or level 2 (normal)

- Interrupt causes (1) In the case of a synchronous serial interface
  The interrupt is generated when the transmission of the specified number of send bits of serial data has been completed (at the data latch timing of the last serial data).



The pin NSBO0 gets tied to the High level after the transmission has been completed. The timing at which this pin gets tied to the High level is one half of the clock period after the data latching of the final send data.

(2) In the case of a start-stop synchronization serial interface (UART) The interrupt is generated when the transmission of the specified number of stop bits (1 or 2) has been completed after sending the UART data.



- Interrupt source selection

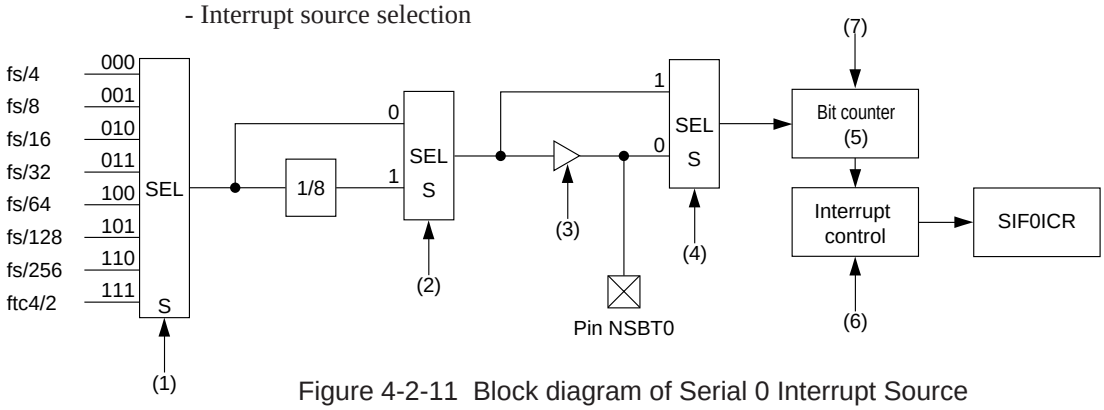


Figure 4-2-11 Block diagram of Serial 0 Interrupt Source

(1) Clock source specification (internal clock)

..... Set by the SC0CK2 to SC0CK0 flags (bp2 to bp0) of register SC0MD1 (x'3F98'.)

(000: fs/4 100: fs/64 001: fs/8 101: fs/128
010: fs/16 110: fs/256 011: fs/32 111: ftc4/2)

(2) 1/8 Frequency division specification

..... Specified by the SC0CKM flag (bp6) of register SC0MD1 (x'3F98').

(0: No 1/8 frequency division (synchronous mode only),
1: 1/8 frequency division present (synchronous mode/start-stop mode))

(3) - Internal/external clock transfer setting

Set by the P0DIR2 flag (bp2) of register P0DIR (x'3F2E').

(0: Input (using external clock transfer), 1: output (using internal clock transfer))

- Selection of the NSBT0 pin function

..... Set by the SC0SBTS flag (bp0) of register SC0MD3 (x'3F9A').

(0: Port, 1: serial clock pin)

- Setting of the NSBT0 pin output format

..... Set by the SC0SBTM flag (bp3) of register SC0MD3 (x'3F9A').

(0: Push-pull (CMOS) output, 1: N-Channel open drain output)

(4) Synchronous/start-stop (UART) mode setting

..... Set by the SC0CMD flag (bp6) of register SC0CTR (x'3F9B').

(0: Synchronous mode serial communication, 1: start-stop mode (UART))

(5) Specification of the number of transferred bits

..... Set by the SC0LNG2 to SC0LNG0 flags (bp2 to bp0) of register SC0MD0 (x'3F97').

(000: 8 Bits, 100: 4 bits, 001: 7 bits, 101: 3 bits, 010: 6 bits, 110: 2 bits,
011: 5 bits, 111: 1 bit)

(6) Frame mode specification when UART transfer is used

..... Specified by the SC0FM1 to SC0FM0 flags (bp4 to bp3) of register SC0MD2 (x'3F99').

(0: 7 Data bits + 1 stop bit, 01: 7 data bits + 2 stop bits,
10: 8 data bits + 1 stop bit, 11: 8 data bits + 2 stop bits)

(7) Start specification when internal clock transfer is used

..... Written into register SC0TRB (x'3F95').

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

SI0LV1	SI0LV0	Interrupt request level setting (bp7 and 6 of register SIF0ICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The Serial 0 interrupt is disabled)

SI0IE	Serial 0 interrupt enable (bp1 of register SIF0ICR)
0	Interrupt disabled
1	Interrupt enabled

SI0IR	Serial 0 interrupt request (bp0 of register SIF0ICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

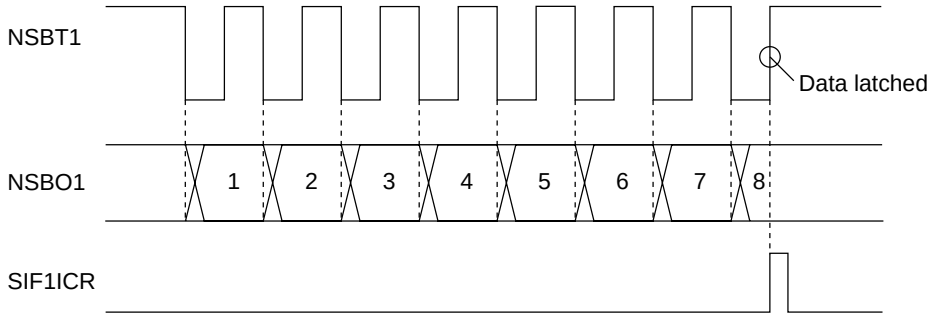
Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (SIF0ICR) ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (SIF0ICR) 1      ; Enables the Serial 0 interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

- Related items Section 6-1 Serial 0 Functions
Figure 6-1-7 Serial 0 Block Diagram

4-2-19 Serial 1 Interrupt

- Interrupt vector x'04068'
- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), or level 2 (normal)
- Interrupt cause The interrupt is generated when the transfer of one byte of serial data is completed (at the time of data latching of the 8th bit).



- Interrupt source selection

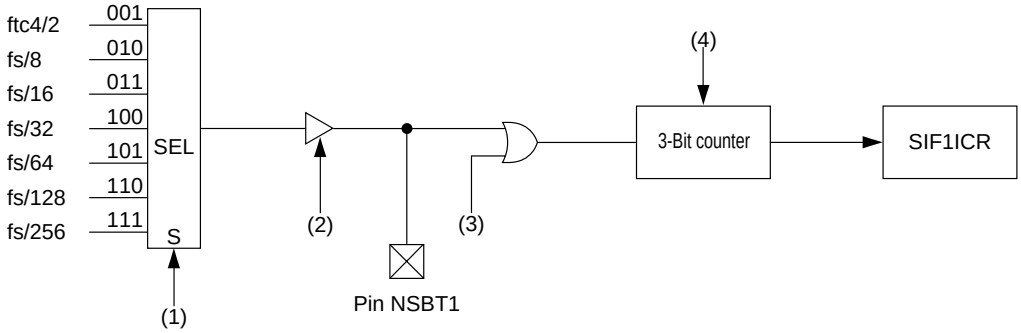


Figure 4-2-12 Block diagram of Serial 1 Interrupt Source

(1) Clock source specification (internal clock)
..... Set by the SI1CM2 to SI1CM0 flags (bp2 to bp0) of register SIM1 (x'3F9C').

(000 : External clock 100 : fs/32 001 : ftc4/2 101 : fs/64
010 : fs/8 110 : fs/128 011 : fs/16 111 : fs/256)

(2) Internal/external clock transfer setting
- Selection of the NSBT1 pin function
..... Set by the SI1OE flag (bp0) of register SIC1 (x'3F9E').
(0: Port input/output, 1: serial input/output (SBT1, SBO1))
- Setting of the NSBT1 pin output format
..... Set by the SI1OD flag (bp6) of register SIC1 (x'3F9E').
(0: Push-pull (CMOS) output, 1: N-Channel open drain output)

(3) Serial 1 clock enable/disable setting

..... Specified by the SI1CKE flag (bp1) of register SIC1 (x'3F9E').

(0: Enable, 1: disable)

(4) Start specification when using internal clock transfer

..... Written into register SIBUF1 (x'3F9D).

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

SI1LV1	SI1LV0	Interrupt request level setting (bp7 and 6 of register SIF1ICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The Serial 1 interrupt is disabled)

SI1IE	Serial 1 interrupt enable (bp1 of register SIF1ICR)
0	Interrupt disabled
1	Interrupt enabled

SI1IR	Serial 1 interrupt request (bp0 of register SIF1ICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (SIF1ICR) ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (SIF1ICR) 1      ; Enables the Serial 1 interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

4-2-20 Serial 2 Interrupt

- Interrupt vector x'0406C'

- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), or level 2 (normal)

- Interrupt causes
  1. When the transfer of one byte is completed (including when the arbitration has been lost).
  2. When the slave address is recognized by the addressing format. Otherwise, when a general address (x'00') is received (during the ALS=0 slave address recognition mode).
  3. When the data is received in the free data format (during the ALS=1 slave address non-recognition mode).

- Interrupt source selection
  - Since the specifications of this I²C operation are complex, they will be omitted here. See Section 6-3, Serial 2 Functions, for details of the settings.

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

SI2LV1	SI2LV0	Interrupt request level setting (bp7 and 6 of register SIF2ICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The Serial 2 interrupt is disabled)

SI2IE	Serial 2 interrupt enable (bp1 of register SIF2ICR)
0	Interrupt disabled
1	Interrupt enabled

SI2IR	Serial 2 interrupt request (bp0 of register SIF2ICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing The above control register is set as shown in the following sample program.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (SIF2ICR) ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (SIF2ICR) 1      ; Enables the Serial 2 interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

- Related item Section 6-3 Serial 2 Functions

4-2-21 AD/PWM14 Interrupt

- Interrupt vector x'04070'
- Interrupt level Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), or level 2 (normal)
- Interrupt causes
  1. The interrupt is generated at the end of A/D conversion (PWM14S = 0).
  2. The interrupt is generated at the falling edge (during positive polarity output) of the basic period of PWM14 (PWM14S = 1).

- Interrupt source selection

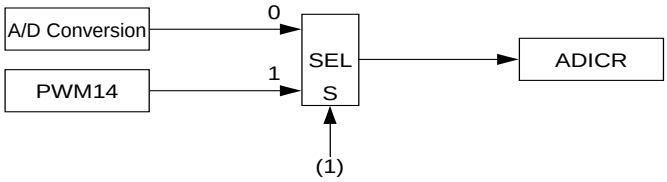


Figure 4-2-13 Block Diagram of AD/PWM14 Interrupt Source

(1) Interrupt source selection specification

..... Specified by the PWM14S flag (bp6) of register PWM14 (x'3F6F').

(0: AD interrupt, 1: PWM14 interrupt)

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

ADLV1	ADLV0	Interrupt request level setting (bp7 and 6 of register ADICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The AD/PWM14 interrupt is disabled)

ADIE	AD/PWM14 interrupt enable (bp1 of register ADICR)
0	Interrupt disabled
1	Interrupt enabled

ADIR	AD/PWM14 interrupt request (bp0 of register ADICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing — The above control register is set as shown in the following sample program.

```
At the time of initial setting (Processed when the task selection has stopped)
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (ADICR)   ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (ADICR) 1        ; Enables the AD/PWM14 interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

- Related items — Section 7-1 A/D Converter
  Section 9-4-3 PWM14 Settings

4-2-22 OSD VSYNC Interrupt

- Interrupt vector — x'04074'
- Interrupt level — Maskable interrupt
  Can be set to level 0 (highest priority), level 1 (priority), and level 2 (normal).
- Interrupt cause — The interrupt is generated at the rising edge of the TVSSG VSYNC signal when internal synchronization has been selected.
  When external synchronization has been selected, the interrupt is generated at the rising edge of the VSYNC signal obtained by vertical sync separation from the signal input at the pin CVIN2.

- Interrupt source selection

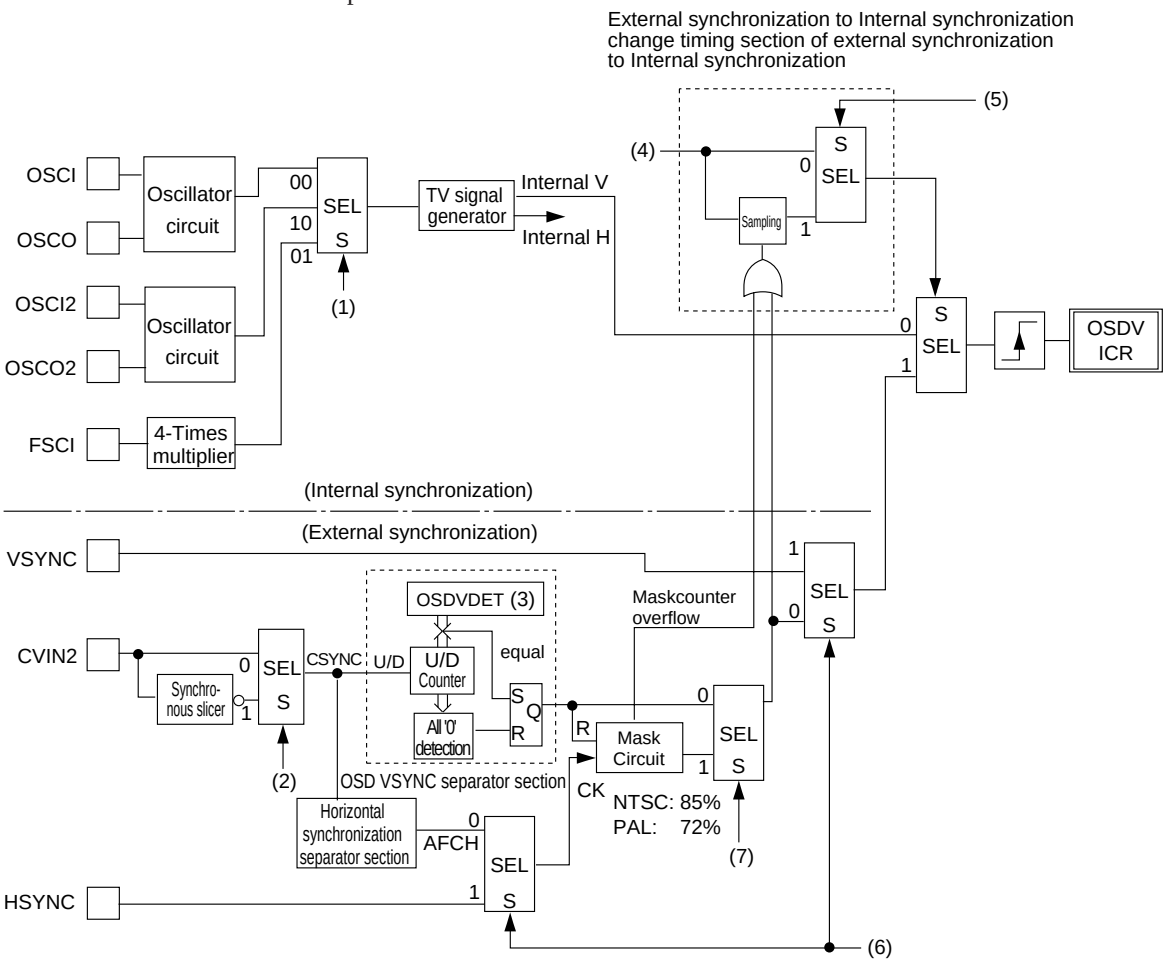


Figure 4-2-14 Block Diagram of OSD VSYNC Interrupt

(1) OSD Section operating clock selection

..... Set by the OSDCK1 flag (bp7) and the OSDCK0 flag (bp6) of register OSDCNT1 (x'3FC8').

(00: Operation using fosc,
01: Operation using the 4-times clock,
10: Operation using fosc2.
11: Specification not permitted)

(2) Pin CVIN2 input signal setting

..... Set by the CSCLPW flag (bp3) of register CLCNT (x'3FD1').

(0: Specified when the clamping power supply is OFF and when inputting the digital signal after synchronization slicing;
1: Specified when the clamping power supply is ON and the synchronization slicer is used, and when the video signal is input.)

(3) OSD VSYNC detection sensitivity setting of the VSYNC separator circuit

..... Register OSDVDET (x'3FBA')

(4) Synchronization mode selection

..... The SYNMOD flag (bp4) of register OSDXNT1 (x'3FC8').

(0: Internal synchronization mode, 1: external synchronization mode)

(5) Synchronization mode automatic change setting
..... The VSELM flag (bp0) of register OSDCNT3 (x'3FB9).
(0: Automatic change OFF, 1: automatic change ON)
(6) External synchronization signal selection
..... The EXSSEL flag (bp5) of register OSDCNT1 (x'3FC8').
(0: Synchronization separator section output, 1: HSYNC pin input)
(7) VSYNC Masking function ON/OFF selection
..... Set by the VMASK flag (bp7) of register DOTCNT (x'3FD2').

Caution: The VSYNC for servo processing and the VSYNC for the OSD display are different.

- Interrupt control register

MIE	IM1	IM0	Interrupt enable/level setting (bp6, 5, and 4 of PSW)
0	—	—	Disable maskable interrupt
1	0	0	
1	0	1	Interrupt mask level 1 (Interrupt is accepted only for interrupt request level 0)
1	1	0	Interrupt mask level 2 (Interrupt is accepted only for interrupt request levels 0 and 1)
1	1	1	Interrupt mask level 3 (Interrupt is accepted for interrupt request levels 0, 1, and 2)

VSYLV1	VSYLV0	Interrupt request level setting (bp7 and 6 of register OSDVICR)
0	0	Interrupt request level 0 (highest priority)
0	1	Interrupt request level 1 (priority)
1	0	Interrupt request level 2 (normal)
1	1	Interrupt request level 3 (The OSD VSYNC interrupt is disabled)

VSYIE	OSD VSYNC Interrupt enable (bp1 of register OSDVICR)
0	Interrupt disabled
1	Interrupt enabled

VSYIR	OSD VSYNC Interrupt request (bp0 of register OSDVICR)
0	No interrupt request is present.
1	Interrupt request has been generated (cleared automatically upon interrupt acceptance)

- Interrupt processing — The above control register is set as shown in the following sample program.

At the time of initial setting (Processed when the task selection has stopped)

```
MOV     PSW, Dn          ; Backs up PSW in Dn
AND     x'BF', PSW       ; Disables all maskable interrupts
BSET    (MEMCTR)2        ; Preparation for IR flag operation (IRWE='1')
MOV     x'00', (OSDVICR) ; Clears IR flag, request level 0
BCLR    (MEMCTR) 2       ; IRWE = '0'
BSET    (OSDVICR) 1      ; Enables the OSD VSYNC interrupt
MOV     Dn, PSW          ; Restores the backed up contents to PSW
```

Since the hardware automatically clears the IR flag upon acceptance of an interrupt, there is no need to clear the IR flag by software except during the initial setting in this microcomputer. When manipulating the IR flag by software, see the Appendix, Precautions in Manipulating the Interrupt Flag.

- Related items — Figure 3-2-37 PB4 (CVIN2) Configuration
Figure 11-1-1 Block Diagram of OSD Section
Figure 11-4-1 Block Diagram of Synchronization Separator

4-2-23 Undefined instruction Interrupt (during ICE operation)

The undefined instruction interrupt during modes other than ICE operation is generated as the non-maskable interrupt described in Section 4-2-2.

- Interrupt vector — x'0407C'
- Interrupt level — Non-maskable interrupt (with a priority lower than the reset interrupt and higher than maskable interrupts).
- Interrupt causes (during ICE operation)
  (1) This interrupt is generated when an undefined instruction code is executed.
  (2) This interrupt is generated when a write operation is made to an uninstalled ROM area of the memory. (The uninstalled ROM area is the area beyond the internal ROM area indicated for each type in the memory map of Figure 2-1-1 up to the last address location of x'3FFFF'.
- Return from interrupt — When executing a return from interrupt (rti) operation in the undefined instruction interrupt service routine, it is necessary to write x'00' in register DCR (x'3FE0').
- Precautions
  (1) The OSD display cannot be made during the processing of the undefined instruction interrupt.
  (2) Register DCR cannot be manipulated outside the undefined instruction interrupt service routine.

4-3 Task Control Section

4-3-1 Overview

In the MN101D02 Series, it is possible to automatically switch among a maximum of four tasks using two sets of registers.
The task switching sequence is started either by an underflow of the task timer register (TSKBC) or by an instruction. It is possible to set the counter value in the TSKBC register separately for each task. The number of execution cycles required for task changing is 11. The control of starting/stopping each task is done by the task start control register (TSKCTR).

4-3-2 Initial Processing of Tasks

Only Task 0 is started when starting after a reset. It is possible to start the second, third, or the fourth task depending on the need. A maximum of four tasks can be started.
The control of starting the tasks is carried out by registers TSKCTR and TSKMD. When starting several tasks, set values in the task timer (TSK0TT) and in the TSKBC register during the initial processing at the time of starting after a reset.

■ Task Timer Register (TSKBC)

Counting is stopped during a reset and when only one task has been started. Counting is started when multiple tasks have been started. The task switching operation is started when the task timer register underflows. At the time of task switching, the initial value for each task is set again in this register.

	7	6	5	4	3	2	1	0
TSKBC	TSKBC7	TSKBC6	TSKBC5	TSKBC4	TSKBC3	TSKBC2	TSKBC1	TSKBC0
During reset:	×	×	×	×	×	×	×	×

Figure 4-3-1 Task timer register (TSKBC : x'2EB0', R/W)

Table 4-3-1 Set Values and Task Switching Interval

Set value	Switching interval
00	2µsec
01	4µsec
⎛	
FE	510µsec
FF	512µsec

(The counter clock is fs/16 when fosc = 16MHz)
TSKBCSEL=0

■ Task Starting Control Register (TSKCTR)

It is not possible to stop all the tasks. When TSK1EN to TSK3EN all become '0', the TSK0EN flag is set to '1'.

	7	6	5	4	3	2	1	0
TSKCTR	TSKBCEN	TSKBCSEL	——	——	TSK3EN	TSK2EN	TSK1EN	TSK0EN
During reset:	1	0	–	–	0	0	0	1

TSK0EN	Task 0 start/stop control
0	Stop
1	Start

TSK1EN	Task 1 start/stop control
0	Stop
1	Start

TSK2EN	Task 2 start/stop control
0	Stop
1	Start

TSK3EN	Task 3 start/stop control
0	Stop
1	Start

TSKBCSEL	Count lock/selection of task timer
0	fs/16
1	fs/32

TSKBCEN	Task timer start/stop control
0	Stop
1	Start

- Valid only when multiple tasks have been started.
- Set at the time of task switching.

Figure 4-3-2 Task Starting Control Register (TSKCTR : x'3F05', R/W)

Make the entries as follows for stopping the counting operation by the task timer register (TSKBC):

```
Check   BCLR  (TSKCTR)TSKBCEN          ; Stop task timer.
        NOP
        TBNZ (TSKCTR)TSKBCEN,check    ; Verify that the
                                        task timer has
                                        stopped.
```

Always use the BSET instruction or the BCLR instruction for manipulating the TSKBCEN flag.

■ Task number Control Register (TSKMD)

The operation of switching to the specified task is made when a value is written in the flags TSKNXT1 and TSKNXT0.

	7	6	5	4	3	2	1	0
TSKMD	TSWBSY	——	TSKNOW1	TSKNOW0	——	——	TSKNXT1	TSKNXT0
During reset:	0	–	0	0	–	–	0	0

TSKNXT1	TSKNXT0	Number of the task to be executed next
0	0	Task 0
0	1	Task 1
1	0	Task 2
1	1	Task 3

Write both bits at the same time.

TSKNOW1	TSKNOW0	Number of the task currently being executed
0	0	Task 0
0	1	Task 1
1	0	Task 2
1	1	Task 3

These are read only flags

TSWBSY	Register save status
0	Saving completed
1	Saving in progress

These are read only flags
The data written in bits 7 to bit 2 are ignored.

Figure 4-3-3 Task Number Control Register (TSKMD : x'3F06', R/W)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
X'0000'	TSK0PC				TSK1PC				TSK2PC				TSK3PC				PC
X'0010'	RC0V		RC1V														
X'0020'																	
X'2EB0'	TSKBC																
X'2EC0'	TSK0SP		TSK0DW0		TSK0DW1		TSK0A0		TSK0A1		TSK0PSW	TSK0TT	TSK0HA				TSK0
X'2ED0'	TSK1SP		TSK1DW0		TSK1DW1		TSK1A0		TSK1A1		TSK1PSW	TSK1TT	TSK1HA				TSK1
X'2EE0'	TSK2SP		TSK2DW0		TSK2DW1		TSK2A0		TSK2A1		TSK2PSW	TSK2TT	TSK2HA				TSK2
X'2EF0'	TSK3SP		TSK3DW0		TSK3DW1		TSK3A0		TSK3A1		TSK3PSW	TSK3TT	TSK3HA				TSK3
	SP		DW0		DW1		A0		A1		PSW	TT	HA				

Figure 4-3-4 Register Value Storage Area of Different Tasks.

4-3-3 Method of Starting the Second Task

There are two methods of starting the second task.

■ Method of Starting Using Underflow of Task Timer

It is possible to start the second task by the underflow of register TSKBC, if the TSKnEN flag of register TSKCTR is set after initializing the values of TSKnPC, TSKnSP, TSKnPSW, TSKnHA, and TSKnTT of the task to be started. Register TSKBC starts counting when the number of tasks that have been started becomes more than one.

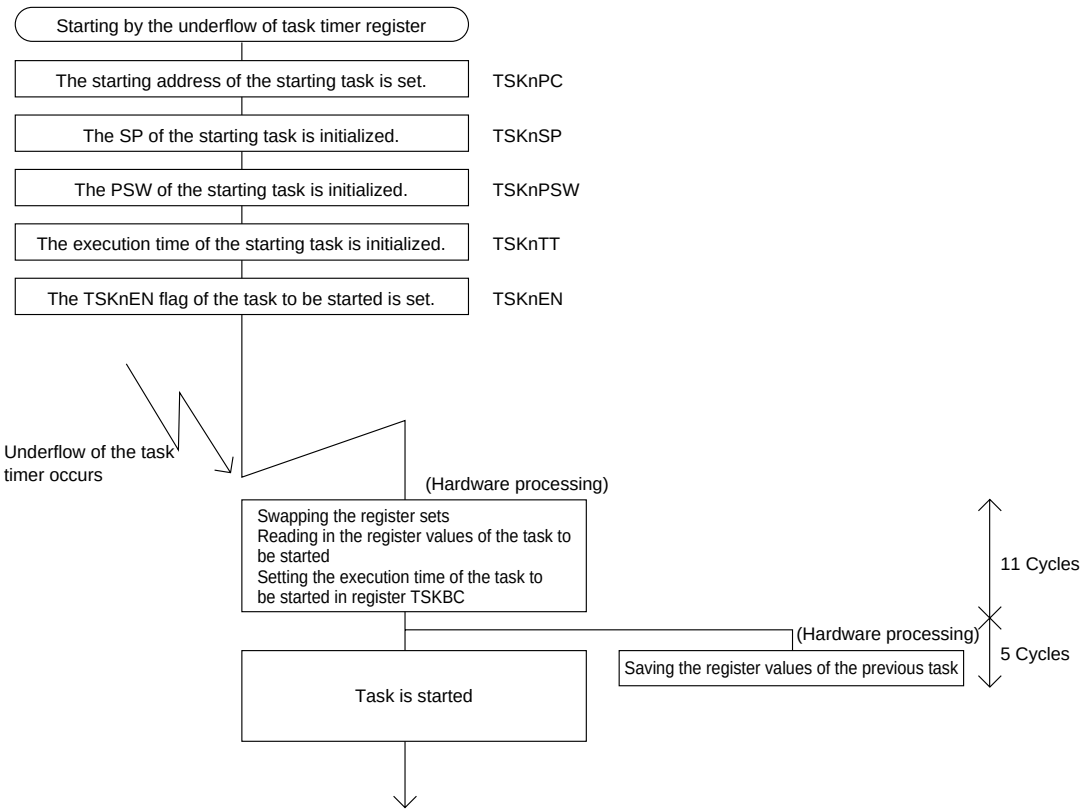


Figure 4-3-5 Starting by Underflow of Task Timer Register

■ Method of Starting Immediately Using an Instruction

It is possible to start the second task immediately without waiting an underflow of register TSKBC by writing the number of the task to be started in the TSKNXT1 and TSKNXT0 flags of the task number control register (TSKMD) after initializing the values of TSKnPC, TSKnSP, TSKnPSW, TSKnHA, and TSKnTT of the task to be started. Register TSKBC starts counting when the number of tasks that have been started becomes more than one.

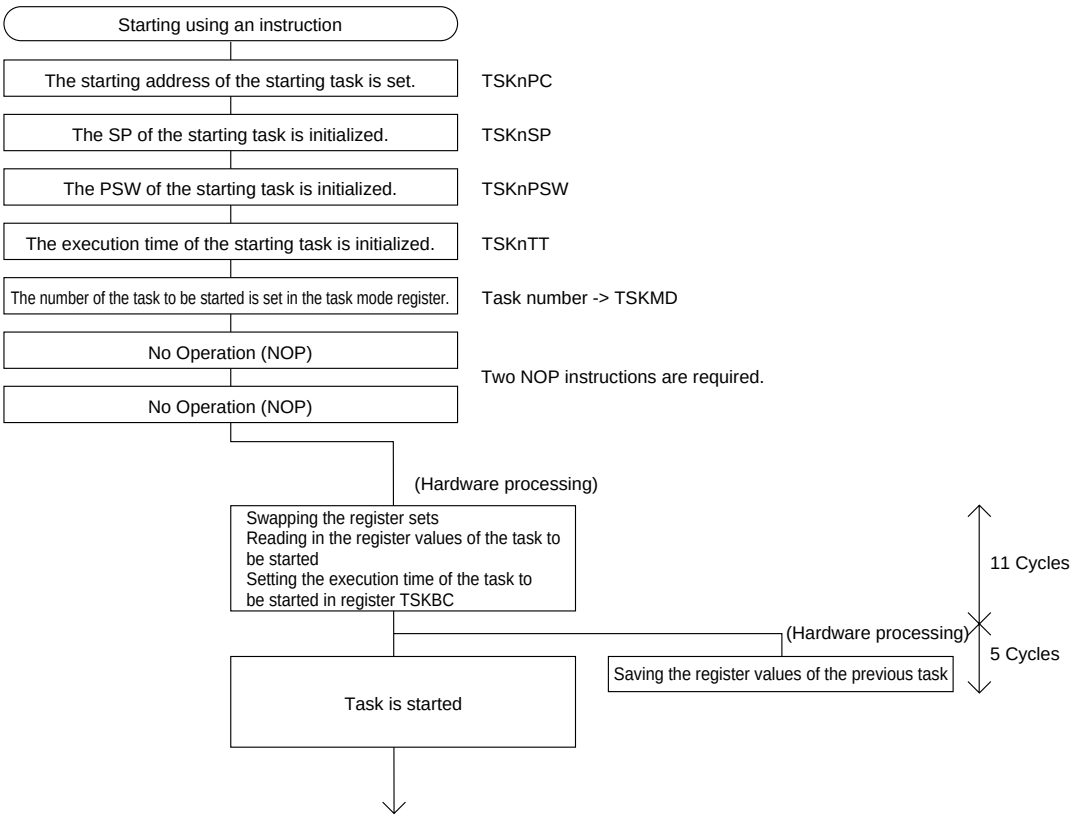


Figure 4-3-6 Starting Using an Instruction

When specifying the task number in the TSKNXT1 and TSKNXT0 flags of register TSKMD, specify both bits at the same time using a MOV instruction. (The data writing to the upper six bits is ignored.)

4-3-4 Method of Switching Tasks

■ Task Switching Sequence

The following operations are made in the sequence of task switching. The number of execution cycles required for these operations is 11 cycles.

(1) Changing the register sets
    Started task: Front
    Stopped task: Back

(2) Restore the values of the register set (front) of the started task from the RAM.
    PC, SP, An, Dn, PSW, HA, TT

(3) Set the count value in register TSKBC.
    TSKnTT → Register TSKBC

(4) Branches to the instruction pointed to by the PC of the started task.

Thereafter, the values of the register set (back) of the stopped task are saved in the RAM in preparation for the execution of the started task. The number of cycles required for saving the register set is 5.

It is not possible to switch to the next task until the saving operations are completed. When starting a new task, always carry out the initialization of the register set. After this task switching is completed, it is not possible to access the register value saving area of the different tasks (x'2EC0' to x'2EFF') for 5 cycles.

Carry out such accesses after stopping the task timer.

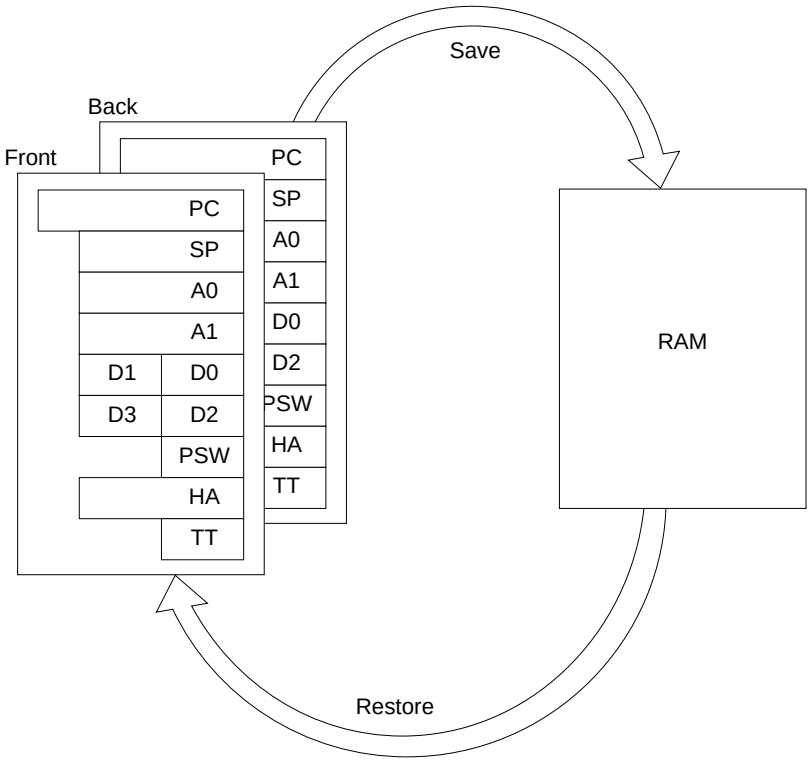


Figure 4-3-7 Swapping Register Sets

■ Method of Task Switching Based On Task Timer Underflow

The task switching sequence is started and the execution of the next task begins when an underflow is generated in register TSKBC, which is carrying out the counting operation. The order of task switching is the order of the task numbers for which the TSKnEN flag was set.

For example, when the start of Task 0, Task 1, and Task 3 has been enabled, the task switching is made in the order -

Task 0 → Task 1 → Task 3 → Task 0

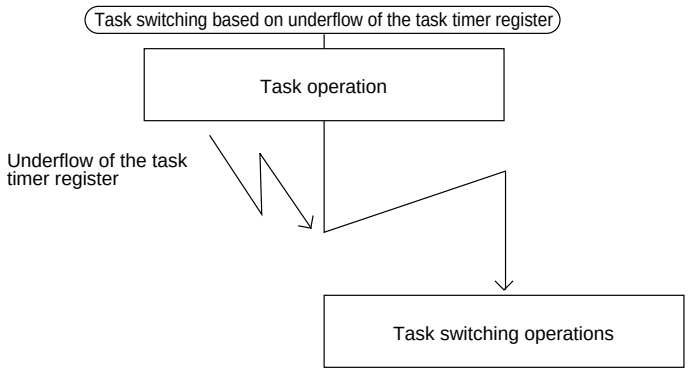


Figure 4-3-8 Task Switching Based on Underflow of Task Timer Register

■ Method of Immediate Task Switching by an Instruction

It is possible to switch the task without waiting for an underflow of register TSKBC by writing the number of the task to be started in the TSKNXT1 and TSKNXT0 flags of register TSKMD using an instruction.

Any underflow of the TSKBC register is ignored during the task switching sequence that has been started by an instruction. In addition, if the task switching request due to an instruction and an underflow of register TSKBC occur simultaneously, both of these are treated as the same task switching request and the task switching sequence is started only once.

In this case, the task that is started is the one that has been specified by the instruction.

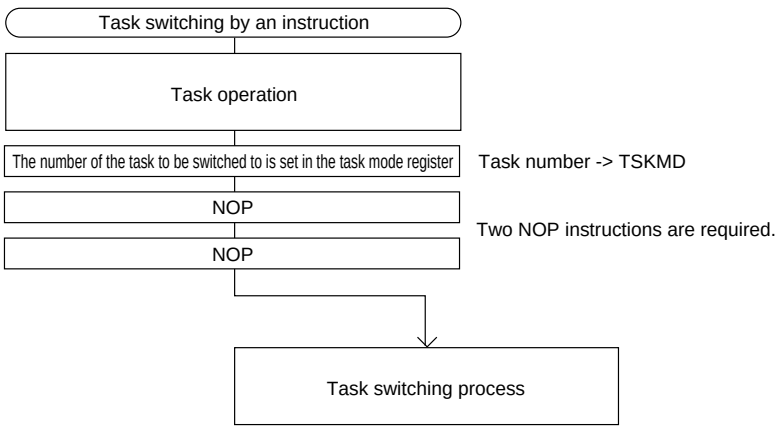


Figure 4-3-9 Task Switching by an Instruction

4-3-5 Method of Stopping a Task

There are the following two methods of stopping a currently executing task.

■ Method of Stopping a Task Based on Underflow of register TSKBC

To stop the current task, the user should disable (clear) the TSKnEN flag for the current task. Thereafter, the execution switches to the next task by the task switching sequence when there is an underflow in register TSKBC, which is carrying out the counting operation. The task for which the TSKnEN flag has been disabled remains stopped.

When the number of started task becomes 1 due to the task stopping process, register TSKBC stops counting.

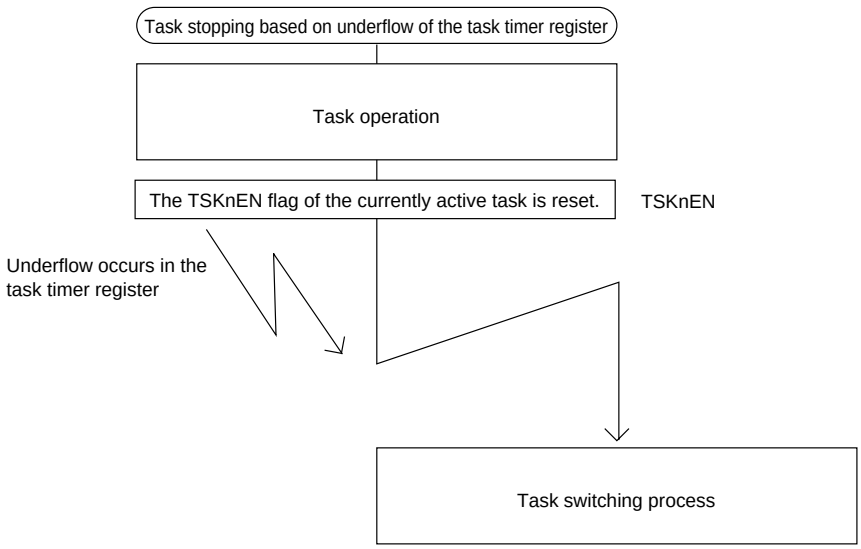


Figure 4-3-10 Task Stopping Based on Underflow of Task Timer Register

■ Method of Immediately Stopping a Task Using an Instruction

To stop the current task, the user should disable (clear) the TSKnEN flag for the current task. Thereafter, if the number of the task to be switched to is written in the TSKNXT1 and TSKNXT0 flags of register TSKMD, it is possible to immediately stop a currently active task by initiating the task switching sequence without waiting for an underflow of register TSKBC.

Any underflow of the TSKBC register is ignored during the task switching sequence that has been started by an instruction. In addition, if the task switching request due to an instruction and an underflow of register TSKBC occur simultaneously, both of these are treated as the same task switching request and the task switching sequence is started only once.

In this case, the task that is started is the one that has been specified by the instruction. When the number of started tasks becomes 1 due to the task stopping process, register TSKBC stops counting.

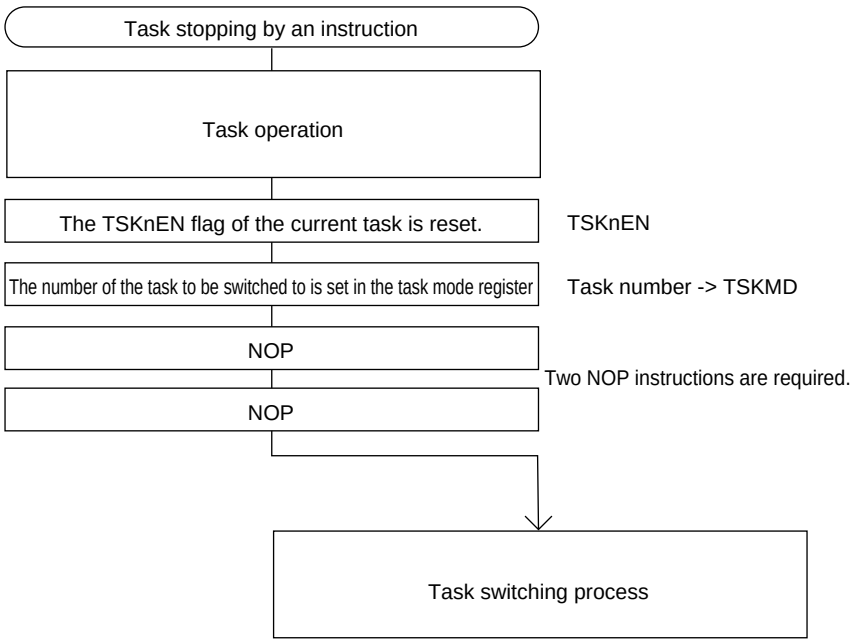


Figure 4-3-11 Task Stopping by an Instruction

4-3-6 Task Manipulating Instructions and Operation during Conflict with the Task Timer

When multiple tasks have been permitted to be started, the following operations are made when there is a conflict between the task manipulating instructions (writing data into register TSKMD for starting the third/fourth task, task switching, task stopping operation, etc.,) and an underflow from the task timer register.

■ Underflow of the Task Timer Register Immediately Before a Task Manipulation Instruction

The task switching sequence based on the underflow from the task timer is started immediately before the task manipulation instruction. Although the task manipulation instruction is executed immediately after the task switching sequence to the next applicable task, since this occurs during the save cycle (5 cycles) of the previous task, the task manipulation instruction is ignored (that is, the task switching sequence is not started). (However, the TSKnEn flag is set.)

■ When the Task Manipulation Instruction and the Task Timer Underflow Occur Simultaneously

These are treated as the same task switching request and the task switching sequence is started only once. The task to be started is the one specified by the instruction.

■ Task Timer Underflow During a Task Switching Sequence Due to a Task Switching Instruction

The underflow of the task timer is ignored.

When manipulating tasks using instructions while multiple tasks have been permitted to be started, execute the task manipulating instructions either while the task timer underflow does not occur or after a task timer count stopping operation.

```
check   BCLR (TSKCTR)TSKBCEN          ; The task timer is stopped.
        NOP
        TBNZ (TSKCTR)TSKBCEN,check    ; Checking that the task timer has stopped.
        MOV  Next_task_NO,(TSKMD)     ; Task manipulation
```

4-3-7 Interrupt Acceptance Operation during Task Switching

When an interrupt is accepted during multitask operation, the task execution is suspended temporarily and the interrupt service routine is started. The interrupt processing sequence in this case is the same as that during single task operation. When the execution of the interrupt service routine is completed, the execution returns to and restarts the task processing that was suspended earlier.

Register TSKBC stops counting during the execution of the interrupt service routine.

When the execution of the interrupt service routine is completed, register TSKBC starts counting again from the count value at the time counting was stopped.

Therefore, the interrupt processing is given priority when an interrupt is accepted and the normal task processing (task switching) is not made, and the normal task processing (task switching) is restarted after returning from the interrupt. Also, during the execution of the interrupt service routine, the stack pointer is changed from SP to SPI. The saving of PC and PSW upon accepting the interrupt, saving the PC during the execution of a JSR instruction within the interrupt service routine, and the execution of PUSH/POP instructions and SP indirect relative addressing instructions are all done for the RAM space addressed by SPI.

SPI is a stack pointer dedicated to interrupt processing. The saving of PC and PSW upon accepting an interrupt, saving of the PC during the execution of a JSR instruction within the interrupt service routine, and the execution of the PUSH/POP instructions are all done for the RAM space pointed to by SPI.

4-3-8 Task Timer Register (TSKBC) Operation

Register TSKBC is an 8-bit down counter that controls the execution time of each task. It is possible to select the counting clock between fs/16 and fs/32. When an underflow occurs in the task timer, the task switching sequence is started and the execution is switched to the next task. The order of task switching is the ascending order of the task numbers of the tasks for which the TSKnEN flag has been enabled.

Task 0 → Task 1
↑ ↓
Task 3 ← Task 2

The tasks for which the TSKnEN flag has not been enabled are skipped. For example, when only Task 1 and Task 3 have been enabled to start, the task is switched in the order -

Task 1 → Task 3 → Task 1 → Task 3

Counting stops when the number of tasks whose start has been enabled is 1 (including during a reset). When the number of tasks whose start has been enabled changes from several tasks to one task, the count value at that time is retained. Counting is restarted from that value when the number of tasks to be started again exceeds one.
When the number of tasks to be started changes from more than one to one to more than one again, counting is restarted from the retained count value, and the next task is started by initiating the task switching sequence when an underflow occurs. In this case, set the value again in register TSKBC depending on the need. however, such setting should be made before setting the TSKnEN flag of the second task.
Since the count value of register TSKBC is initialized within the task switching sequence, it is possible to set the count value for each task. This count value is loaded into register TSKBC when that task is switched to.

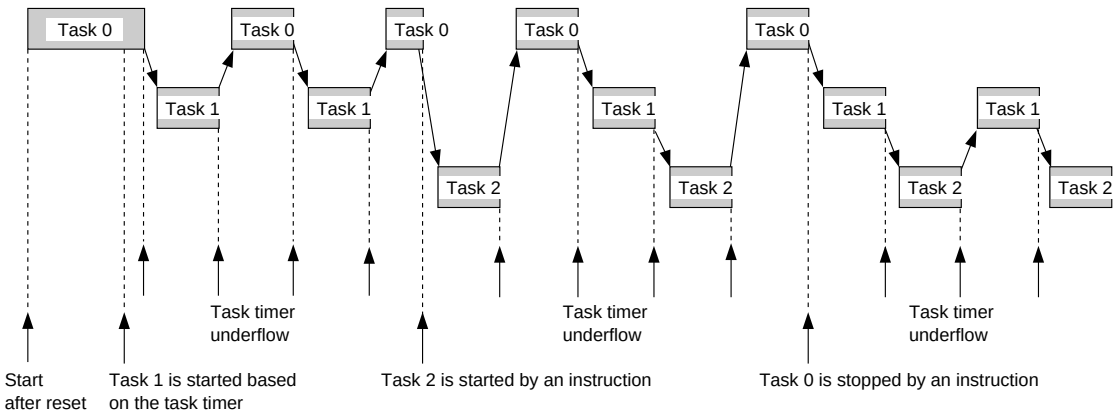


Figure 4-3-12 Task Switching Operation

Register TSKBC stops counting in the following situations:

(1) When the number of tasks to be started is one (including after reset).
(Counting is started when the start of several tasks is enabled.)
(2) During interrupt processing.
(Counting is restarted after returning from the interrupt.)
(3) When the TSKBCEN flag is cleared while the start of multiple tasks has been enabled.
(Counting is restarted when the TSKBCEN flag is set again.)

The count value at the time counting stops is retained. This count value does not change unless register TSKBC is overwritten.
The contents of register TSKBC are overwritten at task switching or by using an over-writing instruction.

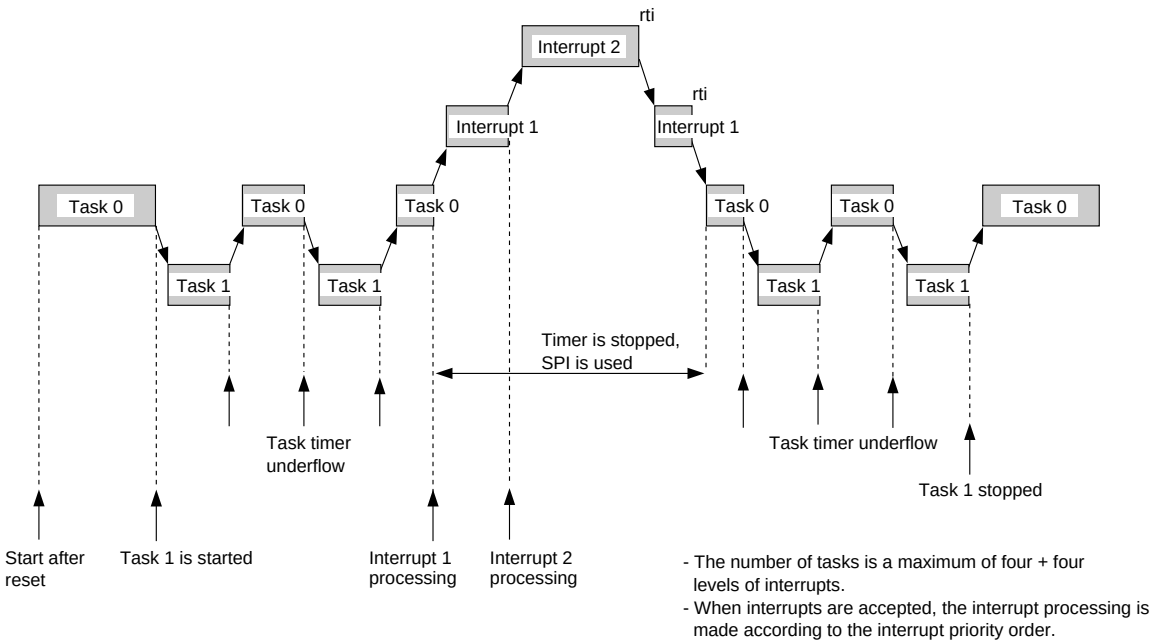


Figure 4-3-13 Task Switching Operation

4-3-9 Task Starting during Interrupt Processing

It is possible to start a new task or specify the task to be switched to while an interrupt is processed. Such operations become effective upon return from the interrupt.
In this case, upon return from the interrupt, the processing returns to the task that had been started upon acceptance of the interrupt, and then is immediately switched to the specified task.

When setting the number of the task to be started in register TSKMD during processing of an interrupt, there is no need for the immediately following NOP instructions.

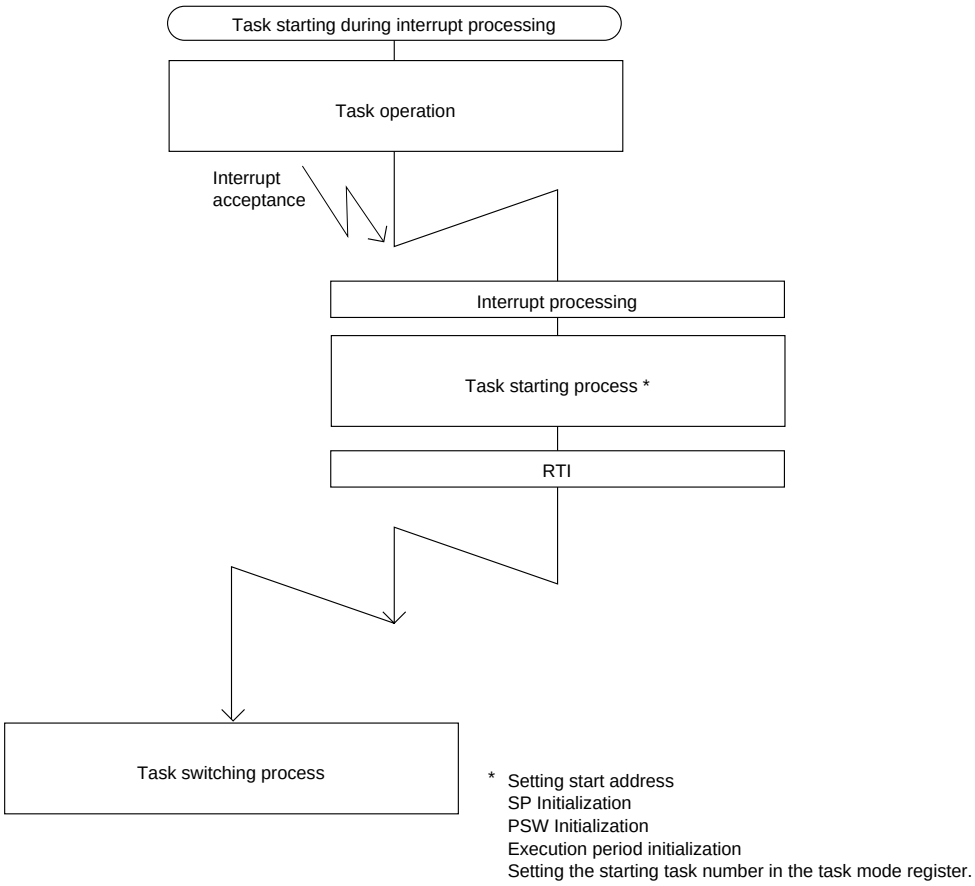


Figure 4-3-14 Task Starting during Interrupt Processing

Chapter 5 Timer Functions

5

5-1 Timer 0 Functions

Timer 0 is a 16-bit timer that can be preset and can realize timer and clock functions. Further, it is possible to make this timer operate as a timer counter with a maximum of 32 bits by connecting in cascade with Timer 6.

5-1-1 Timer 0 Settings

The following settings are common for all the Timer 0 functions: clock source selection, control to operate or halt counting, and interrupt source selection.

(1) The Timer 0 clock source is selected by the Timer 0 Clock Source Flags (TC0M0, TC0M1: bp0, bp1) of the Timer Mode Register 1 (TM1: x'3F5B').
(2) The Timer 0 counter operation is controlled by the Timer 0 Count Operate/Halt Flag (TC0E: bp4) of the Timer Mode Register 0 (TM0: x'3F5A').
(3) Timer 0 interrupt is generated by overflow of Binary Counter 0 (BC0: x'3F4C', x'3F4D').

When using ftc6 as the clock source, always set TB0 and TB6 in the condition in which timers 0 and 6 are stopped. In addition, it is necessary to start timer 0 after starting timer 6 before it overflows.

5-1-2 How to Use Timer 0

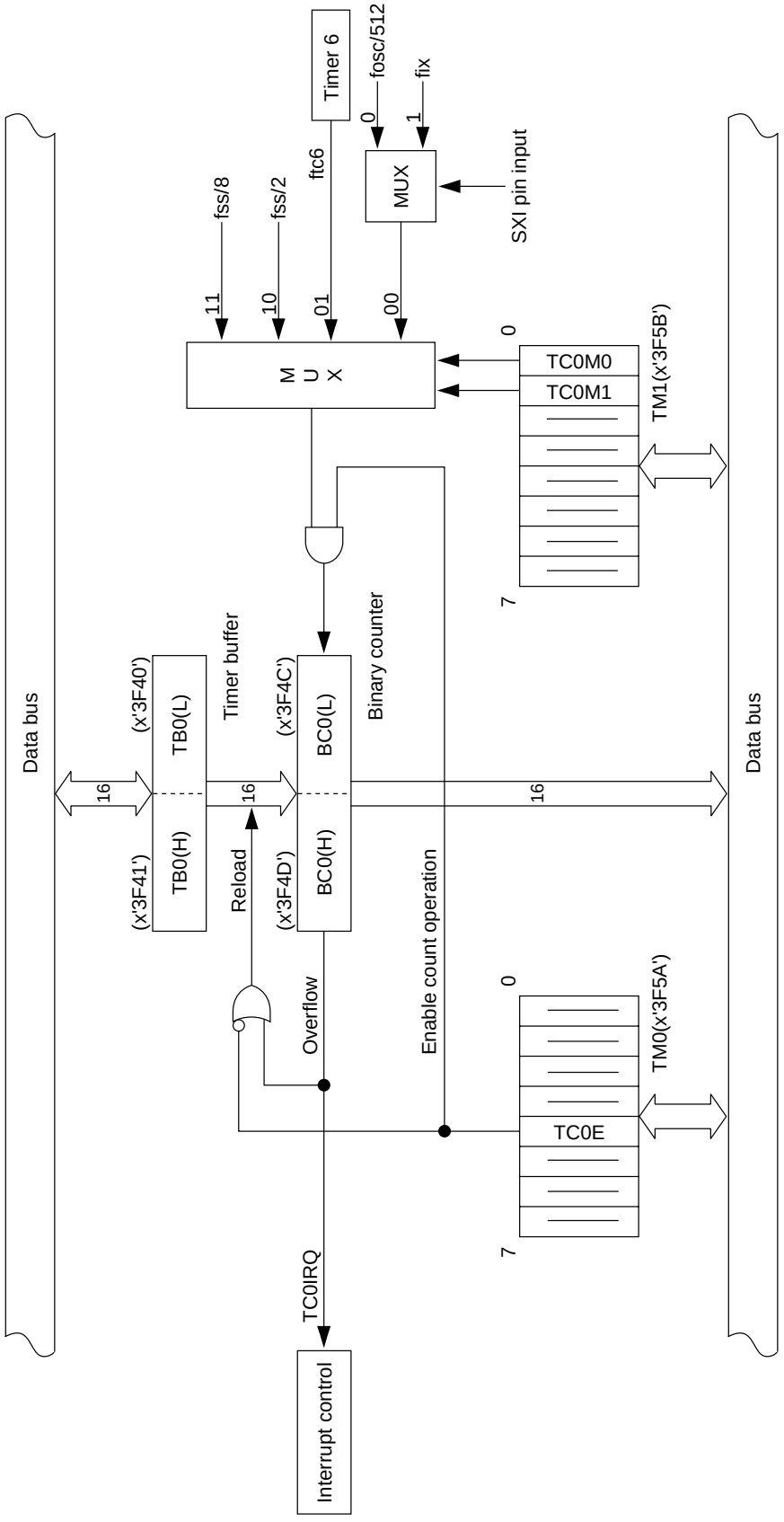
(1) Select the Timer 0 clock source.
(2) Write the preset data to Timer Buffer 0 (TB0: x'3F40', x'3F41').
(3) While the Timer 0 Count Operate/Halt Flag is '0', Binary Counter 0 (BC0) is loaded with the value in Timer Buffer 0. When the Timer 0 Count Operate/Halt Flag changes to '1', counting begins in the upward direction. At the same time as overflow of the Binary Counter 0 (BC0) Register is detected (when the count value changes from x'FFFF' to x'0000'), the value of Timer Buffer 0 (TB0) is reloaded into the Binary Counter 0 (BC0) Register.
(4) The Timer 0 Interrupt Request Flag is set as soon as overflow is detected.

When manipulating the interrupt request flag (xxxIR) by software, it is necessary to set beforehand the interrupt request IR enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, at the end of this operation, reset IRWE to '0'.

5-1-3 Timer 0 Interrupt Processing Settings

Perform the following procedure to enable Timer 0 interrupts.

(1) Specify the interrupt priority level with the TC0LV0 and TC0LV1 flags (bp6 and bp7) of the Timer 0 Interrupt Control Register (TC0ICR: x'3FEA').
(2) Clear the Timer 0 Interrupt Request Flag (TC0IR: bp0) of the Timer 0 Interrupt Control Register (TC0ICR).
(3) Set the Timer 0 Interrupt Enable Flag (TC0IE: bp1) of the Timer 0 Interrupt Control Register (TC0ICR).



Note: If fs/2 is used as the clock source, set the I/O wait to 'no-wait'.
Note: When using ftc6 as the clock source, always set TB0 and TB6 in the condition in which timers 0 and 6 are stopped. In addition, it is necessary to start timer 0 after starting timer 6 before it overflows.

Figure 5-1-1 Timer 0 Block Diagram

5-2 Timer 1 Functions

Timer 1 is a 16-bit timer that can be preset and can realize timer and event counter (linear timer counter) functions.

5-2-1 Timer 1 Settings

(1) The Timer 1 clock source is selected by the Timer 1 Clock Source Flags (TC1M0, TCM1M1: bp3, bp4) of the Timer Mode Register 1 (TM1: x'3F5B').
(2) The Timer 1 counter operation is controlled by the Timer 1 Count Operate/Halt Flag (TC1E: bp5) of the Timer Mode Register 0 (TM0: x'3F5A').
(3) Timer 1 interrupt is generated by overflow of Binary Counter 1 (BC1: x'3F4E', x'3F4F').

5-2-2 How to Use Timer 1

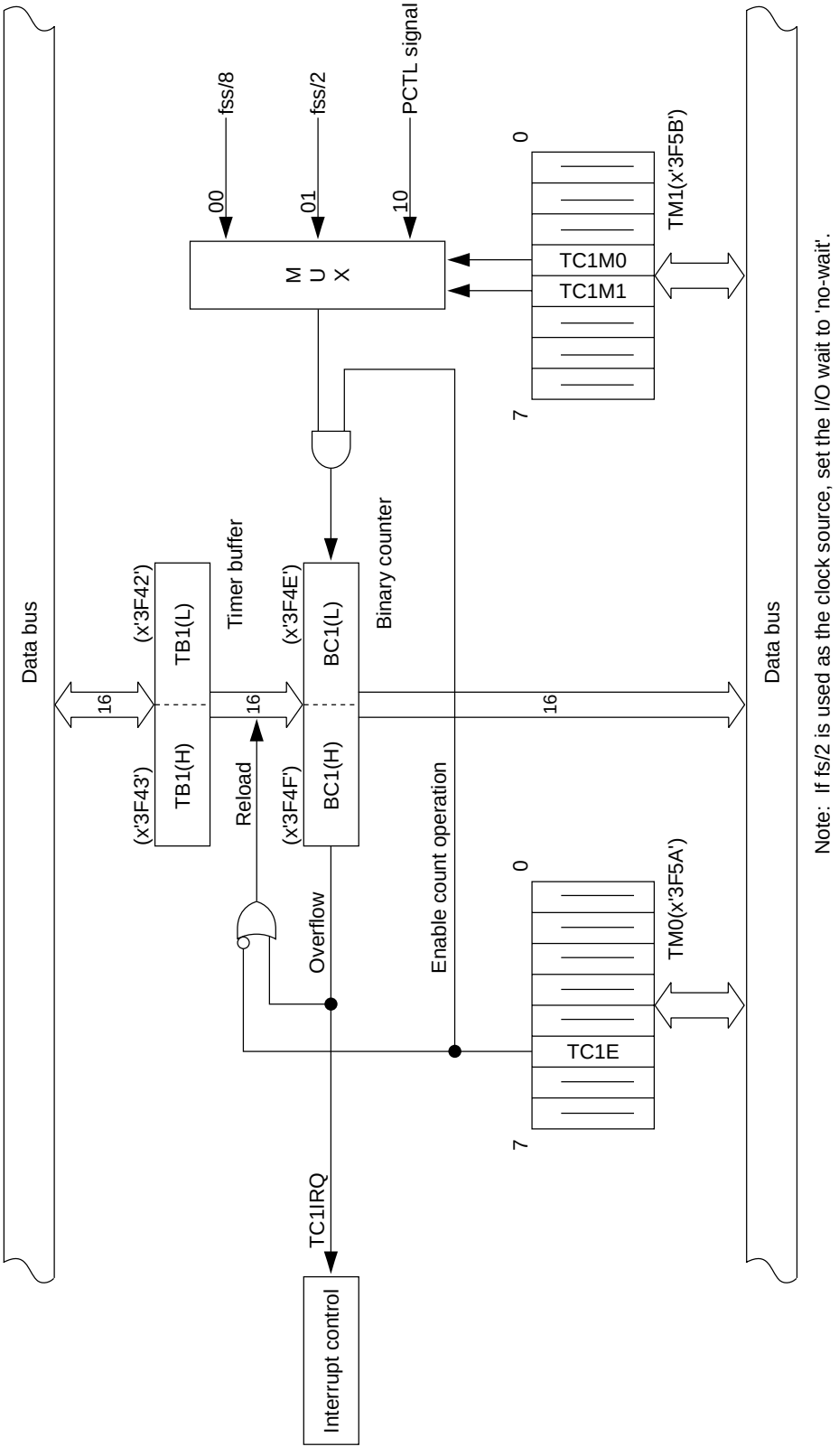
(1) Select the Timer 1 clock source.
(2) Write the preset data to Timer Buffer 1 (TB1: x'3F42', x'3F43').
(3) While the Timer 1 Count Operate/Halt Flag is '0', Binary Counter 1 (BC1: x'3F4E', x'3F4F') is loaded with the value in Timer Buffer 1. When the Timer 1 Count Operate/Halt Flag changes to '1', counting begins in the upward direction. At the same time as overflow of the Binary Counter 1 (BC1) Register is detected (when the count value changes from x'FFFF' to x'0000'), the value of Timer Buffer 1 (TB1) is reloaded into the Binary Counter 1 (BC1) Register.
(4) The Timer 1 Interrupt Request Flag is set as soon as overflow is detected.

When manipulating the interrupt request flag (xxxIR) by software, it is necessary to set beforehand the interrupt request IR enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, at the end of this operation, reset IRWE to '0'. It is necessary to set IRWE to '0'.

5-2-3 Timer 1 Interrupt Processing Settings

Perform the following procedure to enable Timer 1 interrupts.

(1) Specify the interrupt priority level with the TC1LV0 and TC1LV1 flags (bp6 and bp7) of the Timer 1 Interrupt Control Register (TC1ICR: x'3FEB').
(2) Clear the Timer 1 Interrupt Request Flag (TC1IR: bp0) of the Timer 1 Interrupt Control Register (TC1ICR).
(3) Set the Timer 1 Interrupt Enable Flag (TC1IE: bp1) of the Timer 1 Interrupt Control Register (TC1ICR).



Note: If fs/2 is used as the clock source, set the I/O wait to 'no-wait'.

Figure 5-2-1 Timer 1 Block Diagram

5-3 Timer 2 Functions

Timer 2 is a 16-bit timer that can be preset and can realize timer, input capture and duty judgment functions. The duty judgment function stores results of the judgment consecutively in a 16-bit shift register and has a function to compare this data with previously set data, as well as a function to count the number of DCTL signal pulses with a 4-bit counter that can be preset.

5-3-1 Timer 2 Settings

The following settings are common for all the Timer 2 functions: clock source selection, control to operate or halt counting, and interrupt source selection.

(1) The Timer 2 clock source is selected by the Timer 2 Clock Source Flags (TC2M0 to TC2M2: bp0, bp4, bp7) of the Timer Mode Register 2 (TM2: x'3F5C'). If the duty judgment function is to be used, specify fss/8 or fss/12.

(2) The Timer 2 counter operation is controlled by the Timer 2 Count Operate/Halt Flag (TC2E: bp6) of the Timer Mode Register 0 (TM0: x'3F5A').

(3) The Timer 2 Interrupt Source Flag of the Timer Mode Register 2 (TM2) specifies one of the following four Timer 2 interrupt sources.
- Overflow of Binary Counter 2 (BC2: x'3F50', x'3F51')
- When the specified edge of the DCTL signal is input
- When the 4-bit counter of the Timer 2 Shift Register (SRBC2: x'3F64') underflows
- When the value of Timer 2 Shift Register (SR2F: x'3F5F' to x'3F60') matches the value of Timer 2 Shift Register Compare Register (CMPR2: x'3F61' to x'3F62')

5-3-2 How to Use Timer 2 (Timer Function)

(1) Select the Timer 2 clock source.

(2) Specify BC2 overflow as the Timer 2 interrupt source.

(3) Write the preset data to Timer Buffer 2 (TB2: x'3F44', x'3F45'h).

(4) While the Timer 2 Count Operate/Halt Flag is '0', Binary Counter 2 (BC2) is loaded with the value in Timer Buffer 2. When the Timer 2 Count Operate/Halt Flag changes to '1', counting begins in the upward direction. At the same time as overflow of the Binary Counter 2 (BC2) Register is detected (when the count value changes from x'FFFF' to x'0000'), the value of Timer Buffer 2 (TB2) is reloaded into the Binary Counter 2 (BC2) Register.

(5) The Timer 2 Interrupt Request Flag is set as soon as overflow is detected.

5-3-3 Timer 2 Interrupt Processing Settings

When manipulating the interrupt request flag (xxxIR) by software, it is necessary to set beforehand the interrupt request IR enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, at the end of this operation, reset IRWE to '0'.

Perform the following procedure to enable Timer 2 interrupts.

(1) Specify the interrupt priority level with the TC2LV0 and TC2LV1 flags (bp6 and bp7) of the Timer 2 Interrupt Control Register (TC2ICR: x'3FEC').
(2) Clear the Timer 2 Interrupt Request Flag (TC2IR: bp0) of the Timer 2 Interrupt Control Register (TC2ICR).
(3) Set the Timer 2 Interrupt Enable Flag (TC2IE: bp1) of the Timer 2 Interrupt Control Register (TC2ICR).

5-3-4 How to Use Timer 2 (Input Capture Function)

When the DCTL signal changes polarity, the value of Binary Counter 2 (BC2) can be captured and input to the Timer 2 Input Capture Register (ICR2). Measurements such as pulse width and pulse cycle are possible with this function.

(1) To use the input capture function, enable the Input Capture Operation Enable/ Disable Flag (ICR2EG: bp2) of the Timer Mode Register 2 (TM2) and specify the valid edge for capture with the Timer 2 Valid Edge Flag (TC2EG: bp1). Set the Interrupt Source Flags (TC2SEL0, TC2SEL1: bp5, bp6) to generate an interrupt when the specified valid edge of the DCTL signal is input.
(2) To measure the pulse width, enable the BC2CL Flag (bp3) of the Timer Mode Register 2 (TM2) so that the BC2 Register is initialized at either edge of the DCTL signal and set the initial value of the TB2 register to x'0000'. Specify measurement of high-level or low-level widths with the Timer 2 Valid Edge Flag of Timer Mode Register 2 (TM2).
(3) When the specified valid edge of the DCTL signal is input, the value at that moment of the BC2 counter (that is counting up) is captured and input to the ICR2 Register.
(4) If the Interrupt Source Flags (TC2SEL0, TC2SEL1: bp5, bp6) are set in advance to the specified valid edge of the DCTL signal input, an interrupt will be generated at the same time as the BC2 value is captured by the ICR2 Register.
(5) Pulse width measurement is possible by reading the ICR2 Register after an interrupt is accepted.

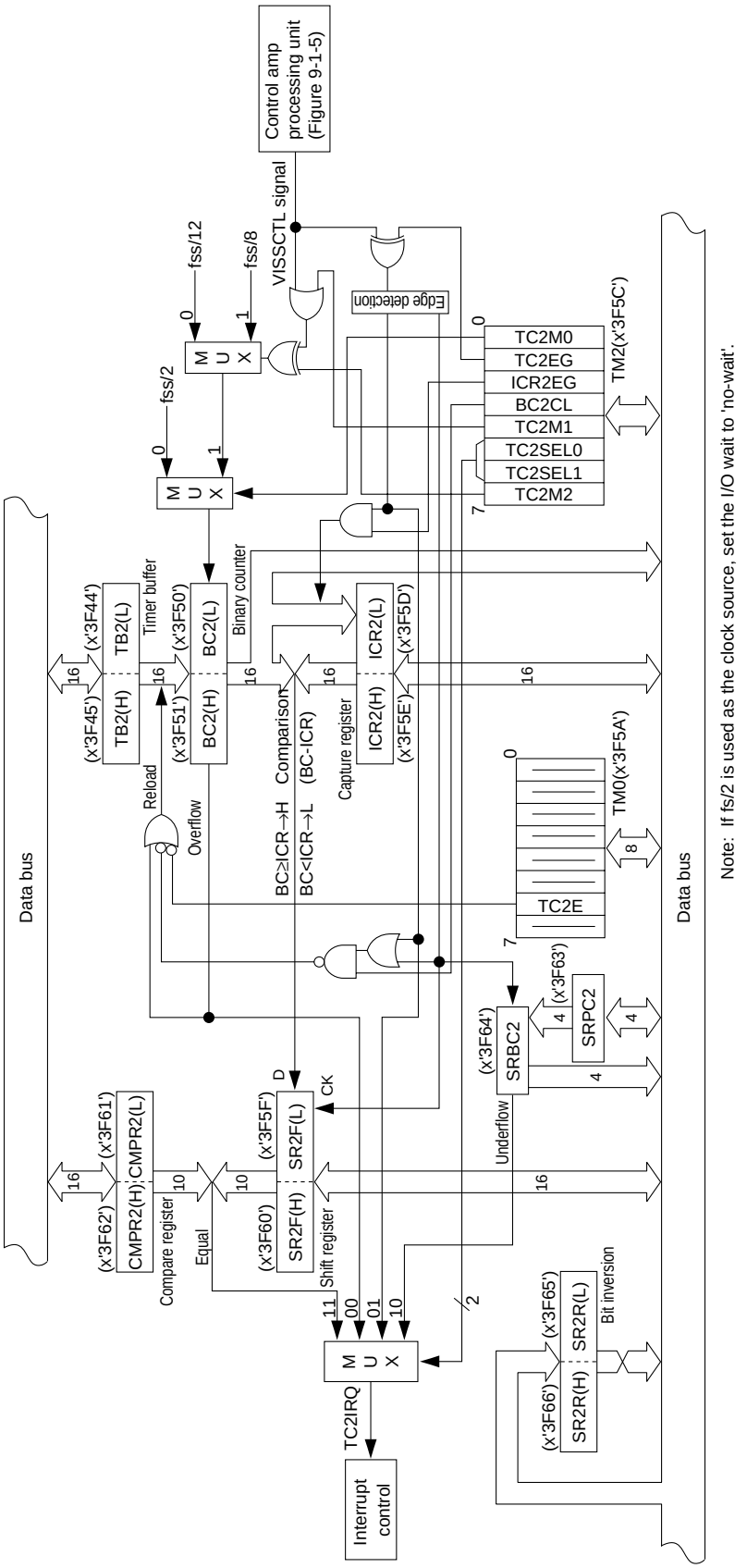
5-3-5 How to Use Timer 2 (Duty Judgment Function)

(1) The initial value of Binary Counter 2 (BC2: x'3F50', x'3F51') is usually set by writing x'0000' to Timer Buffer 2 (TB2: x'3F44', x'3F45').
(2) When determining the duty of consecutive Timer 2 pulses with a threshold value of 40%, the Clock Source Flag of Timer Mode Register 2 (TM2: x'3F5C') specifies whether the signal width for duty evaluation is a high-level or low-level width.

Duty judgment results are sequentially stored in the Timer 2 Shift Register. This register and the Compare Register for Timer 2 Shift Register (CMPR2: x'3F61', x'3F62') form a comparator. The upper 5 bits and the lowest 1 bit of CMPR2 are excluded so that only 10 bits of data are compared. The Timer 2 Interrupt Source Flags (TC2SEL0, TC2SEL1: bp5, bp6) of Timer Mode Register 2 (TM2: x'3F5C') can be set to generate an interrupt when the comparison results match.

The SRBC2 Register contains a function to count the number of DCTL signal pulses. Using the SRBC2 Register, judgment data for an arbitrary number of pulses (up to 16) can be read from the SR2F Register.

(1) Preset data is set by writing to the SRPC2 Register. The SRBC2 Register will be initialized at that time, even if counting is in progress.
(2) When the edge specified by the Timer 2 Valid Edge Flag is input, the SRBC2 Register begins counting downward. Counting by the SRBC2 Register and shifting by the SR2F Register are performed simultaneously.
(3) At the same time as underflow is detected (when the value of SRBC2 changes from x'0' to x'f'), the value in the SRPC2 Register is reloaded. The Timer 2 Interrupt Source Flags (TC2SEL0, TC2SEL1: bp5, bp6) can be set to generate an interrupt when underflow is detected.



Note: If fs/2 is used as the clock source, set the I/O wait to 'no-wait'.

Figure 5-3-1 Timer 2 Block Diagram

5-4 Timer 3 Functions

Timer 3 is a 16-bit timer that can be preset and can realize timer and serial transfer header functions.

5-4-1 Timer 3 Settings

(1) The Timer 3 clock source is selected by the Timer 3 Clock Source Flag (TC3M: bp5) of the Timer Mode Register 1 (TM1: x'3F5B').
(2) The Timer 3 counter operation is controlled by the Timer 3 Count Operate/Halt Flag (TC3E: bp7) of the Timer Mode Register 0 (TM0: x'3F5A').
(3) Timer 3 interrupt is generated by overflow of Binary Counter 3 (BC3: x'3F52', x'3F53').

5-4-2 How to Use Timer 3 (Timer Function)

(1) Select the Timer 3 clock source.
(2) Write the preset data to Timer Buffer 3 (TB3: x'3F46', x'3F47').
(3) While the Timer 3 Count Operate/Halt Flag is '0', Binary Counter 3 (BC3) is loaded with the value in Timer Buffer 3. When the Timer 3 Count Operate/Halt Flag changes to '1', counting begins in the upward direction. At the same time as overflow of the Binary Counter 3 (BC3) Register is detected (when the count value changes from x'FFFF' to x'0000'), the value of Timer Buffer 3 (TB3) is reloaded into the Binary Counter 3 (BC3) Register.
(4) The Timer 3 Interrupt Request Flag is set as soon as overflow is detected.

5-4-3 How to Use Timer 3 (Serial Transfer Header Function)

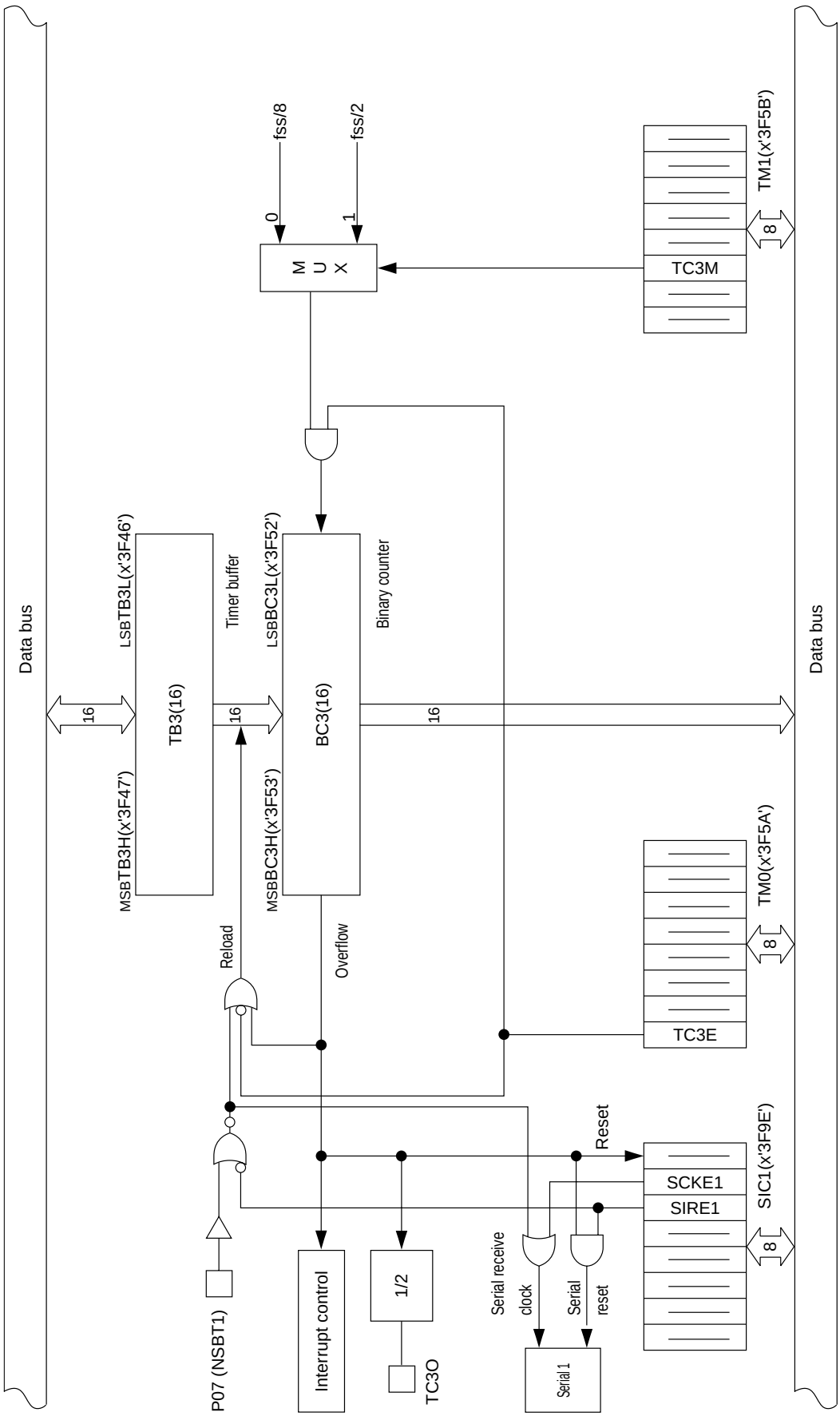
(1) Set the SIF1 Header Function Flag (SIRE1: bp2) of the Serial Control Register (SIC1: x'3F9E') to "Header".
(2) Set the preset data in Timer Buffer 3 (TB3).
(3) Binary Counter 3 operates as a 16-bit up-counter. Set the Timer 3 Count Operate/Halt Flag to '1' to begin operation of Timer 3. When a high-level signal is input to the serial clock pin, the BC3 Counter begins to count upward. When a low-level signal is input, the value of TB3 is loaded into BC3. If fixed-interval low-level signals are not input, the BC3 Register will overflow, causing the 3-bit counter of Serial Interface 1 and the Serial 1 Clock Disable Flag (SIC1: x'3F9E') to be cleared.

5-4-4 Timer 3 Interrupt Processing Settings

Perform the following procedure to enable Timer 3 interrupts.

(1) Specify the interrupt priority level with the TC3LV0 and TC3LV1 flags (bp6 and bp7) of the Timer 3 Interrupt Control Register (TC3ICR: x'3FED').
(2) Clear the Timer 3 Interrupt Request Flag (TC3IR: bp0) of the Timer 3 Interrupt Control Register (TC3ICR).
(3) Set the Timer 3 Interrupt Enable Flag (TC3IE: bp1) of the Timer 3 Interrupt Control Register (TC3ICR).

When manipulating the interrupt request flag (xxxIR) by software, it is necessary to set beforehand the interrupt request IR enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, at the end of this operation, reset IRWE to '0'.



Note: If fs/2 is used as the clock source, set the I/O wait to "no-wait".

Figure 5-4-1 Timer 3 Block Diagram

5-5 Timer 4 Functions

Timer 4 is a 16-bit timer that can be preset and can function as a timer, an event counter, and can generate the clock for synchronous serial 0 and synchronous serial 1 interfaces. A remote control function can be realized through combination with the Serial 1 Interface.

5-5-1 Timer 4 Settings

(1) The Timer 4 clock source is selected by the Timer 4 Clock Source Flag (TC4M: bp7) of the Timer Mode Register 1 (TM1: x'3F5B').
(2) The Timer 4 counter operation is controlled by the Timer 4 Count Operate/Halt Flag (TC4E: bp6) of the Timer Mode Register 1.
(3) Timer 4 interrupt is generated by overflow of Binary Counter 4 (BC4: x'3F54', x'3F55').

5-5-2 How to Use Timer 4 (Timer Function)

(1) Select the Timer 4 clock source.
(2) Write the preset data to Timer Buffer 4 (TB4: x'3F48', x'3F49').
(3) While the Timer 4 Count Operate/Halt Flag is '0', Binary Counter 4 (BC4) is loaded with the value in Timer Buffer 4. When the Timer 4 Count Operate/Halt Flag changes to '1', counting begins in the upward direction. At the same time as overflow of the Binary Counter 4 (BC4) Register is detected (when the count value changes from x'FFFF' to x'0000'), the value of Timer Buffer 4 (TB4) is reloaded into the Binary Counter 4 (BC4) Register.
(4) The Timer 4 Interrupt Request Flag is set as soon as overflow is detected.

5-5-3 Serial Transfer Clock Generation Function

Timer 4 can be used to generate the transfer clock for the synchronous serial interface. To use this function, it is necessary to perform the same settings as for the timer function. The synchronous serial clock is 1/2 the frequency of overflow of Binary Counter 4 (BC4: x'3F54', x'3F55').

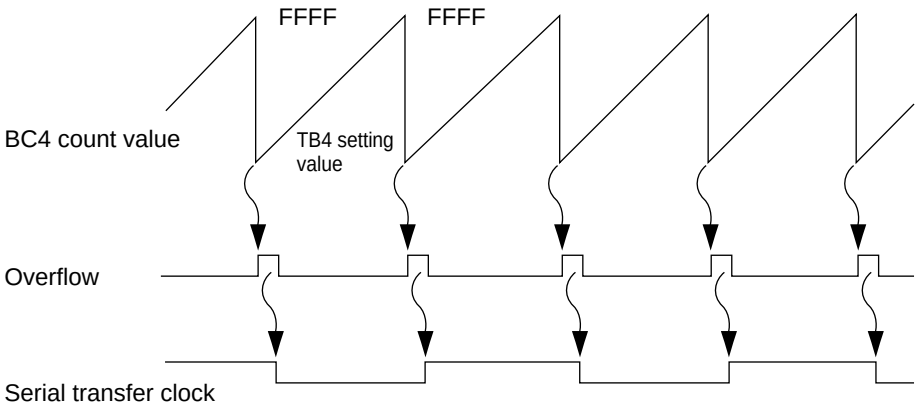


Figure 5-5-1 Synchronous Serial Clock Generation by Timer 4

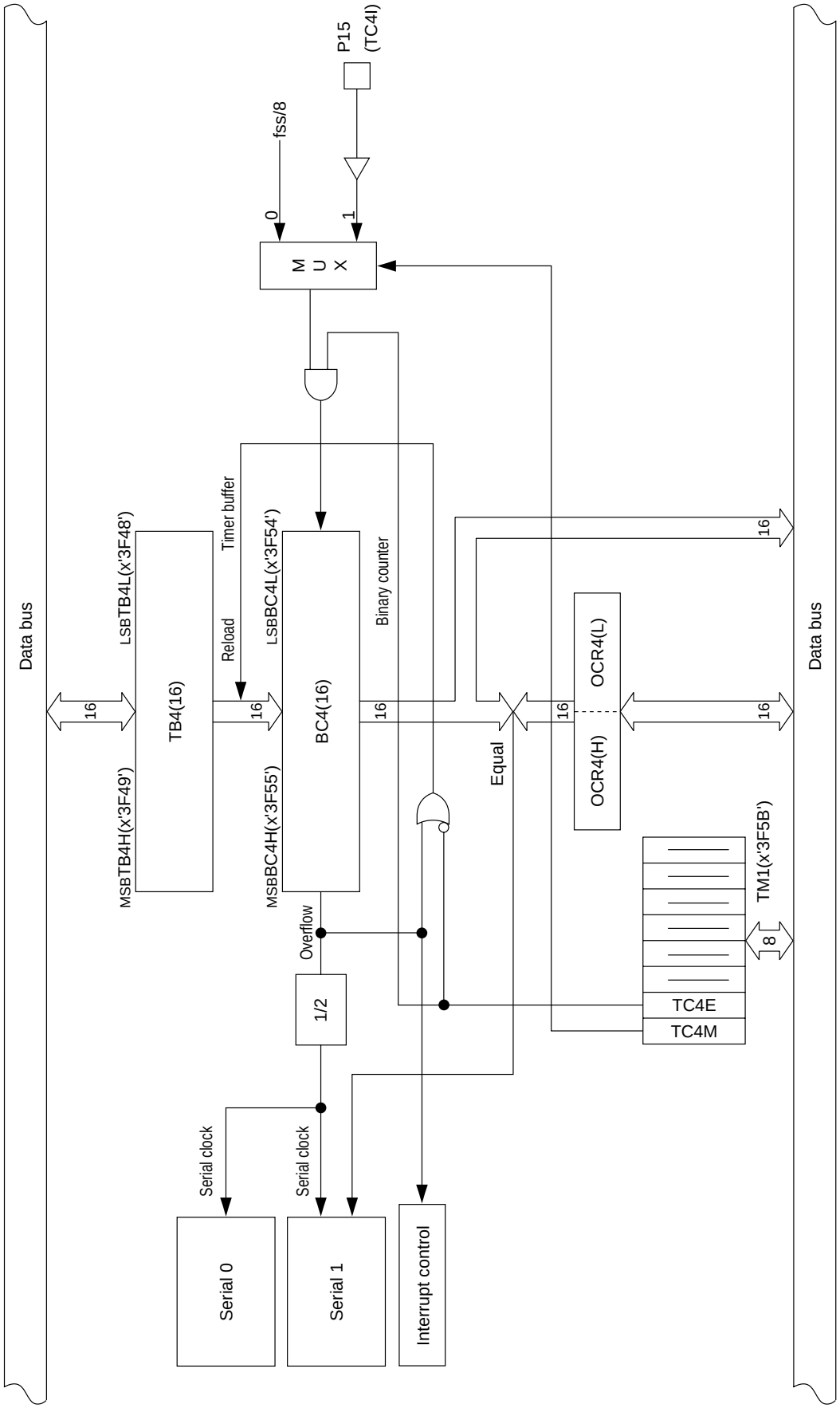
5-5-4 Timer 4 Interrupt Processing Settings

Perform the following procedure to enable Timer 4 interrupts.

(1) Specify the interrupt priority level with the TC4LV0 and TC4LV1 flags (bp6 and bp7) of the Timer 4 Interrupt Control Register (TC4ICR: x'3FEE').
(2) Clear the Timer 4 Interrupt Request Flag (TC4IR: bp0) of the Timer 4 Interrupt Control Register (TC4ICR).
(3) Set the Timer 4 Interrupt Enable Flag (TC4IE: bp1) of the Timer 4 Interrupt Control Register (TC4ICR).

Refer to section 6-2, "Serial 1" for the remote control function.

When manipulating the interrupt request flag (xxxIR) by software, it is necessary to set beforehand the interrupt request IR enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, at the end of this operation, reset IRWE to '0'.



Note: If fs/2 is used as the clock source, set the I/O wait to "no-wait".

Figure 5-5-2 Timer 4 Block Diagram

5-6 Timer 5 Operation

5-6-1 Timer 5 Functions and Common Settings

Timer 5 is a 19-bit binary counter that can realize watchdog timer (time base function) and oscillation stabilization wait timer functions.

5-6-2 Setting and Operation of the Watchdog Timer Function

When activating the watchdog timer, always make sure to clear beforehand the watchdog timer (WDCLR flag ='1'). In particular, care should be taken immediately after recovering from the HALT mode.

(1) First select the watchdog interval with the WDS Flag of the Watchdog Timer Control Register (WDCNT). Next, set the WDCLR Flag to '1' and clear the watchdog timer. Then, set the WDEN Flag to '1' to start operation of the watchdog timer.

(2-1) If the watchdog interval is set to Twd=2^{16}/fs (WDS Flag = 0), WDCLR Flag to '1' within the fixed amount of time (Twd=2^{16}/fs) and the watchdog timer will begin counting from x'0'.
If the watchdog interval is set to Twd=2^{19}/fs (WDS Flag = 1), set the WDCLR Flag to '1' within the fixed amount of time (Twd=2^{19}/fs) and the watchdog timer will begin counting from x'0'.

If the WDCLR Flag is not set, a WD interrupt will be generated after a fixed amount of time passes. Also, the Watchdog Interrupt Request Flag (WDIR: bp1) of the Watchdog Interrupt Control Register (NMICR: x'3FE1') will be set, and interrupts will be processed unconditionally.

Clear the WDEN Flag to '0' to stop operation of the watchdog timer. (The counter operates but does not generate WD interrupts. Setting the WDCLR Flag to '1' clears the counter.)

The watchdog interval is different for each CPU mode.

[☞ 10-1-4 Transfer between SLOW and NORMAL Modes]

The count status of the upper 8 bits in the BC5 Register can be monitored by reading the BC5 Register (x'3F56'). When the watchdog interval is Twd=2^{16}/fs (WDS Flag = 0), the lower 5 bits of the value read correspond to the upper 5 bits of the counter. (The upper 3 bits are valid when the watchdog interval of Twd=2^{19}/fs (WDS Flag =1) is selected.)

5-6-3 Oscillation Stabilization Wait Function

Binary Counter 5 (BC5: x'3F56', R) operates as an oscillation stabilization wait timer when reset is activated and when operation returns from the STOP mode.

(1) When the reset signal or a return-from-STOP-mode signal is detected, all bits of the BC5 counter are reset, counting begins upward from x'0', and each mode waits for oscillation stabilization as indicated below. The oscillation stabilization wait time (Tw) when reset is activated is:

$Tw=2^{18}/fosc$ (18.3ms@fosc=14.32MHz)

regardless of the level at the SXI pin.

Before changing over to the STOP0 mode from the NORMAL mode of fs=fosc/2, always make sure to set x'01' in the register DLYCNT.

[☞ See Figs 10-1-3 and 10-1-5.]

The oscillation stabilization wait time (Tw) when returning from the STOP 0 Mode is:

$Tw=2^{18}/fosc$ (18.3ms@fosc=14.32MHz)

Before transferring to the STOP 1 Mode, the DLYCNT Register must be set to x'02'. When Xi is used and the SXI pin = HIGH, the oscillation stabilization wait time (Tw) upon returning from the STOP 1 Mode is:

[☞ See Fig. 10-1-5.]

$Tw=2^{15}/fxi$ (1.0sec@fxi=32.768kHz)

Refer to Chapter 10, "Standby, Clock Operation/Halt, and Reset Functions", for XI/XO oscillation stabilization instructions.

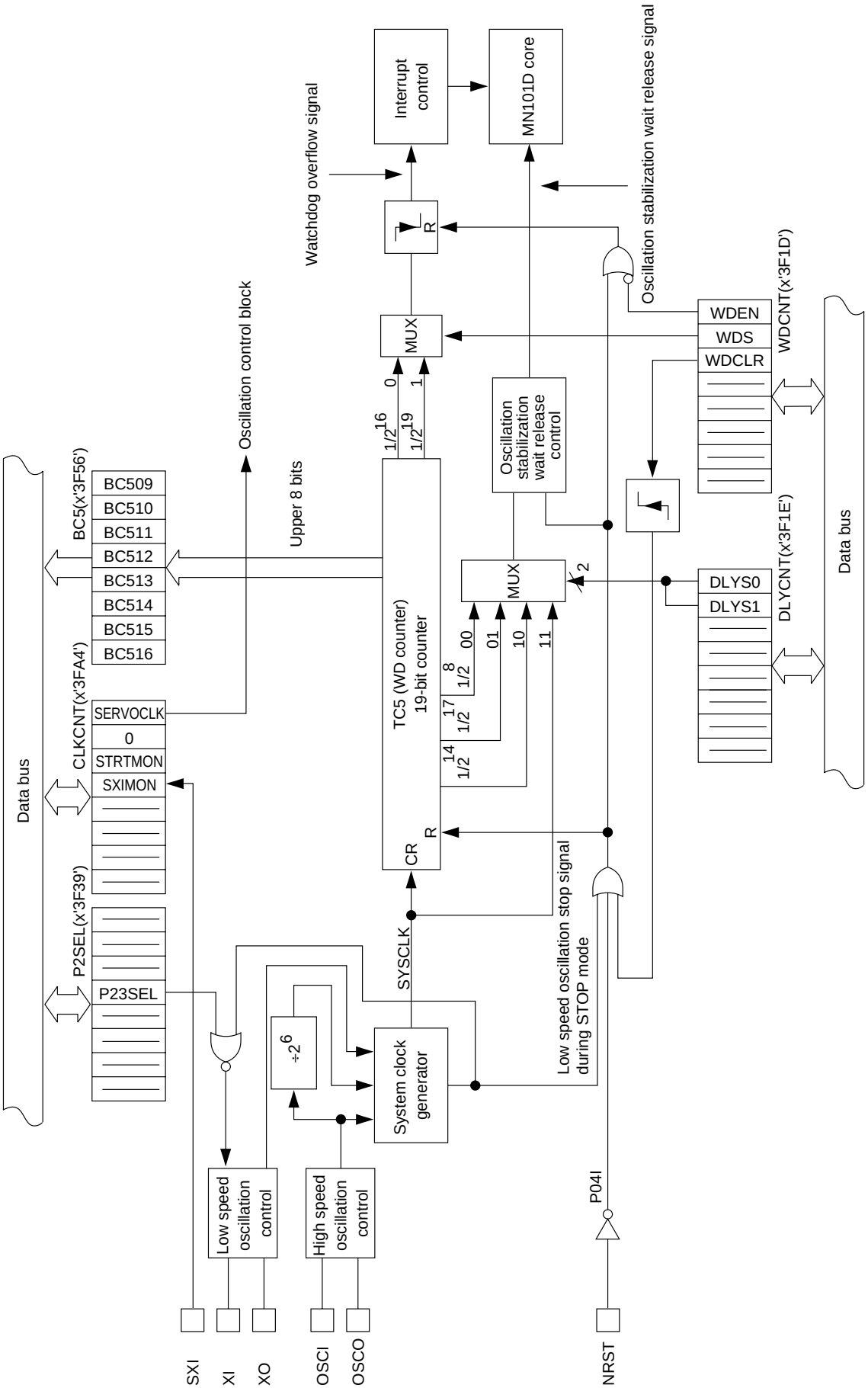


Figure 5-6-1 Timer 5 (Watchdog, Oscillation Stabilization Wait, Oscillation Control) Block Diagram

5-7 Timer 6 Functions

Timer 6 is a 16-bit timer that can be preset and can realize clock functions.

5-7-1 Timer 6 Settings

(1) The Timer 6 clock source is selected by the Timer 6 Clock Source Flags (TC6CK0, TC6CK1: bp5, bp6) of the Timer 6 Control Register (TC6CNT: x'3F6B').

(2) The Timer 6 counter operation is controlled by the Timer 6 Count Operate/Halt Flag (TC6E: bp0).

(3) There are four sources of interrupts for Timer 6. The Timer 6 External Pin Interrupt Flags 0 and 1 (TC6IRQ0, TC6IRQ1: bp1, bp2) of the Timer 6 Control Register specify the interrupt source. Interrupt sources for Timer 6 are listed below.

- Clock source frequency divided by 2^{13}
- Clock source frequency divided by 2^{14}
- Clock source frequency divided by 2^{15}
- Clock source frequency divided by 2^{16} (overflow)

(4) So that counting can be performed when power is turned on, the circuitry of Timer 6 is not reset. These circuits can be specified for reset by the TC6RST Flag of the Timer 6 Control Register (TC6CNT).

If ftc6 is set as the clock source in Timer 0, it is possible to operate it as a timer/counter with a maximum of 32 bits.

5-7-2 Timer 6 Operation

(1) The clock function operates in modes other than STOP0 or STOP1.

(2) Interrupts are requested when the specified interrupt source overflows.

5-7-3 Timer 6 Interrupt Processing Settings

Perform the following procedure to enable Timer 6 interrupts.

(1) Specify the interrupt priority level with the TC6LV0 and TC6LV1 flags (bp6 and bp7) of the Timer 6 Interrupt Control Register (TC6ICR: x'3FEF').

(2) Clear the Timer 6 Interrupt Request Flag (TC6IR: bp0) of the Timer 6 Interrupt Control Register (TC6ICR).

(3) Set the Timer 6 Interrupt Enable Flag (TC6IE: bp1) of the Timer 6 Interrupt Control Register (TC6ICR).

When manipulating the interrupt request flag (xxxIR) by software, it is necessary to set beforehand the interrupt request IR enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, at the end of this operation, reset IRWE to '0'.

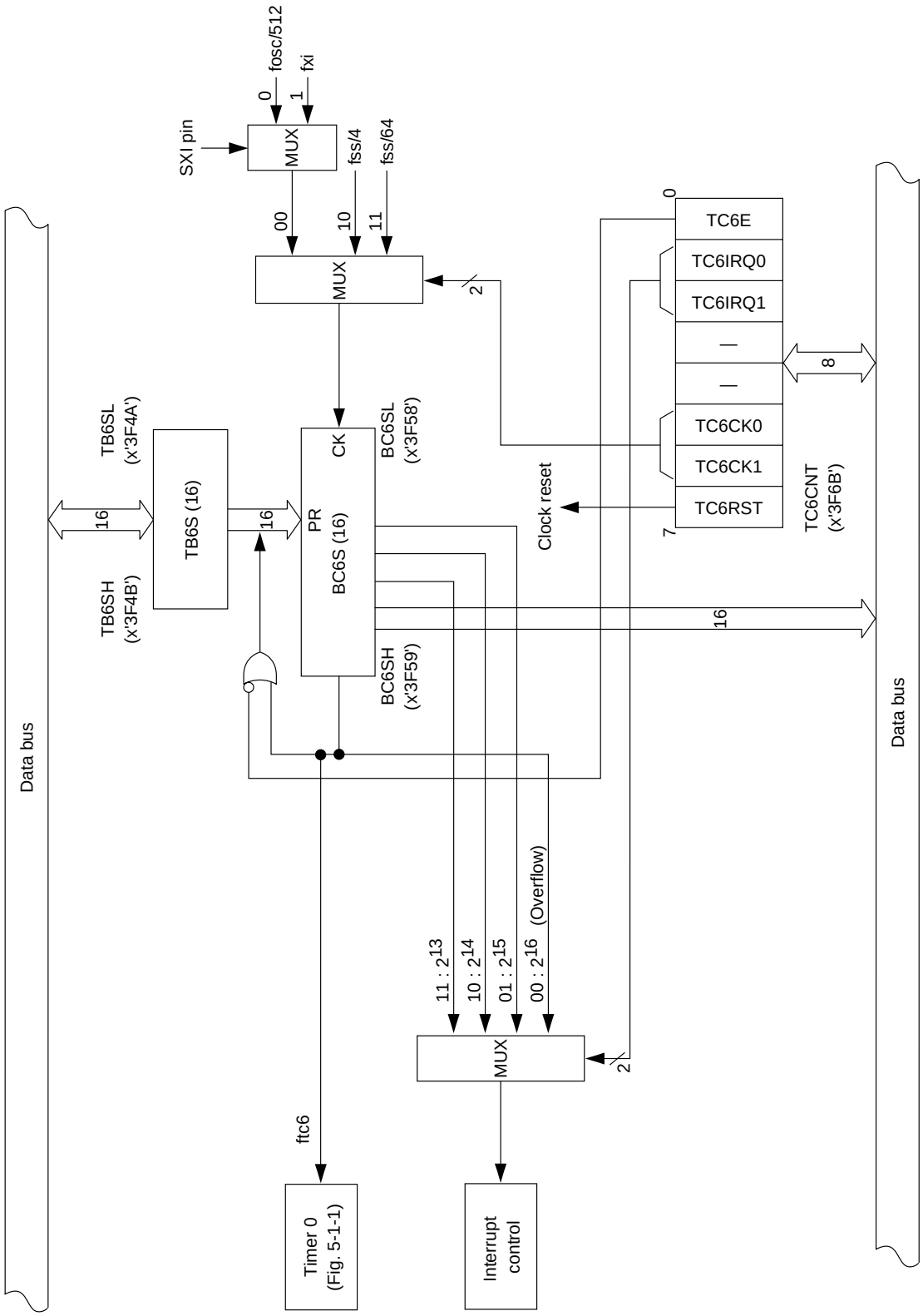


Figure 5-7-1 Timer 6 (Clock) Block Diagram

5-8 Buzzer Output Function

A buzzer signal can be output from Port 46.

5-8-1 Buzzer Output Setting

(1) Set the buzzer output frequency in the Frequency Setting Flags (BUZS0, BUZS1: bp2, bp1) of the Buzzer Control Register (BUZCNT: x'3F3F').
(2) Set Port 46 as a general-purpose port.
(3) When buzzer output is to be performed, set the Buzzer Output Enable Flag (BUZOE: bp0) of the Buzzer Control Register (BUZCNT: x'3F3F').
Clear the Buzzer Output Enable Flag (BUZOE: bp0) to stop the buzzer output.

The four buzzer frequencies are listed below.

Table 5-8-1

BUZCNT Register		Buzzer Frequency	
BUZS1	BUZS0	Normal Mode (servoclk=0)	Normal Mode (servoclk=1)
0	0	6992Hz	3496Hz
0	1	3496Hz	1748Hz
1	1	1748Hz	874Hz
1	0	874Hz	437Hz

The buzzer frequencies in the table to the left are given for the condition of fosc=14.32 MHz.

Buzzer output is not compatible with modes other than NORMAL mode.

Since the buzzer output utilizes the output of a free-running counter, the initial buzzer frequency immediately after output is enabled will differ from the set value.

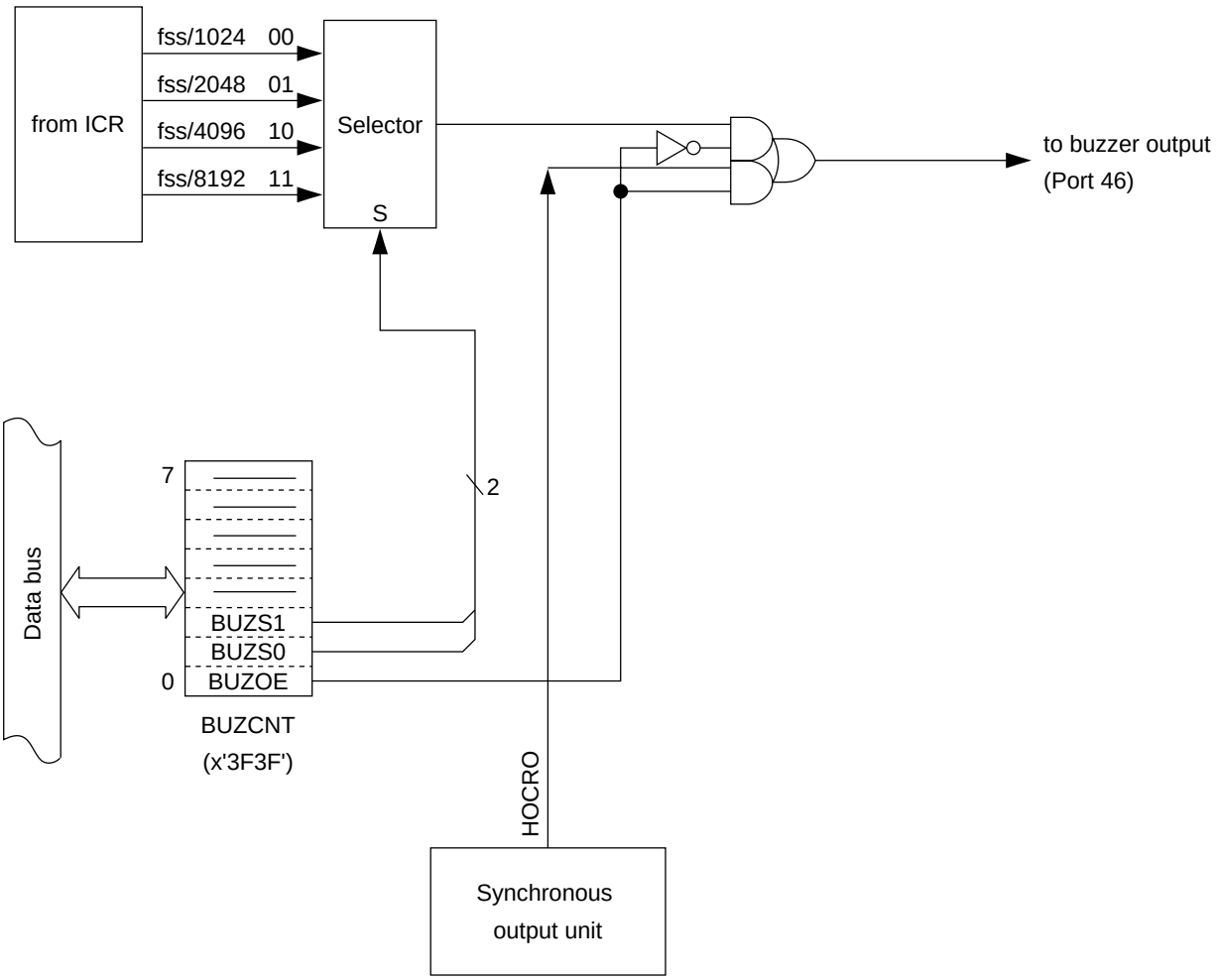


Figure 5-8-1 Buzzer Output Block Diagram

Table 5-8-2 Timer Clock Source Summary (1/2)

		SXI Pin				Comments
		0		1		
		Oscillation Control Register CLKCNT(x'3FA4')		Oscillation Control Register CLKCNT(x'3FA4')		
		SERVOCLK (bp0)		SERVOCLK(bp0)		
Timer Name	Setting for Selected Clock Source	0	1	0	1	
Timer 0	TC0M 00	fosc/512		fxi		TC0M0 and TC0M1 are bits 0 and 1 of TM1 (x'3F5B')
	01	ftc6		Same as values at left		
	10	fs/2	fs/4			
	11	fs/8	fs/16			
Timer 1	TC1M 00	fs/8	fs/16	Same as values at left		TC1M0 and TC1M1 are bits 3 and 4 of TM1 (x'3F5B')
	01	fs/2	fs/4			
	10	PCTL signal				
Timer 2	TC2M 000	fs/2	fs/4	Same as values at left		TC2M0, TC2M1 and TC2M2 are bits 0, 4 and 7 of TM2 (x'3F5C')
	011	fs/12	fs/24			
	111	fs/8	fs/16			
		VISSCTL=0				
	001	fs/12	fs/24			
	101	fs/8	fs/16			
		VISSCTL=1				
	001	fs/8	fs/16			
	101	fs/12	fs/24			
Timer 3	TC3M 0	fs/8	fs/16	Same as values at left		TC3M is bit 5 of TM1 (x'3F5B')
	1	fs/2	fs/4			
Timer 4	TC4M 0	fs/8	fs/16	Same as values at left		TC4M is bit 7 of TM1 (x'3F5B')
	1	TC4I				
Timer 6	TC6CK 00	fosc/512		fxi		TC6CK0 and TC6CK1 are bits 5 and 6 of TC6CNT (x'3F6B')
	01	fs/4	fs/8	Same as values at left		
	10	fs/64	fs/128			

Table 5-8-2 Timer Clock Source Summary (2/2)

Timer name	Microcomputer mode	Clock source selection setting	SIXI Terminal 0	SIXI Terminal 1	Comments
Timer 5	NORMAL SLOW IDLE	wds 0	fs/65536	fs/65536	Functions as a watchdog timer.
		1	fs/524288	fs/524288	
	HALT STOP	–	STOP	STOP	The watchdog timer stops.
	When released from the reset condition	DLYCNT 00	fs/256	fs/256	fs=fosc/1024 fixed
	After recovery from STOP	DLYCNT 00	fs/256	fs/256	Functions for waiting for the oscillations to stabilize.
		01	fs/131072	fs/131072	
		10	fs/16384	fs/16384	
		11	fs	fs	Use prohibited

The value of fs in the table is set by the OSCSEL1 and OSCSEL2 flags (bp6 and bp5) of the CPU Mode Control Register (CPUM: x'3F00'). (Refer to Figure 12-16, "CPU Mode Control Register.")

5-8-2 Clock Output Function

This product incorporates the functions of outputting various clock signals at different terminals as follows.

Function	Output terminal	Output characteristics (Note 2)	Waveform & duty ratio (Note 1) (Note 2)	Clock frequency	Interrupt	Reference
High speed oscillator output (Note 1)	OSCO	Inverter, feedback, 2.5V center	Sinewave, 50%	fosc = 14.32 Mhz (17.72 Mhz) Depends on the externally connected crystal oscillator	Absent	1-3-2
Low speed oscillator output (Note 1)	XO	Inverter, feedback, 2.5V center	Sinewave, 50%	fxi = 32768 kHz (Normally) Depends on the externally connected crystal oscillator	Absent	1-3-2
Oscillator output for OSD (Note 1)	Port 21 (OSCO2)	Inverter, feedback, 2.5V center	Sinewave, 50%	f_{OSC02} = For generating the dot clock. Depends on the externally connected crystal oscillator	Absent	1-3-2 Fig. 3-2-13
Small amplitude OSC frequency divided output (Note 1)	Port 17 (OSCDIV)	Resistance divider, 2.5V center, 1Vpp output, 1k Ω output impedance	Charging and discharging waveform (pseudo-triangular waveform) 50%	f_{OSCDIV} = fosc/4 (P17DIR = 1) = 3.58MHz (Note 3)	Absent	3-2-2 Fig. 3-2-11
		Resistance divider, 2.5V center, 1Vpp output, 1k Ω output impedance		f_{OSCDIV} = fosc/2 (P17DIR = 0) = 7.16MHz (Note 3)		
Timer 3 1/2 frequency divided output	Port 14 (TC3O)	CMOS	Rectangular waveform 50%	f_{TC3O} = fss/ (16 × Nc) (Note 4) (TC3M = 0)	Present	5-4 Fig. 3-2-8
				f_{TC3O} = fss/ (4 × Nc) (Note 4) (TC3M = 1)		
Variable duty clock output	Port 25 (PWM14)	CMOS	Pulse waveform 0 to 100% variable in units of 1/256	$f_{PWM14} = fosc/2^{14}$ = 874Hz (Note 3)	Present	9-4-3
Buzzer clock output	Port 46 (BUZZER)	CMOS	Rectangular waveform 50%	$f_{BUZZER} = fss/2^{10}$ (BUZS = 0)	Absent	5-8
				$f_{BUZZER} = fss/2^{11}$ (BUZS = 1)		
				$f_{BUZZER} = fss/2^{12}$ (BUZS = 2)		
				$f_{BUZZER} = fss/2^{13}$ (BUZS = 3)		

Note 1: Do not provide extended interconnections since the load capacitances, etc., affect the waveform.

Note 2: These are the characteristics for:
- Vdd =5.0V
- fosc = 14.32 Mhz
- Temperature = 25 °C
- No load

Note 3: The values are calculated for fosc = 14.32 Mhz.

Note 4:

Microcomputer mode	Servoclk flag	TC3M Flag = 0	TC3M Flag = 1
NORMAL Mode (fs=fosc/2)	0	f_{TC3O}=fosc/(32 × Nc)	f_{TC3O}=fosc/(8 × Nc)
	1	f_{TC3O}=fosc/(64 × Nc)	f_{TC3O}=fosc/(16 × Nc)
SLOW Mode (fs=fxi/2)	0	f_{TC3O}=fxi/(32 × Nc)	f_{TC3O}=fxi/(8 × Nc)
	1	f_{TC3O}=fxi/(64 × Nc)	f_{TC3O}=fxi/(16 × Nc)

However, Nc = x'10000'-N

(N: The value set in the register TB3 0 to 65535)

Example: When outputting 1/64 frequency divided output of the signal at XI.

(1) x'FFFE' is set in the register TB3. (similar as Nc=2)
(2) Set the TC3M flag to '0'.
(3) Set the Servoclk flag to '0'.
(4) Set x'03' in the register CPUM and change over to the SLOW mode of half the frequency of fxi.
(5) Set '1' in the P14SEL flag. (Selects TC3O output)
(6) Set '1' in the P1DIR4 flag. (Port 14 is set to the output mode.)
(7) Set '1' in the TC3E flag. (Timer 3 operation)

When the above operations are made, a clock signal with 1/64 frequency of Xi is output from port 14.
(f_{TC3O} = fxi/(32 × Nc) = fxi/64)

Chapter 6 Serial Functions

6

6-1 Serial 0 Functions

6-1-1 Serial 0 Configuration

Serial Interface 0 (SIF0) can be synchronized to an 8-bit clock or can be used asynchronously (with a UART). Data transfer is performed with either three pins: Clock Pin (NSBT0), Transmit Data Pin (NSBO0) and Receive Data Pin (NSBI0), or two pins: Clock Pin (NSBT0) and Transmit Data Pin (NSBO0). The NSBO0, NSBI0, and NSBT0 pins output positive logic data. Serial Interface 0 consists of the following components.

If Serial Interface 0 is used as a synchronous serial interface, the start condition function cannot be used. (The start condition function must be running when using the UART serial interface.)

(1) Serial Transfer Clock Generator Unit
(2) Shift Register Unit
(3) I/O Control Unit
(4) Interrupt Control Unit

(1) Serial Transfer Clock Generator Unit

The serial transfer clock generator unit permits selection of the following as the serial transfer clock: external clock, timer 4 output frequency divided by 2 (ftc4/2), or the system clock (fss) divided by 4, 8, 16, 32, 64, 128, or 256. The serial transfer clock is set by the SC0CK0 to SC0CK2 Flags (bits 3 to 5) of the Serial Interface 0 Mode 1 Register (SC0MD1: x'3F98', R/W). An external clock can be selected by configuring the NSBT Pin as an input.

(2) Shift Register Unit

The shift register unit transmits and receives data, one bit at a time. If this device is configured as a synchronous serial interface, the number of bits for transfer (from 8 bits to 1 bit) can be set in binary code in the SC0LNG0 to SC0LNG2 (bp 0 to bp2) Flags of Serial Interface 0 Mode Register 0 (SC0MD0: x'3F97', R/W). On the other hand, if the UART serial interface is set, the number of bits for transfer (7 or 8 bits) can be specified by the SC0FM1 and SC0FM2 (bp3 and bp4) Flags of Serial Interface 0 Mode Register 2 (SCOMD2: x'3F99', R/W).

(3) I/O Control Unit

The I/O control unit performs the following controls.
- Input control for serial transfer clock
- Serial data output control
- Shift direction control (LSB or MSB) of the shift register
- Switching between synchronous and asynchronous (UART) clocks
- Switching between 2-wire and 3-wire use of Serial 0

(4) Interrupt Control Unit

The interrupt control unit generates an interrupt when the Serial Interface 0 transfer is complete.

6-1-2 Synchronous Serial Interface Transmission Settings for Serial 0

Operation of the serial interface begins when data is written to the shift buffer. The bit counter is incremented for each bit transferred. Transfer stops when the bit counter overflows.

An arbitrary number of bits from 1 to 8 can be transferred. The number of bits to be transferred must be set before beginning the transfer.

—Operation and Setting Procedure—

◆ Transmission

Perform the following procedure to set transmission of the synchronous serial interface.

(1) Set the SC0CMD Flag of the Serial Interface 0 Control Register (SC0CTR) to '0' to select synchronous serial data transfer.
(2) Select the number of bits for transfer, from 1 to 8 bits, with the SC0LNG2 to SC0LNG0 Flags of the Serial Interface 0 Mode Register 0 (SC0MD0).
(3) Specify the first bit for transfer (MSB first or LSB first) with the SC0DIR Flag of the SC0MD0 Register.
(4) Select the polarity of the valid clock edge with the SC0CE1 and SC0CE0 Flags of the SC0MD0 Register.
(5) In the case of an internal clock source:
- Select the clock source with the SC0CK1 and SC0CK0 Flags of the Serial Interface 0 Mode Register 1 (SC0MD1).
- Set the SC0CKM Flag of the SC0MD1 Register to specify whether or not the clock source is to be divided by 8.

If the values of the SC0CE1 and SC0CE0 Flags of the SC0MD0 Register are changed while the serial port is enabled, the number of bits for transfer, specified by the SC0LNG2 to SC0LNG0 Flags of the SC0MD0 Register, may be incremented.

At the end of transmission, the SBO0 pin gets fixed to the High level.
The data hold period is half the set internal clock period when transmission is performed using the internal clock, and is 1/2 to 1 period of the set internal clock when transmission is done using an external clock.

If the SC0IOM Flag of the SC0MD3 Register is set for "pin connection" (2-wire method), the SBIO Pin may be used as a port.
Data is received when the SBO0 Pin is set to input mode and data is transmit when the SBO0 Pin is set to output mode.

- Set to SC0SBTS Flag of the Serial Interface 0 Mode Register 3 (SC0MD3) to '1' to set the serial clock.
- Set the SC0SBTM Flag of the SC0MD3 Register.
- Set bit 2 of the Port 0 Direction Control Register (P0DIR) to output mode.

In the case of an external clock source (SBT0 Pin input):

- Set the SC0SBTM Flag of the SC0MD3 Register.
- Set bit 0 of the P0DIR Register to input mode.

(6) Set the SC0BTM Flag of the SC0MD3 Register.

(7) Set the SC0IOM Flag of the SC0MD3 Register.

(8) Set the SC0SBOS Flag of the SC0MD3 Register to '1' to specify serial communication.

(9) Set the transfer data in the Serial Interface 0 Transmit-Receive Shift Register (SC0TRB). Operation of the serial transmission will begin.

(10) When serial transmission begins, the SC0BSY Flag of the SC0CTR Register will change to '1' to indicate that a serial transfer is in progress.

(11) When the serial transmission is complete, the Serial 0 Interface Request Flag (SI0IR: bp0) of the Serial 0 Interface Control Register (SIF0ICR: x'3FF9') is set.

When manipulating the interrupt request flag (xxxIR), it is necessary to set beforehand the IR write enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, reset IRWE to '0' after the manipulation.

After a transfer is complete, the number of bits for transfer, as specified by the SC0LNG1 to SC0LNG0 Flags of the SC0MD0 Register, will change. Except in the case of an 8-bit transfer, at the time of the next reception, reset the number of bits to be transferred.

When switching from transmission to reception, first set the SC0SBOS Flag of the SC0MD3 Register to '0', and then set the SC0SBIS Flag to '1'. Do not change the values of both flags at the same time. (Common to 2-wire and 3-wire methods)

Before setting the SC0SBOS Flag of the SC0MD3 Register to '1', the SC0SBTS Flag of the SC0MD3 Register must be set to '1'. Always follow the setting sequence.

When transmitting using the slave clock, a waiting period is necessary from the start of transmission to the clock input. This waiting period is the time required for the data to be transferred from the send data buffer (SC0TRB) to the shift register. This waiting period is determined based on the transfer clock set by SC0CKM and SC0CK2-0, and two transfer clock periods are required after writing data in the send data buffer.

6-1-3 Synchronous Serial Interface 0 Transmission Timing

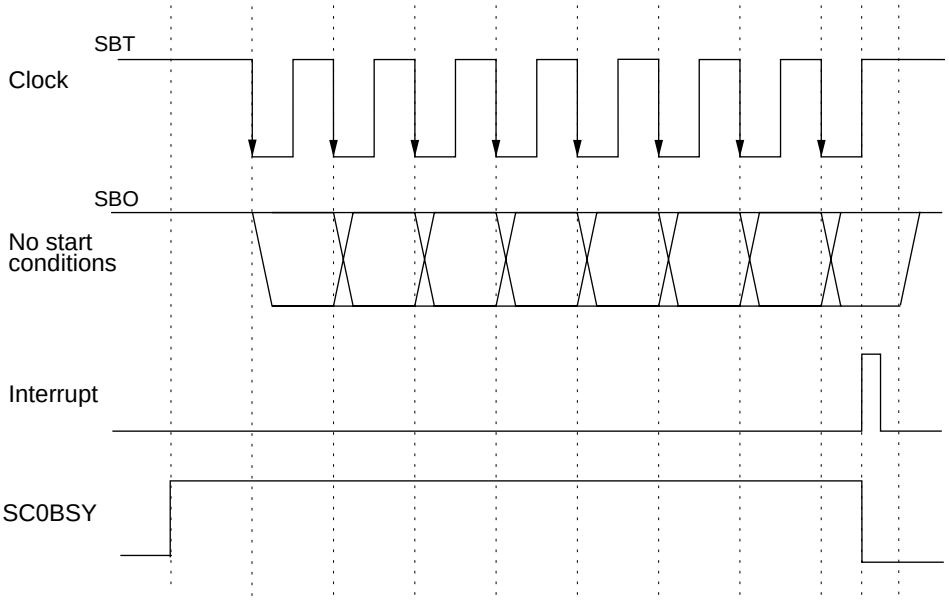


Figure 6-1-1 Synchronous Serial Interface Transmission Timing (Falling Edge Transmission)

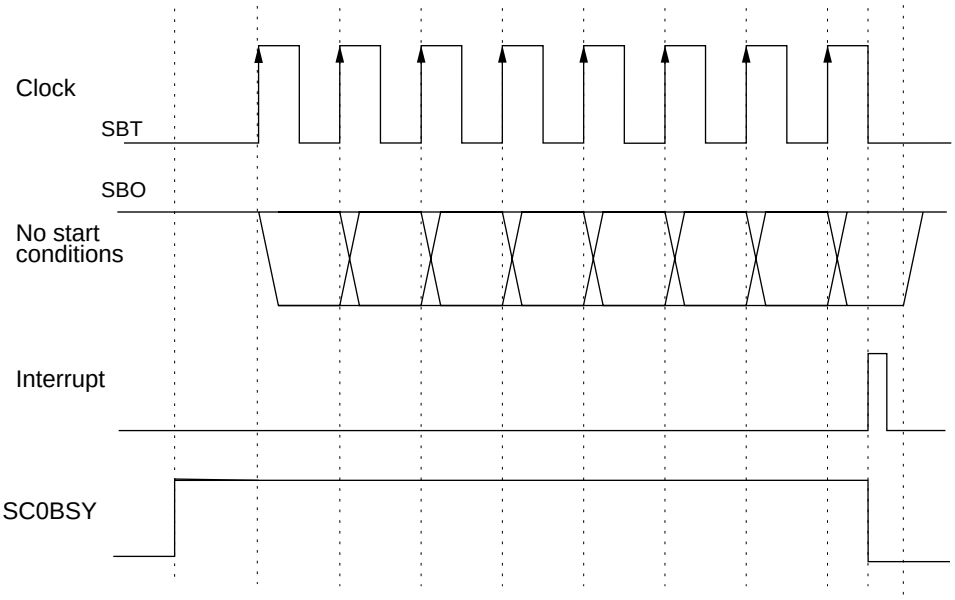


Figure 6-1-2 Synchronous Serial Interface Transmission Timing (Rising Edge Transmission)

6-1-4 Synchronous Serial Interface Reception Settings for Serial 0

◆ Reception

Perform the following procedure to set reception of the synchronous serial interface.

(1) Set the SC0CMD Flag of the Serial Interface 0 Control Register (SC0CTR) to '0' to select synchronous serial data transfer.

(2) Select the number of bits for transfer, from 1 to 8 bits, with the SC0LNG2 to SC0LNG0 Flags of the Serial Interface 0 Mode Register 0 (SC0MD0).

(3) Specify the first bit for transfer (MSB first or LSB first) with the SC0DIR Flag of the SC0MD0 Register.

(4) Select the polarity of the valid clock edge with the SC0CE1 and SC0CE0 Flags of the SC0MD0 Register.

(5) In the case of an internal clock source:
- Select the clock source with the SC0CK2 and SC0CK0 Flags of the Serial Interface 0 Mode Register 1 (SC0MD1).
- Set the SC0CKM Flag of the SC0MD1 Register to specify whether or not the clock source is to be divided by 8.
- Set to SC0SBTS Flag of the Serial Interface 0 Mode Register 3 (SC0MD3) to '1' to set the serial clock pin
- Set the SC0SBTM Flag of the SC0MD3 Register.
- Set bit 2 of the Port 0 Direction Control Register (P0DIR) to output mode. (P02/SBT0 Output Mode)

In the case of an external clock source (SBT0 Pin input):
- Set bit 2 of the P0DIR Register to input mode.

(6) Set the SC0IOM Flag of the SC0MD3 Register.

(7) Set the SC0SBIS Flag of the SC0MD3 Register to '1' to specify serial communication. (The reception wait state is entered.)

(8) When serial transmission begins, the SC0BSY Flag of the SC0CTR Register will change to '1' to indicate that a serial transfer is in progress.

(9) When the serial transmission is complete, SC0BSY Flag of the SC0CTR Register changes to '0' and the Serial 0 Interface Request Flag (SI0IR: bp0) of the Serial 0 Interface Control Register (SIF0ICR: x'3FF9') is set.

If the values of the SC0CE1 and SC0CE0 Flags of the SC0MD0 Register are changed while the serial port is enabled, the number of bits for transfer, specified by the SC0LNG2 to SC0LNG0 Flags of the SC0MD0 Register, may be incremented.

Before setting the SC0SBIS Flag of the SC0MD3 Register to '1', the SC0SBTS Flag of the SC0MD3 Register must be set to '1'. (Always follow the setting sequence.)

When an internal clock source is used, after setting the SC0SBIS Flag of the SC0MD3 Register to '1', write dummy data to the SC0TRB Register. Also write dummy data to the SC0TRB Register at the time of the next data reception.

When manipulating the interrupt request flag (xxxIR), it is necessary to set beforehand the IR write enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, reset IRWE to '0' after the manipulation.

After a transfer is complete, the number of bits for transfer, as specified by the SC0LNG1 to SC0LNG0 Flags of the SC0MD0 Register, will change. Except in the case of an 8-bit transfer, at the time of the next reception, reset the number of bits to be transferred.

When switching from reception to transmission, first set the SC0SBIS Flag of the SC0MD3 Register to '0', and then set the SC0SBOS Flag to '1'. Do not change the values of both flags at the same time. (Common to 2-wire and 3-wire methods)
When switching between transmission and transmission/reception, maintain the settings of the SC0SBIS flag and the SC0SBOS flag of SC0MD3 kept at '1'.

6-1-5 Synchronous Serial Interface Reception Timing

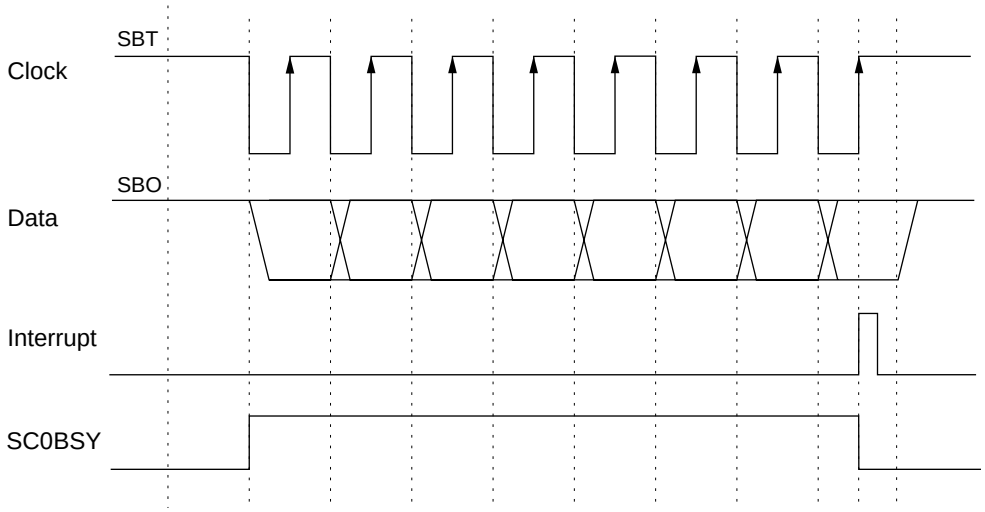


Figure 6-1-3 Synchronous Serial Interface Reception Timing (Rising Edge)

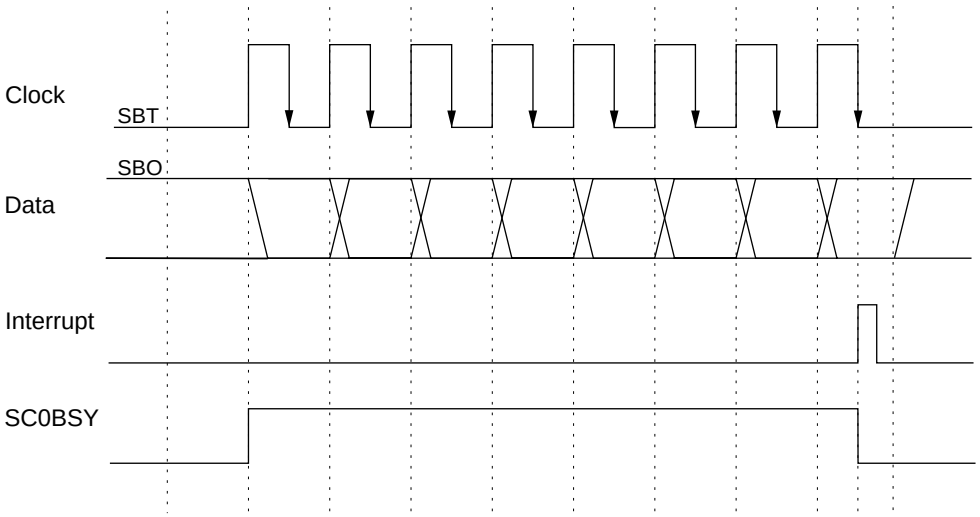


Figure 6-1-4 Synchronous Serial Interface Reception Timing (Falling Edge)

6-1-6 UART Serial Interface Operation

Data to be transferred by the UART serial interface can be set by the frame setting. In the frame mode, 7 or 8-bit transfer data and 1 or 2 stop bits can be selected. The existence of a parity bit can also be selected. The parity bit can be selected as fixed 0, fixed 1, even or odd.

—UART Serial Interface Setting Procedure—

◆ Transmission

Perform the following procedure to set transmission of the UART serial interface.

(1) Set bit 0 of the P0OUT Register to '1'.
(2) Set the SC0CMD Flag of the Serial Interface 0 Control Register (SC0CTR) to '1' to select UART serial data transfer.
(3) Specify the first bit for transfer (MSB first or LSB first) with the SC0DIR Flag of the SC0MD0 Register.
(4) Select the clock source with the SC0CK2 and SC0CK0 Flags of the Serial Interface 0 Mode Register 1 (SC0MD1).
(5) Set the SC0CKM Flag of the SC0MD1 Register to '1'. The clock source must be divided by 8.
(6) Specify whether there will be a parity bit with the SC0NPE Flag of the Serial Interface 0 Mode Register 2 (SC0MD2).
(7) If a parity bit was selected in (6), select the added bit with the SC0PM1 and SC0PM0 Flags of the Serial Interface 0 Mode Register 2 (SC0MD2).
(8) Specify the frame mode with the SC0FM1 and SC0FM0 Flags of the SC0MD2 Register.
(9) Select the break status transmit control with the SC0BRKE Flag of the SC0MD2 Register.
(10) Set the SC0SBOM Flag of the SC0MD3 Register.
(11) Set the SC0IOM Flag of the SC0MD3 Register.
(12) Set bit 0 of the Port 0 Direction Control Register (P0DIR) to output mode.
(13) Set the SC0SBOS Flag of the SC0MD3 Register to '1' to select serial communication.
(14) Set the data to be transferred in the Serial Interface 0 Transmit-Receive Shift Register (SC0TRB).

This will start the serial transmission.

If using as a UART serial interface, the clock source must be divided by 8. (Set the SC0CKM Flag of the SC0MD1 Register to '1'.)

If using as a UART serial interface, set the SC0CE1 and SC0CE0 Flags of the SC0MD0 Register to '00'. The polarity of the valid edge for data I/O cannot be changed while using the UART.

If the values of the SC0CE1 and SC0CE0 Flags of the SC0MD0 Register are changed while the serial port is enabled, the number of bits for transfer, specified by the SC0LNG2 to SC0LNG0 Flags of the SC0MD0 Register, may be incremented.

If the SC0IOM Flag of the SC0MD3 Register is set for "pin connection" (2-wire method), the SBIO Pin may be used as a port.
Data is received when the SBO0 Pin is set to input mode and data is transmit when the SBO0 Pin is set to output mode.

If the SC0FM Flag of the SC0MD2 Register is set, the SC0LNG2 to SC0LNG0 Flags of the SC0MD0 Register will be automatically set.

After transmission is complete, the SBO0 Pin goes to a '1' level.

(15) When serial transmission begins, the SC0BSY Flag of the SC0CTR Register will change to '1' to indicate that a serial transfer is in progress.

(16) When the serial transmission is complete, the SC0BSY Flag of the SC0CTR Register changes to '0' and the Interrupt Request Flag (bit 0) of the SIF0ICR register is set to '1'.

If using as a UART serial interface, the clock source must be divided by 8. (Set the SC0CKM Flag of the SC0MD1 Register to '1'.)

If using as a UART serial interface, set the SC0CE1 and SC0CE0 Flags of the SC0MD0 Register to '00'. The polarity of the valid edge for data I/O data cannot be changed while using the UART.

If the values of the SC0CE1 and SC0CE0 Flags of the SC0MD0 Register are changed while the serial port is enabled, the number of bits for transfer, specified by the SC0LNG2 to SC0LNG0 Flags of the SC0MD0 Register, may be incremented.

If the SC0FM Flag of the SC0MD2 Register is set, the SC0LNG2 to SC0LNG0 Flags of the SC0MD0 Register will be automatically set.

If the SC0FM Flag of the SC0MD2 Register is set, the SC0LNG2 to SC0LNG0 Flags of the SC0MD0 Register will be automatically set.

After reception is complete, the SBI0 Pin goes to a '1' level.

◆ Reception

Perform the following procedure to set reception of the UART serial interface.

(1) Set the SC0CMD Flag of the Serial Interface 0 Control Register (SC0CTR) to '1' to select UART serial data transfer.

(2) Specify the first bit for transfer (MSB first or LSB first) with the SC0DIR Flag of the SC0MD0 Register.

(3) Select the clock source with the SC0CK2 and SC0CK0 Flags of the Serial Interface 0 Mode Register 1 (SC0MD1).

(4) Set the SC0CKM Flag of the SC0MD1 Register to '1'. The clock source must be divided by 8.

(5) Specify whether there will be a parity bit with the SC0NPE Flag of the Serial Interface 0 Mode Register 2 (SC0MD2).

(7) If a parity bit was selected in (5), select the added bit with the SC0PM1 and SC0PM0 Flags of the Serial Interface 0 Mode Register 2 (SC0MD2).

(8) Set the SC0IOM Flag of the SC0MD3 Register.

(9) If the SC0IOM Flag of the SC0MD3 Register has selected the pin to be unconnected, set the input mode by setting bit 1 of the Port 0 Direction Control Register (P0DIR) to '0'.

(10) Set the SC0SBIS Flag of the SC0MD3 Register to '1' to specify serial communication.

(11) Serial reception begins when the reception data signal is received. The SC0BSY Flag of the SC0CTR Register will change to '1' to indicate that a serial transfer is in progress.

(12) When the serial reception is complete, the SC0BSY Flag of the SC0CTR Register changes to '0' and the Interrupt Request Flag (bit 0) of the SIF0ICR register is set to '1'.

When manipulating the interrupt request flag (xxxIR), it is necessary to set beforehand the IR write enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, reset IRWE to '0' after the manipulation.

◆ Operation during a Reception Error

After the serial reception is complete, the error flag is set to '1' if there is an error in the received data.

The error flag is set to '1' when an error occurs. Although this flag is automatically cleared at the time the next serial transfer complete interrupt occurs, it can also be cleared by directly writing in '0' in the flag.

- Parity error: Evaluate the parity bit added to the received data and indicate the result. Set the error flag to '1' if there is a parity error.
- Framing error: Evaluate whether the stop bit of the received data was correctly received and indicate the result.
- Overrun error: Determine whether the received data has been read. When two or more receive interrupts have been generated and the SC0RXC Register has not been read, set the flag to '1'.
- Error monitor flag: This flag is set to '1' if any of the parity, framing, or overrun errors have occurred.

◆ Operation of the Break Status Reception Monitor (SC0BRKF)

The break state reception monitor (SC0BRKF) is set to '1' when a break data is received.

6-1-7 UART Serial Interface Transfer Timing Diagram

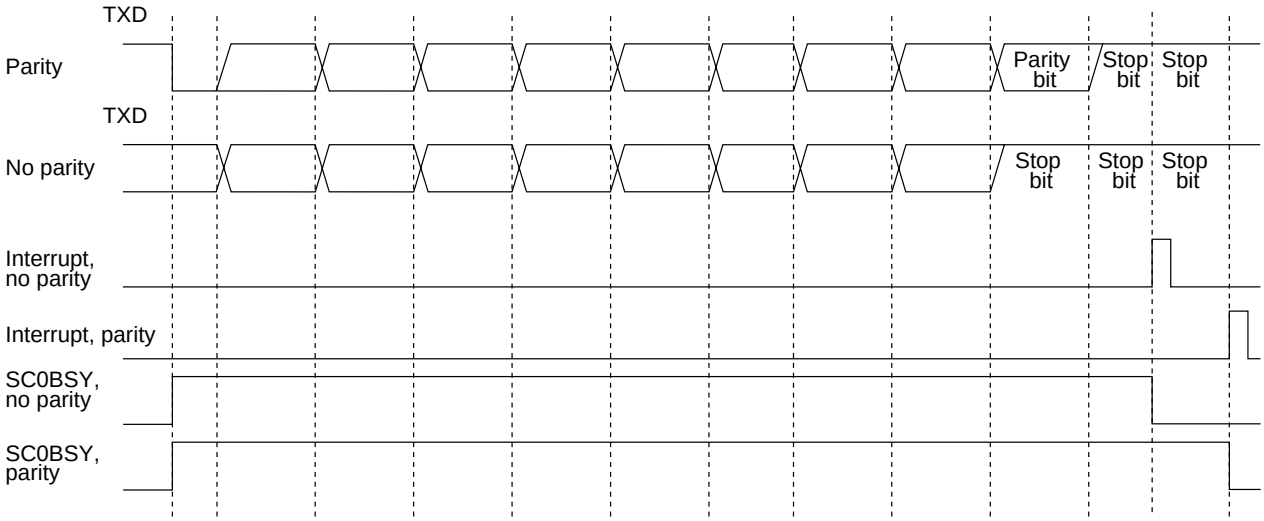


Figure 6-1-5 UART Transmission Timing Diagram

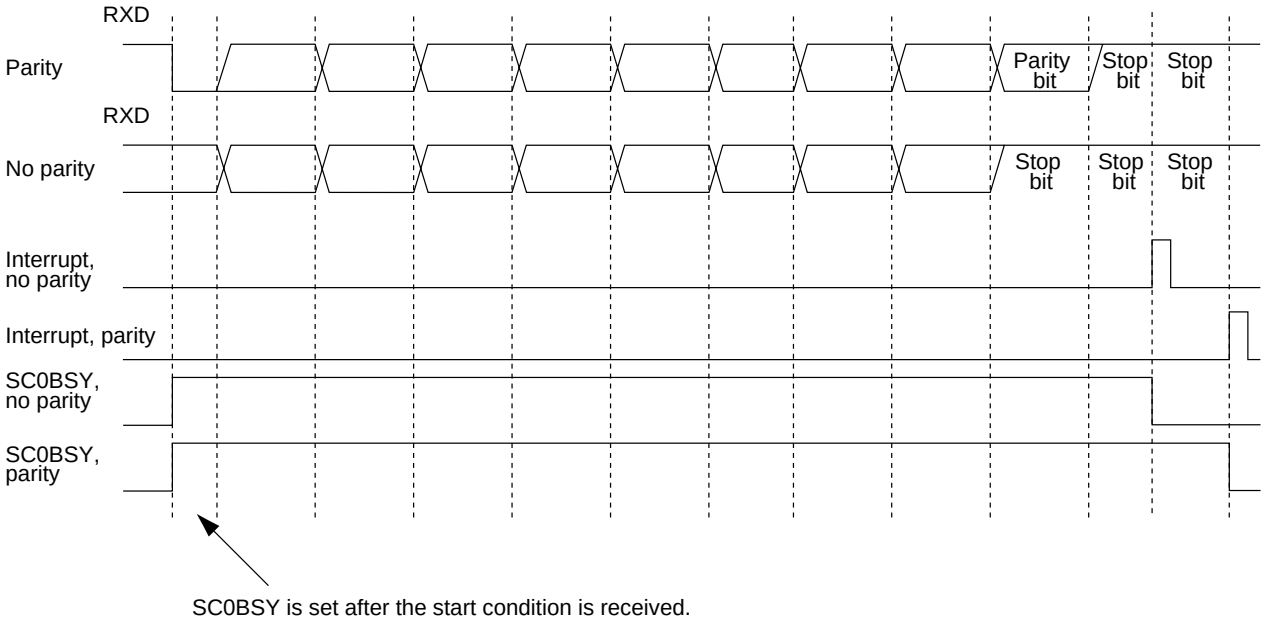


Figure 6-1-6 UART Reception Timing Diagram

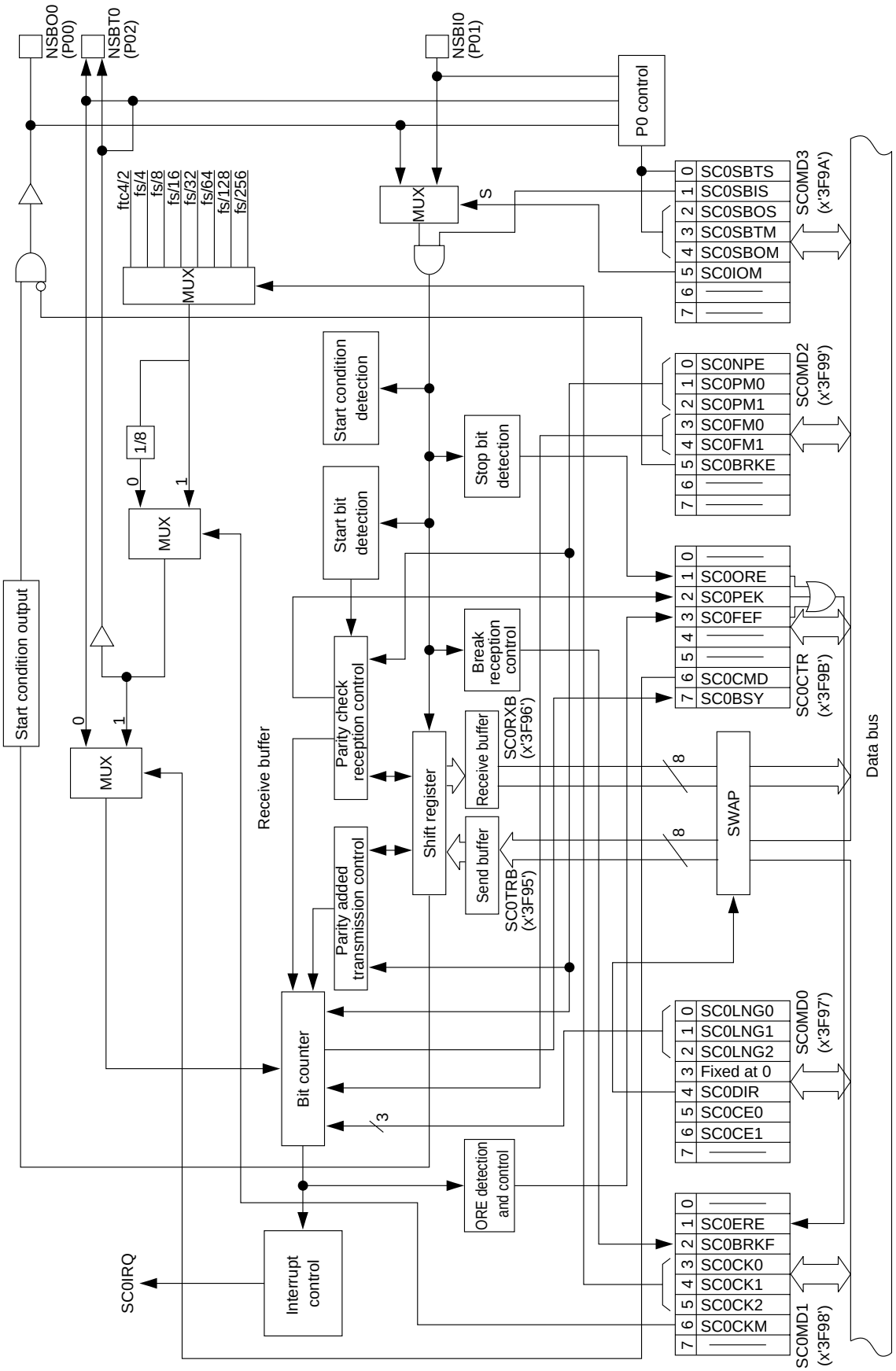


Figure 6-1-7 Serial 0 Block Diagram

6-2 Serial 1

6-2-1 Serial 1 Configuration

Serial Interface 1 (SIF1) is an 8-bit synchronous serial interface. Data transfer is performed with three pins: Clock Pin (NSBT1), Transmit Data Pin (NSBO1) and Receive Data Pin (NSBI1). Serial Interface 1 consists of the following components.

(1) Serial Transfer Clock Generator Unit
(2) Shift Register Unit
(3) I/O Control Unit
(4) Interrupt Control Unit

(1) Serial Transfer Clock Generator Unit

The serial transfer clock generator unit permits selection of the following as the serial transfer clock: external clock, timer 4 output frequency divided by 2 (ftc4/2), or the system clock (fss) divided by 8, 16, 32, 64, 128, or 256. The serial transfer clock is set by the SI1CM0 to SI1CM2 Flags (bits 0 to 2) of the Serial 1 Mode Register (SIM1: x'3F9C', R/W).

(2) Shift Register Unit

The shift register unit transmits and receives data, one bit at a time. Data for transmission is set by writing to the Transmit Data Buffer Register. When an internal clock is used as the transfer clock, the serial transfer begins as soon as data is written to the Transmit Data Register. Data is output with negative logic.
Received data is stored in the Receive Data Buffer Register. The received data may be obtained by reading the Receive Data Buffer Register. Data is input with negative logic.

Serial Interface 1 has a simple remote control receive function. Shifting can be performed continuously with this feature.

Data cannot be written during a serial transfer (while the shift register is operating). Data must be written when the transfer clock is at a high-level.

(3) I/O Control Unit

The I/O control unit performs the following controls.

- Input control for serial transfer clock
- Serial data output control
- Start condition function control
- Shift direction control (LSB or MSB) of shift register

(4) Interrupt Control Unit

When the transfer of the 1-byte interrupt source is complete, the interrupt control unit will generate an interrupt request.

When manipulating the interrupt request flag (xxxIR), it is necessary to set beforehand the IR write enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, reset IRWE to '0' after the manipulation.

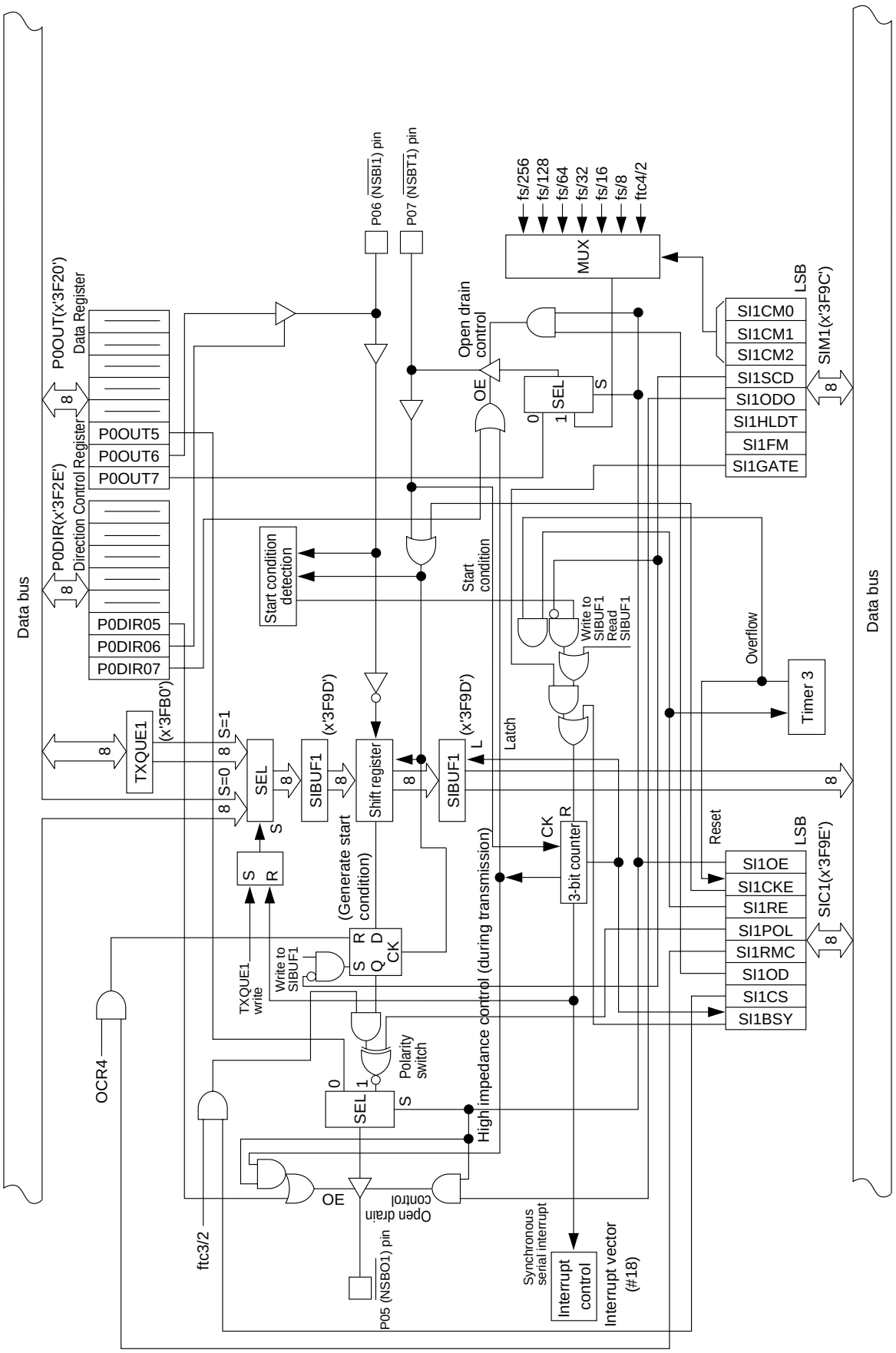


Figure 6-2-1 Serial 1 Block Diagram

6-2-2 Serial Interface 1 (SIF1) Operation

Serial Interface 1 (SIF1) is an 8-bit synchronous serial interface. Data transfer is performed with three pins: Clock Pin (NSBT1), Transmit Data Pin (NSBO1) and Receive Data Pin (NSBI1). Start conditions can be specified.

(1) After data to be transmit is written to the Transmit Data Buffer Register (SIBUF1: x'3F9D'), the serial transfer will begin at the falling edge of the transfer clock (NSBT1). Transmission data is output from the Transmit Data Pin (NSBO1).

(2) At the rising edge of the transfer clock (NSBT1), reception data from the Receive Data Pin (NSBI1) is input to the shift register. After the reception is complete, received data can be obtained by reading the Receive Data Buffer Register (SIBUF1).

(3) The start condition function can be enabled or disabled for both transmission and reception. If enabled during transmission, the NSBO1 pin outputs a low-level signal before data is output from the shift register. During reception, when the NSBT1 pin is at a high-level, the falling edge at the NSBI1 pin's transition from high to low-level is recognized as the start condition and will be detected.

(4) The Timer 4 divided-by-2 clock is a clock source dedicated for remote control transmission. When Timer 4 divided-by-2 is selected as the clock source, a serial clock is output while operation of Timer 4 continues.

If for some reason, the clock input to the NSBT1 pin is delayed with respect to the data input to the NSBI1 pin, the start condition will be recognized and incorrect operation may result.

When using the remote control output function, do not use Serial 1 data reception. It may not be possible to receive correct data.

The remote control output function cannot specify the first bit during transfers. When using the remote control output function, data is output 'MSB first.' 'LSB first' cannot be specified.

Consecutive serial transfers that use the TXQUE1 Register are not possible.

As a general rule when using the remote control output function, divide the Timer 4 output by 2 for use as the clock source. If a different clock source must be used, chose a clock slower than fs/32. Do not use fs/8 or fs/16.

Set the NSBO1 and NSBT1 pins to '1' before performing the serial transfer.

6-2-3 Serial Interface 1 (SIF1) Transmission Timing

The transfer clock can be selected from the following: 1/8, 1/16, 1/32, 1/64, 1/128, or 1/256 of the system clock, 1/2 of Timer 4 (ftc4/2), or an external clock. The data hold interval for transmission can be selected as approximately 10 or 22 machine cycles.

The timing diagrams below show operation when approximately 22 machine cycles have been specified.

(1) The following timing diagram illustrates internal clock transmission with the start condition enabled.

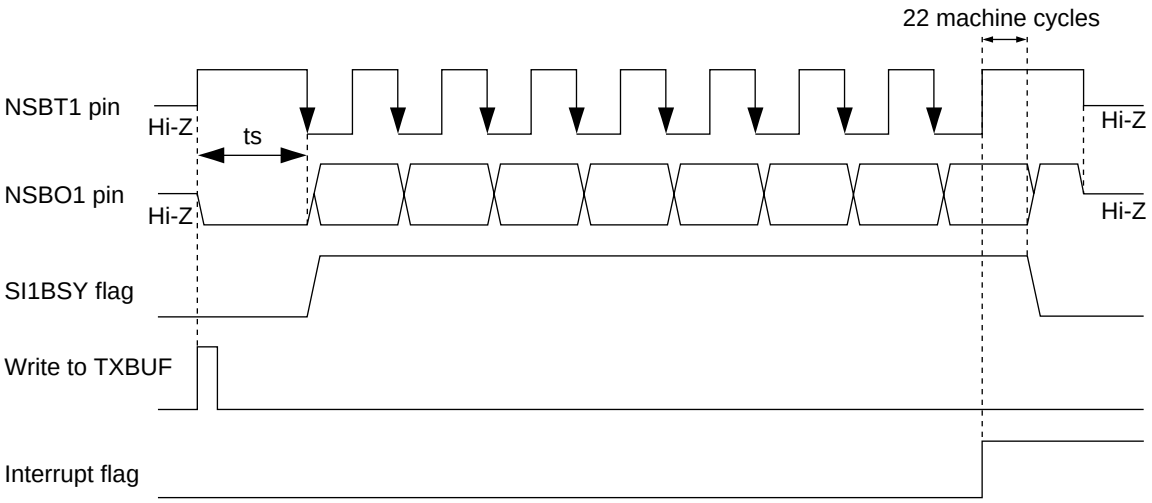


Figure 6-2-2 Transmission Timing (for Internal Clock Transmission and Start Condition Enabled)

When data is written to the SIBUF1 Register, the 3-bit counter (SBC) is cleared and the NSBO1 and NSBT1 pins change to low and high levels respectively. Clock output begins after approximately 1/2 of the transfer clock cycle. In synchronization with that clock, data to be transmit is output to the NSBO1 pin. Transmission data output from the NSBO1 pin changes at the falling edge of the clock at the NSBT1 pin. The data value is held for approximately 22 machine cycles beginning from the 8th rising edge of the transfer clock. Next, the NSBO1 pin outputs a high-level signal for approximately 1 machine cycle, and then enters a high impedance state. At that time, the NSBT1 pin also enters a high impedance state. After writing data to the SIBUF1 Register, the Transmit-Receive Busy Flag SI1BSY (SIC1: x'3F9E', bp7) is set at the falling edge of the first transfer clock. This flag is reset when the data transfer is complete. The Interrupt Request Flag is set at the rising edge of the 8th transfer clock.

(2) The following timing diagram illustrates internal clock transmission with the start condition disabled.

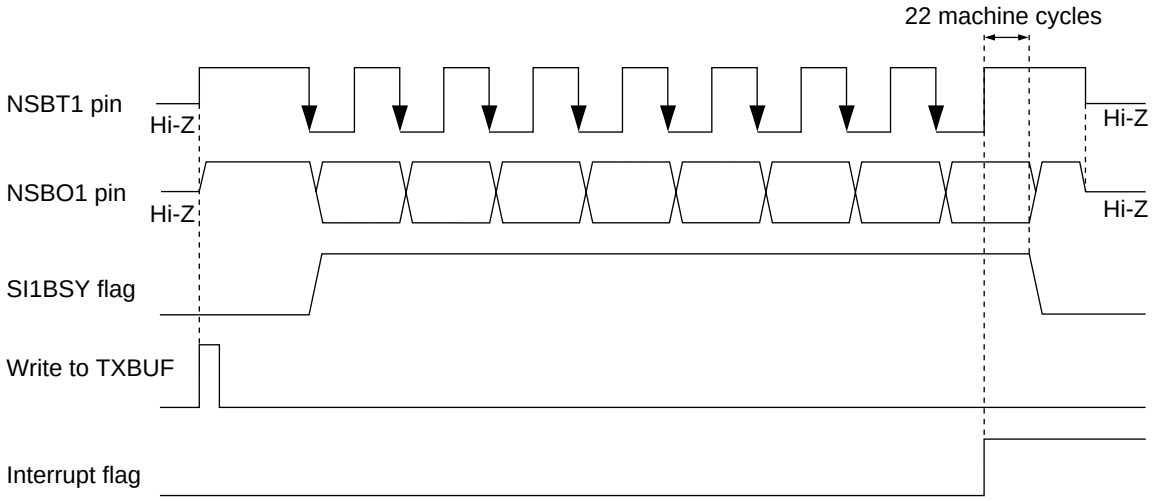


Figure 6-2-3 Transmission Timing (for Internal Clock Transmission and Start Condition Disabled)

When data is written to the SIBUF1 Register, the 3-bit counter (SBC) is cleared and the NSBO1 and NSBT1 pins change to high-levels. After approximately 1 transfer clock cycle (ts), the clock output begins. All other operation is the same as for the case with enabled start condition.

(3) The following timing diagram illustrates external clock transmission with the start condition enabled.

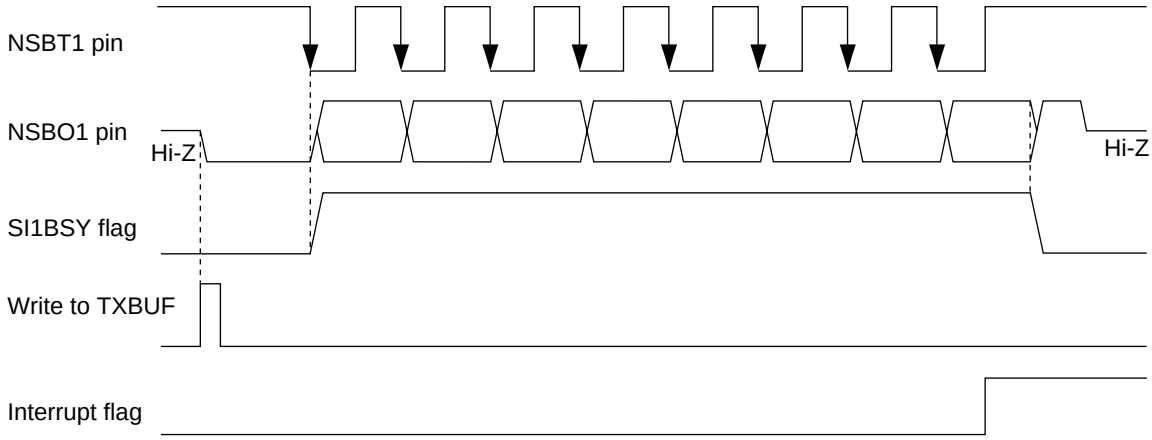


Figure 6-2-4 Transmission Timing (for External Clock Transmission and Start Condition Enabled)

When data is written to the SIBUF1 Register, the 3-bit counter (SBC) is cleared and the NSBO1 pin goes to a low-level. Then, if a clock signal is input to the NSBT pin, data to be transmit will be output to the NSBO1 pin in synchronization with that clock. Transmission data output from the NSBO1 pin changes at the falling edge of the clock at the NSBT1 pin. After approximately 22 machine cycles from the 8th rising edge of the transfer clock, the NSBO1 pin will output a high-level for approximately 1 machine cycle and then enter a high impedance state. At that time, the NSBT1 pin also enters a high impedance state. After writing data to the SIBUF1 Register, the Transmit-Receive Busy Flag SI1BSY (SIC1: x'3F9E', bp7) is set at the falling edge of the first transfer clock. This flag is reset when the data transfer is complete. The Interrupt Request Flag is set at the rising edge of the 8th transfer clock.

(4) The following timing diagram illustrates external clock transmission with the start condition disabled.

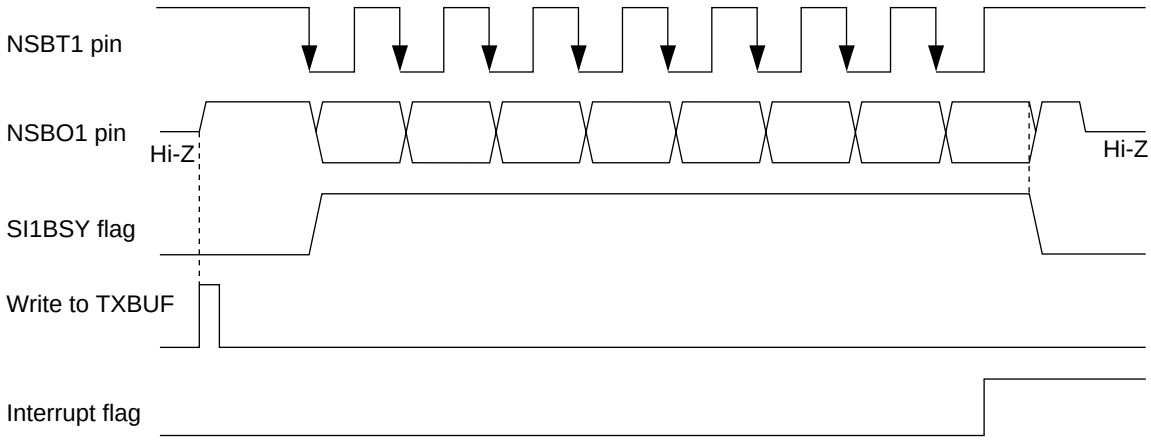


Figure 6-2-5 Transmission Timing (for External Clock Transmission and Start Condition Disabled)

When data is written to the SIBUF1 Register, the 3-bit counter (SBC) is cleared and the NSBO1 and NSBT1 pins change to high-levels. Then, if the clock is input to the NSBT1 pin, transmission data will be output to the NSBO1 pin in synchronization with this clock. All other operation is the same as for the case with enabled start condition.

6-2-4 Serial Interface 1 (SIF1) Reception Timing

The transfer clock can be selected from the following: 1/8, 1/16, 1/32, 1/64, 1/128, or 1/256 of a machine cycle, 1/2 of Timer 4 (ftc4/2), or an external clock. The data hold interval for transfer can be selected as approximately 10 or 22 machine cycles.

The timing diagrams below show operation when approximately 22 machine cycles have been specified.

(1) The following timing diagram illustrates internal clock reception with the start condition enabled.

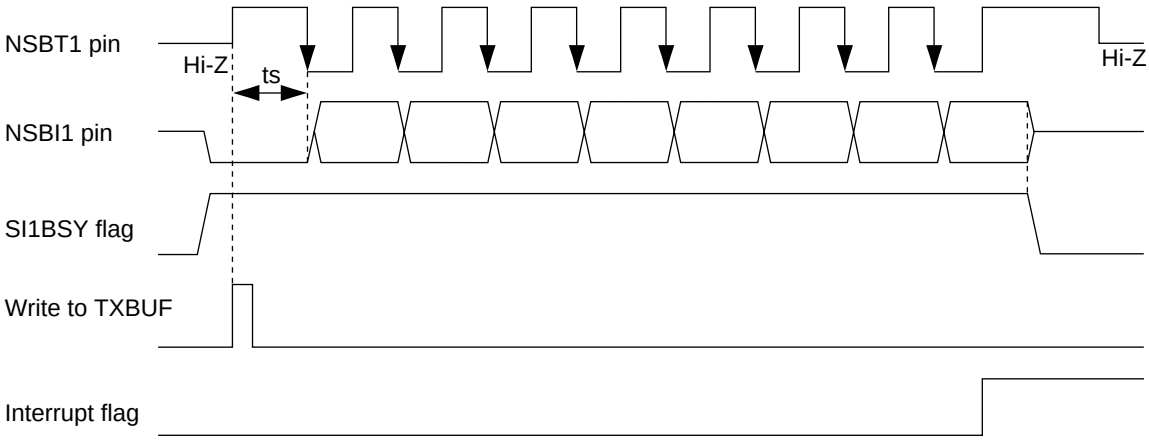


Figure 6-2-6 Reception Timing (for Internal Clock Transmission and Start Condition Enabled)

The 3-bit counter (SBC) is cleared if the start condition is detected before the clock is output. When data is written to the SIBUF1 Register, the 3-bit counter (SBC) is cleared even if the start condition has not been detected.

When data is written to the SIBUF0 Register, the NSBT1 pin goes to a high-level. Clock output begins after approximately 1 transfer clock cycle. Data input from the NSBI1 pin is latched into the shift register at the rising edge of the clock output from the NSBT1 pin. The shift register shifts one bit at the falling edge of the clock and latches sequential data in synchronization with the clock. After 8 clocks are output, the received data can be read by reading the SIBUF1 Register. The Transmit-Receive Busy Flag SI1BSY (SIC1: x'3F9E', bp7) is set if the start condition is detected before the clock is output. However, if the start condition is not detected, this flag is set at the falling edge of the first transfer clock after data is written to the SIBUF1 Register, and is reset after approximately 22 machine cycles from the rising edge of the 8th clock. The Interrupt Request Flag is set at the rising edge of the 8th transfer clock.

(2) The following timing diagram illustrates internal clock reception with the start condition disabled.

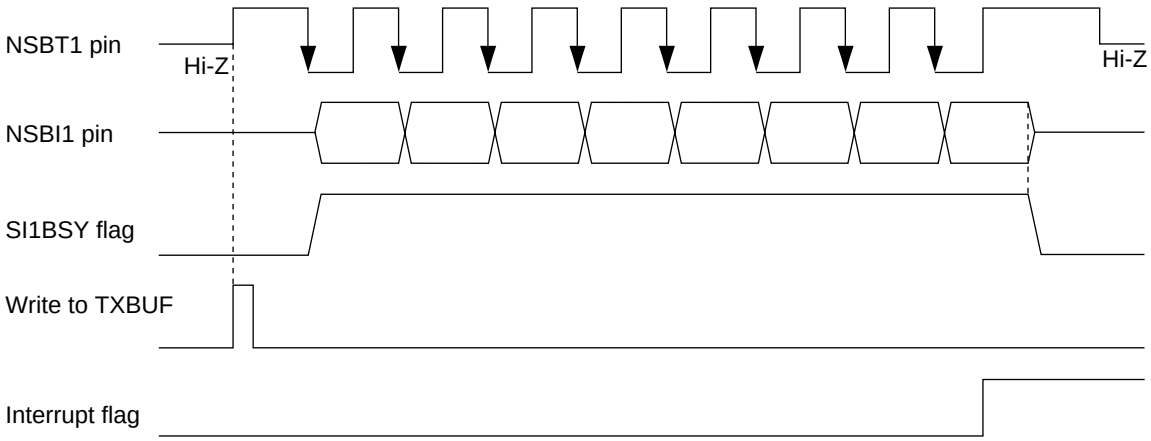


Figure 6-2-7 Reception Timing (for Internal Clock Transmission and Start Condition Disabled)

When data is written to the SIBUF1 Register, the 3-bit counter (SBC) is cleared and the NSBT1 pin goes to a high-level. Clock output begins after approximately 1 transfer clock cycle. Data input from the NSBI1 pin is latched into the shift register at the rising edge of the clock output from the NSBT1 pin. The shift register shifts one bit at the falling edge of the clock and latches sequential data in synchronization with the clock. After 8 clocks are output, the received data can be read by reading the SIBUF1 Register. Even if the start condition is detected before the clock is output, the Transmit-Receive Busy Flag SI1BSY (SIC1: x'3F9E', bp7) is not set. However, this flag is set at the falling edge of the first transfer clock after data is written to the SIBUF1 Register, and is reset after approximately 22 machine cycles from the rising edge of the 8th clock. The Interrupt Request Flag is set at the rising edge of the 8th transfer clock.

(3) The following timing diagram illustrates external clock reception with the start condition enabled.

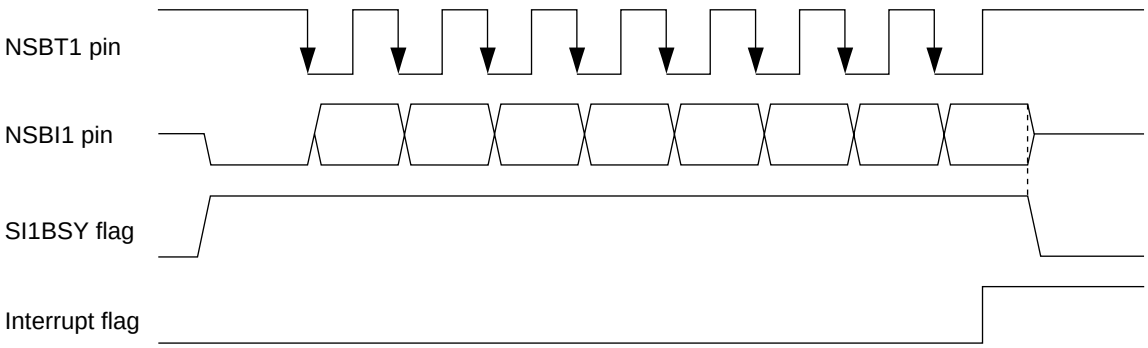


Figure 6-2-8 Reception Timing (for External Clock Transmission and Start Condition Enabled)

The 3-bit counter (SBC) is cleared if the start condition is detected before the clock is input to the NSBT1 pin. Data input from the NSBI1 pin is latched into the shift register at the rising edge of the clock input to the NSBT1 pin. The shift register shifts one bit at the falling edge of the clock and latches sequential data in synchronization with the clock. After 8 clocks have been input, the received data can be read by reading the SIBUF1 Register. The Transmit-Receive Busy Flag SI1BSY (SIC1: x'3F9E', bp7) is set if the start condition is detected before the clock is input. However, if the start condition is not detected, this flag is set at the falling edge of the first clock and is reset after approximately 22 machine cycles from the rising edge of the 8th clock. The Interrupt Request Flag is set at the rising edge of the 8th transfer clock.

(4) The following timing diagram illustrates external clock reception with the start condition disabled.

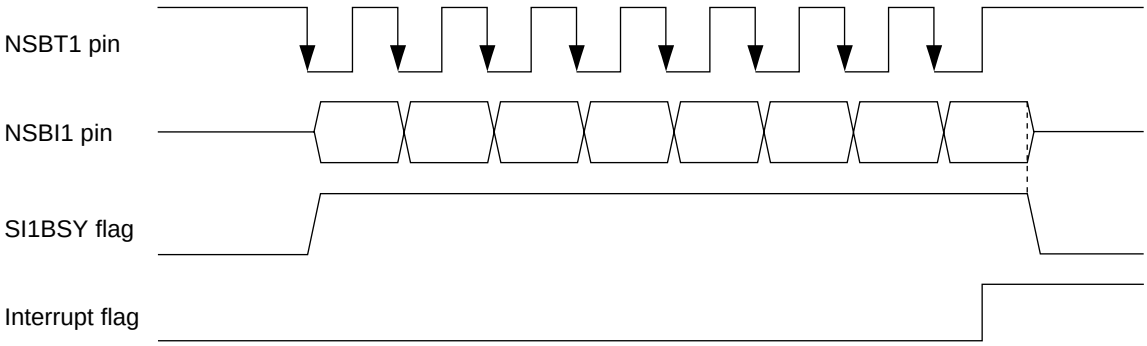


Figure 6-2-9 Reception Timing (for External Clock Transmission and Start Condition Disabled)

Input of the start condition does not cause the 3-bit counter (SBC) to be cleared. Data input from the NSBI1 pin is latched into the shift register at the rising edge of the clock input to the NSBT1 pin. The shift register shifts one bit at the falling edge of the clock and latches sequential data in synchronization with the clock. After 8 clocks have been input, the received data can be read by reading the SIBUF1 Register. Even if the start condition is detected, the Transmit-Receive Busy Flag SI1BSY (SIC1: x'3F9E', bp7) is not set. However, this flag is set at the falling edge of the first clock and is reset after approximately 22 machine cycles from the rising edge of the 8th clock. The Interrupt Request Flag is set at the rising edge of the 8th transfer clock.

6-2-5 Serial Interface 1 (SIF1) Registers

(1) Serial 1 Mode Register (SIM1)

	7	6	5	4	3	2	1	0	
SIM1	SI1GATE	SI1HLDT	SI1FM	SI1ODO	SI1SCD	SI1CM2	SI1CM1	SI1CM0	(At reset: 00000000)

Serial 1 Mode Register (SIM1) is a read and write accessible 8-bit register.

SI1CM0 to SI1CM2 (bp0 to bp2) specify the transfer clock for the Serial 1 Interface. These bits can be set to select the following clocks: external clock, 1/2 of Timer 4 output (ftc4/2), and 1/8, 1/16, 1/32, 1/64, 1/128, or 1/256 of the system clock. If an external clock is to be selected, in addition to setting bits SI1CM0 through SI1CM2, also configure P07 as an input.
SI1SCD (bp3) specifies the Serial 1 start condition. In the initial state, the start condition is enabled.
SI1ODO (bp4) specifies the output of the NSBO pin. In the initial state, this bit is '0' and output is possible. Setting this bit to '1' changes the output configuration to an n-channel, open-drain output.
SI1FM (bp5) specifies the shift direction of the Serial 1 Interface. If SI1FM is set to '0', the shift direction is 'LSB first.' If set to '1', the shift direction is 'MSB first.' During remote control output, the shift direction is always 'MSB first' as it is not possible to specify the first bit.
SI1HLDT (bp6) sets the data hold time of the Serial 1 transfer. If set to '0', the hold time is approximately 10 machine cycles. Or, if set to '1', the hold time is approximately 22 machine cycles. If a clock source of 1/8 or 1/16 of the system clock, or an external clock faster than 1/16 of the system clock is input as the transmit clock, set SI1HLDT to '0' to select approximately 10 machine cycles for the data hold time.
SI1GATE (bp7) specifies shift operation after input of the serial transfer clock. In the initial state, shift operation is disabled after the serial transfer clock is input.

(2) Transmit-Receive Buffer (SIBUF1)

	7	6	5	4	3	2	1	0	
SIBUF1	SIBUF7	SIBUF6	SIBUF5	SIBUF4	SIBUF3	SIBUF2	SIBUF1	SIBUF0	(At reset: XXXXXXXX)

Transmission data can be set by writing to theTransmit-Receive Buffer (SIBUF1). Received data can be obtained by reading this register. The Serial 1 Interface begins transmission when the transmission data is set.

(3) Serial 1 Control Register (SIC1)

	7	6	5	4	3	2	1	0	
SIC1	SI1BSY	SI1CS	SI1OD	SI1RMC	SI1POL	SI1RE	SI1CKE	SI1OE	(At reset: 00000000)

Serial 1 Control Register (SIC1) is a read and write accessible 8-bit register.

The SI1OE Flag (bp0) specifies port or serial I/O. In the initial state, this flag is set to port I/O. If Serial Interface 1 is to be used, set this bit to '1'.
The SI1CKE Flag (bp1) enables or disables the clock supply to Serial 1. Setting this bit to '1' disables the clock supply for Serial 1.
The SI1RE Flag (bp2) specifies the Serial 1 transfer header function. Setting this bit to '1' enables the Serial 1 Transfer Header Function.
The SI1POL Flag (bp3) switches output polarity of the Serial 1 Output (NSBO1) Pin. In the initial state, this bit is '0' and the output is with negative logic.
The SI1RMC flag (bp4) specifies whether or not to use the remote control output pulse width control function based on OCR4 when using remote control output. When this bit is set to '1', the remote control output pulse width control function based on OCR4 becomes effective.
The SI1OD Flag (bp5) specifies the output format of the NSBT Pin. In the initial state, push-pull output is set. Setting this bit to '1' changes the configuration to an n-channel, open-drain output.
The SI1CS Flag (bp6) specifies whether the TC3 output divided by 2 is used.
The SI1BSY Flag (bp7) indicates when a transmission or reception is in progress over the Serial 1 Interface. When this bit is read, a value of '1' indicates that a transmission or reception is in progress over the Serial 1 Interface. The Serial 1 Interface can be initialized by setting this bits to '1'. This bit will be automatically cleared to '0'. Normal serial operation is performed when set to '0'.

6-2-6 Remote Control Output Function

For Timer 3 and Timer 4, refer to Chapter 5, "Timer Functions".

As shown in the diagram below, the remote function is realized with the combination of Timer 3, Timer 4 and the Serial 1 Interface.

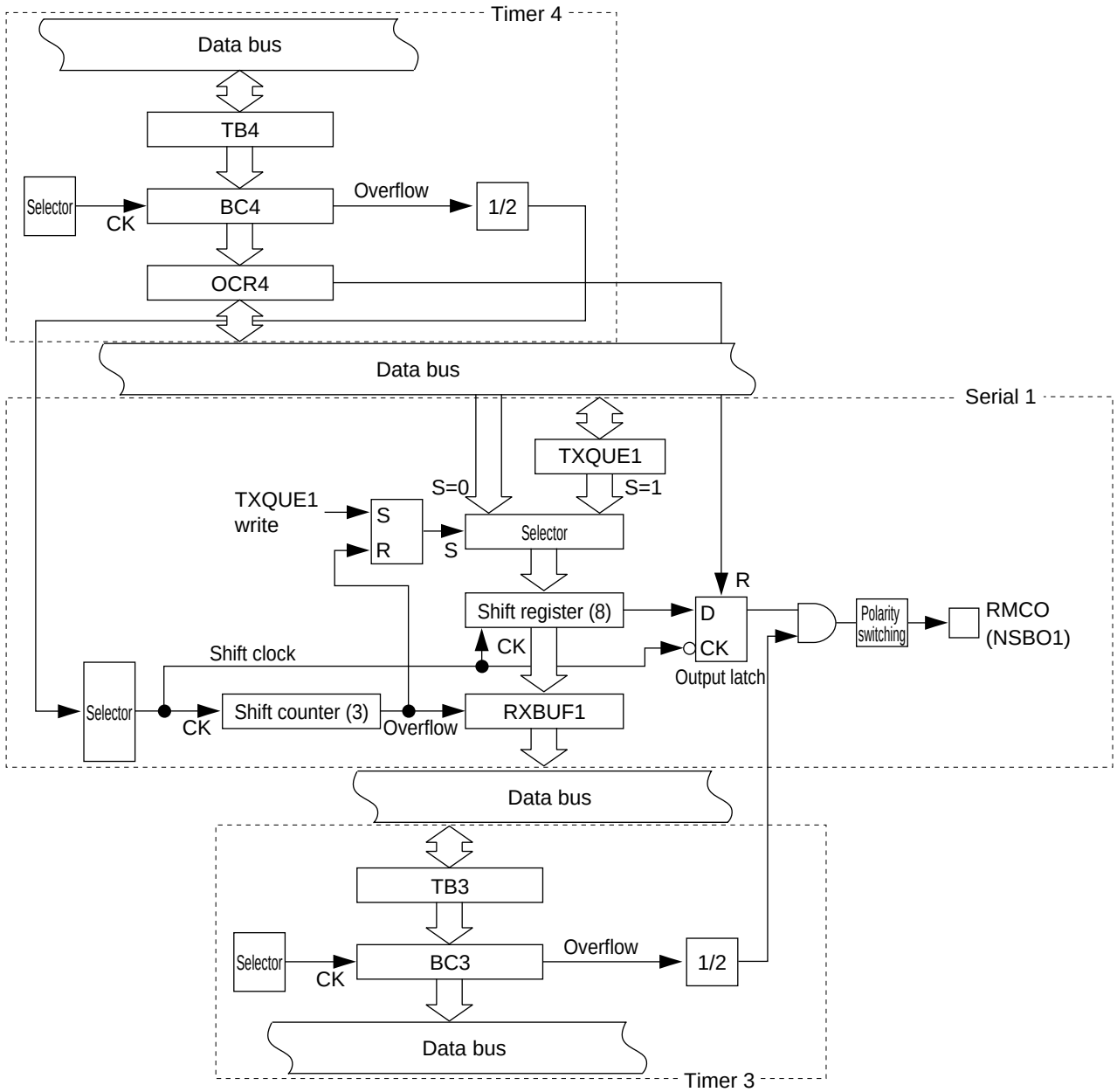


Figure 6-2-10 Remote Control Output Circuit

1. Remote Control Output with No Carrier

As shown in the diagram below, the remote control pulse is generated by Timer 4 and Serial 1.

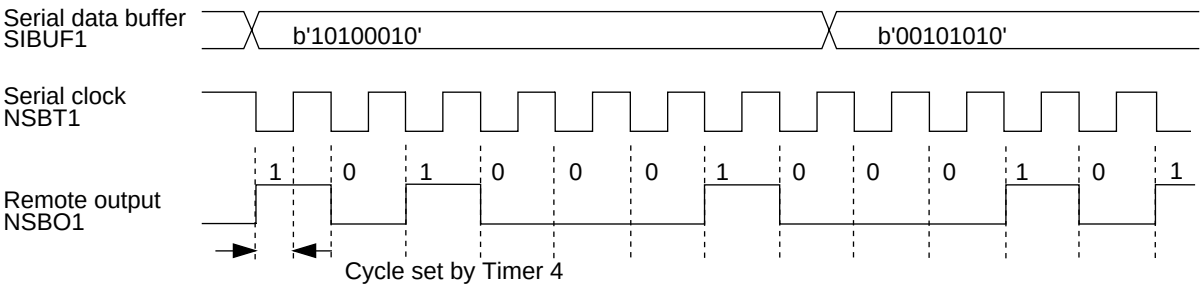


Figure 6-2-11

Remote control output without carrier is realized by the following settings.

(1) Set the Timer 4 period in the Timer 4 period setting register (TB4: x'3F49' to x'3F48').
(2) Set bit 7 to '1' and bit 5 to '0' in the Port 0 data output register (P0OUT: x'3F20').
(3) Set bits 7 and 5 to '1' in the Port 0 direction control register (P0DIR: x'3F2E').
(4) Set x'a9' in the Serial 1 mode register (SIM1: x'3F9C').

If not wanting to output the transfer clock to the NSBT1 pin when using the remote control output function, set bit 3 of register P0DRV (x'3F3C') to '1'.
In this case, the NSBT1 pin (P07) can be used as a general purpose input/output pin.

The reasons for setting x'a9' in SIM1 are the following:
- ***Making the clock the 1/2 frequency signal of Timer 4 output.***
- ***Making the start condition as 'None'.***
- ***Making the NSBO1 pin a CMOS output.***
- ***Making the output with the MSB first.***
- ***Making the data hold period equal to 22 machine cycles.***
- ***Specifying the shift operation after serial transfer.***

The reasons for setting x'88' in SIC1 are the following:
- ***Enabling the clock supply of Serial 1.***
- ***Disabling start search of Serial 1 transfer.***
- ***Making the NSBO1 output a positive logic output.***
- ***Disabling pulse width control based on OCR4.***
- ***Making the NSBT1 pin a CMOS output.***
- ***Attaching no carrier in the TC3 output.***
- ***Resetting Serial 1.***

The reason for setting bit 0 after setting bits 1 to 7 is to ensure that there is no unwanted output made at the SBO1 pin when selecting bit 1 to bit 7.

(5) After setting x'88' in the Serial 1 control register (SIC1: x'3F9E'), set bit 0 to '1'.

(6) Set the data to be transmitted in the serial interface 1 send/receive shift buffer register (SIBUF1: x'3F9D).

(7) When performing continuous transmissions, set the data at this point in the remote control output data queue buffer register (TXQUE1: x'3FB0').

(8) To start remote control transmission, set bit 6 to '1' of the timer mode register 1 (TM1: x'3F5B') and initiate Timer 4 operation.
When Timer 4 overflows, the clock of Serial 1 changes thereby starting remote control transmission.

* When transmitting continuous data, set the next data in the TXQUE1 register after the transmission of the previous byte of data has been started.

■ Procedure of Interrupting Remote Control Transmission

Set x'88' in register SIC1 after confirming that the last data of continuous transmission has been transmitted and bit 0 of register SIF1ICR has become '1'.

■ Procedure During Remote Control Re-transmission

(1) Set the dummy data x'00' in register SIBUF1.

(2) After waiting for a Timer 4 overflow at which the serial clock changes state, set x'09' in register SIC1.

(3) Set the send data in register TXQUE1.

2. Remote Control Output with Carrier

As shown in the diagram below, the remote control pulse is generated by Timer 3, Timer 4 and the Serial 1 Interface.

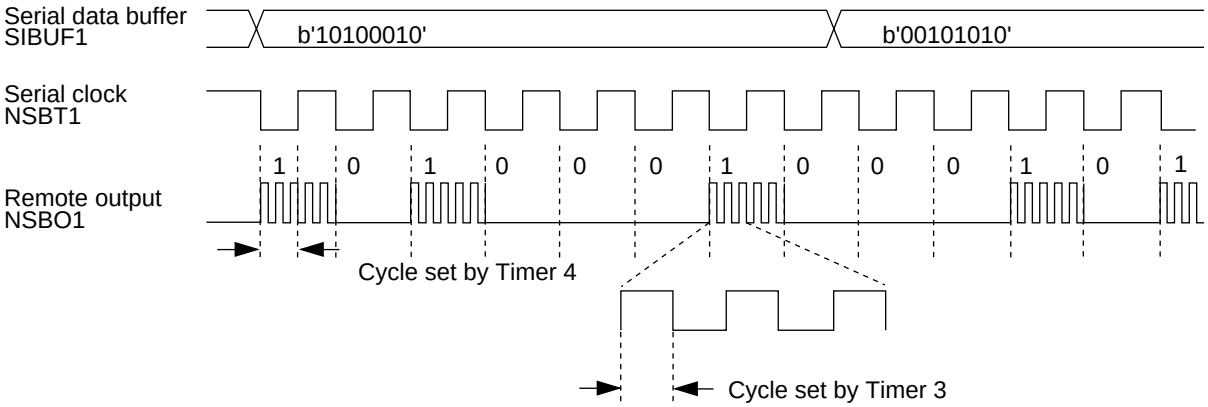


Figure 6-2-12

The remote control output with carrier is realized by the following settings:

(1) Set the Timer 3 period for generating the carrier in the Timer 3 period setting register (TB3: x'3F47' to x'3F46').

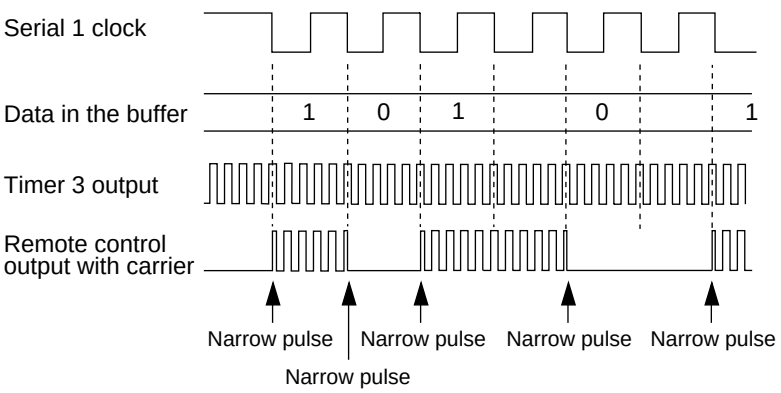
(2) Set the Timer 4 period for generating the serial clock in the Timer 4 period setting register (TB4: x'3F49' to x'3F48').

(3) Set bit 7 to '1' and bit 5 to '0' in the Port 0 data output register (P0OUT: x'3F20').

(4) Set bits 7 and 5 to '1' in the Port 0 direction control register (P0DIR: x'3F2E').

(5) Set x'a9' in the Serial 1 mode register (SIM1: x'3F9C').

In the case of remote control output with carrier, the pulse width may become narrow in the data change-over part of the remote control output.



In the case of remote control output with carrier, the pulse width may become narrow in the remote control output. (Since the start timing of Timer 3 is initiated by software, it is not possible to synchronize it with the serial clock (Timer 4).)

The reasons for setting x'a9' in SIM1 are the following:

- ***Making the clock equal to 1/2 frequency signal of Timer 4 output.***
- ***Making the start condition as 'None'.***
- ***Making the NSBO1 pin a CMOS output.***
- ***Making the output with MSB first.***
- ***Making the data hold period equal to 22 machine cycles.***
- ***Specifying the shift operation after serial transfer.***

Setting x'88' in the initial SIC1 is for making the NSBO1 pin output '0'.
If bit 6 of SIC1 is set to '1' before making the initial serial transmission, some unwanted data is likely to be output during the initial serial transmission. Because of this, the dummy data x'ff' should be set as the first byte of data, and bit 6 of SIC1 should be set to '1' during the transmission of the dummy byte of data.

(6) Set x'88' in the Serial 1 control register (SIC1: x'3F9E').
(7) Set the dummy data x'ff' in the serial interface 1 send/receive shift buffer register (SIBUF1: x'3F9D').
(8) Set bit 6 to '1' in the timer mode register 1 (TM1: x'3F5B) to start Timer 4 operation.
(9) Wait until Timer 4 overflows.
(10) After setting x'40' in the SIC1 register, set bit 0 to '1'.
(11) Set the first byte of send data in the remote control output data queue buffer register (TXQUE1: x'3FB0').
(12) While the dummy data set in register SIBUF1 is being transmitted, set bit 7 to '1' in the timer mode register 0 (TM0: x'3F5A') and start Timer 3 operation.

■ Procedure of Interrupting Remote Control Data Transmission

(1) Before interrupting transmission, set the dummy data x'ff' in register TXQUE1 while the final send data is being transmitted.
(2) After confirming that the interrupt request flag (SIF1ICR: x'3FFA') bit 0 has become '1', indicating that the final send data transmission has been completed, set x'00' in register SIC1.
* At this time, the dummy data is not transmitted.

■ Procedure for Retransmission of Remote Control Data

(1) Set the dummy data x'ff' in register SIBUF1.
(2) Set x'41' in register SIC1.
(3) Set the send data in register TXQUE1.

3. Variable-Pulse-Width Remote Control Output

As shown in the table below, remote control cycle "t1" and remote control output pulse width "t2" are generated by Timer 4 and synchronous output OCR4 respectively.
In order to control pulse width with OCR4 (to use the remote control width control function (OCR)), set SI1RMC (bp4) of the Serial 1 Control Register (SIC1: x'3F9E') to '1'.

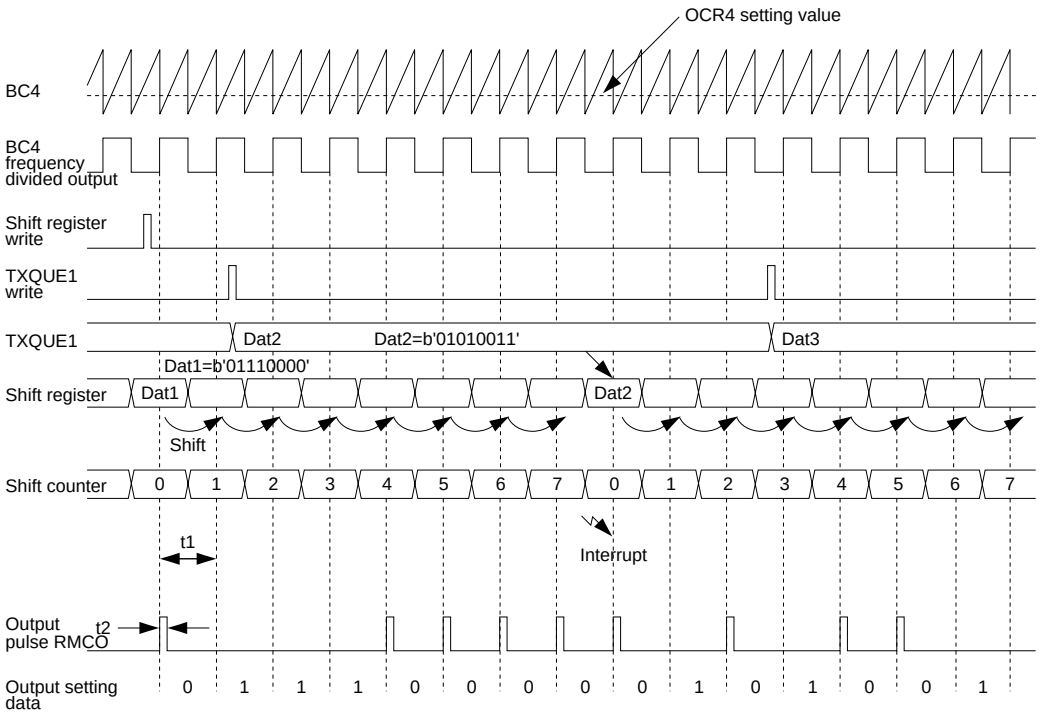


Figure 6-2-13

Variable pulse width remote control output is realized by the following settings:

(1) Set the Timer 4 period for generating the serial clock in the Timer 4 period setting register (TB4: x'3F49' to x'3F48').
(2) Set the same value as that set in register TB4 in step (1) in the Timer 4 output compare register (OCR4: x'3F68' to x'3F67').
(3) Set bit 7 to '1' and bit 5 to '0' in the Port 0 data output register (P0OUT: x'3F20').
(4) Set bits 7 and 5 to '1' in the Port 0 direction control register (P0DIR: x'3F2E').

If the same values are not set in registers TB4 and OCR4 before the first serial data transmission, unwanted data may transmitted during the first serial transmission. Therefore, transmit the dummy data x'ff' as the first byte, and while this dummy data is being transmitted, set the required pulse width value in OCR4.

The reasons for setting x'a9' in SIM1 are the following:
- ***Making the clock equal to 1/2 frequency signal of Timer 4 output.***
- ***Making the start condition 'None'.***
- ***Making the NSBO1 pin a CMOS output.***
- ***Making the output with MSB first.***
- ***Making the data hold period equal to 22 machine cycles.***
- ***Specifying the shift operation after serial transfer.***

(5) Set x'a9' in the Serial 1 mode register (SIM1: x'3F9C').

The reasons for making the settings x'90' -> x'10' -> x'11' in register SIC1 are the following:
Setting x'90' is for resetting Serial 1 before setting data in register SIBUF1.
Next, setting x'10' -> x'11' is to prevent the transmission of unwanted data that is likely when bits 1 to 7 and bit 0 of register SIC1 are set simultaneously.

(6) Set x'90' in the Serial 1 control register (SIC1: x'3F9E').
(7) Set the dummy data x'ff' in the serial interface 1 send/receive shift buffer register (SIBUF1: x'3F9D').
(8) Set bit 6 to '1' in the timer mode register 1 (TM1: x'3F5B) to start Timer 4 operation.
(9) Wait until Timer 4 overflows.
(10) After setting x'10' in the SIC1 register, set bit 0 to '1'.
(11) Set the first byte of send data in the remote control output data queue buffer register (TXQUE1: x'3FB0').

If the same values are not set in registers TB4 and OCR4 before the first serial data transmission, unwanted data may be transmitted during the first serial transmission.
Therefore, transmit the dummy data x'ff' as the first byte, and while this dummy data is being transmitted, set the required pulse width value in OCR4.

(12) While the dummy data is being transmitted, set the output pulse width value in register OCR4.

■ Procedure of Interrupting Remote Control Data Transmission

(1) Before interrupting transmission, set the dummy data x'ff' in register TXQUE1 while the final send data is being transmitted.
(2) After confirming that bit 0 of the interrupt request flag (SIF1ICR: x'3FFA') has become '1', indicating that the final send data transmission has been completed, set x'00' in register SIC1.
* At this time, the dummy data is not transmitted.

■ Procedure for Retransmission of Remote Control Data

(1) Set the dummy data x'ff' in register SIBUF1.
(2) Set x'11' in register SIC1.
(3) Set the send data in the TXQUE1 register.

t1 is specified by the TB4 setting. Timer 4 cycle set to t1/2. t2 is specified by the OCR4 setting.
The range for setting t2 is from 0 up to 1/2 cycle of the remote control transfer clock.

The set output data is transmitted in the MSB first sequence as set in the remote control output data queue buffer register (TXQUE1: x'3FB0').
'LSB first' cannot be specified as the first bit for transfer.

Use Timer 4 divided by 2 as the clock source when using the remote control output function. If there is no other choice but to use a different clock, use a clock that is slower than fs/32. (Do not use fs/8 or fs/16 as the clock source.)

If the remote control output function is to be used, do not enable the start condition.

6-2-7 Serial Transfer Header

While performing a serial transfer, the Serial Header Function, built into the serial interface, detects sections where there is no serial clock signal.
Setting the SIF1 Header Function Flag, SI1RE Flag (SIC1: bit 2 of x'3F9E'), to '1' specifies the Serial Header Function.

(1) In response to overflow of Timer 3 and no signal sections of the Serial Control Register, Timer 3 counts while the SBT1 Pin is at a high-level (no signal sections between data blocks). Timer 3 is preset when a low-level is input to the SBT1 Pin. When Timer 3 overflows, the 3-bit counter (SBC) of Serial Interface 1 is cleared, and the pass/disable flag for the SIF1 clock input, SI1CKE (SIC1: bit 2 of x'3F9E'), is reset. Resetting the SI1CKE Flag causes the serial clock to be transferred inside the chip and starts the serial transfer (header).

(2) Serial Transfer Header Timing Diagram

1. Set an appropriate value of ts as the initial value in the TB3 Register, where Timer 3 overflows at time ts and t1<ts<t2.
2. When the SI1RE and SI1CKE Flags are set to '1' and operation begins, the serial clock input will be masked.
3. The clock is input while the serial clock is masked and Timer 3 begins counting after the 8th clock input. Then, if Timer 3 does not overflow before the next clock is input, the mask for the serial clock will be continued.
4. When Timer 3 overflows, the SI1RE Flag is set to '1', the SI1CKE Flag is set to '0', the serial clock is transferred inside the chip and serial transfer begins.
5. At the completion of the serial transfer, programmed instructions set both the SI1RE and SI1CKE Flags to '1' and overflow of Timer 3 is waited for again.

When using the Serial Transfer Header Function, set the SI1GATE Flag (bp7) of the Serial 1 Mode Register (SIM1: x'3F9C') to disable shifting after the serial transfer.

In the above sequence, the Serial Transfer Header Function and masking of the serial clock can be realized.

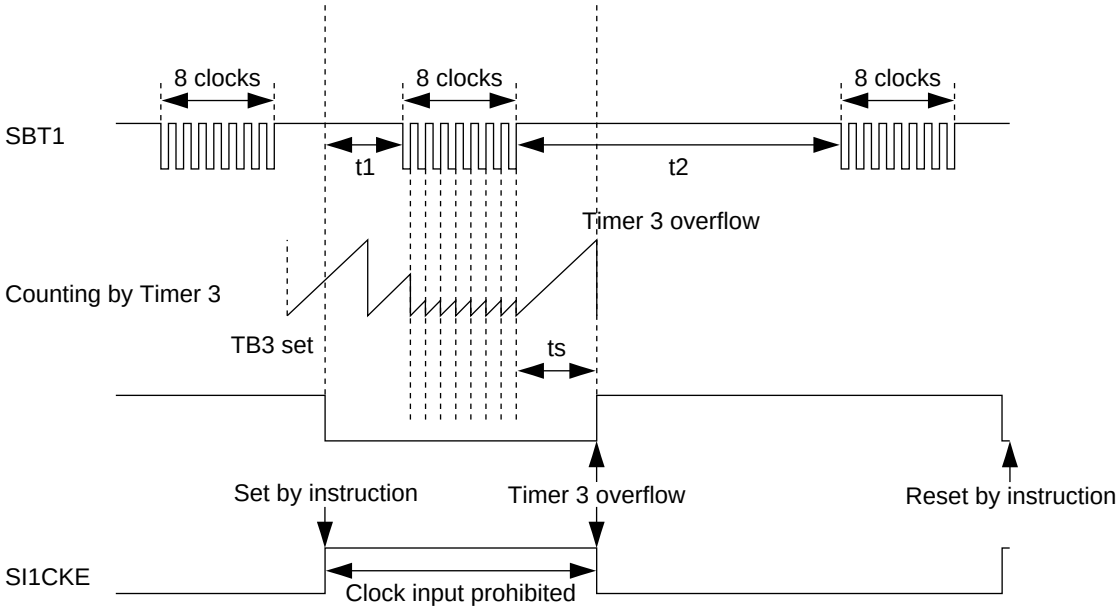


Figure 6-2-14 Serial Transfer Header Timing Diagram

6-3 Serial 2

Be aware that the clock control registers, instructions and serial interrupts differ from those used by Philips Corp.

6-3-1 Overview

The I²C Serial Interface has almost identical functions as the I²C (Inter IC) SIO serial interface of Philips Corp.

The I²C Serial Interface is constructed from 1 data line and 1 clock line. Data is transferred using an I²C bus.

6-3-1-1 I²C Bus

Figure 6-3-1-1 shows an example of a typical system that uses an I²C bus. A device that controls data transfer is called a master and a device controlled by a master is called a slave. In this example, the master is either of the 2 microcomputers and the serial memory and display are slaves.
The master generates timing signals to control the bus and controls data transmission and reception.
An important feature of the I²C bus is that even if 2 or more masters control an I²C bus at the same time, the result will be a multi-master bus where data is not destroyed. If several microcomputers start a transfer at approximately the same time, the master will be determined through arbitration. The master that wins arbitration can address the master that lost arbitration as a slave.

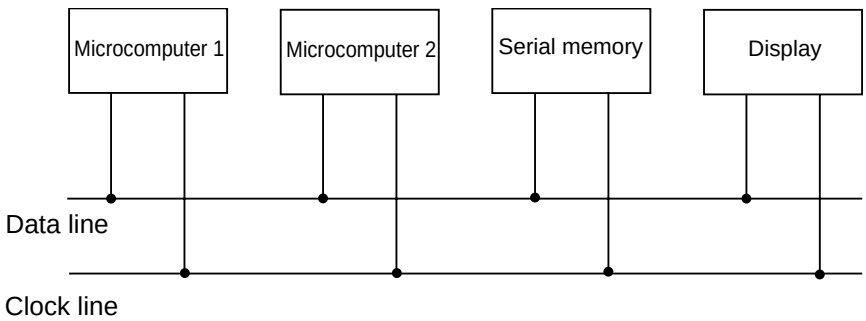


Figure 6-3-1-1 Typical Serial I/O Connection Example

A serial bus is constructed from 2 bi-directional lines, as shown in the serial bus configuration of Figure 6-3-1-2. Each bus line, SDA (data line) and SCK (clock line), are connected to VDD through pull-up resistor Rp. The output circuit for each device has an open-drain configuration and forms a wired AND configuration with the bus. Each device can be programmed through software to operate in either master or slave mode and as a transmitting device or receiving device.

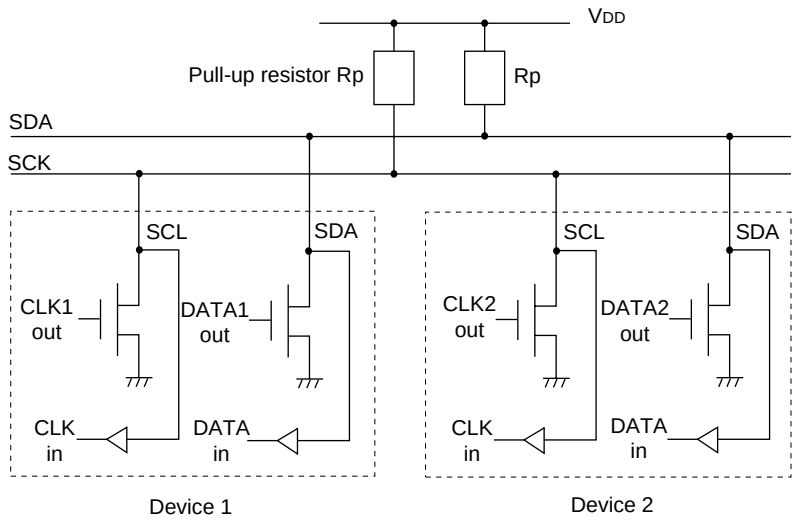


Figure 6-3-1-2 Connecting 2 Devices to the I^2C Serial Bus

6-3-1-2 Operating Modes

The I^2C Serial Interface has the following 4 operating modes.

1. Master transmission: Generates the clock for serial transfer (SCK) by itself and transmits serial data in synchronization with SCK.

2. Slave reception: Receives data in synchronization with SCL generated by the master.

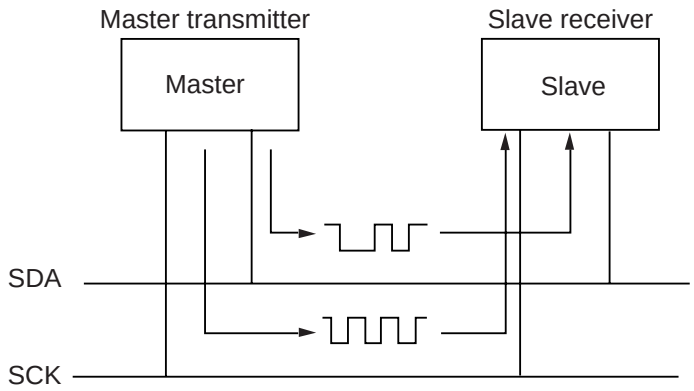


Figure 6-3-1-3 Master Transmission and Slave Reception

For further information regarding the data format, refer to section 6-3-1-4, "Data Format".

3. Master reception: After transmitting addressing format data of R/$\overline{W}$=1, the master enters the master receive mode. At this time, in synchronization with the self-generated SCK, the master receives data transmit from the slave.

4. Slave transmission: If the slave receives addressing format data of R/$\overline{W}$=1, the slave enters the slave transmit mode and transmits data in synchronization with SCK generated by the master.

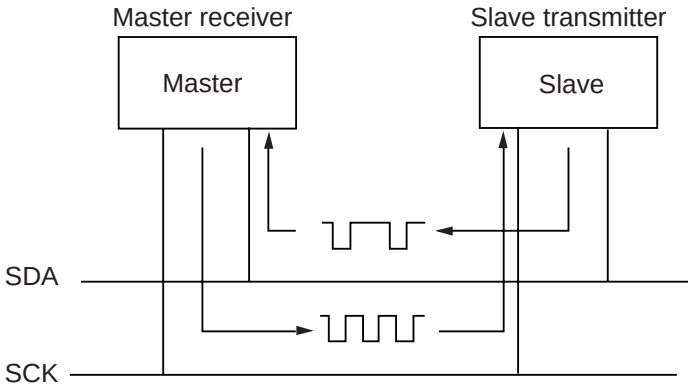


Figure 6-3-1-4 Master Reception and Slave Transmission

6-3-1-3 Data Transfer Sequence

Figure 6-3-1-5 shows the data transfer sequence of the I²C Serial Interface. The sequence is divided into the following states.

F (free)..... The bus is free (idle). Data transfer is not performed in this state. (The SDA and SCK lines are at high-levels.)

S (start) ..... With the SCK line at a high-level, data transfer starts (start condition) when the SDA line changes from a high to a low-level. At this time, the bus is occupied.

C (change) ..... During the interval when the clock is at a low-level, 1 bit of data from a transmitting device is sent along the SDA line. Therefore, the level of the SDA line may change during this interval.

D (data)..... During the interval when the clock is at a high-level, a receiving device receives bit data from the SDA line. So that it is not regarded as a start or stop signal, the level (bit status) of the SDA line must be constant during this interval.

P (stop) ..... With the SCK line at a high-level, data transfer stops (stop condition) when the SDA line changes from a low to a high-level. After the transfer stops, the bus will be free again.

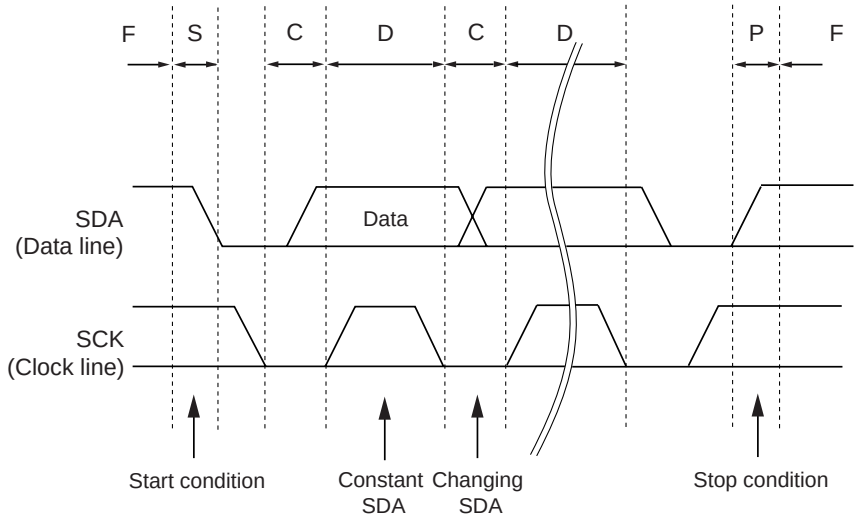


Figure 6-3-1-5 Bit Transfer Procedure on the I²C Serial Bus

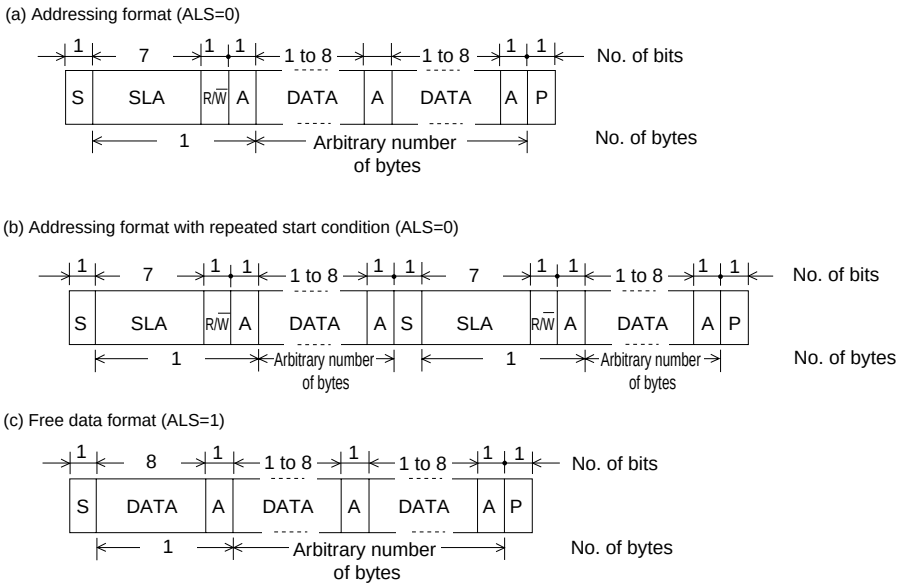
6-3-1-4 Data Format

Figure 6-3-1-6 shows three data formats that have been prepared for use with the I²C Serial Interface. The first byte (SLA and R/$\overline{W}$) that follows start signal (S) normally consists of 8 bits.

Use of an acknowledge bit (A) can be specified by setting the appropriate register.

In addressing formats (a) and (b), the SLA byte specifies the selected slave device. The direction of the next data transfer is determined by the LSB (R/$\overline{W}$ bit) of the SLA byte.
If R/$\overline{W}$=0, the master device writes data to the selected slave device. (Master transmission)
If R/$\overline{W}$=1, the master device reads data from the slave device. (Master reception)

In free data format (c), the transfer direction is constant while data is being transferred and the receiving device receives all of the data from the I²C bus.



S: start condition
SLA: slave address
R/$\overline{W}$: transfer direction bit
A: acknowledge bit
P: stop condition
DATA: data

Figure 6-3-1-6 I²C Serial Bus Transfer Format

6-3-1-5 Block Diagram

Figure 6-3-1-7 shows a block diagram of the I²C Serial Interface.

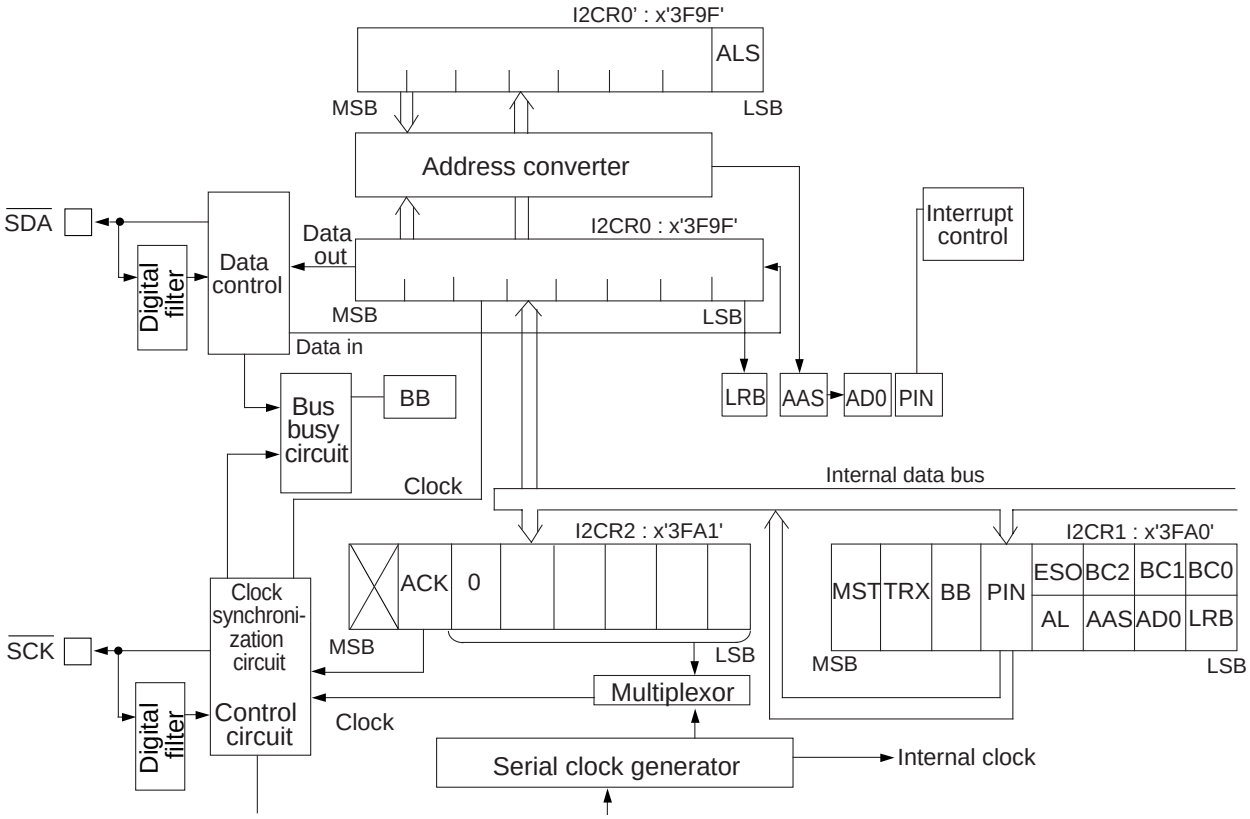


Figure 6-3-1-7 I²C Interface Block Diagram

6-3-2 I²C Serial Interface Control Registers

Table 6-3-2-1 shows the I²C Serial Interface control registers.

Table 6-3-2-1 I²C Serial Interface Control Registers

Register Name	Address	R/W	Function
I2CR0	x'3F9F'	R/W	Data shift register
I2CR0'	x'3F9F'	W	Address register
I2CR1	x'3FA0'	R/W	Status register
I2CR2	x'3FA1'	W	Clock control register
I2CCN	x'3FA2'	R/W	I²C system control register
I2CDIR	x'3FA3'	R/W	I²C direction control register

With the exception of the PIN (Pending Interrupt Not) bit of I2CR1, all bits in the registers are set to '0' at reset. After reset, the PIN bit is set to '1'.

6-3-2-1 Data Shift Register (I^2CR0)

I2CR0: x'3F9F' (Both reading and writing are possible when the ESO flag is 1 in bit 3 of register I2CR1 x'3FA0'.)

Bit	7	6	5	4	3	2	1	0
Flag name	I2CR0 7	I2CR0 6	I2CR0 5	I2CR0 4	I2CR0 3	I2CR0 2	I2CR0 1	I2CR0 0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Permitted value	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

It is possible to both read and write the data shift register I2CR0 when the ESO flag in bit 3 of register I2CR1 (x'3FA0') is '1'.
This register performs data format conversion between serial data and parallel data. After the data to be transmitted has been loaded in I2CR0, it is shifted to pin SDA starting from the MSB. The data received from pin SDA is shifted into I2CR0 starting from the LSB. In the acknowledgement mode, the acknowledge bit ACK does not get entered into the data shift register I2CR0 but gets entered into bit 0, LRB (Last Received Bit) of the status register I2CR1 (x'3FA0'). An interrupt is generated when one byte of data has been transmitted or received.

To properly operate the I^2C interface, programmer must implement a "wait" loop with the following code. The "wait" begins when the setting of transfer data is complete and lasts until the start of the transfer.

```
        ~
   transfer setting

   mov     x'10',  d0   \
loop                     |
        nop              | required
        dec     d0       |
        bne     loop    /
i2cr1   x'xx'
   start of transfer
        ~
```

6-3-2-2 Address Register (I^2CR0')

I2CR0': x'3F9F' (Writing is possible when the ESO flag is 0 in bit 3 of register I2CR1 x'3FA0'.)

Bit	7	6	5	4	3	2	1	0
Flag name	ADRL6	ADRL5	ADRL4	ADRL3	ADRL2	ADRL1	ADRL0	ALS
Read/Write	W	W	W	W	W	W	W	W
After reset	0	0	0	0	0	0	0	0
Permitted value	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

The address register can be written when the ESO flag is '0' in bit 3 of register I2CR1 (x'3FA0').
This register consists of a 7-bit address latch (bits 1 to 7) and the ALS (Always Selected) flag (bit 0). The address latch stores the slave address allocated to the device and the ALS flag enables/disables the address recognition mode.

ALS=1 Slave addresses are not recognized. I^2C Serial Interface is used with the free data format. Data is received from the serial data bus.

ALS=0 Slave addresses are recognized. I^2C Serial Interface is used with the addressing format. The interface only operates when the first received data is the slave address of the device or a general address (x'00'). In this mode, the LSB of the first received data determines the direction of transmission. Read and/or write instructions are executed according to the transmission direction.

6-3-2-3 Status Register (I²CR1)

I2CR1: x'3FA0'

Bit	7	6	5	4	3	2	1	0
Flag name	MST	TRX	BB	PIN	ESO	BC2	BC1	BC0
					AL	AAS	AD0	LRB
Read/Write	R/W	R/W	R/W	R/W	W	W	W	W
					R	R	R	R
After reset	0	0	0	1	0	0	0	0
Permitted value	0/1	0/1	0/1	0/1	0/1	0/1	0/1	0/1

The status register I2CR1 can be written when the PLAINH flag (bit 0) is '1' in register I2CCN (x'3FA2').
The status word in this register is shown in Fig. 6-3-2-1. I2CR1 contains all the information pertaining to the state of the I²C Serial Interface. The configuration of the lower 4 bits of I2CR1 differs during read and write operations. ESO, BC2, BC1, and BC0 are write-only control bits. AL, AAS, AD0 and LRB are read-only status bits. The upper 4 bits, MST, TRX, BB and PIN, are both read and write accessible.

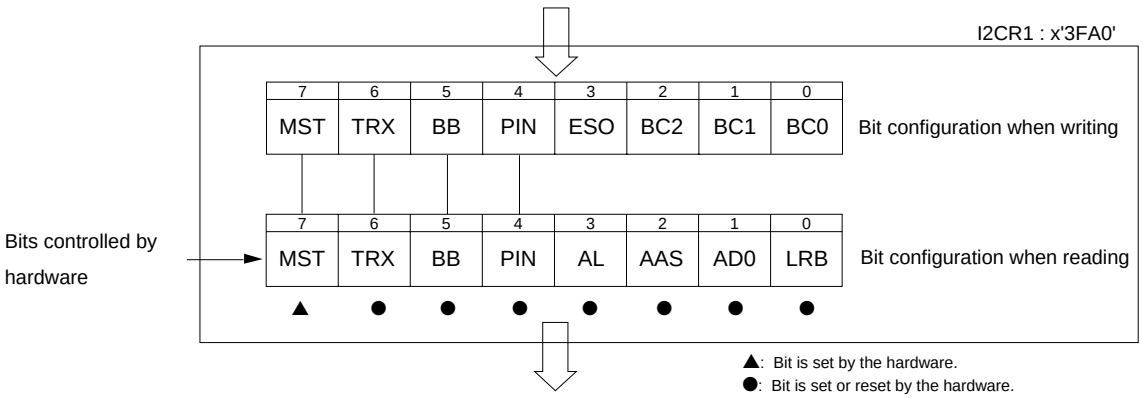


Figure 6-3-2-1 Status Register I2CR1 Configuration

Functions of each bit in the Status Register are described below.

■ MST: Master Bit

MST=1 In this case, the I²C serial interface is in the master mode and the clock SCK for serial data transmission and reception is generated.

MST=0 In this case, the I²C serial interface is in the slave mode and the clock SCK is received from the master device.

Condition under which MST is reset to '0'

MST is reset to '0' by the hardware when the master device generates the stop condition either when the master device loses arbitration, or after the data transfer has been completed.

■ TRX: Transmit Bit

When TRX is '1', the I²C Serial Interface is in transmit mode and synchronized to the pulses on clock line SCK, data is shifted out of Data Shift Register I2CR0 into data line SDA. When TRX is '0', the I²C Serial Interface is in receive mode and synchronized to the pulses on clock line SCK, data from data line SCK is shifted into Data Shift Register I2CR0. If arbitration is lost, the TRX bit is reset to '0' by the hardware. When a data transfer is complete, the device that is in master mode generates the stop condition. The TRX bit for this device in master mode is reset by the hardware.
When serial transfer data is in the addressing mode, the TRX bit is set by hardware based on the state of Read/Write Bit (R/W). [☞ 6-3-1-4, "Data Format"]
Operating mode settings based on MST and TRX of the status word are summarized in Table 6-3-2-2.

Table 6-3-2-2 Operating Mode Setting Based on MST and TRX

MST	TRX	Operating Mode
0	0	Slave reception
1	0	Master reception
0	1	Slave transmission
1	1	Master transmission

■ BB: Bus Occupied Bit

BB indicates the status of the serial bus. When BB is '0', the bus is free (idle). If BB is '1', the busy is occupied (busy). In the slave mode, a bus busy circuit generates the start condition to set BB to '1'. After the stop condition is generated, BB is reset to '0' when the internal serial clock is at a low-level. In the master mode, BB is controlled by the software.
If '1' is written to the MST, TRX and BB bits of I2CR1, the start condition will be generated.
If '1' is written to the MST and TRX bits and '0' to the BB bit of I2CR1, the stop condition will be generated.
The Bus Occupied Bit (BB) is set to '1' after reset is released, but 7 µs later is reset to '0' by the hardware. Also, BB is set to '1' when data is set in Clock Control Register I2CR2, but is reset to '0' after one-half cycle of the specified clock frequency.

When applying the start condition, verify that the Bus Occupied Bit (BB) is set to '0'.

■ PIN: Pending Interrupt Not Bit

The serial 2 interrupt request flag (bp0 R/W of SIF2ICR: x'3FFB') is set to '1' when an I²C interrupt is generated.

When an I²C interrupt is generated, PIN is reset to '0'. While PIN is '0', since clock line SCK is held at a low-level, the clock pulse is invalid.

PIN is set to '0' in the following cases.

1. When a 1-byte transfer is completed (including the case where arbitration is lost)
2. When based on the addressing format, a slave address is recognized or a general address (x'00') is received (ALS=0)
3. When data is received in the free data format (ALS=1)

PIN is set to '1' in the following cases.

1. When Data Shift Register I2CR0 is read from or written to
2. When the software writes '1' to the PIN bit

When manipulating the interrupt request flag (xxxIR), it is necessary to set beforehand the IR write enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, reset IRWE to '0' after the manipulation.

■ ESO: Enable Serial Output

When ESO is '1', the I²C Serial Interface can be used to transmit and receive data. At this time, both ports SCK and SDA change over to I²C. (Direction Control Register I2CDIR must also be set.)
When ESO is '0', the I²C Serial Interface cannot be used and SDA and SCL enter high impedance states. Data can be written to Address Register I2CR0' when ESO is '0'. While ESO is '0', the PIN and AL bits are set to '1' by the hardware. When ESO is '0', the contents of Data Shift Register I2CR0 are maintained.

■ BC2, BC1, BC0: Bit Count

The number of bits to be transmit or received are set in binary code.

Table 6-3-2-3 Number of Bits for Transmission or Reception

BC2	BC1	BC0	No. of data bits when no acknowledge bit	No. of data bits with acknowledge bit
0	0	1	1	2
0	1	0	2	3
0	1	1	3	4
1	0	0	4	5
1	0	1	5	6
1	1	0	6	7
1	1	1	7	8
0	0	0	8	9

The start condition resets the bit count to '000'. Therefore, not counting the acknowledge bit, the first byte to be transmit or received always consists of 8 bits. After all data bits are transmit or received, the bit count is again reset to '000'.

■ AL: Arbitration Lost (Transfer Lost)

AL is set to '1' when in the master transmit mode and the I2C Serial Interface loses arbitration. If BB is already '1' or even if '1' is written to BB, AL will be set to '1'. If arbitration is lost, PIN is reset to '0' when the data transmission is complete and I2CR1 enters the slave reception mode (MST=0, TRX=0).
Writing to I2CR0, reading from I2CR0 and writing to I2CR1 cause AL to be reset to '0'.

■ AAS: Slave Address Recognition

AAS is set to '1' when the address comparator recognizes its own slave address as a general address (x'00'). AAS is also set to '1' if the first data is received in the free data format (ALS=1). Writing to I2CR0 and reading from I2CR0 cause AAS to be reset to '0'.

■ AD0: Zero Address

AD0 is set to '1' when the address comparator recognizes a general address (x'00').
AD0 is reset to '0' when the start condition or stop condition is detected.

■ LRB: Last Receive Bit

During acknowledgement mode, the acknowledge bit generated by the receiver is input to the LRB Status Register I2CR1 of the transmitter. At this time, a LRB value of '0' verifies that the transmission was received. If an acknowledge bit is not included in the transfer data, the last bit of the transmission or reception byte is input as LRB.

6-3-2-4 Clock Control Register (I²CR2)

I2CR2: x'3FA1'

Bit	7	6	5	4	3	2	1	0
Flag name	-	ACK	0	I2CR24	I2CR23	I2CR22	I2CR21	I2CR20
Read/Write	-	W	W	W	W	W	W	W
After reset	0	0	0	0	0	0	0	0
Permitted value	0	0/1	0	0/1	0/1	0/1	0/1	0/1

Clock Control Register I2CR2 is an 8-bit read-only register. Bits 4 to 0 control the clock frequency of the I²C Serial Interface. The clock frequencies shown in Table 6-3-2-4 can be set for each oscillation frequency. Always set bit 5 to '0' to indicate that the HIGH/LOW ratio of SCK is 1:1. Bit 6 (ACK) specifies the acknowledgement mode. Bit 7 is not used. Set bit 7 to '0'.

Table 6-3-2-4 Clock Pulse Frequency Settings (units: kHz)

CLKSEL	I2CR24 to 0	Frequency Division Ratio	Transfer Frequency (kHz)				SCK High to Low Ratio
			14.32MHz	16.00MHz	17.73MHz	20.00MHz	
1	x'09'	72	99.44				1:1
0	x'10'	80	89.50	100.00			
1	x'0A'	84	85.24	95.24			
0	x'11'	96	74.58	83.33	92.34		
1	x'0B'	102	70.20	78.43	86.91	98.04	
0	x'12'	112	63.93	71.43	79.15	89.29	
1	x'0C'	120	59.67	66.67	73.88	83.33	
0	x'13'	128	55.94	62.50	69.26	78.13	
1	x'0D'	144	49.72	55.56	61.56	69.44	
0	x'14'	160	44.75	50.00	55.41	62.50	
1	x'0E'	168	42.62	47.62	52.77	59.52	
0	x'15'	192	37.29	41.67	46.17	52.08	
0	x'16'	224	31.96	35.71	39.58	44.64	
0	x'17'	256	27.97	31.25	34.63	39.06	
0	x'18'	320	22.38	25.00	27.70	31.25	

6-3-2-5 I²C System Control Register (I²CCN)

I2CCN: x'3FA2'

Bit	7	6	5	4	3	2	1	0
Flag name	STA ERR	ERR EN	-	SCK ODC	SDA ODC	-	-	PLA INH
Read/Write	R/W	R/W	-	R/W	R/W	-	-	R/W
After reset	0	0	0	0	1	0	0	0
Permitted value	0/1	0/1	0	0/1	0/1	0	0	0/1

When using the I²C Serial Interface be sure to set the I2CCN Register first.

The I²C system control register I2CCN is an 8-bit register that can be both read from and written to.

■ STAERR
■ ERREN

If ERREN is set in advance to '1', STAERR will be set to '1' when the start condition cannot be generated correctly.

Start condition errors are processed by the following procedure.

1. Enable ERREN (set to 1) in advance.
2. Read STAERR after the start condition is set.
   If STAERR is '1' (start error), perform the return processing.

Start error return processing is performed by the following procedure.

1. Set the stop condition.
2. Set STAERR to '0'.
   The stop condition is generated when the clock line becomes free. If the data transfer is to be attempted again, reset the transfer data.

■ SDAODC, SCKODC

SDAODC and SCKODC control ESO, SDAOE, SCLOE as well as the SCK and SDA pins.
If SDAODC is '1', the SDA pin is configured to have an n-channel, open-drain output. If SCAODC is '0' the SDA pin has a push-pull output. If SCKODC is '1', the SCK pin is configured to have an n-channel, open-drain output. If SCKODC is '0' the SCK pin has a push-pull output.

■ PLAINH

PLAINH is the inhibit signal for I²C Status Register ICR1.
To prevent erroneous operation, the Status Register is disabled immediately after reset.

6-3-2-6 I²C Direction Control Register (I²CDIR)

I2CDIR: x'3FA3'

Bit	7	6	5	4	3	2	1	0
Flag name	CLK SEL	-	-	-	-	-	SCK OE	SDA OE
Read/Write	R/W	-	-	-	-	-	R/W	R/W
After reset	0	0	0	0	0	0	0	0
Permitted value	0/1	0	0	0	0	0	0/1	0/1

I2CDIR controls ESO, SCKODC and SDAODC as well as the SDA and SCK pins. When SDAOE and SCKOE are '1', the SDA and SCK pins are enabled for I/O. CLKSEL sets whether or not the clock pulse frequency is divided by 3. If the clock pulse frequency is to be divided by 3, set CLKSEL to '1'. Refer to Table 6-3-2-4 for clock pulse frequency setting values.

6-3-3 Transmission Control

6-3-3-1 Recognition Sequence in Acknowledgement Mode

Figure 6-3-3-1 shows the recognition sequence for the I²C Serial Interface.

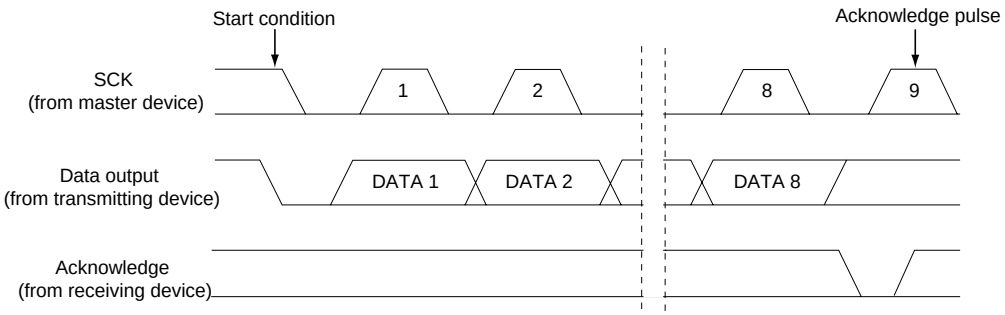


Figure 6-3-3-1 Recognition Sequence for I²C Serial Interface

Recognition sequences for each communication mode are described below.

■Master Transmission

When the ACK flag (bit 6) is '1' in register I2CR2 (x'3FA1'), the master transmitting device places an acknowledge pulse on the SCK line at the end of the transmitted data. During the period of this acknowledge pulse, the SDA pin of the master transmitting device goes into the input mode, and the LRB flag (bit 0) becomes '0' in register I2CR1 (x'3FA0') of the master transmitting device when the receiving side generates the acknowledge pulse (SDA line goes to the "L" level).
If the master transmitting device cannot receive the acknowledge pulse, the SDA line remains in the "H" level, and the LRB flag (bit 0) of register I2CR1 (x'3FA0') gets set to '1'.

■Master Reception

When the ACK flag (bit 6) of register I2CR2 (x'3FA1') is '1', the master receiving device places the acknowledge pulse on the SCK line after receiving one byte of data. While this acknowledge pulse is being generated, the SDA pin of the master receiver device is in output mode. A low-level reception confirmation signal is generated and output on the SDA line.

■Slave Transmission

Except for reception of SCL acknowledge pulses generated by the master device, slave transmission operates in the same manner as a master transmitter device. When the acknowledge pulse generated by the master device is received, the LRB flag (bit 0) of register I2CR1 (x'3FA0') becomes '0'.

■Slave Reception

Except for reception of acknowledge pulses generated by the master device, slave reception operates in the same manner as a master receiver device. While the acknowledge pulse is being generated, the slave receiver device generates a low-level on the SDA line.

For example, in the case of master transmission followed by master reception, the slave device that received the first 1 byte of transfer data (slave address) generates an acknowledge signal to inform the master that data has been received. From the transfer of the 2nd byte on, the master generates an acknowledge signal for the data transmit by the slave. The slave receives this acknowledge signal.

When a '0' is loaded into the ACK flag (bit 6) of the clock control register I2CR2 (x'3FA1'), the I²C serial interface goes out of the acknowledgement mode. In this case, the acknowledge pulse is not generated in the case of a master device, and the "L" level for confirming reception is not generated in the receiving side.

6-3-3-2 Arbitration Sequence

If two or more master transmitting devices begin transferring data to the same device at approximately the same time, arbitration will be performed to mediate between the two competing devices. If two or more transmitting devices output both high and low-level signals to the SDA line, the SDA line will be at a low-level. This is due to the open-drain output of the SDA Pin of the I^2C Interface.
The transmitting device that output the "H" level senses that there is a competing transmitting device by detecting that the SDA line did not go to the "H" level, and changes over to the reception mode by setting a '1' in the AL flag (bit 3) of register I2CR1 (x'3FA0').
Figure 6-3-3-2 shows the arbitration sequence between two devices. If two or more devices transmit the same first byte, arbitration is not performed. However, arbitration will be performed if successive bytes of data are different.

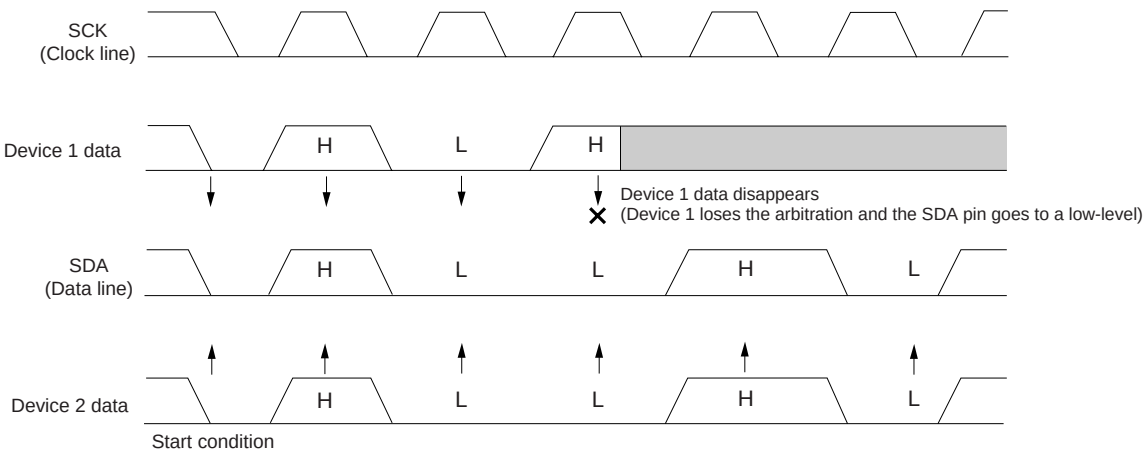


Figure 6-3-3-2 Arbitration Sequence

6-3-3-3 Clock Synchronization Characteristics

In a normal sequence, a single master device generates the clock. In an arbitration sequence, 2 or more master devices generate clocks. In this case, the clock signals generated by each master device must be synchronized so that output data can be shared. Because the clock line has a wired AND configuration, if one master device outputs a low-level on the clock line, other master devices will not be able to change the clock line to a high-level. Therefore, the master device having the longest low-level duration will hold the clock line at a low-level. Other master devices whose low-level clock is completed within this interval will wait until the clock line is released before generating their high-level clock signals.
For multi-master transfers, the clock pulse is determined by the master device having the slowest low-level clock and the master device having the fastest high-level clock. In this manner, since fast devices are synchronized to the speed of slow devices, the timing necessary for processing will be guaranteed even if a combination of slow and fast devices are used.

Figure 6-3-3-3 shows a detailed view of clock synchronization characteristics.

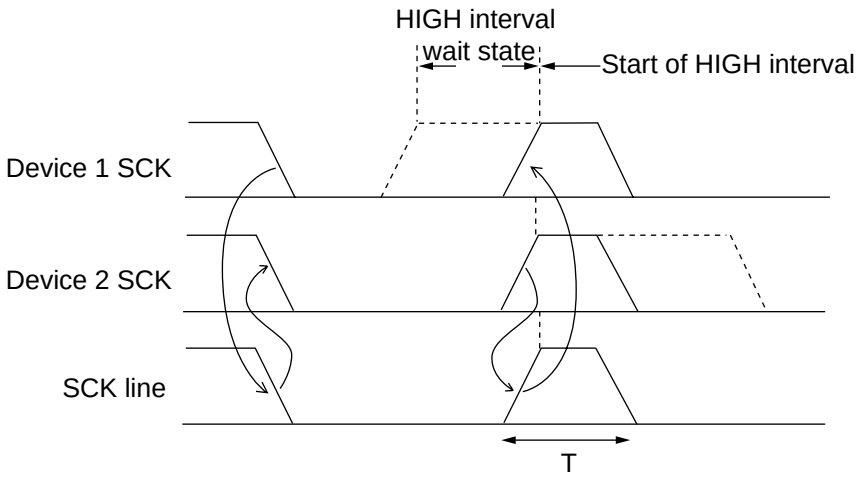


Figure 6-3-3-3 Synchronization of Two Clocks

In an actual system configured with an I²C bus, because it is common for each device to operate non-synchronously, the clock synchronization characteristics during multi-master operation can cause clock pulse width, T, to become so narrow it cannot be recognized. This may hinder the data transfer. If multiple masters are used, do not use complex software programs and take measures to avoid simultaneous data transmission as much as possible.

6-4 I²C Serial Transfer Settings

6-4-1 Master Transmission Procedure

The master transmission procedure is given below.

(1) Set pins P27 and P26 as outputs by setting bits 7 and 6 to '1' in register P2DIR (x'3F30').

(2) Set SCK and SDA as outputs by setting bits 1 and 0 to '1' in register I2CDIR (x'3FA3').

(3) Set the PLAINH flag (bit 0) to '1' in register I2CCN (x'3FA2'). Also, set the output formats of the SCK and SDA pins using bits 4 and 3.

(4) Select the acknowledgement mode using the ACK flag (bit 6) of register I2CR2 (x'3FA1'). Specify the clock frequency using bits 4 to 0.

(5) Set the PIN flag (bit 4) and the ESO flag (bit 3) of register I2CR1 (x'3FA0').

(6) Set the send data in register I2CR0 (x'3F9F'). During the first transmission, specify a 7-bit slave address and data transfer direction ('0' in the case of master transmission).

(7) When using the Serial 2 interrupt, set the SIF2ICR register (x'3FFB'). Set the interrupt level with flags SI2LV0 and SI2LV1 (bits 7 and 6), set interrupt enable using the SI2IE flag (bit 1), and set the SI2IR flag (bit 0) to '0'.

(8) Set bits 7 and 6 to '1' in register P2SEL (x'3F39') to set pins P26 and P27 as dedicated outputs of the SCK and SDA signals.

(9) Wait until the hardware accepts the transfer settings using the following instructions:

```
        mov   x'10',d0
loop
        nop
        dec   d0
        bne   loop
```

(10) Set register I2CR1 (x'3FA0').
Set the number of bits transmitted using flags BC2 to 0 (bits 2 to 0). Set the MST flag, TRX flag, and BB flag (bits 7, 6, and 5) to '1' to set the master transmission mode. Also set the PIN flag and the ESO flag (bits 4 and 3) to '1'. Master transmission is started at the end of this setting.

(11) When the transmission of one byte is completed, the SI2IR flag (bit 0) changes to '1' in register SIF2ICR (x'3FFB') and generates an interrupt.

(12) If data is to be transmitted again, set the data in register I2CR0 (x'3F9F'). Further, transmission start condition is read/write of register I2CR0 and write of register I2CR1.

To end data transmission, set b'110' in bits 7, 6, and 5 of register I2CR1. This generates a stop condition and releases the ownership of the bus.

6-4-2 Master Reception Procedure

The master reception procedure is given below.

(1) Set pins P27 and P26 as outputs by setting bits 7 and 6 to '1' in register P2DIR (x'3F30').

(2) Set SCK and SDA as outputs by setting bits 1 and 0 to '1' in register I2CDIR (x'3FA3').

(3) Set the PLAINH flag (bit 0) to '1' in register I2CCN (x'3FA2'). Also, set the output formats of the SCK and SDA pins using bits 4 and 3.

(4) Select the acknowledgement mode using the ACK flag (bit 6) of register I2CR2 (x'3FA1'). Specify the clock frequency using bits 4 to 0.

(5) When using the Serial 2 interrupt, set the SIF2ICR register (x'3FFB'). Set the interrupt level with flags SI2LV0 and SI2LV1 (bits 7 and 6), set interrupt enable using the SI2IE flag (bit 1), and set the SI2IR flag (bit 0) to '0'.

(6) Set bits 7 and 6 to '1' in register P2SEL (x'3F39') to set pins P26 and P27 as dedicated outputs of SCK and SDA signals.

(7) Wait until the hardware accepts the transfer settings using the following instructions:

```
        mov   x'10',d0
loop
        nop
        dec   d0
        bne   loop
```

(8) Set register I2CR1 (x'3FA0').
Set the number of bits transferred using flags BC2 to 0 (bits 2 to 0). Set the MST flag, TRX flag, and BB flag (bits 7, 6, and 5) to b'101' to set the master reception mode. Also set the PIN flag and the ESO flag (bits 4 and 3) to '1'. The clock is output at the SCK pin upon completion of the above setting.

(9) When the reception of one byte is completed, the SI2IR flag (bit 0) changes to '1' in register SIF2ICR (x'3FFB') and generates an interrupt.

(10) If data is to be received again, either read from or write to register I2CR0, or write to register I2CR1.

To end data reception, set b'110' in bits 7, 6, and 5 of register I2CR1. This generates a stop condition and releases ownership of the bus.

6-4-3 Slave Transmission Procedure

The slave transmission procedure is given below.

(1) Set pins P27 and P26 as outputs by setting bits 7 and 6 to '1' in register P2DIR (x'3F30').

(2) Set SCK and SDA as outputs by setting bits 1 and 0 to '1' in register I2CDIR (x'3FA3').

(3) Set the PLAINH flag (bit 0) to '1' in register I2CCN (x'3FA2'). Also, set the output formats of the SCK and SDA pins using bits 4 and 3.

(4) Select the acknowledgement mode using the ACK flag (bit 6) of register I2CR2 (x'3FA1'). Specify the clock frequency using bits 4 to 0.

(5) Set the address of the slave device in bits 7 to 1 of register I2CCR0 (x'3F9F'). Also, because the transmission is a slave transmission, set the data transfer direction in bit 0 to '1'.

(6) Set the PIN flag (bit 4) and the ESO flag (bit 3) of register I2CR1 (x'3FA0').

(7) Set the send data in register I2CR0.

(8) When using the Serial 2 interrupt, set the SIF2ICR register (x'3FFB'). Set the interrupt level with flags SI2LV0 and SI2LV1 (bits 7 and 6), set interrupt enable using the SI2IE flag (bit 1), and set the SI2IR flag (bit 0) to '0'.

(9) Set bits 7 and 6 to '1' in register P2SEL (x'3F39') to set pins P26 and P27 as dedicated outputs of SCK and SDA signals.

(10) Wait until the hardware accepts the transfer settings using the following instructions:

```
        mov   x'10',d0
loop
        nop
        dec   d0
        bne   loop
```

(11) Set register I2CR1.
Set the number of bits transmitted using flags BC2 to 0 (bits 2 to 0). Set the MST flag, TRX flag, and BB flag (bits 7, 6, and 5) to b'011' to set the slave transmission mode. Also set the PIN flag and the ESO flag (bits 4 and 3) to '1'. Slave transmission is started after this setting when the clock is transmitted by the master device.

(12) When the transmission of one byte is completed, the SI2IR flag (bit 0) changes to '1' in register SIF2ICR (x'3FFB') and generates an interrupt.

(13) If data is to be transmitted again, set the data in register I2CR0. The transmission is started when the master sends the clock.

To end data transmission, make the master generate a stop condition.

6-4-4 Slave Reception Procedure

The slave reception procedure is given below.

(1) Set pins P27 and P26 as outputs by setting bits 7 and 6 to '1' in register P2DIR (x'3F30').

(2) Set SCK and SDA as outputs by setting bits 1 and 0 to '1' in register I2CDIR (x'3FA3').

(3) Set the PLAINH flag (bit 0) to '1' in register I2CCN (x'3FA2'). Also, set the output formats of the SCK and SDA pins using bits 4 and 3.

(4) Select the acknowledgement mode using the ACK flag (bit 6) of register I2CR2 (x'3FA1'). Specify the clock frequency using bits 4 to 0.

(5) Set the slave address in bits 7 to 1 of register I2CR0 (x'3F9F'). Also, since the reception is a slave reception, set the data transfer direction of bit 0 to '0'.

(6) When using the Serial 2 interrupt, set the SIF2ICR register (x'3FFB'). Set the interrupt level with flags SI2LV0 and SI2LV1 (bits 7 and 6), set interrupt enable using the SI2IE flag (bit 1), and set the SSI2IR flag (bit 0) to '0'.

(7) Set bits 7 and 6 to '1' in register P2SEL (x'3F39') to set pins P26 and P27 as dedicated outputs of the SCK and SDA signals.

(8) Wait until the hardware accepts the transfer settings using the following instructions:

```
        mov   x'10',d0
loop
        nop
        dec   d0
        bne   loop
```

(9) Set register I2CR1 (x'3FA0'). Set the slave reception mode by setting b'001' in the MST flag, TRX flag, and BB flag (bits 7, 6, and 5). Also set the PIN flag and the ESO flag (bits 4 and 3) to '1'. Data reception is started hereafter when the clock and data are sent by the master.

(10) When the reception of one byte is completed, the SI2IR flag (bit 0) changes to '1' in register SIF2ICR (x'3FFB') and generates an interrupt.

(11) If data is to be received again, either read from or write to register I2CR0 or write to register I2CR1.
Thereafter, reception is started when the clock and data are transmitted by the master.

To end data reception, make the master generate a stop condition.

Chapter 7 A/D Converter and D/A Converter

7

7-1 A/D Converter

7-1-1 A/D Converter Configuration

This IC contains a built-in 8-bit A/D converter with 12 analog inputs.
All 12 analog input pins function as both I/O Port 8 (P8: x'3F29', R) and as I/O Port C (PC: lower 4 bits of x'3F2D', R).

The A/D converter consists of the following components.

(1)A/D Conversion Control Unit
(2) A/D Buffer
(3) Interrupt Request Unit

(1) A/D Conversion Control Unit
The A/D conversion control unit specifies the input channel for A/D conversion and controls operation of the A/D conversion.

(2) A/D Buffer
The A/D buffer stores results of the A/D conversion. Data values are maintained until the next results are stored.

(3) Interrupt Request Unit
When A/D conversion (an interrupt source) is completed, the interrupt request unit sets the Interrupt Request Flag (ADIR) of the A/D Interrupt Control Register (ADICR: x'3FFC', R/W). If interrupts have been enabled, the interrupt will be processed.

It is possible to select either the end of A/D conversion or the basic period of PWM14 as the A/D interrupt source. When selecting end of A/D conversion, clear PWM14S (bp6 (R/W) of PWM14H: x'3F6F').

7-1-2 A/D Conversion Operation and Register Configuration

(1) To perform A/D conversion, first, set flags ADSEL0 to ADSEL3 (bp0 to bp3) of A/D Mode Register (ADM: x'3F6D', R/W) to select the analog pin input. Set the direction control for the selected input pin (P8DIR: x'3F34' or PCDIR: 3F37h, bp0 to bp7) to "input" (reset to '0').

(2) Set the A/D Operate/Stop Flag (ADEN: bp4 of ADM, R/W) to "operate" (set to '1').

(3) Set the A/D Conversion Start Flag (ADCONV: bp5 of ADM, R/W) to "start" (set to '1') to begin A/D conversion at the specified pin. The time required for conversion is approximately 7.8 µs (when running at 14.32 MHz). While the A/D conversion is in progress, A/D Conversion Operation Flag (ADBUSY: bp6 of ADM) is '1'. This flag is reset to '0' when the conversion is complete. After the conversion begins, ADCONV is automatically reset. Set the A/D Conversion Start Flag to "start" when the A/D Operate/Stop Flag is '1'.

(4) After A/D conversion is completed, conversion data can be obtained by reading the A/D Buffer (ADBUF: x'3F6C', R/W).

When carrying out A/D conversion from the A/D power supply OFF state (ADEN flag = '0'), make sure to start the A/D conversion (ADCONV flag = '1') only after switching ON the A/D power supply (ADEN flag = '1'). Setting both flags simultaneously to '1' may cause the conversion to be incorrect.

7-1-3 Interrupt Processing Settings

Perform the following procedure to enable A/D interrupts.

(1) Clear the A/D interrupt source select flag PWM14S (PWM14H: bp6 R/W of x'3F6F') and set the A/D conversion end interrupt.

(2) Set the interrupt priority level with ADLV0 and ADLV1 flags (bp6 and bp7) of the A/D Interrupt Control Register (ADICR: x'3FFC').

(3) Clear the A/D Interrupt Request Flag (ADIR: bp0) of the A/D Interrupt Control Register (ADICR). (This setting must be made when ADEN='1'.)

(4) Set the A/D Interrupt Enable Flag (ADIE: bp1) of the A/D Interrupt Control Register (ADICR).
If A/D conversion is operated after performing the above settings, the A/D Interrupt Request Flag ADIR (ADICR: x'3FFC', bp0, R/W) will be set to '1' upon completion of the A/D conversion. Also, do not set the A/D Operate/Stop Flag to 'stop' (do not set to '0') during an A/D interrupt request.

When manipulating the interrupt request flag (xxxIR), it is necessary to set beforehand the IR write enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, reset IRWE to '0' after the manipulation.

7-1-4 Warnings when Using the A/D Converter

In general, one must be careful of the following when using a sample-and-hold type of A/D converter. Since capacitors in the sample-and-hold circuit are charged during A/D conversion, an excessive current will flow in the A/D input pins. Therefore, it may not be possible to guarantee precision during A/D conversion.

<Counter Measures>

In order to guarantee precision of the A/D conversion, the following items must be implemented when using a microcomputer.

1. The impedance of the analog signal to be converted should be 28 kΩ or less.
2. If the impedance of the analog signal cannot be 28 kΩ or less, connect a 1000 pF or greater capacitor to limit voltage fluctuation at the A/D input pin.
3. To prevent fluctuations in the voltage of the power supply during A/D conversion, do not change the output level of the microcomputer from HIGH to LOW or from LOW to HIGH. Also, do not switch peripheral load circuits ON or OFF.

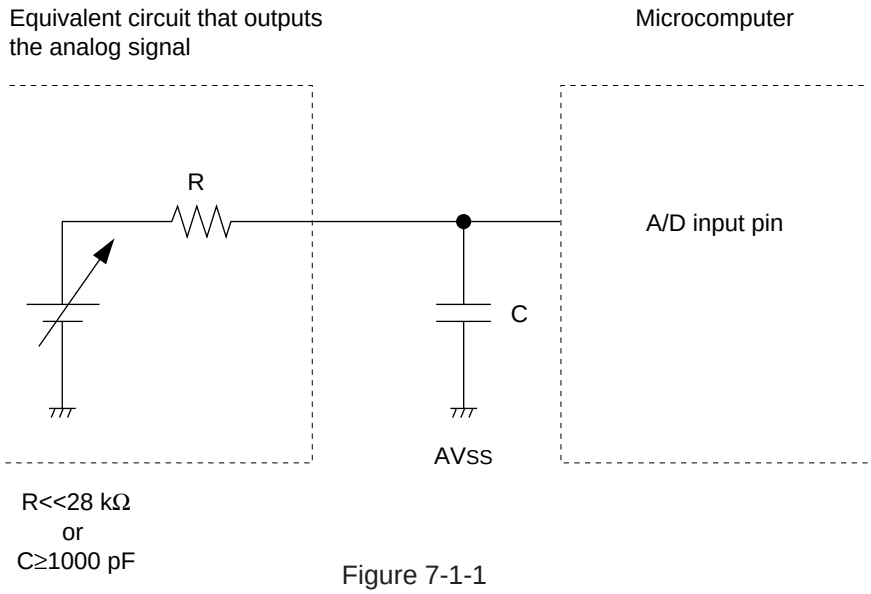


Figure 7-1-1

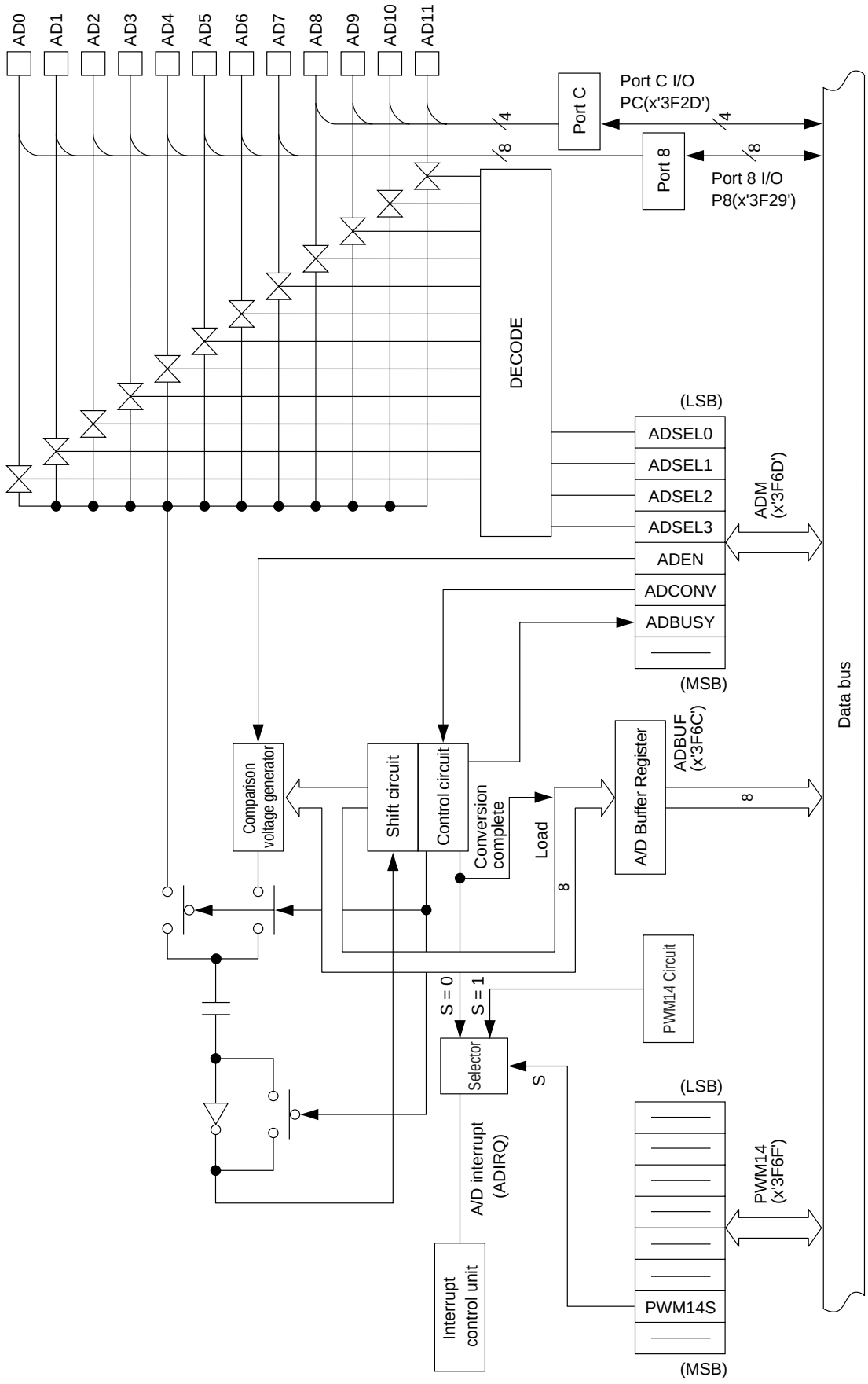


Figure 7-1-2 A/D Converter Block Diagram

7-2 D/A Converter

This IC contains a built-in 8-bit D/A converter. The D/A conversion output either uses a free-running counter for synchronous output, or the value written to the buffer is directly converted and then directly output. Refer to Chapter 8, "Synchronous Output", for instructions on the use of the synchronous output method.
The D/A conversion output pin functions as both the HBUFD2 Pin and as Port 4 (P42).

7-2-1 D/A Converter Configuration

The D/A converter consists of the following components.

(1) D/A Conversion Output Data Buffer (DABUF)
(2) 8-bit D/A conversion circuit

(1) This buffer stores data to be converted. This D/A Conversion Output Data Buffer is written to with either a synchronous write synchronously outputs the D/A Conversion Synchronous Output Data Register, or with a direct write where the data is output directly. The writing method is switched by setting General-Purpose Output Buffer/Synchronous Output Switching Register 3 (HOCRSEL3: HOCRCNT, bp4 of 3F7Ch, R/W) to "Synchronous output buffer" (to '1') to specify synchronous write, or to "General-purpose output buffer" (to '0') to specify direct writing.

(2) The 8-bit D/A conversion circuit converts the 8-bit data in the D/A Conversion Output Data Buffer and outputs the converted data to the D/A Conversion Output Pin. The D/A Conversion Output Enable Flag (DAEN: bp5 of HOCRCNT) controls the 8-bit D/A conversion circuit. If the D/A Conversion Output Enable Flag is set to "D/A conversion output" (to '1'), D/A conversion will be performed. If set to "stop D/A conversion" (to '0'), output of the D/A conversion is terminated and the 8-bit D/A conversion circuit enters the power-down mode.

7-2-2 How to Use the D/A Converter

Perform the following procedure to set direct output of the D/A converter.

(1) Set the General-Purpose Output Buffer/Synchronous Output Switching Register 3 (HOCRSEL3: HOCRCNT bp4 of x'3F7C') to "General-purpose output buffer" (to '0') to set direct output of the D/A conversion.
(2) Set the D/A Conversion Output Enable Flag (DAEN: bp5 of HOCRCNT) to "D/A conversion output" (to '1') to configure the output pin as D/A output.
(3) Write the desired value to be converted to the D/A Conversion Output Data Buffer (DABUF: x'3FAF', R/W).

In the above procedure, the value written to the D/A Conversion Output Data Buffer is converted from a digital to analog signal and output from the D/A Output Pin. For instructions on the use of synchronous output, refer to Chapter 8, "Synchronous Output Function".

[☞ See Chapter 8 Fig. 8-4-3 "Synchronous Output Unit 2" for the D/A block diagram.]

Chapter 8 Synchronous Output Functions

8

8-1 Synchronous Output Configuration

The synchronous output is configured from the following hardware components.

[☞ Figure 8-1-1, Figure 8-1-2]

- 16-bit Synchronous Output Timing Data Register (SPGTIM)
- 8-bit Synchronous Output Data Register (SPGDAT)
- 8-bit Synchronous Output Data Register 0 (HOCRDAT0)
- 5-bit Synchronous Output Data Register 1 (HOCRDAT1)
- 1-bit Synchronous Output Data Register 2 (HOCRDAT2)
- 26-bit Free-Running Counter (FRC)
- 18-bit Free-Running Counter Data Register (FRCD)
- 16-bit Output Compare Register (FOCR0, FOCR1, FOCR2)
- 8-bit D/A Conversion Synchronous Output Data Register (DADAT)

These hardware components can generate an output pattern synchronized to an event that has been input and captured.

8-2 Synchronous Output Control Registers

The synchronous output function is controlled by the following registers.

[☞ Table 8-2-1]

Table 8-2-1 Synchronous Output Control Registers

Register Name		RAM Address	R/W	Function
FOCR0	FOCR0L	x'2E16'	R/W	Free-Running Counter Output Compare Register 0 (lower)
	FOCR0H	x'2E17'		Free-Running Counter Output Compare Register 0 (upper)
FOCR1	FOCR1L	x'2E18'	R/W	Free-Running Counter Output Compare Register 1 (lower)
	FOCR1H	x'2E19'		Free-Running Counter Output Compare Register 1 (upper)
FOCR2	FOCR2L	x'2E1A'	R/W	Free-Running Counter Output Compare Register 2 (lower)
	FOCR2H	x'2E1B'		Compare Register 2 (upper)
P1(SPGBUF)		x'3F22'	R/W	Synchronous Output Buffer
SPGTIM	SPGTIML	x'2E1C'	R/W	Synchronous Output Timing Register 0 (lower)
	SPGTIMH	x'2E1D'		Synchronous Output Timing Register 0 (upper)
SPGDAT		x'3F7E'	R/W	Synchronous Output Data Register
SPGCNT		x'3F7F'	R/W	Synchronous Output Control Register
P1DIR (SPGDIR)		x'3F2F'	R/W	Synchronous Output Buffer Direction Control Register
FRCLH		x'2E15'	R/W	Free-Running Counter Data Capture Control Register (read-only)
FRCD	FRCDL FRCDM FRCDH	x'2E12' x'2E13' x'2E14'	R	Free-Running Counter Data Register (read-only)
HOCRDAT0		x'3F69'	R/W	Synchronous Output Data Register 0
HOCRDAT1		x'3F79'	R/W	Synchronous Output Data Register 1
HOCRDAT2		x'3F7A'	R/W	Synchronous Output Data Register 2
P4OUT (HOCRBUF0)		x'3F24'	R/W	Synchronous Output Data Buffer 0
HOCRBUF1		x'3F7B'	R/W	Synchronous Output Data Buffer 1
HOCRCNT		x'3F7C'	R/W	Synchronous Output Control Register
RCTLBUF		x'3F7D'	R/W	RCTL Buffer Register
DADAT		x'3FAE'	R/W	D/A Conversion Synchronous Output Data Register
DABUF		x'3FAF'	R/W	D/A Conversion Output Data Buffer

Each register is described below.

(1) Free-Running Counter Output Compare Registers

(FOCR0: x'2E17', x'2E16', R/W)

This is a 16-bit general-purpose output compare register for use if the VSYNC signal is missing (NF signal).

The NF signal is a dummy signal used when VSYNC is missing to capture the FRC count value in the ICRVSYN Register instead of the VSYNC signal. The NF signal is generated when the value set in FOCR0 matches the FRC count value.

[☞ 9-2-2, "VSYNC Weak Electric Field Backup Function (NF, N8)"]

(FOCR1: x'2E19', x'2E18', R/W)

This 16-bit general-purpose Output Compare Register controls output of the P4OUT Register, HOCRBUF1 Register, and DABUF Register. When data set in the FOCR1 Register matches the free-running counter, the below operations are performed.

1) Data set in Synchronous Output Data Register 0 (HOCRDAT0: x'3F69', R/W) is output to Synchronous Output Data Buffer 0 (P4OUT (HOCRBUF0): x'3F24', R/W).
2) Data set in Synchronous Output Data Register 1 (HOCRDAT1: x'3F79', R/W) is output to Synchronous Output Data Buffer 1 (HOCRBUF1: x'3F7B', R/W).
3) Data set in D/A Conversion Synchronous Output Data Register (DADAT: x'3FAE', R/W) is output to D/A Conversion Output Data Buffer (DABUF: x'3FAF', R/W).

Write data to the (16-bit) registers that set synchronous timing by using MOVW instructions to set all 16 bits at the same time.

(FOCR2: x'2E1B', x'2E1A', R/W)

This 16-bit general-purpose output compare register generates recording control signals. When data set in the FOCR2 Register matches the free-running counter, data set in Synchronous Output Data Register 2 (HOCRDAT2: x'3F7A', R/W) is output to the RCTL Buffer Register (RCTLBUF: x'3F7D', R/W).

(2) Synchronous Output Buffer (P1 (SPGBUF): x'3F22', R/W)

When the Synchronous Output Timing Register (SPGTIM: x'2E1D', x'2E1C', R/W) matches the free-running counter, this buffer outputs data set in the Synchronous Output Data Register (SPGDAT: x'3F7E', R/W).

(3) Synchronous Output Timing Register (SPGTIM: x'2E1D', x'2E1C', R/W)

This 16-bit data register sets the synchronous output timing. If Synchronous Output Control Register (SPGCNT: x'3F7F', R/W) is set to "Synchronous output", when the SPGTIM register matches the free-running counter, output data set in the Synchronous Output Data Register (SPGDAT: x'3F7E', R/W) is output to the SPGBUF Register.

Do not write to the SPGTIM and SPGDAT Registers at or near the time of synchronous output. Use a MOVW instruction to set all 16 bits of the SPGTIM Register at the same time.

(4) Synchronous Output Data Register (SPGDAT: x'3F7E', R/W)

When SPGTIM matches the free-running counter, this register sets the data desired to be output to the P1 Register (SPGBUF).

(5) Synchronous Output Control Register (SPGCNT: x'3F7F', R/W)

The SPGSEL1 to SPGSEL7 Flags (bp1 to bp7) set whether bits 1 through 7 of the SPGBUF Register are used as a synchronous output buffer, or as a general-purpose I/O port. If set to '0', those bits can be used as a general-purpose I/O port. If set to '1', those bits can be used as a synchronous output buffer. SPGSEL0 (bp0) is the SPGHSW Flag. If this flag is set to '0', the value written to bp0 of the P1 Register is used as the HSW signal. If set to '1', the value in the SPGDAT Register is synchronously output and the HSW signal is generated.

When the output finishes (after interrupt processing is completed), set the next sequential synchronous output.

(6) Synchronous Output Control Register (HOCRCNT: x'3F7C', R/W)

This control register sets the HOCRBUF0, HOCRBUF1 and DABUF Registers to either synchronous output or general-purpose output. Synchronization is performed by the Output Compare Register (FOCR1).

The HOCRSEL0, HOCRSEL1 and HOCRSEL3 Flags (bp1, bp2 and bp4) select whether HOCRBUF0, HOCRBUF1 and DABUF are used as synchronous output ports, synchronous output buffers, general-purpose I/O ports or general-purpose output buffers.
If the HOCRSEL0 Flag is set to '0', HOCRBUF0 can be used as a general-purpose I/O port. If the HOCRSEL1 Flag is set to '1', the HOCRBUF1 Register is a general-purpose output buffer.
If the HOCRSEL2 Flag of the HOCRDAT2 Register is set to '0', writing to RCTLBUF will cause the RCTLD signal to be generated. If the HOCRSEL2 Flag is set to '1', when Free-Running counter (FRC) matches FOCR2, bp 0 of HOCRDAT2 (x'3F7A') will be synchronously output and the RCTLD signal will be generated.
If HOCRSEL3 (bp4) is set to '1', when FOCR1 matches Free-Running Counter (FRC), the value set in DADAT (x'3FAE') is synchronously output to DABUF (x'3FAF'). If set to '0', a value is directly set in the DABUF Register.
DAEN (bp5) enables or disables the D/A conversion output. If DAEN is set to '1', regardless of the HOCRBUF0 setting, the HBUFD2 output will be D/A conversion output.

(7) Synchronous Output Data Register (HOCRDAT0, 1: x'3F69', x'3F79', R/W)

When the FOCR1 Register matches FRC, the HOCRBUF0 and HOCRBUF1 Registers set data to be output. The HOCRDAT0 Register is synchronously output to the P4OUT (HOCRBUF0) Register and the HOCRDAT1 Register is synchronously output to the HOCRBUF1 Register.

The following bits can be used as general-purpose I/O ports: bp0, bp2, bp4, and bp6.

(8) Synchronous Output Data Buffer 0 (P4OUT (HOCRBUF0): x'3F7B', R/W)

When FOCR1 matches FRC, this output buffer will output the output data set in the HOCRDAT0 Register as 3 values (HIGH, LOW or Hi-Z). This buffer can also be used as a 4-bit general-purpose I/O port. To use as a general-purpose input, set the odd bits (HBUFD1, HBUFD3, HBUFD5 and HBUFD7 Flags) to '0'. To use as a general-purpose output, set those bits to '1'.

(9) Synchronous Output Data Buffer 1 (HOCRBUF1: x'3F7C', R/W)

When the FOCR1 Register matches FRC, this output buffer will output the output data set in the HOCRDAT1 Register. If HOCRSEL (bp2 of HOCRCNT Register) is set to '0', this buffer can be used as a 5-bit internal output buffer.

(10) Synchronous Output Buffer Direction Control Register
(P1DIR: x'3F2F', R/W)
This register sets the input or output direction individually for bp1 to bp7 of the P1 (SPGBUF) Register.

(11) Free-Running Counter Data Capture Register
(FRCD: x'2E14', x'2E13', x'2E12', R)
This register is used to capture data of the free-running counter. If an instruction to read the Free-Running Counter Data Capture Control Register (FRCLH: x'2E15', R/W) is executed, the value of the free-running counter will be captured by FRCD.

(12) Free-Running Counter Data Capture Control Register
(FRCLH: x'2E15', R)
If an instruction to read this register is executed, the value of the free-running counter will be captured by FRCD.

(13) D/A Conversion Synchronous Output Data Register
(DADAT: x'3FAE', R/W)
When the FOCR1 Register becomes equal to FRC, this register sets data for synchronous output to the DABUF Register .

For the D/A conversion function, also refer to Chapter 7.

(14) D/A Conversion Output Data Buffer (DABUF: x'3FAF', R/W)
When the FOCR1 Register matches FRC, this output buffer will synchronously output the output data set in the DATAT Register to the D/A converter. If data is set directly in the D/A Conversion Output Data Buffer, when FOCR1 matches FRC, direct output of the D/A conversion will be possible without synchronization. If synchronous output is to be performed, set the HOCRSEL3 Flag (bp4 of HOCRCNT) to '1'. If synchronous output is not to be performed, set the HOCRSEL3 Flag to '0'. If D/A conversion is to be used, set DAEN (bp5 of HOCRCNT) to '1'.

When the output finishes (after interrupt processing is completed), set the next sequential synchronous output.

8-3 Synchronous Output Function 1 (SPG Function)

8-3-1 Synchronous Output Pin Configuration

Pins SBUFD1 to SBUFD7 are synchronous output pins.
The SBUFD1 to SBUFD7 Pins also function as Port 1 (P11 to P17).
Refer to Chapter 3, "Port Functions", for instructions on how to use the port functions of these pins.

The Synchronous Output Control Register (SPGCNT: x'3F7F', R/W) specifies whether SBUFD1 to SBUFD7 Pins are used as general-purpose I/O or as synchronous output.

8-3-2 How to Use Synchronous Output 1

Operation of the SPG function and settings for each register are described below with an example.

Example 1) Pins SBUFD1 to SBUFD3 are used as synchronous output pins and Pin SBUFD4 is used as a general-purpose I/O pin.

(1) At time (1) in Figure 8-3-1, set the following: initial values for each output data, synchronous output timing, and output data.

- Initialize the synchronous output pins with the Synchronous Output Control Register (SPGCNT: x'3F7F', R/W) and Synchronous Output Buffer Direction Control Register (P1DIR: x'3F2F', R/W).
  SPGCNT=B'***00000'
  P1DIR=B'***1111*'
  P1 (SPGBUF)=B'***1010*'

- Set the synchronous output pins.
  SPGCNT=B'***0111*'

- Set the synchronous output timing (SPGTIM: x'2E1D', x'2E1C') and the synchronous output data (SPGDAT).
  SPGTIM=($N_{ICRHSW(n)}$+$N_{\Delta t1}$)/4
  SPGDAT=B'***1011*'

With the above settings, SPGDAT data is output when FRC=SPGTIM.
An SPG interrupt is generated at this time.

- When processing an FOCR1 interrupt, set output data at the time of the next synchronous output and then finish processing the interrupt.
  (Perform the same settings as time t1.)
  SPGTIM=(NICRHSW(n)+NΔt2)/4
  SPGDAT=B'***1101*'

(2) At time (2) in Figure 8-3-1, an instruction outputs a low-level signal from pin SBUFD4. At this time, settings for each register are the same as time (1). In other words, except for synchronous output pins, software instructions may freely modify values.

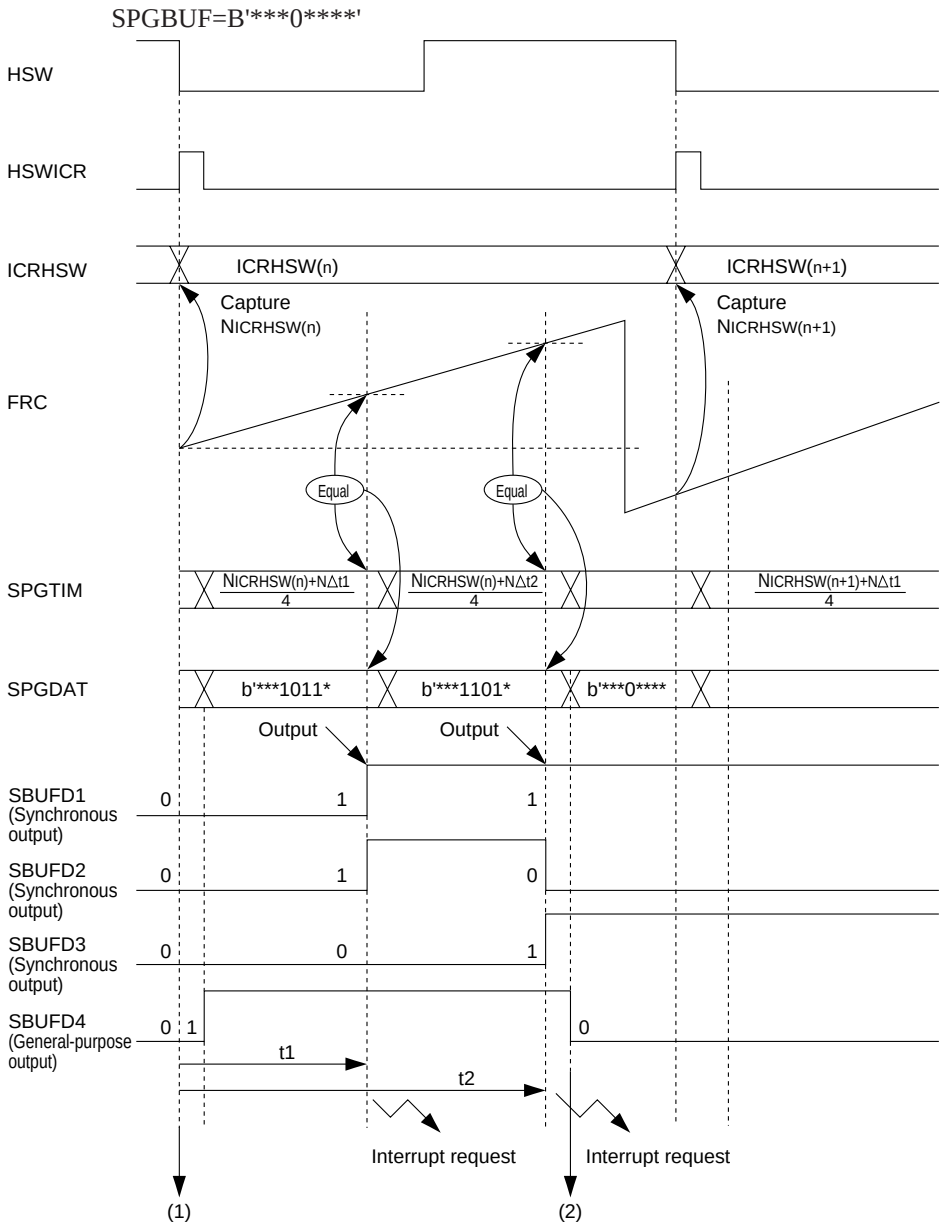


Figure 8-3-1 Synchronous Output Function 1 (SPG Function)

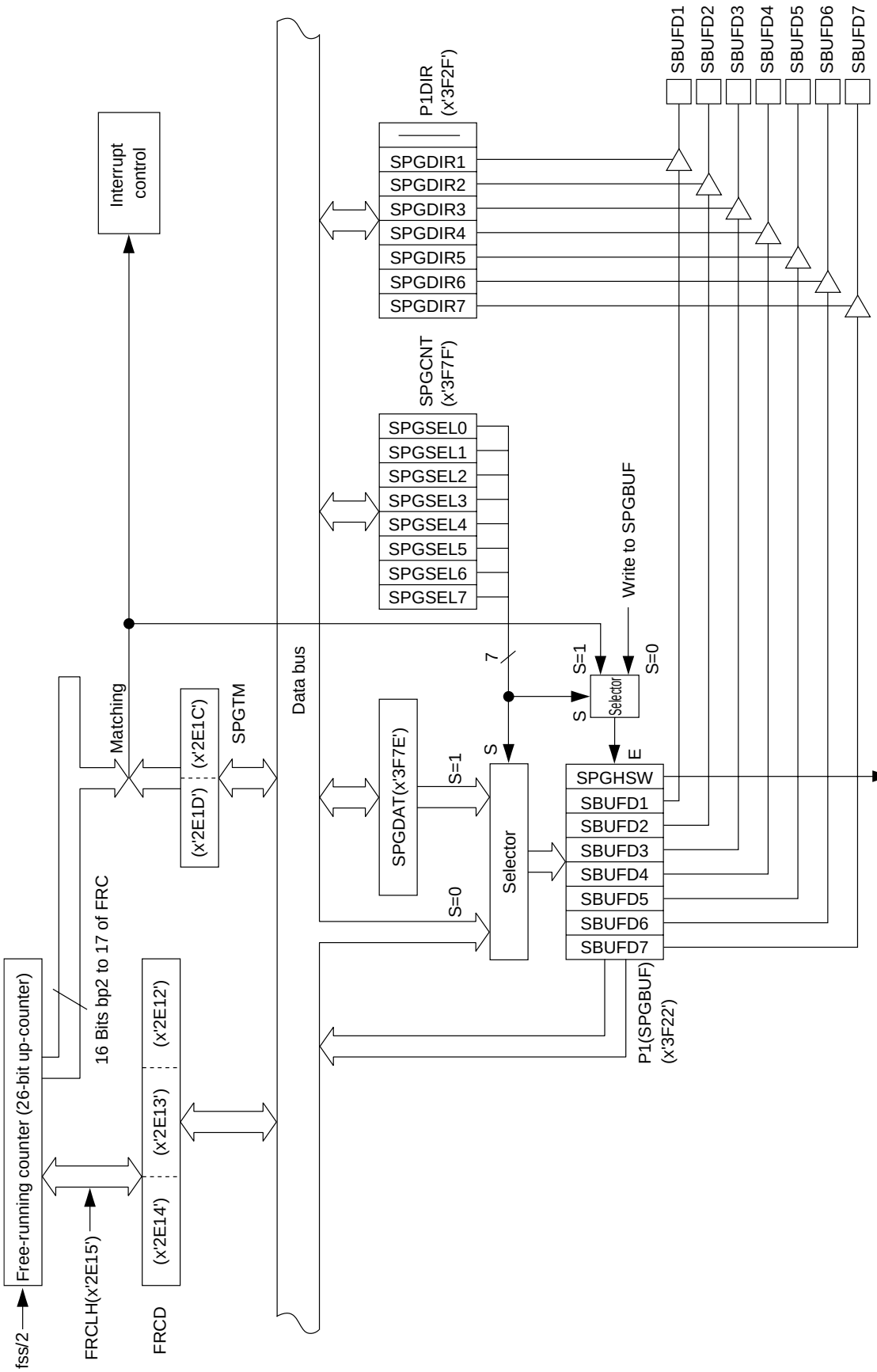


Figure 8-3-2 Synchronous Output Unit 1 (SPG)

8-4 Synchronous Output Function 2 (HOCR Function)

8-4-1 Synchronous Output Pin Configuration

Pins HBUFD0, HBUFD2, HBUFD4 and HBUFD6 are synchronous output pins.
The HBUFD0 to HBUFD6 Pins also function as Port 4 (P40 to P46).
The HBUFD2 Pin also functions as a D/A conversion output pin.
The HBUFD6 Pin also functions as a buzzer output pin.

Refer to Chapter 3, "Port Function", for instructions on how to use the port function of these pins. Refer to section 7-2, "D/A Converter", and section 8-5, "D/A Conversion Synchronous Output Function", for instructions on how to use the D/A conversion.

The HOCRSEL0 Flag (bp1 of HOCRCNT) of the Synchronous Output Control Register (HOCRCNT: x'3F7C', R/W) specifies whether HBUFD0, HBUFD2, HBUFD4 and HBUFD6 will be used as general-purpose I/O pins or as synchronous output pins. The DAEN Flag (bp5 of HOCRCNT) of the Synchronous Output Control Register (HOCRCNT: x'3F7C', R/W) specifies whether the HBUFD2 pin will be used as D/A conversion output.

8-4-2 How to Use Synchronous Output 2

Operation of the HOCR function and settings for each register are described below with examples.

Example 1) FOCR1 (Free-Running Counter Output Compare Register 1) is used and P4OUT (HOCRBUF0) (Synchronous Output Data Buffer 0) is used as a synchronous output.

(1) At time (1) in Figure 8-4-1, set the following: initial values for each output data, synchronous output timing, and output data.

- With the HOCRSEL0 (bp1) and DAEN (bp5) flags of the Synchronous Output Control Register (HOCRCNT: x'3F7C', R/W) set the P4OUT (HOCRBUF0) Register to general-purpose I/O. Make initial settings with the Synchronous Output Data Buffer 0 (P4OUT (HOCRBUF0: x'3F24', R/W), Synchronous Output Data Register (HOCRDAT0: x'3E69', R/W) and Free-Running Counter Output Compare Register 1 (FOCR1: x'2E19', x'2E18', R/W).
  HOCRCNT=B'**0***0*'
  P4OUT (HOCRBUF0)=B'10100*11'
- Set the HOCRSEL0 Flag (bp1) of the Synchronous Output Control Register (HOCRCNT: x'3F7C', R/W) to '1' to specify synchronous output of HOCRBUF0.
  HOCRCNT=B'**0***1*'
- Set the synchronous output timing (FOCR1) and the output data (HOCRDAT0).
  FOCR1=($N_{ICRHSW2}+N\Delta t1$)/4
  HOCRDAT0=B'10111110'

(2) With the above settings, when FRC=FOCR1, the data of HOCRDAT0 is output. At this time, an FOCR1 interrupt will be generated.

- When processing an FOCR1 interrupt, set output data at the time of the next synchronous output and then finish processing the interrupt.
  After the next HSW capture, at time (2), set the output data and synchronous timing.
  FOCR1=($N_{ICRHSW2}+N\Delta t2$)/4
  HOCRDAT0=B'110*110*'
  HOCRCNT=B'**0***1*'

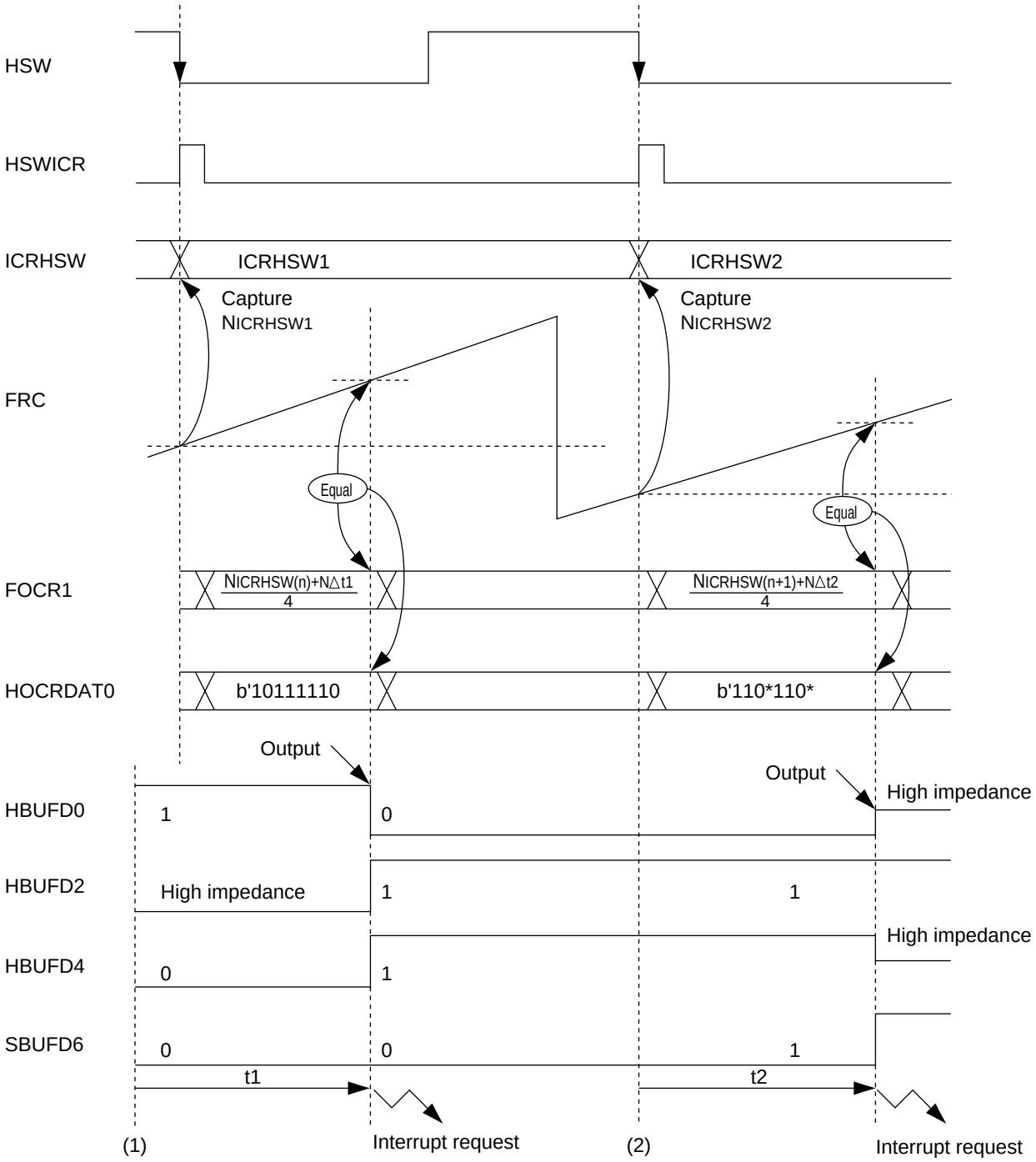


Figure 8-4-1 Synchronous Output Function 1 (HOCR Function), Example 1

Example 2) FOCR1 (Free-Running Counter Output Compare Register 1) is used and HOCRBUF1 (Synchronous Output Data Buffer 1) is used as a synchronous output.

(1) At time (1) in Figure 8-4-2, set the following: initial values for each output data, synchronous output timing, and output data.

- With the HOCRSEL1 (bp2) of the Synchronous Output Control Register (HOCRCNT: x'3F7C', R/W) set the HOCRBUF1 Register to general-purpose I/O. Make initial settings with the Synchronous Output Data Buffer 1 (HOCRBUF1: x'3F7B', R/W), Synchronous Output Data Register (HOCRDAT1: x'3F79', R/W) and Free-Running Counter Output Compare Register 1 (FOCR1: x'2E18', x'2E19', R/W).
  HOCRCNT=B'*****0**'
  HOCRBUF1=B'***11001'

- Set the HOCRSEL1 Flag of the Synchronous Output Control Register (HOCRCNT: x'3F7C', R/W) to '1' to specify synchronous output of HOCRBUF1.
  HOCRCNT=B'*****1**'

- Set the synchronous output timing (FOCR1) and the output data (HOCRDAT1).
  $FOCR1=(N_{ICRHSW(n)}+N\Delta t1)/4$
  HOCRDAT1=B'***10010'

(2) With the above settings, at time (2) when FRC=FOCR1, the data of HOCRDAT1 is output. At this time, an FOCR1 interrupt will be generated.

- When processing an FOCR1 interrupt, set output data at the time of the next synchronous output and then finish processing the interrupt. (2)
  $FOCR1=(N_{ICRHSW(n)}+N\Delta t2)/4$
  HOCRDAT1=B'***01100'

After the next HSW capture, at time (3), set the output data and synchronous timing.

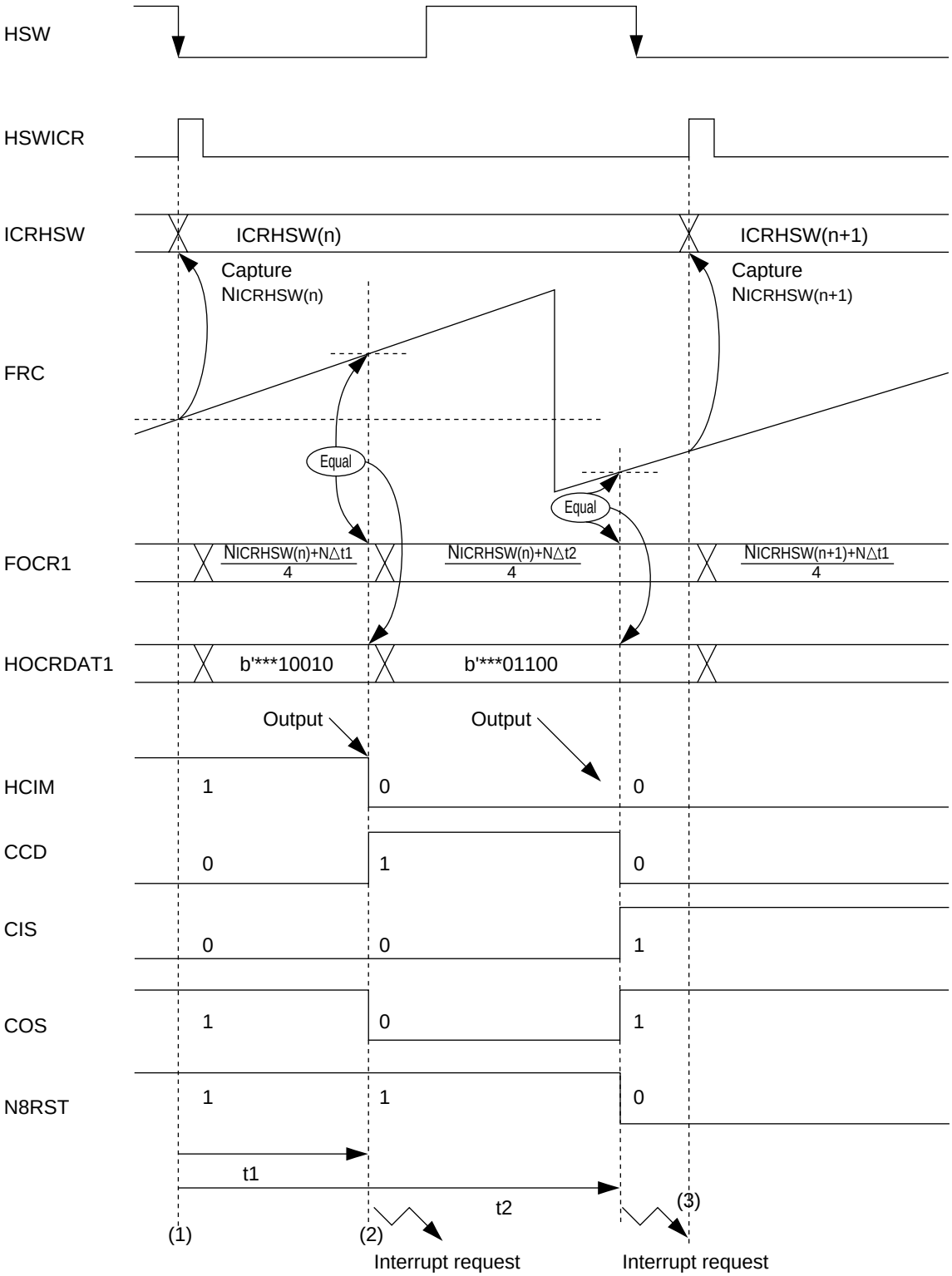


Figure 8-4-2 Synchronous Output Function 2 (HOCR Function), Example 2

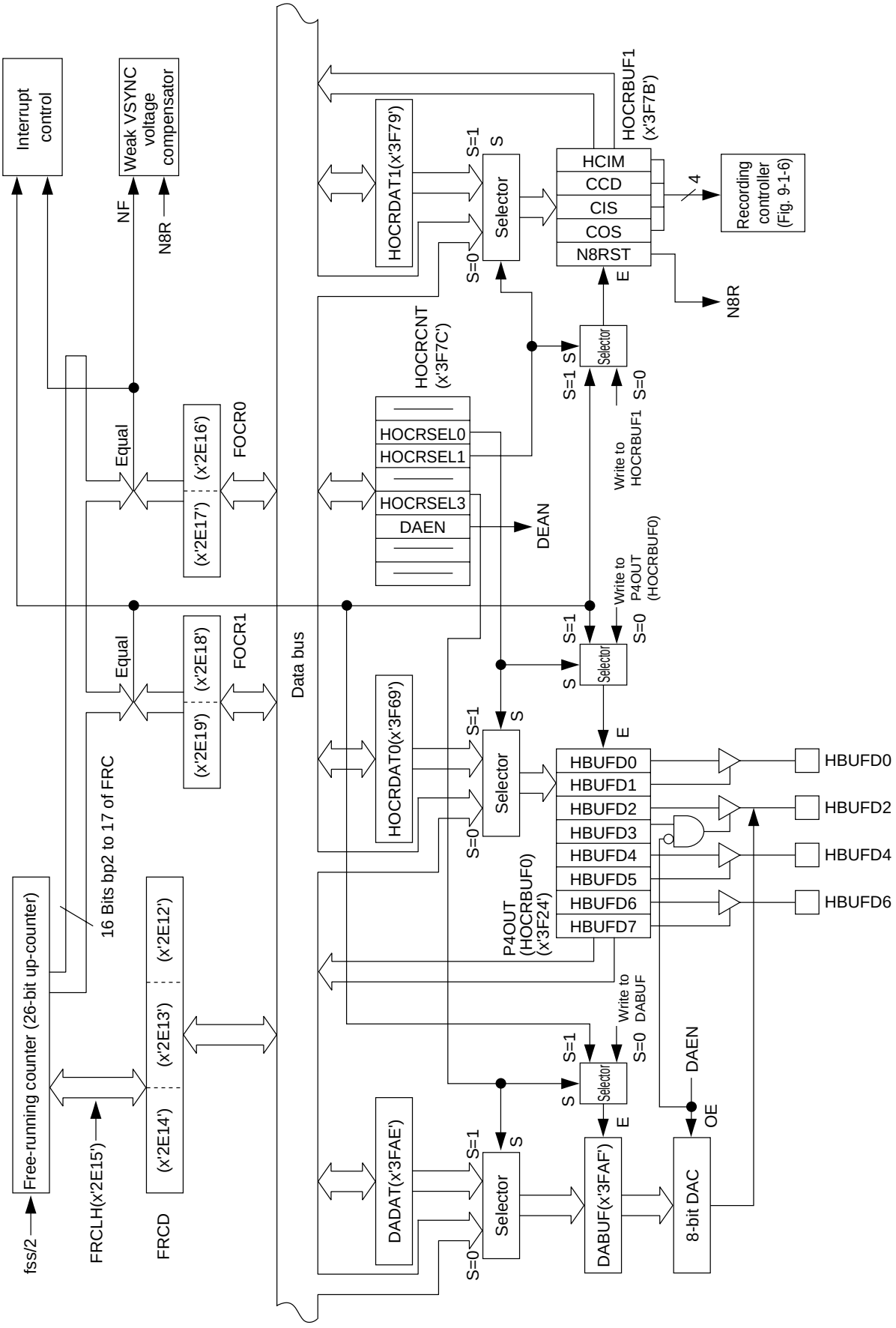


Figure 8-4-3 Synchronous Output Unit 2 (HOCR, D/A Conversion Output)

8-5 D/A Conversion Synchronous Output

8-5-1 D/A Conversion Output Pin Configuration

The D/A conversion output pin functions as the HBUFD2 pin for the HOCR function and as Port 4 (P42).
Refer to section 8-4, "Synchronous Output Function 2" for instructions on how to use the HOCR function of this pin. Refer to Chapter 3, "Port Functions" for instructions on how to use the port function.

To use D/A conversion output as a synchronous output, set the HOCRSEL3 Flag (bp4 of HOCRCNT) of the Synchronous Output Control Register (HOCRCNT: x'3F7C', R/W). To set values directly in the D/A Conversion Output Data Buffer and not use the synchronous output, refer to section 7-2, "D/A Converter."

8-5-2 How to Use the D/A Conversion Synchronous Output

Operation of the D/A conversion synchronous output function and settings for each register are described below with an example.

Example) Synchronized to FOCR1 (Free-Running Counter Output Compare Register 1), the value written to D/A Conversion Synchronous Output Data Register (DADAT: x'3FAE', R/W) is converted from a digital to analog signal, and is synchronously output to the HBUFD2 pin.

(1) At time (1) in Figure 8-5-1, set the following: initial values for each output data, synchronous output timing, and output data.

- Set the D/A converter output data buffer (DABUF: x'3FAF' R/W) as a general purpose output buffer using flag HOCRSEL3 (bp4) of the synchronous output control register (HOCRCNT: x'3F7C' R/W), and further set the initial value of D/A converter output in the D/A converter output data buffer (DABUF: x'3FAF' R/W).
  HOCRCNT=B'**00****'
  DABUF="Initial value of D/A conversion output"

- Set the DAEN flag of Synchronous Output Control Register (HOCRCNT: x'3F7C', R/W) to '1' to start D/A conversion. Set the HOCRSEL3 Flag (bp4 of HOCRCNT) to '1' to specify synchronous output of DABUF.
  HOCRCNT=B'**11****'

- After capture (at time (2)), set the synchronous output timing in FOCR1 (x'2E19', x'2E18') and the D/A conversion output data in the DADAT Register (DADAT: x'3FAE', R/W) .
  FOCR1=(N_{ICR1}+$N_{\Delta t1}$)/4
  DADAT="D/A conversion output data 1"
  Where, N_{ICR1}: FRC value at time of capture
  $N_{\Delta t1}$: Amount FRC increases from time of capture until synchronous output, setting value 1

(2) With the above settings, when FRC=FOCR1, the data of DADAT is output and an FOCR1 interrupt will be generated.

- After the next capture (at time (3)), set the next synchronous output timing and D/A conversion output data.
  FOCR1=(N_{ICR2}+$N_{\Delta t2}$)/4
  DADAT="D/A conversion output data 2"
  Where, N_{ICR2}: FRC value at time of capture
  $N_{\Delta t2}$: Amount FRC increases from time of capture until synchronous output, setting value 2

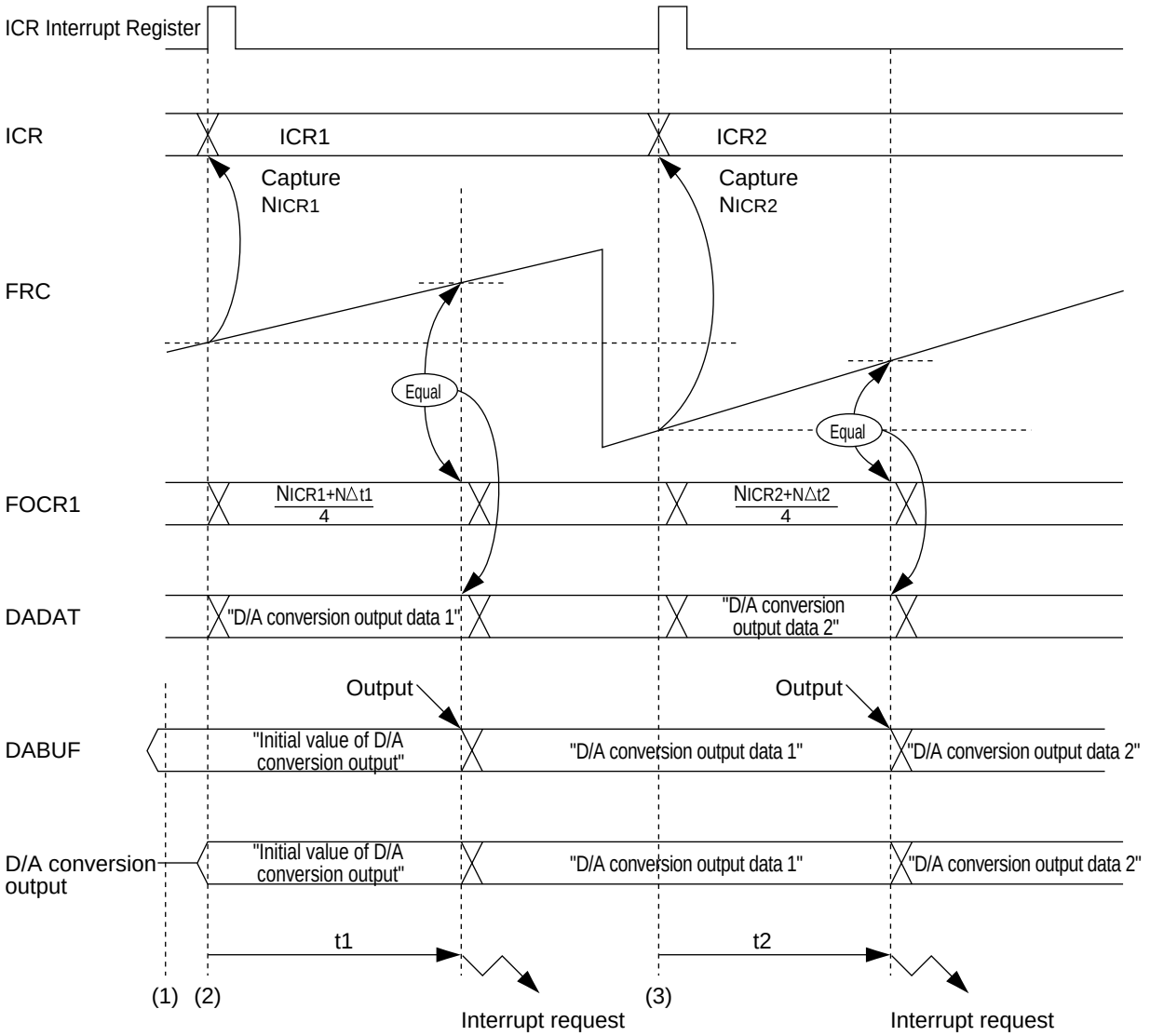


Figure 8-5-1 Synchronous Output of D/A Conversion, Example

8-6 Synchronous Output Interrupts

Synchronous output interrupts, synchronized to the Free-Running Counter (FRC), are generated when data that has been set is output. There are 4 types of interrupts that are synchronized to the Free-Running Counter (FRC).

(1) SPG Interrupt (SPGICR)

An SPG interrupt is generated when the 16-bit value set in SPGTIM (x'2E1C', x'2E1D') becomes equal to the value of the Free-Running Counter (FRC) (16 bits from bp2 of bp17).

Perform the following procedure to set interrupt processing.

i) Set the interrupt priority level with the SPGLV0 and SPGLV1 flags (bp6 and bp7) of the SPG Interrupt Control Register (SPGICR: x'3FF3').

ii) Clear the SPG Interrupt Request Flag (SPGIR: bp0) of the SPG Interrupt Control Register (SPGIR: x'3FF3').

iii) Set the SPG Interrupt Enable Flag (SPGIE: bp1) of the SPG Interrupt Control Register (SPGIR: x'3FF3').

(2) FOCR0 Interrupt (FOCR0ICR)

An FOCR0 interrupt is generated when the 16-bit value set in FOCR0 (x'2E16', x'2E17') becomes equal to the value of the Free-Running Counter (FRC) (16 bits from bp2 of bp17).

Perform the following procedure to set interrupt processing.

i) Set the interrupt priority level with the FO0LV0 and FO0LV1 flags (bp6 and bp7) of the FOCR0 Interrupt Control Register (FOCR0ICR: x'3FF4').

ii) Clear the FOCR0 Interrupt Request Flag (FO0IR: bp0) of the FOCR0 Interrupt Control Register (FOCR0ICR: x'3FF4').

iii) Set the FOCR0 Interrupt Enable Flag (FOCR0IE: bp1) of the FOCR0 Interrupt Control Register (FOCR0ICR: x'3FF4').

(3) FOCR1 Interrupt (FOCR1ICR)

An FOCR1 interrupt is generated when the 16-bit value set in FOCR1 (x'2E18', x'2E19') becomes equal to the value of the Free-Running Counter (FRC) (16 bits from bp2 of bp17).

Perform the following procedure to set interrupt processing.

i) Set the interrupt priority level with the FO1LV0 and FO1LV1 flags (bp6 and bp7) of the FOCR1 Interrupt Control Register (FOCR1ICR: x'3FF5').

ii) Clear the FOCR1 Interrupt Request Flag (FO1IR: bp0) of the FOCR1 Interrupt Control Register (FOCR1ICR: x'3FF5').

iii) Set the FOCR1 Interrupt Enable Flag (FOCR1IE: bp1) of the FOCR1 Interrupt Control Register (FOCR1ICR: x'3FF5').

(4) FOCR2 Interrupt (FOCR2ICR)

An FOCR2 interrupt is generated when the 16-bit value set in FOCR2 (x'2E1A', x'2E1B') becomes equal to the value of the Free-Running Counter (FRC) (16 bits from bp2 of bp17).

Perform the following procedure to set interrupt processing.

i) Set the interrupt priority level with the FO2LV0 and FO2LV1 flags (bp6 and bp7) of the FOCR2 Interrupt Control Register (FOCR2ICR: x'3FF6').

ii) Clear the FOCR2 Interrupt Request Flag (FO2IR: bp0) of the FOCR2 Interrupt Control Register (FOCR2ICR: x'3FF6').

iii) Set the FOCR2 Interrupt Enable Flag (FOCR2IE: bp1) of the FOCR2 Interrupt Control Register (FOCR2ICR: x'3FF6').

8-7 RCTLD Signal Generation

The RCTLD signal is used during playback as a reference signal for the capstan phase control, and during recording as a recording control signal.

Usually, the RCTLD signal is generated either by HOCRDAT2 (x'3F7A') which is enabled when FOCR2 (x'2E1A', x'2E1B') matches the FRC (Free-Running Counter), or by an arbitrary value written to RCTLBUF (x'3F7D').
If HOCRSEL2 (bp3) of HOCRCNT (x'3F7C') is set to '1', the RCTLD signal is generated when FOCR2 becomes equal to FRC. If set to '0', RCTLD is generated when an arbitrary value is written.

Example 1) RCTLD signal is generated when FRC becomes equal to FOCR2
The case where the RCTLD signal is set at time t1 after the falling edge of HSW and the RCTLD signal is cleared at time t2 after the falling edge of HSW is described below. (See Figure 8-7-1).

(1) Specify the edge and set the interrupt so that a capture interrupt will be generated at the falling edge of HSW.

(2) Set the output data in HOCRDAT2. Set bp0 of HOCRDAT2 to '1'. Other bits may be set to arbitrary values.

(3) Set TOCRD2 to the value of (NICRHSW(n)+NΔt1)/8.

(4) At the rising edge of HSW, when the value of FRC is captured (capture value of NICRHSW(n)) , set the HOCRSEL2 flag (bp3) of HOCRCNT (x'3F7C') so that the value of HOCRDAT2 will be output to RCTLBUF when FRC becomes equal to FOCR2.

(5) When FRC=FOCR2, the data of HOCRDAT2 is output and the RCTLD signal changes to '1'. At this time, an FOCR2 interrupt is generated.

(6) During processing of an FOCR2 interrupt, set the value of (NICRHSW(n)+NΔt2)/4 in FOCR2 as the next timing data, and set the next data to be output, '0', in HOCRDAT2. Complete the interrupt processing.

(7) When FRC=FOCR2, the data of HOCRDAT2 is output and the RCTLD signal changes to '0'.

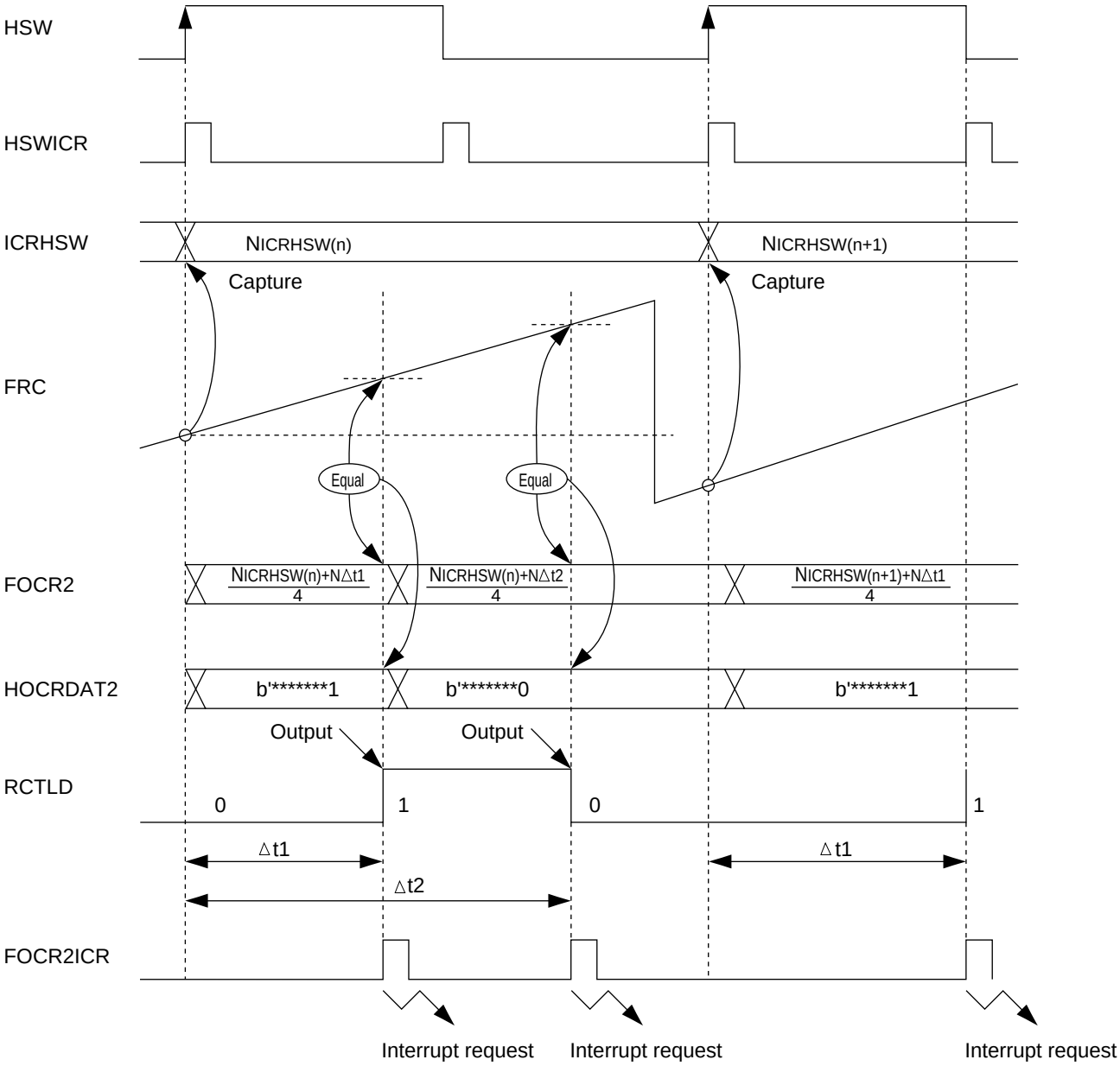


Figure 8-7-1 RCTLD Signal Generation (by HOCRDAT2)

The RCTLD signal is generated when data is written directly to RCTLBUF

(1) Set the HOCRSEL2 flag (bp3) of HOCRCNT (x'3F7C') to '0' so that values can be directly written to the RCTLBUF.

(2) The value written to RCTLBUF becomes the RCTLD signal.

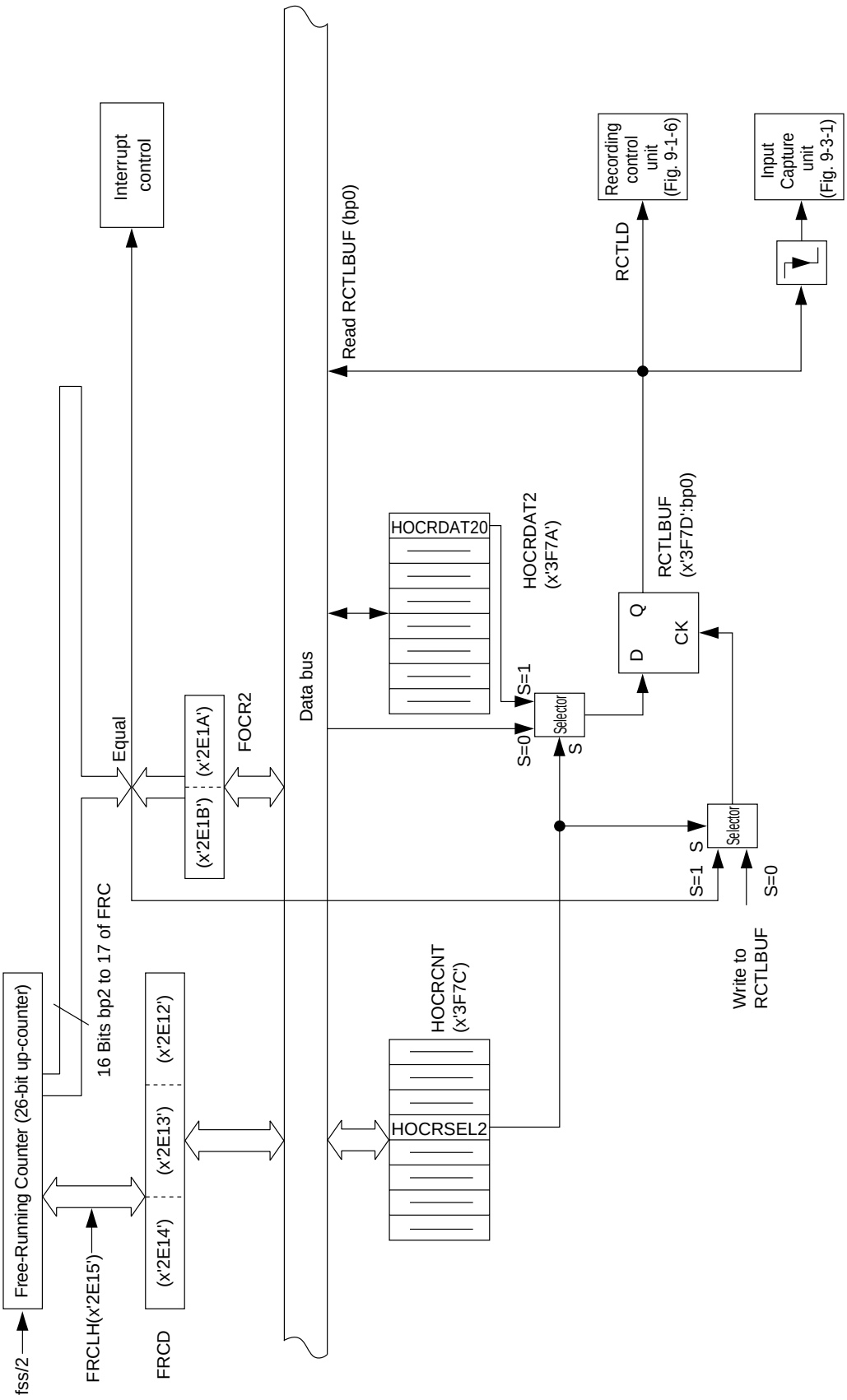


Figure 8-7-2 Recording Control Signal Generator Unit

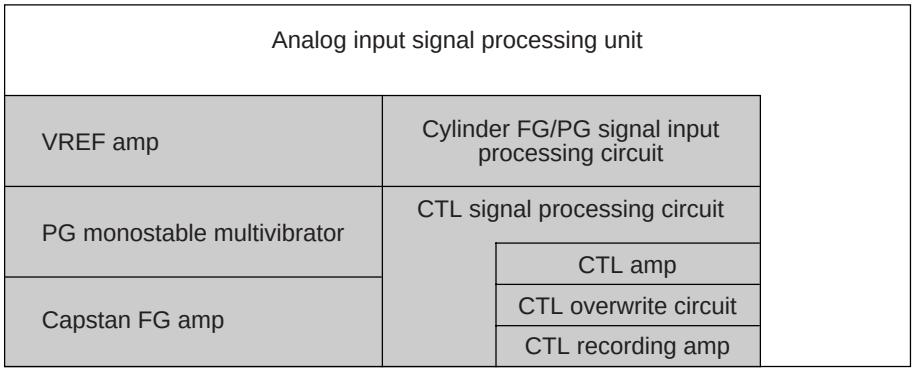
Chapter 9 Servo Functions

9

9-1 Analog Input Signal Processing Unit

9-1-1 Analog Input Signal Processing Circuits

The analog input signal processing unit consists of the following circuits.



VREF Amp

This circuit supplies reference voltage, Vref (2.5 V), both internally to the microcomputer, and externally. Internally, the reference voltage is created by means of a resistance divider. Connect capacitors to the VRI and VRO pins, and ground the AC signals.

PG Monostable Multivibrator

This analog monostable multivibrator circuit electrically adjusts the mounting position of the PG head. Connect a resistor and capacitor to the PGMM pin.

Capstan FG Amp

As shown in Figure 9-1-3, the capstan FG amp consists of an 1st stage limiter amp and a 2nd stage Schmitt amp. Set the gain of the 1st stage limiter amp by attaching external resistors.

Cylinder FG/PG Signal Input Processing Circuit

The input method can be switched by the cylinder PFG 3-value separation circuit and by the PFGS Flag (bp 4) of the Analog Control Register (ANACNT: x'3F81', R/W) when both YFG amp and PFG amp input circuits are ready.

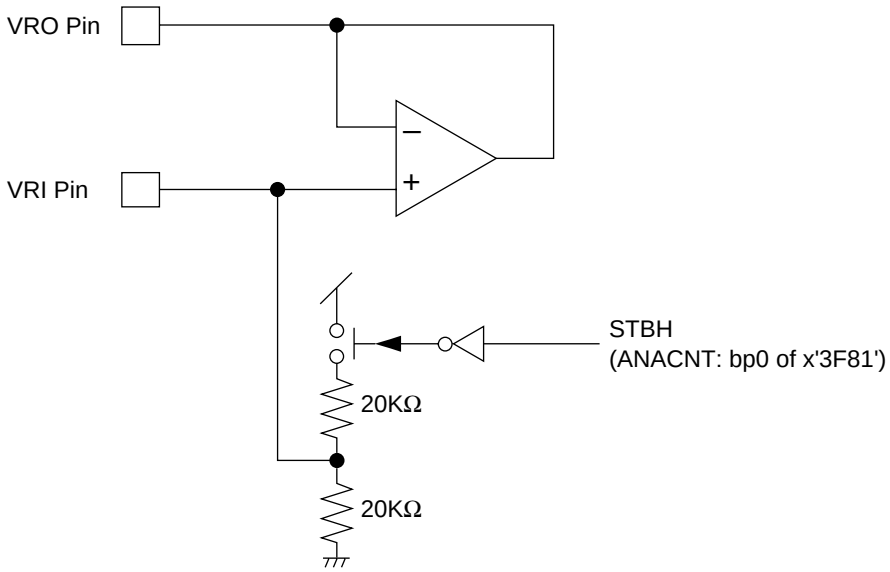


Figure 9-1-1 VREF Amp Configuration

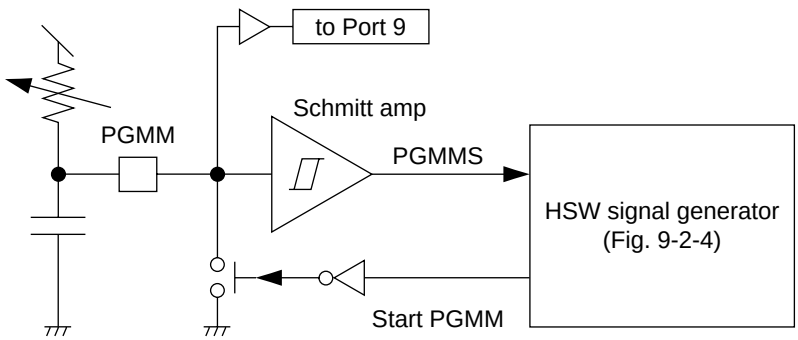


Figure 9-1-2 PG Monostable Multivibrator Configuration

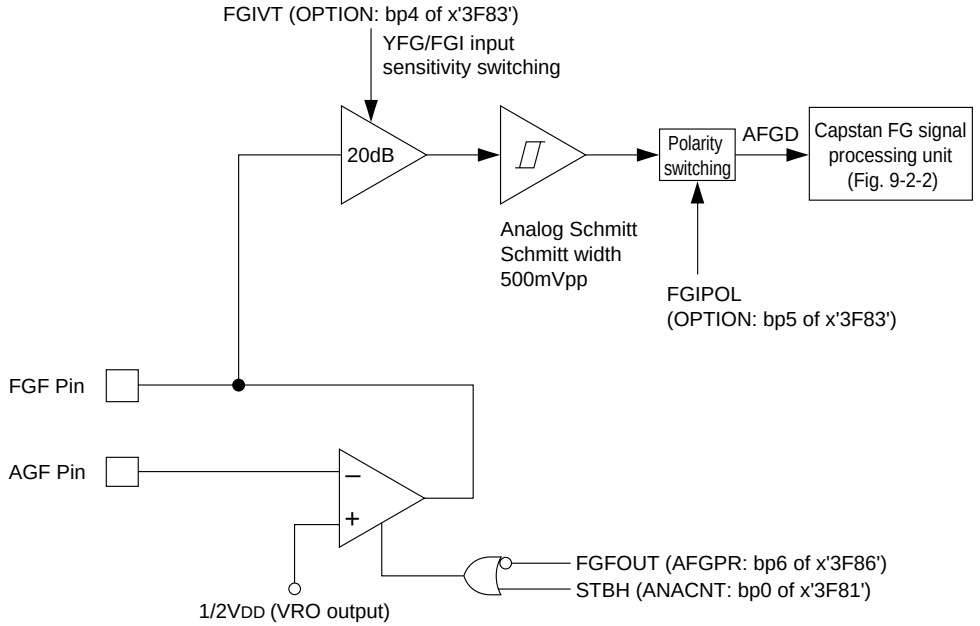


Figure 9-1-3 Capstan FG Amp Configuration

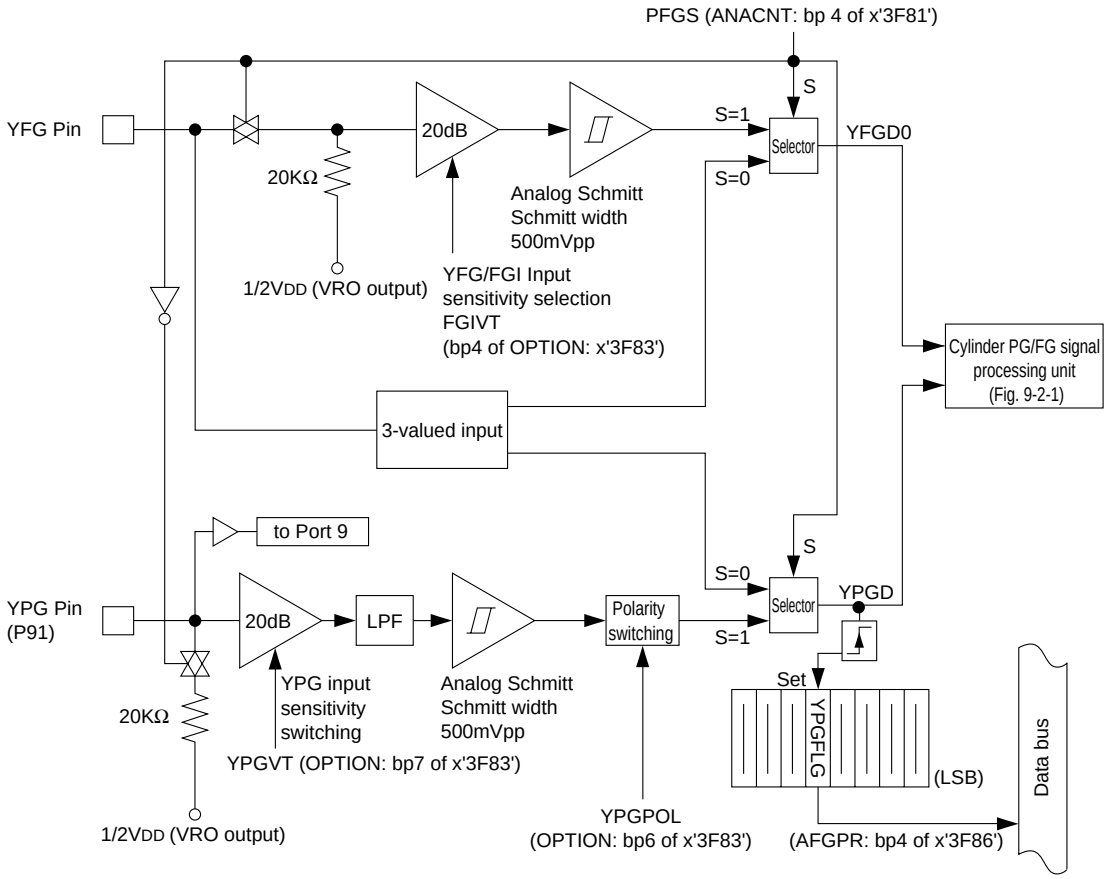


Figure 9-1-4 Cylinder PG/FG Signal Input Processing Unit Configuration

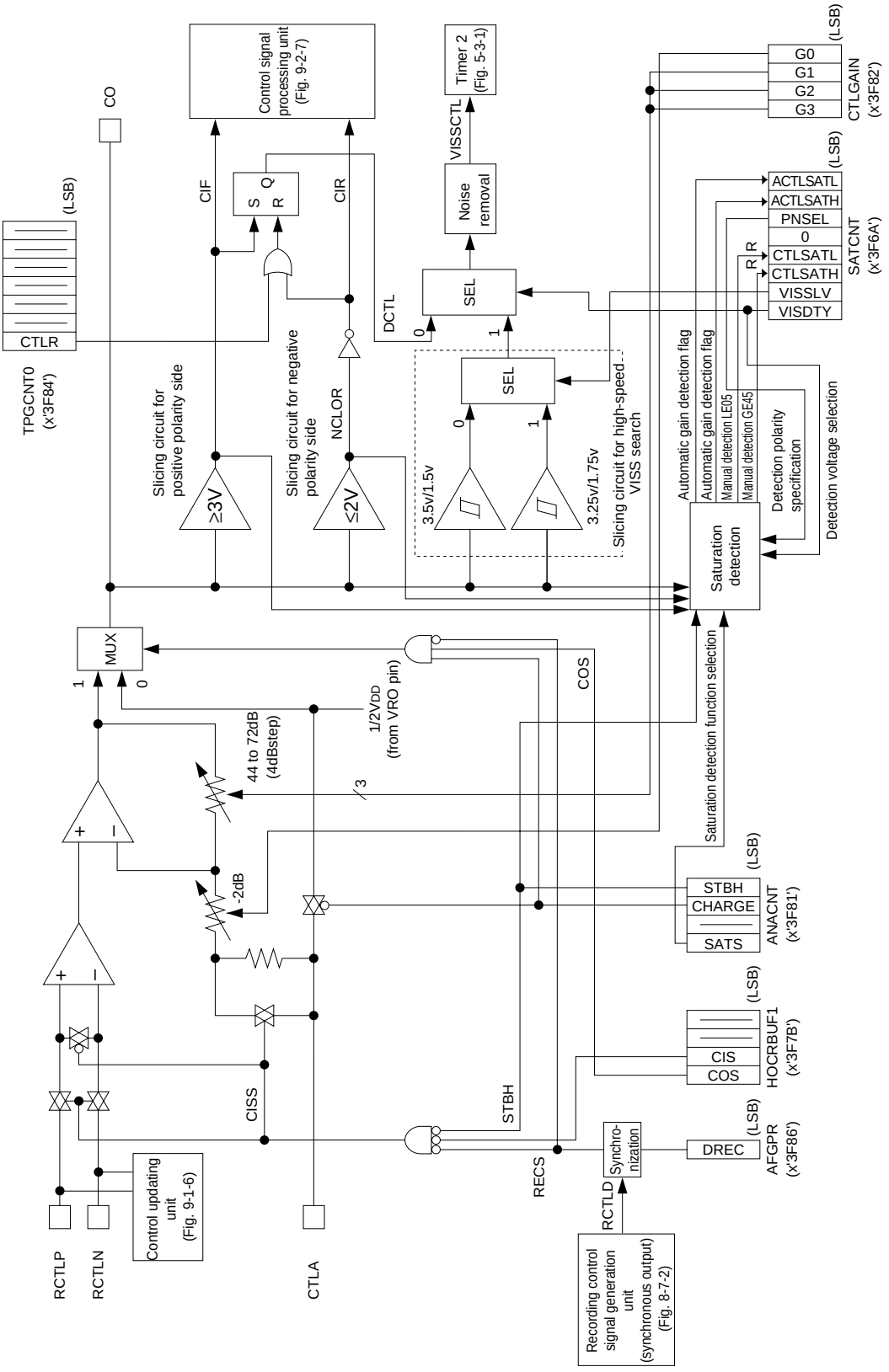


Figure 9-1-5 CTL Amp Block Diagram

CTL Signal Processing Circuit

The CTL signal input processing circuit consists of 3 components: CTL amp, CTL overwrite circuit and CTL recording amp circuit. A single chip can handle the functions of normal recording or playback through VISS/VASS overwrite.

(1) CTL Amp

Gain of the CTL amp can be set in 2 dB steps from 42 dB to 72 dB. The CTLGAIN Register (x'3F82', R/W) sets the gain.

During playback, the CTL signal is input from the RCTLP and RCTLN pins and the amp output is sent to the CO pin. The CTLA pin is used for AC ground. Connect a capacitor to the VRO pins. A slicing circuit rectifies the waveform of the CTL signal output from the CO pin.

(2) CTL Overwrite Circuit

The CTL overwrite circuit receives control signals from the HSW synchronous output, activates the trapezoidal wave recording circuit, and overwrites the CTL signal.

(3) Recording Amp

The RCTLD signal generated by HSW synchronous output causes the recording amp to record the CTL signal.

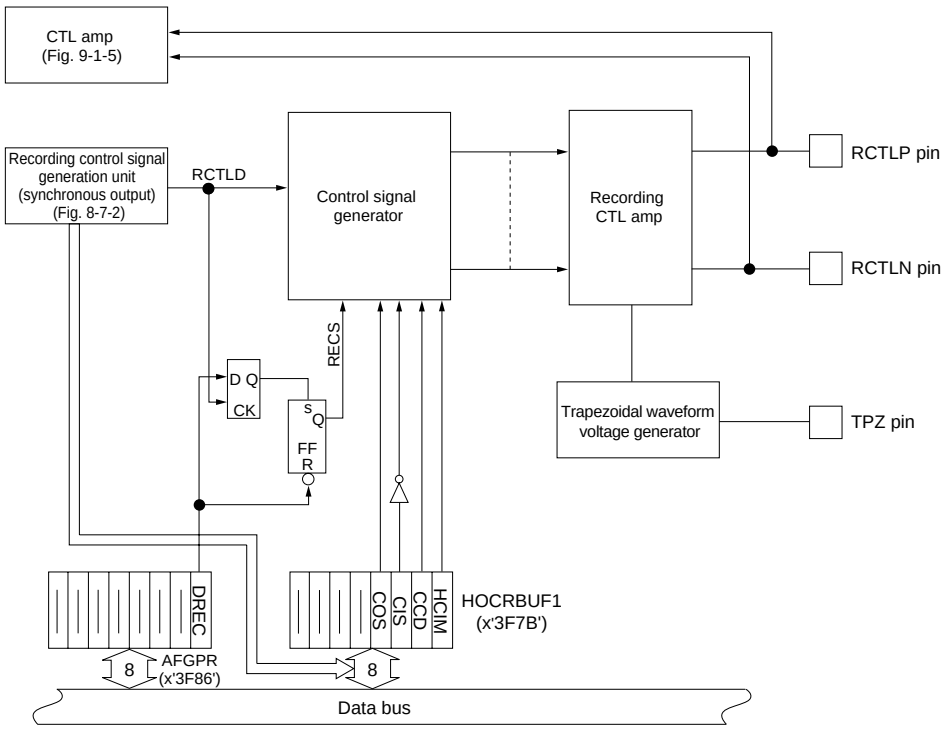


Figure 9-1-6 Control Overwrite Unit

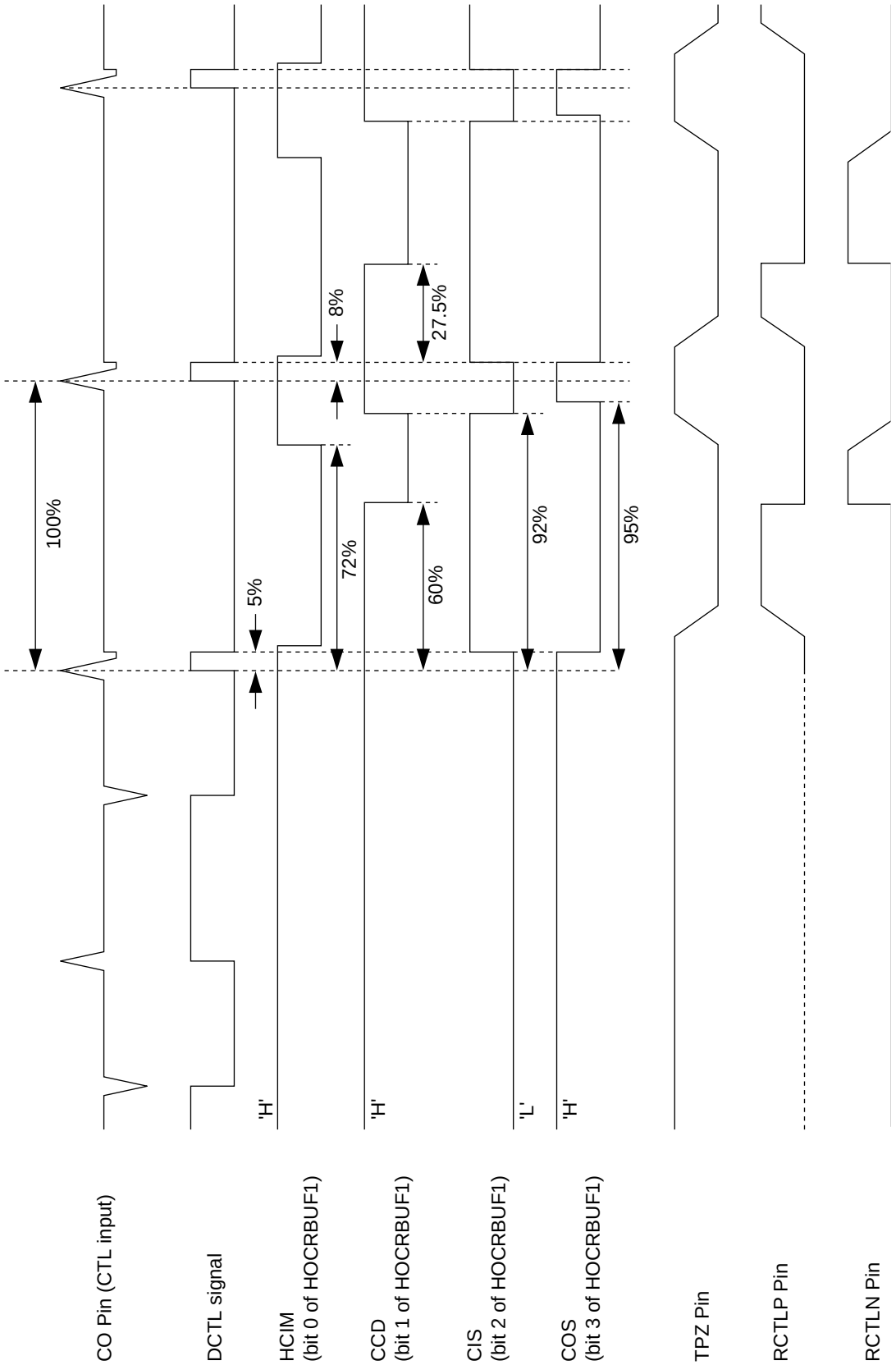


Figure 9-1-7 Control Signal Overwrite Timing Diagram

Saturation Detection Function

A 4.5V/0.5V saturation detection function is provided.
Similar to the MN101D01 Series, only manual detection is supported (initialization of the detection flag is implemented by software).

(1) Saturation detection (SATS flag = '0')

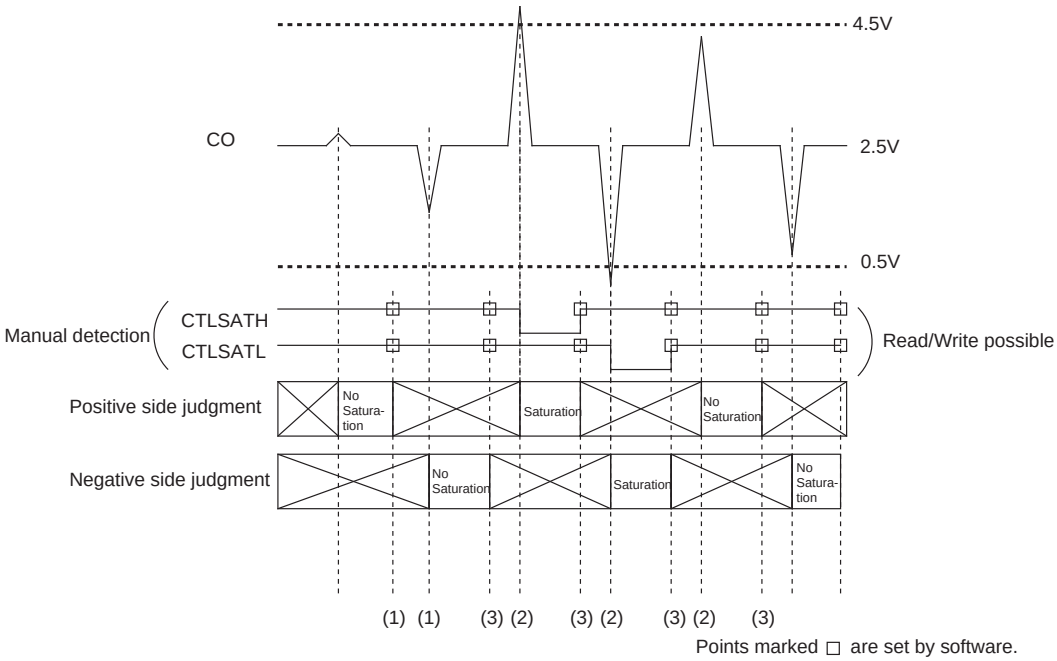


Figure 9-1-8

Method of use in the saturation detection mode

(1) Initializing the detection result saving flag (TEMPSATH, TEMPSATL)

The detection flags (CTLSATH, CTLSATL) are set by software at a timing synchronized with the CTL period, and preparations are made for saturation detection.

(2) Gain detection

The detection flags (CTLSATH, CTLSATL) are cleared when the saturation detection level (±2.0Vop) is exceeded when the CTL signal is input. (Saturation)

The detection flags are left in the set condition if the saturation detection level has not been reached. (No saturation)

(3) Judgment of result

The judgment of saturation/no saturation is made by reading the detection flags (CTLSATH, CTLSATL) at a timing synchronized to the CTL period. After the judgment is made, the detection flags (CTLSATH, CTLSATL) are set and preparations are made for the next saturation detection. At this time, the gain setting processing is made based on the previous saturation detection judgment result.

Steps (2) and (3) above are repeated to maintain the CTL signal level at the appropriate value.

The SATS flag is bit 3 of register ANACNT (x'3F81').

Gain Detection Function

The gain detection function is provided for carrying out automatic gain control (AGC) to maintain the CTL signal at the optimum amplitude level.
While only manual detection (initialization of the detection flag was implemented by software) and negative side detection was performed in the MN101D01 Series, automatic detection (the detection result is latched based on the DCTL signal) and positive side detection are supported from the MN101D02 Series onward.

(2) Gain Detection (SATS flag = '1', VISDTY flag = '0')

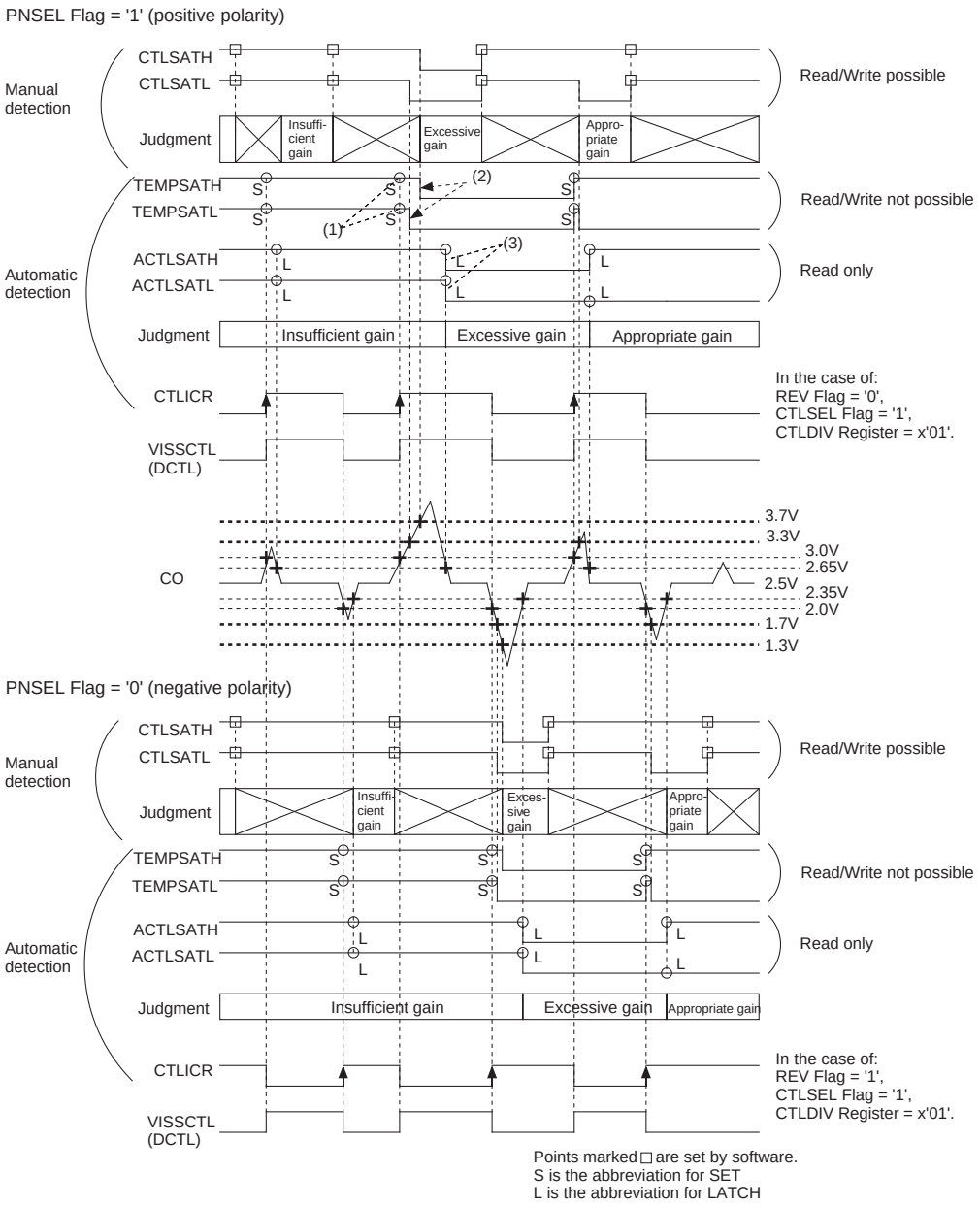


Figure 9-1-9

(3) Gain Detection (SATS flag = '1', VISDTY flag = '1', VISSLV flag = '0')

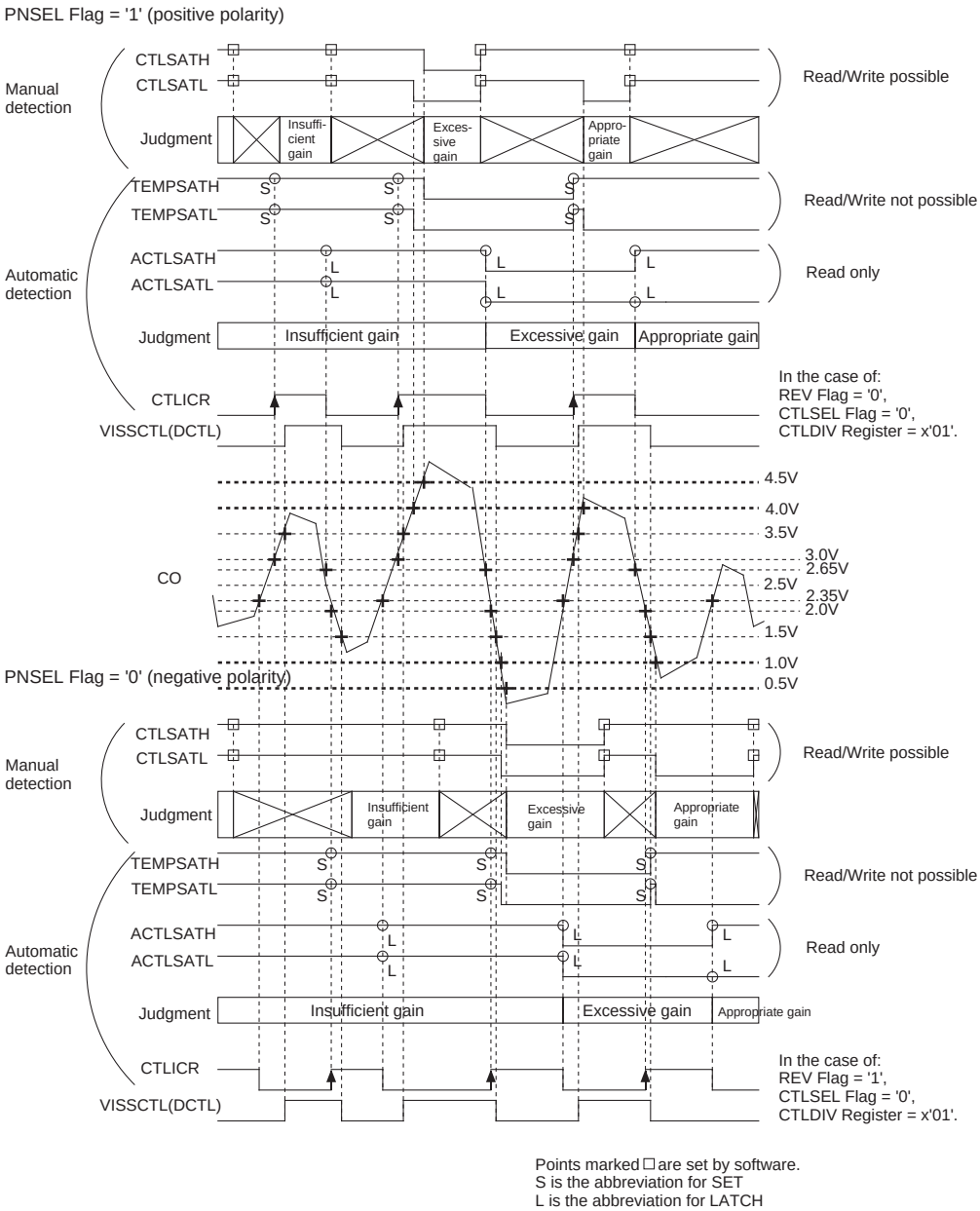


Figure 9-1-10

Operation in the Automatic Gain Detection Mode (VISDTY = '0', PNSEL = '1' in the case of positive pulse side detection)

(1) Initialization of detection result saving flags (TEMPSATH, TEMPSATL)

-> No initialization is done in the case of pulses that do not exceed 3.0V. (Even the automatic gain detection flags (ACTLSATH, ACTLSATL) remains unchanged.)

-> When a pulse of 3.0V or greater is input, the detection result saving flags (TEMPSATH, TEMPSATL) are initialized (set) when the positive pulse rising edge voltage is 3.0V.

(2) Gain detection

-> Pulses 3.0V or greater but less than 3.3V do not change the detection result saving flags. (Insufficient gain)

-> The detection result saving flag TEMPSATL is cleared when a pulse of 3.3V or greater but less than 3.7V is input. TEMPSATH remains set and does not change. (Appropriate gain)

-> The detection result saving flags (TEMPSATH, TEMPSATL) are both cleared when a pulse of 3.7V or greater is input. (Excessive gain)

(3) Gain detection result

The detection result saving flags are stored in the automatic gain detection flags (ACTLSATH, ACTLSATL) at the timing at which the falling edge voltage of a positive pulse of 3.0V or greater is 2.65V.

(4) Judgment of result

When a positive pulse of 3.0V or greater is input, a CTL interrupt is generated and at that timing, the automatic gain detection flags (ACTLSATH, ACTLSATL) are read in, and the judgment of insufficient gain/appropriate gain/excessive gain is made. Based on the result of this judgment, the gain setting processing is performed.

When gain control processing is to be done due to a CTL interrupt, the gain detection result of the previous event is read.

Since steps (1), (2), and (3) above are processed automatically by the hardware, it is sufficient to implement only step (4) by software.

A similar operation is performed when performing settings of VISDTY and PNSEL other than the above, except that the gain detection level and detection pulse polarity are different. (Description is omitted.)

Saturation detection control register (SATCNT: x'3F6A' R/W, partly R)

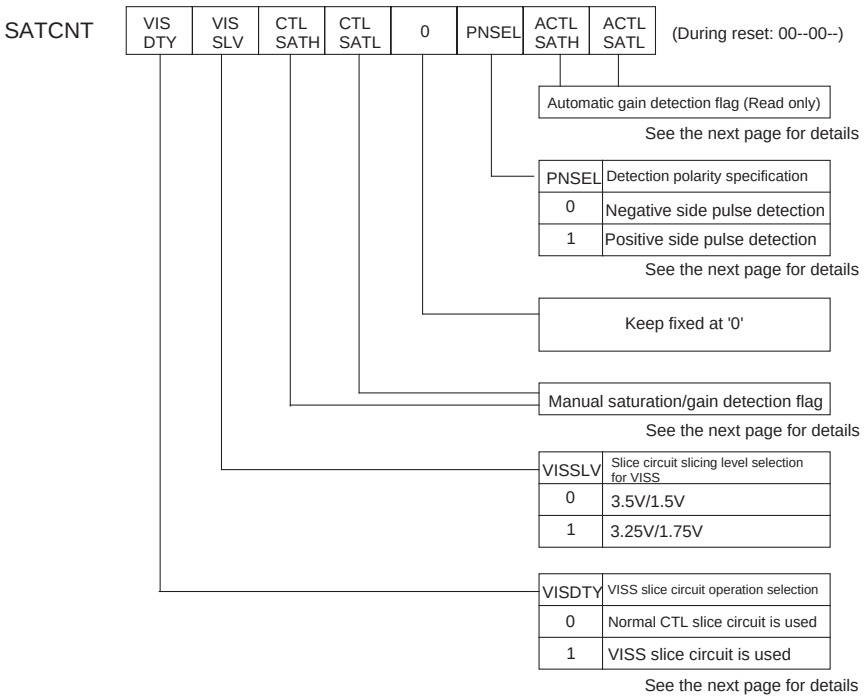


Figure 9-1-11

Supplementary Description of Saturation Detection Control Register (SATCNT: x'3F6A' R/W, partly R)

SATCNT

VIS DTY	VIS SLV	CTL SATH	CTL SATL	0	PNSEL	ACTL SATH	ACTL SATL

Saturation detection (SATS flag = '0')

* Can only be used during manual detection.

CTLSATL	Detection of CTL input of 0.5V or less (saturation)
0	CTL Output voltage≤0.5V(-2.0Vop) (saturation)
1	CTL Output voltage>0.5V(-2.0Vop) (no saturation)

CTLSATH	Detection of CTL input of 0.5V or less (saturation)
0	CTL Output voltage≥4.5V(+2.0Vop) (saturation)
1	CTL Output voltage<4.5V(+2.0Vop) (no saturation)

Gain detection (SATS flag = '1')

* Can be used during manual detection and automatic detection.

Setting		Gain judgment		Detection level	
VISDTY	PNSEL	CTLSATL, ACTLSATL	CTLSATH, ACTLSATH		
0	0 (Negative)	1	1	CTL Output of 1.7V (-0.8Vop) or more (insufficient gain)	-0.8Vop/-1.2Vop (1.7V/1.3V) Detection
		0	1	CTL Output of 1.7V to 1.3V (appropriate gain)	
		0	0	CTL Output of less than 1.3V (-1.2Vop)	
	1 (Positive)	1	1	CTL Output of 3.3V (+0.8Vop) or more (insufficient gain)	+0.8Vop/+1.2Vop (3.3V/3.7V) Detection
		0	1	CTL Output of 3.3V to 3.7V (appropriate gain)	
		0	0	CTL Output of less than 3.7V (+1.2Vop)	
1	0 (Negative)	1	1	CTL Output of 1.0V (-1.5Vop) or more (insufficient gain)	-1.5Vop/-2.0Vop (1.0V/0.5V) Detection
		0	1	CTL Output of 1.0V to 0.5V (appropriate gain)	
		0	0	CTL Output of less than 0.5V (-2.0Vop)	
	1 (Positive)	1	1	CTL Output of 4.0V (+1.5Vop) or more (insufficient gain)	+1.5Vop/+2.0Vop (4.0V/4.5V) Detection
		0	1	CTL Output of 4.0V to 4.5V (appropriate gain)	
		0	0	CTL Output of less than 4.5V (-2.0Vop)	

Automatic gain detection: The ACTLSATL/H flags are set when the input exceeds ±0.50Vop, and the gain detection result is latched in the ACTLSATL/H flags when the input falls below ±0.15V. The saturation detection cannot be made automatically.

Manual detection: It is necessary to positively set the CTLSATH/L flags by software before inputting the control pulse. The results of saturation/gain detection are immediately reflected in the CTLSATH/L flags.

Figure 9-1-12

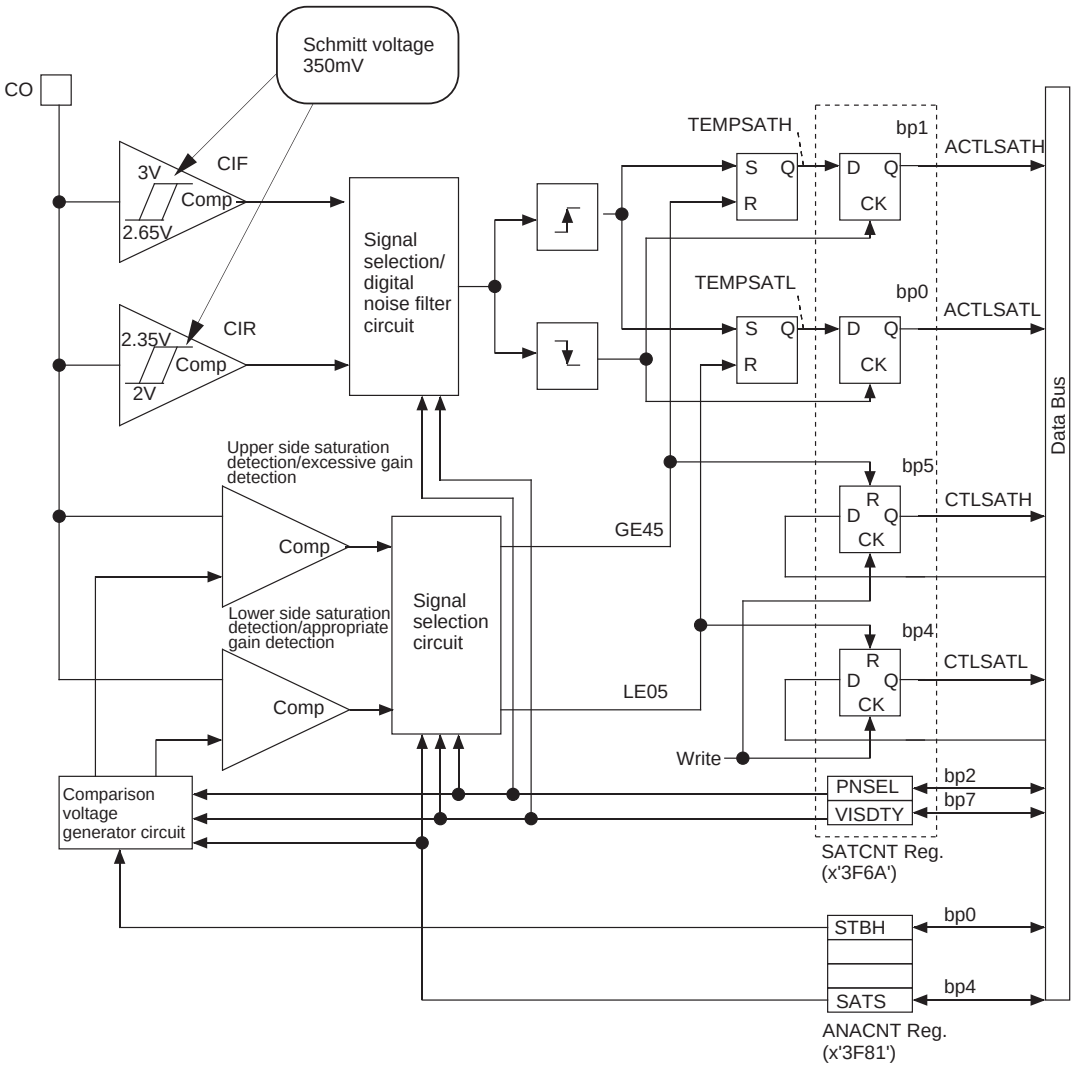
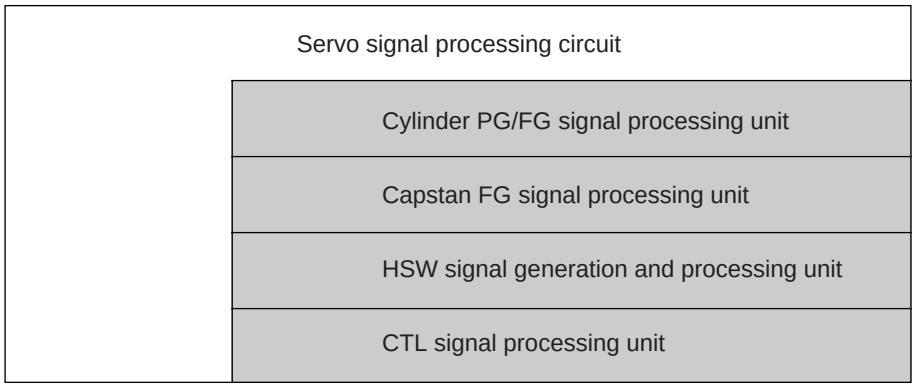


Figure 9-1-13 Saturation/Gain Detection Circuit Block Diagram

9-2 Servo Signal Processing Unit

9-2-1 Servo Signal Processing Circuit

The servo signal processing unit consists of the following individual processing units.



Cylinder PG/FG Signal Processing Unit

The cylinder PG/FG signal processing unit generates the YPGD and YFGD signals. If 3-value separation is used, the cylinder FG signal (YFGA) can be isolated by setting the PFGS Flag (ANACNT: bp 4 of x'3F81') to '0'. If the PFGS Flag is set to '1', the cylinder FG signal input from the YFG pin and the cylinder PG signal input from the YPG pin pass through their respective amps and are then output. Polarity of the cylinder PG signal can be switched with the PGPOL Flag (OPTION: bp6 of x'3F83'). Polarity of the cylinder FG signal can be switched with the PGPOL0 Flag.

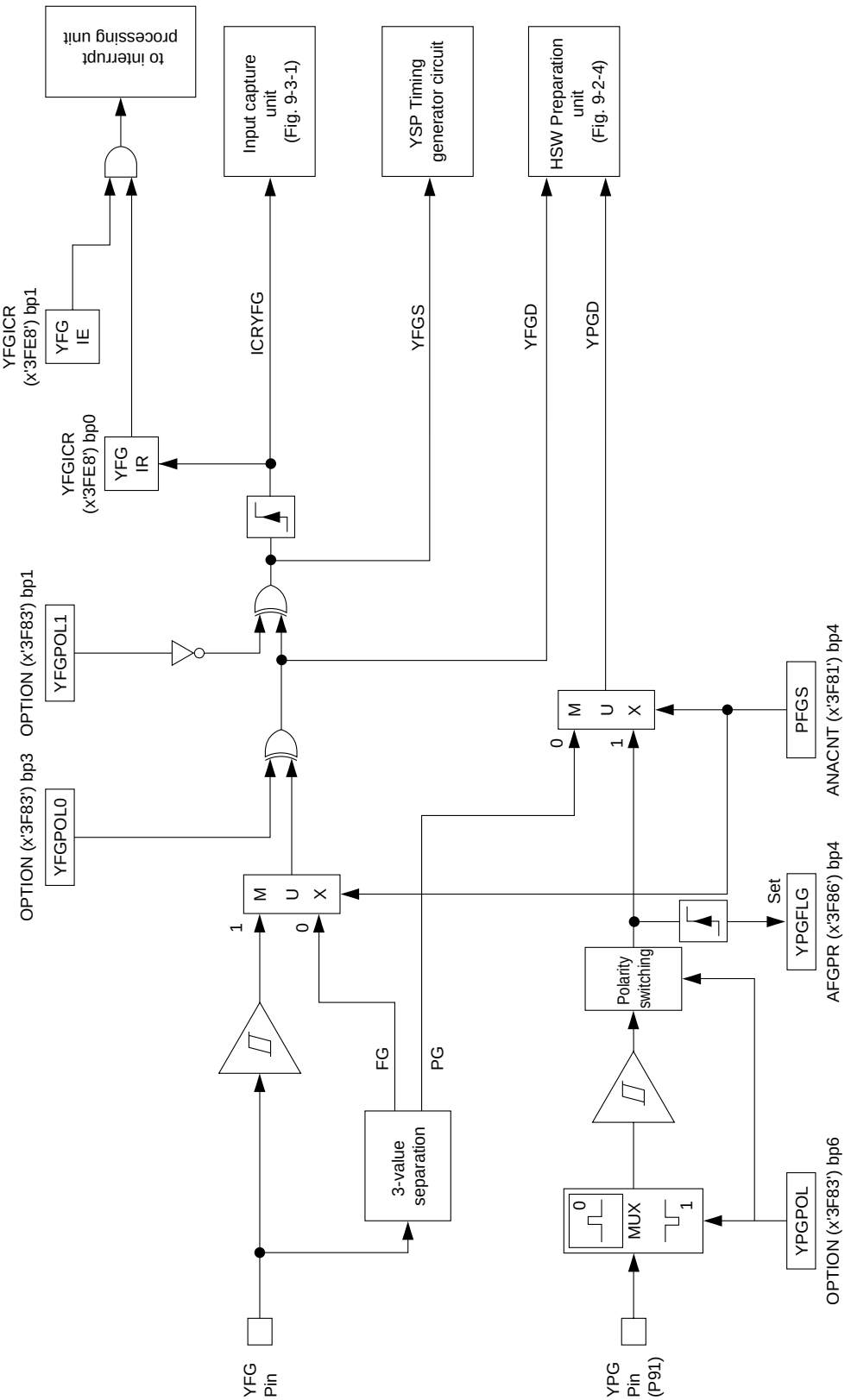


Figure 9-2-1 Cylinder PG/FG Signal Processing Unit

Capstan FG Signal Processing Unit

Capstan FG Signal Processing Unit

The capstan FG signal processing unit consists of a capstan FG prescaler, a capstan FG frequency divider circuit, a capstan FG mask timer circuit, and capstan FG interrupt and capture circuits.

Capstan FG Prescaler

The capstan FG prescaler is set to select whether the frequency of the capstan FG signal rectified by the FG amp and Schmitt amp is multiplied by 2 or divided by 2. This selection is made by the FGD0 (bp2) and AFGD1 (bp3) Flags of the AFG Control Register (AFGPR: x'3F86', R/W).

Table 9-2-1 Capstan FG Prescaler

AFGD0	AFGD1	Prescaler
0	0	1
1	0	Multiply frequency by 2
0	1	Divide frequency by 2
1	1	Prohibited

Capstan FG Frequency Divider Circuit

The capstan FG frequency divider circuit consists of the 7-bit AFG Frequency Divider Register (AFGDIV: x'3F87', R/W) and a 7-bit up-counter. Frequency division ratios from 1/1 to 1/127 can be set with the lower 7 bits of the AFGDIV Register (bp0 to bp6).

Capstan FG Mask Timer Circuit

After receiving the capstan FG signal, the capstan FG mask timer circuit disables input of the capstan FG signal for a fixed amount of time.

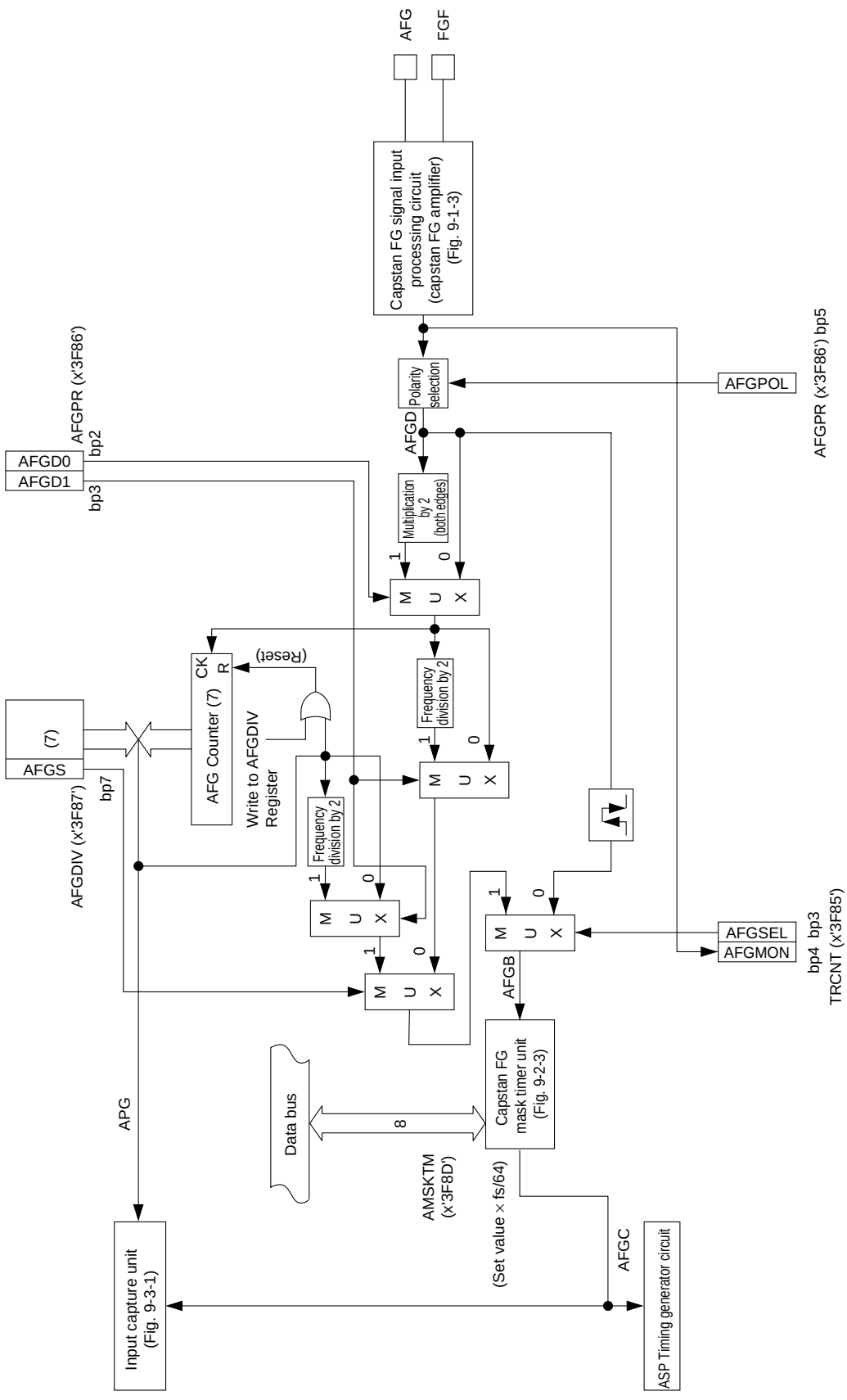


Figure 9-2-2 Capstan FG Signal Processing Unit

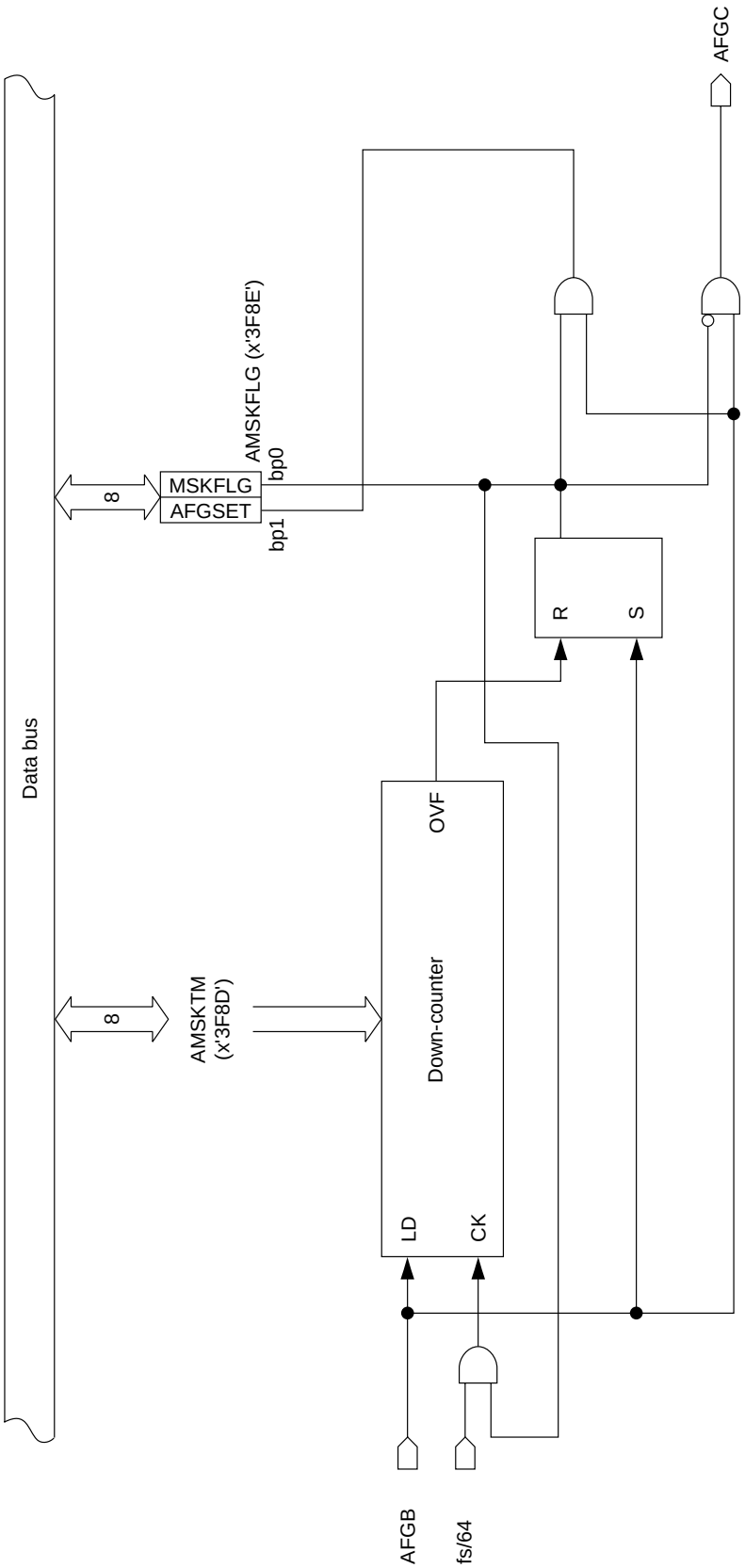


Figure 9-2-3 Capstan FG Mask Timer Unit

HSW Signal Generation and Processing Unit

HSW Signal Generation and Processing Unit

The HSW signal generation and processing unit consists of cylinder FG frequency divider and HSW signal generation circuits, a 2H compensating circuit, a cylinder PG frequency divide-by-3 circuit, and HSW interrupt and capture circuits.

Cylinder FG Frequency Divider and HSW Signal Generation Circuit

For example, if the cylinder FG signal frequency is 180 Hz (NTSC method), the setting value is '3'.

The cylinder FG frequency division ratio is set the lower 6 bits of the YFG Frequency Division Register (YFGDIV: x'3F88', R/W) so that the frequency of the frequency division output will result in the desired HSW signal frequency (NTSC: 30 Hz, PAL: 25 Hz). Bit 1 (FF15SEL) of the TRCNT Register (TRCNT: x'3F85', R/W one section is read only) can be set to specify that SPG synchronous output is output to the HSW pin.

2H Compensating Circuit

The 2H compensating circuit uses a binary counter to count a fixed 8-bit value, and generates an appropriate 2H delay interval with the fosc/8 clock.
The SIF2H Flag (bp7) of the YFG Frequency Division Register (YFGDIV: x'3F88', R/W) selects whether 2H compensation is to be performed.

Cylinder PG Frequency Division by 3 Circuit

Having been converted into a digital signal by 3-value separation or by the YPG amp, and after passing through a noise removal circuit, the PG signal frequency can be divided by 3. The YPG0 Flag (bp1) of the AFG Control Register (AFGPR: x'3F86', R/W) selects whether the cylinder PG signal will be divided by 3.

HSW Interrupt and Capture Circuits

The HSWIR Flag is not reset by the hardware during cycles when an interrupt is accepted.

Reset by software instructions. The HSW signal polarity can be determined by reading the HSWPOL Flag (bp6 of TRCNT).

At the HSW signal falling edge or at both edges (set by bit 5 of TRCNT: x'3F85'), a capture signal (HSWI) is generated. Set the HSWIR Flag (bp0) of the HSW Interrupt Register (HSWICR: x'3FF1', R/W) so that an interrupt request will be generated at both edges of the HSW signal. The HSWLV0 and HSWLV1 Flags (bp6 and bp7) can set the interrupt priority level.

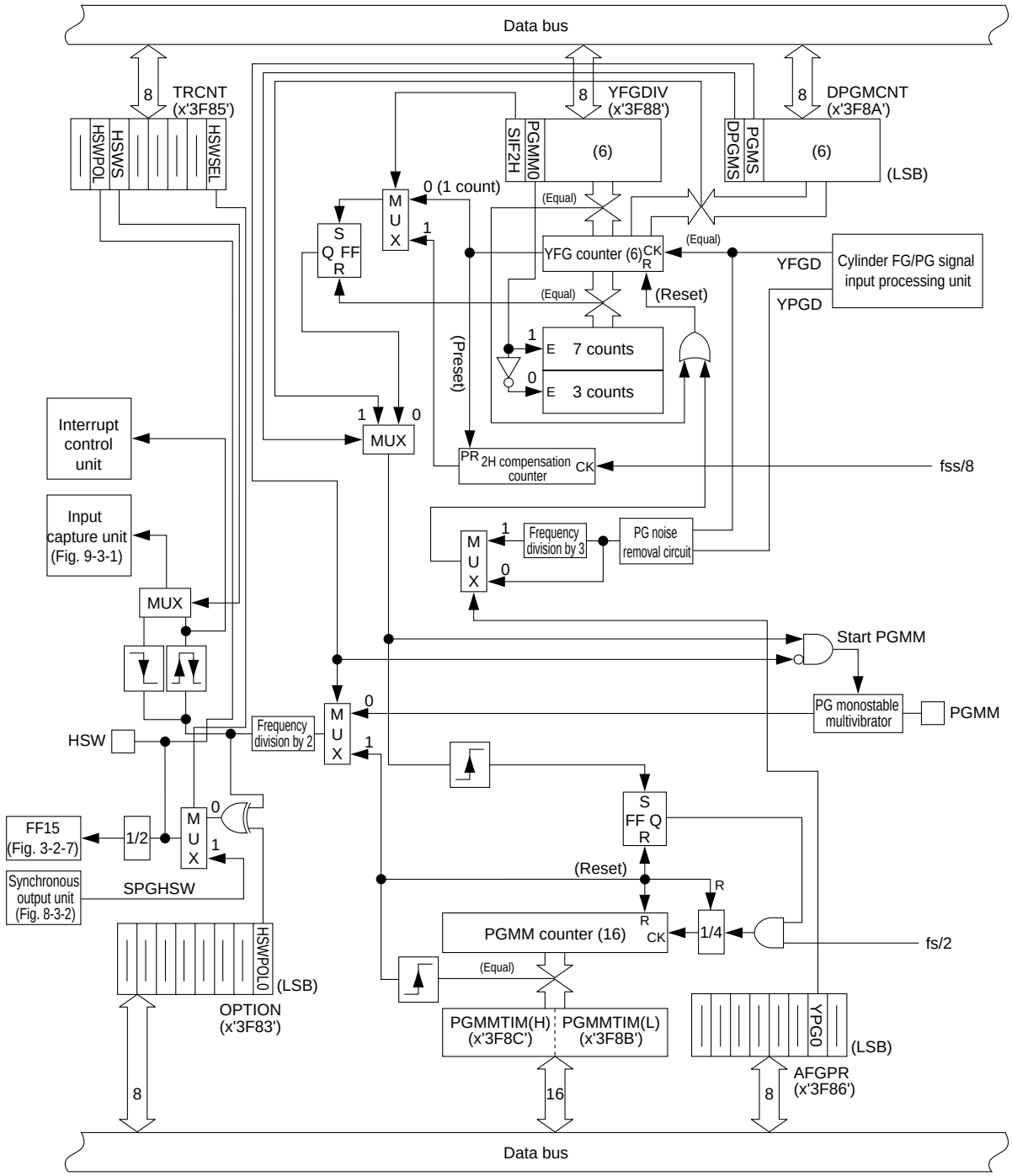


Figure 9-2-4 HSW Generator Unit Block Diagram

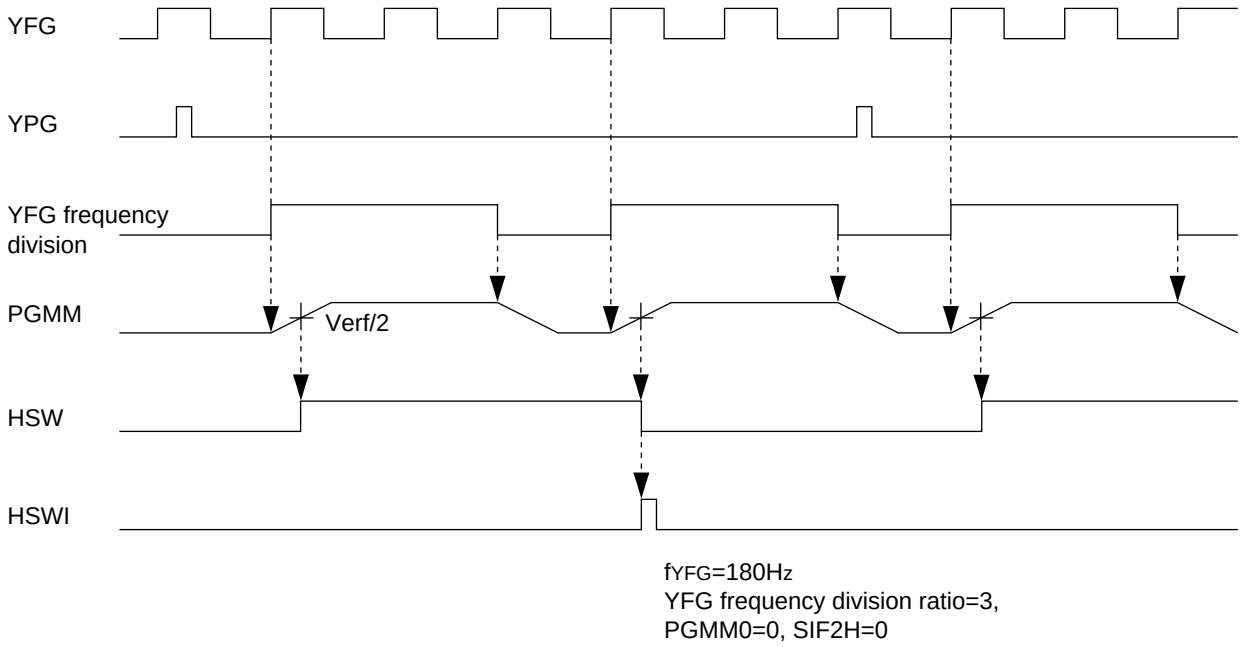


Figure 9-2-5 HSW Signal Preparation Timing Chart (without 2H correction)

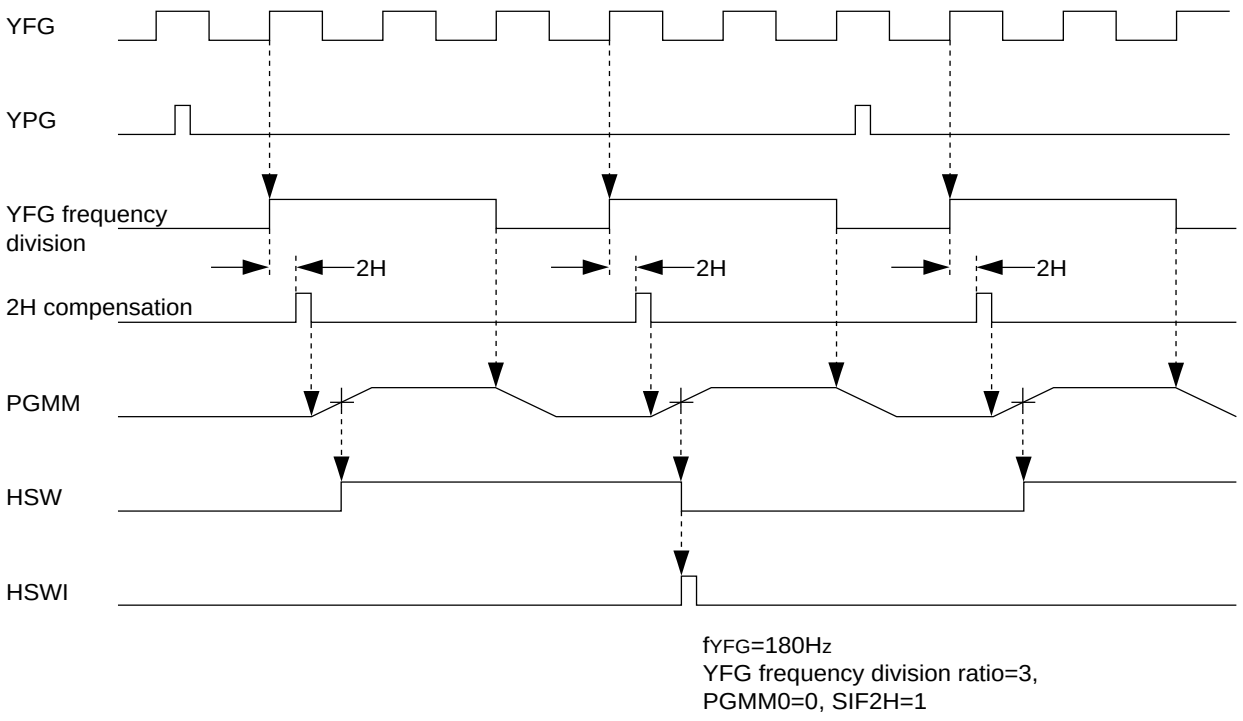


Figure 9-2-6 HSW Signal Preparation Timing Chart (with 2H correction)

CTL Signal Processing Unit

CTL Signal Processing Unit

The CTL signal processing unit consists of a PCTL signal generating circuit, a CTL noise removal circuit, a CTL frequency divider circuit, interrupt and capture processing circuits, and a DCTL signal generating circuit.

PCTL Signal Generating Circuit

This circuit creates the PCTL signal for the linear time count. The REVH Flag (bp2) of the Analog Control Register (ANACNT: x'3F81', R/W) selects either the CIF signal (the positive polarity slicing circuit output) or the CIR signal (the negative polarity slicing circuit output). The selected signal is output as the PCTL signal.

CTL Noise Removal Circuit

Only CTL pulses longer than 3 counts of the fs/64 clock are passed through. The PCTL signal passing through this circuit becomes the input signal for the CTL frequency divider circuit and becomes the input signal for the Timer 1 block.

CTL Frequency Divider Circuit

The CTL frequency divider circuit contains the CTL Register (CTLDIV: x'3F89', R/W) that sets the frequency division ratio in a 6-bit binary counter. Setting a value in the CTLDIV Register allows the frequency of the CTL signal to be divided. Also, setting a value in the CTLDIV Register will cause the CTL counter to be reset.

These flags are not reset during cycles when an interrupt has been accepted. Reset these flags with software instructions.

Interrupt and Capture Processing Circuit

The DREC Flag (bp0) of the AFG Control Register (AFGPR: x'3F89', R/W) selects whether the frequency divided CTL output or the frequency divided PG output will be used with the control signal interrupt and capture signal. The CTLIR Flag (bp0) of the Control Signal Interrupt Register (CTLICR: x'3FF0', R/W) is set according to the selected interrupt source. The CTLLV0 and CTLLV1 flags (bp6 and bp7) can set the interrupt priority level.

DCTL Signal Generating Unit

A DCTL signal that reproduces the duty information of the CTL signal can be generated by setting with the positive polarity signal and resetting with the negative polarity signal of the playback CTL signal. In the CTL input processing unit, a DCTL signal that reproduces the duty information of the original recording CTL signal can be generated by setting and resetting a RS flip-flop from the positive polarity slicing circuit output (CIF signal) and the negative polarity slicing circuit output (CIR signal).
This signal is input into the timer 2 block and makes VISS/VASS processing easy.

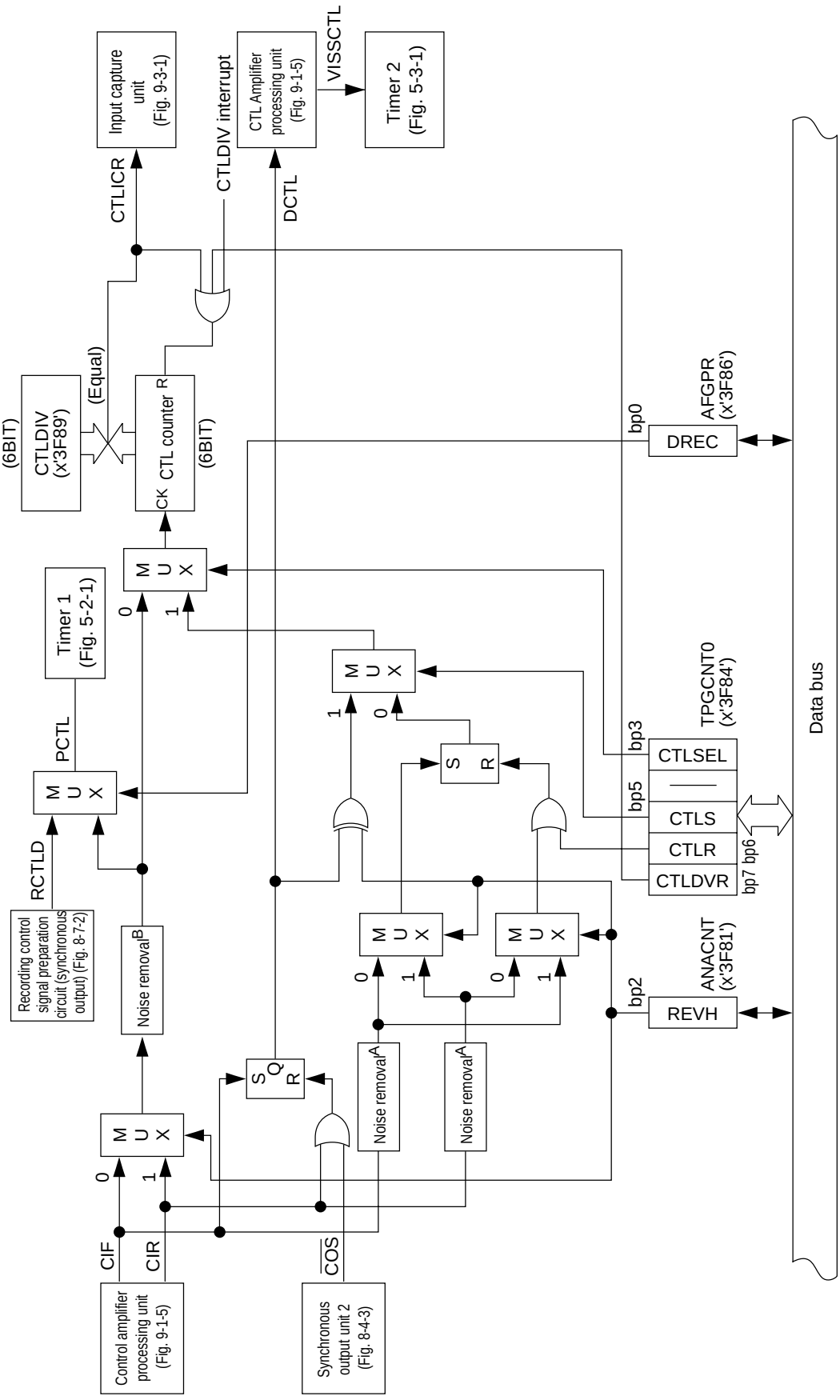


Figure 9-2-7 CTL Signal Processing Unit

VSYNC Input Processing Unit

SYNC Input Processing Unit

The SYNC input processing unit consists of a VSYNC signal separation unit, a field evaluation unit, a HSYNC separation unit, and a VSYNC mask unit.

(1) VSYNC Signal Separation Unit
This unit separates the VSYNC signal from a composite synchronous signal.

(2) Field Evaluation Unit
This unit evaluates the composite synchronous signal and determines whether it is an odd field or an even field.

(3) HSYNC Separation Unit
This unit separates the HSYNC signal from a composite synchronous signal and generates a pseudo-VSYNC signal.

(4) VSYNC Mask Unit
Functioning to reduce noise in a weak electric field, this unit realizes the mask function for the VSYNC signal and provides a backup when VSYNC is missing.

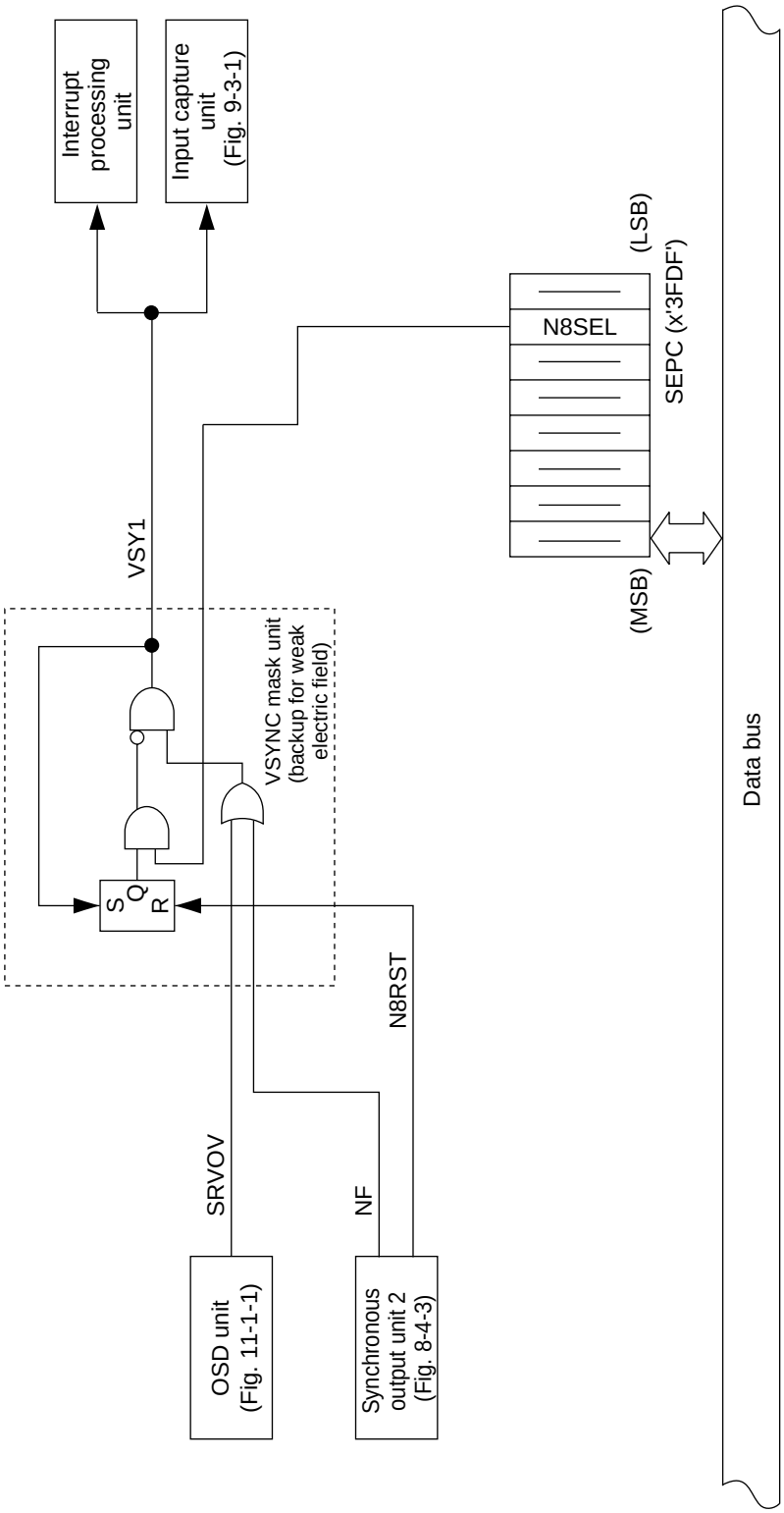


Figure 9-2-8 VSYNC Input Processing Unit

9-2-2 VSYNC Weak Electric Field Backup Function (NF, N8)

The count value of Free-Running Counter (FRC) is captured by the VSYNC Input Capture Register (ICRVYSN) at a time basically determined by the VSYNC signal, but the electromagnetic radiation of the VSYNC signal may generate noise, creating an unstable condition that causes the VSYNC signal to drop out. So that the servo will operate stably even with this type of unstable VSYNC signal, this LSI chip is equipped with the following two built-in hardware functions, a VSYNC noise mask (N8) function, and a backup function (NF) for when VSYNC is missing.

(1) The VSYNC noise mask (N8) signal masks the actual capture signals (VSY1 signal in Fig. 9-2-8) of the VSYNC Input Capture Register (ICRVSYN: x'2E0C', x'2E0D', x'2E0E', R). When the VSYNC signal or NF signal causes the FRC count value to be captured by the ICRVSYN Register, masking begins and capture operation is disabled. The mask can be removed by setting the N8RST Flag (bp4) of the Synchronous Output Data Register (HOCRBUF1: x'3F7B', R/W) to '1'.

(2) The VSYNC missing backup (NF) signal is a dummy signal used when the VSYNC signal is missing to capture the FRC count value in the ICRVSYN Register. The NF signal is generated when the value set in the Free-Running Counter Output Compare Register (FOCR0: x'2E16', x'2E17', R/W) becomes equal to the FRC count value.

Example: Method of Realizing VSYNC Mask (N8) and VSYNC Escape Countermeasure Function (NF)

(1) Specify the edge so that capture starts at the falling edge of HSW and make the setting so that the HSW both edge interrupt is generated.

(2) Specify the VSYNC to be used (sync separator section output VSYNC/ VSYNC pin input). (SVVSEL flag (bp0) of register SEPC (x'3FDF'))

Enable the VSYNC weak electric field countermeasure function (NF).
(Set the N8SEL flag (bp1) to '1' in register SEPC (x'3FDF').)

Make the setting so that the VSYNC mask release timing is realized by synchronous output.
(Set the HOCRSEL1 flag (bp2) to '1' in register HOCRCNT (x'3F7C').)

(3) The value of FRC is captured (capture value NICRVSYN(n)) using the VSYNC signal input or the VSYNC escape countermeasure (NF) signal, and the VSYNC signal mask signal (N8) is set. (Even if the VSYNC signal is input hereafter during the period up to mask release, the VSYNC capture operation and the VSYNC interrupt operation is inhibited.)

(4) Next, when the HSW both edge interrupt is generated, the HSWPOL flag is read and the rising edge or the falling edge of HSW is judged. Here, in the case of a falling edge, the VSYNC mask release timing (N8) is set.
FOCR1(n) = (NICRVSYNC(n)+NΔ1)/4
Example: Δ1=25.864ms (77.5%)
Also, the VSYNC escape countermeasure pulse generation timing (NF) is set.
FOCR0(n) = (NICRVSYNC(n)+NΔ2)/4
Example: Δ2=34.543ms (103.4%)

(5) The N8 signal becomes 'L' at the timing of FRC=FOCR1, the VSYNC mask is released, and the VSYNC signal can be accepted again.

(6) Here, if the VSYNC signal that has to be input normally becomes missing, the NF signal is generated at the timing of FRC=FOCR0 and is input as a dummy VSYNC signal.

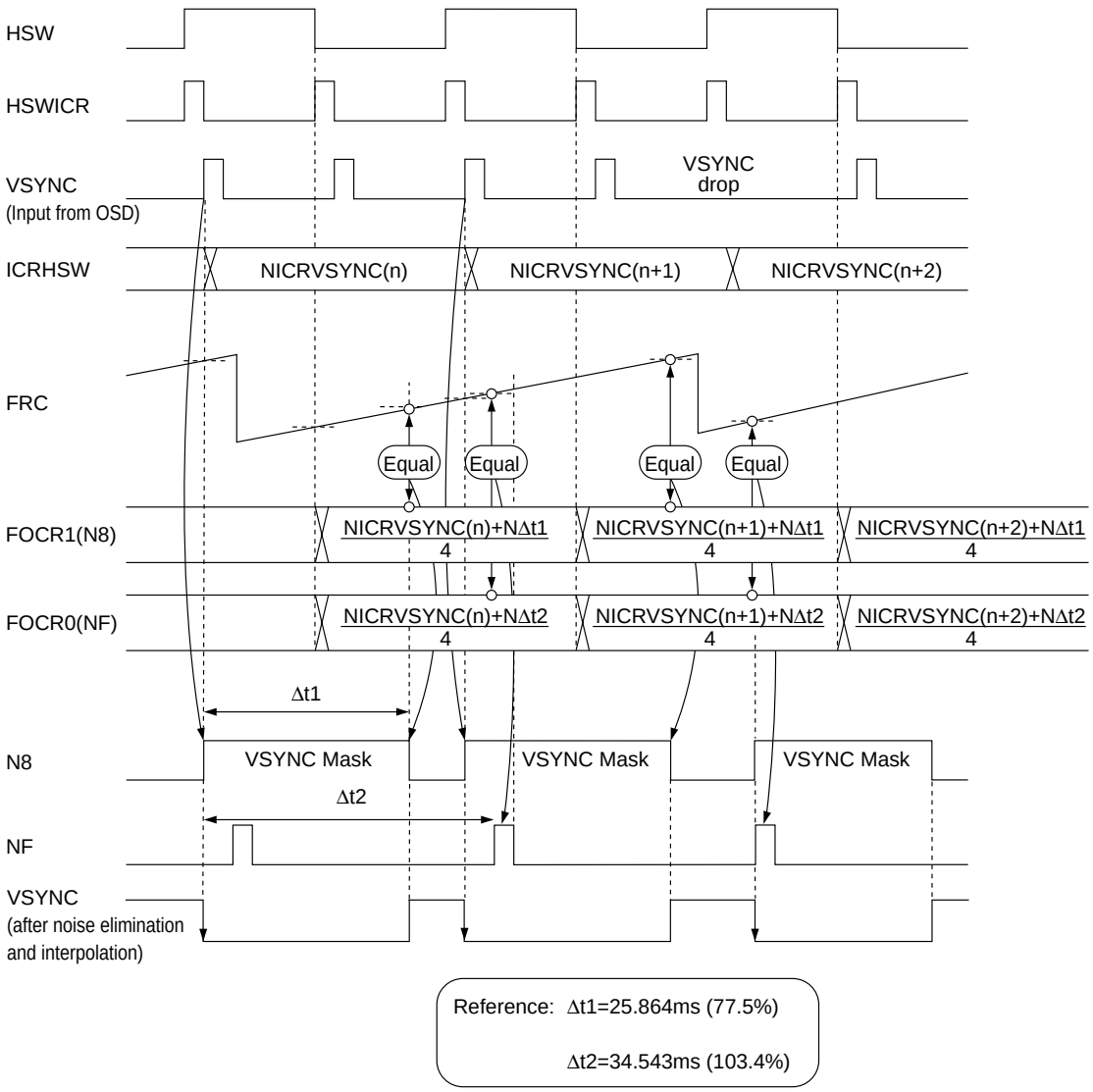


Figure 9-2-9 VSYNC Weak Electric Field Countermeasure Timing

9-3 Input Capture Unit

9-3-1 Input Capture Unit Configuration

The input capture unit consists of one 26-bit Free-Running Counter (FRC) that counts at fss/2 and six 18-bit Input Capture Registers.

9-3-2 Control Registers of the Input Capture Unit

Control registers of the input capture unit are listed below. [☞ Table 9-3-1]

In the NORMAL mode of fosc/2:
fss/2 = fs/2 = fosc/4 when servoclk = '0'; and
fss/2 = fs/4 = fosc/8 when servoclk = '1'.

Table 9-3-1 Control Registers of the Input Capture Unit

Register Name		RAM Address	R/W	Function
ICRAFG	ICRAFGL	x'2E00'	R	AFG Input Capture Register (lower)
	ICRAFGM	x'2E01'		AFG Input Capture Register (middle)
	ICRAFGH	x'2E02'		AFG Input Capture Register (upper)
ICRCTPG	ICRCTPGL	x'2E03'	R	CTL/PG Input Capture Register (lower)
	ICRCTPGM	x'2E04'		CTL/PG Input Capture Register (middle)
	ICRCTPGH	x'2E05'		CTL/PG Input Capture Register (upper)
ICRRCTLD	ICRRCTLDL	x'2E06'	R	RCTLD Input Capture Register (lower)
	ICRRCTLDM	x'2E07'		RCTLD Input Capture Register (middle)
	ICTRCTLDH	x'2E08'		RCTLD Input Capture Register (upper)
ICRHSW	ICRHSWL	x'2E09'	R	HSW Input Capture Register (lower)
	ICRHSWM	x'2E0A'		HSW Input Capture Register (middle)
	ICRHSWH	x'2E0B'		HSW Input Capture Register (upper)
ICRVSYN	ICRVSYNL	x'2E0C'	R	VSYNC Input Capture Register (lower)
	ICRVSYNM	x'2E0D'		VSYNC Input Capture Register (middle)
	ICRVSYNH	x'2E0E'		VSYNC Input Capture Register (upper)
ICRYFG	ICRYFGL	x'2E0F'	R	YFG Input Capture Register (lower)
	ICRYFGM	x'2E10'		YFG Input Capture Register (middle)
	ICRYFGH	x'2E11'		YFG Input Capture Register (upper)
FRCD	FRCDL	x'2E12'	R	Free-Running Counter Data Register (lower)
	FRCDM	x'2E13'		Free-Running Counter Data Register (middle)
	FRCDH	x'2E14'		Free-Running Counter Data Register (upper)
FRCLH		x'2E15'	R	Free-Running Counter Data Capture Control Register (upper)

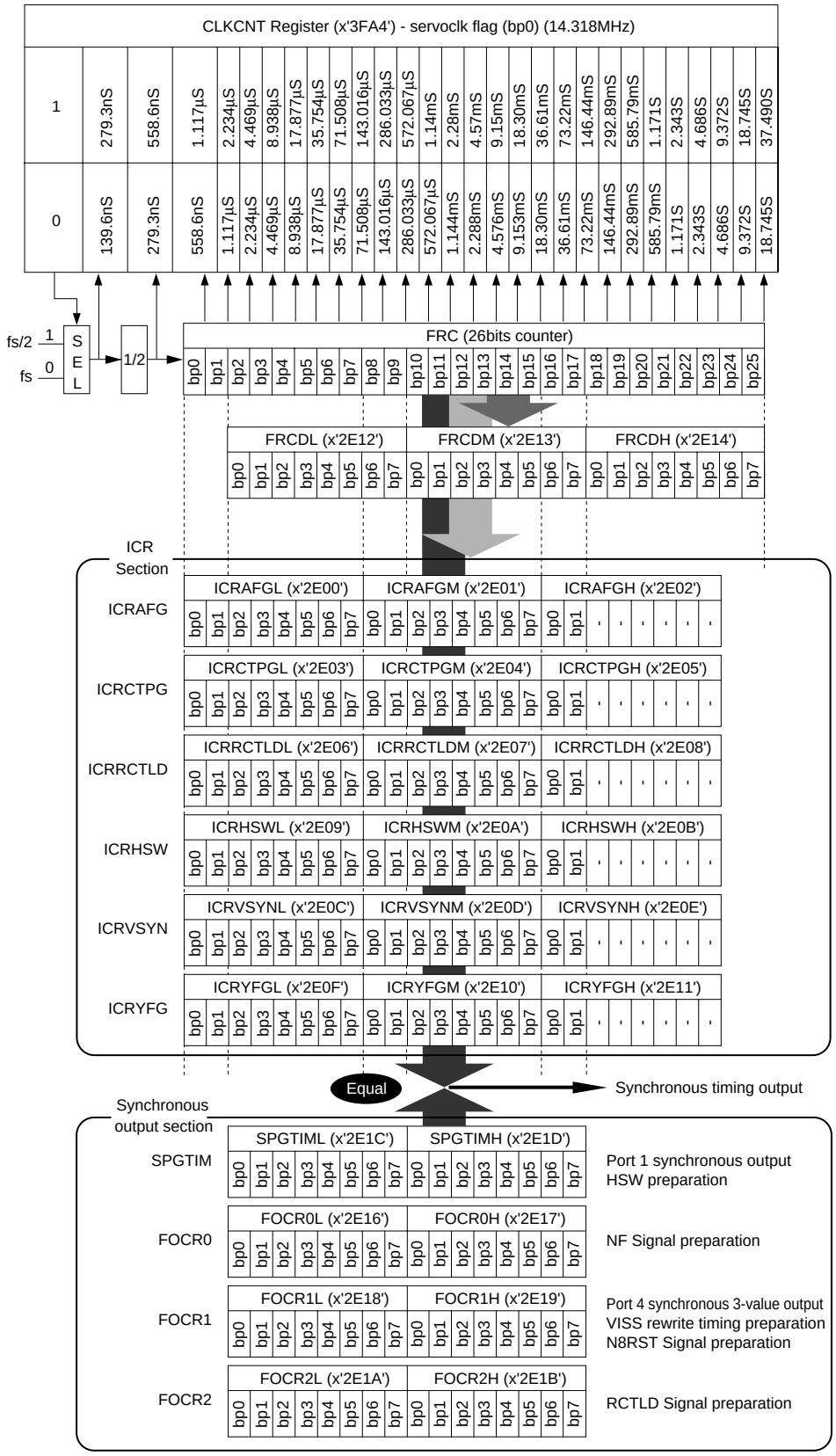


Figure 9-3-1 Diagram of List of Relationships with FRC

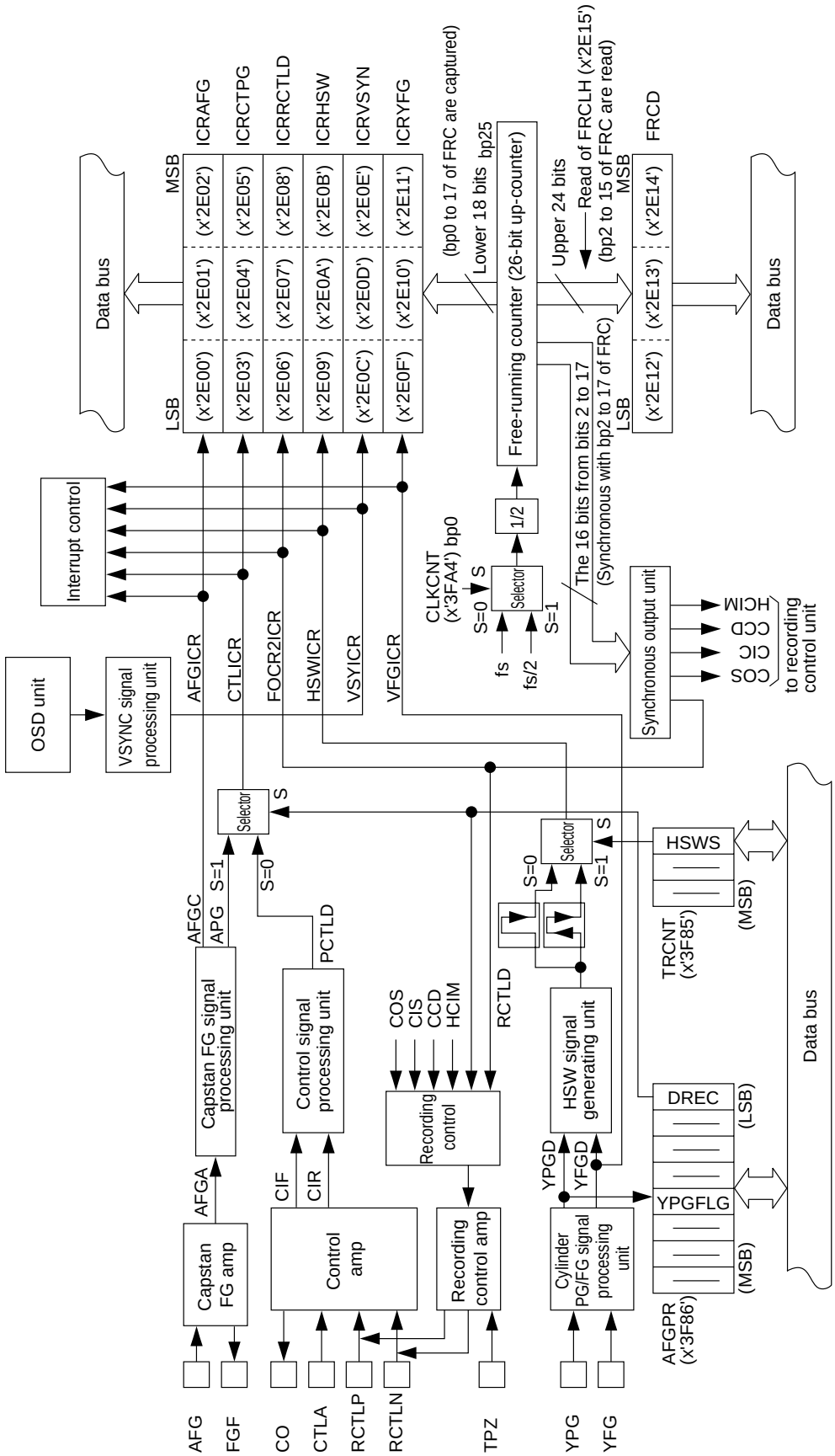


Figure 9-3-2 Input Capture Unit

Control registers of the input capture unit are described below.

(1) AFG Input Capture Register
(ICRAFG: x'2E00', x'2E01', x'2E02', R)

This read-only register captures signals with the capstan FG signal (AFGC).

(2) CTL/PG Input Capture Register
(ICRCTPG:x'2E03', x'2E04', x'2E05', R)

This read-only register captures signals during playback with the CTL signal and during recording with the capstan PG (APG: frequency divided AFG) signal.
The DREC Flag (bp0) of the AFG Control Register (AFGPR: x'3F86', R/W) switches both signals.

(3) RCTLD Input Capture Register
(ICRRCTLD: x'2E06', x'2E07', x'2E08', R)

This read-only register captures signals when the data set in the FOCR2 Register becomes equal to the Free-Running Counter.

(4) HSW Input Capture Register
(ICRHSW: x'2E09', x'2E0A', x'2E0B', R)

The HSW interrupt is always generated at both edges of HSW. (It is not possible to select single edge/both edge.)

This read-only register captures signals at both edges or the falling edge of the HSW signal.
The HSWS Flag (bp5) of the Tracking Control Register (TRCNT: x'3F85', R/W) switches both signals.

(5) VSYNC Input Capture Register
(ICRVSYN: x'2E0C', x'2E0D', x'2E0E', R)

This read-only register captures signals with the VSYNC signal or the timing (NF) set in the FOCR0 Register.

(6) YFG Input Capture Register
(ICRYFG: x'2E0F', x'2E10', x'2E11', R)

This read-only register captures signals with the cylinder FG signal.

(7) Free-Running Counter Data Register
(FRCD: x'2E12', x'2E13', x'2E14', R/W)

This register is used to read the value of the Free-Running Counter. Reading FRCLH (x'2E15', R) causes the value of the Free-Running Counter to be transferred to FRCD.

(8) Free-Running Counter Data Capture Control Register
(FRCLH: x'2E15', R)

If an instruction to read this register is executed, the value of the Free-Running Counter will be transferred to FRCD.

Do not write to FRCLH. If FRCLH is written to, the value in FRCD will be transferred to the Free-Running Counter.

9-3-3 Input Capture Interrupt

An input capture interrupt is generated when the value of the Free-Running Counter (FRC) is captured by an Input Capture Register. The generated interrupt differs depending upon which Input Capture Register has captured the data. Interrupt registers are listed below, corresponding to the register that captured the FRC data.

Although it is possible to specify falling edge or both edge for HSW capture, the HSW interrupt is always generated at both edges.

Table 9-3-2 Interrupt Registers Corresponding to Data Capture Registers

Register that Captured Data	Corresponding Interrupt Register
ICRAFG	AFGICR
ICRCTPG	CTLICR
ICRRCTLD	FOCR2ICR
ICRHSW	HSWICR
ICRVSYN	VSYICR
ICRYFG	YFGICR

When manipulating the interrupt request flag (xxxIR), it is necessary to set beforehand the IR write enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, reset IRWE to '0' after the manipulation.

The input capture interrupt is enabled through the following procedure.

(1) Interrupt priority is specified in flags LV0 and LV1 (bp6, 7) of the interrupt control register (xxxICR) corresponding to input capture, and the interrupt request flag is cleared by set flag xxxIR (bp0) to '0'.

(2) The interrupt enable setting is made by setting flag xxxIE (bp1) to '1' of the interrupt control register (xxxICR) corresponding to input capture.

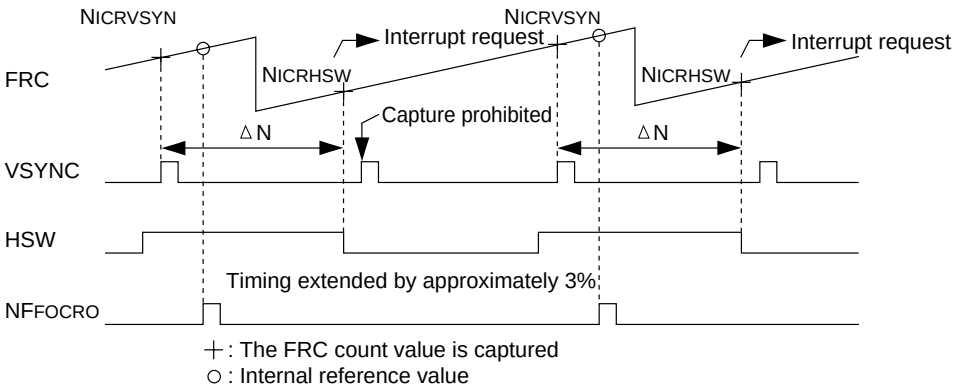
9-3-4 Cylinder and Capstan Phase Control

Four examples of cylinder and capstan phase control are presented using input capture registers.

Example 1) During recording, phase difference is controlled to be constant between the cylinder phase control vertical synchronization signal (VSYNC) and the Head Switch Signal (HSW) for a VCR. [☞ Figure 9-3-3]

The HSWS Flag (bp5) of TRCNT (x'3F85') can be set to select the HSW capture signal.
0: falling edge of HSW signal
1: both edges of HSW signal
The HSW signal can be monitored by the HSWPOL Flag (bp 6 of TRCNT).

At the rising edge of the VSYNC signal, the FRC value is captured to the VSYNC Input Capture Register (ICRVSYN). Then, at the falling edge of the HSW signal, the FRC value is captured to the HSW Input Capture Register (ICRHSW). At the same time as the ICRHSW Register captures this data, the HSWIR Flag (bp0) of the HSW Input Capture Interrupt Register (HSWICR) is set and an interrupt is requested.
The HSWS Flag (bp5) of TRCNT (x'3F85') can be set to select the HSW capture signal.



The phase difference, Δ N, from the rising edge of VSYNC until the falling edge of HSW is:

$$\Delta N = NICRHSW - NICRVSYN$$

the phase error, E, is:

$$E = \left(\frac{1}{2fr} - Trec \right) \times \left(\frac{fss}{16} - \Delta N \right)$$

where:
Trec : desired phase difference between HSW and VSYNC during recording
fr : frame frequency
fss : frequency of clock source

The value, NFOCR0, set in the FOCR0 Register is:

$$NFOCR0 = NICRVSYN + \frac{1.03}{fr} \times \frac{fss}{16}$$

Figure 9-3-3 Cylinder Phase Control during Recording

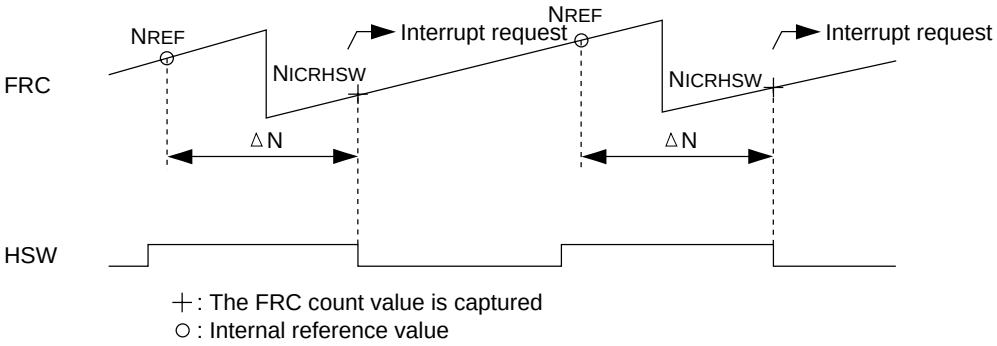
When the VSYNC signal is missing due to a weak electric field or other cause, the NF signal is output at the timing set in the FOCR0 Register. The NF signal may be used instead of the VSYNC signal to capture data in the ICRVSYN Register.

The HSWS Flag (bp5) of TRCNT (x'3F85') can be set to select the HSW capture signal.
0: falling edge of HSW signal
1: both edges of HSW signal
The HSW signal can be monitored by the HSWPOL Flag (bp 6 of TRCNT).

Example 2) Cylinder Phase Control during Playback

During playback, phase difference between the FRC reference signal and the HSW signal is controlled to be constant. [☞ Figure 9-3-4]

Set the FRC internal reference value (N_{REF}). Then, at the falling edge of the HSW signal, the FRC value is captured to the HSW Input Capture Register (ICRHSW). At the same time as the ICRHSW Register captures this data, the HSWIR Flag is reset and an interrupt request is generated.



The FRC internal reference value (N_{REF}) is set as follows. The phase difference, ΔN, from the internal reference value (N_{REF}) until the falling edge of HSW is:

$$\Delta N = N_{ICRHSW} - N_{REF}$$

the phase error, E, is:

$$E = \left(\frac{1}{2fr} - Trec\right) \times \left(\frac{fss}{2} - \Delta N\right)$$

where:
Trec : desired phase difference between HSW and the internal reference value during recording
fr : frame frequency
fss : frequency of clock source

The value set for the internal reference, N_{REF}, is:

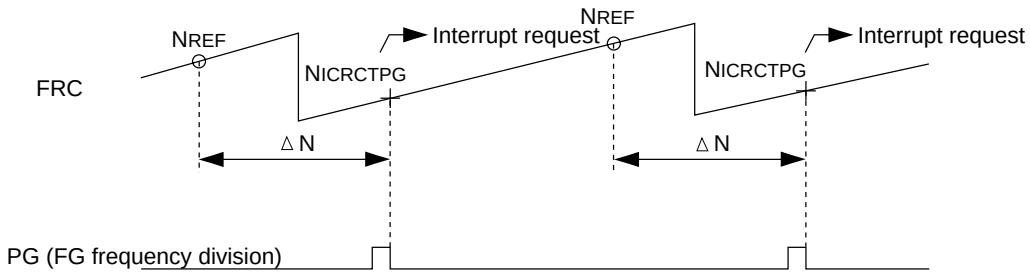
$$N_{REF} \leftarrow N_{REF} + \frac{1}{fr} \times \frac{fss}{2}$$

Figure 9-3-4 Cylinder Phase Control during Playback

Example 3) Capstan Phase Control during Recording

During recording, the capstan phase control operates to control the phase difference between the FRC reference value and the capstan PG signal (frequency divided capstan FG signal: APG) to be constant. [☞ Figure 9-3-5]

Set the FRC internal reference value (N_{REF}). Then, at the rising edge of the capstan PG signal, the FRC value is captured to the CTL/PG Input Capture Register (ICRCTPG). At the same time as the ICRCTPG Register captures this data, the CTL Interrupt Request Flag is set and an interrupt request is generated. The next internal reference value (N_{REF}) is also set at this time.



+ : The FRC count value is captured
○ : Internal reference value: NREF

The FRC internal reference value (NREF) is set as follows. The phase difference, △N, from the internal reference value (NREF) until the rising edge of the PG signal is:

$$\triangle N = N_{ICRCTPG} - N_{REF}$$

where:
f_{PG} : period of the PG signal
fss : frequency of clock source

The value set for the internal reference, NREF, is:

$$N_{REF} \leftarrow N_{REF} + \frac{1}{f_{PG}} \times \frac{fss}{2}$$

Figure 9-3-5 Capstan Phase Control during Recording

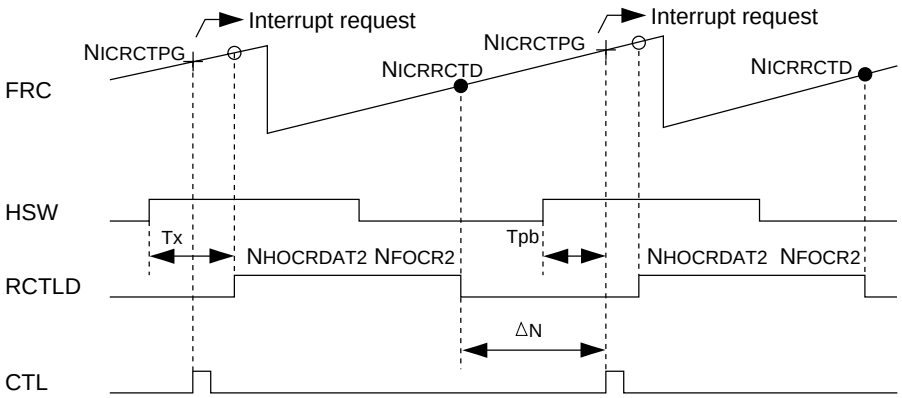
Example 4) Capstan Phase Control during Playback

During playback, the capstan phase control generates the RCTLD signal at the synchronous output unit with the HSW signal as a reference. [☞ Figure 9-3-6]

The CTL/PG Input Capture Register (ICRCTPG) captures data when triggered by the CTL signal during playback and the capstan PG signal during recording. These two signals can be switched by setting the DREC Flag (bp0) of the AFG Control Register (AFGPR: x'3F86', R/W).
'1': capstan PG signal (recording)
'0': CTL signal (playback)

At the falling edge of the RCTLD signal, the FRC value is captured to the RCTLD Input Capture Register (ICRRCTLD). Next, the signal that has been divided from CTL signal at the CTL signal processing unit (PCTLD) causes the FRC value to be capture and stored in the CTL/PG Input Capture Register (ICRCTPG). At the same time as the ICRCTPG Register captures this data, the CTLIR flag is set and an interrupt request is generated.

High or low levels of the HSW signal can be verified by monitoring bit 6 of TRCNT (x'3F85').



+ : FRC count value is captured
◯ : HOCRDAT2 matches FRC
● : FOCR0 matches FRC, FRC count value is captured

The phase difference, ΔN, from the falling edge of the RCTLD signal until the rising edge of the CTL signal is:

$$\Delta N = N_{ICRCTPG} - N_{ICRRCTD}$$

the phase error, E, is:

$$E = \left(\frac{1}{fr} - TRMM(C) \quad Tx + Tpb \right) \times \frac{fss}{2} - \Delta N$$

where:

Trec	:	desired phase difference between HSW and VSYNC during recording
TRMM (C)	:	center value of the tracking monostable multivibrator
Tx	:	X value shift
Tpb	:	desired phase difference between HSW and CTL during playback
fss	:	frequency of clock source

Figure 9-3-6 Capstan Phase Control during Playback

9-4 PWM Unit

9-4-1 PWM Unit Configuration

The PWM unit consists of 5 PWM channels: PWM0 for cylinder control, PWM1 for capstan control, PWM2 and PWM3 for general-purpose PWM, and PWM14 for the tuner.

PWM0 and PWM1 are 13-bit PWM channels. To generate basic components with the upper 8 bits and added bits with the lower 5 bits, PWM is repeated at a frequency of fosc/2^{13}, and basic components are converted at a frequency of fosc/2^{8}.
[Block Diagram, ☞ Figure 9-4-5, Figure 9-4-6]
[PWM Output Waveform, ☞ Figure 9-4-1]

PPWM2 and PWM3 are both 10-bit PWM channels that generate basic components with the upper 6 bits and added bits with the lower 4 bits. PWM is repeated at a frequency of fosc/2^{10} and basic components are converted at a frequency of fosc/2^{6}.
[Block Diagram, ☞ Figure 9-4-7, Figure 9-4-8]
[PWM Output Waveform, ☞ Figure 9-4-2]

If the program directly writes output data to the PWM Register, the PWM can output the desired waveform.

14-bit PWM is output from the PWM14 pin. The PWM14 pin also functions as the P25 (Port 25) pin. The function of this pin is switched by the 14-bit PWM Control Register (PWMCNT: x'3F78', R/W), the Port 2 Direction Control Register (P2DIR: x'3F30', R/W) and the Port 2 Selection Register (P2SEL: x'3F39', R/W).

Table 9-4-1 Summary of PWM Specifications

	PWM0	PWM1	PWM2	PWM3	PWM14
Total number of bits	13	13	10	10	14
Number of basic component bits	8	8	6	6	8
Number of added bits	5	5	4	4	6
PWM conversion frequency (basic components)	fosc/2^8	fosc/2^8	fosc/2^6	fosc/2^6	fosc/2^8
PWM repetition frequency	fosc/2^{13}	fosc/2^{13}	fosc/2^{10}	fosc/2^{10}	fosc/2^{14}
Data register	x'3F71', x'3F70'	x'3F73', x'3F72'	x'3F75', x'3F74'	x'3F77', x'3F76'	x'3F6F', x'3F6E'
Comments	For cylinder control	For capstan control			When PWM14S='0'

9-4-2 PWM Unit Output Control

PWM0

PWM0 is a 13-bit PWM for control use and is configured with 8 bits for basic components and 5 bits for added bits.
This port is a High-Z output immediately after reset.
The output polarity of PWM0 can be selected by setting the PP0 flag of the PWM control register (PWMCNT: x'3F78', bp0).
The output of PWM0 is enabled when the PWM01OE flag (PWMCNT: x'3F78', bp7) is set to '1'. Conversion data for PWM0 must be set directly.

PWM1

PWM1 is a 13-bit PWM for control use and is configured with 8 bits for basic components and 5 bits for added bits.
This port is a High-Z output immediately after reset.
The output polarity of PWM1 can be selected by setting the PP1 flag of the PWM control register (PWMCNT: x'3F78', bp1).
The output of PWM1 is enabled by setting the PWM01OE flag (PWMCNT: x'3F78', bp7) to '1'. Conversion data for PWM1 must be set directly.

PWM2

PWM2 is a 10-bit PWM for control use and is configured with 6 bits for basic components and 4 bits for added bits. The PWM2 functions as both SBUFD1 and P11 pins. Therefore, to output PWM2, the P11SEL Flag (P11SEL: x'3F38', bp1) of the Port 1 Selection Register, and the Port 1 Direction Control Register (P1DIR: x'3F2F', bp1) must be set to '1'. Also, the output polarity can be selected by setting the PP2 flag of the PWM control register (PWMCNT: x'3F78', bp2). Conversion data for PWM2 must be set directly.

PWM3

PWM3 is a 10-bit PWM for control use and is configured with 6 bits for basic components and 4 bits for added bits. The PWM3 functions as both SBUFD2 and P12 pins. Therefore, to output PWM3, the P12SEL Flag (P11SEL: x'3F38', bp2) of the Port 1 Selection Register, and the Port 1 Direction Control Register (P1DIR: x'3F2F', bp2) must be set to '1'. Also, the output polarity can be selected by setting the PP3 flag of the PWM control register (PWMCNT: x'3F78', bp3). Conversion data for PWM3 must be set directly.

PWM14

When setting output data in the PWMn registers, use a MOVW instruction to simultaneously set the PWMnL and PWMnH registers.

PWM4 is a 14-bit PWM for use with a tuner and is configured with 8 bits for basic components and 6 bits for added bits. PWM14 shares pin P25. Therefore, to output PWM14, it is necessary to set the Port 2 selection register (P2SEL: x'3F39', bp5) and the Port 2 direction control register (P2DIR: x'3F30', bp5) to '1'.
Also, the output polarity can be selected by setting the PPWM14 flag of the PWM control register (PWMCNT: x'3F78', bp6). Conversion data for PWM14 must be set directly.

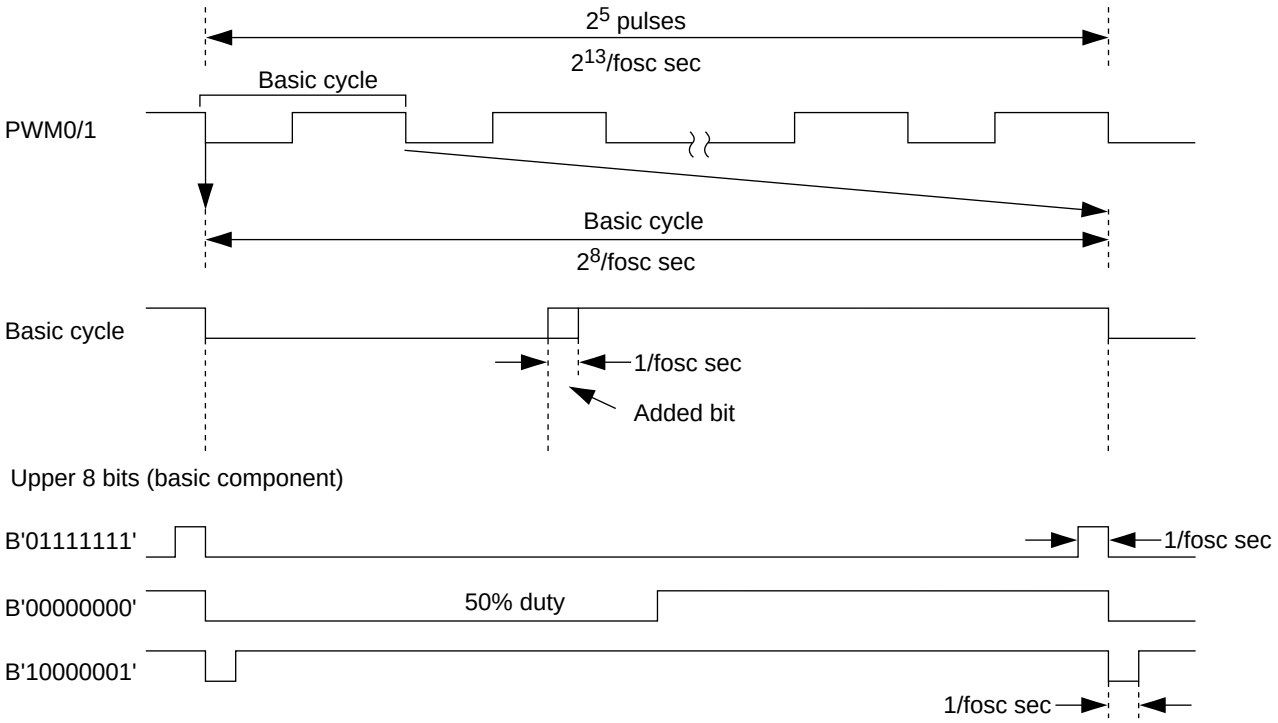


Figure 9-4-1 PWM0 and PWM1 Output Waveform

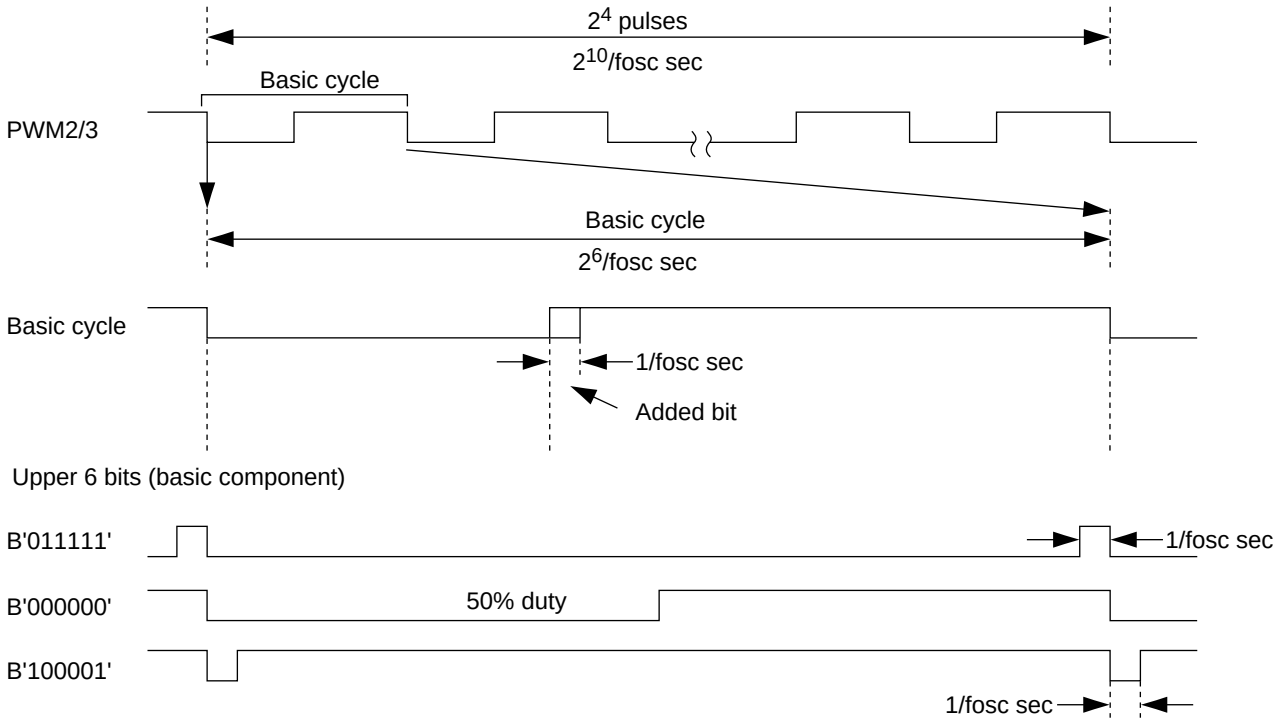


Figure 9-4-2 PWM2 and PWM3 Output Waveform

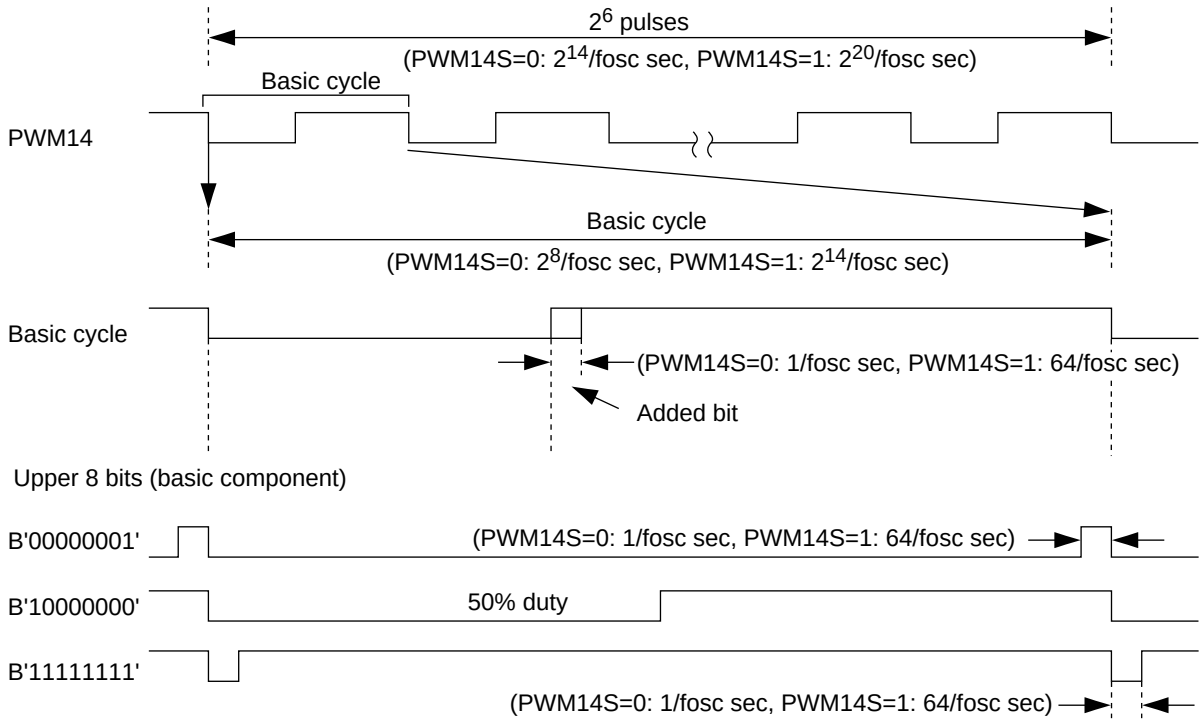


Figure 9-4-3 14-Bit PWM Output Waveform

9-4-3 PWM14 Settings

Perform the following setting procedure to output 14-bit PWM from pin 35.

1. Set P2SEL (bp5) of P2SEL (x'3F39') to '1'.
2. Set P2DIR5 (bp5) of P2DIR (x'3F30') to '1'.

When setting the output data, use a MOVW instruction to simultaneoulsy set the PWM14L and PWM14H registers.

Settings for output polarity and fixed forced output are listed below.

Table 9-4-2 Output Polarity and Fixed Forced Output

PPWM14 (bp6 of PWMCNT)	PWM14SEL (bp7 of PWM14H)	PWM14H13 (bp5 of PWM14H)	Output level of 14-bit PWM
0	1	0	Fixed LOW
0	1	1	Fixed HIGH
1	1	0	Fixed HIGH
1	1	1	Fixed LOW

When the PWM14H register (bp6) is set to '1', the 6 added bits must be set to x'00'.

The following two types of usage are possible depending on the setting of the PWM14S flag (bp6 of register PWM14H).

Table 9-4-3 PWM14 clock source setting

PWM14S flag	PWM14 clock source	PWM14 basic period (seconds)	A/D Interrupt source	Remarks	Application
0	fosc	2^8/fosc	After end of A/D conversion		Tuner control, etc.
1	fosc/64	2^{14}/fosc	At every PWM14 basic period	Set PWM14L5 to PWM14L0 to '0'.	FL Dimmer control, etc.

9-4-4 Method of Adding Bits

The method of adding bits is to add one bit of output to the basic component PWM output of n bits. Precise control is possible depending upon where the bit is added among the m number of PWM repetitions.

To shorten the basic cycle of PWM output and to obtain high precision output, PWM0 to PMW3 use added bits method.

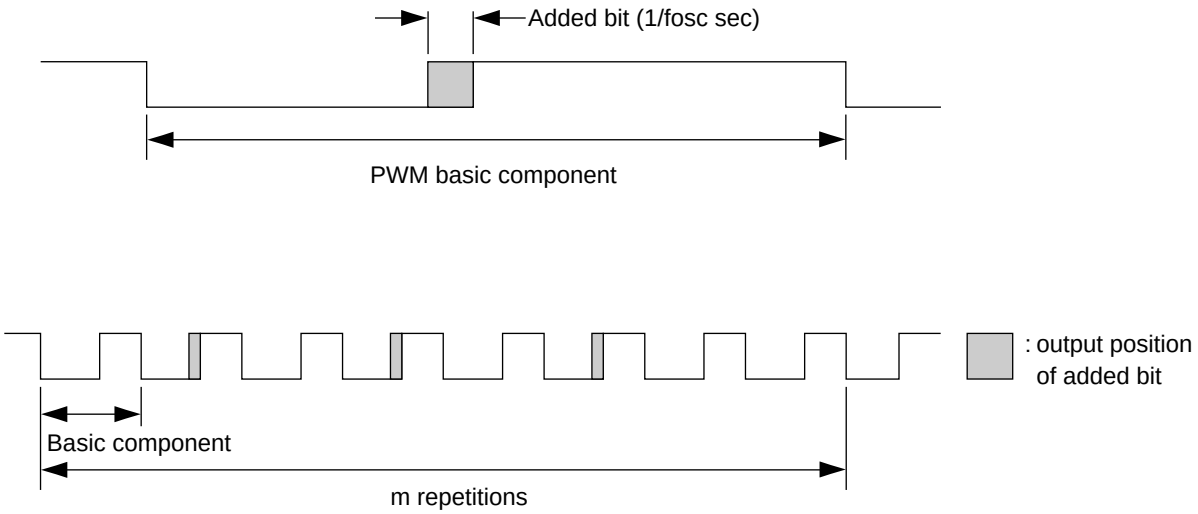


Figure 9-4-4 Method of Added Bits

PWM2 and PWM3 are 10-bit precision PWM channels. Output is controlled so that the upper 6 bits are the basic component and the lower 4 bits are added bits.

In other words,

PWM repetition frequency = fosc/2^{10}

PWM conversion frequency = fosc/2^{6}

Number of repetitions of PWM basic component = 2^{4}

The position at which the additional bit is output within the 16 (=2^{4}) repetitions of the PWM basic component is determined by the lower 4 bits of the data set in each PWM register.

PWM0 and PWM1 are 13-bit precision PWM channels. Output is controlled so that the upper 8 bits are the basic component and the lower 5 bits are added bits.

In other words,

PWM repetition frequency = fosc/2^{13}

PWM conversion frequency = fosc/2^{8}

Number of repetitions of PWM basic component = 2^{5}

The position at which the additional bit is output within the 32 (=2^{5}) repetitions of the PWM basic component is determined by the lower 5 bits of the data set in each PWM register.

While PWM14 has an accuracy of 14 bits, the basic component is output in the higher 8 bits and the additional bits are output in the lower 6 bits.

That is, when the PWM14S flag is '0',

PWM repetition frequency = fosc/2^{14},

PWM conversion frequency = fosc/2^{8}, and

Number of repetitions of PWM basic component = 2^{6}

The position at which the additional bit is output within the 64 (=2^{6}) repetitions of the PWM basic component is determined by the lower 6 bits of the data set in each PWM register.

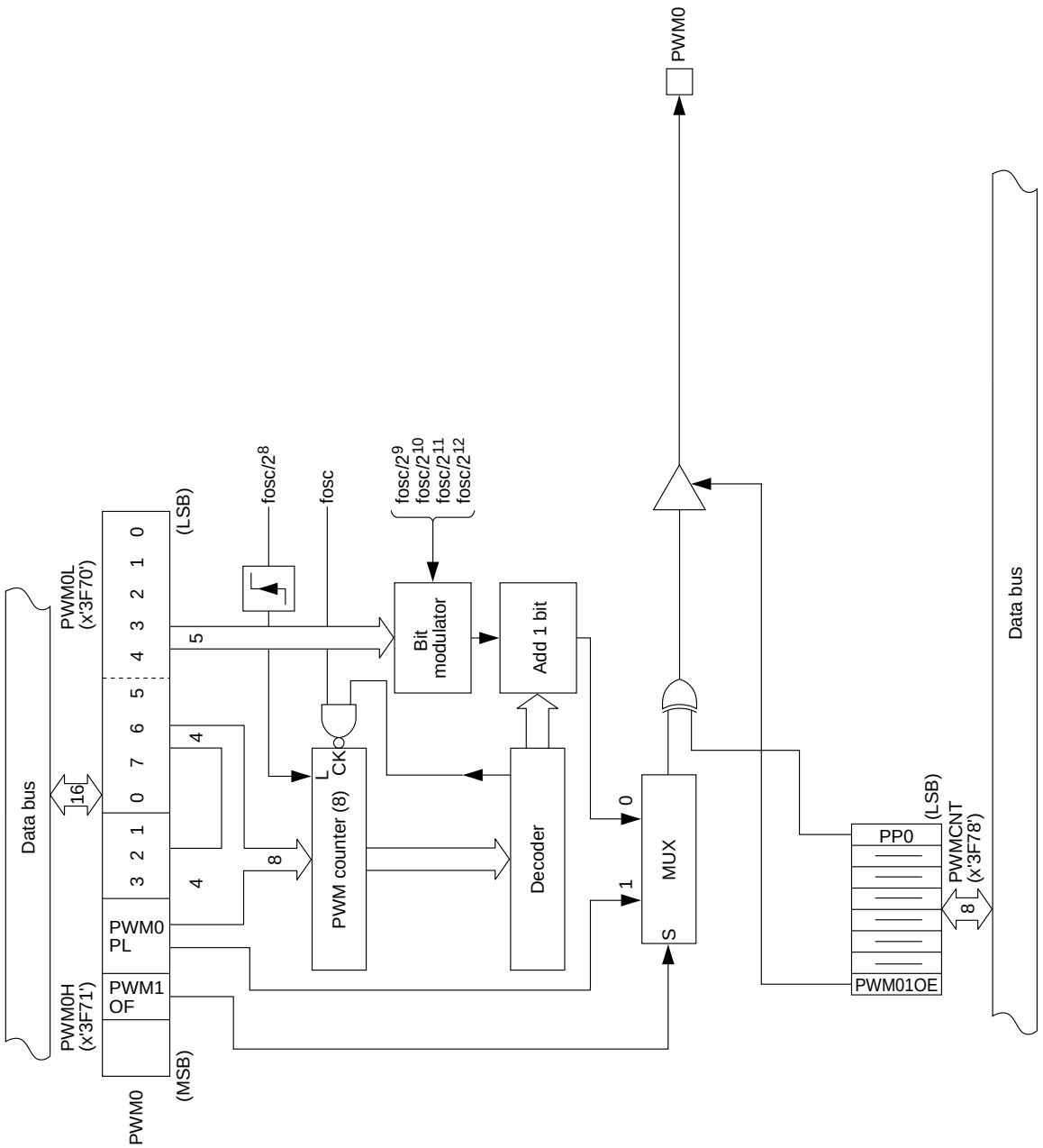


Figure 9-4-5 PWM0 Block Diagram

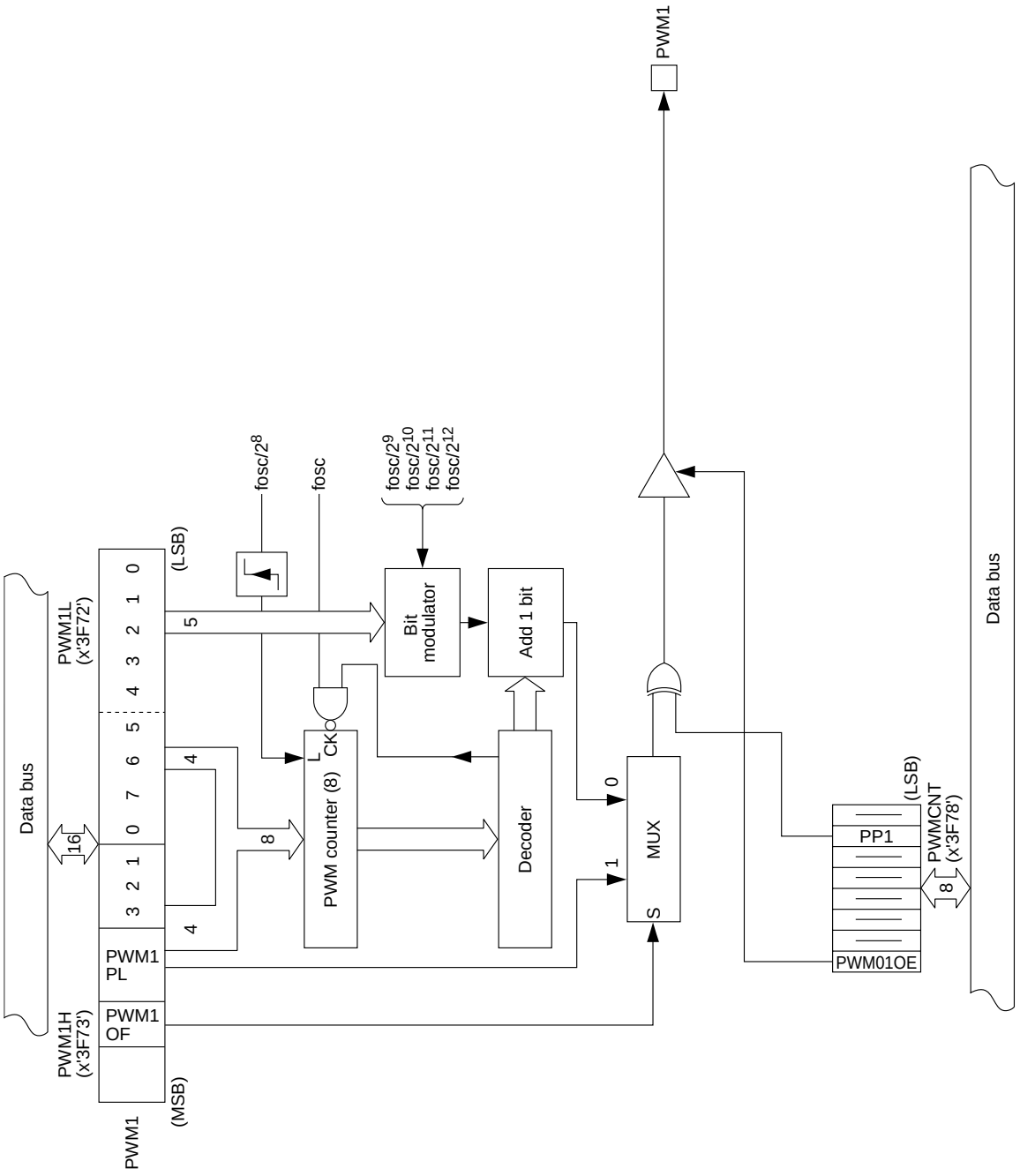


Figure 9-4-6 PWM1 Block Diagram

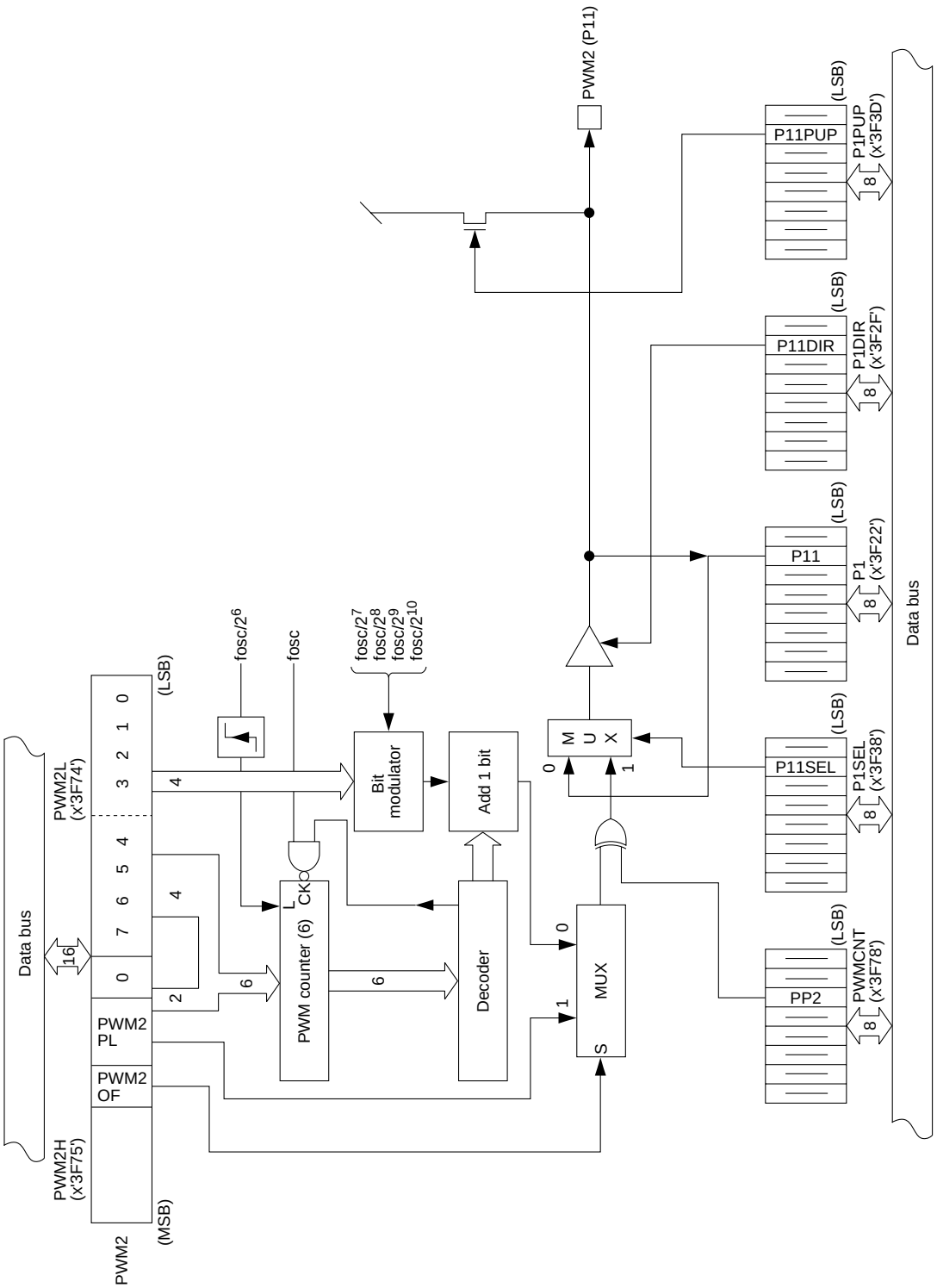


Figure 9-4-7 PWM2 Block Diagram

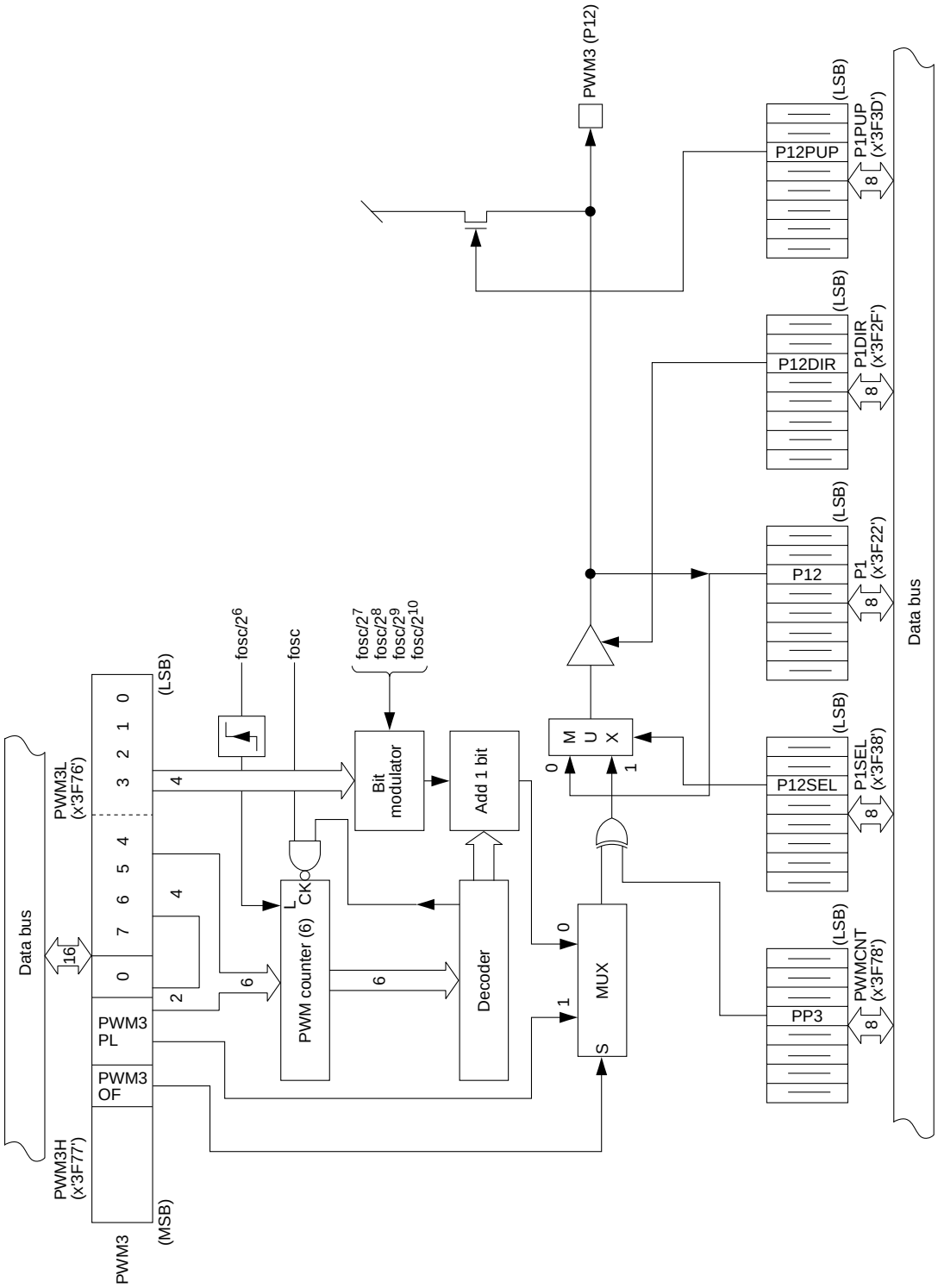


Figure 9-4-8 PWM3 Block Diagram

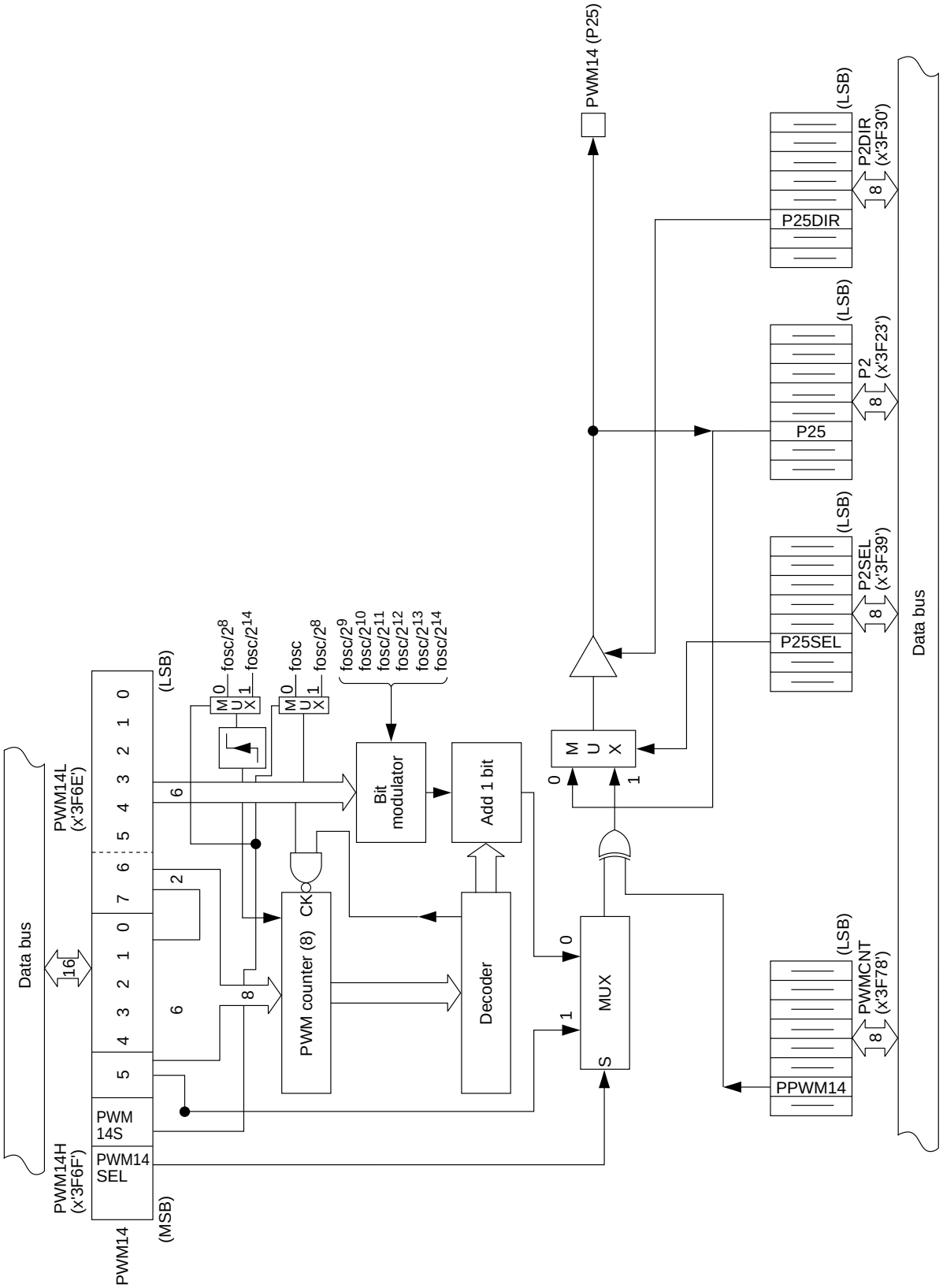


Figure 9-4-9 14-Bit PWM Block Diagram

9-5 Pseudo-V Signal Generating Unit

During special playback, this function outputs a signal (pseudo-VSYNC) that substitutes for the VSYNC signal.

At both edges the HSW signal, a high impedance level is output to the VLP pin. At the same time, the value set in the Pseudo-V Counter Preset Register (AVPR0: x'3F8F', AVPR1: x'3F90') is loaded into the upper 8 bits of the 11-bit Pseudo-V Counter. The Pseudo-V Counter counts at fs/8. A high-level signal is output to the VLP pin when the upper 8 bits of the Pseudo-V Counter match the value set in the Pseudo-V Counter Output Compare Register 0 (AVOCR0: x'3F91'). Setting the AVZC Flag of the Pseudo-V Control Register (AVCNT: x'3F93') to '1', enables use of AVOCR1. A Hi-Z level is output to the VLP pin when the upper 8 bits of the Pseudo-V Counter match the value set in the Pseudo-V Counter Output Compare Register 1 (AVOCR1: x'3F92'). Overflow of the Pseudo-V Counter resets the VLP pin to a low-level. If the Pseudo-V signal is not output to the VLP pin, the VLP pin can output the value set in the AVC0 Flag (bp2) of the AVCNT Register. The AVC0 Flag sets polarity of the Pseudo-V signal.

9-5-1 Pseudo-V Signal Control Registers

Control registers of the Pseudo-V signal generating unit are listed below.

Table 9-5-1 Control Registers of the Pseudo V-Signal Generating Unit

Register Name	RAM Address	R/W	Function
AVPR0	x'3F8F'	R/W	Pseudo-V Counter Preset Register 0
AVPR1	x'3F90'	R/W	Pseudo-V Counter Preset Register 1
AVOCR0	x'3F91'	R/W	Pseudo-V Counter Output Compare Register 0
AVOCR1	x'3F92'	R/W	Pseudo-V Counter Output Compare Register 1
AVCNT	x'3F93'	R/W	Pseudo-V Control Register

Details of the Pseudo-V signal control registers are described below.

(1) Pseudo-V Counter Preset Register 0 (AVPR0: x'3F8F', R/W)

At the rising edge of the HSW signal, the upper 8 bits of the Pseudo-V Counter are preset with the value set in the AVPR0 Register.

(2) Pseudo-V Counter Preset Register 1 (AVPR1: x'3F90', R/W)

At the falling edge of the HSW signal, the upper 8 bits of the Pseudo-V Counter are preset with the value set in the AVPR1 Register.

(3) Pseudo-V Counter Output Compare Register 0 (AVOCR0: x'3F91', R/W)

After the 11-bit Pseudo-V Counter is preset at either edge of HSW by the value set in the AVPR0 Register or the AVPR1 Register, it counts upward at fs/8. A high-level signal is output to the VLP pin when the upper 8 bits of the Pseudo-V Counter match the value set in the AVOCR0 Register.

(4) Pseudo-V Counter Output Compare Register 1 (AVOCR1: x'3F92', R/W)

After the 11-bit Pseudo-V Counter is preset at either edge of HSW by the value set in the AVPR0 Register or the AVPR1 Register, it counts upward at fs/8. A Hi-Z level is output to the VLP pin when the upper 8 bits of the Pseudo-V Counter match the value set in the AVOCR1 Register.

(5) Pseudo-V Control Register (AVCNT: x'3F93', R/W)

The Pseudo-V Control Register specifies polarity of the Pseudo-V signal, specifies the Pseudo-V output (normal or fixed), indicates whether HSYNC is a composite signal, and enables or disables use of AVOCR1.

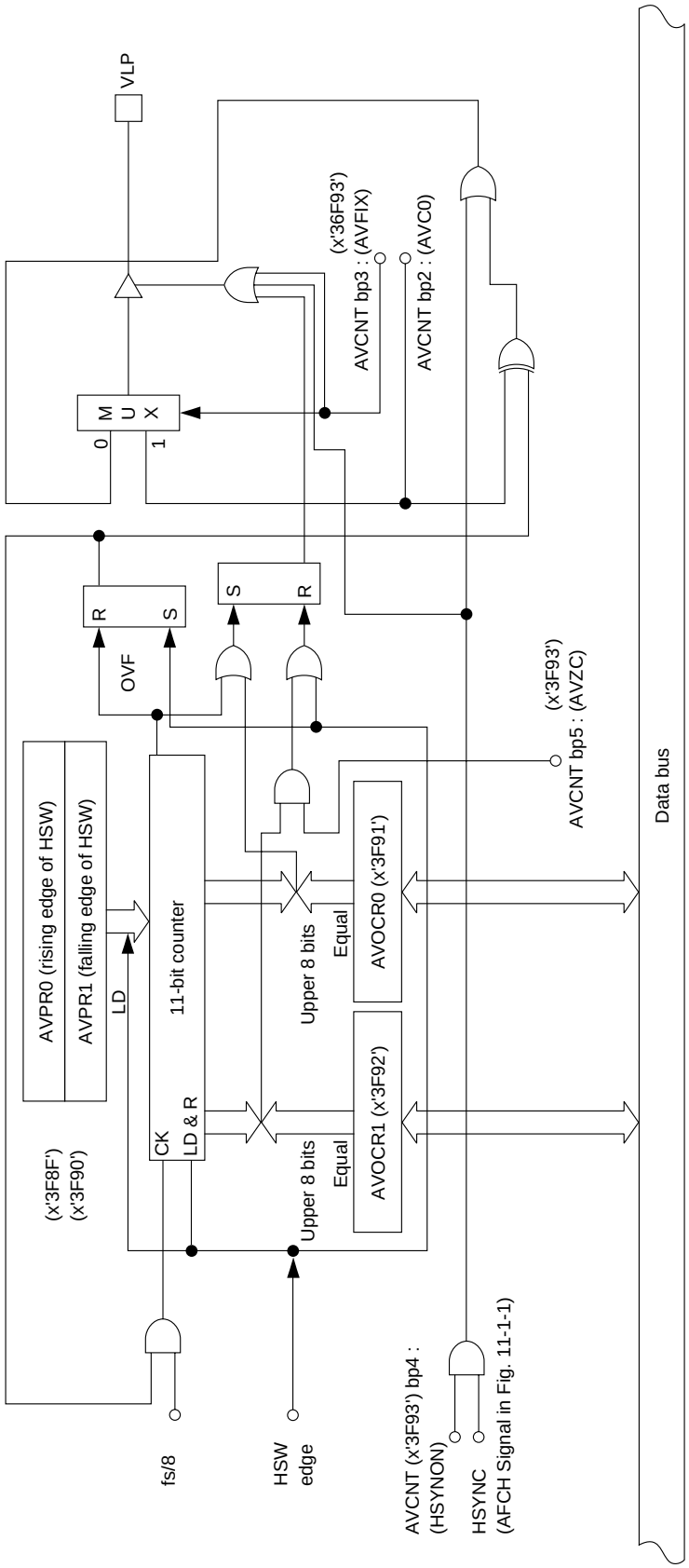
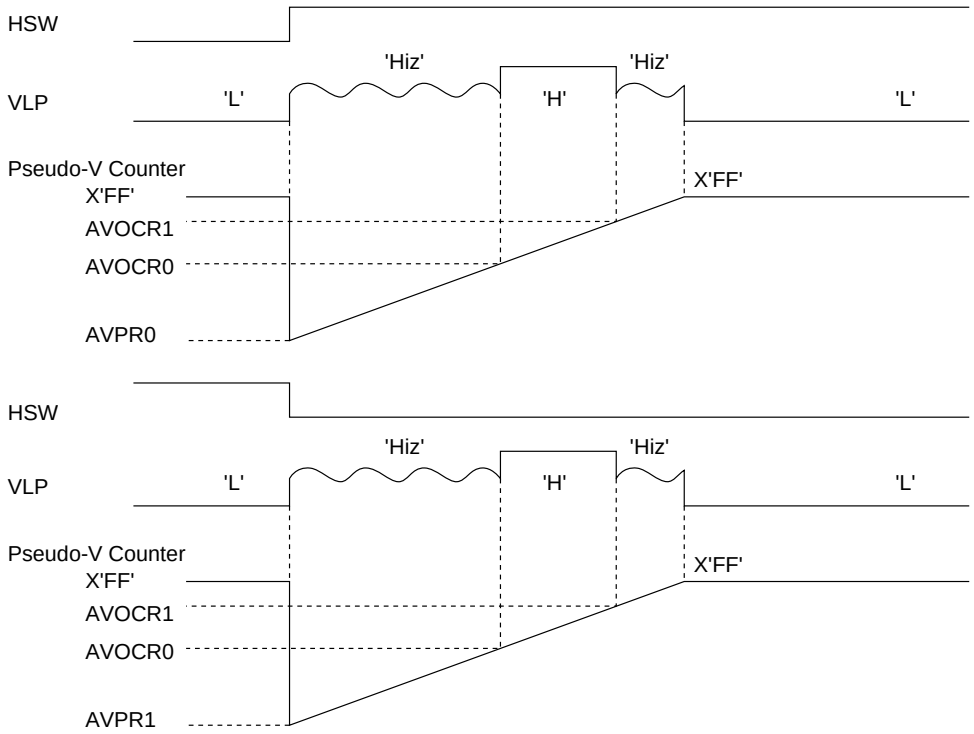


Figure 9-5-1 Pseudo-V Signal Generating Unit

(When using AVOCR1)
AVCNT (x'3F93')
bp5: when AVZC = 1



(When AVOCR1 has not been used)
AVCNT (x'3F93')
bp5: when AVZC = 0

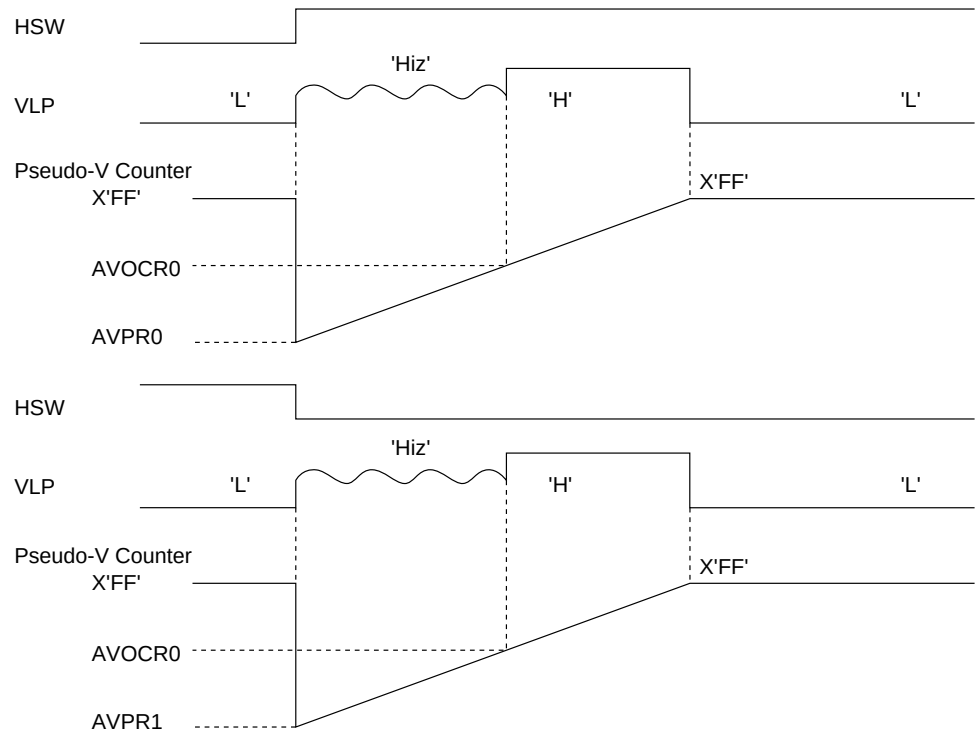


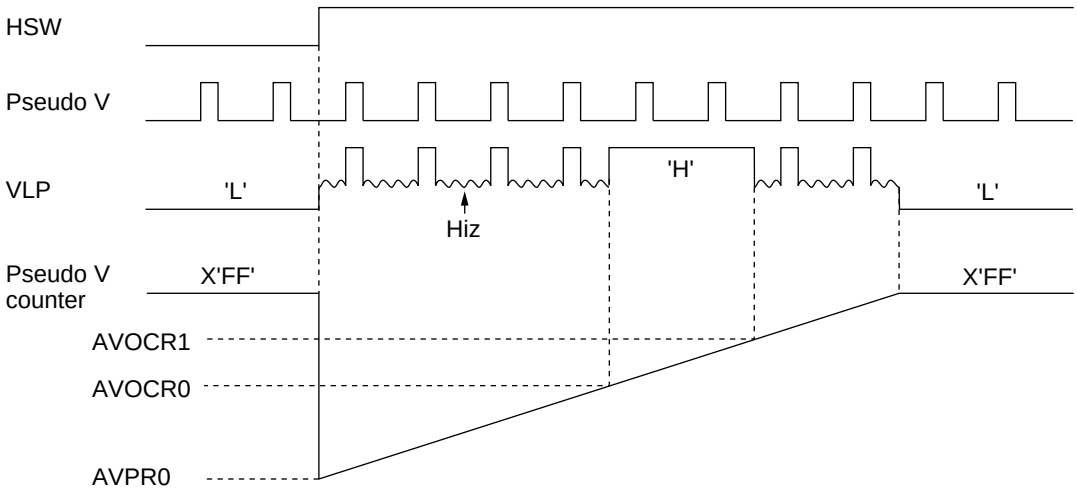
Figure 9-5-2 Pseudo-V Signal Generating Unit, Timing Diagram

9-5-2 Pseudo-H Signal Superimposition Function

The HSYNC signal (pseudo H signal) generated by the horizontal synchronization AFC circuit can be superimposed on the pseudo V signal.

Setting method

The pseudo H signal is superimposed on the pseudo V signal output when the HSYNON flag (AVCNT: x'3F93', bit 4) is set to '1' in addition to the pseudo V output settings.



When AVOCR1 is used (AVZC flag = '1') and HSYNC superimposition is ON (HSYNON flag = '1').

Figure 9-5-3 Timing chart of pseudo V signal generation during pseudo H signal superimposition

9-6 Head Amp and Rotary Control Unit

9-6-1 Dedicated Output Functions for Head Amp and Rotary Control

Output functions, dedicated for the head amp and rotary control, are built-in.

When the direction control for Ports P66 and P67 is set to "Output", the respective pins become dedicated HAMP and ROT outputs. The Head Amp/Rotary Control Register (HAMPCNT: x'3F94', R/W) controls the HAMP and ROT output. Data set in the HAMPCNT Register is latched at either edge of the HSW signal, and through combination logic, becomes the HAMP and ROT output. Therefore, even if the value set in the HAMPCNT Register is modified, the previous state is maintained until either a rising or falling edge of the HSW signal occurs.

When input mode is set, pin levels can be monitored as general-purpose inputs.

The combination logic for HAMP and ROT output signals is as follows.

$$HAMP = (\underline{HAMPC1} \bullet DENV + \overline{HAMPC1} \bullet HSW) \bullet \overline{HAMPC0} + HAMPC2$$

$$ROT = (ROTC1 \bullet DENV + HSW) \bullet \overline{ROTC0} + ROTC2$$

When the HSW signal is at a low-level : The DENV input signal is output
When the HSW signal is at a high level: The DENV signal will be output with negative polarity

For example, if HAMPC0 and HAMPC1 are set to '0' and HAMPC2 is set to '1', the HAMP output will be the HSW signal with negative polarity.
If ROTC0 and ROTC2 are set to '0' and ROTC1 is set to '1', the ROT output will be the exclusive-or (XOR) of the DENV signal and the HSW signal.

The HAMP signal forced fixing flag (HAMPFX) and the ROT signal forced fixing flag (ROTFX) are set asynchronously in the HSW signal.

If the port direction of P66 and P67 is set to "Output" and HAMP and ROT dedicated output is selected, because the output signal will be latched by the output latch at either edge of the HSW signal, set the HAMPCNT Register sufficiently before the edge of the HSW signal.

If the HAMPCNT Register is set at the same time as an edge of the HSW, the output may differ from the set value.

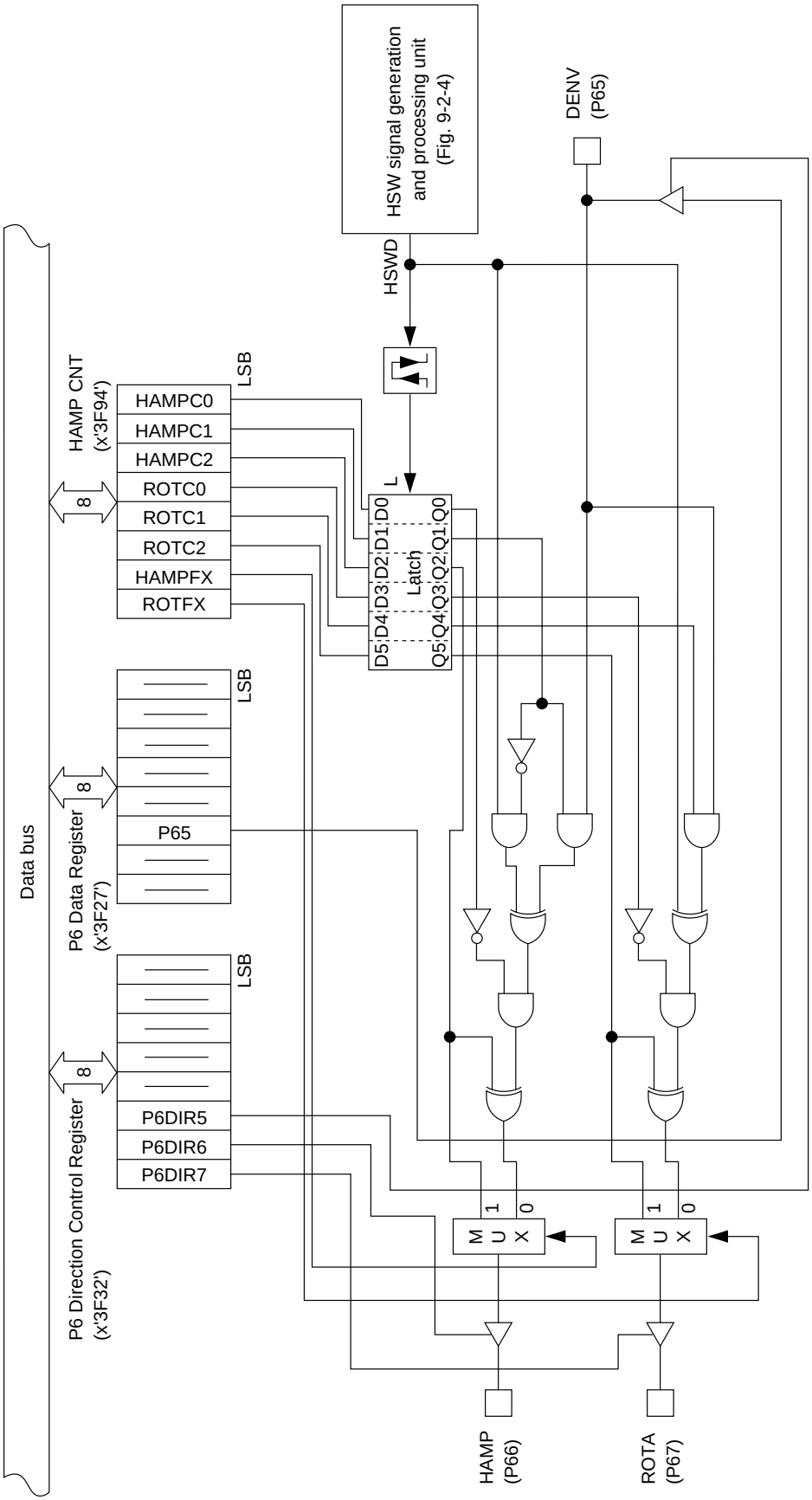


Figure 9-6-1 Head Amp and Rotary Control Unit

9-7 Digital PGMM

9-7-1 Digital PGMM Configuration

A digital PGMM function is built-in to generate HSW signals.

The HSW signal may also be generated from analog PGMM. Register settings select either analog PGMM or digital PGMM. If digital PGMM is selected, the PGMM pin can be used as a general-purpose input port (Port 93). Digital PGMM specifications are summarized below.

Table 9-7-1 Summary of Digital PGMM Specifications

		(When fs=7.16 MHz)
Jitter	fs/2	279.3ns
Setting precision	fs/8	1.117µs
Maximum delay time	$8/fs \times (2^{16}-1)$	73.2ms

When setting the delay time in the PGMMTIM Register, set a value that is 1/2 of the frame cycle or lower.

The following settings are required depending on the method of selecting digital PGMM.

1. Set the DPGMS flag to '0' and the PGMS flag to '1' when starting digital PGMM with the YFG counter equal to '1'.
2. Set both the DPGMS flag and the PGMS flag to '1' when starting digital PGMM when the YFG counter is equal to the value set in DPGMC5-0.

9-7-2 Digital PGMM Operation

Calculate the digital PGMM setting value, N, from the amount of PGMM delay. Set 16-bit data in the PGMMTIMH and PGMMTIML Registers.

Set the upper 8 bits in PGMMTIMH (x'3F8C') and the lower 8 bits in PGMMTIML (x'3F8B'). Use a MOVL instruction to simultaneously set both the PGMMTIMH and PGMMTIML Registers.

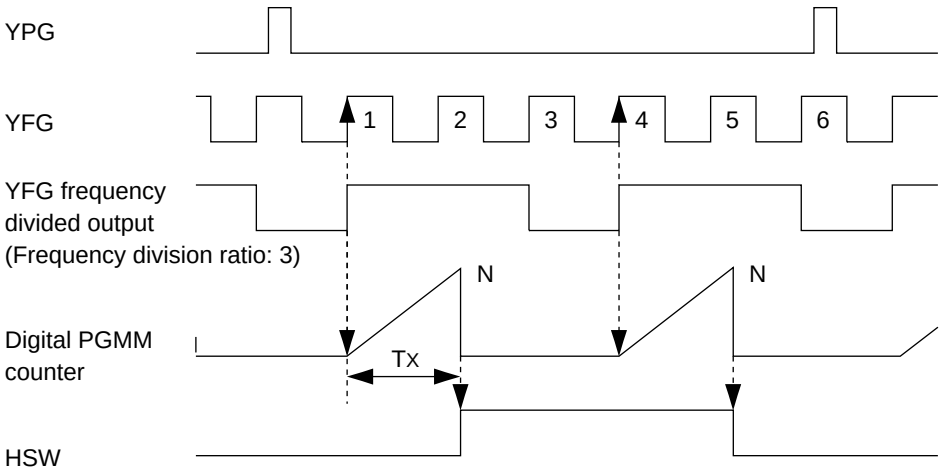


Figure 9-7-1 Timing Diagram

Calculation of digital PGMM delay

$$Tx = N \times \frac{8}{fs}$$

Tx: Digital PGMM delay time
N: PGMMTIM setting value
fs: System clock frequency

Chapter 10
Standby, Clock Operation/Halt, and Reset Functions

10

10-1 Standby Functions

10-1-1 Entering Standby Mode

CPU operating modes for the internal OSD servo are NORMAL mode and SLOW mode. The 2 types of STOP modes and 2 types of HALT modes are standby modes. Power consumption can be reduced through the effective use of these modes.

The word OSC used throughout this chapter refers to the oscillations of OSCI/OSCO.

■ High-Speed Operation State (NORMAL)

- The state in which the program is being executed using the clock (OSC) for high-speed operation.

■ Halt State (HALT0, HALT1)

- In this state, the CPU is stopped but the oscillator continues to run. An interrupt can instantly change the mode to an operating state. In the HALT0 mode, OSC is always oscillating. In the HALT1 mode, XI oscillates continuously and OSC stops.

■ Stop State (STOP0, STOP1)

The following functions cannot be used in the SLOW mode:
- PWM0, 1, 2, 3, 14

- In this state neither the CPU nor the oscillator is running. An interrupt will start the oscillator, and after waiting for the oscillation to stabilize, the mode will transfer to an operating state.
- An interrupt will transfer the STOP0 mode to the NORMAL state.
- An interrupt will transfer the STOP1 mode to the SLOW state.
- Before the operation switches over to the STOP0 or STOP1 modes, the watchdog timer is stopped and the watchdog counter is cleared.
- Transition to the STOP0 mode is only possible from the NORMAL mode of fs=fosc/2. Transition to the STOP1 mode is only possible from the SLOW mode of fs=fxi/2.

■ Slow Operating State (SLOW)

- This state executes the program with a low-speed clock (XI).

■ Idle State (IDLE)

- The state in which the program is waiting for the clock oscillation of the target mode to become stable when changing over from the SLOW mode of fs=fxi/2 to the NORMAL mode of fs=fosc/2 or when changing over from the NORMAL mode of fs-fosc/1024 to the SLOW mode of fs=fxi/32. The program operation is the same as in the SLOW mode.

In the servo microcomputer with built-in OSD functions, OSC is used as the clock for the NORMAL mode and XI is used as the clock for the SLOW mode. The mode is specified by the CPU mode control register (CPUM).
The reason for returning from the standby mode can be a normal reset operation or an interrupt. The wait period for oscillation stabilization is inserted when returning from the STOP mode, and a stabilization period is inserted in the program when changing over from the SLOW mode of fs=fxi/2 to the NORMAL mode of fs=fosc/2 or in the XI when changing over from the NORMAL mode to the SLOW mode during release from reset or immediately after return from the STOP0 mode. There is no oscillation stabilization wait period when returning from the HALT mode. The oscillation state of the system clock automatically returns to its state prior to entering a standby mode.

■ When reset is released

- When the SXI pin is at a high or low-level, after waiting 2^{18} counts of fosc for oscillation stabilization, the mode is transferred to NORMAL mode with fs=focs/10^{24}.

Method to Reduce Power Consumption during STOP and HALT Modes

1. Be careful that input/output current to/from pins and the signal level at pin inputs do not become unstable.
2. Set output pins so that their output matches external levels, or set the direction control to "Input". (Clear (input) or set (output) the flags PnDIR0~7 of register PnDIR.)
3. Set input pins so that external levels will become fixed (VDD or GND). However, it is not necessary to fix pins P90 and P91 if input port 9 has been input-disabled. It is necessary to fix pin P92 (PGMM). (The input is disabled by clearing flags P92IE, P91IE, and P90IE (bits 7, 6, and 5).)
   (The pin can be at any level between GND and Vdd.)
   Also, there is no need to fix ports 87~80, ports C3~C0, and ports B6~B0. (These ports remain in the input disabled condition at all times other than when they are being read.)
4. Turn off the analog power source.
   (Clear the STBH flag of register ANACNT (bit 0 of x'3F81').)
5. Turn off the power source for the A/D converter.
   (Clear the ADEN flag of register ADM (bit 4 of x'3F6D').)
6. Turn off the power source for the D/A converter.
   (Clear the DAEN flag of register HOCRCNT (bit 5 of x'3F7C').)
7. Turn off the VCO power supply. (Write x'00' in register VCOCNT (x'3FD0').)
8. Turn off the character brightness bias power supply. (Clear the BIASPW flag of register CLCNT (bit 5 of x'3FD1').)
9. Turn off the power supply of the buffer for halftone display. (Clear the BUFPW flag of register CLCNT (bit 4 of x'3FD1').)

P0DIR register x'3F2E'
P1DIR register x'3F2F'
P2DIR register x'3F30'
P5DIR register x'3F31'
P6DIR register x'3F32'
P7DIR register x'3F33'
P8DIR register x'3F34'
PADIR register x'3F35'
PBDIR register x'3F36'
PCDIR register x'3F37'

10. Turn off the pin CVIN2 clamping power supply. (Clear the CSCLPW flag of register CLCNT (bit 3 of x'3FD1').)
11. Turn off the clamping power supply of pin CVIN. (Clear the CVCLPW flag of register CLCNT (bit 2 of x'3FD1').)
12. Turn off the dot clock phase adjuster power supply. (Clear the DCKPW flag of register DOTCNT (bit 4 of x'3FD2').)
13. Turn off the AFC circuit LPF power supply. (Clear the LPFPW flag of register PCSTA (bit 5 of x'3FD3').)
14. Turn off the OSCDIV output. (Clear the P17SEL flag of register P1SEL (bit 7 of x'3F38').)
15. Turn off the OSCI2/OSCO2 oscillation function (clear the P21SEL flag of register P2SEL (bit 1 of x'3F39')), and then set it as output (set the flags P2DIR1 and P2DIR2 of register P2DIR (bits 2 and 1 of x'3F30')).
16. Clear the OPTION register (x'3F83') to x'00'.
17. Set PWM0 and PWM1 as outputs. (Set the flag PWM01OE of register PWMCNT (bit 7 of x'3F78').)
18. Halt the pin selection and operation of the serial interface.
    (Clear bits 0, 1, 2, 3, and 4 of register SC0MD3 (x'3F9A').
    Clear bits 4 and 7 of register SIM1 (x'3F9C').
    Clear bits 0 and 5 of register SIC1 (x'3F9E').)

10-1-2 CPU Mode Control Register

Entering each mode can be made by changing the flags of the CPU Mode Control Register (CPUM).

(1) When there is no XI oscillation (SXI pin = low-level start)

CPU Mode Control Register (CPUM : x'3F00' R/W)

7	6	5	4	3	2	1	0
0	OSC SEL1	OSC SEL0	0	STOP	HALT	OSC1	OSC0
At reset: 1	1	0 (See note 1)	0	0	0	0 (See note 5)	

Operating Mode	OSC SEL1	OSC SEL0	STOP	HALT	OSC1	OSC0	Status: OSC1/OSC0 pin (See note 6)	Status: XI/XO pin (See note 7)	Status: System clock (fs)	Status: CPU	Voltage at which operation is guaranteed (See note 3)	Comments
STOP0 (See note 8)	0	0	1	0	0	0	Stopped	Stopped	—	Stopped	2.5 to 5.5	Wait for OSC oscillation stabilization when returning
HALT0	0	0	0	1	0	0	Oscillating	Stopped	—	Stopped	4.0 to 5.5 (4.5 to 5.5)	
NORMAL (See note 2)	0	0	0	0	0	0	Oscillating	Stopped	fosc/2	NORMAL operation	4.0 to 5.5 (4.5 to 5.5)	It is possible to change over from this state to the STOP0 mode/ HALT0 mode. (See Note 9)
	0	1							fosc/8			
	1	0							fosc/32			
	1	1							fosc/1024		2.5 to 5.5	When reset is released (See note 4)

Figure 10-1-1 Control of Operating Modes and Stopped or Oscillating Clock

Note 1: ***This bit (bp4) must be set to '0'.***

Note 2: ***"NORMAL" means normal operation.***

Note 3: ***The value of voltage at which operation is guaranteed is given under the conditions of fosc=14.32 MHz (17.72 MHz) and fxi=32.768 kHz.***

Note 4: ***When reset is released, after waiting for stabilization of fosc oscillation (262,144 counts of fosc), the system will start in NORMAL mode and the system clock (fs) is fosc divided by 1024.***

Note 5: ***The system clock can be verified by reading the status monitor, the OSC0 flag (bp0) of the CPUM register (x'3F00').***

Note 6: ***While the reset input pin (NRST) is at a low level, oscillation of OSC is stopped.***

Note 7: ***While the reset input pin (NRST) is at a low level, the XI/XO pin can oscillate. Therefore, during this interval, the XI pin input is inverted and output to the XO pin. Be careful when using this XI/XO pin as a general-purpose port.***

Note 8: ***Stop the WD timer and clear the WD counter immediately before changing over to the STOP mode.***

Note 9: ***It is possible to change over to the STOP0 mode or the HALT0 mode only from the NORMAL mode of fs=fosc/2.***

(2) When there is XI oscillation (SXI pin = high-level start)

CPU Mode Control Register (CPUM : x'3F00' R/W)

7	6	5	4	3	2	1	0
0	OSC SEL1	OSC SEL0	0	STOP	HALT	OSC1	OSC0
At reset:	1	1	0 (See note 1)	0	0	0	0 (See note 5)

Operating Mode	OSC SEL1	OSC SEL0	STOP	HALT	OSC1	OSC0	Status: OSCI/OSCO pin (See note 6)	Status: XI/XO pin (See note 7)	Status: System clock (fs)	Status: CPU	Voltage at which operation is guaranteed (See note 3)	Comments
STOP0 (See note 8)	0	0	1	0	0	0	Stopped	Stopped	—	Stopped	2.5 to 5.5	Wait for OSC oscillation stabilization when returning
HALT0	0	0	0	1	0	0	Oscillating	Oscillating	—	Stopped	4.0 to 5.5 (4.5 to 5.5)	
NORMAL (See note 2)	0	0	0	0	0	0	Oscillating	Oscillating	fosc/2	NORMAL operation	4.0 to 5.5 (4.5 to 5.5)	It is possible to change over from this state to the STOP0 mode/ HALT0 mode. (See Note 9)
	0	1							fosc/8			
	1	0							fosc/32			
	1	1							fosc/1024		2.5 to 5.5	When reset is released (See note 4)
IDLE	0	0	0	0	0	1	Oscillating	Oscillating	fxi/2	SLOW operation	2.2 to 5.5	Stabilize fosc when transferring from SLOW mode to NORMAL mode
	0	1							fxi/8			
	1	*							fxi/32			
SLOW	0	0	0	0	1	1	Stopped	Oscillating	fxi/2	SLOW operation	2.2 to 5.5	It is possible to change over from this state to the STOP0 mode/ HALT0 mode. (See Note 10)
	0	1							fxi/8			
	1	*							fxi/32			
HALT1	0	0	0	1	1	1	Stopped	Oscillating	—	Stopped	2.2 to 5.5	
STOP1 (See note 8)	0	0	1	0	1	1	Stopped	Stopped	—	Stopped	2.2 to 5.5	Wait for XI oscillation stabilization when returning

Figure 10-1-2 Control of Operating Modes and Stopped or Oscillating Clock

Note 1: ***This bit (bp4) must be set to '0'.***

Note 2: ***"NORMAL" means normal operation.***

Note 3: ***The value of voltage at which operation is guaranteed is given under the conditions of fosc=14.32 MHz (17.72 MHz) and fxi=32.768 kHz.***

Note 4: ***When reset is released, after waiting for stabilization of fosc oscillation (262,144 counts of fosc), the system will start in NORMAL mode and the system clock (fs) is fosc divided by 1024.***

Note 5: ***The system clock can be verified by reading the status monitor, the OSC0 flag (bp0) of the CPUM register (x'3F00').***

Note 6: ***While the reset input pin (NRST) is at a low level, oscillation of OSC is stopped.***

Note 7: ***While the reset input pin (NRST) is at a low level, the XI/XO pin oscillates.***

Note 8: ***Stop the WD timer and clear the WD counter immediately before changing over to the STOP mode.***

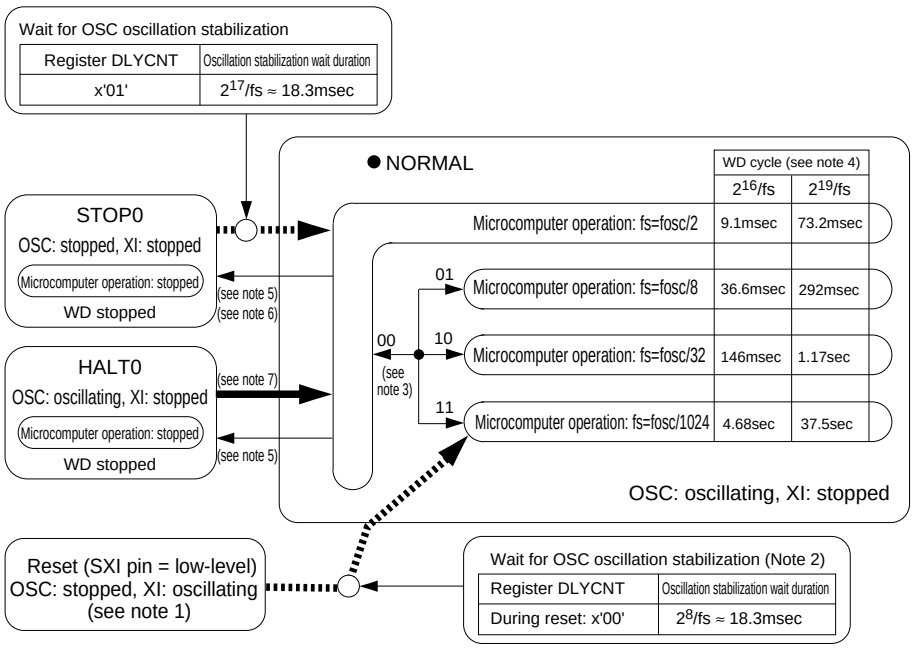
Note 9: ***It is possible to change over to the STOP0 mode or the HALT0 mode only from the NORMAL mode of fs=fosc/2.***

Note 10: ***It is possible to change over to the STOP1 mode or the HALT1 mode only from the SLOW mode of fs=fxi/2.***

10-1-3 Transition between Operating Modes

■ Operating mode transition (when there is no XI oscillation)

The NORMAL mode is the one in which the microcomputer is operating. The NORMAL and SLOW modes are operating modes. During the HALT mode, operation of the microcomputer stops; it remains oscillating but the system clock is halted. In STOP mode, oscillation is stopped and the system clock also stops. Transitions (when there is no XI oscillation) between operating modes are described below.



Note 1: ***While the reset input pin (NRST) is at a low level, oscillation of OSC is stopped.***
While the reset input pin (NRST) is at a low level, the XI/XO pin can oscillate. Therefore, during this interval, the XI pin input is inverted and output to the XO pin. Be careful when using this XI/XO pin as a general-purpose port.

Note 2: ***When reset is released, after waiting for stabilization of fosc oscillation (262,144 counts of fosc), the system will start in NORMAL mode and the system clock (fs) is fosc divided by 1024.***

Note 3: ***The frequency of the system clock can be selected by the software. (The flags OSCSEL1 and OSCSEL0 of register CPUM (x'3F00'))***

Note 4: ***The WD cycle of 2^{16}/fs or 2^{19}/fs is selected by the software. (The WDS flag of register WDCNT (x'3F1D'))***

Note 5: ***Changing over to the HALT0 and STOP0 modes is done from the NORMAL mode of fs=fosc/2. Change over to the standby mode immediately after stopping the WD timer and clearing the WD counter.***

Note 6: ***When changing over to the STOP0 mode from the NORMAL mode of fs=fosc/2, always be sure to set register DLYCNT to x'01'.***

Note 7: ***When starting the watchdog timer after returning from the HALT mode, be sure to start the watchdog timer operation only after clearing it (WDCLR flag (bp2) of register WDCNT (x'3F1D')).***

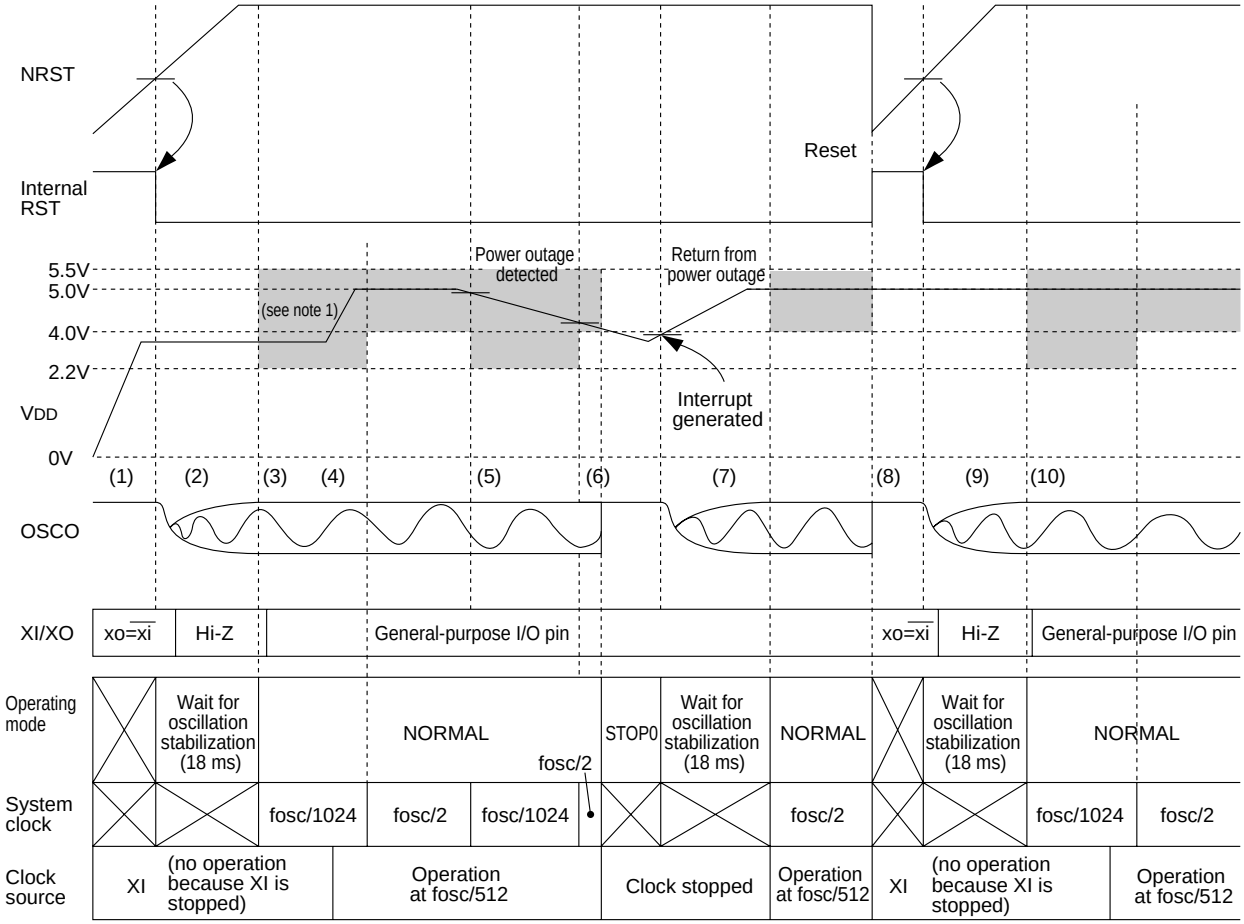
Conditions

S_{XI} = fixed at a low-level
f_{OSC} = 14.32MHz

Addendum

Bold arrow: return due to an interrupt
Dotted arrow: Wait for oscillation stabilization

Figure 10-1-3 Microcomputer Mode Transition Diagram (for No XI)



Note 1: Shaded voltage areas indicate voltages at which operation is guaranteed for the operating modes. Operation is guaranteed at these voltages when fosc=14.32 MHz.

Figure 10-1-4 Operating Mode Transition Example when there is No XI

(1) After the time power is turned on, until reset is released, OSC1/OSC2 oscillation is stopped and the XI/XO pin oscillates. The XI pin input is inverted and output to the XO pin. Be careful when using this pin as a general-purpose I/O port.

(2) After reset is released, an OSC oscillation stabilization wait interval of approximately 18 ms (@14.32 MHz) is automatically inserted.

(3) After the oscillation stabilization wait is completed, operation starts in the NORMAL mode with fs (system clock) = fosc/1024.

(4) When the power source voltage (Vdd) becomes sufficient to operate at fs=fosc/2, the clock can be switched to NORMAL divided by 2 (fs=fosc/2). If, on the other hand, the power supply voltage falls at this point, the operation moves to step (6).

(5) Before the power source voltage drops through the fosc/2 NORMAL operating range, the mode switches to the low power consumption NORMAL mode (fs=fosc/1024).

(6) Further, before the voltage falls below the operating power supply voltage range of the 1/1024 frequency NORMAL mode, the operation changes to the STOP0 mode after temporarily passing through the 1/2 frequency NORMAL mode.

Before entering the standby mode, the following are necessary: low power consumption processing [☞ 10-1-1, "Transfer to Standby Mode"], return interrupts must be enabled, and the oscillation stabilization wait clock must be set (set the DLYCTR Register to x'00').

(7) After receiving a voltage return interrupt, an OSC oscillation stabilization wait interval of approximately 18 ms (@14.32 MHz) is automatically inserted, and the mode returns to the 1/2 frequency NORMAL mode.

(8) If a reset has occurred for some reason (external input or self-reset), OSC1/OSC2 oscillation is stopped and the XI/XO pin oscillates until reset is released.

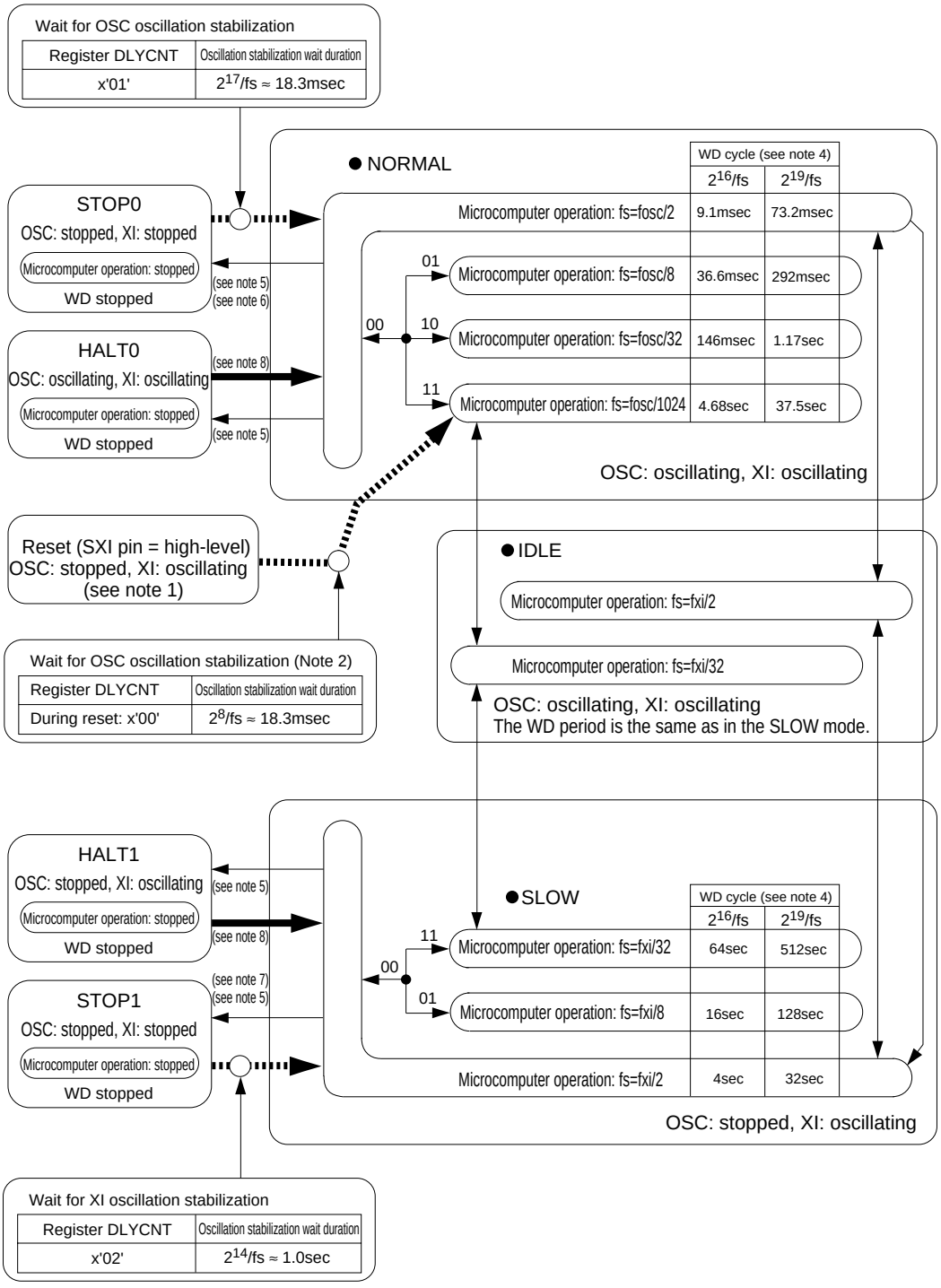
(9) After reset is released, an OSC oscillation stabilization wait interval of approximately 18 ms (@14.32 MHz) is automatically inserted.

(10) After the oscillation stabilization wait is completed, operation starts in the NORMAL mode with fs (system clock) = fosc/1024.

The microcomputer operating mode is changed by making settings in register CPUM (x'3F00'). Also, the watchdog timer period is selected by the WDS flag of register WDCNT (x'3F1D'). When WDS=0, the WD period is 2^{19}/fs (seconds), and when WDS=1, the WD period is 2^{19}/fs (seconds). See Fig. 10-1-3 and Fig. 10-1-5 for details of the relationship between the microcomputer mode and the WD period.

■ Operating mode transition (when XI is oscillating)

The NORMAL and SLOW modes are operating modes. During the HALT mode, operation of the microcomputer stops; it remains oscillating but the system clock is halted. In STOP mode, oscillation is stopped and the system clock also stops. Transitions (with XI oscillation) between operating modes are described below.



Note 1: ***While the reset input pin (NRST) is at a low level, oscillation of OSC is stopped.***
While the reset input pin (NRST) is at a low level, the XI/XO pin oscillates.

Note 2: ***When reset is released, after waiting for stabilization of fosc oscillation (262,144 counts of fosc), the system will start in NORMAL mode and the system clock (fs) is fosc divided by 1024.***

Note 3: ***The frequency of the system clock can be selected by the software. (The flags OSCSEL1, 0 of register CPUM (x'3F00').)***

Note 4: ***The WD cycle of 2^{16}/fs or 2^{19}/fs is selected by the software. (The WDS flag of register WDCNT (x'3F1D').)***

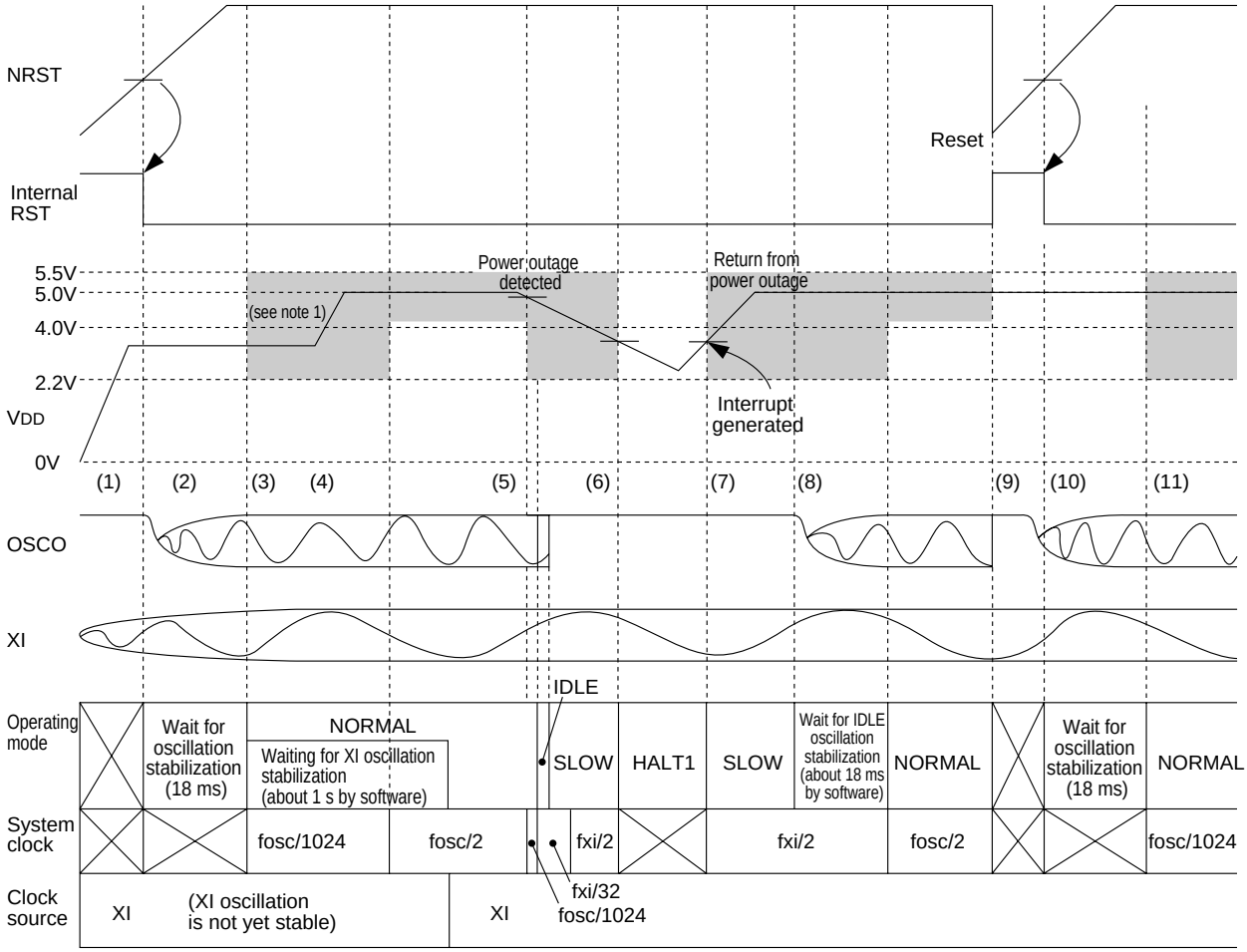
Note 5: ***Changing over to the HALT0 and STOP0 modes is done from the NORMAL mode of fs=fosc/2, and changing to the HALT1 and STOP1 modes is done from the SLOW mode of fs=fxi/2. Change over to the standby mode immediately after stopping the WD timer and clearing the WD counter.***

Note 6: ***When changing over to the STOP0 mode from the NORMAL mode of fs=fosc/2, always be sure to set register DLYCNT to x'01'.***

Note 7: ***When changing over to the STOP1 mode from the SLOW mode of fs=fxi/2, always be sure to set register DLYCNT to x'02'.***

Note 8: ***When starting the watchdog timer after returning from the HALT mode, be sure to start the watchdog timer operation only after clearing it (WDCLR flag (bp2) of register WDCNT (x'3F1D')).***

Figure 10-1-5 Microcomputer Mode Transition Diagram (with XI)



Note 1: Shaded voltage areas indicate voltages at which operation is guaranteed for the operating modes. Operation is guaranteed at these voltages when fosc=14.32 MHz and fxi=32.768 MHz.

Figure 10-1-6 Operating Mode Transition Example with XI

(1) After the time power is turned on, until reset is released, OSC1/OSC2 oscillation is stopped. XI/XO begins oscillating at reset.

(2) After reset is released, an OSC oscillation stabilization wait interval of approximately 18 ms (@14.32 MHz) is automatically inserted.

(3) After the oscillation stabilization wait is completed, operation starts in the NORMAL mode with fs (system clock) = fosc/1024.

(4) The XI/XO oscillation stabilization wait interval is inserted by the software. Further, it is possible to continue software processing during the XI/XO oscillation stabilization period. See (4') below.

The clock (Timer 0 or 6) is set and its operation is started. A software reset can be made if the power supply voltage is sufficient for operation at fs=fosc/1024. When the power source voltage (Vdd) becomes sufficient to operate at fs=fosc/2, the clock can be switched to NORMAL divided by 2 (fs=fosc/2).

If, on the other hand, the power supply voltage drops at this point of time, it is possible to change over to the STOP0 mode. The processing of return from STOP0 mode in this case is the same as item (6) in the operating mode transition (when XI is not oscillating).

(4') An example of the program for waiting for the XI oscillation to become stable (about 1 second by software) is given below. (Supplementary explanation for (4).)

```
        MOV    16, D2
LOOP1   MOVW   x'FFFF', DW0
LOOP0   SUBW   1, DW0
        BNE    LOOP0        ;(4+3) × 65535 = 458745 [Mcycle]
        ADD    -1, D2
        BNE    LOOP1        ;16 × 458745 = 7339920 [Mcycle]
                            ;7339920 × 139usec = 1.025 s
```

(When using a 14.32MHz clock)

The microcomputer operating mode is changed by making settings in register CPUM (x'3F00'). Also, the watchdog timer period is selected by the WDS flag of register WDCNT (x'3F1D'). When WDS=0, the WD period is 2^{16}/fs (seconds), and when WDS=1, the WD period is 2^{19}/fs (seconds). See Fig. 10-1-3 and Fig. 10-1-5 for details of the relationship between the microcomputer mode and the WD period.

(5) In the case of the 1/2 frequency NORMAL mode, the mode changes directly to the 1/2 frequency SLOW mode before the power supply voltage falls outside the operating range. In the case of the 1/8 and 1/32 frequency NORMAL modes, the mode changes first to the 1/1024 frequency NORMAL mode and then to the 1/32 frequency IDLE mode, a software wait is carried out for 20 machine cycles or more, and then the mode changes to the 1/32 frequency SLOW mode. Next, the mode changes to the 1/2 frequency SLOW mode. When transferring from the 1/2 frequency NORMAL mode to the 1/2 frequency SLOW mode, the system clocks fosc/1024 and fxi/32 between steps (5) and (6) in Fig. 10-1-6 are omitted.

(6) Before dropping through the operating power source voltage range of the SLOW mode, the mode switches to the HALT1 mode.
In this case, it is necessary to transfer from the 1/2 frequency SLOW mode to the HALT1 mode. (Even the clock stops in the STOP1 mode since the XI/XO oscillation stops in this mode. The clock counter continues to count in the HALT1 mode.) Before entering the standby mode, it is necessary to carry out low power consumption processing (☛ 10-1-1 Entering the Standby Mode) and to enable return from interrupt.

(7) The voltage recovery interrupt is accepted and the mode returns to the 1/2 frequency SLOW mode.

(8) When the power source voltage (Vdd) becomes sufficient to operate at fs=fosc/2, the mode is switched to the 1/2 frequency and the program inserts an OSC stabilization wait interval. After the oscillation stabilization wait is completed, the mode is switched to NORMAL divided by 2 (fs=fosc/2).

(9) If a reset has occurred for some reason (external input or self-reset), OSC1/OSC2 oscillation is stopped and the XI/XO pin continues to oscillate until reset is released. (The clock counting with XI source also continues.) The contents of RAM are not initialized at reset.

(10) After reset is released, an OSC oscillation stabilization wait interval of approximately 18 ms (@14.32 MHz) is automatically inserted.

(11) After the oscillation stabilization wait is completed, operation starts in the NORMAL mode with fs (system clock) = fosc/1024.

10-1-4 Transition between SLOW and NORMAL Modes

There are two types of CPU operation modes in the microcomputer with on-chip OSD servo, namely, the NORMAL and SLOW modes with XI oscillation present (pin SXI = 'H'). It is necessary to temporarily pass through the IDLE mode when changing over from a frequency divided NORMAL mode other than the 1/2 frequency Normal mode to the SLOW mode and when changing over from the SLOW mode to the NORMAL mode.

The system clock can be monitored by reading the OSC0 Flag (bp0) of the CPUM Register (x'3F00'). This bit can be used to verify whether the system clock supplied to peripheral devices is a high-speed clock or a low-speed clock. When OSC0 is '0', a high-speed clock (fosc) is being supplied. If OSC0 is '1', a low-speed clock (fxi) is being supplied.

■ Changing over from the NORMAL mode to the SLOW mode

(1) It is possible to change over directly from the 1/2 frequency NORMAL mode to the 1/2 frequency SLOW mode.

(2) When changing over from the 1/1024 frequency NORMAL mode to the 1/32 frequency SLOW mode, it is necessary to first wait for the low-speed oscillation to become sufficiently stable, then change the mode temporarily from the 1/1024 frequency NORMAL mode to the 1/32 frequency IDLE mode, waiting 20 or more machine cycles by software, change the mode to the 1/32 frequency SLOW mode, and then finally to the 1/2 frequency SLOW mode.
When low-speed oscillation is still unstable immediately after reset is released, the program as shown below is necessary to change over from the 1/1024 frequency NORMAL mode to the 1/2 frequency SLOW mode. The loop for stabilizing the low-speed clock oscillation is not required when the low-speed clock oscillation has already become stable.

Program 2 (The following is executed in the 1/1024 frequency NORMAL mode.)

```
        MOV   x'FF', D2      ;Loop for stabilizing the oscillation of the low-speed clock.
LOOP3   MOV   x'FF', D1      ;T=(((5 × x'15')+7) × x'FF'+7) × x'FF'/fs
LOOP2   MOV   x'15', D0      ;=1S
LOOP1   ADD   -1, D0
        BNE   LOOP1
        ADD   -1, D1
        BNE   LOOP2
        ADD   -1, D2
        BNE   LOOP3
        BSET  (CPUM)0        ;Transfer from 1/1024 frequency NORMAL mode to 1/32 frequency IDLE mode.
        MOV   x'04',D0       ;Stabilization loop for changing over from high-speed to low-speed clock.
LOOP4   ADD   -1, D0         ;20 or more machine cycles are required.
        BNE   LOOP4
        BSET  (CPUM)1        ;Transfer from 1/32 frequency IDLE mode to 1/32 frequency SLOW mode.
        MOV   (CPUM), D0     ;Transfer from 1/32 frequency SLOW mode to 1/2 frequency SLOW mode.
        AND   x'0F, D0
        MOV   D0, (CPUM)
```

■ Changing over from SLOW mode to NORMAL mode

Changing over from a SLOW mode to a NORMAL mode can be made either from the 1/2 frequency SLOW mode to the 1/2 frequency NORMAL mode or from the 1/32 frequency SLOW mode to the 1/1024 NORMAL mode.
When changing over from the 1/2 frequency SLOW mode to the 1/2 frequency NORMAL mode, it is necessary to first change the mode temporarily to the 1/2 frequency IDLE mode, and wait by programming until the high-speed clock oscillation has become sufficiently stable.
The required period of waiting by programming for the oscillation to become stable in the IDLE mode when changing over from the SLOW mode to the NORMAL mode is given below.

Mode Change	Source for waiting for oscillation to become stable	Oscillation stabilization waiting period
From the 1/2 frequency SLOW mode to the 1/2 frequency NORMAL mode	OSC	18 ms by software (See Note 1)

Note 1:
The period waiting for oscillation to become stable when changing over from the SLOW mode to the NORMAL mode should be set to twice the time duration in which the oscillation reaches the full-swing state by actually monitoring the OSCO pin.

Program 3

```
        BCLR    (CPUM) 1        ;1/2 frequency SLOW -> 1/2 frequency IDLE
        MOV     x'3B', D0       ;Loop for high-speed clock oscillation to stabilize
LOOP    ADD     -1, D0          ;T= (5 × x'3B') / fs=18mS
        BNE     LOOP
        BCLR    (CPUM) 0        ;1/2 frequency IDLE -> 1/2 frequency NORMAL
```

10-1-5 Entering Standby Mode

Changing over from the NORMAL or SLOW mode to the standby mode is done by programming, and the return from the standby mode to the original NORMAL or SLOW mode is done by an interrupt. The transfer to the standby mode should always be made from the 1/2 frequency NORMAL mode or from the 1/2 frequency SLOW mode.

The following settings must be made before entering to a standby mode.

(1) Disable all interrupts by clearing the Interrupt Enable Flag (MIE) of the Processor Status Word (PSW) and the Interrupt Enable Flags (xxxIE) of the Maskable Interrupt Control Registers (xxxICR).

(2) Specify the interrupt source that will cause a return from the standby mode to the CPU mode. Set only the Interrupt Enable Flag (xxxIE) of that Maskable Interrupt Control Register (xxxICR). Then, set the MIE Flag of the PSW.

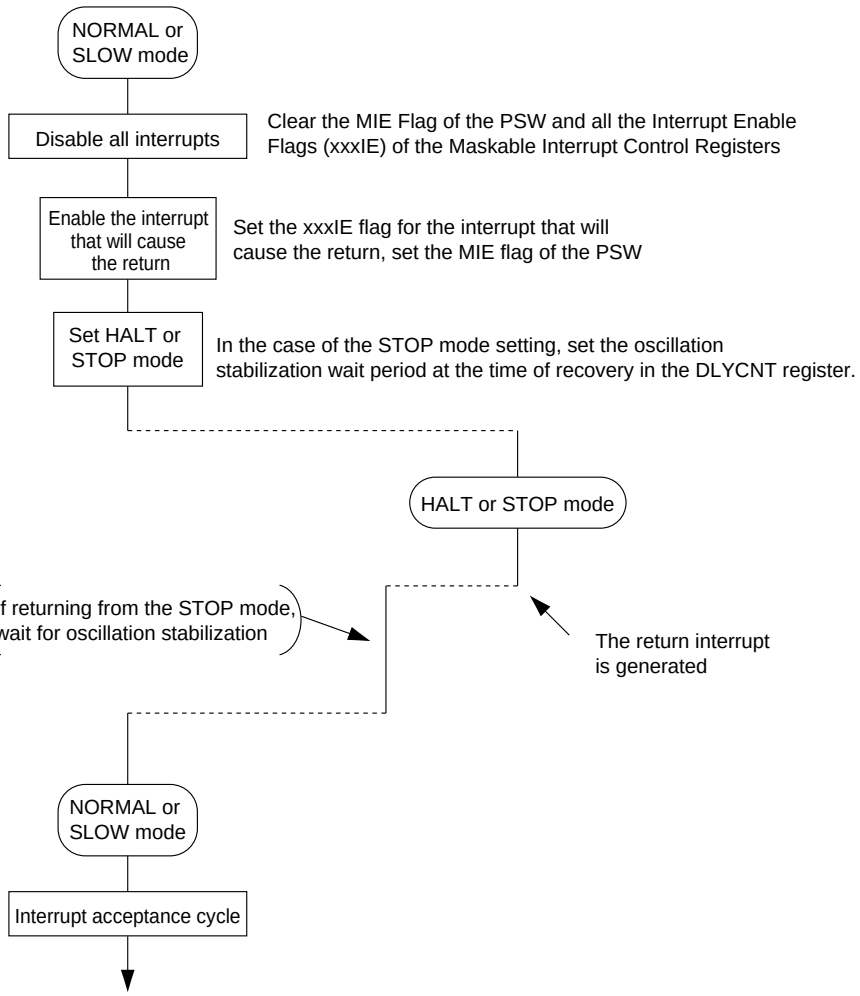


Figure 10-1-7 Transfer to Standby Mode and the Subsequent Return Sequence

Before entering the HALT or STOP modes, if interrupts are enabled but the priority level of the interrupts used is less than the mask level set in PSW, maskable interrupts will not be able to cause a return to a CPU operating state.

■ Entering HALT Mode

It is possible to enter the HALT0 mode only from the 1/2 frequency NORMAL mode and to enter the HALT1 mode only from the 1/2 frequency SLOW mode. In both cases, it remains oscillation and only the CPU stops. Returns from the HALT mode are performed by interrupts or by reset. When reset, the normal reset procedure will be performed. When interrupted, the mode will return to the one before entered the HALT mode. When entered, the HALT mode with the watchdog timer enabled, the watchdog timer will stop counting. Upon return to a CPU operating mode, counting will resume from where it left off.

Program 4 (The following is executed in the 1/2 frequency NORMAL mode or the 1/2 frequency SLOW mode.)

```
BSET    (CPUM)2         ;Set HALT mode.
NOP                     ;After writing in register CPUM, several dummy instructions
NOP                     ;(three or fewer instructions) are executed depending
NOP                     ;on the status of the pipeline.
```

■ Transfer to STOP Mode

It is possible to enter the STOP0 mode only from the 1/2 frequency NORMAL mode and the STOP1 mode only from the 1/2 frequency SLOW mode. In both cases, both the oscillation and the CPU stop. Returns from the STOP0 and STOP1 modes are performed by interrupts or by reset. When reset, the normal reset procedure will be performed. When interrupted, the mode will return to the one before entered the STOP mode.

Program 5 (The following is executed in the 1/2 frequency NORMAL mode or the 1/2 frequency SLOW mode.)

```
BCLR    (WDCNT) 0       ;The WD timer is stopped.
BSET    (WDCNT) 2       ;The WD counter is cleared.
BSET    (CPUM)3         ;The STOP mode is set.
NOP                     ;After writing in register CPUM, several dummy instructions
NOP                     ;(three or fewer instructions) are executed depending
NOP                     ;on the status of the pipeline.
```

When entering the STOP mode or the HALT mode, the watchdog timer should be disabled and the watchdog timer's counter should be cleared before switching the mode (clear the WDEN flag and set the WDCLR flag or the WDCNT register). Further, if the monitoring by the watchdog timer is necessary after returning from these STOP and HALT modes, always be sure to set the WDCLR flag and then enable again the watchdog timer (set the WDEN flag of register WDCNT).

10-1-6 Return from Standby Mode

The table below lists sources that cause a return from standby mode.

Table 10-1-1 Sources that Cause a Return from Standby Mode

Return cause \ Operating Mode	HALT0	HALT1	STOP0	STOP1
RESET (Reset CPU)	❍	❍	❍	❍
IRQ0 to 4, KEYIRQ0 to 4 (External signal interrupt)	❍	❍	❍	❍
TC0ICR (Timer 0 interrupt)	Δ*1	Δ*2	×	×
TC6ICR (Timer 6 interrupt)	Δ*3	Δ*4	×	×

❍: causes a return
Δ : conditional return
×: does not cause a return

****1 Operating condition (1 or 2 in the table below)***

		Clock source	Remarks
1	Timer 0	f_{OSC}/512	
2	Timer 0	f_{TC6}	32-Bit timer
	Timer 6	f_{OSC}/512	

****2 Operating condition (1 or 2 in the table below)***

		Clock source	Remarks
1	Timer 0	f_{XI}	
2	Timer 0	f_{TC6}	32-Bit timer
	Timer 6	f_{XI}	

****3 Operating condition (1 in the table below)***

		Clock source	Remarks
1	Timer 6	f_{OSC}/512	

****4 Operating condition (1 in the table below)***

		Clock source	Remarks
1	Timer 6	f_{XI}	

10-2 Reset Function

If the NRST pin is pulled to a low-level, the CPU will be reset, and flags, registers, etc. will be initialized. (There are registers that are not initialized. Even the RAM contents are undefined.)

After switching on the power and the RAM contents have been initialized, the RAM contents are retained even after a reset unless the power is switched off again or the power supply voltage is decreased below the voltage necessary for RAM data retention.

(1) When the NRST pin changes from a low to high-level, Timer 5 operates as an oscillation stabilization wait timer and begins counting at either the system clock generated at OSCI/OSCO. Reset is released when the oscillation stabilization wait timer overflows.

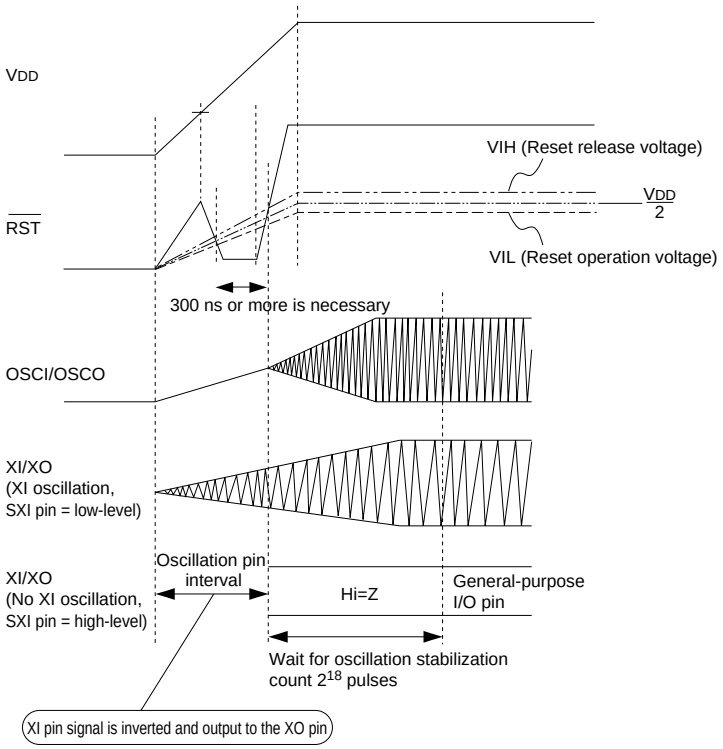


Figure 10-2-1 Reset Release Sequence

Maintain the NRST pin at the 'L' level for at least 300 ns.

(2) There are 2 methods of transferring to the reset state: when the value of VDD drops and the NRST pin is brought to a low-level, or by setting the P0OUT4 flag (bp4) of the Port 0 Data Output Register (P0OUT) to '0' and resetting by software instruction.

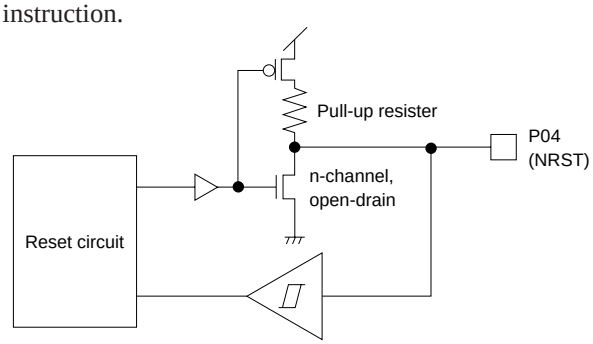


Figure 10-2-2 Internal Configuration of the Reset Pin

Reset can be initiated by software instructions for the initial start-up after a runaway interrupt has been generated (P1).

The P0OUT4 flag is write-only. When setting other bits in the P0OUT Register, the P0OUT4 flag must be set to '1'.

10-3 Power Supply Control

■ When turning off AVDD

It is possible to transfer from the state of VDD=VDD=AVDD=5V to the state in which VDD=VDD2=5V and AVDD=Off.

(1) Functions that no longer operate when AVDD is turned off

- A/D Conversion
- Cylinder/capstan amplifier
- Control playback amplifier, slicer, saturation detection
- Trapezoidal waveform generation (TPZ)
- PG Shifter (PGMM)

(2) Procedure for turning off AVDD

1. Turn off the analog power supply.
   (Clear the STBH flag (bp0 of register ANACNT x'3F81').)
2. Turn off the A/D converter power supply.
   (Clear the ADEN flag (bp4 of register ADM x'3F81').)
3. Set the AD ports (port 8, port C) as outputs 'L'.
   Set x'FF' in P8DIR (x'3F34'), x'FF' in PCDIR (x'3F37'), x'00' in P8 (x'3F29'), and x'00' in PC (x'3F2D').
4. Turn off AVDD.

■ When Turning Off VDD2

It is possible to transfer from the state of VDD=VDD=AVDD=5V to the state in which VDD=AVDD=5V and VDD2=Off.

(1) Functions that no longer operate when VDD2 is turned off

- OSD Functions, XDS functions

(2) Procedure for turning off VDD2

1. Carry out steps 7-13 described in "Method of reducing the power supply consumption in the STOP mode and the HALT mode" to be described later in Section 10-1-1 Transfer to the Standby Mode.
2. Turn off VDD2.

■ When Turning Off AVDD and VDD2

Turn off AVDD and VDD2 after making the register settings for turning off AVDD and VDD2 described above.

Chapter 11 OSD and XDS Functions

11

11-1 OSD Function

11-1-1 OSD Function Summary

Compatible with menu display or superimposed display

Compatible broadcast formats:	NTSC, PAL, PAL-M, PAL-N
Screen configuration:	24 characters × 2n lines (where n = 1 to 6)
Character types:	512 character types maximum (variable in 1 character units)
Character size:	12 dot × 18 dot
Character enlargement:	Settings of 2× , 3× , 4× are possible in both horizontal and vertical directions
Character interpolation:	None
Background color:	8 colors (during menu display, can be specified for each row)
Background brightness:	8 level adjustments, can be specified for each row
Character color:	White (When digital output is used, one of 8 colors can be specified for each character.)
Character brightness:	8 level adjustments, can be specified for each row
Border function:	1 dot border in 4 or 8 directions
Border brightness:	4 level adjustments, can be specified for each row
BOX shadow function:	Can be specified for each character (only when there are 128 or more character types during composite output)
Blinking:	None (Blinking can be implemented in software.)
Highlighted characters:	Can be specified for each character.
Halftone:	Can be specified for each row, 2 brightness level adjustments (set for each row)
Input:	Composite video signal input
Clamp method:	Sync chip clamp, 4 clamp levels
Output:	Composite video signal output Digital output (6 pins) During digital output, the color and background of each character can be selected from 8 colors
Vibration resistant display:	Built-in AFC circuit
Interrupt:	Generated after displaying the row specified by a register

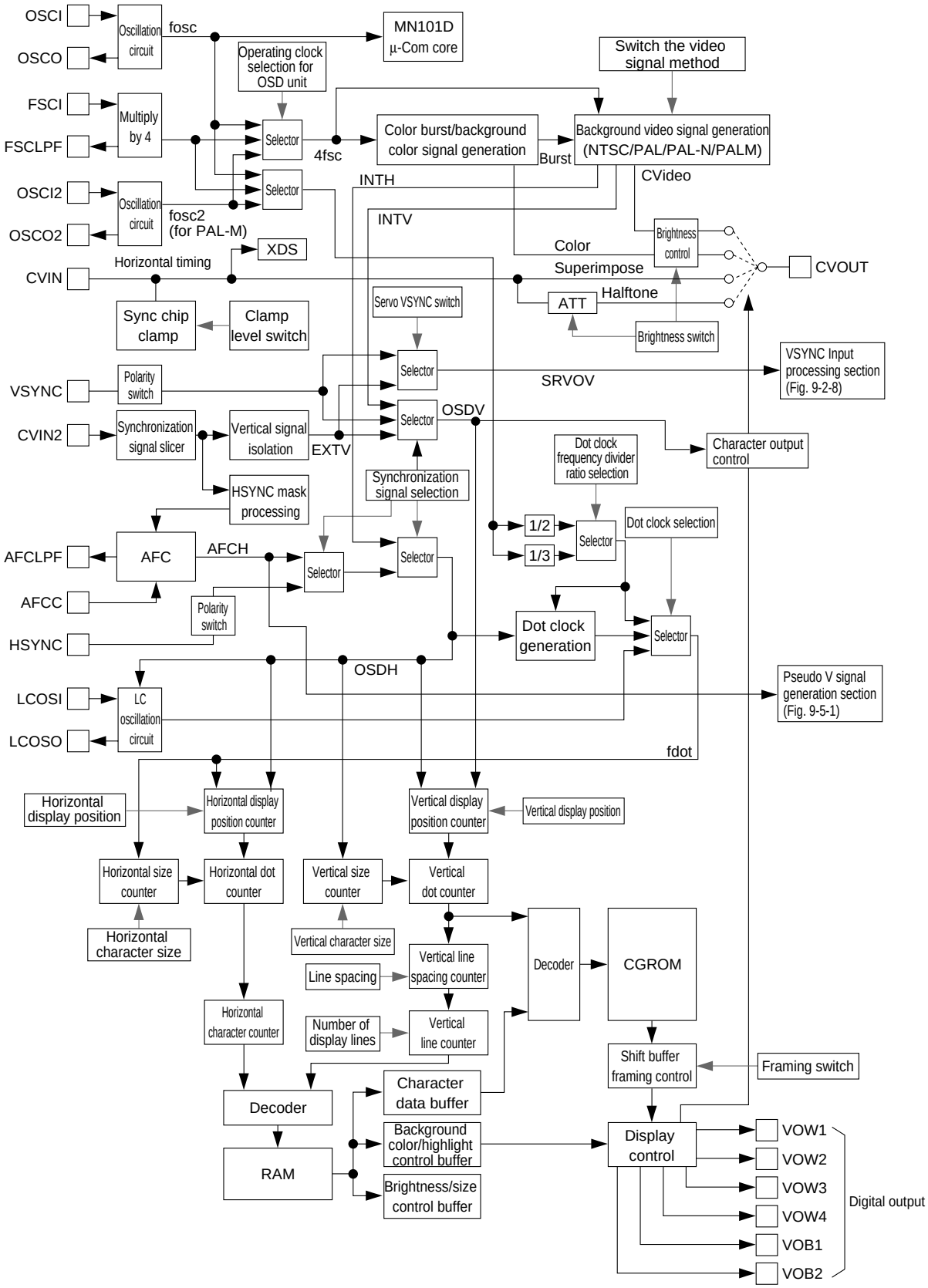


Figure 11-1-1 OSD Unit, Block Diagram

11-1-2 Pin Configuration

The OSD pin configuration is shown in Table 11-1-1.

Table 11-1-1 OSD Pin Summary

Pin Name	Pin Number	I/O	Function
VDD2	51	Input	OSD analog voltage (+) terminal
VSS2	48	Input	OSD analog voltage (–) terminal
VSYNC	45	Input	External vertical sync signal input (also functions as port 20)
HSYNC	46	Input	External horizontal sync signal input (also functions as port B7)
CVOUT	47	Output	Video signal (composite) output pin (also functions as port B6)
CVIN	49	Input	External video signal (composite) input pin (also functions as port B5) Also functions as XDS data input pin
CVIN2	50	Input	External video (composite) input pin for synchronous, separate signals (also functions as port B4) Also functions as XDS synchronous, separate input
AFCC	52	Input	Mounting pin 1 for horizontal sync AFC active RC filter (also functions as port B3)
AFCLPF	53	Output	Mounting pin 2 for horizontal sync AFC active RC filter (also functions as port B2)
FSCI	54	Input	Clock input pin when OSD clock is multiplied by 4 or 2 (also functions as port B1)
FSCLPF	55	Output	Filter mounting pin when clock is multiplied by 4 or 2 (also functions as port B0)
LCOSO	56	Output	LC connection pin 1 when the OSD dot clock is LC oscillation (also functions as port A7)
LCOSI	57	Input	LC connection pin 2 when the OSD dot clock is LC oscillation (also functions as port A6)
VOB2	58	Output	Character frame output pin during digital OSD output (also functions as port A5)
VOB1	59	Output	Display interval output pin during digital OSD output (also functions as port A4)
VOW4	60	Output	RGB mix output pin during digital OSD output (also functions as port A3)
VOW3	61	Output	R output pin during digital OSD output (also functions as port A2)
VOW2	62	Output	G output pin during digital OSD output (also functions as port A1)
VOW1	63	Output	B output pin during digital OSD output (also functions as port A0)
OSCI2	43	Input	Crystal connection pin 1 when dedicated OSD clock is selected (also functions as port 22)
OSCO2	44	Output	Crystal connection pin 2 when dedicated OSD clock is selected (also functions as port 21)

11-1-3 Control Registers

OSD control registers are listed in Table 11-1-2.

Table 11-1-2 OSD Register Summary

Register Name	Address	R/W	Function
OSDCNT1	x'3FC8'	R/W	OSD Operation Control Register 1 (OSD clock selection, TV mode selection)
OSDCNT2	x'3FDD'	R/W	OSD Operation Control Register 2 (Output mode, Character type, Display format selection)
OSDCNT3	x'3FB9'	R/W	OSD Operation Control Register 3 (Synchronization method, Screen background selection)
BKALCL	x'3F0E'	R/W	Screen Background Color Control Register
VP	x'3FD8'	R/W	Vertical Display Start Position Control Register
HP	x'3FD9'	R/W	Horizontal Display Start Position Control Register
VOFS	x'3FDA'	R/W	Display Line Spacing Control Register
VLIN	x'3FDB'	R/W	Display Control Register (display line, background control)
VCOCNT	x'3FD0'	R/W	VCO Control Register (VCO control for horizontal sync AFC)
CLCNT	x'3FD1'	R/W	Video Signal Control Register (clamp, through, output level, halftone, character brightness)
DOTCNT	x'3FD2'	R/W	Dot Clock Control Register (clock source, phase adjustment)
PCSTA	x'3FD3'	R/W	Horizontal Sync AFC Control Register (LPF power source, OSD clock multiplication selection)
CMPLIN	x'3FDC'	R/W	OSD Interrupt Display Line Control Register
ROMEND	x'3FC9'	R/W	CGROM Last Address Control Register

11-1-3-1 OSDCNT1 OSD Operation Control Register 1

The OSDCNT1 register sets the operation mode. Settings are listed below.

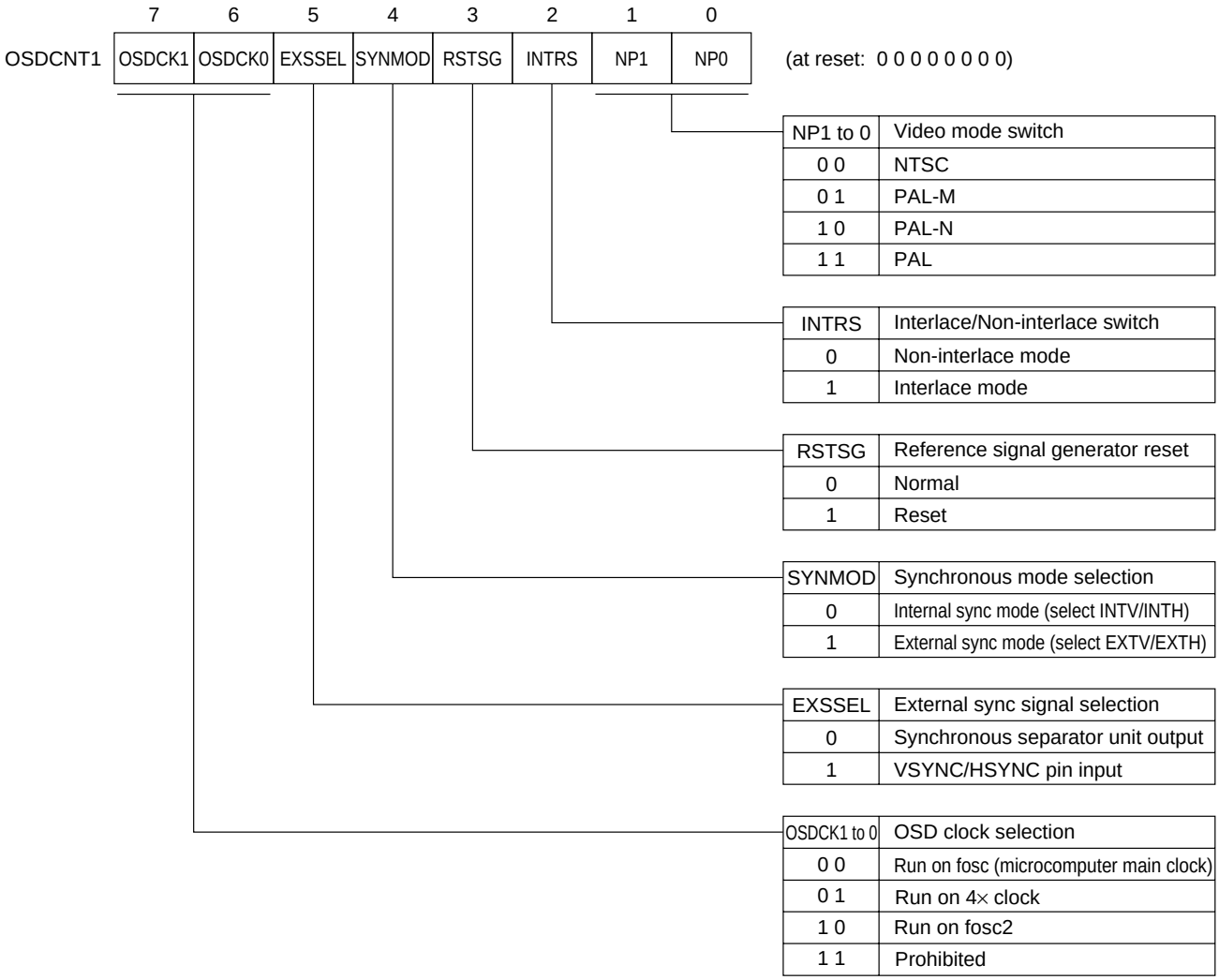


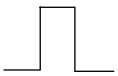
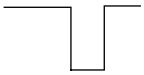
Figure 11-1-2 OSD Operation Control Register 1 (OSDCNT1: x'3FC8', R/W)

1. OSD Clock Selection (bits 7 and 6)

These bits select the clock of the TV signal generator for OSD, the XDS operation clock, and the clock of the summation counter of the sync separator section. This selection sets the clock as 4× the TV mode sub-carrier frequency.

2. External sync signal selection (bit 5)

This bit selects the horizontal and vertical sync signals for the OSD unit during superimposed video. If this bit is '0', the signal input to CVIN2 is generated by the internal synchronous separator circuit and the horizontal sync AFC. If this bit is '1', externally input vertical and horizontal sync signals are selected. The horizontal and vertical signals are input to HSYNC (pin 46) and VSYNC (pin 45), respectively. A polarity inversion function is available.

	PASEL (bit 6)		P2SEL (bit 0)	
	0	1	0	1
HSYNC Pin input waveform	Active High 	Active Low 		
VSYNC Pin input waveform			Active High 	Active Low 

3. Synchronous mode selection (bit 4)

The internal sync mode outputs the video signal generated by the internal clock of the microcomputer. The external sync mode is used during superimposed video. This bit selects these two modes.

4. TV reference signal generator reset (bit 3)

Setting this bit to '1' will stop the microcomputer's internal TV reference signal generator. Depending upon the selected TV mode, changing the setting from '1' to '0' will start the generator at a position ±1H from the vertical sync signal start position. When switching from external sync to internal sync mode, continuity of the vertical signal can be preserved by using the VSYNC position to temporarily reset the generator.
The TV reference signal generator creates the speed reference signal for the horizontal sync AFC and the VCO calibration reference signal. Therefore, do not reset the generator during calibration or during external sync mode.

5. Interlace/Non-interlace switch (bit 2)

This bit selects either interlace or non-interlace during the internal sync mode. Immediately after this bit is changed, the microcomputer's TV reference signal generator will be reset, and will start from the vertical sync signal.

6. Video mode switch (bits 1 and 0)

These bits select the TV signal mode. Because signals from the TV reference signal generator are used for the horizontal sync AFC and VCO calibration, it is not recommended to write or modify these bits after initial settings have been made. As in the case of bit 5, immediately after these bits are changed, the microcomputer's TV reference signal generator is reset, and will start from the vertical sync signal.

11-1-3-2 OSDCNT2 OSD Operation Control Register 2

The OSDCNT2 register sets the OSD display format. Settings are listed below.

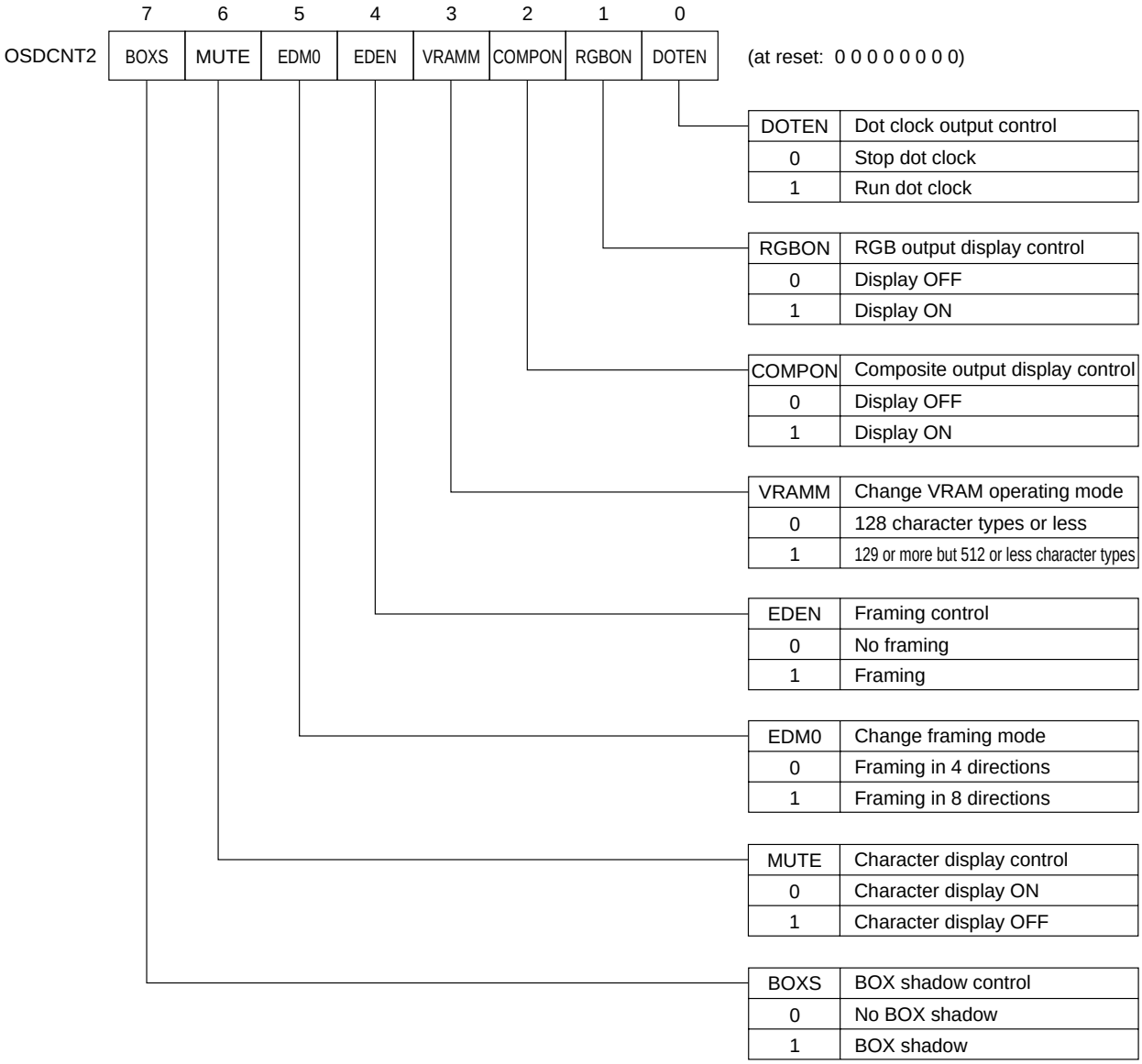


Figure 11-1-3 OSD Operation Control Register 2 (OSDCNT2: x'3FDD', R/W)

1. BOX shadow control (bit 7)

This bit selects whether or not there will be BOX shadowing. This bit is only valid for composite signals when VRAMM (bit 4) is '1'. During digital output, this bit has no effect. (Fig. 11-1-28)

2. Character display control (bit 6)

This bit controls the character output. When this bit is '1', although the line background and screen background are output, character output, frame output, and box shadow output are not performed (during composite output).

3. Framing mode switch (bit 5)

This bit switches the character framing mode. If '1', the character is framed in eight vertical, horizontal and diagonal directions. (Fig. 11-1-27) If '0', framing is in four vertical and horizontal directions. (Fig. 11-1-26)

4. Framing control (bit 4)

This bit selects whether or not there is character framing. (Fig. 11-1-25) This selection is valid for both composite and digital output.

5. VRAM operation mode switch (bit 3)

This bit switches the OSD function and the type of characters that can be used. If '0', 128 character or less can be used. If '1', between 129 and 512 characters can be used. For composite signals, the BOX shadow function is selected. For RGB digital output, the color of each individual character can be specified. Be careful when setting this bit, VRAM allocation will change depending upon the selection.

6. Composite output control (bit 2)

This bit selects whether a composite signal will be output. If set to '1', a video signal is output from the CVOUT pin.

7. RGB digital output display control (bit 1)

This bit turns ON/OFF the digital output of display data to pins VOB1, VOB2, VOW1, VOW2, VOW3, and VOW4. During digital and composite displays, VRAM CCn and CDn settings for individual character control will differ. Be careful.

In the case of the digital OSD output signal, it is not possible to delete characters or output only the line background color mentioned at right.

8. Dot clock output control (bit 0)

This bit selects whether to supply the dot clock to the OSD unit. By making this bit '0', it is possible to delete the character display data in the video signal of CVOUT, and to output only the line background color specified in VRAM. (This is not possible for external mode composite signals.)

11-1-3-3 OSDCNT3 OSD Operation Control Register 3

The OSDCNT3 register controls the OSD operation mode and consists of the following settings.

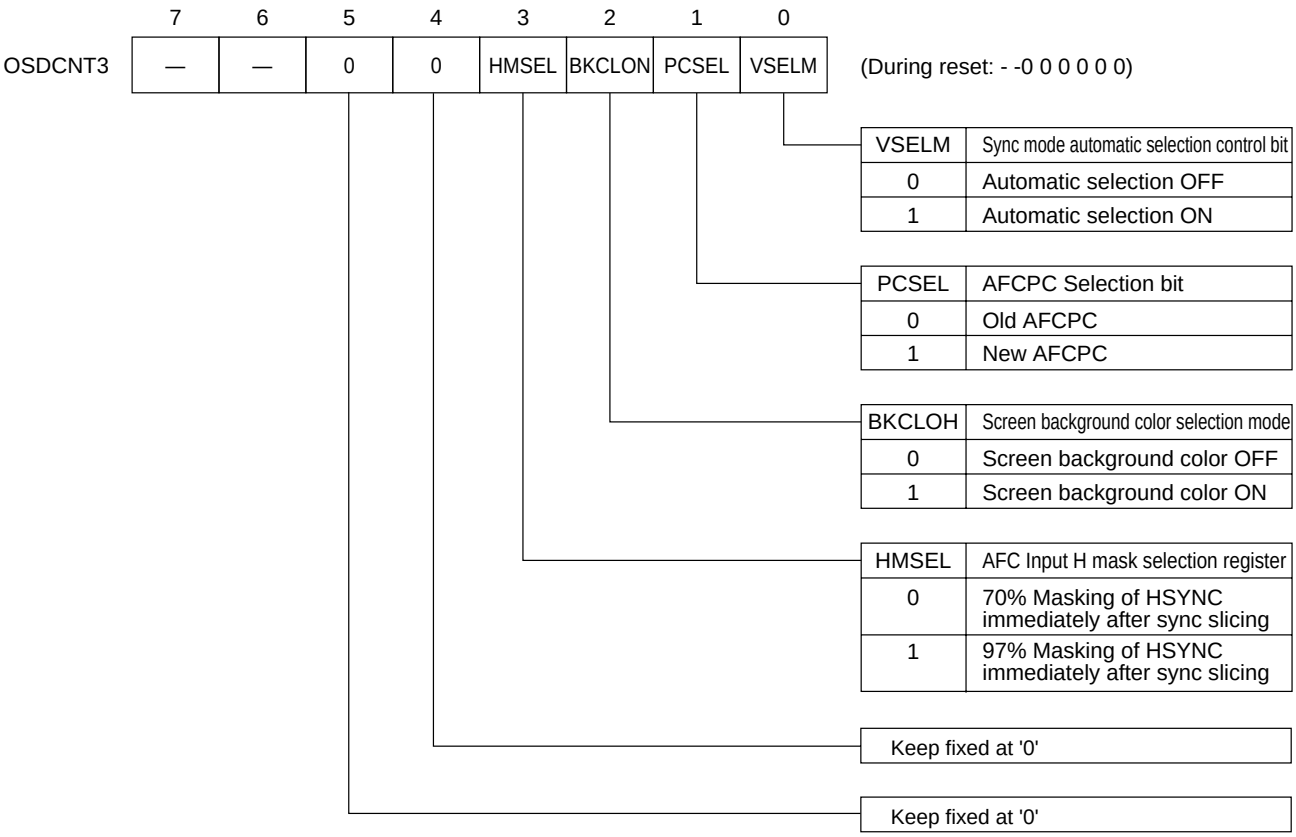


Fig. 11-1-4 OSD Operation Control Register 3 (OSDCNT3: x'3FB9', R/W)

1. AFC Input H mask selection (bit 3)

It is possible to mask the signal immediately after sync slicing for the Hsync input signal to the AFC circuit generating the AFCH signal, which is the clock for OSD display. (EXTHSEL flag = '1', bit 4 of the SDFLG register x'3FBF')
70% masking is done when the value of the HMSEL flag is '0', and 97% masking is done when it is '1'.

2. Screen background color selection mode (bit 2)

This bit turns ON and OFF the screen background color. This selection is effective in the composite output during internal synchronization.
The specification of the screen background color is made in the BKALCL register (x'3F0E').
The following settings (1) to (4) are required for enabling the screen background color.

(1) BKCLON Flag (OSDCNT3 register x'3FB9', bit 2) = 1
(2) SYNMOD Flag (OSDCNT1 register x'3FC8', bit 4) = 0
(3) BKMONO Flag (VLIN register x'3FDB', bit 5) = 0
(4) BAKSEL Flag (VLIN register x'3FDB', bit 6) = 0

3. AFCPC Selection (bit 1)

This bit selects the phase comparator. When this bit is '0', the same comparator (old) as that in the MN101D01 Series is enabled and the new phase comparator is disabled. When this bit is '1', the new phase comparator is enabled, the old comparator is disabled, and the power supply of the active filter becomes OFF.

4. Synchronization mode automatic selection control (bit 0)

This bit turns on or off the function for continuous automatic switching of vertical synchronization from external synchronization to internal synchronization (the vertical dancing prevention function). When set to 0, automatic switching is turned off, and the synchronization mode can be switched by directly writing the SYNMOD flag (OSDCNT1 register (x’3FC8) bit 4). When set to 1, automatic switching is turned on, and, after the user writes the SYNMOD flag, the synchronization mode switches at the VSYNC position. (When there is no VSYNC signal, this occurs with the timing of the vertical counter that uses OSDHSYNC as its clock.)

11-1-3-4 BKALCL Screen Background Color Control Register

This BKALCL register is used for specifying the color phase, brightness, and chrominance of the screen background color when the screen background color has been made ON, and permits the following settings.

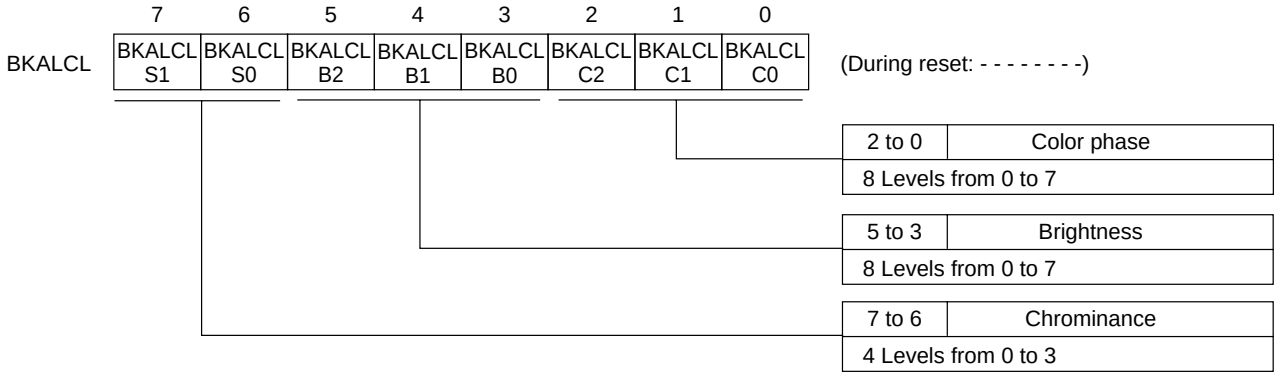


Fig. 11-1-5 Screen background color control register (BKALCL: x'3F0E', R/W)

This BKALCL register is effective during composite output when the screen background color is ON. (This register does not affect the digital output.)
The areas other than the character display area and the "through" area in the screen are changed to the screen background color. The background color of the character display area (including space codes) other than the "through" codes are the line background color.
The following settings are required for enabling the screen background color.

BKCLON	(OSDCNT3 x'3FB9' bp2)=1
SYNMOD	(OSDCNT1 x'3FC8' bp4)=0
BKMONO	(VLIN x'3FDB' bp5)=0
BAKSEL	(VLIN x'3FDB' bp6)=0

Color spreading may appear at the boundary areas depending on the combination of the line background color and the screen background color.

Figure 11-1-38 shows the screen display when the screen background color has been enabled in the top part settings (screen background color, character background present, reverse specification enabled) in the internal synchronization mode screen display.

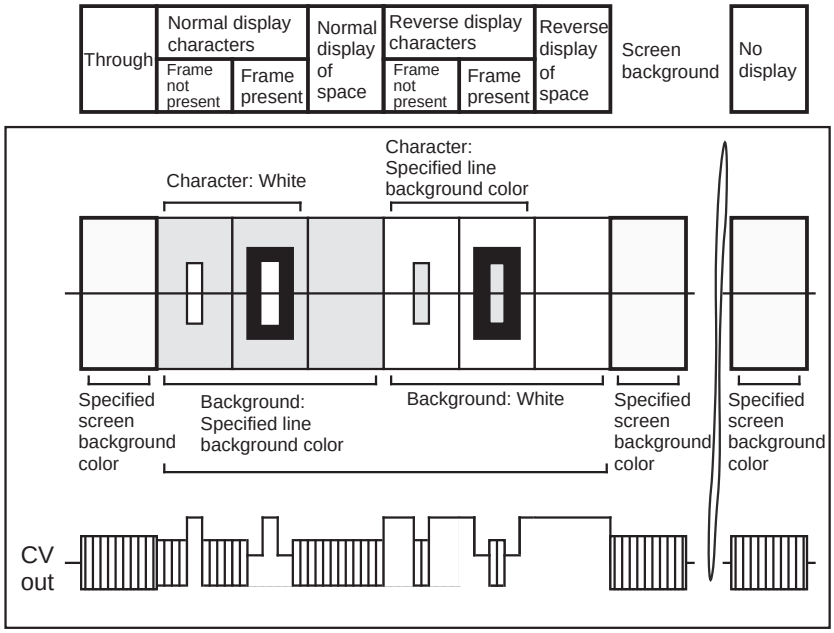


Fig. 11-1-6 Screen display when the screen background color is enabled.

1. Screen background color chrominance control (bits 7, 6)

These bits specify the chrominance of the specified screen background color when the screen background color has been enabled during composite output (4 levels).
These control bits do not affect the control of digital output.

2. Screen background color brightness control (bits 5~3)

These bits specify the brightness of the specified screen background color when the screen background color has been enabled during composite output (8 levels).
These control bits do not affect the control of digital output.

3. Screen background color phase control (bits 2~0)

These bits specify the color phase of the specified screen background color when the screen background color has been enabled during composite output (8 levels).
These control bits do not affect the control of digital output.

11-1-3-5 VP Vertical Display Start Position Control Register

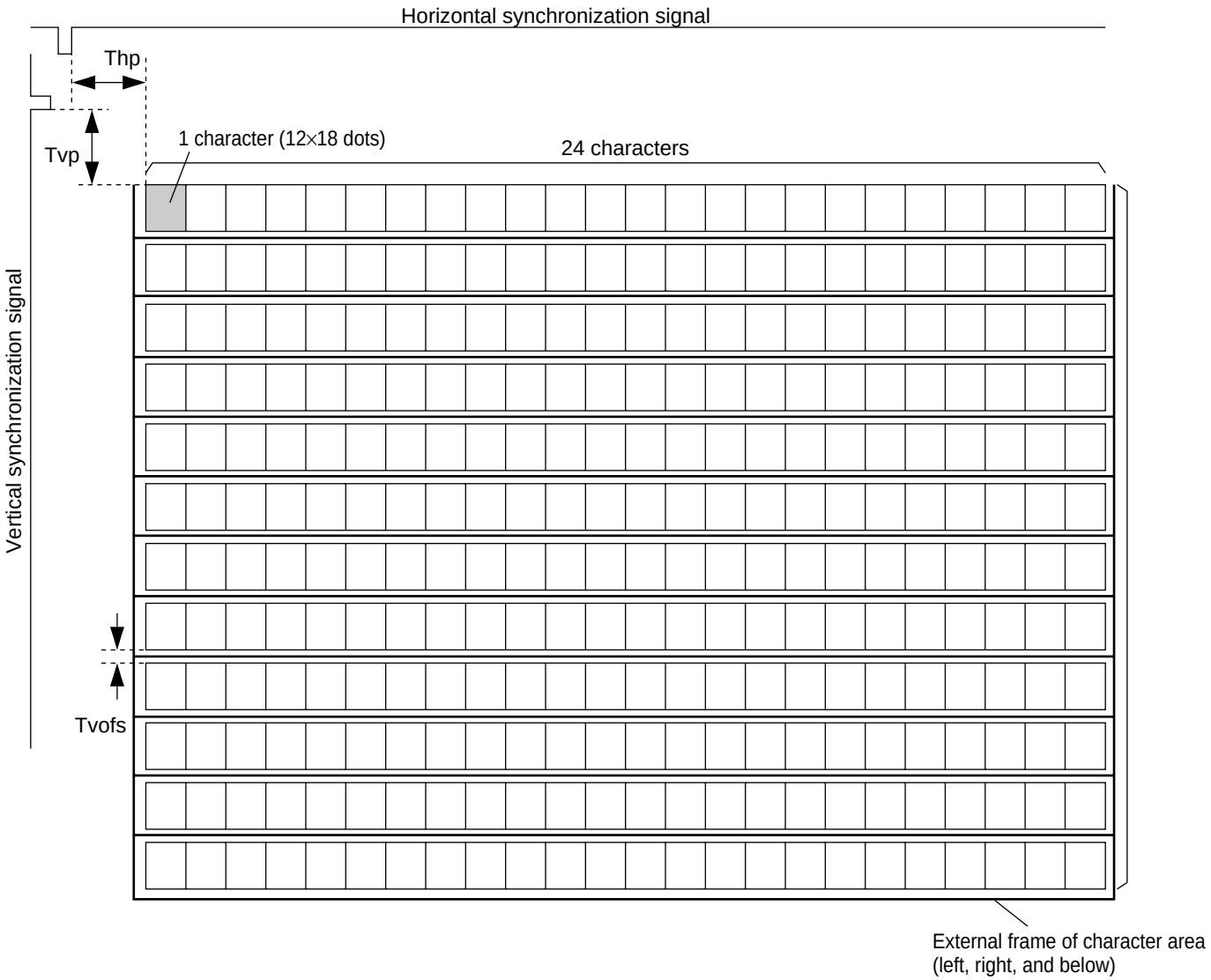


Figure 11-1-7 Screen Configuration

Figure 11-1-7 shows the screen configuration. One square corresponds to a single character consisting of 12× 18 dots. The VP vertical display start control register sets the value of Tvp shown the figure. Each setting value of 1 corresponds to moving two horizontal lines. After VSYNC is completed, set 1/2 the value of the HSYNC number.

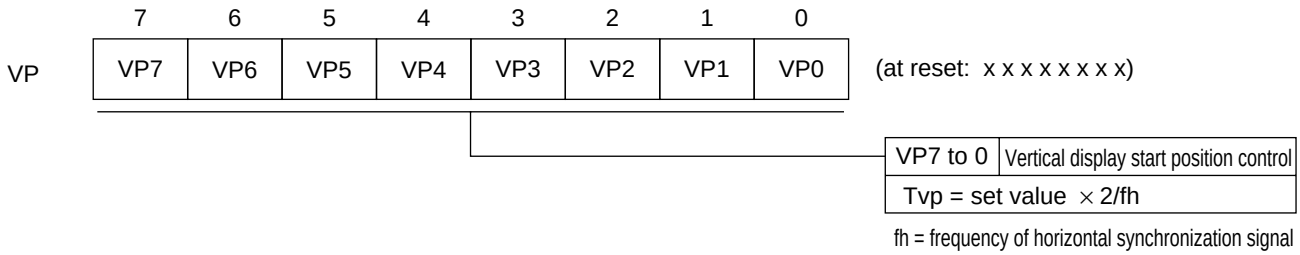


Figure 11-1-8 Vertical Display Start Position Control Register
(VP: x'3FD8', R/W)

11-1-3-6 HP Horizontal Display Start Position Control Register

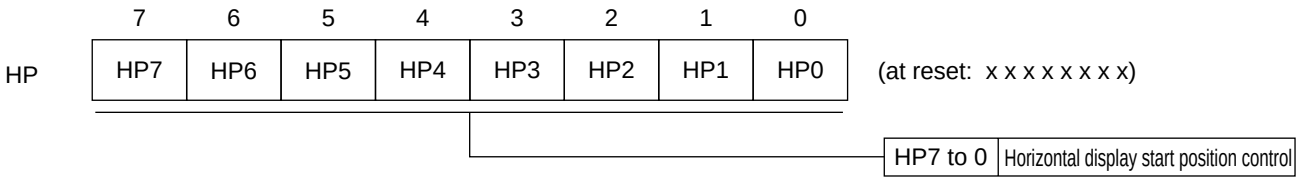


Figure 11-1-9 Horizontal Display Start Position Control Register
(HP: x'3FD9', R/W)

HP set value =(time difference from HSYNC until the display position) ÷ dot clock cycle – 27

The HP horizontal display start position control register sets the value of Thp shown in Figure 11-1-7. At a setting value of 1, two dot clocks are moved. After HSYNC is completed, the setting value of 0 becomes the position at 27 dot clocks.

11-1-3-7 VOFS Display Line Interval Control Register

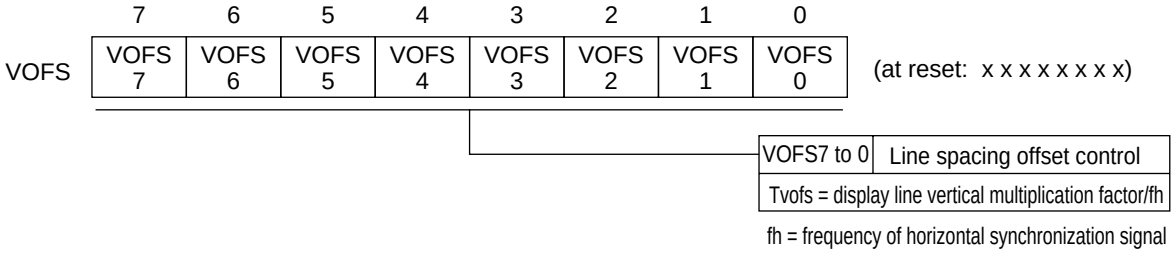


Figure 11-1-10 Display Line Interval Control Register (VOFS: x'3FDA', R/W)

VOFS display line interval control register sets the value of Tvofs shown in Figure 11-1-7. At a setting value of 1, the number of horizontal lines moved are the same as the vertical multiplication factor displayed previously. For example, with a setting of 1, the line spacing below the display line of a character having a vertical multiplication factor of 4, will be 4 horizontal lines. If VOFS is set to a value of 1 or greater, framing display is only possible one line below the display line.

11-1-3-8 VLIN Display Control Register

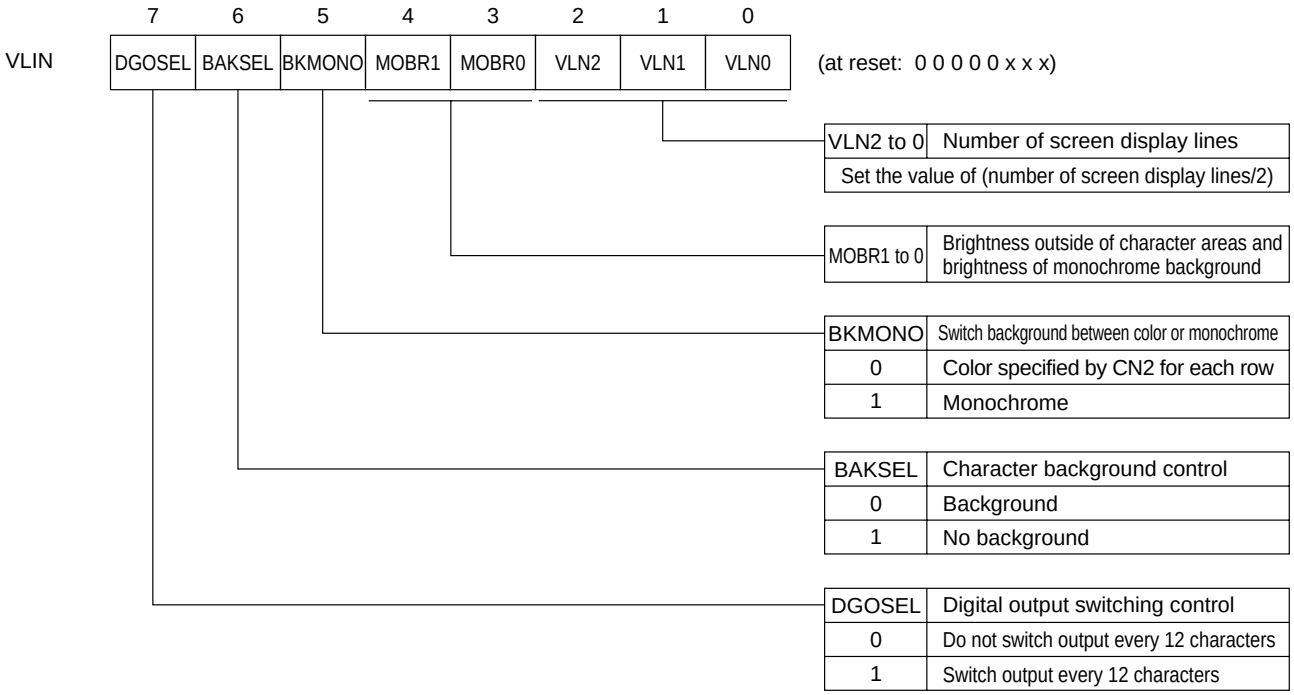


Figure 11-1-11 Display Control Register (VLIN: x'3FDB', R/W)

The VLIN Display Control Register controls the status of the OSD display. The control by each bit is described below.

1. Digital output switching control (bit 7)

There are two display modes, display by the composite signal output, VOB2 and VOW4, and display by VOB1, VOW1, VOW2 and VOW3. This bit enables both outputs to be switched at a position located at the halfway point of the OSD display area. If set to '1', the selection of bits 7 and 6 of the Line Control VRAM CN2 Register will be valid. If bit 7 (DOL) of that register is set to '1', the former OSD display will be valid on the left side of the display area, and the latter digital output will be invalid. If DOL is '0', the former OSD display will be invalid and the latter digital output display will be valid. Bit 6 of CN2 (DOR) controls the right side in a similar manner.
If this bit is '0', both the former and latter displays are valid. However, if bits 1 and 2 of OSDCNT2 (RGBON and COMPON) are not set to '1', neither the former or latter will be output.

2. Character background control (bit 6)

When this bit is set to 0, the system is set to a mode in which the character background is given a color in the composite video output. In this mode, line background color control by CN2 in the line control VRAM, background color level control (composite only), and CN1 background brightness control (composite only) will be enabled. During superimposed display, the character background will be the TV screen. During digital output, the VRAM CCN background color selection is enabled.
If this bit is set to '1', when the menu is displayed, the character background settings will be changed to monochrome, and the monochrome background brightness control of this register will be enabled. During superimposed video, the character background can be blackened, and VRAM CN0 framing brightness control 1 will be enabled. During digital output, if this bit is set to '1', character backgrounds will not be colored and the TV screen will be displayed "through". Under these conditions, highlighted characters will be white instead of TV screen "through".

3. Screen background color, monochrome switch (bit 5)

This bit sets color or monochrome for the screen background. If this bit is set to '1', the screen background will be monochrome. If '0', the screen background will be the line background color set by the VRAM CN2.

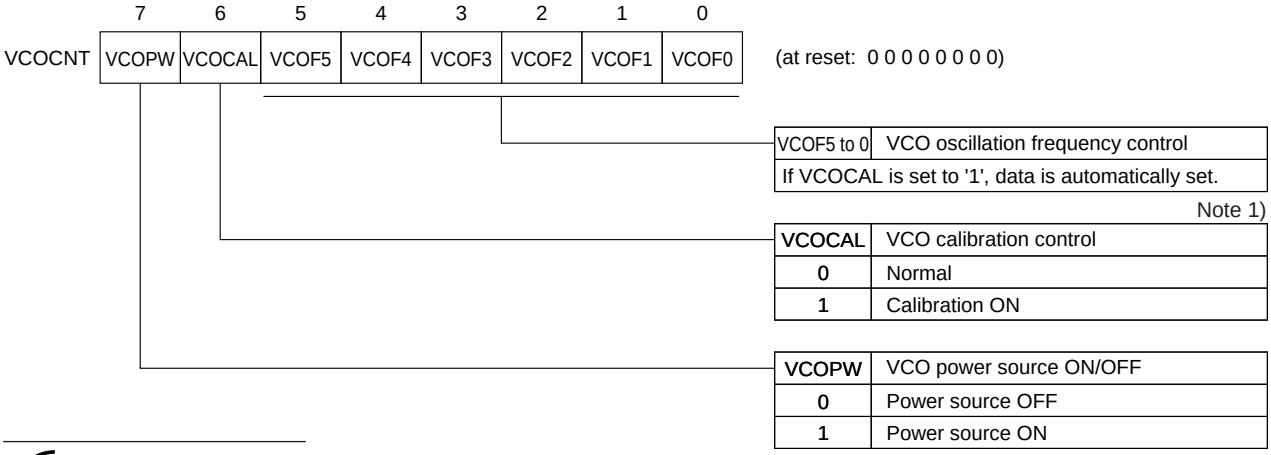
4. Monochrome background brightness setting (bits 4 and 3)

This bit controls brightness of the monochrome background when BAKSEL is set to '1' (to select a monochrome character background) and when BKMONO is set to '1' (to select a monochrome screen background).

5. Screen display line number specification (bits 2, 1 and 0)

These bits specify the number of lines to display on the screen. For every '1' set, the display will change by two lines. If '0' is set, the display will stop. The VRAM area can be decreased by setting a smaller value. Setting all 3 bits to '1' (x'7') is prohibited as it will exceed the display area.

11-1-3-9 VCOCNT VCO Control Register



Note1: In the case that VCOCAL='1', VCOF5 through VCOF0 must be set to '100000'.

Figure 11-1-12 VCO Control Register (VCOCNT: x'3FD0', R/W)

The VCOCNT (VCO Control register) controls the VCO status for horizontal sync AFC. The control functions of each bit are listed below.

1. VCO power ON/OFF (bit 7)

This bit controls the VCO power supply. During OSD display, the power supply must be ON. If displayed immediately after the power is turned on, with external sync mode, the display may be distorted. Please wait for the AFC horizontal sync stabilization interval before displaying. The power supply must also be ON if XDS data is to be read.
During low-power consumption states such as STOP or HALT, set this bit to '0' to turn OFF the power supply.

2. VCO calibration control (bit 6)

This bit starts and controls VCO calibration (center frequency calibration). Calibration begins when this bit is set to '1'. At that time, x'1000000' must be written to bit 5 through bit 0.
When calibration is completed, verify that this bit has become '0'. Calibration is performed in reference to the timing generated by the microcomputer's internal TV reference signal generator. During calibration, start the TV reference signal generator only after setting it to usable status (see OSDCNT1 settings).
The calibration interval is approximately 1 msec. Perform calibration when initializing register settings.

3. VCO oscillation frequency control (bits 5 through 0)

The value of these bits set the VCO center frequency. If the calibration described above is performed, an optimal value will automatically be written to these bits. When register settings are initialized, the calibration automatically sets this optimal value. Thereafter, it is not recommended to modify the setting.

11-1-3-10 CLCNT Video Signal Control Register

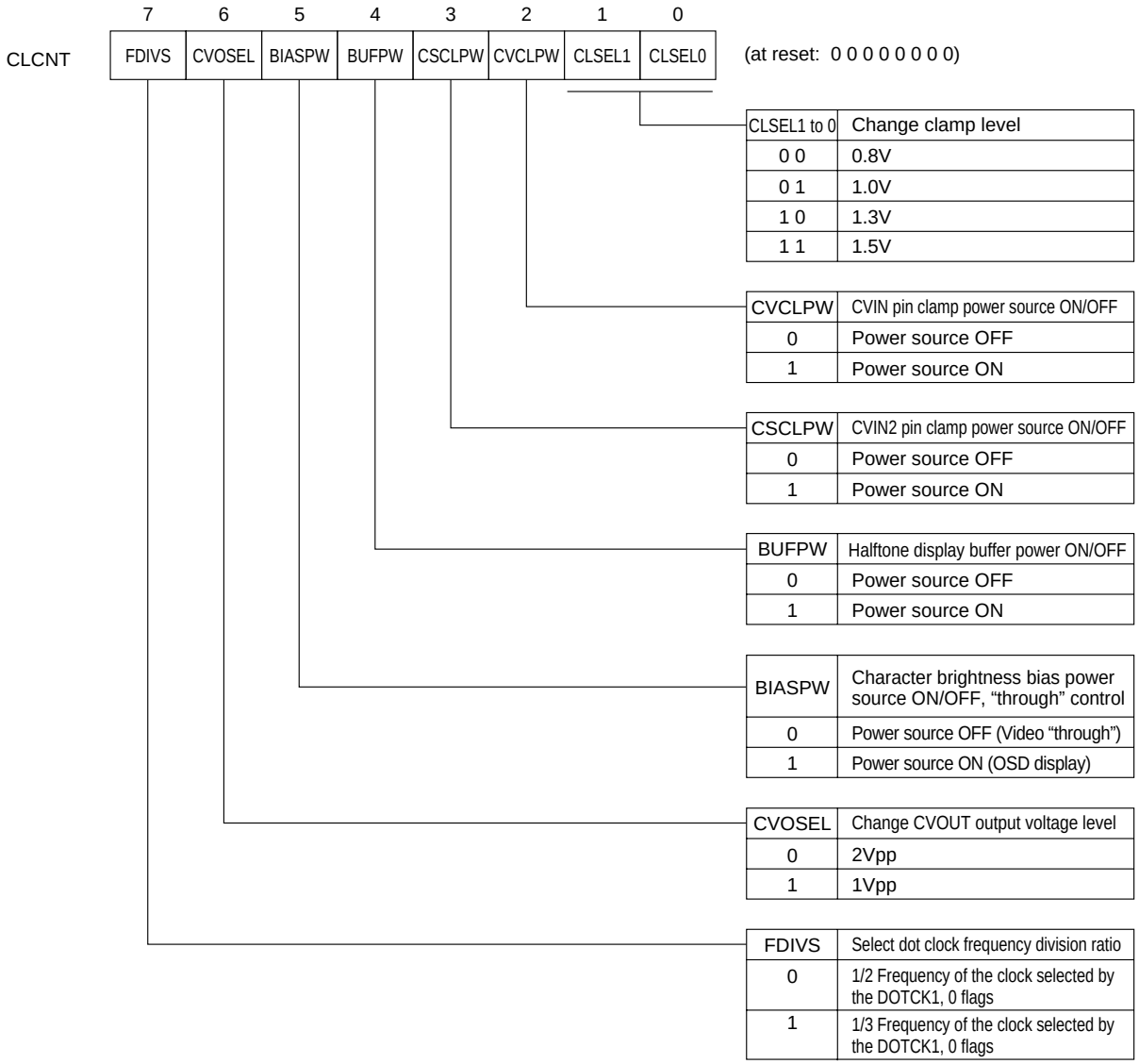


Figure 11-1-13 Video Signal Control Register (CLCNT: x'3FD1', R/W)

The CLCNT register performs control related to the composite video signal, and sets the frequency division ratio of the dot clock.

1. Dot clock frequency division ratio selection (bit 7)

This bit switches the dot clock frequency division ratio between 2 and 3. If a dot clock running on LC oscillation is used, this bit will be invalid and the frequency division ratio will be fixed at 1.

2. CVOUT output voltage level switch (bit 6)

This bit is used to set whether the OSD signal, mixed with the video signal output from CVOUT, is 1 Vpp or 2 Vpp.

3. Character brightness bias power ON/OFF control (bit 5)

This bit turns ON/OFF the power source for the circuit that generates the OSD signal output to the composite video signal (CVOUT pin). If set to '0', the video signal will be in the "through state." If digital output is to be used, set this bit to '0'.
For low-power consumption modes such as STOP or HALT, set this bit to '0'.

4. Power ON/OFF for halftone display buffer (bit 4)

The video signal is buffered and mixed during halftone display. This bit sets whether the buffer power is ON or OFF.
When displaying with the internal sync mode, set this bit to '0'. If it is not set, the external sync video signal (CVIN pin) will be faintly displayed on the internal sync display screen.
Similar to other OSD power ON/OFF settings, set this bit to '0' for low-power consumption modes such as STOP or HALT.

5. CVIN2 pin clamp power ON/OFF (bit 3)

This bit selects whether the video signal input to the CVIN2 pin is clamped. If the CVIN2 input video signal is synchronously separated, this bit must be set to '1'. If set to '0', CSYNC (0-VDD) may be directly input to the CVIN2 pin.
Similar to other OSD power ON/OFF settings, set this bit to '0' for low-power consumption modes such as STOP or HALT.

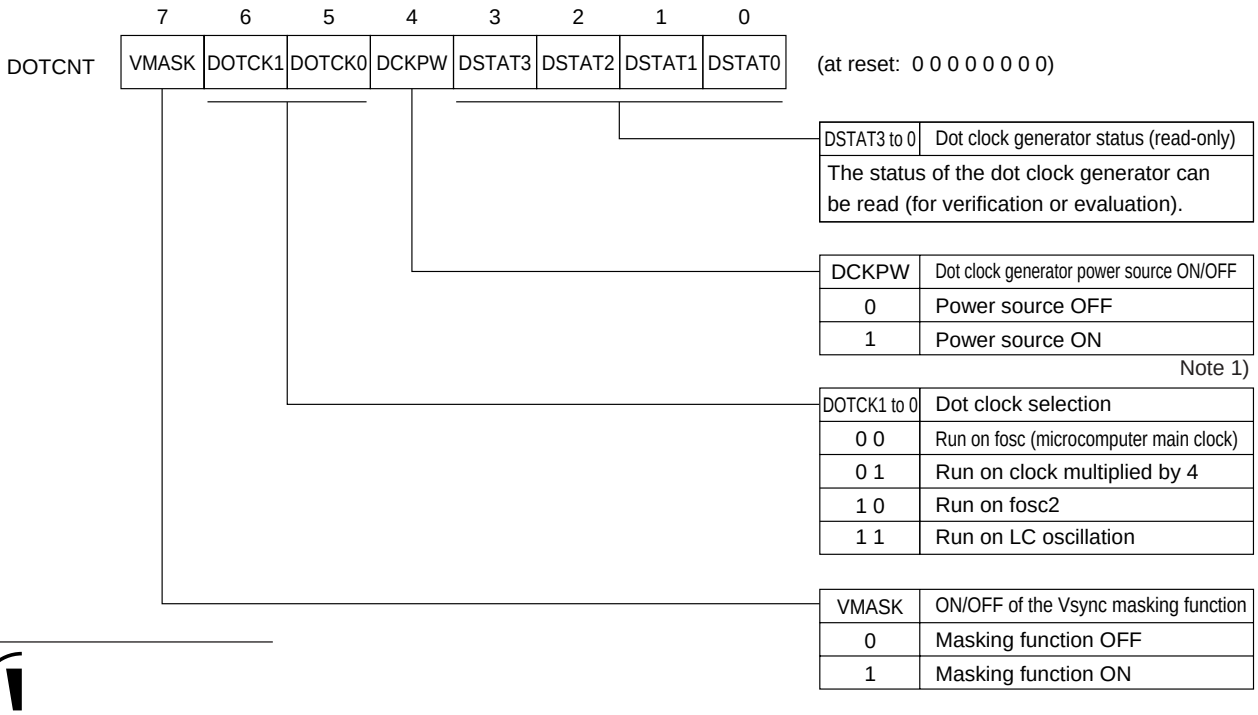
6. CVIN pin clamp power ON/OFF (bit 2)

This bit selects whether the video signal input to the CVIN pin and output from the CVOUT pin is clamped.
When the video signal output from the CVOUT pin is connected to an output driver, the bias can be set.
Similar to other OSD power ON/OFF settings, set this bit to '0' for low-power consumption modes such as STOP or HALT.

7. Clamp level switch (bits 1 and 0)

These bits select the sync clamp voltage for the video signal input to the CVIN pin and output from the CVOUT pin. Both the external sync (superimpose) and the internal sync (microcomputer video signal output) will be at the set voltage. This setting is unrelated to the CVIN2 clamp level.

11-1-3-11 DOTCNT Dot Clock Control Register



Note1: When running on the LC oscillation, set bp7 and bp6 of the PADIR register (x'3F35') to '0' (input).

Figure 11-1-14 Dot Clock Control Register (DOTCNT: x'3FD2', R/W)

The dot clock is the OSD horizontal direction clock and is controlled by the DOTCNT register. Control settings are listed below.

1. Vertical sync mask function on/off selection (bit 7)

☞ [See the OSDVsync interrupt source block diagram in Fig. 4-2-14.]

This bit selects whether or not to mask (NTSC: 85%, PAL: 72%) the VSYNC signal of the OSD section. The masking function is enabled when this bit is '1'. This function can be used to eliminate character position shift due to black blur during reproduction.

2. Dot clock selection (bits 5 and 6)

These bits specify which signal will be used to generate the dot clock. If fosc, 4× the fsci pin, or fosc2 is selected, the dot clock frequency is divided by the frequency division ratio set by FDIVS, bit 7 of CLCNT. In the case of LC oscillation, the dot clock frequency is the oscillation frequency at the pin. When using LC oscillation, bits 6 and 7 of PADIR must be set to '0'.

3. Power ON/OFF control for the dot clock phase adjuster (bit 4)

If fosc, 4× the fsci pin, or fosc2 is selected as the dot clock, the HSYN endpoint and the dot clock phase must be uniform. For this reason, there is an internal phase adjuster. This bit turns ON/OFF power to the phase adjuster. Always set this bit to '1' when selecting 4-times clock of the signals at the fosc and fsci pins or fosc2 as the dot clock during superimposed display (except during LC oscillations).
When LC oscillation is selected, settings for this bit are invalid.
Similar to other OSD power ON/OFF settings, set this bit to '0' for low-power consumption modes such as STOP or HALT.

4. Dot clock phase adjuster status (bits 3 through 0)

These bits can be used to read the status of the dot clock phase adjuster. This register is used for test purposes.

11-1-3-12 PCSTA Horizontal Sync AFC Control Register

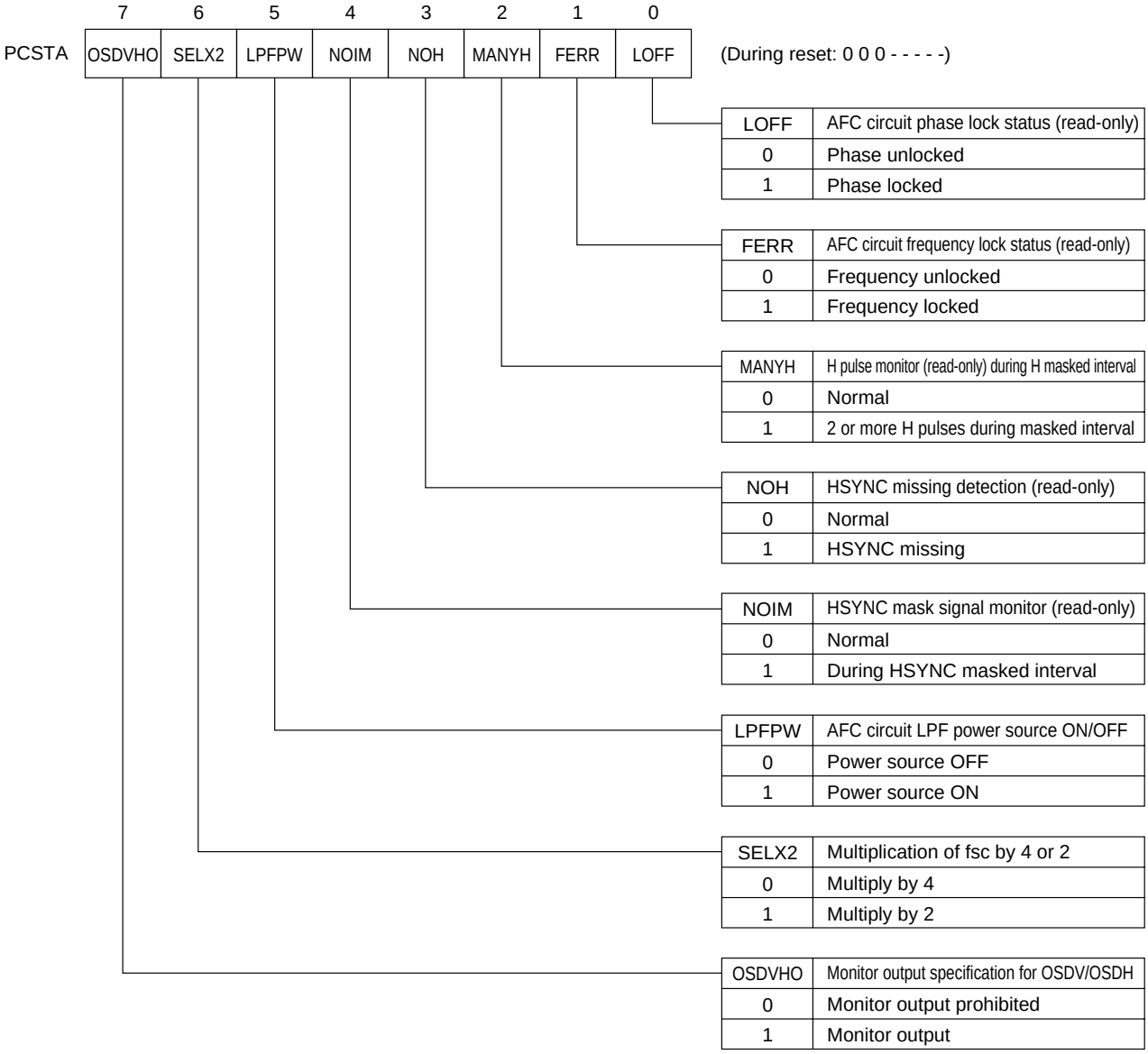


Figure 11-1-15 Horizontal Sync AFC Control Register (PCSTA: x'3FD3', R)

The PCSTA register controls the horizontal sync AFC that generates HSYNC in the external sync mode, and controls the FSCI multiplication factor of the OSD clock. Control settings are listed below.

1. OSDV, OSDH monitor output specification (bit 7)

This bit selects whether the HSYNC and VSYNC of the OSD unit will be output to bit 7 (OSDV) and bit 6 (OSDH) of port 1. If these signals are to be output, set bits 6 and 7 of the Port 1 Direction Control Register (P1DIR) to '1'. In the external sync mode, these outputs can also be used as synchronous separate output of the video signal input from the CVIN2 pin.

2. 2× or 4× multiplier for fsc generation (bit 6)

This bit sets the multiplication factor in the case where the clock input from FSCI is to be multiplied and used as the OSD unit clock (where bits 7 and 6 of OSDCNT1 are set to '01', and bits 6 and 5 of DOTCNT are set to '01').

3. Power ON/OFF for LPF in AFC circuit (bit 5)

This bit sets the power source ON or OFF for the active filter internal to the horizontal sync AFC. If the external sync mode is to be used, this bit must be set to '1'.
Similar to other OSD power ON/OFF settings, set this bit to '0' for low-power consumption modes such as STOP or HALT.

4. HSYNC mask signal monitor (bits 0 through 4)

These bits indicate the status of the phase comparator of the horizontal sync AFC circuit. Bit 4 indicates the interval for which the external horizontal sync signal is recognized. Bit 3 indicates the status of consecutive missing external horizontal sync signals. Bit 2 indicates two or more pulses during the mask interval of bit 4. If two or more pulses are counted, bit 1 indicates that the AFC frequency cannot be locked and bit 0 indicates that the phase cannot be locked.

11-1-3-13 CMPLIN OSD Interrupt Display Line Control Register

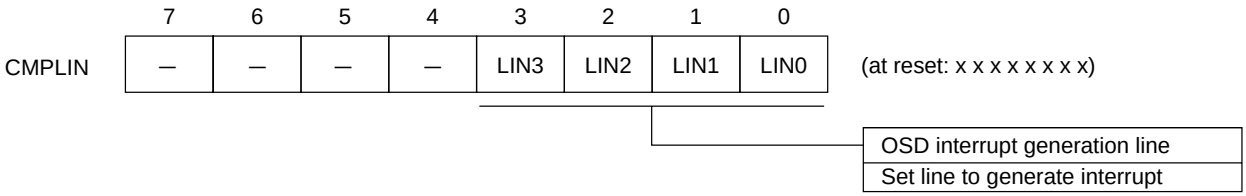


Figure 11-1-16 OSD Interrupt Display Line Control Register
(CMPLIN: x'3FDC', R/W)

The CMPLIN register specifies the OSD line that will generate an interrupt. The four bits from bits 3 through 0 specify lines 1 through 12. (Interrupts are not generated for values of these four bits of Dh or higher.) For example, to generate an interrupt after display of the 1st line is completed, set a value of '1'. The interrupt is generated at the start of the horizontal sync signal, after the specified line is displayed. If there is a line spacing setting, after completing the display and the line spacing, the interrupt is generated when the horizontal sync signal starts at the beginning of the next display.
If this register is set to the endpoint of the displayed line, an interrupt will be generated at the end of the display operation. This interrupt can be used to rewrite VRAM even faster than methods in which the vertical synchronizing signal is used to rewrite VRAM.

11-1-3-14 ROMEND CGROM Last Address Register

	7	6	5	4	3	2	1	0	
ROMED	—	—	—	—	A17	A16	A15	A14	(at reset: x x x x x x x x)

Setting value	CGROM last address
0000	Prohibited
0001	x'07FFF'
0010	x'0BFFF'
0011	x'0FFFF'
0100	x'13FFF'
0101	x'17FFF'
0110	x'1BFFF'
0111	x'1FFFF'
1000	x'23FFF'
1001	x'27FFF'
1010	x'2BFFF'
1011	x'2FFFF'
1100	x'33FFF'
1101	x'37FFF'
1110	x'3BFFF'
1111	x'3FFFF'

Figure 11-1-17 CGROM Last Address Register (ROMEND: x'3FC9', R/W)

The ROMEND register specifies the last address of the OSD character ROM (CGROM). An increase of '1' corresponds to shifting the last address by 16K. Do not set a value of '0'. Further, do not set this register during the period that VRAM is accessed during OSD display. The display may become unstable immediately after setting this register. This register should be written either when the display has stopped or during the vertical blanking period.

11-1-4 V-RAM

VRAM has two types of registers: CN2, CN1, and CN0 that control each display line,
and CCn and CDn that control each character or specify characters to be displayed. The allocation of these registers is shown in Figures 11-1-18 and 11-1-19. Be aware that the allocation changes due to bit 3 of OSDCNT (VRAMM). RAM power consumption is lower when VRAM is '0', but the number of characters used and OSD functions will change. When VRAMM is '1', a BOX shadow function is available for composite signals, and with digital output, background color and character color (contents of CCn) can be specified for each character. However, these functions are not available when VRAMM is '0'.
Shaded areas can be used as program or stack areas. Bits 2 through 0 of VLIN can set the display lines in two line units. At least that number of lines of VRAM is necessary. If bits 2 through 0 of VLIN are set to '1', the display line changes to 2. At that time, the necessary VRAM will be addresses x'0380' through x'03FF' when VRAMM is '1', and addresses x'03C0' through x'03FF' when VRAMM is '0'. The VRAMEND address is fixed at x'03FF'.

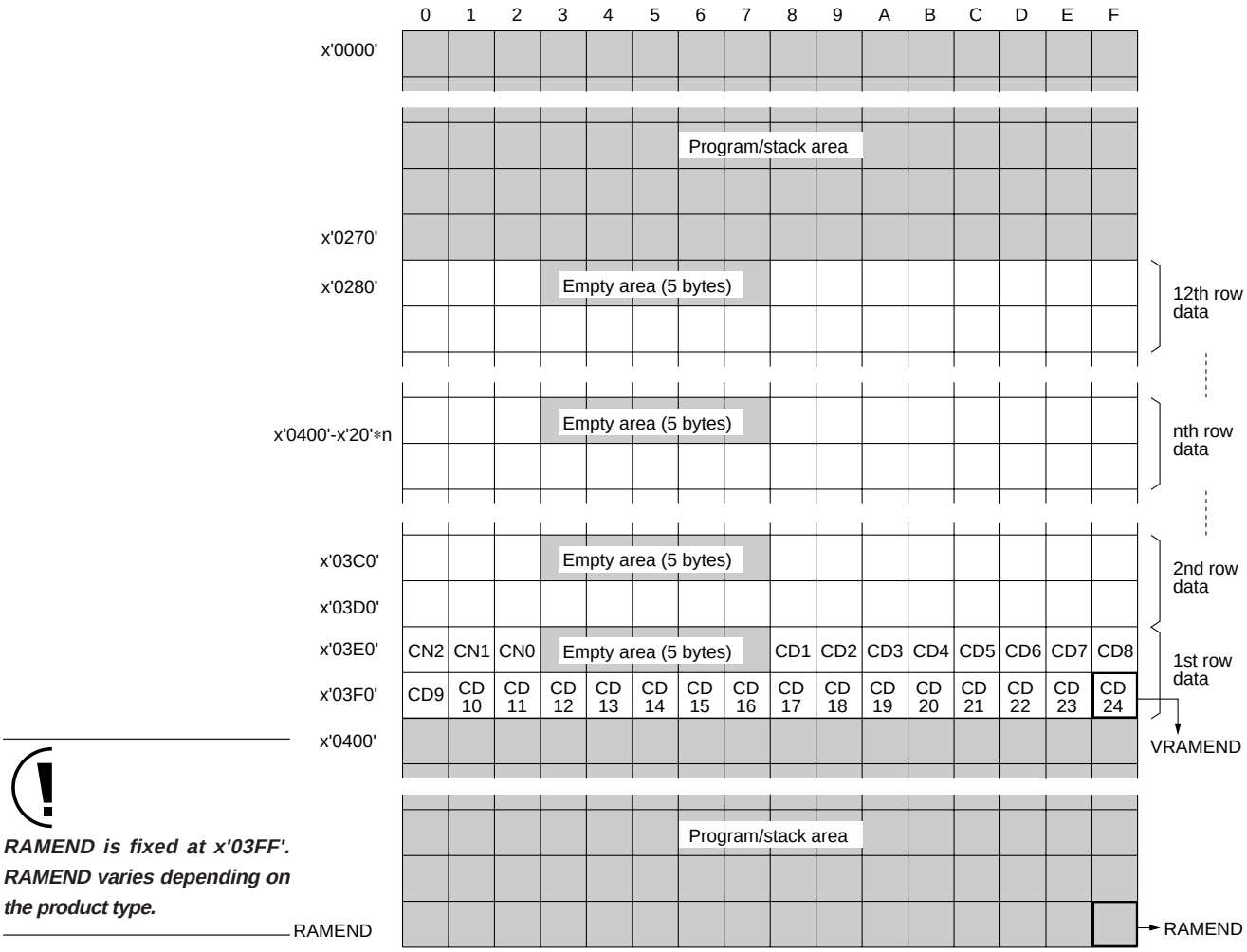


Figure 11-1-18 V-RAM Allocation (for 128 or less character types)
VRAMM Flag = '0'

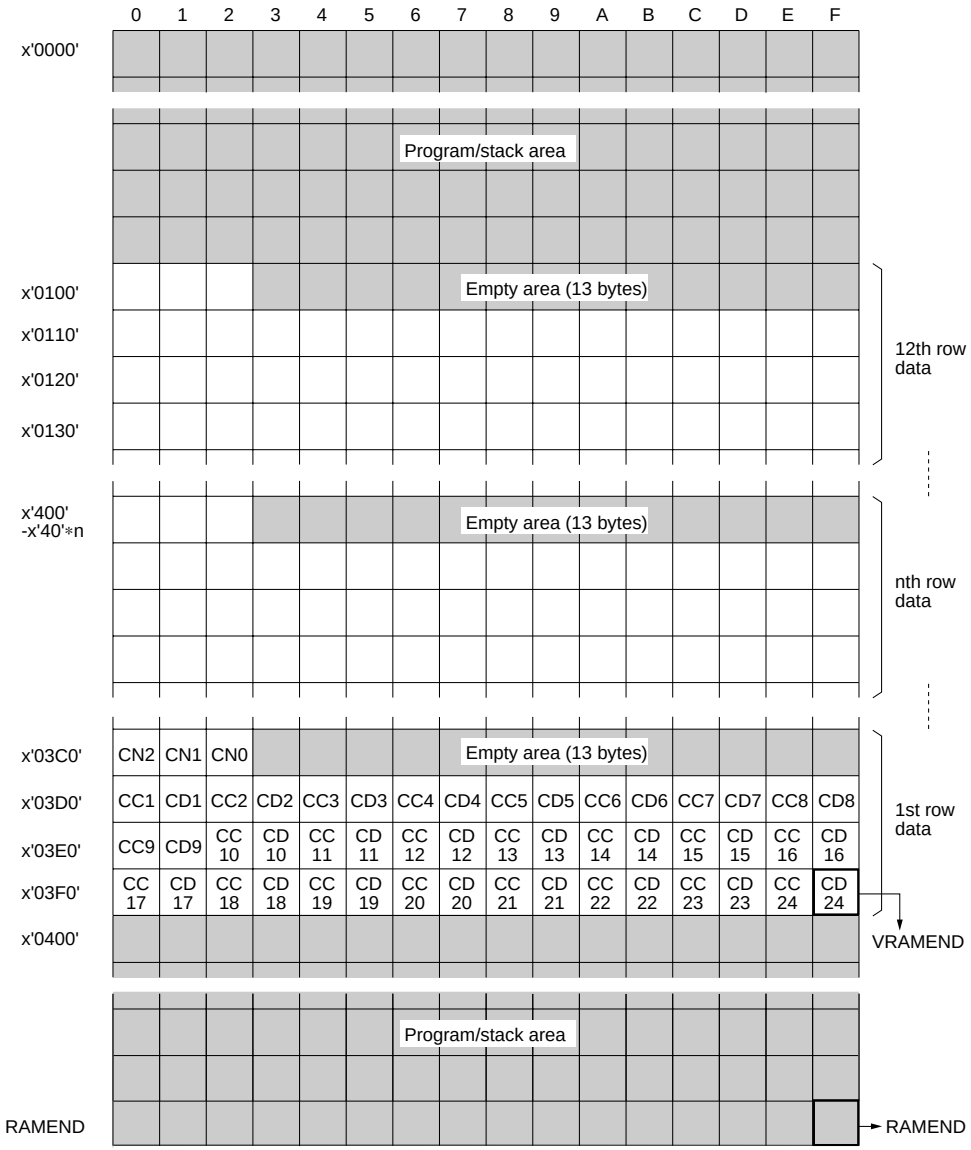


Figure 11-1-19 VRAM Allocation (when there are 129 or more character types, when there are 512 or less character types)
VRAMM Flag = '1'

RAMEND is fixed at x'03FF'. RAMEND varies depending on the product type.

11-1-4-1 Line Control VRAM CN2

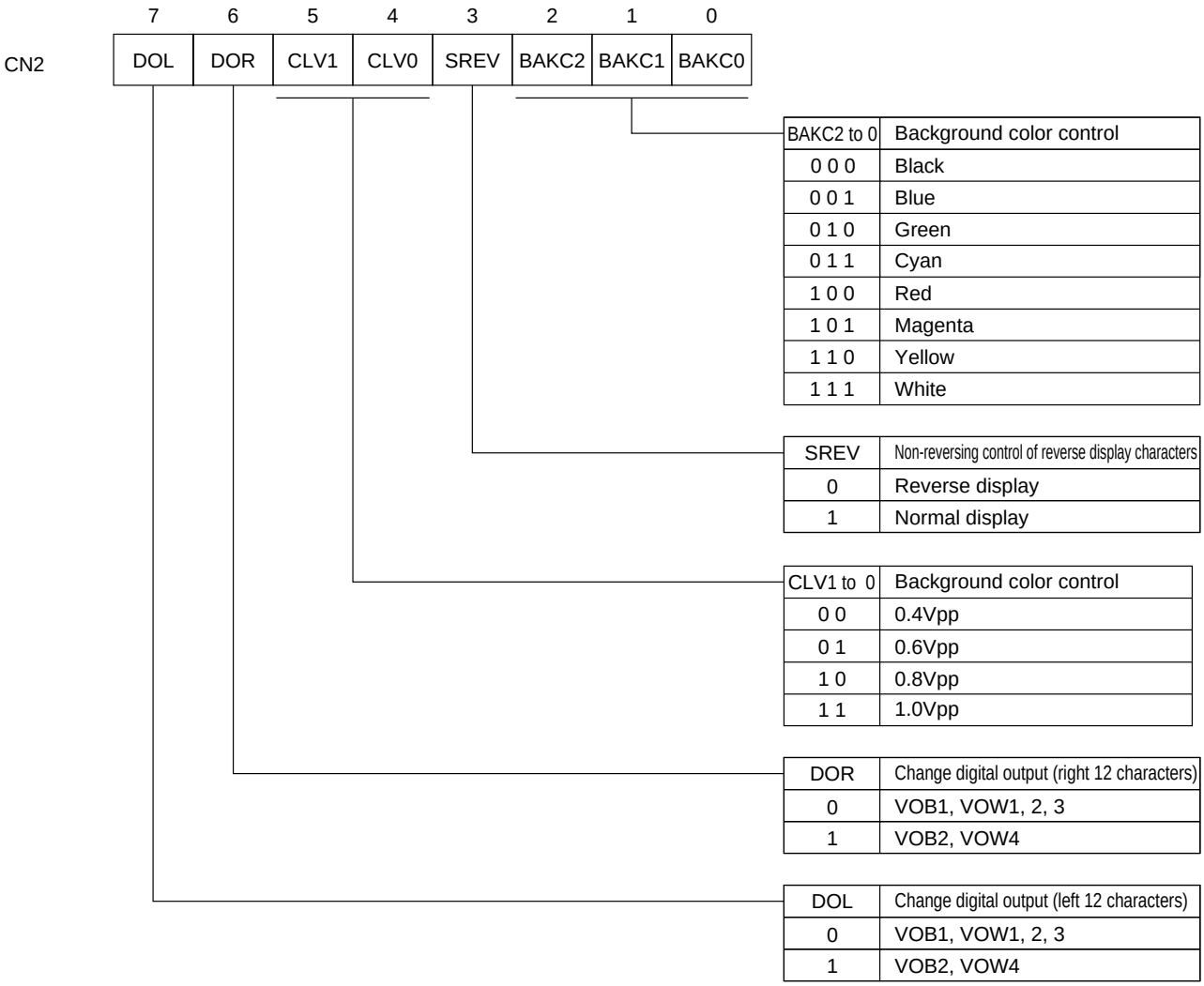


Figure 11-1-20 Line Control VRAM2 (CN2)

1. Digital output switch (left half portion) (bit 7)

This bit controls and selects the left side display of the screen partition function during digital output. Setting DGOSEL (bit 7 of VLIN) to '1', enables this control. If this bit is set to '0', VOB1, VOW1, VOW2, and VOW3 are output, VOB2 and VOW4 are fixed at '0', and OSD composite display will no longer be output. If this bit is set to '1', the result is inverted: VOB2 and VOW4 are output, VOB1, VOW1, VOW2, and VOW3 are fixed at '0', and the OSD composite display is output.

2. Digital output switch (right half portion) (bit 6)

This bit controls and selects the right side display of the screen partition function during digital output. Setting DGOSEL (bit 7 of VLIN) to '1', enables this control. If this bit is set to '0', VOB1, VOW1, VOW2, and VOW3 are output, VOB2 and VOW4 are fixed at '0', and OSD composite display will no longer be output. If this bit is set to '1', the result is inverted: VOB2 and VOW4 are output, VOB1, VOW1, VOW2, and VOW3 are fixed at '0', and the OSD composite display is output.

3. Background color level control (bits 5 and 4)

During internal sync output, these bits set the background tint of the composite signal. There is no effect on the digital output.

4. Non-highlight control for highlighted characters (bit 3)

This bit is used only for highlighted characters. If set to '1', characters specified for highlighting will no longer be highlighted, and special control of the character background will be performed. During internal sync mode, when BAKSEL (bit 6 of the VLIN register) is set to '1', the character background is normally black and white. However if this bit (bit 3) is also set to '1', characters within that line and specified for highlighting by REV (bit 0 of the CDn VRAM) will not be highlighted, but will instead be displayed as white characters with the line background color (specified by BAKC2 to BAKC0 of VRAM's CN2) as the character background. For lines where this bit is '0', characters specified for highlighting will be highlighted as usual, and characters will have the line background color.
During external sync mode, if this bit is set to '1', similar to internal sync mode, characters specified for highlighting will not be highlighted. A special use of this bit is as follows. If set to '1' and BAKSEL (bit 6 of the VLIN register) is set to '1', characters specified for highlighting will be white characters with a black background (with a brightness specified by EDGB01 and EDGB00 of VRAM's CN0).
Similarly for RGB, if this bit is set to '1', characters specified for highlighting will not be highlighted.

5. Background color control (bits 2, 1, and 0)

The background color control sets the background color. This bit is used only for the internal sync mode. During composite display, with the exception of black and white, 6 color phases (in 45 degree units) can be selected. Apart from this, the color setting is determined by the settings of the background brightness by bits 7~5 of CN1 and the chrominance by bits 5 and 4 of this register. Eight colors can be specified with digital output.
During composite display, black and white are controlled by the background brightness control of CN1 bits 7 through 5.

11-1-4-2 Line Control VRAM CN1

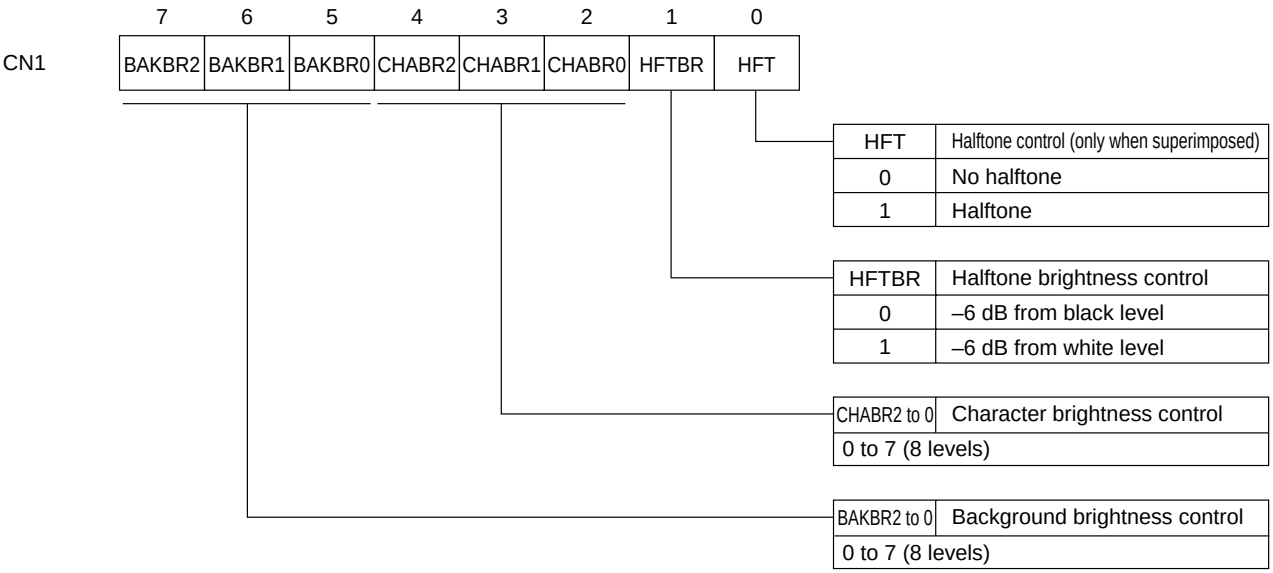


Figure 11-1-21 Line Control VRAM (CN1)

1. Background brightness control (bits 7 through 5)

These bits control the background brightness of the composite signal. Similar to the background tint control, settings are only valid for the internal sync mode.
These control bits have no effect upon the digital output.

2. Character brightness control (bits 4 through 2)

These bits control the character brightness of the composite signal. Settings are valid for both internal and external sync modes.
These control bits have no effect upon the digital output.

3. Halftone brightness control (bit 1)

During the external sync mode, there is a halftone function for the composite output. This bit selects whether the composite signal will be 1/2 (–6 dB) of a white signal, or 1/2 of the pedestal signal. If set to '1', the display will be brighter (whiter) than the original screen. If set to '0', the display will dim.
This control bit has no effect upon the digital output.

4. Halftone control (bit 0)

During the external sync mode, this bit controls ON/OFF of the halftone function for composite output.
This control bit has no effect upon the digital output.

11-1-4-3 Line Control VRAM CN0

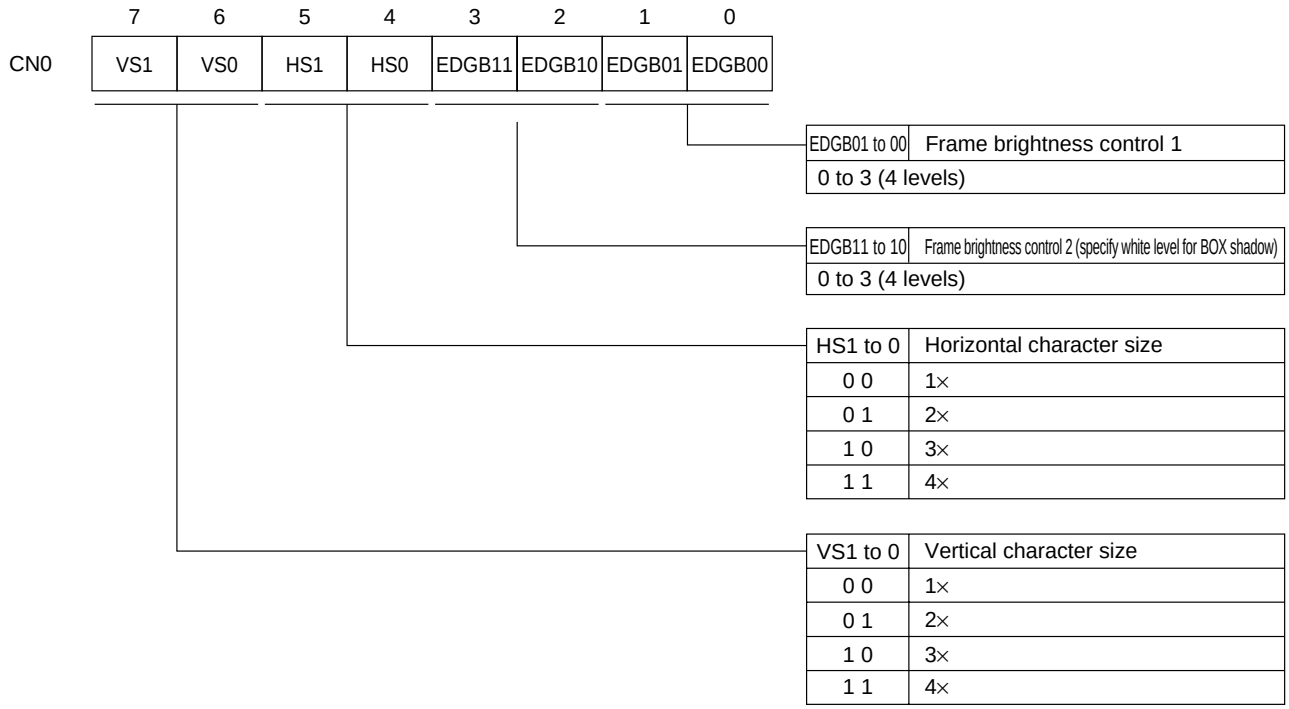


Figure 11-1-22 Line Control VRAM0 (CN0)

1. Vertical character size (bits 7 and 6)

These bits set the vertical size of characters. In addition to characters, this setting also changes the line spacing number. For example, if 10 is to be multiplied by 3, each increase of 1 in the line spacing correspond to 3 lines.

2. Horizontal character size (bits 5 and 4)

These bits set the horizontal size of characters. During external sync mode, even if there is no horizontal blanking of the OSD display when the horizontal multiplication factor is 1, there will be blanking if the horizontal multiplication factor is set to 2 or greater. Therefore, it is recommended to insert space codes into the VRAM CDn and CCn areas where the OSD is not displayed due to horizontal blanking. (Space codes must be set for areas of horizontal blanking.)

3. Framing brightness control 2 (bits 3 and 2)

These bits control the BOX shadow function for composite output. These bits specify the white level of the BOX shadow. There are four stages of settings from 0 to 3. These control bits have no effect upon the control of digital output.

4. Framing brightness control 1 (bits 1 and 0)

This setting is only valid during the composite output mode. These bits set the black level for the character framing brightness and BOX shadow. These control bits have no effect upon the control of digital output.

11-1-4-4 Character Control VRAM CCn (n=1 to 24)

Character control VRAM consists of CDn registers that specify characters and control the display of highlighted characters, and CCn registers that are valid when VRAMM (bit 3 of OSDCNT2) is set to '1'. The CCn registers control and set the background and character color for each character during digital mode, and control the BOX shadow display position during composite output.

■ Composit Output

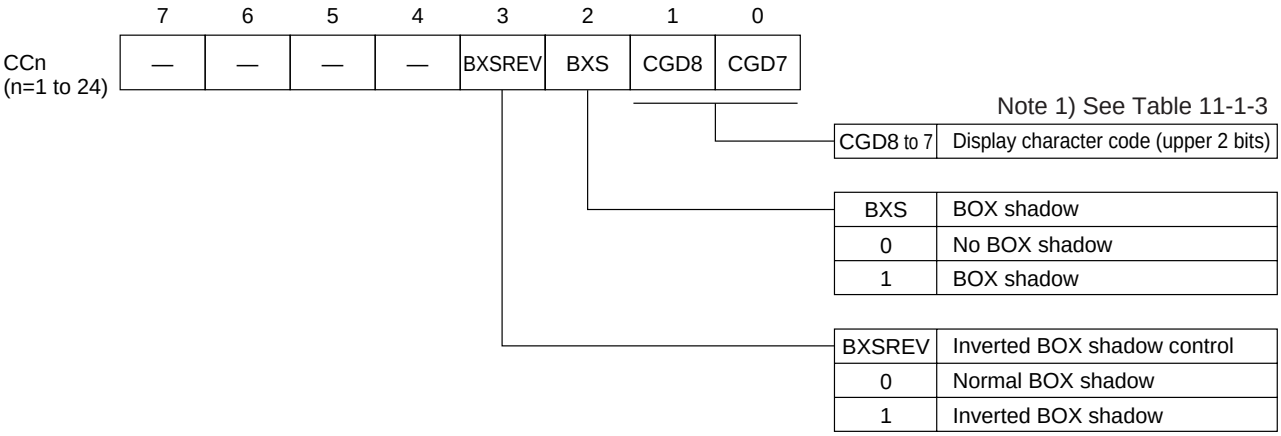


Figure 11-1-23 Character Control VRAM1 (CCn)

1. BOX Shadow Highlight Control (bit 3)

This bit selects whether the BOX shadow is displayed normally or highlighted. If '0', the BOX shadow is displayed normally. If the next BXS is set to '1' while there is a BOX shadow, and if this bit is changed from '1' to '0' or from '0' to '1', the BOX shadow immediately prior to this character position is will terminate, and the BOX shadow specified here will begin with this character.

2. BOX shadow specification (bit 2)

This bit specifies whether specified characters will have a BOX shadow. A BOX starts when this bit changes from ‘0’ to '1' and ends when this bit changes from '1' to '0'. This forms one BOX. However, if the above highlighted bits are changed or if there is a modification before completion, then two BOX shadows will be created.

3. Display character code specification (upper 2 bits) (bits 1 and 0)

In order to select 512 character types, 2 bits are added to the 7-bit CDn. These bits are added as the upper two bits. In the case of through display or space display, set these bits to '11'.

■ RGB Output

	7	6	5	4	3	2	1	0
CCn (n=1 to 24)	BCL2	BCL1	BCL0	CCL2	CCL1	CCL0	CGD8	CGD7

Note 1) See Table 11-1-3

CGD8 to 7	Display character code specification (upper 2 bits)

CCL2 to 0	Character color specification
000	Black
001	Blue
010	Green
011	Cyan
100	Red
101	Magenta
110	Yellow
111	White

BCL2 to 0	Background color specification
000	Black
001	Blue
010	Green
011	Cyan
100	Red
101	Magenta
110	Yellow
111	White

Figure 11-1-24 Character Control VRAM1 (CCn)

1. Background color specification (bits 7 through 5)

These bits set the background color of each bit for RGB output (VOW1, VOW2, and VOW3).

2. Character color specification (bits 4 through 2)

These bits set the character color of each bit for RGB output (VOW1, VOW2, and VOW3)

3. Display character code specification (upper 2 bits) (bits 1 and 0)

In order to select 512 character types, 2 bits are added to the 7-bit CDn. These bits are added as the upper two bits. In the case of through display or space display, set these bits to '11'.

11-1-4-5 Character Control VRAM CDn (n=1 to 24)

VRAM CD is RAM that specifies characters and controls the display of highlighted characters. Special character codes such as "space" or "through" can also be specified.

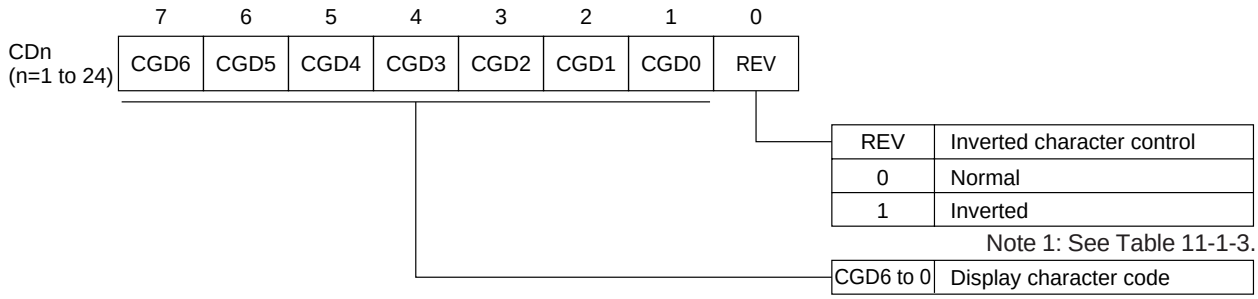


Figure 11-1-25 Character Control VRAM0 (CDn)

Table 11-1-3 VRAM Special Character Codes

Note 1: Special character code

Special character name	Function	Background data description			Display character code	
		During composite output		During digital output	128 characters or less VRAMM=0	128 characters or more VRAMM=1
		During menu display	While superimposed			
Space	Blank character	Character background color	Screen through, halftone valid	Character background color	x'7F'	x'1FF'
Through	Blank character	Character background color	Screen through, halftone invalid	Screen through	x'7E'	x'1FE'

1. Display character code specification (bits 7 through 1)

These 7 bits are used to specify 128 characters, including "space" and "through".

2. Character highlight control (bit 0)

This bit controls character highlighting. Settings can be made for each character.

11-1-5 CGROM

The CGROM (Character ROM) stores several different types of characters that consist of 12×8 dots. This ROM is allocated in the user program area. The ROMEND address can be set in 16K units by the ROMEND register. Figure 11-26 shows the allocation for 256 characters. Figure 11-27 shows the character configuration.

ROM Allocation when Using 256 Characters

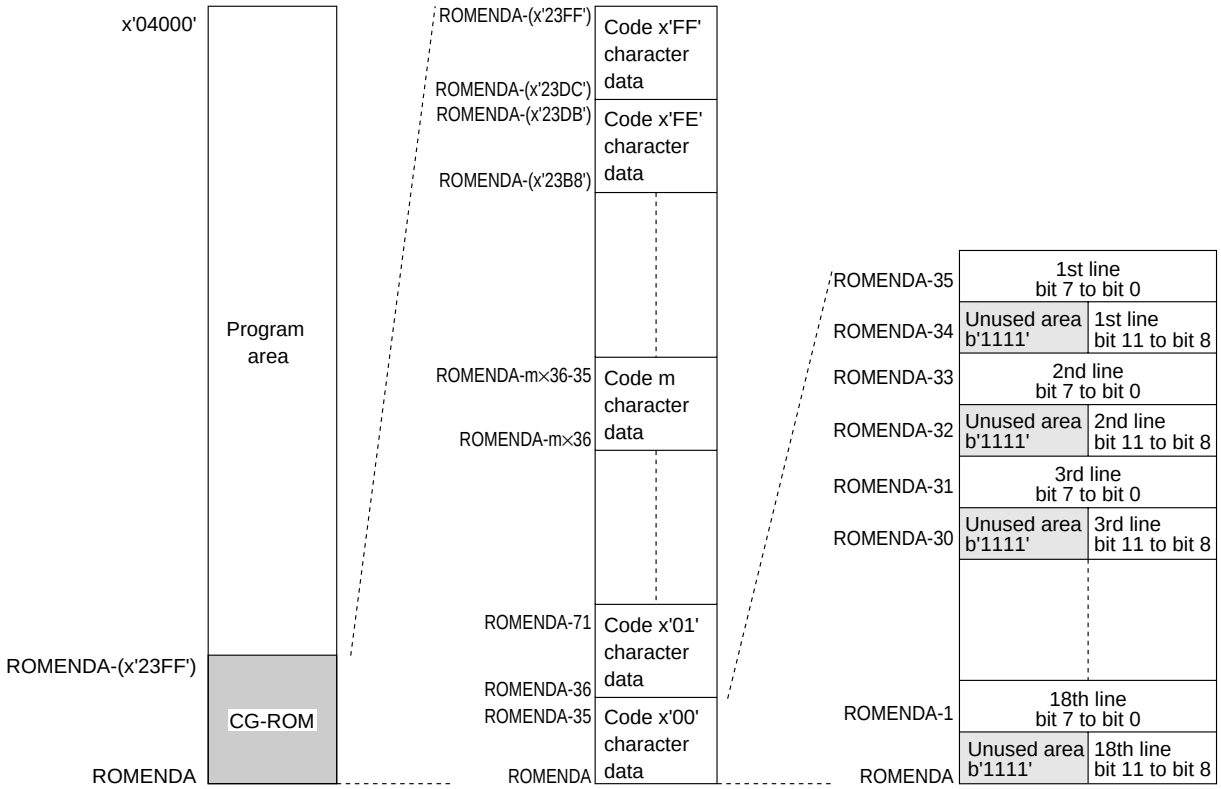


Figure 11-1-26 CGROM Allocation

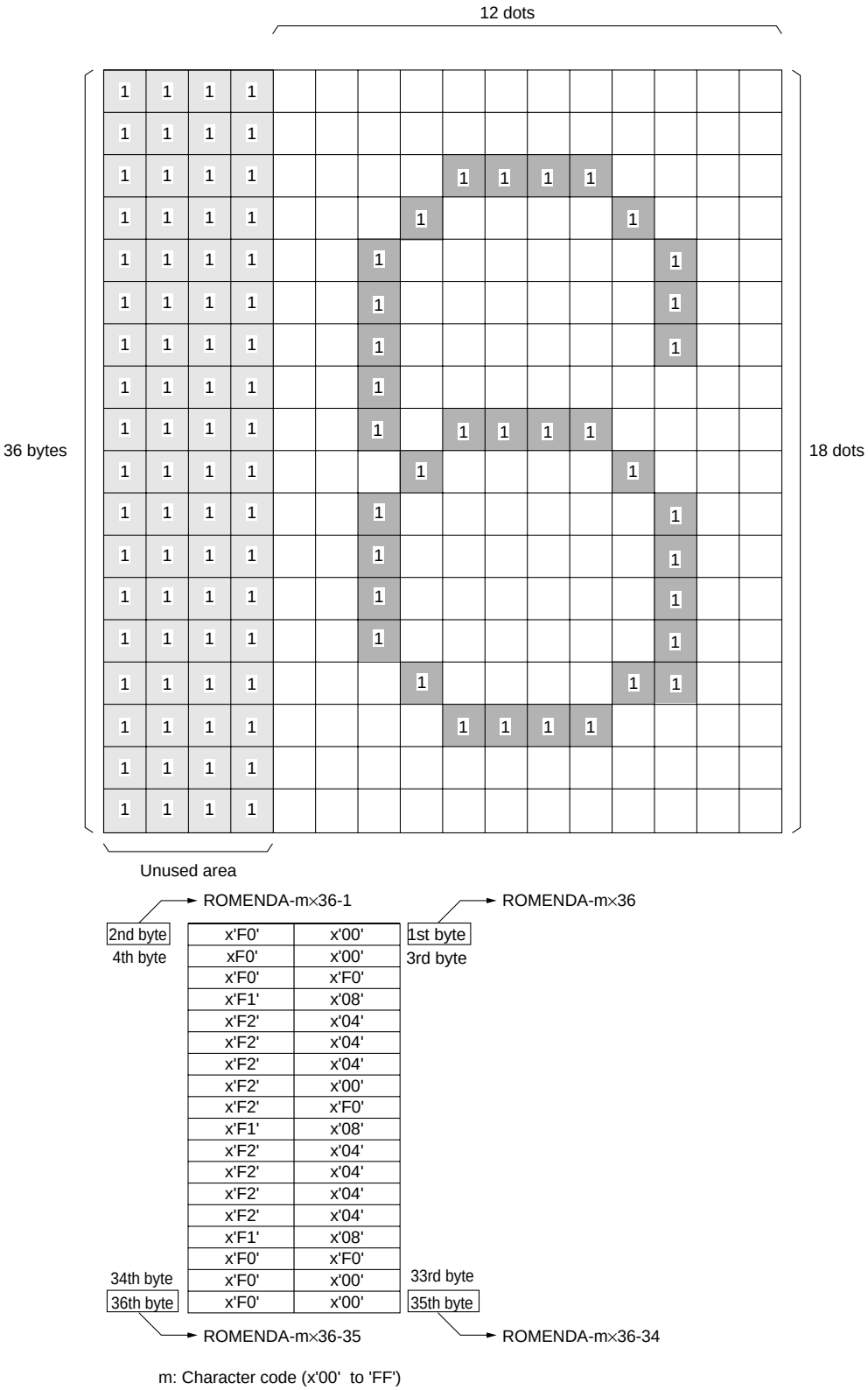
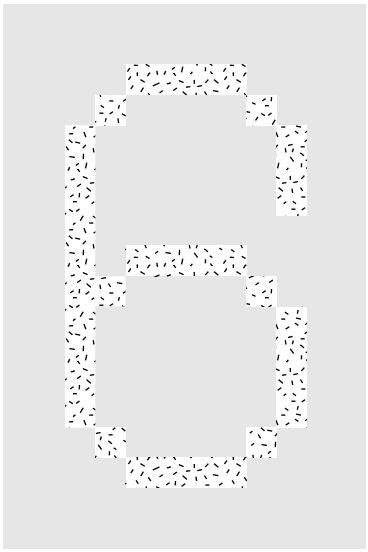


Figure 11-1-27 Character Configuration

■ Character Framing



■☞ [See Fig. 11-1-3-2 OSDCNT2.]

Figure 11-1-28 Display with No Frame

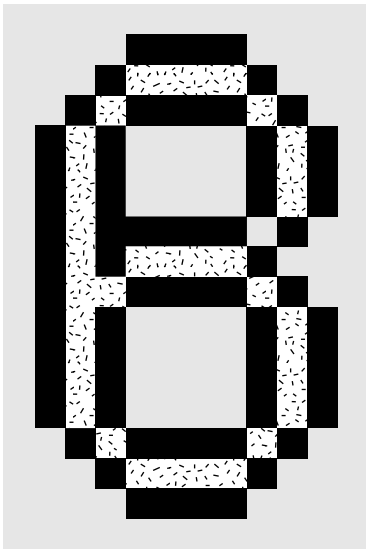


Figure 11-1-29 Display with Frame (4 directions)

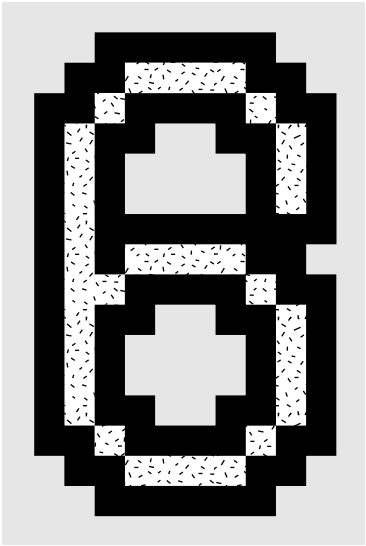


Figure 11-1-30 Display with Frame (8 directions)

The box shadow display is valid only in the case of composite signals when the VRAMM flag is '1'.

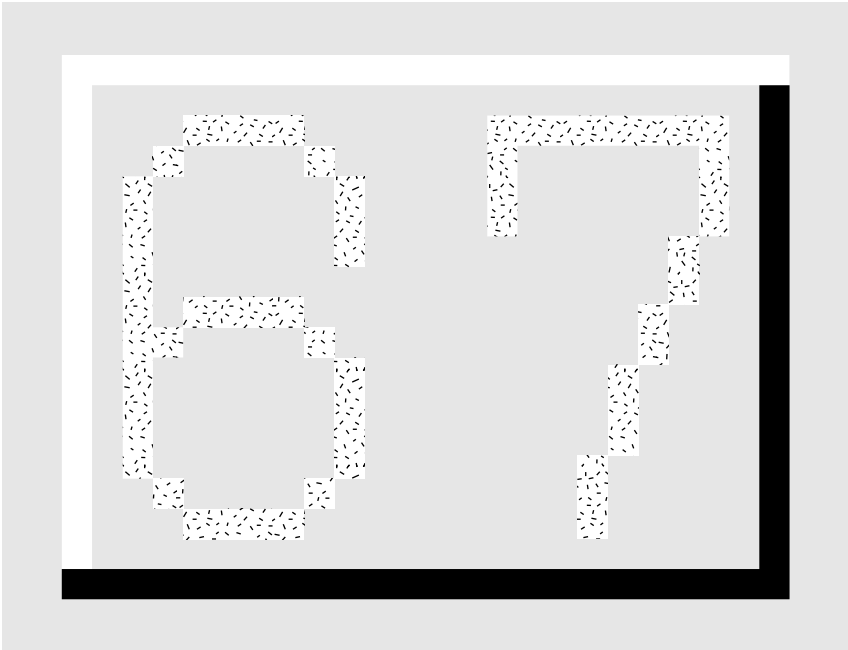


Figure 11-1-31 BOX Shadow Display

■ Various Examples of Character, Character Background, and Screen Background Settings (During Composite Output)

[1] Set entirely blue background

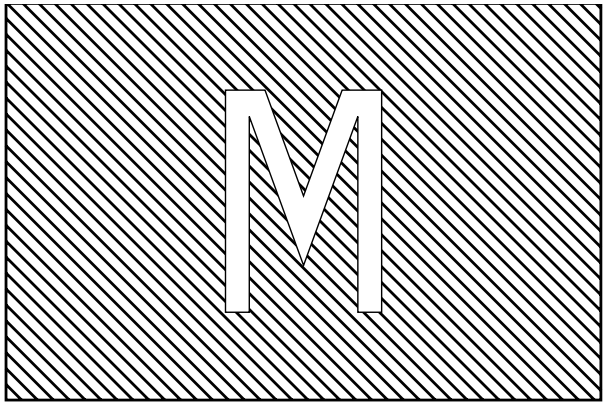


Figure 11-1-32

Register
- (OSDCNT1)SYNMOD=0
- (OSDCNT2)COMPON=1
- (VLIN)DGOSEL=0
- (VLIN)BAKSEL=0
- (VLIN)BKMONO=0

VRAM
- (CN2)BAKC2=0
- (CN2)BAKC1=0
- (CN2)BAKC0=1
- Set (CN1) background brightness control and character brightness control

[2] Set entirely black and white background

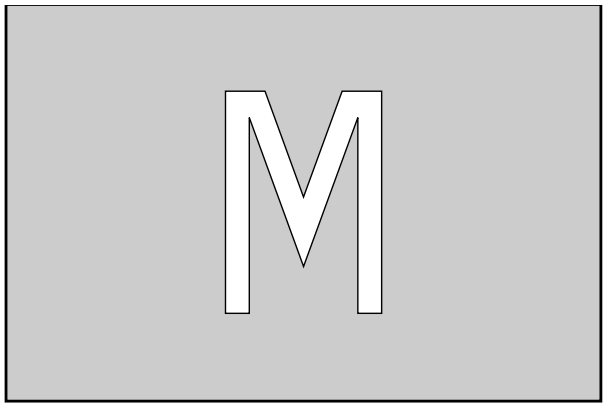


Figure 11-1-33

Register
- (OSDCNT1)SYNMOD=0
- (OSDCNT2)COMPON=1
- (VLIN)DGOSEL=0
- (VLIN)BAKSEL=1
- (VLIN)BKMONO=1
- Set black and white background brightness with (VLIN) MOBR1, MOBR0

VRAM
- (CN2)SREV=0
- (CDn)REV=0
- Set (CN1) character brightness control

[3] With an entirely black and white background, set the desired character background to color

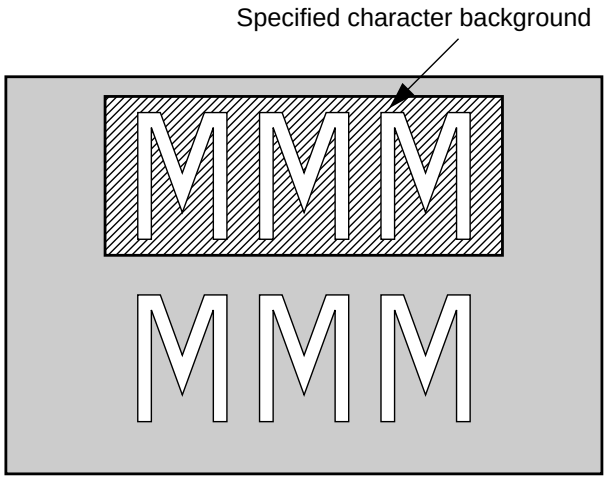


Figure 11-1-34

Register
- (OSDCNT1)SYNMOD=0
- (OSDCNT2)COMPON=1
- (VLIN)DGOSEL=0
- (VLIN)BAKSEL=1
- (VLIN)BKMONO=1
- Set black and white background brightness with (VLIN) MOBR1, MOBR0

VRAM for colored background
- (CN2)SREV=1
- (CDn)REV=1
- Set (CN2) background color control

VRAM for black and white background
- (CN2)SREV=0
- (CDn)REV=0
- Set (CN1) character brightness control

[4] With an entirely black and white background, set the desired character to color

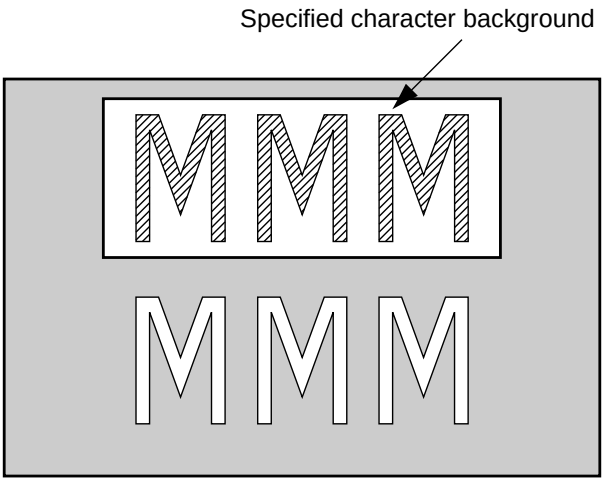


Figure 11-1-35

Register
- (OSDCNT1)SYNMOD=0
- (OSDCNT2)COMPON=1
- (VLIN)DGOSEL=0
- (VLIN)BAKSEL=1
- (VLIN)BKMONO=1
- Set location to add color to black and white background with (VLIN) MOBR1, MOBR0

VRAM for location with added color
- (CN2)SREV=0
- (CDn)REV=1
- Set (CN2) background color control

VRAM for location with black and white background
- (CN2)SREV=0
- (CDn)REV=0
- Set (CN1) character brightness control

[5] Set white character display for superimposed video

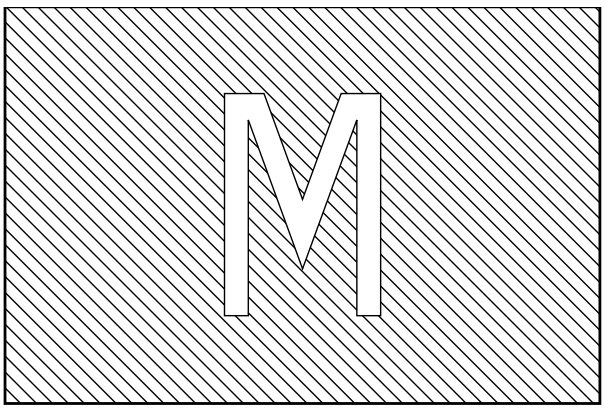


Figure 11-1-36

Register
- (OSDCNT1)SYNMOD=1
- (OSDCNT2)COMPON=1
- (VLIN)DGOSEL=0
- (VLIN)BAKSEL=0

VRAM
- (CN2)SREV=0
- (CDn)REV=0
- Set (CN1) character brightness control

[6] Set black character background for superimposed video

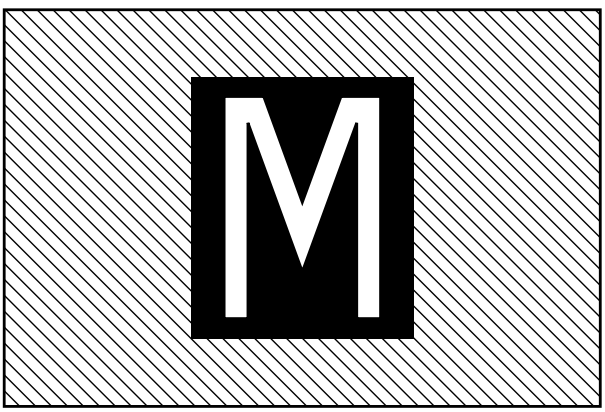


Figure 11-1-37

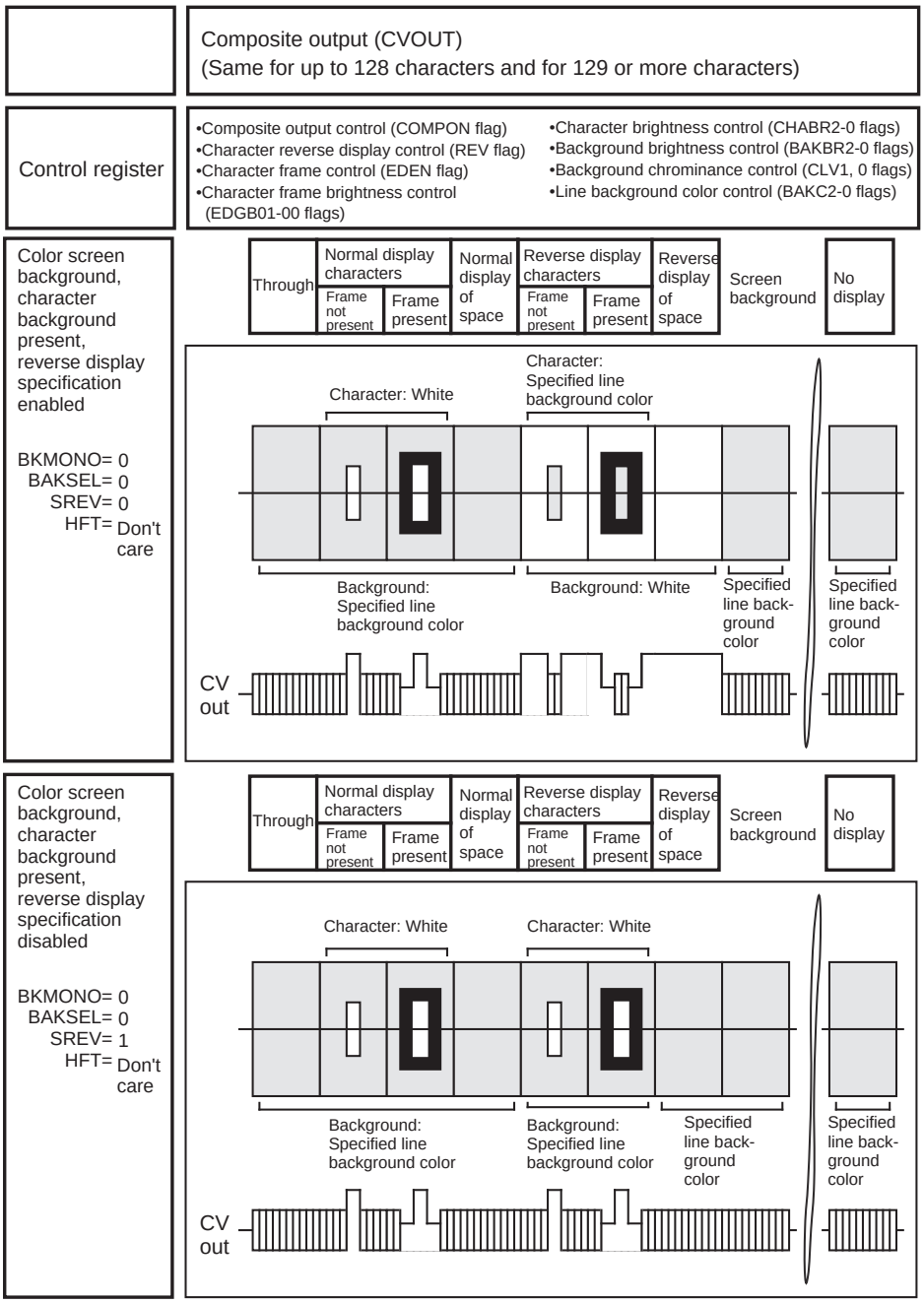
Register
- (OSDCNT1)SYNMOD=1
- (OSDCNT2)COMPON=1
- (VLIN)DGOSEL=0
- (VLIN)BAKSEL=1

VRAM
- (CN2)SREV=1
- (CDn)REV=1
- Set (CN1) character brightness control
- Set (CN0) framing brightness control 1

Internal Synchronization Mode Screen Display

SYNMOD-flag (OSDCNT1-reg. x'3FC8' bp4)='0' (Internal Synchronization Mode)

Color screen background, character background present



The following are not discussed:
Box shadow control (BOXS/BXSREV flags)
Frame mode selection (EDM0 flag)
Brightness during monochrome background (MOBR1-0 flags)

Figure 11-1-38

Internal Synchronization Mode Screen Display

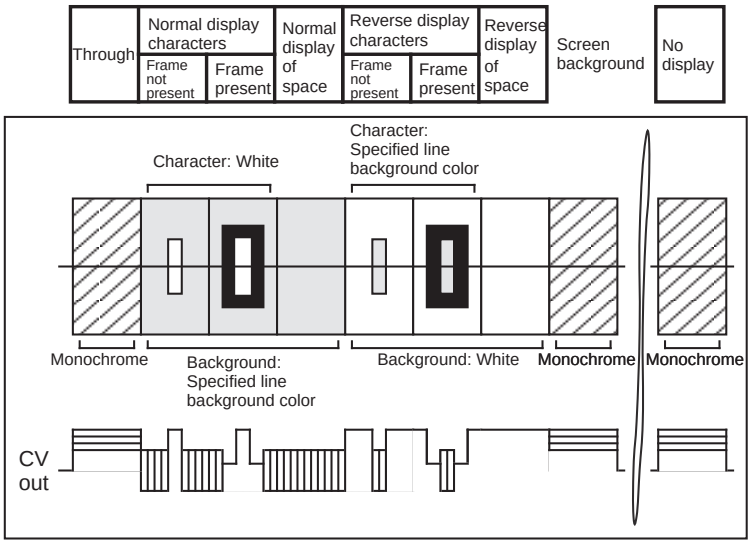
SYNMOD-flag (OSDCNT1-reg. x'3FC8' bp4)='0' (Internal Synchronization Mode)

Monochrome screen background, character background present

	Composite output (CVOUT) (Same for up to 128 characters and for 129 or more characters)
Control register	•Composite output control (COMPON flag) •Character reverse display control (REV flag) •Character frame control (EDEN flag) •Character frame brightness control (EDGB01-00 flags) •Character brightness control (CHABR2-0 flags) •Background brightness control (BAKBR2-0 flags) •Background chrominance control (CLV1, 0 flags) •Line background color control (BAKC2-0 flags)

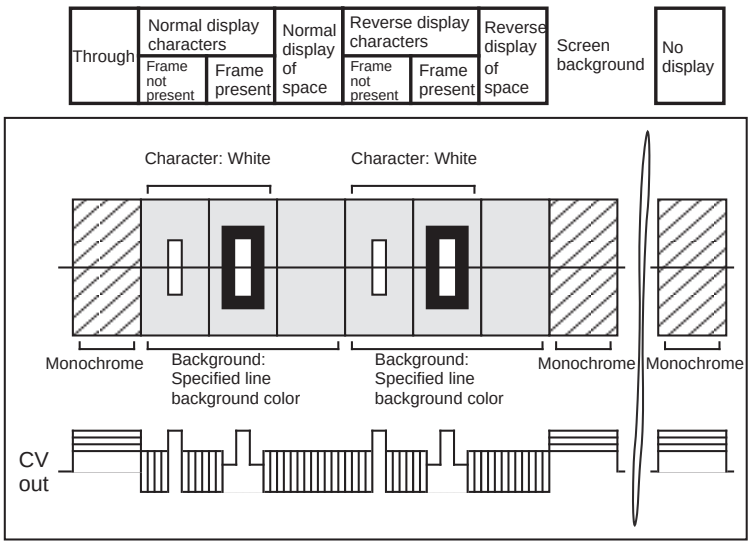
Monochrome screen background, character background present, reverse display specification enabled

BKMONO= 1
BAKSEL= 0
SREV= 0
HFT= Don't care



Monochrome screen background, character background present, reverse display specification disabled

BKMONO= 1
BAKSEL= 0
SREV= 1
HFT= Don't care



The following are not discussed:
Box shadow control (BOXS/BXSREV flags)
Frame mode selection (EDM0 flag)
Brightness during monochrome background (MOBR1-0 flags)

Figure 11-1-39

Internal Synchronization Mode Screen Display

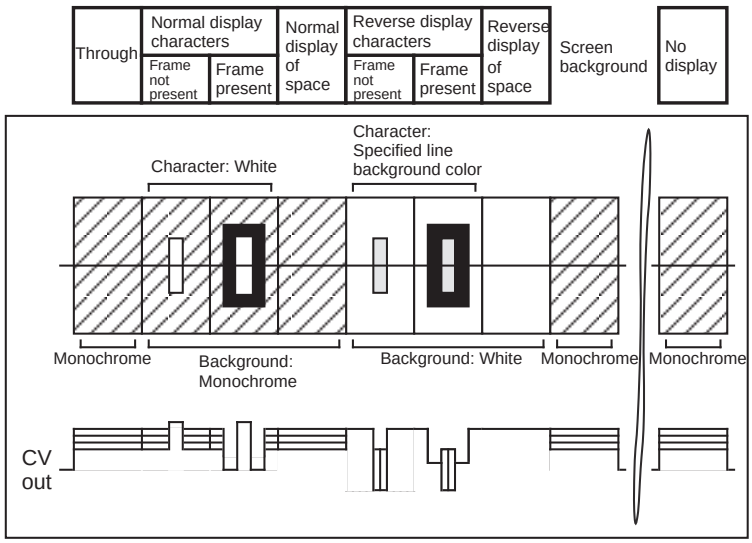
SYNMOD-flag (OSDCNT1-reg. x'3FC8' bp4)='0' (Internal Synchronization Mode)

Monochrome screen background, character background not present

	Composite output (CVOUT) (Same for up to 128 characters and for 129 or more characters)
Control register	•Composite output control (COMPON flag) •Character reverse display control (REV flag) •Character frame control (EDEN flag) •Character frame brightness control (EDGB01-00 flags) •Character brightness control (CHABR2-0 flags) •Background brightness control (BAKBR2-0 flags) •Background chrominance control (CLV1, 0 flags) •Line background color control (BAKC2-0 flags)

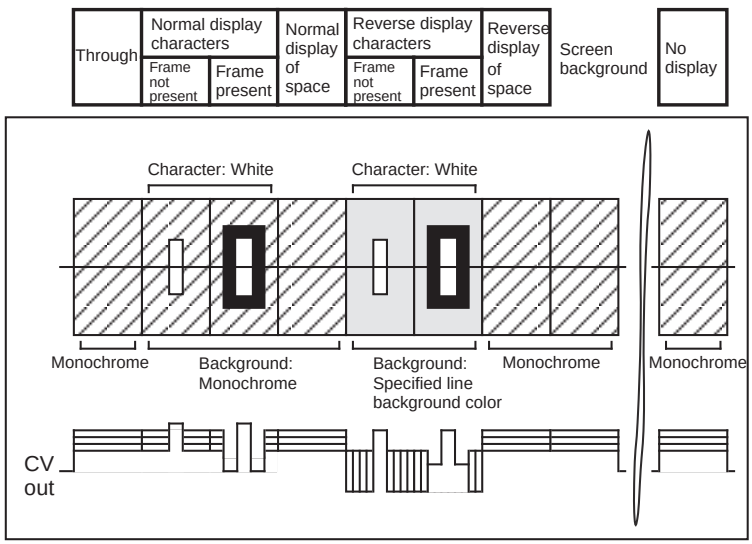
Monochrome screen background, character background not present, reverse display specification enabled

BKMONO= 1
BAKSEL= 1
SREV= 0
HFT= Don't care



Monochrome screen background, character background not present, reverse display specification disabled

BKMONO= 1
BAKSEL= 1
SREV= 1
HFT= Don't care



The following are not discussed:
Box shadow control (BOXS/BXSREV flags)
Frame mode selection (EDM0 flag)
Brightness during monochrome background (MOBR1-0 flags)

Figure 11-1-40

External Synchronization Mode Screen Display

SYNMOD-flag (OSDCNT1-reg. x'3FC8' bp4)='0' (External Synchronization Mode)

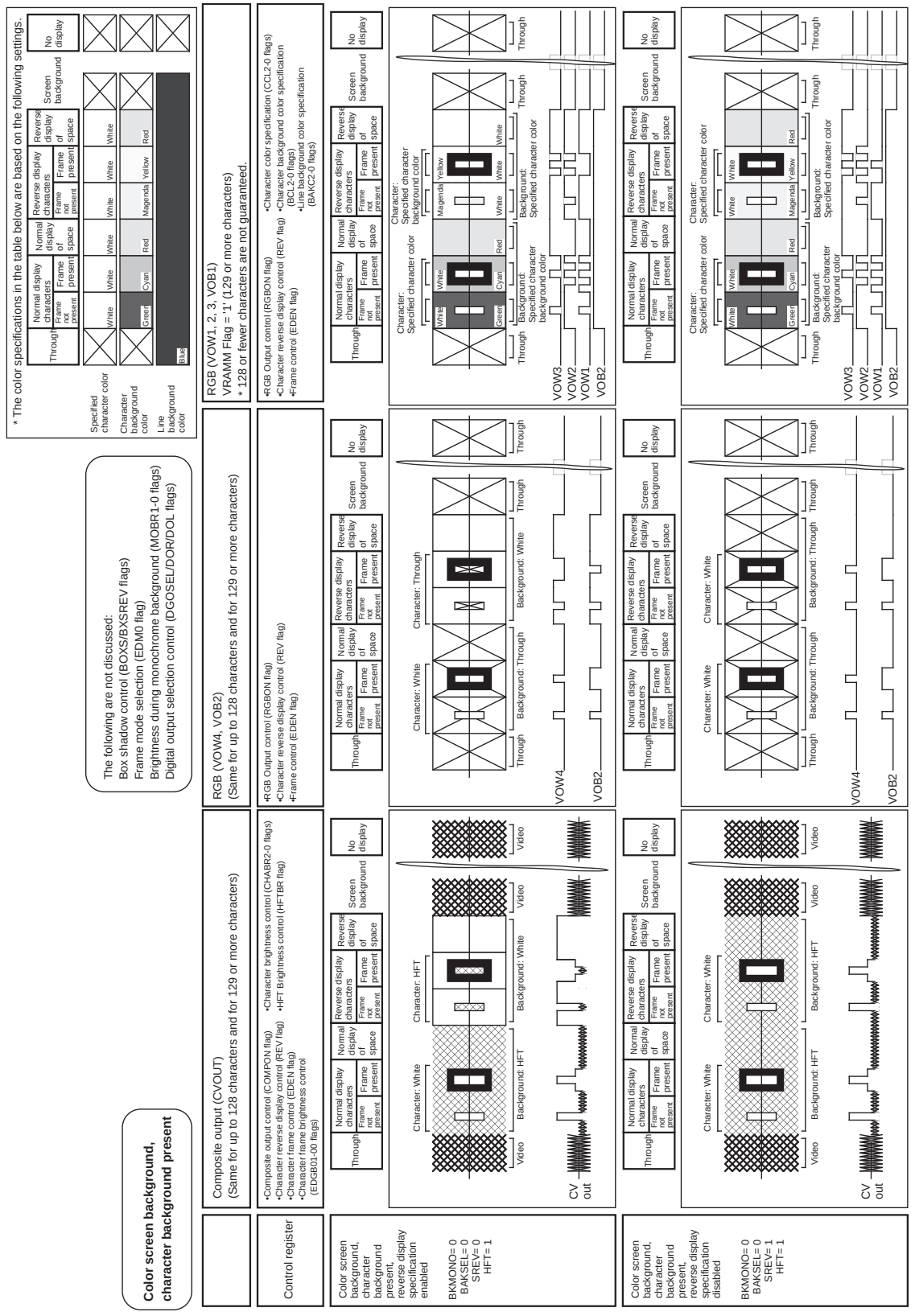


Figure 11-1-41

External Synchronization Mode Screen Display

SYNMOD-flag (OSDCNT1-reg. x'3FC8' bp4)='0' (External Synchronization Mode)

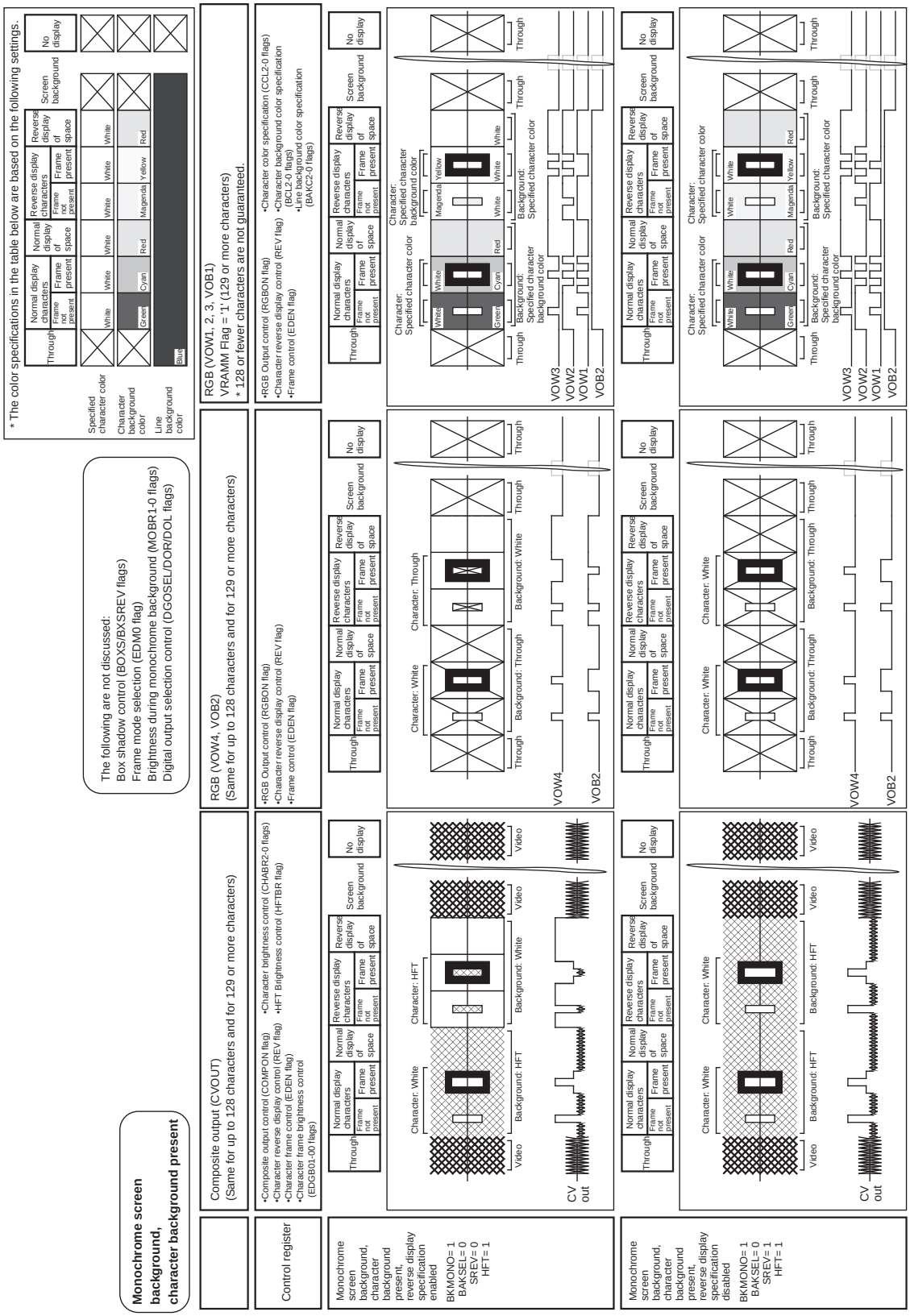


Figure 11-1-42

External Synchronization Mode Screen Display

SYNMOD-flag (OSDCNT1-reg. x'3FC8' bp4)='0' (External Synchronization Mode)

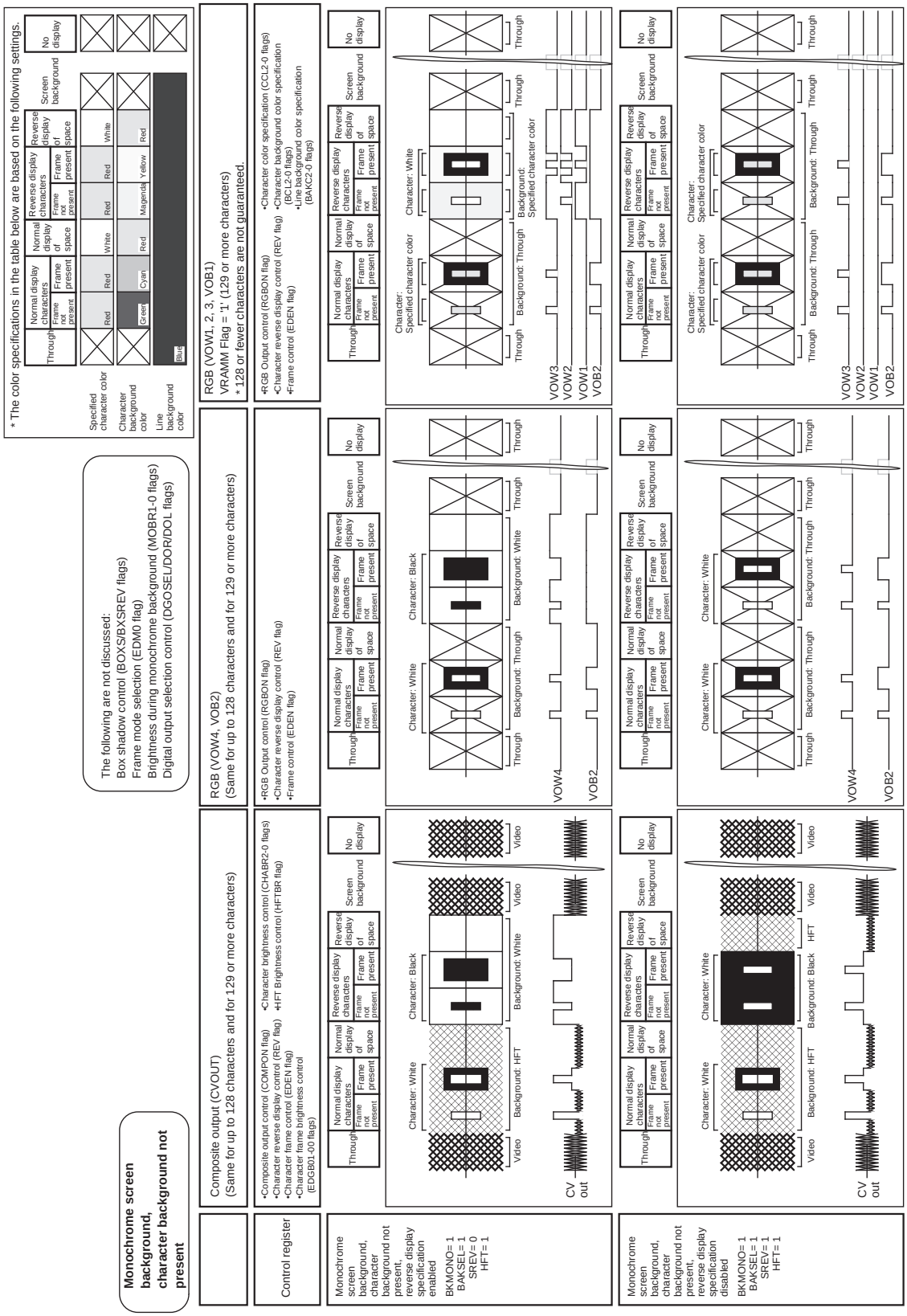


Figure 11-1-43

11-1-6 OSD Peripheral Circuits

11-1-6-1 CVIN2 Input Filter Setting

A sample filter circuit is shown in Fig. 11-1-44. High-frequency components such as the chroma component, etc., are removed from the input signal by the RC filter. Further, it is possible to set the sync slicing position by the value of R in this filter. The value is 13.5×R×6.7μA(V) as shown in Fig. 11-1-45. If the high-frequency noise is not removed sufficiently by an RC filter, a third order filter is used as shown in Fig. 11-1-46 (when the value of fc in the figure is 1MHz). In this case, the slicing position is 13.5×(R1+R2)×6.7μA(V). Although the filter characteristics vary depending on the peripheral circuits, fc (the -3dB point) is set near 1MHz. Do not set fc to too small a value since the delay due to this filter affects the data sampling of XDS.

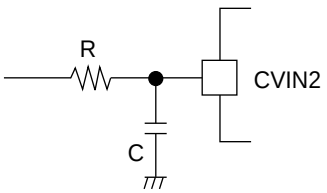


Figure 11-1-44

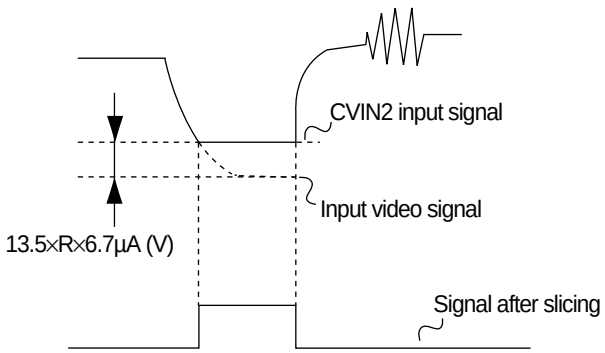


Figure 11-1-45

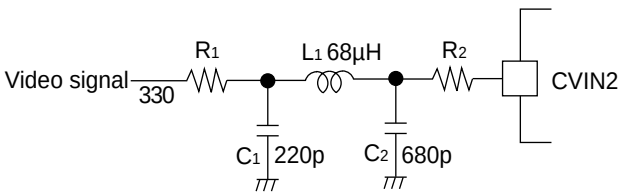


Figure 11-1-46

11-1-6-2 AFC Filter Setting

The filter circuit configuration is shown in Fig. 11-1-47. The loop gain can be adjusted by the setting of R. The response becomes faster when R is increased, thereby reducing the jitter during stable conditions. However, when jitter occurs as in received signals, too fast a response can cause the display characters to become ragged. Therefore, the values of R=2.2k and C=1µF are recommended.

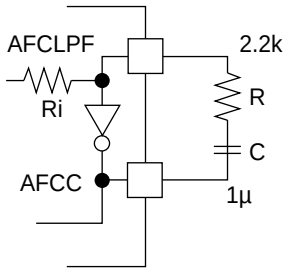
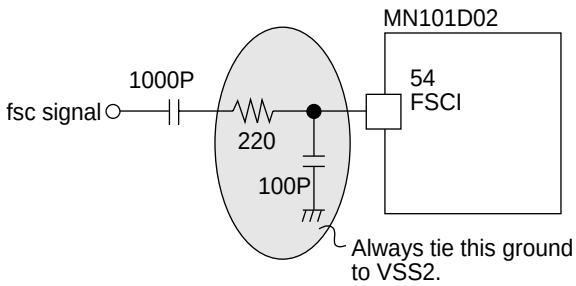


Figure 11-1-47

11-1-6-3 Precautions on the FSCI Pin Input

A hysteresis of 20mV has been set for the FSCI input to comply with small amplitude inputs. Therefore, faulty operation is made if the potential difference between the microcomputer ground (VSS2) and the video section ground is 20 mV. Although it depends on the level of noise between the two grounds, it is recommended to insert an RC filter before the input in order to prevent faulty operation. In addition, the recommended lag-lead filter constants of the frequency quadrupling circuit in this case are also indicated in the figure.

Example of FSCI input RC low pass filter settings



RC Filter setting:
Select the values of R and C so that the noise is attenuated as much as possible by the filter and the input at pin 54 becomes more than 100mVpp (about 300mVpp).

Figure 11-1-48

Example of frequency quadrupling lag-lead filter

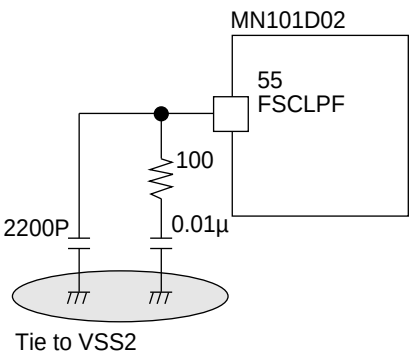


Figure 11-1-49

11-2 XDS Data Slice Function

This function is used exclusively for signal extraction of U.S. closed caption signals. Caption signals inserted in the vertical blanking interval of the video signal are extracted.

(1) Two horizontal line locations can be specified for data insertion. (Locations are set in XDSLIN1 and XDSLIN2.)
(2) The sampling clock phase can be modified. (XDSCKPH setting)
This function allows delay compensation (of up to 1260 ns) to be set when the low-pass filtering for CVIN2 synchronous separation has a large time constant, and an optimum sampling position to be set in case the waveform is distorted.
(3) Data slice levels are automatically detected during the caption data clock cycle.
(4) An XDS interrupt is generated immediately after the last data of the horizontal line specified by XDSLIN2 is stored. Data capture can be detected by this interrupt or by monitoring XDSIR.
(5) Fields can be evaluated by the synchronous separation function. [■☞ 11-4, "Synchronous Separation Function"] Evaluation results can be obtained from FLAME (bit 2 of SEPCNT (x'3FDF')).
(6) Data is captured at the VSYNC period set by XDSLIN. The program must process this data to determine whether it is valid or invalid. Therefore, the start bit data is captured by XDSDAT3.
(7) With the horizontal sync AFC locked, data is sampled at 32× the horizontal sync frequency. Therefore, it is necessary to lock the horizontal sync AFC. (OSD register setting)
(8) The data can be read out irrespective of the CVIN input clamping ON/OFF setting. However, the L level of data needs to be 1.2V or higher.
(9) The settings of XDSLIN and XDSCKPH can be verified by the function of outputting the sampling clock at P16 and the data at P17 by setting both XDSCKD (bit 6 of the special register P1SEL) and OSDVH0 (bit 7 of PCSTA) to '1'.

Register Setting Summary

Register Name	Address	R/W	Function
XDSLIN1	x'3FC6	R/W	Data extraction line 1 setting
XDSLIN2	x'3FC7'	R/W	Data extraction line 2 setting
XDSCKPH	x'3FC5'	R/W	Sampling clock phase adjustment
XDSDAT1L	x'3FC0'	R	Lower 8 bits of data specified by line 1
XDSDAT1H	x'3FC1'	R	Upper 8 bits of data specified by line 1
XDSDAT2L	x'3FC2'	R	Lower 8 bits of data specified by line 2
XDSDAT2H	x'3FC3'	R	Upper 8 bits of data specified by line 2
XDSDAT3	x'3FC4'	R	Start bit data for line 1 and line 2
XDSICR	x'3FF8'	R/W	XDS interrupt control register
FLAME (SEPCNT)2	x'3FDF'	R	Field evaluation
XDSCKD (PISEL)bp6	x'3F38'	R/W	XDS Data and clock monitoring output control
OSDVH0 (PCSTA)bp7	x'3FD3'	R/W	XDS Data and clock monitoring output control

11-2-1 XDSLIN1, XDSLIN2 Settings

Two lines can be specified by the location of closed caption data insertion. (0 to 31H)

XDS data insertion horizontal line position 1: XDSLIN1 (*x*'3FC6') R/W (5 bits)

7	6	5	4	3	2	1	0	
—	—	—	XDSL 14	XDSL 13	XDSL 12	XDSL 11	XDSL 10	
0	0	x	0	0	0	0	0	reset

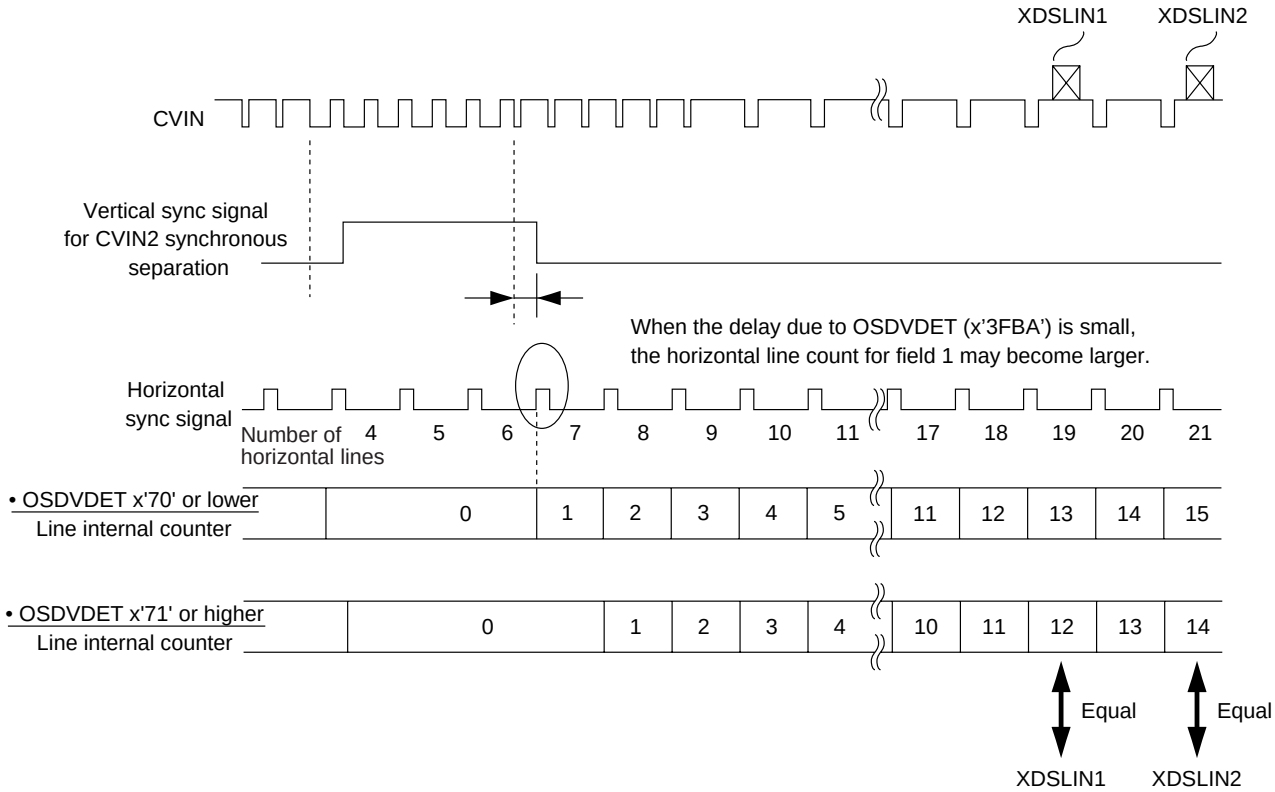
XDS data insertion horizontal line position 2: XDSLIN2 (*x*'3FC7') R/W (5 bits)

7	6	5	4	3	2	1	0	
—	—	—	XDSL 24	XDSL 23	XDSL 22	XDSL 21	XDSL 20	
x	x	x	1	0	1	0	1	reset

Specify a line that is larger than the XDSLIN1 setting value.

	Field 1	Field 2
OSDVDET <x'70'	XDSLIN1: (Initial caption insertion line) - 7 XDSLIN2: (Next caption insertion line) - 7	XDSLIN1: (Initial caption insertion line) - 6 XDSLIN2: (Next caption insertion line) - 6
OSDVDET ≥x'71'	XDSLIN1: (Initial caption insertion line) - 7 XDSLIN2: (Next caption insertion line) - 7	

See figure 11-2-1 and section 11-4, “Sync Separation” for details on changing the XDSLIN1 and XDSLIN2 settings using vertical sync separation.
Always set bits 7 and 6 of XDSLIN1 to '0'.



If the horizontal sync signal is set in the vicinity of transition points of the vertical sync signal, the line to be read may shift or be unreadable.
XDSIR will be set immediately after the registers are set. After setting XDSLIN, clear XDSIR.

When manipulating the interrupt request flag (xxxIR), it is necessary to set beforehand the IR write enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, reset IRWE to '0' at the end of such manipulations.

11-2-2 Sampling Clock Phase Adjustment

To reduce incorrect operation caused by noise, the sampling clock for caption data is generated from the horizontal sync signal obtained with horizontal AFC, multiplied by 32. The signal input from CVIN is detected at the clock cycle for closed captioned data. Slice levels are determined from maximum and minimum values. Data is compared to those values and sliced.
Therefore, if a low-pass filter with a large time constant is inserted into the path of a signal input to CVIN2, a shift will occur between the clock and data. Distortion of the video input signal waveform will prevent data from being sampled at its center position and the percentage of data read with a shifted sampling position will increase. This register is used to adjust these values.
The position at which sampling is being done can be verified by setting to '1' both XDSCKD (bit 6 of register P1SEL (x'3F38')) and bit 7 of PCSTA (x'3FD3'). At this time, P16 outputs the sampling clock and P17 outputs the video signal converted into a data of 0's and 1's.

XDS sampling phase: XDSCKPH (x'3FC5') (R/W, 5 bits)

7	6	5	4	3	2	1	0	
—	—	—	XDSCP 4	XDSCP 3	XDSCP 2	XDSCP 1	XDSCP 0	
x	x	x	1	0	1	0	1	reset

XDSCKPH = {(set amount of shift from data center)–(delay of LPF of CVIN2)}×(OSD operation clock frequency selected by OSDCNT1)+4

Relation between register setting and sampling clock (When the delay is 0 of the LPF connected to the CVIN2 input.)

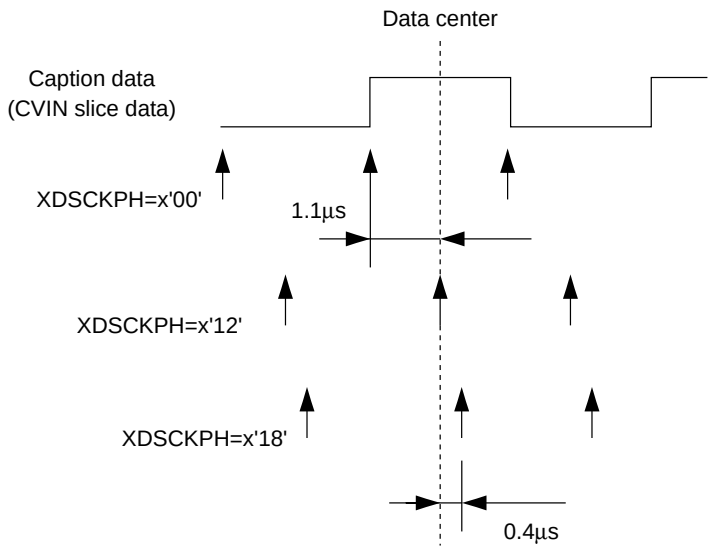


Figure 11-2-2

If XDSCKPH is set to a value larger than x'18', the value read will be unstable.

11-2-3 Data Slice Levels

Maximum and minimum data values on the horizontal line specified by XDSLIN1 and XDSLIN2, are detected at the clock period of the caption data. The slice level is 1/2 of those levels.

Slice Level Detection Operation

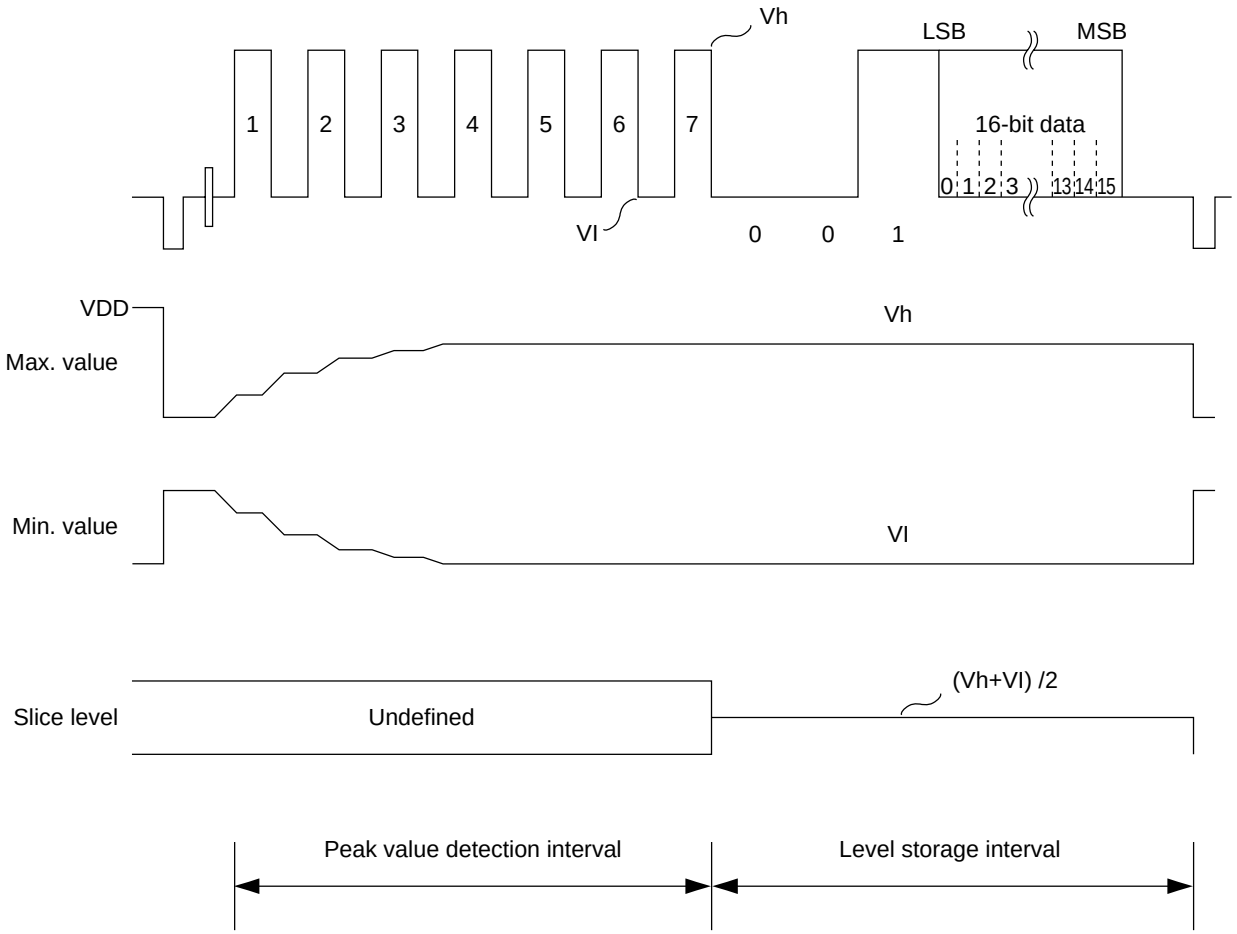


Figure 11-2-3

11-2-4 XDS Interrupt Control

When manipulating the interrupt request flag (xxxIR), it is necessary to set beforehand the IR write enable flag (IRWE) of the memory control register (MEMCTR) to '1'. However, reset IRWE to '0' at the end of such manipulations.

Enable XDS interrupts in the following sequence.

(1) Set XDSLIN1 and XDSLIN2
(2) Set XDSCKPH
(3) Specify interrupt priority level with XDSLV1 and XDSLV0 (bits 1 and 0) of XDSICR, and clear XDSIR (bit 0)
(4) Set XDSIR (bit 1) of XDSICR

XDS interrupt control: XDSIR (x'3FF8') (R/W, 4 bits)

7	6	5	4	3	2	1	0	
XDSLV1	XDSLV0	—	—	—	—	XDSIE	XDSIR	(at reset: 0 0 - - - - 0 0)

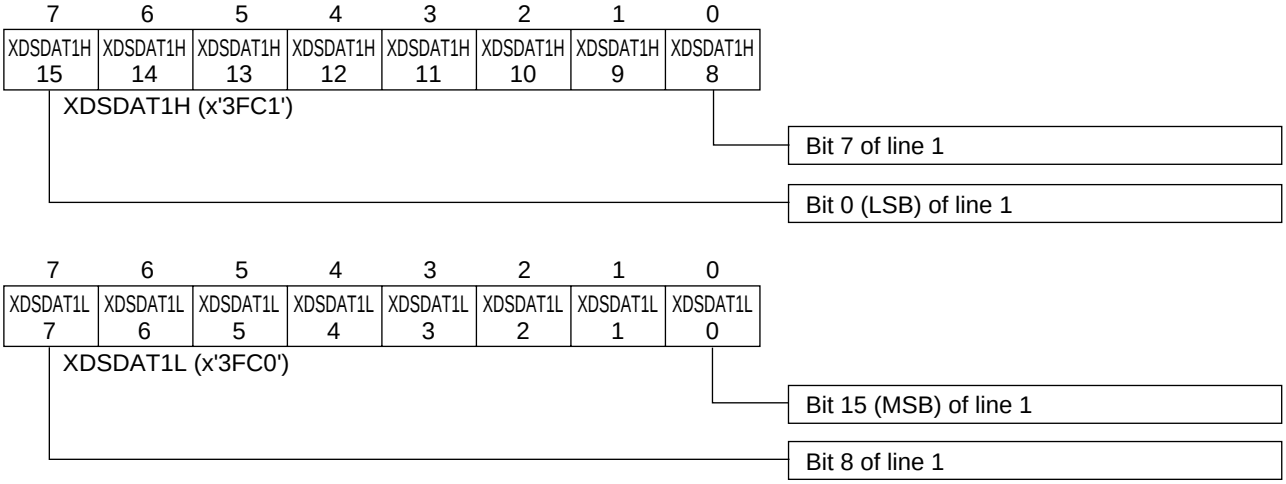
XDSIR	XDS interrupt control flag
0	No interrupt request
1	Generate interrupt request

XDSIE	XDS interrupt enable flag
0	Disable interrupts
1	Enable interrupts

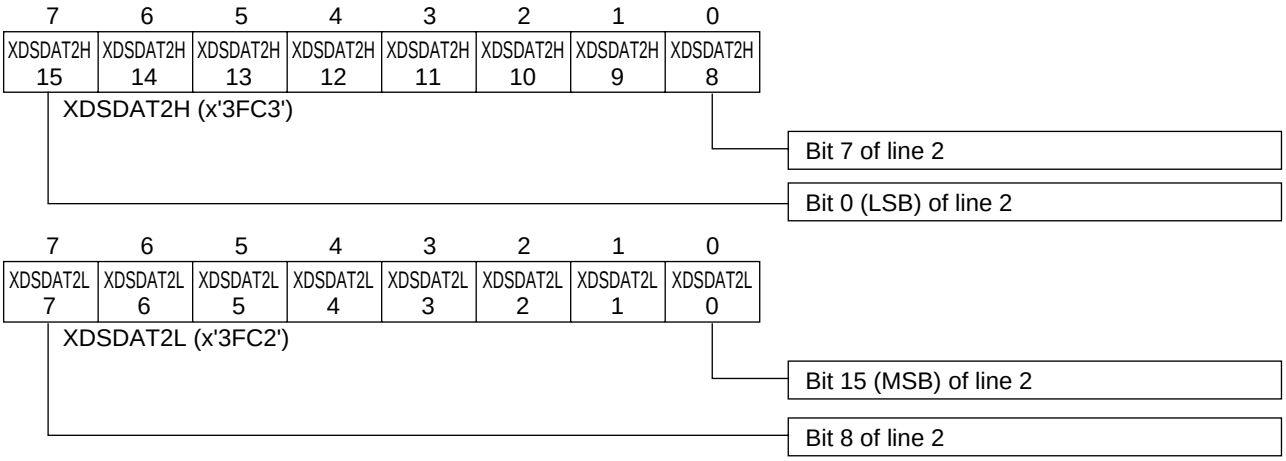
XDSLV1 to 0	XDS interrupt priority level
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Interrupts prohibited (level 3)

11-2-5 XDS Data Register

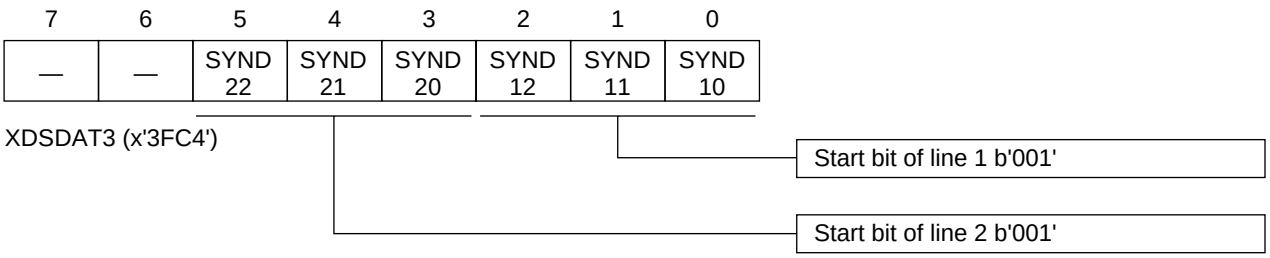
XDS line 1 data: XDSAT1 (read-only)



XDS line 2 data: XDSDAT2 (read-only)



XDS start bit data: XDSDAT3 (read-only)



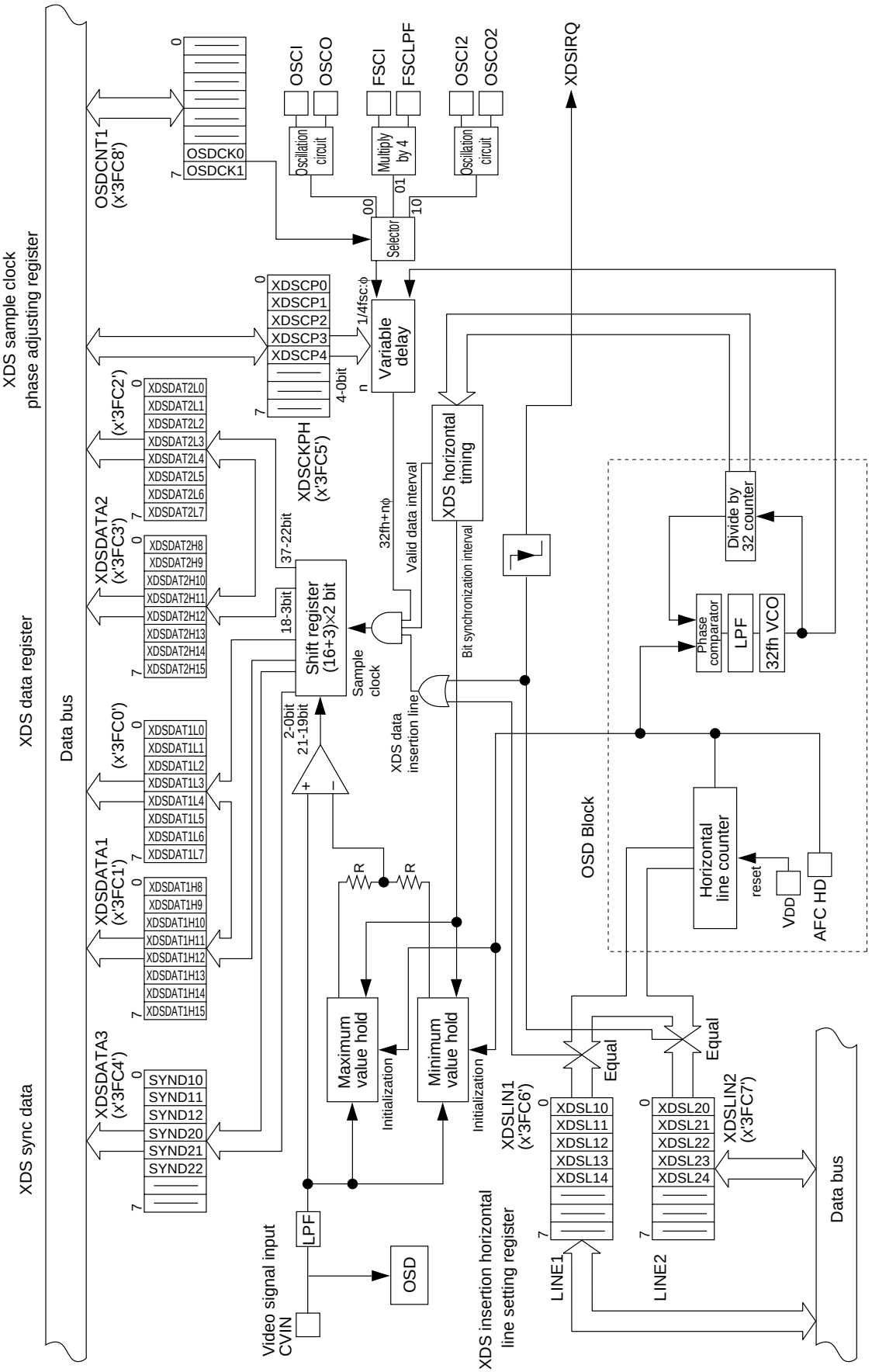


Figure 11-2-4 XDS Unit, Block Diagram

11-3 Synchronization Judgment

This is the function of judging whether or not the video signal is being input. The period of the signal after sync slicing the signal input at CVIN2 is measured by counting the microcomputer system clock, and the decision is made whether that period is approximately equal to the period of the horizontal synchronization signal. Further, the counter is made UP/DOWN based on the result of this period judgment, the signal is summed, and the decision of whether or not the video signal is being input is made based on the result of summation. Based on the result of this judgment, the registers of OSD are manipulated by software to change over to blue-back, etc.

11-3-1 Synchronization Judgment Circuit Configuration

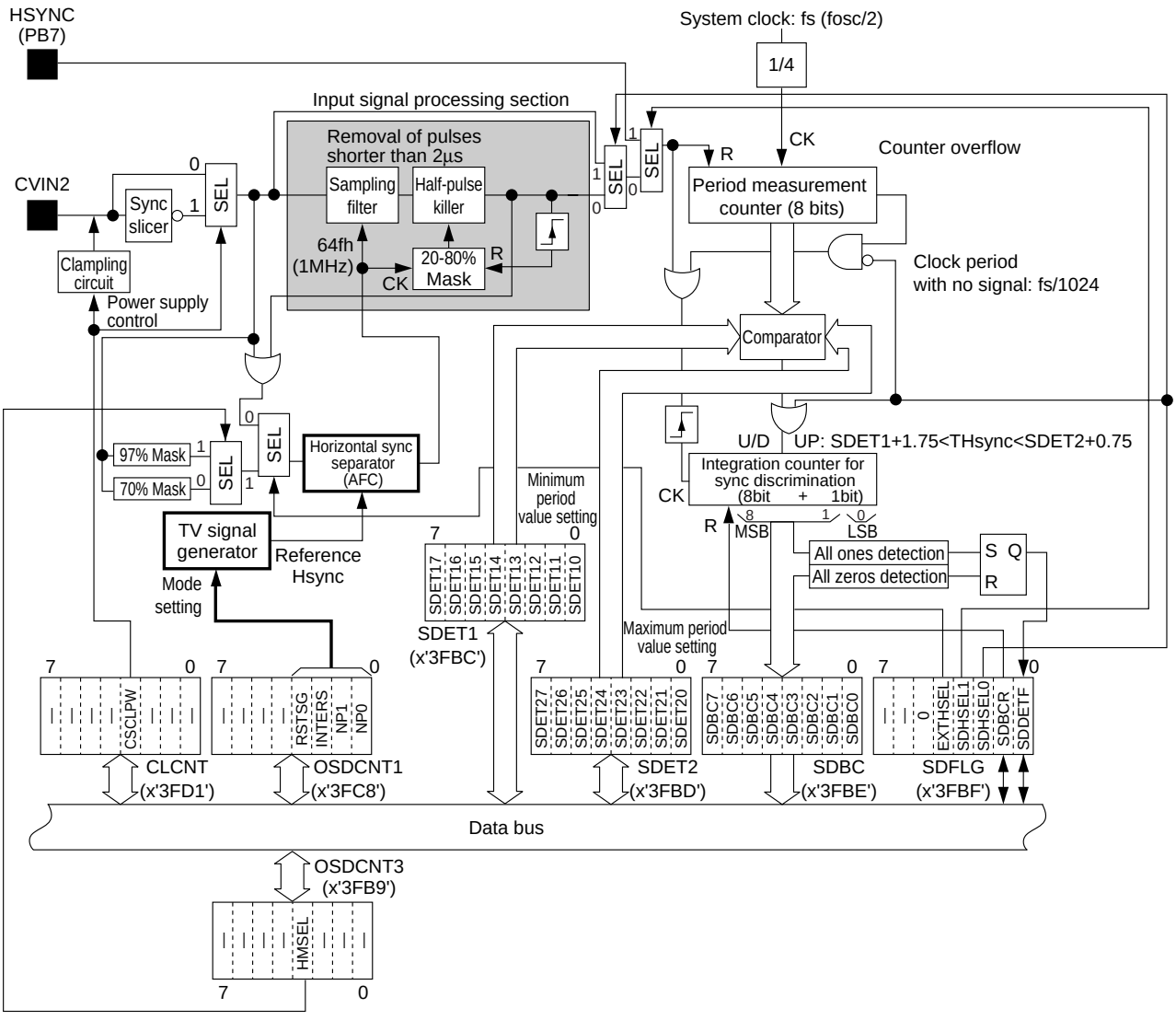


Figure 11-3-1 Synchronization judgment circuit block diagram

The overall block diagram is shown in Fig. 11-3-1. This block diagram is described below by dividing it into the following four sections.

(1) Input signal processing section
(2) Period measurement section
(3) Period judgment section
(4) Periodic signal judgment section

(1) Input signal processing section

The noise filter and half-pulse killer circuits are constructed using the signal obtained by frequency division of the 128-times clock of the horizontal synchronization signal generated by the OSD horizontal sync separator (see Section 11-4 "Sync separation function"). The noise filter removes noise of less than 2µs by checking for two successive matches of sampling using the 64fh signal generated by frequency halving of the 128fh signal (fh is the horizontal synchronization signal frequency) that is the VCO output of the AFC circuit of the horizontal sync separator. The half-pulse killer removes the equivalent noise at 50% using the 20-80% mask signal generated by the AFC circuit. Since the mask signal operates by counting the AFC clock, the masking position shifts towards the smaller side if the resistance connected to the AFCC and AFCLP pins becomes larger. This phenomenon affects the field judgment described in the next chapter. Also, it is necessary to operate the AFC of the horizontal sync separator when using this function.
Figure 11-3-2 illustrates the operation explained in this section.

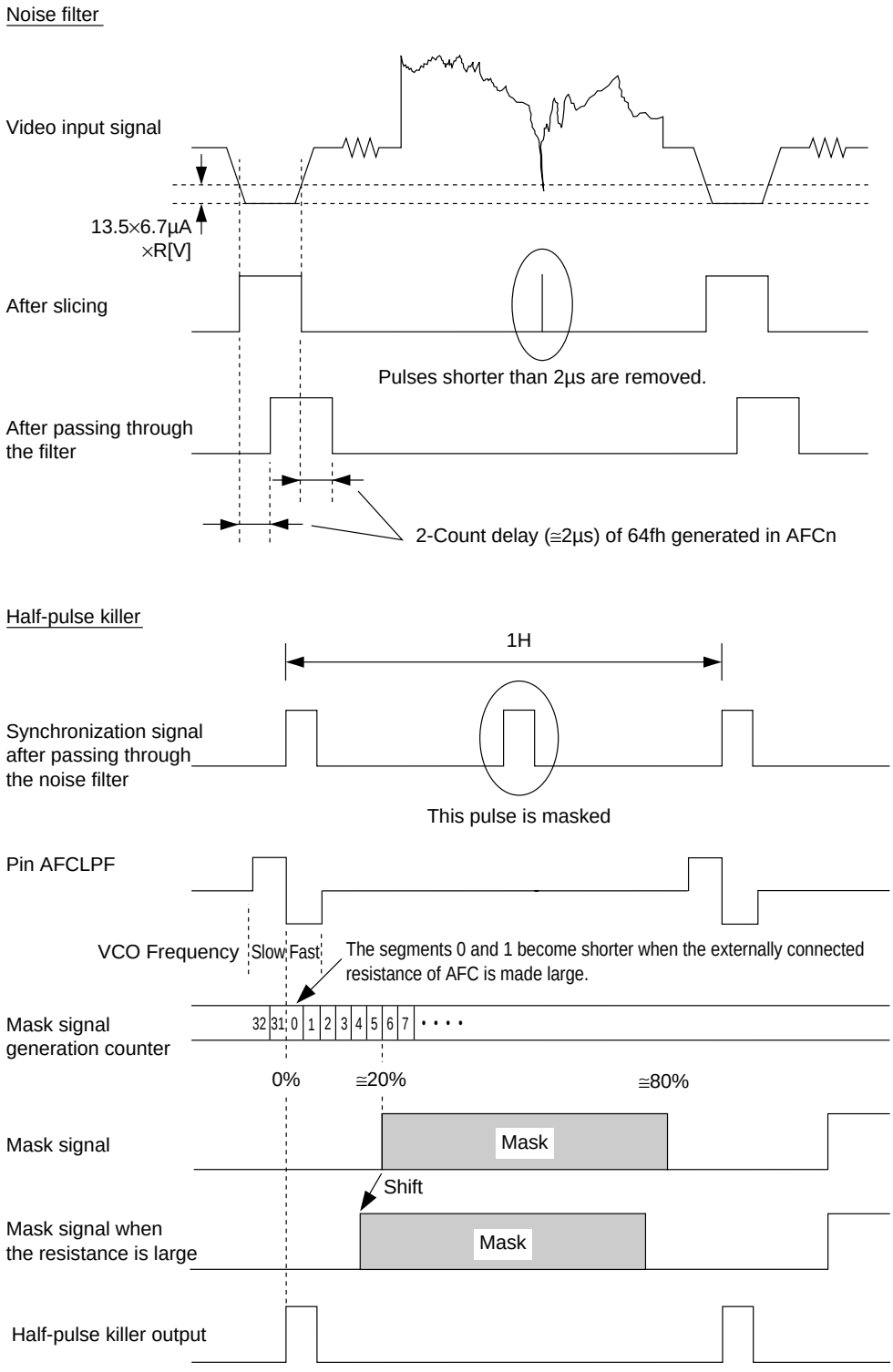


Figure 11-3-2 Operation of the input signal processing section

(2) Period measurement section

The period is measured by counting the fs/4 signal obtained by 1/4 frequency division of the system clock between successive rising edges of the pulses passing through the input signal processing section. The measurement result is -

Edge interval x fs/4,

and the center value is the following when fosc is 14.31818MHz and fh is 15.734kHz (NTSC):

$1/15.734 \times 10^3 \times (14.31818\ \text{MHz}/2)/4 = 114(72h)$

(3) Period judgment section

The result of measurement done in the period measurement section is compared with registers SDET1 (x'3FBC', R/W) and SDET2 (x'3FBD', R/W) every time an edge is input. Also, since no edges are input when the input signal is not present, the timing of comparison is taken as the time when an overflow occurs in the counter of the period measurement section. A '1' is output when the result of comparison satisfies the following inequality, and a '0' is output otherwise:

SDET1+1.75<Measured period value (THsync)<SDET2+0.75

The period between edges when there is no input signal is 1024/fs[s]. The set values of SDET1 and SDET2 in the case of NTSC is:

SDET1 $\leftarrow$ 114(72h) - α

SDET2 $\leftarrow$ 114(72h) + α

(4) Sync signal judgment section

The count is incremented or decremented (UP/DOWN) at every input edge and overflow pulse of the period judgment counter, and the sync judgment is made based on the result from this counter. This sync judgment summation counter SDBC (x'3FBE', Read only) can be started under software control using the corresponding reset register SDBCR (bit 1 of x'3FBF', R/W). In addition, there are two judgment methods available, namely, the method of monitoring register SDDETF (bit 0 of x'3FBF', Read only) which is set or reset by ALL '1' or ALL '0' states of the period judgment summation counter SDCR, and the method of judging by software sync judgment summation counter SDBC.

11-3-2 Method of Using Sync Judgment

It is necessary to activate the AFC part of the horizontal sync separator for carrying out sync judgment. This requires the following settings.

(1) Set the OSD section clock by setting OSDCK1 and OSDCK0 (bits 7 and 6 of register OSDCNT1 (x'3FC8')), and start the operation of the TV signal generator by setting NP1, NP0, INTERS, and RSTSG (bits 3 to 0 of OSDCNT1 (x'3FC8')).

(2) Set the CVIN2 pin as an input by setting to '0' bit 4 of PBDIR (x'3F36').

(3) If the signal input to CVIN2 is a video signal, turn on the clamping circuit power supply by setting the CSCLPW flag (bit 3 of CLCNT: x'3FD1') to '1'. Also, when inputting the synchronization signal (logic signal), make the clamping circuit power supply '0'. In this case, input the signal so that the horizontal synchronization signal becomes active High.

(4) Turn on the power supply of the AFC low pass filter by setting the LPFPW flag (bit 1 of PCSTA, x'3FD3') to '1'.

(5) Write x'E0' in VCOCNT (x'3FD0'), thereby turning on the VCO power supply and executing its calibration. The calibration gets completed 1 ms after this setting is made.

The above are the settings for operating the AFC circuit of the horizontal sync separator, and are carried out during the initial settings. The following are the settings for sync judgment.

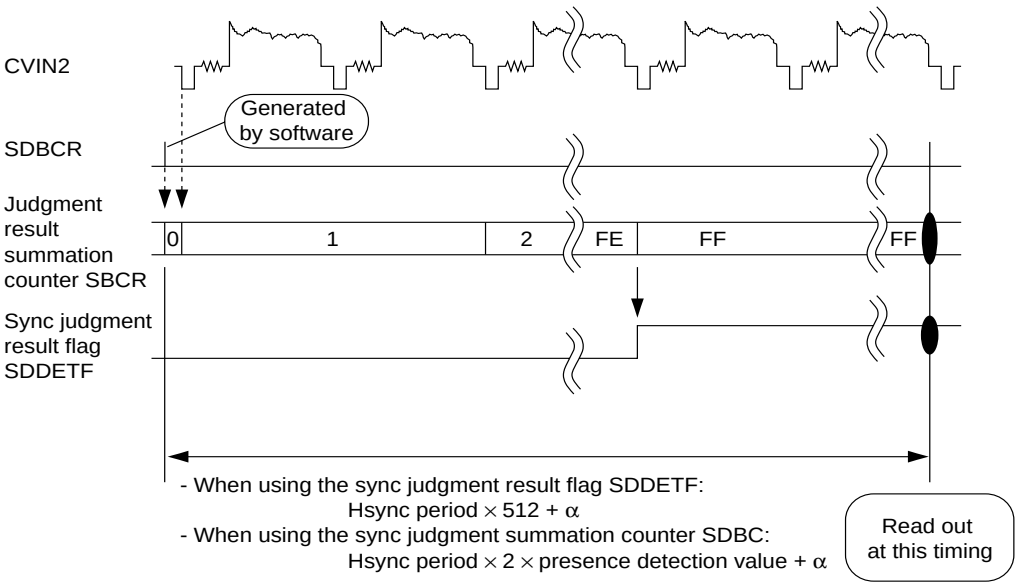
(6) Set the period judgment registers SDET1 (x'3FBC') and SDET2 (x'3FBD'). Set a value in SDET2 that is higher than the value in SDET1. For example, the following settings need to be made when a range of ±10% is to be set when fosc is 14.31818MHz and fh is 15.734kHz (in the case of NTSC signals):

SDET1 ← x'65'
SDET2 ← x'7C'

(7) There are two methods of sync judgment. One method consists of reading out the sync judgment result flag SDDETF (bit 0) of the sync judgment register SDFLG (x'3FBF'), and the other method consists of reading out the judgment result summation counter SDBC (x'3FBE', Read only) and making the judgment by software. When judging by software, since the decision may flutter near the threshold value, hysteresis is incorporated by providing a difference between the value at which sync signal is judged to be present and the value at which the sync signal is judged to be absent.

Operation timing chart

In the case of a normal signal



In the case of a change from normal signal to no signal

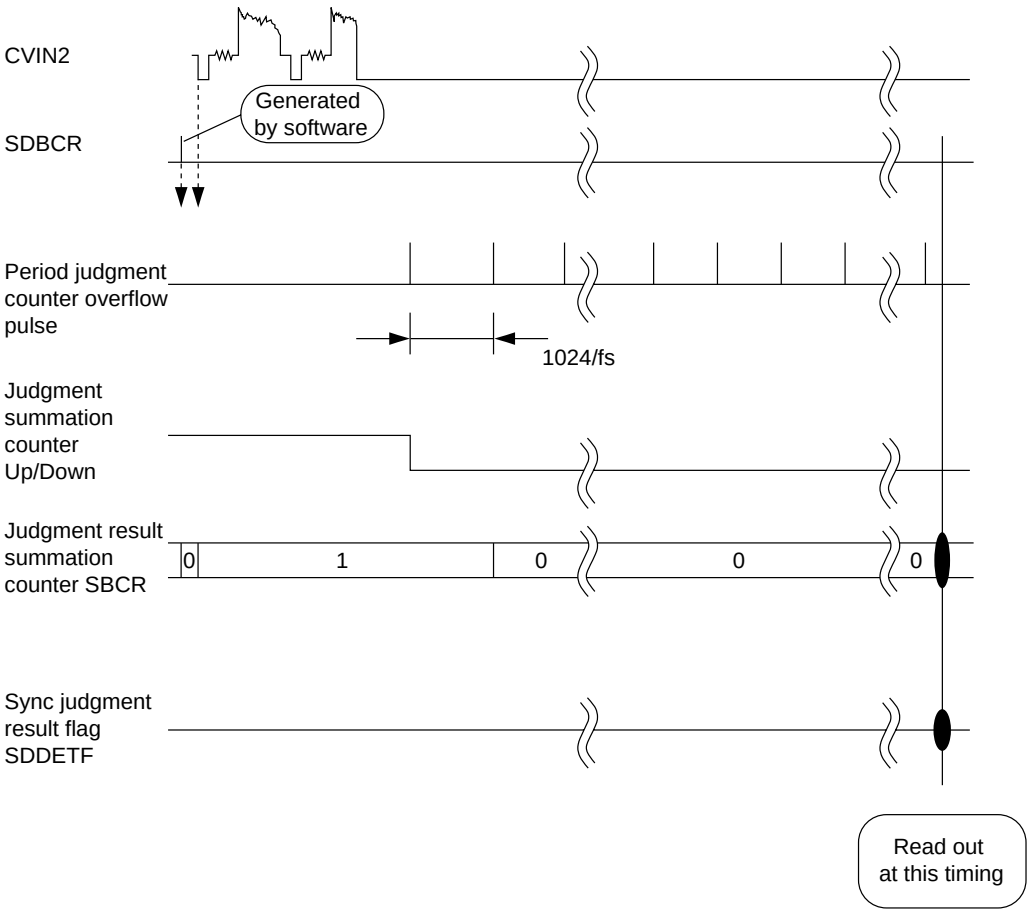


Figure 11-3-3 Operation of the sync judgment section

(8) The operation is started by setting and clearing SDBCR by software. Next, the sync judgment flag SDDETF or the judgment result summation counter SDBC is monitored after a fixed period of time. The minimum time duration required for judgment is -

- In the case of sync judgment result flag SDDETF:
  Hsync period×512

- In the case of sync judgment summation counter SDBC:
  Hsync period×2×(software signal presence detection set value)

In actuality, the set values are greater than the above values.

Further, there is also the method in which the resetting is done by software only at the beginning, and thereafter, resetting is not done but the decision is made based on sync judgment flag SDDETF or counter SDBC. In this case, the minimum time duration for recognizing signal absence from the signal present state are -

- In the case of sync judgment result flag SDDETF:
  2048/fs [s]

- In the case of sync judgment summation counter SDBC:
  4/fs×2×(set value of signal absence detection by software) [s]

Therefore, the period of judging by software is set to be somewhat longer than the above time durations.

11-4 Synchronization Signal Separation Function

11-4-1 Synchronization Signal Separation Function Summary

The synchronization signal separation function consists of the function of separating the vertical synchronization signal and the horizontal synchronization signal from the composite synchronization signal input from the CVIN2 pin and the field judgment function.

The block diagram is shown in Fig. 11-4-1. The sync separator consists of the three blocks of the horizontal sync separator section, the vertical sync separator section, and the field discrimination section.

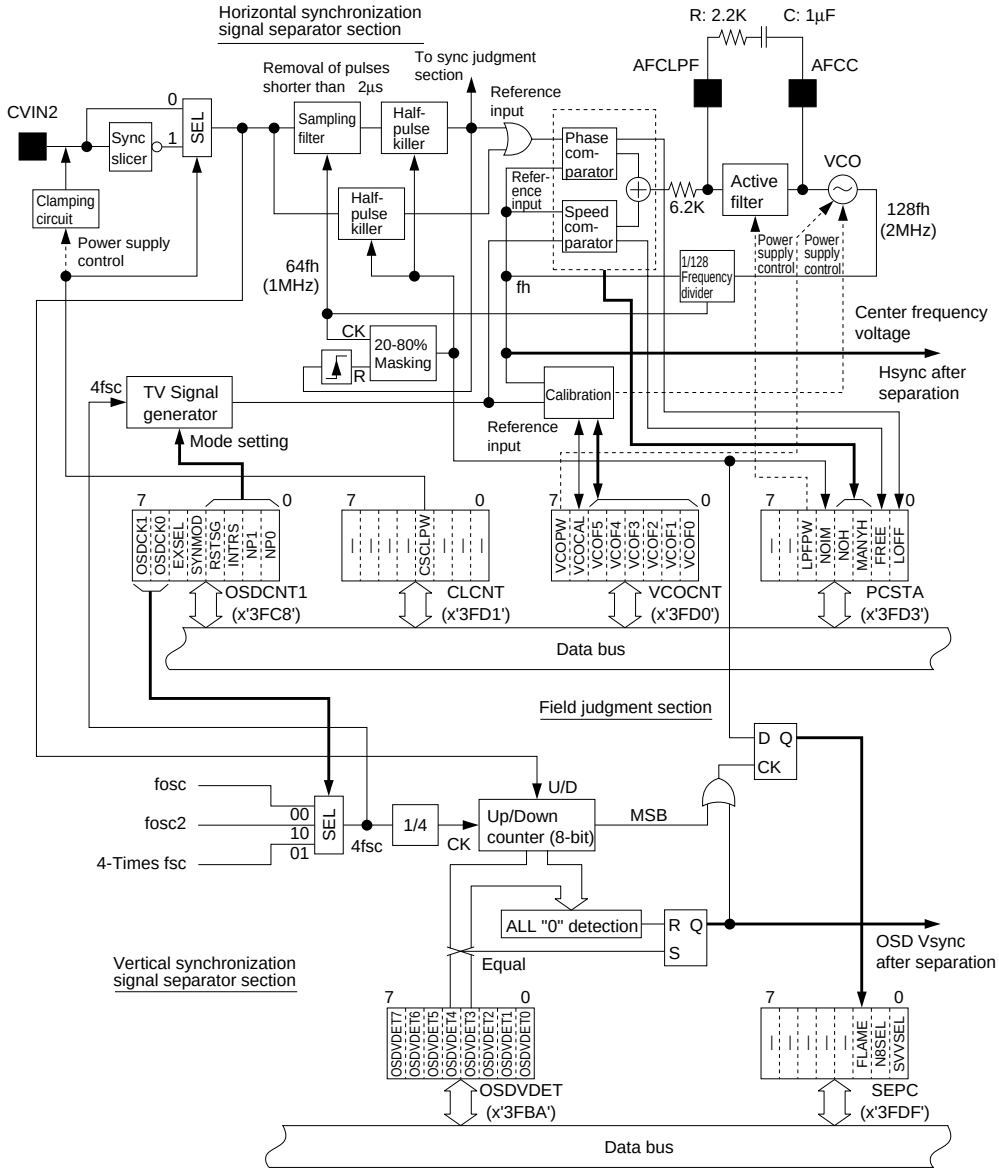


Figure 11-4-1 Synchronization signal separator block diagram

11-4-2 Horizontal Synchronization Signal Separation Function

The video signal input at CVIN2 is either passed through the sync slicer or is directly input to the noise filter and half-pulse killer circuits (selected by the CSCLPW flag, bit 3 of register CLCNT x'3FD1'; 1: video signal input, 0: sync signal input) and the Hsync signal is extracted from these circuit blocks. Further, this signal is taken as the reference in an AFC circuit to generate a stable horizontal synchronization signal even during weak electric fields, during ghost image generation, special reproduced screens, and during tear off of the reproduced screen.

The AFC section comprises phase and speed comparators, low pass filter, VCO, and 128fh frequency divider circuit. The phase comparator compares the phase of the internal Hsync signal generated by 1/128 frequency division of the VCO output with the phase of the reference Hsync signal generated from the signal input at CVIN2, and the speed comparator compares the frequency of the internal Hsync signal with the horizontal synchronization signal generated by the TV signal generator. Because of this, it is necessary to ensure normal operation of the TV signal generator even during external synchronization. In addition, since the horizontal synchronization signal generated by the TV signal generator is used as the reference for calibration of the VCO, it is necessary to ensure normal operation even during calibration. When the operating clock of OSD is selected as the quadruple frequency (4-times fsc) signal, the frequency of the fsci signal can become unstable during external synchronization. In such situations, make sure change the OSD clock to a stable clock and also set the TV mode corresponding to that frequency by making appropriate settings of bits 7 and 6 and bits 1 and 0 of OSDCNT1 (x'3FC8'). For example, when fosc is 14.31818MHz, fsci is 4.43362MHz, the 4-times frequency signal is selected as the OSD clock, and PAL mode is selected, it is necessary to ensure the operation so that stable fsci is input during calibration in the external synchronization mode. However, if it is difficult to satisfy this condition by the peripheral circuits, select fosc by setting bits 7 and 6 of OSDCNT1 to '00' and also select the NTSC mode by setting bits 1 and 0 to '00'.

The low pass filter and VCO have corresponding power supply control bits, LPFPW (bit 5 of PCSTA x'3FD3') and VCOPW (bit 7 of VCOCNT x'3FD0'). When operating OSD in the external synchronization mode, ensure that these bits are set to '1' thereby turning ON the power supply during XDS, sync judgment, and calibration.

Bits 6~0 of register VCOCNT (x'3FD0') are for calibration control and control of the VCO center frequency. Calibration is started when the VCOCAL flag (bit 6) is set to '1'. Calibration ends after about 1 ms. It is possible to confirm the end of calibration by verifying that this bit has become '0'. Bits 5~0 are the data of the VCO center frequency. At the time of calibration, a center value of b'100000' is written in these bits. The new center value is set automatically in these bits at the end of calibration.

When wanting to intentionally shift the center frequency, read out the value after calibration and set a newly computed value.

Bits 4~0 of register PCSTA (x'3FD3') output the AFC status. Bit 4 is the 20-80% masking signal NOIM. Bit 3 NOH and bit 2 MANYH indicate the Hsync input condition. NOH outputs a '1' when the synchronization is lost, and MANYH outputs a '1' when the input is excessive. Bit 1 FREE outputs the result of speed comparison and outputs a '1' when the PLL is locked. Bit 0 LOFF outputs the result of phase comparison and similarly outputs a '1' when the PLL is locked.

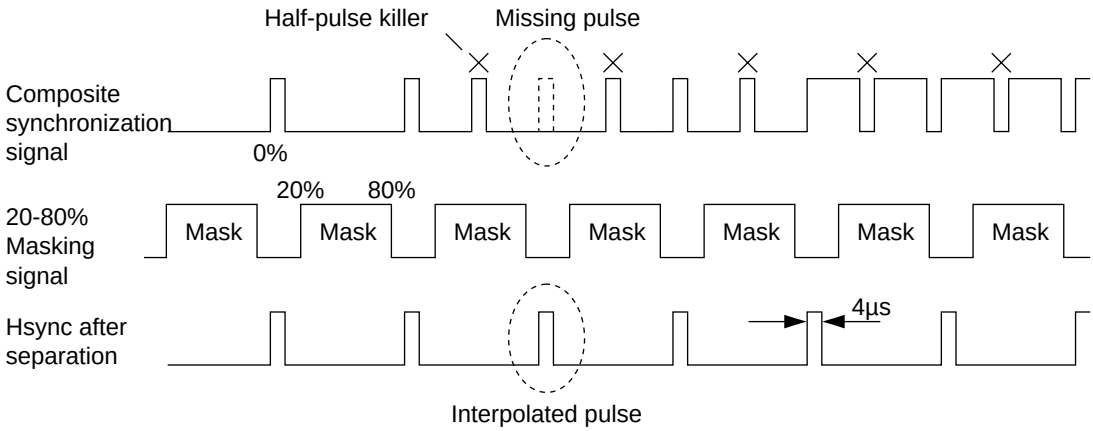


Figure 11-4-2 Diagram explaining horizontal sync separator operation

The external circuit connected to pins AFCLPF and AFCC consist of a resistor R that determines the AFC loop gain and a capacitor C for low band compensation. The gain increases when R is increased and the frequency response also becomes faster. However, when there is jitter in the signal after slicing, such as during weak electric fields, the circuit responds also to the frequency of such jitter. Hence the values of R=2.2K and C=1µF are recommended. Since the changes in the loop gain due to this resistance are determined by the ratio of the resistance with the internal 6.2K resistor, select a value that does not cause problems even when there is shift of 10%.
In addition, a low pass filter is required to eliminate high-frequency components such as chroma component from the input at the CVIN2 pin. Even the slicing level is determined by the resistance part of this filter. (See Section 11-1-6 "OSD Peripheral Circuits".)

11-4-3 Vertical Synchronization Signal Separation Function

Similar to horizontal sync separation, the sync sliced signal is used to carry out UP/DOWN control of the 8-bit counter that counts the OSD section clock selected by bits 7 and 6 of OSDCNT1, and the value summed in the counter is compared with the value set in register OSDVDET (x'3FBA', R/W), thereby separating the Vsync signal for OSD.

The value set in OSDVDET is the width up to judging as Vsync, and this width is determined according to the following equation:

Value set in OSDVDET×(4/OSD operating clock),

and since the normal OSD operating click is 4fsc, this width is the product of fsc and the value set in OSDVDET. In the case of NTSC, this width is -

Value set in OSDVDET×1/3.579545MHz.

Do not set a value near 1/2 (31.8µs) in OSDVDET (setting near x'71' in the case of NTSC). A shift of 1 occurs in the value of the counter that counts the Hsync pulses after OSD separation, and hence the OSD characters may "dance" in the vertical direction.

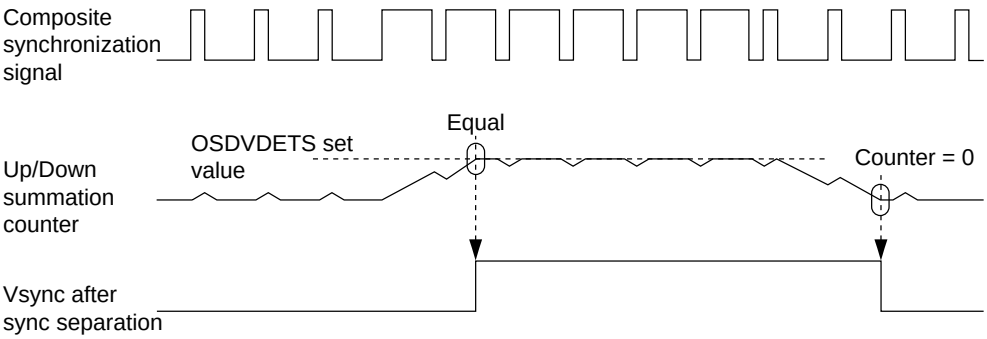


Figure 11-4-3 Diagram explaining vertical sync separator operation

Further, even the XDS line specification may get shifted. This is described in Fig. 11-4-4. The counter determining the XDS position in the vertical direction counts Hsync after sync separation. Also, the reset interval is the Vsync period after sync separation. Therefore, if the width for judging Vsync based on the value set in OSDVDET is less than 1/2H (the normal method of use), the counter of field 2 becomes larger by 1. Therefore, the values of XDSLIN1 and XDSLIN2 so that the value of field 2 is one larger than the value of field 1.

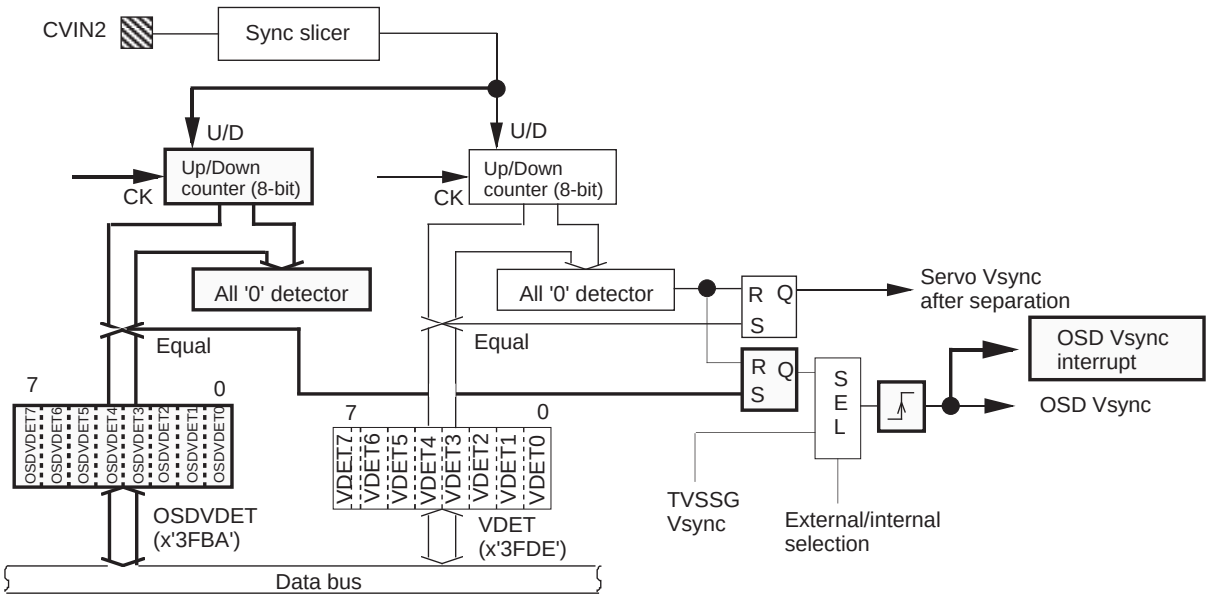
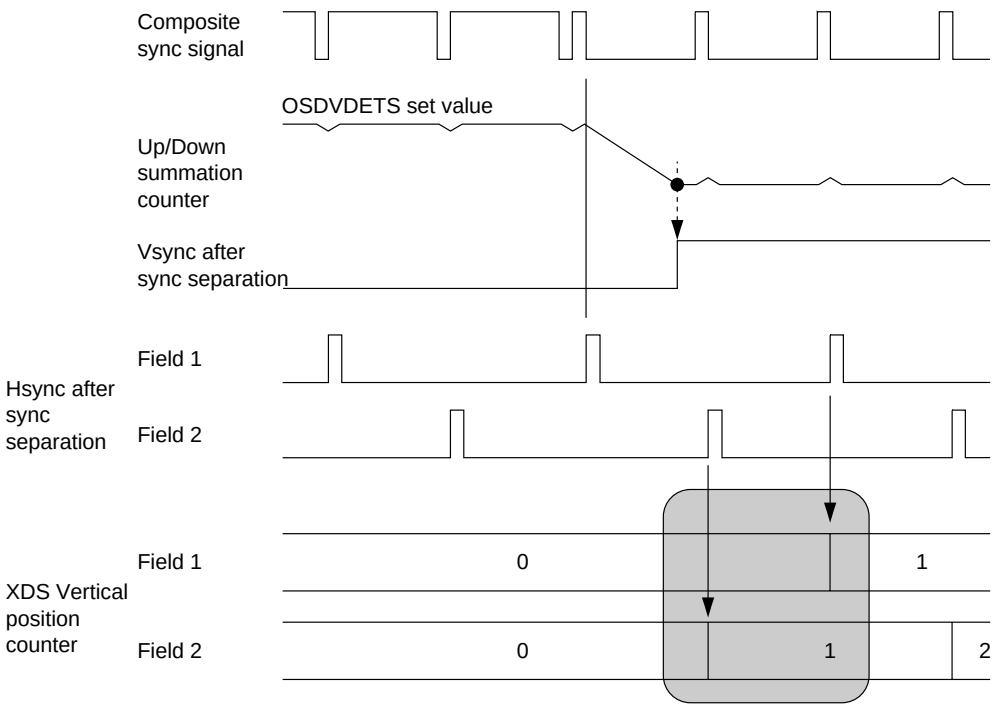


Figure 11-4-4

When the displacement due to OSDVDET is less than 1/2 H (under x'70').



A displacement of one count occurs.
The values of XDSLIN1 and XDSLIN2 for field 2 must be set to values one larger than those for field 1.

When the displacement due to OSDVDET is 1/2 H or greater (x'70' or over).

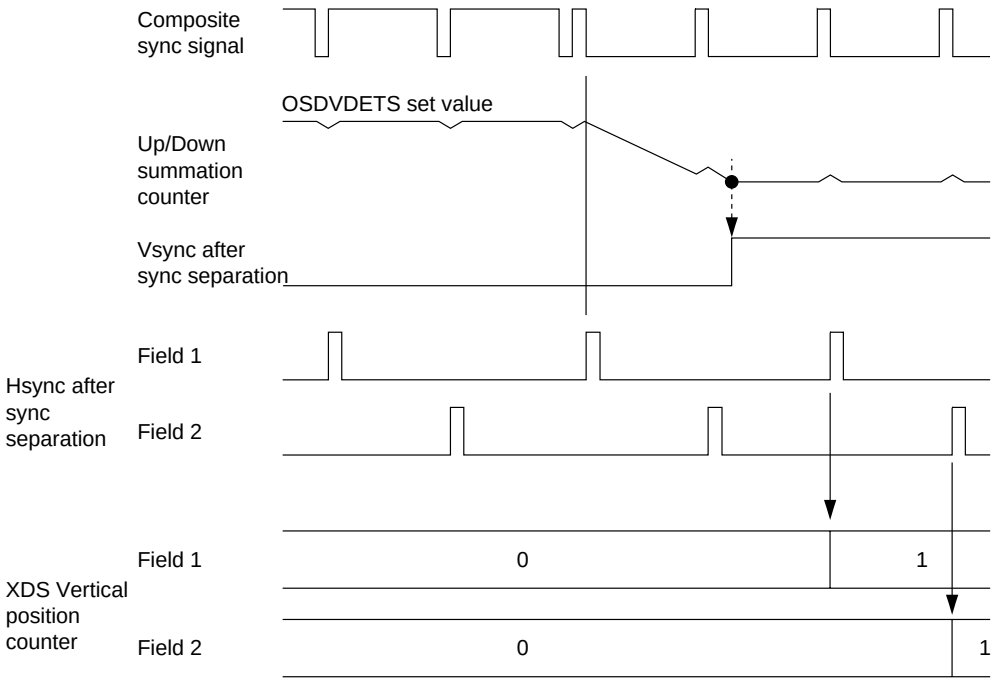
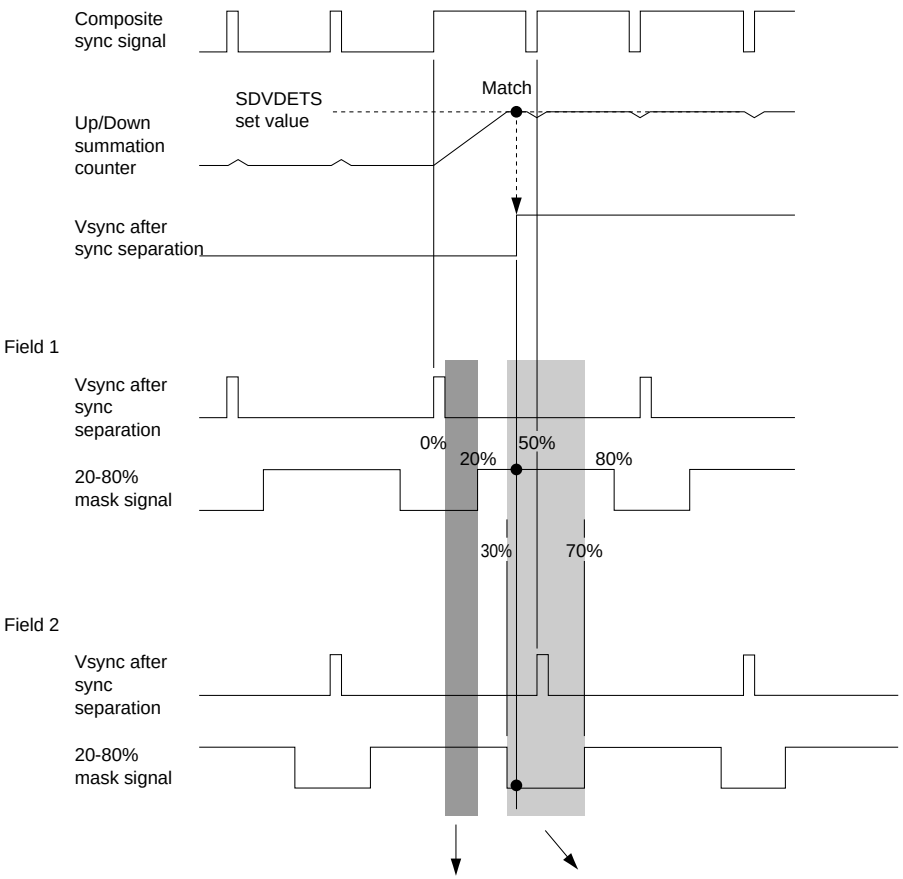


Figure 11-4-5 Diagram explaining the operation of shifting XDSLIN data setting

11-4-4 Field Judgment Function

Field judgment uses the 1/2H shift of Vsync and Hsync in field 1 and field 2, and the 20-80% masking signal that masks the Hsync half-pulses is latched and output at the rising edge of the above Vsync after separation. Therefore, the judgment result varies depending on the position of Vsync after separation based on OSDVDET (x'3FBA', R/W). Figure 11-4-5 shows the operation of the field judgment function. Since there is a 50% (1/2H) shift in field 1 and field 2, even the 20-80% mask signal also gets shifted by 50%. Therefore, the two ranges in which the field 1 and field 2 mask signals differ are 0 to 20% (range 1) and 30 to 70% (range 2). Although field 1 will be 0 and field 2 will be 1 for range 1 (the 0 to 20% range), these are reversed for range 2 (the 30 to 70% range); that is field 1 will be 1 and field 2 will be 0 for the 30 to 70% range. The relationship between the set values of OSDVDET and the output is shown in the table in Fig. 11-4-5. Since the operation is not stable near the boundary, values with sufficient margin are given in the table. In the range of x'80' ~ x'FF' of the set value of OSDVDET, the judgment result is output when the up/down counter is x'80'.



Field judgment result / OSD Clock	Judgment range 1	Judgment range 2
	Field 2: 1, Field 1: 0	Field 2: 0, Field 1: 1
14.31818 MHz	13h to 2Bh	47h to 70h
17.7345 MHz	17h to 32h	58h to FFh(80h)

Figure 11-4-6 Diagram explaining the field judgment operation

11-4-5 Method of Using Sync Separation Function

The common initial settings for the sync separator section and field judgment are made according to the following procedure.

(1) Set OSDCK1, OSDCK0, NP1, NP0, INTERS, and RSTSG of OSDCNT1 (bits 7, 6, 3, 2, 1, and 0 of OSDCNT1 x'3FC8'), thereby determining the operating clock of the OSD section and starting the TV signal generator.
(2) Input CVIN2. Set bit 4 of PBDIR (x'3F36') as input (='0').
(3) When a video signal is input, turn ON ('1') the CVIN2 pin clamping circuit power supply CSCLPW (bit 3 of CLCNT x'3FD1') thereby supplying power to the sync signal slicer circuit. When a sync signal is input, set this bit to '0'. In this case, the signal polarity of Hsync is active High.
(4) Set the Vsync detection sensitivity (OSDVDET: x'3FBA') of the Vsync separator circuit.
(5) Input composite sync signal from CVIN2.

The initial setting of the sync separator section is made according to the following procedure.

(6) Set the external synchronization signal selection EXSSEL and sync selection SYNMOD (bits 5 and 4 of OSDCNT1: x'3FC8') to sync separator section output (='01).
(7) Set LPFPW (PCSTA: x'3FD3', bit 5 ='1') and supply power to the LPF of the AFC circuit.
(8) Set VCOPW (VCOCNT: x'3FD0', bit 7 = '1') and turn on the VCO power supply.
(9) Set VCOCAL (VCOCNT: x'3FD0', bit 6 = 0 or 1), and select the AUTO/MANUAL mode of VCO calibration control. (0: MANUAL, 1: AUTO)
When AUTO is selected: Be sure to set VCOF5~F0 (VCOCNT: x'3FD0', bits 5 ~ 0) to '100000'.
When MANUAL is selected: By setting any suitable value in VCOF5~F0 (VCOCNT: x'3FD0', bits 5 ~ 0) it is possible to change the VCO center frequency.

Field judgment is done as follows:

(10) The field judgment result can be obtained by reading out FLAME (bit 2 of SEPC: x'3FDF').
When monitoring the separated sync signal, if OSDVHO (bit 7 of PCSET: x'3FD3') is set to '1' and bits 7 and 6 of P1DIR (x'3F2F') are set to '11', the horizontal synchronization signal is output from Port P16 and the vertical synchronization signal is output from Port P17.

Chapter 12 Appendix

12

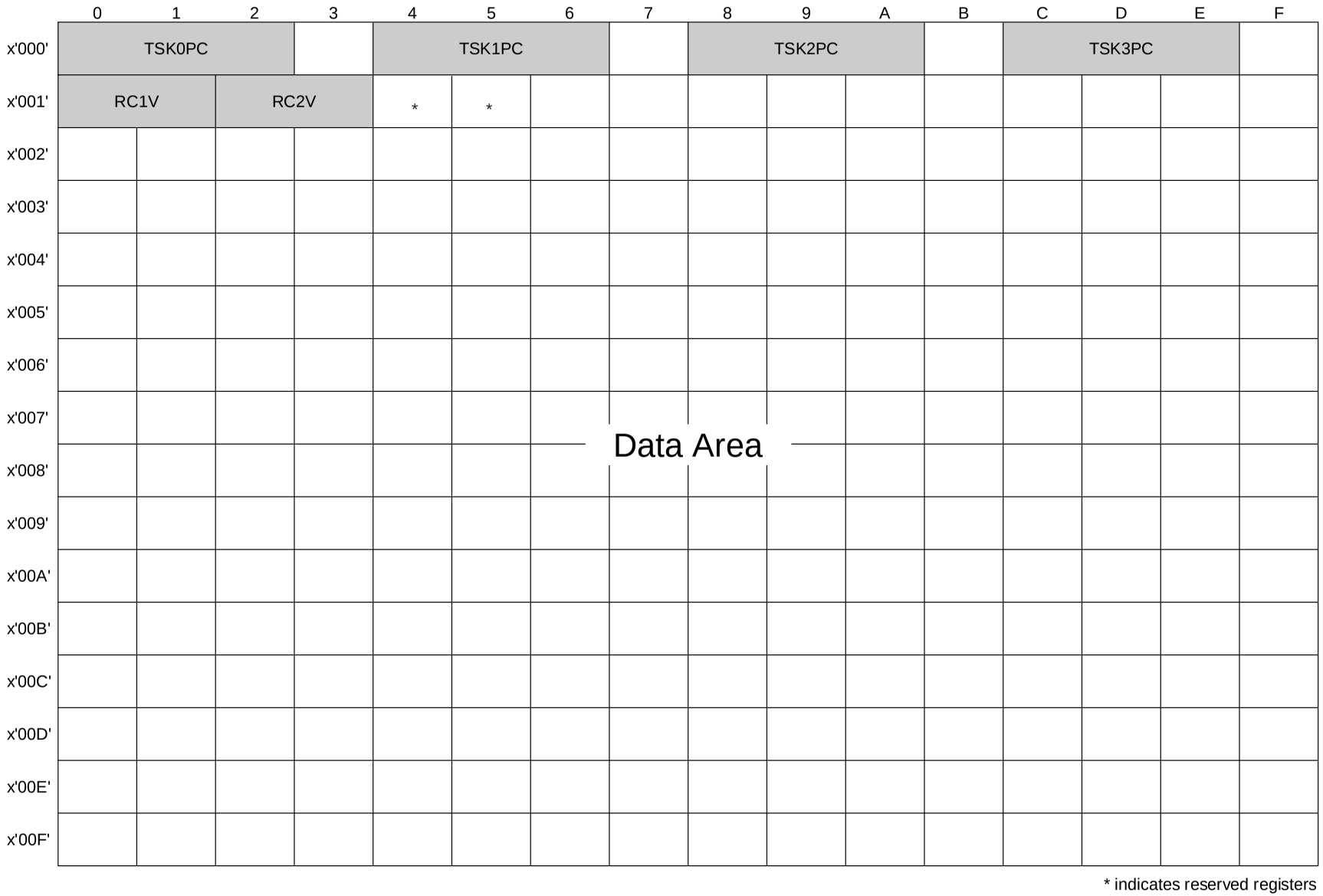


Figure 12-1 RAM Address Map (1/3)

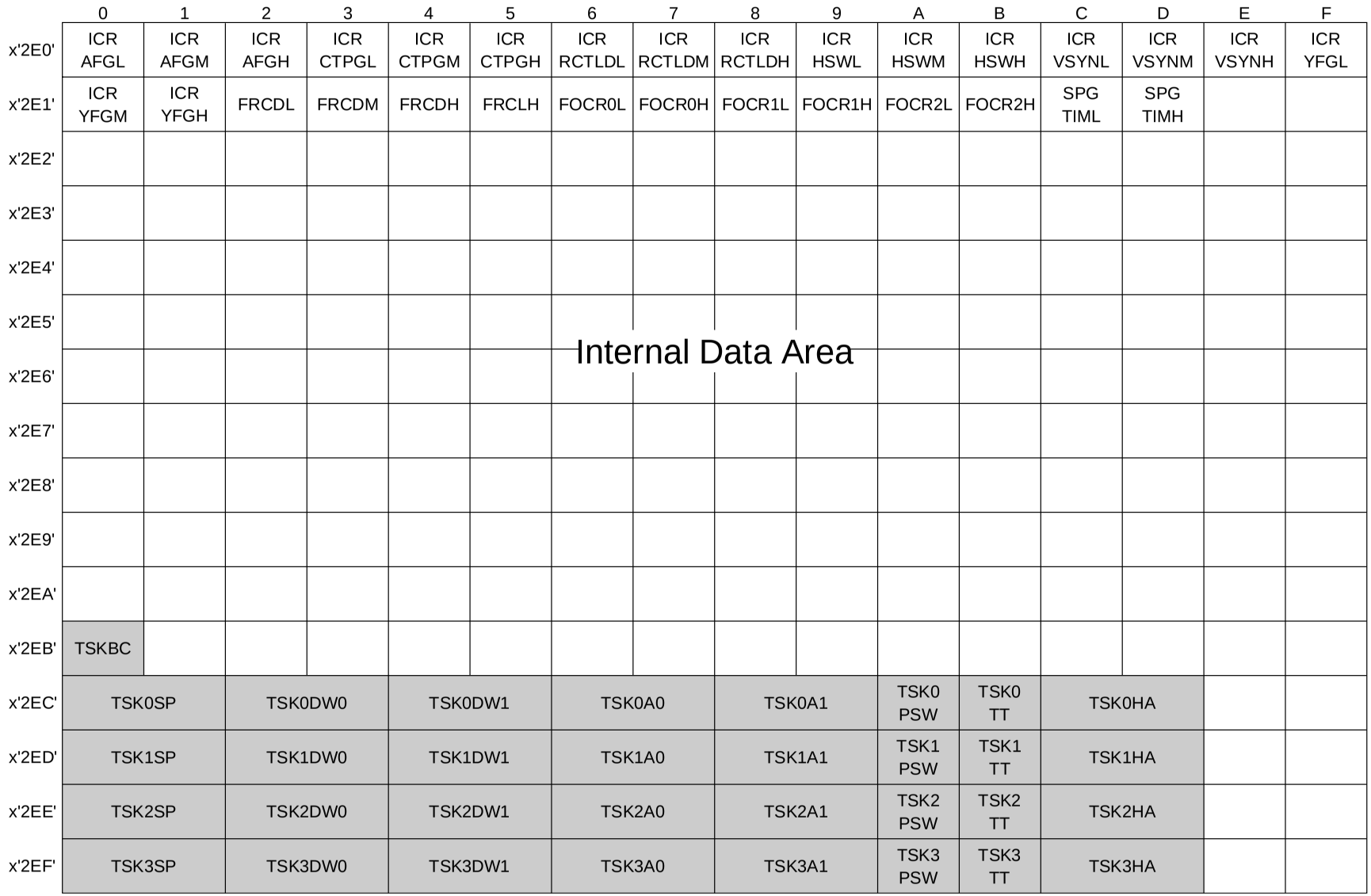


Figure 12-2 RAM Address Map (2/3)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
x'3F0'	CPUM	MEM CTR		*	*	TSK CTR	TSK MD	AU CTR	*	RC CTR	BMEM SR	BMEM DS			BKA LCL	*
x'3F1'	RC0APL	RC0APM	RC0APH	RC1APL	RC1APM	RC1APH	*	*	*	*	*	*	*	WDCNT	DLYCNT	P6CT
x'3F2'	P0OUT	P0IN	P1	P2	P4OUT	P4IN	P5	P6	P7	P8	P9	PA	PB	PC	P0DIR	P1DIR
x'3F3'	P2DIR	P5DIR	P6DIR	P7DIR	P8DIR	PADIR	PBDIR	PCDIR	P1SEL	P2SEL	P6SEL	PASEL	P0DRV	P1PUP	P7PUP	BUZCNT
x'3F4'	TB0L	TB0H	TB1L	TB1H	TB2L	TB2H	TB3L	TB3H	TB4L	TB4H	TB6SL	TB6SH	BC0L	BC0H	BC1L	BC1H
x'3F5'	BC2L	BC2H	BC3L	BC3H	BC4L	BC4H	BC5	ANAMON	BC6SL	BC6SH	TM0	TM1	TM2	ICR2L	ICR2H	SR2FL
x'3F6'	SR2FH	CMPR2L	CMPR2H	SRPC2	SRBC2	SR2RL	SR2RH	OCR4L	OCR4H	HOCR DAT0	SAT CNT	TC6CNT	ADBUF	ADM	PWM14L	PWM14H
x'3F7'	PWM0L	PWM0H	PWM1L	PWM1H	PWM2L	PWM2H	PWM3L	PWM3H	PWMCNT	HOCR DAT1	HOCR DAT2	HOCR BUF1	HOCR CNT	RCTLBUF	SPGDAT	SPGCNT
x'3F8'	AFGCR	ANACNT	CTL GAIN	OPTION	TPG CNT0	TRCNT	AFGPR	AFGDIV	YFGDIV	CTLDIV	DPGM CNT	PGMM TIML	PGMM TIMH	AMSKTM	AMSK FLG	AVPR0
x'3F9'	AVPR1	AVOCR0	AVOCR1	AVCNT	HAMP CNT	SC0TRB	SC0RXB	SC0MD0	SC0MD1	SC0MD2	SC0MD3	SC0CTR	SIM1	SIBUF1	SIC1	I2CR0
x'3FA'	I2CR1	I2CR2	I2CCN	I2CDIR	CLKCNT	SPCLK CNT	YSERL	YSERH	ASERL	ASERH	YSPRL	YSPRH	ASPRL	ASPRH	DADAT	DABUF
x'3FB'	TXQUE1	*	IRQCNT	KEYCNT		*	*	*	*	OSD CNT3	OSD VDET	*	SDET1	SDET2	SDBC	SDFLG
x'3FC'	XDS DAT1L	XDS DAT1H	XDS DAT2L	XDS DAT2H	XDS DAT3	XDS CKPH	XDSLIN1	XDSLIN2	OSD CNT1	ROM ENDA	*	*	*	*	*	*
x'3FD'	VCOCNT	CLCNT	DOTCNT	PCSTA	*	*	*	*	VP	HP	VOFS	VLIN	CMPLIN	OSD CNT2	VDET	SEPC
x'3FE'	*	NMICR	IRQ0 ICR	IRQ1 ICR	IRQ2 ICR	IRQ3 ICR	IRQ4 ICR	KEY ICR	YFG ICR	AFG ICR	TC0 ICR	TC1 ICR	TC2 ICR	TC3 ICR	TC4 ICR	TC6 ICR
x'3FF'	CTL ICR	HSW ICR	VSY ICR	SPG ICR	FOCR 0ICR	FOCR 1ICR	FOCR 2ICR	OSD ICR	XDS ICR	SIF0 ICR	SIF1 ICR	SIF2 ICR	AD ICR	OSDV ICR	*	*

* indicates reserved registers

Figure 12-3 RAM Address Map (3/3)

RAM for Saving Task Program Counter

This RAM is for saving the program counter for each task. (For system use)

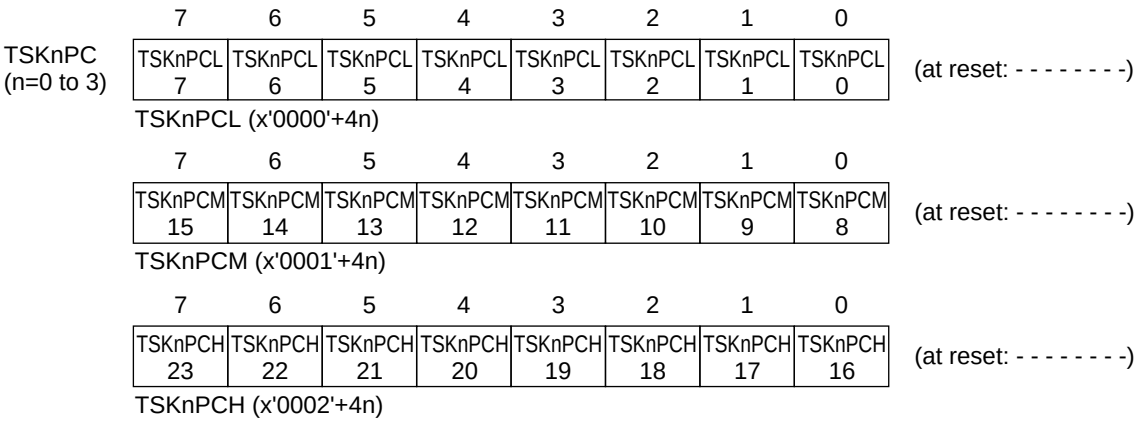


Figure 12-4 RAM for Saving Program Counter of Task n
(TSK0PC: x'0000' to x'0002', R/M)
(TSK1PC: x'0004' to x'0006', R/M)
(TSK2PC: x'0008' to x'000A', R/M)
(TSK3PC: x'000C' to x'000E', R/M)

ROM Correction Indirect Branch Table RAM

When the ROM correction function is operating, if the program counter (PC) value indicated by the leading address of the instruction under execution matches with the ROM correction address setting register RCnAP, the RCnV in this ROM correction indirect branch table is referenced and control is transferred to the program in the RAM.

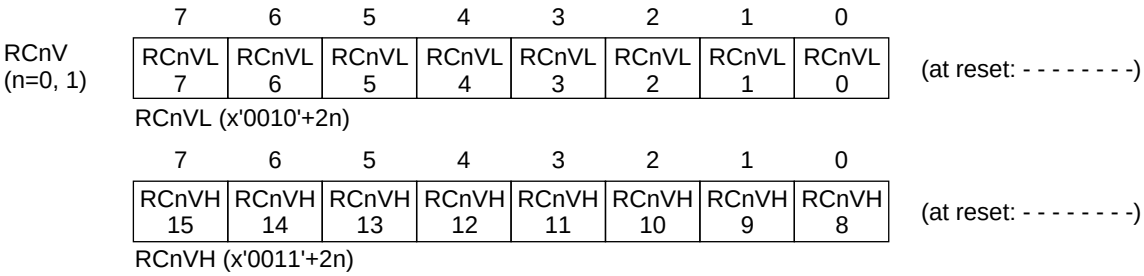


Figure 12-5 ROM Correction Indirect Branch Table
(RC0V: x'0010' to x'00011')
(RC1V: x'0012' to x'00013')

Indirect jumping in the ROM correction function is possible only in the RAM space from x'0000' to x'FFFF', and is restricted to byte boundaries.

AFG Input Capture Register

Read-only register that captures the signal value at occurrence of the AFG signal

ICRAFG

7	6	5	4	3	2	1	0	
—	—	—	—	—	—	ICRAFGH 17	ICRAFGH 16	(at reset: - - - - - - - -)

ICRAFGH (x'2E02')

7	6	5	4	3	2	1	0	
ICRAFGM 15	ICRAFGM 14	ICRAFGM 13	ICRAFGM 12	ICRAFGM 11	ICRAFGM 10	ICRAFGM 9	ICRAFGM 8	(at reset: - - - - - - - -)

ICRAFGM (x'2E01')

7	6	5	4	3	2	1	0	
ICRAFGL 7	ICRAFGL 6	ICRAFGL 5	ICRAFGL 4	ICRAFGL 3	ICRAFGL 2	ICRAFGL 1	ICRAFGL 0	(at reset: - - - - - - - -)

ICRAFGL (x'2E00')

Figure 12-6 AFG Input Capture Register
(ICRAFG: x'2E02' to x'2E00', R)

CTL/PG Input Capture Register

Read-only register that captures the signal value at occurrence of the CTL signal during playback or the YPG signal during recording

ICRCTPG

7	6	5	4	3	2	1	0	
—	—	—	—	—	—	ICRCTPGH 17	ICRCTPGH 16	(at reset: - - - - - - - -)

ICRCTPGH (x'2E05')

7	6	5	4	3	2	1	0	
ICRCTPGM 15	ICRCTPGM 14	ICRCTPGM 13	ICRCTPGM 12	ICRCTPGM 11	ICRCTPGM 10	ICRCTPGM 9	ICRCTPGM 8	(at reset: - - - - - - - -)

ICRCTPGM (x'2E04')

7	6	5	4	3	2	1	0	
ICRCTPGL 7	ICRCTPGL 6	ICRCTPGL 5	ICRCTPGL 4	ICRCTPGL 3	ICRCTPGL 2	ICRCTPGL 1	ICRCTPGL 0	(at reset: - - - - - - - -)

ICRCTPGL (x'2E03')

Figure 12-7 CTL/PG Input Capture Register
(ICRCTPG: x'2E05' to x'2E03', R)

RCTLD Input Capture Register

Read-only register that captures the signal value at the falling edge of the internally generated RCTLD signal

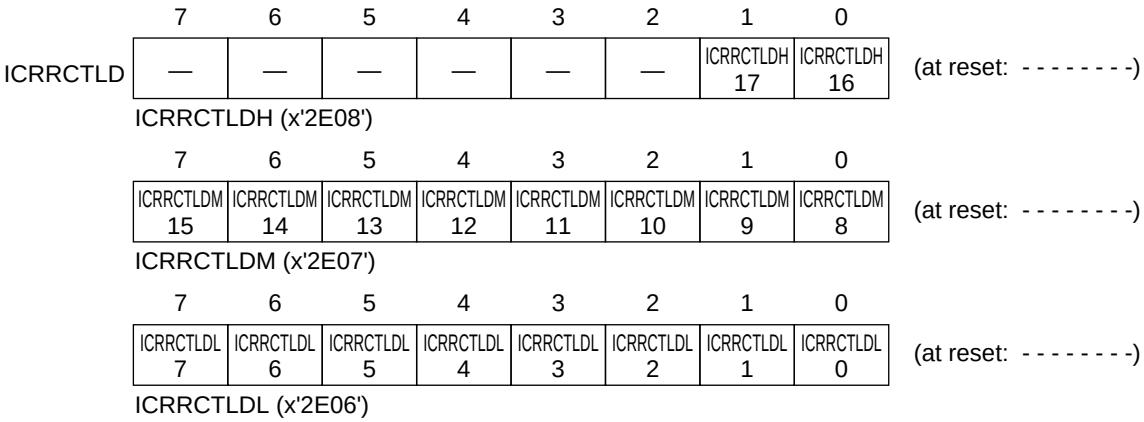


Figure 12-8 RCTLD Input Capture Register
(ICRRCTLD: x'2E08' to x'2E06', R)

HSW Input Capture Register

Read-only register that captures the signal value at the falling edge of the HSW signal

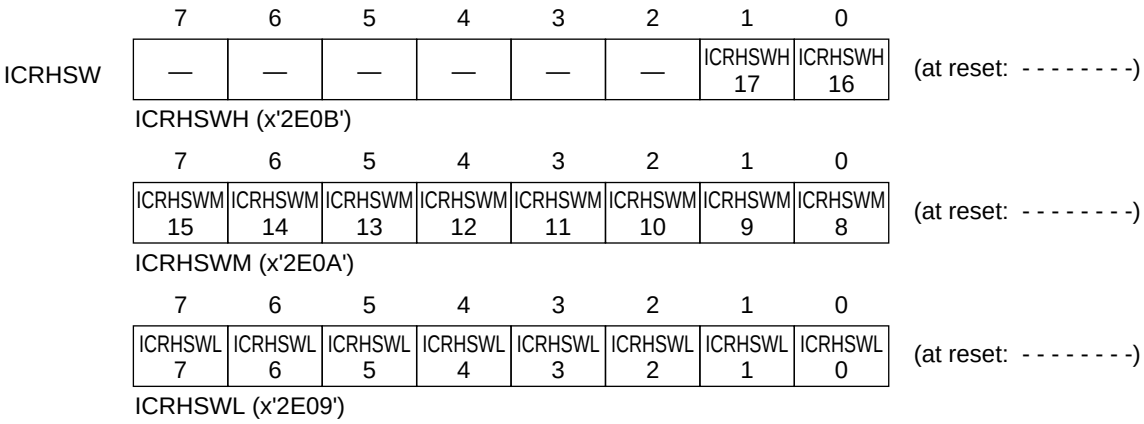


Figure 12-9 HSW Input Capture Register
(ICRHSW: x'2E0B' to x'2E09', R)

VSYNC Input Capture Register

Read-only register that captures the signal value at occurrence of the vertical synchronization signal or at the timing (NF) set in the FOCR0 register

ICRVSYN

7	6	5	4	3	2	1	0	
—	—	—	—	—	—	ICRVSYNH 17	ICRVSYNH 16	(at reset: - - - - - - - -)

ICRVSYNH (x'2E0E')

7	6	5	4	3	2	1	0	
ICRVSYNM 15	ICRVSYNM 14	ICRVSYNM 13	ICRVSYNM 12	ICRVSYNM 11	ICRVSYNM 10	ICRVSYNM 9	ICRVSYNM 8	(at reset: - - - - - - - -)

ICRVSYNM (x'2E0D')

7	6	5	4	3	2	1	0	
ICRVSYNL 7	ICRVSYNL 6	ICRVSYNL 5	ICRVSYNL 4	ICRVSYNL 3	ICRVSYNL 2	ICRVSYNL 1	ICRVSYNL 0	(at reset: - - - - - - - -)

ICRVSYNL (x'2E0C')

Figure 12-10 VSYNC Input Capture Register
(ICRVSYN: x'2E0E' to x'2E0C', R)

YFG Input Capture Register

Read-only register that captures the signal value at occurrence of the YFG signal

ICRYFG

7	6	5	4	3	2	1	0	
—	—	—	—	—	—	ICRYFGH 17	ICRYFGH 16	(at reset: - - - - - - - -)

ICRYFGH (x'2E11')

7	6	5	4	3	2	1	0	
ICRYFGM 15	ICRYFGM 14	ICRYFGM 13	ICRYFGM 12	ICRYFGM 11	ICRYFGM 10	ICRYFGM 9	ICRYFGM 8	(at reset: - - - - - - - -)

ICRYFGM (x'2E10')

7	6	5	4	3	2	1	0	
ICRYFGL 7	ICRYFGL 6	ICRYFGL 5	ICRYFGL 4	ICRYFGL 3	ICRYFGL 2	ICRYFGL 1	ICRYFGL 0	(at reset: - - - - - - - -)

ICRYFGL (x'2E0F')

Figure 12-11 YFG Input Capture Register
(ICRYFG: x'2E11' to x'2E0F', R)

Free-Running Counter Data Register

Data register of the upper 24 bits of the 26-bit FRC

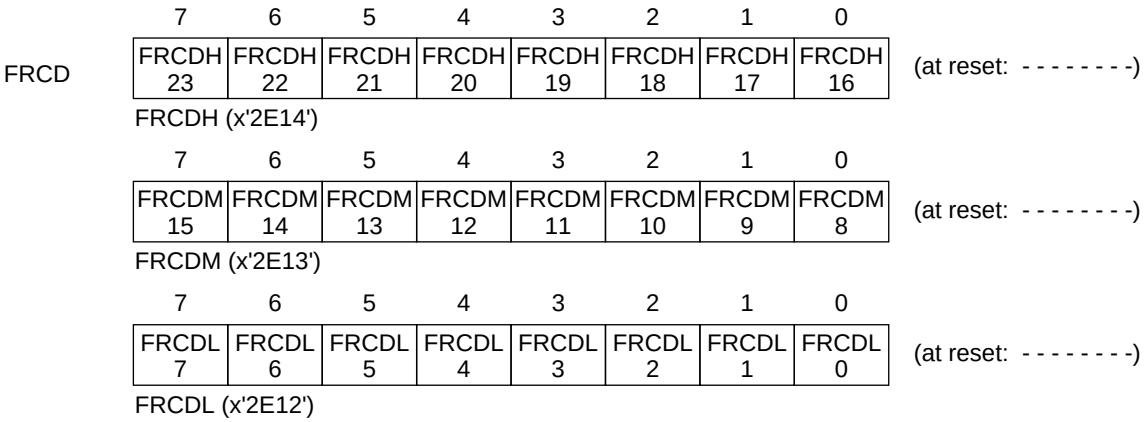
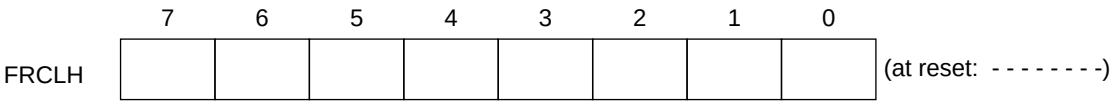


Figure 12-12 Free-Running Counter Data Register
(FRCD: x'2E14' to x'2E12', R/W)

Free-Running Counter Data Capture Control Register

By using a MOV instruction to read this register, FRC data is captured and input to FRCD.



Do not write to FRCLH. If FRCLH is written to, the value of FRCD is input to FRC.

Figure 12-13 Free-Running Counter Data Capture Control Register
(FRCLH: x'2E15', R)

Free-Running Counter Output Compare Register 0

Sets the desired timing (generates NF signal) to synchronize the 3rd through 18th bits of the Free-Running Counter FRC.

FOCR0

7	6	5	4	3	2	1	0	
FOCR0H 15	FOCR0H 14	FOCR0H 13	FOCR0H 12	FOCR0H 11	FOCR0H 10	FOCR0H 9	FOCR0H 8	(at reset: - - - - - - - -)

FOCR0H (x'2E17')

7	6	5	4	3	2	1	0	
FOCR0L 7	FOCR0L 6	FOCR0L 5	FOCR0L 4	FOCR0L 3	FOCR0L 2	FOCR0L 1	FOCR0L 0	(at reset: - - - - - - - -)

FOCR0L (x'2E16')

Figure 12-14 Free-Running Counter Output Compare Register 0
(FOCR0: x'2E17' to x'2E16', R/W)

Free-Running Counter Output Compare Register 1

Sets the desired timing (HOCRBUF0, HOCRBUF1) to synchronize the 3rd through 18th bits of the Free-Running Counter FRC.

FOCR1

7	6	5	4	3	2	1	0	
FOCR1H 15	FOCR1H 14	FOCR1H 13	FOCR1H 12	FOCR1H 11	FOCR1H 10	FOCR1H 9	FOCR1H 8	(at reset: - - - - - - - -)

FOCR1H (x'2E19')

7	6	5	4	3	2	1	0	
FOCR1L 7	FOCR1L 6	FOCR1L 5	FOCR1L 4	FOCR1L 3	FOCR1L 2	FOCR1L 1	FOCR1L 0	(at reset: - - - - - - - -)

FOCR1L (x'2E18')

Figure 12-15 Free-Running Counter Output Compare Register 1
(FOCR1: x'2E19' to x'2E18', R/W)

Free-Running Counter Output Compare Register 2

Sets the desired timing (generate recording CTL) to synchronize the 3rd through 18th bits of the Free-Running Counter FRC.

FOCR2

7	6	5	4	3	2	1	0	
FOCR2H 15	FOCR2H 14	FOCR2H 13	FOCR2H 12	FOCR2H 11	FOCR2H 10	FOCR2H 9	FOCR2H 8	(at reset: - - - - - - - -)

FOCR2H (x'2E1B')

7	6	5	4	3	2	1	0	
FOCR2L 7	FOCR2L 6	FOCR2L 5	FOCR2L 4	FOCR2L 3	FOCR2L 2	FOCR2L 1	FOCR2L 0	(at reset: - - - - - - - -)

FOCR2L (x'2E1A')

Figure 12-16 Free-Running Counter Output Compare Register 2
(FOCR2: x'2E1B' to x'2E1A', R/W)

SPG Synchronous Output Compare Register

Sets the desired timing (SPG synchronous output) to synchronize the 3rd through 18th bits of the Free-Running Counter FRC.

SPGTIM

7	6	5	4	3	2	1	0	
SPGTIMH 15	SPGTIMH 14	SPGTIMH 13	SPGTIMH 12	SPGTIMH 11	SPGTIMH 10	SPGTIMH 9	SPGTIMH 8	(at reset: - - - - - - - -)

SPGTIMH (x'2E1D')

7	6	5	4	3	2	1	0	
SPGTIML 7	SPGTIML 6	SPGTIML 5	SPGTIML 4	SPGTIML 3	SPGTIML 2	SPGTIML 1	SPGTIML 0	(at reset: - - - - - - - -)

SPGTIML (x'2E1C')

Figure 12-17 SPG Synchronous Output Compare Register
(SPGTIM: x'2E1D' to x'2E1C', R/W)

Task Timer Register

The counting operation is stopped after reset and when only one task has been started. The counting operation is started when multiple tasks have been started. The task switching operation is started when there is an underflow of the task timer register. The initial value for each task is loaded at task switching.

	7	6	5	4	3	2	1	0	
TSKBC	TSKBC 7	TSKBC 6	TSKBC 5	TSKBC 4	TSKBC 3	TSKBC 2	TSKBC 1	TSKBC 0	(at reset: - - - - - - - -)

Set value	Switching interval (µSec)	
	fosc=14.32MHz	fosc=17.72MHz
00	2.23	1.81
01	4.47	3.61
⋮	⋮	⋮
n	2.23 (n+1)	1.81 (n+1)
⋮	⋮	⋮
FF	572.1	462.3

Count clock = fs/16, TSKBCSEL = 0

When multiple tasks are to be started, set the initial value in the task timer register (TSKBC) at the time of a reset start.

Figure 12-18 Task Timer Register
(TSKBC: x'2EB0', R/W)

RAM for Saving System Registers of Tasks

This RAM is for saving the system registers for different tasks. (For system use)
The contents of the RAM for saving system registers of task after reset are undefined.

TSKnSP	For saving register SP	(x'2EC0'+16n)
TSKnDW0	For saving register DW0	(x'2EC2'+16n)
TSKnDW1	For saving register DW1	(x'2EC4'+16n)
TSKnA0	For saving register A0	(x'2EC6'+16n)
TSKnA1	For saving register A1	(x'2EC8'+16n)
TSKnPSW	For saving register PSW	(x'2ECA'+16n)
TSKnTT	For saving register TT	(x'2ECB'+16n)
TSKnHA	For saving register HA	(x'2ECC'+16n)

(n=0 to 3)

When multiple tasks are to be started, set the initial value in the task timer (TSKnTT) at the time of a reset start.

Figure 12-19 RAM for Saving System Registers of Tasks
(Task 0: x'2EC0' to x'2ECD')
(Task 1: x'2ED0' to x'2EDD')
(Task 2: x'2EE0' to x'2EED')
(Task 3: x'2EF0' to x'2EFD')

CPU Mode Control Register

	7	6	5	4	3	2	1	0	
CPUM	0	OSC SEL1	OSC SEL0	0	STOP	HALT	OSC1	OSC0	(at reset: 0 1 1 0 0 0 0 0)
				(note 1)					

CPU Mode Settings

SXI pin input	Operating mode	OSCSEL1	OSCSEL0	STOP	HALT	OSC1	OSC0	State: OSC1/OSC2 pins (note 5)	State: XI/XO pins (note 6)	State: System clock (fs)	State: CPU	Voltage (V) at which operation is guaranteed (note 2)	Comments
When held at a low level (no XI)	STOP0 (Note 7)	0	0	1	0	0	0	Stop	Stop	—	Stop	2.2 to 5.5	Upon return, waits for stabilization of OSC oscillation
	HALT0	*	*	0	1	0	0	Oscillate	Stop	—	Stop	4.0 to 5.5 (4.5 to 5.5)	
	NORMAL	0	0	0	0	0	0	Oscillate	Stop	fosc/2	NORMAL operation	4.0 to 5.5 (4.5 to 5.5)	(Note 8)
		0	1							fosc/8			
		1	0							fosc/32			
		1	1							fosc/1024		2.2 to 5.5	When reset is released (note 3)
When held at a high level (XI present)	STOP0 (Note 7)	0	0	1	0	0	0	Stop	Stop	—	Stop	2.2 to 5.5	Upon return, waits for stabilization of OSC oscillation
	HALT0	*	*	0	1	0	0	Oscillate	Oscillate	—	Stop	4.0 to 5.5 (4.5 to 5.5)	
	NORMAL	0	0	0	0	0	0	Oscillate	Oscillate	fosc/2	NORMAL operation	4.0 to 5.5 (4.5 to 5.5)	(Note 8)
		0	1							fosc/8			
		1	0							fosc/32			
		1	1							fosc/1024		2.2 to 5.5	When reset is released (note 3)
	IDLE	0	0	0	0	0	1	Oscillate	Oscillate	fxi/2	SLOW operation	2.2 to 5.5	Stabilizes fosc during transition from SLOW mode to NORMAL mode
		0	1							fxi/8			
		1	*							fxi/32			
	SLOW	0	0	0	0	1	1	Stop	Oscillate	fxi/2	SLOW operation	2.2 to 5.5	(Note 9)
		0	1							fxi/8			
		1	*							fxi/32			
	HALT1	*	*	0	1	1	1	Stop	Oscillate	—	Stop	2.2 to 5.5	
	STOP1 (Note 7)	0	0	1	0	1	1	Stop	Stop	—	Stop	2.2 to 5.5	Upon return, waits for stabilization of xi oscillation

Figure 12-20 CPU Mode Control Register (CPUM: x'3F00', R/W)

Note 1: This bit (bp 4) must be set to '0'.

Note 2: The value of voltage at which operation is guaranteed is given under the conditions of fosc=14.32 MHz (17.72 MHz) and fxi=32.768 kHz.

Note 3: When reset is released, after waiting for stabilization of fosc oscillation (262,144 counts of fosc), the system will start in NORMAL mode and the system clock (fs) is fosc divided by 1024.

Note 4: The system clock can verified by reading the status monitor, OSC0 flag (bp0) of the CPUM register (x'3F00').

Note 5: While the reset input pin (NRST) is at a low level, oscillation of OSC is halted.

Note 6: While the reset input pin (NRST) is at a low level, the XI pin oscillates.

Note 7: Stop the WD timer and clear the WD counter before changing over to the STOP mode.

Note 8: It is not possible to change over from the fs=fosc/2 NORMAL mode to the STOP0 mode or the HALT0 mode.

Note 9: It is not possible to change over from the fs=fxi/2 SLOW mode to the STOP1 mode or the HALT1 mode.

Memory Control Register

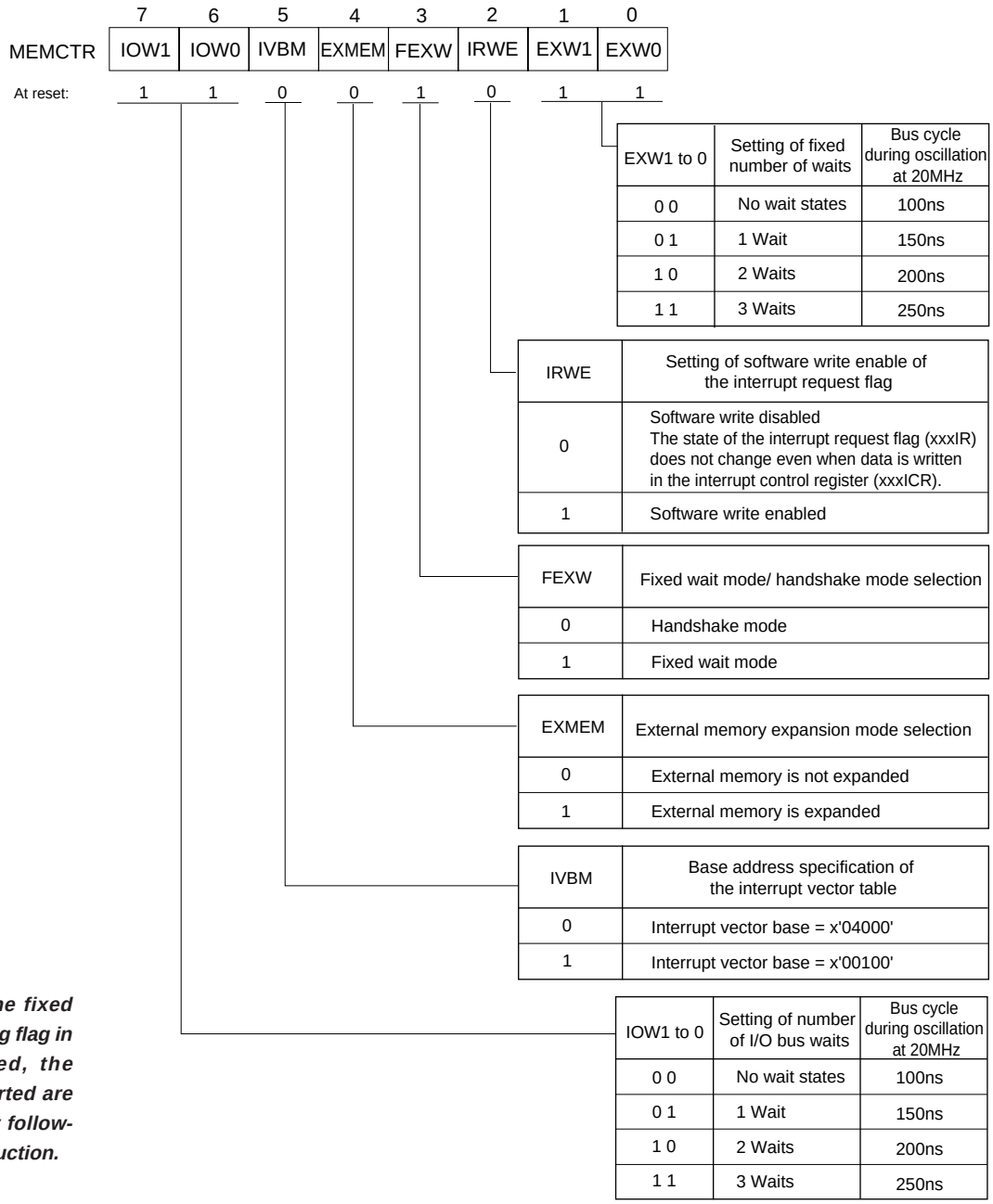


Figure 12-21 Memory Control Register (MEMCTR: x'3F01', R/W)

- The wait target space of IOW1, 0 is the special register space of x'3F00' ~ x'3FFF'.
- The wait target space of EXW1, 0 is the externally connected devices in the processor mode and memory expansion mode.

When the value of the fixed number of waits setting flag in MEMCTR is changed, the number of waits inserted are changed immediately following the rewriting instruction.

Limit writing '1' in IRWE to the initialization program, etc. After manipulating the interrupt control register by making IRWE '1', make sure to return it to '0'. Keeping IRWE in the '1' state can cause interrupt requests to be missed.

Task Start Control Register (TSKCTR)

It is not possible to stop all tasks. When TSK1EN to TSK3EN are all '0', TSK0EN flag are set to '1'.

	7	6	5	4	3	2	1	0
TSKCTR	TSKBCEN	TSKBCSEL	——	——	TSK3EN	TSK2EN	TSK1EN	TSK0EN
At reset:	1	0	—	—	0	0	0	1

TSK0EN	Task 0 start/stop control
0	Stopped
1	Started

TSK1EN	Task 1 start/stop control
0	Stopped
1	Started

TSK2EN	Task 2 start/stop control
0	Stopped
1	Started

TSK3EN	Task 3 start/stop control
0	Stopped
1	Started

TSKBCSEL	Task timer count clock selection
0	fs/16
1	fs/32

TSKBCEN	Task timer start/stop control
0	Stopped
1	Started

- Valid only when multiple tasks have been started.
- Set at the time of task switching.

Figure 12-22 Task Start Control Register (TSKCTR: x'3F05', R/W)

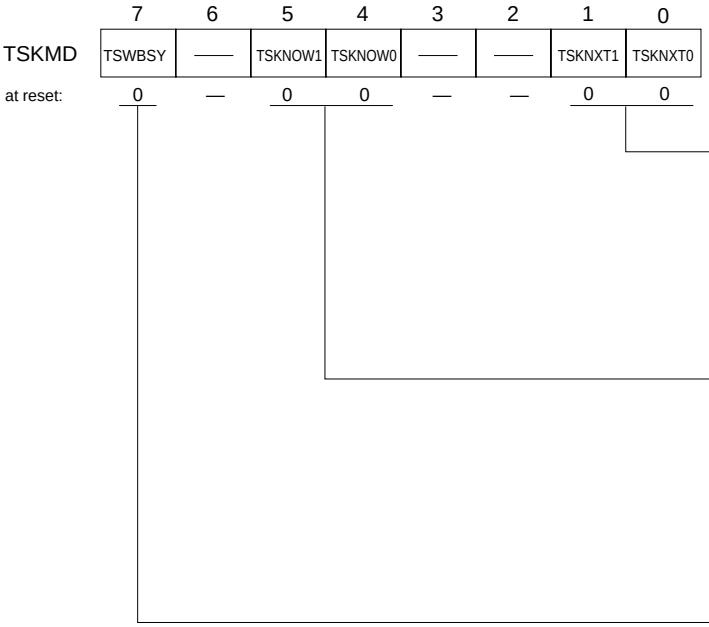
Write the program as follows for stopping the counting of the task timer register (TSKBC).

```
check  BCLR   (TSKCTR)TSKBCEN    ;Stopping the task timer.
       NOP
       TBNZ   (TSKCTR)TSKBCEN    ;Verifying that the task timer has stopped.
```

To manipulate the TSKBCEN flag, always use the BSET instruction or the BCLR instruction.

Task Number Control Register (TSKMD)

When some values are written in the flags TSKNXT1 ~ TSKNXT0, switching to the specified task is done without waiting for an underflow of register TSKBC.



TSKNXT1	TSKNXT0	Number of the task to be executed next
0	0	Task 0
0	1	Task 1
1	0	Task 2
1	1	Task 3

Write the two bits simultaneously.

TSKNOW1	TSKNOW0	Number of the task currently being executed
0	0	Task 0
0	1	Task 1
1	0	Task 2
1	1	Task 3

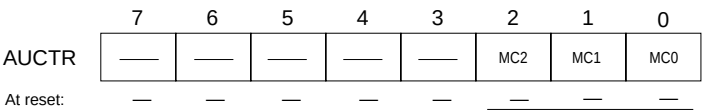
These are read only flags.

TSWBSY	Register saving status
0	Saving completed
1	Saving in progress

This is a write only flag. When this flag is set all other bits, i. e., bits 0-6, will be ignored.

Figure 12-23 Task Number Control Register (TSKMD: x'3F06', R/W)

Extended Computation Control Register (AUCTR)



MC2 to 0	Macro instruction (Write Only)
001	Execution of MULWU instruction
010	Execution of MULW instruction
100	Execution of DIVWU instruction

Note 1: This is a write-only register whose value is always indefinite when read out.
Note 2: Do not write to this register under normal conditions.

Figure 12-23-2 Extended Computation Control Register (AUCTR: x'3F07', W)

ROM Correction Control Register (RCCTR)

The ROM correction function is started when the RCnEN flag is set.
An indirect branch is made to the address in the RAM set in the RCnV table when there is a match between the corresponding register (correction address setting register) and ROM address. Set the RCnEN flag after setting the correction address setting register. (n=0,1)

Do not set the same address in several RCnAPH/M/L registers.
The following priority order is followed when the same address is specified in several registers.
RCnAPH → RCnAPM → RCnAPL

	7	6	5	4	3	2	1	0
RCCTR	—	—	—	—	—	0	RC1EN	RC0EN
At reset:	—	—	—	—	—	0	0	0

RC0EN	ROM Correction control of first address
0	Disabled
1	Enabled

RC1EN	ROM Correction control of second address
0	Disabled
1	Enabled

Figure 12-24 ROM Correction Control Register (RCCTR: x'3F09', R/W)

Bank Register for Source Address

Register BMEMSR is for specifying the bank for a load instruction from memory to register. When a bank is specified in this register, the bank control are effective for all addressing modes other than I/O short instructions and SP relative indirect instructions.

The bit operation instructions (BSET, BCLR, BTST) refer to the value in register BMEMSR.

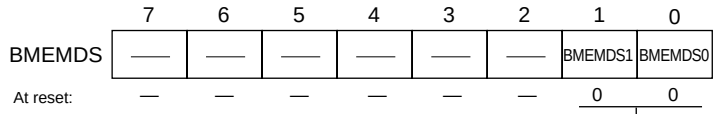
	7	6	5	4	3	2	1	0
BMEMSR	—	—	—	—	—	—	BMEMSR1	BMEMSR0
At reset:	—	—	—	—	—	—	0	0

BMEMSR1	BMEMSR0	Bank selection for source address
0	0	Bank 0
0	1	Bank 1 (setting prohibited)
1	0	Bank 2 (setting prohibited)
1	1	Bank 3 (setting prohibited)

Figure 12-25 Bank Register for Source Address (BMEMSR: x'3F0A', R/W)

Bank Register for Destination Address

Register BMEMDS is for specifying the bank for a store instruction from register to memory. When a bank is specified in this register, the bank control are effective for all addressing modes other than I/O short instructions, SP relative indirect instructions, and bit operation instructions.

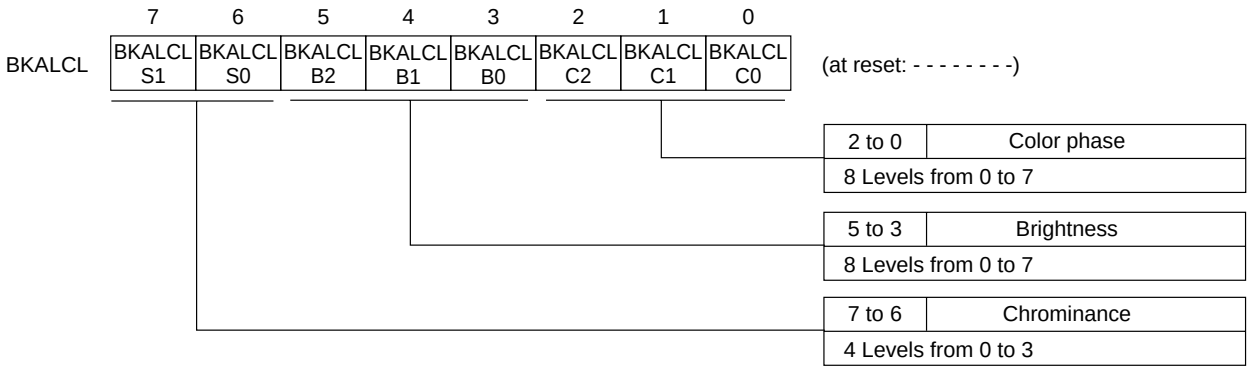


BMEMDS1	BMEMDS0	Bank selection for destination address
0	0	Bank 0
0	1	Bank 1 (setting prohibited)
1	0	Bank 2 (setting prohibited)
1	1	Bank 3 (setting prohibited)

The bank selection function is not supported in the MN101D02 Series. Therefore, make sure to set registers BMEMSR and BMEMDS to x'00'.

Figure 12-26 Bank Register for Destination Address (BMEMDS: x'3F0B', R/W)

Screen Background Color Control Register



Note 1: The part of the screen that is the screen background color is the area other than the character display area and the through area. The background color of the character display area (including spaces) other than through codes are the line background color.

Note 2: The following settings are necessary for enabling the screen background color.

BKCLON	(OSDCNT3	x'3FB9' bp2)=1
SYNMOD	(OSDCNT1	x'3FC8' bp4)=0
BKMONO	(VLIN	x'3FDB' bp5)=0
BAKSEL	(VLIN	x'3FDB' bp6)=0

Note 3: Depending on the combination of the line background color and the screen background color, there can be color smear in the boundary areas.

Figure 12-27 Screen Background Color Control Register (BKALCL: x'3F0E', R/W)

ROM Correction Address Setting Registers (RCnAPL, RCnAPM, RCnAPH)

The address storing the leading address of the instruction requiring correction is set in these registers. An indirect branch is made to the address set in the RCnV table when there is a match between the value set in this register and the execution address of the instruction. To enable the ROM correction function, first set the desired address in this register and then set the RCnEN flag of register RCCTR. (n=0,1)

(1) ROM Correction Address Setting Register (RCnAPL) (lower 8 bits)

	7	6	5	4	3	2	1	0	
RCnAPL	RCnAPL7	RCnAPL6	RCnAPL5	RCnAPL4	RCnAPL3	RCnAPL2	RCnAPL1	RCnAPL0	(Indefinite at reset)

Figure 12-28 ROM Correction address setting register (lower 8 bits)

(2) ROM Correction Address Setting Register (RCnAPM) (middle 8 bits)

	7	6	5	4	3	2	1	0	
RCnAPM	RCnAPM7	RCnAPM6	RCnAPM5	RCnAPM4	RCnAPM3	RCnAPM2	RCnAPM1	RCnAPM0	(Indefinite at reset)

Figure 12-29 ROM Correction address setting register (middle 8 bits)

(3) ROM Correction Address Setting Register (RCnAPH) (upper 8 bits)

	7	6	5	4	3	2	1	0	
RCnAPH	RCnAPH7	RCnAPH6	RCnAPH5	RCnAPH4	RCnAPH3	RCnAPH2	RCnAPH1	RCnAPH0	(Indefinite at reset)

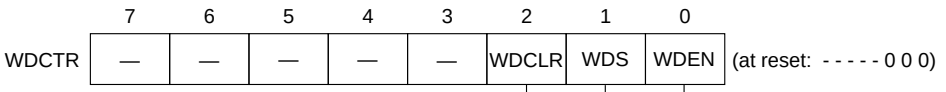
Figure 12-30 ROM Correction address setting register (upper 8 bits)

The relationships between the ROM correction address setting register, RCnV table, and ROM correction control register are summarized in the following table.

Table 12-1 Correspondence with the ROM Correction Address Setting Register

No.	ROM Correction address setting register			Address of RC vector table	ROM Correction control flag
	Register name abbreviation	RAM Address	R/W		
0	RC0APL	X'3F10'	R/W	X'0010' to X'0011'	RC0EN
	RC0APM	X'3F11'	R/W		
	RC0APH	X'3F12'	R/W		
1	RC1APL	X'3F13'	R/W	X'0012' to X'0013'	RC1EN
	RC1APM	X'3F14'	R/W		
	RC1APH	X'3F15'	R/W		

Watchdog Timer Control Register



WDEN		
0	WDS=0	Halt watchdog timer, reset upper 2 bits of 2^{16} counter
	WDS=1	Halt watchdog timer, reset upper 5 bits of 2^{19} counter
1	Operate watchdog timer	

Upper 2 bits of the watchdog timer

Note 1)

WDS	Watchdog interval setting		
Micro-computer mode		WD interval	
		WDS=0 (Twd=2^{16}/fs)	WDS=1 (Twd=2^{19}/fs)
fs=fosc/1024	NORMAL	4.68sec	37.5sec
fs=fosc/32	NORMAL	146msec	1.17sec
fs=fosc/8	NORMAL	36.6msec	292msec
fs=fosc/2	NORMAL	9.1msec	73.2msec
fs=fxi/32	SLOW	64sec	512sec
fs=fxi/8	SLOW	16sec	128sec
fs=fxi/2	SLOW	4sec	32sec

Note 2)

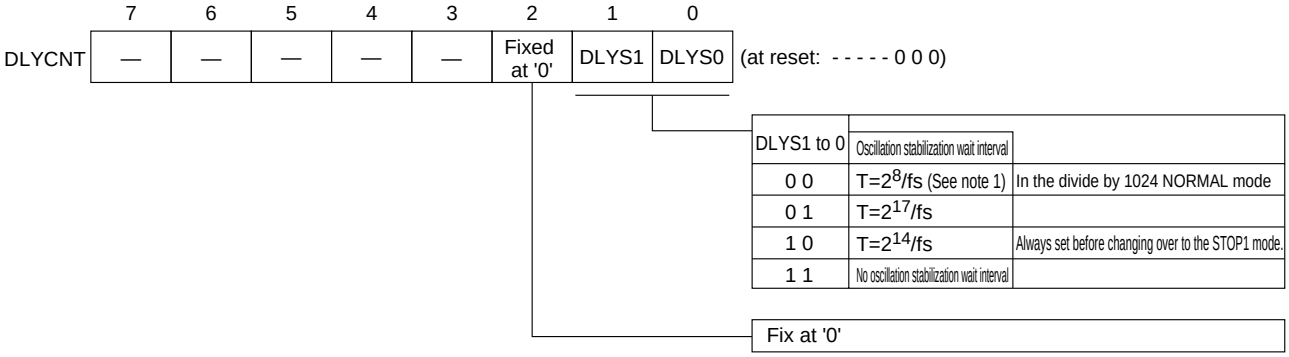
WDCLR	Clear watchdog counter
0	Normal
1	Clear the 19-bit watchdog counter

Note1: fs is the system clock.

Note2: If the WDCLR Flag is set to '1', the 19-bit WD Counter will be cleared.

Figure 12-31 Watchdog Timer Control Register (WDCTR: x'3F1D', R/W)

Oscillation Stabilization Wait Control Register



The oscillation stabilization waiting period at the time of release from reset is T=2^{18}/ fosc = 18.3 ms.

Figure 12-32 Oscillation Stabilization Wait Control Register
(DLYCTR: x'3F1E', R/W)

Port 6 Input Level Control Register

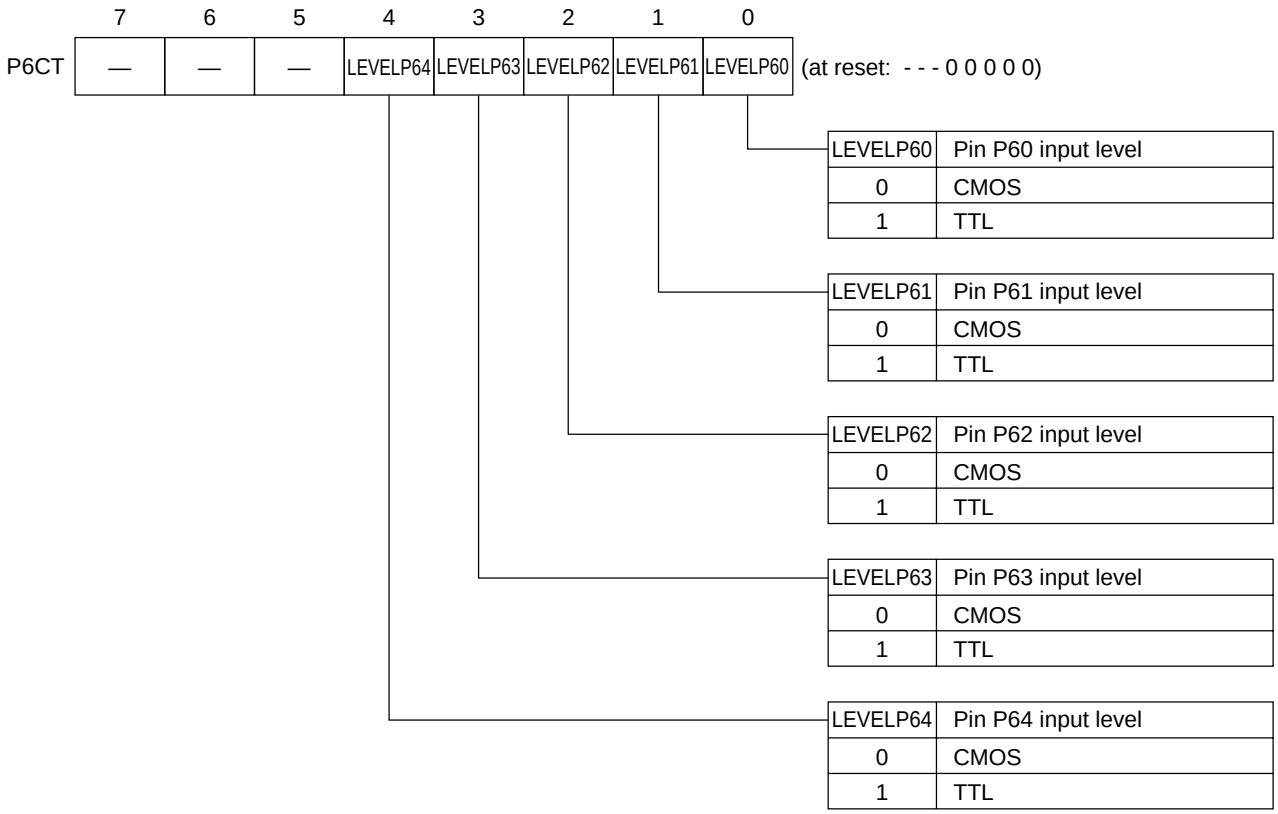


Figure 12-33 Port 6 Input Level Control Register (P6CT: x'3F1F', R/W)

Port 0 Data Output Register

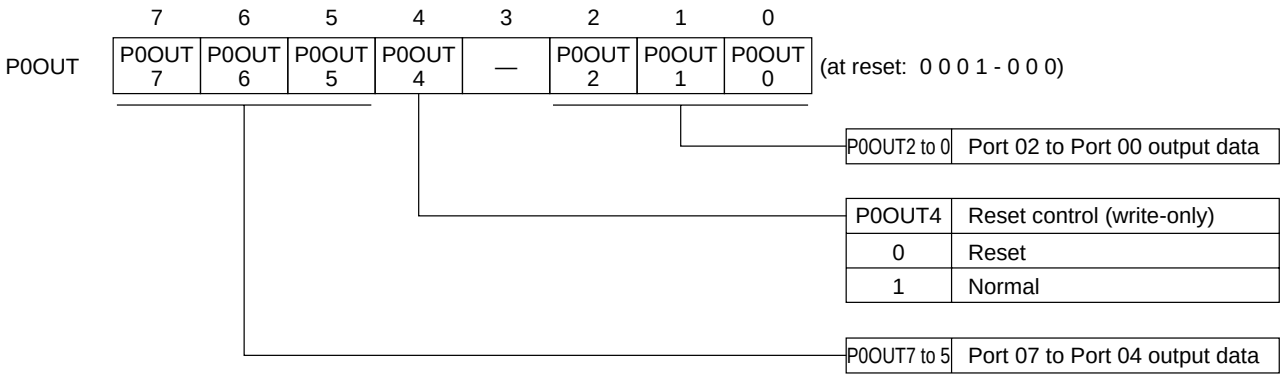


Figure 12-34 Port 0 Data Output Register (P0OUT: x'3F20', R/W one section is write-only)

Port 0 Data Input Register

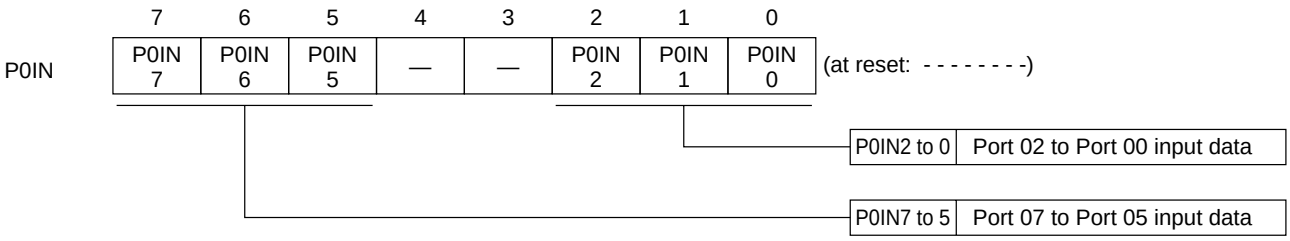


Figure 12-35 Port 0 Data Input Register (P0IN: x'3F21', R)

Port 1 Data (Synchronous Output Buffer) Register

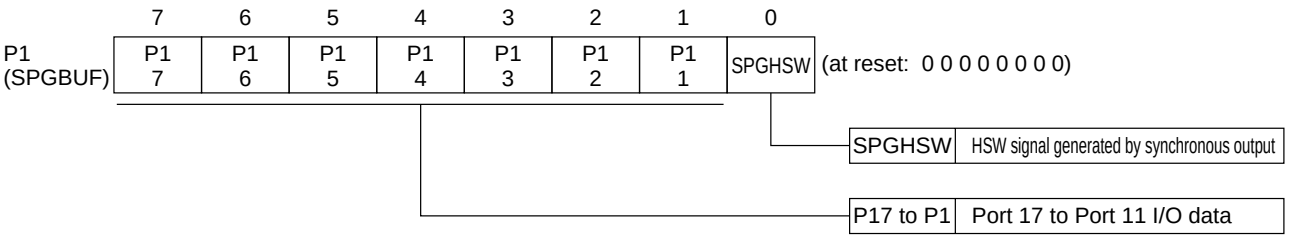


Figure 12-36 Port 1 Data (Synchronous Output Buffer) Register
(P1 [SPGBUF]: x'3F22', R/W)

Port 2 Data Register

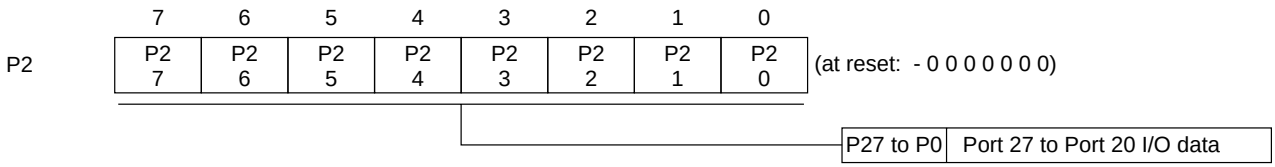


Figure 12-37 Port 2 Data Register (P2: x'3F23', R/W)

Port 4 Data (HSW Synchronous Output Buffer 0) Register

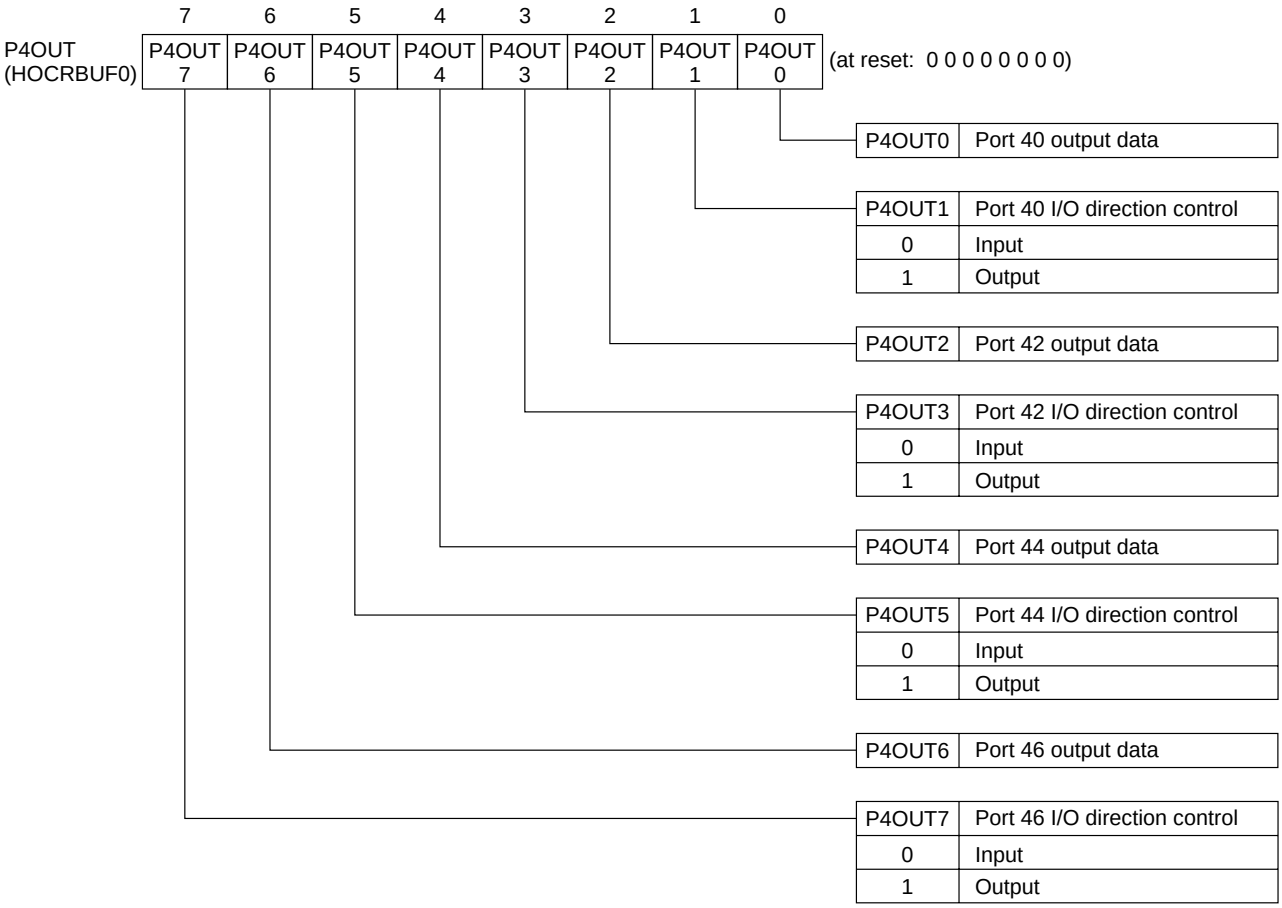


Figure 12-38 Port 1 Data (HSW Synchronous Output Buffer 0) Register
(P4OUT [HOCRBUF0]: x'3F24', R/W)

Port 4 Data Input Register

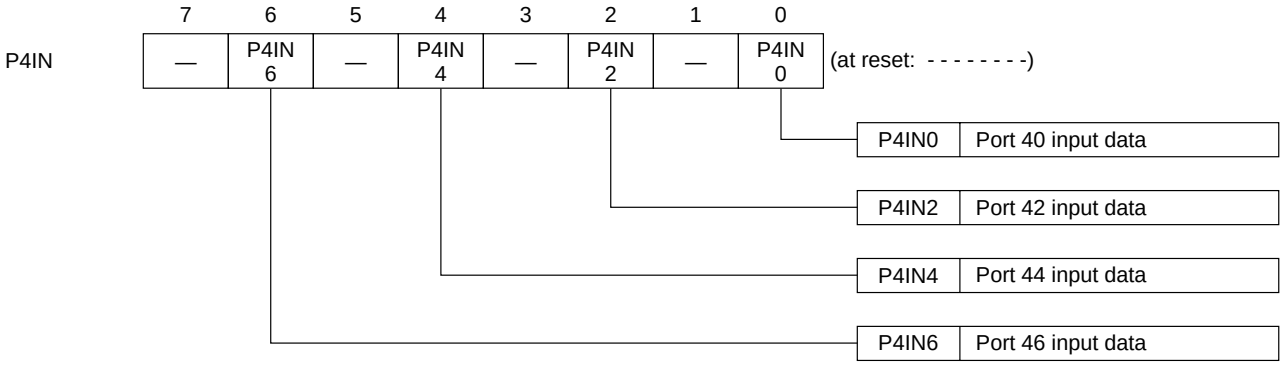


Figure 12-39 Port 4 Data Input Register (P4IN: x'3F25', R)

Port 5 Data Register

	7	6	5	4	3	2	1	0	
P5	—	—	—	P54	P53	P52	P51	P50	(at reset: - - - 0 0 0 0 0)

Figure 12-40 Port 5 Data Register (P5: x'3F26', R/W)

Port 6 Data Register

	7	6	5	4	3	2	1	0	
P6	P67	P66	P65	P64	P63	P62	P61	P60	(at reset: 0 0 0 0 0 0 0 0)

Figure 12-41 Port 6 Data Register (P6: x'3F27', R/W, one section is read only)

Port 7 Data Register

	7	6	5	4	3	2	1	0	
P7	P77	P76	P75	P74	P73	P72	P71	P70	(at reset: 0 0 0 0 0 0 0 0)

Figure 12-42 Port 7 Data Register (P7: x'3F28', R/W)

Port 8 Data Register

	7	6	5	4	3	2	1	0	
P8	P87	P86	P85	P84	P83	P82	P81	P80	(at reset: 0 0 0 0 0 0 0 0)

Figure 12-43 Port 8 Data Register (P8: x'3F29', R/W)

Port 9 Data Register

	7	6	5	4	3	2	1	0	
P9	—	—	—	—		P92	P91	P90	(at reset: - - - - - - - -)

Figure 12-44 Port 9 Data Register (P9: x'3F2A', R)

Port A Data Register

	7	6	5	4	3	2	1	0	
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	(at reset: 0 0 0 0 0 0 0 0)

Figure 12-45 Port A Data Register (PA: x'3F2B', R/W)

Port B Data Register

	7	6	5	4	3	2	1	0	
PB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	(at reset: 0 0 0 0 0 0 0 0)

Figure 12-46 Port B Data Register (PB: x'3F2C', R/W)

Port C Data Register

	7	6	5	4	3	2	1	0	
PC	—	—	—	—	PC3	PC2	PC1	PC0	(at reset: - - - - 0 0 0 0)

Figure 12-47 Port C Data Register (PC: x'3F2D', R/W)

Port 0 Direction Control Register

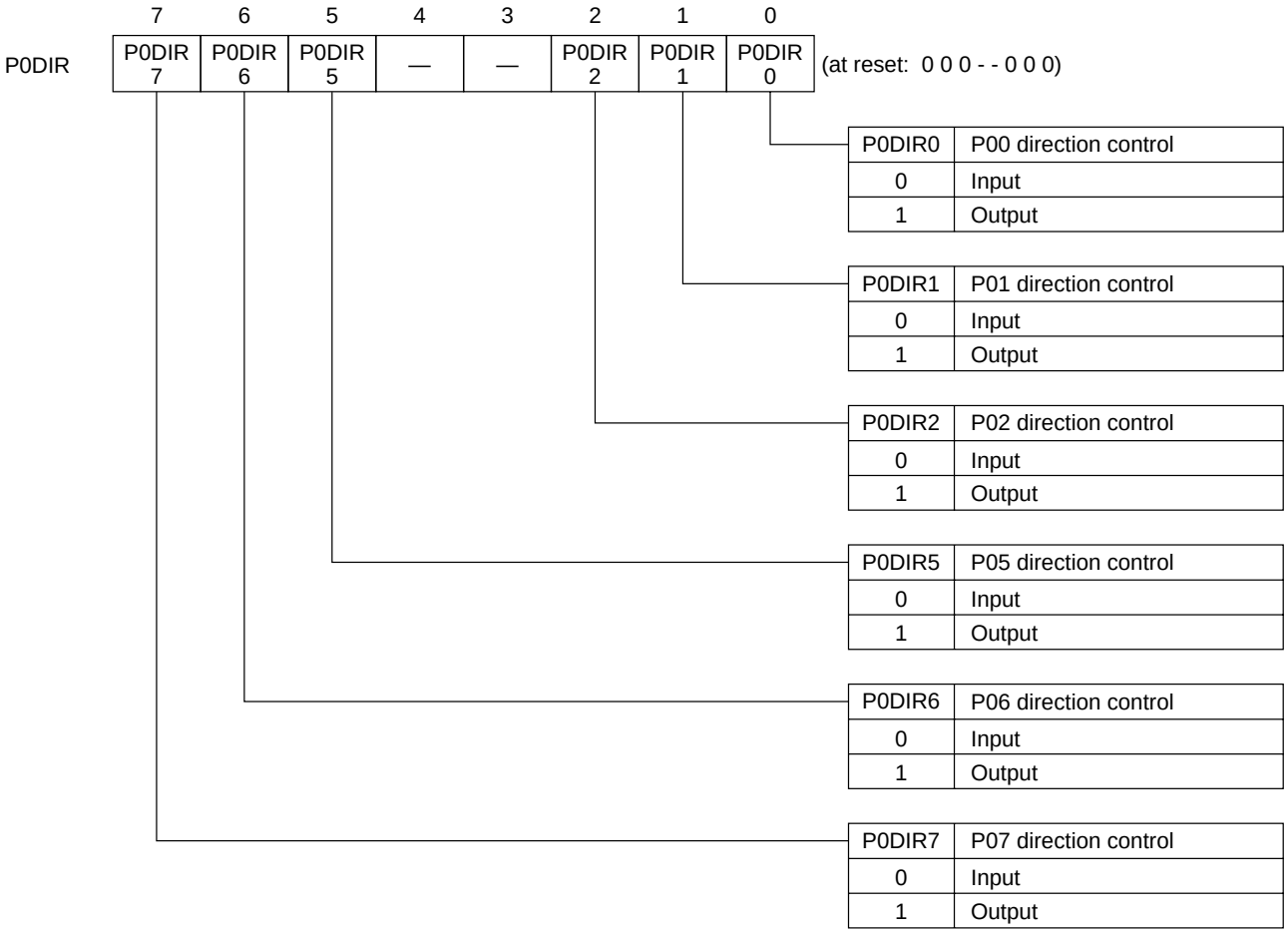


Figure 12-48 Port 0 Direction Control Register (P0DIR: x'3F2E', R/W)

Port 1 (SPG) Direction Control Register

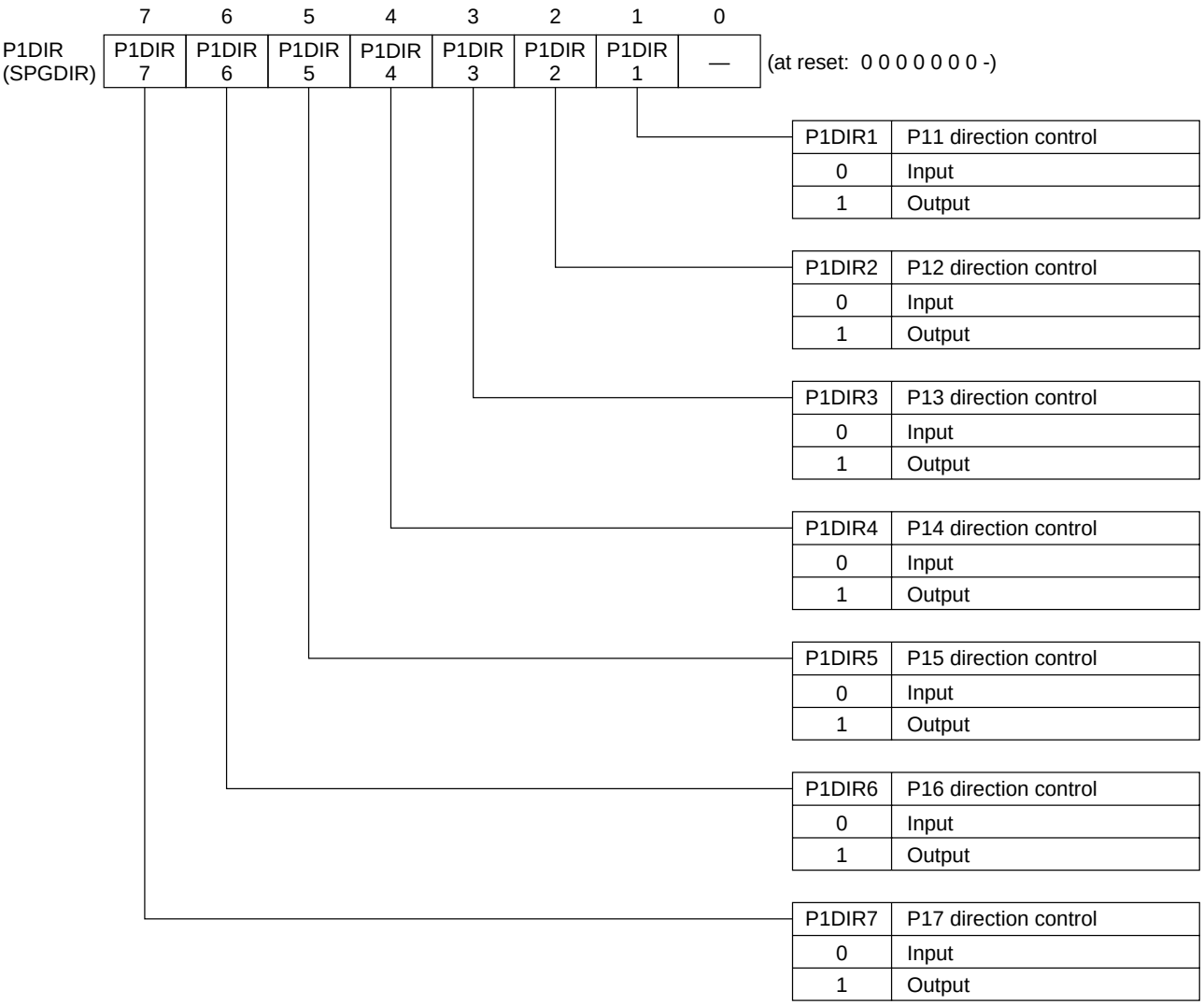


Figure 12-49 Port 1 (SPG) Direction Control Register
(P1DIR [SPGDIR]: x'3F2F', R/W)

Port 2 Direction Control Register

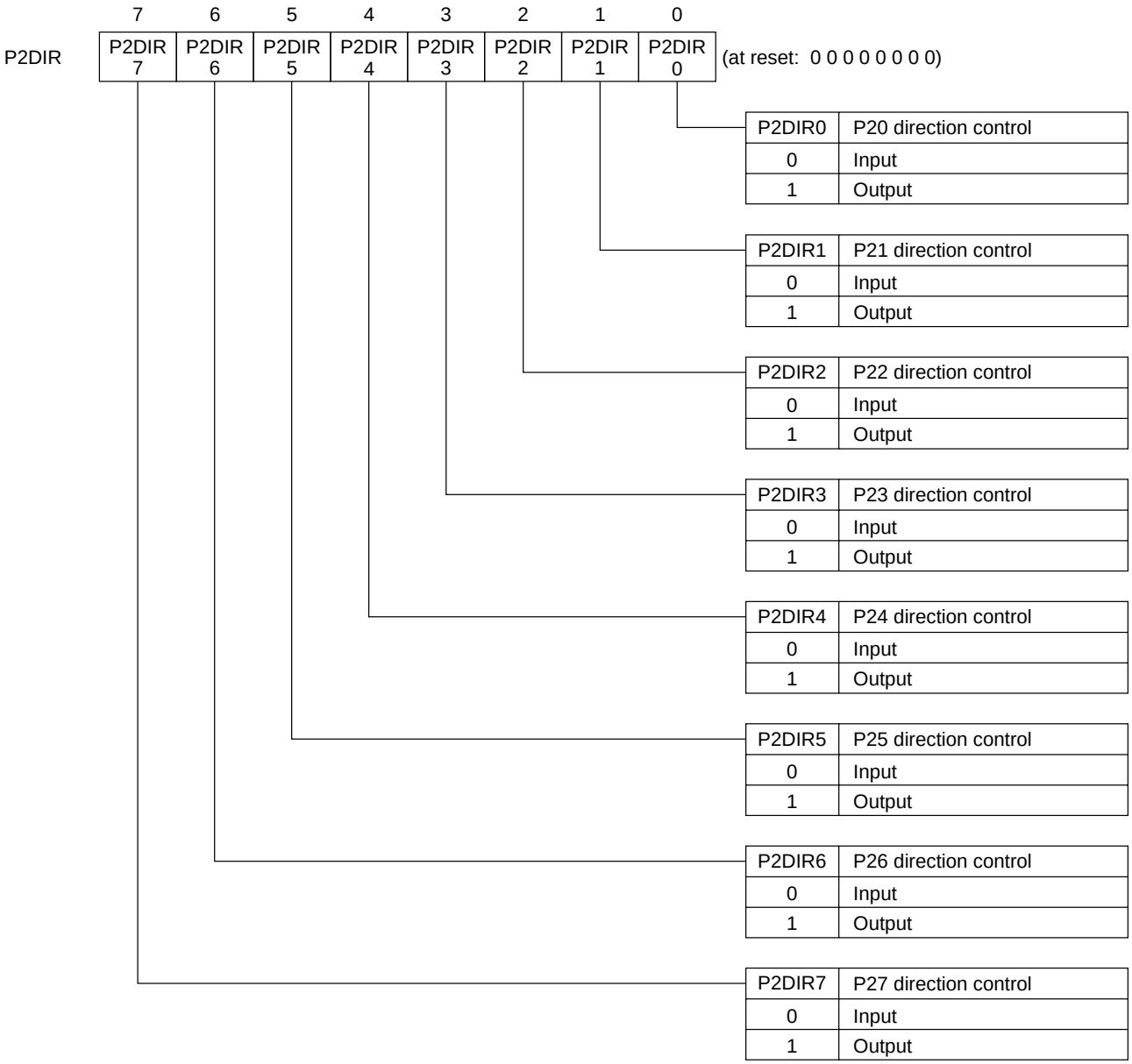


Figure 12-50 Port 2 Direction Control Register (P2DIR: x'3F30', R/W)

Port 5 Direction Control Register

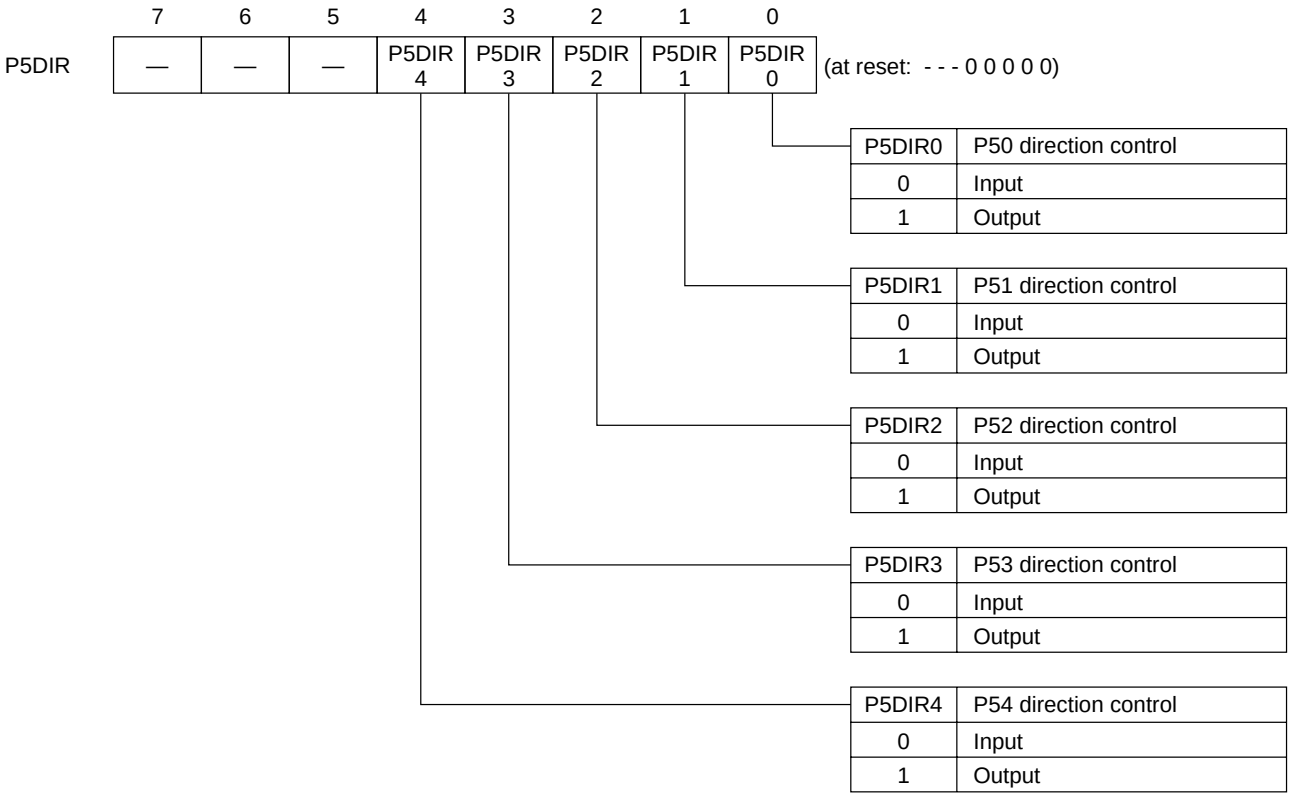


Figure 12-51 Port 5 Direction Control Register (P5DIR: x'3F31', R/W)

Port 6 Direction Control Register

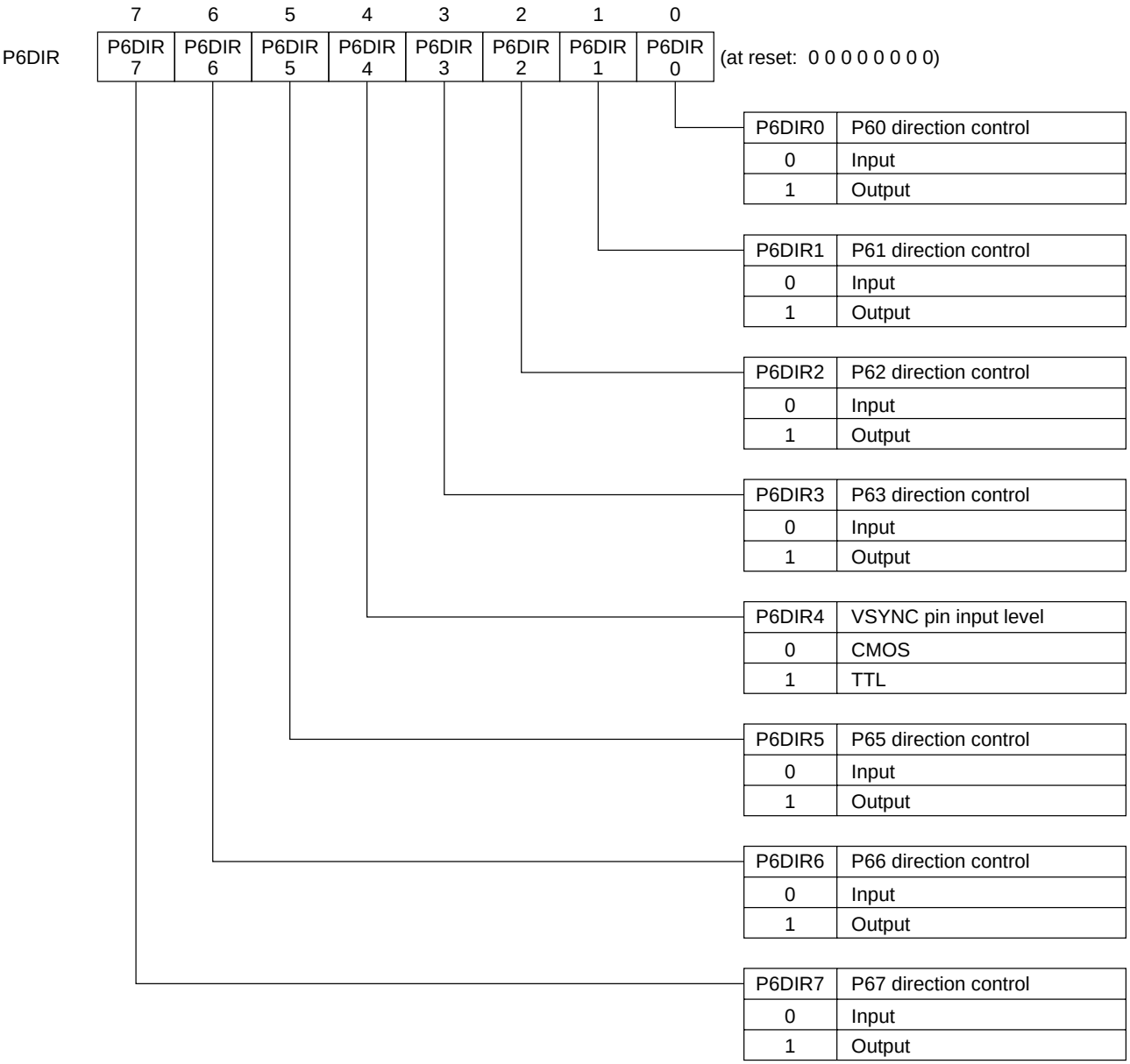


Figure 12-52 Port 6 Direction Control Register (P6DIR: x'3F32', R/W)

Port 7 Direction Control Register

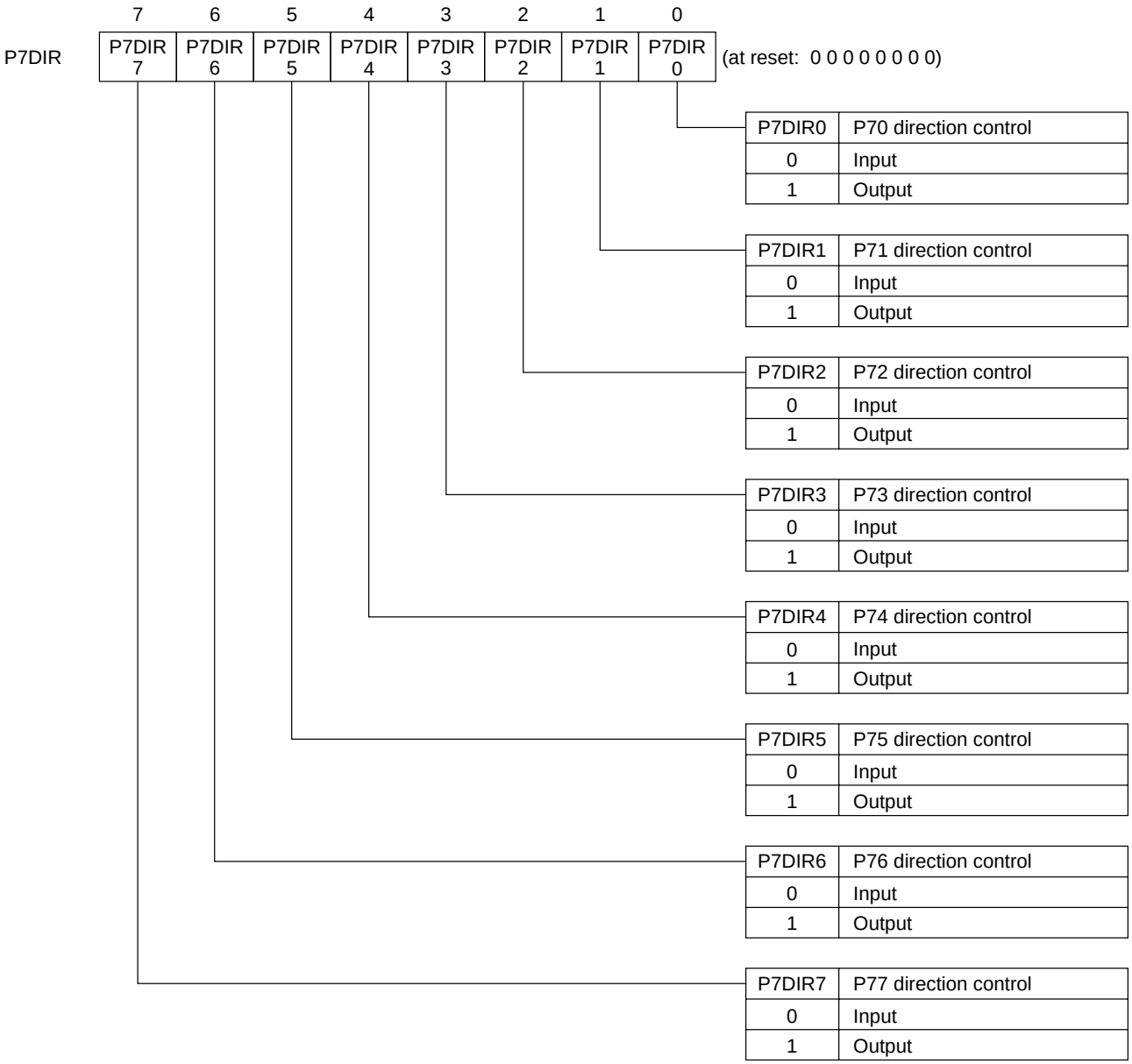


Figure 12-53 Port 7 Direction Control Register (P7DIR: x'3F33', R/W)

Port 8 Direction Control Register

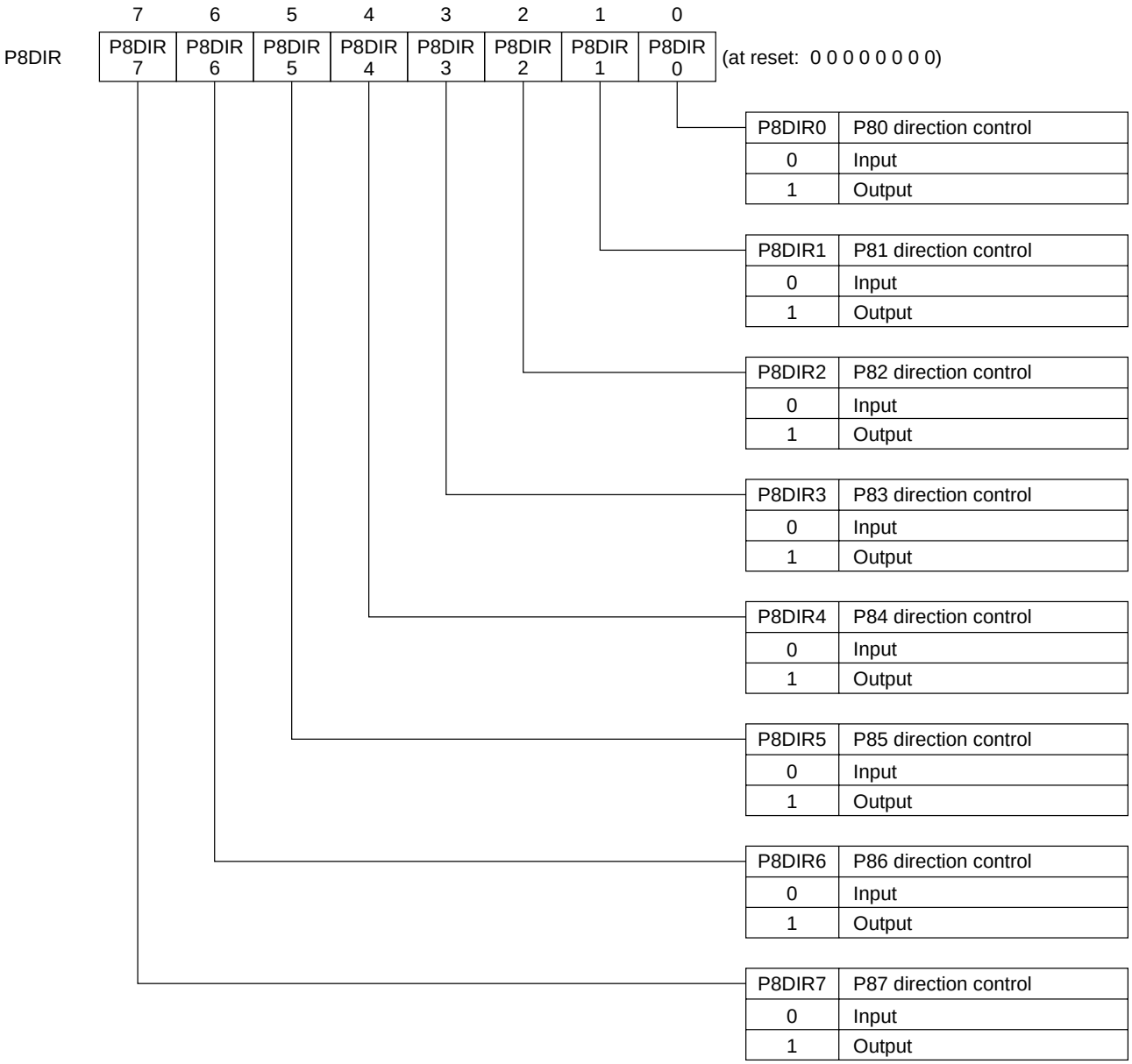


Figure 12-54 Port 8 Direction Control Register (P8DIR: x'3F34', R/W)

Port A Direction Control Register

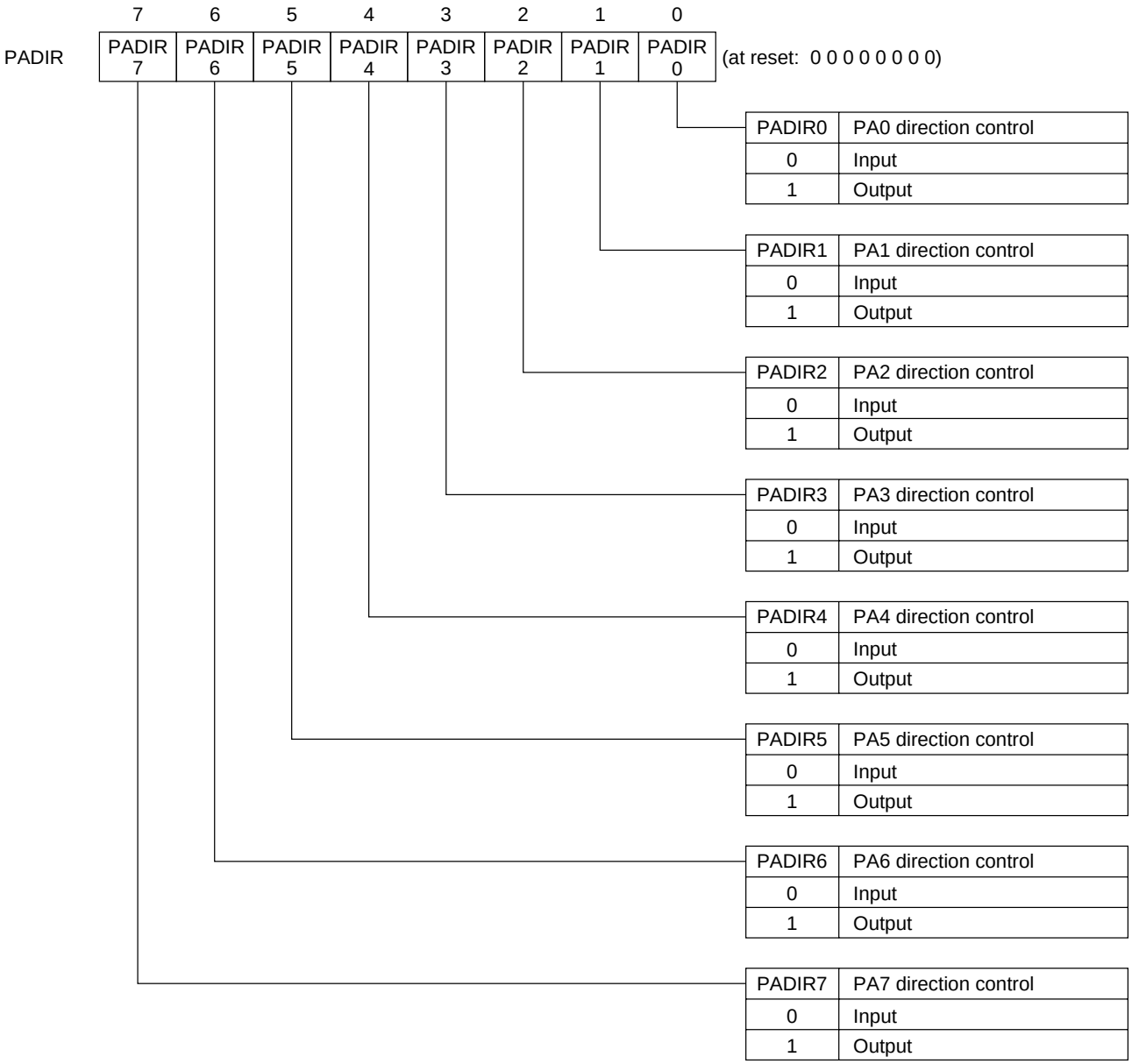


Figure 12-55 Port A Direction Control Register (PADIR: x'3F35', R/W)

Port B Direction Control Register

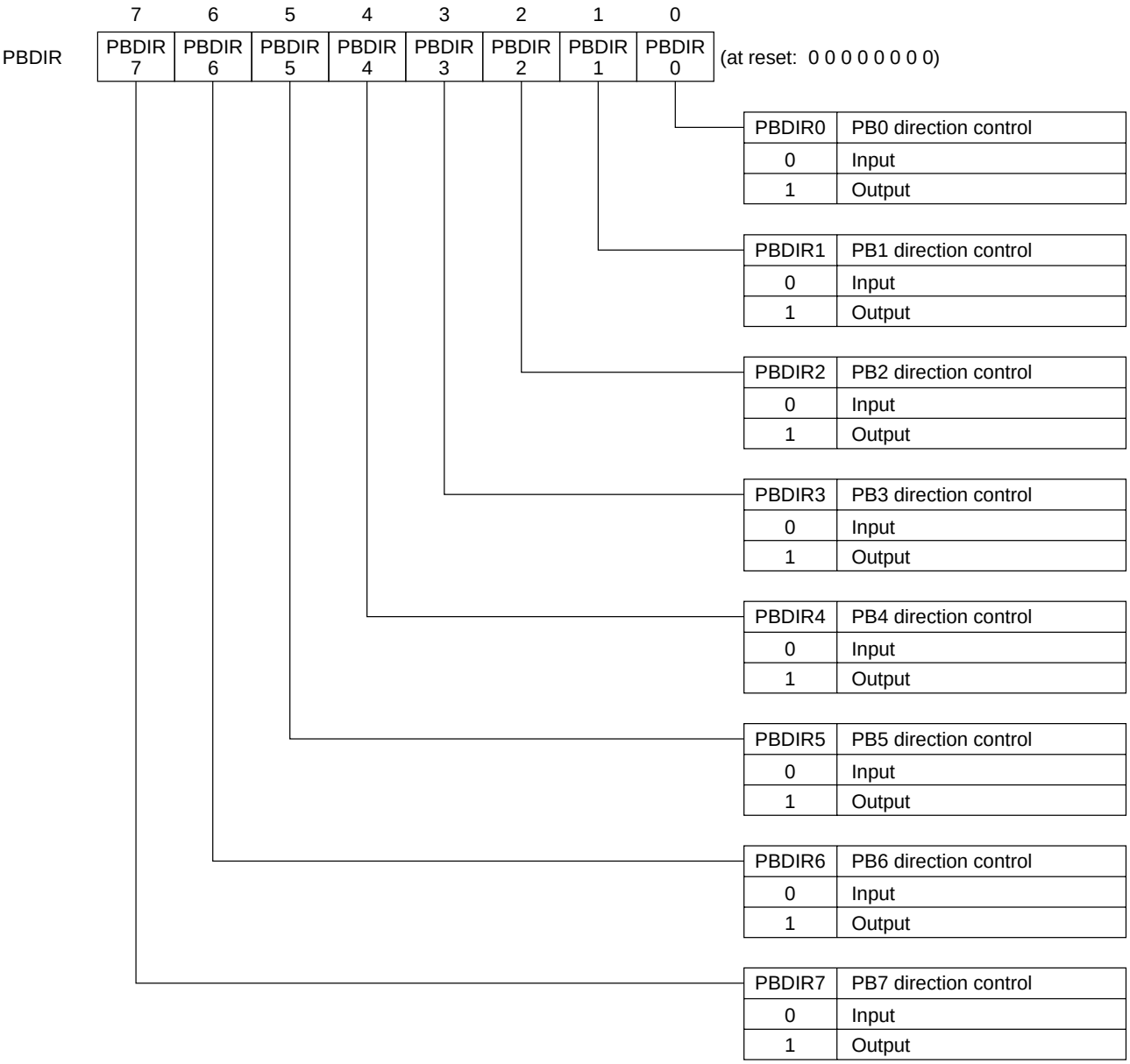


Figure 12-56 Port B Direction Control Register (PBDIR: x'3F36', R/W)

Port C Direction Control Register

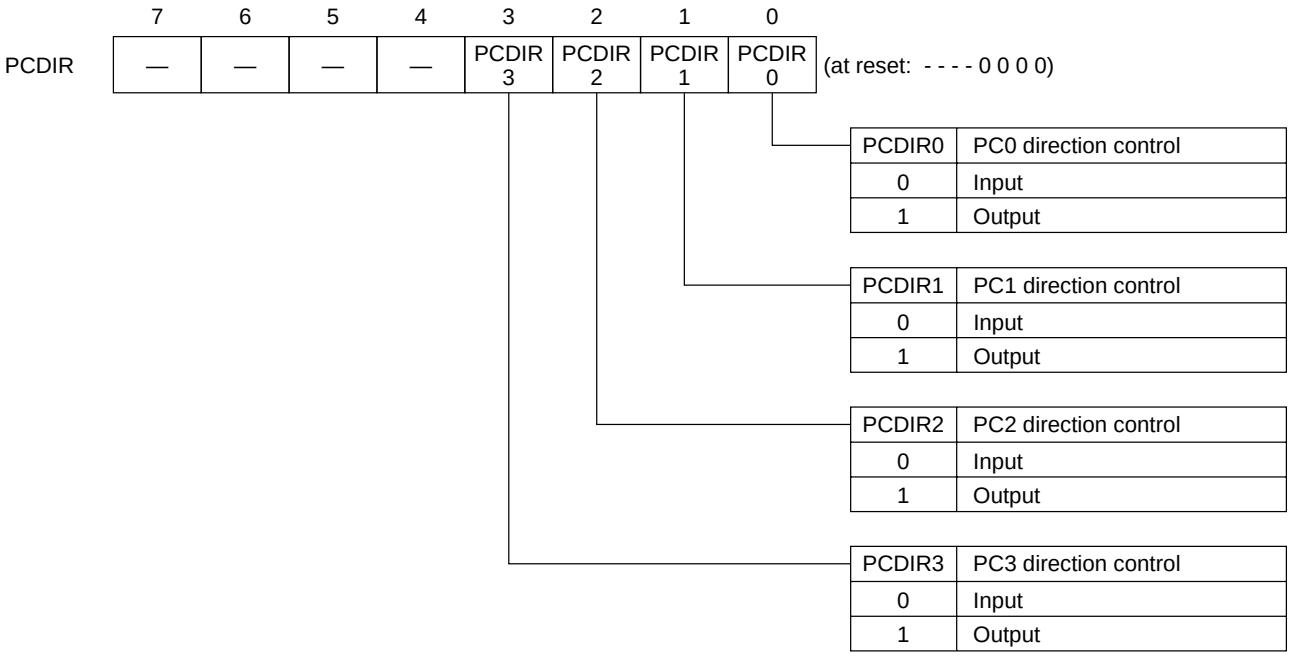


Figure 12-57 Port C Direction Control Register (PCDIR: x'3F37', R/W)

Port 1 Selection Register

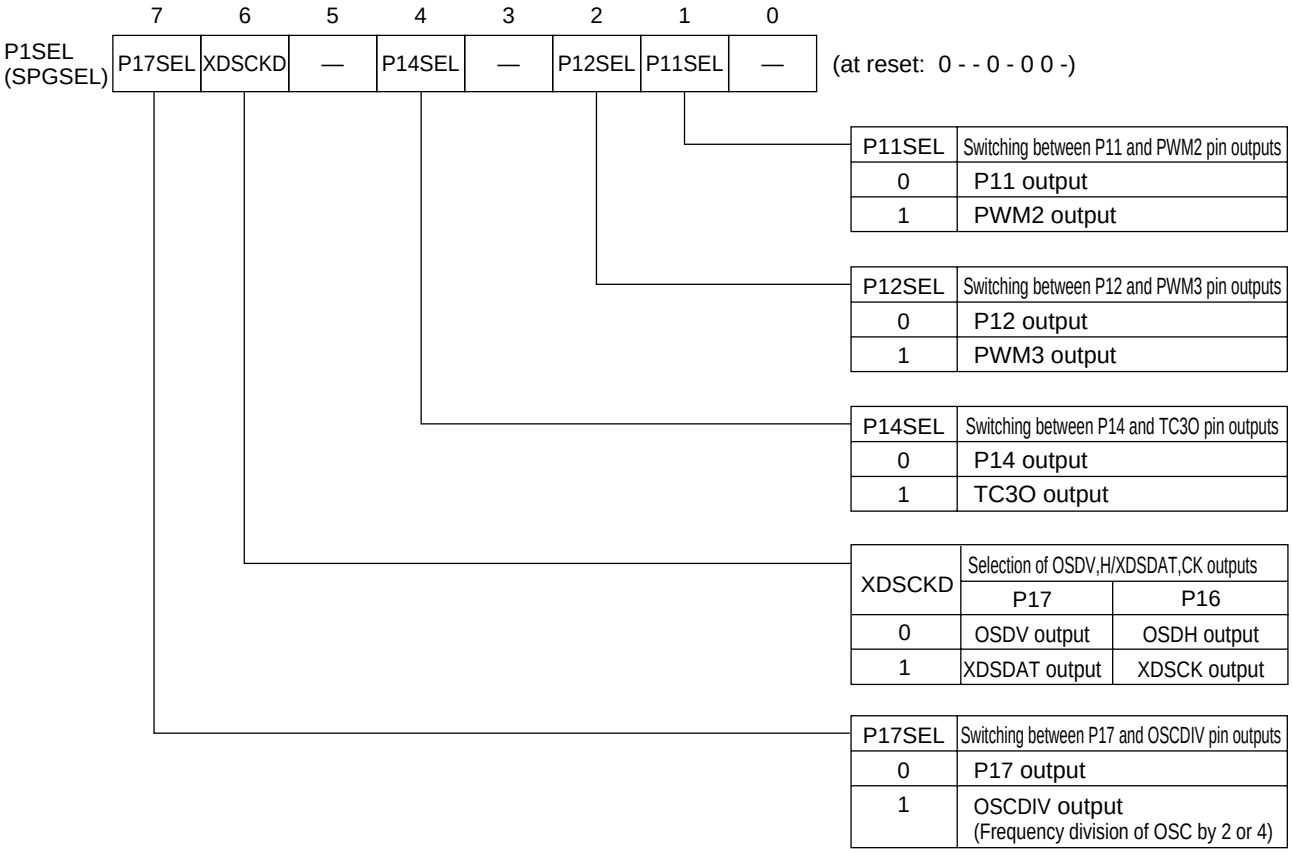


Figure 12-58 Port 1 Selection Register (P1SEL [SPGSEL]: x'3F38', R/W)

Port 2 Selection Register

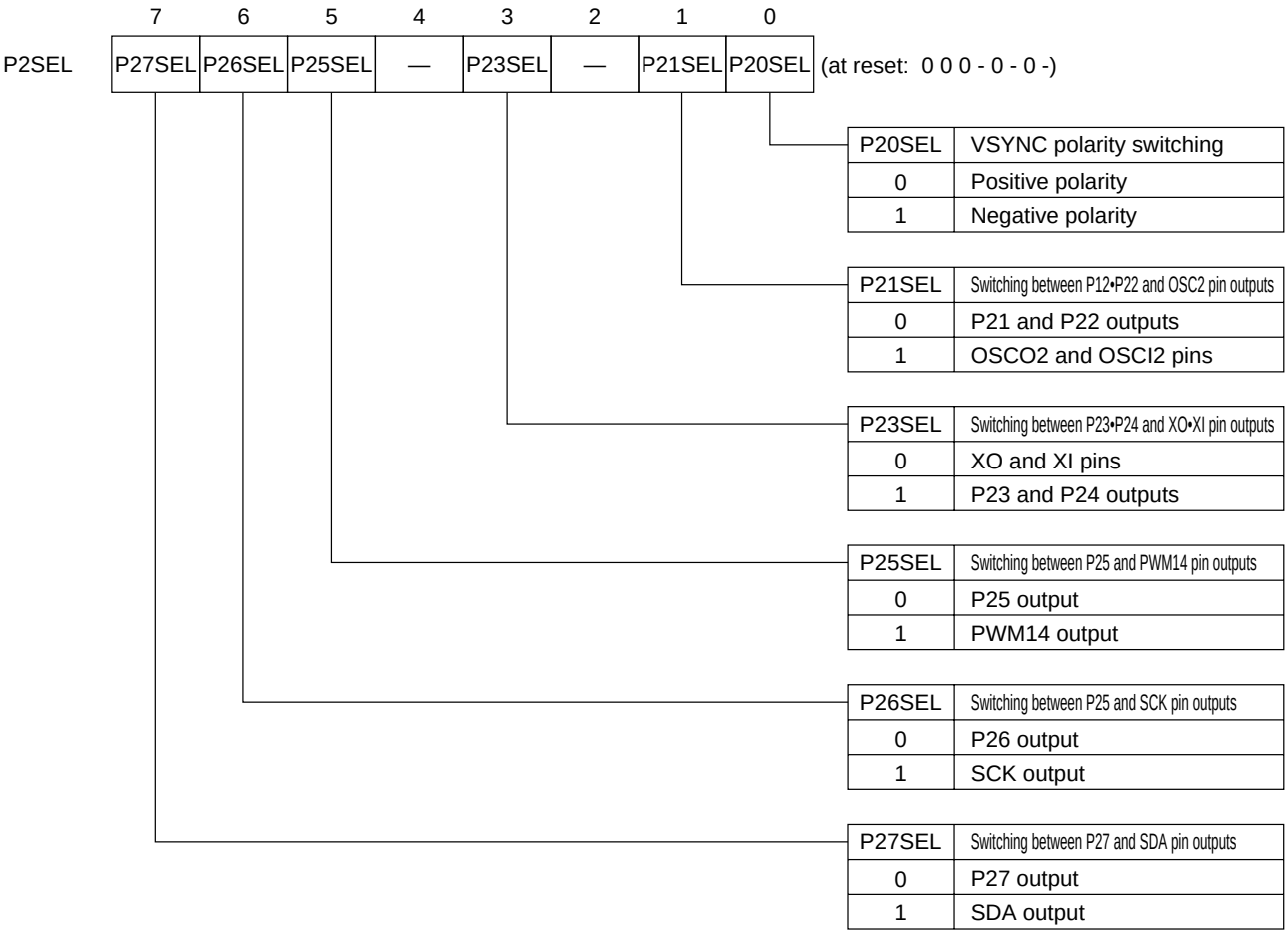


Figure 12-59 Port 2 Selection Register (P2SEL: x'3F39', R/W)

Port 6 Selection Register

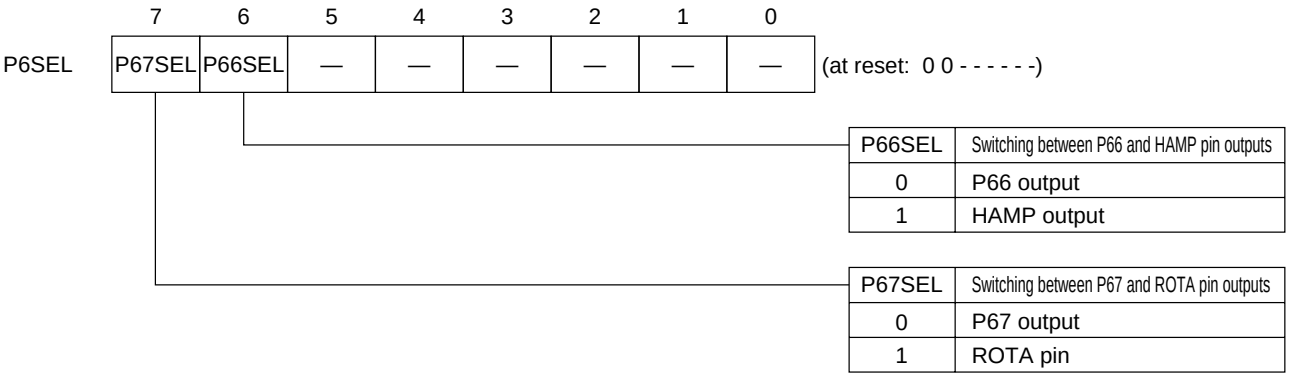


Figure 12-60 Port 6 Selection Register (P6SEL: x'3F3A', R/W)

Port A Selection Register

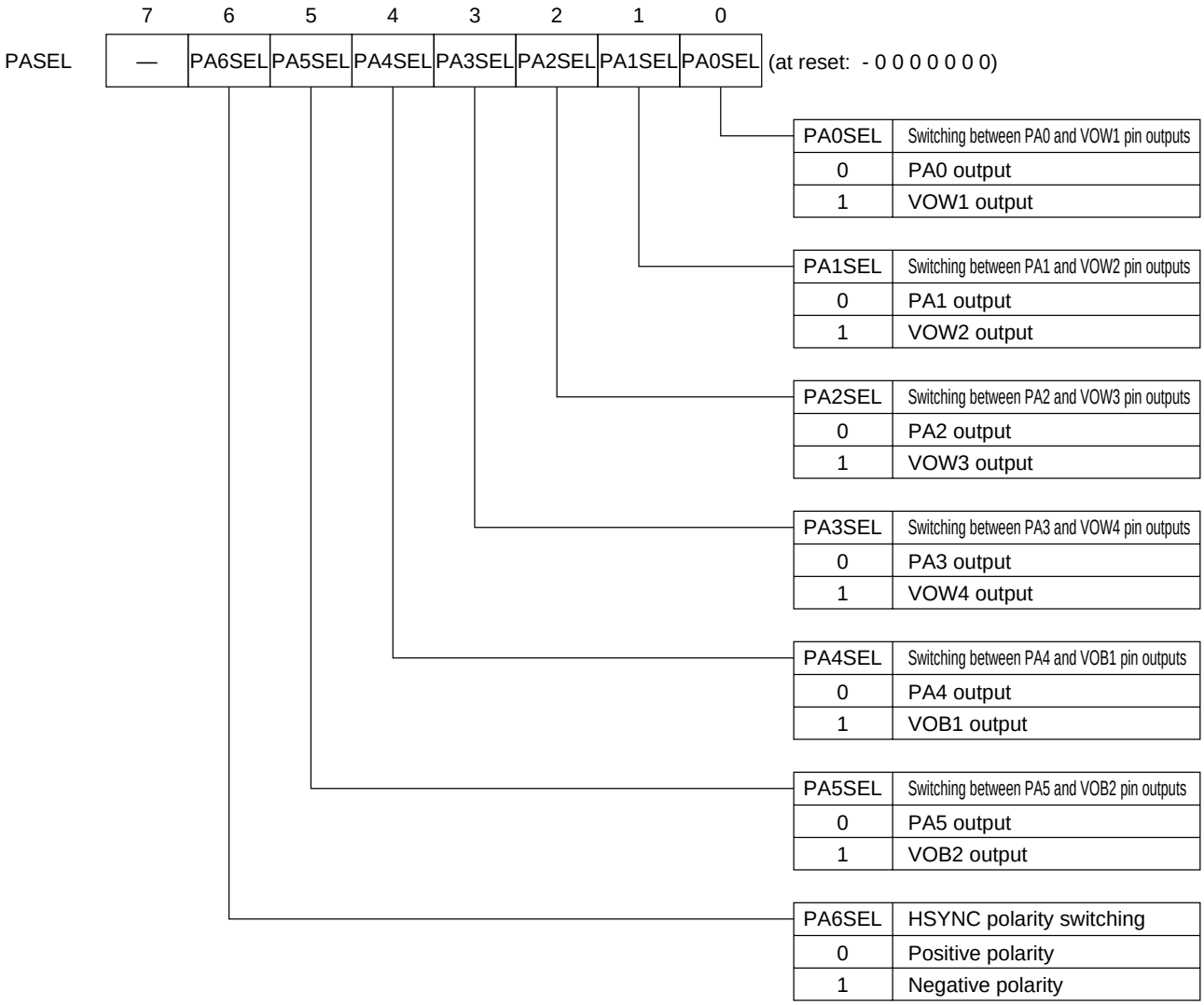


Figure 12-61 Port A Selection Register (PASEL: x'3F3B', R/W)

Serial Output Function Control Register

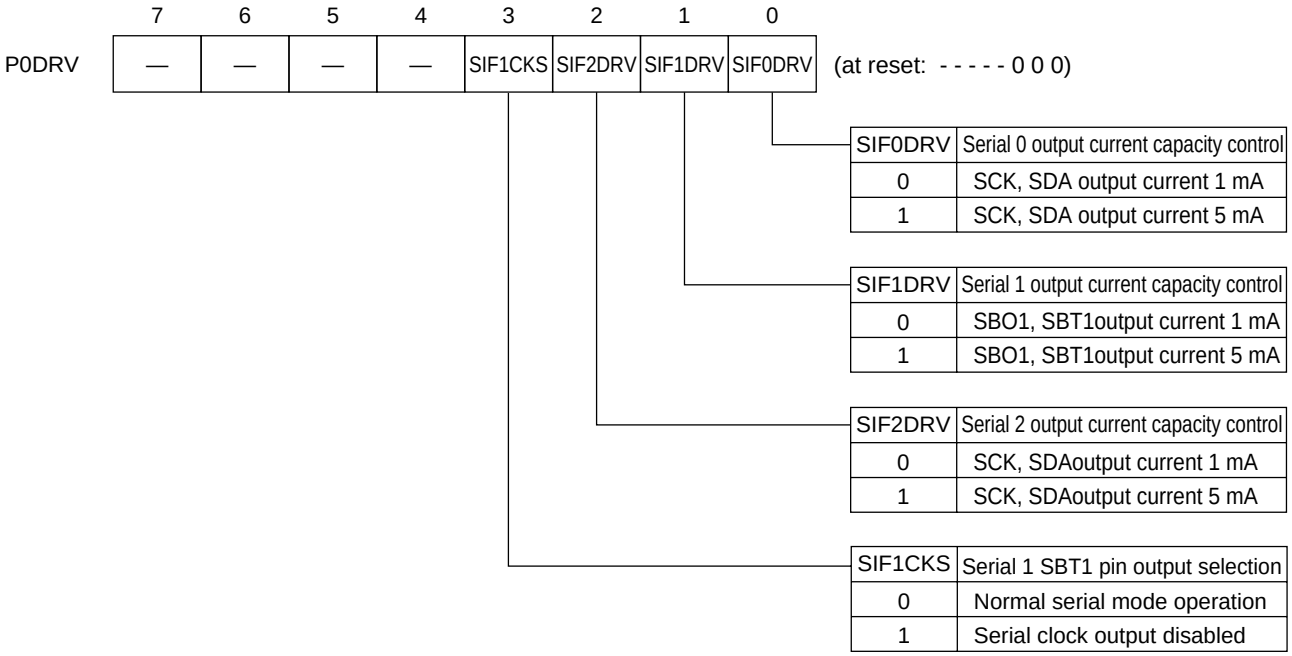


Figure 12-62 Serial Output Function Control Register (P0DRV: x'3F3C', R/W)

Port 1 Pull-Up Control Register

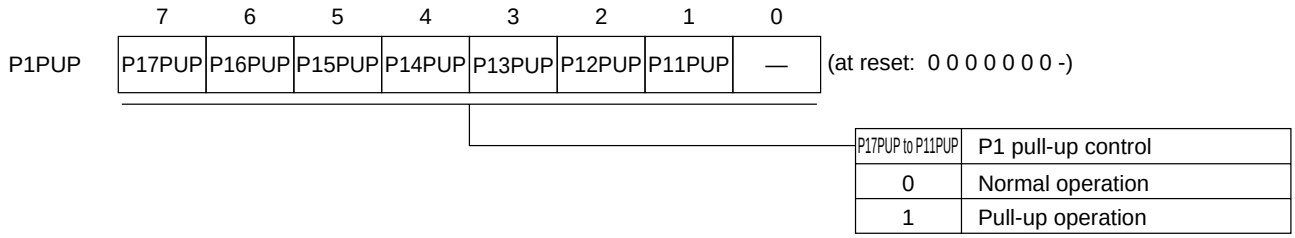
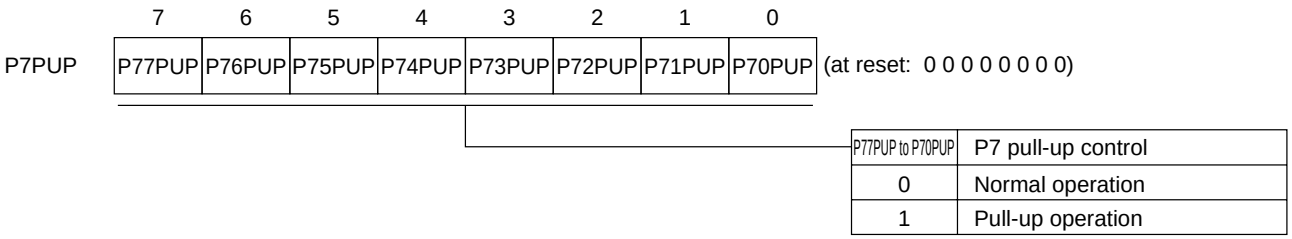


Figure 12-63 Port 1 Pull-Up Control Register (P1PUP: x'3F3D', R/W)

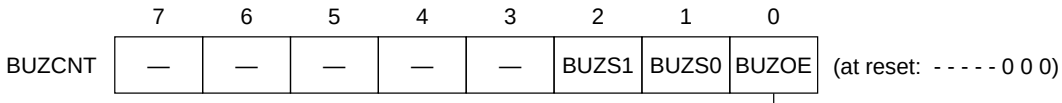
Port 7 Pull-Up Control Register



P77PUP to P70PUP	P7 pull-up control
0	Normal operation
1	Pull-up operation

Figure 12-64 Port 7 Pull-Up Control Register (P7PUP: x'3F3E', R/W)

Buzzer Control Register



BUZOE	Enable buzzer output
0	Stop buzzer output
1	Enable buzzer output

BUZS1 to 0	Buzzer output frequency setting	
	NORMAL mode servoclk=0	NORMAL mode servoclk=1
0 0	6992Hz	3496Hz
0 1	3496Hz	1748Hz
1 0	1748Hz	874Hz
1 1	874Hz	437Hz

The above buzzer frequency is for fosc = 14.32MHz.

Figure 12-65 Buzzer Control Register (BUZCNT: x'3F3F', R/W)

Timer 0 Interval Setting Register

Setting value = x'10000' - (count value)

TB0

7	6	5	4	3	2	1	0	
TB0H 15	TB0H 14	TB0H 13	TB0H 12	TB0H 11	TB0H 10	TB0H 9	TB0H 8	(at reset: - - - - - - - -)

TB0H (x'3F41')

7	6	5	4	3	2	1	0	
TB0L 7	TB0L 6	TB0L 5	TB0L 4	TB0L 3	TB0L 2	TB0L 1	TB0L 0	(at reset: - - - - - - - -)

TB0L (x'3F40')

Use a MOVW instruction to change these bits.

Figure 12-66 Timer 0 Interval Setting Register
(TB0: x'3F41' to x'3F40', R/W)

Timer 1 Interval Setting Register

Setting value = x'10000' - (count value)

TB1

7	6	5	4	3	2	1	0	
TB1H 15	TB1H 14	TB1H 13	TB1H 12	TB1H 11	TB1H 10	TB1H 9	TB1H 8	(at reset: - - - - - - - -)

TB1H (x'3F43')

7	6	5	4	3	2	1	0	
TB1L 7	TB1L 6	TB1L 5	TB1L 4	TB1L 3	TB1L 2	TB1L 1	TB1L 0	(at reset: - - - - - - - -)

TB1L (x'3F42')

Use a MOVW instruction to change these bits.

Figure 12-67 Timer 1 Interval Setting Register
(TB1: x'3F43' to x'3F42', R/W)

Timer 2 Interval Setting Register

Setting value = x'10000' - (count value)

TB2

7	6	5	4	3	2	1	0	
TB2H 15	TB2H 14	TB2H 13	TB2H 12	TB2H 11	TB2H 10	TB2H 9	TB2H 8	(at reset: - - - - - - - -)

TB2H (x'3F45')

7	6	5	4	3	2	1	0	
TB2L 7	TB2L 6	TB2L 5	TB2L 4	TB2L 3	TB2L 2	TB2L 1	TB2L 0	(at reset: - - - - - - - -)

TB2L (x'3F44')

Use a MOVW instruction to change these bits.

Figure 12-68 Timer 2 Interval Setting Register
(TB2: x'3F45' to x'3F44', R/W)

Timer 3 Interval Setting Register

Setting value = x'10000' - (count value)

TB3

7	6	5	4	3	2	1	0	
TB3H 15	TB3H 14	TB3H 13	TB3H 12	TB3H 11	TB3H 10	TB3H 9	TB3H 8	(at reset: - - - - - - - -)

TB3H (x'3F47')

7	6	5	4	3	2	1	0	
TB3L 7	TB3L 6	TB3L 5	TB3L 4	TB3L 3	TB3L 2	TB3L 1	TB3L 0	(at reset: - - - - - - - -)

TB3L (x'3F46')

Use a MOVW instruction to change these bits.

Figure 12-69 Timer 3 Interval Setting Register
(TB3: x'3F47' to x'3F46', R/W)

Timer 4 Interval Setting Register

Setting value = x'10000' - (count value)

TB4

7	6	5	4	3	2	1	0	
TB4H 15	TB4H 14	TB4H 13	TB4H 12	TB4H 11	TB4H 10	TB4H 9	TB4H 8	(at reset: - - - - - - - -)

TB4H (x'3F49')

7	6	5	4	3	2	1	0	
TB4L 7	TB4L 6	TB4L 5	TB4L 4	TB4L 3	TB4L 2	TB4L 1	TB4L 0	(at reset: - - - - - - - -)

TB4L (x'3F48')

Use a MOVW instruction to change these bits.

Figure 12-70 Timer 4 Interval Setting Register
(TB4: x'3F49' to x'3F48', R/W)

Timer 6 Interval Setting Register

Setting value = x'10000' - (count value)

TB6S

7	6	5	4	3	2	1	0	
TB6SH 15	TB6SH 14	TB6SH 13	TB6SH 12	TB6SH 11	TB6SH 10	TB6SH 9	TB6SH 8	(at reset: - - - - - - - -)

TB6SH (x'3F4B')

7	6	5	4	3	2	1	0	
TB6SL 7	TB6SL 6	TB6SL 5	TB6SL 4	TB6SL 3	TB6SL 2	TB6SL 1	TB6SL 0	(at reset: - - - - - - - -)

TB6SL (x'3F4A')

Use a MOVW instruction to change these bits.

Figure 12-71 Timer 6 Interval Setting Register
(TB6S: x'3F4B' to x'3F4A', R/W)

Timer 0 Binary Counter

The 16-bit count value can be monitored.

BC0

7	6	5	4	3	2	1	0	
BC0H 15	BC0H 14	BC0H 13	BC0H 12	BC0H 11	BC0H 10	BC0H 9	BC0H 8	(at reset: - - - - - - - -)

BC0H (x'3F4D')

7	6	5	4	3	2	1	0	
BC0L 7	BC0L 6	BC0L 5	BC0L 4	BC0L 3	BC0L 2	BC0L 1	BC0L 0	(at reset: - - - - - - - -)

BC0L (x'3F4C')

Use a MOVW instruction to change these bits.

Figure 12-72 Timer 0 Binary Counter (BC0: x'3F4D' to x'3F4C', R)

Timer 1 Binary Counter

The 16-bit count value can be monitored.

BC1

7	6	5	4	3	2	1	0	
BC1H 15	BC1H 14	BC1H 13	BC1H 12	BC1H 11	BC1H 10	BC1H 9	BC1H 8	(at reset: - - - - - - - -)

BC1H (x'3F4F')

7	6	5	4	3	2	1	0	
BC1L 7	BC1L 6	BC1L 5	BC1L 4	BC1L 3	BC1L 2	BC1L 1	BC1L 0	(at reset: - - - - - - - -)

BC1L (x'3F4E')

Use a MOVW instruction to change these bits.

Figure 12-73 Timer 1 Binary Counter (BC1: x'3F4F' to x'3F4E', R)

Timer 2 Binary Counter

The 16-bit count value can be monitored.

BC2

7	6	5	4	3	2	1	0	
BC2H 15	BC2H 14	BC2H 13	BC2H 12	BC2H 11	BC2H 10	BC2H 9	BC2H 8	(at reset: - - - - - - - -)

BC2H (x'3F51')

7	6	5	4	3	2	1	0	
BC2L 7	BC2L 6	BC2L 5	BC2L 4	BC2L 3	BC2L 2	BC2L 1	BC2L 0	(at reset: - - - - - - - -)

BC2L (x'3F50')

Use a MOVW instruction to change these bits.

Figure 12-74 Timer 2 Binary Counter (BC2: x'3F51' to x'3F50', R)

Timer 3 Binary Counter

The 16-bit count value can be monitored.

BC3

7	6	5	4	3	2	1	0	
BC3H 15	BC3H 14	BC3H 13	BC3H 12	BC3H 11	BC3H 10	BC3H 9	BC3H 8	(at reset: - - - - - - - -)

BC3H (x'3F53')

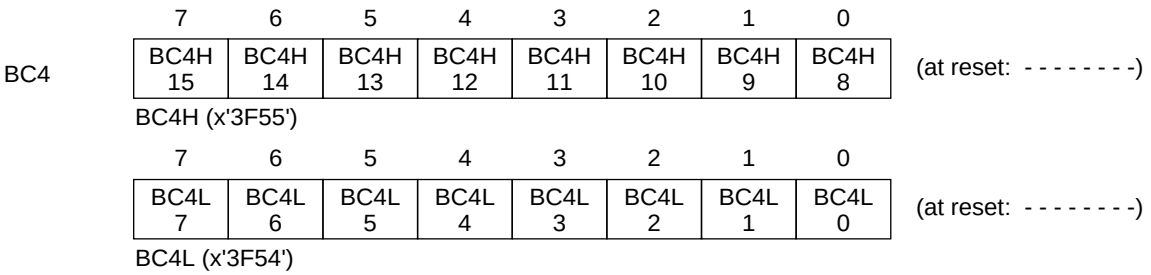
7	6	5	4	3	2	1	0	
BC3L 7	BC3L 6	BC3L 5	BC3L 4	BC3L 3	BC3L 2	BC3L 1	BC3L 0	(at reset: - - - - - - - -)

BC3L (x'3F52')

Use a MOVW instruction to change these bits.

Figure 12-75 Timer 3 Binary Counter (BC3: x'3F53' to x'3F52', R)

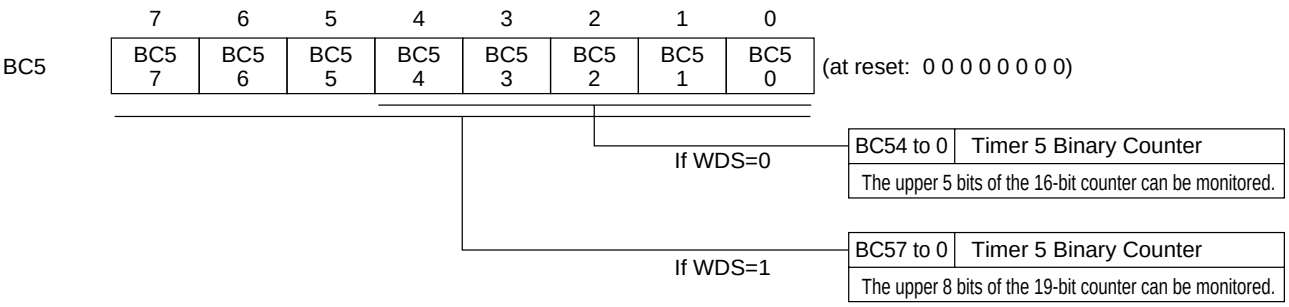
Timer 4 Binary Counter



Use a MOVW instruction to change these bits.

Figure 12-76 Timer 4 Binary Counter (BC4: x'3F55' to x'3F54', R)

Timer 5 Binary Counter



The lower 11 bits of the 16-bit counter cannot be monitored.

Figure 12-77 Timer 5 Binary Counter (BC5: x'3F56', R)

Timer 6 Binary Counter

BC6S

7	6	5	4	3	2	1	0	
BC6SH 15	BC6SH 14	BC6SH 13	BC6SH 12	BC6SH 11	BC6SH 10	BC6SH 9	BC6SH 8	(at reset: - - - - - - - -)

BC6SH (x'3F59')

7	6	5	4	3	2	1	0	
BC6SL 7	BC6SL 6	BC6SL 5	BC6SL 4	BC6SL 3	BC6SL 2	BC6SL 1	BC6SL 0	(at reset: - - - - - - - -)

BC6SL (x'3F58')

Use a MOVW instruction to change these bits.

Figure 12-78 Timer 6 Binary Counter (BC6S: x'3F59' to x'3F58', R)

Analog Monitor Output Selection Register

	7	6	5	4	3	2	1	0	
ANAMON	SNF	YPG MON	VSY MON	YFG MON	AFG MON1	IRQ3 MON	IRQ2 MON	VISS MON	(At reset: 0000000)

VISS MON	P70 and VISSCTL monitor output selection
0	P70 output
1	VISSCTL Monitor output

IRQ2 MON	P71 and IRQ2 monitor output
0	P71 output
1	IRQ2 monitor output

IRQ3 MON	P72 and IRQ3 monitor output
0	P72 output
1	IRQ3 monitor output

AFG MON1	P73 and AFG monitor output selection
0	P73 output
1	AFG Monitor output

YFG MON	P74 and YFG monitor output selection
0	P74 output
1	YFG Monitor output

YSY MON	P75 and VSYNC monitor output selection
0	P75 output
1	VSYNC Monitor output

YPG MON	P76 and YPG monitor output selection
0	P76 output
1	YPG Monitor output

SNF	Noise filter clock selection	IRQ2	IRQ3	AFG	PFG	YFG
0	Noise filter OFF	fs/2	fs/2	—	fs/2	—
1	Noise filter ON	fs/8	fs/8	fs/4	fs/4	fs/4

Figure 12-79 Analog Monitor Output Selection Register (ANAMON: x'3F57')

Timer Mode Register 0

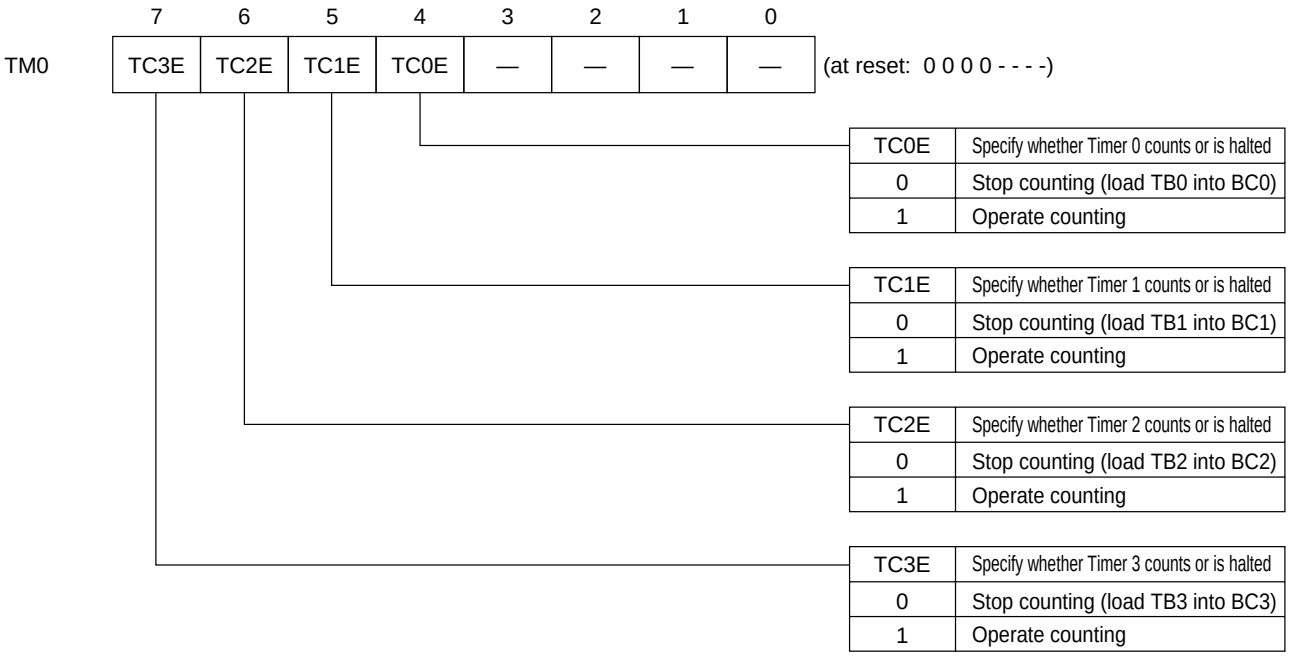


Figure 12-80 Timer Mode Register 0 (TM0: x'3F5A', R/W)

Timer Mode Register 1

	7	6	5	4	3	2	1	0	
TM1	TC4M	TC4E	TC3M	TC1M1	TC1M0	—	TC0M1	TC0M0	(at reset: 0 0 0 0 0 - 0 0)

TC0M1 to 0	Specify timer 0 clock source		
	SXI pin = LOW		SXI pin = HIGH
	SERVOCLK=0	SERVOCLK=1	SERVOCLK
0 0	fosc/512		fxi
0 1	Timer 6 overflow		Same as values at left
1 0	fs/2	fs/4	
1 1	fs/8	fs/16	

TC1M1 to 0	Specify timer 1 clock source	
	SERVOCLK=0	SERVOCLK=1
0 0	fs/8	fs/16
0 1	fs/2	fs/4
1 0	PCTL signal	

TC3M	Specify timer 3 clock source	
	SERVOCLK=0	SERVOCLK=1
0	fs/8	fs/16
1	fs/2	fs/4

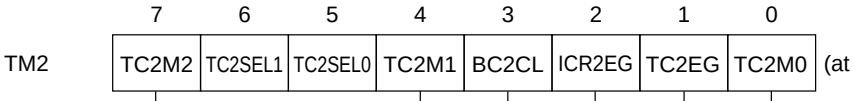
TC4E	Specify whether Timer 4 counts or is halted
0	Stop counting (load TB4 into BC4)
1	Operate counting

TC4M	Specify timer 4 clock source	
	SERVOCLK=0	SERVOCLK=1
0	fs/8	fs/16
1	P15 (TC41) external clock input	

If fs/2 is used as the clock source for the timer, set the I/O wait to "no-wait".

Figure 12-81 Timer Mode Register 1 (TM1: x'3F5B', R/W)

Timer Mode Register 2



TC2M2 to 0	Specify timer 0 clock source			
	SERVOCLK=0		SERVOCLK=1	
0 0 0	fs/2		fs/4	
0 1 1	fs/12		fs/24	
1 1 1	fs/8		fs/16	
	VISS CTL=0	VISS CTL=1	VISS CTL=0	VISS CTL=1
0 0 1	fs/12	fs/8	fs/24	fs/16
1 0 1	fs/8	fs/12	fs/16	fs/24

TC2EG	Specify timer 2 valid edge	
	ICR2 edge	SR2F edge
0	Falling	Rising
1	Rising	Falling

ICR2EG	Timer 2 input capture operation control
0	Stop capture
1	Operate capture

BC2CL	BC2 clear operation control
0	Normal
1	Cleared at both edges of the VISSCTL signal

TC2SEL1 to 0	Specify Timer 2 interrupt source
0 0	Overflow of binary counter (BC2)
0 1	ICR2 capture signal
0 1	Shift counter underflow
1 0	Match with value in compare register (CMPR2)

Figure 12-82 Timer Mode Register 2 (TM2: x'3F5C', R/W)

If fs/2 is used as the clock source for the timer, set the I/O wait to "no-wait".

Timer 2 Input Capture Register

Value of BC2 is captured when the valid edge of the VISSCTL signal is input.

ICR2

7	6	5	4	3	2	1	0	
ICR2H 15	ICR2H 14	ICR2H 13	ICR2H 12	ICR2H 11	ICR2H 10	ICR2H 9	ICR2H 8	(at reset: - - - - - - - -)

ICR2H (x'3F5E')

7	6	5	4	3	2	1	0	
ICR2L 7	ICR2L 6	ICR2L 5	ICR2L 4	ICR2L 3	ICR2L 2	ICR2L 1	ICR2L 0	(at reset: - - - - - - - -)

ICR2L (x'3F5D')

Use a MOVW instruction to change these bits.

Figure 12-83 Timer 2 Input Capture Register
(ICR2: x'3F5E' to x'3F5D', R)

Timer 2 Shift Register

Results of the duty determing function are shifted in LSB first.

SR2F

7	6	5	4	3	2	1	0	
SR2FH 15	SR2FH 14	SR2FH 13	SR2FH 12	SR2FH 11	SR2FH 10	SR2FH 9	SR2FH 8	(at reset: - - - - - - - -)

SR2FH (x'3F60')

7	6	5	4	3	2	1	0	
SR2FL 7	SR2FL 6	SR2FL 5	SR2FL 4	SR2FL 3	SR2FL 2	SR2FL 1	SR2FL 0	(at reset: - - - - - - - -)

SR2FL (x'3F5F')

Use a MOVW instruction to change these bits.

Figure 12-84 Timer 2 Shift Register (SR2F: x'3F60' to x'3F5F', R/W)

Compare Register for Timer 2 Shift Register

Comparison with SR2F data.
With the exception of the upper 5 bits and the lowest 1 bit, 10 bits of data can be compared.

CMPR2

7	6	5	4	3	2	1	0	
CMPR2H 15	CMPR2H 14	CMPR2H 13	CMPR2H 12	CMPR2H 11	CMPR2H 10	CMPR2H 9	CMPR2H 8	(at reset: - - - - - - - -)

CMPR2H (x'3F62')

7	6	5	4	3	2	1	0	
CMPR2L 7	CMPR2L 6	CMPR2L 5	CMPR2L 4	CMPR2L 3	CMPR2L 2	CMPR2L 1	CMPR2L 0	(at reset: - - - - - - - -)

CMPR2L (x'3F61')

Use a MOVW instruction to change these bits.

Figure 12-85 Compare Register for Timer 2 Shift Register
(CMPR2: x'3F62' to x'3F61', R/W)

Count Value Setting Register for Timer 2 Shift Counter

Specifies the number of shifts.
Setting value = (count value)-1

SRPC2

7	6	5	4	3	2	1	0	
—	—	—	—	SRPC2 3	SRPC2 2	SRPC2 1	SRPC2 0	(at reset: - - - - - - - -)

Figure 12-86 Count Value Setting Register for Timer 2 Shift Counter
(SRPC2: x'3F63', R/W)

4-Bit Counter for Timer 2 Shift Register

The count value of the shift counter can be monitored.

	7	6	5	4	3	2	1	0	
SRBC2	—	—	—	—	SRBC2 3	SRBC2 2	SRBC2 1	SRBC2 0	(at reset: - - - - - - - -)

Figure 12-87 4-Bit Counter for Timer 2 Shift Register (SRBC2: x'3F64', R)

Data Bit Exchange Register

This register rearranges bits.
Bits are rearranged as: Dn <==> D(15-n) (where n= 8 to 15)
: Dn <==> D(7-n) (where n= 0 to 7)

SR2R

7	6	5	4	3	2	1	0	
SR2RH 15	SR2RH 14	SR2RH 13	SR2RH 12	SR2RH 11	SR2RH 10	SR2RH 9	SR2RH 8	(at reset: - - - - - - - -)

SR2RH (x'3F66')

7	6	5	4	3	2	1	0	
SR2RL 7	SR2RL 6	SR2RL 5	SR2RL 4	SR2RL 3	SR2RL 2	SR2RL 1	SR2RL 0	(at reset: - - - - - - - -)

SR2RL (x'3F65')

Use a MOVW instruction to change these bits.

Figure 12-88 Data Bit Exchange Register
(SR2R: x'3F66' to x'3F65', R/W)

Output Compare Register for Timer 4

Sets the output pulse width for remote control output data.

OCR4

7	6	5	4	3	2	1	0	
OCR4H 15	OCR4H 14	OCR4H 13	OCR4H 12	OCR4H 11	OCR4H 10	OCR4H 9	OCR4H 8	(at reset: - - - - - - - -)

OCR4H (x'3F68')

7	6	5	4	3	2	1	0	
OCR4L 7	OCR4L 6	OCR4L 5	OCR4L 4	OCR4L 3	OCR4L 2	OCR4L 1	OCR4L 0	(at reset: - - - - - - - -)

OCR4L (x'3F67')

Use a MOVW instruction to change these bits.

If this function is to be used, divide the Timer 4 frequency by 2 and use as the clock source for serial interface 1.

Figure 12-89 Output Compare Register for Timer 4
(OCR4: x'3F68' to x'3F67', R/W)

Synchronous Output Data Register 0

Sets the data for synchronous output to P4OUT (HOCRBUF0).

HOCRDAT0

7	6	5	4	3	2	1	0	
HOCRDAT0 7	HOCRDAT0 6	HOCRDAT0 5	HOCRDAT0 4	HOCRDAT0 3	HOCRDAT0 2	HOCRDAT0 1	HOCRDAT0 0	(at reset: - - - - - - - -)

Figure 12-90 Synchronous Output Data Register 0
(HOCRDAT0: x'3F69', R/W)

Saturation Detection Control Register

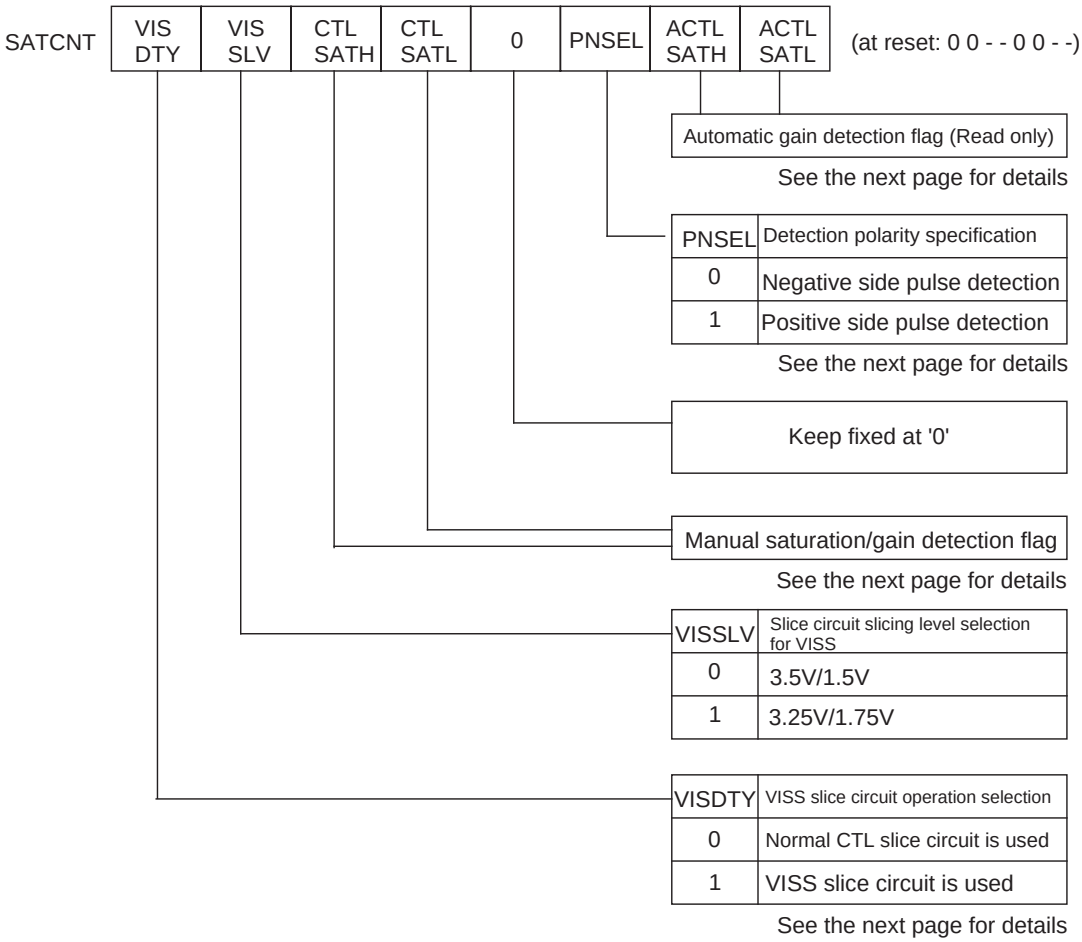


Figure 12-91 Saturation Detection Control Register
(SATCNT: x'3F6A', R/W, one section is read only)

Supplementary Description of Saturation Detection Control Register (SATCNT: x'3F6A' R/W, partly R)

SATCNT	VIS DTY	VIS SLV	CTL SATH	CTL SATL	0	PNSEL	ACTL SATH	ACTL SATL

Saturation detection (SATS flag = '0')

* Can only be used during manual detection.

CTLSATL	Detection of CTL input of 0.5V or less (saturation)
0	CTL Output voltage≤0.5V(-2.0Vop) (saturation)
1	CTL Output voltage>0.5V(-2.0Vop) (no saturation)

CTLSATH	Detection of CTL input of 4.5V or more (saturation)
0	CTL Output voltage≥4.5V(+2.0Vop) (saturation)
1	CTL Output voltage<4.5V(+2.0Vop) (no saturation)

Gain detection (SATS flag = '1')

* Can be used during manual detection and automatic detection.

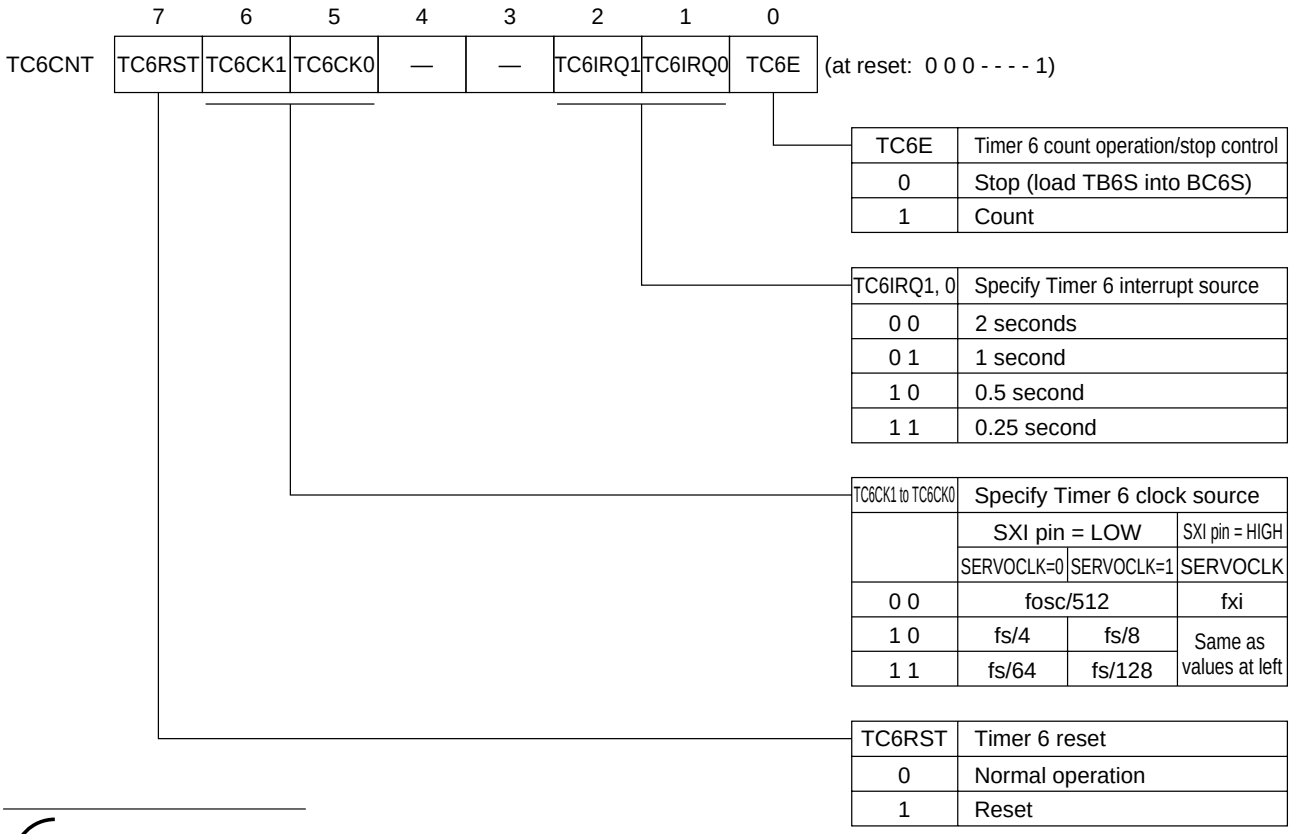
Setting		Gain judgment		Detection level	
VISDTY	PNSEL	CTLSATL, ACTLSATL	CTLSATH, ACTLSATH		
0	0 (Negative)	1	1	CTL Output of 1.7V (-0.8Vop) or more (insufficient gain)	-0.8Vop/-1.2Vop (1.7V/1.3V) Detection
		0	1	CTL Output of 1.7V to 1.3V (appropriate gain)	
		0	0	CTL Output of less than 1.3V (-1.2Vop) (excessive gain)	
	1 (Positive)	1	1	CTL Output of 3.3V (+0.8Vop) or more (insufficient gain)	+0.8Vop/+1.2Vop (3.3V/3.7V) Detection
		0	1	CTL Output of 3.3V to 3.7V (appropriate gain)	
		0	0	CTL Output of less than 3.7V (+1.2Vop) (excessive gain)	
1	0 (Negative)	1	1	CTL Output of 1.0V (-1.5Vop) or more (insufficient gain)	-1.5Vop/-2.0Vop (1.0V/0.5V) Detection
		0	1	CTL Output of 1.0V to 0.5V (appropriate gain)	
		0	0	CTL Output of less than 0.5V (-2.0Vop) (excessive gain)	
	1 (Positive)	1	1	CTL Output of 4.0V (+1.5Vop) or more (insufficient gain)	+1.5Vop/+2.0Vop (4.0V/4.5V) Detection
		0	1	CTL Output of 4.0V to 4.5V (appropriate gain)	
		0	0	CTL Output of less than 4.5V (-2.0Vop) (excessive gain)	

Automatic gain detection: The ACTLSATL/H flags are set when the input exceeds ±0.50Vop, and the gain detection result is latched in the ACTLSATL/H flags when the input falls below ±0.15V. The saturation detection cannot be made automatically.

Manual detection: It is necessary to positively set the CTLSATH/L flags by software before inputting the control pulse. The results of saturation/gain detection are immediately reflected in the CTLSATH/L flags.

Figure 12-92 Supplementary Explanation of the Saturation Detection Control Register

Timer 6 Control Register



TC6E	Timer 6 count operation/stop control
0	Stop (load TB6S into BC6S)
1	Count

TC6IRQ1, 0	Specify Timer 6 interrupt source
0 0	2 seconds
0 1	1 second
1 0	0.5 second
1 1	0.25 second

TC6CK1 to TC6CK0	Specify Timer 6 clock source		
	SXI pin = LOW		SXI pin = HIGH
	SERVOCLK=0	SERVOCLK=1	SERVOCLK
0 0	fosc/512		fxi
1 0	fs/4	fs/8	Same as values at left
1 1	fs/64	fs/128	

TC6RST	Timer 6 reset
0	Normal operation
1	Reset

Setting bp7 to '1' will cause the TC6CNT register to become x'000**001. The value of bp7 is automatically reset to '0'.

Figure 12-93 Timer 6 Control Register (TC6CNT: x'3F6B', R/W)

A/D Buffer

A/D conversion results of the channel specified by the ADM register are stored in this register. The previous value is saved before storing the next conversion result.

	7	6	5	4	3	2	1	0	
ADBUF	ADBUF 7	ADBUF 6	ADBUF 5	ADBUF 4	ADBUF 3	ADBUF 2	ADBUF 1	ADBUF 0	(at reset: - - - - - - - -)

Figure 12-94 A/D Buffer (ADBUF: x'3F6C', R/W)

A/D Mode Register

	7	6	5	4	3	2	1	0	
ADM	—	ADBUSY	ADCONV	ADEN	ADSEL3	ADSEL2	ADSEL1	ADSEL0	(at reset: - - 0 0 0 0 0 0)

ADSEL3 to ADSEL0	Specify A/D conversion channel
0 0 0 0	Convert AD0
0 0 0 1	Convert AD1
0 0 1 0	Convert AD2
0 0 1 1	Convert AD3
0 1 0 0	Convert AD4
0 1 0 1	Convert AD5
0 1 1 0	Convert AD6
0 1 1 1	Convert AD7
1 0 0 0	Convert AD8
1 0 0 1	Convert AD9
1 0 1 0	Convert AD10
1 0 1 1	Convert AD11
1 1 - -	Prohibited

ADEN	A/D power source ON/OFF
0	Power OFF
1	Power ON

ADCONV	A/D conversion start control
0	Stop A/D conversion
1	Start A/D conversion

ADBUSY	A/D conversion operation flag (read-only)
0	A/D conversion halted
1	A/D conversion in progress

Figure 12-95 A/D Mode Register (ADM: x'3F6D', R/W one section is read-only)

PWM14 Data Register

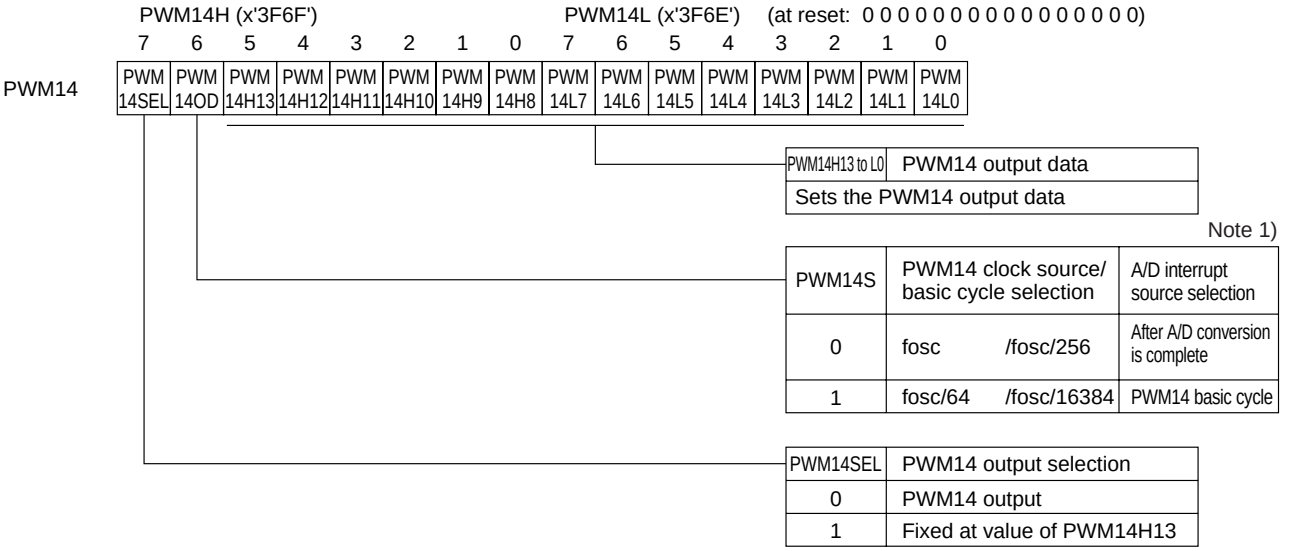


Figure 12-96 PWM14 Data Register (PWM14: x'3F6F' to x'3F6E', R/W)

Note1: ***When PWM14S=1, set PWM14L5 through PWM14L0 to x'00'.***

PWM0 Data Register

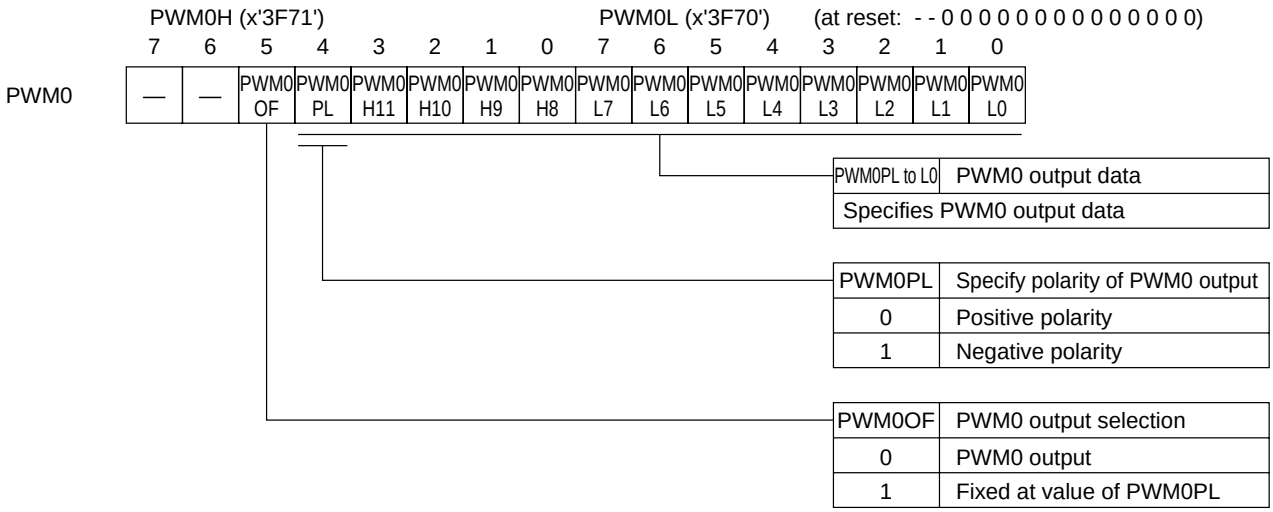


Figure 12-97 PWM0 Data Register (PWM0: x'3F71' to x'3F70', R/W)

PWM1 Data Register

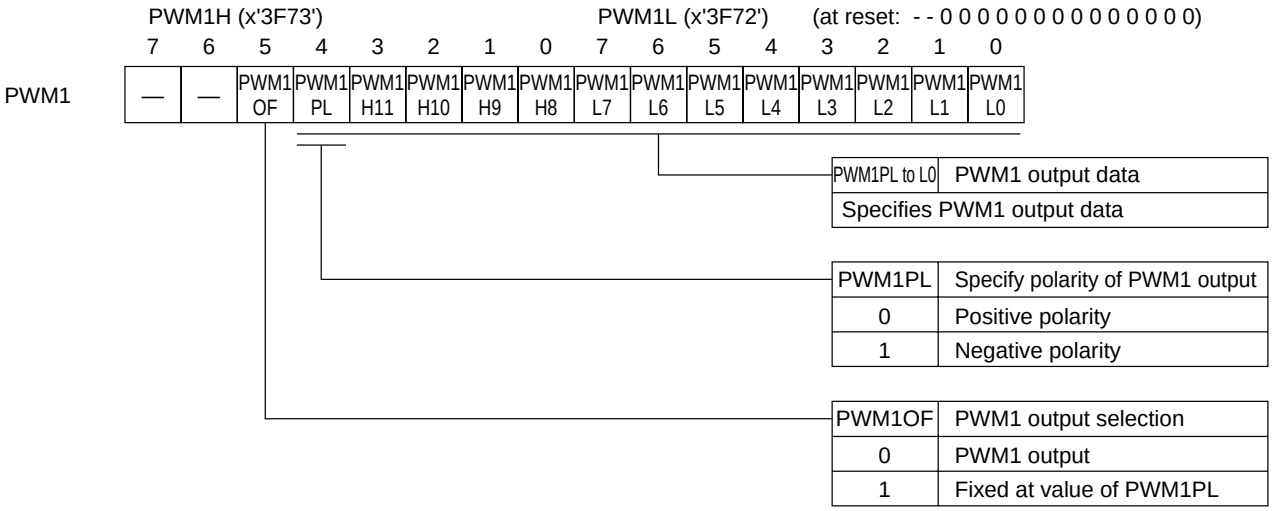


Figure 12-98 PWM1 Data Register (PWM1: x'3F73' to x'3F72', R/W)

PWM2 Data Register

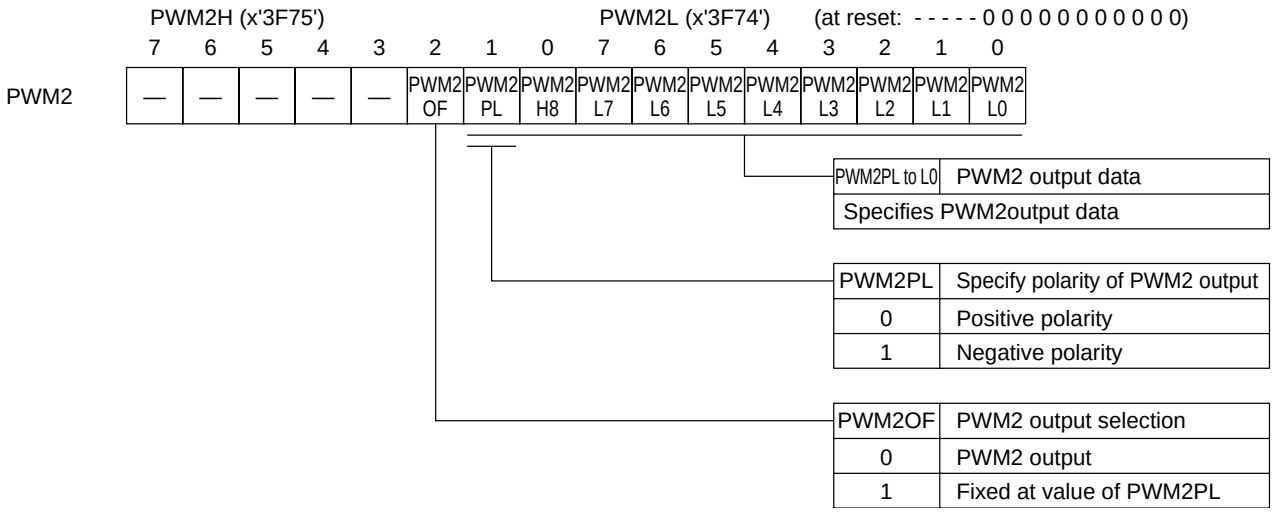


Figure 12-99 PWM2 Data Register (PWM2: x'3F75' to x'3F74', R/W)

PWM3 Data Register

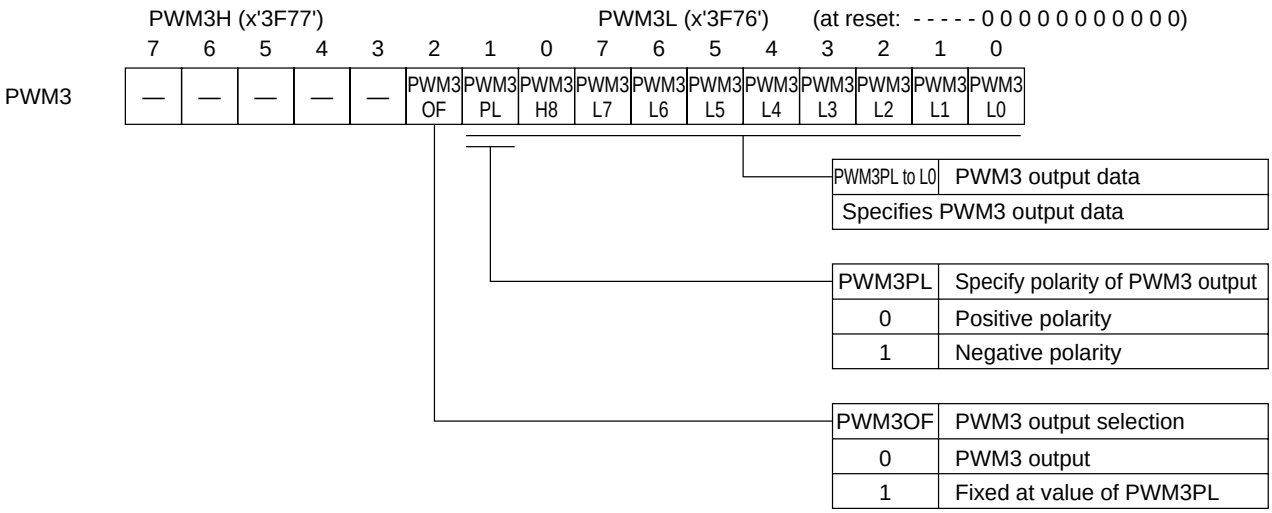


Figure 12-100 PWM3 Data Register (PWM3: x'3F77' to x'3F76', R/W)

PWM Control Register

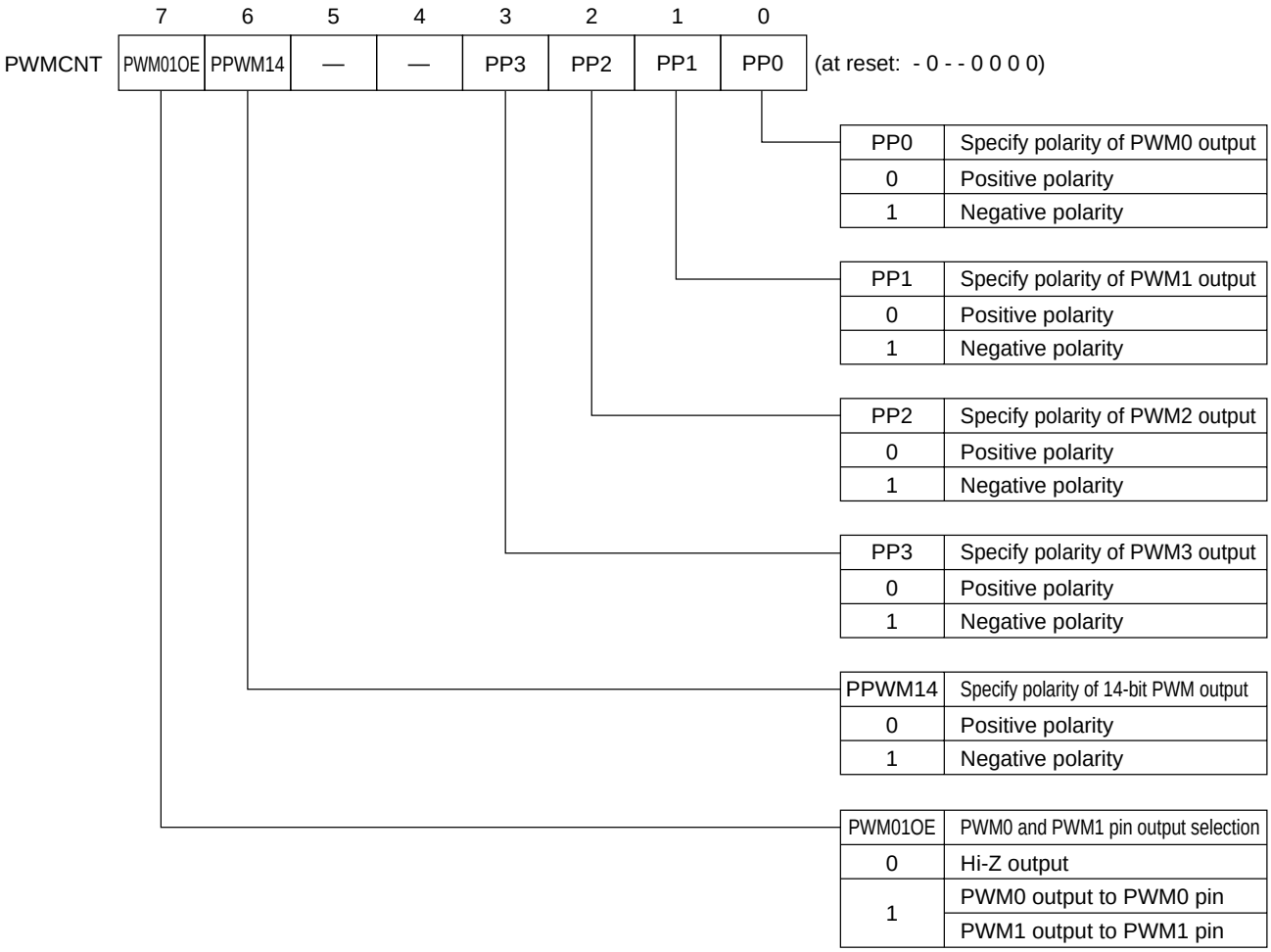


Figure 12-101 PWM Control Register (PWMCNT: x'3F78', R/W)

Synchronous Output Data Register 1

Sets data for synchronous output to HOCRBUF1.

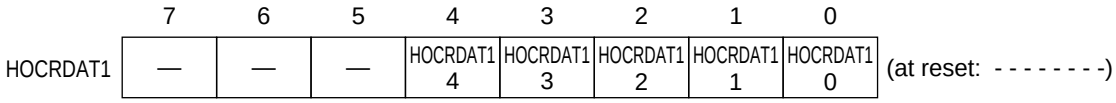


Figure 12-102 Synchronous Output Data Register 1 (HOCRDAT1: x'3F79', R/W)

Synchronous Output Data Register 2

Sets data for synchronous output to RCTLBUF.

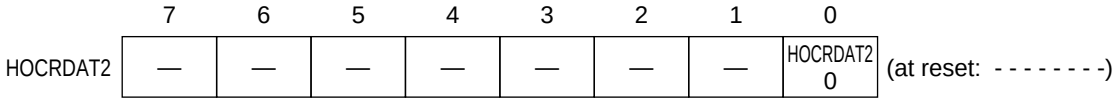


Figure 12-103 Synchronous Output Data Register 2 (HOCRDAT2: x'3F7A', R/W)

Synchronous Output Buffer Register 1

	7	6	5	4	3	2	1	0	
HOCRBUF1	—	—	—	N8RST	COS	CIS	CCD	HC1M	(at reset: - - - - - - - -)

HC1M	Recording current register
0	The recording current is applied.
1	The recording current is not applied.
HOCRSEL1=1	Internal output of bit 0 of HOCRDAT1
HOCRSEL1=0	Internal output buffer

CCD	Recording current direction control register
0	RCTLP→RCTLN
1	RCTLN→RCTLP
HOCRSEL1=1	Internal output of bit 1 of HOCRDAT1
HOCRSEL1=0	Internal output buffer

CIS	CTL amp input switching register (during playback mode)
0	Normal playback
1	Connect to ground
HOCRSEL1=1	Internal output of bit 2 of HOCRDAT1
HOCRSEL1=0	Internal output buffer

COS	CTL amp output (CO) control register
0	Fixed at 2.5 V
1	Amp output
HOCRSEL1=1	Internal output of bit 3 of HOCRDAT1
HOCRSEL1=0	Internal output buffer

Note 1)

N8RST	N8 reset control register
0	No reset
1	N8 reset
HOCRSEL1=1	Internal output of bit 4 of HOCRDAT1
HOCRSEL1=0	Internal output buffer

Note 1: ***Writing a '1' will reset the N8 signal. If captured in the ICRVSYN register at the occurrence of NF or VSYNC signals, N8RST is reset by the hardware and the N8 signal is set.***

Figure 12-104 Synchronous Output Buffer Register 1
(HOCRBUF1: x'3F7B', R/W)

HOCR Control Register

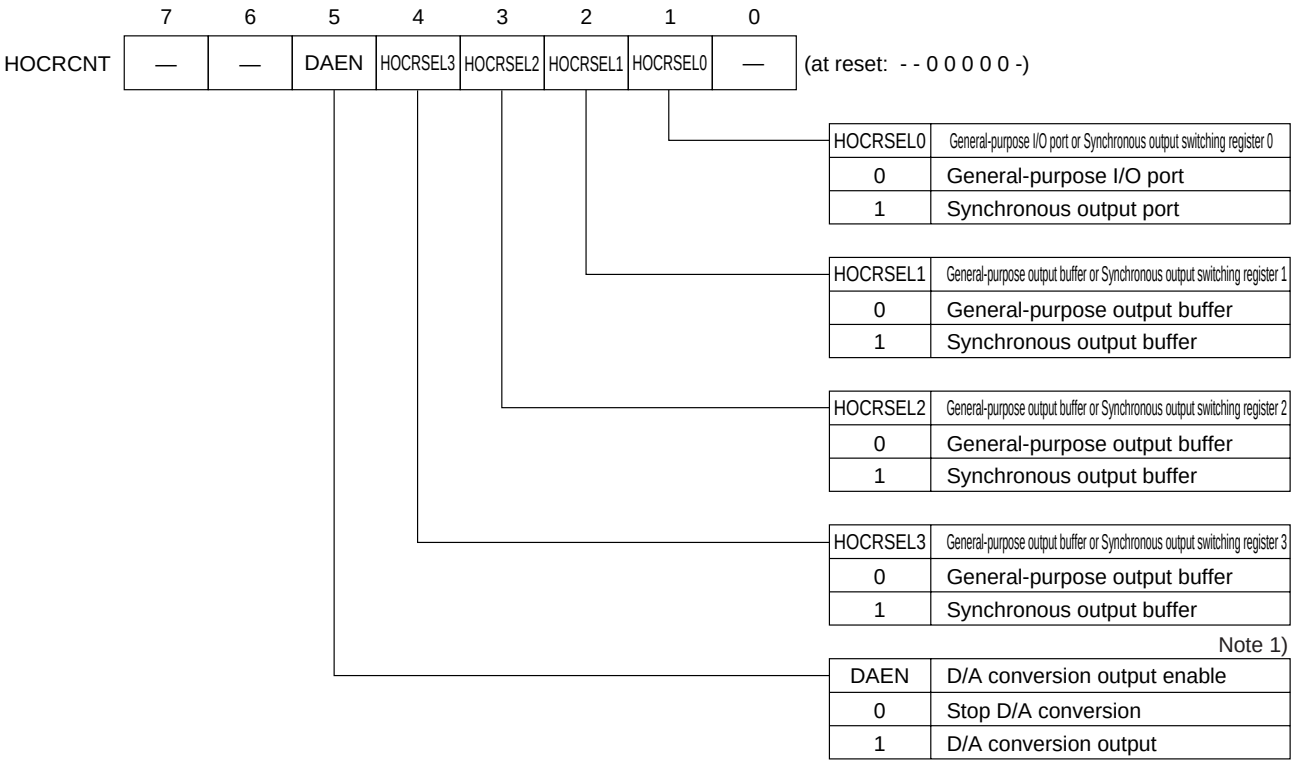


Figure 12-105 HOCR Control Register (HOCRCNT: x'3F7C', R/W)

Note1: Halt the D/A conversion during standby mode (STOP, HALT). When DAEN = '1', the D/A conversion is output regardless of the HOCRSEL0 settings.

RCTL Buffer Register

Internal output of the bit 0 data of HOCRDAT2

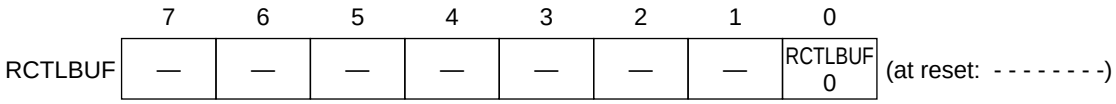


Figure 12-106 RCTL Buffer Register (RCTLBUF: x'3F7D', R/W)

SPG Synchronous Output Data Register

This register sets data for synchronous output to SPGBUF when the contents of the SPGTIM register match those of FRC.

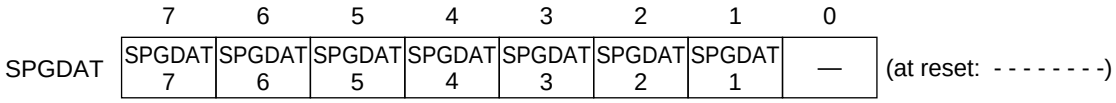


Figure 12-107 SPG Synchronous Output Data Register (SPGDAT: x'3F7E', R/W)

Synchronous Output Control Register

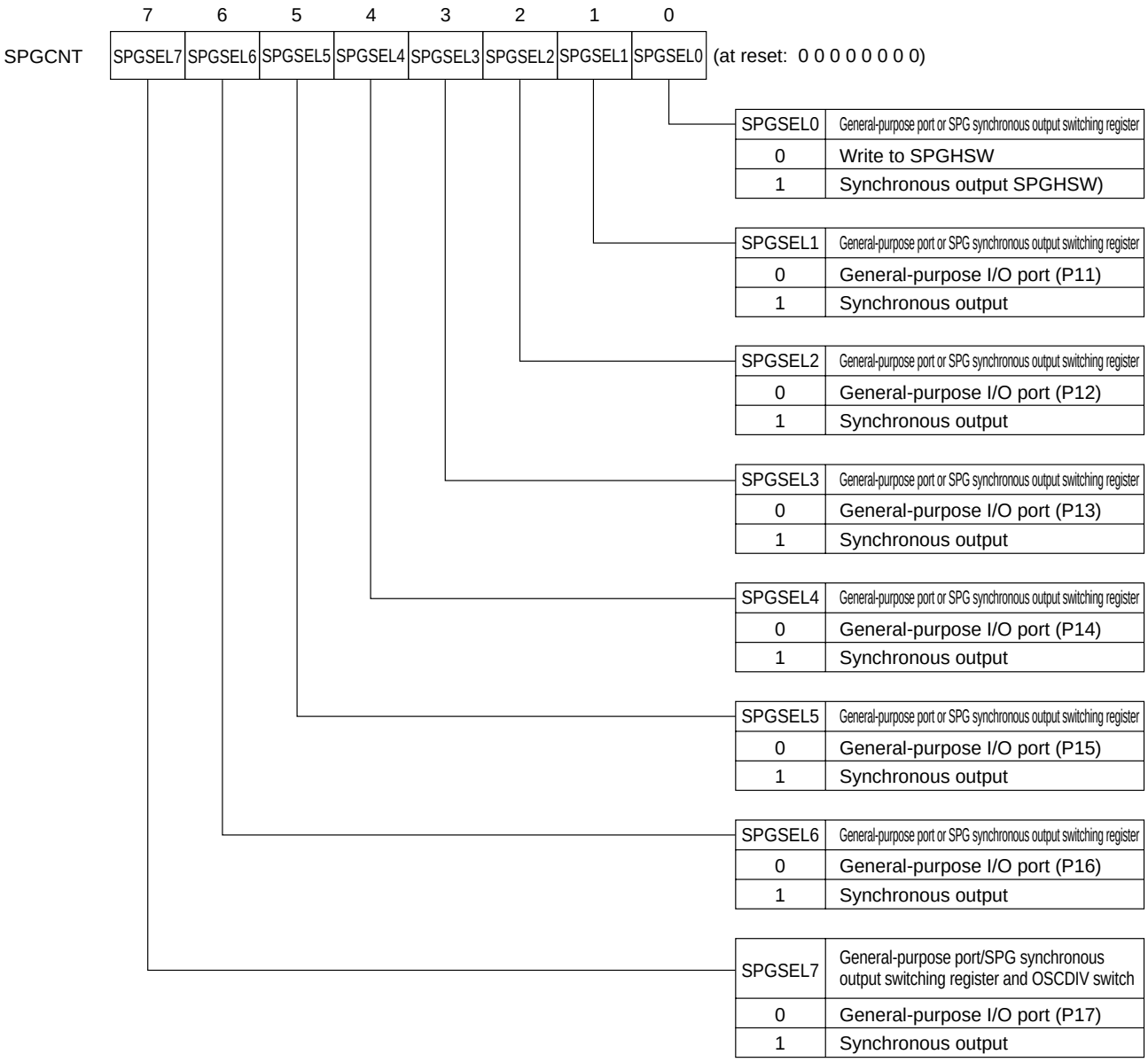


Figure 12-108 Synchronous Output Control Register
(SPGCNT: x'3F7F', R/W)

Read Register for AFG Counter

The count value of the AFG counter can be read.

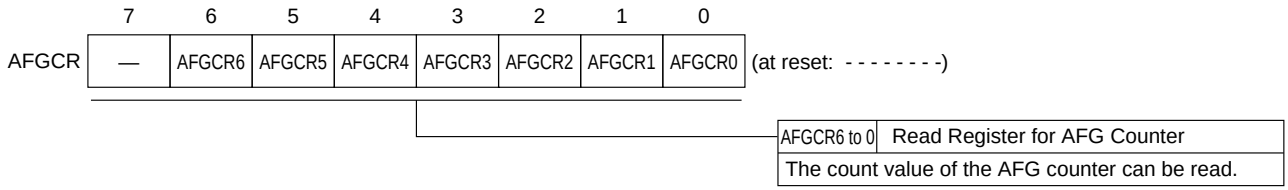
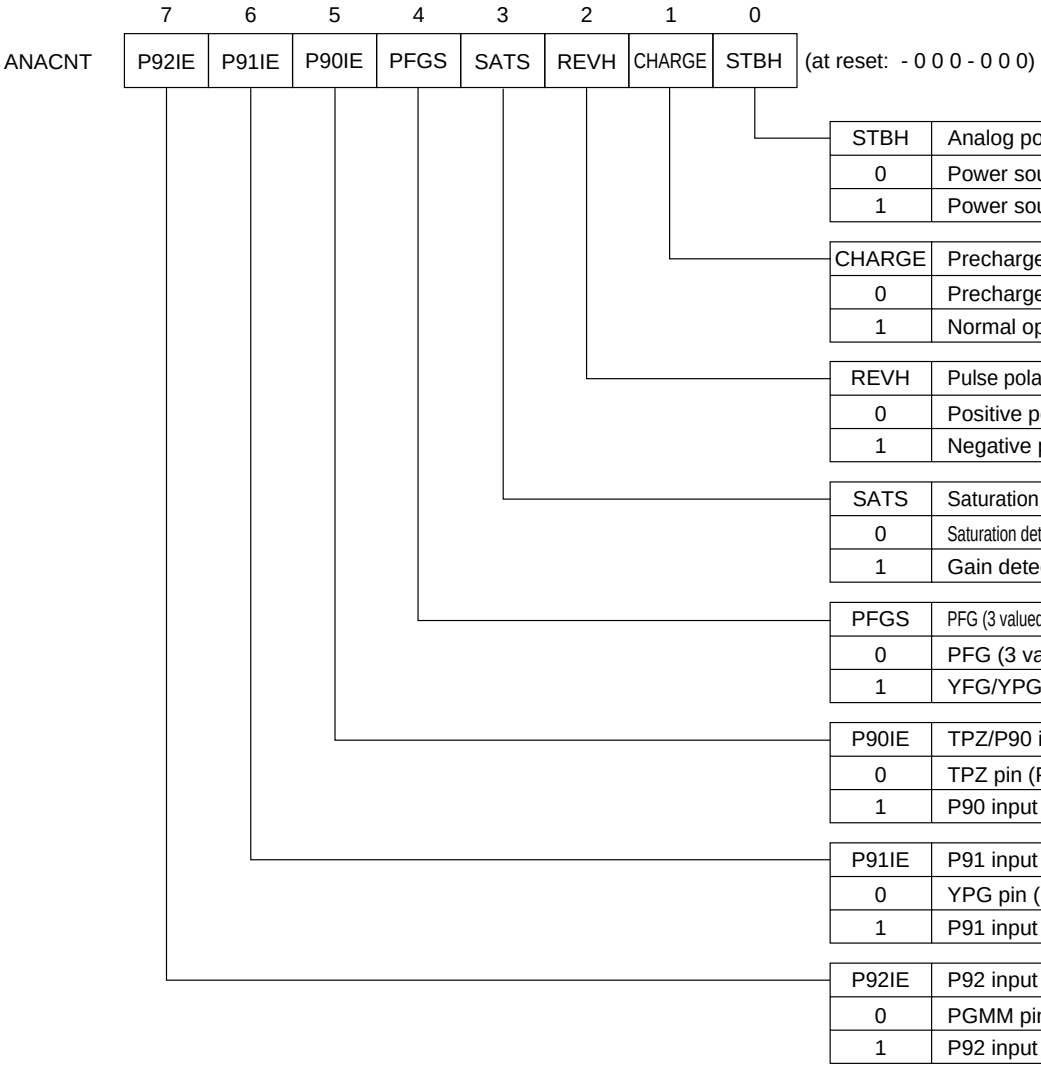


Figure 12-109 Read Register for AFG Counter (AFGCR: x'3F80', R)

Analog Control Register



STBH	Analog power source ON/OFF
0	Power source OFF
1	Power source ON

CHARGE	Precharge control for CTL amp
0	Precharge
1	Normal operation

REVH	Pulse polarity selection for CTL signal
0	Positive polarity (CIF pin input)
1	Negative polarity (CIR pin input)

SATS	Saturation detection function selection
0	Saturation detection function (0.5V/4.5V detection)
1	Gain detection function

PFGS	PFG (3 valued) input and YFG/YPG input switching
0	PFG (3 valued) input
1	YFG/YPG amp input

P90IE	TPZ/P90 input switching
0	TPZ pin (P90 input disabled)
1	P90 input enabled

P91IE	P91 input enable/disable switching
0	YPG pin (P91 input disabled)
1	P91 input enabled

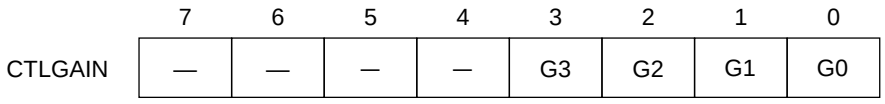
P92IE	P92 input enable/disable switching
0	PGMM pin (P92 input disabled)
1	P92 input enabled

See the description of register TPGCNT0 (x'3F84') for details of the REVH flag.

Do not set the CHARGE flag to '0' (precharge) while the CTL amplifier is operating.

Figure 12-110 Analog Control Register (ANACNT: x'3F81', R/W)

Control Register for Control Amp



G3 to G0	CTL Amplifier gain setting
0 0 0 0	72dB
0 0 0 1	70dB
0 0 1 0	68dB
0 0 1 1	66dB
0 1 0 0	64dB
0 1 0 1	62dB
0 1 1 0	60dB
0 1 1 1	58dB
1 0 0 0	56dB
1 0 0 1	54dB
1 0 1 0	52dB
1 0 1 1	50dB
1 1 0 0	48dB
1 1 0 1	46dB
1 1 1 0	44dB
1 1 1 1	42dB

Figure 12-111 Control Register for Control Amp (CTLGAIN: x'3F82', R/W)

Option Register

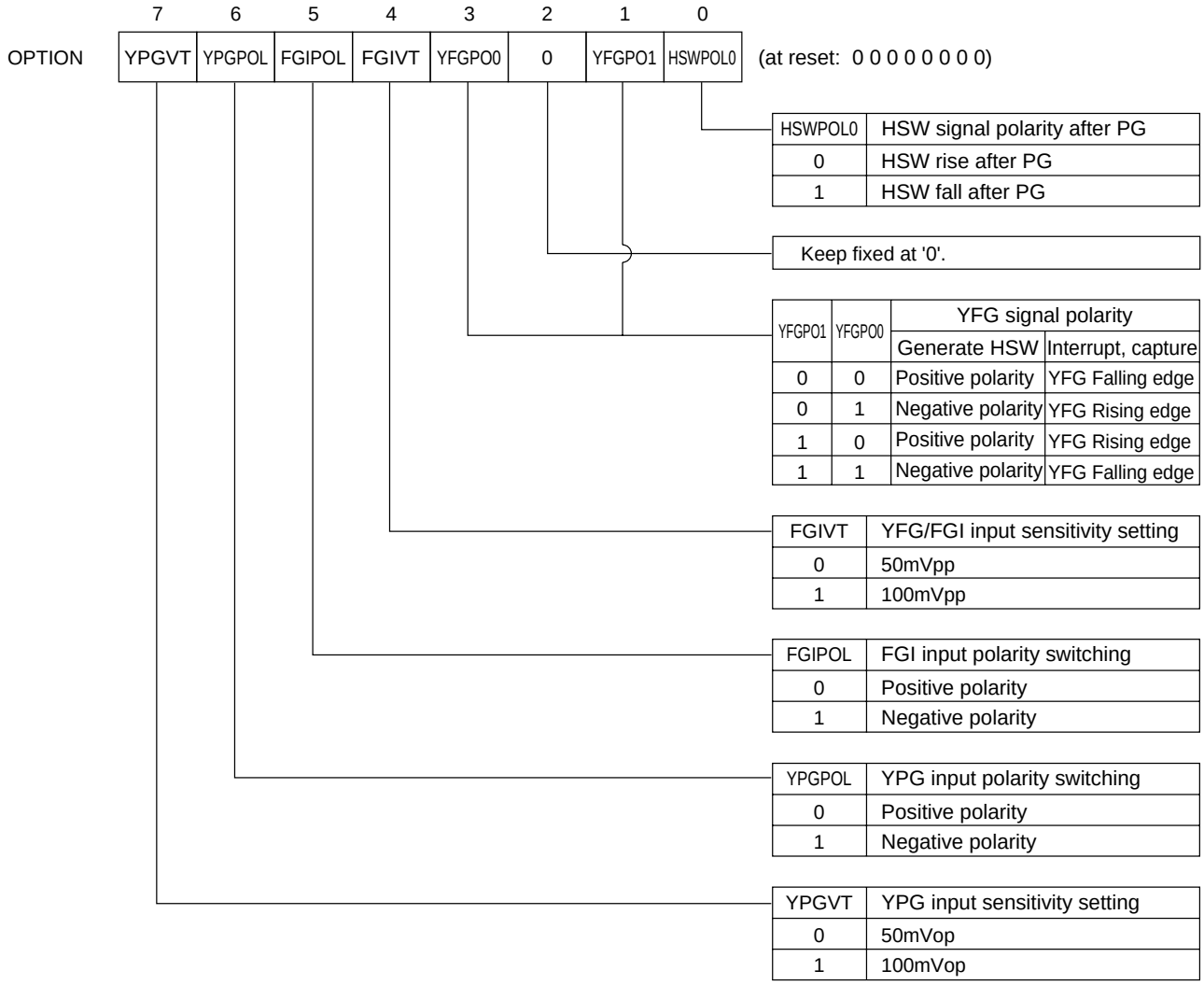


Figure 12-112 Option Register (OPTION: x'3F83', R/W)

Playback Control Register

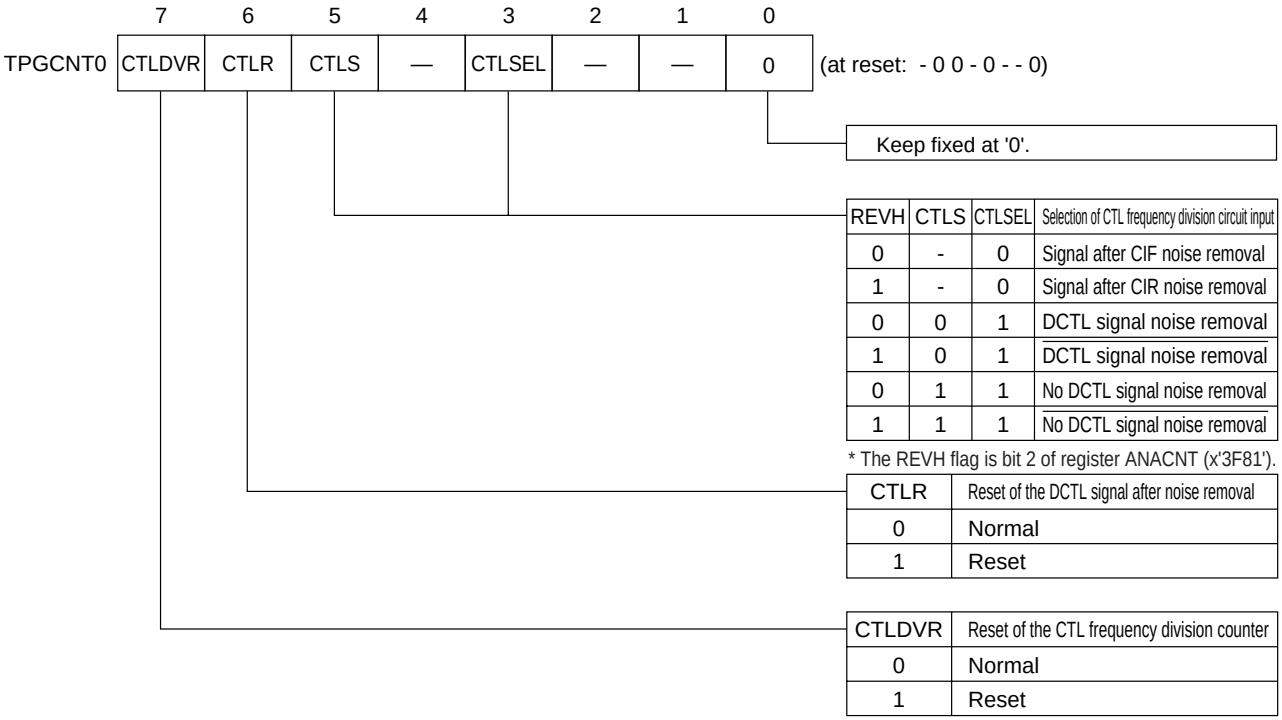


Figure 12-113 Playback Control Register (TPGCNT0: x'3F84', R/W)

TRCNT Register

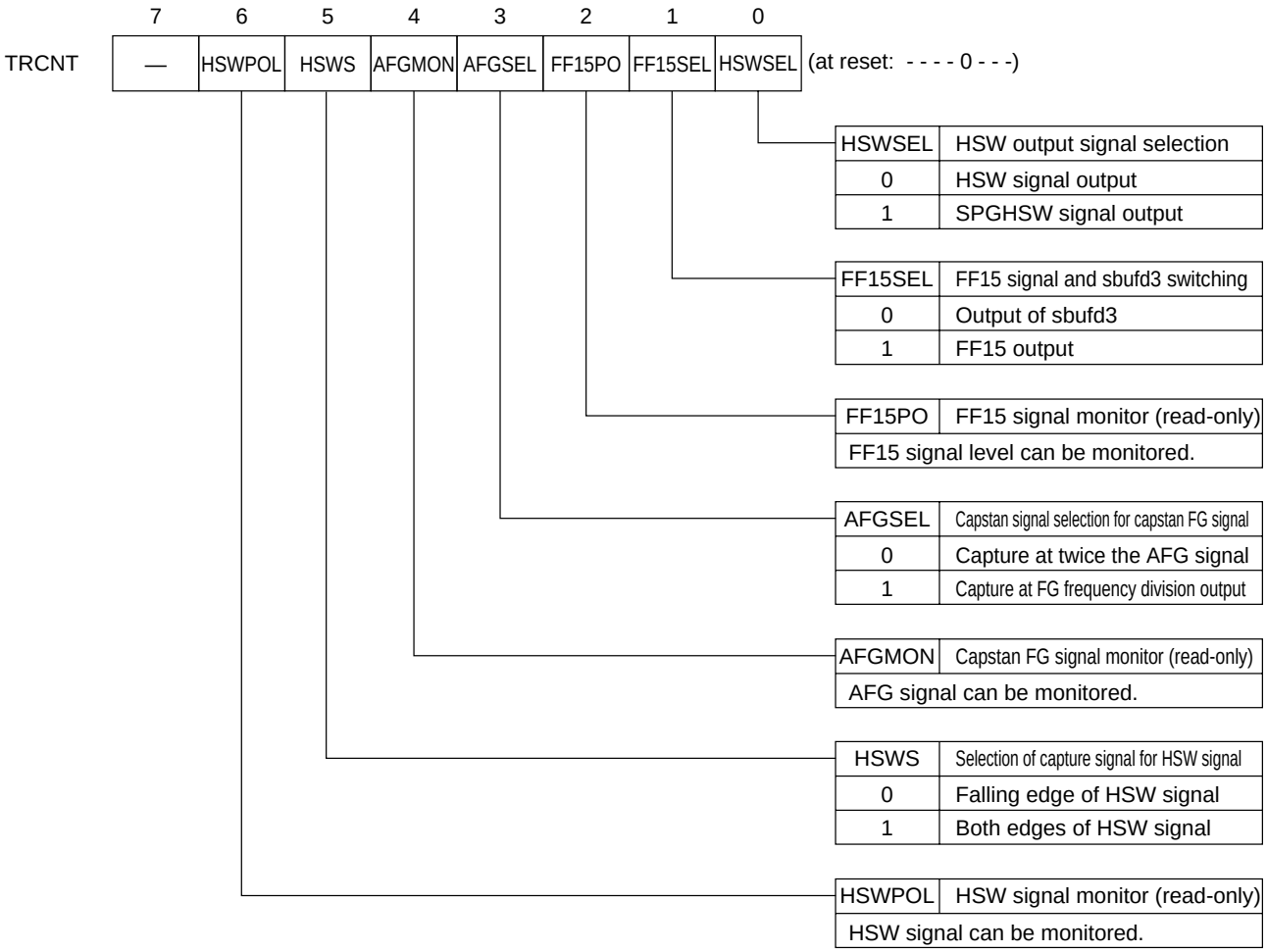
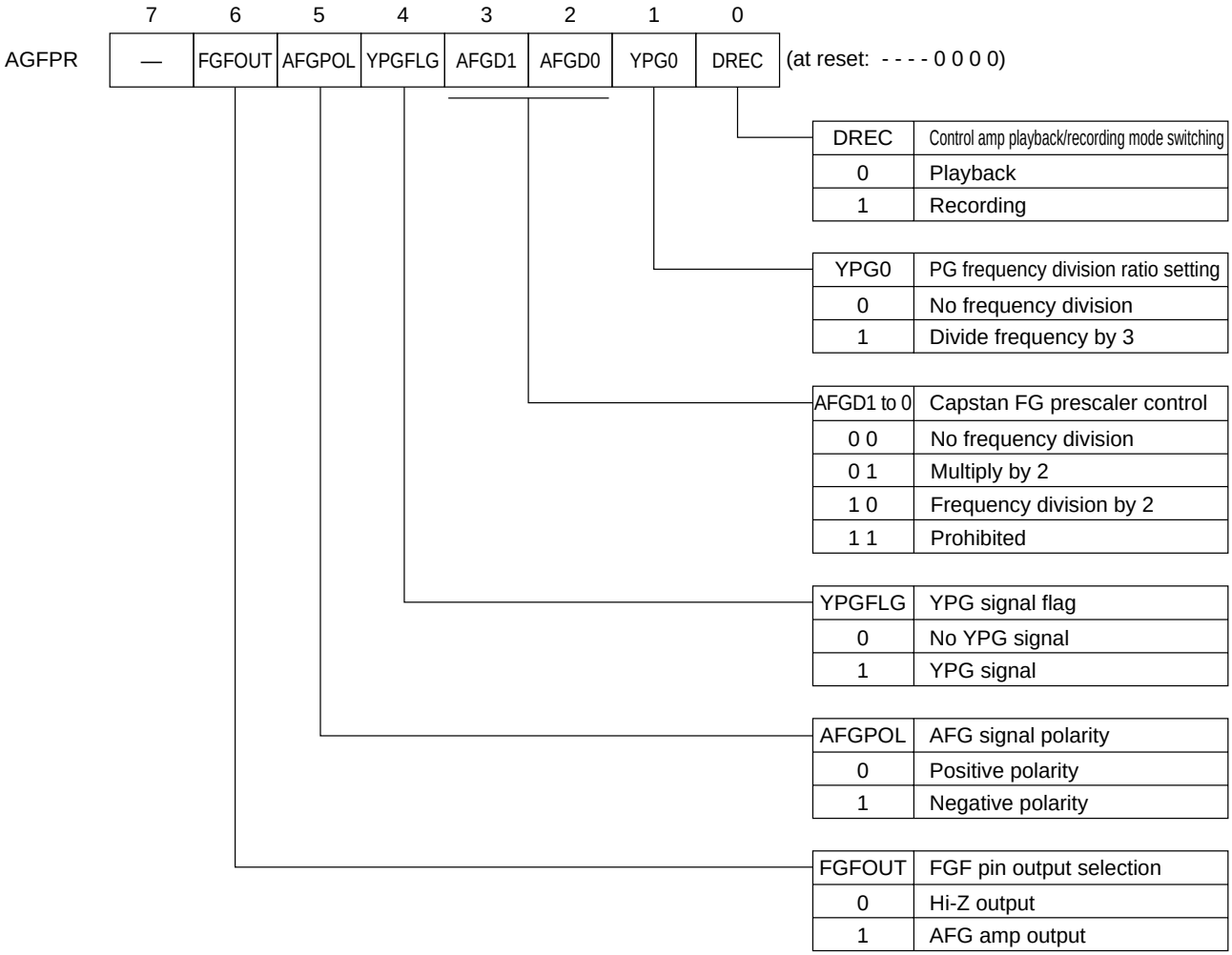


Figure 12-114 TRCNT Register (TRCNT: x'3F85', R/W one section is read-only)

AFG Control Register



The YPG signal flag (YPGFLG) is set by hardware based on the YPG signal. This flag is cleared by writing '0' by software.

Figure 12-115 AFG Control Register (AFGPR: x'3F86', R/W)

AFG Frequency Division Register

Setting value = Frequency division ratio

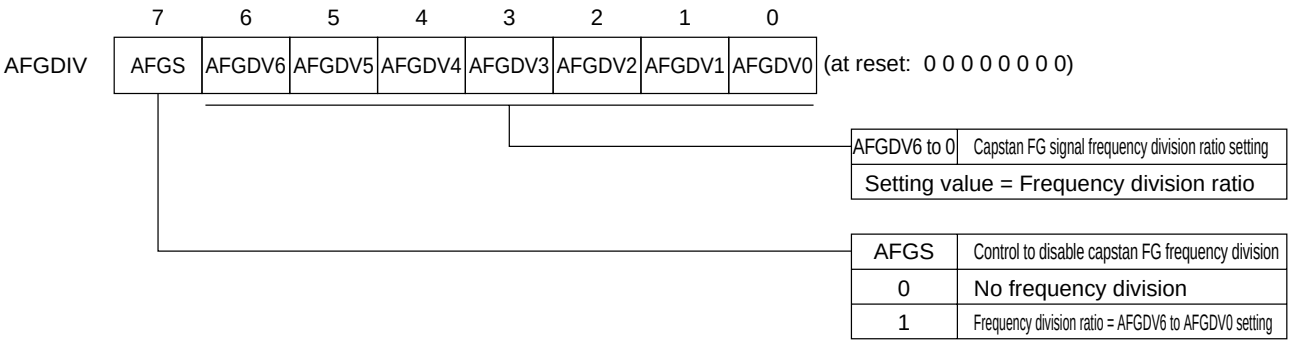


Figure 12-116 AFG Frequency Division Register (AFGDIV: x'3F87', R/W)

YFG Frequency Division Register

Setting value = Frequency division ratio

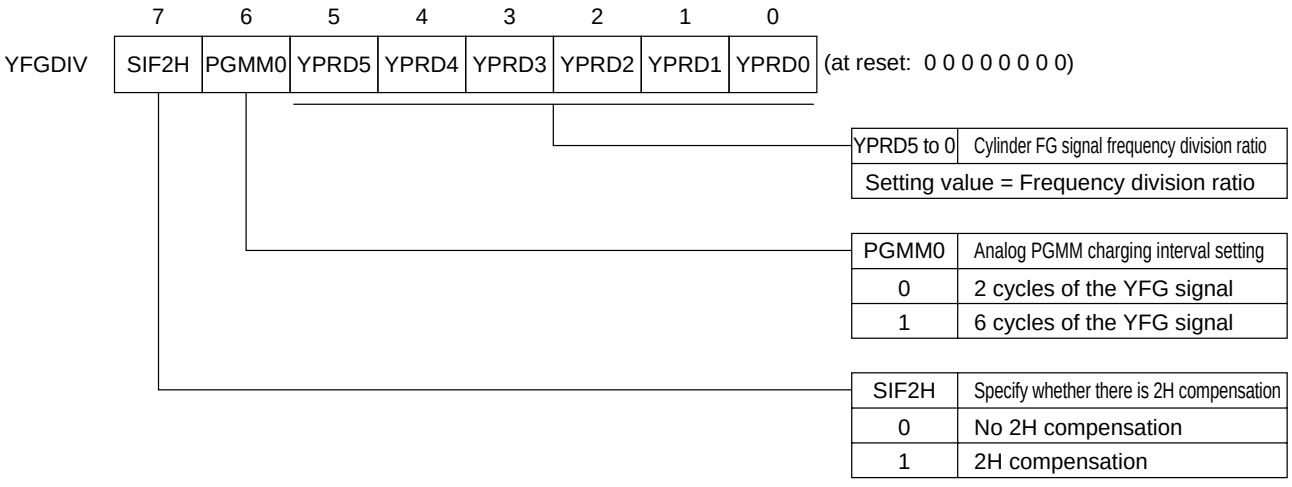


Figure 12-117 YFG Frequency Division Register (YFGDIV: x'3F88', R/W)

CTL Frequency Division Register

Setting value = Frequency division ratio

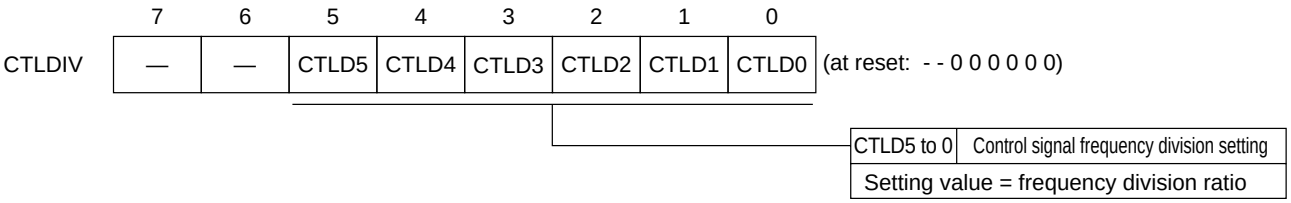


Figure 12-118 CTL Frequency Division Register (CTLDIV: x'3F89', R/W)

PGMM Control Register

When the setting value and YFG counter match, digital PGMM is started.

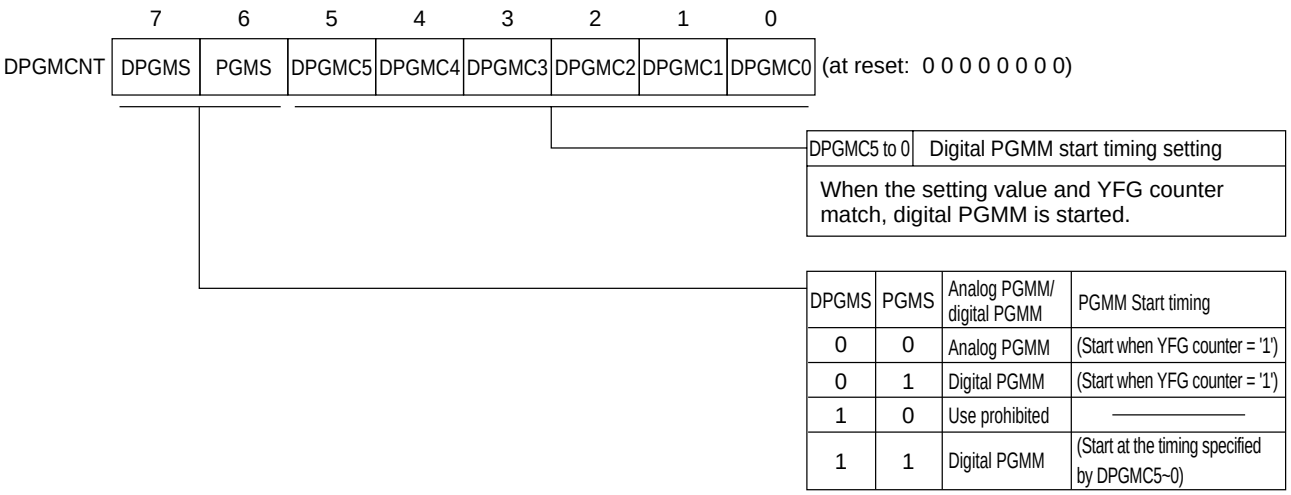


Figure 12-119 PGMM Control Register (DPGMCNT: x'3F8A', R/W)

Digital PGMM Delay Setting Register

Setting value = (count value)

PGMMTIM

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

(at reset: - - - - - - - -)

PGMMTIMH (x'3F8C')

7	6	5	4	3	2	1	0
PGMMTIML 7	PGMMTIML 6	PGMMTIML 5	PGMMTIML 4	PGMMTIML 3	PGMMTIML 2	PGMMTIML 1	PGMMTIML 0

(at reset: - - - - - - - -)

PGMMTIML (x'3F8B')

Use a MOVW instruction to change these bits.

Figure 12-120 Digital PGMM Delay Setting Register
(PGMMTIM: x'3F8C' to x'3F8B', R/W)

Mask Timer Setting Value for Capstan FG Signal

Setting value = (count value)

AMSKTM

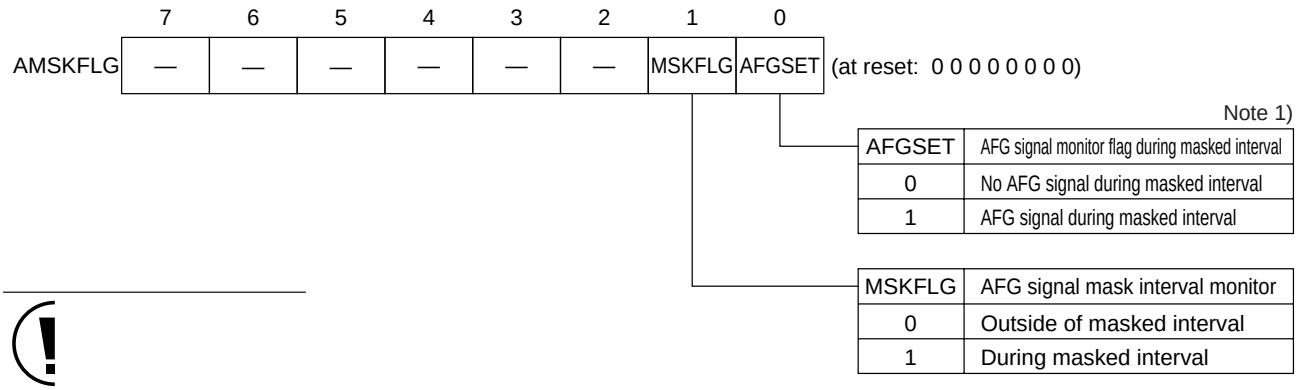
7	6	5	4	3	2	1	0
AMSKTM 7	AMSKTM 6	AMSKTM 5	AMSKTM 4	AMSKTM 3	AMSKTM 2	AMSKTM 1	AMSKTM 0

(at reset: 0 0 0 0 0 0 0 0)

AMSKTM7 to 0	Mask timer setting value for capstan FG signal
Setting value = (count value)	

Figure 12-121 Mask Timer Setting Value for Capstan FG Signal
(AMSKTM: x'3F8D', R/W)

Capstan FG Mask Monitor Register



Note 1: ***This bit is set by the hardware when an AFG signal is input during a masked interval. Clear this bit with the software.***

Figure 12-122 Capstan FG Mask Monitor Register
(AMSKFLG: x'3F8E', R/W one section is read-only)

Pseudo-V Counter Preset Register 0

Sets the reference value when HSW is rising.

	7	6	5	4	3	2	1	0	
AVPR0	AVPR07	AVPR06	AVPR05	AVPR04	AVPR03	AVPR02	AVPR01	AVPR00	(at reset: - - - - - - - -)

Figure 12-123 Pseudo-V Counter Preset Register 0
(AVPR0: x'3F8F', R/W)

Pseudo-V Counter Preset Register 1

Sets the reference value when HSW is falling.

	7	6	5	4	3	2	1	0	
AVPR1	AVPR17	AVPR16	AVPR15	AVPR14	AVPR13	AVPR12	AVPR11	AVPR10	(at reset: - - - - - - - -)

Figure 12-124 Pseudo-V Counter Preset Register 1
(AVPR1: x'3F90', R/W)

Pseudo-V Counter Output Compare Register 0

Sets the rising edge of the Pseudo-V pulse.

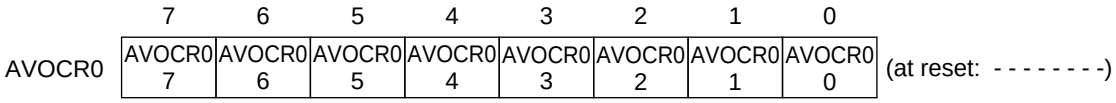


Figure 12-125 Pseudo-V Counter Output Compare Register 0
(AVOCR0: x'3F91', R/W)

Pseudo-V Counter Output Compare Register 1

Sets the falling edge of the Pseudo-V pulse interval.

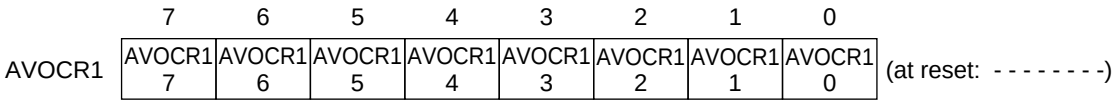


Figure 12-126 Pseudo-V Counter Output Compare Register 1
(AVOCR1: x'3F92', R/W)

Pseudo-V Control Register

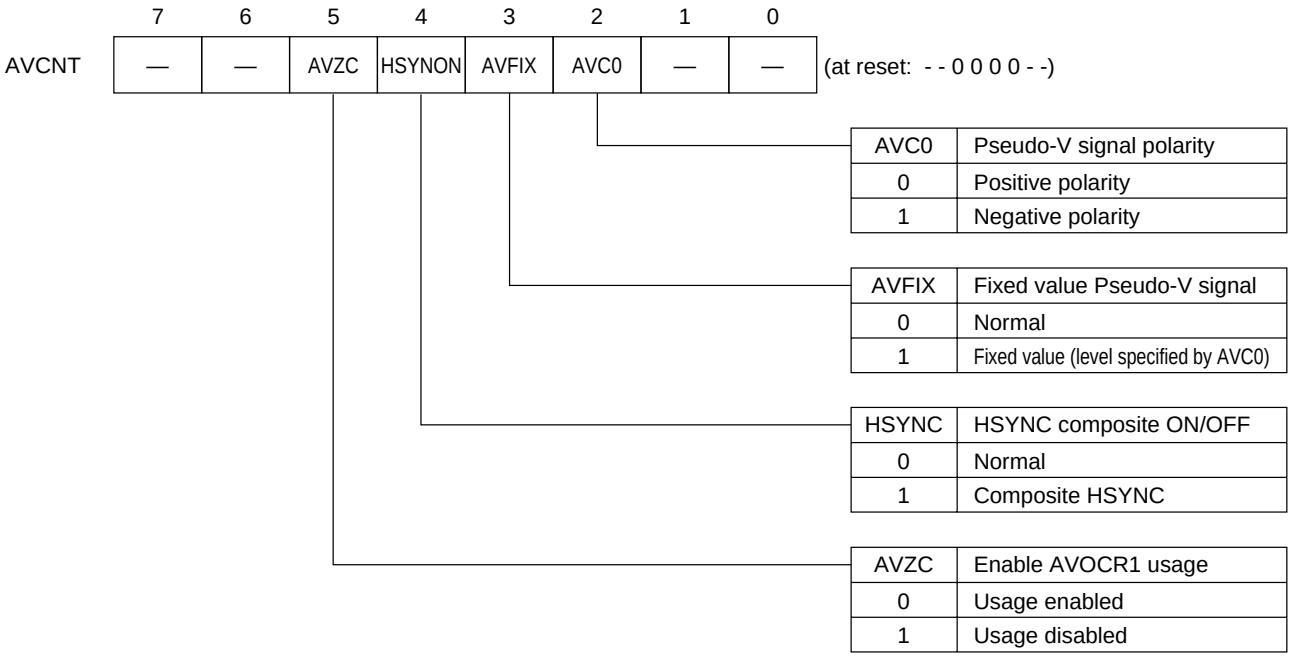
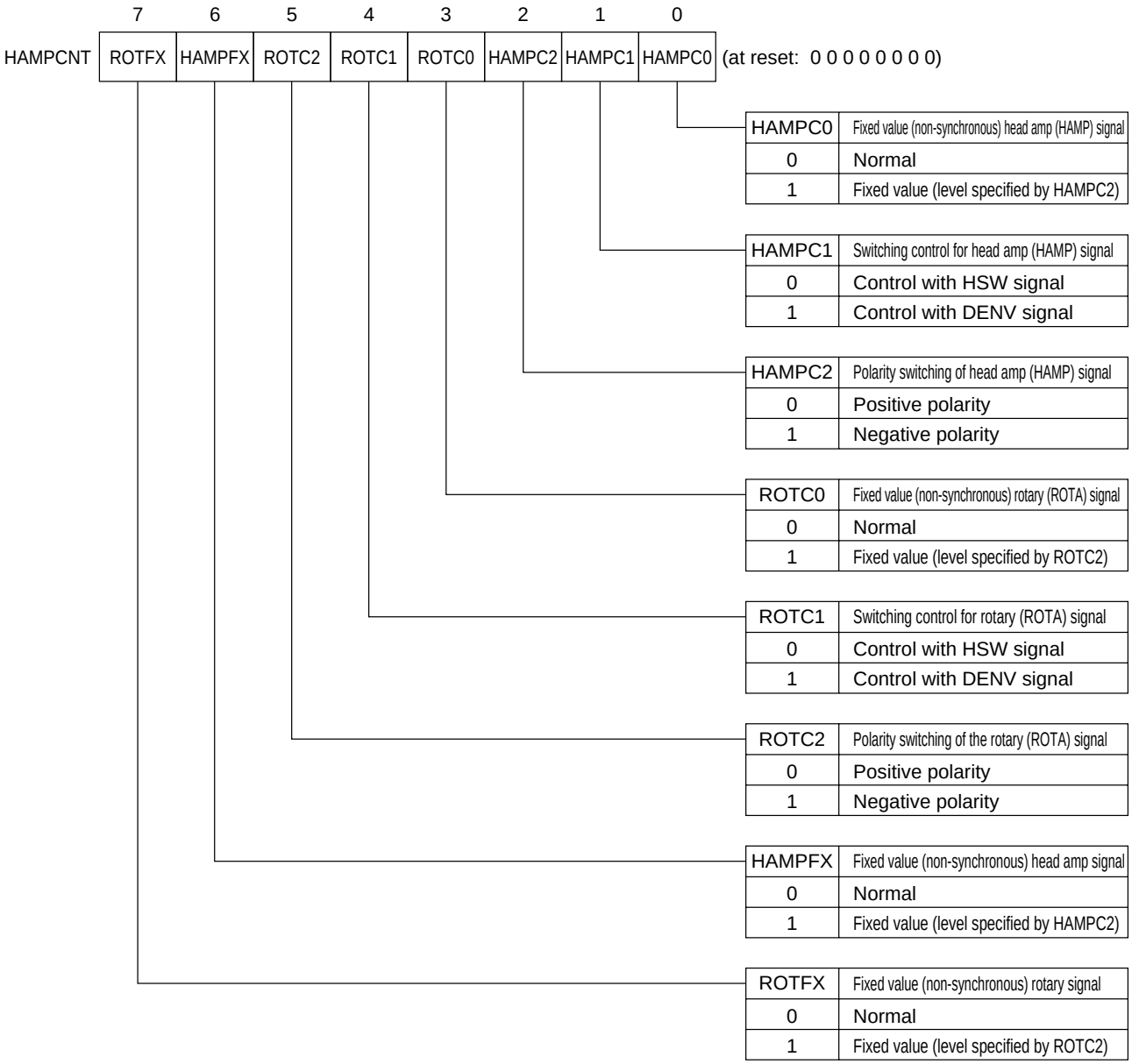


Figure 12-127 Pseudo-V Control Register (AVCNT: x'3F93', R/W)

Head Amp/Rotary Control Register



The HAMPFX and ROTFX flags are not synchronized with the HSW signal.

Figure 12-128 Head Amp/Rotary Control Register (HAMPCNT: x'3F94', R/W)

Serial Interface 0 Send-Receive Shift Register

Transmission data can be set by using a MOV instruction to write to this register.

	7	6	5	4	3	2	1	0	
SC0TRB	SC0TRB7	SC0TRB6	SC0TRB5	SC0TRB4	SC0TRB3	SC0TRB2	SC0TRB1	SC0TRB0	(at reset: - - - - - - - -)

Figure 12-129 Serial Interface 0 Send-Receive Shift Register
(SC0TRB: x'3F95', W)

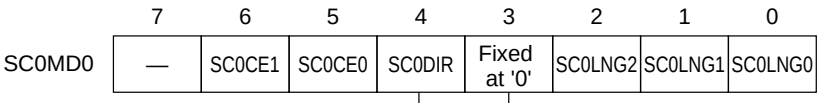
Serial Interface 0 Receive Data Buffer Register

Received data can be obtained by using a MOV instruction to read this register.

	7	6	5	4	3	2	1	0	
SC0RXB	SC0RXB7	SC0RXB6	SC0RXB5	SC0RXB4	SC0RXB3	SC0RXB2	SC0RXB1	SC0RXB0	(at reset: - - - - - - - -)

Figure 12-130 Serial Interface 0 Receive Data Buffer Register
(SC0RXB: x'3F96', R)

Serial Interface 0 Mode Register 0



SC0LNG2 to 0	Number of bits for transfer
0 0 0	8 bits
0 0 1	7 bits
0 1 0	6 bits
0 1 1	5 bits
1 0 0	4 bits
1 0 1	3 bits
1 1 0	2 bits
1 1 1	1 bit

Fix at '0'

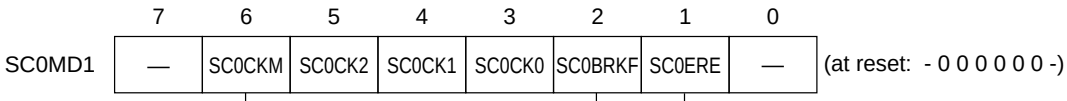
SC0DIR	Specification of the first bit during transfers
0	MSB first
1	LSB first

SC0CE0 to 1	Receive data input edge	Transmit data output edge
0 0	Rising	Falling
0 1	Falling	Rising
1 0	Falling	Falling
1 1	Rising	Rising

The input edge specification is valid only when the SC0CKM flag of register SC0MD1 (x'3F98') is set to '0', thereby making the clock source other than the 1/8 frequency signal. This specification cannot be used when the SC0CKM flag is set to '1', thereby making the 1/8 frequency signal the clock source.

Figure 12-131 Serial Interface 0 Mode Register 0
(SC0MD0: x'3F97', R/W)

Serial Interface 0 Mode Register 1



SC0ERE	Error monitor
0	No errors
1	Error

SC0BRKF	Break status reception monitor
0	Data
1	Break

SC0CK2 to 0	Clock source specification
0 0 0	fs/4
0 0 1	fs/8
0 1 0	fs/16
0 1 1	fs/32
1 0 0	fs/64
1 0 1	fs/128
1 1 0	fs/256
1 1 1	ftc4/2

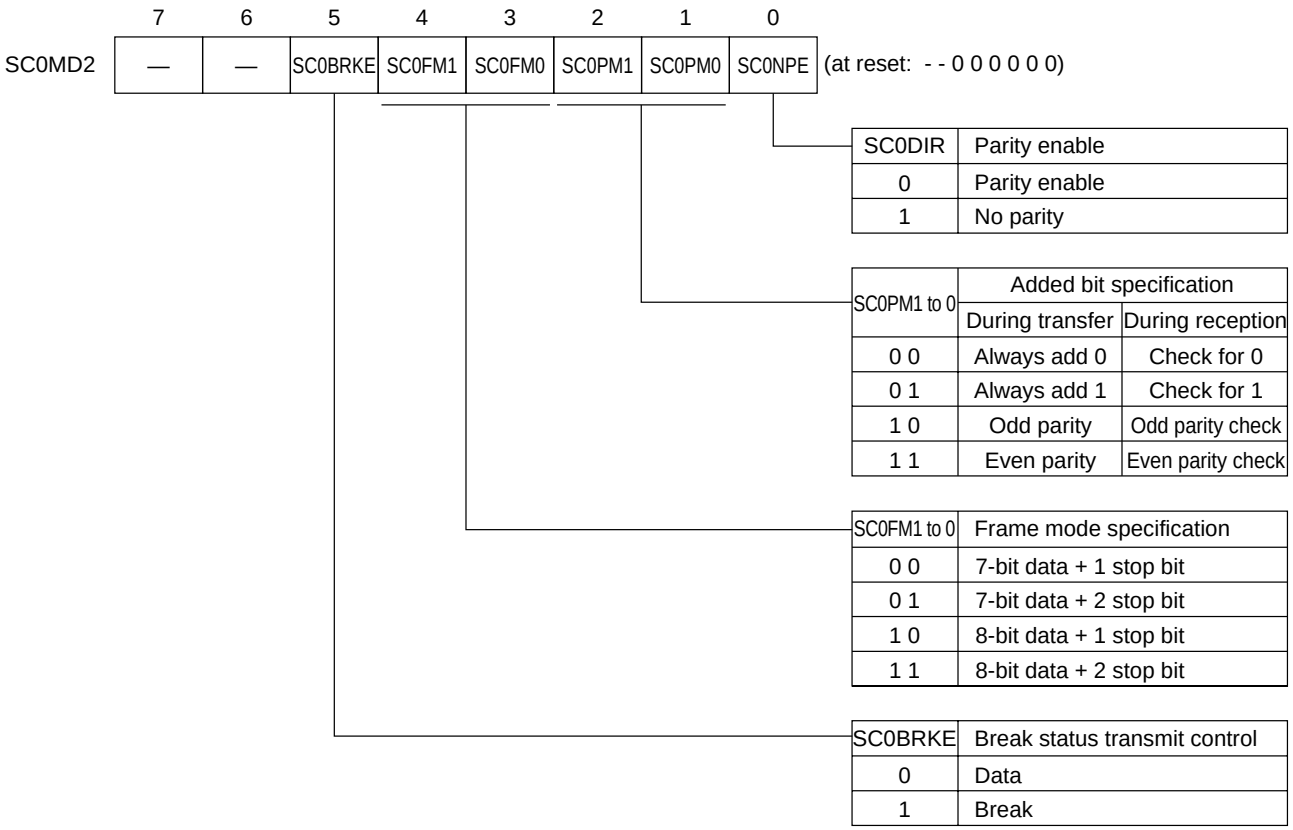
SC0CKM	Selection of frequency division by 8
0	Do not divide frequency by 8
1	Divide frequency by 8

If using an external clock as the clock source, set the SBT0 pin to input mode.

If using with the UART, frequency division by 8 must be selected.

Figure 12-132 Serial Interface 0 Mode Register 0
(SC0MD1: x'3F98', R/W)

Serial Interface 0 Mode Register 2



SC0DIR	Parity enable
0	Parity enable
1	No parity

SC0PM1 to 0	Added bit specification	
	During transfer	During reception
0 0	Always add 0	Check for 0
0 1	Always add 1	Check for 1
1 0	Odd parity	Odd parity check
1 1	Even parity	Even parity check

SC0FM1 to 0	Frame mode specification
0 0	7-bit data + 1 stop bit
0 1	7-bit data + 2 stop bit
1 0	8-bit data + 1 stop bit
1 1	8-bit data + 2 stop bit

SC0BRKE	Break status transmit control
0	Data
1	Break

Figure 12-133 Serial Interface 0 Mode Register 2 (SC0MD2: x'3F99', R/W)

Serial Interface 0 Mode Register 3

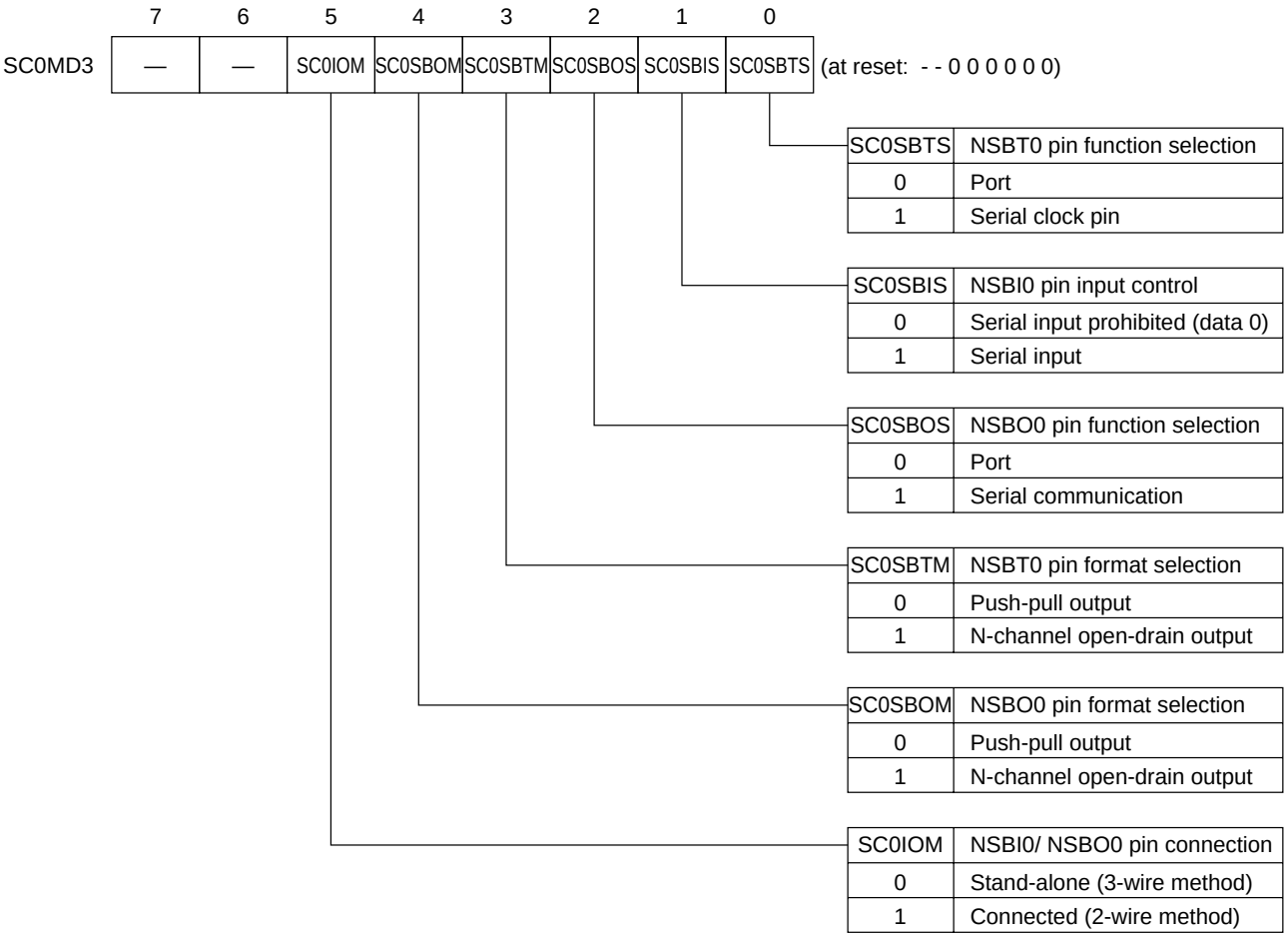


Figure 12-134 Serial Interface 0 Mode Register 3
(SC0MD3: x'3F9A', R/W)

Serial Interface 0 Control Register

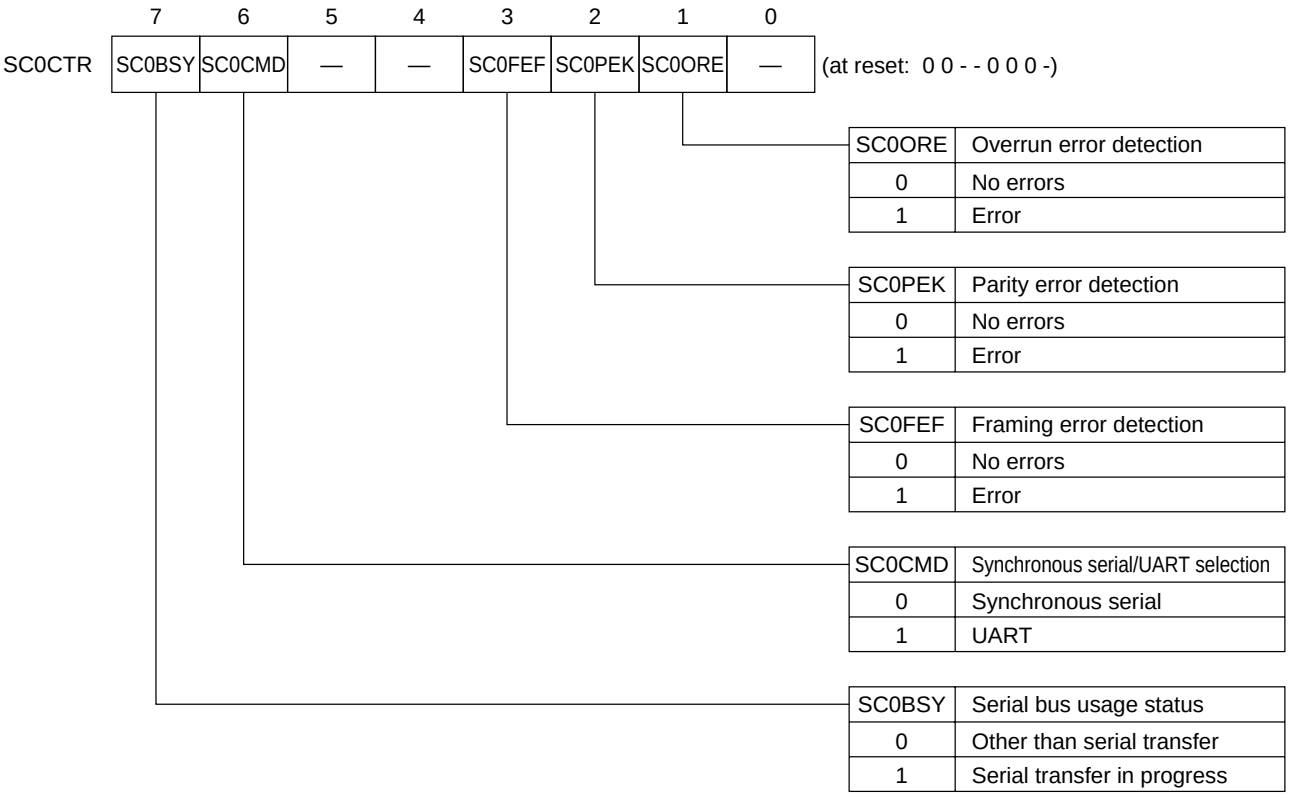


Figure 12-135 Serial Interface 0 Control Register
(SC0CTR: x'3F9B', R/W)

Serial 1 Mode Register

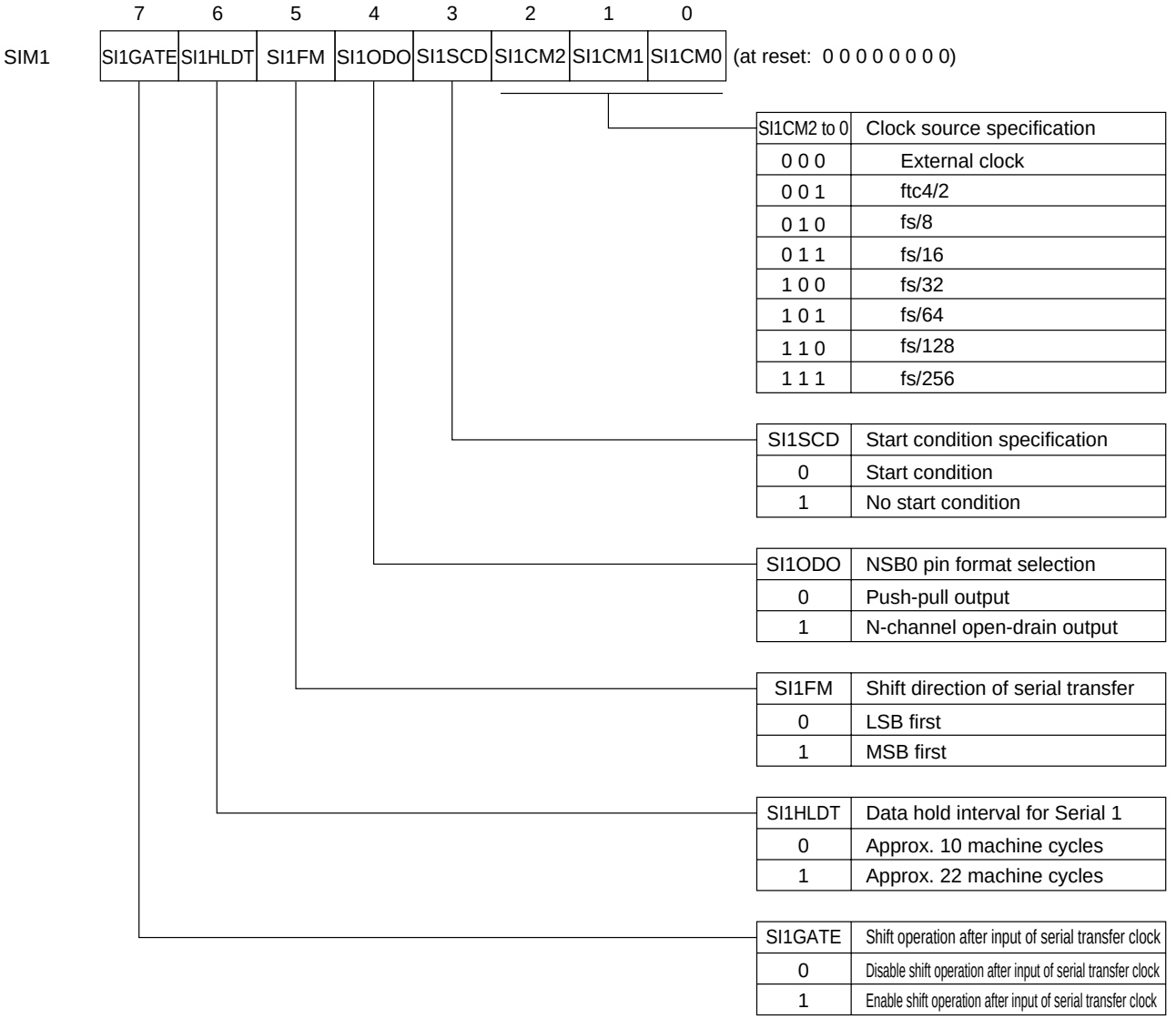


Figure 12-136 Serial 1 Mode Register (SIM1: x'3F9C', R/W)

If an external clock is to be set, configure the P07 (SBT1) pin as an input.

If the clock source is fs/8 or fs/16 and an external clock faster than fs/16 is to be used, the data hold interval must be set to approximately 10 machine cycles.

Serial Interface 1 Transmit-Receive Shift Buffer Register

Transmission data can be set by writing to this register with a MOV instruction. The received data can be read with a MOV instruction.

	7	6	5	4	3	2	1	0	
SIBUF1	SIBUF17	SIBUF16	SIBUF15	SIBUF14	SIBUF13	SIBUF12	SIBUF11	SIBUF10	(at reset: - - - - - - - -)

Figure 12-137 Serial Interface 1 Transmit-Receive Shift Buffer Register (SIBUF1: x'3F9D', R/W)

Serial 1 Control Register

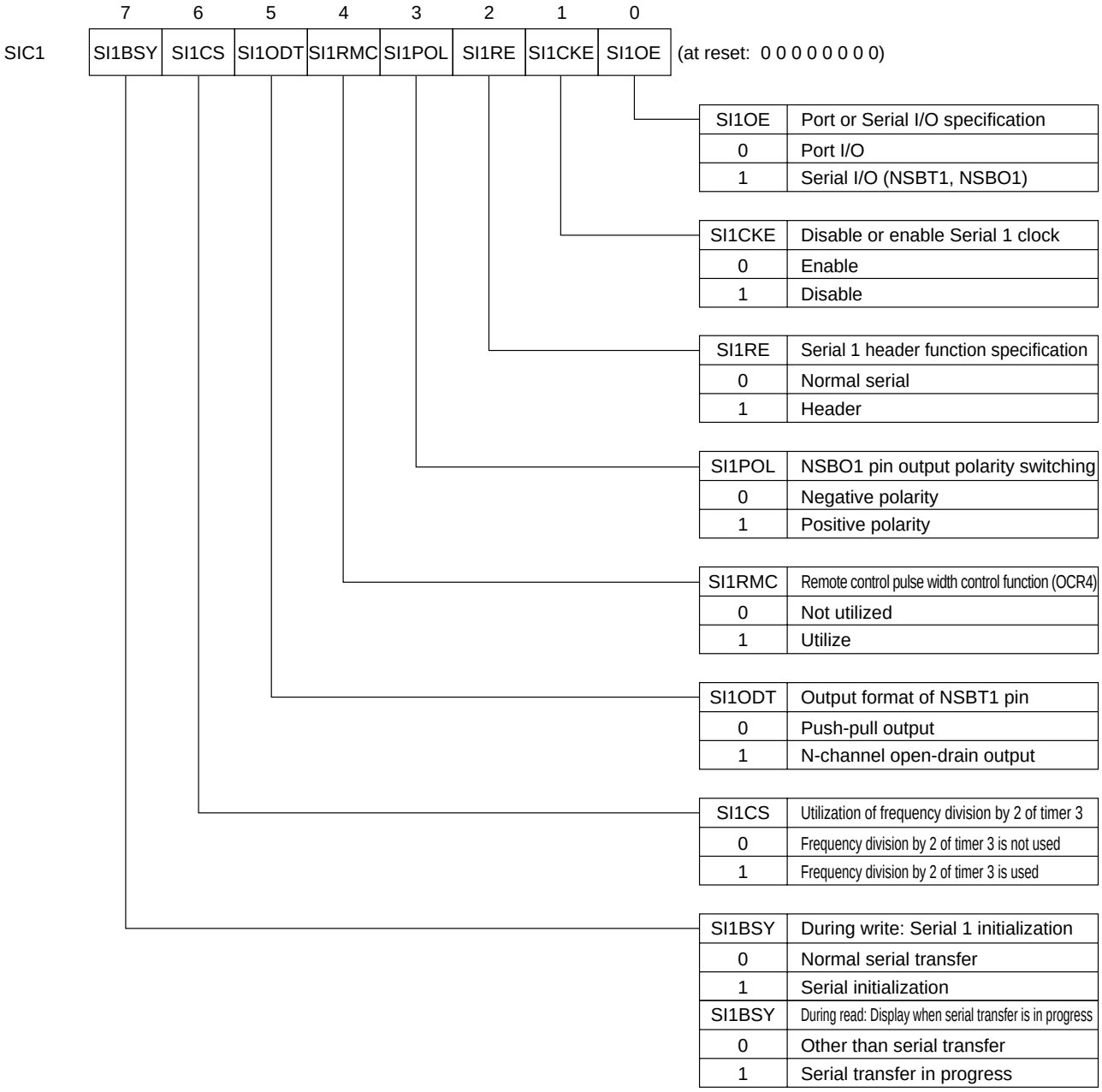


Figure 12-138 Serial 1 Control Register (SIC1: x'3F9E', R/W)

If the BUSY flag (bp7) is set for serial initialization, this bit will automatically be reset for normal serial operation.

Serial Interface 2 Data Shift Register

Transmission data can be set by writing to this register with a MOV instruction. The received data can be read with a MOV instruction.

	7	6	5	4	3	2	1	0	
I2CR0	I2CR7	I2CR6	I2CR5	I2CR4	I2CR3	I2CR2	I2CR1	I2CR0	(at reset: 0 0 0 0 0 0 0 0)

If the ESO flag (bit 3) of the I2CR1 register is '1', reading and writing to the I2CR0 register is possible.

Figure 12-139 Serial Interface 2 Data Shift Register
(I2CR0: x'3F9F', R/W)

Serial Interface 2 Address Register

This register stores the slave address assigned to a device.

	7	6	5	4	3	2	1	0	
I2CR0'	ADRL6	ADRL5	ADRL4	ADRL3	ADRL2	ADRL1	ADRL0	ALS	(at reset: 0 0 0 0 0 0 0 0)

ALS	Slave address recognition
0	Recognize (addressing format)
1	Do not recognize (free data format)

ADRL6 to 0	Serial Interface 2 Address Register
This register stores the slave address assigned to a device.	

IIf the ESO flag (bit 3) of the I2CR1 register is '0', writing to the I2CR0 register is possible.

Figure 12-140 Serial Interface 2 Address Register
(I2CR0': x'3F9F', W)

Status Register

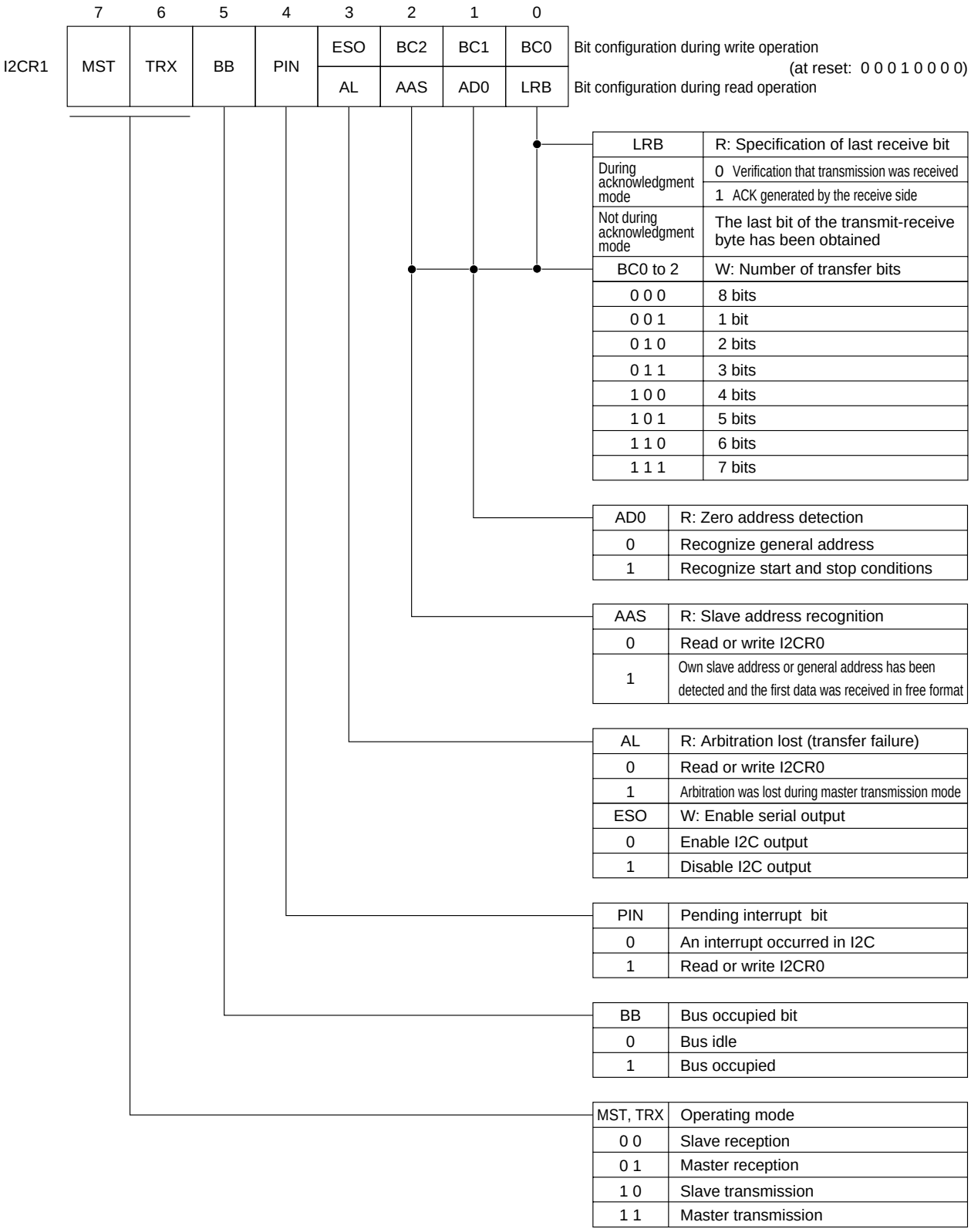


Figure 12-141 Status Register (I2CR1: x'3FA0', R/W)

Clock Control Register

	7	6	5	4	3	2	1	0	
I2CR2	—	ACK	0	I2CR24	I2CR23	I2CR22	I2CR21	I2CR20	(at reset: 0 0 0 0 0 0 0 0)

CLKSEL	I2CR24 to 0	Clock source specification
1	x'09'	fs/72
0	x'10'	fs/80
1	x'0A'	fs/84
0	x'11'	fs/96
1	x'0B'	fs/102
0	x'12'	fs/112
1	x'0C'	fs/120
0	x'13'	fs/128
1	x'0D'	fs/144
0	x'14'	fs/160
1	x'0E'	fs/168
0	x'15'	fs/192
0	x'16'	fs/224
0	x'17'	fs/256
0	x'18'	fs/320

* The CLKSEL flag is bit 7 of register IC2DIR (x'3FA3').

0	HIGH/LOW ratio specification for SCK
Must be set to '0'. When '0', the SCK HIGH/LOW ratio is 1:1.	

ACK	Acknowledge mode specification
0	Invalid
1	Valid

Figure 12-142 Clock Control Register (I2CR2: x'3FA1', W)

I²C System Control Register

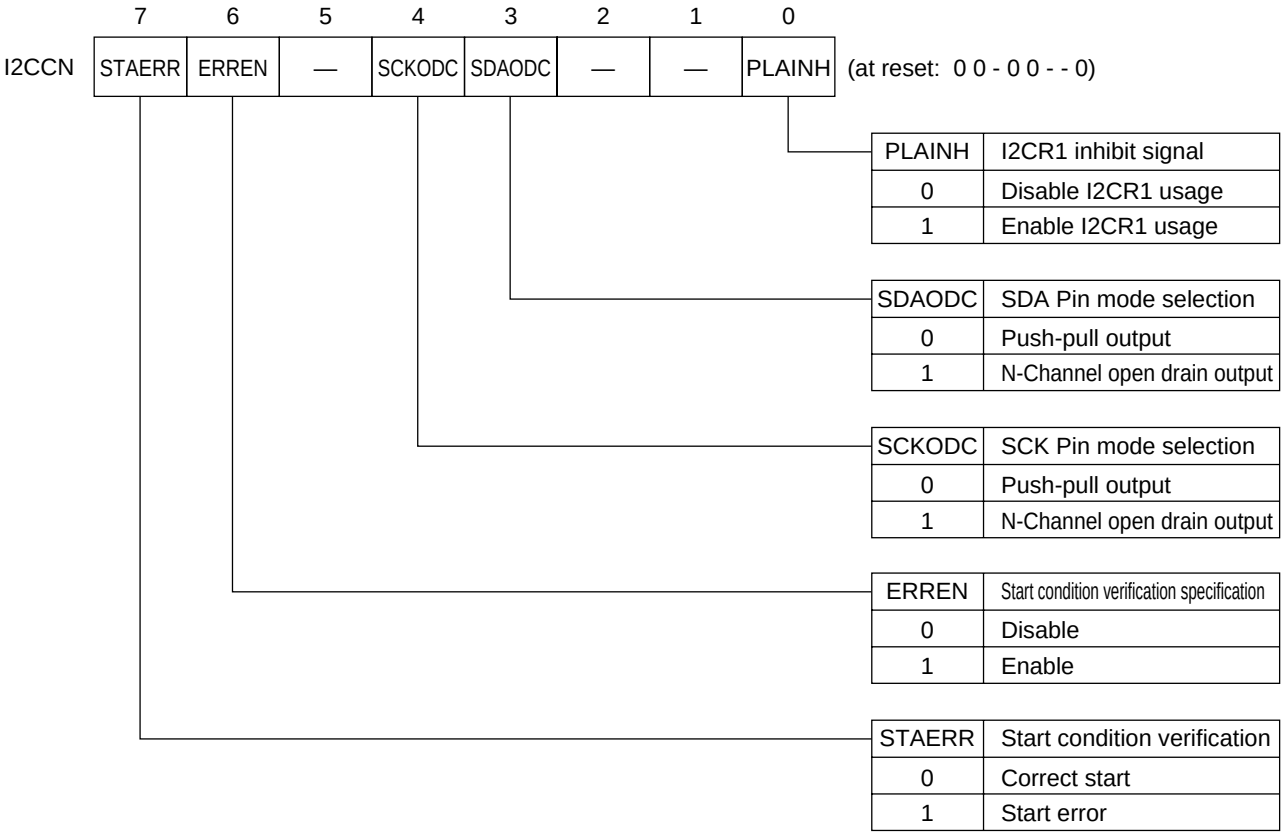


Figure 12-143 I²C System Control Register (I2CCN: x'3FA2', R/W)

I²C Direction Control Register

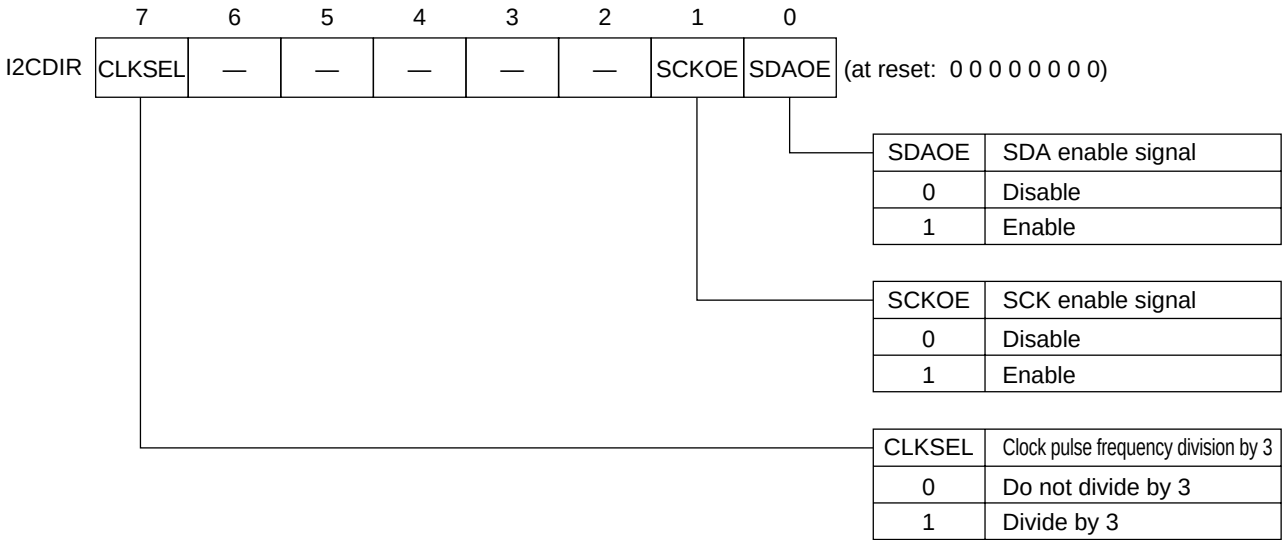
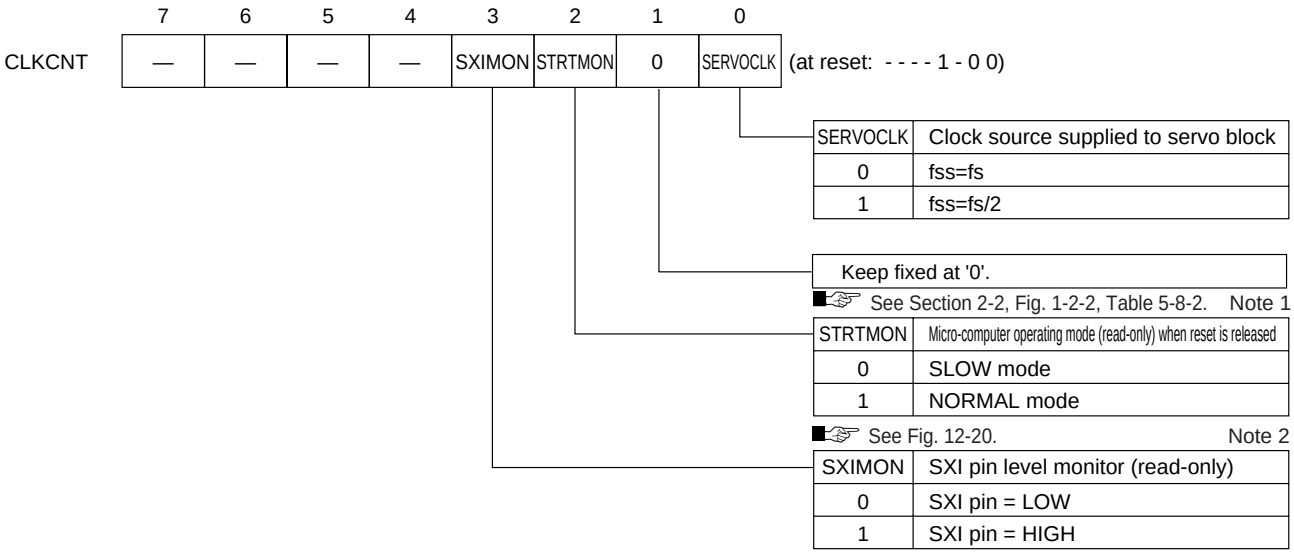


Figure 12-144 I²C Direction Control Register (I2CDIR: x'3FA3', R/W)

Oscillation Control Register

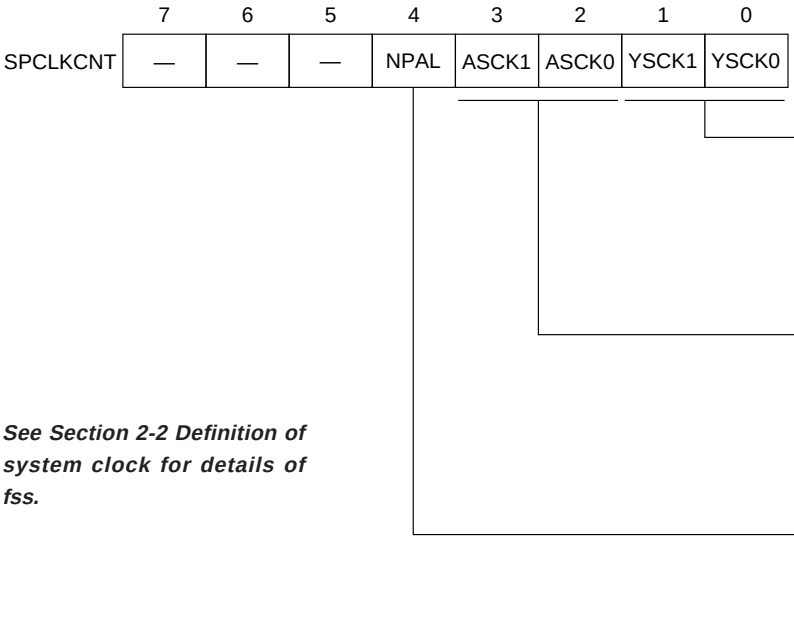


Note 1: ***This chip is ordinarily '1' (start in NORMAL mode).***

Note 2: ***The SXI pin level is latched when reset is released. This level is stored until the next reset release operation.***

Figure 12-145 Oscillation Control Register
(CLKCNT: x'3FA4', R/W one section is read-only)

Cylinder and Capstan Speed Counter Control Register



YSCK1	YSCK0	Changing the cylinder output counter and clock
0	0	fss/2
0	1	fss/4
1	0	fss/8
1	1	Prohibited

ASCK1	ASCK0	Changing the capstan output counter and clock
0	0	fss/4
0	1	fss/8
1	0	fss/16
1	1	fss/32

See Section 2-2 Definition of system clock for details of fss.

NPAL	Changing the 2H compensation value
0	1024/fss
1	1016/fss

Figure 12-146 Cylinder and Capstan Speed Counter Control Register (SPCLKCNT: x'3FA5', R/W)

YFG Speed Error Register

Sets the cylinder error data.

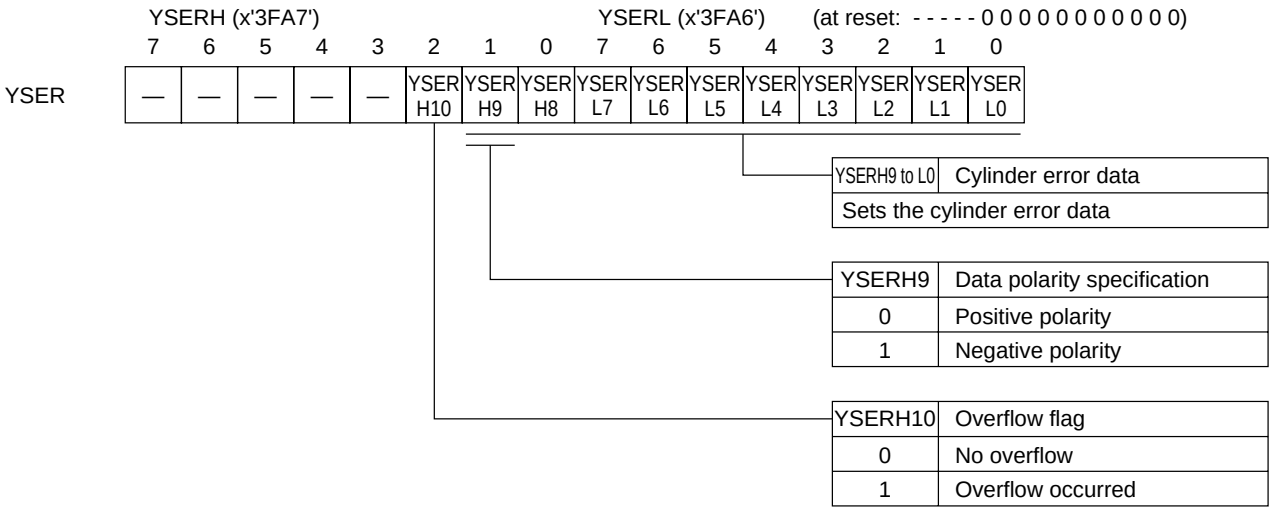


Figure 12-147 YFG Speed Error Register (YSER: x'3FA7' to x'3FA6', R)

AFG Speed Error Register

Sets the capstan error data.

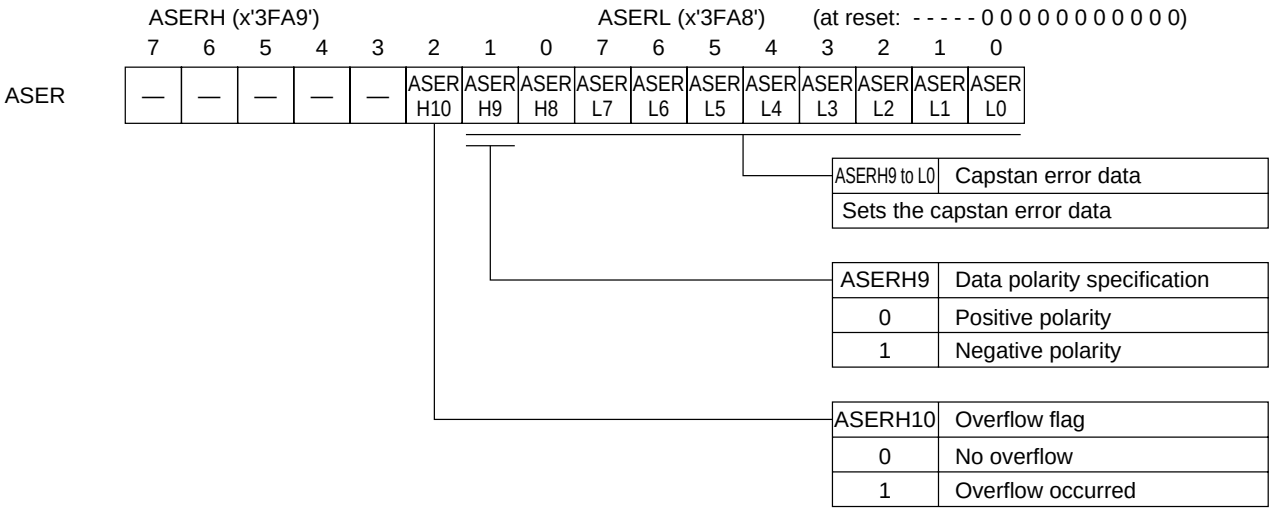


Figure 12-148 AFG Speed Error Register (ASER: x'3FA9' to x'3FA8', R)

YFG Speed Counter Preset Register

Sets cylinder counter preset data.
YSCS is set by the SPCLKCNT (x'3FA5') register.

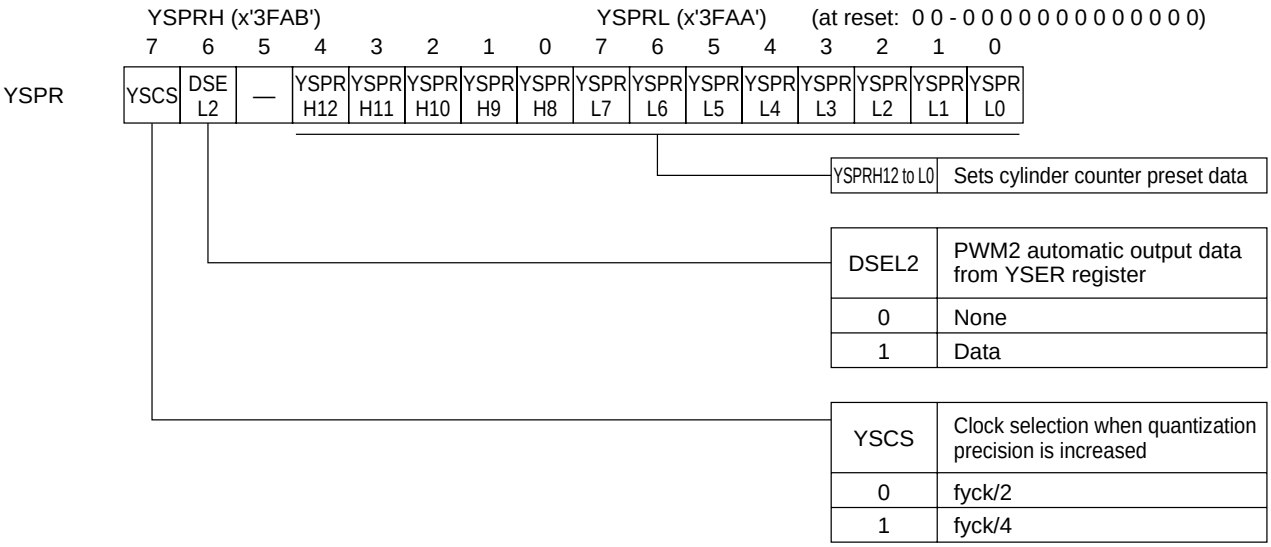


Figure 12-149 YFG Speed Counter Preset Register
(YSPR: x'3FAB' to x'3FAA', R/W)

AFG Speed Counter Preset Register

Sets capstan counter preset data.

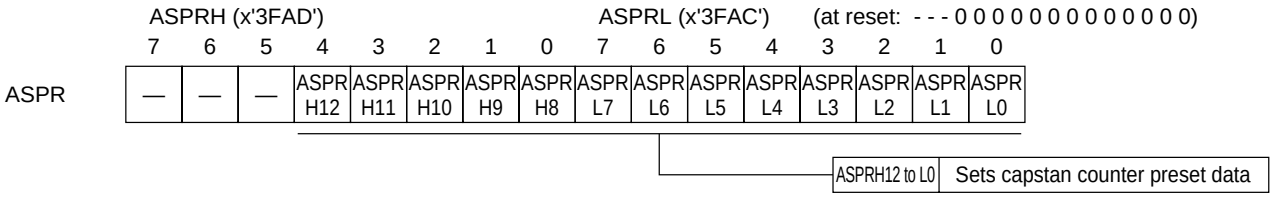


Figure 12-150 AFG Speed Counter Preset Register
(ASPR: x'3FAD' to x'3FAC', R/W)

D/A Conversion Synchronous Output Data Register

Sets the data for synchronous output to DABUF.

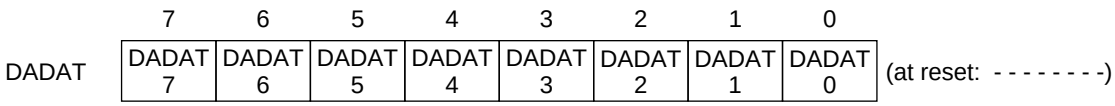


Figure 12-151 D/A Conversion Synchronous Output Data Register
(DADAT: x'3FAE', R/W)

D/A Conversion Output Data Buffer

Sets the D/A conversion value.

	7	6	5	4	3	2	1	0	
DABUF	DABUF 7	DABUF 6	DABUF 5	DABUF 4	DABUF 3	DABUF 2	DABUF 1	DABUF 0	(at reset: 0 0 0 0 0 0 0 0)

Figure 12-152 D/A Conversion Output Data Buffer (DABUF: x'3FAF', R/W)

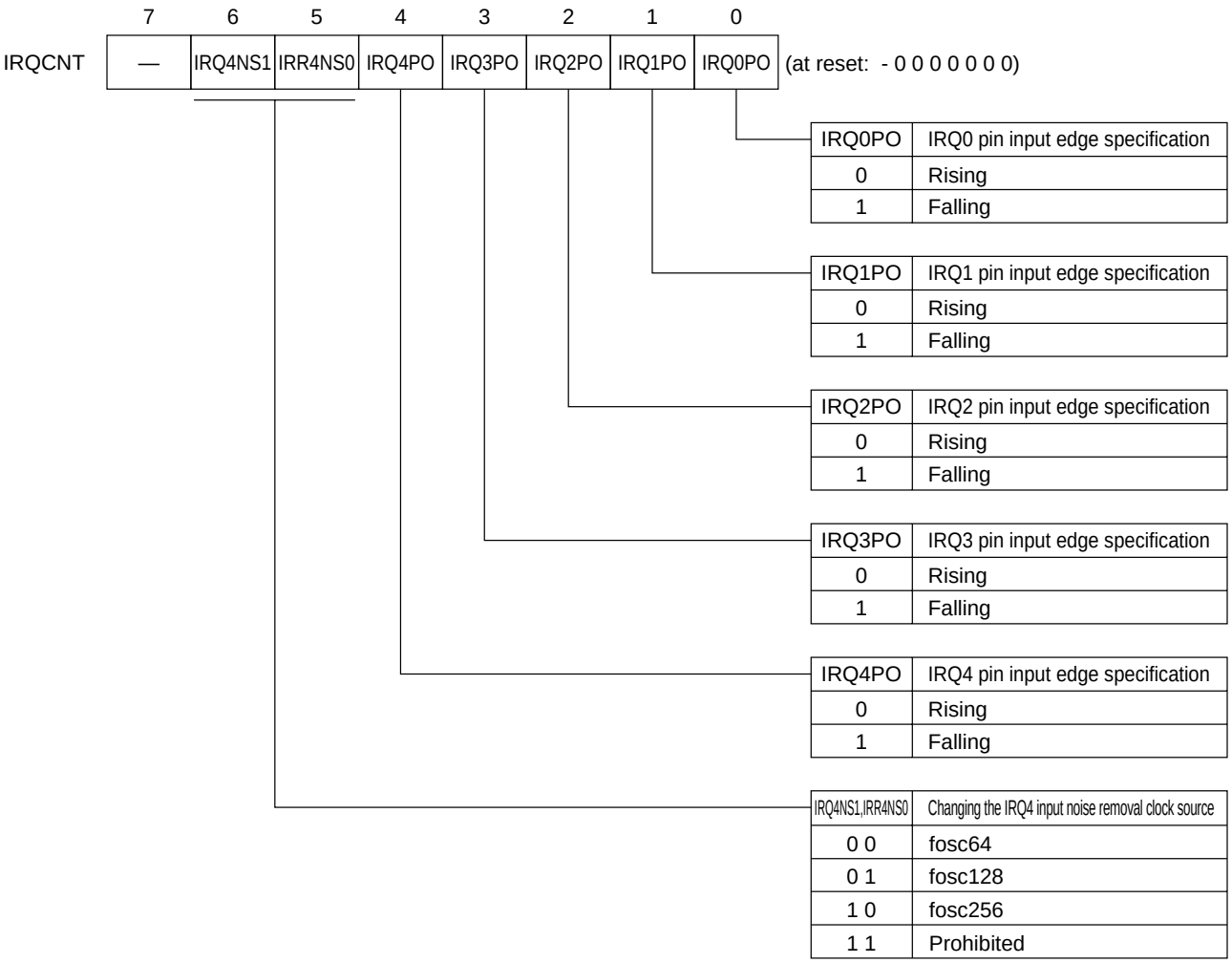
Remote Control Output Data Queue Buffer Register (Serial 1)

After an 8-bit transfer is completed by the shift register, the remote control output data already in TXQUE1 will be automatically set as the next transfer data.

	7	6	5	4	3	2	1	0	
TXQUE1	TXQUE1 7	TXQUE1 6	TXQUE1 5	TXQUE1 4	TXQUE1 3	TXQUE1 2	TXQUE1 1	TXQUE1 0	(at reset: - - - - - - - -)

Figure 12-153 Remote Control Output Data Queue Buffer Register (Serial 1)
(TXQUE1: x'3FB0', R/W)

Port or Interrupt Pin Switching Register



External interrupts are generated when writing to this register.
Disable external interrupts during write operations, and then clear the interrupt request flag after the write is complete.

Figure 12-154 Port or Interrupt Pin Switching Register (IRQCNT: x'3FB2', R/W)

Key Interrupt Control Register

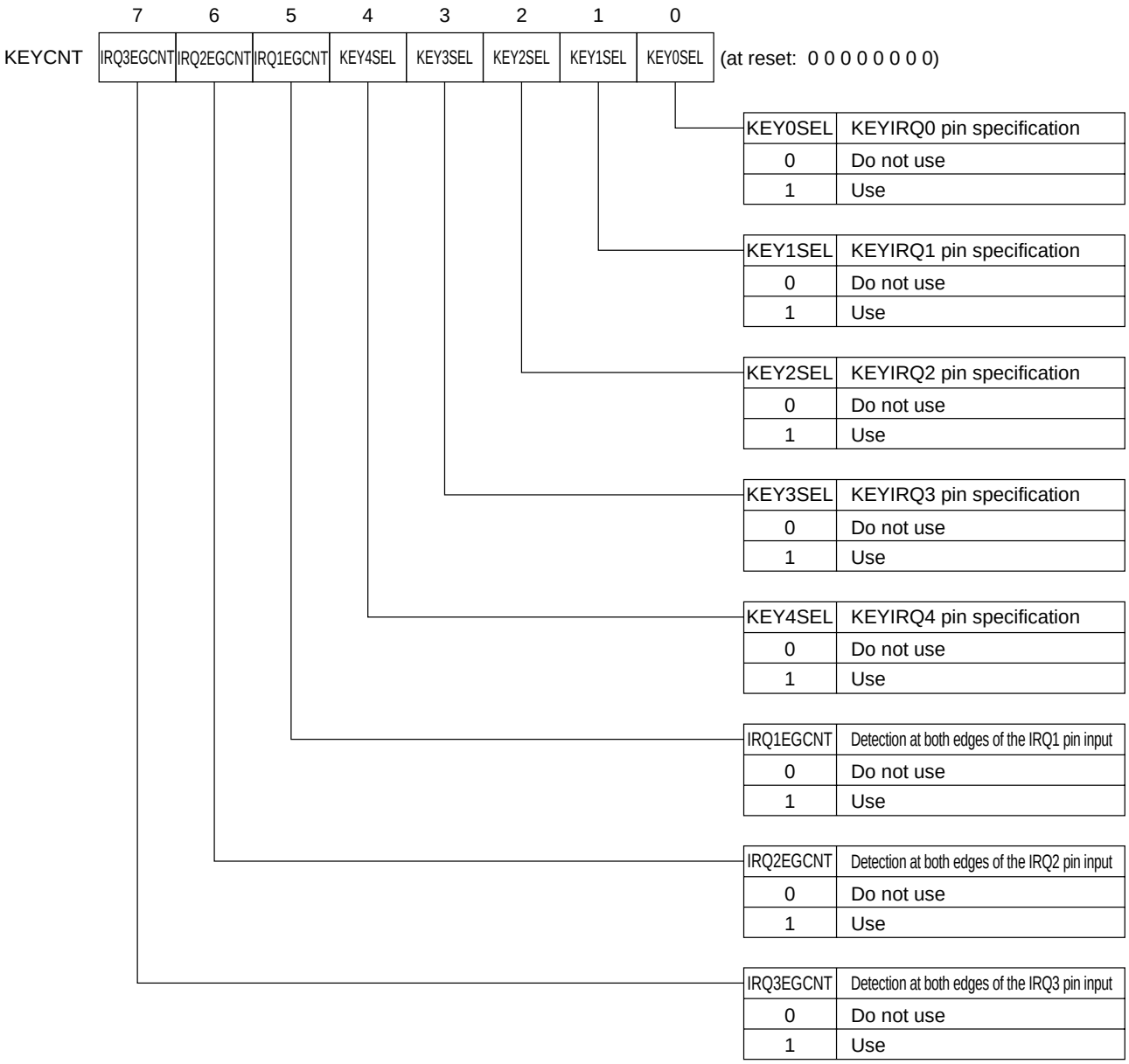


Figure 12-155 Key Interrupt Control Register (KEYCNT: x'3FB3', R/W)

KEY interrupts are generated when writing to this register. Disable KEY interrupts during write operations, and then clear the interrupt request flag after the write is complete.

OSD Operation Control Register 3

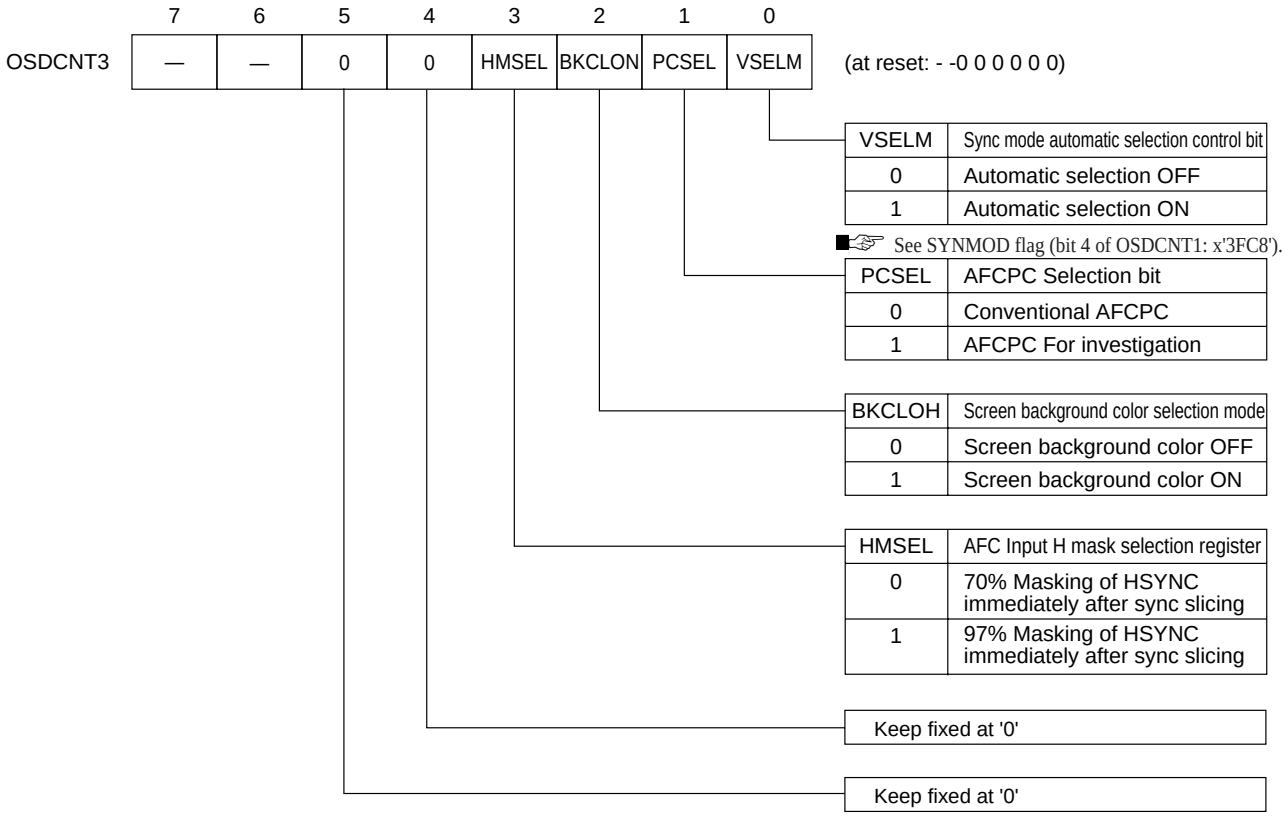


Figure 12-156 OSD Operation Control Register 3 (OSDCNT3: x'3FB9', R/W)

VSYNC Separator Circuit: OSD VSYNC Detection Sensitivity Setting Register

Setting value = (count value)

	7	6	5	4	3	2	1	0	
OSDVDET	OSDVDET 7	OSDVDET 6	OSDVDET 5	OSDVDET 4	OSDVDET 3	OSDVDET 2	OSDVDET 1	OSDVDET 0	(at reset: 00000000)

Figure 12-157 VSYNC Separator Circuit: OSD VSYNC Detection Sensitivity Setting Register (OSDVDET: x'3FBA', R/W)

Synchronous Evaluation Circuit Detection Period Setting Register 1

Setting value=(count value)

	7	6	5	4	3	2	1	0	
SDET1	SDET17	SDET16	SDET15	SDET14	SDET13	SDET12	SDET11	SDET10	(at reset: 0 0 0 0 0 0 0 0)

Figure 12-158 Synchronous Evaluation Circuit Detection Period Setting Register 1
(SDET1: x'3FBC', R/W)

Synchronous Evaluation Circuit Detection Period Setting Register 2

Setting value=(count value)

	7	6	5	4	3	2	1	0	
SDET2	SDET27	SDET26	SDET25	SDET24	SDET23	SDET22	SDET21	SDET20	(at reset: 0 0 0 0 0 0 0 0)

Figure 12-159 Synchronous Evaluation Circuit Detection Period Setting Register 2
(SDET2: x'3FBD', R/W)

Synchronous Evaluation Integration Value

The value of the up/down counter can be monitored.

	7	6	5	4	3	2	1	0	
SDBC	SDBC 7	SDBC 6	SDBC 5	SDBC 4	SDBC 3	SDBC 2	SDBC 1	SDBC 0	(at reset: - - - - - - - -)

Figure 12-160 Synchronous Evaluation Integration Value (SDBC: x'3FBE', R)

Synchronous Evaluation Register

	7	6	5	4	3	2	1	0	
SDFLG	—	—	0	EXT HSEL	SDH SEL1	SDH SEL0	SDBCR	SDDETF	(at reset: - - - - - - - -)

SDDETF	Synchronous evaluation results flag
0	No signal
1	Signal present

SDBCR	Reset the synchronous evaluation integration
0	Normal operation
1	Reset up/down counter

SDH SEL1	SDH SEL0	Hsync input to sync judgment section
0	0	Hsync after input signal processing (filter, half-pulse killer)
0	1	Hsync immediately after sync slicing
1	—	Hsync (PB7) pin input

EXT HSEL	HM SEL	Hsync input to AFC circuit
0	—	Hsync immediately after sync slicing (no masking)
1	0	Hsync immediately after sync slicing (70% masking)
	1	Hsync immediately after sync slicing (97% masking)

* The HMSEL flag is bit 3 of register OSDCNT3 (x'3FB9').

☞ Figure 11-3-1

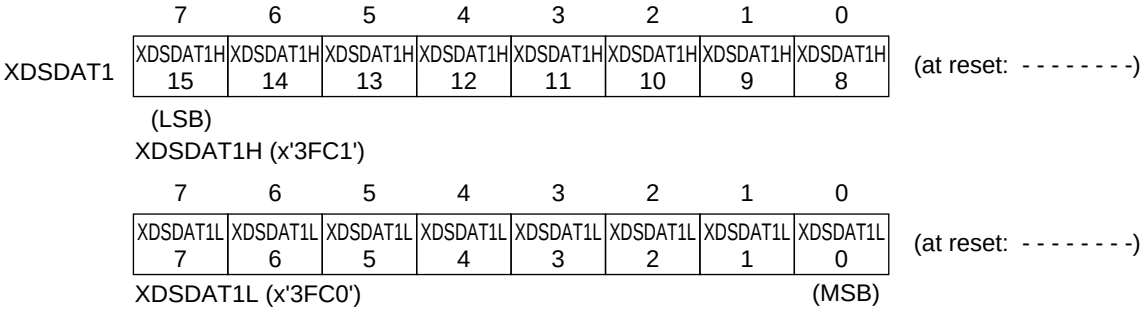
Keep fixed at '0'.

The SDDETF flag is cleared automatically by the hardware.

Figure 12-161 Synchronous Evaluation Register (SDFLG: x'3FBF', R/W)

XDS Data Storage Register 1

Stores the line of XDS data set by XDSLIN1.



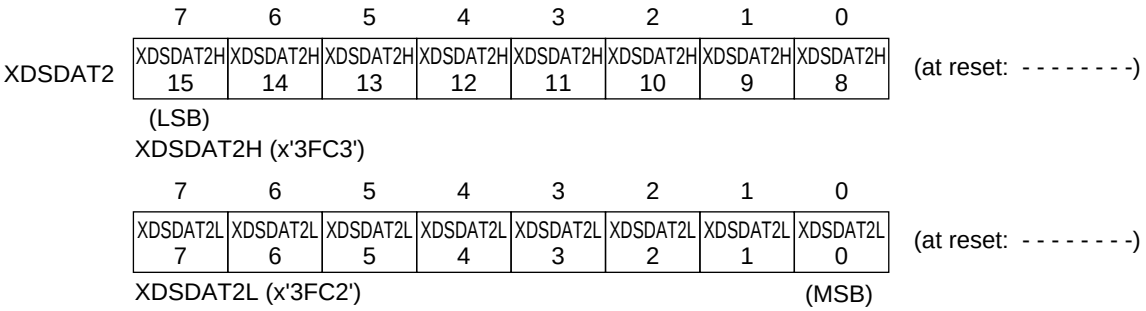
Use a MOVW instruction to change these bits.

The MSB of XDS data is stored in XDSDAT1L0, and the LSB is stored in XDSDAT1H15.

Figure 12-162 XDS Data Storage Register 1
(XDSDAT1: x'3FC1' to x'3FC0', R)

XDS Data Storage Register 2

Stores the line of XDS data set by XDSLIN2.



Use a MOVW instruction to change these bits.

The MSB of XDS data is stored in XDSDAT2L0, and the LSB is stored in XDSDAT2H15.

Figure 12-163 XDS Data Storage Register 2
(XDSDAT2: x'3FC3' to x'3FC2', R)

XDS Start Bit Data Register

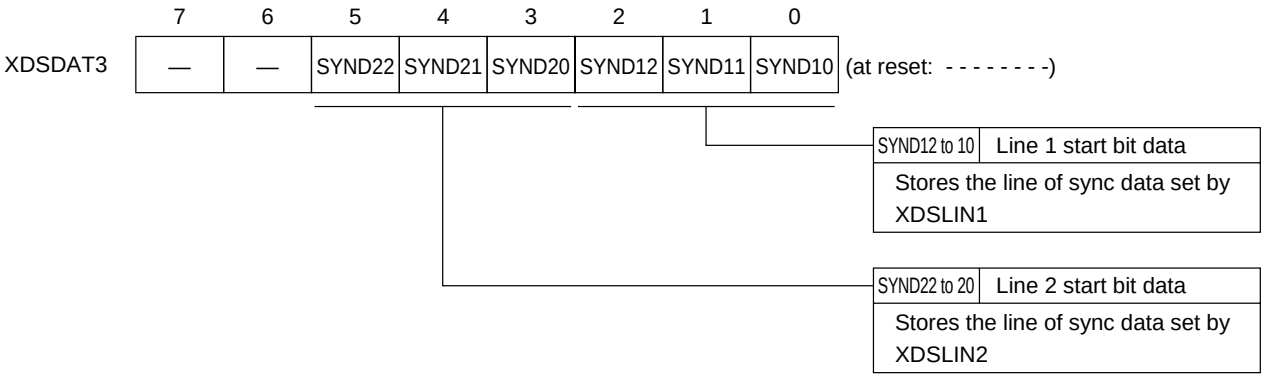
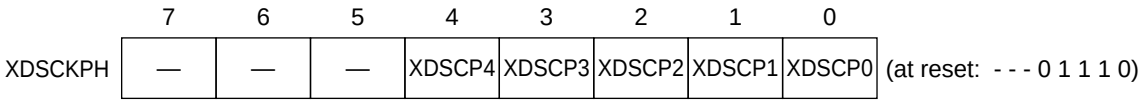


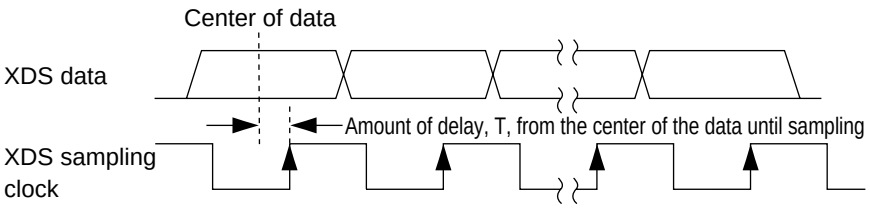
Figure 12-164 XDS Start Bit Data Register (XDSDAT3: x'3FC4', R)

Sampling Clock Phase Adjustment

Setting value=((amount of delay T)+0.300 μs) × 4fsc

The OSD section clock selected by the OSDCK1,0 flags (bits 7 and 6 of OSDCNT1).





Set a value less than x'18'. 4fsc is the clock for the OSD unit as specified by OSDCK1 and OSDCK0 of OSDCNT1.

Figure 12-165 Sampling Clock Phase Adjustment (XDSCKPH: x'3FC5', R/W)

XDS Data Sampling Line Specification 1

Setting value=(sampling line number)-1

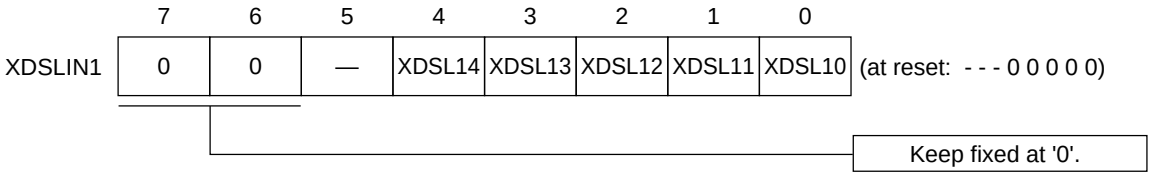
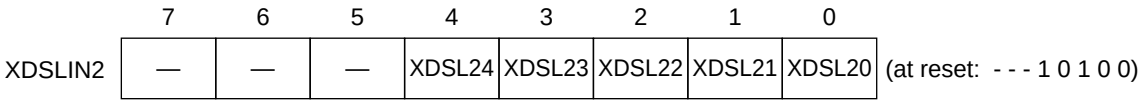


Figure 12-166 XDS Data Sampling Line Specification 1
(XDSLIN1: x'3FC6', R/W)

XDS Data Sampling Line Specification 2



Set a value larger than the value set for XDSLIN2.

Figure 12-167 XDS Data Sampling Line Specification 2
(XDSLIN2: x'3FC7', R/W)

OSD Operation Control Register 1

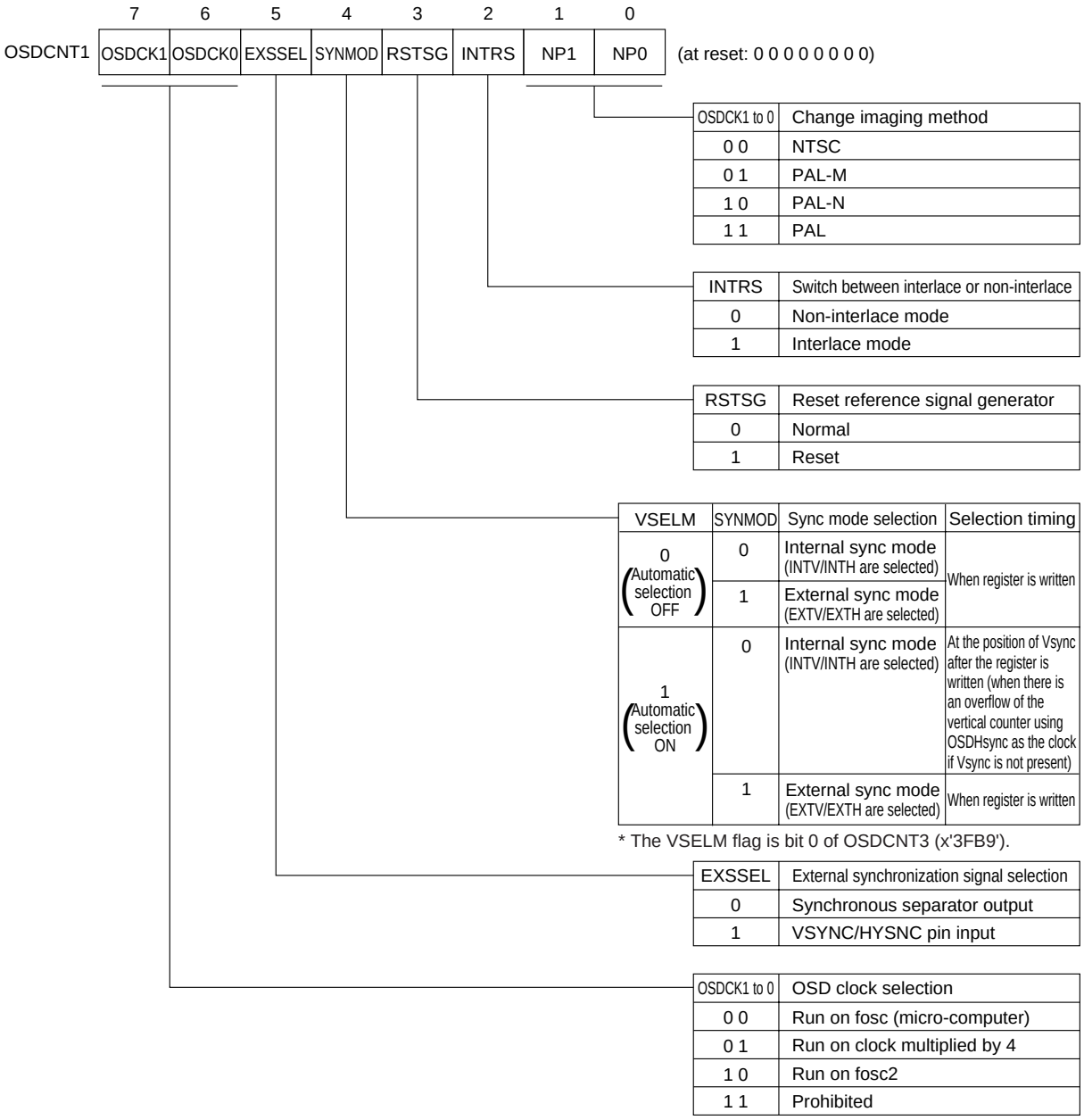


Figure 12-168 OSD Operation Control Register 1 (OSDCNT1: x'3FC8', R/W)

Character Generate ROM End Address Setting Register

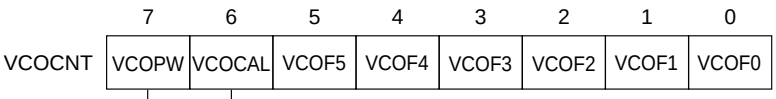
	7	6	5	4	3	2	1	0	
ROMENDA	—	—	—	—	A17	A16	A15	A14	(at reset: - - - - - - - -)

A17 to A14	OSD character generate end address specification

A17	A16	A15	A14	End address
0	0	0	0	Prohibited
0	0	0	1	x'07FFF'
0	0	1	0	x'0BFFF'
0	0	1	1	x'0FFFF'
0	1	0	0	x'13FFF'
0	1	0	1	x'17FFF'
0	1	1	0	x'1BFFF'
0	1	1	1	x'1FFFF'
1	0	0	0	x'23FFF'
1	0	0	1	x'27FFF'
1	0	1	0	x'2BFFF'
1	0	1	1	x'2FFFF'
1	1	0	0	x'33FFF'
1	1	0	1	x'37FFF'
1	1	1	0	x'3BFFF'
1	1	1	1	x'3FFFF'

Figure 12-169 Character Generate ROM End Address Setting Register
(ROMENDA: x'3FC9', R/W)

VCO Control Register



VCOF5 to 0	VCO oscillation frequency control
If VCOCAL is set to '1', data is automatically set.	

VCOCAL	VCO calibration control
0	Normal
1	Calibration ON

VCOPW	VCO power source ON/OFF
0	Power source OFF
1	Power source ON

In the case that VCOCAL='1', after data has been set for VCOF5 through VCOF0, reset only VCOF0.
In the case that VCOCAL='1', VCOF5 through VCOF0 must be set to '100000'.

Write x'00' when changing over to the standby mode (power saving).

Figure 12-170 VCO Control Register (VCOCNT1: x'3FD0', R/W)

Video Signal Control Register

	7	6	5	4	3	2	1	0	
CLCNT	FDIVS	CVOSEL	BIASPW	BUFPW	CSCLPW	CVCLPW	CLSEL1	CLSEL0	(at reset: 0 0 0 0 0 0 0 0)

CLSEL1 to 0	Change clamp level
0 0	0.8V
0 1	1.0V
1 0	1.2V
1 1	1.5V

CVCLPW	CVIN pin clamp power source ON/OFF
0	Power source OFF
1	Power source ON

CSCLPW	CVIN2 pin clamp power source ON/OFF
0	Power source OFF
1	Power source ON

BUFPW	Halftone buffer power ON/OFF
0	Power source OFF
1	Power source ON

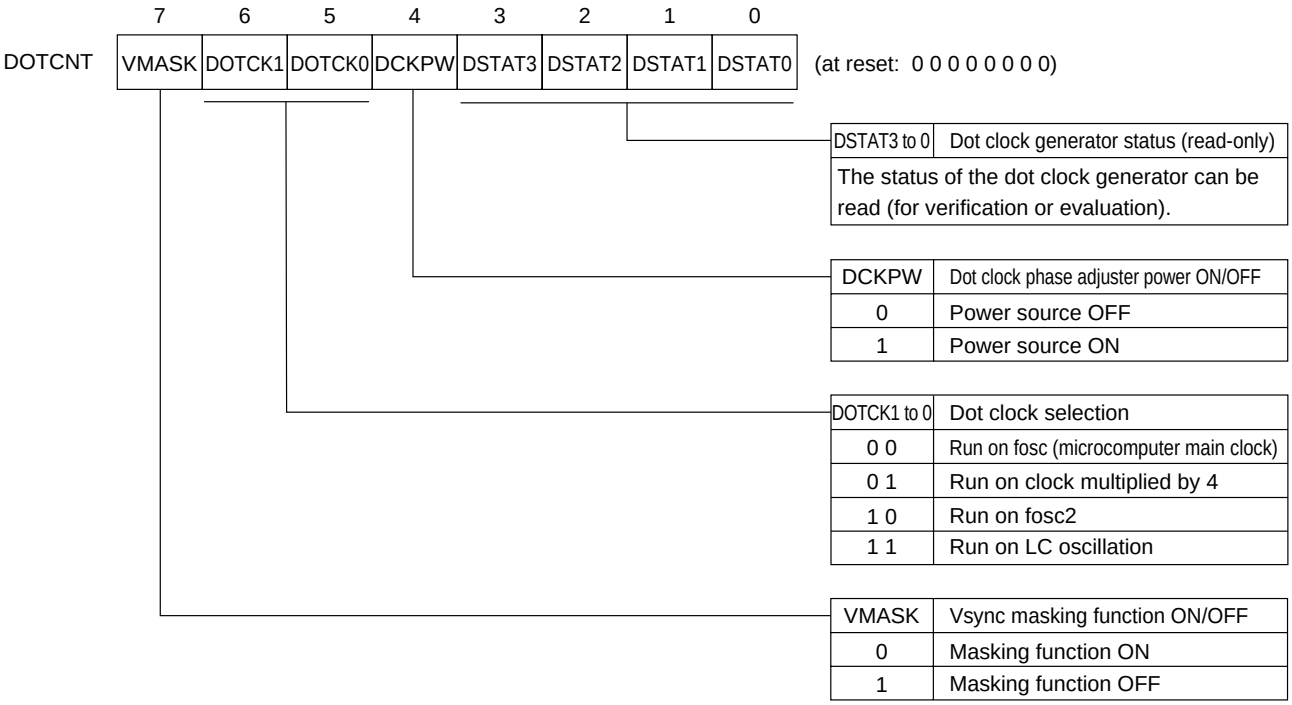
BIASPW	Video signal character brightness bias power ON/OFF (video through control)
0	Power source OFF (video through)
1	Power source ON (OSD display ON)

CVOSEL	Change CVOUT output voltage level
0	2Vpp
1	1Vpp

FDIVS	Select dot clock frequency division ratio
0	1/2 Frequency of the clock selected by the DOTCK1 and DOTCK0 flags.
1	1/3 Frequency of the clock selected by the DOTCK1 and DOTCK0 flags.

Figure 12-171 Video Signal Control Register (CLCNT: x'3FD1', R/W)

Dot Clock Control Register



When running on the LC oscillation, set bp7 and bp6 of the PADIR register (x'3F35'). to '0' (input).
When running on the fosc2 clock, set bp1 of the P2SEL register (x'3F39') to '1' (oscillate).
When running on the multiply by 4 clock, set bp1 and bp0 of the PBDIR register (x'3F36') to '0' (input).

Figure 12-172 Dot Clock Control Register (DOTCNT: x'3FD2', R/W)

Horizontal Sync AFC Control Register

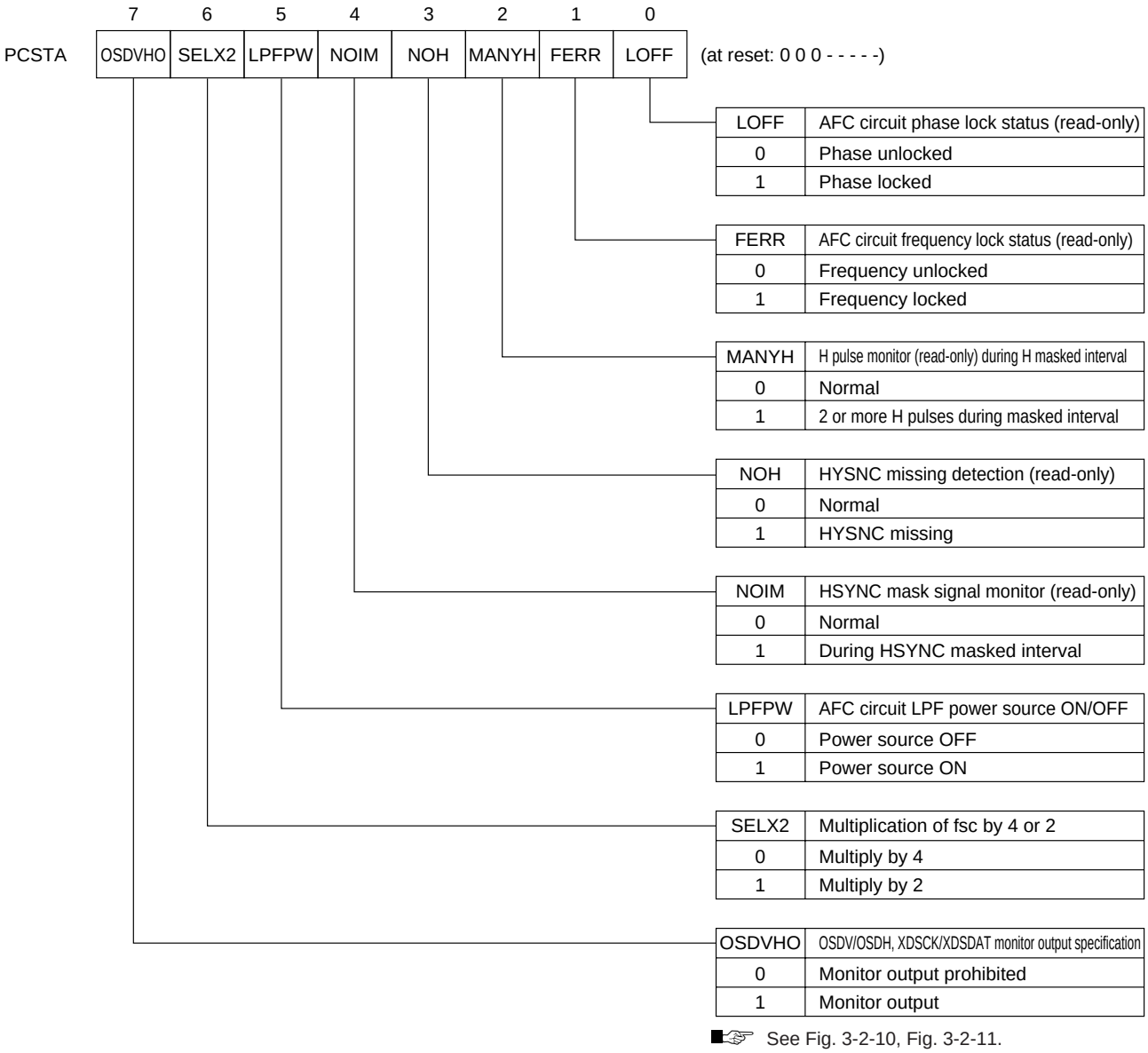
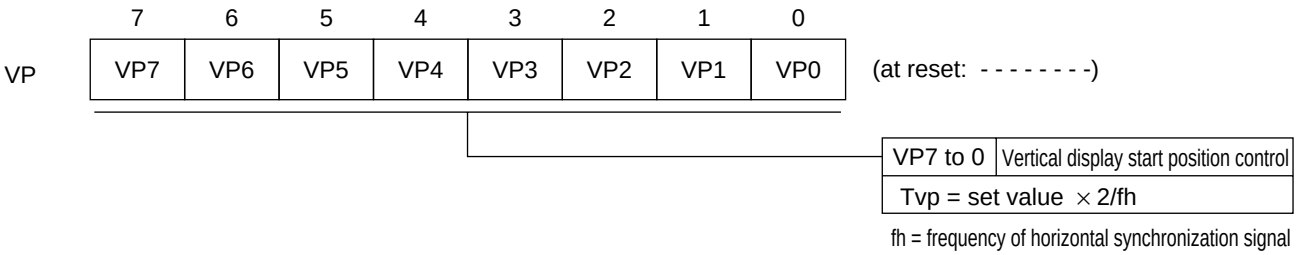


Figure 12-173 Horizontal Sync AFC Control Register (PCSTA: x'3FD3', R/W)

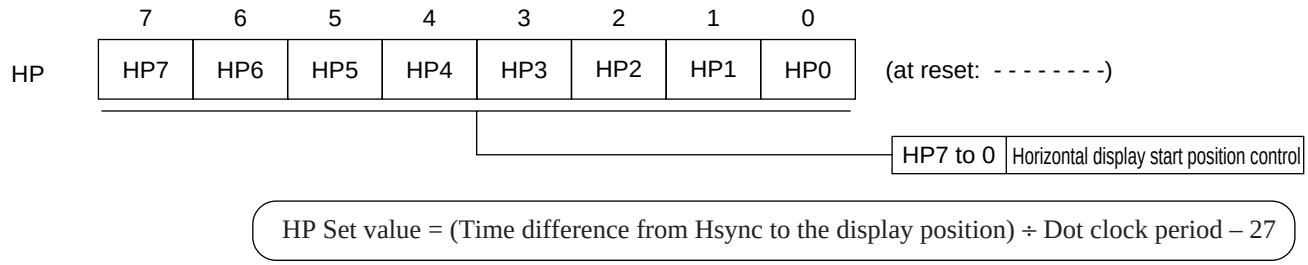
VP Vertical Display Start Position Control Register



The screen configuration is shown in Fig. 11-1-4. Each rectangle corresponds to on character composed of 12x18 dots. The VP vertical display start position control register is for setting the value of Tvp in the figure. For each set value of 1, the position shifts by two horizontal lines. Set a value equal to half the number of Hsync pulses at the end of Vsync.

Figure 12-174 Vertical Display Start Position Control Register
(VP: x'3FD8', R/W)

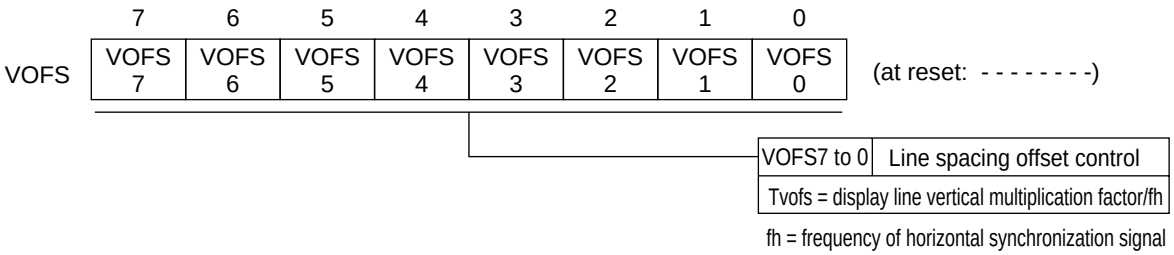
HP Horizontal Display Start Position Control Register



The HP horizontal display start position control register is for setting the value of Thp. For each set value of 1, the position shifts by two dot clocks. The position of the 27th dot clock at the end of Hsync corresponds to a set value of 0.

Figure 12-175 Horizontal Display Start Position Control Register
(HP: x'3FD9', R/W)

VOFS Display Line Spacing Control Register



The VOFS display line spacing control register is for setting Tvofs in Fig. 11-1-4. For each set value of 1, the display moves by the number of horizontal lines of the vertical magnification used during the immediately previous display. For example, the spacing of the line below the character display line of vertical magnification of 4 moves by 4 horizontal lines for each setting of 1.

When VOFS is set to a value equal to 1 or more, the border display is possible only for one line below the display line.

Figure 12-176 Display Line Spacing Control Register (VOFS: x'3FDA', R/W)

VLIN Display Control Register

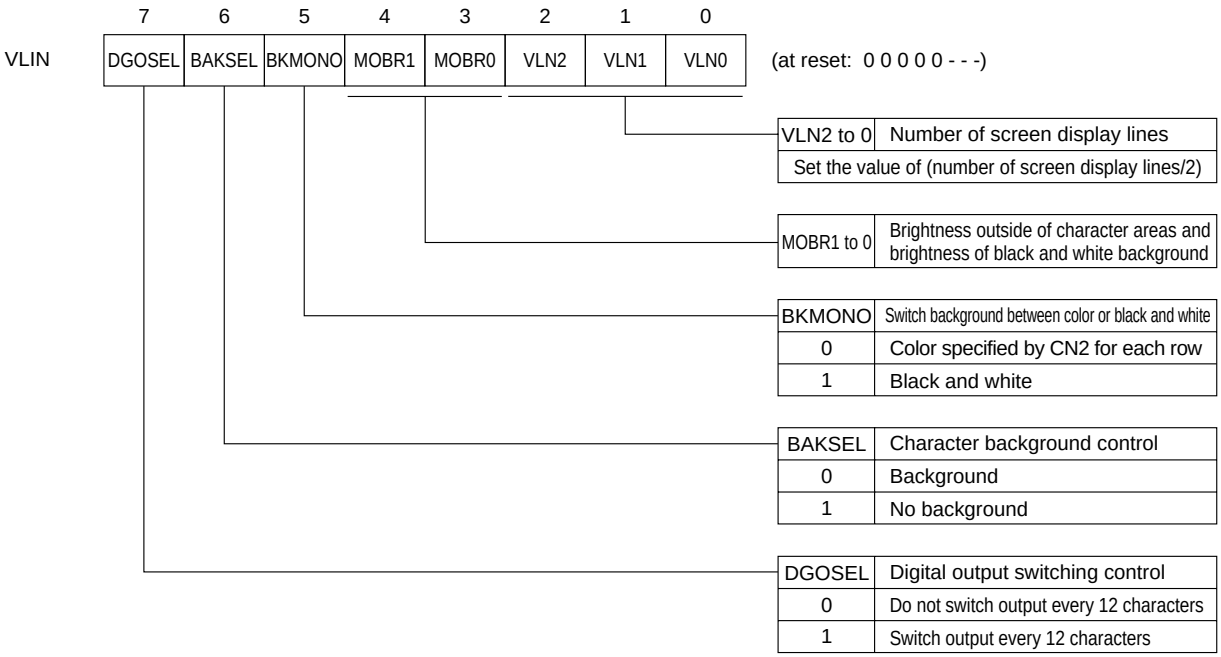


Figure 12-177 Display Control Register (VLIN: x'3FDB', R/W)

OSD Interrupt Display Line Setting Register

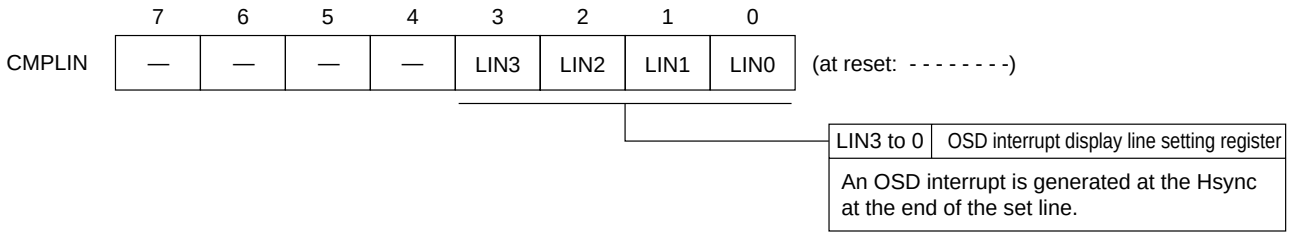


Figure 12-178 OSD Interrupt Display Line Setting Register
(CMPLIN: x'3FDC', R/W)

OSD Operation Control Register 2

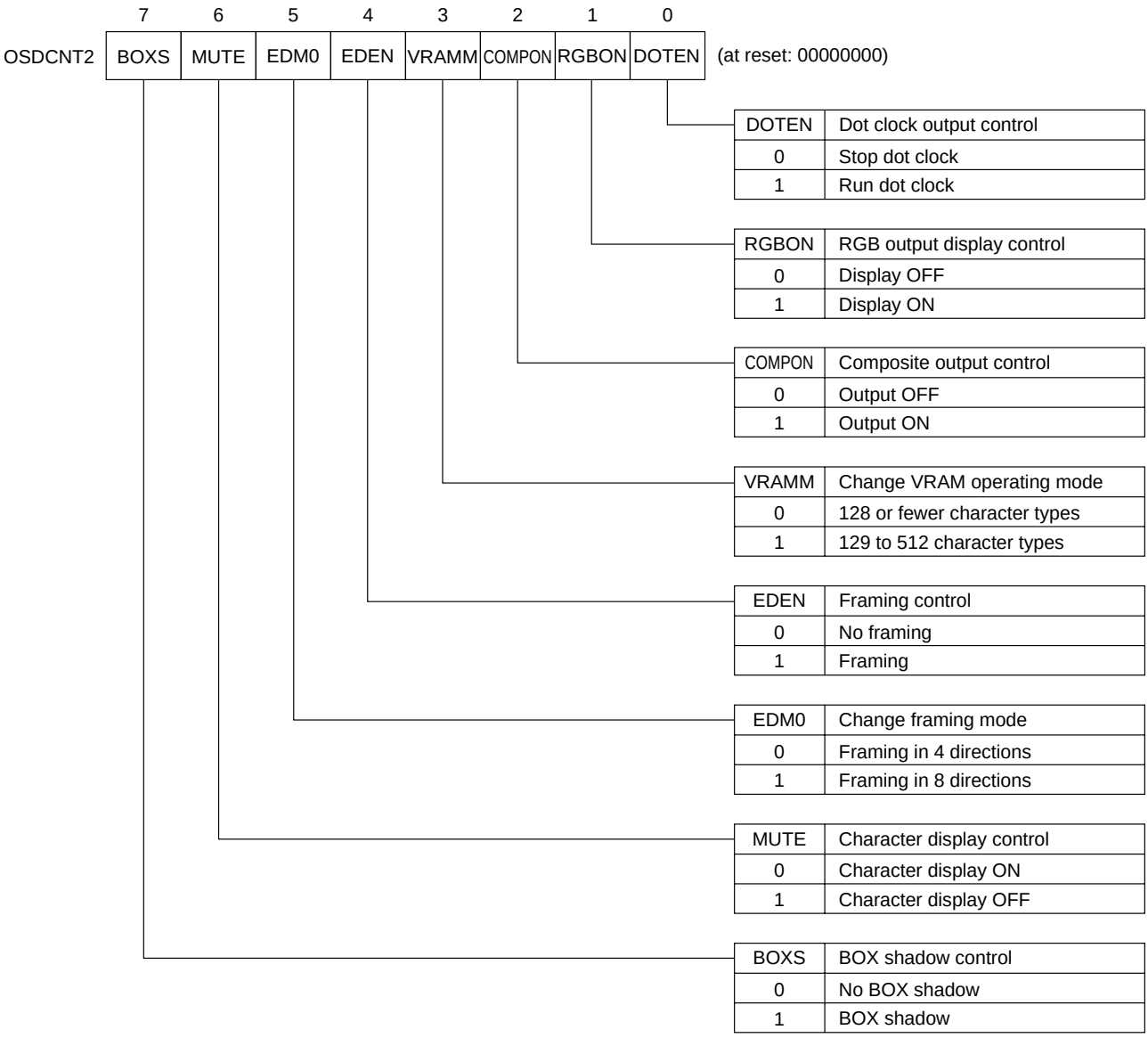


Figure 12-179 OSD Operation Control Register 2 (OSDCNT2: x'3FDD', R/W)

Vsync Separator Circuit: Setting of Detection Sensitivity of Vsync for Servo

Setting value=(count value)

	7	6	5	4	3	2	1	0	
VDET	VDET 7	VDET 6	VDET 5	VDET 4	VDET 3	VDET 2	VDET 1	VDET 0	(at reset: 0 0 0 0 0 0 0 0)

Figure 12-180 Vsync Separator Circuit: Setting of Detection Sensitivity of Vsync for Servo (VDET: x'3FDE', R/W)

VSYNC Control Register

	7	6	5	4	3	2	1	0	
SEPC	SVMON	OVMON	—	0	0	FLAME	N8SEL	SVVSEL	(at reset: - - - 0 - - 00)

SVVSEL	Servo Vsync selection
0	Sync separator section output
1	VSYNC Pin input

N8SEL	Specification of presence/absence of Servo Vsync weak electric field countermeasure function
0	Not present
1	Present

FLAME	Field judgment result (Read only)	
	Judgment range 1	Judgment range 2
0	Odd field	Even field
1	Even field	Odd field

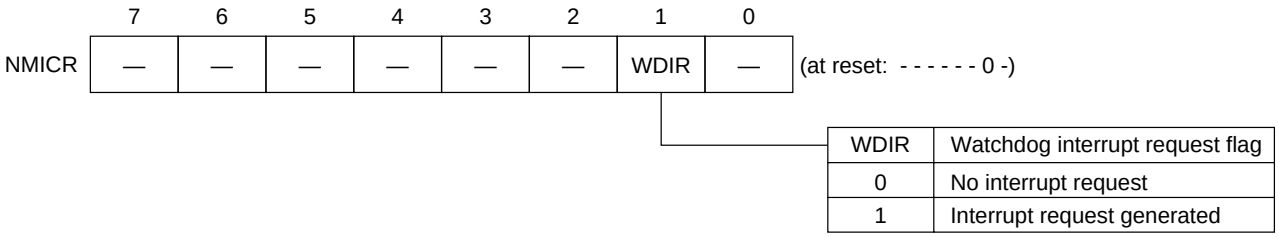
Keep fixed at '0'.

OSDVsync monitor

Servo Vsync monitor

Figure 12-181 VSYNC Control Register (SEPC: x'3FDF', R/W one section is read-only)

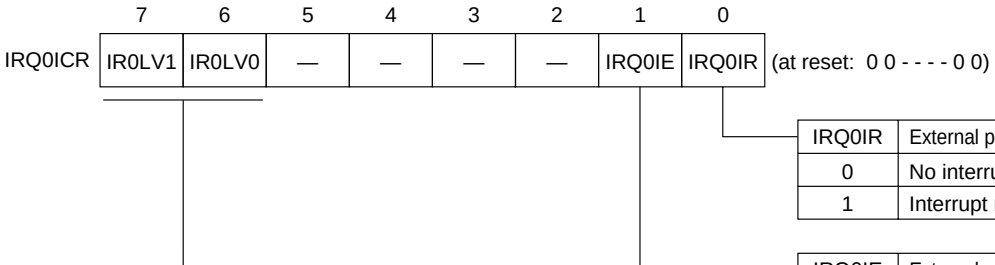
Non-Maskable Interrupt Control Register



WDIR	Watchdog interrupt request flag
0	No interrupt request
1	Interrupt request generated

Figure 12-182 Non-Maskable Interrupt Control Register (NMICR: x'3FE1', R/W)

External Interrupt 0 Control Register



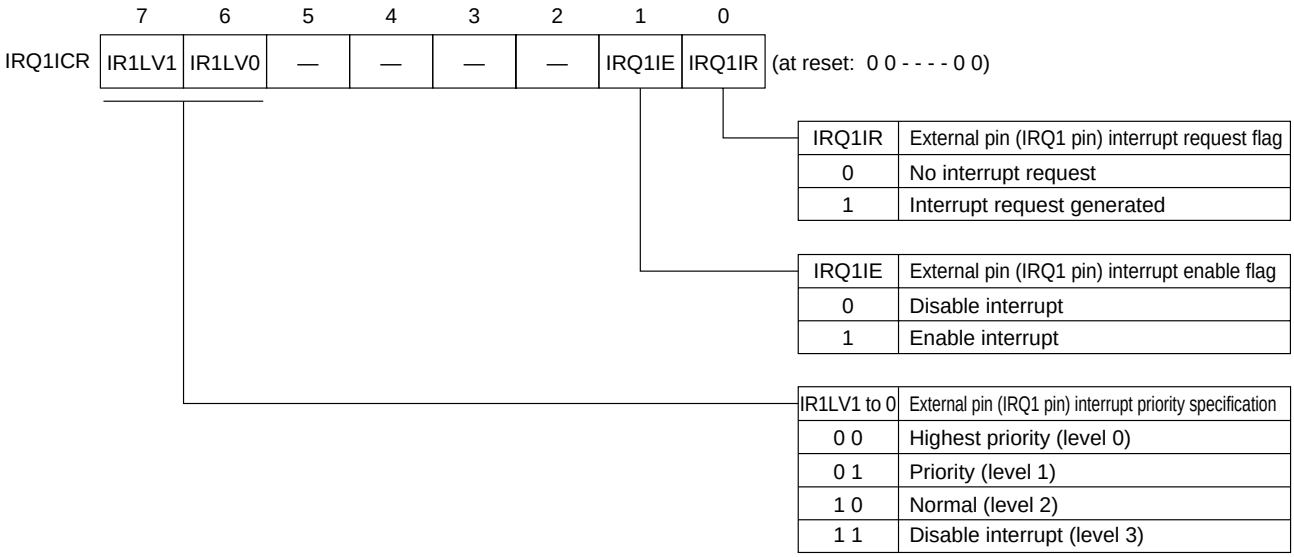
IRQ0IR	External pin (IRQ0 pin) interrupt request flag
0	No interrupt request
1	Interrupt request generated

IRQ0IE	External pin (IRQ0 pin) interrupt enable flag
0	Disable interrupt
1	Enable interrupt

IR0LV1 to 0	External pin (IRQ0 pin) interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-183 External Interrupt 0 Control Register (IRQ0ICR: x'3FE2', R/W)

External Interrupt 1 Control Register



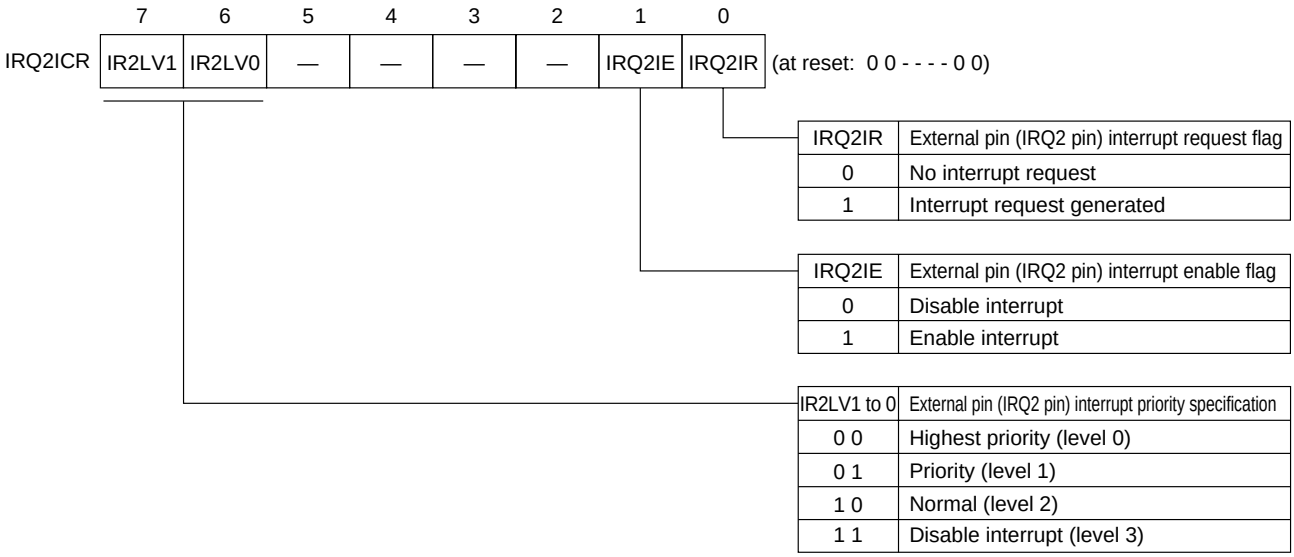
IRQ1IR	External pin (IRQ1 pin) interrupt request flag
0	No interrupt request
1	Interrupt request generated

IRQ1IE	External pin (IRQ1 pin) interrupt enable flag
0	Disable interrupt
1	Enable interrupt

IR1LV1 to 0	External pin (IRQ1 pin) interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-184 External Interrupt 1 Control Register (IRQ1ICR: x'3FE3', R/W)

External Interrupt 2 Control Register



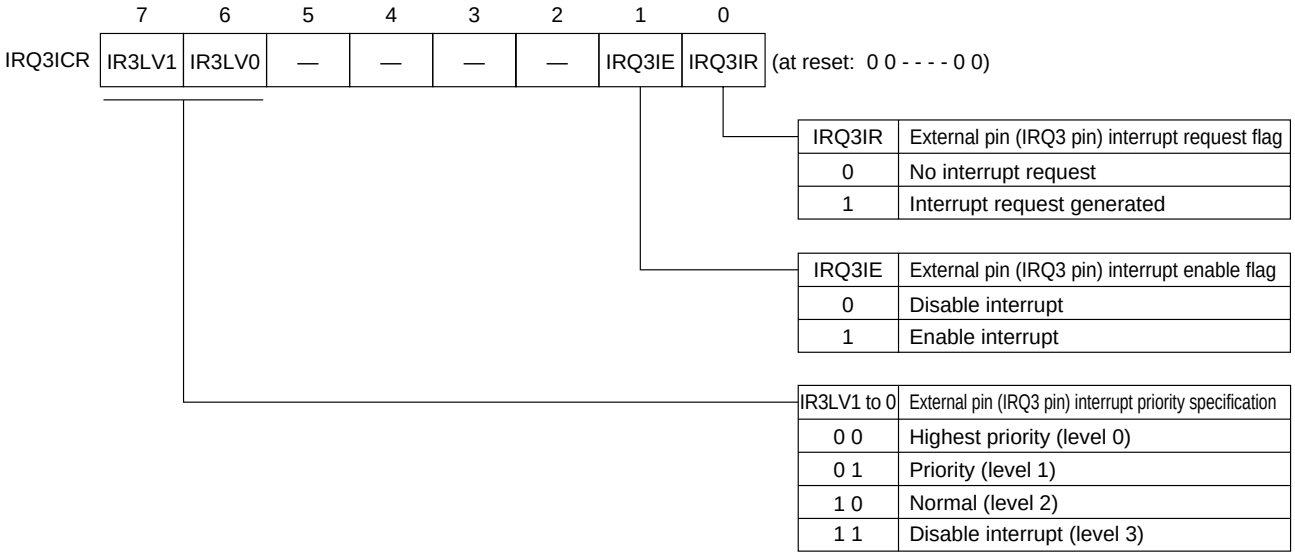
IRQ2IR	External pin (IRQ2 pin) interrupt request flag
0	No interrupt request
1	Interrupt request generated

IRQ2IE	External pin (IRQ2 pin) interrupt enable flag
0	Disable interrupt
1	Enable interrupt

IR2LV1 to 0	External pin (IRQ2 pin) interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-185 External Interrupt 2 Control Register (IRQ2ICR: x'3FE4', R/W)

External Interrupt 3 Control Register



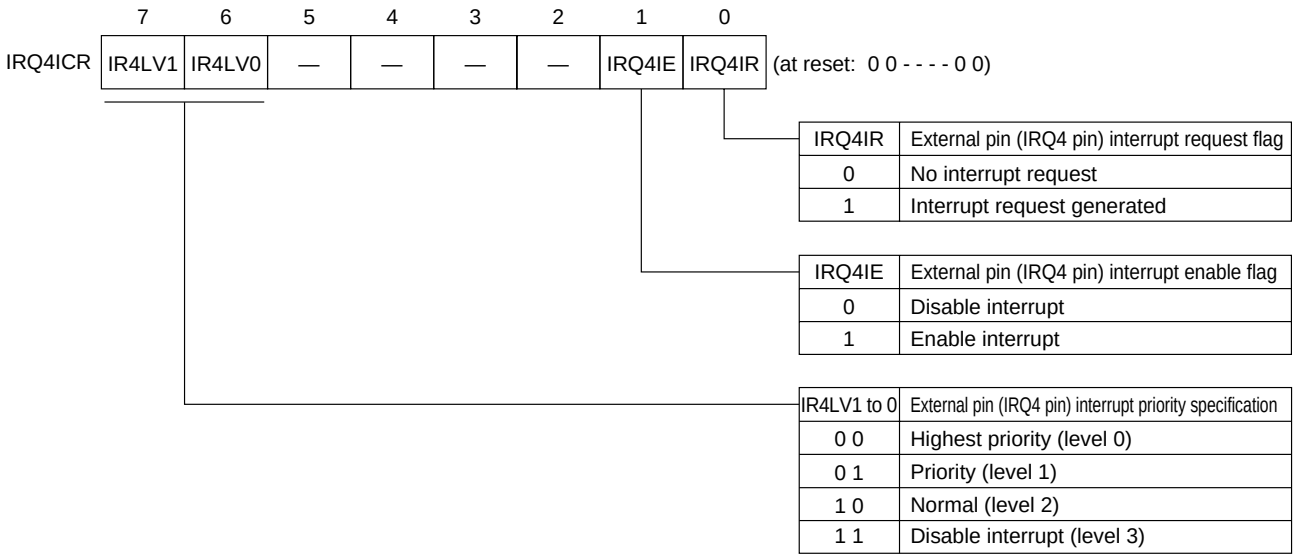
IRQ3IR	External pin (IRQ3 pin) interrupt request flag
0	No interrupt request
1	Interrupt request generated

IRQ3IE	External pin (IRQ3 pin) interrupt enable flag
0	Disable interrupt
1	Enable interrupt

IR3LV1 to 0	External pin (IRQ3 pin) interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-186 External Interrupt 3 Control Register (IRQ3ICR: x'3FE5', R/W)

External Interrupt 4 Control Register



IRQ4IR	External pin (IRQ4 pin) interrupt request flag
0	No interrupt request
1	Interrupt request generated

IRQ4IE	External pin (IRQ4 pin) interrupt enable flag
0	Disable interrupt
1	Enable interrupt

IR4LV1 to 0	External pin (IRQ4 pin) interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-187 External Interrupt 4 Control Register (IRQ4ICR: x'3FE6', R/W)

KEY Interrupt Control Register

	7	6	5	4	3	2	1	0	
KEYICR	KEYLV1	KEYLV0	—	—	—	—	KEYIE	KEYIR	(at reset: 0 0 - - - - 0 0)

KEYIR	KEY interrupt request flag
0	No interrupt request
1	Interrupt request generated

KEYIE	KEY interrupt enable flag
0	Disable interrupt
1	Enable interrupt

KEYLV1 to 0	KEY interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-188 KEY Interrupt Control Register (KEYICR: x'3FE7', R/W)

Cylinder FG Interrupt Control Register

	7	6	5	4	3	2	1	0	
YFGICR	YFGLV1	YFGLV0	—	—	—	—	YFGIE	YFGIR	(at reset: 0 0 - - - - 0 0)

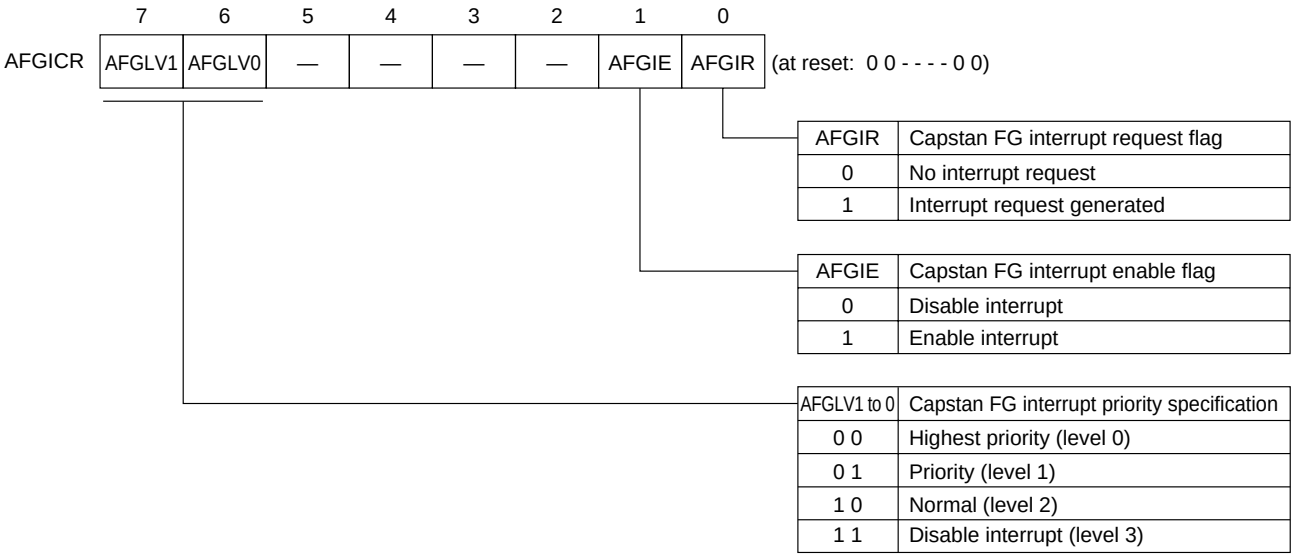
YFGIR	Cylinder FG interrupt request flag
0	No interrupt request
1	Interrupt request generated

YFGIE	Cylinder FG interrupt enable flag
0	Disable interrupt
1	Enable interrupt

YFGLV1 to 0	Cylinder FG interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-189 Cylinder FG Interrupt Control Register (YFGICR: x'3FE8', R/W)

Capstan FG Interrupt Control Register



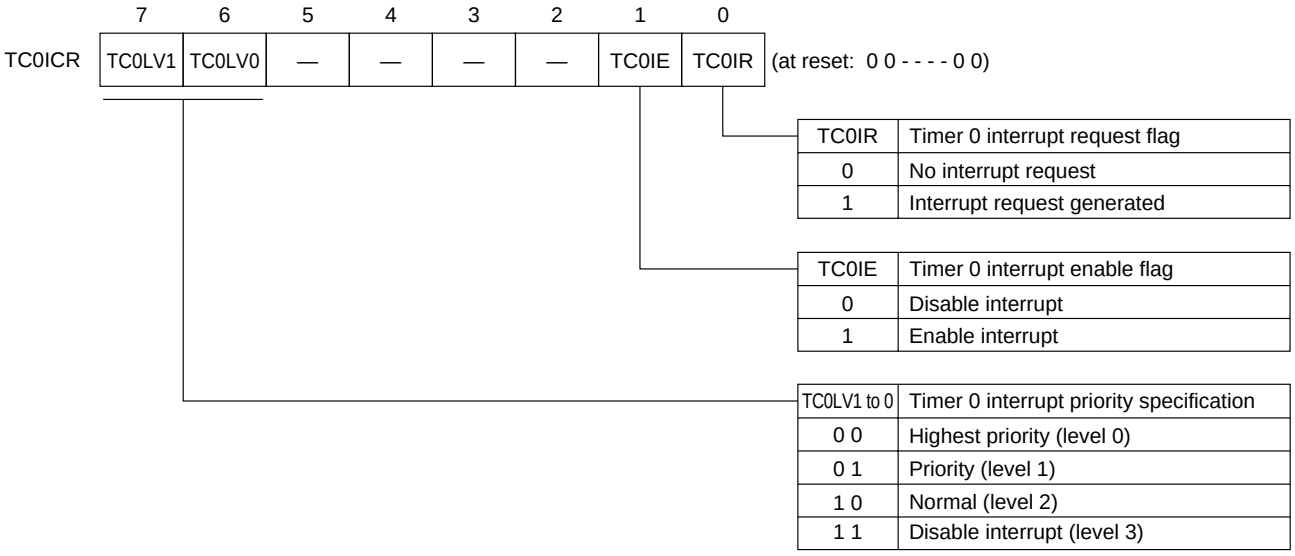
AFGIR	Capstan FG interrupt request flag
0	No interrupt request
1	Interrupt request generated

AFGIE	Capstan FG interrupt enable flag
0	Disable interrupt
1	Enable interrupt

AFGLV1 to 0	Capstan FG interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-190 Capstan FG Interrupt Control Register (AFGICR: x'3FE9', R/W)

Timer 0 Interrupt Control Register



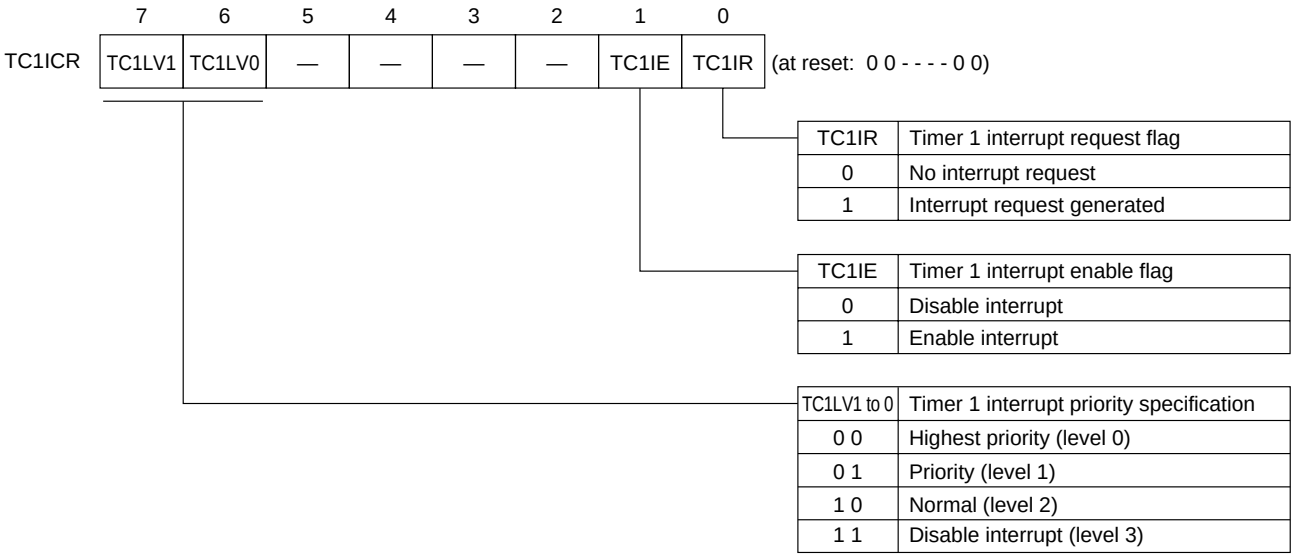
TC0IR	Timer 0 interrupt request flag
0	No interrupt request
1	Interrupt request generated

TC0IE	Timer 0 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

TC0LV1 to 0	Timer 0 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-191 Timer 0 Interrupt Control Register (TC0ICR: x'3FEA', R/W)

Timer 1 Interrupt Control Register



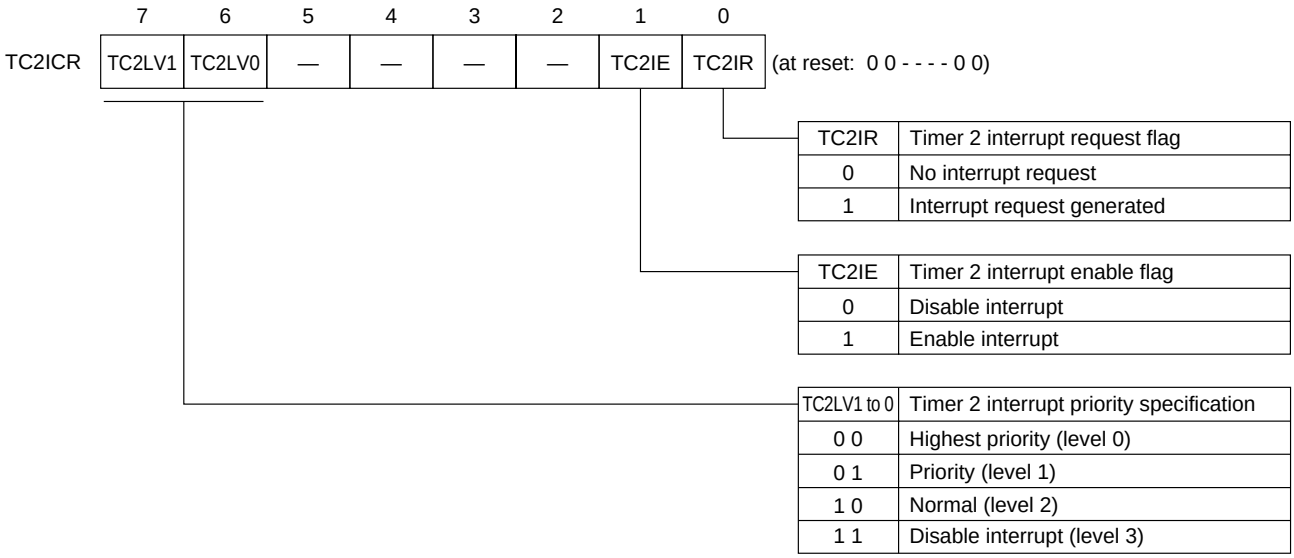
TC1IR	Timer 1 interrupt request flag
0	No interrupt request
1	Interrupt request generated

TC1IE	Timer 1 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

TC1LV1 to 0	Timer 1 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-192 Timer 1 Interrupt Control Register (TC1ICR: x'3FEB', R/W)

Timer 2 Interrupt Control Register



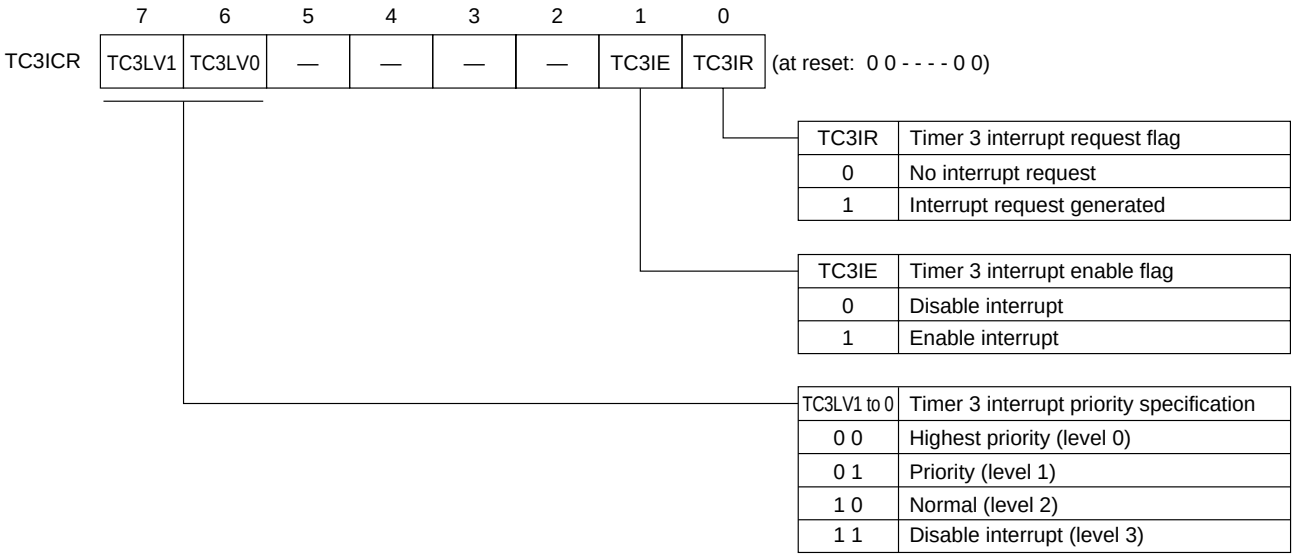
TC2IR	Timer 2 interrupt request flag
0	No interrupt request
1	Interrupt request generated

TC2IE	Timer 2 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

TC2LV1 to 0	Timer 2 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-193 Timer 2 Interrupt Control Register (TC2ICR: x'3FEC', R/W)

Timer 3 Interrupt Control Register



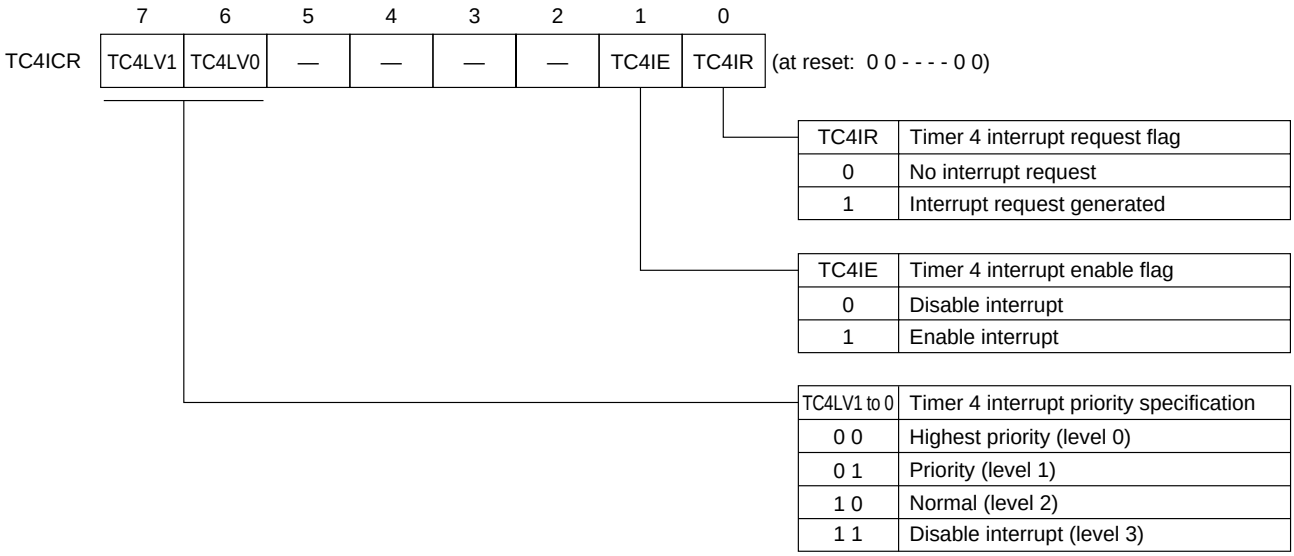
TC3IR	Timer 3 interrupt request flag
0	No interrupt request
1	Interrupt request generated

TC3IE	Timer 3 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

TC3LV1 to 0	Timer 3 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-194 Timer 3 Interrupt Control Register (TC3ICR: x'3FED', R/W)

Timer 4 Interrupt Control Register



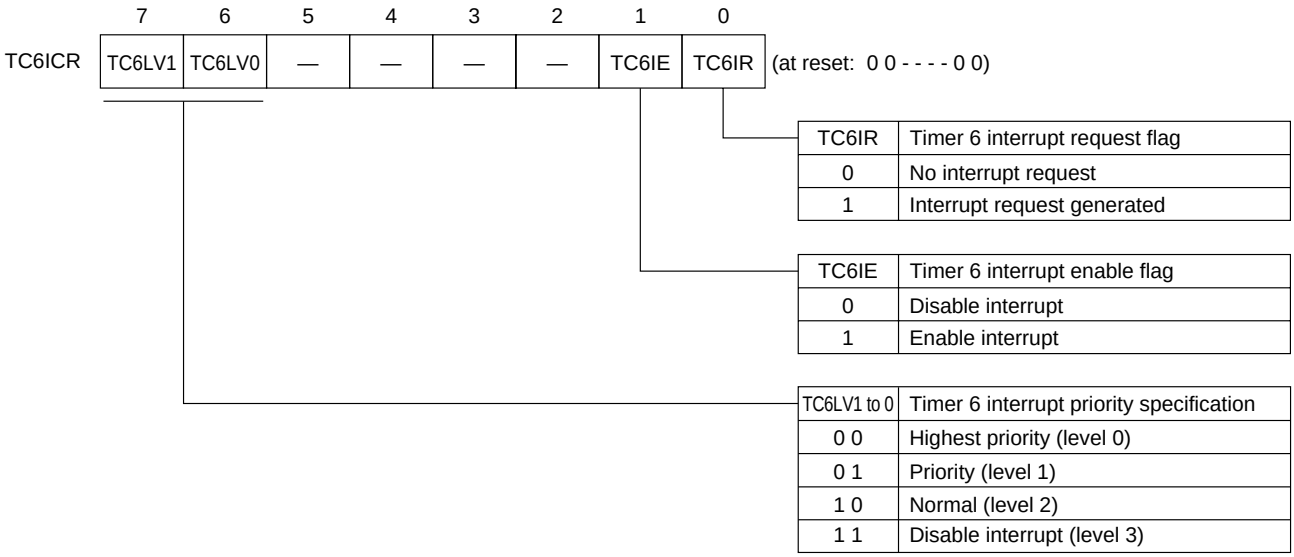
TC4IR	Timer 4 interrupt request flag
0	No interrupt request
1	Interrupt request generated

TC4IE	Timer 4 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

TC4LV1 to 0	Timer 4 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-195 Timer 4 Interrupt Control Register (TC4ICR: x'3FEE', R/W)

Timer 6 Interrupt Control Register



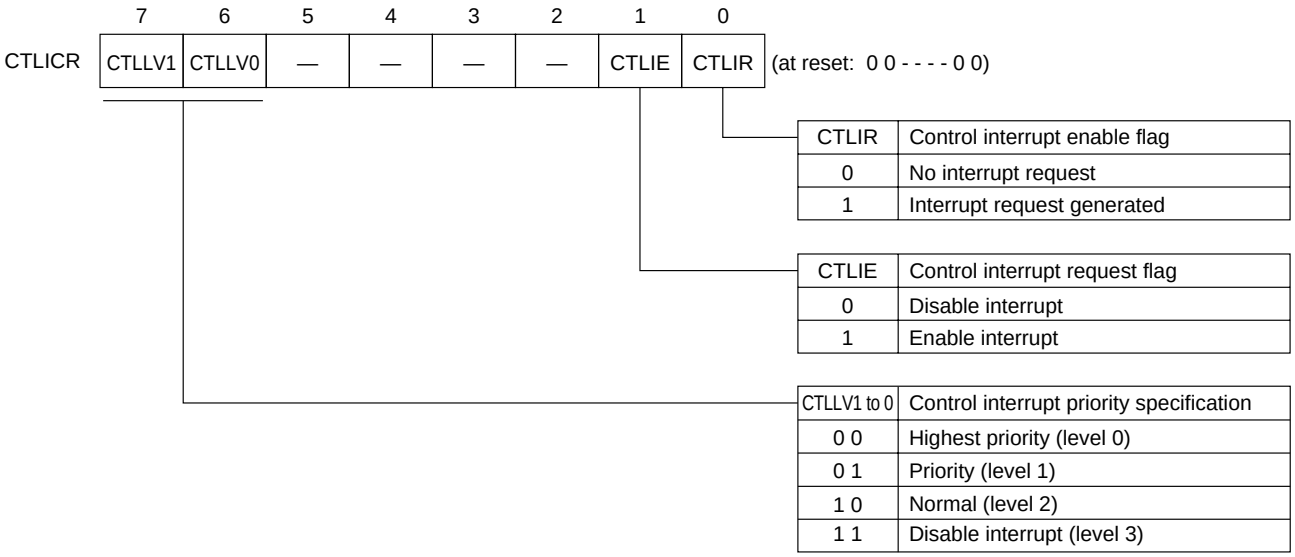
TC6IR	Timer 6 interrupt request flag
0	No interrupt request
1	Interrupt request generated

TC6IE	Timer 6 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

TC6LV1 to 0	Timer 6 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-196 Timer 6 Interrupt Control Register (TC6ICR: x'3FEF', R/W)

Control Signal Interrupt Control Register



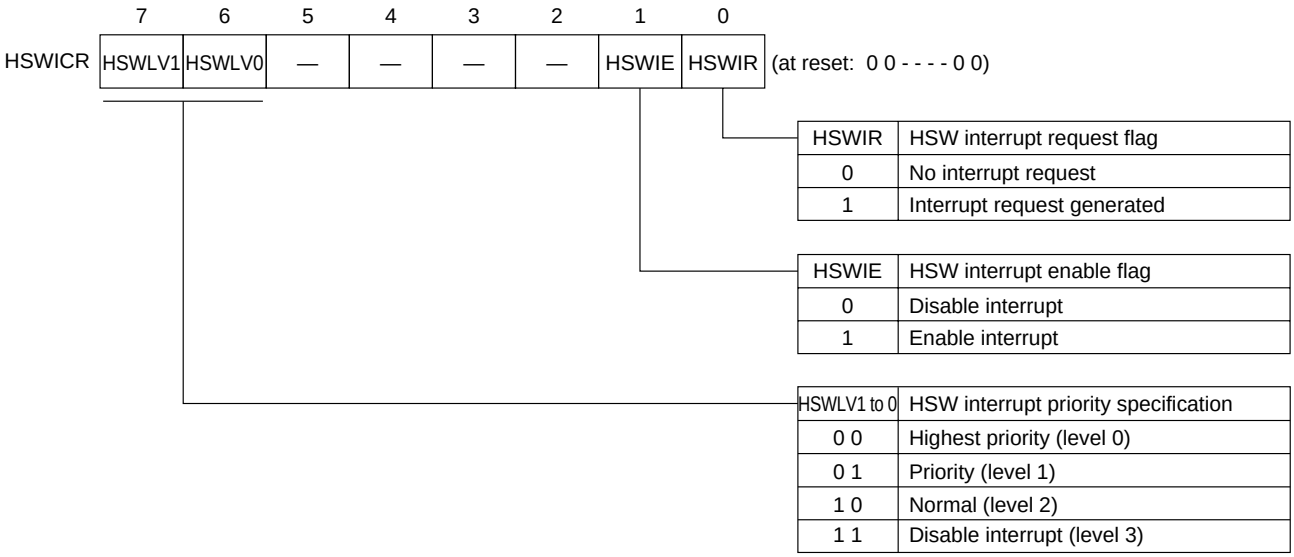
CTLIR	Control interrupt enable flag
0	No interrupt request
1	Interrupt request generated

CTLIE	Control interrupt request flag
0	Disable interrupt
1	Enable interrupt

CTLLV1 to 0	Control interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-197 Control Signal Interrupt Control Register (CTLICR: x'3FF0', R/W)

HSW Interrupt Control Register



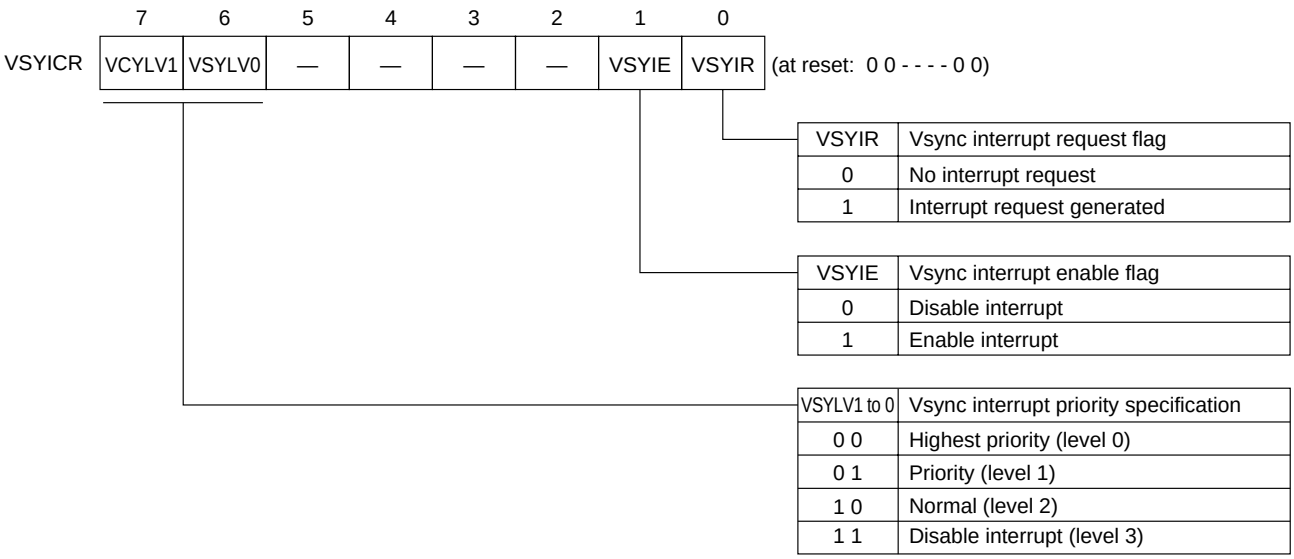
HSWIR	HSW interrupt request flag
0	No interrupt request
1	Interrupt request generated

HSWIE	HSW interrupt enable flag
0	Disable interrupt
1	Enable interrupt

HSWLV1 to 0	HSW interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-198 HSW Interrupt Control Register (HSWICR: x'3FF1', R/W)

VSYNC Interrupt Control Register



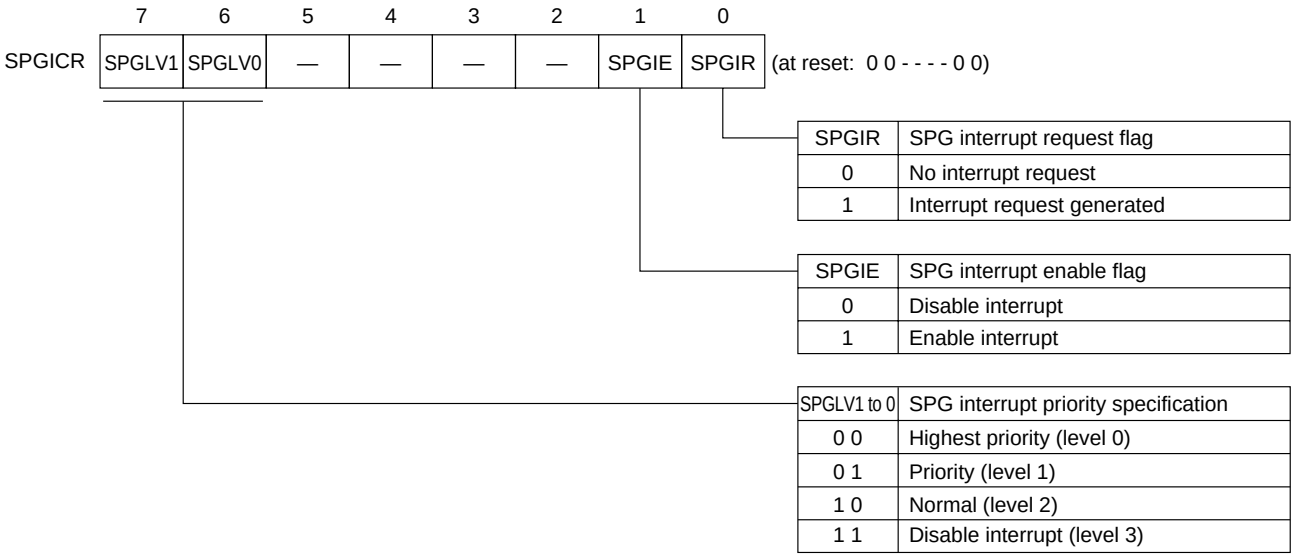
VSYIR	Vsync interrupt request flag
0	No interrupt request
1	Interrupt request generated

VSYIE	Vsync interrupt enable flag
0	Disable interrupt
1	Enable interrupt

VSYLV1 to 0	Vsync interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-199 VSYNC Interrupt Control Register (VSYICR: x'3FF2', R/W)

SPG Interrupt Control Register



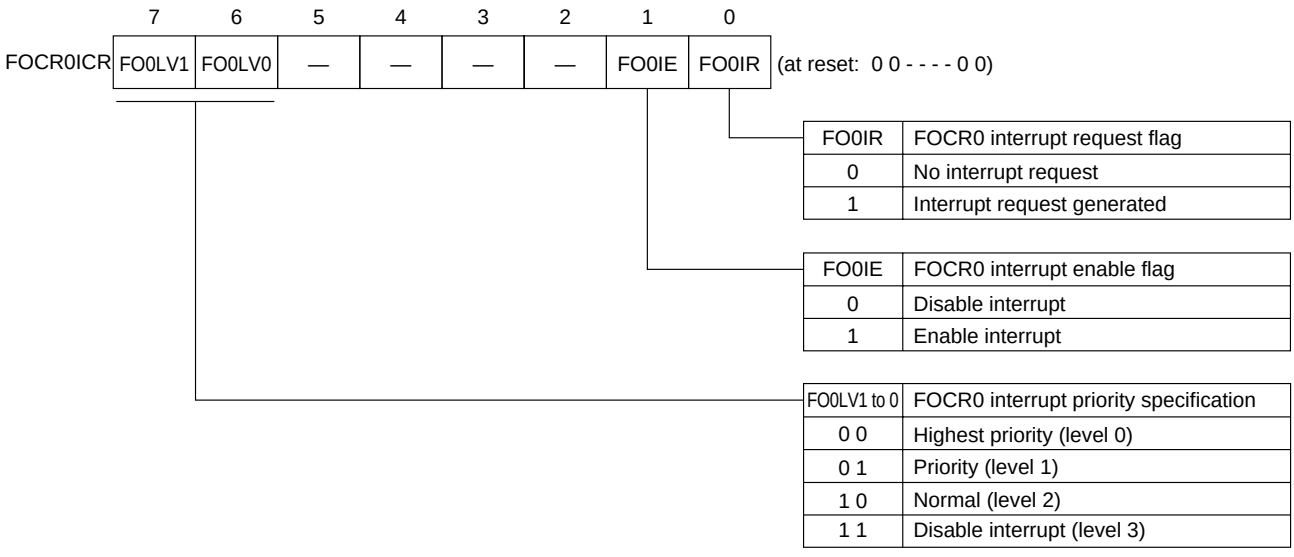
SPGIR	SPG interrupt request flag
0	No interrupt request
1	Interrupt request generated

SPGIE	SPG interrupt enable flag
0	Disable interrupt
1	Enable interrupt

SPGLV1 to 0	SPG interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-200 SPG Interrupt Control Register (SPGICR: x'3FF3', R/W)

FOCR0 Interrupt Control Register



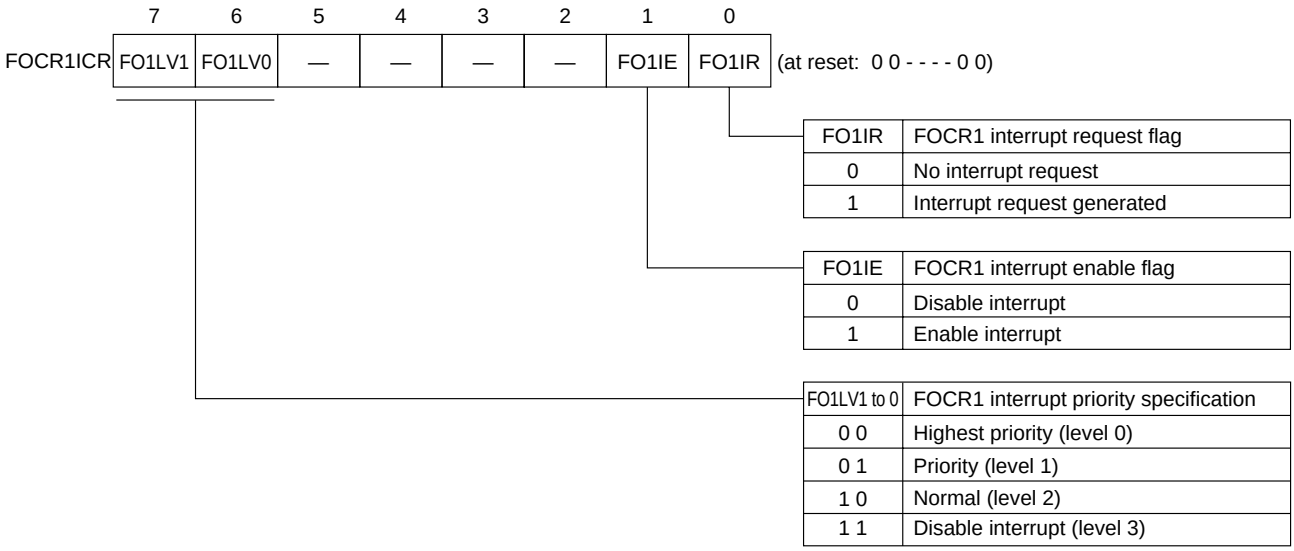
FO0IR	FOCR0 interrupt request flag
0	No interrupt request
1	Interrupt request generated

FO0IE	FOCR0 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

FO0LV1 to 0	FOCR0 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-201 FOCR0 Interrupt Control Register (FOCR0ICR: x'3FF4', R/W)

FOCR1 Interrupt Control Register



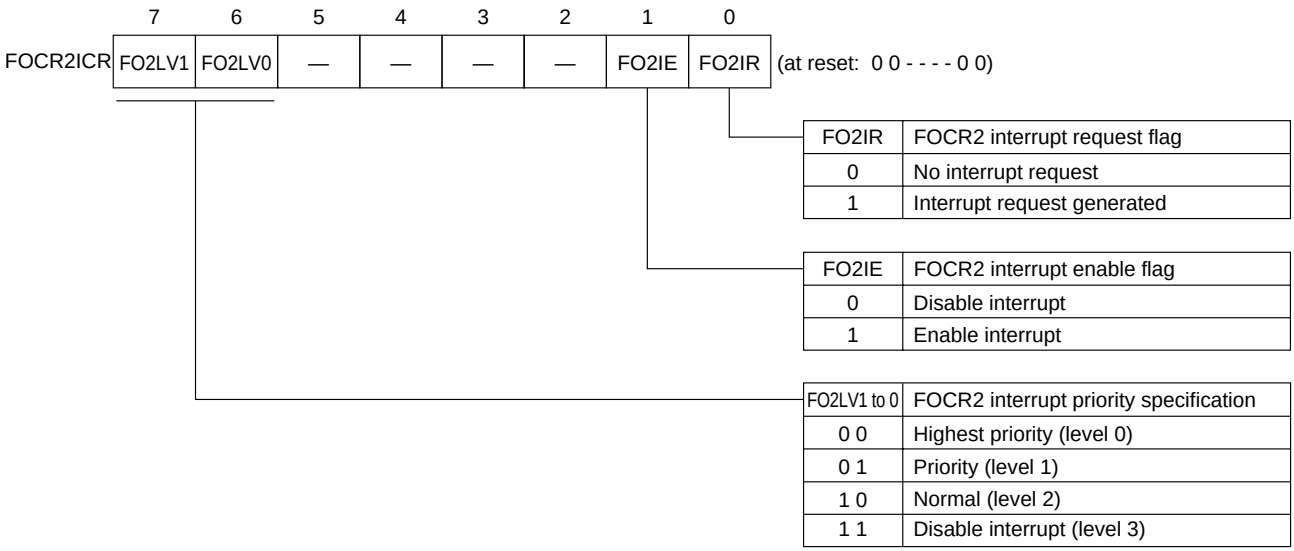
FO1IR	FOCR1 interrupt request flag
0	No interrupt request
1	Interrupt request generated

FO1IE	FOCR1 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

FO1LV1 to 0	FOCR1 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-202 FOCR1 Interrupt Control Register (FOCR1ICR: x'3FF5', R/W)

FOCR2 Interrupt Control Register



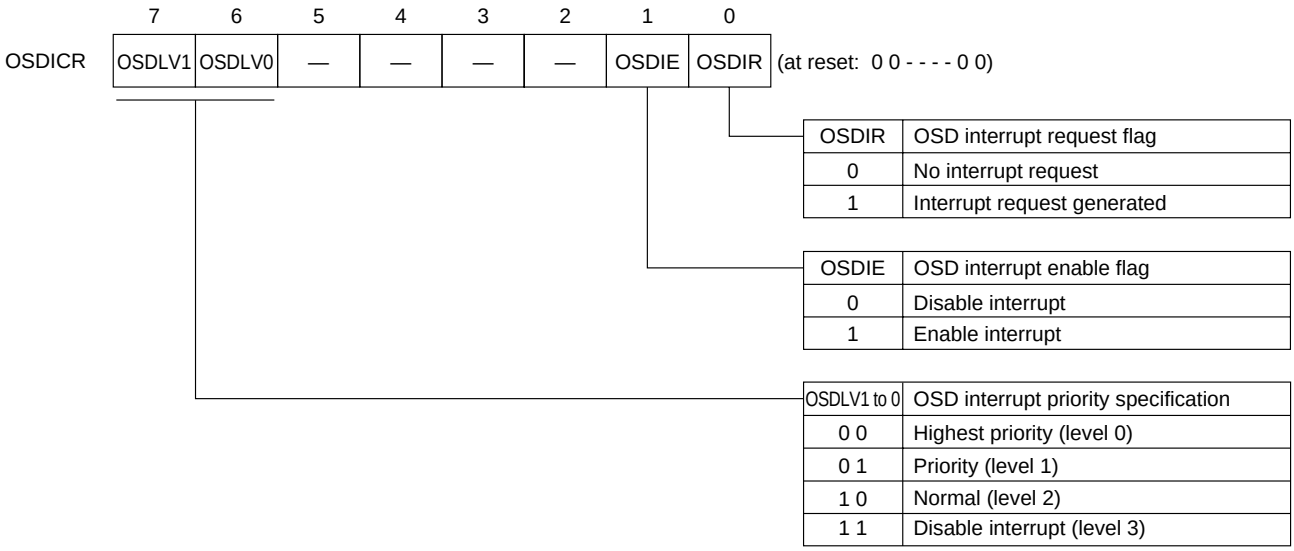
FO2IR	FOCR2 interrupt request flag
0	No interrupt request
1	Interrupt request generated

FO2IE	FOCR2 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

FO2LV1 to 0	FOCR2 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-203 FOCR2 Interrupt Control Register (FOCR2ICR: x'3FF6', R/W)

OSD Interrupt Control Register



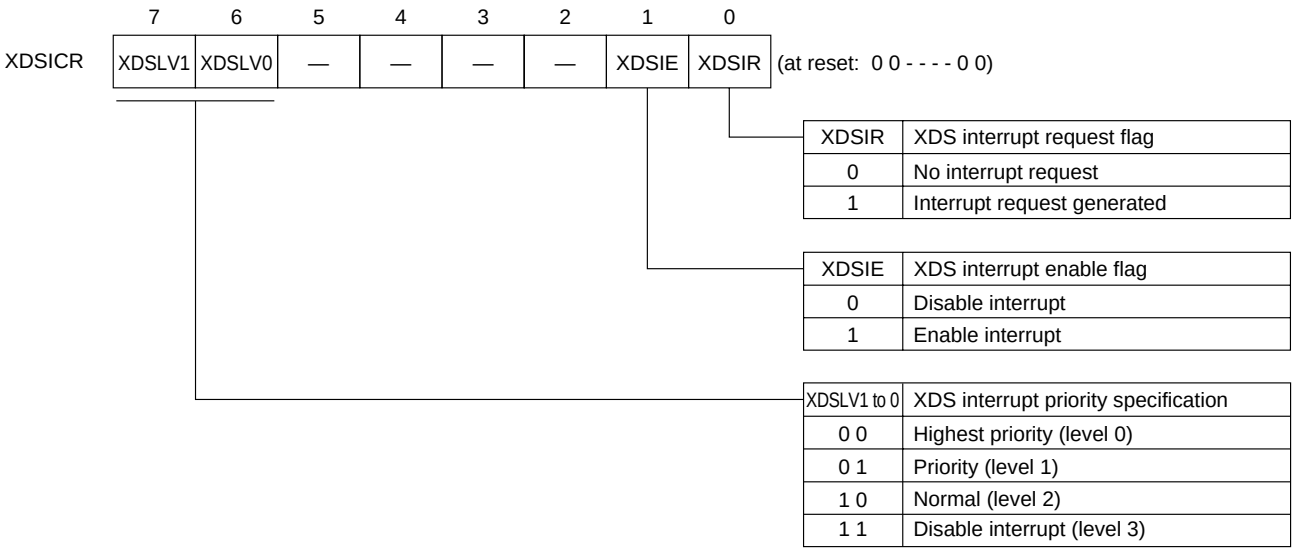
OSDIR	OSD interrupt request flag
0	No interrupt request
1	Interrupt request generated

OSDIE	OSD interrupt enable flag
0	Disable interrupt
1	Enable interrupt

OSDLV1 to 0	OSD interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-204 OSD Interrupt Control Register (OSDICR: x'3FF7', R/W)

XDS Interrupt Control Register



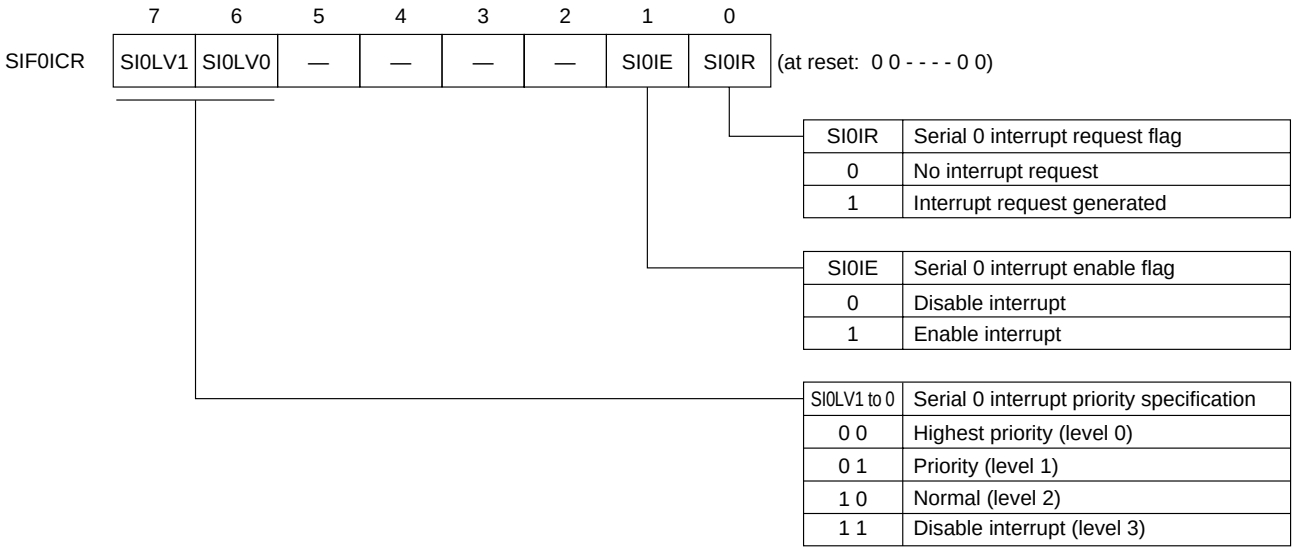
XDSIR	XDS interrupt request flag
0	No interrupt request
1	Interrupt request generated

XDSIE	XDS interrupt enable flag
0	Disable interrupt
1	Enable interrupt

XDSLV1 to 0	XDS interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-205 XDS Interrupt Control Register (XDSICR: x'3FF8', R/W)

Serial 0 Interrupt Control Register



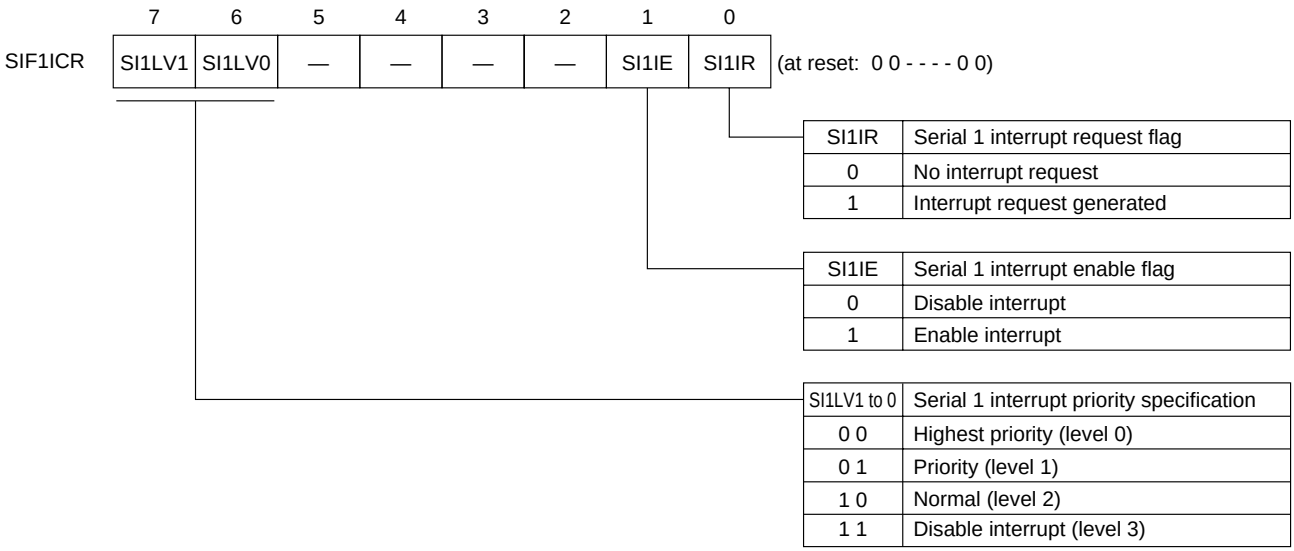
SI0IR	Serial 0 interrupt request flag
0	No interrupt request
1	Interrupt request generated

SI0IE	Serial 0 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

SI0LV1 to 0	Serial 0 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-206 Serial 0 Interrupt Control Register (SIF0ICR: x'3FF9', R/W)

Serial 1 Interrupt Control Register



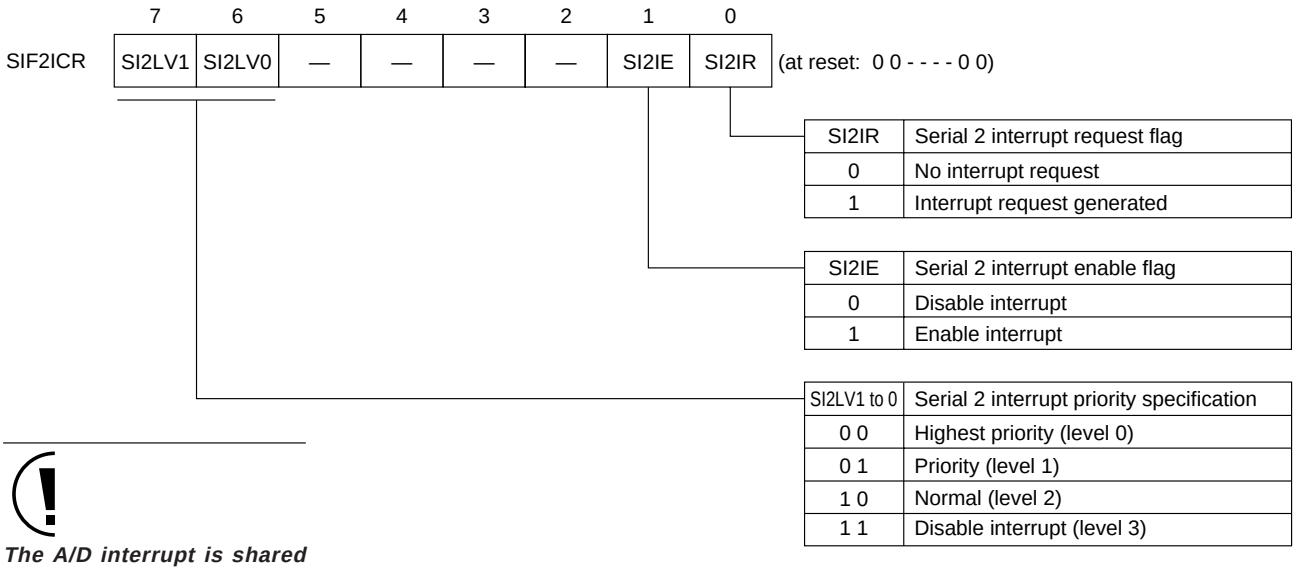
SI1IR	Serial 1 interrupt request flag
0	No interrupt request
1	Interrupt request generated

SI1IE	Serial 1 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

SI1LV1 to 0	Serial 1 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-207 Serial 1 Interrupt Control Register (SIF1ICR: x'3FFA', R/W)

Serial 2 Interrupt Control Register



SI2IR	Serial 2 interrupt request flag
0	No interrupt request
1	Interrupt request generated

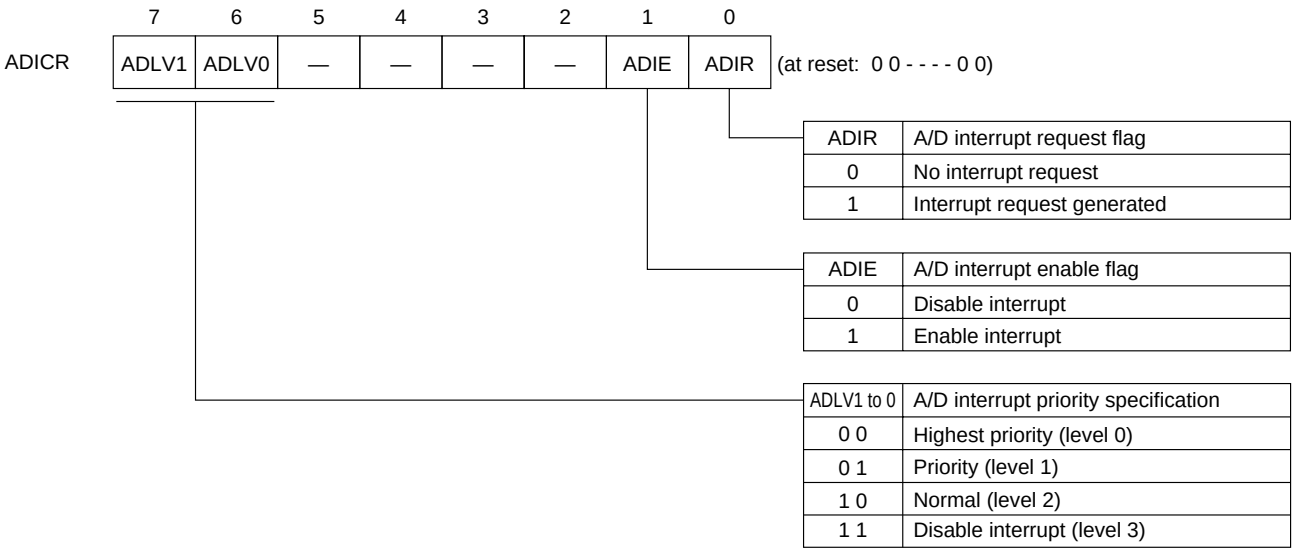
SI2IE	Serial 2 interrupt enable flag
0	Disable interrupt
1	Enable interrupt

SI2LV1 to 0	Serial 2 interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

The A/D interrupt is shared with the PWM14 interrupt. (Selected by the PWM14S flag.)

Figure 12-208 Serial 2 Interrupt Control Register (SIF2ICR: x'3FFB', R/W)

A/D Interrupt Control Register



ADIR	A/D interrupt request flag
0	No interrupt request
1	Interrupt request generated

ADIE	A/D interrupt enable flag
0	Disable interrupt
1	Enable interrupt

ADLV1 to 0	A/D interrupt priority specification
0 0	Highest priority (level 0)
0 1	Priority (level 1)
1 0	Normal (level 2)
1 1	Disable interrupt (level 3)

Figure 12-209 A/D Interrupt Control Register (ADICR: x'3FFC', R/W)

OSDVsync Interrupt Control Register

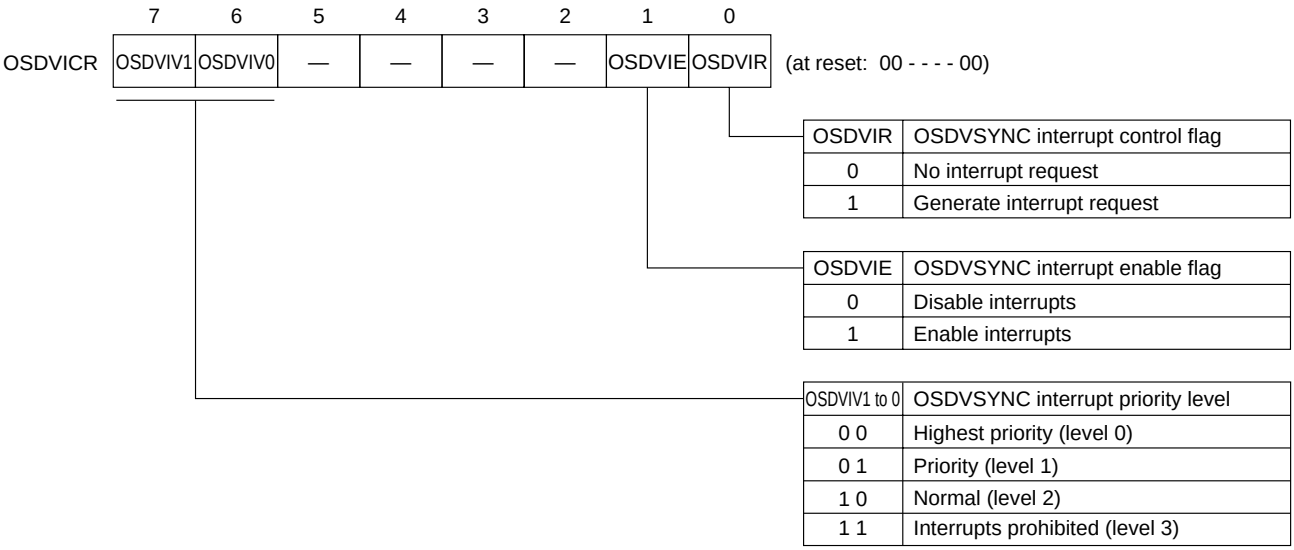


Figure 12-210 OSDVsync Interrupt Control Register (OSDVICR: x'3FFD', R/W)

Cylinder Speed Controller

Configuration of the Cylinder Speed Controller

By simply setting a reference value in the cylinder speed controller, the built-in cylinder speed control circuit will automatically output the PWM2 error. The cylinder speed controller consists of a register for the reference value, a binary counter that can be preset, a register for storing error data, and YFG (cylinder FG) signal processing circuitry.

[☞ Figure 12-211]

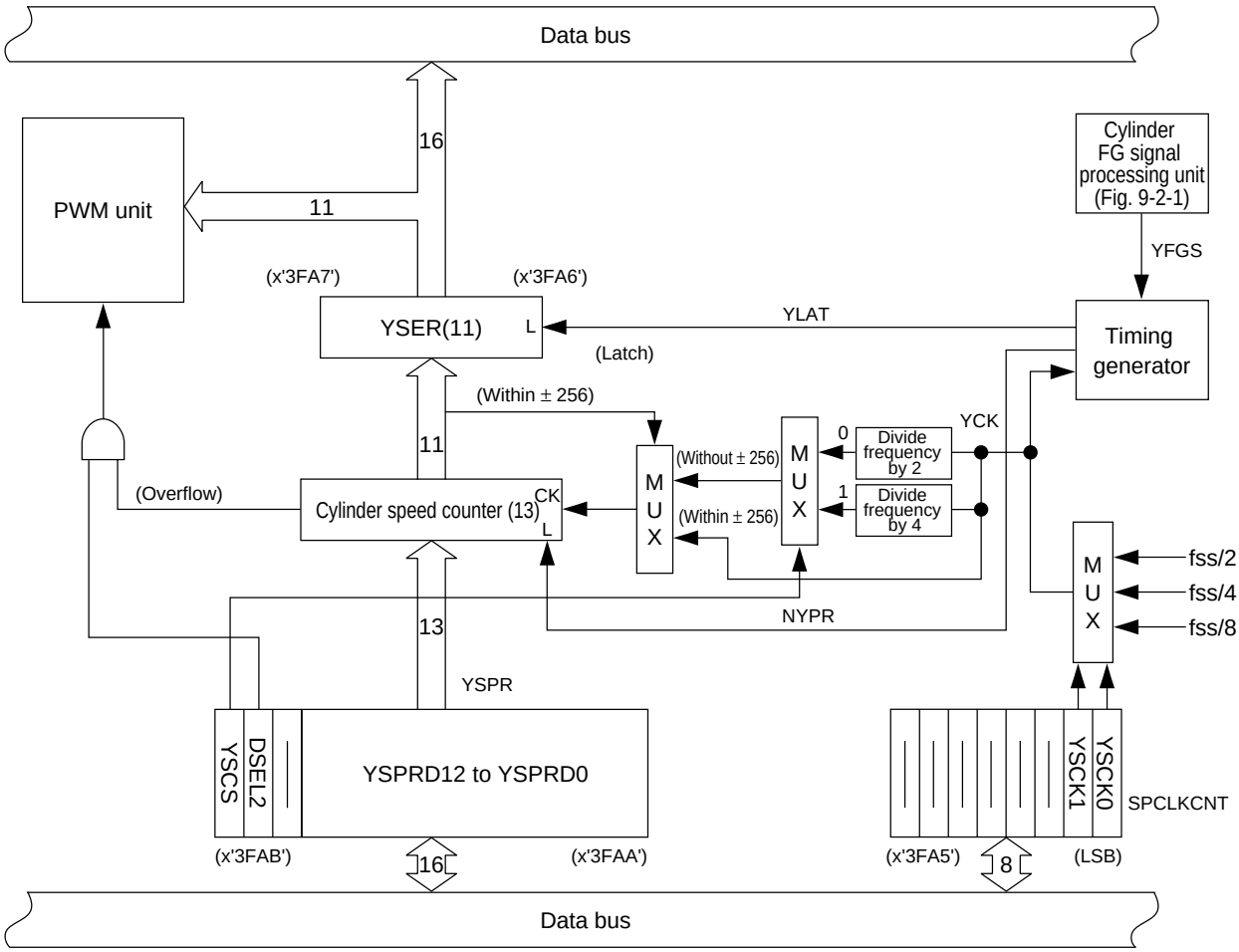


Figure 12-211 Cylinder Speed Controller Block Diagram

Cylinder Speed Control Registers

Cylinder speed is controlled by the following registers.

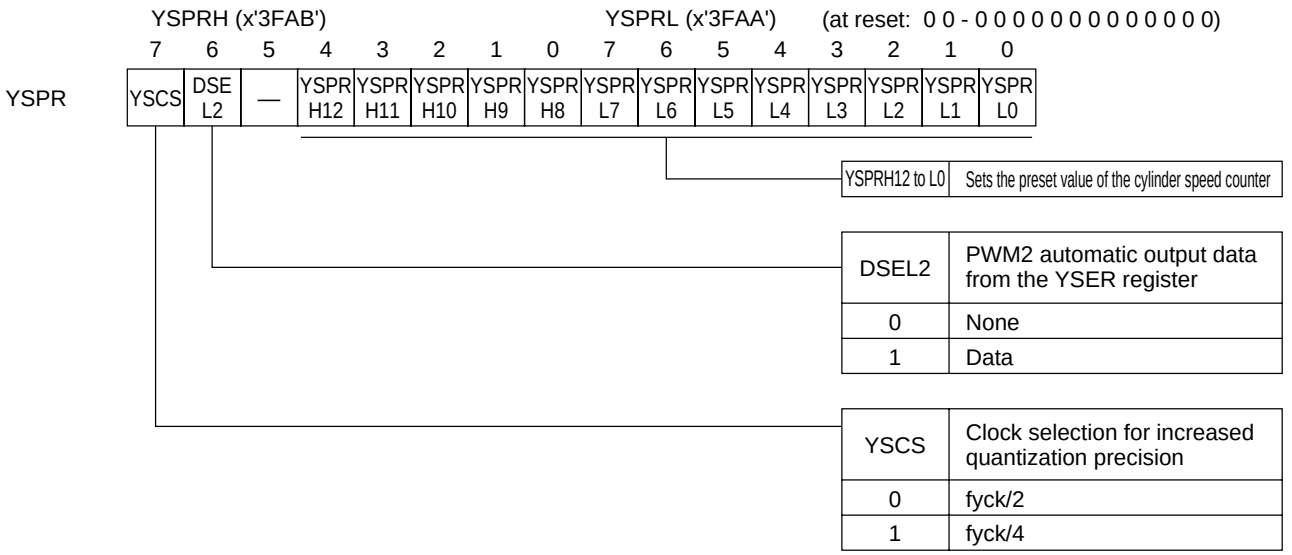
Table 12-2 Cylinder Speed Control Registers

Register Name		RAM Address	R/W	Function
YSPR	YSPRL	x'3FAA'	R/W	YFG Speed Counter Preset Register (lower)
	YSPRH	x'3FAB'		YFG Speed Counter Preset Register (upper)
YSER	YSERL	x'3FA6'	R	YFG Speed Error Register (lower)
	YSERH	x'3FA7'		YFG Speed Error Register (upper)
SPCLKCNT		x'3FA5'	R/W	Cylinder and Capstan Speed Counter Control Register

The cylinder speed control registers are described below.

(1) YFG Speed Counter Preset Register

The YSPR register is a 13-bit register into which the preset value is set for the cylinder speed control counter.



fyck is set by the SPCLKCNT register.

Figure 12-212 YFG Speed Counter Preset Register (YSPR: x'3FAA', x'3FAB' R/W)

YSPR D12 to YSPR D0

Use a MOVW instruction to set all 13 bits simultaneously.

Exercise care that this register is not regularly set.

The YSPRD0 through YSPRD12 flags (bp0 to bp12) set the preset value of the cylinder speed control counter. The cylinder FG signal immediately following a write operation to this register does not cause data to be latched from the YFG Speed Error Register (YSER) to the PWM Output Data Register.

YSCS

The YSCS flag (bp15) selects the clock for increased quantization precision. By setting a low precision count clock (fyck/2 or fyck/4) for the cylinder speed counter outside of the interval ±256 clock cycles from the center value (NC) of a sloped section, the dynamic range of the control can be expanded without decreasing the precision of quantization.

(2) YFG Speed Error Register

The YFG Speed Error Register is an 11-bit read-only register (with an overflag flag) that contains the cylinder speed error data. The overflow flag is set when the count value of the cylinder speed control counter is outside the sloped interval. The count value of the cylinder speed control is compared to the center value (NC) of the sloped interval and the result is stored in the YSER register. Therefore, negative error data will also exist.

When reading the contents of the YFG Speed Error Register, since the timing may cause erroneous data to be read, be sure to read the values twice and compare the results.

When the error is '0', the lower 11 bits of the value read from the register are all zeros and the upper 5 bits of data are undefined.

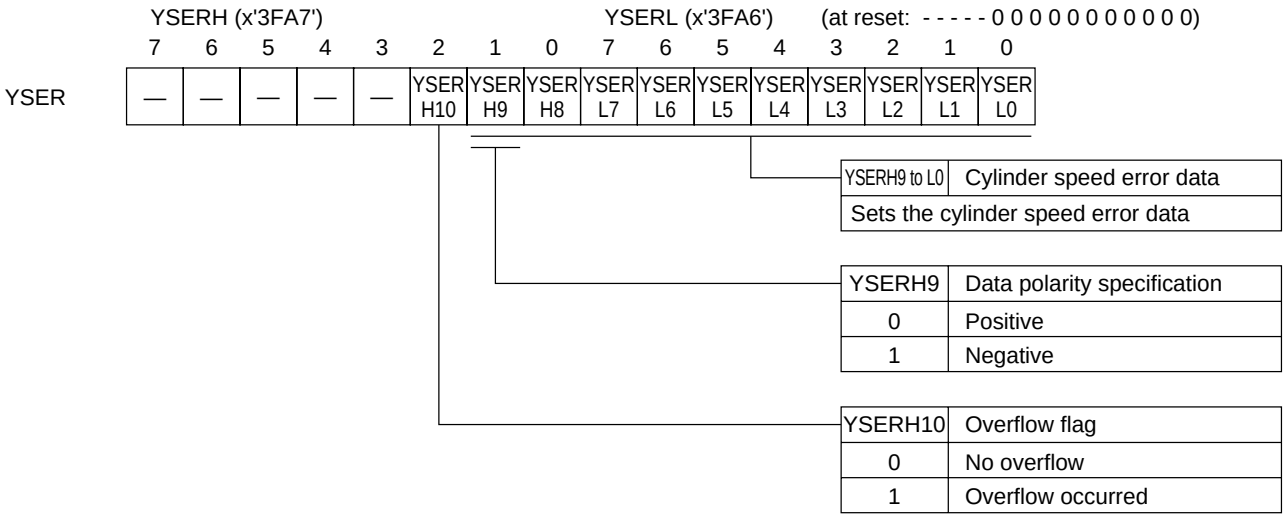
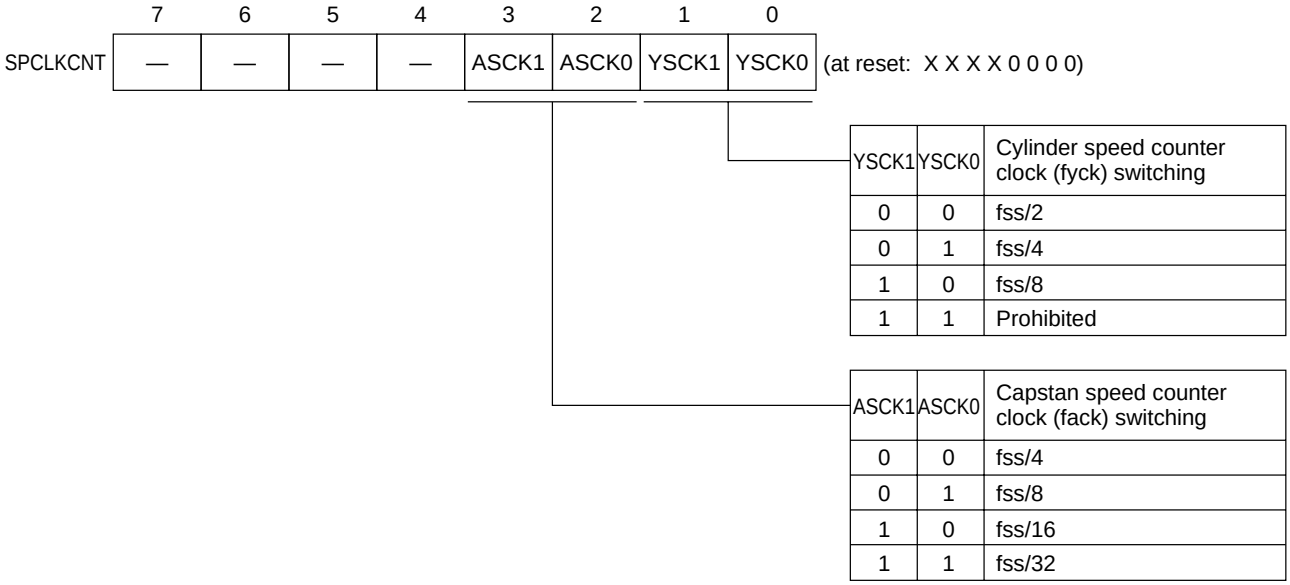


Figure 12-213 YFG Speed Error Register (YSER: x'3FA6', x'3FA7' R)

(3) Cylinder and Capstan Speed Counter Control Register

The SPCLKCNT register switches the cylinder and capstan speed counter clocks.



YSCK1	YSCK0	Cylinder speed counter clock (fyck) switching
0	0	fss/2
0	1	fss/4
1	0	fss/8
1	1	Prohibited

ASCK1	ASCK0	Capstan speed counter clock (fack) switching
0	0	fss/4
0	1	fss/8
1	0	fss/16
1	1	fss/32

Figure 12-214 Cylinder and Capstan Speed Counter Control Register (SPCLKCNT: x'3FA5' R/W)

YSCK1 to YSCK0

The YSCK0 and YSCK1 flags (bp0 and bp1) set one of three frequencies (fss/2, fss/4 or fss/8) as the input clock (YCK) for the cylinder speed control counter unit.

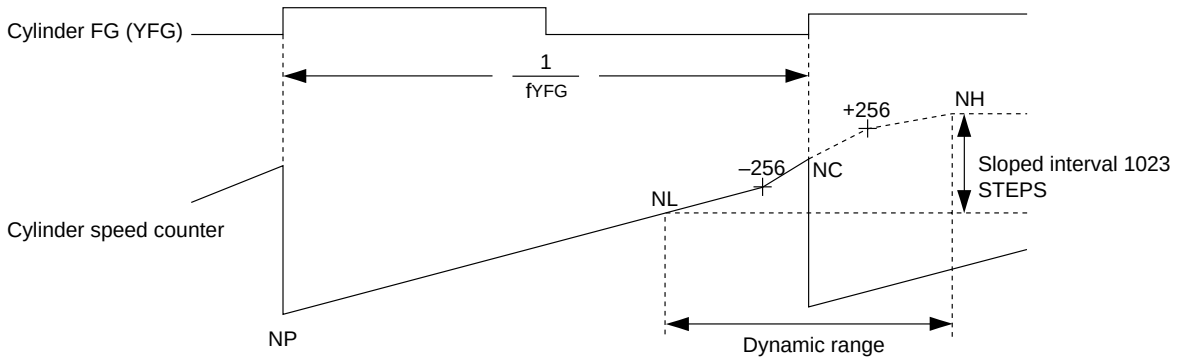
Cylinder Speed Control Reference Value (Preset Value) Setting

The count clock for the cylinder speed counter that can be preset is set for high precision (fyck) within the interval ±256 clock cycles from the center value (NC) of a sloped section. However, low precision (fyck/2 or fyck/4) is set for regions outside this section. Therefore, the dynamic range of the control can be expanded without decreasing the precision of quantization.

The YSCS flag (bit 15) of the YFG Speed Counter Preset Register (YSPR) selects a clock frequency of either fyck/2 or fyck/4 for the regions of low clock precision. The YSCK0 and YSCK1 flags (bp0 and bp1) of the Cylinder and Capstan Speed Counter Control Register (SPCLKCNT) can select one of three frequencies (fss/2, fss/4, or fss/8) as the input clock frequency (fyck) for the cylinder speed counter. If the YSCS, YSCK0 and YSCK1 flags are set, the cylinder speed control reference value (NP) can be computed.

[☞ Figure 12-215]

In addition, there is also a function to control the cylinder phase error output (PWM0) by means of the cylinder speed error. If the cylinder speed error is outside the dynamic range (outside of range from NL to NH), the duty of the cylinder phase error output (PWM0) waveform will be fixed at 50%.



The following equations describe the case where theYSCS flag of the YSPR Register has selected a count clock frequency of fyck/n as the cylinder speed counter for the interval outside the sloped section ±256.

$$\frac{1}{f_{YFG}} = \frac{(4+256)}{fyck} + \frac{(NL+256-NP) \times n}{fyck}$$

here,

$$NL = 8191 - 1023 = 7168$$

Counts of the cylinder speed counter
Max. value $2^{13}-1=8191$
Sloped interval $2^{10}-1=1023$

$$\frac{1}{f_{YFG}} = \frac{260}{fyck} + \frac{(7424-NP) \times n}{fyck}$$

$$\frac{NP \times n}{fyck} = \frac{260}{fyck} + \frac{7424 \times n}{fyck} - \frac{1}{f_{YFG}}$$

∗: 4

delay from when count value of cylinder speed counter is latched until it is preset (= 4 clocks) ∗

$$NP = 7424 + \frac{260}{n} - \frac{1}{n} \times \frac{fyck}{f_{YFG}}$$

(1) When the YSCS flag (bit 15 of YSPR) = '0',

$n = 2$

therefore,

$$NP = 7554 - \frac{1}{2} \times \frac{fyck}{f_{YFG}}$$

(2) When the YSCS flag (bit 15 of YSPR) = '1',

$n = 4$

therefore,

$$NP = 7489 - \frac{1}{4} \times \frac{fyck}{f_{YFG}}$$

where,
f_{YFG} : Cylinder FG frequency
fyck : Input clock frequency of cylinder speed counter
(selected by YSCK0 and YSCK1)

Figure 12-215 Computation of Cylinder Speed Control Reference Value

Capstan Speed Controller

Configuration of Capstan Speed Controller

By simply setting a reference value in the capstan speed controller, built-in capstan speed control circuitry will automatically compute the error. The capstan speed controller consists of a register for setting reference values, a binary counter that can be reset, a register for storing error data, and capstan FG signal processing circuitry.

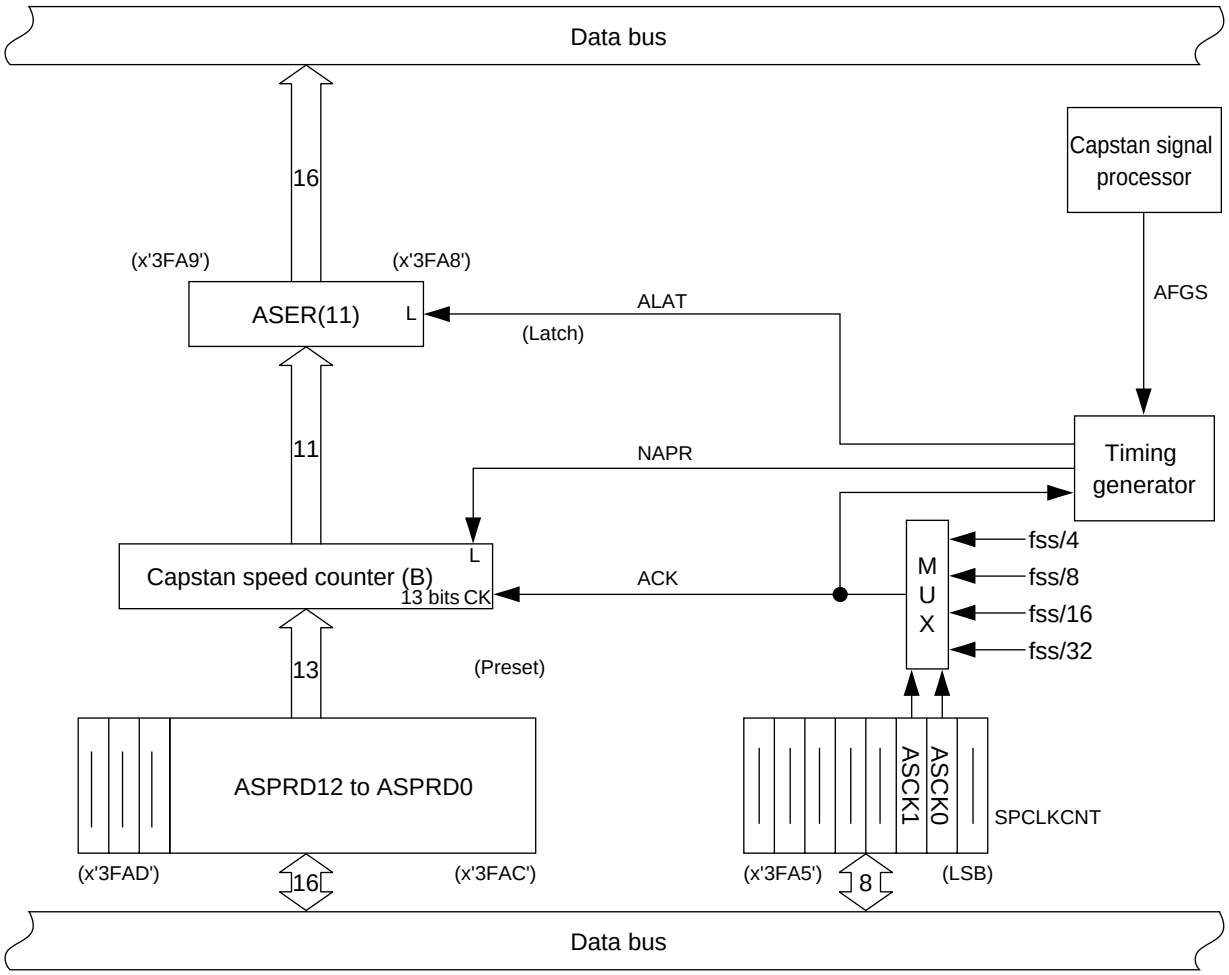


Figure 12-216 Capstan Speed Controller Block Diagram

Capstan Speed Control Register

Capstan speed is controlled by the following registers.

Table 12-3 Capstan Speed Control Registers

Register Name		RAM Address	R/W	Function
ASPR	ASPRL	x'3FAC'	R/W	AFG Speed Counter Preset Register (lower)
	ASPRH	x'3FAD'		AFG Speed Counter Preset Register (upper)
ASER	ASERL	x'3FA8'	R	AFG Speed Error Register (lower)
	ASERH	x'3FA9'		AFG Speed Error Register (upper)
SPCLKCNT		x'3FA5'	R/W	Cylinder and Capstan Speed Counter Control Register

Each register is described below.

(1) AFG Speed Counter Preset Register

The ASPR register is a 13-bit register that sets the preset value for the capstan speed control counter.

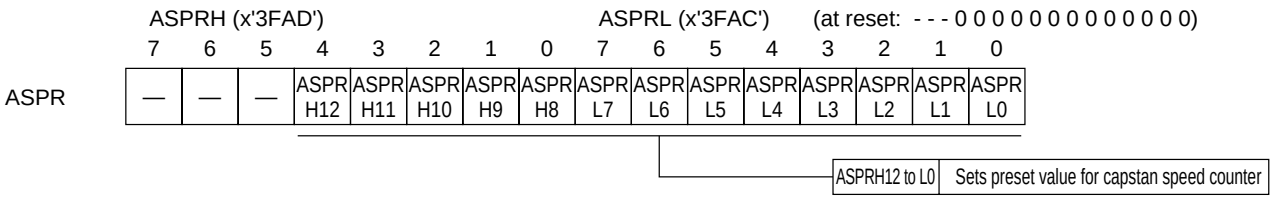


Figure 12-217 AFG Speed Counter Preset Register
(ASPR: x'3FAC', x'3FAD' R/W)

The ASPRD0 to ASPRD12 flags (bp0 to bp12) set the preset value for the capstan speed control counter. The capstan FG signal that immediately follows a write operation to this register does not cause data to be latched from the AFG Speed Error Register (ASER) to the PWM Output Data Register.

ASPR D12 to ASPR D0

Use a MOVW instruction to set all 13 bits simultaneously.

Exercise care that this register is not regularly set.

(2) Capstan Speed Error Register

The ASER Register is an 11-bit read-only register (with an overflow flag) that contains the capstan speed error data. The overflow flag is set when the count value of the capstan speed control counter is outside the sloped interval. The count value of the capstan speed control is compared to the center value (NC) of the sloped interval and the result is stored in the ASER register. Therefore, negative error data will also exist.

When reading the contents of the ASER Register, since the timing may cause erroneous data to be read, be sure to read the values twice and compare the results.

When the error is '0', the lower 11 bits of the value read from the register are all zeros and the upper 5 bits of data are undefined.

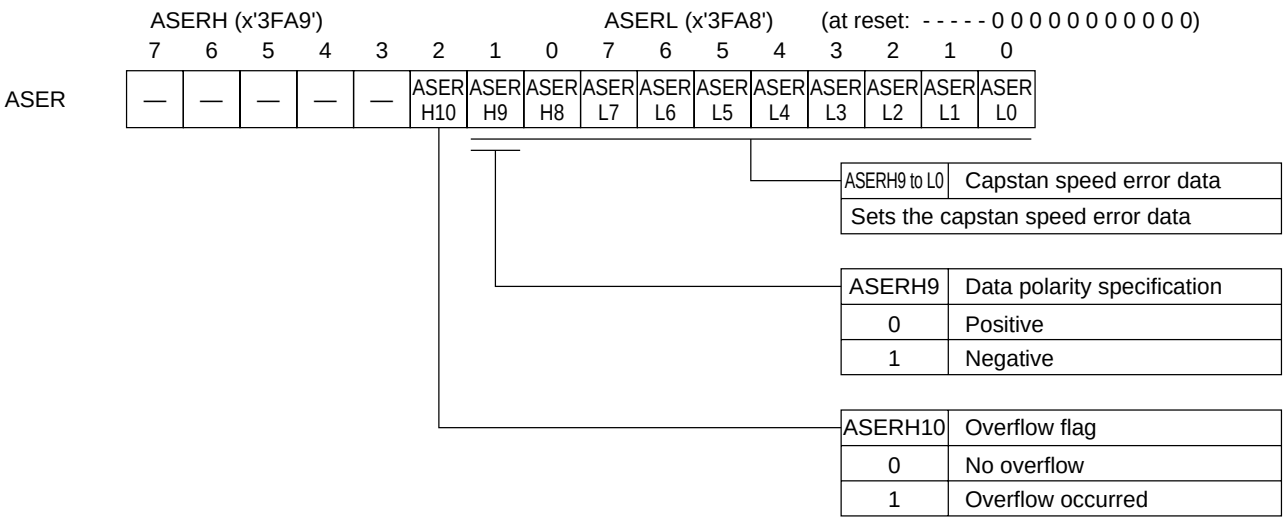
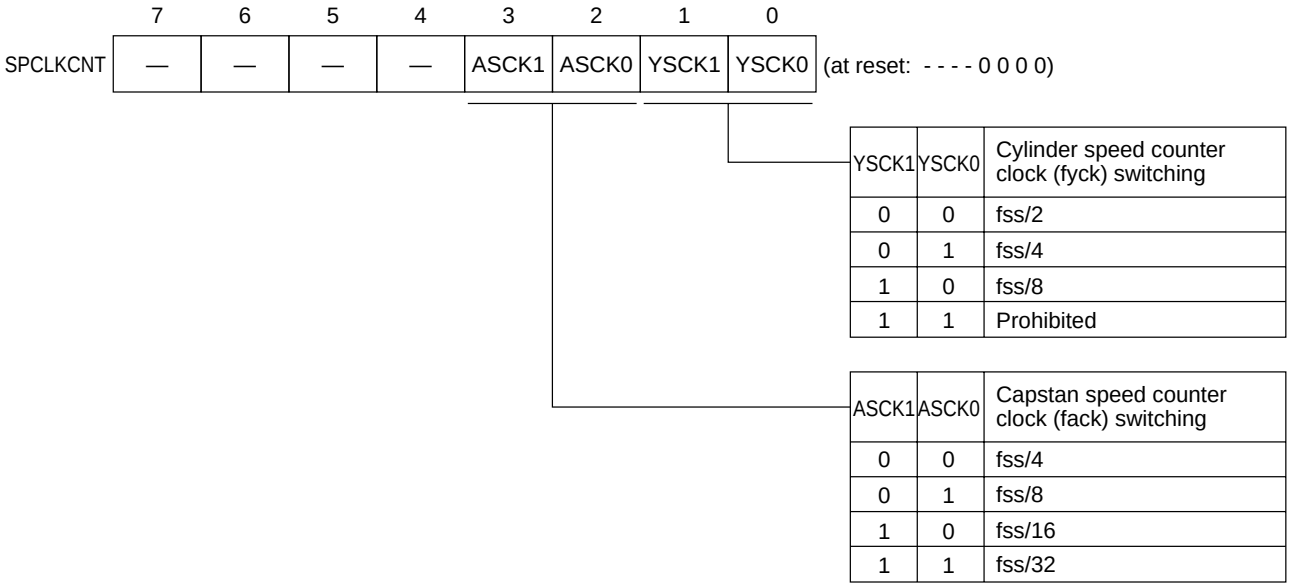


Figure 12-218 AFG Speed Error Register (ASER: x'3FA8', x'3FA9' R)

(3) Cylinder and Capstan Speed Counter Control Register

The SPCLKCNT register changes the cylinder and capstan speed counter clocks by switching the clock input to the cylinder and capstan speed control counter unit.



YSCK1	YSCK0	Cylinder speed counter clock (fyck) switching
0	0	fss/2
0	1	fss/4
1	0	fss/8
1	1	Prohibited

ASCK1	ASCK0	Capstan speed counter clock (fack) switching
0	0	fss/4
0	1	fss/8
1	0	fss/16
1	1	fss/32

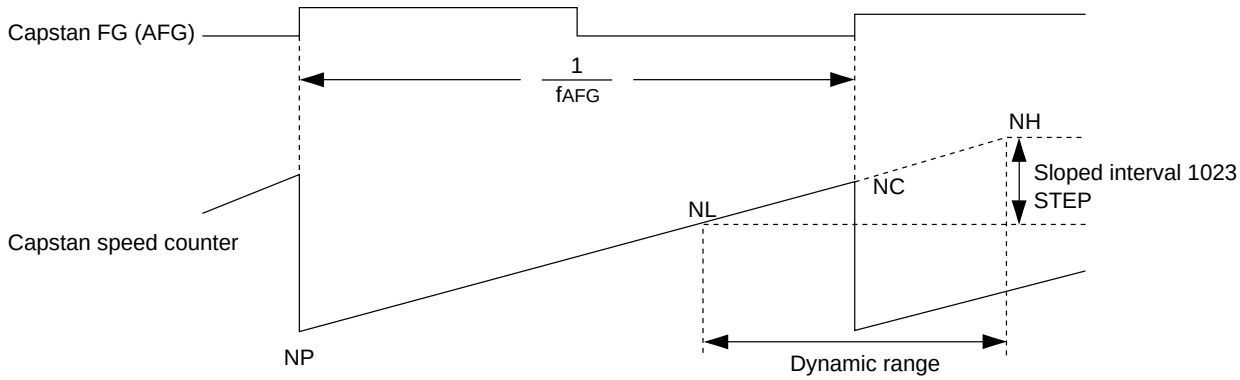
Figure 12-219 Cylinder and Capstan Speed Counter Control Register (SPCLKCNT: x'3FA5' R/W)

ASCK1 to ASCK0

The ASCK0 and ASCK1 flags (bp2 and bp3) set one of four frequencies (fss/4, fss/8, fss/16 or fss/32) as the input clock (ACK) for the capstan speed control counter.

Capstan Speed Control Reference Value (Preset Value) Setting

The ASCK0 and ASCK1 flags (bits 2 and 3) of the Cylinder and Capstan Speed Counter Control Register (SPCLKCNT: x'3FA5', R/W) can select one of four frequencies (fss/4, fss/8, fss/16 or fss/32) as the input clock frequency (fack) to the capstan speed counter. If the ASCK0 and ASCK1 flags are set, the cylinder speed control reference value (NP) can be computed.
[☞ Figure 12-220]



The sloped section has 1023 STEPS
By denoting the sloped section as m, the following equations describe this case.

$$\frac{1}{f_{AFG}} = \frac{5}{fack} + \frac{NC-NP}{fack}$$

$$\frac{NP}{fack} = \frac{5}{fack} + \frac{NC-NP}{f_{AFG}} - \frac{5}{f_{AFG}}$$

$$NP = 5 + NC - \frac{fack}{f_{AFG}}$$

here,

Maximum count value of the capstan speed counter
$2^{13}-1=8191$

$NC = 8191 - m/2$

delay from when count value of capstan speed counter is latched until it is preset (= 5 clocks)*

$$NP = 8196 - \frac{m}{2} - \frac{fack}{f_{AFG}}$$

therefore, since m=1023,

$$NP = 7685 - \frac{fack}{f_{AFG}}$$

where,
f_{AFG} : Capstan FG frequency
fack : Input clock frequency of capstan speed counter
(selected by ASCK0 and ASCK1)

Figure 12-220 Computation of Capstan Speed Control Reference Value (NP)

List of Instructions

The MN101D00 series assembler instruction set consists of 37 instructions organized as shown in the following table.
[See the "MN101D00 Series Instruction Set Manual" for further details.]

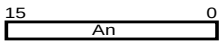
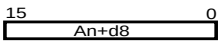
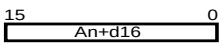
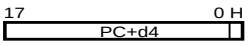
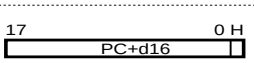
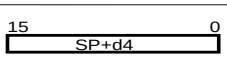
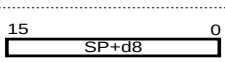
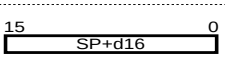
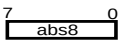
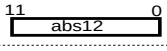
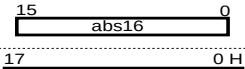
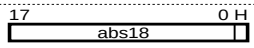
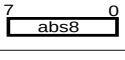
Table 12-4 List of Instructions

Instruction group	Instruction	Function
Data transfer instructions	MOV	8-Bit data move between memory and register
	MOVW	16-Bit data move between memory and register
	PUSH	Save register contents
	POP	Restore register contents
	EXT	Sign extension
Arithmetic operation instructions	ADD	8-Bit addition
	ADDC	Addition with carry
	ADDW	16-Bit addition
	ADDUW	16-Bit addition with zero extension
	ADDSW	16-Bit addition with sign extension
	SUB	8-Bit subtraction
	SUBC	Subtraction with borrow
	SUBW	16-Bit subtraction
	MULU	8-Bit unsigned multiplication
	MULWU	16-Bit unsigned multiplication
	MULW	16-Bit signed multiplication
	DIVU	Unsigned division (16 ÷ 8 bits)
	DIVWU	Unsigned division (32 ÷ 16 bits)
	CMP	8-Bit comparison
	CMPW	16-Bit comparison
Logical operation instructions	AND	Logical product
	OR	Logical sum
	XOR	Logical exclusive sum
	NOT	Inversion (1's complement)
	ASR	Arithmetic right shift
	LSR	Logical right shift
	ROR	Rotate right
Bit manipulation instructions	BSET	Bit test and set (processing unit is a byte)
	BCLR	Bit test and clear (processing unit is a byte)
	BTST	Bit test
Branch instructions	Bcc	Conditional branch (relative to PC)
	CBcc	Compare and conditional branch (relative to PC)
	TBcc	Bit test and conditional branch (relative to PC)
	JMP	Unconditional branch (absolute, register indirect)
	JSR	Subroutine call (absolute, register indirect branch)
	JSRV	Subroutine call (vector table indirect branch)
	NOP	No operation
	RTS	Return from subroutine
	RTI	Return from maskable interrupt
Control instruction	REP	Repeat

Addressing Modes

The MN101D00 series supports the following nine addressing modes. During register operations, it is possible to use the two types of addressing modes of register direct addressing and immediate addressing modes. [See the "MN101D00 Series Instruction Set Manual" for further details.]

Table 12-5 Summary of Addressing Modes

Addressing mode		Effective address	Description
Register direct	Dn/DWn An/SP PSW	——	The register is specified directly. The registers that can be specified are the internal registers.
Immediate value	imm3/imm4 imm8/imm16	——	The operand value, mask value, etc., is specified directly along with the op code.
Register indirect	(An)		The address is specified through the address register.
Register relative indirect	(d8, An)		The address is specified through the address register and an 8-bit displacement.
	(d16, An)		The address is specified through the address register and a 16-bit displacement.
	(d4, PC) (Branch instruction only)		The address is specified through the program counter, a 4-bit displacement, and an H bit.
	(d7, PC) (Branch instruction only)		The address is specified through the program counter, a 7-bit displacement, and an H bit.
	(d11, PC) (Branch instruction only)		The address is specified through the program counter, an 11-bit displacement, and an H bit.
	(d12, PC) (Branch instruction only)		The address is specified through the program counter, a 12-bit displacement, and an H bit.
	(d16, PC) (Branch instruction only)		The address is specified through the program counter, a 16-bit displacement, and an H bit.
Stack relative indirect *2	(d4, SP)		The address is specified through the stack pointer and a 4-bit displacement.
	(d8, SP)		The address is specified through the stack pointer and an 8-bit displacement.
	(d16, SP)		The address is specified through the stack pointer and a 16-bit displacement.
Relative	(abs8)		The address is specified by an operand value appended to the op code. The specified address can be made to have an optimal operand length.
	(abs12)		
	(abs16)		
	(abs18) (Branch instruction only)		
RAM Short addressing	(abs8)		Specified by an 8-bit offset from the address location x'00000'.
I/O Short addressing	(io8)		Specified by an 8-bit offset from the top address of the special registers area.
Handy addressing	(HA)	——	This is the addressing mode in which the address during the previous memory access is used again, and can only used with the MOV and MOVW instructions. It is possible to reduce the code size by combining this mode with absolute addressing.

*Note 1: H denotes a half-byte bit.
*Note 2: During an interrupt, the stack relative indirect addressing is made relative to SPI.

Internal Registers

Address Registers

The registers in the MN101D00 series can be classified into the internal registers in the CPU core and the special registers for control. The internal registers consist of the address registers and operation registers, and the processor status word (PSW).

The contents of AN, SP, and SPI will be uncertain immediately after a reset start. Carry out initial settings for these registers in the initialization program.

The address registers are the 19-bit program counter (PC), the addressing register (An), and the stack pointers (SP, SPI).

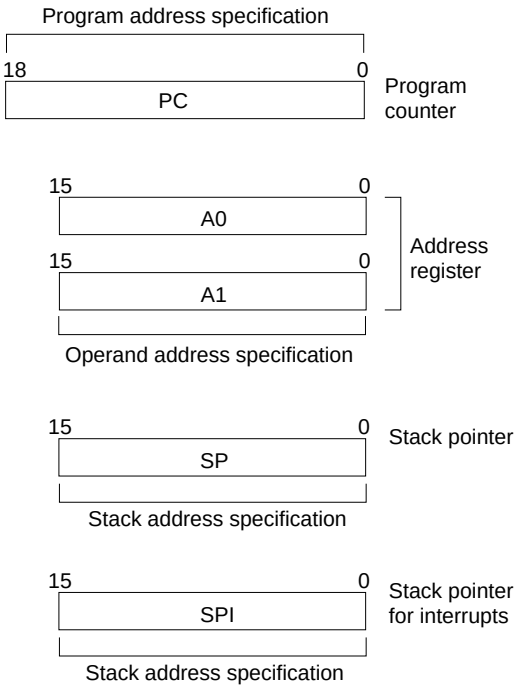


Figure 12-221 Organization of Address Registers

■ Program Counter (PC)

This is the register pointing to the address of the instruction under execution. Since the instruction separator denotes a 256KB instruction space in units of 4 bits, the program counter has a length of 19 bits. Although the op codes of the program can use 256KB as a linear space, it is necessary to place the table data in the space of the leading 64KB. In the case of subroutine call instructions, these 19 bits are saved on the stack (3 bytes) as the return address.

■ Address Registers (A0, A1)

These are the registers used as address pointers and are used by instructions for address computation (addition, subtraction, comparison). The contents of the address registers are actually pointers (2-byte data), and the data transfer between these registers and memory is always done in 16-bit units. (There is no need for the address to be aligned for such 16-bit transfers with the memory.)

■ Stack Pointer (SP)

This is the register that constantly points to the current top address of the stack area. The stack pointer is decremented during saving (pushing) and incremented during data restoration popping).

■ Stack Pointers for Interrupts (SPI)

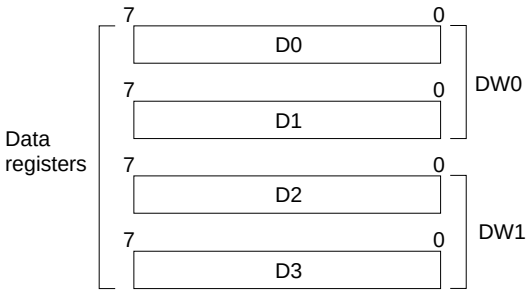
When an interrupt is acknowledged, this stack pointer points to the top address of the stack area. This stack pointer is decremented when data is saved (pushed) and incremented when the data is restored (popped).

SPI is a stack pointer dedicated to interrupts. The pushing of PC/PSW at the time of interrupt acknowledgement, saving of PC during the execution of a JSR instruction and PUSH/POP instructions within the interrupt servicing routine are executed for the RAM space pointed to by the SPI register. The SPF flag of the PSW is set while the CPU is using SPI. The SPF flag is set at the time an interrupt is acknowledged and is reset when an RTI instruction is executed. However, this flag is not reset by the RTI instruction when nested interrupts are being serviced. In addition, it is also possible to set/reset this flag using instructions. The contents of the SPI register will not be definite at the time of a recovery from a reset. Always make sure to initialize the SPI register after recovery from a reset. The following operations are required for such initialization.

(1) Disabling all interrupts.
(2) Setting the SPF flag of PSW.
(3) Initializing SPI using the MOV An,SP instruction.
(4) Resetting the SPF flag of PSW.
(5) Enabling interrupts as required.

Operation Registers

The operation registers are organized as 8-bit data registers (Dn).



The contents of the registers Dn will not be definite after a reset start. Carry out initial setting of the registers in the initialization program.

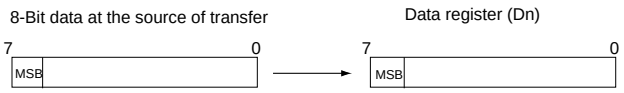
Figure 12-222 Operation Register Organization

■ Data Registers (D0 ~ D3)

The data registers (D0 ~ D3) are 8-bit general purpose registers and can be used in all arithmetic operations, logical operations, and shift operations. Further, all these registers can be used for data exchange with the memory.

These data registers can handle 16-bit data by combining D0, D1 and D2, D3.

- 8-Bit Transfer



- 16-Bit Transfer

Figure 12-223

Processor Status Word

The processor status word (PSW) is an 8-bit register containing various flags such as the status of the interrupt control circuit and result of operations of the CPU.

The four flags VF, NF, CF, and ZF indicate the result of operation and can be used by the branch instruction (the Bcc instruction) in the program. All these flags will be reset to '0' at the time of a reset start. IME1~0 and MIE are flags controlling interrupts, and will be reset to IM0, IM1 = '00', and IME = '0' at the time of a reset start.

The contents of PSW are automatically saved on to the stack area when an interrupt is acknowledged, and restored from the stack at the time of a return from interrupt.

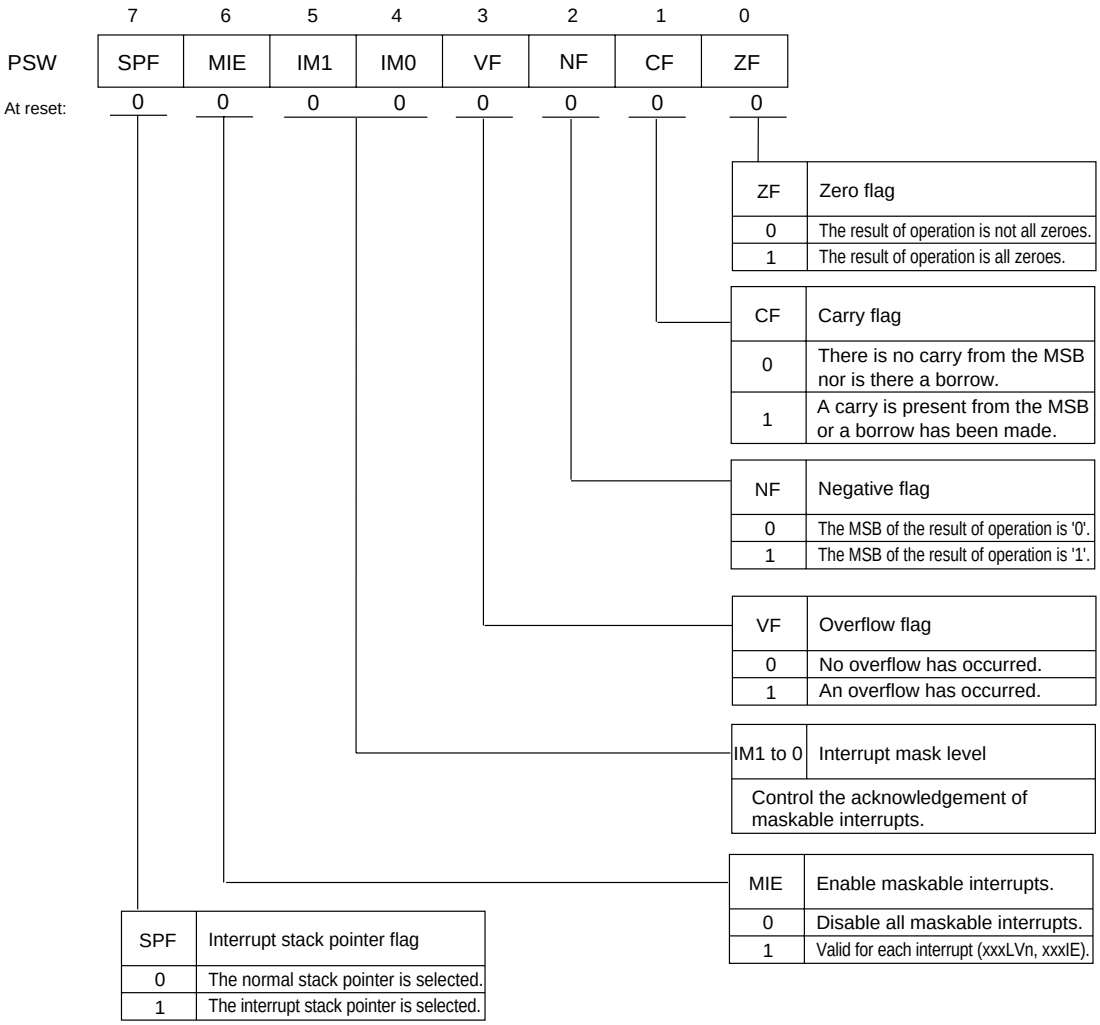


Figure 12-224 Processor Status Word Organization

■ Zero flag (ZF)

This flag becomes '1' when all bits of the result of operation are '0', and becomes '0' otherwise.

■ Carry flag (CF)

This flag is set to '1' when there is a carry or a borrow at the MSB in the result of calculation, and is reset to '0' when there is no carry or borrow.

■ Negative flag (NF)

This flag is set to '1' when the MSB in the result of calculation is '1', and is reset to '0' when the MSB is '0'. This NF flag is used when handling the data as a signed value.

■ Overflow flag (VF)

This flag is set to '1' when there is an overflow in the result of operation when it is handled as a signed value, and reset to '0' when there is no overflow. This flag is used when handling the data as a signed value.

■ Interrupt mask level flags (IM1 ~ IM0)

The interrupt mask level flags (IM1, IM0) are for controlling the acknowledgement of maskable interrupts depending on the priority level of the interrupt cause. The interrupt level is defined from Level '0' ('00') to Level '3' ('11') according to the combination of these flags of which Level '0' is the highest masking level. The interrupt will be acknowledged only when its priority level set in the interrupt level flags (xxxLVn) of the corresponding interrupt control register (xxxICR) is higher than the level set in these two flag bits. When an interrupt is acknowledged, the contents of that interrupt level flags is set in these two flag bits IM1 and IM0, and any interrupt of same or lower level than that interrupt will not be acknowledged until the interrupt servicing is completed.

Table 12-6 Interrupt Mask Level and Interrupt Acknowledgement

Interrupt mask level		Interrupt levels accepted	Priority level
IM1	IM0		
0	0	Only non-maskable interrupts (NMI)	High
0	1	Level 0, NMI	↑
1	0	Levels 0 to 1, NMI	
1	1	Levels 0 to 2, NMI	Low

■ Maskable interrupt enable flag (MIE)

This flag is for enabling maskable interrupts. Maskable interrupts can be acknowledged when this flag is '1'. When MIE is '0', all maskable interrupts will be disabled irrespective of the values of the interrupt mask level flags (IM) of the PSW.
MIE does not get modified by interrupts.

■ Interrupt stack pointer flag (SPF)

This flag is for switching between the interrupt stack pointer and the normal stack pointer. This flag is set to '1' when an interrupt is acknowledged and the interrupt stack pointer is selected. This flag becomes '0' when an RTI instruction is executed thereby selecting the normal stack pointer. However, this flag remains '1' after the execution of intermediate RTI instructions when servicing nested interrupts.

The Instruction Set

MN101D00 SERIES INSTRUCTION SET

Group	Mnemonic	Operation	Flag VF	Flag NF	Flag CF	Flag ZF	Code Size	Cycle	Re-peat	Machine Code Extension	1	2	3	4	5	6	7	8	9	10	11	Notes	Page
Data transfer instructions																							
MOV	MOV Dn,Dm	Dn→Dm	—	—	—	—	2	1			1010	DnDm											25
	MOV imm8,Dm	imm8→Dm	—	—	—	—	4	2			1010	DmDm	<#8.	...>									25
	MOV Dn,PSW	Dn→PSW	●	●	●	●	3	3		0010	1001	01Dn											26
	MOV PSW,Dm	PSW→Dm	—	—	—	—	3	2		0010	0001	01Dm											26
	MOV (An),Dm	mem8(An)→Dm	—	—	—	—	2	2			0100	1ADm											27
	MOV (d8,An),Dm	mem8(d8+An)→Dm	—	—	—	—	4	2			0110	1ADm	<d8.	...>								*1	27
	MOV (d16,An),Dm	mem8(d16+An)→Dm	—	—	—	—	7	4		0010	0110	1ADm	<d16	....	....	...>							28
	MOV (d4,SP),Dm	mem8(d4+SP)→Dm	—	—	—	—	3	2			0110	01Dm	<d4>									*2	28
	MOV (d8,SP),Dm	mem8(d8+SP)→Dm	—	—	—	—	5	3		0010	0110	01Dm	<d8.	...>								*3	29
	MOV (d16,SP),Dm	mem8(d16+SP)→Dm	—	—	—	—	7	4		0010	0110	00Dm	<d16	....	....	...>							29
	MOV (io8),Dm	mem8(IOTOP+io8)→Dm	—	—	—	—	4	2			0110	00Dm	<io8	...>									30
	MOV (abs8),Dm	mem8(abs8)→Dm	—	—	—	—	4	2			0100	01Dm	<abs	8..>									30
	MOV (abs12),Dm	mem8(abs12)→Dm	—	—	—	—	5	2			0100	00Dm	<abs	12..	...>								31
	MOV (abs16),Dm	mem8(abs16)→Dm	—	—	—	—	7	4		0010	1100	00Dm	<abs	16..	....	...>							31
	MOV Dn,(Am)	Dn→mem8(Am)	—	—	—	—	2	2			0101	1aDn											32
	MOV Dn,(d8,Am)	Dn→mem8(d8+Am)	—	—	—	—	4	2			0111	1aDn	<d8.	...>								*1	32
	MOV Dn,(d16,Am)	Dn→mem8(d16+Am)	—	—	—	—	7	4		0010	0111	1aDn	<d16	....	....	...>							33
	MOV Dn,(d4,SP)	Dn→mem8(d4+SP)	—	—	—	—	3	2			0111	01Dn	<d4>									*2	33
	MOV Dn,(d8,SP)	Dn→mem8(d8+SP)	—	—	—	—	5	3		0010	0111	01Dn	<d8.	...>								*3	34
	MOV Dn,(d16,SP)	Dn→mem8(d16+SP)	—	—	—	—	7	4		0010	0111	00Dn	<d16	....	....	...>							34
	MOV Dn,(io8)	Dn→mem8(IOTOP+io8)	—	—	—	—	4	2			0111	00Dn	<io8	...>									35
	MOV Dn,(abs8)	Dn→mem8(abs8)	—	—	—	—	4	2			0101	01Dn	<abs	8..>									35
	MOV Dn,(abs12)	Dn→mem8(abs12)	—	—	—	—	5	2			0101	00Dn	<abs	12..	...>								36
	MOV Dn,(abs16)	Dn→mem8(abs16)	—	—	—	—	7	4		0010	1101	00Dn	<abs	16..	....	...>							36
	MOV imm8,(io8)	imm8→mem8(IOTOP+io8)	—	—	—	—	6	3			0000	0010	<io8	...>	<#8.	...>							37
	MOV imm8,(abs8)	imm8→mem8(abs8)	—	—	—	—	6	3			0001	0100	<abs	8..>	<#8.	...>							37
	MOV imm8,(abs12)	imm8→mem8(abs12)	—	—	—	—	7	3			0001	0101	<abs	12..	...>	<#8.	...>						38
	MOV imm8,(abs16)	imm8→mem8(abs16)	—	—	—	—	9	5		0011	1101	1001	<abs	16..	....	...>	<#8.	...>					38
	MOV Dn,(HA)	Dn→mem8(HA)	—	—	—	—	2	2			1101	00Dn											39
MOVW	MOVW (An),DWm	mem16(An)→DWm	—	—	—	—	2	3			1110	00Ad											40
	MOVW (An),Am	mem16(An)→Am	—	—	—	—	3	4		0010	1110	10Aa										*4	40
	MOVW (d4,SP),DWm	mem16(d4+SP)→DWm	—	—	—	—	3	3			1110	011d	<d4>									*2	41
	MOVW (d4,SP),Am	mem16(d4+SP)→Am	—	—	—	—	3	3			1110	010a	<d4>									*2	41
	MOVW (d8,SP),DWm	mem16(d8+SP)→DWm	—	—	—	—	5	4		0010	1110	011d	<d8.	...>								*3	42
	MOVW (d8,SP),Am	mem16(d8+SP)→Am	—	—	—	—	5	4		0010	1110	010a	<d8.	...>								*3	42
	MOVW (d16,SP),DWm	mem16(d16+SP)→DWm	—	—	—	—	7	5		0010	1110	001d	<d16	....	....	...>							43
	MOVW (d16,SP),Am	mem16(d16+SP)→Am	—	—	—	—	7	5		0010	1110	000a	<d16	....	....	...>							43
	MOVW (abs8),DWm	mem16(abs8)→DWm	—	—	—	—	4	3			1100	011d	<abs	8..>									44
	MOVW (abs8),Am	mem16(abs8)→Am	—	—	—	—	4	3			1100	010a	<abs	8..>									44
	MOVW (abs16),DWm	mem16(abs16)→DWm	—	—	—	—	7	5		0010	1100	011d	<abs	16..	....	...>							45
	MOVW (abs16),Am	mem16(abs16)→Am	—	—	—	—	7	5		0010	1100	010a	<abs	16..	....	...>							45
	MOVW DWn,(Am)	DWn→mem16(Am)	—	—	—	—	2	3			1111	00aD											46
	MOVW An,(Am)	An→mem16(Am)	—	—	—	—	3	4		0010	1111	10aA										*4	46
	MOVW DWn,(d4,SP)	DWn→mem16(d4+SP)	—	—	—	—	3	3			1111	011D	<d4>									*2	47
	MOVW An,(d4,SP)	An→mem16(d4+SP)	—	—	—	—	3	3			1111	010A	<d4>									*2	47
	MOVW DWn,(d8,SP)	DWn→mem16(d8+SP)	—	—	—	—	5	4		0010	1111	011D	<d8.	...>								*3	48
	MOVW An,(d8,SP)	An→mem16(d8+SP)	—	—	—	—	5	4		0010	1111	010A	<d8.	...>								*3	48
	MOVW DWn,(d16,SP)	DWn→mem16(d16+SP)	—	—	—	—	7	5		0010	1111	001D	<d16	....	....	...>							49
	MOVW An,(d16,SP)	An→mem16(d16+SP)	—	—	—	—	7	5		0010	1111	000A	<d16	....	....	...>							49
	MOVW DWn,(abs8)	DWn→mem16(abs8)	—	—	—	—	4	3			1101	011D	<abs	8..>									50
	MOVW An,(abs8)	An→mem16(abs8)	—	—	—	—	4	3			1101	010A	<abs	8..>									50
	MOVW DWn,(abs16)	DWn→mem16(abs16)	—	—	—	—	7	5		0010	1101	011D	<abs	16..	....	...>							51
	MOVW An,(abs16)	An→mem16(abs16)	—	—	—	—	7	5		0010	1101	010A	<abs	16..	....	...>							51
	MOVW DWn,(HA)	DWn→mem16(HA)	—	—	—	—	2	3			1001	010D											52
	MOVW An,(HA)	An→mem16(HA)	—	—	—	—	2	3			1001	011A											52
	MOVW imm8,DWm	sign(imm8)→DWm	—	—	—	—	4	2			0000	110d	<#8.	...>								*5	53
	MOVW imm8,Am	zero(imm8)→Am	—	—	—	—	4	2			0000	111a	<#8.	...>								*6	53
	MOVW imm16,DWm	imm16→DWm	—	—	—	—	6	3			1100	111d	<#16	....	....	...>							54

Note: "Page" refers to the corresponding page in the Instruction Manual.

*Note 1: d8 Sign extension *Note 4: A=An, a=Am
*Note 2: d4 Zero extension *Note 5: #8 Sign extension
*Note 3: d8 Zero extension *Note 6: #8 Zero extension

MN101D00 SERIES INSTRUCTION SET

Group	Mnemonic	Operation	Flag VF	Flag NF	Flag CF	Flag ZF	Code Size	Cycle	Re-peat	Machine Code: Extension	1	2	3	4	5	6	7	8	9	10	11	Notes	Page
	MOVW imm16,Am	imm16→Am	—	—	—	—	6	3			1101	111a	<#16	....	....	...>							54
	MOVW SP,Am	SP→Am	—	—	—	—	3	3			0010	0000	100a										55
	MOVW An,SP	An→SP	—	—	—	—	3	3			0010	0000	101A										55
	MOVW DWn,DWm	DWn→DWm	—	—	—	—	3	3			0010	1000	00Dd									*1	56
	MOVW DWn,Am	DWn→Am	—	—	—	—	3	3			0010	0100	11Da										56
	MOVW An,DWm	An→DWm	—	—	—	—	3	3			0010	1100	11Ad										57
	MOVW An,Am	An→Am	—	—	—	—	3	3			0010	0000	00Aa									*2	57
PUSH	PUSH Dn	SP-1→SP,Dn→mem8(SP)	—	—	—	—	2	3			1111	10Dn											58
	PUSH An	SP-2→SP,An→mem16(SP)	—	—	—	—	2	5			0001	011A											58
POP	POP Dn	mem8(SP)→Dn,SP+1→SP	—	—	—	—	2	3			1110	10Dn											59
	POP An	mem16(SP)→An,SP+2→SP	—	—	—	—	2	4			0000	011A											59
EXT	EXT Dn,DWm	sign(Dn)→DWm	—	—	—	—	3	3			0010	1001	000d									*3	60
Arithmetic operation instructions																							
ADD	ADD Dn,Dm	Dm+Dn→Dm	●	●	●	●	3	2	❍		0011	0011	DnDm										61
	ADD imm4,Dm	Dm+sign(imm4)→Dm	●	●	●	●	3	2			1000	00Dm	<#4>									*6	61
	ADD imm8,Dm	Dm+imm8→Dm	●	●	●	●	4	2			0000	10Dm	<#8.	...>									62
ADDC	ADDC Dn,Dm	Dm+Dn+CF→Dm	●	●	●	●	3	2	❍		0011	1011	DnDm										63
ADDW	ADDW DWn,DWm	DWm+DWn→DWm	●	●	●	●	3	3	❍		0010	0101	00Dd									*1	64
	ADDW DWn,Am	Am+DWn→Am	●	●	●	●	3	3	❍		0010	0101	10Da										64
	ADDW imm4,Am	Am+sign(imm4)→Am	●	●	●	●	3	2			1110	110a	<#4>									*6	65
	ADDW imm8,Am	Am+sign(imm8)→Am	●	●	●	●	5	3			0010	1110	110a	<#8.	...>							*7	65
	ADDW imm16,Am	Am+imm16→Am	●	●	●	●	7	4			0010	0101	011a	<#16	....	....	...>						66
	ADDW imm4,SP	SP+sign(imm4)→SP	—	—	—	—	3	2			1111	1101	<#4>									*6	66
	ADDW imm8,SP	SP+sign(imm8)→SP	—	—	—	—	4	2			1111	1100	<#8.	...>								*7	67
	ADDW imm16,SP	SP+imm16→SP	—	—	—	—	7	4			0010	1111	1100	<#16	....	....	...>						67
	ADDW imm16,DWm	DWm+imm16→DWm	●	●	●	●	7	4			0010	0101	010d	<#16	....	....	...>						68
ADDUW	ADDUW Dn,Am	Am+zero(Dn)→Am	●	●	●	●	3	3	❍		0010	1000	1aDn									*8	69
ADDSW	ADDSW Dn,Am	Am+sign(Dn)→Am	●	●	●	●	3	3	❍		0010	1001	1aDn										70
SUB	SUB Dn,Dm (Dn≠Dm)	Dm-Dn→Dm	●	●	●	●	3	2	❍		0010	1010	DnDm										71
	SUB Dn,Dn	Dn-Dn→Dn	0	0	0	1	2	1			1000	01Dn											71
	SUB imm8,Dm	Dm-imm8→Dm	●	●	●	●	5	3			0010	1010	DmDm	<#8.	...>								72
SUBC	SUBC Dn,Dm	Dm-Dn-CF→Dm	●	●	●	●	3	2	❍		0010	1011	DnDm										73
SUBW	SUBW DWn,DWm	DWm-DWn→DWm	●	●	●	●	3	3			0010	0100	00Dd									*1	74
	SUBW DWn,Am	Am-DWn→Am	●	●	●	●	3	3			0010	0100	10Da										74
	SUBW imm16,DWm	DWm-imm16→DWm	●	●	●	●	7	4			0010	0100	010d	<#16	....	....	...>						75
	SUBW imm16,Am	Am-imm16→Am	●	●	●	●	7	4			0010	0100	011a	<#16	....	....	...>						75
MULU	MULU Dn,Dm	Dm*Dn→DWk	0	●	●	●	3	8			0010	1111	111D									*4	76
MULWU	MULWU	DW0*DW1→(DW1,DW0)	0	—	—	—	6	15			0000	0010	0111	0000	0001	0000							77
MULW	MULW	DW0*DW1→(DW1,DW0)	0	—	—	—	6	15			0000	0010	0111	0000	0010	0000							78
DIVU	DIVU Dn,DWm	DWm/Dn→DWm-l•••DWm-h	●	●	●	●	3	9			0010	1110	111d									*5	79
DIVWU	DIVWU	(DW1,DW0)/A0→DW0•••DW1	●	—	—	—	6	17			0000	0010	0111	0000	0100	0000							80
CMP	CMP Dn,Dm	Dm-Dn...PSW	●	●	●	●	3	2			0011	0010	DnDm										81
	CMP imm8,Dm	Dm-imm8...PSW	●	●	●	●	4	2			1100	00Dm	<#8.	...>									81
	CMP imm8,(abs8)	mem8(abs8)-imm8...PSW	●	●	●	●	6	3			0000	0100	<abs	8..>	<#8.	...>							82
	CMP imm8,(abs12)	mem8(abs12)-imm8...PSW	●	●	●	●	7	3			0000	0101	<abs	12..	...>	<#8.	...>						82
	CMP imm8,(abs16)	mem8(abs16)-imm8...PSW	●	●	●	●	9	5			0011	1101	1000	<abs	16..	....	...>	<#8.	...>				83
CMPW	CMPW DWn,DWm	DWm-DWn...PSW	●	●	●	●	3	3			0010	1000	01Dd									*1	84
	CMPW DWn,Am	Am-DWn...PSW	●	●	●	●	3	3			0010	0101	11Da										84
	CMPW An,Am	Am-An...PSW	●	●	●	●	3	3			0010	0000	01Aa									*2	85
	CMPW imm16,DWm	DWm-imm16...PSW	●	●	●	●	6	3			1100	110d	<#16	....	....	...>							85
	CMPW imm16,Am	Am-imm16...PSW	●	●	●	●	6	3			1101	110a	<#16	....	....	...>							86
Logical operation instructions																							
AND	AND Dn,Dm	Dm&Dn→Dm	0	●	0	●	3	2			0011	0111	DnDm										87
	AND imm8,Dm	Dm&imm8→Dm	0	●	0	●	4	2			0001	11Dm	<#8.	...>									87
	AND imm8,PSW	PSW&imm8→PSW	●	●	●	●	5	3			0010	1001	0010	<#8.	...>								88
OR	OR Dn,Dm	DmIDn→Dm	0	●	0	●	3	2			0011	0110	DnDm										89
	OR imm8,Dm	DmIimm8→Dm	0	●	0	●	4	2			0001	10Dm	<#8.	...>									89
	OR imm8,PSW	PSWIimm8→PSW	●	●	●	●	5	3			0010	1001	0011	<#8.	...>								90
XOR	XOR Dn,Dm	Dm^Dn→Dm	0	●	0	●	3	2			0011	1010	DnDm									*9	91
	XOR imm8,Dm	Dm^imm8→Dm	0	●	0	●	5	3			0011	1010	DmDm	<#8.	...>								91

*Note 1: D=DWn, d=DWm
*Note 2: A=An, a=Am
*Note 3: d=DWm
*Note 4: D=DWk
*Note 5: D=DWm
*Note 6: #4 Sign extension
*Note 7: #8 Sign extension
*Note 8: Dn Zero extension
*Note 9: m≠n

MN101D00 SERIES INSTRUCTION SET

Group	Mnemonic	Operation	Flag VF	Flag NF	Flag CF	Flag ZF	Code Size	Cycle	Re-peat	Exten-sion	Machine Code 1	2	3	4	5	6	7	8	9	10	11	Notes	Page
NOT	NOT Dn	¯Dn→Dn	0	●	0	●	3	2			0010	0010	10Dn										92
ASR	ASR Dn	Dn.msb→temp,Dn.lsb→CF Dn>>1→Dn,temp→Dn.msb	0	—	●	●	3	2	❍		0010	0011	10Dn										93
LSR	LSR Dn	Dn.lsb→CF,Dn>>1→Dn 0→Dn.msb	0	0	●	●	3	2	❍		0010	0011	11Dn										94
ROR	ROR Dn	Dn.lsb→temp,Dn>>1→Dn CF→Dn.msb,temp→CF	0	●	●	●	3	2	❍		0010	0010	11Dn										95
Bit manipulation instructions																							
BSET	BSET (io8)bp	mem8(IOTOP+io8)&bpdata...PSW 1→mem8(IOTOP+io8)bp	0	●	0	●	5	5			0011	1000	0bp.	<io8	...>								96
	BSET (abs8)bp	mem8(abs8)&bpdata...PSW 1→mem8(abs8)bp	0	●	0	●	4	4				1011	0bp.	<abs	8..>								96
	BSET (abs16)bp	mem8(abs16)&bpdata...PSW 1→mem8(abs16)bp	0	●	0	●	7	6			0011	1100	0bp.	<abs	16..	....	...>						97
BCLR	BCLR (io8)bp	mem8(IOTOP+io8)&bpdata...PSW 0→mem8(IOTOP+io8)bp	0	●	0	●	5	5			0011	1000	1bp.	<io8	...>								98
	BCLR (abs8)bp	mem8(abs8)&bpdata...PSW 0→mem8(abs8)bp	0	●	0	●	4	4				1011	1bp.	<abs	8..>								98
	BCLR (abs16)bp	mem8(abs16)&bpdata...PSW 0→mem8(abs16)bp	0	●	0	●	7	6			0011	1100	1bp.	<abs	16..	....	...>						99
BTST	BTST imm8,Dm	Dm&imm8...PSW	0	●	0	●	5	3			0010	0000	11Dm	<#8.	...>								100
	BTST (abs16)bp	mem8(abs16)&bpdata...PSW	0	●	0	●	7	5			0011	1101	0bp.	<abs	16..	....	...>						100
Branch instructions																							
Bcc	BEQ label	if(ZF=1), PC+3+d4(label)+H→PC if(ZF=0), PC+3→PC	—	—	—	—	3	2/3				1001	000H	<d4>								*1	101
	BEQ label	if(ZF=1), PC+4+d7(label)+H→PC if(ZF=0), PC+4→PC	—	—	—	—	4	2/3				1000	1010	<d7.	...H							*2	101
	BEQ label	if(ZF=1), PC+5+d11(label)+H→PC if(ZF=0), PC+5→PC	—	—	—	—	5	2/3				1001	1010	<d11	....	...H						*3	102
	BNE label	if(ZF=0), PC+3+d4(label)+H→PC if(ZF=1), PC+3→PC	—	—	—	—	3	2/3				1001	001H	<d4>								*1	103
	BNE label	if(ZF=0), PC+4+d7(label)+H→PC if(ZF=1), PC+4→PC	—	—	—	—	4	2/3				1000	1011	<d7.	...H							*2	103
	BNE label	if(ZF=0), PC+5+d11(label)+H→PC if(ZF=1), PC+5→PC	—	—	—	—	5	2/3				1001	1011	<d11	....	...H						*3	104
	BGE label	if((VF^NF)=0),PC+4+d7(label)+H→PC if((VF^NF)=1),PC+4→PC	—	—	—	—	4	2/3				1000	1000	<d7.	...H							*2	105
	BGE label	if((VF^NF)=0),PC+5+d11(label)+H→PC if((VF^NF)=1),PC+5→PC	—	—	—	—	5	2/3				1001	1000	<d11	....	...H						*3	105
	BCC label	if(CF=0),PC+4+d7(label)+H→PC if(CF=1), PC+4→PC	—	—	—	—	4	2/3				1000	1100	<d7.	...H							*2	106
	BCC label	if(CF=0), PC+5+d11(label)+H→PC if(CF=1), PC+5→PC	—	—	—	—	5	2/3				1001	1100	<d11	....	...H						*3	106
	BCS label	if(CF=1),PC+4+d7(label)+H→PC if(CF=0), PC+4→PC	—	—	—	—	4	2/3				1000	1101	<d7.	...H							*2	107
	BCS label	if(CF=1), PC+5+d11(label)+H→PC if(CF=0), PC+5→PC	—	—	—	—	5	2/3				1001	1101	<d11	....	...H						*3	107
	BLT label	if((VF^NF)=1),PC+4+d7(label)+H→PC if((VF^NF)=0),PC+4→PC	—	—	—	—	4	2/3				1000	1110	<d7.	...H							*2	108
	BLT label	if((VF^NF)=1),PC+5+d11(label)+H→PC if((VF^NF)=0),PC+5→PC	—	—	—	—	5	2/3				1001	1110	<d11	....	...H						*3	108
	BLE label	if((VF^NF)|ZF=1),PC+4+d7(label)+H→PC if((VF^NF)|ZF=0),PC+4→PC	—	—	—	—	4	2/3				1000	1111	<d7.	...H							*2	109
	BLE label	if((VF^NF)|ZF=1),PC+5+d11(label)+H→PC if((VF^NF)|ZF=0),PC+5→PC	—	—	—	—	5	2/3				1001	1111	<d11	....	...H						*3	109
	BGT label	if((VF^NF)|ZF=0),PC+5+d7(label)+H→PC if((VF^NF)|ZF=1),PC+5→PC	—	—	—	—	5	3/4			0010	0010	0001	<d7.	...H							*2	110

*Note 1: d4 Sign extension
*Note 2: d7 Sign extension
*Note 3: d11 Sign extension

MN101D00 SERIES INSTRUCTION SET

Group	Mnemonic	Operation	Flag VF	Flag NF	Flag CF	Flag ZF	Code Size	Cycle	Re-peat	Machine Code Exten-sion	1	2	3	4	5	6	7	8	9	10	11	Notes	Page
Bcc	BGT label	if((VF^NF)|ZF=0),PC+6+d11(label)+H→PC if((VF^NF)|ZF=1),PC+6→PC	—	—	—	—	6	3/4		0010	0011	0001	<d11	....	...H							*3	110
	BHI label	if(CFIZF=0),PC+5+d7(label)+H→PC if(CFIZF=1), PC+5→PC	—	—	—	—	5	3/4		0010	0010	0010	<d7.	...H								*2	111
	BHI label	if(CFIZF=0),PC+6+d11(label)+H→PC if(CFIZF=1), PC+6→PC	—	—	—	—	6	3/4		0010	0011	0010	<d11	....	...H							*3	111
	BLS label	if(CFIZF=1),PC+5+d7(label)+H→PC if(CFIZF=0), PC+5→PC	—	—	—	—	5	3/4		0010	0010	0011	<d7.	...H								*2	112
	BLS label	if(CFIZF=1),PC+6+d11(label)+H→PC if(CFIZF=0), PC+6→PC	—	—	—	—	6	3/4		0010	0011	0011	<d11	....	...H							*3	112
	BNC label	if(NF=0),PC+5+d7(label)+H→PC if(NF=1),PC+5→PC	—	—	—	—	5	3/4		0010	0010	0100	<d7.	...H								*2	113
	BNC label	if(NF=0),PC+6+d11(label)+H→PC if(NF=1),PC+6→PC	—	—	—	—	6	3/4		0010	0011	0100	<d11	....	...H							*3	113
	BNS label	if(NF=1),PC+5+d7(label)+H→PC if(NF=0),PC+5→PC	—	—	—	—	5	3/4		0010	0010	0101	<d7.	...H								*2	114
	BNS label	if(NF=1),PC+6+d11(label)+H→PC if(NF=0),PC+6→PC	—	—	—	—	6	3/4		0010	0011	0101	<d11	....	...H							*3	114
	BVC label	if(VF=0),PC+5+d7(label)+H→PC if(VF=1),PC+5→PC	—	—	—	—	5	3/4		0010	0010	0110	<d7.	...H								*2	115
	BVC label	if(VF=0),PC+6+d11(label)+H→PC if(VF=1),PC+6→PC	—	—	—	—	6	3/4		0010	0011	0110	<d11	....	...H							*3	115
	BVS label	if(VF=1),PC+5+d7(label)+H→PC if(VF=0),PC+5→PC	—	—	—	—	5	3/4		0010	0010	0111	<d7.	...H								*2	116
	BVS label	if(VF=1),PC+6+d11(label)+H→PC if(VF=0),PC+6→PC	—	—	—	—	6	3/4		0010	0011	0111	<d11	....	...H							*3	116
	BRA label	PC+3+d4(label)+H→PC	—	—	—	—	3	3			1110	111H	<d4>									*1	117
	BRA label	PC+4+d7(label)+H→PC	—	—	—	—	4	3			1000	1001	<d7.	...H								*2	117
	BRA label	PC+5+d11(label)+H→PC	—	—	—	—	5	3			1001	1001	<d11	....	...H							*3	118
CBEQ	CBEQ imm8,Dm,label	if(Dm=imm8),PC+6+d7(label)+H→PC if(Dm≠imm8),PC+6→PC	●	●	●	●	6	3/4			1100	10Dm	<#8.	...>	<d7.	...H						*2	119
	CBEQ imm8,Dm,label	if(Dm=imm8),PC+8+d11(label)+H→PC if(Dm≠imm8),PC+8→PC	●	●	●	●	8	4/5		0010	1100	10Dm	<#8.	...>	<d11	....	...H					*3	119
	CBEQ imm8,(abs8),label	if(mem8(abs8)=imm8),PC+9+d7(label)+H→PC if(mem8(abs8)≠imm8),PC+9→PC	●	●	●	●	9	6/7		0010	1101	1100	<abs	8..>	<#8.	...>	<d7.	...H				*2	120
	CBEQ imm8,(abs8),label	if(mem8(abs8)=imm8),PC+10+d11(label)+H→PC if(mem8(abs8)≠imm8),PC+10→PC	●	●	●	●	10	6/7		0010	1101	1101	<abs	8..>	<#8.	...>	<d11	....	...H			*3	120
	CBEQ imm8,(abs16),label	if(mem8(abs16)=imm8),PC+11+d7(label)+H→PC if(mem8(abs16)≠imm8),PC+11→PC	●	●	●	●	11	7/8		0011	1101	1100	<abs	16..	....	...>	<#8.	...>	<d7.	...H		*2	121
	CBEQ imm8,(abs16),label	if(mem8(abs16)=imm8),PC+12+d11(label)+H→PC if(mem8(abs16)≠imm8),PC+12→PC	●	●	●	●	12	7/8		0011	1101	1101	<abs	16..	....	...>	<#8.	...>	<d11	....	...H	*3	121
CBNE	CBNE imm8,Dm,label	if(Dm≠imm8),PC+6+d7(label)+H→PC if(Dm=imm8),PC+6→PC	●	●	●	●	6	3/4			1101	10Dm	<#8.	...>	<d7.	..H>						*2	122
	CBNE imm8,Dm,label	if(Dm≠imm8),PC+8+d11(label)+H→PC if(Dm=imm8),PC+8→PC	●	●	●	●	8	4/5		0010	1101	10Dm	<#8.	...>	<d11	....	...H					*3	122
	CBNE imm8,(abs8),label	if(mem8(abs8)≠imm8),PC+9+d7(label)+H→PC if(mem8(abs8)=imm8),PC+9→PC	●	●	●	●	9	6/7		0010	1101	1110	<abs	8..>	<#8.	...>	<d7.	...H				*2	123
	CBNE imm8,(abs8),label	if(mem8(abs8)≠imm8),PC+10+d11(label)+H→PC if(mem8(abs8)=imm8),PC+10→PC	●	●	●	●	10	6/7		0010	1101	1111	<abs	8..>	<#8.	...>	<d11	....	...H			*3	123
	CBNE imm8,(abs16),label	if(mem8(abs16)≠imm8),PC+11+d7(label)+H→PC if(mem8(abs16)=imm8),PC+11→PC	●	●	●	●	11	7/8		0011	1101	1110	<abs	16..	....	...>	<#8.	...>	<d7.	...H		*2	124
	CBNE imm8,(abs16),label	if(mem8(abs16)≠imm8),PC+12+d11(label)+H→PC if(mem8(abs16)=imm8),PC+12→PC	●	●	●	●	12	7/8		0011	1101	1111	<abs	16..	....	...>	<#8.	...>	<d11	....	...H	*3	124
TBZ	TBZ (abs8)bp,label	if(mem8(abs8)bp=0),PC+7+d7(label)+H→PC if(mem8(abs8)bp=1),PC+7→PC	0	●	0	●	7	6/7		0011	0000	0bp.	<abs	8..>	<d7.	...H						*2	125
	TBZ (abs8)bp,label	if(mem8(abs8)bp=0),PC+8+d11(label)+H→PC if(mem8(abs8)bp=1),PC+8→PC	0	●	0	●	8	6/7		0011	0000	1bp.	<abs	8..>	<d11	....	...H					*3	125

*Note 1: d4 Sign extension
*Note 2: d7 Sign extension
*Note 3: d11 Sign extension

MN101D00 SERIES INSTRUCTION SET

Group	Mnemonic	Operation	Flag VF	Flag NF	Flag CF	Flag ZF	Code Size	Cycle	Re-peat	Extension	Machine Code 1	2	3	4	5	6	7	8	9	10	11	Notes	Page
TBZ	TBZ (io8)bp,label	if(mem8(IOTOP+io8)bp=0),PC+7+d7(label)+H→PC if(mem8(IOTOP+io8)bp=1),PC+7→PC	0	●	0	●	7	6/7			0011	0100	0bp.	<io8	...>	<d7.	...H					*1	126
	TBZ (io8)bp,label	if(mem8(IOTOP+io8)bp=0),PC+8+d11(label)+H→PC if(mem8(IOTOP+io8)bp=1),PC+8→PC	0	●	0	●	8	6/7			0011	0100	1bp.	<io8	...>	<d11	....	...H				*2	126
	TBZ (abs16)bp,label	if(mem8(abs16)bp=0),PC+9+d7(label)+H→PC if(mem8(abs16)bp=1),PC+9→PC	0	●	0	●	9	7/8			0011	1110	0bp.	<abs	16..	....	...>	<d7.	...H			*1	127
	TBZ (abs16)bp,label	if(mem8(abs16)bp=0),PC+10+d11(label)+H→PC if(mem8(abs16)bp=1),PC+10→PC	0	●	0	●	10	7/8			0011	1110	1bp.	<abs	16..	....	...>	<d11	....	...H		*2	127
TBNZ	TBNZ (abs8)bp,label	if(mem8(abs8)bp=1),PC+7+d7(label)+H→PC if(mem8(abs8)bp=0),PC+7→PC	0	●	0	●	7	6/7			0011	0001	0bp.	<abs	8..>	<d7.	...H					*1	128
	TBNZ (abs8)bp,label	if(mem8(abs8)bp=1),PC+8+d11(label)+H→PC if(mem8(abs8)bp=0),PC+8→PC	0	●	0	●	8	6/7			0011	0001	1bp.	<abs	8..>	<d11	....	...H				*2	128
	TBNZ (io8)bp,label	if(mem8(io)bp=1),PC+7+d7(label)+H→PC if(mem8(io)bp=0),PC+7→PC	0	●	0	●	7	6/7			0011	0101	0bp.	<io8	...>	<d7.	...H					*1	129
	TBNZ (io8)bp,label	if(mem8(io)bp=1),PC+8+d11(label)+H→PC if(mem8(io)bp=0),PC+8→PC	0	●	0	●	8	6/7			0011	0101	1bp.	<io8	...>	<d11	....	...H				*2	129
	TBNZ (abs16)bp,label	if(mem8(abs16)bp=1),PC+9+d7(label)+H→PC if(mem8(abs16)bp=0),PC+9→PC	0	●	0	●	9	7/8			0011	1111	0bp.	<abs	16..	....	...>	<d7.	...H			*1	130
	TBNZ (abs16)bp,label	if(mem8(abs16)bp=1),PC+10+d11(label)+H→PC if(mem8(abs16)bp=0),PC+10→PC	0	●	0	●	10	7/8			0011	1111	1bp.	<abs	16..	....	...>	<d11	....	...H		*2	130
JMP	JMP (An)	0→PC.17~16,An→PC.15~0,0→PC.H	—	—	—	—	3	4			0010	0001	00A0										131
	JMP label	abs18(label)+H→PC	—	—	—	—	7	5			0011	1001	0aaH	<abs	18.b	p15~	0..>					*5	131
JSR	JSR (An)	SP-3→SP,(PC+3).bp7~0→mem8(SP) (PC+3).bp15~8→mem8(SP+1) (PC+3).H→mem8(SP+2).bp7, 0→mem8(SP+2).bp6~2, (PC+3).bp17~16→mem8(SP+2).bp1~0 0→PC.bp17~16 An→PC.bp15~0,0→PC.H	—	—	—	—	3	7			0010	0001	00A1										132
	JSR label	SP-3→SP,(PC+5).bp7~0→mem8(SP) (PC+5).bp15~8→mem8(SP+1) (PC+5).H→mem8(SP+2).bp7, 0→mem8(SP+2).bp6~2, (PC+5).bp17~16→mem8(SP+2).bp1~0 PC+5+d12(label)+H→PC	—	—	—	—	5	6			0001	000H	<d12	....	...>							*3	132
	JSR label	SP-3→SP,(PC+6).bp7~0→mem8(SP) (PC+6).bp15~8→mem8(SP+1) (PC+6).H→mem8(SP+2).bp7, 0→mem8(SP+2).bp6~2, (PC+6).bp17~16→mem8(SP+2).bp1~0 PC+6+d16(label)+H→PC	—	—	—	—	6	7			0001	001H	<d16	....	....	...>						*4	133
	JSR label	SP-3→SP,(PC+7).bp7~0→mem8(SP) (PC+7).bp15~8→mem8(SP+1) (PC+7).H→mem8(SP+2).bp7, 0→mem8(SP+2).bp6~2, (PC+7).bp17~16→mem8(SP+2).bp1~0 abs18(label)+H→PC	—	—	—	—	7	8			0011	1001	1aaH	<abs	18.b	p15~	0..>					*5	133
	JSRV (tbl4)	SP-3→SP,(PC+3).bp70→mem8(SP) (PC+3).bp15~8→mem8(SP+1) (PC+3).H→mem8(SP+2).bp7 0→mem8(SP+2).bp6~2, (PC+3).bp17~16→mem8(SP+2).bp1~0 mem8(x'004080+tbl4<<2)→PC.bp7~0 mem8(x'004080+tbl4<<2+1)→PC.bp15~8 mem8(x'004080+tbl4<<2+2).bp7→PC.H mem8(x'004080+tbl4<<2+2).bp1~0→ PC.bp17~16	—	—	—	—	3	9			1111	1110	<t4>										134
NOP	NOP	PC+2→PC	—	—	—	—	2	1	❍		0000	0000											135

*Note 1: d7 Sign extension
*Note 2: d11 Sign extension
*Note 3: d12 Sign extension
*Note 4: d16 Sign extension
*Note 5: aa=abs18.17 to 16

MN101D00 SERIES INSTRUCTION SET

Group	Mnemonic	Operation	Flag VF	Flag NF	Flag CF	Flag ZF	Code Size	Cycle	Re-peat	Machine Code Exten-sion	1	2	3	4	5	6	7	8	9	10	11	Notes	Page
RTS	RTS	mem8(SP)→(PC).bp7~0 mem8(SP+1)→(PC).bp15~8 mem8(SP+2).bp7→(PC).H mem8(SP+2).bp1~0→(PC).bp17~16 SP+3→SP	—	—	—	—	2	7		0000 0001													136
RTI	RTI	mem8(SP)→PSW mem8(SP+1)→(PC).bp7~0 mem8(SP+2)→(PC).bp15~8 mem8(SP+3).bp7→(PC).H mem8(SP+3).bp1~0→(PC).bp17~16 mem8(SP+4)→HA-l mem8(SP+5)→HA-h SP+6→SP	●	●	●	●	2	11		0000 0011													137
Control instruction																							
REP	REP imm3	imm3-1→RPC	—	—	—	—	3	2		0010 0001 1rep												*1	138

*Note 1: The number of repetitions is 0 when imm3-1=0, (rep: imm3-1)

Apart from the instructions of the MN101D00 series, the following instructions are also incorporated in the assembler as macro instructions.
These macro instructions are replaced by equivalent instructions by the assembler before execution.

Macro instruction	Replacing instruction	Remarks
INC Dn	ADD 1,Dn	
DEC Dn	ADD -1,Dn	
INC An	ADDW 1,An	
DEC An	ADDW -1,An	
INC2 An	ADDW 2,An	
DEC2 An	ADDW -2,An	
CLR Dn	SUB Dn,Dm	n=m
ASL Dn	ADD Dn,Dm	n=m
ROL Dn	ADDC Dn,Dm	n=m
NEG Dn	NOT Dn ADD 1,Dn	
NOPL	MOVW DWn,DWm	n=m
MOV (SP),Dn	MOV (0,SP),Dn	
MOV Dn,(SP)	MOV Dn,(0,SP)	
MOVW (SP),DWn	MOVW (0,SP),DWn	
MOVW DWn,(SP)	MOVW DWn,(0,SP)	
MOVW (SP),An	MOVW (0,SP),An	
MOVW An,(SP)	MOVW An,(0,SP)	

Ver3.0 (1998.4.3)

Instruction Map

MN101D00 SERIES INSTRUCTION MAP

First nibble / second nibble

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	RTS	MOV #8,(io8) *1	RTI	CMP #8,(abs8)/(abs12)		POP An		ADD #8,Dm				MOVW #8,DWm		MOVW #8,Am	
1	JSR d12(label)		JSR d16(label)		MOV #8,(abs8)/(abs12)		PUSH An		OR #8,Dm				AND #8,Dm			
2	When the extension code is b'0010'.															
3	When the extension code is b'0011'.															
4	MOV (abs12),Dm				MOV (abs8),Dm				MOV (An),Dm							
5	MOV Dn,(abs12)				MOV Dn,(abs8)				MOV Dn,(Am)							
6	MOV (io8),Dm				MOV (d4,SP),Dm				MOV (d8,An),Dm							
7	MOV Dn,(io8)				MOV Dn,(d4,SP)				MOV Dn,(d8,Am)							
8	ADD #4,Dm				SUB Dn,Dn				BGE d7	BRA d7	BEQ d7	BNE d7	BCC d7	BCS d7	BLT d7	BLE d7
9	BEQ d4		BNE d4		MOVW DWn,(HA)		MOVW An,(HA)		BGE d11	BRA d11	BEQ d11	BNE d11	BCC d11	BCS d11	BLT d11	BLE d11
A	MOV Dn,Dm / MOV #8,Dm															
B	BSET (abs8)bp								BCLR (abs8)bp							
C	CMP #8,Dm				MOVW (abs8),Am		MOVW (abs8),DWm		CBEQ #8,Dm,d7				CMPW #16,DWm		MOVW #16,DWm	
D	MOV Dn,(HA)				MOVW An,(abs8)		MOVW DWn,(abs8)		CBNE #8,Dm,d7				CMPW #16,Am		MOVW #16,Am	
E	MOVW (An),DWm				MOVW (d4,SP),Am		MOVW (d4,SP),DWm		POP Dn				ADDW #4,Am		BRA d4	
F	MOVW DWn,(Am)				MOVW An,(d4,SP)		MOVW DWn,(d4,SP)		PUSH Dn				ADDW #8,SP	ADDW #4,SP	JSRV (tbl4)	

*Note 1: MULWU, MULW, and DIVWU are macro instructions using MOV #8,(i08).

Extension code: b'0010'

Second nibble / third nibble

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	MOVW An,Am				CMPW An,Am				MOVW SP,Am		MOVW An,SP		BTST #8,Dm			
1	JMP (A0)	JSR (A0)	JMP (A1)	JSR (A1)	MOV PSW,Dm				REP #3							
2		BGT d7	BHI d7	BLS d7	BNC d7	BNS d7	BVC d7	BVS d7	NOT Dn				ROR Dn			
3		BGT d11	BHI d11	BLS d11	BNC d11	BNS d11	BVC d11	BVS d11	ASR Dn				LSR Dn			
4	SUBW DWn,DWm				SUBW #16,DWm		SUBW #16,Am		SUBW DWn,Am				MOVW DWn,Am			
5	ADDW DWn,DWm				ADDW #16,DWm		ADDW #16,Am		ADDW DWn,Am				CMPW DWn,Am			
6	MOV (d16,SP),Dm				MOV (d8,SP),Dm				MOV (d16,An),Dm							
7	MOV Dn,(d16,SP)				MOV Dn,(d8,SP)				MOV Dn,(d16,Am)							
8	MOVW DWn,DWm (NOPL @n=m)				CMPW DWn,DWm				ADDUW Dn,Am							
9	EXT Dn,DWm		AND #8,PSW	OR #8,PSW	MOV Dn,PSW				ADDSW Dn,Am							
A	SUB Dn,Dm / SUB #8,Dm															
B	SUBC Dn,Dm															
C	MOV (abs16),Dm				MOVW (abs16),Am		MOVW (abs16),DWm		CBEQ #8,Dm,d11				MOVW An,DWm			
D	MOV Dn,(abs16)				MOVW An,(abs16)		MOVW DWn,(abs16)		CBNE #8,Dm,d11				CBEQ #8,(abs8),d7/d11		CBNE #8,(abs8),d7/d11	
E	MOVW (d16,SP),Am		MOVW (d16,SP),DWm		MOVW (d8,SP),Am		MOVW (d8,SP),DWm		MOVW (An),Am				ADDW #8,Am		DIVU Dn,DWm	
F	MOVW An,(d16,SP)		MOVW DWn,(d16,SP)		MOVW An,(d8,SP)		MOVW DWn,(d8,SP)		MOVW An,(Am)				ADDW #16,SP		MULU Dn,Dm	

Extension code: b'0010'

Second nibble / third nibble

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	TBZ (abs8)bp,d7								TBZ (abs8)bp,d11							
1	TBNZ (abs8)bp,d7								TBNZ (abs8)bp,d11							
2	CMP Dn,Dm															
3	ADD Dn,Dm															
4	TBZ (io8)bp,d7								TBZ (io8)bp,d11							
5	TBNZ (io8)bp,d7								TBNZ (io8)bp,d11							
6	OR Dn,Dm															
7	AND Dn,Dm															
8	BSET (io8)bp								BCLR (io8)bp							
9	JMP abs18(label)								JSR abs18(label)							
A	XOR Dn,Dm / XOR #8,Dm															
B	ADDC Dn,Dm															
C	BSET (abs16)bp								BCLR (abs16)bp							
D	BTST (abs16)bp								cmp #8,(abs16)	mov #8,(abs16)			CBEQ #8,(abs16),d7/11		CBNE #8,(abs16),d7/11	
E	TBZ (abs16)bp,d7								TBZ (abs16)bp,d11							
F	TBNZ (abs16)bp,d7								TBNZ (abs16)bp,d11							

Ver2.0 (1997.10.06)

Precautions in Manipulating the Interrupt Flags

Always take the following software precautions when manipulating the interrupt register of the MN101D02 series in order to avoid the CPU from entering infinite loops.

[1] When clearing the interrupt request flag (IR)

When clearing the interrupt request flag by software, use the procedure shown below.

(1) Disable task switching.
(2) Disable interrupts.
(3) Disable task switching.
(4) Enable writing of the interrupt request flag (IRWE).
(5) Clear the interrupt request flag (IR).
(6) Disable writing of the interrupt request flag (IRWE).
(7) Enable interrupts.
(8) Enable task switching.

This is because interrupt drop out or CPU endless loop processing can occur if the interrupt control register is manipulated in the condition in which IRWE='1'. Make sure to set the interrupt request write enable flag IRWE to '0' except when necessary.

- The steps (1), (3) , and (8) above are not needed when multitasking operation is not being made.
- The steps (2) and (7) above are not needed when interrupts have not been enabled.
- The step (3) above is not needed when task initiation is not made within the interrupt servicing routine.

Sample Program

```
check1
    bclr    (x'3f05') 7             .... Stop task timer.
    nop
    tbnz    (x'3f05') 7, check1     .... Verify that the task timer has stopped.
    and     x'bf', psw              .... Disable interrupts (MIE=0).
check2
    bclr    (x'3f05') 7             .... Stop task timer.
    nop
    tbnz    (x'3f05') 7, chech2     .... Verify that the task timer has stopped.
    bset    (x'3f01') 2             .... IRWE=1
    bclr    (xxxxICR) 0             .... Clear interrupt request flag.
    bclr    (x'3f01') 2             .... IRWE=0
    or      x'40', psw              .... Enable interrupts (MIE=1).
    bset    (x'3f05') 7             .... Start task timer.
```

[2] When setting the interrupt request flag (IR)

When setting the interrupt request flag by software, use the procedure shown below.

(1) Disable task switching.
(2) Disable interrupts.
(3) Disable task switching.
(4) Enable writing of the interrupt request flag (IRWE).
(5) Set the interrupt request flag (IR).
(6) Disable writing of the interrupt request flag (IRWE).
(7) Enable interrupts.
(8) Enable task switching.

- The steps (1) ~ (3), (7), and (8) above are not needed when multitasking operation is not being made.
- The step (3) above is not needed when task initiation is not made within the interrupt servicing routine.

[3] When clearing the interrupt enable flag (IE) and when changing the interrupt priority level (LV1~0)

Disable interrupts by clearing the MIE flag (bit 6 of PSW) before clearing the interrupt enable flag (IE) and before changing the interrupt priority level (LV1~0).

(1) Disable interrupts.
(2) Clear the interrupt enable flag (IE) or change the interrupt priority level (LV1~0).
(3) Enable interrupts.

- The steps (1) and (3) above are not required within an interrupt servicing routine.

Sample Program 1

```
and     x'bf', psw          ; Clear MIE (disable interrupts).
bclr    (xxxICR)  1         ; Clear IE.
or      x'40', psw          ; Set MIE (enable interrupts).
```

Sample Program 2

```
and     x'bf', psw          ; Clear MIE (disable interrupts).
bclr    (xxxICR)  7         ; Clear LV1.
bset    (xxxICR)  6         ; Set LV0.
or      x'40', psw          ; Set MIE (enable interrupts).
```

[4] When setting the interrupt enable flag (IE)

There is no need to disable interrupts.

MN101D02D/02E/02F/02G/02H
LSI User's Manual

July, 1999 2nd Edition

Issued by Matsushita Electric Industrial Co., Ltd.
Matsushita Electronics Corporation

Semiconductor Company Matsushita Electronics Corporation

Nagaokakyo, Kyoto, 617-8520 Japan
Tel: (075) 951-8151
http://www.mec.panasonic.co.jp

SALES OFFICES

■ U.S.A. SALES OFFICE

Panasonic Industrial Company [PIC]

- **New Jersey Office:**
  2 Panasonic Way, Secaucus, New Jersey 07094
  Tel: 201-392-6173
  Fax: 201-392-4652
- **Milpitas Office:**
  1600 McCandless Drive, Milpitas, California 95035
  Tel: 408-945-5630
  Fax: 408-946-9063
- **Chicago Office:**
  1707 N. Randall Road, Elgin, Illinois 60123-7847
  Tel: 847-468-5829
  Fax: 847-468-5725
- **Atlanta Office:**
  1225 Northbrook Parkway, Suite 1-151,
  Suwanee, Georgia 30174
  Tel: 770-338-6940
  Fax: 770-338-6849
- **San Diego Office:**
  9444 Balboa Avenue, Suite 185
  San Diego, California 92123
  Tel: 619-503-2940
  Fax: 619-715-5545

■ CANADA SALES OFFICE

Panasonic Canada Inc. [PCI]
5700 Ambler Drive Mississauga, Ontario, L4W 2T3
Tel: 905-624-5010
Fax: 905-624-9880

■ GERMANY SALES OFFICE

Panasonic Industrial Europe G.m.b.H. [PIEG]

- **Munich Office:**
  Hans-Pinsel-Strasse 2 85540 Haar
  Tel: 89-46159-156
  Fax: 89-46159-195

■ U.K. SALES OFFICE

Panasonic Industrial Europe Ltd. [PIEL]

- **Electric component Group:**
  Willoughby Road, Bracknell, Berkshire RG12 8FP
  Tel: 1344-85-3773
  Fax: 1344-85-3853

■ FRANCE SALES OFFICE

Panasonic Industrial Europe G.m.b.H. [PIEG]

- **Paris Office:**
  270, Avenue de President Wilson
  93218 La Plaine Saint-Denis Cedex
  Tel: 14946-4413
  Fax: 14946-0007

■ ITALY SALES OFFICE

Panasonic Industrial Europe G.m.b.H. [PIEG]

- **Milano Office:**
  Via Lucini N19, 20125 Milano
  Tel: 2678-8266
  Fax: 2668-8207

■ HONG KONG SALES OFFICE

Panasonic Shun Hing Industrial Sales (Hong Kong) Co., Ltd. [PSI(HK)]
11/F, Great Eagle Centre, 23 Harbour Road,
Wanchai, Hong Kong.
Tel: 2529-7322
Fax: 2865-3697

■ TAIWAN SALES OFFICE

Panasonic Industrial Sales Taiwan Co.,Ltd. [PIST]

- **Head Office:**
  6th Floor, Tai Ping & First Building No.550. Sec.4,
  Chung Hsiao E. Rd. Taipei 10516
  Tel: 2-2757-1900
  Fax: 2-2757-1906
- **Kaohsiung Office:**
  6th Floor, Hsien 1st Road Kaohsiung
  Tel: 7-223-5815
  Fax: 7-224-8362

■ SINGAPORE SALES OFFICE

Panasonic Semiconductor of South Asia [PSSA]
300 Beach Road # 16-01
The Concourse Singapore 199555
Tel: 390-3688
Fax: 390-3689

■ MALAYSIA SALES OFFICE

Panasonic Industrial Company (Malaysia) Sdn. Bhd.

- **Head Office: [PICM]**
  Tingkat 16B Menara PKNS PJ No.17,Jalan Yong
  Shook Lin 46050 Petaling Jaya Selangor Darul Ehsan
  Malaysia
  Tel: 03-7516606
  Fax: 03-7516666
- **Penang Office:**
  Suite 20-17,MWE PLAZA No.8,Lebuh Farquhar,10200
  Penang Malaysia
  Tel: 04-2625550
  Fax: 04-2619989
- **Johore Sales Office:**
  39-01 Jaran Sri Perkasa 2/1,Taman Tampoi
  Utama,Tampoi 81200 Johor Bahru,Johor Malaysia
  Tel: 07-241-3822
  Fax: 07-241-3996

■ CHINA SALES OFFICE

Panasonic SH Industrial Sales (Shenzhen) Co., Ltd. [PSI(SZ)]
7A-107, International Business & Exhibition Centre,
Futian Free Trade Zone, Shenzhen 518048
Tel: 755-359-8500
Fax: 755-359-8516

Panasonic Industrial (Shanghai) Co., Ltd. [PICS]
1F, Block A, Development Mansion, 51 Ri Jing Street,
Wai Gao Qiao Free Trade Zone, Shanghai 200137
Tel: 21-5866-6114
Fax: 21-5866-8000

■ THAILAND SALES OFFICE

Panasonic Industrial (Thailand) Ltd. [PICT]
252/133 Muang Thai-Phatra Complex Building,31st
Fl.Rachadaphisek Rd.,Huaykwang,Bangkok 10320
Tel: 02-6933407
Fax: 02-6933423