www.DataSheet4U.com

FUJITSU SEMICONDUCTOR

CONTROLLER MANUAL

CM44-10118-2E

F²MC™-16LX

16-BIT MICROCONTROLLER

MB90385 Series

HARDWARE MANUAL

DataSheet4U.com

DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com

DataShee

DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

F²MC™-16LX

16-BIT MICROCONTROLLER

MB90385 Series

HARDWARE MANUAL

et4U.com
DataShee
DataSheet4U.com

FUJITSU LIMITED

DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

www.DataSheet4U.com

et4U.com

DataShee

DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

PREFACE

■ Manual Objectives and Readers

Thank you very much for your continued patronage of Fujitsu semiconductor products.

The MB90385 series is one of the general-purpose products in the F^{2}MC^{TM}-16LX family of 16-bit single-chip microcontrollers that is developed by using an application-specific integrated circuit (ASIC).

This manual covers the functions and operations of the MB90385 series for engineers to develop LSIs using this series.

■ Trademarks

F^{2}MC is the abbreviation of FUJITSU Flexible Microcontroller.

Other system and product names in this manual are trademarks of respective companies or organizations.

The symbols ^{TM} and ® are sometimes omitted in this manual.

- The contents of this document are subject to change without notice.
  Customers are advised to consult with FUJITSU sales representatives before ordering.
- The information, such as descriptions of function and application circuit examples, in this document are presented solely for the purpose of reference to show examples of operations and uses of Fujitsu semiconductor device; Fujitsu does not warrant proper operation of the device with respect to use based on such information. When you develop equipment incorporating the device based on such information, you must assume any responsibility arising out of such use of the information. Fujitsu assumes no liability for any damages whatsoever arising out of the use of the information.
- Any information in this document, including descriptions of function and schematic diagrams, shall not be construed as license of the use or exercise of any intellectual property right, such as patent right or copyright, or any other right of Fujitsu or any third party or does Fujitsu warrant non-infringement of any third-party' s intellectual property right or other right by using such information. Fujitsu assumes no liability for any infringement of the intellectual property rights or other rights of third parties which would result from the use of information contained herein.
- The products described in this document are designed, developed and manufactured as contemplated for general use, including without limitation, ordinary industrial use, general office use, personal use, and household use, but are not designed, developed and manufactured as contemplated (1) for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could have a serious effect to the public, and could lead directly to death, personal injury, severe physical damage or other loss (i.e., nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system), or (2) for use requiring extremely high reliability (i.e., submersible repeater and artificial satellite).
  Please note that Fujitsu will not be liable against you and/or any third party for any claims or damages arising in connection with above-mentioned uses of the products.
- Any semiconductor devices have an inherent chance of failure. You must protect against injury, damage or loss from such failures by incorporating safety design measures into your facility and equipment such as redundancy, fire protection, and prevention of over-current levels and other abnormal operating conditions.
- If any products described in this document represent goods or technologies subject to certain restrictions on export under the Foreign Exchange and Foreign Trade Law of Japan, the prior authorization by Japanese government will be required for export of those products from Japan.

DataSheet4U.com ©2005 FUJITSU LIMITED Printed in Japan www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

HOW TO READ THIS MANUAL

■ Page Structure

Each section content can be read easily because it is mentioned within one page or double spread.

A summary under the title in each section outlines the section contents.

The top-level title at the top of a double spread indicates where you are reading without returning to the table of contents or the chapter title page.

■ How to Find Information

To find information in each section, use the following index in addition to general table of contents and index.

● Register index

This index helps you find the page containing the explanation of the corresponding register from a register name or related resource name. You can also check the mapped addresses on memory and reset values.

● Pin function index

This index helps you find the page containing the explanation or block diagram of the corresponding pin from a pin number, pin name, or related resource name. You can also check the circuit types.

et4U.com DataShee

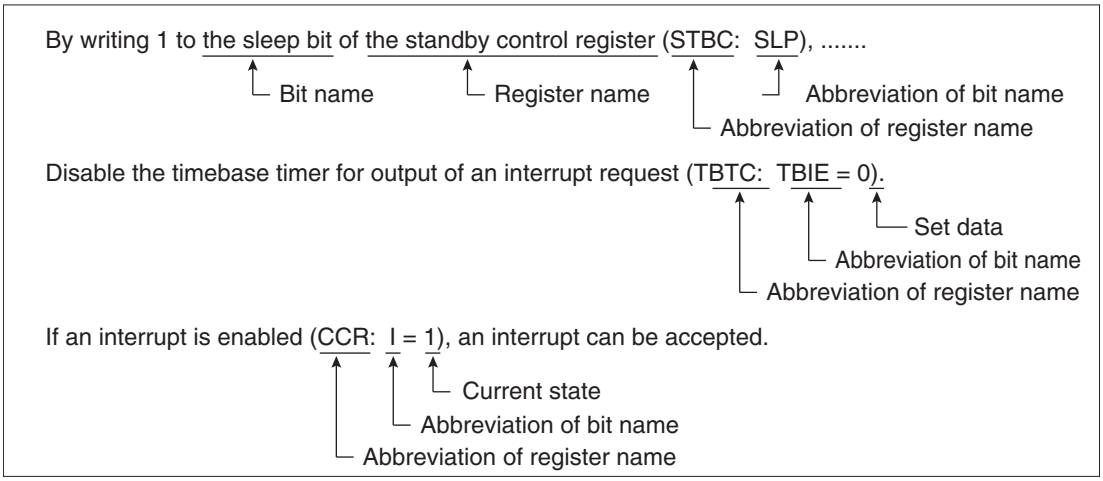
DataSheet4U.com

● Interrupt vector index

This index helps you find the page containing the explanation of a corresponding interrupt from a name of resource generating the interrupt or an interrupt number. You can also check the names and addresses of interrupt control registers (ICRs), and the interrupt vector addresses.

■ Representation of Register Name and Pin Name

● Representation of register name and bit name



DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Representation of dual-purpose pin

P25/SCK pin

Some pins are dual-purpose pins which functions can be switched by the setting of program. A slash (/) separates and represents the names corresponding to the functions of the dual-purpose pins.

■ Register Representation

The F^{2}MC-16LX family is a CPU with a 16-bit bus width. The bit position of each control register and data register is given in 16 bits.

In 16-bit registers, bits 15 to 8 are allocated to odd addresses and bits 7 to 0 even addresses.

Even in 8-bit registers, the position of bits allocated to odd addresses is given in bits 15 to 8.

The F^{2}MC-16LX family enables access to 8-bit data in order to increase the efficiency of programs. So, if odd-address registers are accessed in 8 bits, bits 7 to 0 in data correspond to bits 15 to 8 in the manual representation.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com

DataShee

DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CONTENTS

CHAPTER 1 OVERVIEW 1

1.1 Features of MB90385 Series 2
1.2 Product Lineup for MB90385 Series 5
1.3 Block Diagram of MB90385 Series 8
1.4 Pin Assignment 9
1.5 Package Dimension 10
1.6 Pin Description 11
1.7 I/O Circuit 14

CHAPTER 2 HANDLING DEVICES 17

2.1 Precautions when Handling Devices 18

CHAPTER 3 CPU 21

3.1 Memory Space 22
3.1.1 Mapping of and Access to Memory Space 24
3.1.2 Memory Map 26
3.1.3 Addressing 27
3.1.4 Linear Addressing 28
3.1.5 Bank Addressing 29
3.1.6 Allocation of Multi-byte Data on Memory 31
3.2 Dedicated Registers 33
3.2.1 Dedicated Registers and General-purpose Register 35
3.2.2 Accumulator (A) 36
3.2.3 Stack Pointer (USP, SSP) 39
3.2.4 Processor Status (PS) 42
3.2.4.1 Condition Code Register (PS: CCR) 43
3.2.4.2 Register Bank Pointer (PS: RP) 45
3.2.4.3 Interrupt Level Mask Register (PS: ILM) 46
3.2.5 Program Counter (PC) 47
3.2.6 Direct Page Register (DPR) 48
3.2.7 Bank Register (PCB, DTB, USB, SSB, and ADB) 49
3.3 General-purpose Register 50
3.4 Prefix Codes 52
3.4.1 Bank Select Prefix (PCB, DTB, ADB, and SPB) 53
3.4.2 Common Register Bank Prefix (CMR) 55
3.4.3 Flag Change Inhibit Prefix (NCC) 56
3.4.4 Restrictions on Prefix Code 57
3.5 Interrupt 59
3.5.1 Interrupt Factor and Interrupt Vector 61
3.5.2 Interrupt Control Registers and Resources 64
3.5.3 Interrupt Control Register (ICR00 to ICR15) 66
3.5.4 Function of Interrupt Control Register 68
3.5.5 Hardware Interrupt 71

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.6 Operation of Hardware Interrupt 74
3.5.7 Procedure for Use of Hardware Interrupt 76
3.5.8 Multiple Interrupts 77
3.5.9 Software Interrupt 79
3.5.10 Interrupt by EI^{2}OS 80
3.5.11 EI^{2}OS Descriptor (ISD) 82
3.5.12 Each Register of EI^{2}OS Descriptor (ISD) 84
3.5.13 Operation of EI^{2}OS 87
3.5.14 Procedure for Use of EI^{2}OS 88
3.5.15 EI^{2}OS Processing Time 89
3.5.16 Exception Processing Interrupt 91
3.5.17 Time Required to Start Interrupt Processing 92
3.5.18 Stack Operation for Interrupt Processing 94
3.5.19 Program Example of Interrupt Processing 95
3.6 Reset 99
3.6.1 Reset Factors and Oscillation Stabilization Wait Time 101
3.6.2 External Reset Pin 103
3.6.3 Reset Operation 104
3.6.4 Reset Factor Bit 106
3.6.5 State of Each Pin at Reset 108
3.7 Clocks 109
3.7.1 Block Diagram of Clock Generation Section 112
3.7.2 Register in Clock Generation Section 114
3.7.3 Clock Select Register (CKSCR) 115
3.7.4 Clock Mode 118
3.7.5 Oscillation Stabilization Wait Time 122
3.7.6 Connection of Oscillator and External Clock 123
3.8 Low-power Consumption Mode 124
3.8.1 Block Diagram of Low-power Consumption Circuit 127
3.8.2 Registers for Setting Low-power Consumption Modes 129
3.8.3 Low-power Consumption Mode Control Register (LPMCR) 130
3.8.4 CPU Intermittent Operation Mode 133
3.8.5 Standby Mode 134
3.8.5.1 Sleep Mode 135
3.8.5.2 Watch mode 137
3.8.5.3 Timebase Timer Mode 139
3.8.5.4 Stop Mode 141
3.8.6 State Transition in Standby Mode 144
3.8.7 Pin State in Standby Mode, at Reset 145
3.8.8 Precautions when Using Low-power Consumption Mode 146
3.9 CPU Mode 149
3.9.1 Mode Pins (MD2 to MD0) 150
3.9.2 Mode Data 152
3.9.3 Memory Access Mode 154
3.9.4 Selection of Memory Access Mode 155

CHAPTER 4 I/O PORT 157

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.1 Overview of I/O Port 158
4.2 Registers of I/O Port 160
4.3 Port 1 161
4.3.1 Registers for Port 1 (PDR1, DDR1) 163
4.3.2 Operation of Port 1 164
4.4 Port 2 166
4.4.1 Registers for Port 2 (PDR2, DDR2) 168
4.4.2 Operation of Port 2 169
4.5 Port 3 171
4.5.1 Registers for Port 3 (PDR3, DDR3) 173
4.5.2 Operation of Port 3 174
4.6 Port 4 176
4.6.1 Registers for Port 4 (PDR4, DDR4) 178
4.6.2 Operation of Port 4 179
4.7 Port 5 181
4.7.1 Registers for Port 5 (PDR5, DDR5, ADER) 183
4.7.2 Operation of Port 5 185

CHAPTER 5 TIMEBASE TIMER 187
5.1 Overview of Timebase Timer 188
5.2 Block Diagram of Timebase Timer 190
5.3 Configuration of Timebase Timer 192
5.3.1 Timebase Timer Control Register (TBTC) 193
5.4 Timebase Timer Interrupt 195
5.5 Explanation of Operation of Timebase Timer 196
5.6 Precautions when Using Timebase Timer 200
5.7 Program Example of Timebase Timer 201

CHAPTER 6 WATCHDOG TIMER 203
6.1 Overview of Watchdog Timer 204
6.2 Configuration of Watchdog Timer 205
6.3 Watchdog Timer Registers 207
6.3.1 Watchdog Timer Control Register (WDTC) 208
6.4 Explanation of Operation of Watchdog Timer 210
6.5 Precautions when Using Watchdog Timer 213
6.6 Program Examples of Watchdog Timer 214

CHAPTER 7 16-BIT INPUT/OUTPUT TIMER 215
7.1 Overview of 16-bit Input/Output Timer 216
7.2 Block Diagram of 16-bit Input/Output Timer 217
7.2.1 Block Diagram of 16-bit Free-run Timer 218
7.2.2 Block Diagram of Input Capture 220
7.3 Configuration of 16-bit Input/Output Timer 222
7.3.1 Timer Counter Control Status Register (TCCS) 225
7.3.2 Timer Counter Data Register (TCDT) 227
7.3.3 Input Capture Control Status Registers (ICS01 and ICS23) 229
7.3.4 Input Capture Data Registers 0 to 3 (IPCP0 to IPCP3) 232

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

7.4 Interrupts of 16-bit Input/Output Timer 233
7.5 Explanation of Operation of 16-bit Free-run Timer 234
7.6 Explanation of Operation of Input Capture 236
7.7 Precautions when Using 16-bit Input/Output Timer 239
7.8 Program Example of 16-bit Input/Output Timer 240

CHAPTER 8 16-BIT RELOAD TIMER 243
8.1 Overview of 16-bit Reload Timer 244
8.2 Block Diagram of 16-bit Reload Timer 246
8.3 Configuration of 16-bit Reload Timer 248
8.3.1 Timer Control Status Registers (High) (TMCSR0: H, TMCSR1: H) 251
8.3.2 Timer Control Status Registers (Low) (TMCSR0: L, TMCSR1: L) 253
8.3.3 16-bit Timer Registers (TMR0, TMR1) 255
8.3.4 16-bit Reload Registers (TMRLR0, TMRLR1) 256
8.4 Interrupts of 16-bit Reload Timer 257
8.5 Explanation of Operation of 16-bit Reload Timer 258
8.5.1 Operation in Internal Clock Mode 260
8.5.2 Operation in Event Count Mode 265
8.6 Precautions when Using 16-bit Reload Timer 268
8.7 Program Example of 16-bit Reload Timer 269

CHAPTER 9 WATCH TIMER 273
9.1 Overview of Watch Timer 274
9.2 Block Diagram of Watch Timer 276
9.3 Configuration of Watch Timer 278
9.3.1 Watch Timer Control Register (WTC) 279
9.4 Watch Timer Interrupt 281
9.5 Explanation of Operation of Watch Timer 282
9.6 Program Example of Watch Timer 284

CHAPTER 10 8-/16-BIT PPG TIMER 285
10.1 Overview of 8-/16-bit PPG Timer 286
10.2 Block Diagram of 8-/16-bit PPG Timer 289
10.2.1 Block Diagram for 8-/16-bit PPG Timer 0 290
10.2.2 Block Diagram of 8-/16-bit PPG Timer 1 293
10.3 Configuration of 8-/16-bit PPG Timer 296
10.3.1 PPG0 Operation Mode Control Register (PPGC0) 298
10.3.2 PPG1 Operation Mode Control Register (PPGC1) 300
10.3.3 PPG0/1 Count Clock Select Register (PPG01) 302
10.3.4 PPG Reload Registers (PRLL0/PRLH0, PRLL1/PRLH1) 304
10.4 Interrupts of 8-/16-bit PPG Timer 305
10.5 Explanation of Operation of 8-/16-bit PPG Timer 307
10.5.1 8-bit PPG Output 2-channel Independent Operation Mode 308
10.5.2 16-bit PPG Output Operation Mode 310
10.5.3 8+8-bit PPG Output Operation Mode 313
10.6 Precautions when Using 8-/16-bit PPG Timer 316

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 11 DELAYED INTERRUPT GENERATION MODULE 319
11.1 Overview of Delayed Interrupt Generation Module 320
11.2 Block Diagram of Delayed Interrupt Generation Module 321
11.3 Configuration of Delayed Interrupt Generation Module 322
11.3.1 Delayed Interrupt Request Generate/Cancel Register (DIRR) 323
11.4 Explanation of Operation of Delayed Interrupt Generation Module 324
11.5 Precautions when Using Delayed Interrupt Generation Module 325
11.6 Program Example of Delayed Interrupt Generation Module 326

CHAPTER 12 DTP/EXTERNAL INTERRUPT 327
12.1 Overview of DTP/External Interrupt 328
12.2 Block Diagram of DTP/External Interrupt 329
12.3 Configuration of DTP/External Interrupt 331
12.3.1 DTP/External Interrupt Factor Register (EIRR) 332
12.3.2 DTP/External Interrupt Enable Register (ENIR) 333
12.3.3 Detection Level Setting Register (ELVR) (High) 334
12.3.4 Detection Level Setting Register (ELVR) (Low) 335
12.4 Explanation of Operation of DTP/External Interrupt 336
12.4.1 External Interrupt Function 339
12.4.2 DTP Function 340
12.5 Precautions when Using DTP/External Interrupt 341
12.6 Program Example of DTP/External Interrupt Circuit 343

CHAPTER 13 8-/10-BIT A/D CONVERTER 347
13.1 Overview of 8-/10-bit A/D Converter 348
13.2 Block Diagram of 8-/10-bit A/D Converter 349
13.3 Configuration of 8-/10-bit A/D Converter 352
13.3.1 A/D Control Status Register (High) (ADCS: H) 354
13.3.2 A/D Control Status Register (Low) (ADCS: L) 356
13.3.3 A/D Data Register (High) (ADCR: H) 359
13.3.4 A/D Data Register (Low) (ADCR: L) 361
13.3.5 Analog Input Enable Register (ADER) 362
13.4 Interrupt of 8-/10-bit A/D Converter 364
13.5 Explanation of Operation of 8-/10-bit A/D Converter 365
13.5.1 Single Conversion Mode 366
13.5.2 Continuous Conversion Mode 368
13.5.3 Pause-conversion Mode 370
13.5.4 Conversion Using EI^{2}OS Function 372
13.5.5 A/D-converted Data Protection Function 373
13.6 Precautions when Using 8-/10-bit A/D Converter 375

CHAPTER 14 UART1 377
14.1 Overview of UART1 378
14.2 Block Diagram of UART1 380
14.3 Configuration of UART1 383
14.3.1 Serial Control Register 1 (SCR1) 385
14.3.2 Serial Mode Register 1 (SMR1) 387

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.3.3 Serial Status Register 1 (SSR1) 389
14.3.4 Serial Input Data Register 1 (SIDR1) and Serial Output Data Register 1 (SODR1) 391
14.3.5 Communication Prescaler Control Register 1 (CDCR1) 393
14.4 Interrupt of UART1 395
14.4.1 Generation of Receive Interrupt and Timing of Flag Set 397
14.4.2 Generation of Transmit Interrupt and Timing of Flag Set 399
14.5 Baud Rate of UART1 400
14.5.1 Baud Rate by Dedicated Baud Rate Generator 402
14.5.2 Baud Rate by Internal Timer (16-bit Reload Timer) 405
14.5.3 Baud Rate by External Clock 407
14.6 Explanation of Operation of UART1 408
14.6.1 Operation in Asynchronous Mode (Operation Mode 0 or 1) 410
14.6.2 Operation in Clock Synchronous Mode (Operation Mode 2) 415
14.6.3 Bidirectional Communication Function (Operation Modes 0 and 2) 418
14.6.4 Master/Slave Type Communication Function (Multiprocessor Mode) 420
14.7 Precautions when Using UART1 423
14.8 Program Example for UART1 424

CHAPTER 15 CAN CONTROLLER 427

15.1 Overview of CAN Controller 428
15.2 Block Diagram of CAN Controller 429
15.3 Configuration of CAN Controller 433
15.3.1 Control Status Register (High) (CSR: H) 437
15.3.2 Control Status Register (Low) (CSR: L) 439
15.3.3 Last Event Indicate Register (LEIR) 442
15.3.4 Receive/Transmit Error Counter (RTEC) 444
15.3.5 Bit Timing Register (BTR) 446
15.3.6 Message Buffer Valid Register (BVALR) 450
15.3.7 IDE Register (IDER) 452
15.3.8 Transmission Request Register (TREQR) 454
15.3.9 Transmission RTR Register (TRTRR) 456
15.3.10 Remote Frame Receiving Wait Register (RFWTR) 458
15.3.11 Transmission Cancel Register (TCANR) 460
15.3.12 Transmission Complete Register (TCR) 462
15.3.13 Transmission Complete Interrupt Enable Register (TIER) 464
15.3.14 Reception Complete Register (RCR) 466
15.3.15 Reception RTR Register (RRTRR) 468
15.3.16 Reception Overrun Register (ROVRR) 470
15.3.17 Reception Complete Interrupt Enable Register (RIER) 472
15.3.18 Acceptance Mask Select Register (AMSR) 474
15.3.19 Acceptance Mask Register (AMR) 476
15.3.20 Message Buffers 478
15.3.21 ID Register (IDRx, x = 7 to 0) 479
15.3.22 DLC Register (DLCR) 482
15.3.23 Data Register (DTR) 483
15.4 Interrupts of CAN Controller 484
15.5 Explanation of Operation of CAN Controller 486

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.5.1 Transmission 487
15.5.2 Reception 490
15.5.3 Procedures for Transmitting and Receiving 494
15.5.4 Setting Multiple Message Receiving 501
15.6 Precautions when Using CAN Controller 503
15.7 Program Example of CAN Controller 504

CHAPTER 16 8/16 ADDRESS MATCH DETECTION FUNCTION 507
16.1 Overview of Address Match Detection Function 508
16.2 Block Diagram of Address Match Detection Function 509
16.3 Configuration of Address Match Detection Function 510
16.3.1 Address Detection Control Register (PACSR) 511
16.3.2 Detect Address Setting Registers (PADR0 and PADR1) 513
16.4 Explanation of Operation of Address Match Detection Function 515
16.4.1 Example of using Address Match Detection Function 516
16.5 Program Example of Address Match Detection Function 521

CHAPTER 17 ROM MIRRORING FUNCTION SELECT MODULE 523
17.1 Overview of ROM Mirroring Function Select Module 524
17.2 ROM Mirroring Function Select Register (ROMM) 526

CHAPTER 18 512 KBIT FLASH MEMORY 527
18.1 Overview of 512 Kbit Flash Memory 528
18.2 Registers and Sector Configuration of Flash Memory 529
18.3 Flash Memory Control Status Register (FMCS) 530
18.4 How to Start Automatic Algorithm of Flash Memory 533
18.5 Check the Execution State of Automatic Algorithm 535
18.5.1 Data Polling Flag (DQ7) 537
18.5.2 Toggle Bit Flag (DQ6) 539
18.5.3 Timing Limit Over Flag (DQ5) 540
18.5.4 Sector Erase Timer Flag (DQ3) 541
18.5.5 Toggle Bit 2 Flag (DQ2) 542
18.6 Details of Programming/Erasing Flash Memory 544
18.6.1 Read/Reset State in Flash Memory 545
18.6.2 Data Programming to Flash Memory 546
18.6.3 Data Erase from Flash Memory (Chip Erase) 548
18.6.4 Erasing Any Data in Flash Memory (Sector Erasing) 549
18.6.5 Sector Erase Suspension in Flash Memory 551
18.6.6 Sector Erase Resumption in Flash Memory 552
18.7 Program Example of 512 Kbit Flash Memory 553

CHAPTER 19 FLASH SERIAL PROGRAMMING CONNECTION 557
19.1 Basic Configuration of Serial Programming Connection for F^2MC-16LX MB90F387/S 558
19.2 Connection Example in Single-chip Mode (User Power Supply) 561
19.3 Connection Example in Single-chip Mode (Writer Power Supply) 563
19.4 Example of Minimum Connection to Flash Microcontroller Programmer (User Power Supply) 565
19.5 Example of Minimum Connection to Flash Microcontroller Programmer (Writer Power Supply) 567

et4U.com DataSheet4U.com DataShee
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

APPENDIX 569
APPENDIX A Instructions 570
A.1 Instruction Types 571
A.2 Addressing 572
A.3 Direct Addressing 574
A.4 Indirect Addressing 580
A.5 Execution Cycle Count 587
A.6 Effective Address Field 590
A.7 How to Read the Instruction List 591
A.8 F^2MC-16LX Instruction List 594
A.9 Instruction Map 609
APPENDIX B Register Index 631
APPENDIX C Pin Function Index 640
APPENDIX D Interrupt Vector Index 642

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 1
OVERVIEW

This chapter describes the features and basic specifications of the MB90385 series.

1.1 Features of MB90385 Series
1.2 Product Lineup for MB90385 Series
1.3 Block Diagram of MB90385 Series
1.4 Pin Assignment
1.5 Package Dimension
1.6 Pin Description
1.7 I/O Circuit

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

1.1 Features of MB90385 Series

The MB90385 series is a general-purpose, high-performance 16-bit microcontroller designed for control of processors such as consumer products requiring high-speed real-time processing. This series has a full CAN interface.

The instruction system is based on the architecture of the F^{2}MC family and provides additional high-level language instructions, extended addressing modes, enhanced multiply/divide instructions, and enriched bit processing instructions. A 32-bit accumulator enables long-word data (32 bits) processing.

■ Features of MB90385 Series

- Clock
  - Built-in PLL clock multiplying circuit
  - Machine clock (PLL clock) selectable from 1/2 frequency of oscillation clock or 1 to 4-multiplied oscillation clock (4 MHz to 16 MHz when oscillation clock is 4 MHz)
  - Subclock operation (8.192 kHz) (MB90387, MB90F387)
  - Minimum instruction execution time: 62.5 ns (4-MHz oscillation clock, 4-multiplied PLL clock)

et4U.com DataShee

- 16-MB CPU memory space

DataSheet4U.com

  - Internal 24-bit addressing
- Instruction system optimized for controllers
  - Various data types (bit, byte, word, long word)
  - 23 types of addressing modes
  - Enhanced signed instructions of multiplication/division and RETI instruction function
  - High-accuracy operations enhanced by 32-bit accumulator
- Instruction system for high-level language (C language)/multitask
  - System stack pointer
  - Enhanced pointer indirect instructions
  - Barrel shift instructions
- Higher execution speed
  - 4-byte instruction queue
- Powerful interrupt function
  - Powerful interrupt function with 8 levels and 34 factors

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- CPU-independent automatic data transfer function
  - Extended intelligent I/O service (EI^{2}OS): Maximum 16 channels
- Lower-power consumption (standby) modes
  - Sleep mode (stops CPU clock)
  - Timebase timer mode (operates only oscillation clock and subclock, timebase timer and watch timer)
  - Watch mode (operates only subclock and watch timer)
  - Stop mode (stops oscillation clock and subclock)
  - CPU Intermittent operation mode
- Process
  - CMOS Technology
- I/O ports
  - General-purpose I/O ports (CMOS output): 34 ports (for M90387 or M90F387) (included 4 output ports for high current)

  Note : 36 ports (for MB90387S or MB90F387S) on condition of unusing sub-clock.
- Timers
  - Timebase timer, watch timer, watchdog timer: 1 channel
  - 8/16-bit PPG timer: 8 bits × 4 channels or 16 bits × 2 channels
  - 16-bit reload timer: 2 channels
  - 16-bit I/O timer
    - 16-bit free-run timer: 1 channel
    - 16-bit input capture (ICU): 4 channels

  By detecting the edge of the pin input, the count value of the 16-bit free-run timer is latched to generate an interrupt request.
- CAN Controller: 1 channel
  - Conforms to CAN Specification Ver. 2.0A and Ver. 2.0B.
  - Built-in 8 message buffers
  - Transfer rate: 10 Kbps to 1 Mbps (at 16-MHz machine clock frequency)
  - CAN wake-up
- UART1 (SCI): 1 channel
  - Full-duplex double buffer
  - Clock asynchronous or clock synchronous serial transfer
- DTP/external interrupt: 4 channels
  - External input to start EI^{2}OS and external interrupt generation module

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Delayed interrupt generation module
  - Generates interrupt request for task switching
- 8-/10-bit A/D converter: 8 channels
  - 8-bit and 10-bit resolutions
  - Start by external trigger input
  - Conversion time: 6.125µs (including sampling time at 16-MHz machine clock frequency)
- Program patch function
  - Detects address match for two address pointers

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

1.2 Product Lineup for MB90385 Series

The MB90385 series is available in three types. This section provides the product lineup, CPU, and resources.

■ Product Lineup for MB90385 Series

Table 1.2-1 Product Lineup for MB90385 Series

	MB90V495G	MB90F387/S	MB90387/S
Classification	Evaluation product	Flash ROM	Mask ROM
ROM Size	--	64 KB	
RAM Size	6 KB	2 KB	
Clock	Dual system products	MB90F387: dual system products MB90F387S: single system product	MB90387: dual system products MB90387S: single system product
Process	CMOS		
Package	PGA256	LQFP-48 (with 0.50-mm pin pitch),	
Operating supply voltage	4.5 V to 5.5 V	3.5 V to 5.5 V	
Power supply for emulator *	Not provided	--	

*: Setting of DIP Switch (S2) when using emulation pod (MB2145-507). For details, refer to the MB2145-507 Hardware Manual (Section 2.7 Emulator-specific Power Supply).

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ CPU and Resources for MB90385 Series

Table 1.2-2 CPU and Resources for MB90385 Series (1/2)

		MB90V495G	MB90F387/S	MB90387/S
CPU Function		Basic instruction count: 351 Instruction bit length: 8 or 16 bits Instruction length: 1 to 7 bytes Data bit length: 1, 8, or 16 bits		
		Minimum instruction execution time: 62.5 ns (at 16-MHz machine clock frequency)		
		Interrupt processing time: 1.5 µs (at 16-MHz machine clock frequency)		
Low-power consumption (standby) modes		Sleep mode, watch mode, timebase timer mode, stop mode, CPU intermittent operation mode		
I/O Ports		General-purpose I/O ports (CMOS output):34 ports (36 ports *) included 4 output ports for high current (P14 to P17)		
Timebase timer		18-bit free-run counter Interrupt cycle: 1.024, 4.096, 16.834, 131.072 ms (at 4-MHz oscillation clock frequency)		
Watchdog timer		Reset cycle: 3.58, 14.33, 57.23, 458.75 ms (at 4-MHz oscillation clock frequency)		
16-bit I/O timers	16-bit free-run timer	Channel count: 1 Overflow interrupt		
	Input capture	Channel count: 4 Free-run timer values saved by pin input (rising edge, falling edge, both edges)		
16-bit reload timer		Channel count: 2 Operation of 16-bit reload timer Count clock cycle: 0.25µs, 0.5µs, 2.0µs (at 16-MHz machine clock frequency) External event countable		
Watch timer		15-bit free-run counter Interrupt cycle: 31.25, 62.5, 12, 250, 500 ms, and 1.0 s, 2.0 s (at 8.192-kHz subclock frequency)		
8-/16-bit PPG timer		Channel count: 2 (operable with 8 bits x 4 channels) PPG operable with 8 bits x 4 channels or 16 bits x 1 channel Pulse waveform output at arbitrary cycle and duty Count clock: 62.5 ns to 1µs (at 16-MHz machine clock frequency)		
Delayed interrupt generation module		Interrupt generation module for switching task Used for Real-time OS		
DTP/external interrupt		Input count: 4 Start on rising or falling edges and by High- or Low-level inputs External interrupts or extended intelligent I/O service (EI^2OS)		

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 1.2-2 CPU and Resources for MB90385 Series (2/2)

	MB90V495G	MB90F387/S	MB90387/S
8-/10-bit A/D converter	Channel count: 8 Resolution: 10 or 8 bits Conversion time: 6.125µs (including sampling time at 16-MHz machine clock frequency) Two or more continuous channels can be converted sequentially (up to 8 channels) Single conversion mode: Selected channel converted once only Continuous conversion mode: Selected channel converted continuously Stop conversion mode: Selected channel converted and temporary stopped alternately		
UART 1	Channel count: 1 Clock synchronous transfer: 62.5 Kbps to 2 Mbps Clock asynchronous transfer: 9,615 bps to 500 kbps Two-way serial communication function, master/slave-connected communication		
CAN	Conforms to CAN Specification Ver. 2.0A and Ver. 2.0B Transmit/receive message buffer: 8 Transfer bit rate: 10 Kbps to 1 Mbps (at 16-MHz machine clock) CAN wake-up		

*: MB90387S, MB90F387S

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

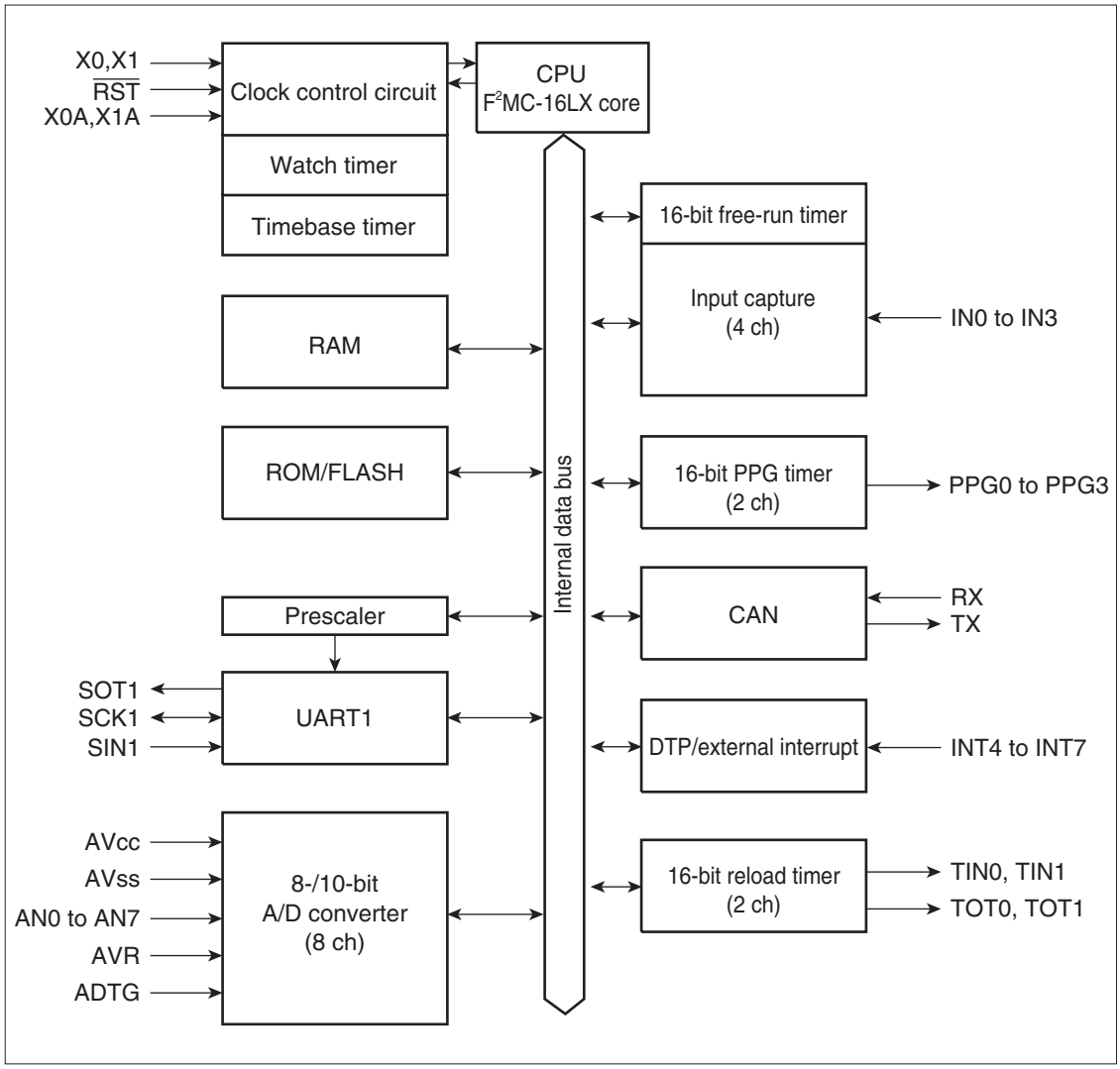
www.DataSheet4U.com

1.3 Block Diagram of MB90385 Series

Block diagram of the MB90385 series is shown in the figure below.

■ Block Diagram of MB90385 Series

Figure 1.3-1 Block Diagram of MB90385 Series



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

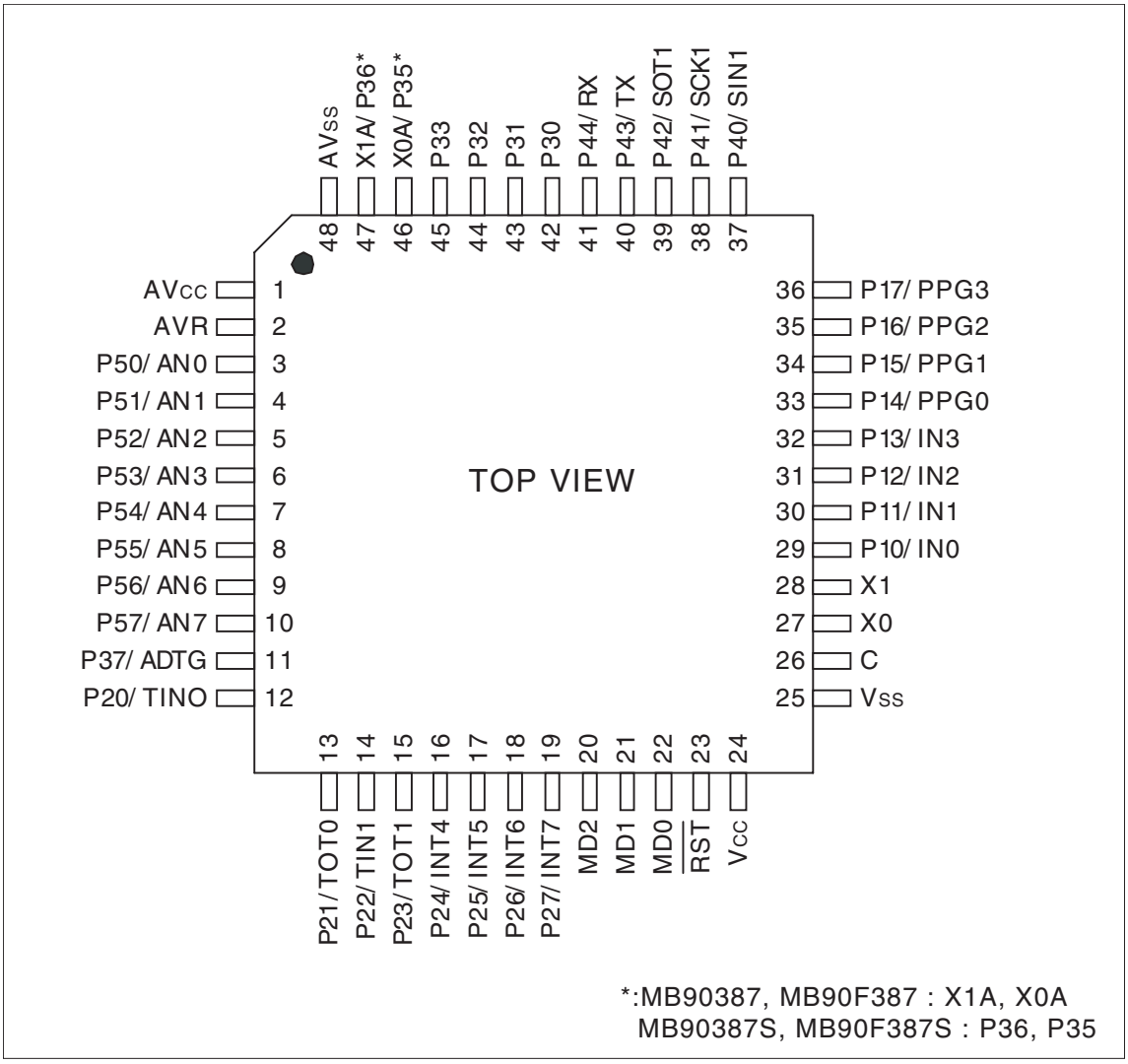
www.DataSheet4U.com

1.4 Pin Assignment

Pin assignment of the MB90385 series is shown in the figure below.

■ Pin Assignment (FPT-48P-M26)

Figure 1.4-1 Pin Assignment (FPT-48P-M26)



DataSheet4U.com
www.DataSheet4U.com

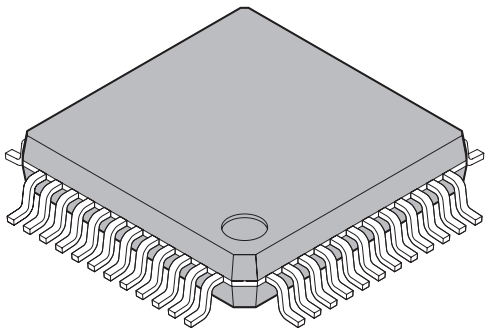
DataSheet4U.com

www.DataSheet4U.com

1.5 Package Dimension

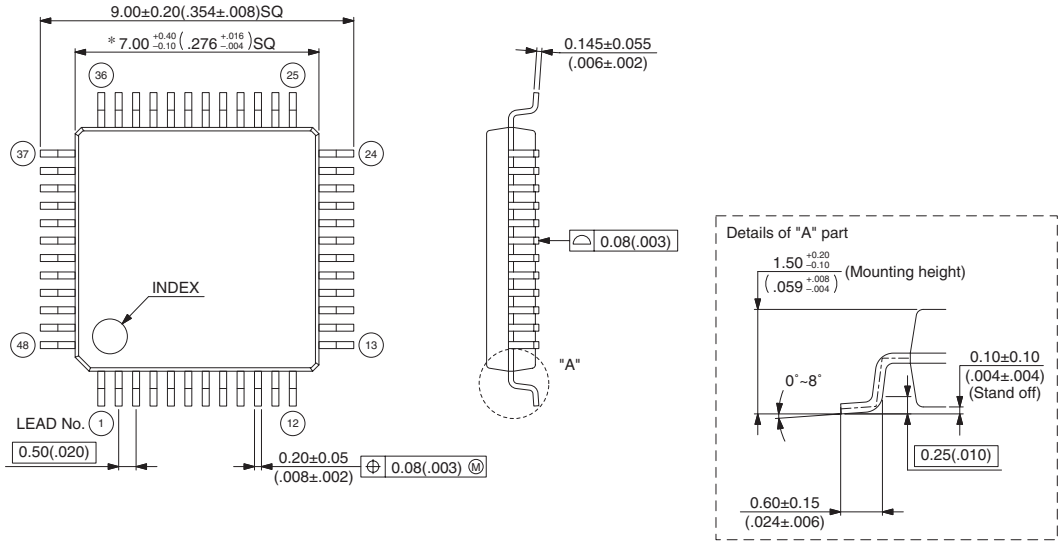
The MB90385 series is available in one type of package.
The package dimensions below are for reference only. Contact Fujitsu for the nominal package dimensions.

■ Package Dimension of FPT-48P-M26

48-pin plastic LQFP  (FPT-48P-M26)	Lead pitch	0.50 mm
	Package width × package length	7 × 7 mm
	Lead shape	Gullwing
	Sealing method	Plastic mold
	Mounting height	1.70 mm MAX
	Weight	0.17 g
	Code (Reference)	P-LFQFP32-7×7-0.50

48-pin plastic LQFP
(FPT-48P-M26)

Note 1) * : These dimensions include resin protrusion.
Note 2) Pins width and pins thickness include plating thickness.
Note 3) Pins width do not include tie bar cutting remainder.



© 2003 FUJITSU LIMITED F48040S-c-2-2

Dimensions in mm (inches).
Note: The values in parentheses are reference values.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

1.6 Pin Description

This section describes the I/O pins and their functions of the MB90385 series.

■ Pin Description

Table 1.6-1 Pin Description (1/3)

Pin No. M26	Pin Name	Circuit Type	Function
1	AV_{CC}	--	V_{CC} power input pin for A/D converter
2	AVR	--	Power (Vref+) input pin for A/D converter. The power supply should not be input exceeding
3 to 10	P50 to P57	E	General-purpose I/O port
	AN0 to AN7		Analog input pin for A/D converter. These pins work when the analog input is set to "enable."
11	P37	D	General-purpose I/O port.
	ADTG		External trigger input pin for A/D converter. This pin should be set to "input port".
12	P20	D	General-purpose I/O port.
	TIN0		Event input pin for reload timer 0. This pin should be set to "input port."
13	P21	D	General-purpose I/O port.
	TOT0		Event output pin for reload timer 0. This pin is enabled only when the output setting is "enabled".
14	P22	D	General-purpose I/O port.
	TIN1		Event input pin for reload timer 1. This pin should be set to "input port."
15	P23	D	General-purpose I/O port.
	TOT1		Event output pin for reload timer 1. This pin is enabled only when the output setting is "enabled".
16 to 19	P24 to P27	D	General-purpose I/O port.
	INT4 to INT7		External interrupt input pins. These pins should be set to "input port."
20	MD2	F	Input pin for selecting operation mode
21	MD1	C	Input pin for selecting operation mode
22	MD0	C	Input pin for selecting operation mode

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 1.6-1 Pin Description (2/3)

Pin No. M26	Pin Name	Circuit Type	Function
23	$\overline{RST}$	B	Input pin for external reset
24	V_{CC}	--	Power (5 V) input pin.
25	V_{SS}	--	Power (0 V) input pin
26	C	--	Capacity pin for stabilizing power supply. This pin should be connected to a ceramic capacitor of approx. 0.1µF.
27	X0	A	High-speed oscillation pin
28	X1	A	High-speed oscillation pin
29 to 32	P10 to P13	D	General-purpose I/O ports.
	IN0 to IN3		Trigger input pins for input capture channels 0 to 3. These pins should be set to "input port".
33 to 36	P14 to P17	G	General-purpose I/O ports. High current output port.
	PPG0 to PPG3		Output pins for PPG timers 01 and 23. These pins are enabled when the output setting is "enabled."
37	P40	D	General-purpose I/O port
	SIN1		Serial data input pin for UART1. This pin should be set to "input port."
38	P41	D	General-purpose I/O port.
	SCK1		Serial clock I/O pin for UART1. This pin is enabled only when the serial clock I/O setting of the UART1 is "enabled."
39	P42	D	General-purpose I/O port.
	SOT1		Serial data output pin for UART1. This pin functions only when the serial data output setting of the UART1 is "enabled".
40	P43	D	General-purpose I/O port.
	TX		CAN transmission output pin. This pin is enabled only when the output setting is "enabled".
41	P44	D	General-purpose I/O port.
	RX		CAN reception input pin. This pin should be set to "input port."
42 to 45	P30 to P33	D	General-purpose I/O port.
46	X0A*	A	Low-speed oscillation pin.
	P35*	D	General-purpose I/O port.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 1.6-1 Pin Description (3/3)

Pin No. M26	Pin Name	Circuit Type	Function
47	X1A*	A	Low-speed oscillation pin.
	P36*	D	General-purpose I/O port.
48	AV_{SS}	--	V_{SS} power input pin for A/D converter

*: MB90387, MB90F387 : X1A, X0A
MB90387S, MB90F387S : P36, P35

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

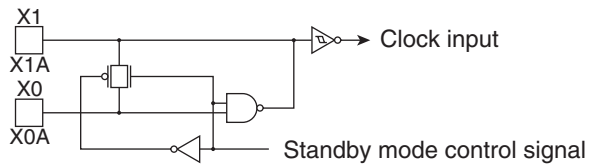
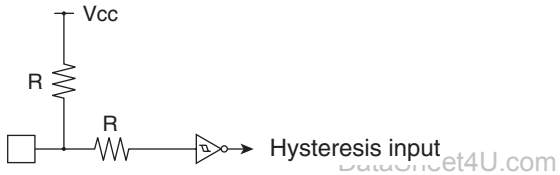
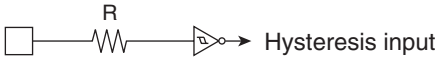
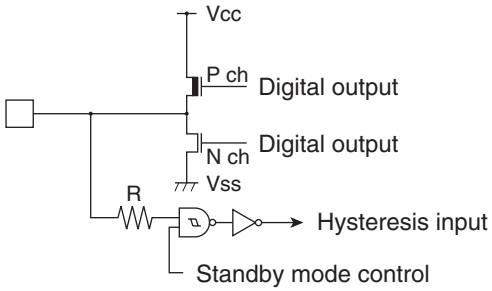
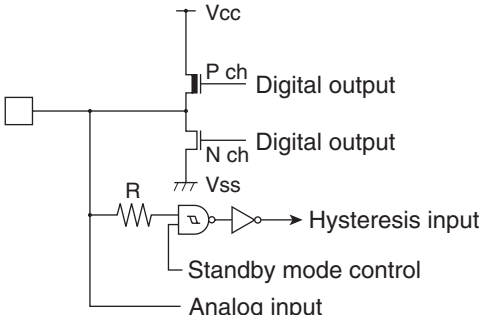
www.DataSheet4U.com

1.7 I/O Circuit

I/O circuit of the MB90385 series is shown in the figure below.

■ I/O Circuit

Table 1.7-1 I/O Circuit (1/2)

Classification	Circuit	Remark
A		• Approximately 1 MΩ high speed oscillation feedback resistor. • Oscillation feedback resistor for low speed approximately 10 MΩ
B		• Hysteresis input with pull-up resistor • Pull-up resistor: about 50kΩ
C		• Hysteresis input
D		• CMOS hysteresis input • CMOS-level output • Standby control provided
E		• CMOS hysteresis input • CMOS-level output • Also used as analog input pin • Standby control provided

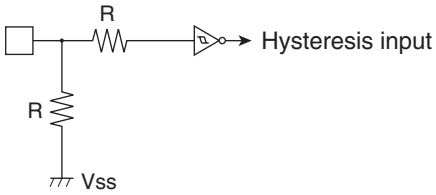
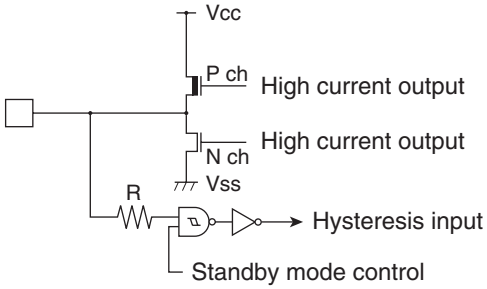
DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 1.7-1 I/O Circuit (2/2)

Classification	Circuit	Remark
F		• Hysteresis input with pull-down resistor • Pull-down resistor: about 50kΩ • There is no pull-down resistor in FLASH product
G		• CMOS hysteresis input • CMOS-level output (for high current output) • Standby control provided

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 2
HANDLING DEVICES

This chapter describes the precautions when handling general-purpose one chip microcontroller.

2.1 Precautions when Handling Devices

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

2.1 Precautions when Handling Devices

This section describes the precautions against the power supply voltage of the device and processing of pin.

■ Precautions when Handling Devices

● Voltage not exceeding maximum ratings (preventing latch-up)

- For a CMOS IC, latch-up may occur when a voltage higher than V_{CC} or a voltage lower than V_{SS} is impressed to the I/O pin other than medium-/high-voltage withstand I/O pins, or when a voltage that exceeds the rated voltage is impressed between V_{CC} and V_{SS}.
- Latch-up may cause a sudden increase in power supply current, resulting in thermal damage to the device. Therefore, the maximum voltage ratings must not be exceeded.
- When turning the analog power supply on and off, the analog power supply voltage (AV_{CC} and AVR) and the analog input voltage should not exceed the digital power supply voltage (V_{CC}).

● Handling not-used pins

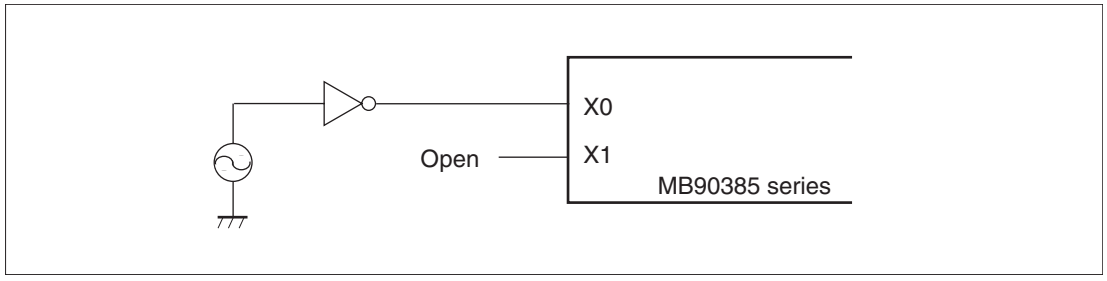
If unused input pins remain open, a malfunction or latch-up may cause permanent damage, so take countermeasures such as pull-up or pull-down using a 2 kΩ or larger resistor.

Leave unused input pins open in the output state or, if left in the input state, treat them in the same manner as for input pins in use.

● Precautions of using external clock

When an external clock is used, drive only the X0 pin and open the X1 pin. Figure 2.1-1 shows an use example of external clock.

Figure 2.1-1 Example of Using External Clock



● Precautions of non-use of subclock

If an oscillator is not connected to the X0A and X1A pins, connect the X0A pin to Pull-down and leave the X1A pin open.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Precautions during operation of PLL clock mode

If the PLL clock mode is selected, the microcontroller attempt to be working with the self-oscillating circuit even when there is no external oscillator or external clock input is stopped. Performance of this operation, however, cannot be guaranteed.

● Power pins

- When plural V_{CC} pins and V_{SS} pins are provided, pins designed to be at the same electric potential are internally connected to the device to prevent malfunctions such as latch-up. However, always connect all same electric potential pins to power supply and ground outside the device to prevent decrease of unnecessary radiation, the malfunction of the strobe signal due to a rise of ground level, and follow the standards of total output current.
- The power pins should be connected to V_{CC} and V_{SS} of the MB90385 series device at the lowest possible impedance from the current supply source.
- It is best to connect approximately 0.1µF capacitor between V_{CC} and V_{SS} as a bypass capacitor near the pins of the MB90385 series device.

● Crystal oscillator circuit

- Noise near the X0 and X1 pins may cause the MB90385 series to malfunction. Design the PC board so that the X0 and X1 pins, the crystal (or ceramic) oscillator, and the bypass capacitor to ground are as close as possible to each other, and so the wiring of the X0 and X1 pins and other wiring do not cross.
- For stable operation, the PC board is recommended to have the artwork with the X0 and X1 pins enclosed by a ground line.

et4U.com DataShee DataSheet4U.com

● Procedure of A/D converter/analog input power-on

- Always apply a power to the A/D converter power and the analog input (AN0 to AN7 pins) after or concurrently with the digital power (V_{CC})-on.
- Always turn off the A/D converter power and the analog input before or concurrently with the digital power-down.
- Note that AVR should not exceed AV_{CC} at turn on or off. (The analog power and digital power can be simultaneously turned on or off with no problem.)

● Handling pins when not using A/D converter

When not using the A/D converter, the pins should be connected so that AV_{CC} = AVR = V_{CC} and AV_{SS} = V_{SS}.

● Precautions at power on

To prevent a malfunction of the internal step-down circuit, the voltage rise time at power-on should be 50 µs or more (between 0.2 V and 2.7 V).

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Stabilization of supply voltage

If the power supply voltage varies acutely even within the operation assurance range of the V_{CC} power supply voltage, a malfunction may occur. The V_{CC} power supply voltage must therefore be stabilized.

As stabilization guidelines, stabilize the power supply voltage so that V_{CC} ripple fluctuations (peak to peak value) in the commercial frequencies (50 Hz to 60 Hz) fall within 10% of the standard V_{CC} power supply voltage and the transient fluctuation rate becomes 0.1V/ms or less in instantaneous fluctuation for power supply switching.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 3
CPU

This chapter explains the CPU function of the MB90385 series.

3.1 Memory Space
3.2 Dedicated Registers
3.3 General-purpose Register
3.4 Prefix Codes
3.5 Interrupt
3.6 Reset
3.7 Clocks
3.8 Low-power Consumption Mode
3.9 CPU Mode

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.1 Memory Space

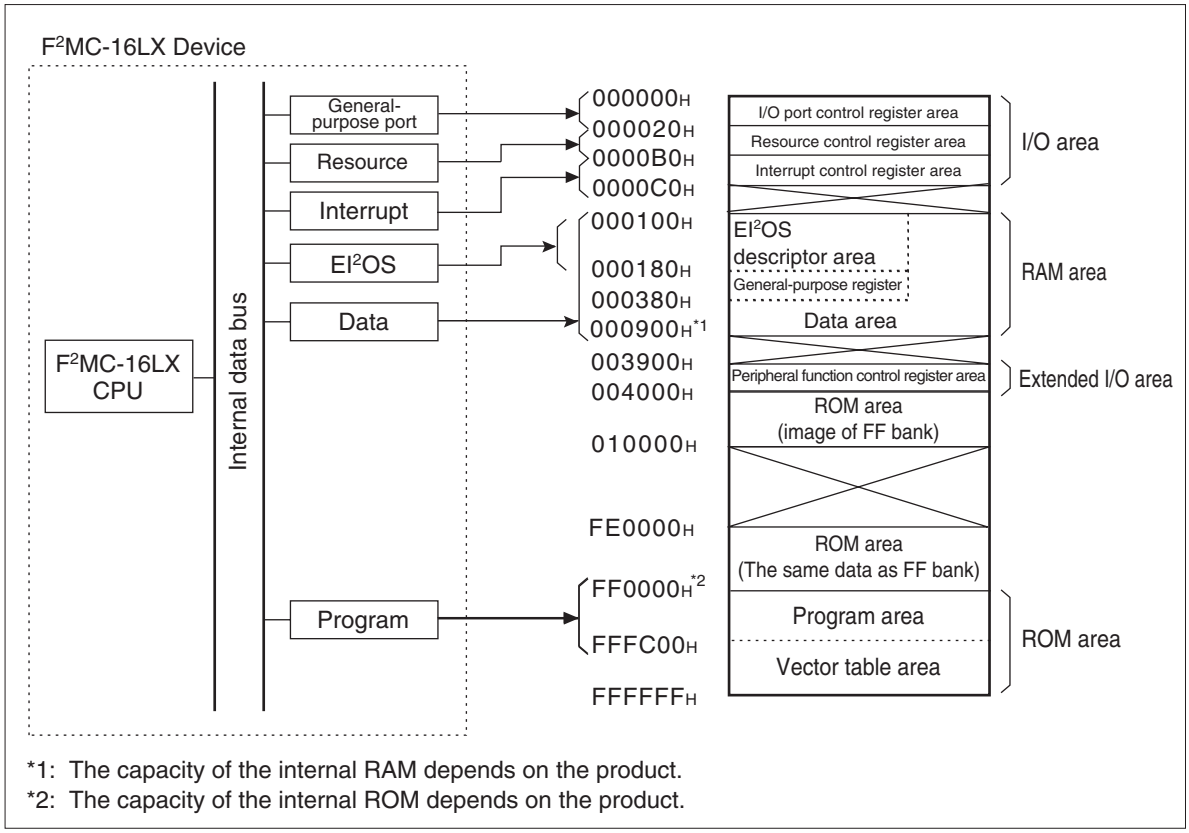
The memory space of the F²MC-16LX is 16 MB and is allocated to I/O, programs, and data. Part of the memory space is used for specific uses such as the expansion intelligent I/O service (EI²OS) descriptors, the general-purpose registers, and the vector tables.

■ Memory Space

I/O, programs and data are all allocated somewhere in the 16-MB memory space of the F²MC-16LX CPU. The CPU can indicate their addresses in the 24-bit address bus to access each resource.

Figure 3.1-1 shows an example of the relationships between the F²MC-16LX and the memory map.

Figure 3.1-1 Example of Relationships between F²MC-16LX System and Memory Map



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ ROM Area

- Vector table area (address: "FFFC00$_H$" to "FFFFFF$_H$")
  - The vector table is provided for reset and interrupts.
  - This area is allocated at the top of the ROM area. The starting address of the corresponding processing routine is set to the address of each vector table as data.
- Program area (address: to "FFFBFF$_H$")
  - ROM is contained as the internal program area.
  - The capacity of the internal ROM depends on the product.

■ RAM Area

- Data area (address: "000100$_H$" to "000900$_H$")
  - Static RAM is contained as the internal data area.
  - The capacity of the internal RAM depends on the product.
- General-purpose register area (address: "000180$_H$" to "00037F$_H$")
  - Auxiliary registers for operations or transfer of the 8-bit, 16-bit, or 32-bit data are allocated in this area.
  - This area is allocated to part of the RAM area, and can also be used as ordinary RAM.
  - When this area is used as general-purpose registers, they can be accessed quickly using a short instruction through general-purpose register addressing.
- Expanded intelligent I/O service (EI^{2}OS) descriptor area (address: "000100$_H$" to "00017F$_H$")
  - This area holds the transfer mode, I/O address, transfer count, and buffer address.
  - This area is allocated to part of the RAM area, and can also be used as ordinary RAM.

■ I/O Area

- Interrupt control register area (address: "0000B0$_H$" to "0000BF$_H$")

  The interrupt control registers (ICR00 to ICR15) correspond to all resources with an interrupt function, and control the setting of interrupt level and EI^{2}OS.
- Resource control register area (address: "000020$_H$" to "0000AF$_H$")

  This area controls the resource function and data I/O.
- I/O port control register area (address: "000000$_H$" to "00001F$_H$")

  This area controls the I/O ports and data I/O.

■ Extended I/O Area

- Peripheral function control register area (address: 003900$_H$ to 003FFF$_H$)

  The registers control peripheral functions and input/output data.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.1.1 Mapping of and Access to Memory Space

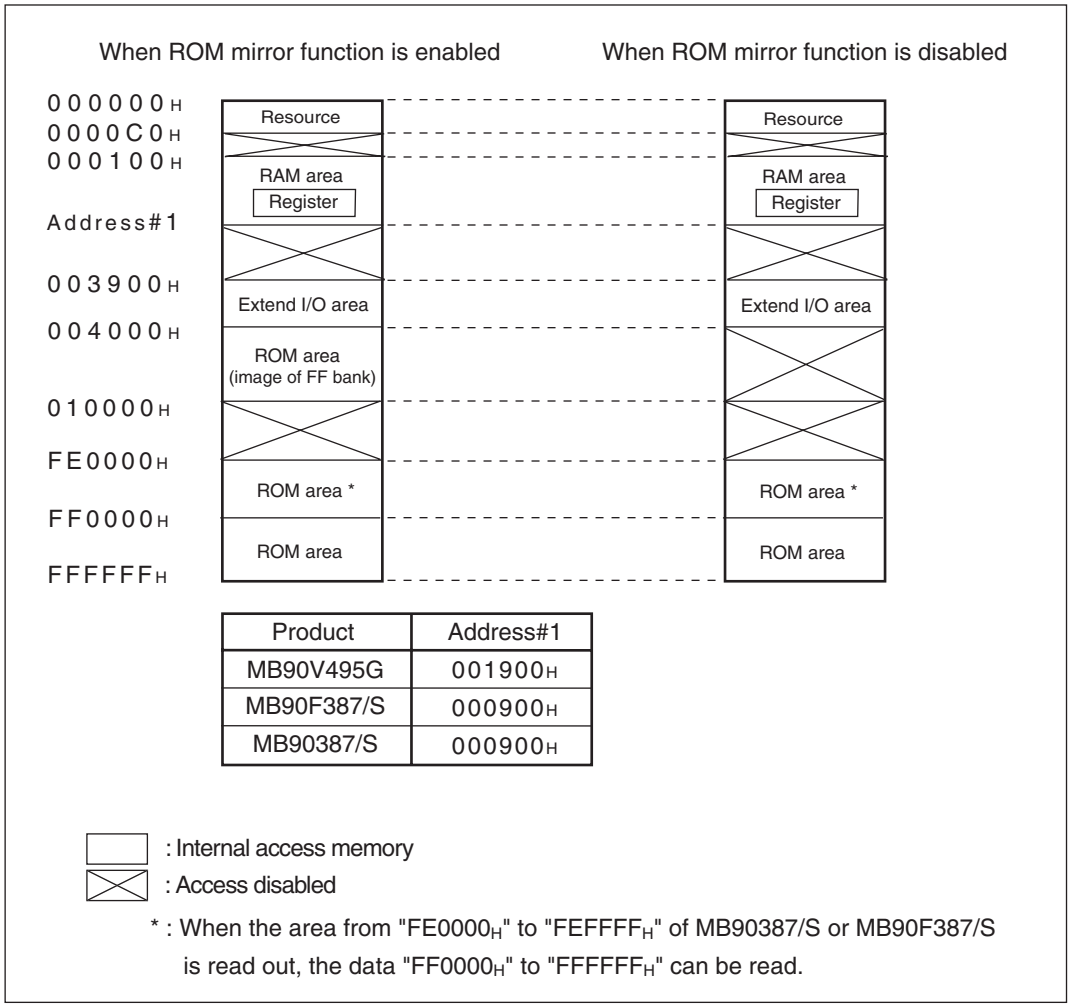
In the MB90385 series, the single-chip mode can be set as memory access modes.

■ Memory Map for MB90385 Series

In the MB90385 series, the internal address bus is output up to a width of 24 bits and the external address bus is output up to a width of 24 bits; the external access memory can access up to the 16-MB memory space.

Figure 3.1-2 shows the memory map when the ROM mirroring function is enabled and disabled.

Figure 3.1-2 Memory Map for MB90385 Series



Product	Address#1
MB90V495G	001900H
MB90F387/S	000900H
MB90387/S	000900H

: Internal access memory

: Access disabled

* : When the area from "$FE0000_H$" to "$FEFFFF_H$" of MB90387/S or MB90F387/S is read out, the data "$FF0000_H$" to "$FFFFFF_H$" can be read.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Image Access to Internal ROM

In the F^{2}MC-16LX family, with the internal ROM in operation, ROM data in the FF bank can be seen as an image in the top 00 bank. This function is called ROM mirroring and enables effective use of a small C compiler.

In the F^{2}MC-16LX family, the lower 16-bit addresses of the FF bank are the same as the lower 16-bit addresses of the 00 bank, so the table in ROM can be referenced without specifying "far" with a pointer.

For example, if "$00C000_H$" is accessed, data in ROM at "$FFC000_H$" is actually accessed. However, the ROM area in the FF bank exceeds 48 KB and all areas cannot be seen as images in the 00 bank. Therefore, ROM data from "$FF4000_H$" to "$FFFFFF_H$" is see as an image from "004000_H" to "$00FFFF_H$" so the ROM data table should be stored in the area from "$FF4000_H$" to "$FFFFF_H$".

Reference:	To disable the ROM mirroring function (ROMM: MI = 0), see Section "17.1 Overview of ROM Mirroring Function Select Module".

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

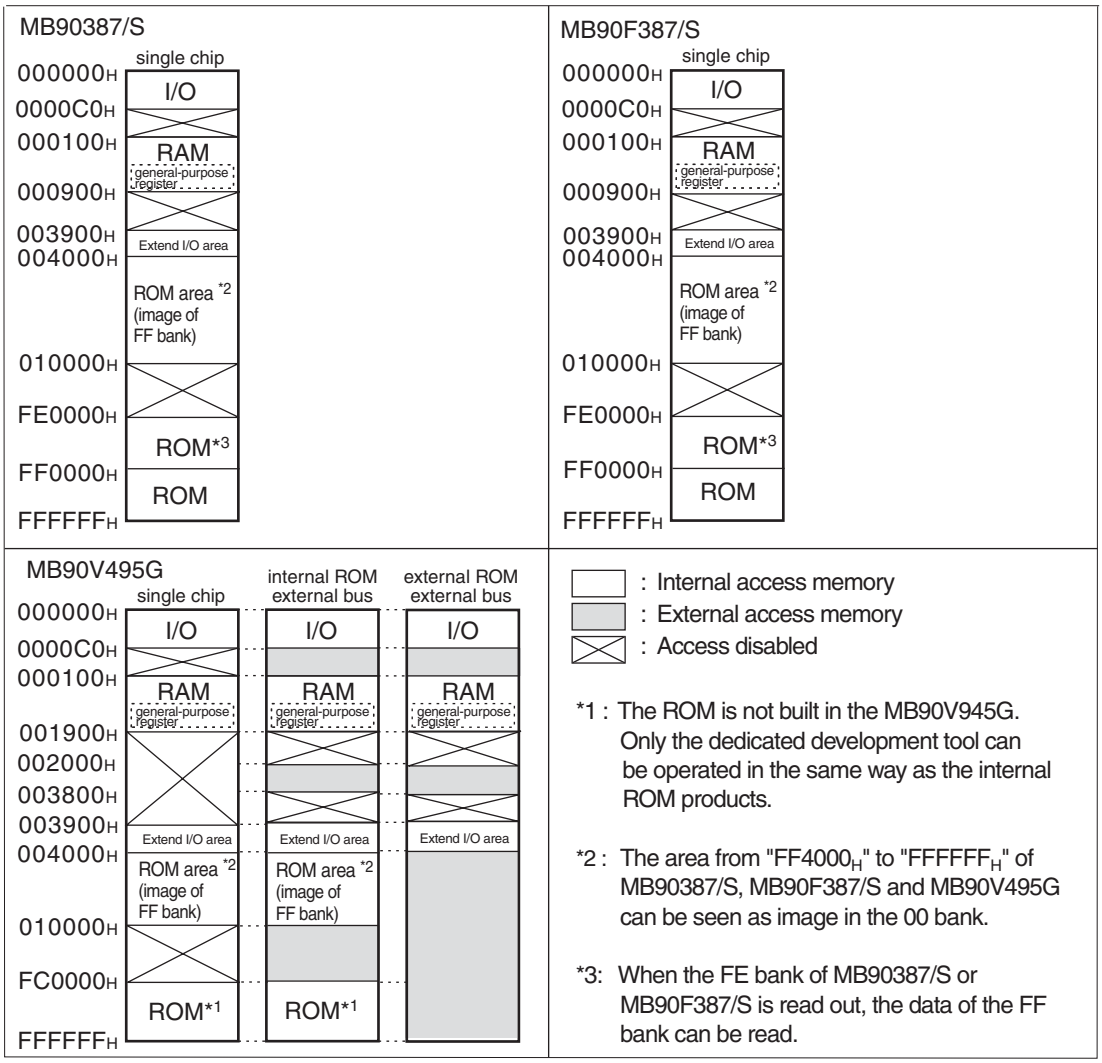
3.1.2 Memory Map

The MB90385 series memory map is shown for each product.

■ Memory Map

Figure 3.1-3 shows the memory map for the MB90385 series.

Figure 3.1-3 Memory Map for MB90385 Series



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.1.3 Addressing

Linear and bank types are available for addressing.

The F^{2}MC-16LX family uses basically bank addressing.

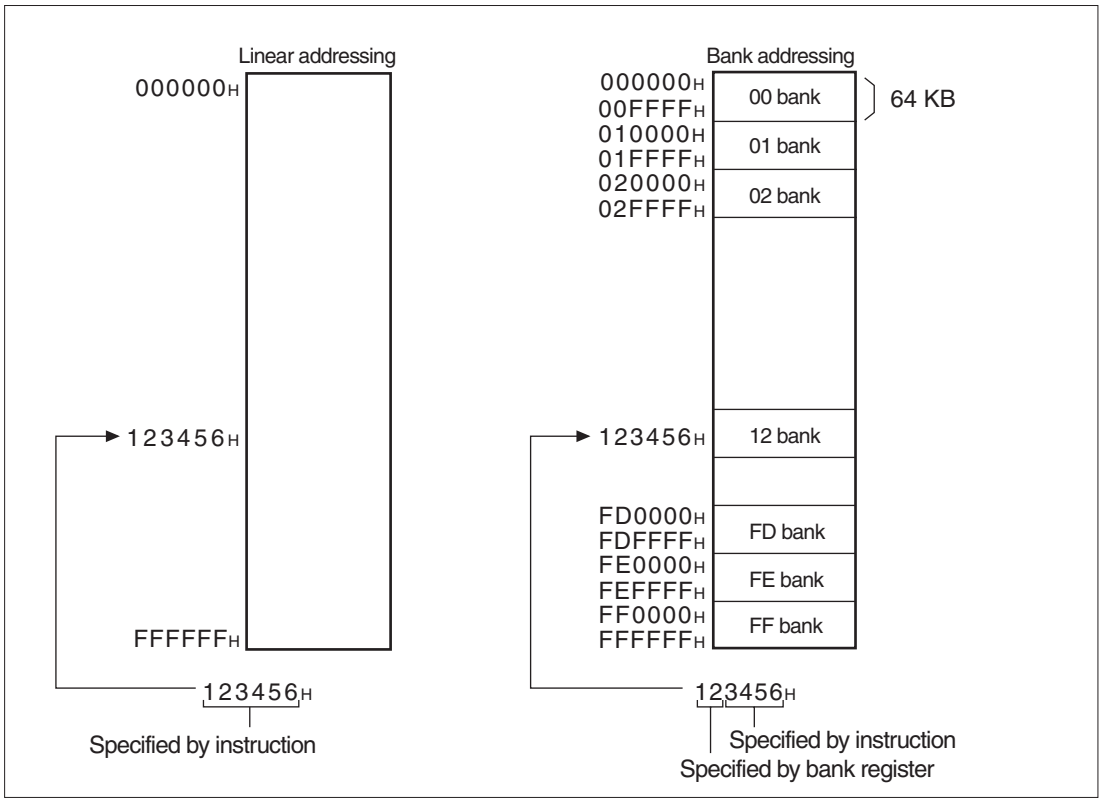
- **Linear type: direct-addressing all 24 bits by instruction**
- **Bank type: addressing higher 8 bits by bank registers suitable for the use, and lower 16 bits by instruction**

■ Linear Addressing and Bank Addressing

The linear addressing is to access the 16-MB memory space by direct-addressing. The bank addressing is to access the 16-MB memory space which divided into 256 banks of 64KB, by specifying banks and addresses in banks.

Figure 3.1-4 shows overview of memory management in linear and bank type.

Figure 3.1-4 Memory Management in Linear and Bank Types



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

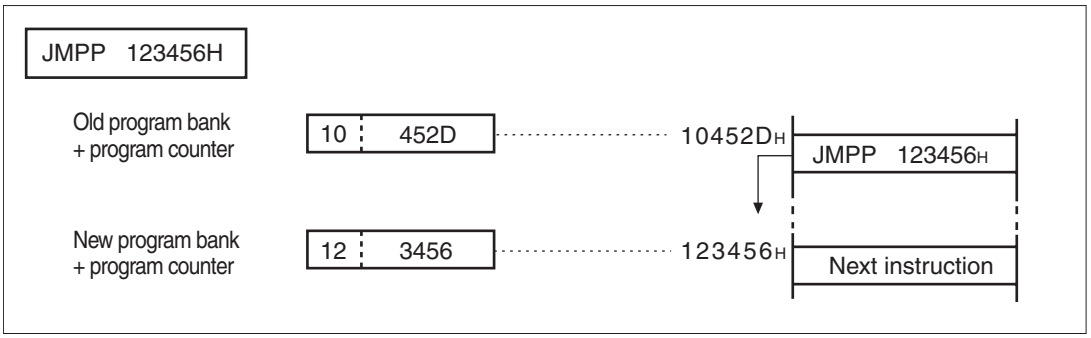
3.1.4 Linear Addressing

The linear addressing has the following two types:

- **Direct-addressing 24 bits by instruction**
- **Using lower 24 bits of 32-bit general-purpose register for address**

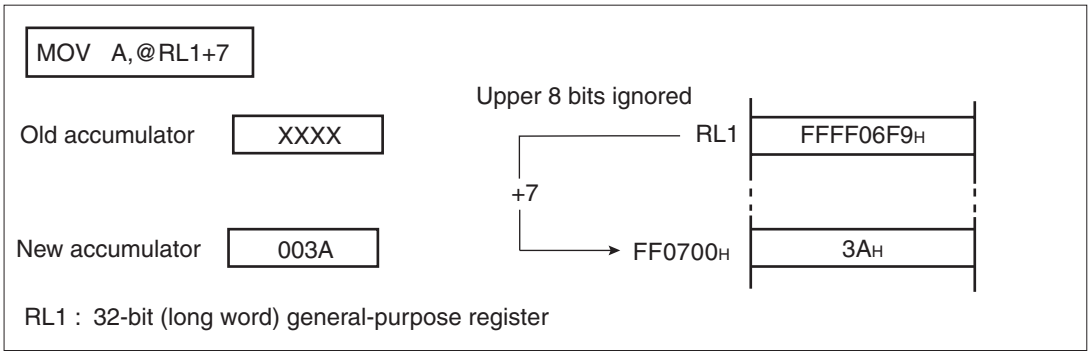
■ Linear Addressing by Specifying 24-bit Operand

Figure 3.1-5 Example of 24-bit Physical Direct Addressing in Linear Type



■ Addressing by Indirect-specifying 32-bit Register

Figure 3.1-6 Example of indirect-specifying 32-bit General-purpose Register in Linear Type



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.1.5 Bank Addressing

The bank addressing is a type of addressing each of 256 banks of 64-KB into which the 16-MB memory space is divided, using the bank register, and the lower 16 bits by an instruction.
Bank register has the following five types depending on the use.

- **Program bank register (PCB)**
- **Data bank register (DTB)**
- **User stack bank register (USB)**
- **System stack bank register (SSB)**
- **Additional bank register (ADB)**

■ Bank Registers and Access Space

Table 3.1-1 shows the access space for each bank register and the major use of it.

Table 3.1-1 Access Space for Each Bank Register and Major Use of Access Space

Bank Register Name	Access Space	Major Use	Reset Value
Program bank register (PCB)	Program (PC) space	Stores instruction code, vector tables, immediate data.	FF_H
Data bank register (DTB)	Data (DT) space	Stores data that can be read/written and can access resource control registers and data registers.	00_H
User stack bank register (USB)	Stack (SP) space	These are used for the stack accessing such as the PUSH/POP instruction and the register saving at an interrupt. When the stack flag (CCR: S) is 1, SSB is used. When the stack flag is 0, USB is used*.	00_H
System stack bank register (SSB)*			00_H
Additional bank register (ADB)	Additional (AD) space	Stores data that cannot be stored in data (DT) space.	00_H

*: SSB is always used for the stack at an interrupt.

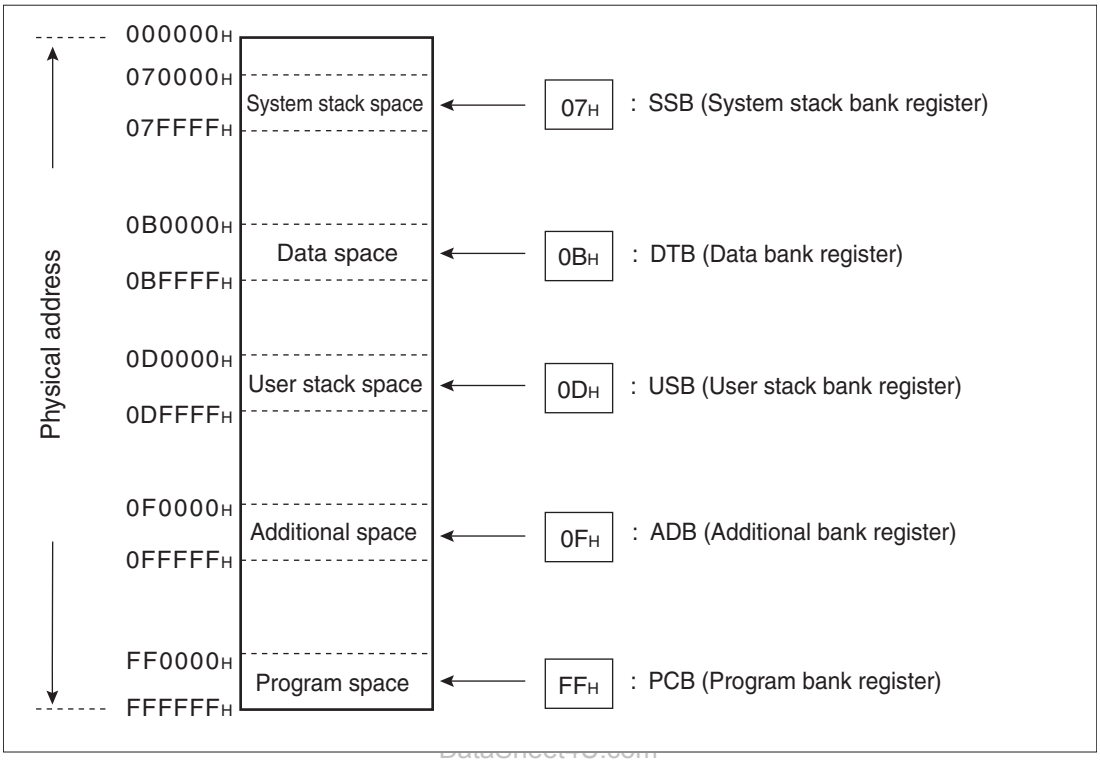
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 3.1-7 shows the relationships between the memory space divided into banks and each register.

Figure 3.1-7 Example of Bank Addressing



Reference: For details, see Section "3.2 Dedicated Registers".

■ Bank Addressing and Default Space

To improve the instruction code efficiency, the default space shown in Table 3.1-2 is determined for each instruction in each addressing type. To use any bank space other than the default space, specify the prefix code for that bank space before the instruction, which makes the arbitrary bank space corresponding to the prefix code accessible.

Table 3.1-2 Addressing and Default Spaces

Default Spaces	Addressing
Program space	PC indirect addressing, program-access addressing, branch instruction addressing
Data space	Addressing with @RW0, @RW1, @RW4, @RW5, @A, addr16, and dir
Stack space	Addressing with PUSHW, POPW, @RW3, and @RW7
Additional space	Addressing with @RW2 and @RW6

Reference: For details of the prefix codes, see Section "3.4 Prefix Codes".

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

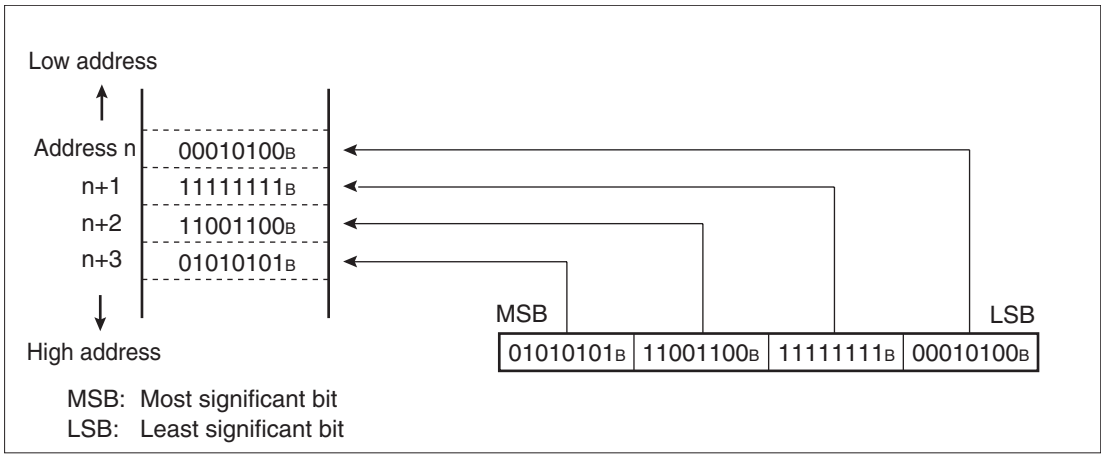
3.1.6 Allocation of Multi-byte Data on Memory

Multi-byte data is written to memory in sequence starting from the low address. For 32-bit length data, the lower 16 bits are written first, and then the higher 16 bits are written. If a reset signal is output immediately after the lower 16 bits is written, the higher 16 bit may not be written.

■ Store of Multi-byte Data in RAM

Figure 3.1-8 shows the order in which multi-byte data is stored. Lower 8 bits are allocated to n address, and in order of n+1, n+2, n+3 and so on.

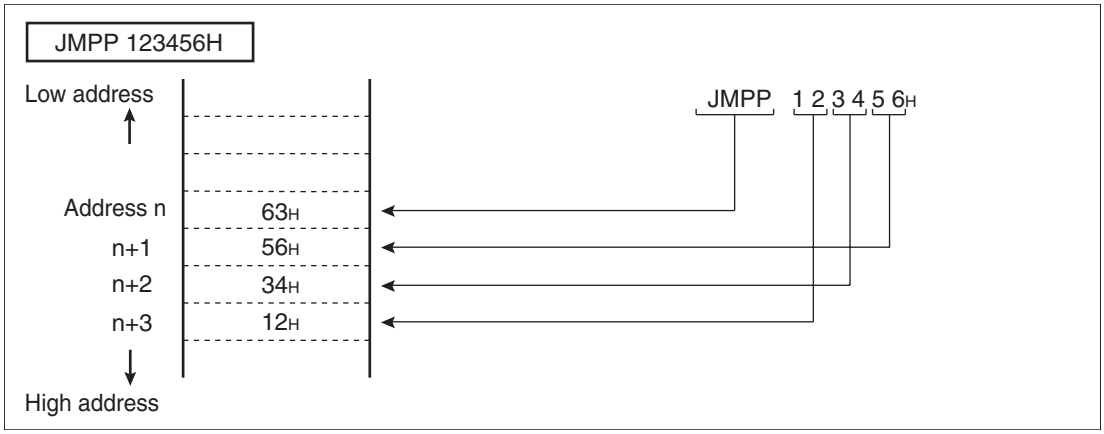
Figure 3.1-8 Storage of Multi-byte Data in RAM



■ Storage of Multi-byte Length Operand

Figure 3.1-9 shows the configuration of a multi-byte length operand in memory.

Figure 3.1-9 Storage of Multi-byte Operand



DataSheet4U.com
www.DataSheet4U.com

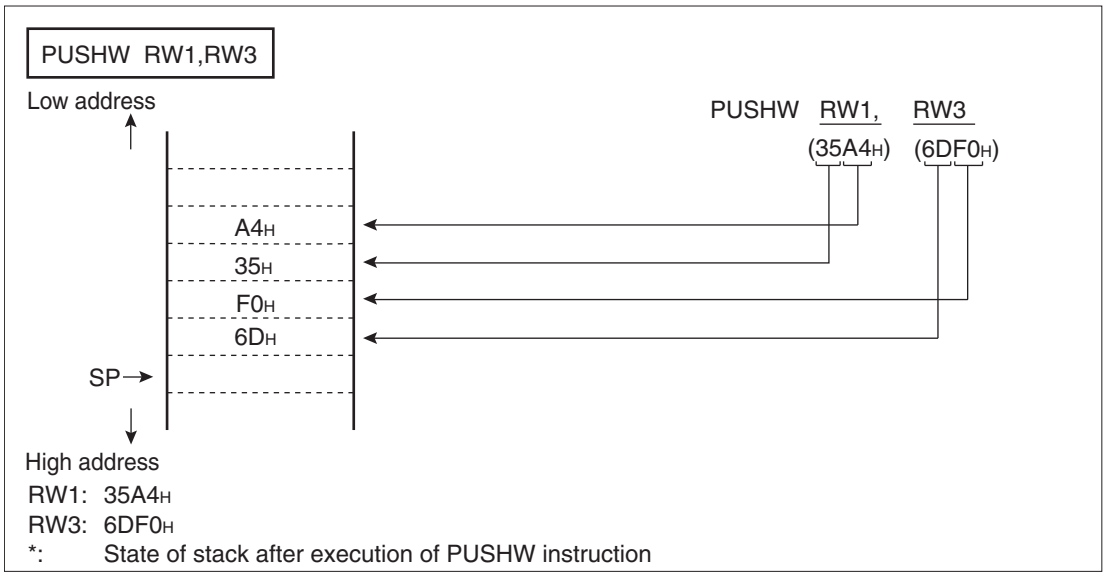
DataSheet4U.com

www.DataSheet4U.com

■ Storage of Multi-byte Data in Stack

Figure 3.1-10 shows the order in which multi-byte data is stored in the stack.

Figure 3.1-10 Storage of Multi-byte Data in Stack

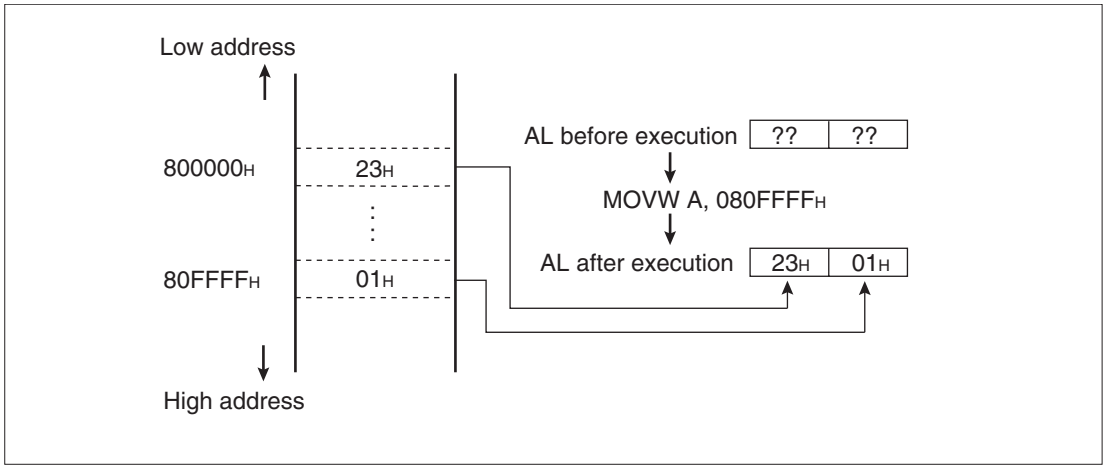


■ Access to Multi-byte Data

All accesses are basically made inside the bank. Consequently, for an instruction that accesses multi-byte data, the address after the "$FFFF_H$" address is the "0000_H" address of the same bank.

Figure 3.1-11 shows an example of access instruction for multi-byte data on the bank boundary.

Figure 3.1-11 Access to Multi-byte Data on Bank Boundary



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

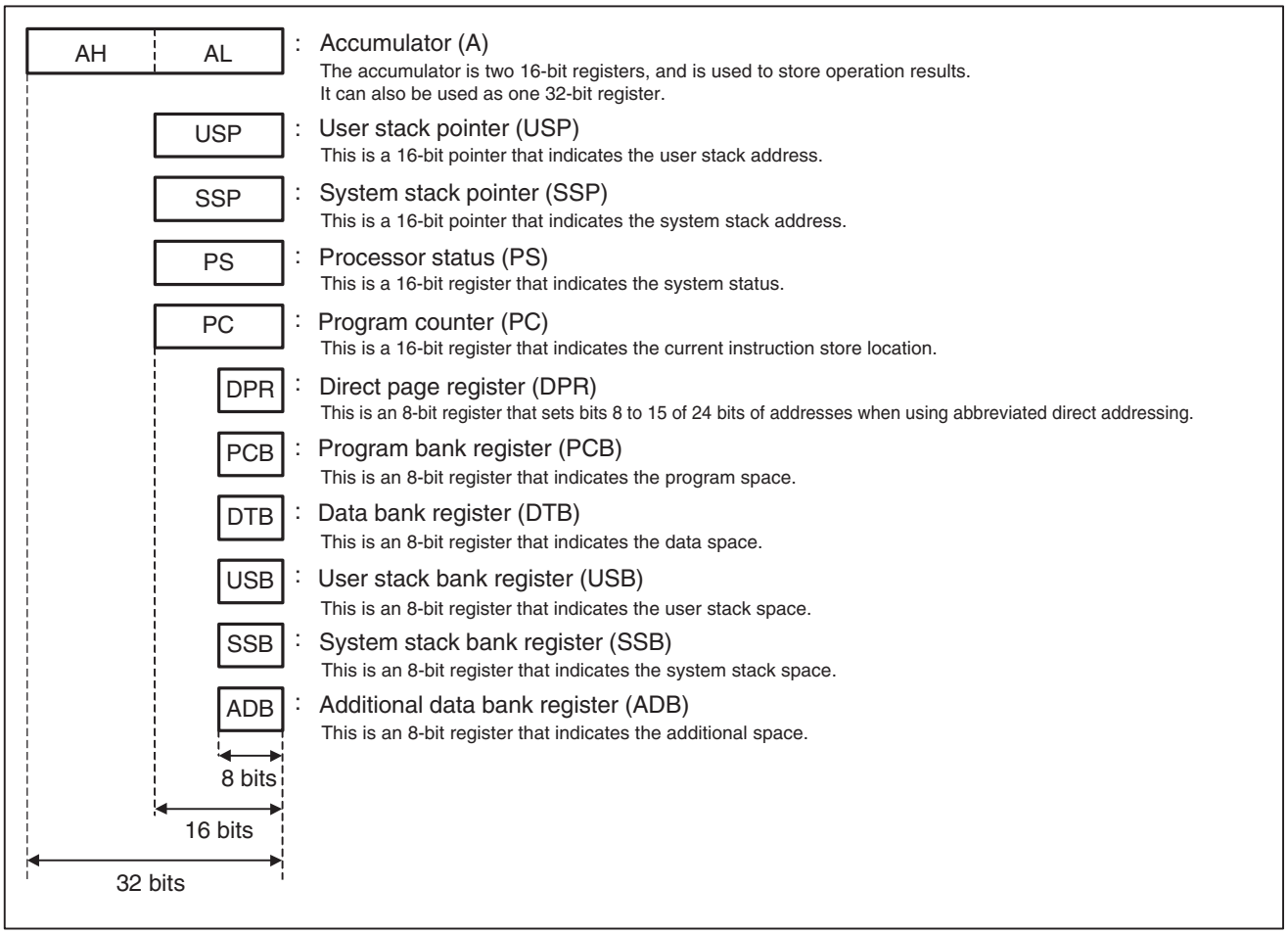
3.2 Dedicated Registers

The CPU has the following dedicated registers.

- **Accumulator**
- **User stack pointer**
- **System stack pointer**
- **Processor status**
- **Program counter**
- **Direct page register**
- **Bank registers (program bank register, data bank register, user stack bank register, system stack bank register, additional data bank register)**

■ Configuration of Dedicated Registers

Figure 3.2-1 Configuration of Dedicated Registers

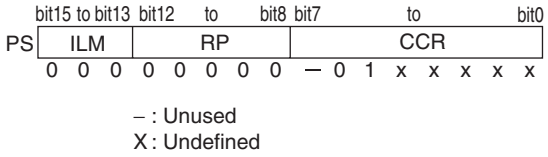


DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 3.2-1 Reset Values of Dedicated Registers

Dedicated Register	Reset Value
Accumulator (A)	Undefined
User stack pointer (USP)	Undefined
System stack pointer (SSP)	Undefined
Processor status (PS)	
Program counter (PC)	Value of reset vector (data at "FFFFDC$_H$" and "FFFFDD$_H$")
Direct page register (DPR)	01_H
Program bank register (PCB)	Value of reset vector (data at "FFFFDE$_H$")
Data bank register (DTB)	00_H
User stack bank register (USB)	00_H
System stack bank register (SSB)	00_H
Additional data bank register (ADB)	00_H

Note: The above reset values are the reset values for the device. The reset values for the ICE (such as emulator) are different from those of the device.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.2.1 Dedicated Registers and General-purpose Register

The F^{2}MC-16LX family has two types of registers: dedicated registers in the CPU and general-purpose register in the internal RAM.

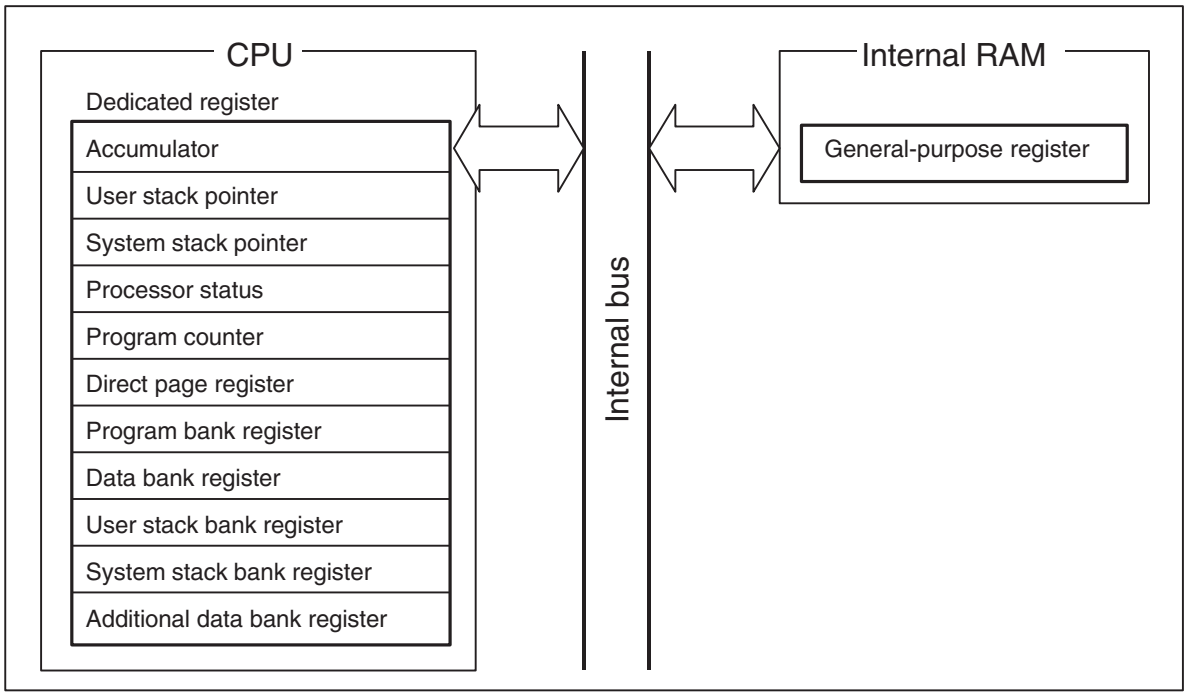
■ Dedicated Registers and General-purpose Register

The dedicated registers are limited to the use in the hardware architecture of the CPU.

The general-purpose registers are in the internal RAM in the CPU address space. As with the dedicated registers, these registers can be used for addressing and the use of these register is not limited.

Figure 3.2-2 shows the allocation of the dedicated registers and the general-purpose registers.

Figure 3.2-2 Dedicated Registers and General-purpose Register



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.2.2 Accumulator (A)

An accumulator (A) consists of two 16-bit length operation registers (AH and AL) used for temporary storage of the operation result or data.
Accumulator can be used as a 32-, 16-, or 8-bit register. Various operations can be performed between the register and memory or the other register, or between the AH register and the AL register.

■ Accumulator (A)

● Data transfer to accumulator

The accumulator can process 32-bit data (long word), 16-bit data (word), and 8-bit data (byte).

- When processing 32-bit data, the AH register and the AL register are concatenated and used.
- When processing 16- or 8-bit data, only the AL register is used.

Data retention function

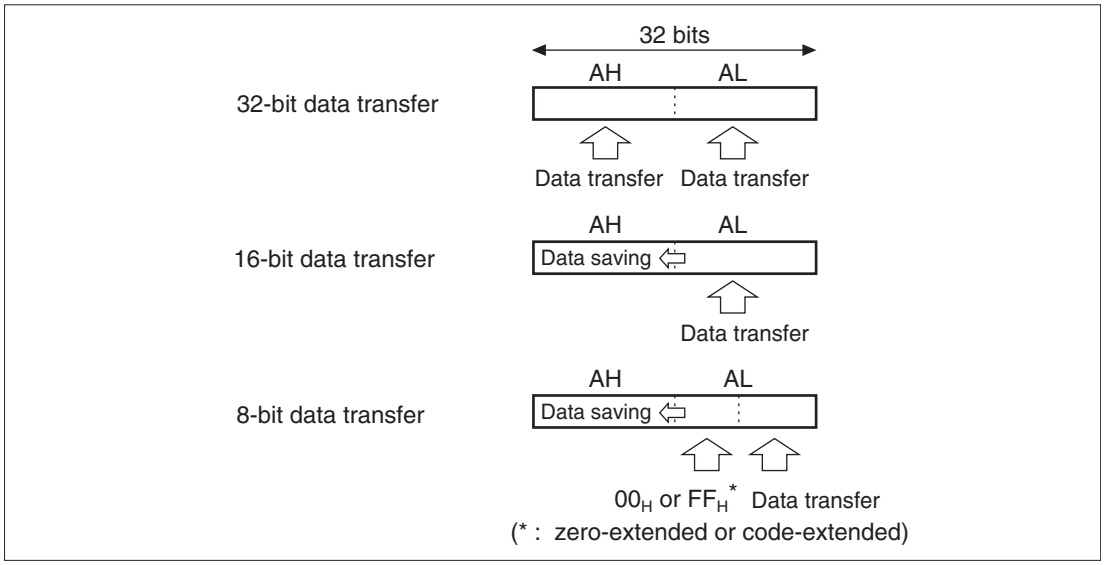
When data of word length or less is transferred to the AL register, data stored in the AL register is transferred automatically to the AH register.

Code-extended function and zero-extended function

When transferring data of byte length or less to the AL register, the data is code-extended (MOVX instruction) or zero-extended (MOVX instruction) to be the 16-bit length and stored in the AL register. Data in the AL register can also be treated in word and byte lengths.

Figure 3.2-3 shows data transfer to the accumulator and a concrete example.

Figure 3.2-3 Data Transfer to Accumulator



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

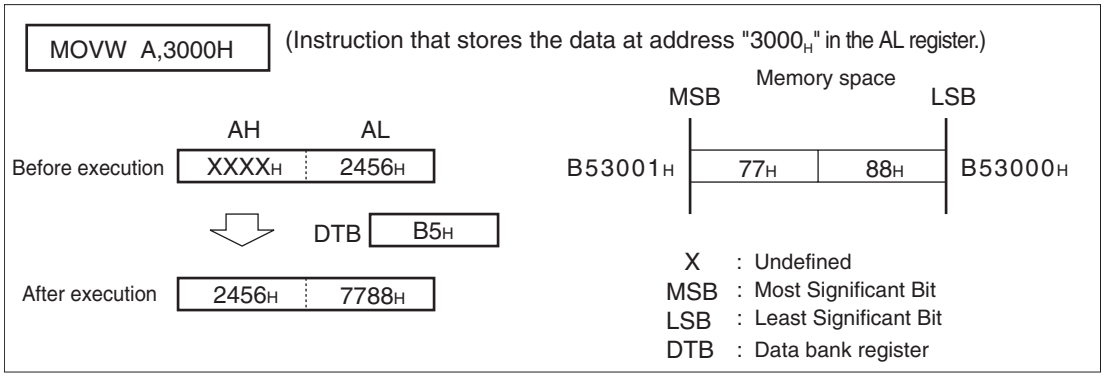
● Byte processing arithmetic operation of accumulator

When the arithmetic operation instruction for byte processing is executed for the AL register, the higher 8 bits of the AL register in pre-operation are ignored, and the higher 8 bits of the operation result become all 0.

● Reset value of accumulator

The reset value is undefined.

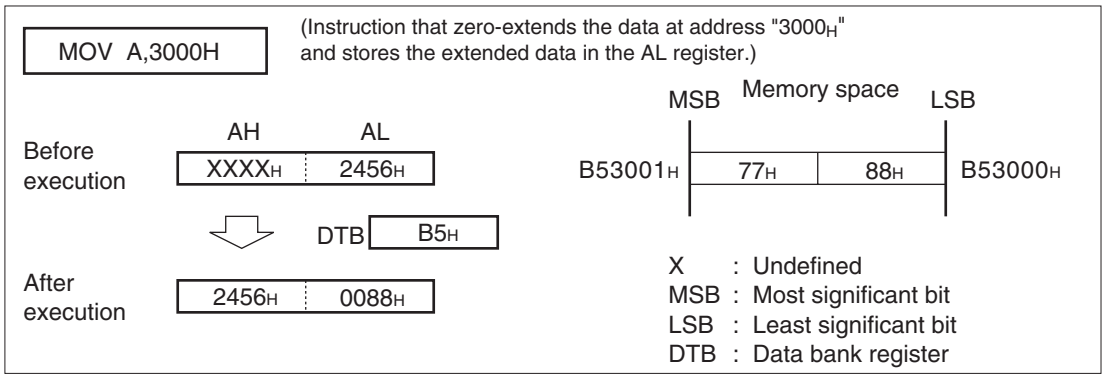
Figure 3.2-4 Example of 8-bit Data Transfer to Accumulator (A) (Data Saving)



et4U.com
DataShee

Figure 3.2-5 Example of 8-bit Data Transfer to Accumulator (A) (Data Saving, Zero-extended)

DataSheet4U.com



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 3.2-6 Example of 16-bit Data Transfer to Accumulator (A) (Data Saving)

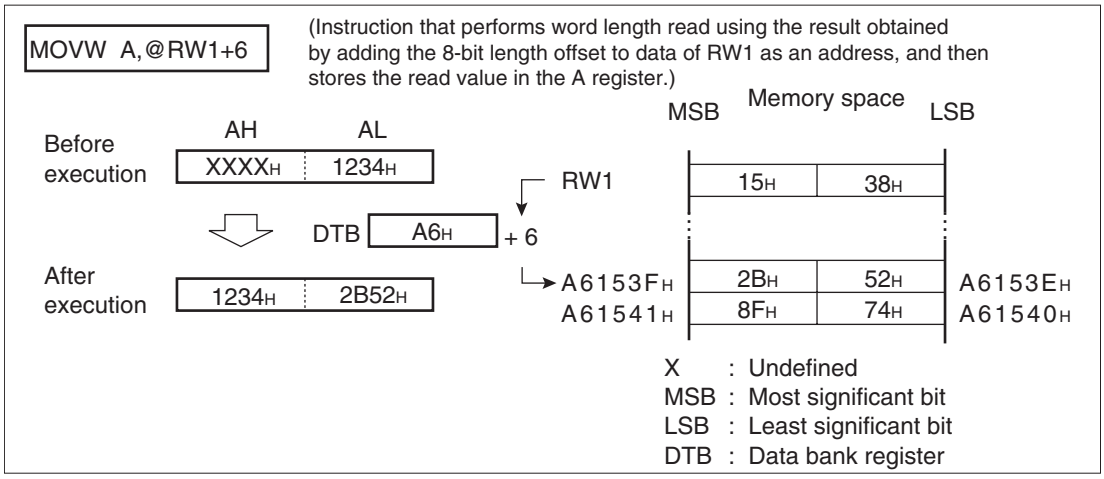
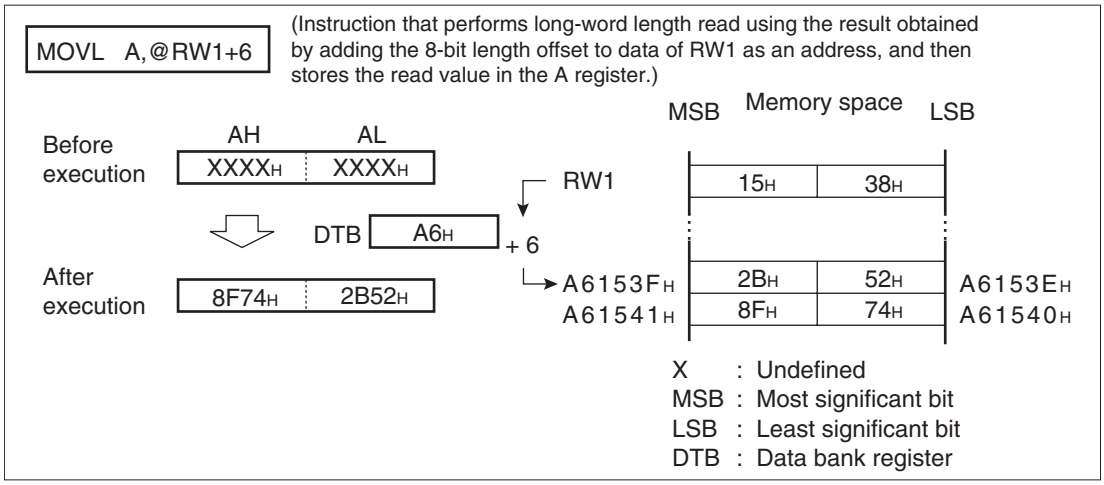


Figure 3.2-7 Example of 32-bit Data Transfer to Accumulator (A) (Register Indirect)



DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.2.3 Stack Pointer (USP, SSP)

The stack pointers include a user stack pointer (USP) and a system stack pointer (SSP). Both these pointers indicate the address where saved data and return data are stored when the PUSH instruction, the POP instruction, and the subroutine is executed.

- **The higher 8 bits of the stack address are set by the user stack bank register (USB) or the system stack bank register (SSB).**
- **When the stack flag (PS: CCR: S) is 0, the USP and USB register are enabled. When the stack flag is 1, the SSP and SSB register are enabled.**

■ Stack Selection

For the F^{2}MC-16LX family, two types of stack pointer can be used: system stack, and user stack.

The addresses of the stack pointers are set by the stack flag of the condition code register (CCR: S) as shown in Table 3.2-2.

Table 3.2-2 Stack Address Specification

S Flag	Stack Address	
	Higher 8 Bits	Lower 16 Bits
0	User stack bank register (USB)	User stack pointer (USP)
1*	System stack bank register (SSB)	System stack pointer (SSP)

*: Reset value

Since the stack flag (CCR: S) is set to 1 by a reset, the system stack pointer is used after reset.

Ordinarily, the system stack pointer is used in processing the stack at the interrupt routine, and the user stack pointer is used in processing the stack other than interrupt routine. When it is not necessary to divide the stack space, use only the system stack pointer.

Note: When an interrupt is accepted, the stack flag (CCR: S) is set and the system stack pointer is always used.

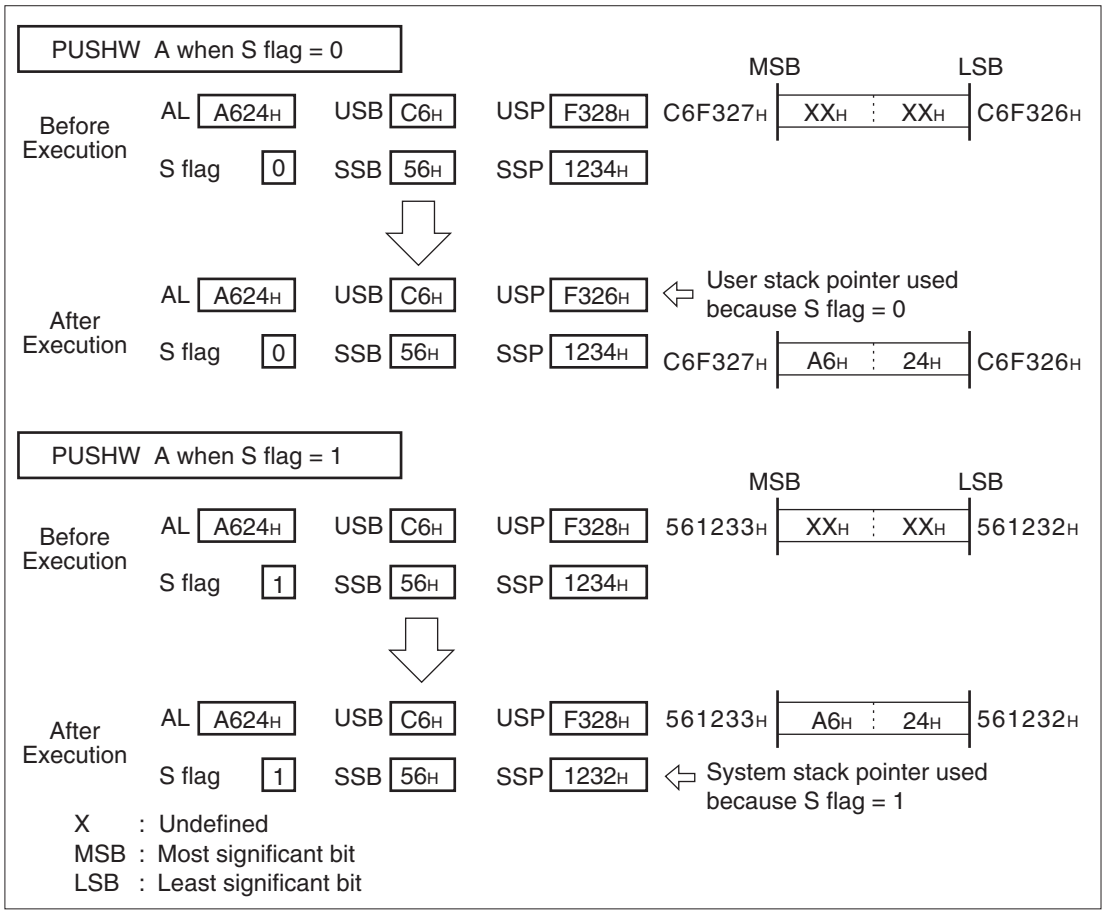
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 3.2-8 shows an example of the stack operation using the system stack.

Figure 3.2-8 Stack Operation Instructions and Stack Pointers



Notes:
- Use even addresses for setting value to the stack pointer. Setting an odd address divides the word access into two accesses, decreasing the efficiency.
- The reset values of the USP and SSP registers are undefined.

■ System Stack Pointer (SSP)

When using the system stack pointer (SSP), the stack flag (CCR: S) is set to 1. The higher 8 bits of the address used in processing the stack are set by the system stack bank register (SSB).

■ User Stack Pointer (USP)

When using the user stack pointer (USP), the stack flag (CCR: S) is set to 0. The higher 8 bits of the address used in processing the stack are set by the user stack bank register (USB).

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Stack Area

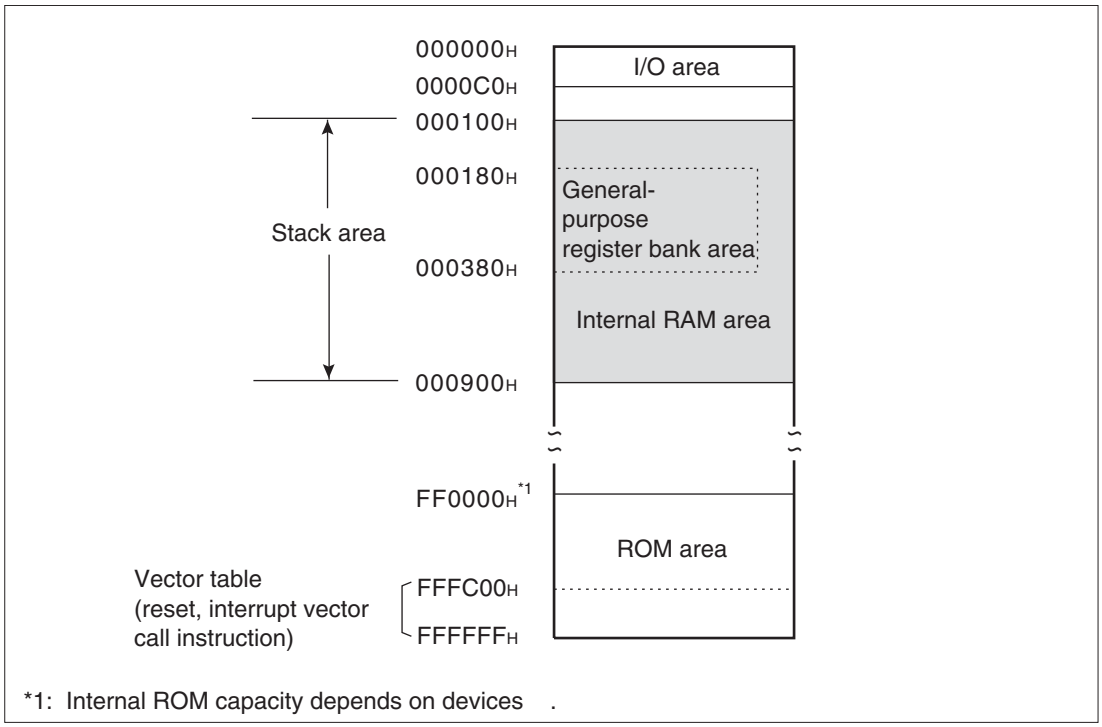
● Securing stack area

The stack area is used to save and return the program counter (PC) at execution of the interrupt processing, subroutine call instruction (CALL) and vector call instruction (CALLV). It is also used to save and return temporary registers using the PUSHW and POPW instructions.

The stack area is secured with the data area in RAM.

The stack area is as shown below:

Figure 3.2-9 Stack Area



Notes:
- As a general rule, even addresses should be set in the stack pointers (SSP and USP).
- The system stack area, user stack area, and data area should not overlap.

● System stack area and user stack area

The system stack area is used for interrupt processing. When an interrupt occurs, even if the user stack area is used, it is switched forcibly to the system stack area. Therefore, in systems mainly using the user stack area also, the system stack area must be set correctly.

In particular, only the system stack area should be used unless it is necessary to divide the stack space.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.2.4 Processor Status (PS)

The processor status (PS) consists of the bits controlling CPU and various bits indicating the CPU status. The PS consists of the following three registers.

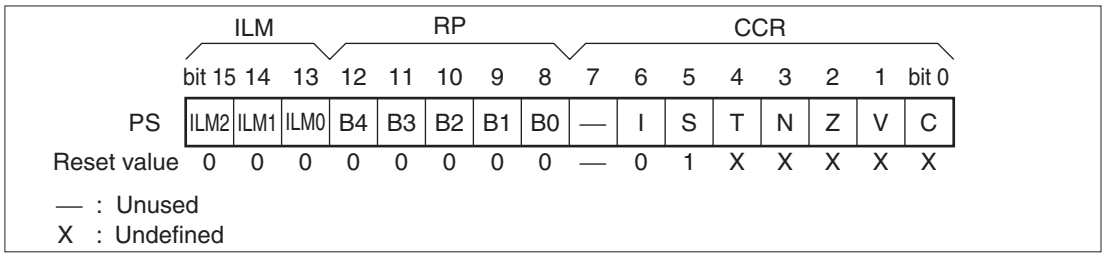
- **Interrupt level mask register (ILM)**
- **Register bank pointer (RP)**
- **Condition code register (CCR)**

■ Configuration of Processor Status (PS)

The processor status (PS) consists of bits controlling CPU and various bits indicating the CPU status.

Figure 3.2-10 shows the configuration of the processor status (PS).

Figure 3.2-10 Processor Status (PS)



DataSheet4U.com

● Interrupt level mask register (ILM)

This register indicates the level of the interrupt that the CPU is currently accepting. The value of this register is compared to the value of the interrupt level setting bits of the interrupt control register (ICR: IL0 to IL2) corresponding to the interrupt request of each resource.

● Register bank pointer (RP)

This register set the memory block (register bank) to be used for the general-purpose registers allocated in the internal RAM.

General-purpose registers can be set for up to 32 banks. The general-purpose register banks to be used are set by setting 0 to 31 in the register bank pointer (RP).

● Condition code register (CCR)

This register consists of various flags that are set (1) or cleared (0) by instruction execution result or acceptance of an interrupt.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

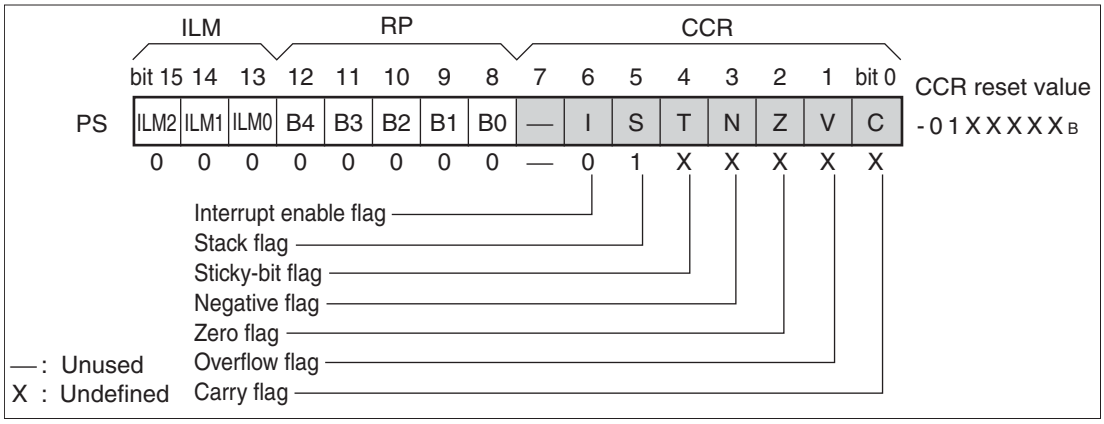
3.2.4.1 Condition Code Register (PS: CCR)

The condition code register (CCR) is an 8-bit register consisting of bits indicating the result of instruction execution, and the bits enabling or disabling the interrupt request.

■ Configuration of Condition Code Register (CCR)

Figure 3.2-11 shows the configuration of the CCR register.

Figure 3.2-11 Configuration of Condition Code Register (CCR)



DataSheet4U.com

● Interrupt enable flag (I)

All interrupts except software interrupts are enabled when the interrupt enable flag (CCR: I) is set to 1, and are disabled when the interrupt enable flag is set to 0. This flag is cleared to 0 by a reset.

● Stack flag (S)

This flag sets the pointer for stack processing.

When the stack flag (CCR: S) is 0, the user stack pointer (USP) is enabled. When the stack flag is 1, the system stack pointer (SSP) is enabled. If an interrupt is accepted or a reset occurs, the flag is set to 1.

● Sticky-bit flag (T)

If either one of the data shifted out of the carry is 1 when the logic right-shift instruction or arithmetic right-shift instruction is executed, this flag is set to 1. If all the shifted-out data is 0 or the shift amount is 0, this flag is set to 0.

● Negative flag (N)

If the most significant bit (MSB) of the operation result is 1, this flag is set to 1. If the MSB is 0, the flag is cleared to 0.

● Zero flag (Z)

If all the bits of the operation result are 0, this flag is set to 1. If any bit is 1, the flag is cleared to 0.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Overflow flag (V)

  If an overflow occurs as a signed numeric value at the execution of operation, this flag is set to 1. If no overflow occurs, the flag is cleared to 0.

- Carry flag (C)

  If a carry from the MSB or to the least significant bit (LSB) occurs at the execution of operation, this flag is set to 1. If no carry occurs, this flag is cleared to 0.

Reference:	For the state of the condition code register (CCR) at instruction execution, refer to the Programming Manual.

et4U.com DataShee
DataSheet4U.com

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

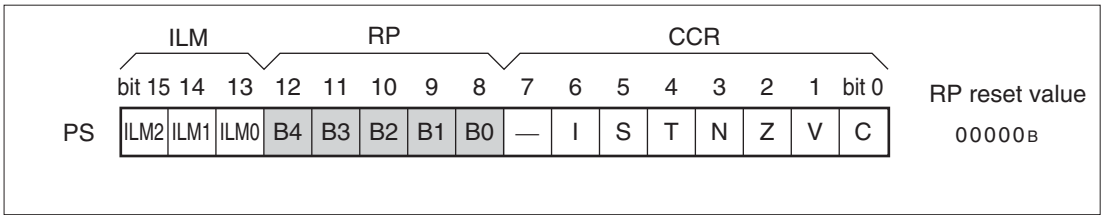
3.2.4.2 Register Bank Pointer (PS: RP)

The register bank pointer (RP) is a 5-bit register that indicates the starting address of the currently used general-purpose register bank.

■ Register Bank Pointer (RP)

Figure 3.2-12 shows the configuration of the register bank pointer (RP).

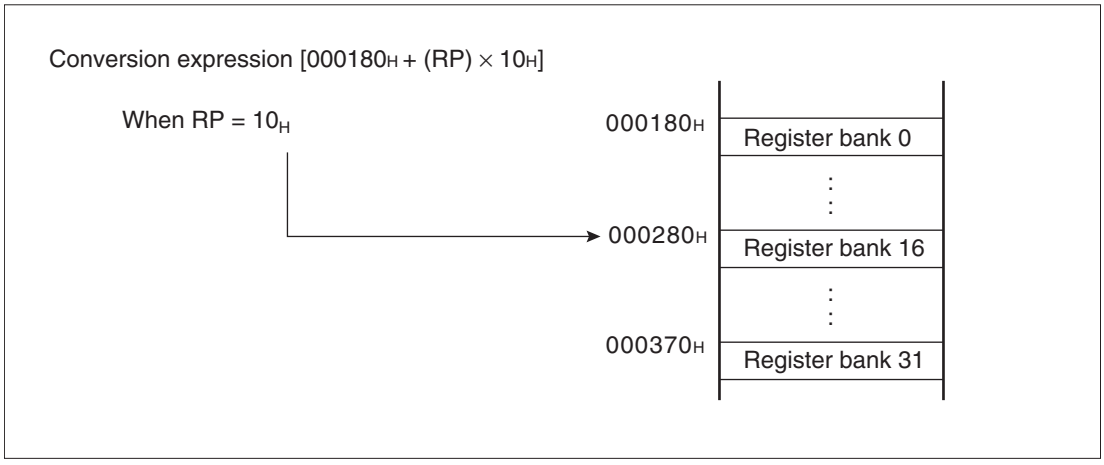
Figure 3.2-12 Configuration of Register Bank Pointer (RP)



■ General-purpose Register Area and Register Bank Pointer

The register bank pointer (RP) indicates the allocation of general-purpose registers used in the internal RAM. The relationship between the values of PR and the actual addresses should conform to the conversion rule shown in Figure 3.2-13.

Figure 3.2-13 Physical Address Conversion Rules in General-purpose Register Area



- The register bank pointer (RP) can take the values from "00_H" to "$1F_H$" so that the starting address of the register bank can be set within the range of "000180_H" to "$00037F_H$".
- The assembler instruction can use the 8-bit immediate value transfer instruction that is transferred to the register bank pointer (RP), but only the lower 5 bits of that data is actually used.
- The reset value of the register bank pointer (RP) is set to "00_H" after a reset.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

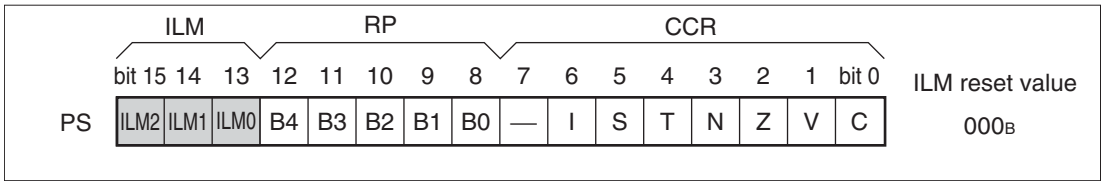
3.2.4.3 Interrupt Level Mask Register (PS: ILM)

The interrupt level mask register (ILM) is a 3-bit register indicating the interrupt level accepted by the CPU.

■ Interrupt Level Mask Register (ILM)

Figure 3.2-14 shows the configuration of the interrupt level mask register (ILM).

Figure 3.2-14 Configuration of Interrupt Level Mask Register (ILM)



The interrupt level mask register (ILM) indicates the level of an interrupt that the CPU is accepting for comparison with the values of the interrupt level setting bits (ICR: IL2 to IL0) set according to interrupt requests from each resource. The CPU performs interrupt processing only when an interrupt with a lower value (interrupt level) than that indicated by the interrupt level mask register (ILM) is requested with an interrupt enabled (CCR: I = 1).

- When an interrupt is accepted, its interrupt level value is set in the interrupt level mask register (ILM). Thereafter, an interrupt with a level value lower than the set level value is not accepted.
- At a reset, the interrupt level mask register (ILM) is always set to 0s to enter the interrupt-disabled (highest interrupt level) state.
- The assembler instruction can use the 8-bit immediate value transfer instruction that is transferred to the interrupt level mask register (ILM), but only the lower 3 bits of that data is actually used.

Table 3.2-3 Interrupt Level Mask Register (ILM) and Interrupt Level (High/Low)

ILM2	ILM1	ILM0	Interrupt Level	Interrupt Level (High/Low)
0	0	0	0	High (Interrupts Disabled) ↑
0	0	1	1	
0	1	0	2	
0	1	1	3	
1	0	0	4	
1	0	1	5	
1	1	0	6	
1	1	1	7	↓ Low

Reference: For details of the interrupts, see Section "3.5 Interrupt".

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.2.5 Program Counter (PC)

The program counter (PC) is a 16-bit counter indicating the lower 16 bits of the address for the next instruction code to be executed by the CPU.

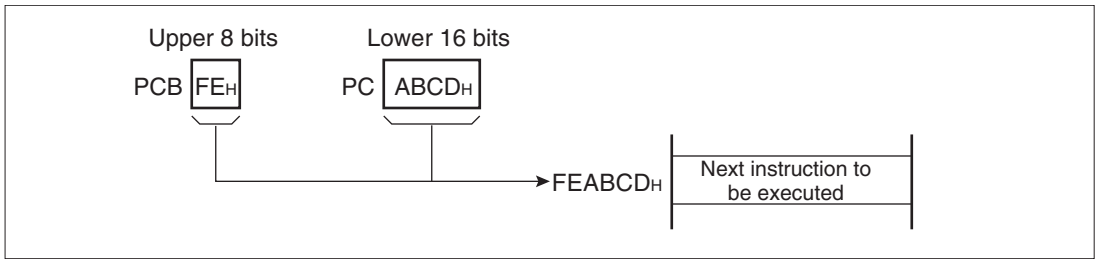
■ Program Counter (PC)

The program bank register (PCB) indicates the higher 8 bits of addresses where the next instruction code to be executed by the CPU is stored; the program counter (PC) indicates the lower 16 bits. As shown in Figure 3.2-15, the actual addresses are combined into 24 bits.

The program counter (PC) is updated by the execution of the conditional branch instruction, the subroutine call instruction, by an interrupt or reset, etc.

The program counter (PC) can also be used as the base pointer when reading the operand.

Figure 3.2-15 Program Counter (PC)



Note:	Neither the program counter (PC) nor the program bank register (PCB) can be rewritten directly by a program (such as MOV PC and #FF).

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.2.6 Direct Page Register (DPR)

The direct page register (DPR) sets bit 8 to bit 15 (addr 15 to addr 8) for the 8 bits of the low address directly specified using the operand when executing the instruction by the abbreviated direct addressing.

■ Direct Page Register (DPR)

The direct page register (DPR) sets bit 8 to bit 15 (addr 15 to addr 8) for the 8 bits of the low address directly specified using the operand when executing the instruction by the abbreviated direct addressing. The direct page register (DPR) is 8 bits long and is set to "01_H" at a reset. It is a read and write register.

Figure 3.2-16 Generation of Physical Address in Direct Page Register (DPR)

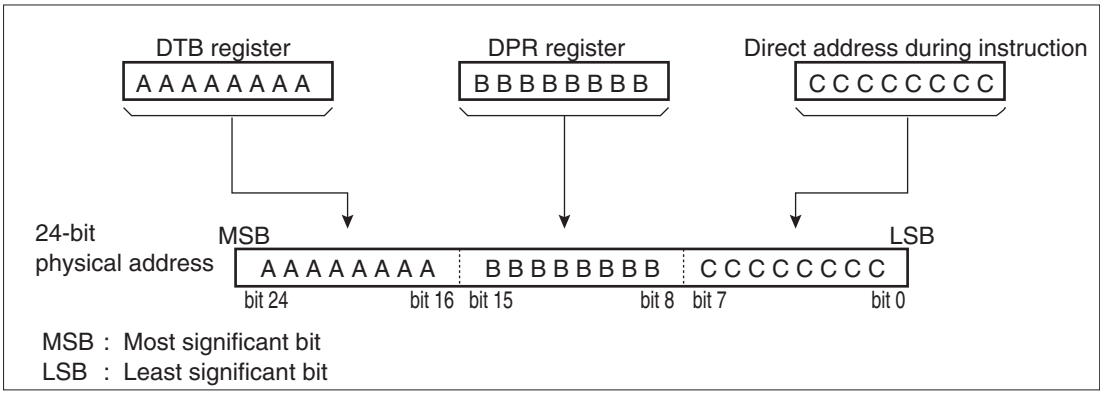
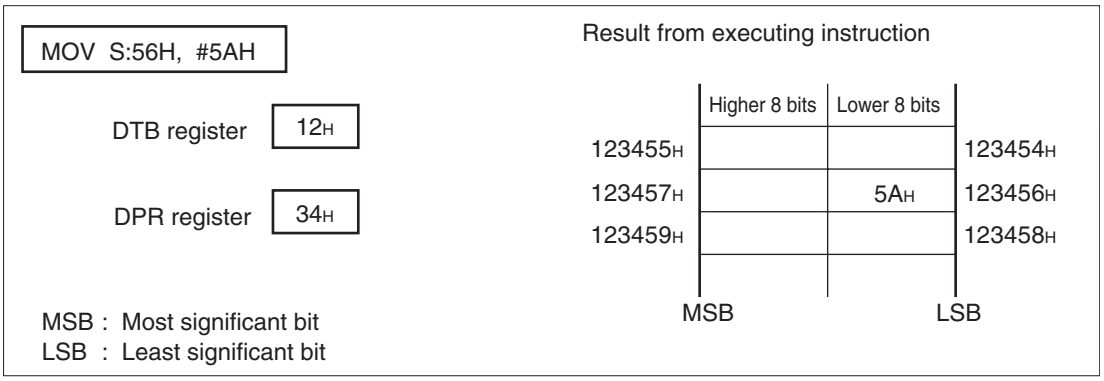


Figure 3.2-17 shows the setting of direct page register (DPR) and an example of data access.

Figure 3.2-17 Setting of Direct Page Register (DPR) and Data Access Example



DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.2.7 Bank Register (PCB, DTB, USB, SSB, and ADB)

The bank register sets the MSB 8 bit of the 24-bit address using bank addressing and consists of the following five registers:

- **Program bank register (PCB)**
- **Data bank register (DTB)**
- **User stack bank register (USB)**
- **System stack bank register (SSB)**
- **Additional bank register (ADB)**

Each of the above registers indicate the memory bank to which the program, data, user stack, system stack, or additional is allocated.

■ Program Bank Register (PCB)

The program bank register (PCB) sets the program (PC) space.

This register is rewritten at execution of the JMPP, CALLP, RETP, or RETI instruction that branches to the entire 16-MB space, at executing a software interrupt instruction, or at a hardware interrupt or exception interrupt.

■ Data Bank Register (DTB)

The data bank register (DTB) sets the data (DT) space.

■ User Stack Bank Register (USB) and System Stack Bank Register (SSB)

The user stack bank register (USB) and system stack bank register (SSB) set the stack (SP) space. The bank register that is used is determined by the value of the stack flag (CCR: S).

■ Additional Bank Register (ADB)

The additional bank register (ADB) sets the additional (AD) space.

■ Setting of Each Bank and Data Access

Each bank register is 8 bits long. At a reset, the program bank register (PCB) is set to "FF_H" and other bank registers are set to "00_H".

The program bank register (PCB) is a read-only register and other bank registers are read and write registers.

Reference: For the operation of each bank register, see Section "3.1 Memory Space".

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.3 General-purpose Register

The general-purpose register is a memory block allocated to addresses "000180_H" to "$00037F_H$" in the internal RAM in bank units of 16 bits x 8. It is configured as follows:

- **General-purpose 8-bit register (byte registers R0 to R7)**
- **16-bit register (word registers RW0 to RW7)**
- **32-bit register (long-word registers RL0 to RL7)**

■ Configuration of General-purpose Register

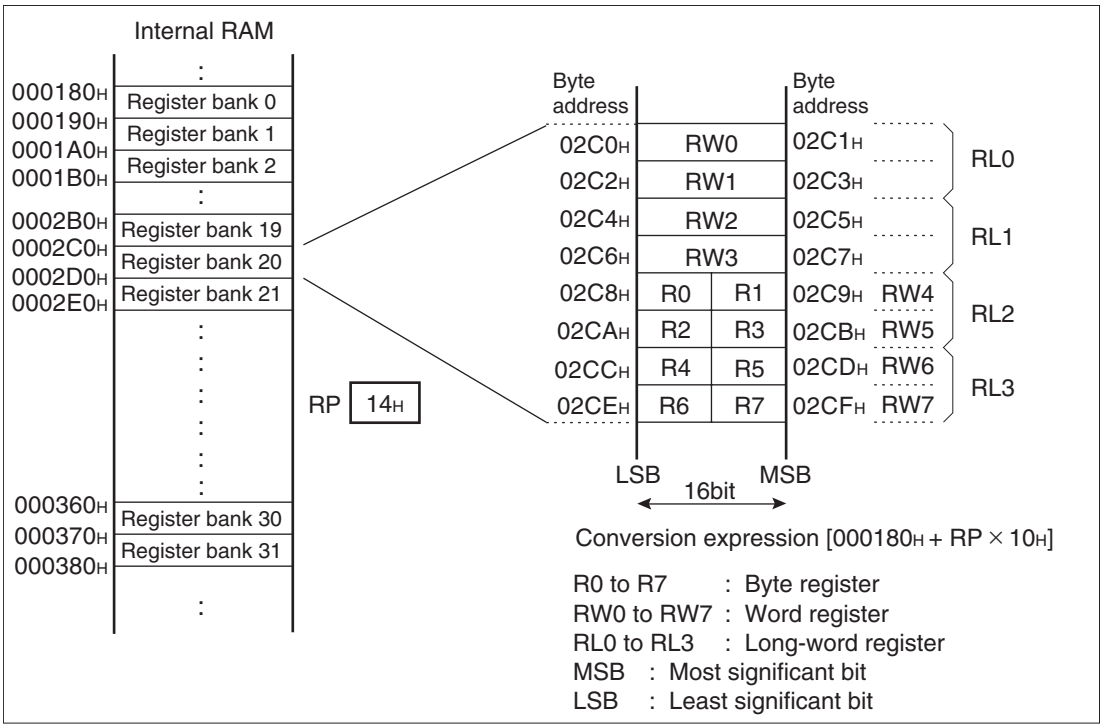
General-purpose registers are provided as 32 banks in the internal RAM from "000180_H" to "$00037F_H$". The banks that are used are set by the register bank pointer (RP). The current banks are indicated by reading the register bank pointer (RP).

The register bank pointer (RP) determines the starting address of each bank as the following expression.

Starting address of general-purpose register = "000180_H" + RP × "10_H"

Figure 3.3-1 shows the allocation and configuration of the general-purpose register banks in memory space.

Figure 3.3-1 Allocation and Configuration of General-Purpose Register Banks in Memory Space



Note: The register bank pointer (RP) is initialized to "00000_B" by a reset.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Register Bank

The register bank can be used as a general-purpose register (byte registers R0 to R7, word registers RW0 to RW7, and long-word registers RL0 to RL3) to perform various operations or to serve as a pointer. The long- word register can also be used as a linear addressing to directly access the entire memory space.

In the same way as ordinary RAM, the value in the general-purpose register is unchanged by a reset, meaning that the state before the reset is held. However, at power-on, the value is undefined.

Table 3.3-1 shows the typical function of the general-purpose register.

Table 3.3-1 Typical Function of the General-purpose Register

Register Name	Function
R0 to R7	Used as operands for various instructions **Note:** R0 can also be used as the barrel shift counter or the normalized instruction counter.
RW0 to RW7	Used as addressing Used as operands for various instructions **Note:** RW0 can also be used as the string instruction counter.
RL0 to RL3	Used as linear addressing Used as operands for various instructions

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.4 Prefix Codes

When prefix code is inserted prior to an instruction, the operation of the instruction can be changed partially. The prefix code has the following three types:

- **Bank select prefix (PCB, DTB, ADB, and SPB)**
- **Common register bank prefix (CMR)**
- **Flag change inhibit prefix (NCC)**

■ Prefix Code

● Bank select prefix (PCB, DTB, ADB, and SPB)

When the bank select prefix (PCB, DTB, ADB, SPB) codes precede an instruction, any memory space to be accessed by the instruction can be selected, regardless of the addressing types.

● Common register bank prefix (CMR)

When the common register bank prefix (CMR) code precedes an instruction for accessing a general-purpose register, the general-purpose register to be accessed by the instruction can be changed to a common bank (register bank selected when the register bank pointer (RP) is 0) at "000180_H" to "$00018F_H$", regardless of the current value of the register bank pointer (RP).

et4U.com DataShee DataSheet4U.com

● Flag change inhibit prefix (NCC)

When the flag change inhibit (NCC) code precedes an instruction for changing various flags of the condition code register (CCR), a flag change with instruction execution can be inhibited.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.4.1 Bank Select Prefix (PCB, DTB, ADB, and SPB)

When the bank select prefix (PCB, DTB, ADB, SPB) codes precede an instruction, any memory space accessed by the instruction can be set, regardless of the addressing types.

■ Bank Select Prefix (PCB, DTB, ADB, SPB)

Memory space used at data access is predetermined for each addressing type. However, when the bank select prefix (PCB, DTB, ADB, SPB) codes precede an instruction statement, any memory space accessed by the instruction statement can be set, regardless of the addressing types. Table 3.4-1 shows the bank select prefix code and the memory space to be selected.

Table 3.4-1 Bank Select Prefix

Bank Select Prefix	Selected Space
PCB	Program space
DTB	Data space
ADB	Additional space
SPB	When the stack flag (CCR: S) is 0, user stack space is selected. When the stack flag is 1, system stack space is selected.

et4U.com DataShee DataSheet4U.com

The use of the bank select prefix (PCB, DTB, ADB, SPB) codes causes some instructions to perform exceptional operations as explained below.

Table 3.4-2 shows the instructions not affected by the bank select prefix code, and Table 3.4-3 shows the instructions requiring precaution when using the bank select prefix.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 3.4-2 Instructions Unaffected by Bank Select Prefix

Instruction Type	Instruction		Effect
String instruction	MOVS SCEQ FILS	MOVSW SCWEQ FILSW	The bank register specified for the operand is used irrespective of the presence or absence of the bank select prefix code.
Stack instruction	PUSHW	POPW	Irrespective of the presence or absence of the bank select prefix code, the user stack bank (USB) is used when the S flag is 0; and the system stack bank (SSB) is used when the S flag is 1

I/O access instruction	MOV A,io
	MOVW A,io
	MOV io,A
	MOV io,#imm8
	MOVB A,io:bp
	SETB io:bp
	BBC io:bp,rel
	WBTC io,bp
		MOVX A, io
		
		MOVW io,A
		MOVW io,#imm16
		MOVB io:bp,A
		CLRB io:bp
		BBS io:bp,rel
		WBTS io:bp
			The I/O space ("000000_H" to "$0000FF_H$") is accessed irrespective of the presence or absence of the bank select prefix code.

Interrupt return instruction	RETI		The system stack bank (SSB) is used irrespective of the presence or absence of the bank select prefix code.

Table 3.4-3 Instructions Requiring Precaution When Using Bank Select Prefix

Instruction Type	Instruction		Explanation
Flag change instruction	AND OR	CCR,#imm8 CCR,#imm8	The bank select prefix code affects up to the next instruction.
ILM setting instruction	MOV	ILM,#imm8	The bank select prefix code affects up to the next instruction.
PS return instruction	POPW	PS	Do not add the bank select prefix code to the PS return instruction.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.4.2 Common Register Bank Prefix (CMR)

When the common register bank prefix (CMR) code precedes an instruction for accessing a general-purpose register, the general-purpose register to be accessed by the instruction can be changed to a common bank register bank selected when the register bank pointer (RP) is 0 at "000180_H" to "$00018F_H$", regardless of the current value of the register bank pointer (RP).

■ Common Register Bank Prefix (CMR)

The F^2MC-16LX family provides common banks at "000180_H" to "$00018F_H$" as register banks that can be commonly accessed by each task, regardless of the values of the register bank pointer (RP).

The use of the common banks facilitates data exchange between two or more tasks.

When the common register bank prefix (CMR) code precedes an instruction for accessing a general-purpose register, the general-purpose register accessed by the instruction can be changed to a common bank (register bank to be selected when the register bank pointer (RP) is 0) at "000180_H" to "$00018F_H$", regardless of the current value of the register bank pointer (RP).

Table 3.4-4 shows the instructions requiring care when using the common register bank prefix.

et4U.com

Table 3.4-4 Instructions Requiring Precaution When Using Common Register Bank Prefix (CMR)

DataSheet4U.com

Instruction Type	Instruction		Explanation
String instruction	MOVS SCEQ FILS	MOVSW SCWEQ FILSW	Do not add the CMR code to string instructions.
Flag change instruction	AND OR	CCR,#imm8 CCR,#imm8	The CMR code affects up to the next instruction.
PS return instruction	POPW	PS	The CMR code affects up to the next instruction.
ILM setting instruction	MOV	ILM,#imm8	The CMR code affects up to the next instruction.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.4.3 Flag Change Inhibit Prefix (NCC)

When the flag change inhibit prefix (NCC) code precedes an instruction for changing various flags of the condition code register (CCR), a flag change caused by instruction execution can be inhibited.

■ Flag Change Inhibit Prefix (NCC)

The flag change inhibit prefix (NCC) code is used to inhibit an unnecessary flag change.

When the flag change inhibit prefix (NCC) code precedes an instruction for changing various flags of the condition code register (CCR), a flag change caused by instruction execution can be inhibited. The inhibited flags are:

- Sticky-bit flag (CCR: T)
- Negative flag (CCR: N)
- Zero flag (CCR: Z)
- Overflow flag (CCR: V)
- Carry flag (CCR: C)

Table 3.4-5 shows the instructions requiring precaution when using the flag change inhibit prefix.

et4U.com

Table 3.4-5 Instructions Requiring Precaution When Using Flag Change Inhibit Prefix (NCC)

DataShee

Instruction Type	Instruction	Explanation
String instruction	MOVS MOVSW SCEQ SCWEQ FILS FILSW	Do not add the NCC code to the string instruction.
Flag change instruction	AND CCR,#imm8 OR CCR,#imm8	The CCR changes by execution of an instruction, regardless of the presence or absence of the NCC code. The NCC code affects the next instruction.
PS return instruction	POPW PS	The CCR changes by execution of an instruction, regardless of the presence or absence of the NCC code. The NCC code affects the next instruction.
ILM setting instruction	MOV ILM,#imm8	The NCC code affects the next instruction.
Interrupt instruction Interrupt return instruction	INT #vct8 INT9 INT INTP addr16 addr24 RETI	The CCR changes by execution of an instruction statement, regardless of the presence or absence of the NCC code.
Context switch instruction	JCTX @ A	The CCR changes by execution of an instruction statement, regardless of the presence or absence of the NCC code.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.4.4 Restrictions on Prefix Code

The use of the prefix codes is restricted as follows:

- **No interrupt request is accepted during execution of a prefix code and interrupt inhibit instruction.**
- **When a prefix code precedes an interrupt inhibit instruction, the effect of the prefix code is delayed.**
- **When conflicting prefix codes are used in succession, the last prefix code is enabled.**

■ Prefix Code and Interrupt Inhibit Instruction

The interrupt inhibit instruction and prefix code are restricted as shown below.

Table 3.4-6 Prefix Code and Interrupt Inhibit Instruction

	Prefix Code	Interrupt/Hold Inhibit Instruction (instruction that delays effect of prefix code)	

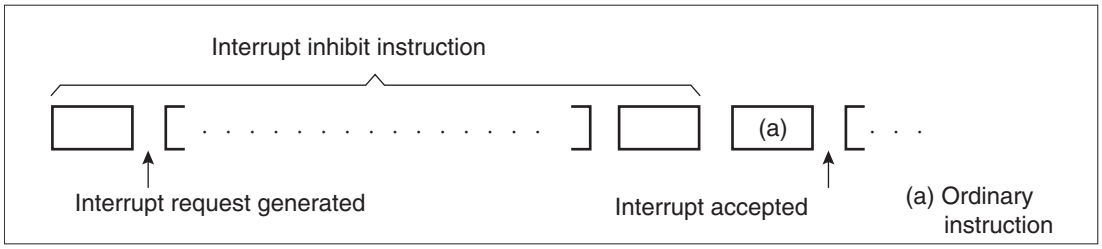
Instruction that does not accept interrupt request	PCB
	DTB
	ADB
	SPB
	CMR
	NCC
		`MOV`
		`OR`
		`AND`
		`POPW`
			`ILM,#imm8`
			`CCR,#imm8`
			`CCR,#imm8`
			`PS`

et4U.com DataShee DataSheet4U.com

● Interrupt Inhibition

Even if an interrupt request is generated, it is not accepted during execution of a prefix code and interrupt inhibit instruction. When other instructions are executed after execution of a prefix code and interrupt inhibit instruction, an interrupt is processed.

Figure 3.4-1 Interrupt Inhibition



DataSheet4U.com www.DataSheet4U.com

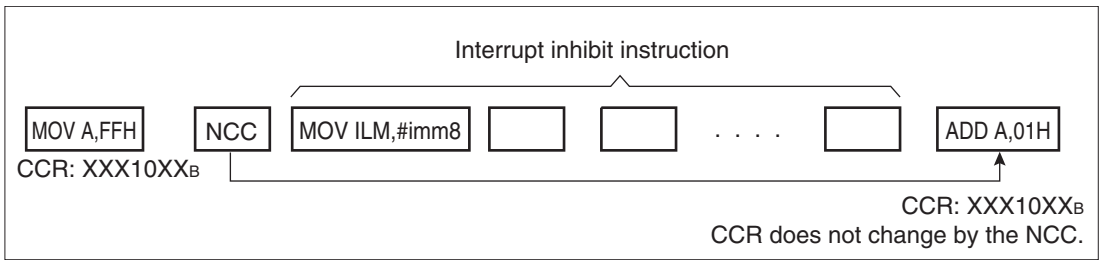
DataSheet4U.com

www.DataSheet4U.com

● Delay of the effect of the prefix code

When a prefix code precedes an interrupt inhibit instruction, it affects an instruction next to the interrupt inhibit instruction.

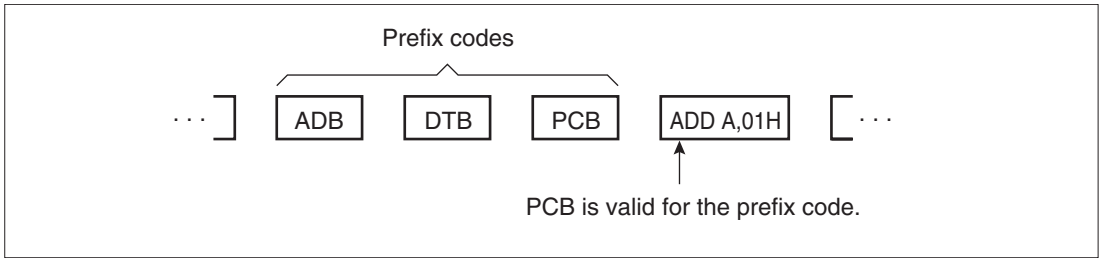
Figure 3.4-2 Interrupt Inhibit Instruction and Prefix Code



■ Array of Prefix Codes

For array of conflicting prefix codes (PCB, ADB, DTB, SPB), the last one is enabled.

Figure 3.4-3 Array of Prefix Codes



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5 Interrupt

The F^{2}MC-16LX family has four interrupt functions for suspending the current processing to transfer control to a program which is defined separately at generation of event.

- **Hardware interrupt**
- **Software interrupt**
- **Interrupts by extended intelligent I/O service (EI^{2}OS)**
- **Exception processing**

■ Type and Function of Interrupt

● Hardware interrupt

This transits control to the interrupt processing program defined by user in response to the interrupt request from resources.

● Software interrupt

This transfers control to the interrupt processing program defined by user by executing an instruction (such as INT instruction) dedicated to the software interrupt.

DataSheet4U.com

● Interrupt by extended intelligent I/O service (EI^{2}OS)

The extended intelligent I/O service (EI^{2}OS) provides automatic data transfer between resources and memory. Data can be transferred just by creating the startup-setting program and end program of the EI^{2}OS. At completion of data transfer, the interrupt processing program is executed automatically.

An interrupt generated by the EI^{2}OS is a type of the above hardware interrupt.

● Exception processing

If an exception (execution of an undefined instruction) is detected among instructions, ordinary processing is suspended to perform exception processing. This is equivalent to the above software interrupt instruction INT10.

DataSheet4U.com www.DataSheet4U.com

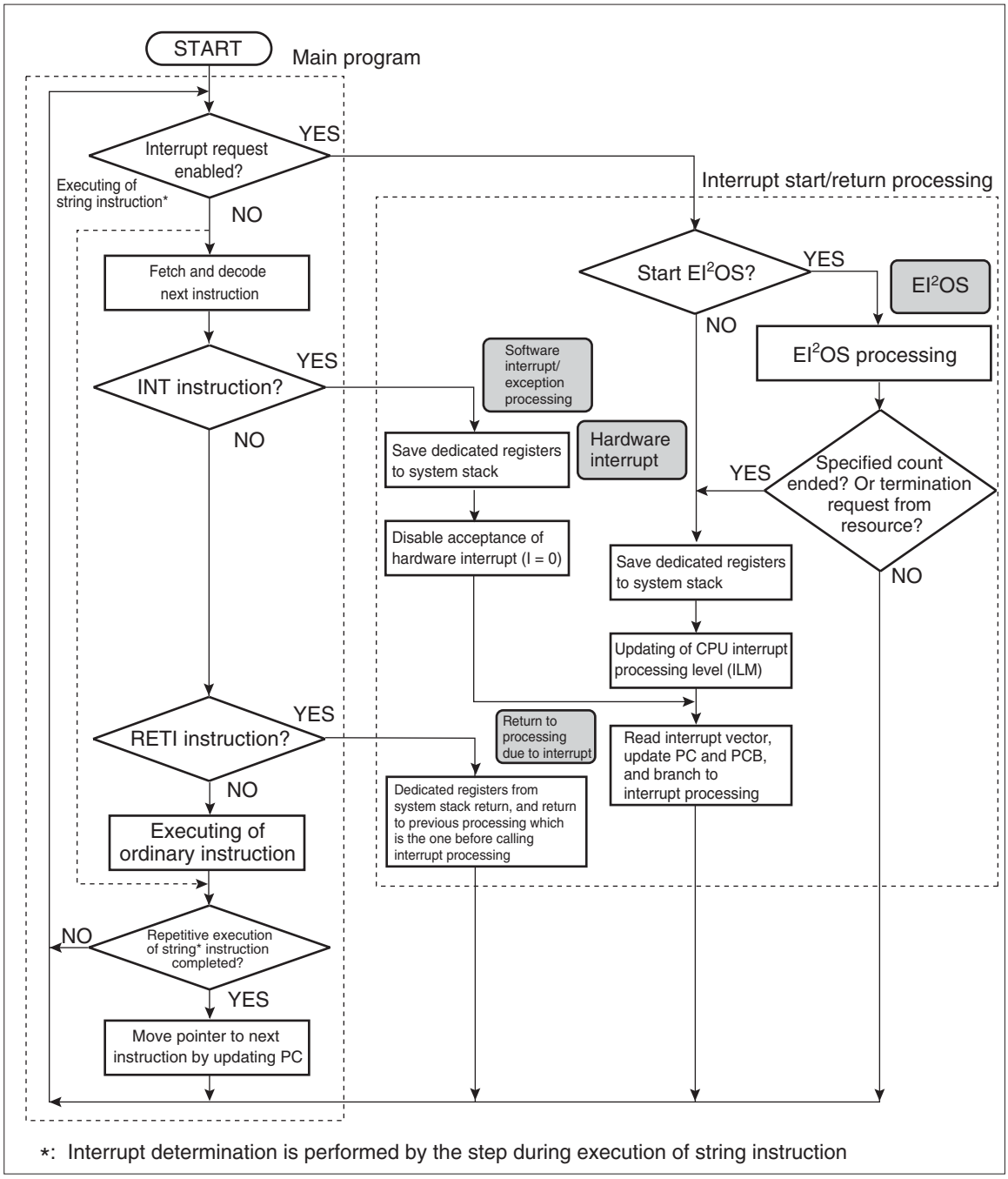
DataSheet4U.com

www.DataSheet4U.com

■ Interrupt Operation

Figure 3.5-1 shows interrupt start and return processing.

Figure 3.5-1 General Flow of Interrupt Operation



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.1 Interrupt Factor and Interrupt Vector

The F^{2}MC-16LX family has vector tables corresponding to 256 types of interrupt factor.

■ Interrupt Vector

The interrupt vector tables referenced at interrupt processing are allocated to the most significant addresses ("FFFC00$_H$" to "FFFFFF$_H$") of the memory area. The interrupt vectors share the same area with the EI^{2}OS, exception processing, and hardware and software interrupts.

- Interrupts (INT0 to INT255) are used as software interrupts.
- At hardware interrupts, the interrupt vectors and interrupt control register (ICR) are fixed for each resource.

Table 3.5-1 shows the interrupt number and allocation of interrupt vector.

Table 3.5-1 List of Interrupt Vectors

Software Interrupt Instruction	Vector Address (Low)	Vector Address (Middle)	Vector Address (High)	Mode Data	Interrupt Number	Hardware Interrupt
INT0	FFFFFC$_H$	FFFFFD$_H$	FFFFFE$_H$	Unused	#0	None
:	:	:	:	:	:	:
INT7	FFFFE0$_H$	FFFFE1$_H$	FFFFE2$_H$	Unused	#7	None
INT8	FFFFDC$_H$	FFFFDD$_H$	FFFFDE$_H$	FFFFDF$_H$	#8	(RESET vector)
INT9	FFFFD8$_H$	FFFFD9$_H$	FFFFDA$_H$	Unused	#9	None
INT10	FFFFD4$_H$	FFFFD5$_H$	FFFFD6$_H$	Unused	#10	<Exception processing>
INT11	FFFFD0$_H$	FFFFD1$_H$	FFFFD2$_H$	Unused	#11	Resource interrupt #0
INT12	FFFFCC$_H$	FFFFCD$_H$	FFFFCE$_H$	Unused	#12	Resource interrupt #1
INT13	FFFFC8$_H$	FFFFC9$_H$	FFFFCA$_H$	Unused	#13	Resource interrupt #2
INT14	FFFFC4$_H$	FFFFC5$_H$	FFFFC6$_H$	Unused	#14	Resource interrupt #3
:	:	:	:	:	:	:
INT254	FFFC04$_H$	FFFC05$_H$	FFFC06$_H$	Unused	#254	None
INT255	FFFC00$_H$	FFFC01$_H$	FFFC02$_H$	Unused	#255	None

Reference: It is recommended to set the unused interrupt vectors to the addresses for exception processing.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Interrupt Factor, Interrupt Vector, and Interrupt Control Register

Table 3.5-2 shows the relationships between the interrupt factor except software interrupt, interrupt vector, and interrupt control register.

Table 3.5-2 Interrupt Factor, Interrupt Vector, and Interrupt Control Register (1/2)

Interrupt Factor	EI^{2}OS-Corresponded	Interrupt Vector			Interrupt Control Register		Priority(*4)
		Number		Address	ICR	Address	
Reset	X	#08	08_H	$FFFFDC_H$	–	–	Highest ↑
INT9 instruction	X	#09	09_H	$FFFFD8_H$	–	–	
Exception processing	X	#10	$0A_H$	$FFFFD4_H$	–	–	
CAN controller receive completion (RX)	X	#11	$0B_H$	$FFFFD0_H$	ICR00	$0000B0_H$(*1)	
CAN controller transmit completion (TX)/node status transition (NS)	X	#12	$0C_H$	$FFFFCC_H$			
Reserved	X	#13	$0D_H$	$FFFFC8_H$	ICR01	$0000B1_H$	
Reserved	X	#14	$0E_H$	$FFFFC4_H$			
CAN wake-up	△	#15	$0F_H$	$FFFFC0_H$	ICR02	$0000B2_H$(*3)	
Timebase timer	X	#16	10_H	$FFFFBC_H$			
16-bit reload timer 0	△	#17	11_H	$FFFFB8_H$	ICR03	$0000B3_H$(*1)	
8-/10-bit A/D converter	△	#18	12_H	$FFFFB4_H$			
16-bit free-run timer overflow	△	#19	13_H	$FFFFB0_H$	ICR04	$0000B4_H$	
Reserved	X	#20	14_H	$FFFFAC_H$			
Reserved	X	#21	15_H	$FFFFA8_H$	ICR05	$0000B5_H$	
PPG timer ch 0/1 underflow	X	#22	16_H	$FFFFA4_H$			
Input capture 0 fetched	△	#23	17_H	$FFFFA0_H$	ICR06	$0000B6_H$(*1)	
External interrupt (INT4/INT5)	△	#24	18_H	$FFFF9C_H$			
Input capture 1 fetched	△	#25	19_H	$FFFF98_H$	ICR07	$0000B7_H$(*2)	
PPG timer ch 2/3 underflow	X	#26	$1A_H$	$FFFF94_H$			
External interrupt (INT6/INT7)	△	#27	$1B_H$	$FFFF90_H$	ICR08	$0000B8_H$(*1)	
Watch timer	△	#28	$1C_H$	$FFFF8C_H$			

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 3.5-2 Interrupt Factor, Interrupt Vector, and Interrupt Control Register (2/2)

Interrupt Factor	EI^{2}OS-Corresponded	Interrupt Vector			Interrupt Control Register		Priority(*4)
		Number		Address	ICR	Address	
Reserved	X	#29	$1D_H$	$FFFF88_H$	ICR09	$0000B9_H$	
Input capture 2 fetched Input capture 3 fetched	X	#30	$1E_H$	$FFFF84_H$			
Reserved	X	#31	$1F_H$	$FFFF80_H$	ICR10	$0000BA_H$	
Reserved	X	#32	20_H	$FFFF7C_H$			
Reserved	X	#33	21_H	$FFFF78_H$	ICR11	$0000BB_H$	
Reserved	X	#34	22_H	$FFFF74_H$			
Reserved	X	#35	23_H	$FFFF70_H$	ICR12	$0000BC_H$	
16-bit reload timer 1	O	#36	24_H	$FFFF6C_H$			
UART1 receive	◎	#37	25_H	$FFFF68_H$	ICR13	$0000BD_H$(*1)	
UART1 transmit	◎	#38	26_H	$FFFF64_H$			
Reserved	X	#39	27_H	$FFFF60_H$	ICR14	$0000BE_H$	
Reserved	X	#40	28_H	$FFFF5C_H$			
Flash memory	X	#41	29_H	$FFFF58_H$	ICR15	$0000BF_H$(*1)	↓ Lowest
Delayed interrupt generation module	X	#42	$2A_H$	$FFFF54_H$			

O: Interrupt factor corresponds to EI^{2}OS

X: Interrupt factor does not correspond to EI^{2}OS

◎: Interrupt factor corresponds to EI^{2}OS and has EI^{2}OS stop function

△: Interrupt factor can be used when not using interrupt sources sharing ICR register

*1:

- The interrupt level for resources sharing an ICR register become the same.
- When two resources share an ICR register, only one can use the EI^{2}OS.
- When two resources share an ICR register and one specifies the EI^{2}OS, the remaining resource cannot use the interrupt.

*2: The only input capture1 corresponds to the EI^{2}OS function and the PPG does not correspond to the EI^{2}OS function. Therefore, if the EI^{2}OS function is used by the input capture1, the PPG is set to disable generation of interrupt requests.

*3: The only CAN wake-up corresponds to the EI^{2}OS function and the timebase timer dose not correspond to the EI^{2}OS function. Therefore, if the EI^{2}OS function is used by the CAN wake-up, the timebase timer is set to disable generation of interrupt requests.

*4: The priority is given when plural interrupts with the same level are generated simultaneously.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.2 Interrupt Control Registers and Resources

The interrupt control registers (ICR00 to ICR15) are allocated in the interrupt controller, and correspond to all resources with interrupt functions. The registers control the interrupt and extended intelligent I/O service (EI^{2}OS).

■ Interrupt Control Register List

Table 3.5-3 lists the resources corresponding to the interrupt control registers.

Table 3.5-3 Interrupt Control Register List

Address	Register	Abbreviation	Corresponding Resource
$0000B0_H$	Interrupt control register 00	ICR00	CAN controller
$0000B1_H$	Interrupt control register 01	ICR01	Reserved
$0000B2_H$	Interrupt control register 02	ICR02	CAN wake-up Timebase timer
$0000B3_H$	Interrupt control register 03	ICR03	16-bit reload timer 0 A/D converter
$0000B4_H$	Interrupt control register 04	ICR04	16-bit free-run timer overflow
$0000B5_H$	Interrupt control register 05	ICR05	PPG 0/1
$0000B6_H$	Interrupt control register 06	ICR06	Input capture 0 External interrupt INT4/INT5
$0000B7_H$	Interrupt control register 07	ICR07	Input capture 1 PPG 2/3
$0000B8_H$	Interrupt control register 08	ICR08	External interrupt INT6/INT7 Watch timer
$0000B9_H$	Interrupt control register 09	ICR09	Input capture 2/3
$0000BA_H$	Interrupt control register 10	ICR10	Reserved
$0000BB_H$	Interrupt control register 11	ICR11	Reserved
$0000BC_H$	Interrupt control register 12	ICR12	16-bit reload timer 1
$0000BD_H$	Interrupt control register 13	ICR13	UART1
$0000BE_H$	Interrupt control register 14	ICR14	Reserved
$0000BF_H$	Interrupt control register 15	ICR15	Flash memory, delayed interrupt

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

The interrupt control register (ICR) has the following four functions.

Some functions of the interrupt control register (ICR) are different at write and read.

- Setting of interrupt level of corresponding resource
- Selection of whether to perform normal interrupt or EI^{2}OS for corresponding resource
- Selection of channel of EI^{2}OS
- Display of end state of EI^{2}OS

Note:	Do not access the interrupt control register (ICR) using the read modify write instruction because it causes a malfunction.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.3 Interrupt Control Register (ICR00 to ICR15)

The functions of the interrupt control registers are shown below.

■ Interrupt Control Register (ICR00 to ICR15)

Part of functions differ depending on whether data is written to or read from the interrupt control registers.

Figure 3.5-2 Interrupt Control Register (ICR00 to ICR15) at Write

At write

7	6	5	4	3	2	1	0
W	W	W	W	R/W	R/W	R/W	R/W

Reset value
0 0 0 0 0 1 1 1 B

bit 2 bit 1 bit 0

IL2	IL1	IL0	Interrupt level setting bits
0	0	0	Interrupt level 0 (highest)
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	
1	1	1	Interrupt level 7 (no interrupt)

bit 3

ISE	EI^{2}OS enable bit
0	Starts normal interrupt processing at an interrupt
1	Starts EI^{2}OS at an interrupt

bit 7 bit 6 bit 5 bit 4

ICS3	ICS2	ICS1	ICS0	EI^{2}OS channel select bit	
				Channel	Descriptor address
0	0	0	0	0	000100H
0	0	0	1	1	000108H
0	0	1	0	2	000110H
0	0	1	1	3	000118H
0	1	0	0	4	000120H
0	1	0	1	5	000128H
0	1	1	0	6	000130H
0	1	1	1	7	000138H
1	0	0	0	8	000140H
1	0	0	1	9	000148H
1	0	1	0	10	000150H
1	0	1	1	11	000158H
1	1	0	0	12	000160H
1	1	0	1	13	000168H
1	1	1	0	14	000170H
1	1	1	1	15	000178H

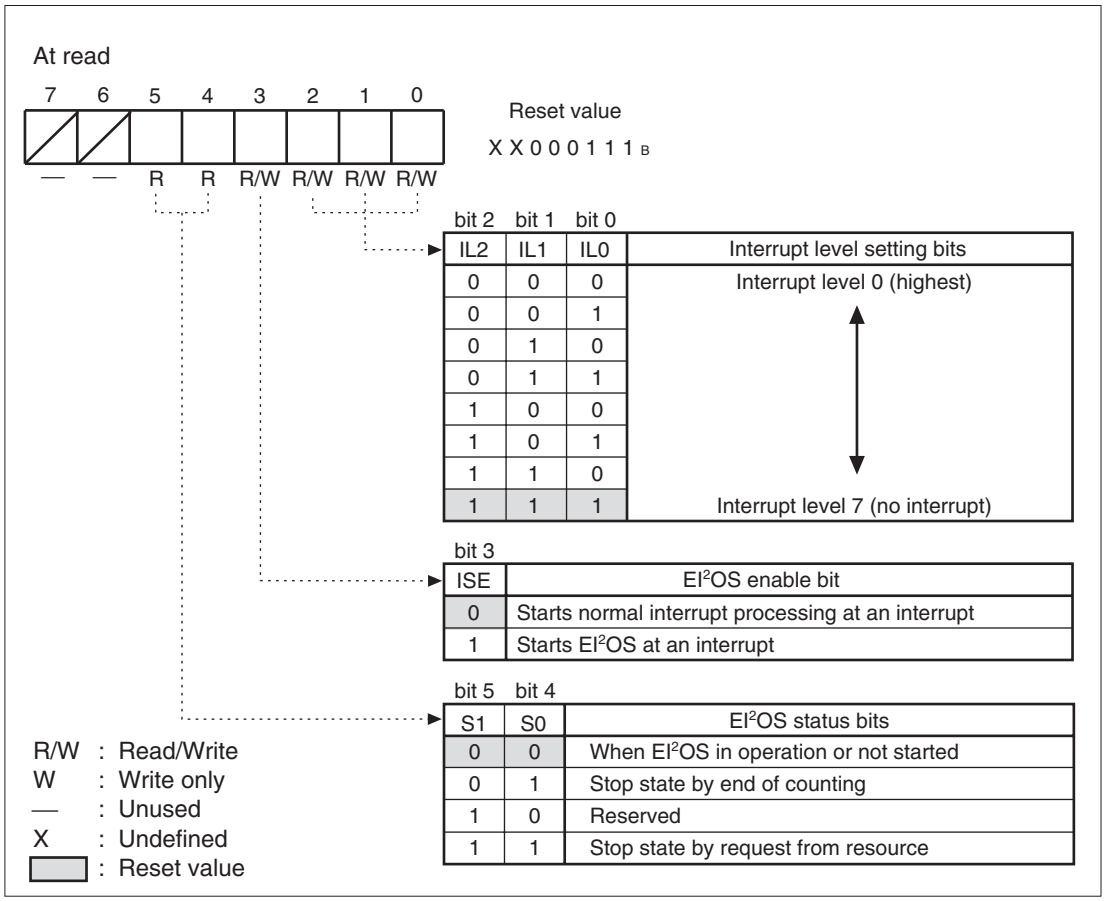
R/W : Read/Write
W : Write only
□ : Reset value

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 3.5-3 Interrupt Control Register (ICR00 to ICR15) at Read



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.4 Function of Interrupt Control Register

The interrupt control registers (ICR00 to ICR15) consist of the following bits with four functions.

- **Interrupt level setting bits (IL2 to IL0)**
- **EI^{2}OS enable bit (ISE)**
- **EI^{2}OS channel select bits (ICS3 to ICS0)**
- **EI^{2}OS status bits (S1 and S0)**

■ Bit Configuration of Interrupt Control Register (ICR)

The bit configuration of the interrupt control registers (ICR) is show below.

Figure 3.5-4 Configuration of Interrupt Control Register (ICR)

Configuration of interrupt control register (ICR) at write

bit 7	6	5	4	3	2	1	bit 0	Reset value
ICS3	ICS2	ICS1	ICS0	ISE	IL2	IL1	IL0	00000111_B
W	W	W	W	W	W	W	W	

Configuration of interrupt control register (ICR) at read

bit 7	6	5	4	3	2	1	bit 0	Reset value
-	-	S1	S0	ISE	IL2	IL1	IL0	$XX000111_B$
-	-	R	R	R	R	R	R	

R: Read only
W: Write only
-: Unused

References:

- The setting of the channel select bits (ICR: ICS3 to ICS0) is enabled only when starting the EI^{2}OS. When starting the EI^{2}OS, set the EI^{2}OS enable bit (ICR: ISE) to 1. When not starting the EI^{2}OS, set the bit to 0.
- The channel select bits (ICR: ICS3 to ICS0) are enabled only at write, and the EI^{2}OS status bits (ICR: S1, S0) are enabled only at read.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Function of Interrupt Control Register

● Interrupt level setting bits (IL2 to IL0)

These bits set the interrupt levels of the corresponding resources. At reset, the bits are set to level 7 (IL2 to IL0 = "111_B": no interrupt).

Table 3.5-4 shows the relationship between the interrupt level setting bits and interrupt levels.

Table 3.5-4 Relationship between Interrupt Level Setting Bits and Interrupt Levels

IL2	IL1	IL0	Interrupt Level
0	0	0	0 (Highest) ↑
0	0	1	
0	1	0	
0	1	1	
1	0	0	
1	0	1	
1	1	0	↓ 6 (Lowest)
1	1	1	7 (No interrupt)

et4U.com DataShee DataSheet4U.com

● EI^{2}OS enable bit (ISE)

When an interrupt occurs with the ISE bit set to 1, the EI^{2}OS is started. When an interrupt occurs with the ISE bit set to 0, ordinary interrupt processing is started. If the EI^{2}OS end condition is satisfied (when the status bits S1 and S0 are not "00_B"), the ISE bit is cleared. When the corresponding resources have no EI^{2}OS function, this bit must be set to 0 by the program. At reset, the ISE bit is set to 0.

● EI^{2}OS channel select bits (ICS3 to ICS0)

These bits select EI^{2}OS channels. The EI^{2}OS descriptor addresses are set according to the setting values of the ICS3 to ICS0 bits. At reset, the ICS3 to ICS0 are set to "0000_B".

Table 3.5-5 shows the correspondence between the EI^{2}OS channel select bits and descriptor addresses.

Table 3.5-5 Correspondence between EI^{2}OS Channel Select Bits and Descriptor Addresses (1/2)

ICS3	ICS2	ICS1	ICS0	Channel to be Selected	Descriptor Address
0	0	0	0	0	000100_H
0	0	0	1	1	000108_H
0	0	1	0	2	000110_H
0	0	1	1	3	000118_H
0	1	0	0	4	000120_H

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 3.5-5 Correspondence between EI^{2}OS Channel Select Bits and Descriptor Addresses (2/2)

ICS3	ICS2	ICS1	ICS0	Channel to be Selected	Descriptor Address
0	1	0	1	5	000128_H
0	1	1	0	6	000130_H
0	1	1	1	7	000138_H
1	0	0	0	8	000140_H
1	0	0	1	9	000148_H
1	0	1	0	10	000150_H
1	0	1	1	11	000158_H
1	1	0	0	12	000160_H
1	1	0	1	13	000168_H
1	1	1	0	14	000170_H
1	1	1	1	15	000178_H

et4U.com

- EI^{2}OS status bits (S1 and S0)

DataShee

When the S1 and S0 bits are read at the termination of the EI^{2}OS, the operating and end states can be checked. At reset, the S1 and S0 bits are set to "00_B".

DataSheet4U.com

Table 3.5-6 shows the relationship between the EI^{2}OS status bits (ICR: S1, S0) and the EI^{2}OS status.

Table 3.5-6 Relationships Between EI^{2}OS Status Bits and EI^{2}OS Status

S1	S0	EI^{2}OS Status
0	0	When EI^{2}OS in operation or not started
0	1	Stop state by end of counting
1	0	Reserved
1	1	Stop state by request from resource

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.5 Hardware Interrupt

The hardware interrupt responds to the interrupt request from a resource, suspends the current-executing program and transfers control to the interrupt processing program defined by user.

The hardware interrupt corresponds to the EI^{2}OS.

■ Hardware Interrupt

- Function of hardware interrupt

  When a hardware interrupt is generated, the interrupt level (IR: IL) of an interrupt request from a resource is compared with the interrupt level mask register (PS: ILM) and the state of the interrupt enable flag (CCR: I) is referenced to determine whether to accept the hardware interrupt.

  When the hardware interrupt is accepted, registers in the CPU are automatically saved in the system stack. The interrupt level of the accepted interrupt is stored in the interrupt level mask register (ILM), then branches to the corresponding interrupt vector.

- Multiple interrupts

  Multiple hardware interrupts can be started.

DataSheet4U.com

- EI^{2}OS

  When the EI^{2}OS function ends, normal interrupt processing is performed. No multiple EI^{2}OSs are started. Other interrupt requests and EI^{2}OS requests are held during EI^{2}OS processing.

- External interrupt

  The external interrupt (wake-up interrupt included) is accepted as a hardware interrupt via the resource (interrupt request detector).

- Interrupt vector

  The interrupt vector tables referenced during interrupt processing are allocated to "FFFC00$_H$" to "FFFFFF$_H$" in the memory and shared with software interrupts.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Mechanism of Hardware Interrupt

The mechanism related to the hardware interrupt consists of the four sections.

When starting the hardware interrupt, these four sections must be set by the program.

Table 3.5-7 Mechanism Related to Hardware Interrupt

	Mechanism Related to Hardware Interrupt	Function
Resource	Interrupt enable bit, interrupt request bit	Controls interrupt request from resource
Interrupt controller	Interrupt control register (ICR)	Sets interrupt level and controls EI^2OS
CPU	Interrupt enable flag (I)	Identifies interrupt enable state
	Interrupt level mask register (ILM)	Compares requested interrupt level and current interrupt level
	Microcode	Executes interrupt routine
"$FFFC00_H$ to "$FFFFFF_H$" in memory	Interrupt vector table	Stores branch destination address at interrupt processing

■ Hardware Interrupt Inhibition

et4U.com

No hardware interrupt requests are inhibited under following conditions.

DataShee

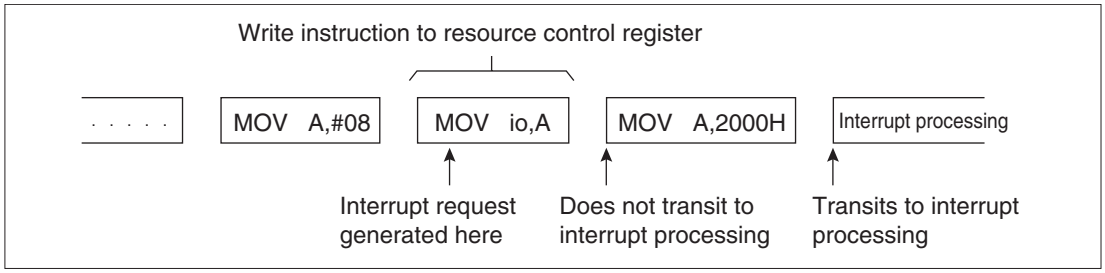
DataSheet4U.com

- Hardware interrupt inhibition during write to resource control register in I/O area

No hardware interrupt requests are accepted during write to resource control register. This prevents the CPU from malfunctioning with respect to interrupt requests generated during rewrite related to interrupt control registers of each resource.

Figure 3.5-5 shows the hardware interrupt operation during write to the resource control register.

Figure 3.5-5 Hardware Interrupt Request During Write to the Resource Control Register



DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Hardware interrupt inhibition by interrupt inhibit instruction

Table 3.5-8 shows the hardware interrupt inhibit instructions.

If a hardware interrupt is generated during execution of a hardware interrupt inhibit instruction, it is processed after execution of the hardware interrupt inhibit instruction, then and other instruction.

Table 3.5-8 Hardware Interrupt Inhibit Instructions

	Prefix code	Interrupt Inhibit Instruction	
Instruction that does not accept interrupt request	PCB DTB ADB SPB CMR NCC	MOV OR AND POPW	ILM,#imm8 CCR,#imm8 CCR,#imm8 PS

● Hardware interrupt inhibition during execution of software interrupt

When a software interrupt is started, the interrupt enable flag (CCR: I) is cleared to 0 and the interrupt is disabled.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.6 Operation of Hardware Interrupt

The operation from the generation of hardware interrupt request to the completion of interrupt processing is explained below.

■ Start of Hardware Interrupt

● Operation of resource (generation of interrupt request)

The resources with a hardware interrupt request function have an interrupt request flag indicating the generation of an interrupt request, as well as an interrupt enable flag selecting between enabling and disabling an interrupt request. The interrupt request flag is set when events inherent to resources occur. When the interrupt enable flag is set to "enabled", an interrupt request is generated to the interrupt controller.

● Operation of interrupt controller (control of interrupt request)

The interrupt controller compares the interrupt level (ICR: IL2 to IL0) of simultaneously generated interrupt requests, selects the request with the highest level (with the smallest IL setting value), and posts it to the CPU. If there are two or more interrupt requests with the same level, the interrupt request with the smallest interrupt number is preferred.

et4U.com DataShee

● Operation of CPU (interrupt request acceptance and interrupt processing)

The CPU compares the received interrupt level (ICR: IL2 to IL0) with the value of the interrupt level mask register (ILM) and generates an interrupt processing microcode after end of the current instruction execution if the interrupt level (IL) is smaller than the value of the interrupt level mask register (ILM) and an interrupt is enabled (CCR: I = 1).

The EI^2OS enable bit (ICR: ISE) is referenced at the beginning of the interrupt processing microcode. When the EI^2OS enable bit (ICR: ISE) is set to 0, ordinary interrupt processing is performed. If the bit is set to 1, the EI^2OS starts.

At interrupt processing, the values of the dedicated registers (A, DPR, ADB, DTB, PCB, PC and PS) are saved in the system stack (system stack space indicated by SSB and SSP) first.

Next, the address of the vector table corresponding to the generated interrupt is loaded to the program counter (PCB, PC), the interrupt level mask register (ILM) is updated, and the stack flag (CCR: S) is set to 1.

■ Return from Hardware Interrupt

When the interrupt processing program clears the interrupt request flag in the resource that causes the interrupt to execute the RETI instruction, the values of the dedicated registers saved in the system stack are returned to each register and the operation returns to the processing executed before transition to interrupt processing.

The interrupt request output to the interrupt controller by the resource is cleared by clearing the interrupt request flag.

DataSheet4U.com www.DataSheet4U.com

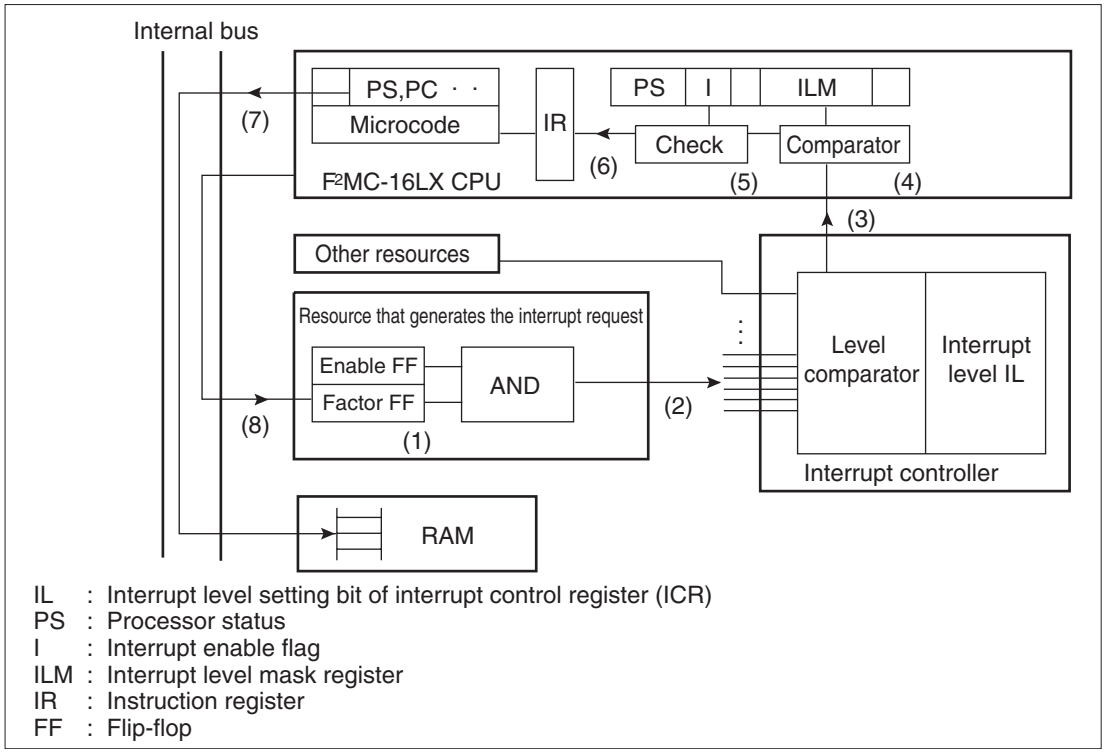
DataSheet4U.com

www.DataSheet4U.com

■ Operation of Hardware Interrupt

Figure 3.5-6 shows the operation from the generation of hardware interrupt to the completion of interrupt processing.

Figure 3.5-6 Operation of Hardware Interrupt



1. The resource generates an interrupt request.
2. When the interrupt enable bit in the resource is set to "enabled", the resource generates an interrupt request to the interrupt controller.
3. The interrupt controller that is received the interrupt request determines the priority of interrupts simultaneously requested and posts the interrupt level (IL) corresponding to the appropriate interrupt request to the CPU.
4. The CPU compares the interrupt level (IL) requested from the interrupt controller with the value of the interrupt level mask register (ILM).
5. If the interrupt request is preferred to the interrupt level mask register (ILM), the interrupt enable flag (CCR: I) is checked.
6. When an interrupt is enabled by the interrupt enable flag (CCR: I = 1), the requested interrupt level (IL) is set to the interrupt level mask register (ILM) after completion of the current instruction execution.
7. The values of the dedicated registers are saved, and processing transits to interrupt processing.
8. The program clears the interrupt request generated from the resource and executes the interrupt return instruction (RETI) to terminate interrupt processing.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

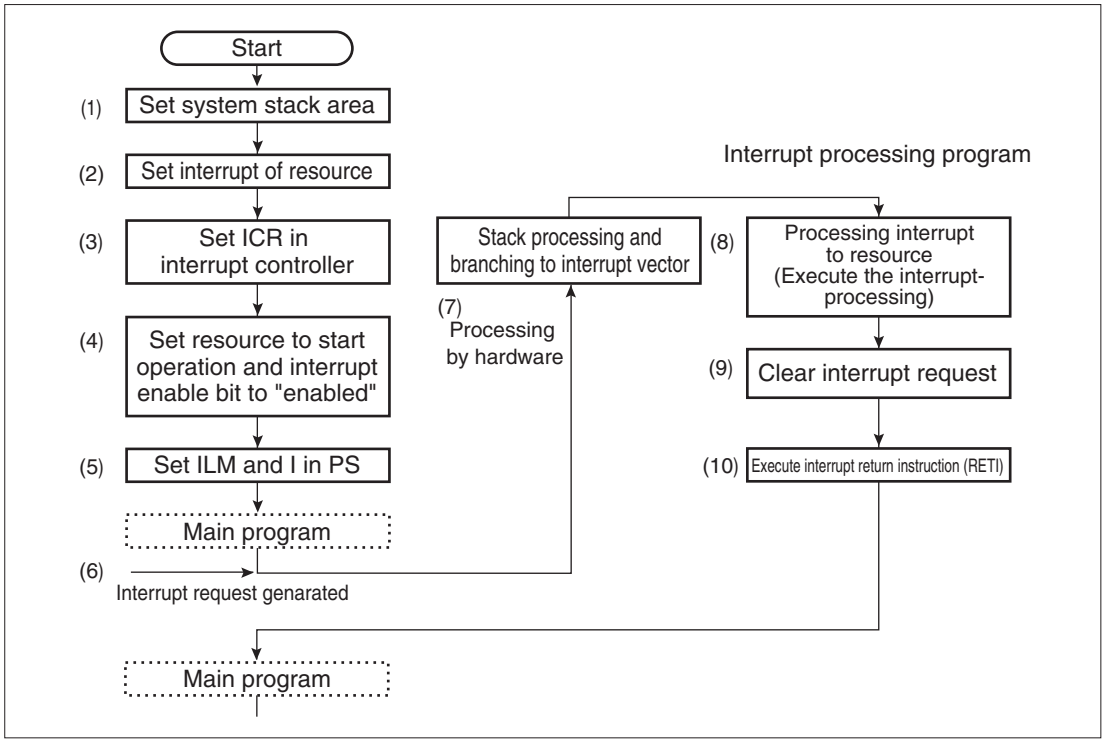
3.5.7 Procedure for Use of Hardware Interrupt

The settings of the system stack area, resources, interrupt control registers (ICR) are required for using the hardware interrupt.

■ Procedure for Use of Hardware Interrupt

Figure 3.5-7 shows an example of the procedure for use of the hardware interrupt.

Figure 3.5-7 Procedure for Use of Hardware Interrupt



1. Set the system stack area.
2. Set an interrupt of the resource with the interrupt request function.
3. Set the interrupt control register (ICR) in the interrupt controller.
4. Set the resource to start operation and the interrupt enable bit to "enabled".
5. Set the interrupt level mask register (ILM) and the interrupt enable flag (CCR: I) ready to accept an interrupt (CCR: I = 1).
6. An interrupt generated from the resource generates a hardware interrupt request.
7. The interrupt controller saves data in the dedicated registers, and processing transits to interrupt processing.
8. Execute the program for interrupt generation at interrupt processing.
9. Clear the interrupt request from the resource.
10. Execute the interrupt return instruction (RETI) to return to the program executed before transition to interrupt processing.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.8 Multiple Interrupts

Multiple hardware interrupts can be generated by setting different interrupt levels in the interrupt level setting bits of the interrupt control register (ICR: ILO to IL2) in response to plural interrupt requests from the resource. However, multiple EI^{2}OS cannot be started.

■ Multiple Interrupts

● Operation of multiple interrupts

If an interrupt request with a higher priority than the interrupt level of the current interrupt processing is generated during interrupt processing, the current interrupt processing is suspended to accept the generated higher-level interrupt request. When the higher-level interrupt processing is terminated, the suspended interrupt processing is resumed. The interrupt level (IL) can be set to 0 to 7. The interrupt request set to level 7 is never accepted.

If an interrupt request with a priority equal to or lower than the interrupt level of the current-executing interrupt is generated during interrupt processing, unless the setting of the interrupt enable flag (CCR: I) or the interrupt level mask register (ILM) is changed, the new interrupt request is held until the current interrupt processing is completed.

Starting of multiple interrupts generated during interrupt processing can be disabled temporarily by setting the interrupt enable flag (CCR: I) to "disabled" (CCR: 1= 0) or the interrupt level mask register (ILM) to "disabled" (ILM = 000).

Note: Multiple EI^{2}OS cannot be started. During EI^{2}OS processing, other interrupt requests and other EI^{2}OS requests are all held.

DataSheet4U.com www.DataSheet4U.com

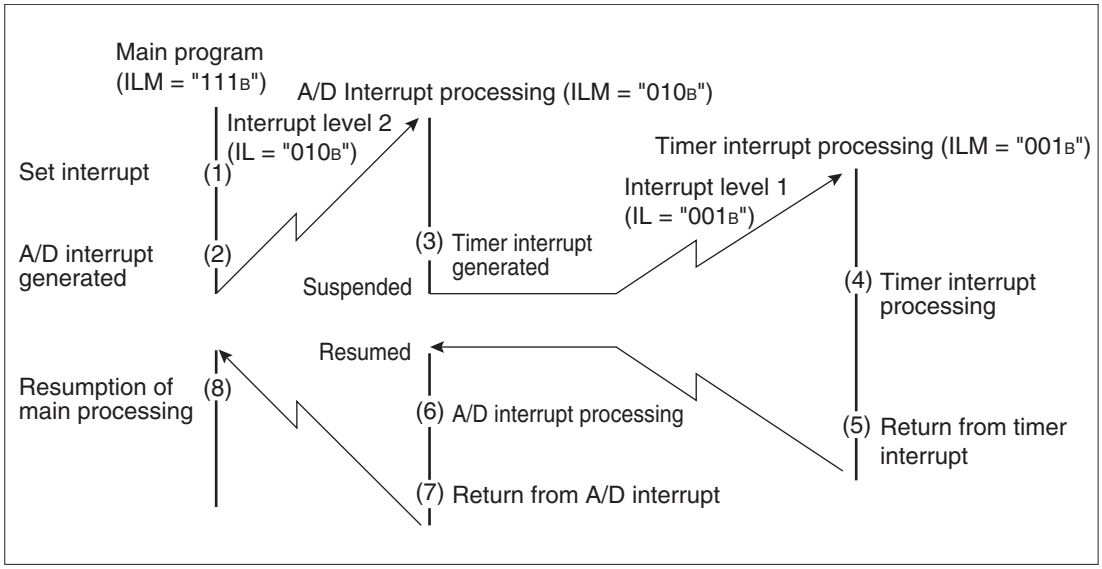
DataSheet4U.com

www.DataSheet4U.com

● Example of multiple interrupts

As an example of multiple interrupt processing, assuming that a timer interrupt is preferred to an A/D converter interrupt, set the interrupt level of the A/D converter to 2 and the interrupt level of the timer to 1. Figure 3.5-8 shows the processing of the timer interrupt generated during processing of the A/D converter interrupt.

Figure 3.5-8 Example of Multiple Interrupts



et4U.com
DataShee
DataSheet4U.com

- When processing of the A/D converter interrupt is started, the interrupt level mask register (ILM) is automatically set to the value (2 in example) of the interrupt level (ICR: IL2 to IL0) of the A/D converter. When an interrupt request with an interrupt level of 1 or 0 is generated under this condition, processing the generated interrupt is preferred.
- When the interrupt return instruction (RETI) is executed after the completion of interrupt processing, the values of the dedicated registers (A, DPR, ADB, DTB, PCB, PC, PS) saved in the system stack are returned to each register and the interrupt level mask register (ILM) is returned to the value before interrupt processing was suspended.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.9 Software Interrupt

The software interrupt is a function for transiting control from the current-executing program to the interrupt processing program defined by user by execution of a software interrupt instruction (INT instruction). The hardware interrupt is held during execution of a software interrupt.

■ Start and Operation of Software Interrupt

● Start of software interrupt

A software interrupt is started by executing the INT instruction. It does not have an interrupt request flag or an interrupt enable flag. An interrupt request is generated immediately after the INT instruction is executed.

● Hardware interrupt inhibition

Interrupts by the INT instruction have no interrupt level and the interrupt level mask register (ILM) is not updated. During execution of the INT instruction, the interrupt enable flag (CCR: I) is set to 0 and a hardware interrupt is masked.

When enabling a hardware interrupt during software interrupt processing, set the interrupt enable flag (CCR: I) to 1 during software interrupt processing.

et4U.com DataShee DataSheet4U.com

● Operation of software interrupt

When the INT instruction is executed, the software interrupt processing microcode in the CPU is started. The software interrupt processing microcode saves the values of the dedicated registers in the system stack; branching to the address of the corresponding interrupt vector table after a hardware interrupt is masked (CCR: I = 0).

■ Return from Software Interrupt

When the interrupt return instruction (RETI) is executed in the interrupt processing program, the values of the dedicated registers saved in the system stack are returned to each register and the operation is returned to the processing performed before branching to interrupt processing.

Note: When the program bank register (PCB) is "FF_H", the vector area for the CALLV instruction overlaps the table for the INT #vct8 instruction. A CALLV and INT #vct8 instructions can not use the same address in creating a software.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.10 Interrupt by EI^{2}OS

EI^{2}OS is a function to automatically transfer data between the resources (I/O) and memory. It generates the hardware interrupt at termination of data transfer.

■ EI^{2}OS

The EI^{2}OS provides automatic data transfer between the I/O area and memory. When data transfer is terminated, the termination factor (end condition) is set, branching automatically to the interrupt processing routine. Data can be transferred just by creating a setup program for starting the EI^{2}OS and an end program.

● Advantages of EI^{2}OS

Compared to data transfer using the interrupt-processing routine, EI^{2}OS has the following advantages.

- Since the creation of transfer program is not required, the program size can be reduced.
- The transfer count can be set to prevent transfer of unnecessary data.
- Whether to update the buffer address pointer can be specified.
- Whether to update the I/O address pointer can be specified.

et4U.com DataShee

● Interrupt by EI^{2}OS termination

DataSheet4U.com

At completion of data transfer by the EI^{2}OS, the end condition is set in the EI^{2}OS status bits (ICR: S1, S0), and then the processing automatically transits to interrupt processing.

The EI^{2}OS termination factor can be determined by checking the EI^{2}OS status bits (ICR: S1, S0) using the interrupt processing program.

● Interrupt control register (ICR)

This register is within the interrupt controller, and displays the states at starting, setting channel, and terminating the EI^{2}OS.

● EI^{2}OS descriptor (ISD)

The EI^{2}OS descriptor (ISD), which is allocated between "000100_H" and "$00017F_H$" in internal RAM, is 8-byte data that is used to set the transfer mode, addresses, transfer count and buffer addresses. It has 16 channels, and a channel number is allocated to each of these channels by the interrupt control register (ICR).

Note: The CPU stops while the EI^{2}OS is in operation.

DataSheet4U.com www.DataSheet4U.com

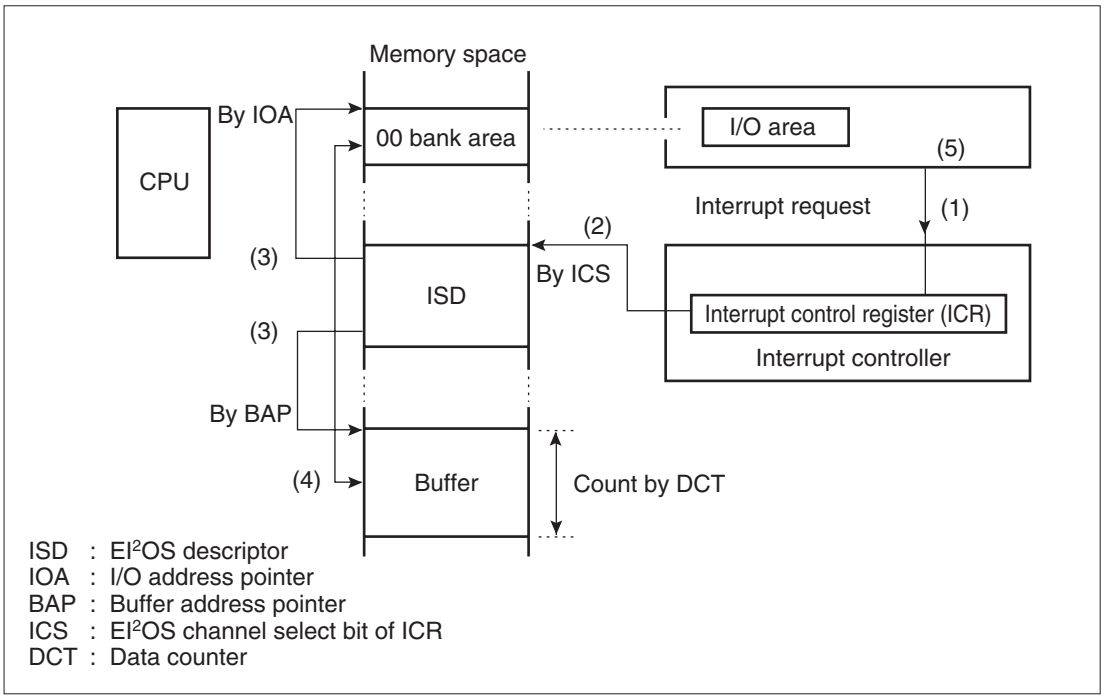
DataSheet4U.com

www.DataSheet4U.com

■ Operation of EI^{2}OS

Figure 3.5-9 shows the operation of the EI^{2}OS.

Figure 3.5-9 Operation of EI^{2}OS



1. An interrupt request is generated and the EI^{2}OS is started.
2. The interrupt controller selects the EI^{2}OS descriptor.
3. The transfer-source and transfer-destination address pointers are read from the EI^{2}OS descriptor.
4. Data is transferred according to the transfer-source and transfer-destination address pointers.
5. An interrupt factor is cleared automatically.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

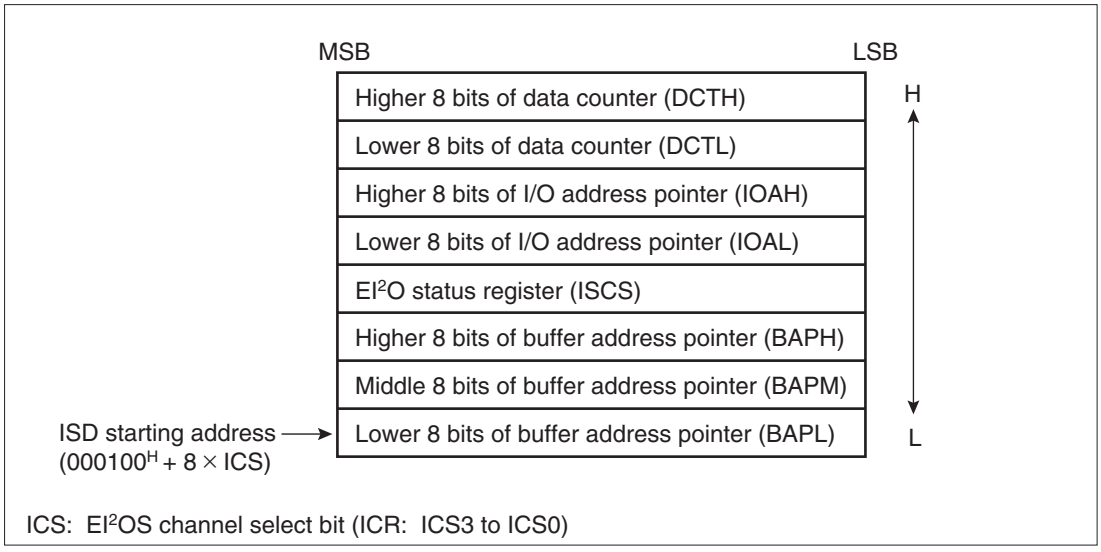
3.5.11 EI^{2}OS Descriptor (ISD)

The EI^{2}OS descriptor (ISD) is allocated to the addresses "000100$_H$" to "00017F$_H$" in the internal RAM, and consists of 8 bytes x 16 channels.

■ Configuration of EI^{2}OS Descriptor (ISD)

ISD consists of 8 bytes x 16 channels, and each ISD is composed as shown in Figure 3.5-10. Table 3.5-9 shows the correspondence between the channel number and ISD address.

Figure 3.5-10 Configuration of EI^{2}OS Descriptor (ISD)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 3.5-9 EI^{2}OS Descriptor (ISD) Area

Channel (ICR: ICS3 to ICS0)	Descriptor Starting Address
0	000100_H
1	000108_H
2	000110_H
3	000118_H
4	000120_H
5	000128_H
6	000130_H
7	000138_H
8	000140_H
9	000148_H
10	000150_H
11	000158_H
12	000160_H
13	000168_H
14	000170_H
15	000178_H

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

www.DataSheet4U.com

3.5.12 Each Register of EI^{2}OS Descriptor (ISD)

The EI^{2}OS descriptor (ISD) consists of the following registers.

- **Data counter (DCT)**
- **I/O address pointer (IOA)**
- **EI^{2}OS status register (ISCS)**
- **Buffer address pointer (BAP)**

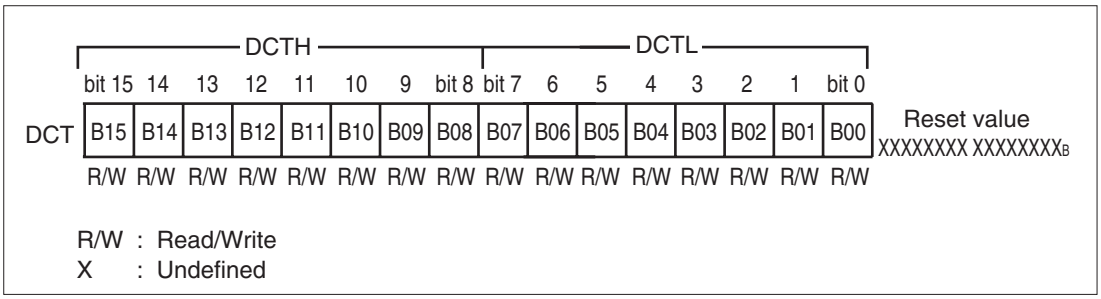
The reset value of each register is undefined and a reset should be performed carefully.

■ Data Counter (DCT)

The data counter (DCT) is a 16-bit register, and corresponds to the transfer data count. It decrements by one each time data is transferred. When the data counter (DCT) reaches 0, the EI^{2}OS is terminated and then the processing transits to interrupt processing.

Figure 3.5-11 shows the bit configuration of the data counter (DCT).

Figure 3.5-11 Configuration of Data Counter (DCT)

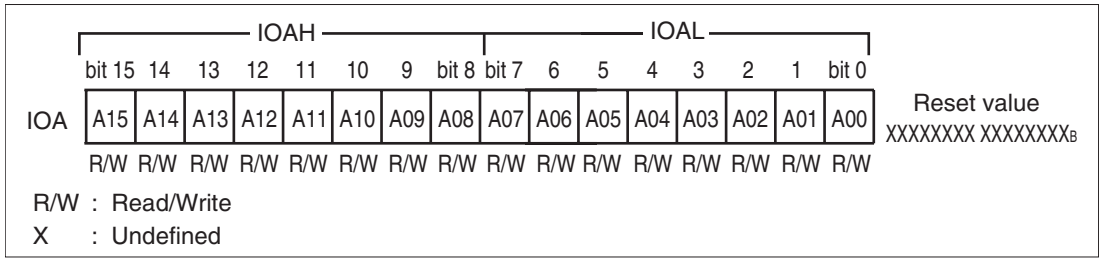


■ I/O Address Pointer (IOA)

The I/O address pointer (IOA) is a 16-bit register that sets the low addresses (A15 to A0) of the 00 bank area where data is transferred to or from the buffer. The high addresses (A23 to A16) are set all to 0 and the area between "000000_H" and "$00FFFF_H$" can be addressed.

Figure 3.5-12 shows the bit configuration of I/O address pointer (IOA).

Figure 3.5-12 Configuration of I/O Address Pointer (IOA)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ EI^{2}OS Status Register (ISCS)

The EI^{2}OS status register (ISCS) is an 8-bit register that sets the method to update the buffer address pointer and I/O address pointer, transfer data format (byte/word), and transfer direction.

Figure 3.5-13 shows the bit configuration of the EI^{2}OS status register (ISCS).

Figure 3.5-13 Configuration of EI^{2}OS Status Register (ISCS)

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value
XXXXXXXX$_B$

bit 0

SE	EI^{2}OS termination control bit
0	Not terminated by a request from resource
1	Terminated by a request from resource

bit 1

DIR	Data transfer direction specify bit
0	I/O address pointer → Buffer address pointer
1	Buffer address pointer → I/O address pointer

bit 2

BF	BAP updating/fixing select bit
0	Buffer address pointer updated after data transfer *1
1	Buffer address pointer not updated after data transfer

bit 3

BW	Transfer data length specify bit
0	Byte
1	Word

bit 4

IF	IOA updating/fixing select bit
0	I/O address pointer updated after data transfer *2
1	I/O address pointer not updated after data transfer

bit 7	bit 6	bit 5	
Reserved	Reserved	Reserved	Reserved bits
0	0	0	Always write 0

R/W : Read/Write
X : Undefined
*1 : The buffer address pointer changes only in lower 16 bits, and can only be incremented.
*2 : I/O address pointer can only be incremented.

DataSheet4U.com
www.DataSheet4U.com

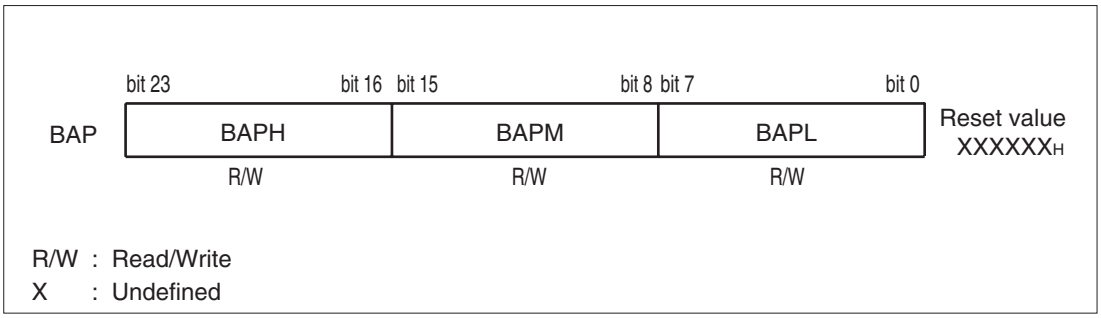
DataSheet4U.com

www.DataSheet4U.com

■ Buffer Address Pointer (BAP)

The buffer address pointer (BAP) is a 24-bit register and sets the 16-MB addresses where data is transferred to or from I/O area. When the BAP updating/fixing select bit of the EI^{2}OS status register (ISCS: BF) is set to "updated", the buffer address pointer (BAP) changes only in the lower 16 bits (BAPH, BAPL) and does not change in the higher 8 bits (BAPH). Figure 3.5-14 shows the bit configuration of the buffer address pointer (BAP).

Figure 3.5-14 Configuration of Buffer Address Pointer (BAP)



References:
- The area that can be set by the I/O address pointer (IOA) is "000000_H" to "$00FFFF_H$".
- The area that can be set by the buffer address pointer (BAP) is "000000_H" to "$FFFFFF_H$".
- The maximum transfer count that can be set by the data counter (DCT) is 65,536.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

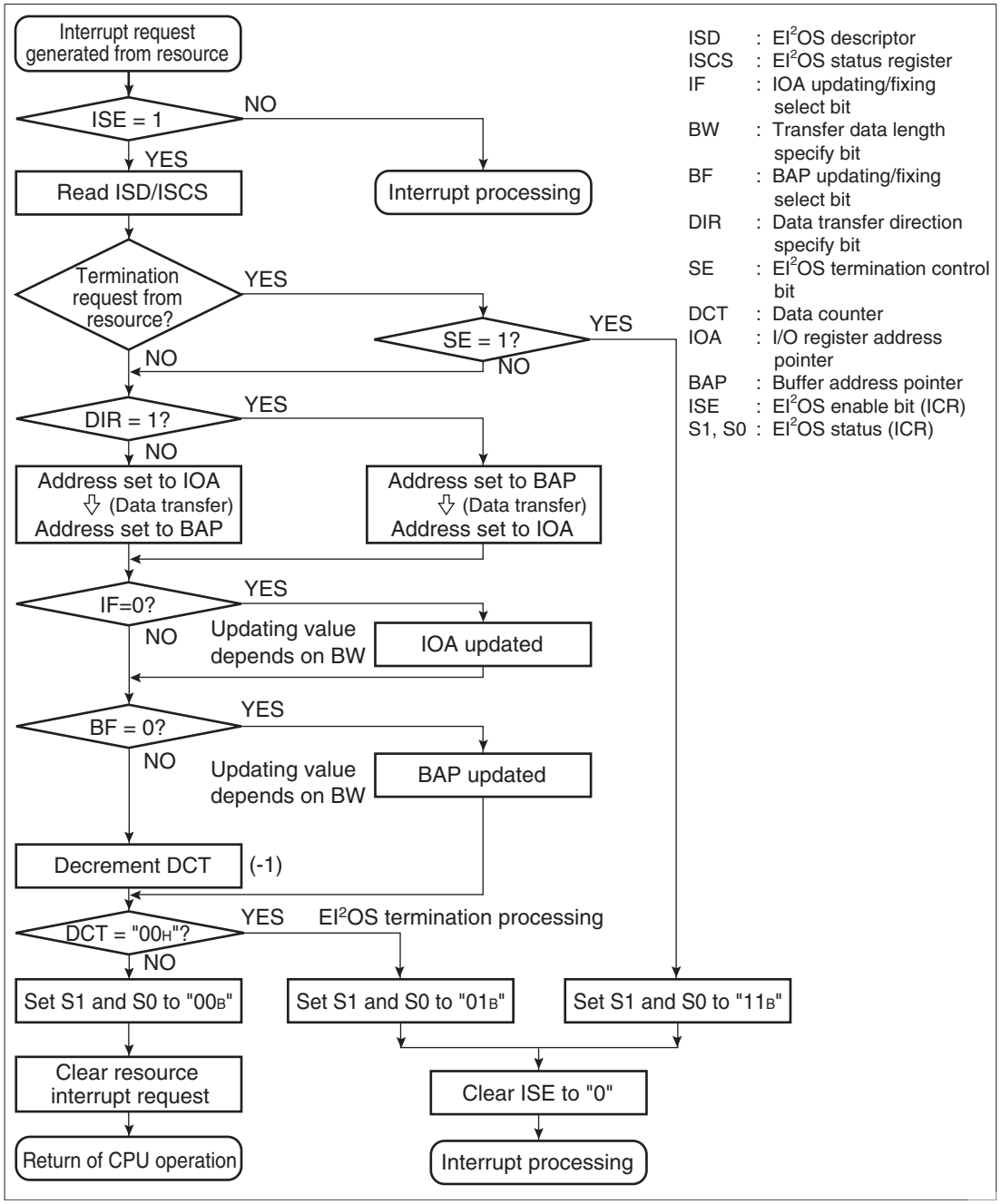
www.DataSheet4U.com

3.5.13 Operation of EI²OS

The flowchart of operation of the EI²OS using the microcode in the CPU is shown below:

■ Operation of EI²OS

Figure 3.5-15 Flowchart of Operation of EI²OS



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

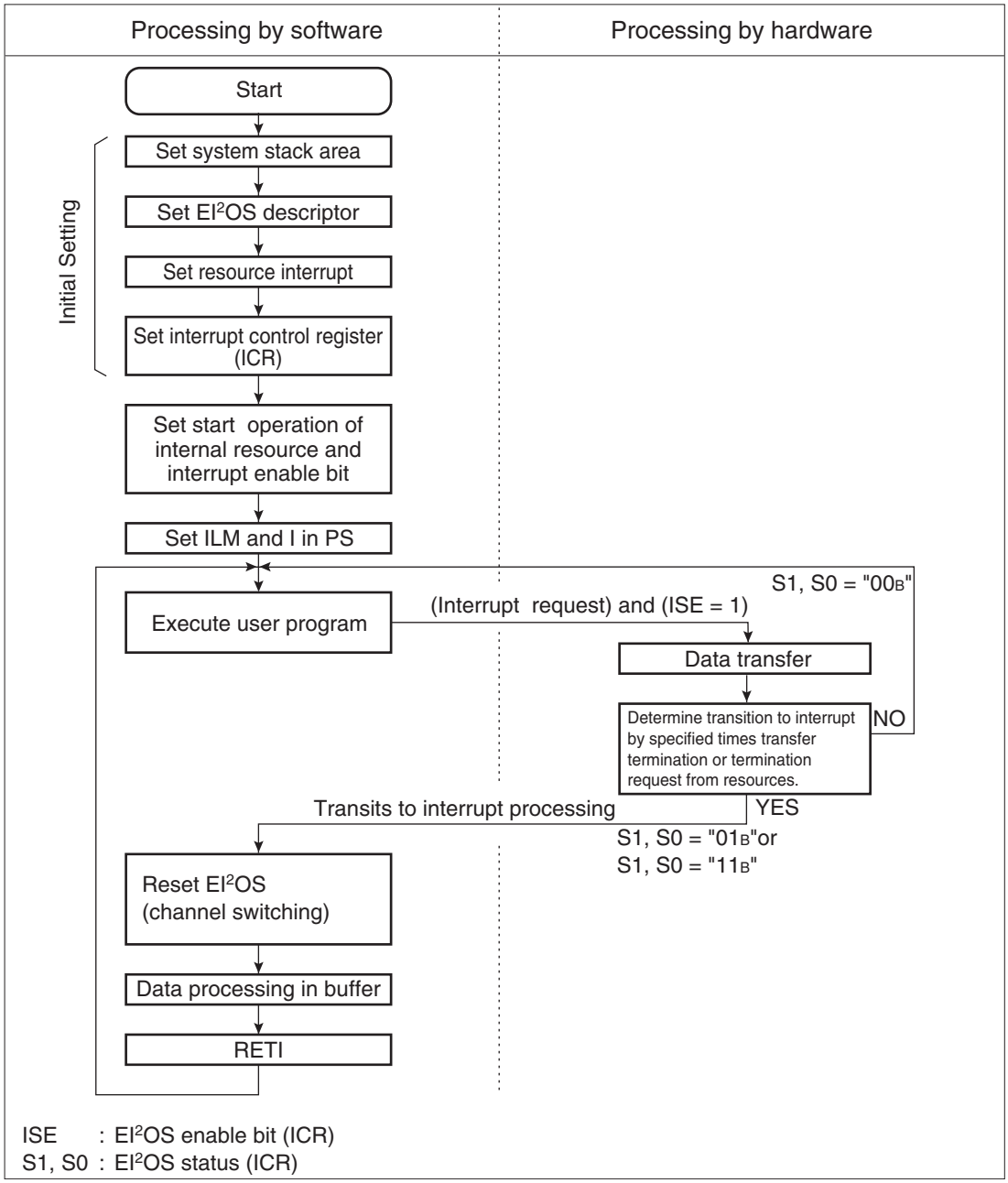
www.DataSheet4U.com

3.5.14 Procedure for Use of EI^{2}OS

The procedure for using the EI^{2}OS is shown below:

■ Procedure for Use of EI^{2}OS

Figure 3.5-16 Procedure for Use of EI^{2}OS



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.15 EI^{2}OS Processing Time

The time required for EI^{2}OS processing depends on the following factors:

- **Setting of EI^{2}OS status register (ISCS)**
- **Data length of transfer data**

Some interrupt handling time is required at the transition to hardware interrupt processing after completion of data transfer.

■ EI^{2}OS Processing Time (time for one transfer)

● At continuing data transfer (DCT ≠ 0, ISCS: SE=0)

The EI^{2}OS processing time at continuing data transfer is determined by the setting of the EI^{2}OS status register (ISCS) as shown in Table 3.5-10.

Table 3.5-10 EI^{2}OS Execution Time

Setting of EI^{2}OS Termination Control Bit (SE)		Termination by Termination Request from Resource		Ignores Termination Request from Resource	
Setting of IOA Updating/Fixing Select Bit (IF)		Fixed	Updated	Fixed	Updated
Setting of BAP address updating/fixing select bit (BF)	Fixed	32	34	33	35
	Updated	34	36	35	37

Unit: Machine cycle (one machine cycle is equal to one clock cycle of the machine clock (ϕ).)

In addition, compensation is required depending on the conditions at executing EI^{2}OS as shown in Table 3.5-11.

Table 3.5-11 Compensation Value for Data Transfer at EI^{2}OS Processing Time

I/O Register Address Pointer			Internal Access	
			B/even	Odd
Buffer address pointer	Internal access	B/even	0	+2
		Odd	+2	+4

B: Byte data transfer
Even: Word transfer at even address
Odd: Word transfer at odd address

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● At end of data counter (DCT) (DCT ≠ 0, ISCS: SE=0)

At completion of data transfer by the EI^{2}OS, since the hardware interrupt is started, the interrupt handling time is added. The EI^{2}OS processing time at the end of counting is calculated by the following expression.

$$\text{EI}^2\text{OS processing time at end of counting} = \text{EI}^2\text{OS processing time at continuing data transfer} + \underbrace{(21 + 6 \times Z)}_{\text{Interrupt handling time}} \text{ machine cycles}$$

(Z: Compensation value of interrupt handling time)

The interrupt handling time depends on the address set by the stack pointer. Table 3.5-12 shows the compensation value (Z) of the interrupt handling time.

Table 3.5-12 Compensation Value (Z) of Interrupt Handling Time

Address Set by Stack Pointer	Compensation Value (Z)
For internal area (even address)	0
For internal area (odd address)	+2

● At termination by termination request from resource (DCT ≠ 0, ISCS=1)

If data transfer by the EI^{2}OS is terminated during its processing by the termination request from a resource (ICR: S1, S0 = "11_B"), processing transits to interrupt processing. The EI^{2}OS processing time at a termination request from a resource is calculated as follows:

$$\text{EI}^2\text{OS processing time at termination during processing} = 36 + 6 \times Z \text{ machine cycles}$$

(Z: Compensation value of interrupt handling time)

Reference: One machine cycle is equal to one clock cycle of the machine clock (ϕ).

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.16 Exception Processing Interrupt

The F^{2}MC-16LX family performs exception processing when an undefined instruction is executed.
Exception processing is basically the same as interrupt. When an exception is detected between instructions, normal processing is suspended to perform exception processing.
Exception processing is performed when an unexpected operation is performed, and should be used only for starting recovery software at debugging or in an emergency.

■ Exception Processing

● Operation of exception processing

The F^{2}MC-16LX family treats all instruction codes not defined in the instruction map as undefined instructions. If an undefined instruction is executed, the processing equal to the software interrupt instruction INT # 10 is performed.

At exception processing, the following processing is performed before the transition to interrupt processing:

- The values of dedicated registers (A, DPR, ADB, DTB, PCB, PC, PS) saved to the system stack
- The interrupt enable flag (CCR: I) cleared to 0 and an interrupt disabled
- The stack flag (CCR: S) set to 1

The value of the program counter (PC) saved in the stack is a value of the address where undefined instructions are stored. For instruction codes of 2 bytes or more, the value of the program counter (PC) is a value of the address where instruction codes that can be identified as undefined are stored. When the type of exception factor must be determined at exception processing, use the saved program counter (PC).

● Return from exception processing

When the program counter (PC) indicates an undefined instruction, the interrupt return instruction (RETI) from exception processing is executed to return to exception processing. Some measures such as performing a software reset should be taken when returning from exception processing.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

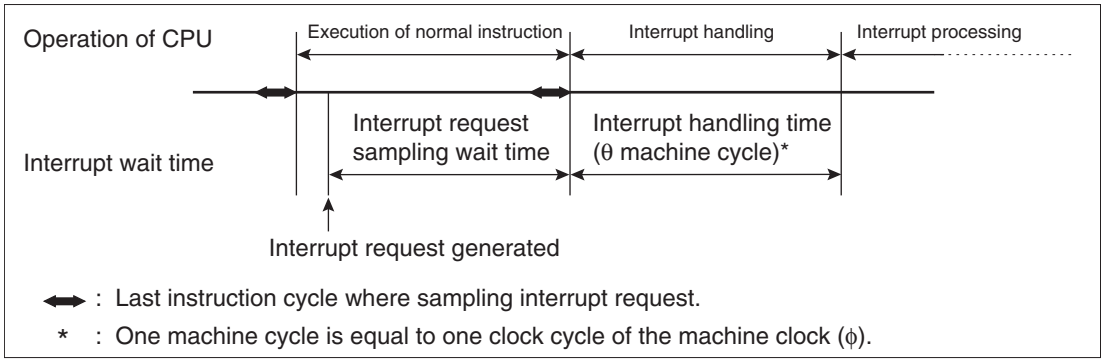
3.5.17 Time Required to Start Interrupt Processing

The time for terminating the currently executing instruction plus the interrupt handling time is required from generation of the hardware interrupt request to execution of the interrupt-processing.

■ Time Required to Start Interrupt Processing

The interrupt request sampling wait time and the interrupt handling time (time required for preparation for interrupt processing) are required from generation of the interrupt request and acceptance of interrupt, to execution of the interrupt processing. Figure 3.5-17 shows the interrupt processing time.

Figure 3.5-17 Interrupt Processing Time



● Interrupt request sampling wait time

It indicates a time from the generation of the interrupt request to the termination of the currently executing instruction.

Whether the interrupt request is generated or not is determined by sampling the interrupt request in the last cycle of each instruction. The CPU cannot recognize the interrupt request during execution of each instruction, as a result wait time occurs.

Reference:	The interrupt request sampling wait time is longest when the interrupt request is generated immediately after starting execution of the POPW, RW0, …RW7 instructions with the longest execution cycle (45 machine cycles).

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Interrupt handling time (θ machine cycles)

The CPU requires an interrupt handling time of θ machine cycles to save the dedicated registers to the system stack and fetch the interrupt vector table address after accepting the interrupt request. The interrupt handling time (θ) is obtained using the following equations.

θ = 24 + 6 x Z machine cycles (Z: compensation value of interrupt handling time)

The interrupt handling time depends on the address set by the stack pointer. Table 3.5-13 shows the compensation value (Z) of the interrupt handling time.

Table 3.5-13 Compensation Value (Z) of Interrupt Handling Time

Address Set by Stack Pointer	Compensation Value (Z)
For internal area (even address)	0
For internal area (odd address)	+2

Reference: One machine cycle is equal to one clock cycle of the machine clock (ϕ).

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.18 Stack Operation for Interrupt Processing

When an interrupt is accepted, the values of dedicated registers are automatically saved to the system stack before transition to interrupt processing. At completion of interrupt processing, the values of the dedicated registers are automatically returned from the system stack.

■ Stack Operation at Starting Interrupt Processing

When an interrupt is accepted, the CPU automatically saves the values of the current-dedicated registers in the system stack in the following order.

- Accumulator (AH, AL)
- Direct page register (DPR)
- Additional data bank register (ADB)
- Data bank register (DTB)
- Program bank register (PCB)
- Program counter (PC)
- Processor status (PS)

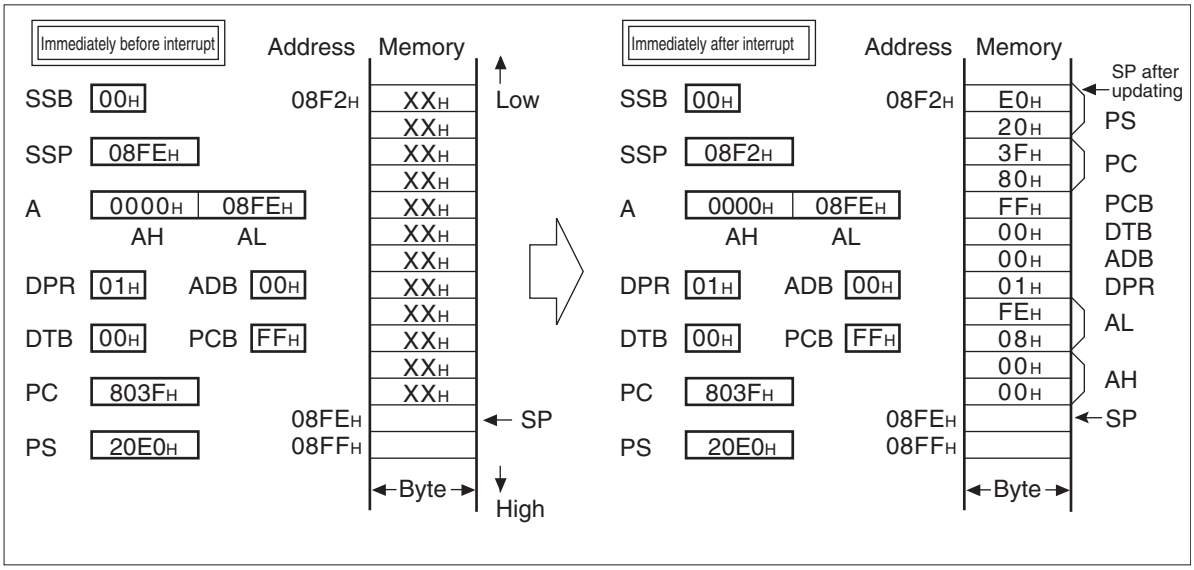
Figure 3.5-18 shows the stack operation at starting interrupt processing.

et4U.com

DataShee

DataSheet4U.com

Figure 3.5-18 Stack Operation at Starting Interrupt Processing



■ Stack Operation at Return from Interrupt Processing

When the interrupt return instruction (RETI) is executed after completion of interrupt processing, the values of the dedicated registers (PS, PC, PCB, DTB, ADB, DPR, AL, AH) are returned to each register from the system stack, and the dedicated registers are returned to the state before interrupt processing was started.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.5.19 Program Example of Interrupt Processing

This section gives a program example of interrupt processing.

■ Program Example of Interrupt Processing

● Processing specification

This is an example of interrupt program using external interrupt 4 (INT4).

● Coding example

```
DDR2    EQU  000012H           ; Port 2 direction register
ENIR    EQU  030H              ; Interrupt/DTP enable register
EIRR    EQU  031H              ; Interrupt/DTP flag
ELVR    EQU  032H              ; Request level setting register
ICR00   EQU  0B0H              ; Interrupt control register
STACK   SSEG                   ; Stack
RW      100
STACK_T RW   1
STACK   ENDS
;-----Main program-----------------------------------------------------------
CODE    CSEG
;
START:
        MOV  RP,#0             ; The general-purpose register uses the starting bank.
        MOV  ILM,#07H          ; ILM in PS set to level 7
        MOV  A,#!STACK_T       ; System stack set
        MOV  SSB,A
        MOVW A,#STACK_T        ; Stack pointer set
        MOVW SP,A              ; In this case, S flag = 1, so set to SSP
        MOV  DDR2,#00000000B   ; The P24/INT4 pin set to input.
        OR   CCR,#40H          ; I flag of CCR in PS set to interrupt enabled
        MOV  I:ICR00,#00H      ; Interrupt level 0 (highest)
        MOV  I:ELVR,#00010000B ; INT4 as an High level request
        MOV  I:EIRR,#00H       ; INT4 interrupt factor cleared
        MOV  I:ENIR,#10H       ; INT4 input enabled
         :
LOOP:   NOP                    ; Dummy loop
        NOP
        NOP
        NOP
        BRA  LOOP              ; Unconditional jump
```

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com


```
;-----Interrupt program-----------------------------------------------------
ED_INT1:
        MOV  I:EIRR,#00H       ; New acceptance of INT4 disabled
        NOP
        NOP
        NOP
        NOP
        NOP
        NOP
        RETI                   ; Return from interrupt
CODE    ENDS
;-----Vector setting--------------------------------------------------------
VECT    CSEG ABS = OFFH
        ORG  OFFDOH            ; Vector set to interrupt  #11 (OB_H)
        DSL  ED_INT1
        ORG  OFFDCH            ; Reset vector set
        DSL  START
        DB   00H               ; Set to single-chip mode
VECT    ENDS
        END   START
```

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Program Example of EI^{2}OS

● Processing specification

- The EI^{2}OS is started by detecting the High level of the signal to be input to the INT4 pin.
- When the High level is input to the INT4 pin, EI^{2}OS is started and the data of port 2 is transferred to memory address "3000_H".
- The transfer data is 100 bytes. After 100 bytes are transferred, an interrupt is generated at completion of transfer by the EI^{2}OS transfer.

● Coding example

```
DDR2    EQU  000012H           ; Port 2 direction register
ENIR    EQU  000030H           ; Interrupt/DTP enable register
EIRR    EQU  000031H           ; Interrupt/DTP factor register
ELVR    EQU  000032H           ; Request level setting register
ICR00   EQU  0000B0H           ; Interrupt control register
BAPL    EQU  000100H           ; Lower of buffer address pointer
BAPM    EQU  000101H           ; Middle of buffer address pointer
BAPH    EQU  000102H           ; Higher of buffer address pointer
ISCS    EQU  000103H           ; EI2OS status
IOAL    EQU  000104H           ; Lower of I/O address pointer
IOAH    EQU  000105H           ; Higher of I/O address pointer
DCTL    EQU  000106H           ; Lower of data counter
DCTH    EQU  000107H           ; Higher of data counter
ER0     EQU  EIRR:0            ; External interrupt request flag bit defined
STACK   SSEG                   ; Stack
        RW   100
STACK_T RW  1
STACK   ENDS
;-----Main program-----------------------------------------------------------
CODE   CSEG
START:
        AND  CCR,#0BFH         ; I flag of CCR in PS cleared to interrupt disabled
        MOV  RP,#00            ; Register bank pointer set
        MOV  A,#!STACK_T       ; System stack set
        MOV  SSB,A
        MOVW A,#STACK_T        ; Stack pointer
                               ; in this case, S flag = 1, so set to SSP
        MOVW SP,A
        MOV  I:DDR2,#00000000B ; P24/INT4 pin set to input
        MOV  BAPL,#00H         ; Buffer address set (003000H)
        MOV  BAPM,#30H
        MOV  BAPH,#00H
```

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com


```
        MOV  ISCS,#00010001B   ; I/O Address not updated, byte transfer performed,
                               ; and buffer address updated
                               ; Data transferred from I/O to buffer,
                               ; and termination by resource
        MOV  IOAL,#00H         ; Transfer source address set
                               ; (port 2:  0000002H)
        MOV  IOAH,#00H
        MOV  DCTL,#64H         ; Transfer byte count set (100 bytes)
        MOV  DCTH,#00H
        MOV  I:ICR00,#00001000B; EI2OS channel 0, EI2OS enabled,
                               ; and interrupt level 0 (highest)
        MOV  I:ELVR,#00010000B ; INT4 set as an High level request
        MOV  I:EIRR,#00H       ; INT4 interrupt factor cleared
        MOV  I:ENIR,#01H       ; INT4 interrupt enabled
        MOV  ILM,#07H          ; ILM in PS set to level 7
        OR   CCR,#40H          ; I flag of CCR in PS set to interrupt enabled
         :
LOOP:   BRA  LOOP              ; Infinite loop
;-----Interrupt program-----------------------------------------------------------
WARI    CLRB ER0               ; Interrupt/DTP request flag cleared
         :
        Processing by user     ; EI2OS termination factor checked,
         :                     ; data processing during buffering
                               ; EI2OS reset
        RETI
CODE    ENDS
;-----Vector setting--------------------------------------------------------------
VECT    CSEG ABS = OFFH
        ORG  OFFDOH            ; Vector set to interrupt  #11 (0BH).
        DSL  WARI
        ORG  OFFDCH            ; Reset vector set
        DSL  START
        DB   00H               ; Set to single-chip mode
VECT    ENDS
        END  START
```

et4U.com DataShee
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.6 Reset

When a reset factor occurs, the CPU immediately suspends the current processing and starts the reset operation.
The reset factors are as follows:

- **Power-on reset**
- **Overflow of watchdog timer**
- **Software reset request**
- **Generation of external reset request ($\overline{\text{RST}}$ pin)**

■ Reset Factors

Table 3.6-1 Reset Factor

Reset	Factor	Machine Clock	Watchdog Timer	Oscillation Stabilization Waiting
Power on reset	At power on	MCLK	Stops	Generated
Watchdog timer reset	Watchdog timer overflow	MCLK	Stops	None
Software reset	0 is written to the RST bit	MCLK	Stops	None
External reset	Input L level to $\overline{\text{RST}}$ pin	MCLK	Stops	None

MCLK: Main clock

● Power on reset

- The power on reset occurs at power on.
- The reset operation is executed after the oscillation stabilization wait time of 2^{18}/HCLK has elapsed.

● Watchdog timer reset

- Unless the watchdog timer is periodically cleared at the interval time to be repeatedly counted after starting, an overflow occurs, causing a reset.
- The oscillation stabilization wait time is not generated by a watchdog timer reset.

Reference: For the details of the watchdog timer, see "CHAPTER 6 WATCHDOG TIMER".

● Software reset

- The software reset occurs when 0 is written to the internal reset signal generate bit (LPMCR: RST) in the low-power consumption mode control register.
- The oscillation stabilization wait time is not generated by a software reset.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● External reset

- The external reset occurs when a Low level is input to the external reset pin ($\overline{RST}$ pin). The time for inputting Low level from the $\overline{RST}$ pin requires at least 16 machine cycles (main clock).
- An external reset does not require the oscillation stabilization wait time.

Notes:
- If an external reset request is generated from the $\overline{RST}$ pin during writing by a transfer instruction (such as MOV), the reset cancel wait state is set after completion of the transfer instruction, so writing is terminated normally. For a string instruction (such as MOVS), the reset cancel wait state may be set before completion of transfer by a specified counter value.
- When stop mode, sub-clock mode, sub-sleep mode and watch mode are returned to main clock mode using an external reset pin ($\overline{RST}$ pin), input level "L" for at least "the oscillation time of the oscillator(*) + 100μs + 16 machine cycles (main clock)".
  *: The oscillation time of the oscillator is the time required to reach 90% of amplitude. It takes several to dozens of ms for crystal oscillators, hundreds of μs to several ms for FAR/ceramic oscillators, and 0 ms for external clocks.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.6.1 Reset Factors and Oscillation Stabilization Wait Time

The oscillation stabilization wait time after reset varies depending on the reset factors.

■ Reset Sources and Oscillation Stabilization Wait Time

Table 3.6-2 Reset Factors and Oscillation Stabilization Wait Times

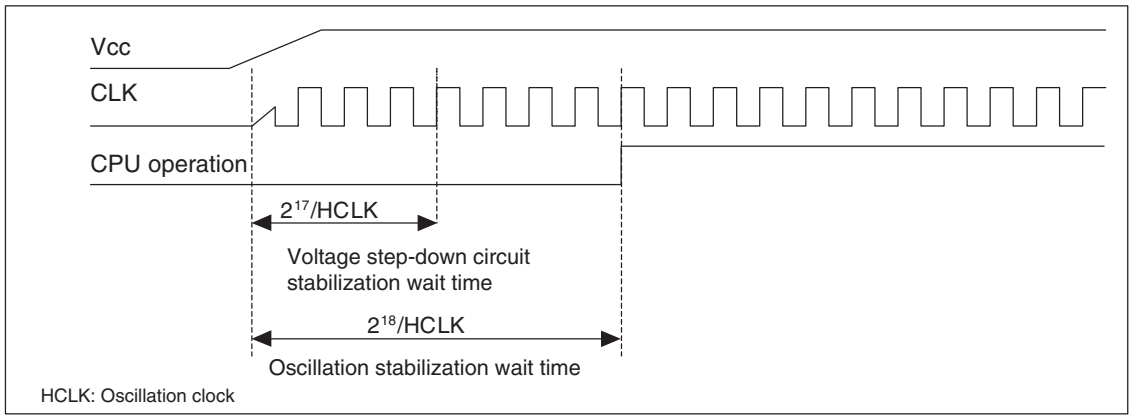
Reset Factor	Oscillation Stabilization Wait Time (The parenthesized value is a calculation example at 4 MHz oscillation clock frequency.)
Power on reset	2^{18}/HCLK (approx. 65.54 ms)
Watchdog reset	None
Software reset	None
External reset	None

HCLK: Oscillation clock

et4U.com DataShee

DataSheet4U.com

Figure 3.6-1 Oscillation Stabilization Wait time for the MB90385 Series during a Power-on Reset



DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 3.6-3 Oscillation Stabilization Wait Time by Clock Select Register (CKSCR)

Clock Select Bit		Oscillation Stabilization Wait Time (The parenthesized value is a calculation example at 4 MHz oscillation clock frequency.)
WS1	WS0	
0	0	2^{10}/HCLK (256 µs)
0	1	2^{13}/HCLK (approx. 2.048 ms)
1	0	2^{15}/HCLK (approx. 8.192 ms)
1	1	2^{17}/HCLK (approx. 32.77 ms) *

HCLK: Oscillation clock

*: At power on, the oscillation stabilization wait time is fixed at 2^{18}/HCLK (approximately 65.54 ms).

Note: Ceramic or crystal oscillators require the oscillation stabilization wait time of several milliseconds to some tens of milliseconds to stabilize oscillation. Set the oscillation stabilization wait time required for the oscillator to be used.

For the details of the clock, see "3.7 Clocks".

DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

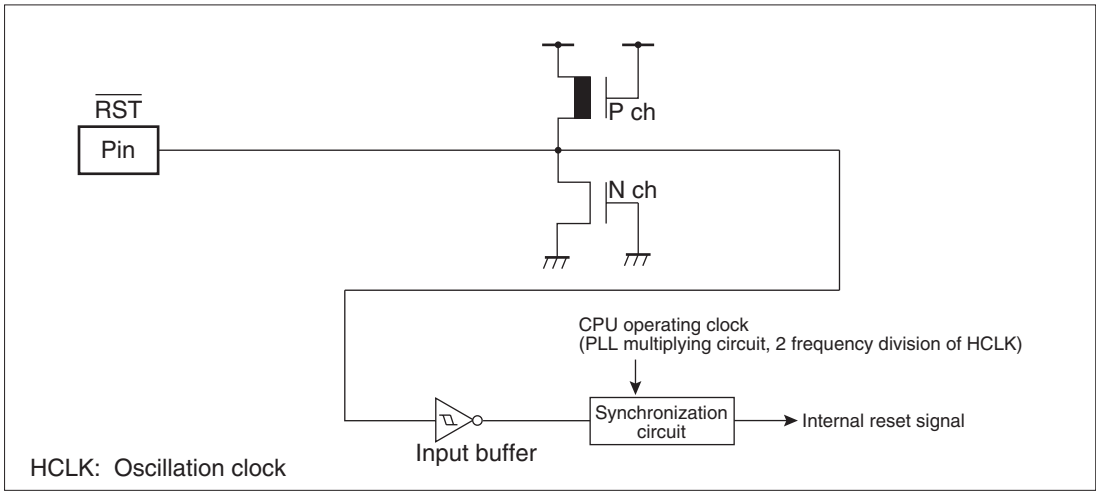
www.DataSheet4U.com

3.6.2 External Reset Pin

The external reset pin ($\overline{\text{RST}}$ pin) is a reset input pin. Input of an external Low level generates a reset factor. The MB90385 series starts the reset operation in synchronization between the CPU and clock.

■ Block Diagram of External Reset Pin

Figure 3.6-2 Block Diagram of External Reset Pin



Notes:
- To prevent damage to memory due to a reset during writing to memory, a Low level is input to the $\overline{\text{RST}}$ pin in a machine cycle in which memory is not damaged.
- The CPU operation clock is required to initialize internal circuits. In particular, at operation on an external clock, the reset signal and CPU operation clock signal must be input.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

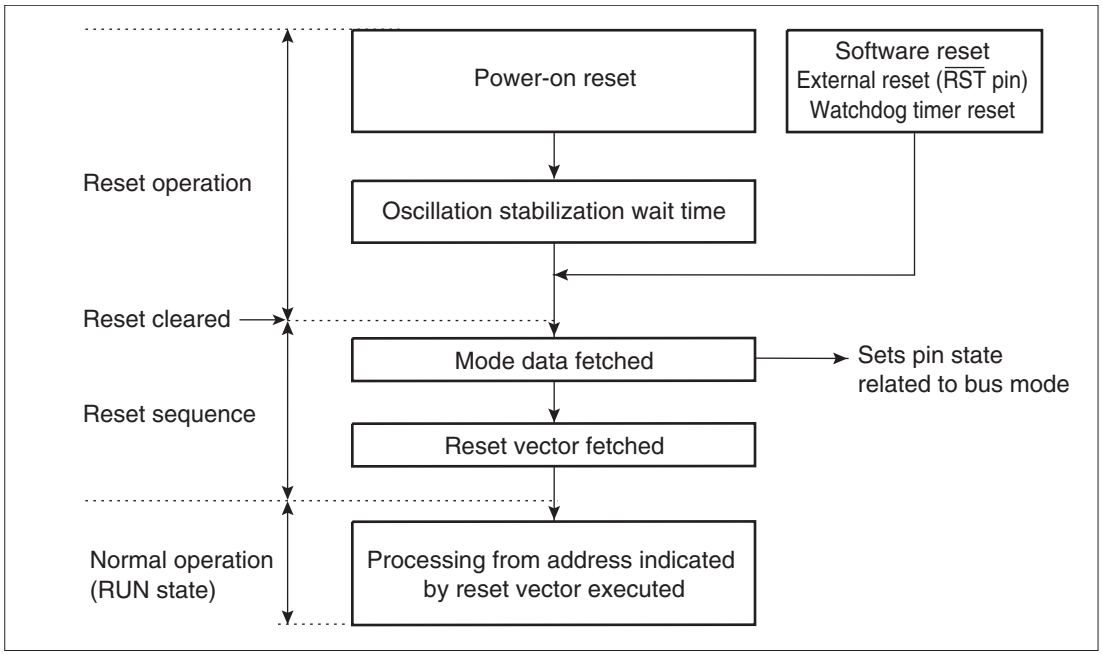
3.6.3 Reset Operation

During reset operation, the mode for reading mode data and reset vectors is set according to the settings of the mode pins (MD0 to MD2) and a mode fetch is executed. When the oscillation clock is returned from stop states (power on, stop mode) by a reset, a mode fetch is executed after the elapse of the main clock oscillation stabilization wait time.

■ Flowchart of Reset Operation

Figure 3.6-3 shows the flowchart of reset operation.

Figure 3.6-3 Flowchart of Reset Operation



■ Oscillation Stabilization Wait Time in Standby Mode

When a reset occurs during operation in a stop mode or subclock mode in which the oscillation clock is stopped, and oscillation stabilization wait time of 2^{17}/HCLK (approximately 32.77 ms when the oscillation clock operates at 4 MHz) is generated.

Reference:	For standby mode operation, see Section "3.8 Low-power Consumption Mode".

■ Mode Pin

The MD0 to MD2 mode pins are external pins. They are used to set the mode for reading data and reset vectors.

Reference:	For the details of the mode pins (MD0 to MD2), see Section "3.9.3 Memory Access Mode".

DataSheet4U.com
www.DataSheet4U.com

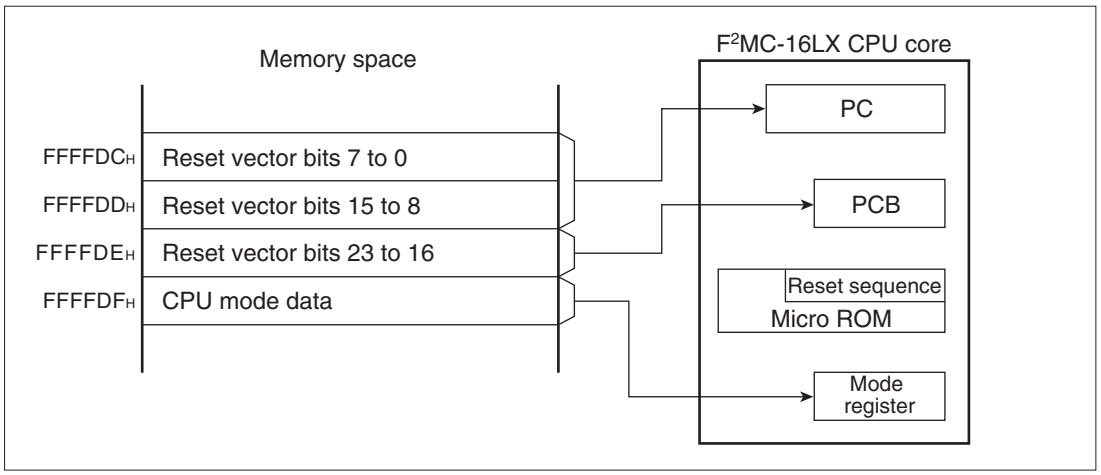
DataSheet4U.com

www.DataSheet4U.com

■ Mode Fetch

At transition to the reset operation, the CPU automatically transfers mode data and reset vectors by hardware to the appropriate register in the CPU core. The mode data and reset vector are allocated to four bytes of addresses "FFFFDC$_H$" to "FFFFDF$_H$". After a reset factor is generated (or after the elapse of the oscillation stabilization wait time), the CPU immediately outputs the addresses of the mode data and reset vectors to the bus to fetch the mode data and reset vectors. This operation is called "mode fetch." At completion of mode fetch, the CPU starts processing from the address indicated by the reset vector.

Figure 3.6-4 Transfer of Mode Data and Reset Vectors



et4U.com
DataShee
DataSheet4U.com

Note: To read the mode data and reset vectors from internal ROM is set by the mode pins (MD0 to MD2). For use in the single-chip mode, the mode pins should be set to the internal vector mode.

● Mode data

The mode data is used to set a memory access type or a memory access area after completion of the reset operation. It is allocated to address "FFFFDF$_H$". During the reset operation, this data is read automatically by a mode fetch and stored in the mode register.

● Reset vectors

The reset vectors are the start addresses of execution after completion of the reset operation. They are allocated to addresses "FFFFDC$_H$" to "FFFFDE$_H$". During the reset operation, these vectors are read automatically by a mode fetch and transferred to the program counter.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

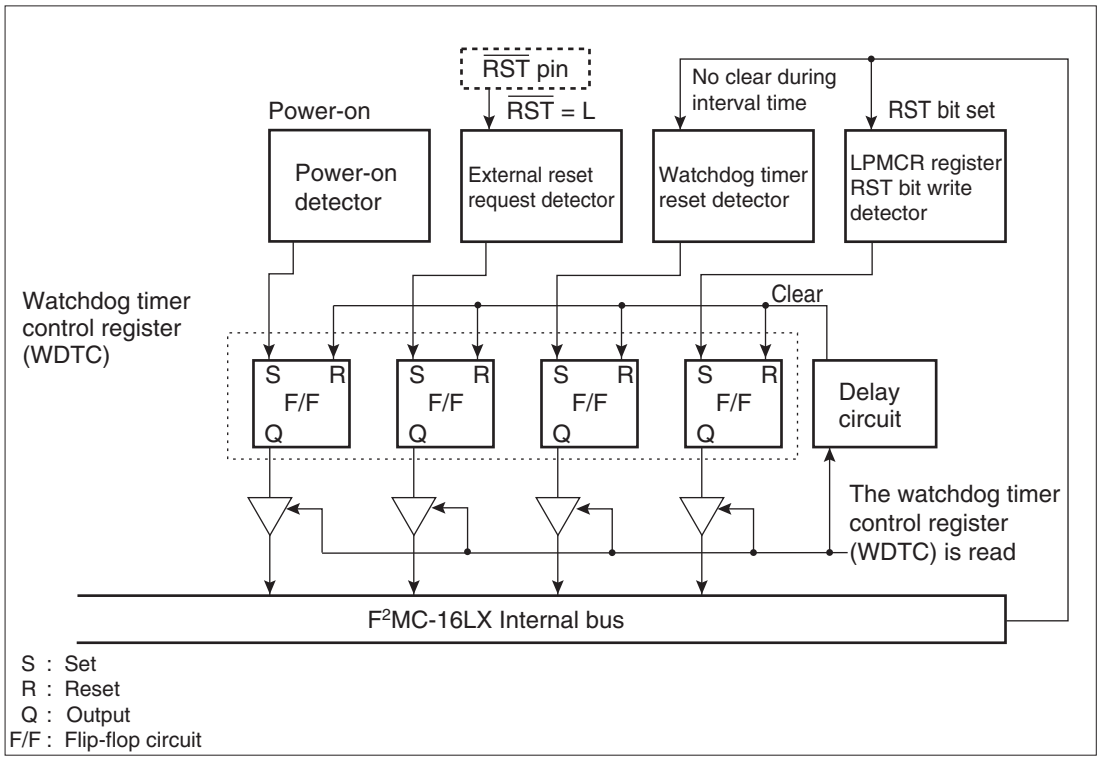
3.6.4 Reset Factor Bit

To check reset factors, read the value of the watchdog timer control register (WDTC).

■ Reset Factor Bit

Each reset factor provides a flip-flop circuit corresponding to each factor. The state of the flip-flop circuit can be checked by reading the value of the watchdog timer control register (WDTC). If it is necessary to identify reset factors after completion of the reset operation, read the value of the watchdog timer control register (WDTC) by software to branch the value to the appropriate program.

Figure 3.6-5 Block Diagram of Reset Factor Bits



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Correspondence of Reset Factor Bit and Reset Factor

Figure 3.6-6 shows the configuration of the reset factor bits in the watchdog timer control register (WDTC: PONR, WRST, ERST, SRST).

Figure 3.6-6 Configuration of Reset Factor Bit

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset value
Watchdog timer control register (WDTC)	PONR	-	WRST	ERST	SRST	WTE	WT1	WT0	$XXXXX111_B$
	R	-	R	R	R	W	W	W	

R: Read only
W: Write only
X: Undefined

Table 3.6-4 Correspondence of Reset Factor Bit Value and Reset Factor

Reset Factor	PONR	WRST	ERST	SRST
Power on reset	1	X	X	X
Watchdog timer reset	*	1	*	*
Input of external reset signal to $\overline{RST}$ pin	*	*	1	*
Software reset (RST bit)	*	*	*	1

*: The previous state is held
X: Undefined

■ Notes on Reset Factor Bit

● Power on reset

When a power on reset is executed, the PONR bit is set to 1 after completion of the reset operation. Any reset factor bit other than the PONR bit is undefined. When the PONR bit is 1 after completion of the reset operation, ignore the value of any bit other than the PONR bit.

● At two or more reset factors

The reset factor bit is set to 1 according to each reset factor even when two or more reset factors are generated. For example, if the watchdog timer overflows and an external reset request is generated from the $\overline{RST}$ pin at the same time, both WRST and ERST bits are set to 1 after completion of the reset operation.

● Clearing of reset factor bit

Once set, the reset factor bit is not cleared even if any reset factor other than the set factor is generated. The reset factor bit is cleared after the completion of reading the watchdog timer control register (WDTC).

Reference: For the watchdog timer, see "CHAPTER 6 WATCHDOG TIMER".

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.6.5 State of Each Pin at Reset

This section explains the state of each pin at reset.

■ State of Pins at Reset

The state of the pins during reset operation is determined by the settings of the mode pins (MD0 to MD2).

- When internal vector mode set:

  If the internal vector mode is set, all I/O pins enter the high-impedance state and mode data is read to internal ROM.

■ State of Pins after Mode Data Read

The I/O pins are all set to the high-impedance state, and the mode data read destination is the internal ROM.

Note: Don't let the device connected to pins that enter the high-impedance state malfunction when the reset factor is generated.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.7 Clocks

The clock generation section controls the internal clock that is an operating clock for the CPU or resources. The clock generated by the clock generation section is called a "machine clock" and one cycle of the machine clock is a machine cycle. The clock to be supplied from a high-speed oscillator is called an "oscillation clock" and the 2-frequency division of the oscillation clock is called a "main clock." The 4-frequency division of a clock to be supplied from a low-speed oscillator is called a "subclock" and the clock to be supplied from the PLL oscillation is called a "PLL clock."

■ Clock

The clock generation section has oscillators and generates an oscillation clock by connecting an oscillator to oscillation pins. External clocks that are input to the oscillation pins can be used as oscillation clocks. The PLL clock multiplying circuit can be used to generate four clocks for multiplying the oscillation clock. The clock generation section controls the oscillation stabilization wait time, PLL clock multiplying, and selects internal clock by the clock selector.

- Oscillation clock (HCLK)

  This clock is generated by connecting an oscillator or inputting an external clock to the high-speed oscillation pins (X0 and X1).

et4U.com DataShee DataSheet4U.com

- Main clock (MCLK)

  This clock is 2-frequency division of oscillation clock, and is an input clock to the timebase timer and clock selector.

- Subclock (SCLK)

  This clock is a clock with 4-frequency division of the clock generated by connecting an oscillator or inputting an external clock to the low-speed oscillation pins (X0A and X1A). It can also be used as an operating clock for the watch timer or as a low-speed machine clock.

- PLL clock (PCLK)

  This clock is multiplied by the PLL clock multiplying circuit (PLL oscillator). It can be selected from four types of clock according to the setting of the multiplication rate select bits (CKSCR: CS1, CS0).

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Machine clock

This clock is an operating clock for the CPU and the resources. One cycle of the machine clock is a machine cycle (1/ϕ). One clock can be selected from the main clock, subclock, and four types of PLL clock.

Notes:
- When the operating voltage is 5 V, the oscillation clock can oscillate at 3 MHz to 16 MHz. The maximum operating frequency of the CPU or resources is 16 MHz. If a multiplication rate that exceeds the maximum operating frequency is set, the device does not operate normally. If the oscillation clock is 16 MHz, the multiplication rate of PLL clock can only be set to x1. The PLL oscillator oscillates in the range of 3 MHz to 16 MHz, which varies depending on the operating voltage and multiplication rate.
- There is no sub-clock in MB90F387S and MB90387S.

■ Clock Supply Map

Machine clocks generated by the clock generation section are supplied as operating clocks of the CPU and resources. The operation of the CPU and resources is affected by switching among the main clock, subclock, and PLL clock (clock mode) or by switching the multiplication rate of PLL clock. The clock-divided output of the timebase timer is supplied to some resources, and the operating clock can be selected for each resource.

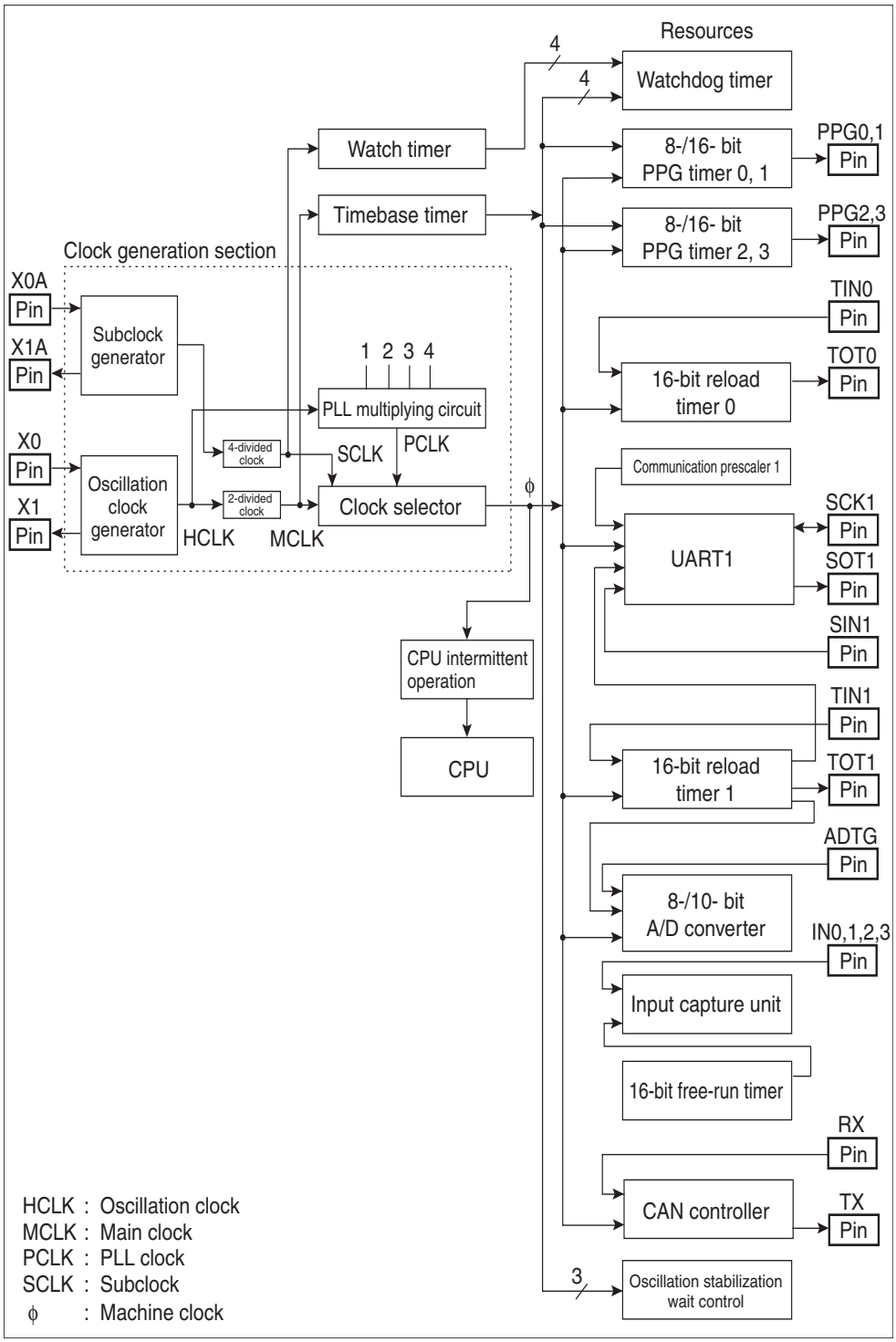
Figure 3.7-1 shows the clock supply map.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 3.7-1 Clock Supply Map



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.7.1 Block Diagram of Clock Generation Section

The clock generation section consists of the following five blocks:

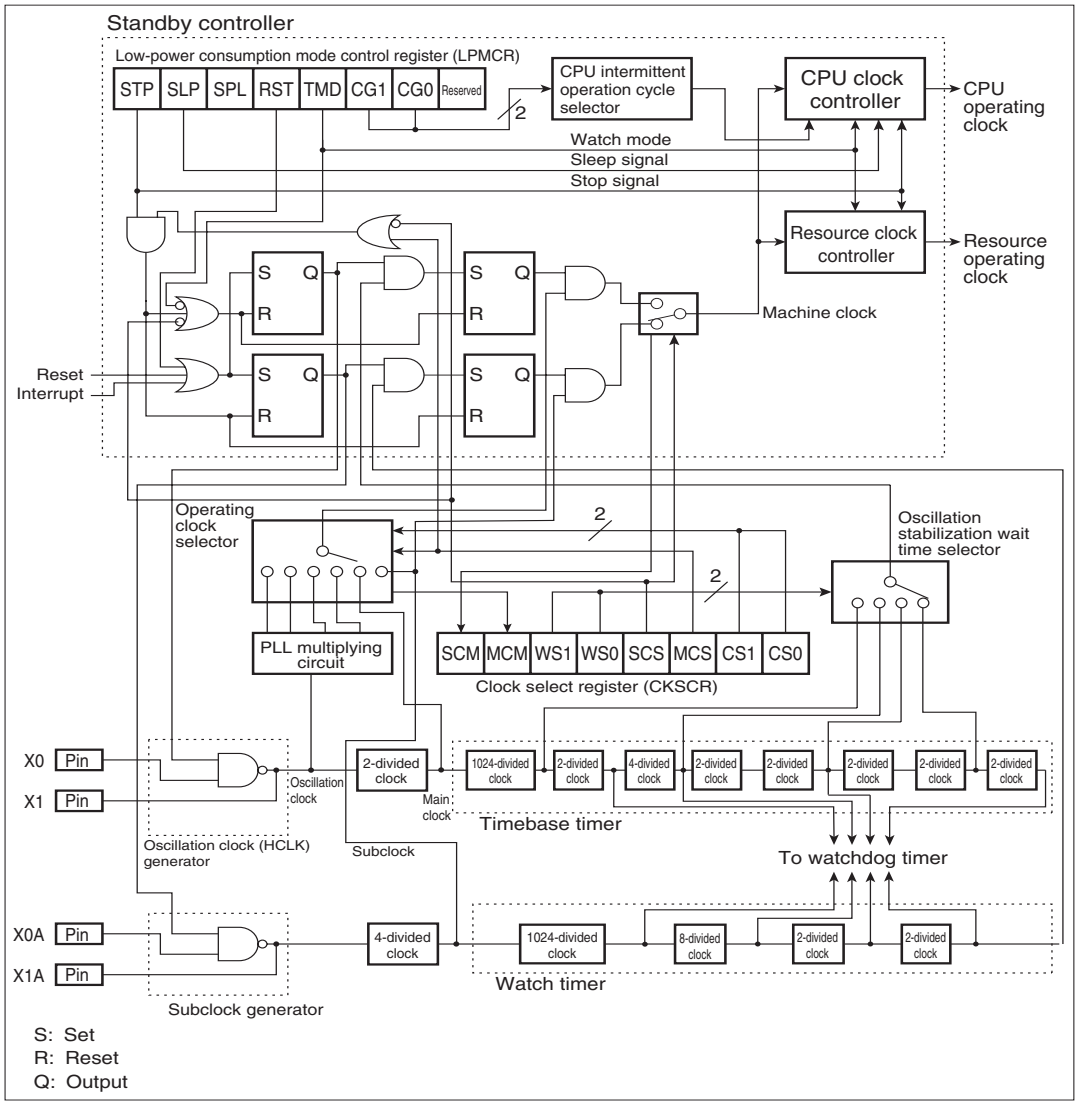
- **Oscillation clock generator/subclock generator**
- **PLL multiplying circuit**
- **Clock selector**
- **Clock select register (CKSCR)**
- **Oscillation stabilization wait time selector**

■ Block Diagram of Clock Generation Section

Figure 3.7-2 shows the block diagram of the clock generation section.

It also includes the standby controller and timebase timer circuit.

Figure 3.7-2 Block Diagram of Clock Generation Section



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Oscillation clock generator

  This generator generates an oscillation clock (HCLK) by connecting an oscillator or inputting an external clock to the high-speed oscillation pins.

- Subclock generator

  This generator generates a subclock (SCLK) by connecting an oscillator or inputting an external clock to the low-speed oscillation pins (X0A, X1A).

- PLL multiplying circuit

  This circuit multiplies the oscillation clock and supplies it as a PLL clock (PCLK) to the clock selector.

- Clock selector

  This selector selects the clock that is supplied to the CPU or resources from the main clock, subclock, and four types of PLL clock.

- Clock select register (CKSCR)

  This register switches between the oscillation clock and the PLL clock, and between the main clock and the subclock, selects the oscillation stabilization wait time, and the multiplication rate of the PLL clock.

- Oscillation stabilization wait time selector

  This selector selects the oscillation stabilization wait time of the oscillation clock. There are four timebase timer outputs to select.

Note:	There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.7.2 Register in Clock Generation Section

This section explains the register in the clock generation section.

■ Register in Clock Generation Section and List of Reset Values

Figure 3.7-3 Clock Select Register and List of Reset Values

bit	15	14	13	12	11	10	9	8
Clock select register (CKSCR)	1	1	1	1	1	1	0	0

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSh4U.com

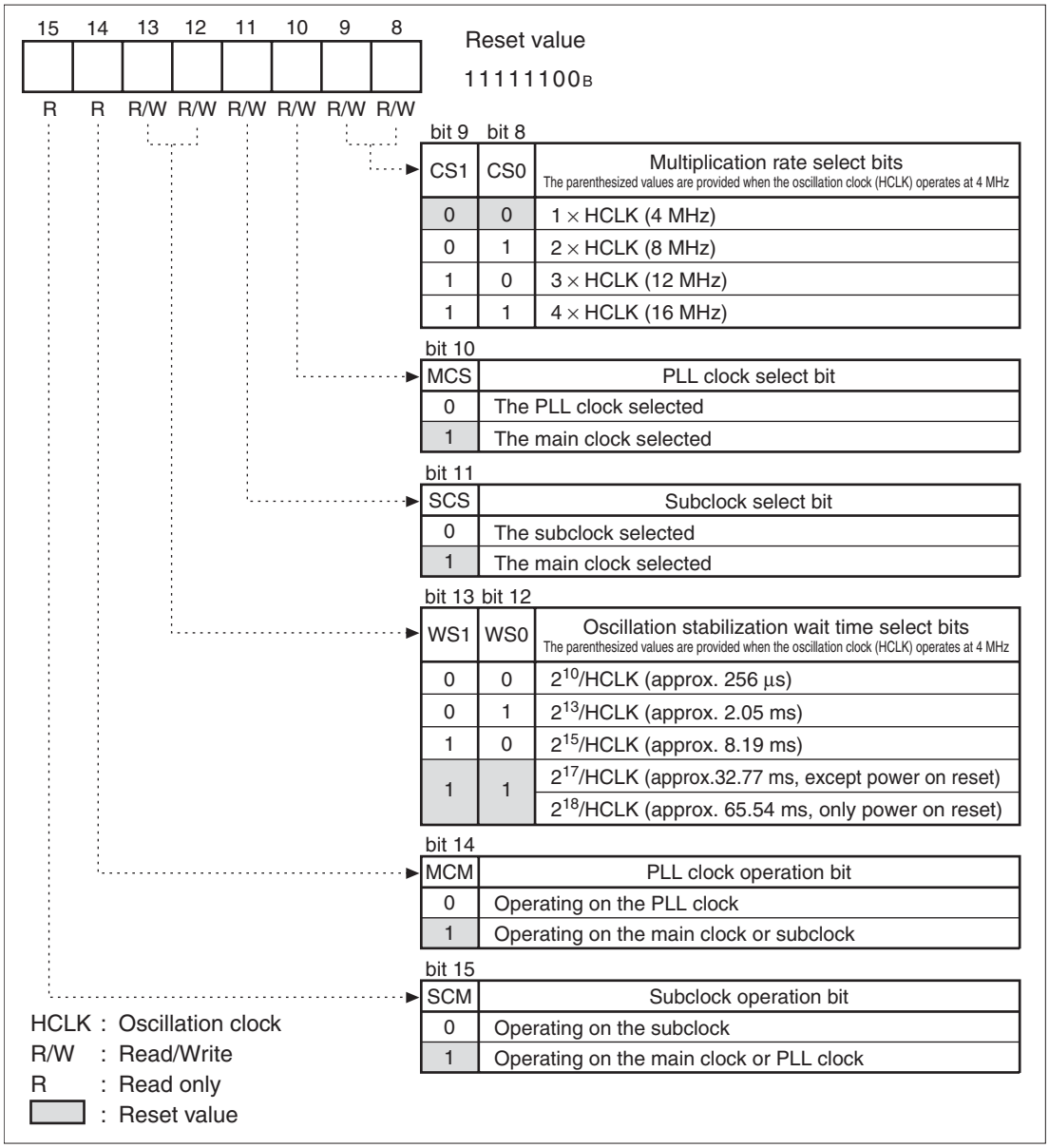
www.DataSheet4U.com

3.7.3 Clock Select Register (CKSCR)

The clock select register (CKSCR) switches between the main clock, subclock, and PLL clock, selects the oscillation stabilization wait time and the multiplication rate of PLL clock.

■ Configuration of Clock Select Register (CKSCR)

Figure 3.7-4 Clock Select Register (CKSCR)



bit 9 bit 8

CS1	CS0	Multiplication rate select bits The parenthesized values are provided when the oscillation clock (HCLK) operates at 4 MHz
0	0	1 × HCLK (4 MHz)
0	1	2 × HCLK (8 MHz)
1	0	3 × HCLK (12 MHz)
1	1	4 × HCLK (16 MHz)

bit 10

MCS	PLL clock select bit
0	The PLL clock selected
1	The main clock selected

bit 11

SCS	Subclock select bit
0	The subclock selected
1	The main clock selected

bit 13 bit 12

WS1	WS0	Oscillation stabilization wait time select bits The parenthesized values are provided when the oscillation clock (HCLK) operates at 4 MHz
0	0	2^{10}/HCLK (approx. 256 μs)
0	1	2^{13}/HCLK (approx. 2.05 ms)
1	0	2^{15}/HCLK (approx. 8.19 ms)
1	1	2^{17}/HCLK (approx.32.77 ms, except power on reset)
		2^{18}/HCLK (approx. 65.54 ms, only power on reset)

bit 14

MCM	PLL clock operation bit
0	Operating on the PLL clock
1	Operating on the main clock or subclock

bit 15

SCM	Subclock operation bit
0	Operating on the subclock
1	Operating on the main clock or PLL clock

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 3.7-1 Function of Each Bit of Clock Select Register (CKSCR) (1/2)

Bit Name		Function
bit 9 bit 8	CS1, CS0: Multiplication rate select bits	These bits are used to select the multiplication rate of the PLL clock from four types. When reset, they all return to their reset value. **Note:** When the PLL clock is selected (CKSCR: MCS = 0), writing is inhibited. When changing the multiplication rate, write 1 to the PLL clock select bit (CKSCR: MCS), rewrite the multiplication rate select bits (CKSCR: CS1, CS0), and then return the PLL clock select bit (CKSCR: MCS) to "0".
bit 10	MCS: PLL clock select bit	This bit sets where to select the main clock or PLL clock as a machine clock. If the machine clock is switched from the main clock to the PLL clock (CKSCR: MCS = 1 --> 0), the oscillation stabilization wait time of the PLL clock is generated and then the mode transits to the PLL clock mode. The timebase timer is automatically cleared. When the main clock mode is switched to PLL clock, the oscillation stabilization wait time is fixed to 2^{14}/HCLK (approximately 4.1 ms when the oscillation clock operates at 4 MHz). When subclock mode is switched to PLL clock, the oscillation stabilization wait time uses the specified values in the oscillation stabilization wait time selection bits (CKSCR: WS1, WS0). When reset, this bit returns to its reset value. **Notes:** 1. If both the MCS and SCS bits are 0, the SCS bit is preferred and the subclock mode is set. 2. When switching the machine clock from the main clock to the PLL clock (CKSCR: MCS = 1 --> 0), use the interrupt enable bit of the timebase timer (TBTC: TBIE) or the interrupt level mask register (ILM: ILM2 to ILM0) to disable the timebase timer interrupts.
bit 11	SCS: Subclock select bit	This bit sets whether to select main clock or subclock as machine clock. • When the machine clock is switched from the main clock to the subclock (CKSCR: SCS = 1 --> 0), the main clock mode transits to the subclock mode in synchrony with the subclock (approximately 130 µs). • When the machine clock is switched from the subclock to the main clock (CKSCR: SCS = 0 --> 1), the subclock mode transits to the main clock mode after the main clock oscillation stabilization wait time is generated. The timebase timer is automatically cleared. When reset, this bit returns to its reset value. **Notes:** 1. If both the MCS and SCS bits are 0, the SCS bit is preferred and the subclock mode is set. 2. If both the subclock select bit (CKSCR: MCS) and PLL clock select bit (CKSCR: SCS) are 0, the subclock is preferred. 3. When switching the machine clock from the main clock to the subclock (CKSCR: SCS = 1 --> 0), use the interrupt enable bit of the timebase timer (TBTC: TBIE) or the interrupt level mask register (ILM: ILM2 to ILM0) to disable the timebase timer. 4. At power on or when the stop mode is cancelled, the subclock oscillation stabilization wait time (approximately 2 s) is generated. Therefore, if the mode is switched from the main clock mode to the subclock mode, the oscillation stabilization wait time is generated. 5. There is no sub-clock in MB90F387S and MB90387S. This bit should be set to initial values.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 3.7-1 Function of Each Bit of Clock Select Register (CKSCR) (2/2)

Bit Name		Function
bit 13 bit 12	WS1, WS0: Oscillation stabilization wait time select bits	These bits select an oscillation stabilization wait time of the oscillation clock when stop mode was released, when transition occurred from subclock mode to main clock mode, or when transition occurred from subclock mode to PLL clock mode. • These bits are used to select from four timebase timer outputs. When reset, they all return to their reset value. **Note:** Set an oscillation stabilization wait time appropriate for an oscillator. For details, see Section "3.6.1 Reset Factors and Oscillation Stabilization Wait Time". When the main clock mode is switched to PLL clock mode, the oscillation stabilization wait time is fixed to 2^{14}/HCLK (approximately 4.1 ms when the oscillation clock operates at 4 MHz). When subclock mode is switched to PLL clock mode or when PLL stop mode is returned to PLL clock mode, the oscillation stabilization wait time uses the specified values in the WS1 and WS0 bits. For PLL clock oscillation stabilization wait time, at least 2^{14} /HCLK is required. Accordingly, when subclock mode is switched to PLL clock mode, or when PLL clock mode is switched to PLL stop mode, set WS1 and WS0 bits to "10_B" or "11_B".
bit14	MCM: PLL clock operation bit	This bit indicates whether to select main clock or PLL clock as machine clock. • If the PLL clock operation flag bit (CKSCR: MCM) is 1 and the PLL clock select bit (CKSCR: MCS) is 0, it indicates that the oscillation stabilization wait time of the PLL clock is taken.
bit 15	SCM: Subclock operation bit	This bit indicates whether to select main clock or the subclock as the machine clock. • If the subclock operation flag bit (CKSCR: SCM) is 0 and the subclock select bit (CKSCR: SCS) is 1, it indicates that the subclock switches to the main clock. If the subclock operation flag bit (CKSCR: SCM) is 1 and the subclock select bit (CKSCR: SCS) is 0, it indicates that the main clock switches to the subclock.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

www.DataSheet4U.com

3.7.4 Clock Mode

Clock modes have a main clock mode, subclock mode, and PLL clock mode.

■ Clock Mode

● Main clock mode

In the main clock mode, a clock with 2-frequency division of the clock generated by connecting an oscillator or inputting an external clock to the high-speed oscillation pins (X0, X1) is used as the operating clock for the CPU or resources.

● Subclock mode

In the subclock mode, a clock with 4-frequency division of the clock generated by connecting an oscillator or inputting an external clock to the low-speed oscillation pins (X0A, X1A) is used as the operating clock for the CPU or resources.

● PLL clock mode

In the PLL clock mode, the oscillation clock multiplied by the PLL clock multiplying circuit (PLL oscillator circuit) is used as the operating clock for the CPU or resources. The PLL clock multiplication rate can be set using the clock select register (CKSCR: CS1, CS0).

et4U.com DataShee DataSheet4U.com

Note: There is no sub-clock in MB90F387S and MB90387S.

■ Transition of Clock Mode

In clock modes, the setting of the PLL clock select bit (CKSCR: MCS) and subclock select bit (CKSCR: SCS) transits to the main clock mode, subclock mode or PLL clock mode.

● Transition from main clock mode to PLL clock mode

If the PLL clock select bit (CKSCR: MCS) is rewritten from 1 to 0, the main clock switches to the PLL clock after the PLL oscillation stabilization wait time (2^{14}/HCLK) has elapsed.

● Transition from PLL clock mode to main clock mode

If the PLL clock select bit (CKSCR: MCS) is rewritten from 0 to 1, the PLL clock switches to the main clock when the edge of the PLL clock matches the edge of the main clock (after 1 to 8 PLL clocks).

● Transition from main clock mode to subclock mode

If the subclock select bit (CKSCR: SCS) is rewritten from 1 to 0, the main clock switches to the subclock synchronizing the subclock (approx.130 µs).

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Transition from subclock mode to main clock mode

When the subclock select bit (CKSCR: SCS) is rewritten from 0 to 1, the subclock switches to the main clock after the main clock oscillation stabilization wait time has elapsed.

Notes:
- When sub-clock mode are returned to main clock mode using an external reset pin ($\overline{RST}$ pin), input level "L" for at least "the oscillation time of the oscillator* + 100μs + 16 machine cycles (main clock)".
  *: The oscillation time of the oscillator is the time required to reach 90% of amplitude. It takes several to dozens of ms for crystal oscillators, hundreds of μs to several ms for FAR/ceramic oscillators, and 0 ms for external clocks.
- There is no sub-clock in MB90F387S and MB90387S.

● Transition from PLL clock mode to subclock mode

When the subclock select bit (CKSCR: SCS) is rewritten from 1 to 0, the PLL clock switches to the subclock.

● Transition from subclock mode to PLL clock mode

When the subclock select bit (CKSCR: SCS) is rewritten from 0 to 1, the subclock switches to the PLL clock after the main clock oscillation stabilization wait time has elapsed.

■ Selection of PLL Clock Multiplication Rate

The PLL clock multiplication rate can be set from x1 to x4 by writing values of "00_B" to "11_B" to the multiplication rate select bits (CKSCR: CS1, CS0).

et4U.com

DataShee

■ Machine Clock

DataSheet4U.com

The PLL clock, main clock, and subclock output from the PLL multiplying circuit are used as machine clocks supplied to the CPU or resources.

Any of the main clock, PLL clock, and subclock can be selected by writing to the subclock select bit (CKSCR: SCS) and the PLL clock select bit (CKSCR: MCS).

Notes:
- The machine clock is not switched immediately even when the PLL clock select bit (CKSCR: MCS) and the subclock select bit (CKSCR: SCS) are rewritten. When running resources that depend on the machine clock, after switching the machine clock, refer the value of the PLL clock operation flag bit (CKSCR: MCM) or the subclock operation flag bit (CKSCR:SCM) to check that the machine clock has been switched.
- When the PLL clock select bit (CKSCR: MCS) is 0 (PLL clock mode) and the subclock select bit (CKSCR: SCS) is 0 (subclock mode), the SCS bit is preferred, transiting to the subclock mode.
- When transiting a clock mode, do not transit a clock mode to any other clock mode or a low-power consumption mode until the completion of transition. Refer the MCM and SCM bits in the clock select register (CKSCR) to check that the transition of a clock mode is completed. If the mode is switched to another clock mode or low-power-consumption mode before completion of switching, the mode may not be switched.
- There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com

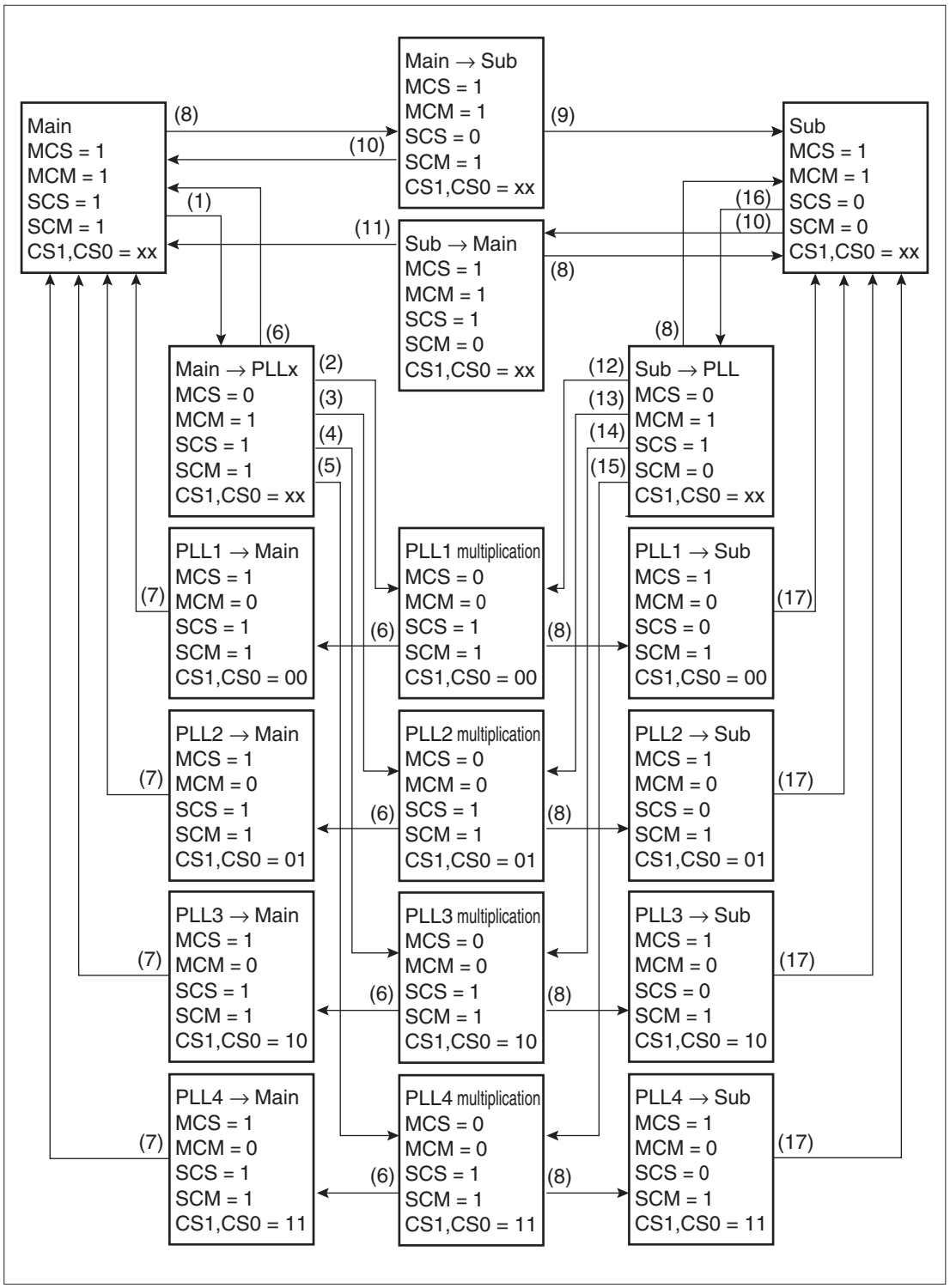
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 3.7-5 shows the transition of a clock mode.

Figure 3.7-5 Clock Mode Transition



DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

(1) 0 write to MCS bit
(2) PLL clock oscillation stabilization waiting termination & CS1, CS0 = 00
(3) PLL clock oscillation stabilization waiting termination & CS1, CS0 = 01
(4) PLL clock oscillation stabilization waiting termination & CS1, CS0 = 10
(5) PLL clock oscillation stabilization waiting termination & CS1, CS0 = 11
(6) 1 write to MCS bit (hardware standby and the watchdog reset included)
(7) Synchronous timing of PLL clock and main clock
(8) 0 write to SCS bit
(9) Subclock oscillation stabilization wait time termination (maximum 2^{14}/SCLK)
(10) 1 write to SCS bit
(11) Main clock oscillation stabilization waiting termination
(12) Main clock oscillation stabilization waiting termination & CS1, CS0 = 00
(13) Main clock oscillation stabilization waiting termination & CS1, CS0 = 01
(14) Main clock oscillation stabilization waiting termination & CS1, CS0 = 10
(15) Main clock oscillation stabilization waiting termination & CS1, CS0 = 11
(16) 1 write to SCS bit and 0 to MCS bit
(17) Synchronous timing of PLL clock and subclock

MCS : PLL clock select bit of clock select register (CKSCR)
MCM : PLL clock display bit of clock select register (CKSCR)
SCS : Subclock select bit of clock select register (CKSCR)
SCM : Subclock display bit of clock select register (CKSCR)
CS1,CS0 : Multiplication rate select bit of clock select register (CKSCR)

et4U.com DataShee
DataSheet4U.com

Notes:
- The reset value of the machine clock is in the main clock mode (MCS = 1, SCS = 1).
- When SCS and MCS are both 0, SCS is preferred, and the subclock is selected.
- When transiting from the subclock mode to the PLL clock mode, set the oscillation stabilization wait time select bit of the CKSCR register (WS1, WS0) to 10_B or 11_B.
- There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.7.5 Oscillation Stabilization Wait Time

At power on or return from the stop mode, when the oscillation clock is stopped, a time taken until the oscillation clock stabilizes (oscillation stabilization wait time) is required after starting an oscillation. The oscillation stabilization wait time is also required for switching the clock mode from main clock mode to PLL clock mode, from main clock mode to subclock mode, from subclock mode to main clock mode, and from subclock mode to PLL clock mode.

■ Operation During Oscillation Stabilization Wait Time

Ceramic and crystal oscillators require several to some tens of milliseconds to reach a stable oscillation frequency after starting oscillation. Therefore when, immediately after an oscillation starts, once the CPU operation is disabled and then an oscillation stabilizes after the elapse of oscillation stabilization wait time, the machine clock is supplied to the CPU.

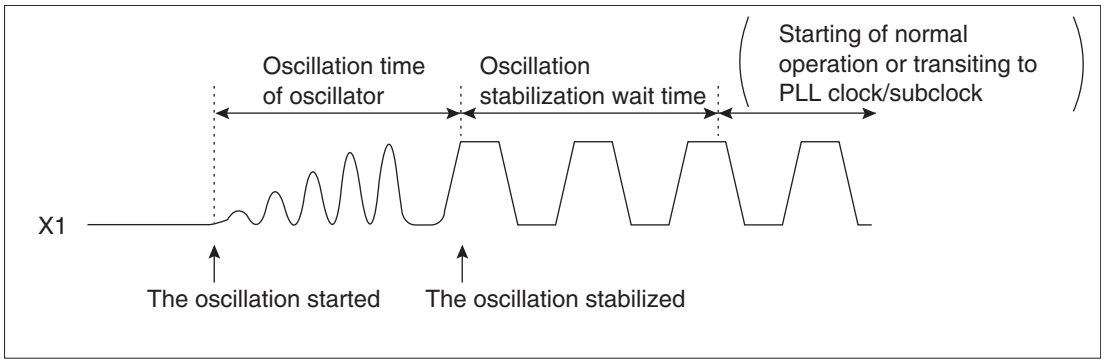
The oscillation stabilization wait time varies with the type of oscillator (ceramic, crystal, etc.).

It is necessary to select a oscillation stabilization wait time appropriate to an oscillator to be used.

The oscillation stabilization wait time can be selected using the clock select register (CKSCR).

When clock mode is switched from main clock to PLL clock, main clock to subclock, subclock to main clock, or subclock to PLL clock, the CPU runs in the clock mode set before switching for the oscillation stabilization wait time. After the oscillation stabilization wait time has elapsed, the CPU changes to the specified clock mode. Figure 3.7-6 shows the oscillating operation immediately after it starts.

Figure 3.7-6 Operation after Oscillation Stabilization Wait Time



Note: There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

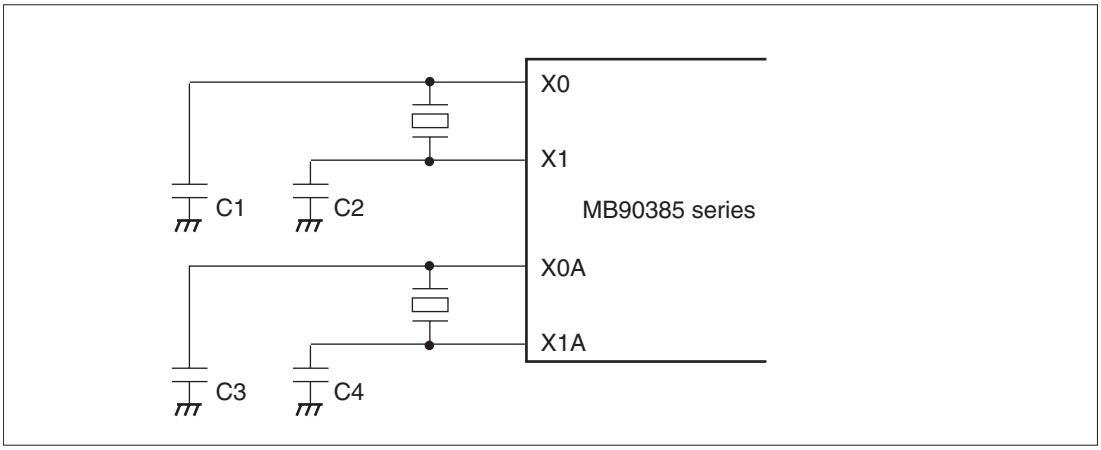
3.7.6 Connection of Oscillator and External Clock

The MB90385 series has a system clock generator and generates an internal clock by connecting an oscillator to the oscillation pins. External clocks input to the oscillation pins can be used as oscillation clocks.

■ Connection of Oscillator and External Clock

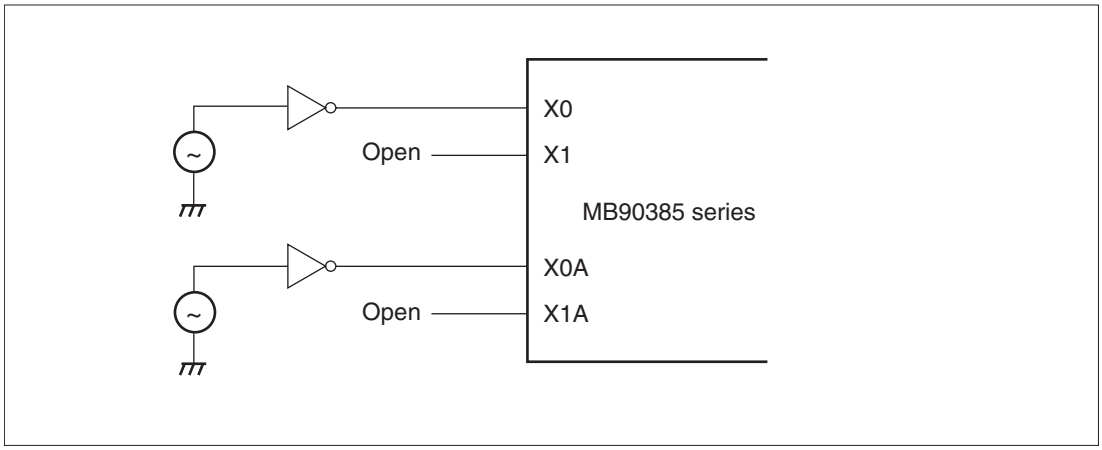
- Example of connection of crystal oscillator or ceramic oscillator

Figure 3.7-7 Example of Connection of Crystal Oscillator or Ceramic Oscillator



- Example of connection of external clock

Figure 3.7-8 Example of Connection of External Clock



Note: There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.8 Low-power Consumption Mode

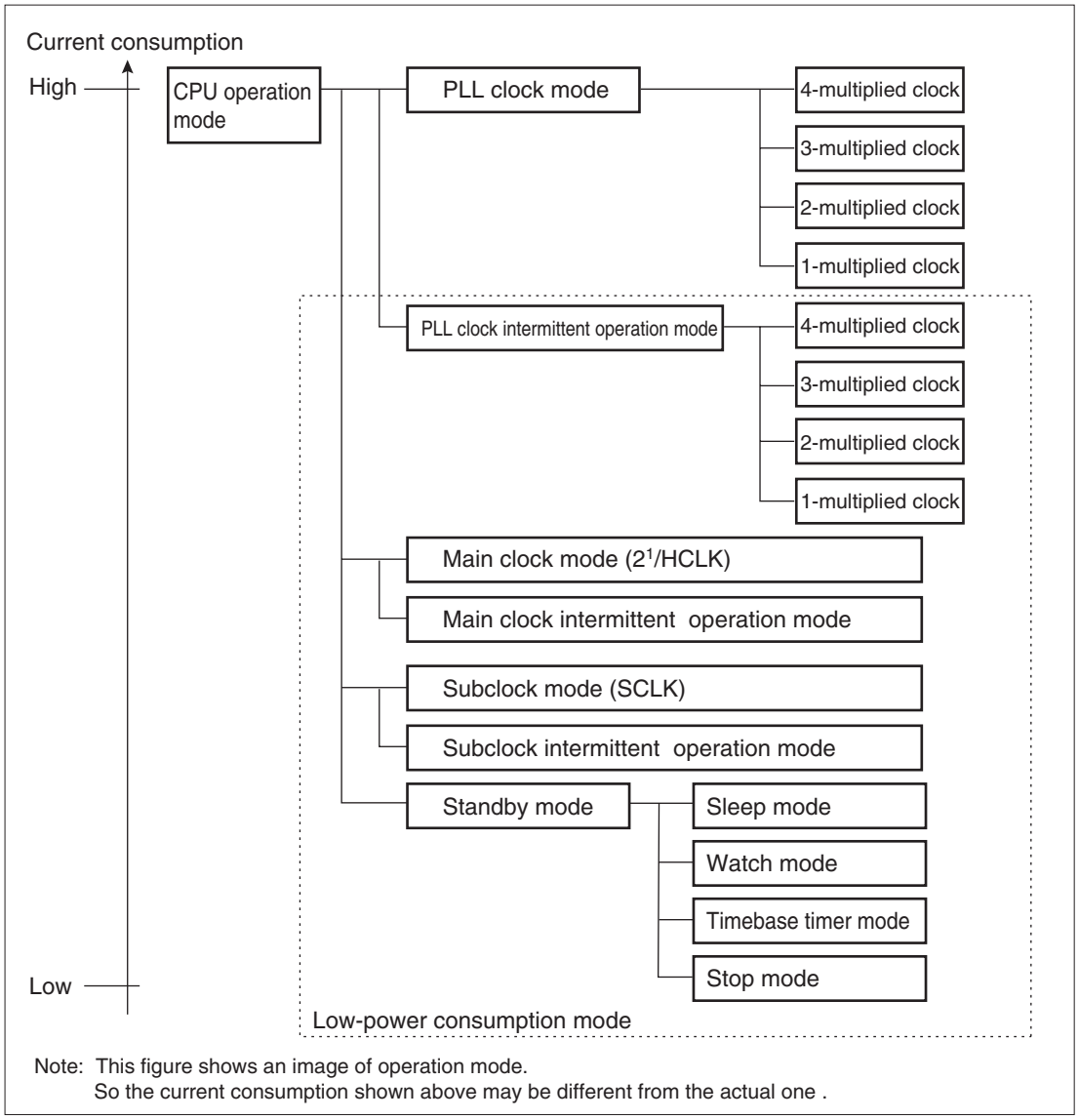
The CPU operation modes are classified as follows according to the selection of the operation clock and the oscillation control of a clock. All the operation modes except the PLL clock mode are low-power consumption modes.

- **Clock modes (main clock, PLL clock and subclock modes)**
- **CPU intermittent operation modes (main clock, PLL clock, and subclock modes)**
- **Standby modes (sleep, stop, watch, and timebase timer modes)**

■ CPU Operation Modes and Current Consumption

Figure 3.8-1 shows the relationships between the CPU operation mode and current consumption.

Figure 3.8-1 CPU Operation Mode and Current Consumption



Note: This figure shows an image of operation mode.
So the current consumption shown above may be different from the actual one .

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Clock Mode

● PLL clock mode

In PLL clock mode, the CPU and resources operate on a PLL multiplying clock of oscillation clock (HCLK).

● Main clock mode

In main clock mode, the CPU and resources operate on a clock with 2-frequency division of oscillation clock (HCLK). In this mode, the PLL multiplying circuit stops.

● Subclock mode

In subclock mode, the CPU and resources operate on a subclock (SCLK). In this mode, the main clock and PLL multiplying circuit stop.

The subclock oscillation stabilization wait time (approximately 2 s) is generated at power on or at cancellation of the stop mode. Therefore, if the clock mode transits from the main clock mode to the subclock mode during that period, the oscillation stabilization wait time is generated.

Note:	There is no sub-clock in MB90F387S and MB90387S.

et4U.com

Reference:	For the clock mode, see Section "3.7 Clocks".

DataShee

■ CPU Intermittent Operation Mode

DataSheet4U.com

In CPU intermittent operation mode, the CPU performs the intermittent operation with the high-speed clock supplied to the resource to reduce the power consumption. In this mode, the intermittent clock is input to only the CPU at accessing registers, internal memory, or resources.

■ Standby Mode

The standby mode causes the standby control circuit to stop the supply of an operation clock to the CPU or resources or to stop the oscillation clock (HCLK) in order to reduce power consumption.

● Sleep mode

The sleep mode stops supply of an operation clock to the CPU during operation in each clock mode. The CPU stops and the resources operate in the clock mode before the transition to the sleep mode. The sleep mode is divided into the main sleep mode, PLL sleep mode, and sub-sleep mode according to the clock mode before the transition to the sleep mode.

● Watch mode

The watch mode operates only the subclock (SCLK) and watch timer. The main clock and PLL clock stop. All resources except the watch timer stop.

● Timebase timer mode

The timebase timer mode operates only the oscillation clock (HCLK), subclock (SCLK), timebase timer, and watch timer. Resources other than the timebase timer and watch timer stop.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Stop mode

The stop mode stops the oscillation clock (HCLK) and subclock (SCLK) during operation in each clock mode. It enables data to be retained with the least power consumption.

Notes:
- When transiting a clock mode, do not transit a clock mode to any other clock mode or a low-power consumption mode until the completion of transition. Refer the MCM and SCM bits in the clock select register (CKSCR) to check that the transition of a clock mode is completed. If the mode is switched to another clock mode or low-power-consumption mode before completion of switching, the mode may not be switched.
- There is no sub-clock in MB90F387S and MB90387S.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

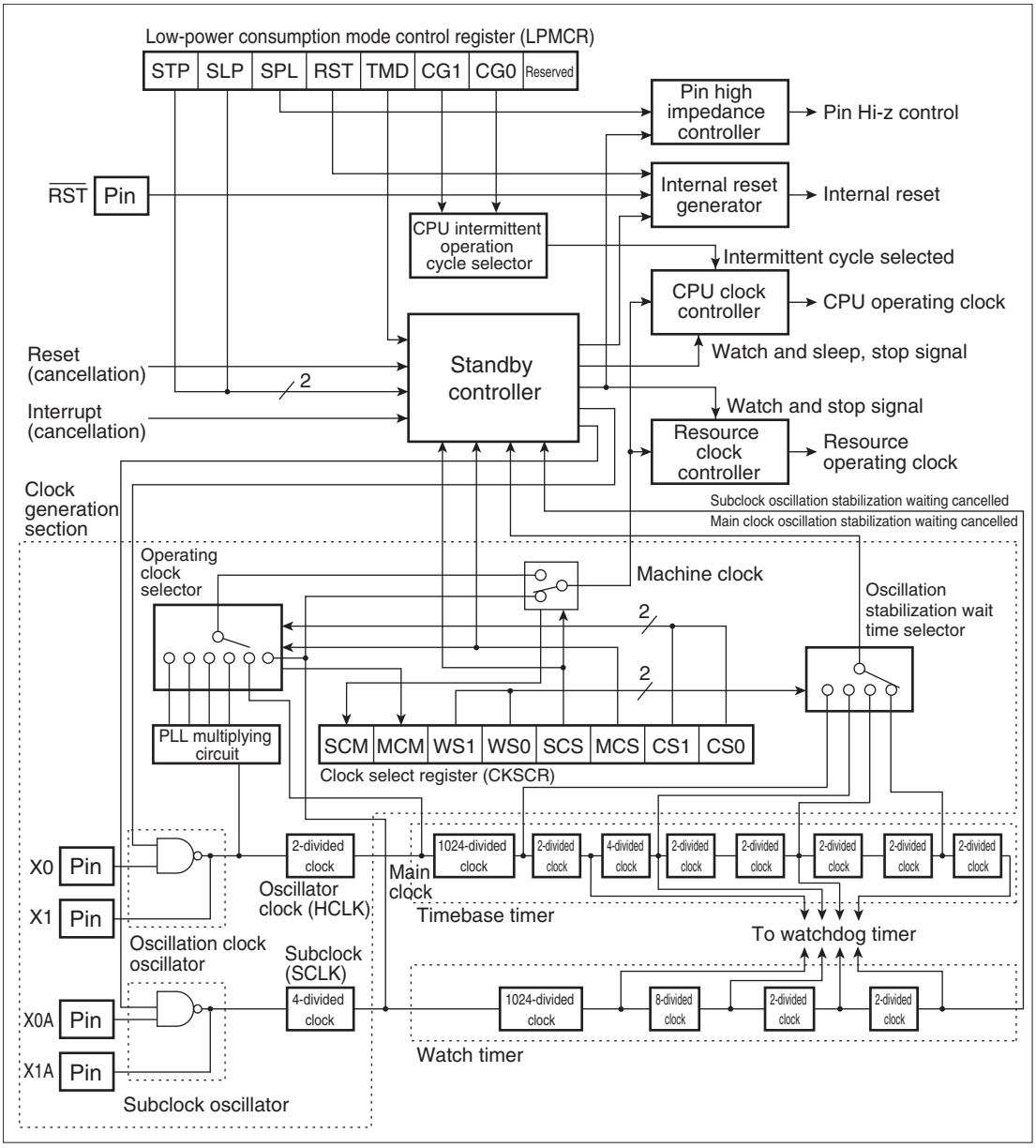
www.DataSheet4U.com

3.8.1 Block Diagram of Low-power Consumption Circuit

This section shows block diagram of low-power consumption circuit.

■ Block Diagram of Low-power Consumption Circuit

Figure 3.8-2 Block Diagram of Low-power Consumption Circuit



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- CPU intermittent operation selector

  This selector selects the halt cycle count of the CPU clock in the CPU intermittent operation mode.

- Standby controller

  This controller causes the CPU clock controller and resource clock controller to switch between the CPU operating clock and the resource operating clock, and to transits a clock mode to and cancel the standby mode.

- CPU clock controller

  This controller supplies an operating clock to the CPU.

- Pin high-impedance controller

  This controller causes the input/output pins to become high impedance in the watch mode, timebase timer mode, and stop mode.

- Internal reset generator

  This generator generates the internal reset signal.

- Low-power consumption mode control register (LPMCR)

  This register transits a clock mode to and cancels the standby mode, and sets the CPU intermittent operation mode.

et4U.com DataShee
DataSheet4U.com

Note:	There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.8.2 Registers for Setting Low-power Consumption Modes

This section explains the registers to be used to set lower-power consumption modes.

■ Low-power Consumption Mode Control Register and Reset Values

Figure 3.8-3 Low-power Consumption Mode Control Register and Reset Values

bit	7	6	5	4	3	2	1	0
Low-power consumption mode control register (LPMCR)	0	0	0	1	1	0	0	0

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSh4U.com

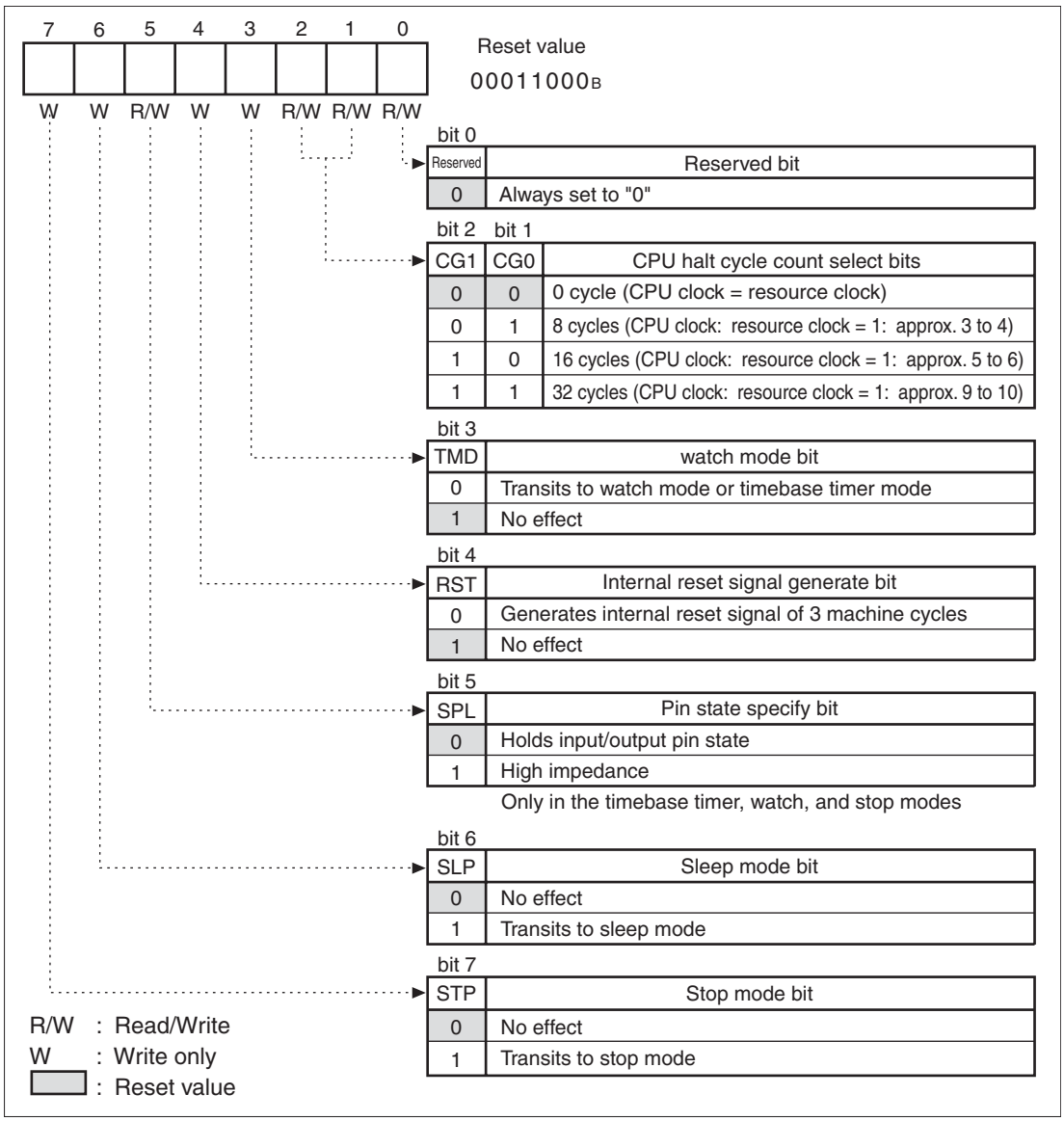
www.DataSheet4U.com

3.8.3 Low-power Consumption Mode Control Register (LPMCR)

The low-power consumption mode control register (LPMCR) transits an operation mode to and cancels the low-power consumption modes, generates an internal reset signal, and sets the halt cycle count in the CPU intermittent operation mode.

■ Low-power Consumption Mode Control Register (LPMCR)

Figure 3.8-4 Low-power Consumption Mode Control Register (LPMCR)



bit 0

Reserved	Reserved bit
0	Always set to "0"

bit 2 bit 1

CG1	CG0	CPU halt cycle count select bits
0	0	0 cycle (CPU clock = resource clock)
0	1	8 cycles (CPU clock: resource clock = 1: approx. 3 to 4)
1	0	16 cycles (CPU clock: resource clock = 1: approx. 5 to 6)
1	1	32 cycles (CPU clock: resource clock = 1: approx. 9 to 10)

bit 3

TMD	watch mode bit
0	Transits to watch mode or timebase timer mode
1	No effect

bit 4

RST	Internal reset signal generate bit
0	Generates internal reset signal of 3 machine cycles
1	No effect

bit 5

SPL	Pin state specify bit
0	Holds input/output pin state
1	High impedance

Only in the timebase timer, watch, and stop modes

bit 6

SLP	Sleep mode bit
0	No effect
1	Transits to sleep mode

bit 7

STP	Stop mode bit
0	No effect
1	Transits to stop mode

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 3.8-1 Function of Each Bit of Low-power Consumption Mode Control Register (LPMCR)

Bit Name		Function
bit 0	Reserved bit	Always set this bit to 0.
bit 1 bit 2	CG1, CG0: CPU halt cycle count select bit	These bits are used to set the halt cycle count of the CPU clock in the CPU intermittent operation mode. • Any reset causes the bit to return to the reset value.
bit 3	TMD: Watch mode bit	This bit is used to transit the operation mode to the watch mode or the timebase timer mode. **When set to 0:** The mode transits to the watch mode. **When set to 1:**Not effect • This bit is set to 1 by a reset or interrupt. **Read:** 1 is always read.
bit 4	RST: Internal reset signal generate bit	This bit is used to generate a software reset. **When set to 0:** Three machine cycles of internal reset signals are generated. **When set to 1:** No effect **Read:** 1 is always read.
bit 5	SPL: Pin state specify bit	This bit is used to set the state of input/output pins in transiting to the stop mode, watch mode or timebase timer mode. **When set to 0:** The current level of input/output pins is held. **When set to 1:** The input/output pins are set to high impedance. • This bit is initialized to 0 by a reset.
bit 6	SLP: Sleep mode bit	This bit is used to transit the mode to the sleep mode. **When set to 0:** No effect **When set to 1:** The mode transits to the sleep mode. • This bit is initialized to 0 by a reset or external interrupt. • When both the STP and SLP bits are set to 1 simultaneously, the STP bit is preferred and the mode transits to the stop mode.
bit 7	STP: Stop mode bit	This bit is used to transit the mode to the stop mode. **When set to 0:** No effect **When set to 1:** The mode is transits to the stop mode. **When read:** 1 is always read. • This bit is initialized to 0 by a reset or external interrupt.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Notes:

- When transiting to a low-power consumption mode using the low-power consumption mode control register (LPMCR), use the instructions listed in Table 3.8-2.
- The low-power consumption mode transition instruction in Table 3.8-2must always be followed by an array of instructions highlighted by a dotted line below.

```
MOV   LPMCR,#H'XX ; the low-power consumption mode transition instruction in Table 3.8-2
NOP
NOP
JMP   $+3         ; jump to next instruction
MOV   A,#H'10     ; any instruction
```

  The devices do not guarantee its operation after returning from the standby mode if you place an array of instructions other than the one enclosed in the dotted line.
- To access the low-power consumption mode control register (LPMCR) with C language, refer to "■ Notes on Accessing the Low-Power Consumption Mode Control Register (LPMCR) to Enter the Standby Mode" in the section "3.8.8 Precautions when Using Low-power Consumption Mode".
- When word-length is used for writing the low-power consumption mode control register, even addresses must be used. Using odd addresses to switch to a low-power consumption mode may result in a malfunction.
- To set a pin to high impedance when the pin is shared by a peripheral function and a port in stop mode, watch mode or timebase timer mode, disable the output of peripheral functions, and set the STP bit of the low-power consumption mode control register (LPMCR) to 1 or set the TMD bit to 0.
  This applies to the following pins:
  P14/PPG0, P15/PPG1, P16/PPG2, P17/PPG3, P21/TOT0, P23/TOT1
- There is no sub-clock in MB90F387S and MB90387S.

et4U.com DataShee

Table 3.8-2 Instructions at Transition to Low-power Consumption Mode

MOV io,#imm8	MOV dir,#imm8	MOV eam,#imm8	MOV eam,Ri
MOV io,A	MOV dir,A	MOV addr16,A	MOV eam,A
MOV @RLi+disp8,A			
MOVW io,#imm16	MOVW dir,#imm16	MOVW eam,#imm16	MOVW eam,RWi
MOVW io,A	MOVW dir,A	MOVW addr16,A	MOVW eam,A
MOVW @RLi+disp8,A			
SETB io:bp	SETB dir:bp	SETB addr16:bp	
CLRB io:bp	CLRB dir:bp	CLRB addr16:bp	

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.8.4 CPU Intermittent Operation Mode

The CPU intermittent operation mode causes the CPU to operate intermittently with an operating clock supplied to the CPU or resources to reduce power consumption.

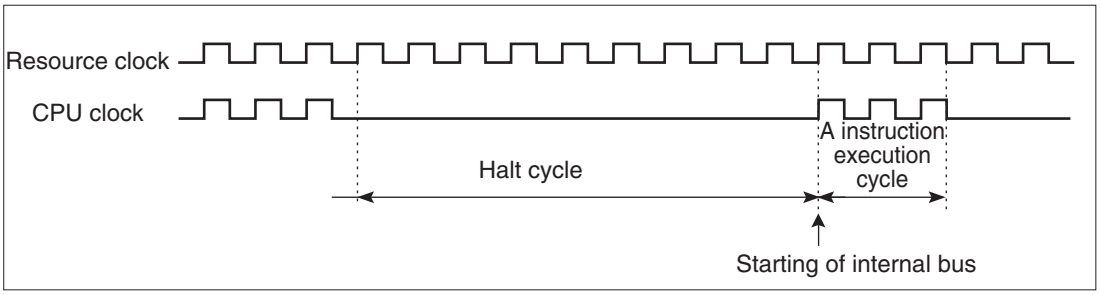
■ Operation in CPU Intermittent Operation Mode

The CPU intermittent operation mode halts the clock supplied to the CPU at every instruction execution when the CPU accesses registers, internal memory, I/O, or resources delaying to start the internal bus. Decreasing the CPU processing speed while supplying a high-speed clock to resources reduces the power consumption.

- The count of machine cycles in which clock supply to the CPU halts is set by the CG1 and CG0 bits in the low-power consumption mode control register (LPMCR).
- The instruction execution time in the CPU intermittent operation mode is determined by adding the "normal execution time" to the "compensation value" obtained by multiplying "count of accesses to registers, internal memory, and resources " by "halt cycle count."

Figure 3.8-5 shows the clock operation in the CPU intermittent operation mode.

Figure 3.8-5 Clock Operation in CPU Intermittent Operation Mode



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.8.5 Standby Mode

The standby mode causes the standby control circuit to either stop supplying an operation clock to the CPU and resources, or to stop the oscillation clock (HCLK) to reduce power consumption.

■ Operating State in Each Standby Mode

Table 3.8-3 shows the operating state in each standby mode.

Table 3.8-3 Operating State in Each Standby Mode

Mode Name		Transition Condition	Oscillation Clock (HCLK)	Subclock (SCLK)	Machine Clock	CPU	Resource	Pin	Cancellation
Sleep mode	Main sleep mode	MCS = 1 SCS = 1 SLP = 1	O	O	O	X	O	O	External reset or interrupt
	Sub-sleep mode	MCS = X SCS = 0 SLP = 1	X	O	O	X	O	O	External reset or interrupt
	PLL sleep mode	MCS = 0 SCS = 1 SLP = 1	O	O	O	X	O	O	External reset or interrupt
Timebase timer mode	SPL = 0	MCS = X SCS = 1 TMD = 0	O	O	X	X	X ∗1	◇	External reset or interrupt ∗4
	SPL = 1	MCS = X SCS = 1 TMD = 0	O	O	X	X	X ∗1	Hi-z ∗3	External reset or interrupt∗4
Watch mode	SPL = 0	MCS =X SCS = 0 TMD = 0	X	O	X	X	X ∗2	◇	External reset or interrupt ∗5
	SPL = 1	MCS = X SCS = 0 TMD = 0	X	O	X	X	X ∗2	Hi-z ∗3	External reset or interrupt ∗5
Stop mode	SPL = 0	STP = 1	X	X	X	X	X	◇	External reset or interrupt ∗6
	SPL = 1	STP = 1	X	X	X	X	X	Hi-z ∗3	External reset or interrupt ∗6

O: Operate X: Stop ◇ : Pre-transition state held Hi-z: High-impedance
∗1: The timebase timer and the watch timer operate.
*2: The watch timer operates.
*3: DTP/external interrupt input pins operates
*4: Watch timer, timebase timer, and external interrupt
*5: Watch timer and external interrupt.
*6: External interrupt
MCS: PLL clock select bit of clock select register (CKSCR)
SCS: Subclock select bit of CKSCR
SPL: Pin state specify bit of low-power consumption mode control register (LPMCR)
SLP: Sleep mode bit of LPMCR
STP: Stop mode bit of LPMCR
TMD: Watch mode bit of LPMCR

Note: There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.8.5.1 Sleep Mode

The sleep mode stops the operating clock to the CPU during an operation in each clock mode. The CPU stops and the resources continue to operate.

■ Transition to Sleep Mode

When the mode transits to the sleep mode by setting the low-power consumption mode control register (LPMCR: SLP = 1, STP = 0), the mode transits to the sleep mode according to the settings of the MCS and SCS bits in the clock select register (CKSCR).

Table 3.8-4 shows the settings of the MCS and SCS bits in the clock select register (CKSCR) and the sleep modes.

Table 3.8-4 Settings of MCS and SCS Bits in Clock Select Register (CKSCR) and Sleep Modes

Clock Select Register (CKSCR)		Sleep Mode to be transited
MCS	SCS	
1	1	Main sleep mode
0	1	PLL sleep mode
1	0	Sub-sleep mode
0	0	

et4U.com DataShee DataSheet4U.com

Notes:
- If both the STP and SLP bits in the low-power consumption mode control register (LPMCR) are set to 1 simultaneously, the STP bit is preferred and the mode transits to the stop mode.
  If the SLP bit is set to 1 and the TMD bit is set to 0 at the same time, the TMD bit is preferred and the mode transits to the timebase timer mode or the watch mode.
- There is no sub-clock in MB90F387S and MB90387S.

● Data hold function

In the sleep mode, data in the dedicated registers such as accumulators and internal RAM are held.

● Operation when interrupt request generated

If an interrupt request is generated when the SLP bit in the low-power consumption mode control register (LPMCR) is set to 1, the mode does not transit to the sleep mode. If the CPU is not ready to accept any interrupt request, the instruction next to the currently executing instruction is executed. If the CPU is ready to accept any interrupt request, an interrupt operation immediately branches to the interrupt processing routine.

● Pin state

In the sleep mode, pins other than those used for bus input/output or bus control are held in the state before transiting to the sleep mode.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Return from Sleep Mode

The sleep mode is cancelled by a reset factor or when an interrupt is generated.

● Return by reset factor

When the sleep mode is cancelled by a reset factor, the mode transits to the main clock mode after the sleep mode is cancelled, transiting to the reset sequence.

Notes:
- When sub-sleep mode are returned to main clock mode using an external reset pin ($\overline{RST}$ pin), input level "L" for at least "the oscillation time of the oscillator(*) + 100µs + 16 machine cycles".
  *: The oscillation time of the oscillator is the time required to reach 90% of amplitude. It takes several to dozens of ms for crystal oscillators, hundreds of µs to several ms for FAR/ceramic oscillators, and 0 ms for external clocks.
- There is no sub-clock in MB90F387S and MB90387S.

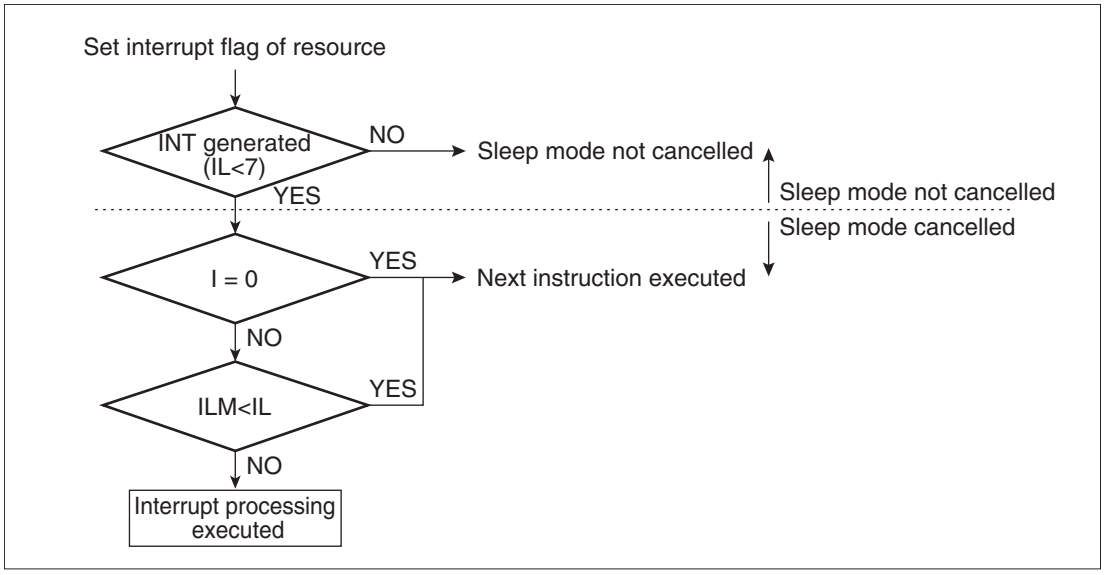
● Return by interrupt

When a higher interrupt request than the interrupt level (IL) of 7 is generated from the resources in the sleep mode, the sleep mode is cancelled. After the sleep mode is cancelled, as with normal interrupt processing, the generated interrupt request is identified according to the settings of the I flag in the condition code register (CCR), the interrupt level mask register (ILM), and the interrupt control register (ICR).

- When the CPU is not ready to accept any interrupt request, the instruction next to the currently executing instruction is executed.
- When the CPU is ready to accept any interrupt request, it branches immediately to the interrupt processing routine.

Figure 3.8-6 shows the cancellation of sleep mode by an interrupt.

Figure 3.8-6 Cancellation of Sleep Mode by Interrupt



Note: When an interrupt processing is executed, the CPU usually proceeds to the interrupt processing after executing the instruction next to the one specifying the sleep mode.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.8.5.2 Watch mode

The watch mode operates only the subclock (SCLK) and the watch timer. The main clock and PLL clock stop.

■ Transition to Watch mode

In the subclock mode, when 0 is written to the TMD bit in the LPMCR register according to the settings of the low-power consumption mode control register (LPMCR), the mode transits to the watch mode.

● Data hold function

In the watch mode, data in the dedicated registers such as an accumulator and internal RAM are held.

● Operation when interrupt request generated

When interrupt request is generated with the TMD bit of the low-power consumption mode control register (LPMCR) set to 0, the mode does not transit to the watch mode. If the CPU is not ready to accept any interrupt request, the instruction next to the currently executing instruction is executed. If the CPU is ready to accept any interrupt request, it immediately branches to the interrupt processing routine.

et4U.com

● Pin state

DataShee

In the watch mode, the input/output pins can be set to the high-impedance state or held in the state before transiting to the watch mode according to the setting of the SPL bit in the low-power consumption mode control register (LPMCR).

DataSheet4U.com

Notes:
- To set a pin to high impedance when the pin is shared by a peripheral function and a port in watch mode, disable the output of peripheral functions, and set the TMD bit to 0.
  This applies to the following pins:
  P14/PPG0, P15/PPG1, P16/PPG2, P17/PPG3, P21/TOT0, P23/TOT1
- There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Return from Watch mode

The watch mode is cancelled by a reset factor or when an interrupt is generated.

● Return by reset factor

When the watch mode is cancelled by a reset factor, the mode transits to the main clock mode after the watch mode is cancelled, transiting to the reset sequence.

Notes:
- When watch mode are returned to main clock mode using an external reset pin ($\overline{RST}$ pin), input level "L" for at least "the oscillation time of the oscillator(*) + 100µs + 16 machine cycles (main clock)".
  *: The oscillation time of the oscillator is the time required to reach 90% of amplitude. It takes several to dozens of ms for crystal oscillators, hundreds of µs to several ms for FAR/ceramic oscillators, and 0 ms for external clocks.
- There is no sub-clock in MB90F387S and MB90387S.

● Return by an interrupt

When an interrupt request higher than the interrupt level (IL) of 7 is generated from the watch timer and external interrupt in the watch mode, the watch mode is cancelled. After the watch mode is cancelled, as with normal interrupt processing, the generated interrupt request is identified according to the settings of the I flag in the condition code register (CCR), the interrupt level mask register (ILM), and the interrupt control register (ICR). In the sub-watch mode, no oscillation stabilization wait time is generated and the interrupt request is identified immediately after return from the watch mode.

et4U.com DataShee

- When the CPU is not ready to accept any interrupt request, the instruction next to the currently executing instruction is executed.

DataSheet4U.com

- When the CPU is ready to accept any interrupt request, it branches immediately to the interrupt processing routine.

Notes:
- When an interrupt processing is executed, the CPU usually proceeds to the interrupt processing after executing the instruction next to the one specifying the watch mode.
- There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.8.5.3 Timebase Timer Mode

The timebase timer mode operates only the oscillation clock (HCLK), subclock (SCLK), timebase timer, and watch timer. Resources other than the timebase timer and watch timer stop.

■ Transition to Timebase Timer Mode

The mode transits to the timebase timer mode when 0 is written to the TMD bit of the low-power consumption mode control register (LPMCR) during operation in the PLL clock mode or the main clock mode (CKSCR: SCM = 1).

- Data hold function

  In the timebase timer mode, data in the dedicated registers such as an accumulator and internal RAM are held.

- Operation when interrupts request generated

  When an interrupt request is generated with the TMD bit of the low-power consumption mode control register (LPMCR) set to 0, the mode does not transit to the timebase timer mode. When the CPU is not ready to accept any interrupt request, the instruction next to the currently executing instruction is executed. When the CPU is ready to accept any interrupt request, it branches immediately to the interrupt processing routine.

et4U.com DataShee DataSheet4U.com

- Pin state

  In the timebase timer mode, the input/output pins can be set to the high-impedance state or held in the state before transiting to the timebase timer mode according to the setting of the SPL bit in the low-power consumption mode control register (LPMCR).

Note: To set a pin to high impedance when the pin is shared by a peripheral function and a port in timebase timer mode, disable the output of peripheral functions, and set the TMD bit to 0.
This applies to the following pins:
P14/PPG0, P15/PPG1, P16/PPG2, P17/PPG3, P21/TOT0, P23/TOT1

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Return from Timebase Timer Mode

The timebase timer mode is cancelled by a reset factor or when an interrupt is generated.

● Return by reset factor

When the timebase timer mode is cancelled by a reset factor, the mode transits to the main clock mode after the timebase timer mode is cancelled, transiting to the reset sequence.

Note: When the timebase timer mode is returned to main clock mode using an external reset pin ($\overline{RST}$ pin), input level "L" for at least 100µs.

● Return by an interrupt

When an interrupt request higher than interrupt level (IL) of 7 is generated from the watch timer, timebase timer, and external interrupt in the timebase timer mode, the timebase timer mode is cancelled. After the timebase timer mode is cancelled, as with normal interrupt processing, the generated interrupt request is identified according to the settings of the I flag in the condition code register (CCR), the interrupt level mask register (ILM), and the interrupt control register (ICR).

- When the CPU is not ready to accept any interrupt request, the instruction next to the currently executing instruction is executed.
- When the CPU is ready to accept any interrupt request, it branches immediately to the interrupt processing routine.
- The following two timebase timer modes are available:
  - Main clock <-- --> timebase timer mode
  - PLL clock <-- --> timebase timer mode

et4U.com DataShee DataSheet4U.com

Notes:
- At interrupt processing, the CPU usually proceeds to the interrupt processing after executing the instruction next to the one specifying the timebase timer mode.
- When the timebase timer mode is returned by an interrupt, the interrupt processing is performed after the maximum 80 µs after the interrupt request is accepted.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.8.5.4 Stop Mode

The stop mode stops the oscillation clock (HCLK) and subclock (SCLK) during operation in each clock mode. Data can be held with the minimum power consumption.

■ Stop Mode

When 1 is written to the STP bit of the low-power consumption mode control register (LPMCR) during operation in the PLL clock mode (CKSCR : MCS=1, SCS=0), the mode transits to the stop mode according to the settings of the MCS bit and SCS bit in the clock select register (CKSCR).

Table 3.8-5 shows the settings of the MCS and SCS bits in the clock select register (CKSCR) and the stop modes.

Table 3.8-5 Settings of MCS and SCS Bits in Clock Select Register (CKSCR) and Stop Modes

Clock Select Register (CKSCR)		Stop Mode to be Transited
MCS	SCS	
1	1	Main stop mode
0	1	PLL stop mode
1	0	Sub-stop mode
0	0	

et4U.com DataShee DataSheet4U.com

Notes:
- If both the STP and SLP bits in the low-power consumption mode control register (LPMCR) are set to 1 simultaneously, the STP bit is preferred and the mode transits to the stop mode.
- There is no sub-clock in MB90F387S and MB90387S.

● Data hold function

In the stop mode, data in the dedicated registers such as accumulators and internal RAM are held.

● Operation when interrupt request generated

When an interrupt request is generated with the STP bit in the low-power consumption mode control register (LPMCR) set to 1, the mode does not transit to the stop mode. When the CPU is not ready to accept any interrupt request, the instruction next to the currently executing instruction is executed. If the CPU is ready to accept any interrupt request, it immediately branches to the interrupt processing routine.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Pin state

In the stop mode, the input/output pins can be set to the high-impedance state or held in the state before transiting to the stop mode according to the setting of the SPL bit in the low-power consumption mode control register (LPMCR).

Note: To set a pin to high impedance when the pin is shared by a peripheral function and a port in stop mode, disable the output of peripheral functions, and set the STP bit to 1.
This applies to the following pins:
P14/PPG0, P15/PPG1, P16/PPG2, P17/PPG3, P21/TOT0, P23/TOT1

■ Return from Stop Mode

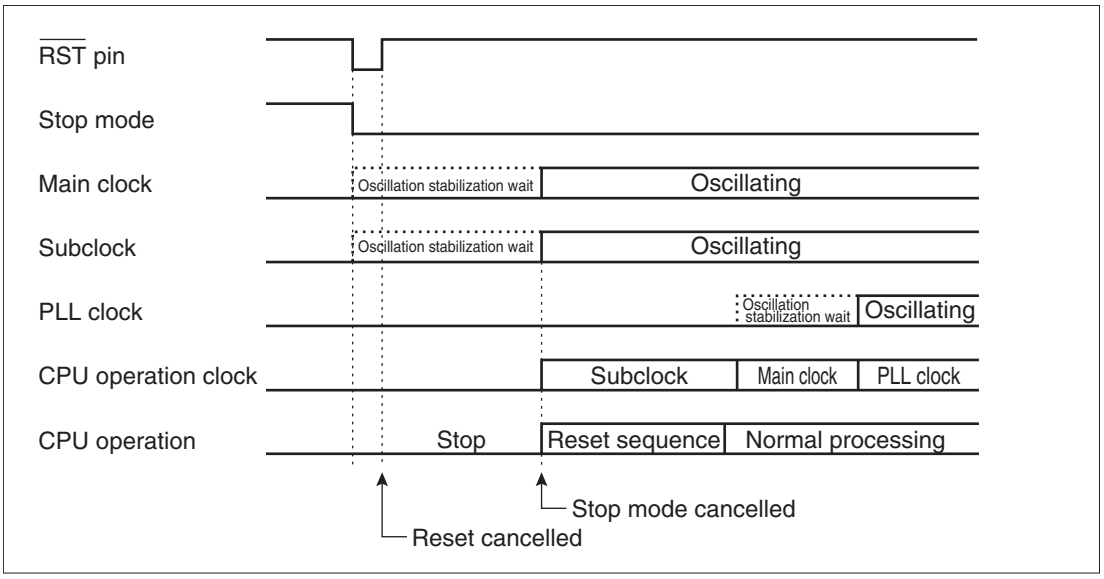
The stop mode is cancelled by a reset factor or when an interrupt is generated. At return from the stop mode, the oscillation clock (HCLK) and the subclock (SCLK) stop, so the stop mode is cancelled after the elapse of the main clock oscillation stabilization wait time or the subclock oscillation stabilization wait time.

● Return by reset factor

When the stop mode is cancelled by a reset factor, the main clock oscillation stabilization wait time is generated. After the termination of the main clock oscillation stabilization wait time, the stop mode is cancelled, transiting to the reset sequence.

Figure 3.8-7 shows the return from the sub-stop mode by an external reset.

Figure 3.8-7 Return from the Sub-stop Mode by an External Reset



Notes:
- When stop mode are returned to main clock mode using an external reset pin ($\overline{RST}$ pin), input level "L" for at least "the oscillation time of the oscillator(*) + 100µs + 16 machine cycles (main clock)".
  *: The oscillation time of the oscillator is the time required to reach 90% of amplitude. It takes several to dozens of ms for crystal oscillators, hundreds of µs to several ms for FAR/ceramic oscillators, and 0 ms for external clocks.
- There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Return by an interrupt

When an interrupt request higher than the interrupt level (IL) of 7 is generated from external interrupt in the stop mode, the stop mode is cancelled. In the stop mode, the main clock oscillation stabilization wait time or the subclock oscillation stabilization wait time is generated after the stop mode is cancelled. After the main clock oscillation stabilization wait time or the subclock oscillation stabilization wait time is terminated, as with normal interrupt processing, the generated interrupt request is identified according to the settings of the I flag in the condition code register (CCR), the interrupt level mask register (ILM), and the interrupt control register (ICR).

- When the CPU is not ready to accept any interrupt request, the instruction next to the currently executing instruction is executed.
- When the CPU is ready to accept any interrupt request, it branches immediately to the interrupt processing routine.

Notes:

- At interrupt processing, the CPU usually proceeds to the interrupt processing after executing the instruction next to the one specifying the stop mode.
- In PLL stop mode, the main clock and PLL multiplication circuit stop. During recovery from PLL stop mode, it is necessary to secure the main clock oscillation stabilization wait time and PLL clock oscillation stabilization wait time. The oscillation stabilization wait times for the main clock and PLL clock are counted simultaneously according to the value specified in the oscillation stabilization wait time selection bits (CKSCR: WS1, WS0) in the clock selection register. The oscillation stabilization wait time selection bits (CKSCR: WS1, WS0) in the clock selection register must be selected accordingly to account for the longer of main clock and PLL clock oscillation stabilization wait time.
  The PLL clock oscillation stabilization wait time, however, requires 2^{14}/HCLK or more. Set the oscillation stabilization wait time selection bits (CKSCR: WS1, WS0) in the clock selection register to "10_B" or "11_B".

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

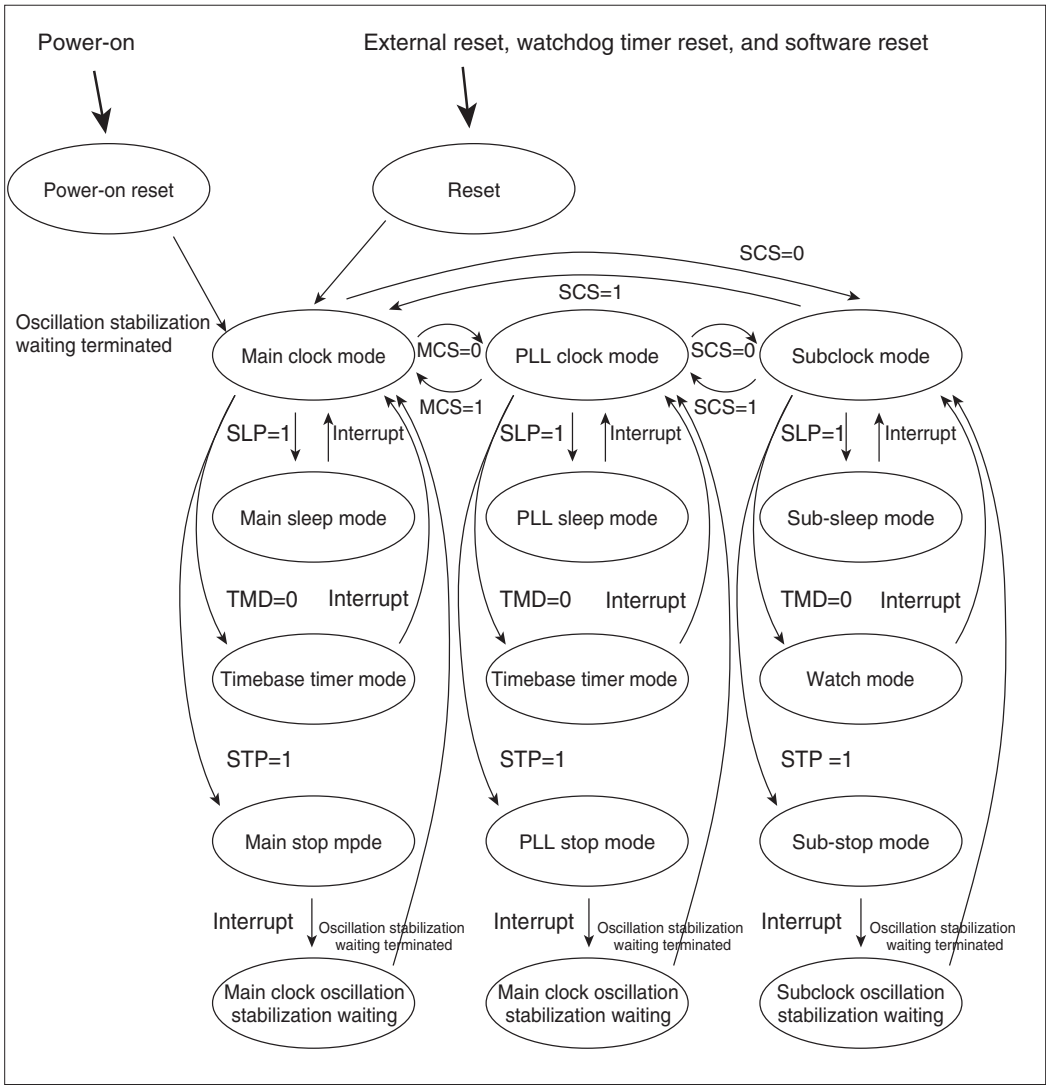
www.DataSheet4U.com

3.8.6 State Transition in Standby Mode

The operating state and state transition in the clock mode and standby mode in the MB90385 series are shown in the diagram.

■ State Transition Diagram

Figure 3.8-8 State Transition Diagram



Notes:
- In attempting to switch the clock mode, do not attempt to switch to another clock mode or low-power consumption mode until the first switching is completed. The MCM and SCM bits of the clock selection register (CKSCR) indicate that switching is completed. If the mode is switched to another clock mode or low-power-consumption mode before completion of switching, the mode may not be switched.
- There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.8.7 Pin State in Standby Mode, at Reset

The state of input/output pins in the standby mode and at reset is shown in each access mode.

■ State of Input/Output Pins (Single-chip Mode)

Table 3.8-6 State of Input/Output Pins (Single-chip Mode)

Pin Name	Sleep	Stop/Watch/Timebase timer		Reset
		SPL = 0	SPL = 1	
P07 to P10	Immediately-preceding state held *1	Input cut off/ immediately-preceding state held*1	Input cut off/ output Hi-Z *2	Input disabled/ output Hi-Z
P27 to P20				
P37 to P35, p33 to p30				
P44 to P40				
P57 to P50				

*1: Indicates that state of pins output immediately before entering each standby mode is output as it is or "input disabled." "State of pins output is output as it is" means that if the resource output is in operation, the state of pins is output according to the state of the resource and if the state of output pins is output, it is held. "Input disabled" means that no pin value can be accepted internally because the operation of the input gates of pins is enabled but the internal circuit stops.

*2: "Input cut off" means that operation of the input gates of pins is disabled and "output Hi-Z" means that the driving of pin driving transistors is disabled to set pins to the high-impedance state.

Note: To set a pin to high impedance when the pin is shared by a peripheral function and a port in stop mode, watch mode or timebase timer mode, disable the output of peripheral functions, and set the STP bit to 1 or set the TMD bit to 0.
This applies to the following pins:
P14/PPG0, P15/PPG1, P16/PPG2, P17/PPG3, P21/TOT0, P23/TOT1

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.8.8 Precautions when Using Low-power Consumption Mode

This section explains the precautions when using the low-power consumption modes.

■ Transition to Standby Mode

When an interrupt request is generated from the resource to the CPU, the mode does not transit to each standby mode even after setting the STP and SLP bits in the low-power consumption mode control register (LPMCR) to 1 and the TMD bit to 0 (and also even after interrupt processing).

If the CPU is in interrupt processing, the interrupt request flag during interrupt processing is cleared and the mode can transit to each standby mode if no other interrupt requests are generated.

■ Cancellation of Standby Mode by Interrupt

When an interrupt request higher than the interrupt level (IL) of 7 is generated from the resource and external interrupt during operation in the sleep mode, watch mode, timebase timer mode, or stop mode, the standby mode is cancelled. The standby mode is cancelled by an interrupt regardless of whether the CPU accept interrupts or not.

Notes:
- Take measures, such as disabling interrupts, not to branch to the interrupt processing immediately after return from the standby mode.
- There is no sub-clock in MB90F387S and MB90387S.

et4U.com DataShee

■ Notes on the Transition to Standby Mode

DataSheet4U.com

To set a pin to high impedance when the pin is shared by a peripheral function and a port in stop mode, watch mode, or timebase timer mode, use the following procedure:

1. Disable the output of peripheral functions.
2. Set the SPL bit to "1", STP bit to "1", or TMD bit to "0" in the low-power consumption mode control register (LPMCR).

Note: There is no sub-clock in MB90F387S and MB90387S.

■ Note on Cancelling Standby Mode

The standby mode can be cancelled by an input according to the settings of an input factor of an external interrupt. The system enters the stop mode. The input factor can be selected from High level, Low level, rising edge, and falling edge.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Oscillation Stabilization Wait Time

● Oscillation stabilization wait time of main clock

In the subclock mode, watch mode, or stop mode, the oscillation of the main clock stops and the oscillation stabilization wait time of the main clock is required. The oscillation stabilization wait time of the main clock is set by the WS1 and WS0 bits in the clock select register (CKSCR).

● Oscillation stabilization wait time of subclock

In the sub-stop mode, the oscillation of the subclock stops and the oscillation stabilization wait time of the subclock is required. The oscillation stabilization wait time of the subclock is fixed at 2^{14}/SCLK (SCLK: subclock).

● Oscillation stabilization wait time of PLL clock

In main clock mode, the PLL multiplication circuit stops. When changing to PLL clock mode, it is necessary to reserve the PLL clock oscillation stabilization wait time. The CPU runs in main clock mode till the PLL clock oscillation stabilization wait time has elapsed. When the main clock is switched to PLL clock mode, the PLL clock oscillation stabilization wait time is fixed at 2^{14}/HCLK (HCLK: oscillation clock).

In subclock mode, the main clock and PLL multiplication circuit stop. When changing to PLL clock mode, it is necessary to reserve the main clock oscillation stabilization wait time and PLL clock oscillation stabilization wait time. The oscillation stabilization wait times for main clock and PLL clock are counted simultaneously according to the value specified in the oscillation stabilization wait time selection bits (CKSCR: WS1, WS0) in the clock selection register. The oscillation stabilization wait time selection bits (CKSCR: WS1, WS0) in the clock selection register must be selected accordingly to account for the longer of the main clock and PLL clock oscillation stabilization wait times. The PLL clock oscillation stabilization wait time, however, requires 2^{14}/HCLK or more. Set the oscillation stabilization wait time selection bits (CKSCR: WS1, WS0) in the clock selection register to "10_B" or "11_B".

In PLL stop mode, the main clock and PLL multiplication circuit stop. During recovery from PLL stop mode, it is necessary to secure the main clock oscillation stabilization wait time and PLL clock oscillation stabilization wait time. The oscillation stabilization wait times for the main clock and PLL clock are counted simultaneously according to the value specified in the oscillation stabilization wait time selection bits (CKSCR: WS1, WS0) in the clock selection register. The oscillation stabilization wait time selection bits (CKSCR: WS1, WS0) in the clock selection register must be selected accordingly to account for the longer of main clock and PLL clock oscillation stabilization wait time. The PLL clock oscillation stabilization wait time, however, requires 2^{14}/HCLK or more. Set the oscillation stabilization wait time selection bits (CKSCR: WS1, WS0) in the clock selection register to "10_B" or "11_B".

■ Transition of Clock Mode

When transiting a clock mode, do not transit a clock mode to any other clock mode or a low-power consumption mode until the completion of transition. Refer the MCM and SCM bits in the clock select register (CKSCR) to check that the transition of a clock mode is completed. If the mode is switched to another clock mode or low-power-consumption mode before completion of switching, the mode may not be switched.

Note: There is no sub-clock in MB90F387S and MB90387S.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Notes on Accessing the Low-Power Consumption Mode Control Register (LPMCR) to Enter the Standby Mode

- To access the low-power consumption mode control register (LPMCR) with assembler language
  - To set the low-power consumption mode control register (LPMCR) to enter the standby mode, use the instruction listed in Table 3.8-2.
  - The low-power consumption mode transition instruction in Table 3.8-2 must always be followed by an array of instructions highlighted by a dotted line below.

```
MOV   LPMCR,#H'XX ; the low-power consumption mode transition instruction in Table 3.8-2
NOP
NOP
JMP   $+3              ; jump to next instruction
MOV   A,#H'10          ; any instruction
```

The devices do not guarantee its operation after returning from the standby mode if you place an array of instructions other than the one enclosed in the dotted line.

- To access the low-power consumption mode (LPMCR) with C language

To enter the standby mode using the low-power consumption mode control register (LPMCR), use one of the following methods (1) to (3) to access the register:

(1) Specify the standby mode transition instruction as a function and insert two _wait_nop() built-in functions after that instruction. If any interrupt other than the interrupt to return from the standby mode can occur within the function, optimize the function during compilation to suppress the LINK and UNLINK instructions from occurring.

Example: Watch mode or timebase timer mode transition function

```
void enter_watch(){
    IO_LPMCR.byte = 0x10;          /* Set LPMCR TMD bit to "0" */
    _wait_nop();
    _wait_nop();
}
```

(2) Define the standby mode transition instruction using _asm statements and insert two NOP and JMP instructions after that instruction.

Example: Transition to sleep mode

```
_asm("   MOVI: _IO_LPMCR,#H'58);   /* Set LPMCR SLP bit to "1" */
_asm("   NOP");
_asm("   NOP");
_asm("   JMP    $+3");             /* Jump to next instruction */
```

(3) Define the standby mode transition instruction between #pragma asm and #pragma endasm and insert two NOP and JMP instructions after that instruction.

Example: Transition to stop mode

```
#pragma asm
    MOV   I: _IO_LPMCR,#H'98      /* Set LPMCR STP bit to "1" */
    NOP
    NOP
    JMP   $+3                     /* Jump to next instruction */
 #pragma endasm
```

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.9 CPU Mode

The F²MC-16LX family enables the transition of operation modes and memory access modes to set the CPU operation and access modes and areas.

■ Classification of Modes

Table 3.9-1 shows the classification of operation modes and memory access modes for the F²MC-16LX family. Each mode is set by mode pins (MD2 to MD0) in reset and mode-fetched mode data.

Table 3.9-1 Classification of Modes

Operation Modes	Memory Access Modes
	Bus Modes
RUN modes	Single-chip mode (Internal-ROM internal-bus mode)
Flash serial programming mode	–
Flash memory mode	–

et4U.com DataShee

■ Operation Mode

The operation modes control the operating state of the device and are set by the mode pins (MD2 to MD0).

DataSheet4U.com

● RUN mode

The RUN mode is the normal CPU operation mode. It provides various low-power consumption modes, such as the main clock mode, PLL clock mode, and subclock mode.

Reference: For details of the low-power consumption modes, see Section "3.8 Low-power Consumption Mode".

● Flash serial programming mode and flash memory mode

Some products in the MB90385 series have user-programmable flash memory.

The flash serial programming mode is that for serially programming data to flash memory.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.9.1 Mode Pins (MD2 to MD0)

The mode pins are three external pins of MD2 to MD0, and enable a combination of these pins to set the following:

- **Operation modes (RUN mode, flash serial programming mode, flash memory mode)**
- **Reading reset vectors and mode data**

■ Setting of Mode Pins (MD2 to MD0)

Table 3.9-2 shows the settings of the mode pins.

Table 3.9-2 Setting of Mode Pins

Mode Pin*			Mode Name
MD2	MD1	MD0	
0	0	0	Setting disabled
0	0	1	
0	1	0	
0	1	1	Internal vector mode
1	0	0	Setting disabled
1	0	1	
1	1	0	Flash serial programming mode
1	1	1	Flash memory mode

*: Set MD2 to MD0: 0 = V_{SS} or 1 = V_{CC}.

et4U.com DataShee DataSheet4U.com

● Internal vector mode

Reset vectors are read from internal ROM.

● Flash serial programming mode

Flash serial programming cannot be performed just by the settings of the mode pins.

Reference: For details of flash serial programming, see "CHAPTER 19 FLASH SERIAL PROGRAMMING CONNECTION".

● Flash memory mode

This mode is set when using a parallel writer.

DataSheet4U.com www.DataSheet4U.com

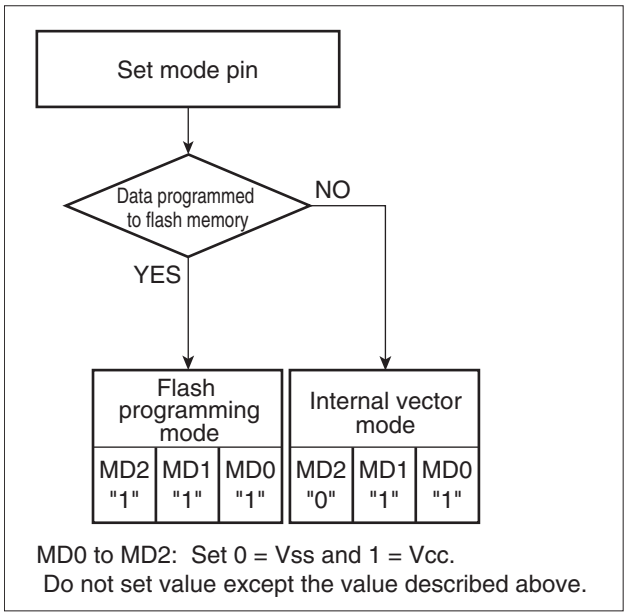
DataSheet4U.com

www.DataSheet4U.com

■ Setting Mode Pins

Set the mode pins as shown in Figure 3.9-1.

Figure 3.9-1 Flow of Mode Pin Setting



et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.9.2 Mode Data

Mode data is used to set the memory access mode. It is automatically read to the CPU by mode fetch.

■ Mode Data

The values of the mode register can be changed only in the reset sequence. The changed mode register values are enabled after the reset sequence.

Figure 3.9-2 Mode Data

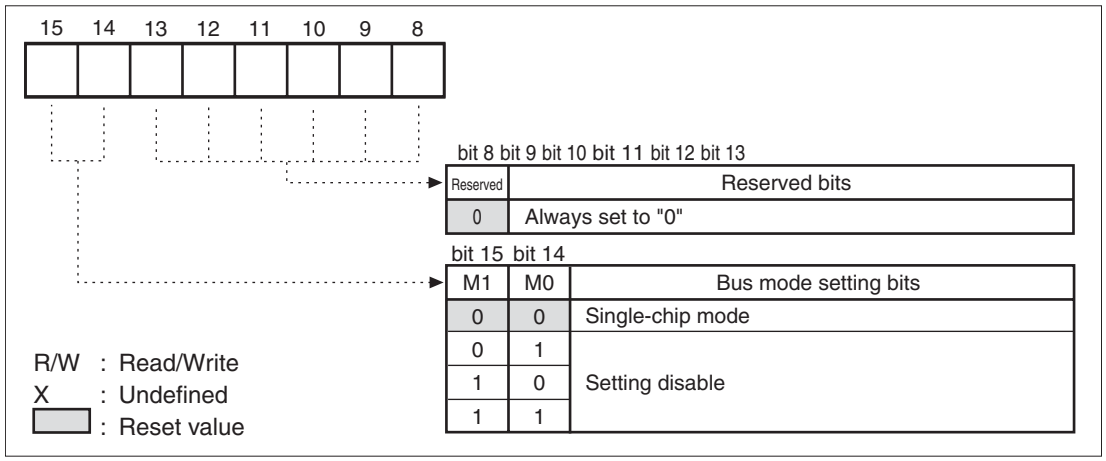


Table 3.9-3 Function of Mode Register

Bit Name		Function
bit 8 to bit 13	Reserved: Reserved bits	Always write 0 to these bits.
bit 14 bit 15	M1, M0: Bus mode setting bit	Always set these bits to "00_B".

DataSheet4U.com

www.DataSheet4U.com

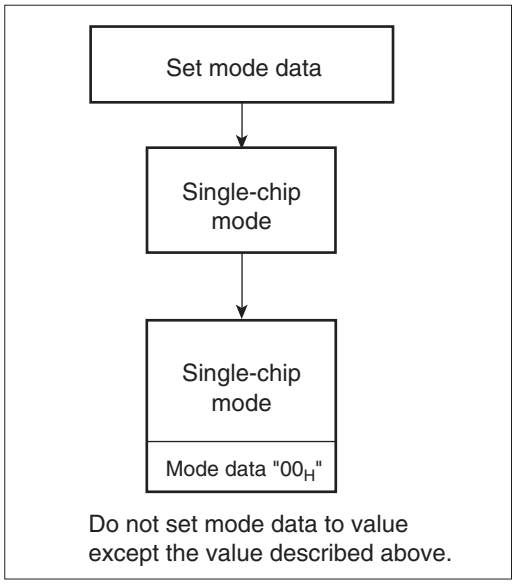
DataSheet4U.com

www.DataSheet4U.com

■ Setting Mode Data

Set mode data according to Figure 3.9-3.

Figure 3.9-3 Flow of Mode Data Setting



et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

3.9.3 Memory Access Mode

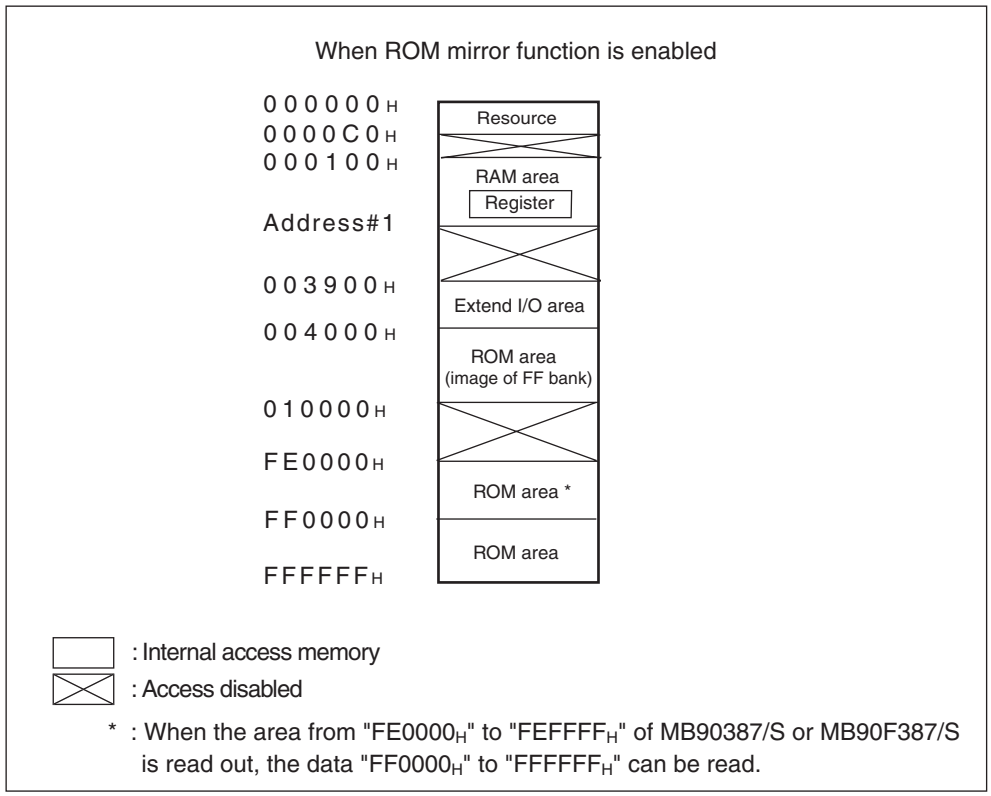
The memory access mode is the following one mode: bus mode and external access mode.

- **Bus mode: Sets access area (internal)**

■ Bus Mode

Figure 3.9-4 shows the memory map in the mode.

Figure 3.9-4 Memory map in the mode



Reference: For details of the access area, see Section "3.1 Memory Space".

● Single-chip mode (internal-ROM internal-access)

- Only internal ROM and internal RAM are used and no external access occurs.
- Ports 1 to 3 can be used as general-purpose I/O ports.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

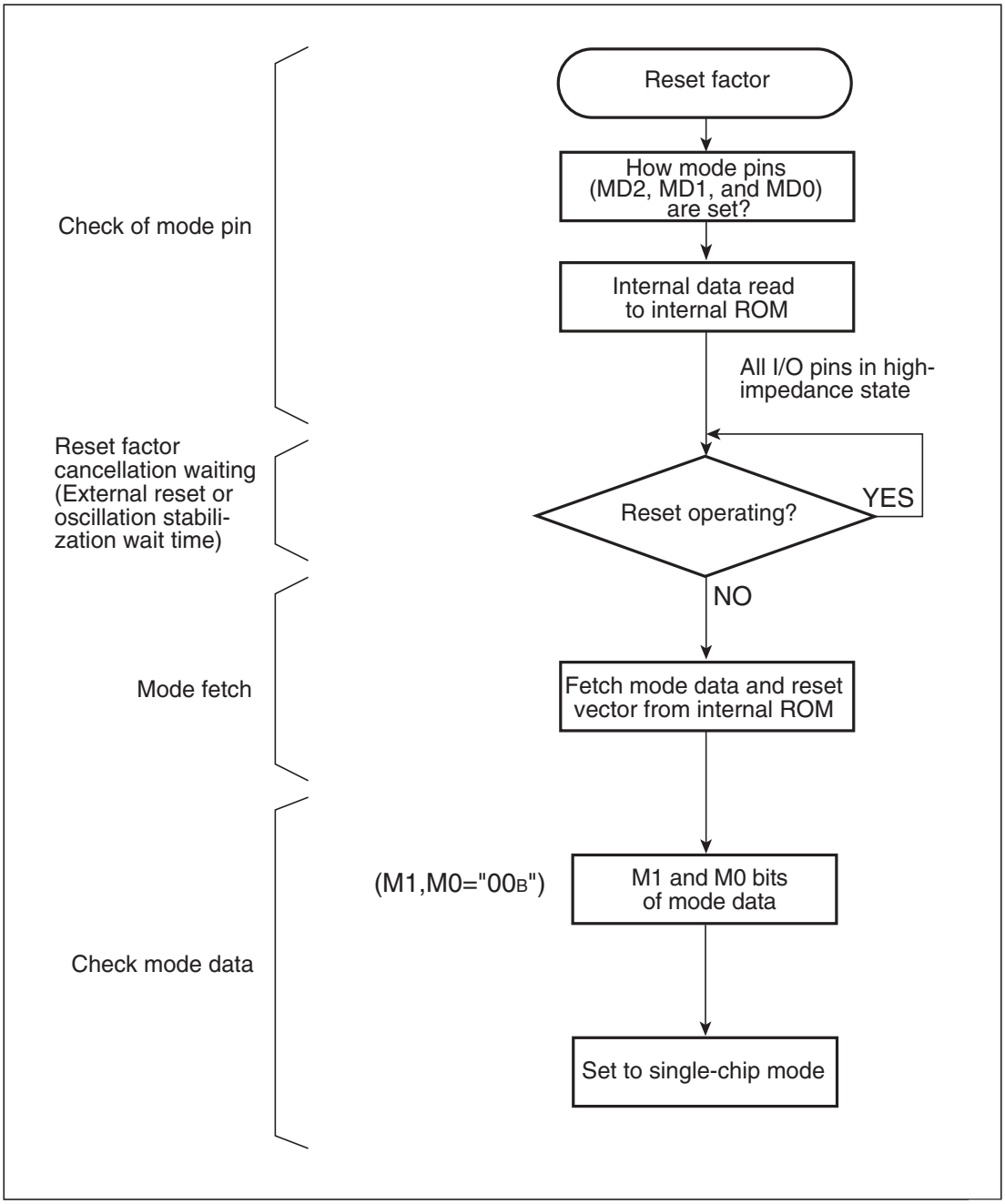
3.9.4 Selection of Memory Access Mode

This section explains selection of the memory access mode in the reset sequence.

■ Selection of Memory Access Mode

After reset is cancelled, the CPU selects the memory access mode according to the procedure shown in Figure 3.9-5 by referencing the settings of the mode pins and mode data.

Figure 3.9-5 Selection of Memory Access Mode



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 4
I/O PORT

This chapter describes the function and operation of the I/O port.

4.1 Overview of I/O Port
4.2 Registers of I/O Port
4.3 Port 1
4.4 Port 2
4.5 Port 3
4.6 Port 4
4.7 Port 5

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.1 Overview of I/O Port

I/O ports can be used as general-purpose I/O ports (parallel I/O ports). In the MB90385 series, there are five ports (34 pins).
Each port also serves as a resource I/O pins.

■ I/O Port Function

The I/O ports enable the port data register (PDR) to output data to the I/O pins from the CPU and fetch signals input to the I/O ports. These also enable the port direction register (DDR) to set a direction for the I/O pins in unit of bits.

The following shows the function of each port, and the resources that it also serves as:

- Port 1: Serves as both general-purpose I/O port and PPG timer output, or input capture input
- Port 2: Serves as both general-purpose I/O port and reload timer I/O, or external interrupt input pin
- Port 3: Serves as both general-purpose I/O port or A/D converter start trigger pin
- Port 4: Serves as both general-purpose I/O port and UART1 I/O or CAN controller transmit/receive pin
- Port 5: Serves as both general-purpose I/O port and analog input pin

Table 4.1-1 List of Each Port Functions

Port Name	Pin Name	Input Type	Output Type	Function	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
Port 1	P10/IN0 to P13/IN3	CMOS (hysteresis)	CMOS	General-purpose I/O port	P17	P16	P15	P14	P13	P12	P11	P10
	P14/PPG0 to P17/PPG3		CMOS high current	Resource	PPG3	PPG2	PPG1	PPG0	IN3	IN2	IN1	IN0
Port 2	P20/TIN0 to P27/INT7		CMOS	General-purpose I/O port	P27	P26	P25	P24	P23	P22	P21	P20
				Resource	INT7	INT6	INT5	INT4	INT3	INT2	INT1	INT0
Port 3	P30 to P33 P35/X0A to P37/ADTG			General-purpose I/O port	P37	P36* / X1A	P35* / X0A	–	P33	P32	P31	P30
				Resource	ADTG	–	–	–	–	–	–	–
Port 4	P40/SIN1 to P44/RX			General-purpose I/O port	–	–	–	P44	P43	P42	P41	P40
				Resource	–	–	–	RX	TX	SOT1	SCK1	SIN1
Port 5	P50/AN0 to P57/AN7	Analog/ CMOS (hysteresis)		General-purpose I/O port	P57	P56	P55	P54	P53	P52	P51	P50
				Analog input pin	AN7	AN6	AN5	AN4	AN3	AN2	AN1	AN0

*: If the low-speed oscillation pin is selected (for MB90387 or MB90F387), P35 and P36 pins cannot be used.

DataSheet4U.com

www.DataSheet4U.com

Note:	Port 5 also serves as analog input pins. When using these ports as general-purpose ports, always set each bit of the analog input enable register (ADER) corresponding to each pin of the ports to 0. ADER bit is 1 at a reset.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.2 Registers of I/O Port

The registers related to I/O port setting are listed as follows.

■ Registers of I/O Ports

Table 4.2-1 lists the registers of each port.

Table 4.2-1 Registers of Each Port

Register Name	Read/Write	Address	Reset Value
Port 1 data register (PDR1)	R/W	000001_H	$XXXXXXXX_B$
Port 2 data register (PDR2)	R/W	000002_H	$XXXXXXXX_B$
Port 3 data register (PDR3)	R/W	000003_H	$XXXXXXXX_B$
Port 4 data register (PDR4)	R/W	000004_H	$XXXXXXXX_B$
Port 5 data register (PDR5)	R/W	000005_H	$XXXXXXXX_B$
Port 1 direction register (DDR1)	R/W	000011_H	00000000_B
Port 2 direction register (DDR2)	R/W	000012_H	00000000_B
Port 3 direction register (DDR3)	R/W	000013_H	$000X0000_B$
Port 4 direction register (DDR4)	R/W	000014_H	$XXX00000_B$
Port 5 direction register (DDR5)	R/W	000015_H	00000000_B
Analog input enable register (ADER)	R/W	$00001B_H$	11111111_B

R/W: Read/Write
X: Undefined value

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.3 Port 1

Port 1 is a general-purpose I/O port that serves as the resource I/O pin. When the single-chip mode is set, use port 1 by switching between the resource pin and the general-purpose I/O port.
The function as a general-purpose I/O port is mainly described here. The configuration, pin assignment, block diagram of the pins, and registers for port 1 are shown below.

■ Configuration of Port 1

Port 1 consists of the following three elements:

- General-purpose I/O port, resource I/O pin (P10/IN0 to P17/PPG3)
- Port 1 data register (PDR1)
- Port 1 direction register (DDR1)

■ Pin Assignment of Port 1

- When the single-chip mode is set, use port 1 by switching between the resource pin and the general-purpose I/O port.
- Since port 1 serves as a resource pin, it cannot be used as a general-purpose I/O port when used as resources.
- When using port 1 as the input pin of the resource, set the pin corresponding to the resource in the DDR1 as an input port.
- When using port 1 as the output pin of the resource, set the output of the corresponding resource to "enabled". Port 1 functions as the output pin of the resource regardless of the settings of the DDR1

Table 4.3-1 shows the pin assignment of port 1.

Table 4.3-1 Pin Assignment of Port 1

Port Name	Pin Name	Port Function		Resource		I/O Type		Circuit Type
						Input	Output	
Port 1	P10/IN0	P10	General-purpose I/O port	IN0	Input capture input	CMOS (hysteresis)	CMOS	D
	P11/IN1	P11		IN1				
	P12/IN2	P12		IN2				
	P13/IN3	P13		IN3				
	P14/PPG0	P14		PPG0	PPG timer output		CMOS high current	G
	P15/PPG1	P15		PPG1				
	P16/PPG2	P16		PPG2				
	P17/PPG3	P17		PPG3				

Reference: For the circuit type, see Section "1.7 I/O Circuit".

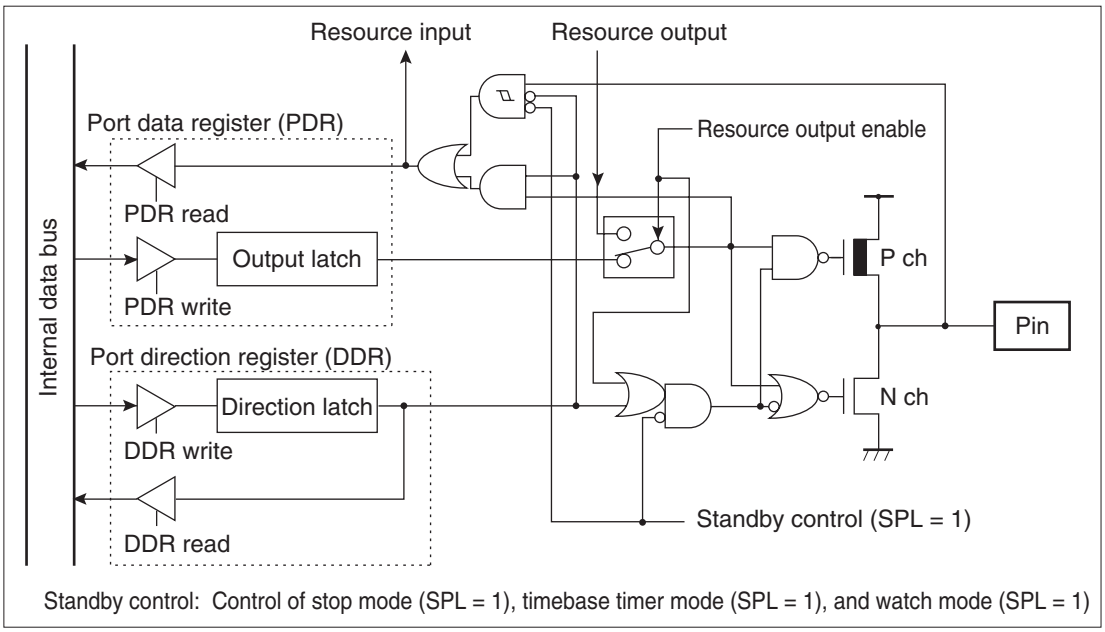
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Block Diagram of Port 1 Pins (in Single Chip Mode)

Figure 4.3-1 Block Diagram of Port 1 Pins



■ Registers for Port 1 (in Single Chip Mode)

- The registers for port 1 are PDR1 and DDR1.
- The bits composing each register correspond to the pins of port 1 one-to-one.

Table 4.3-2 shows the correspondence between the registers and pins of port 1.

Table 4.3-2 Correspondence between Registers and Pins for Port 1

Port Name	Bits of Related Registers and Corresponding Pins								
Port 1	PDR1, DDR1	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	Corresponding pin	P17	P16	P15	P14	P13	P12	P11	P10

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.3.1 Registers for Port 1 (PDR1, DDR1)

The registers for port 1 are explained.

■ Function of Registers for Port 1 (in Single Chip Mode)

● Port 1 data register (PDR1)

- Port 1 data register indicates the state of the pins.

● Port 1 direction register (DDR1)

- The port 1 direction register sets the input/output directions.
- When the bit corresponding to the pin is set to 1, port 1 functions as an output port. When the bit is set to 0, port 1 functions as an input port.

Table 4.3-3 shows the functions of the registers for port 1.

Table 4.3-3 Function of Registers for Port 1

Register Name	Data	At Read	At Write	Read/ Write	Register Address	Reset Value
Port 1 data register (PDR1)	0	The pin state is Low level.	0 is set for the output latch. When the pin is an output port pin, the Low level is output to the pin.	R/W	000001_H	$XXXXXXXX_B$
	1	The pin state is High level.	1 is set for the output latch. When the pin is an output port pin, the High level is output to the pin.			
Port 1 direction register (DDR1)	0	The direction latch is 0.	The output buffer is set to OFF, and the pin becomes an input port pin.	R/W	000011_H	00000000_B
	1	The direction latch is 1.	The output buffer is set to ON, and the pin becomes an output port pin.			

R/W: Read/Write
X: Undefined value

References:
- When using port 1 as the input pin of the resource, clear the bit in the DDR1 corresponding to the input pin of the resource to 0 and set the input pin as an input port.
- When using port 1 as the output pin of the resource, set the output of the corresponding resource to "enabled". Port 1 functions as the output pin of the resource regardless of the settings of the DDR1.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.3.2 Operation of Port 1

The operation of port 1 is explained.

■ Operation of Port 1 (in Single Chip Mode)

● Operation of output port

- When the bit in the port 1 direction register (DDR1) corresponding to the output pin is set to 1, port 1 functions as an output port.
- When the output buffer is turned ON and output data is written to the port 1 data register (PDR1), the data is retained in the output latch and output from the pin.
- When the PDR1 is read, the state of the output latch in the PDR1 is read.

Note: If read modify write instructions (such as the bit set instruction) are used to read the PDR, the pin set as an output port by the DDR outputs the desired data. However, the pin set as an input port outputs data after the input state is written to the output latch. When switching from the input port to the output port, write data to the PDR and set the pin as an output port in the DDR.

● Operation of input port

- If the bit in the DDR corresponding to the input pin is set to 0, port 1 functions as an input port.
- The output buffer is turned OFF and the pin enters the high impedance state.
- When data is written to the PDR1, it is retained in the output latch in the PDR1 but not output to the pin.
- When the PDR1 is read, the level value (Low or High) of the pin is read.

● Operation of resource output

- When using port 1 as the output pin of the resource, set the resource output to "enabled".
- Since the resource output is preferred enabled, the resource output functions regardless of the settings of the DDR1.
- When the pin state is read with the resource output set to "enabled", the output state of the resource is read.

● Operation of resource input

- The state of the pin that serves as the resource input is input to the resource.
- When using port 1 as the input pin of the resource, clear the bit in the DDR1 corresponding to the input pin of the resource to 0 and set the input pin as an input port.

● Operation at reset

- When the CPU is reset, the value of the DDR1 is cleared to 0. Consequently, all output buffers are set to OFF (the pin becomes an input port pin), and the pin enters the high-impedance state.
- The PDR1 is not initialized by reset. Therefore, when using port 1 as an output port, it is necessary to set output data in the PDR1, and then set the bit in the DDR1 corresponding to the output pin to 1, and then, to output.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Operation in stop mode, timebase timer mode or watch mode

When the pin state specification bit of the low power consumption mode control register (LPMCR: SPL) is 1, at a transition to the stop mode, timebase timer mode or watch mode, the pin enters the high-impedance state. Because the output buffer is forcibly set to OFF irrespective of the value of the DDR1.

Table 4.3-4 shows the state of the port 1 pins.

Table 4.3-4 State of Port 1 Pins

Pin Name	Normal Operation	Sleep Mode	Stop Mode, Timebase Timer Mode or Watch Mode	
			SPL=0	SPL=1
P10/IN0 to P17/PPG3	General-purpose I/O port	General-purpose I/O port	General-purpose I/O port	Input cut off, and output becomes Hi-Z (Pull-up resistor disconnected)

SPL: Pin state specification bit of low power consumption mode control register (LPMCR: SPL)
Hi-Z: High impedance

Note: To set a pin to high impedance when the pin is shared by a peripheral function and a port in stop mode, watch mode or timebase timer mode, disable the output of peripheral functions, and set the STP bit to 1 or set the TMD bit to 0.
This applies to the following pins:
P14/PPG0, P15/PPG1, P16/PPG2, P17/PPG3

DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.4 Port 2

Port 2 is a general-purpose I/O port that serves as the resource I/O pin. Use port 2 by switching between the resource pin and the general-purpose I/O port.
The function as a general-purpose I/O port is mainly described here. The configuration, pin assignment, block diagram of the pins, and registers for port 2 are shown below.

■ Configuration of Port 2

Port 2 consists of the following four elements:

- General-purpose I/O port, resource I/O pin (P20/TIN0 to P27/INT7)
- Port 2 data register (PDR2)
- Port 2 direction register (DDR2)
- High address control register (HACR)

■ Pin Assignment of Port 2

- Use port 2 by switching between the resource pin and the general-purpose I/O port.
- Since port 2 serves as resource pin, when used as a resource pin, port 2 cannot be used as general-purpose I/O port.
- When using port 2 as the input pin of the resource, set the pin corresponding to the resource in the DDR2 as an input port.
- When using port 2 as the output pin of the resource, set the output of the corresponding resource to "enabled". Port 2 functions as the output pin of the resource regardless of the settings of the DDR2.

Table 4.4-1 shows the pin assignment for port 2.

Table 4.4-1 Pin Assignment of Port 2

Port Name	Pin Name	Port Function		Resource		I/O Type		Circuit Type
						Input	Output	
Port 2	P20/TIN0	P20	General-purpose I/O port	TIN0	16-bit reload timer 0 input	CMOS (hysteresis)	CMOS	D
	P21/TOT0	P21		TOT0	16-bit reload timer 0 output			
	P22/TIN1	P22		TIN1	16-bit reload timer 1 input			
	P23/TOT1	P23		TOT1	16-bit reload timer 1 output			
	P24/INT4	P24		INT4	External interrupt input			
	P25/INT5	P25		INT5				
	P26/INT6	P26		INT6				
	P27/INT7	P27		INT7				

Reference: For the circuit type, see Section "1.7 I/O Circuit".

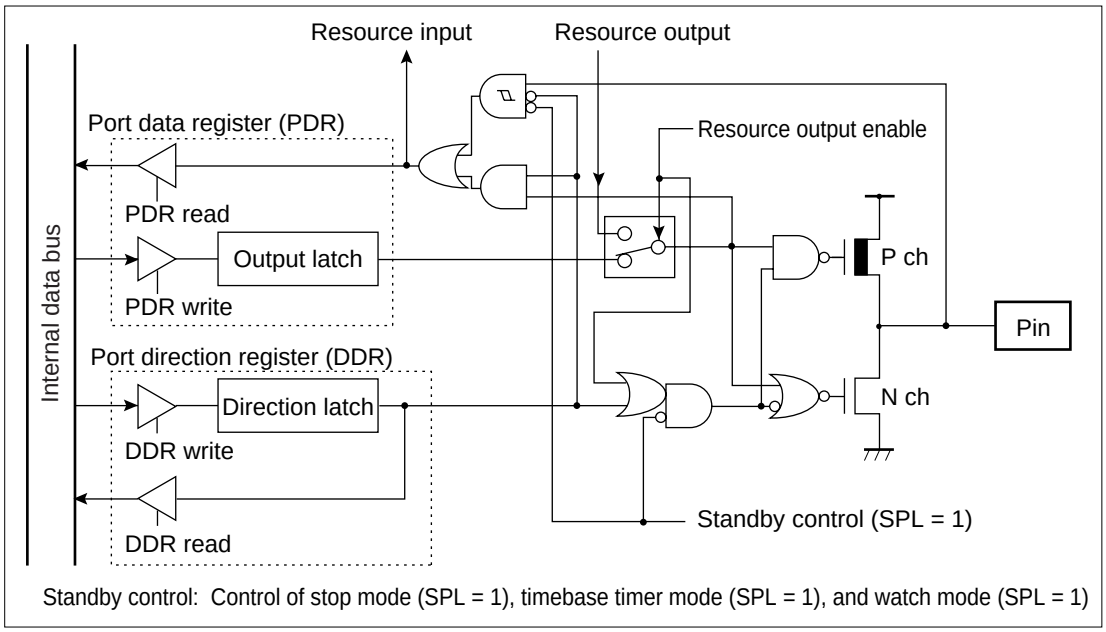
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Block Diagram of Pins of Port 2 (General-purpose I/O Port)

Figure 4.4-1 Block Diagram of Pins of Port 2



■ Registers for Port 2

- The registers for port 2 are PDR2 and DDR2.
- The bits composing each register correspond to the pins of port 2 one-to-one.

Table 4.4-2 shows the correspondence between the registers and pins of port 2.

Table 4.4-2 Correspondence between Registers and Pins for Port 2

Port Name	Bits of Related Registers and Corresponding Pins								
Port 2	PDR2, DDR2	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	Corresponding pin	P27	P26	P25	P24	P23	P22	P21	P20

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.4.1 Registers for Port 2 (PDR2, DDR2)

The registers for port 2 are explained.

■ Function of Registers for Port 2

● Port 2 data register (PDR2)

Port 2 data register indicates the input/output state of the pins.

● Port 2 direction register (DDR2)

- The port 2 direction register sets the input/output directions.
- When the bit corresponding to the pin is set to 1, port 2 functions as an output port. When the bit is set to 0, port 2 functions as an input port.

Table 4.4-3 shows the functions of the registers for port 2.

Table 4.4-3 Function of Registers for Port 2

Register Name	Data	At Read	At Write	Read/ Write	Register Address	Reset Value
Port 2 data register (PDR2)	0	The pin state is Low level.	0 is set for the output latch, and when the pin is an output port pin, the Low level is output to the pin.	R/W	000002_H	$XXXXXXXX_B$
	1	The pin state is High level.	1 is set for the output latch, and when the pin is an output port pin, the High level is output to the pin.			
Port 2 direction register (DDR2)	0	The direction latch is 0."	The output buffer is set to OFF, and the pin becomes an input port pin.	R/W	000012_H	00000000_B
	1	The direction latch is 1.	The output buffer is set to ON, and the pin becomes an output port pin.			

R/W: Read/Write
X: Undefined value

References:
- When using port 2 as the input pin of the resource, clear the bit in the DDR2 corresponding to the input pin of the resource to 0 and set the input pin as an input port.
- When using port 2 as the output pin of the resource, set the output of the corresponding resource to "enabled". Port 2 functions as the output pin of the resource regardless of the settings of the DDR2.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.4.2 Operation of Port 2

The operation of port 2 is explained.

■ Operation of Port 2 (General-purpose I/O Port)

● Operation of output port

- When the bit in the port 2 direction register (DDR2) corresponding to the output pin is set to 1, port 2 functions as an output port.
- When the output buffer is turned ON and output data is written to the port 2 data register (PDR2), the data is retained in the output latch and output from the pin.
- When the PDR2 is read, the state of the output latch in the PDR2 is read.

Note: If read modify write instructions (such as the bit set instruction) are used to read the PDR, the pin set as an output port by the DDR outputs the desired data. However, the pin set as an input port outputs data after the input state is written to the output latch. When switching from the input port to the output port, write data to the PDR and set the pin as an output port in the DDR.

● Operation of input port

- If the bit in the DDR2 corresponding to the input pin is set to 0, port 2 functions as an input port.
- The output buffer is turned OFF and the pin enters the high impedance state.
- When data is written to the PDR2, it is retained in the output latch in the PDR2 but not output to the pin.
- When the PDR2 is read, the level value (Low or High) of the pin is read.

● Operation of resource output

- When using port 2 as the output pin of the resource, set the resource output to "enabled".
- Since the resource output is preferred enabled, the resource output functions regardless of the settings of the DDR2.
- When the pin state is read with the resource output set to "enabled," the output state of the resource is read.

● Operation of resource input

- The state of the pin that serves as the input of the resource is input to the resource.
- When using port 2 as the input pin of the resource, clear the bit in the DDR2 corresponding to the input pin of the resource to 0 and set the input pin to an input port.

● Operation at reset

- When the CPU is reset, the value of the DDR2 is initialized to 0. Consequently, all output buffers are set to OFF (the pin becomes an input port pin), and the pin enters the high-impedance state.
- The PDR2 is not initialized by reset. Therefore, when using port 2 as an output port, it is necessary to set output data in the PDR2, and then set the bit in the DDR2 corresponding to the output pin to 1, and then, to output.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Operation in stop mode, timebase timer mode or watch mode

When the pin state specification bit of the low power consumption mode control register (LPMCR: SPL) is 1, at a transition to the stop mode, timebase timer mode or watch mode, the pin enters the high-impedance state. Because the output buffer is set forcibly to OFF irrespective of the value of the DDR2.

Table 4.4-4 shows the state of the port 2 pins.

Table 4.4-4 State of Port 2 Pins

Pin Name	Normal Operation	Sleep Mode	Stop Mode, Timebase Timer Mode or Watch Mode	
			SPL=0	SPL=1
P20/TIN0 to P27/INT7	General-purpose I/O port	General-purpose I/O port	General-purpose I/O port	Input cut off, and output becomes Hi-Z

SPL: Pin state specification bit of low power consumption mode control register (LPMCR: SPL)
Hi-Z: High impedance

Note: To set a pin to high impedance when the pin is shared by a peripheral function and a port in stop mode, watch mode or timebase timer mode, disable the output of peripheral functions, and set the STP bit to 1 or set the TMD bit to 0.
This applies to the following pins:
P21/TOT0, P23/TOT1

et4U.com DataShee DataSheet4U.com

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.5 Port 3

Port 3 is a general-purpose I/O port that serves as the resource I/O pin. Use port 3 by switching between the resource pin and the general-purpose I/O port.
The function as a general-purpose I/O port is mainly described here. The configuration, pin assignment, block diagram of the pins, and registers for port 3 are shown below.

■ Configuration of Port 3

Port 3 consists of the following three elements:

- General-purpose I/O port, resource input pin (P30 to P33, P35*/X0A, P36*/X1A, P37/ADTG)
- Port 3 data register (PDR3)
- Port 3 direction register (DDR3)

■ Pin Assignment of Port 3

- Use port 3 by switching between the resource pin and the general-purpose I/O port.
- Since port 3 serves as a resource pin, when used as a resource pin, port 3 cannot be used as general-purpose I/O port.
- When using port 3 as the resource input pin, set the pin corresponding to the resource in the DDR3 as an input port.

et4U.com DataShee

DataSheet4U.com

Table 4.5-1 shows the pin assignment of port 3.

Table 4.5-1 Pin Assignment of Port 3

Port Name	Pin Name	Port Function		Resource		I/O Type		Circuit Type
						Input	Output	
Port 3	P30	P30	General-purpose I/O port	–	–	CMOS (hysteresis)	CMOS	D
	P31	P31		–	–			
	P32	P32		–	–			
	P33	P33		–	–			
	P35/X0A	P35*		–	–			D/A
	P36/X1A	P36*		–	–			D/A
	P37/ADTG	P37		ADTG	External trigger input for A/D converter			D

*: If the low-speed oscillation pin is selected (for MB90387 or MB90F387), P35 and P36 pins cannot be used.

Reference: For the circuit type, see Section "1.7 I/O Circuit".

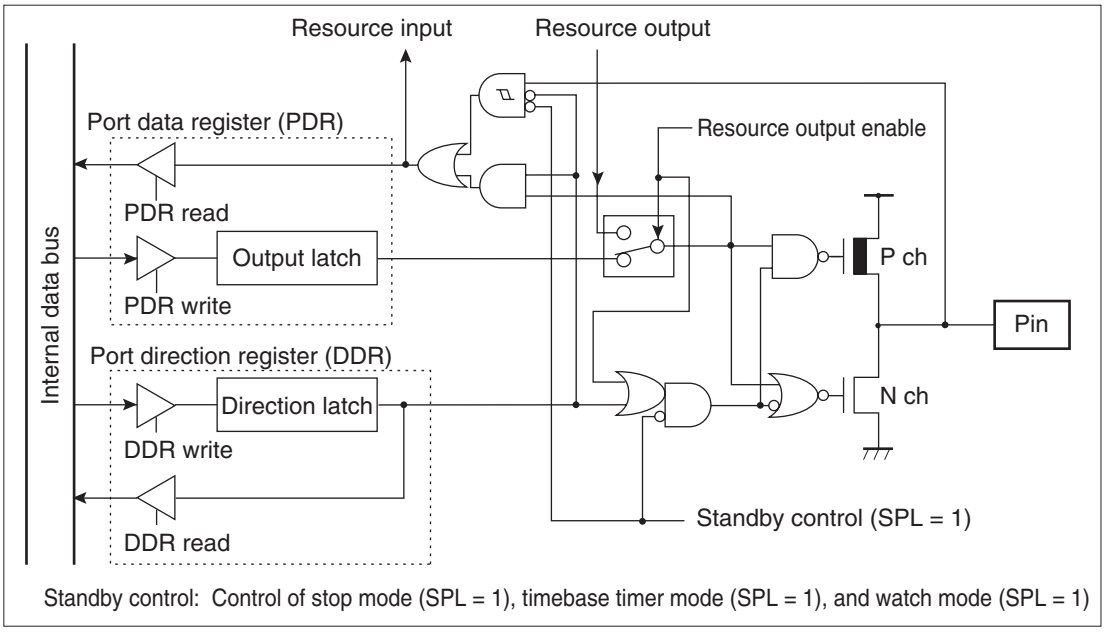
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Block Diagram of Pins of Port 3

Figure 4.5-1 Block Diagram of Pins of Port 3



■ Registers for Port 3

- The registers for port 3 are PDR3 and DDR3.
- The bits composing each register correspond to the pins of port 3 one-to-one.

Table 4.5-2 shows the correspondence between the registers and pins of port 3.

Table 4.5-2 Correspondence between Registers and Pins for Port 3

Port Name	Bits of Related Registers and Corresponding Pin								
Port 3	PDR3, DDR3	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	Corresponding pin	P37	P36*	P35*	–	P33	P32	P31	P30

*: There are no P35 and P36 pins in MB90387 and MB90F387.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.5.1 Registers for Port 3 (PDR3, DDR3)

The registers for port 3 are explained.

■ Function of Registers for Port 3

● Port 3 data register (PDR3)

- Port 3 data register indicates the state of the pins.

● Port 3 direction register (DDR3)

- The port 3 direction register sets the input/output directions.
- When the bit corresponding to the pin is set to 1, port 3 functions as an output port. When the bit is set to 0, port 3 functions as an input port.

Table 4.5-3 shows the functions of the registers for port 3.

Table 4.5-3 Function of Registers for Port 3

Register Name	Data	At Read	At Write	Read/ Write	Register Address	Reset Value
Port 3 data register (PDR3)	0	The pin state is Low level.	0 is set for the output latch, and when the pin is an output port pin, the Low level is output to the pin.	R/W	000003_H	$XXXXXXXX_B$
	1	The pin state is High level.	1 is set for the output latch, and when the pin is an output port pin, the High level is output to the pin.			
Port 3 direction register (DDR3)	0	The direction latch is 0.	The output buffer is set to OFF, and the pin becomes an input port pin.	R/W	000013_H	$000X0000_B$
	1	The direction latch is 1.	The output buffer is set to ON, and the pin becomes an output port pin			

R/W: Read/Write
X: Undefined value

References:
- When using port 3 as the input pin of the resource, clear the bit in the DDR3 corresponding to the input pin of the resource to 0 and set the input pin as an input port.
- When using port 3 as the output pin of the resource, set the output of the corresponding resource to "enabled". Port 3 functions as the output pin of the resource regardless of the settings of the DDR3.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.5.2 Operation of Port 3

The operation of port 3 is explained.

■ Operation of Port 3 (General - purpose I/O port)

● Operation of output port

- When the bit in the port 3 direction register (DDR3) corresponding to the output pin is set to 1, port 3 functions as an output port.
- When the output buffer is turned ON and output data is written to the port 3 data register (PDR3), the data is retained in the output latch and output from the pin.
- When the PDR3 is read, the state of the output latch in the PDR3 is read.

Note:	If read modify write instructions (such as the bit set instruction) are used to read the port data register (PDR), the pin set as an output port by the port direction register (DDR) outputs the desired data. However, the pin set as an input port outputs data after the input state is written to the output latch. When switching from the input port to the output port, write data to the PDR and set the pin as an output port in the DDR.

et4U.com DataShee

● Operation of input port

DataSheet4U.com

- If the bit in the DDR3 corresponding to the input pin is set to 0, port 3 functions as an input port.
- The output buffer is turned OFF and the pin enter the high impedance state.
- When data is written to the PDR3, it is retained in the output latch in the PDR3 but not output to the pin.
- When the PDR3 is read, the level value (Low or High) of the pin is read.

● Operation of resource input

- The state of the pin that serves as a resource is input to the resource.
- When using port 3 as the input pin of the resource, clear the bit in the DDR3 corresponding to the input pin of the resource to 0 and set the input pin as an input port.

● Operation at reset

- When the CPU is reset, the value of the DDR3 is cleared to 0. Consequently, all output buffers are set to OFF (the pin becomes an input port pin), and the pin enters the high-impedance state.
- The PDR3 is not initialized by reset. Therefore, when using port 3 as an output port, it is necessary to set output data in the PDR3, and then set the bit in the DDR3 corresponding to the output pin to 1 and to output.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Operation in stop mode, timebase timer mode or watch mode

- When the pin state specification bit of the low power consumption mode control register (LPMCR: SPL) is 1, at a transition to the stop mode, timebase timer mode or watch mode, the pin enters the high-impedance state. The output buffer is forcibly set to OFF irrespective of the value of the DDR3 register.

Table 4.5-4 shows the state of the port 3 pins.

Table 4.5-4 State of Port 3 Pins

Pin Name	Normal Operation	Sleep Mode	Stop Mode, Timebase Timer Mode or Watch Mode	
			SPL=0	SPL=1
P30 to P33, P35/X0A to P37/ ADTG	General-purpose I/O port	General-purpose I/O port	General-purpose I/O port	Input cut off, and output becomes Hi-Z

SPL: Pin state specification bit of low power consumption mode control register (LPMCR: SPL)
Hi-Z: High impedance

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.6 Port 4

Port 4 is a general-purpose I/O port that serves as the resource I/O pin. Use port 4 by switching between the resource pin and the general-purpose I/O port.
The function as a general-purpose I/O port is mainly described here. The configuration, pin assignment, block diagram of the pins, and registers for port 4 are shown below.

■ Configuration of Port 4

Port 4 consists of the following three elements:

- General-purpose I/O port, resource I/O pin (P40/SIN1 to P44/RX)
- Port 4 data register (PDR4)
- Port 4 direction register (DDR4)

■ Pin Assignment of Port 4

- Use port 4 by switching between the resource pin and the general-purpose I/O port.
- Since port 4 serves as a resource pin, it cannot be used as a general-purpose I/O port when used as a resource.
- When using port 4 as the input pin of the resource, set the pin corresponding to the resource in the DDR4 as an input port.
- When using port 4 as the output pin of the resource, set the output of the corresponding resource to "enabled". Port 4 functions as the output pin of the resource regardless of the settings of the DDR4.

et4U.com DataShee DataSheet4U.com

Table 4.6-1 shows the pin assignment of port 4.

Table 4.6-1 Pin Assignment of Port 4

Port Name	Pin Name	Port Function		Resource		I/O Type		Circuit Type
						Input	Output	
Port 4	P40/SIN1	P40	General-purpose I/O port	SIN1	UART1 serial data input	CMOS (hysteresis)	CMOS	D
	P41/SCK1	P41		SCK1	UART1 serial clock I/O			
	P42/SOT1	P42		SOT1	UART1 serial data output			
	P43/TX	P43		TX	CAN controller send output			
	P44/RX	P44		RX	CAN controller receive input			

Reference: For the circuit type, see Section "1.7 I/O Circuit".

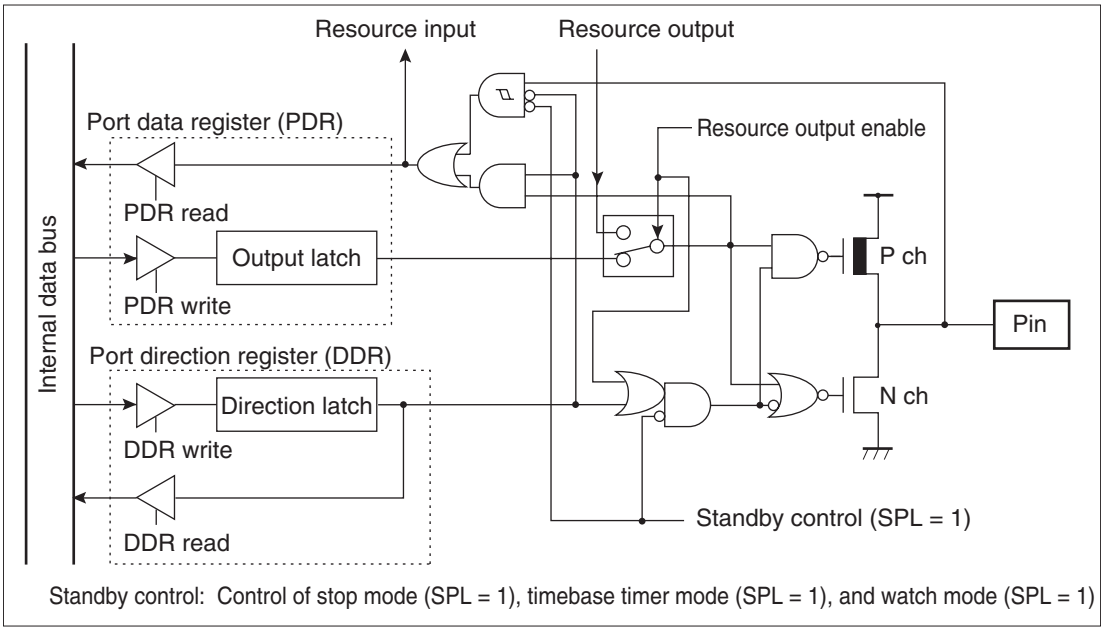
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Block Diagram of Pins of Port 4

Figure 4.6-1 Block Diagram of Pins of Port 4



et4U.com
DataShee

■ Registers for Port 4

DataSheet4U.com

- The registers for port 4 are PDR4 and DDR4.
- The bits composing each register correspond to the pins of port 4 one-to-one.

Table 4.6-2 shows the correspondence between the registers and pins of port 4.

Table 4.6-2 Correspondence between Registers and Pins for Port 4

Port Name	Bits of Related Registers and Corresponding Pins								
Port 4	PDR4, DDR4	–	–	–	bit 4	bit 3	bit 2	bit 1	bit 0
	Corresponding pin	–	–	–	P44	P43	P42	P41	P40

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.6.1 Registers for Port 4 (PDR4, DDR4)

The registers for port 4 are explained.

■ Function of Registers for Port 4

● Port 4 data register (PDR4)

- Port 4 data register indicates the state of the pins.

● Port 4 direction register (DDR4)

- The port 4 direction register sets the input/output directions.
- When the bit corresponding to the pin is set to 1, port 4 functions as an output port. When the bit is set to 0, port 4 functions as an input port.

Table 4.6-3 shows the functions of the registers for port 4.

Table 4.6-3 Function of Registers for Port 4

Register Name	Data	At Read	At Write	Read/ Write	Register Address	Reset Value
Port 4 data register (PDR4)	0	The pin state is Low level.	0 is set for the output latch. When the pin is an output port pin, the Low level is output to the pin.	R/W	000004_H	$XXXXXXXX_B$
	1	The pin state is High level.	1 is set for the output latch. When the pin is an output port pin, the High level is output to the pin.			
Port 4 direction register (DDR4)	0	The direction latch is 0.	The output buffer is set to OFF, and the pin becomes an input port pin.	R/W	000014_H	$XXX00000_B$
	1	The direction latch is 1.	The output buffer is set to ON, and the pin becomes an output port pin.			

R/W: Read/Write
X: Undefined value

References:
- When using port 4 as the input pin of the resource, clear the bit in the DDR4 corresponding to the input pin of the resource to 0 and set the input pin as an input port.
- When using port 4 as the output pin of the resource, set the output of the corresponding resource to "enabled". Port 4 functions as the output pin of the resource regardless of the settings of the DDR4.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.6.2 Operation of Port 4

The operation of port 4 is explained.

■ Operation of Port 4

● Operation of output port

- When the bit in the port 4 direction register (DDR4) corresponding to the output pin is set to 1, port 4 functions as an output port.
- When the output buffer is turned ON and output data is written to the port 4 data register (PDR4), the data is retained in the output latch and output from the pin.
- When the port 4 data register (PDR4) is read, the state of the output latch in the port 4 data register (PDR4) is read.

Note: If read modify write instructions (such as the bit set instruction) are used to read the PDR, the pin set as an output port by the DDR outputs the desired data. However, the pin set as an input port outputs data after the input state is written to the output latch. When switching from the input port to the output port, write data to the PDR and set the pin as an output port in the DDR.

et4U.com

● Operation of input port

DataShee

- If the bit in the DDR4 corresponding to the input pin is set to 0, port 4 functions as an input port.
- The output buffer is turned OFF and the pin enters the high impedance state.
- When data is written to the PDR4, it is retained in the output latch in the PDR4 but not output to the pin.
- When the PDR4 is read, the level value (Low or High) of the pin is read.

● Operation of resource output

- When using port 4 as the output pin of the resource, set the output of the corresponding resource to "enabled".
- Since the resource output is preferred enabled, the resource output functions regardless of the settings of the DDR4.
- When the pin state is read with the resource output set to "enabled", the output state of the resource is read

● Operation of resource input

- The state of the pin that serves as the input of the resource is input to the resource.
- When using port 4 as the input pin of the resource, clear the bit in the DDR4 corresponding to the input pin of the resource to 0 and set the input pin as an input port.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Operation at reset

- When the CPU is reset, the value of the DDR4 is initialized to 0. Consequently, all output buffers are set to OFF (the pin becomes an input port pin), and the pin enters the high-impedance state.
- The PDR4 is not initialized by reset. Therefore, when using port 4 as an output port, it is necessary to set output data in the PDR4, and then set the bit in the DDR4 corresponding to the output pin to 1 and to output.

● Operation in stop mode, timebase timer mode and watch mode

If the pin state specify bit (SPL) of the low-power consumption mode control register (LPMCR) is set to "1" when the CPU operation mode switches to stop mode, timebase timer mode or watch mode, the pin enters the high-impedance state. In this case, the output buffer is forcibly set to "OFF" regardless of the values of the Port 4 direction register (DDR4).

Table 4.6-4 shows the state of the port 4 pins.

Table 4.6-4 State of Port 4 Pins

Pin Name	Normal Operation	Sleep Mode	Stop Mode, Timebase Timer Mode or Watch Mode	
			SPL=0	SPL=1
P40/SIN1 to P47/RX	General-purpose I/O port	General-purpose I/O port	General-purpose I/O port	Input cut off, and output becomes Hi-Z (Pull-up resistor disconnected)

SPL: Pin state specification bit of low power consumption mode control register (LPMCR: SPL)
Hi-Z: High impedance

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.7 Port 5

Port 5 is a general-purpose I/O port that serves as the analog input pin. Use port 5 by switching between the analog input pin and the general-purpose I/O port.
The function as a general-purpose I/O port is mainly described here. The configuration, pin assignment, block diagram of the pins, and registers for port 5 are shown below.

■ Configuration of Port 5

Port 5 consists of the following four elements:

- General-purpose I/O port, analog input pins (P50/AN0 to P57AN7)
- Port 5 data register (PDR5)
- Port 5 direction register (DDR5)
- Analog input enable register (ADER)

■ Pin Assignment of Port 5

- Use port 5 by switching between the analog input pin and the general-purpose I/O port.
- Since port 5 serves as an analog input pin, it cannot be used as a general-purpose I/O port when used as an analog input pin.
- When using port 5 as an analog input pin, set the pin corresponding to the analog input in the DDR5 as an input port.
- When using port 5 as a general-purpose I/O port, do not input any analog signal.

et4U.com DataShee DataSheet4U.com

Table 4.7-1 shows the pin assignment of port 5.

Table 4.7-1 Pin Assignment of Port 5

Port Name	Pin Name	Port Function		Resource		I/O Type		Circuit Type
						Input	Output	
Port 5	P50/AN0	P50	General-purpose I/O port	AN0	Analog input channel 0	CMOS (hysteresis/ analog input)	CMOS	E
	P51/AN1	P51		AN1	Analog input channel 1			
	P52/AN2	P52		AN2	Analog input channel 2			
	P53/AN3	P53		AN3	Analog input channel 3			
	P54/AN4	P54		AN4	Analog input channel 4			
	P55/AN5	P55		AN5	Analog input channel 5			
	P56/AN6	P56		AN6	Analog input channel 6			
	P57/AN7	P57		AN7	Analog input channel 7			

Reference: For the circuit type, see Section "1.7 I/O Circuit".

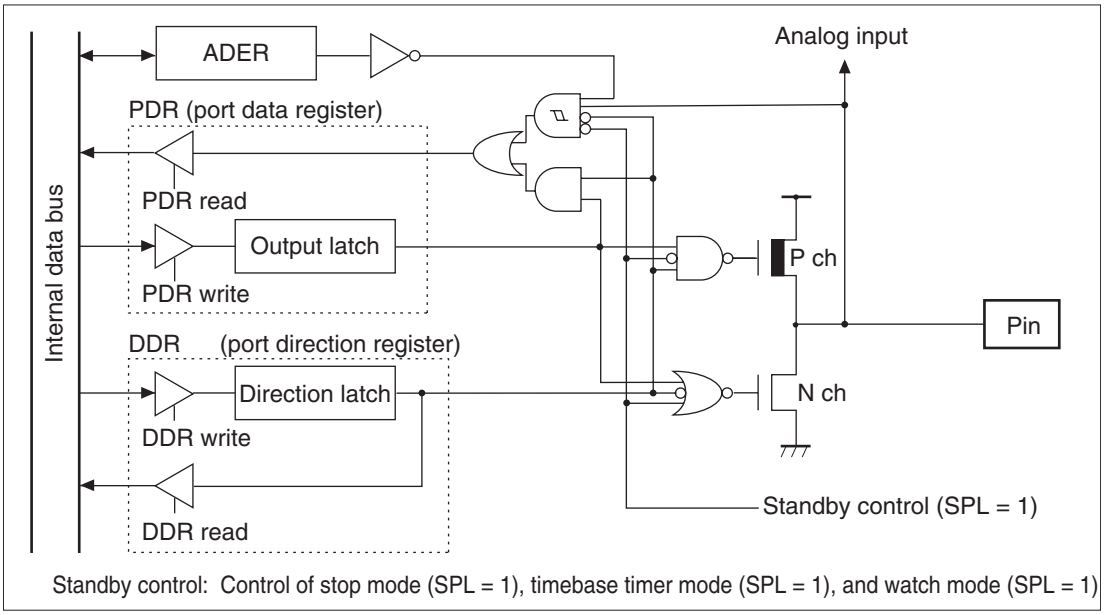
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Block Diagram of Pins of Port 5

Figure 4.7-1 Block Diagram of Pins of Port 5



■ Registers for Port 5

- The registers for port 5 are PDR5, DDR5, and ADER.
- The ADER sets input of an analog signal to the analog input pin to "enabled" or "disabled".
- The bits composing each register correspond to the pins of port 5 one-to-one.

Table 4.7-2 shows the correspondence between the registers and pins of port 5.

Table 4.7-2 Correspondence between Registers and Pins for Port 5

Port Name	Bits of Related Registers and Corresponding Pins								
Port 5	PDR5, DDR5	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
	ADER	ADE7	ADE6	ADE5	ADE4	ADE3	ADE2	ADE1	ADE0
	Corresponding pin	P57	P56	P55	P54	P53	P52	P51	P50

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.7.1 Registers for Port 5 (PDR5, DDR5, ADER)

The registers for port 5 are explained.

■ Function of Registers for Port 5

● Port 5 data register (PDR5)

- Port 5 data register indicates the state of the pins.

● Port 5 direction register (DDR5)

- The port 5 direction register sets the input/output directions.
- When the bit corresponding to the pin is set to 1, port 5 functions as an output port. When the bit is set to 0, port 5 functions as an input port.

● Analog input enable register (ADER)

- The analog input enable register (ADER) sets the general-purpose I/O ports and analog input pin in unit of ports.
- When the ADE bit corresponding to the analog input pin is set to 1, port 5 functions as an analog input pin. When the bit is set to 0, port 5 functions as a general-purpose I/O port.

Table 4.7-3 shows the functions of the registers for port 5.

Note:	When a middle-level signal is input with port 5 set as an input port, input leakage current flows. Therefore, when inputting an analog signal, set the corresponding ADE bit in the ADER to "analog input enabled."

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 4.7-3 Function of Registers for Port 5

Register Name	Data	At Read	At Write	Read/ Write	Register Address	Reset Value
Port 5 data register (PDR5)	0	The pin state is Low level.	0 is set for the output latch. When the pin is an output port pin, the Low level is output to the pin.	R/W	000005_H	$XXXXXXXX_B$
	1	The pin state is High level.	1 is set for the output latch. When the pin is an output port pin, the High level is output to the pin.			
Port 5 direction register (DDR5)	0	The direction latch is 0.	The output buffer is set to OFF, and the pin becomes an input port pin.	R/W	000015_H	00000000_B
	1	The direction latch is 1.	The output buffer is set to ON, and the pin becomes an input port pin.			
Analog input enable register (ADER)	0	General-purpose I/O port		R/W	$00001B_H$	11111111_B
	1	Analog input mode				

R/W: Read/Write
X: Undefined value

et4U.com DataShee
DataSheet4U.com

References:
- When using port 5 as the analog input pin, clear the bit in the DDR5 corresponding to the analog input pin to 0 and set the input pin as an input port.
- When using port 5 as the input pin of the resource, clear the bit in the DDR5 corresponding to the input pin of the resource to 0 and set the input pin as an input port.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

4.7.2 Operation of Port 5

The operation of port 5 is explained.

■ Operation of Port 5

● Operation of output port

- When the bit in the port 5 direction register (DDR5) corresponding to the output pin is set to 1, port 5 functions as an output port.
- When the output buffer is turned ON and output data is written to the port 5 data register (PDR5), the data is retained in the output latch and output from the pin.
- When the port 5 data register (PDR5) is read, the state of the output latch in the PDR5 is read.

Note:	If read modify write instructions (such as the bit set instruction) are used to read the PDR, the pin set as an output port by the DDR outputs the desired data. However, the pin set as an input port outputs data after the input state is written to the output latch. When switching from the input port to the output port, write data to the PDR and set the pin as an output port in the DDR.

● Operation of input port

- If the bit in the DDR5 corresponding to the input pin is set to 0, port 5 functions as an input port.
- The output buffer is turned OFF and the pin enters the high impedance state.
- When data is written to the port 5 data register (PDR5), it is retained in the output latch in the PDR5 but not output to the pin.
- When the PDR5 is read, the level value (Low or High) of the pin is read.

● Operation of analog input

- When using port 5 as an analog input pin, set the bit in the ADER corresponding to the analog input pin to 1. Port 5 is disabled to operate as a general-purpose I/O port, and functions as an analog input pin.
- When the PDR5 is read with the bit set to "analog input enabled," the read value is 0.

● Operation at reset

- When the CPU is reset, the value of the DDR5 is initialized to 0. Consequently, all output buffers are set to OFF (the pin becomes an input port pin), and the pin enters the high-impedance state.
- The PDR5 is not initialized by reset. Therefore, when using port 5 as an output port, it is necessary to set output data in the PDR5, and then set the bit in the DDR5 corresponding to the output pin to 1 and to output.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Operation in stop mode, timebase timer mode or watch mode

When the pin state specification bit of the low power consumption mode control register (LPMCR: SPL) is 1, at a transition to the stop mode, timebase timer mode or watch mode, the pin enters the high-impedance state. The output buffer is set forcibly to OFF irrespective of the value of the DDR5.

Table 4.7-4 shows the state of the port 5 pins.

Table 4.7-4 State of Port 5 Pins

Pine Name	Normal Operation	Sleep Mode	Stop Mode, Timebase Timer Mode or Watch Mode	
			SPL=0	SPL=1
P50/AN0 to P57/AN7	General-purpose I/O port	General-purpose I/O port	General-purpose I/O port	Input cut off, and output becomes Hi-Z

SPL: Pin state specification bit of low power consumption mode control register (LPMCR: SPL)
Hi-Z: High impedance

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 5

TIMEBASE TIMER

This chapter describes the function and operation of the timebase timer.

5.1 Overview of Timebase Timer
5.2 Block Diagram of Timebase Timer
5.3 Configuration of Timebase Timer
5.4 Timebase Timer Interrupt
5.5 Explanation of Operation of Timebase Timer
5.6 Precautions when Using Timebase Timer
5.7 Program Example of Timebase Timer

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

5.1 Overview of Timebase Timer

The timebase timer is an 18-bit free-run counter (timebase timer counter) that increments in synchronization with the main clock (2-divided frequency of main oscillation clock).

- **Four interval times can be selected and an interrupt request can be generated for each interval time.**
- **An operation clock is supplied to the oscillation stabilization wait time timer and other resources.**

■ Functions of Interval Timer

- When the timebase timer counter reaches the interval time set by the interval time select bits (TBTC: TBC1, TBC0), an overflow occurs (TBTC: TBOF = 1) and an interrupt request is generated.
- When an interrupt is enabled due to an overflow (carry) (TBTC: TBIE = 1), an overflow occurs (TBTC: TBOF = 1) and an interrupt is generated.
- The timebase timer has four interval times that can be selected. Table 5.1-1 shows the interval times of the timebase timer.

Table 5.1-1 Interval Times of Timebase Timer

Count Clock	Interval Time
2/HCLK(0.5 µs)	2^{12}/HCLK (approx. 1.0 ms)
	2^{14}/HCLK (approx. 4.1 ms)
	2^{16}/HCLK (approx. 16.4 ms)
	2^{19}/HCLK (approx. 131.1 ms)

HCLK: Oscillation clock
The parenthesized values are provided at 4-MHz oscillation clock.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Clock Supply

The timebase timer supplies an operation clock to the resources such as an oscillation stabilization wait time timer, PPG timer, and watchdog timer. Table 5.1-2 shows the clock cycles supplied from the timebase timer.

Table 5.1-2 Clock Cycles Supplied from Timebase Timer

Where to Supply Clock	Clock Cycle
Oscillation stabilization wait time	2^{10}/HCLK (approx. 256 µs)
	2^{13}/HCLK (approx. 2.0 ms)
	2^{15}/HCLK (approx. 8.2 ms)
	2^{17}/HCLK (approx. 32.8 ms)
Watchdog timer	2^{12}/HCLK (approx. 1.0 ms)
	2^{14}/HCLK (approx. 4.1 ms)
	2^{16}/HCLK (approx. 16.4 ms)
	2^{19}/HCLK (approx. 131.1 ms)
PPG Timer	2^{9}/HCLK (approx. 128 µs)

HCLK: Oscillation clock
The parenthesized values are provided at 4-MHz oscillation clock.
Note: Since the oscillation cycle is unstable immediately after oscillation starts, the oscillation stabilization wait time values are given as a guide.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

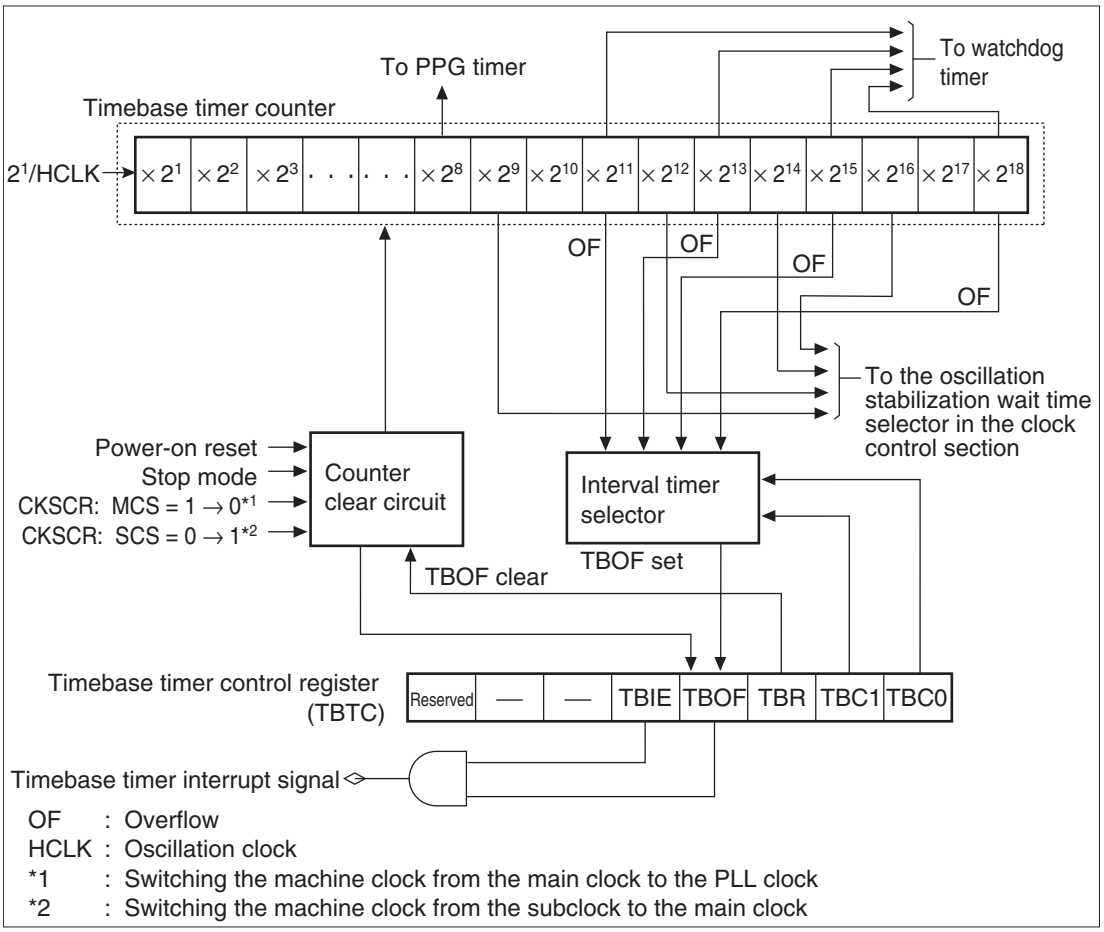
5.2 Block Diagram of Timebase Timer

The timebase timer consists of the following blocks:

- **Timebase timer counter**
- **Counter clear circuit**
- **Interval timer selector**
- **Timebase timer control register (TBTC)**

■ Block Diagram of Timebase Timer

Figure 5.2-1 Block Diagram of Timebase Timer



The actual interrupt request number of the timebase timer is as follows:

Interrupt request number: #16 (10_H)

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Timebase timer counter

The timebase timer counter is an 18-bit up counter that uses a clock with 2-divided frequency of the oscillation clock (HCLK) as a count clock.

● Counter clear circuit

The counter clear circuit clears the value of the timebase timer counter by the following factors:

- Timebase timer counter clear bit in the timebase timer control register (TBTC: TBR = 0)
- Power-on reset
- Transition to main stop mode or PLL stop mode (CKSCR: SCS = 1, LPMCR: STP = 1)
- Clock mode switching (from main clock mode to PLL clock mode, from subclock mode to PLL clock mode, or from subclock mode to main clock mode)

● Interval timer selector

The interval timer selector selects the output of the timebase timer counter from four types. When incrementing causes the selected interval time bit to overflow (carrying), an interrupt request is generated.

● Timebase timer control register (TBTC)

The timebase timer control register (TBTC) selects the interval time, clears the timebase timer counter, enables or disables interrupts, and checks and clears the state of an interrupt request.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

5.3 Configuration of Timebase Timer

This section explains the registers and interrupt factors of the timebase timer.

■ List of Registers and Reset Values of Timebase Timer

Figure 5.3-1 List of Registers and Reset Values of Timebase Timer

	bit	15	14	13	12	11	10	9	8
Timebase timer control register (TBTC)		1	X	X	0	0	1	0	0

■ Generation of Interrupt Request from Timebase Timer

When the selected interval timer counter bit reaches the interval time, the overflow interrupt request flag bit in the timebase timer control register (TBTC: TBOF) is set to 1. If the overflow interrupt request flag bit is set (TBTC: TBOF = 1) when the interrupt is enabled (TBTC: TBIE = 1), the timebase timer generates an interrupt request.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSh4U.com

www.DataSheet4U.com

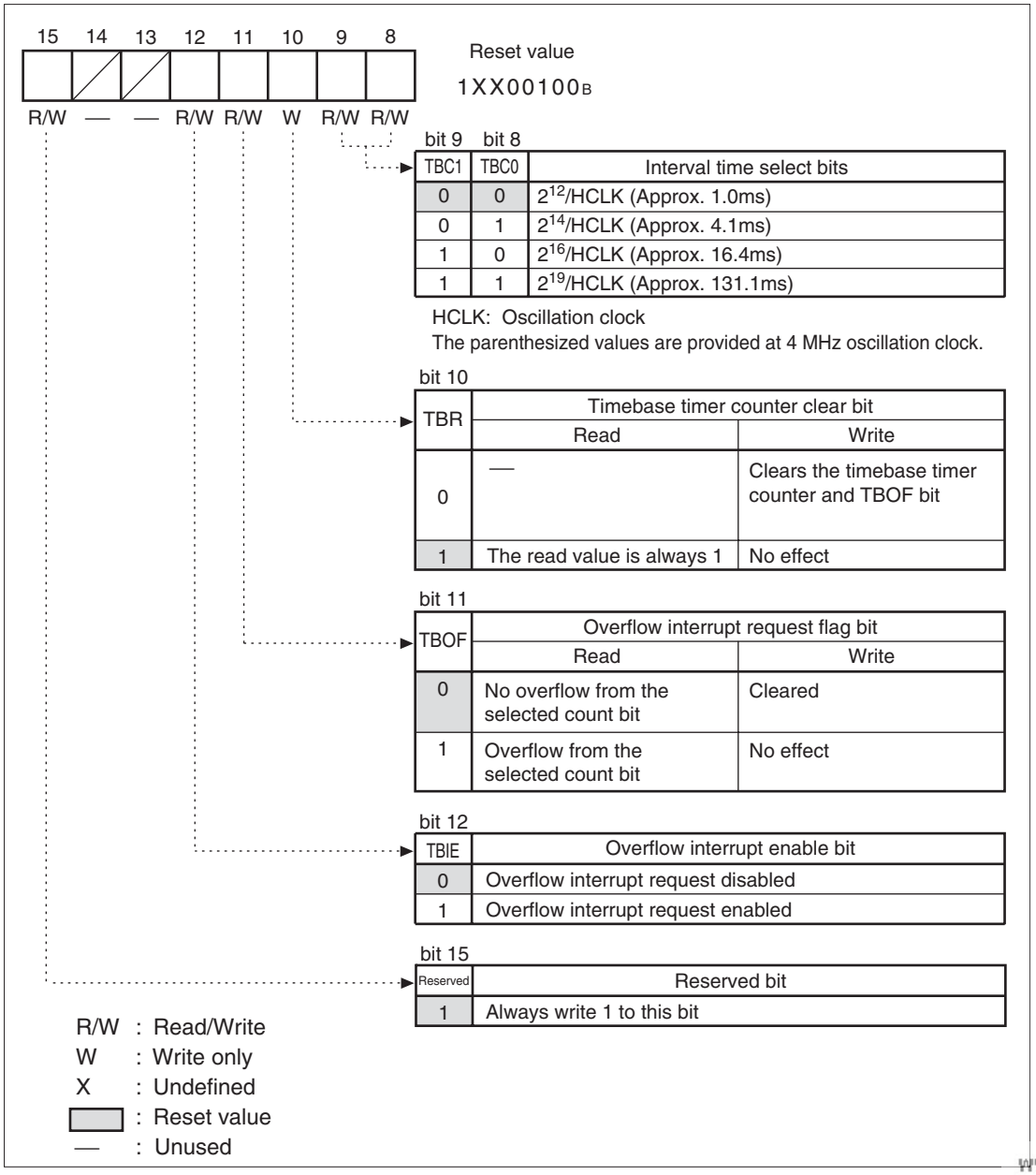
5.3.1 Timebase Timer Control Register (TBTC)

The timebase timer control register (TBTC) provides the following settings:

- **Selecting the interval time of the timebase timer**
- **Clearing the count value of the timebase timer**
- **Enabling or disabling the interrupt request when an overflow occurs**
- **Checking and clearing the state of the interrupt request flag when an overflow occurs**

■ Timebase Timer Control Register (TBTC)

Figure 5.3-2 Timebase Timer Control Register (TBTC)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 5.3-1 Functions of Timebase Timer Control Register (TBTC)

Bit Name		Function
bit 8 bit 9	TBC1, TBC0: Interval time select bits	These bits set the cycle of the interval timer in the timebase timer counter. • The interval time of the timebase timer is set according to the setting of the TBC1 and TBC0 bits. • Four interval times can be set.
bit 10	TBR: Timebase timer counter clear bit	This bit clears all the bits in the timebase timer counter. **When set to 0:** All the bits in the timebase timer counter are cleared to 0. The TBOF bit is also cleared. **When set to 1:** Disabled. The state remains unchanged. **Read:** 1 is always read.
bit 11	TBOF: Overflow interrupt request flag bit	This bit indicates an overflow (carrying) in the interval timer bit in the timebase timer counter. When an overflow (carrying) occurs (TBOF = 1) with interrupts enabled (TBIE = 1), an interrupt request is generated. **When set to 0:** The bit is cleared. **When set to 1:** Disabled. The state remains unchanged. Reading by read-modify-write type instructions always reads "1". **Notes:** 1. To clear the TBOF bit, disable interrupts (TBIE = 0) or mask interrupts using the interrupt mask register (ILM) in the processor status. 2. The TBOF bit is cleared when 0 is written to the bit, a transition to main stop mode, a transition to PLL stop mode, a transition from subclock mode to main clock mode, a transition from subclock mode to PLL clock mode, or a transition from main clock mode to PLL clock mode occurs, 0 is written to the timebase timer counter clear bit (TBR), or by a reset.
bit 12	TBIE: Overflow interrupt enable bit	This bit enables or disables an interrupt when the interval timer bit in the timebase timer counter overflows. **When set to 0:** No interrupt request is generated at an overflow (TBOF = 1). **When set to 1:** An interrupt request is generated at an overflow (TBOF = 1).
bit 13 bit 14	Unused bits	**Read:** The value is undefined. **Write:** No effect
bit 15	Reserved: Reserved bit	Always set this bit to 1.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

5.4 Timebase Timer Interrupt

The timebase timer generates an interrupt request when the interval time bit in the timebase timer counter corresponding to the interval time set by the timebase timer control register overflows (carries) (interval timer function).

■ Timebase Timer Interrupt

- The timebase timer continues incrementing for as long as the main clock (with 2-divided frequency of the oscillation clock) is input.
- When the interval time set by the interval time select bits in the timebase timer control register (TBTC: TBC1, TBC2) is reached, the interval time select bit corresponding to the interval time selected in the timebase timer counter overflows.
- When the interval time select bit overflows, the overflow interrupt request flag bit in the timebase timer control register (TBTC: TBOF) is set to 1.
- When the overflow interrupt request flag bit in the timebase timer control register is set (TBTC: TBOF = 1) with an interrupt enabled (TBTC: TBIE = 1), an interrupt request is generated.
- When the selected interval time is reached, the overflow interrupt request flag bit in the timebase timer control register (TBTC: TBOF) is set regardless of whether an interrupt is enabled or disabled (TBTC: TBIE).
- To clear the overflow interrupt request flag bit (TBTC: TBOF), disable a timebase timer interrupt at interrupt processing (TBTC : TBIE=0) or mask a timebase timer interrupt by using the ILM bit in the processor status (PS) to write 0 to the TBOF bit.

Note:	When an interrupt is enabled (TBTC: TBIE = 1) with the overflow interrupt request flag bit in the timebase timer control register set (TBTC: TBOF = 1), an interrupt request is generated immediately.

■ Correspondence between Timebase Timer Interrupt and EI^{2}OS

- The timebase timer does not correspond to EI^{2}OS.
- For details of the interrupt number, interrupt control register, and interrupt vector address, see "3.5 Interrupt".

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

5.5 Explanation of Operation of Timebase Timer

The timebase timer operates as an interval timer or an oscillation stabilization wait time timer, and supplies a clock to resources.

■ Interval Timer Function

Interrupt generation at every interval time enables the timebase timer to be used as an interval timer.

Operating the timebase timer as an interval timer requires the settings shown in Figure 5.5-1.

● Setting of timebase timer

Figure 5.5-1 Setting of Timebase Timer

Timebase timer control register (TBTC)

bit 15	14	13	12	11	10	9	8
Reserved	-	-	TBIE	TBOF	TBR	TBC1	TBC0
1			◎	0	0	◎	◎

- : Unused bit
◎: Used bit
0 : Set 0
1 : Set 1

DataSheet4U.com

● Operation as interval timer function

The timebase timer can be used as an interval timer by generating an interrupt at every set interval time.

- The timebase timer continues incrementing in synchronization with the main clock (2-divided frequency of the oscillation clock) while the oscillation clock is active.
- When the timebase timer counter reaches the interval time set by the interval time select bits in the timebase timer control register (TBTC: TBC1, TBC0), it causes an overflow (carrying) and the overflow interrupt request flag bit (TBTC: TBOF) is set to 1.
- When the overflow interrupt request flag bit is set (TBTC: TBOF = 1) with interrupts enabled (TBTC: TBIE = 1), an interrupt request is generated.

Note: The interval time may become longer than the one set by clearing the timebase timer counter.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

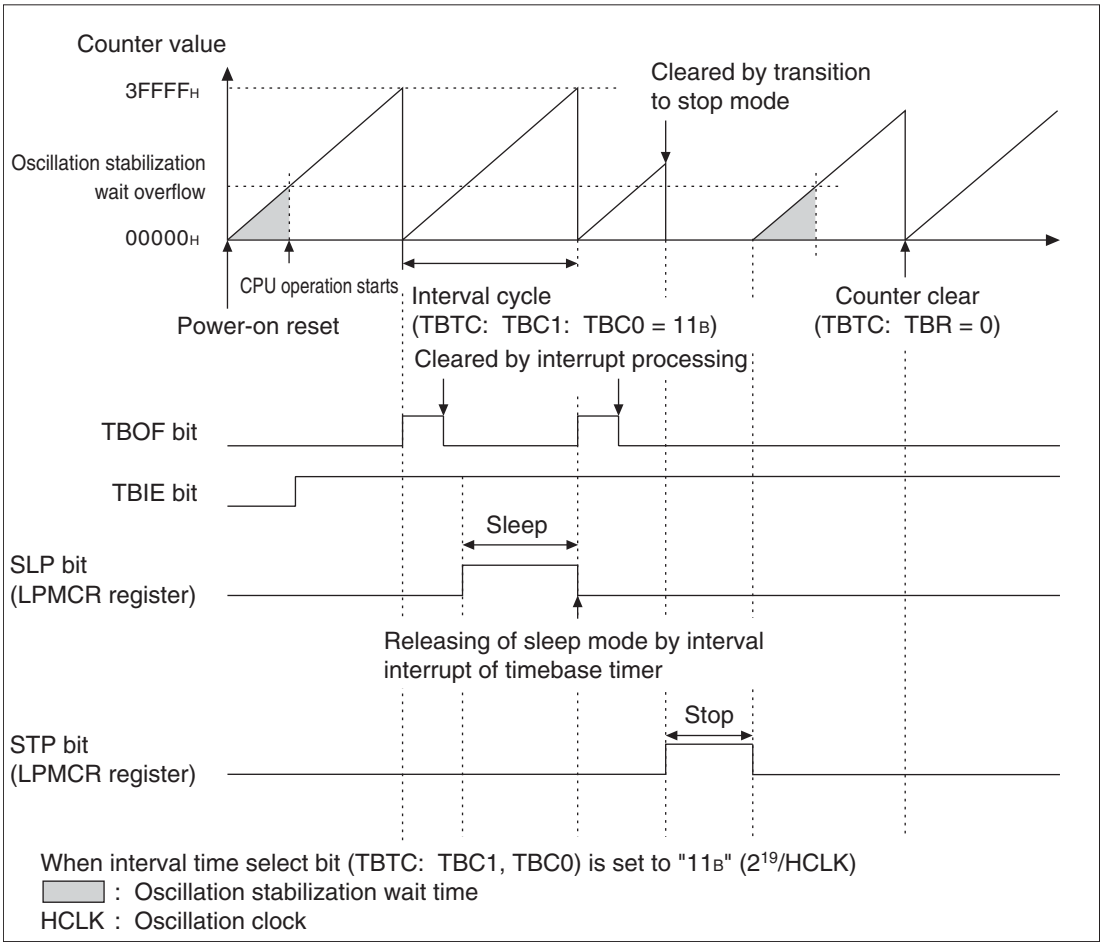
● Example of operation of timebase timer

Figure 5.5-2 gives an example of the operation that the timebase timer performs under the following conditions:

- A power-on reset occurs.
- The mode transits to the sleep mode during the operation of the interval timer.
- The mode transits to the stop mode during the operation of the interval timer.
- A request to clear the timebase timer counter is issued.

At transition to the stop mode, the timebase timer counter is cleared to stop counting. At return from the stop mode, the timebase timer counts the oscillation stabilization wait time of the main clock.

Figure 5.5-2 Example of Operation of Timebase Timer



■ Operation as Oscillation Stabilization Wait Time Timer

The timebase timer can be used as the oscillation stabilization wait time timer of the main clock and PLL clock.

- The oscillation stabilization wait time is the time elapsed from when the timebase timer counter increments from 0 until the set oscillation stabilization wait time select bit overflows (carrying).

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 5.5-1 shows clearing conditions and oscillation stabilization wait time of timebase timer.

Table 5.5-1 Clearing Conditions and Oscillation Stabilization Wait Time of Timebase Timer

Operation	Counter Clear	TBOF Clear	Oscillation Stabilization Wait Time
Writing 0 to timebase timer counter clear bit (TBTC: T BR)	O	O	
Reset			
Power on reset	O	O	Transition to main clock mode after oscillation stabilization wait time of main clock completed
Watchdog reset	X	O	Not provided
External reset	X	O	Not provided
Software reset	X	O	Not provided
Switching between clock modes			
Main clock --> PLL clock (CKSCR: MCS = 1 --> 0)	O	O	Transition to PLL clock mode after oscillation stabilization wait time of PLL clock completed
Main clock --> subclock (CKSCR: SCS = 1 --> 0)	X	X	Transition to subclock mode after oscillation stabilization wait time of subclock completed
Subclock --> main clock (CKSCR: SCS = 0 --> 1)	O	O	Transition to main clock mode after oscillation stabilization wait time of main clock completed
Subclock --> PLL clock (CKSCR: MCS = 0, SCS = 0 --> 1)	O	O	Transition to PLL clock mode after oscillation stabilization wait time of main clock completed
PLL clock --> main clock (CKSCR: MCS = 0 --> 1)	X	X	Not provided
PLL clock --> subclock (CKSCR: MCS = 0, SCS = 1 --> 0)	X	X	Not provided
Cancellation of stop modes			
Cancellation of main stop mode	O	O	Transition to main clock mode after oscillation stabilization wait time of main clock completed
Cancellation of sub-stop mode	X	X	Transition to subclock mode after oscillation stabilization wait time of subclock completed
Cancellation of PLL stop mode	O	O	Transition to PLL clock mode after oscillation stabilization wait time of main clock completed
Cancellation of watch mode			
Cancellation of sub-watch mode	X	X	Not provided
Cancellation of timebase timer modes			
Return to main clock mode	X	X	Not provided
Return to subclock mode	X	X	Not provided
Return to PLL clock mode	X	X	Not provided
Cancellation of sleep modes			
Cancellation of main sleep mode	X	X	Not provided
Cancellation of sub-sleep mode	X	X	Not provided
Cancellation of PLL sleep mode	X	X	Not provided

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Supply of Operation Clock

The timebase timer supplies an operation clock to the PPG timers (PPG01, PPG23) and the watchdog timer.

Note:	Clearing the timebase timer counter may affect the operation of the resources such as the watchdog timer and PPG timers using the output of the timebase timer.

Reference:	For details of the PPG timers, see "CHAPTER 10 8-/16-BIT PPG TIMER". For details of the watchdog timer, see "CHAPTER 6 WATCHDOG TIMER".

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

5.6 Precautions when Using Timebase Timer

This section explains the precautions when using the timebase timer.

■ Precautions when Using Timebase Timer

● Clearing interrupt request

To clear the overflow interrupt request flag bit in the timebase timer control register (TBTC: TBOF = 0), disable interrupts (TBTC: TBIE = 0) or mask the timebase timer interrupt by using the interrupt level mask register in the processor status.

● Clearing timebase timer counter

Clearing the timebase timer counter affects the following operations:

- When the timebase timer is used as the interval timer (interval interrupt).
- When the watchdog timer is used.
- When the clock supplied from the timebase timer is used as the operation clock of the PPG timer.

● Using timebase timer as oscillation stabilization wait time timer

- After power on or in the main stop mode, PLL stop mode, and subclock mode, the oscillation clock stops. Therefore, when oscillation starts, the timebase timer requires the oscillation stabilization wait time of the main clock. An appropriate oscillation stabilization wait time must be selected according to the types of oscillators connected to high-speed oscillation input pins.

Reference:	For details of the oscillation stabilization wait time, see "3.7.5 Oscillation Stabilization Wait Time".

● Resources to which timebase timer supplies clock

- At transition to operation modes (PLL stop mode, subclock mode, and main stop mode) in which the oscillation clock stops, the timebase timer counter is cleared and the timebase timer stops.
- When the timebase timer counter is cleared, an after-clearing interval time is needed. It may cause the clock supplied from the timebase timer to have a short High level or a 1/2 cycle longer Low level.
- The watchdog timer performs normal counting because the watchdog timer counter and timebase timer counter are cleared simultaneously.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

5.7 Program Example of Timebase Timer

This section gives a program example of the timebase timer.

■ Program Example of Timebase Timer

● Processing specification

The 2^{12}/HCLK (HCLK: oscillation clock) interval interrupt is generated repeatedly. In this case, the interval time is approximately 1.0 ms (at 4-MHz operation).

● Coding example

```
ICR02   EQU  0000B2H           ; Timebase timer interrupt control register
TBTC    EQU  0000A9H           ; Timebase timer control register
TBOF    EQU  TBTC:3            ; Interrupt request flag bit
TBIE    EQU  TBTC:2            ; Interrupt enable bit
;-----Main program-------------------------------------------------------------
CODE    CSEG
START:                         ; Stack pointer (SP) already initialized
        AND  CCR,#0BFH         ; Interrupts disabled
        MOV  I:ICR02 #00H      ; Interrupt level 0 (highest)
        MOV  I:TBTC,#10000000B ; Upper 3 bits fixed
                               ; TBOF cleared,
                               ; Counter cleare interval time
                               ; 2^12/HCLK selected
        SETB I:TBIE            ; Interrup enabled
        MOV  ILM,#07H          ; ILM in PS set to level 7
        OR   CCR,#40H          ; Interrupts enabled
LOOP:   MOV  A,#00H            ; Infinite loop
        MOV  A,#01H
        BRA  LOOP
;-----Interrupt program---------------------------------------------------------
WARI:
        CLRB I:TBIE            ; Interrupt enabled bit cleared
        CLRB I:TBOF            ; Interrupt request flag cleared
         :
        Processing by user
         :
        SETB I:TBIE            ; Interrupt enabled
        RETI                   ; Return from interrupt
CODE    ENDS
;-----Vector setting------------------------------------------------------------
VECT    CSEG ABS=0FFH
        ORG  0FFBCH            ; Vector set to interrupt #16 (10H)
        DSL  WARI
        ORG  0FFDCH            ; Reset vector set
        DSL  START
        DB   00H               ; Set to single-chip mode
VECT    ENDS
        END  START
```

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 6

WATCHDOG TIMER

This chapter describes the function and operation of the watchdog timer.

6.1 Overview of Watchdog Timer
6.2 Configuration of Watchdog Timer
6.3 Watchdog Timer Registers
6.4 Explanation of Operation of Watchdog Timer
6.5 Precautions when Using Watchdog Timer
6.6 Program Examples of Watchdog Timer

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

6.1 Overview of Watchdog Timer

The watchdog timer is a 2-bit counter that uses the timebase timer or watch timer as a count clock. If the counter is not cleared within a set interval time, the CPU is reset.

■ Functions of Watchdog Timer

- The watchdog timer is a timer counter that is used to prevent program malfunction. When the watchdog timer is started, the watchdog timer counter must continue to be cleared within a set interval time. If the set interval time is reached without clearing the watchdog timer counter, the CPU is reset.
- The interval time of the watchdog timer depends on the clock cycle input as a count clock and a watchdog reset occurs between the minimum and maximum times.
- The clock source output destination is set by the watchdog clock select bit in the watch timer control register (WTC: WDCS).
- The interval time of the watchdog timer is set by the timebase timer output select bit/watch timer output select bit in the watchdog timer control register (WDTC: WT1, WT0).

Table 6.1-1 lists the interval times of the watchdog timer.

Table 6.1-1 Interval Time of Watchdog Timer

Min.	Max.	Clock cycle	Min.	Max.	Clock cycle
Approx. 3.58 ms	Approx.4.61 ms	$2^{14} \pm 2^{11}$/HCLK	Approx. 0.457 s	Approx. 0.576 s	$2^{12} \pm 2^{9}$/SCLK
Approx. 14.33 ms	Approx. 18.3 ms	$2^{16} \pm 2^{13}$/HCLK	Approx. 3.584 s	Approx. 4.608 s	$2^{15} \pm 2^{12}$/SCLK
Approx. 57.23 ms	Approx. 73.73 ms	$2^{18} \pm 2^{15}$/HCLK	Approx. 7.168 s	Approx. 9.216 s	$2^{16} \pm 2^{13}$/SCLK
Approx. 458.75 ms	Approx. 589.82 ms	$2^{21} \pm 2^{18}$/HCLK	Approx. 14.336 s	Approx. 18.432 s	$2^{17} \pm 2^{14}$/SCLK

HCLK: Oscillation clock (4 MHz), SLCK: Subclock (8.192 kHz)

Notes:
- If the timebase timer output (carry signal) is used as a count clock to the watchdog timer, the timebase timer is cleared and the time for the watchdog reset to occur may be long.
- If the subclock is used as a machine clock, always set the watchdog timer clock source select bit (WDCS) in the watch timer control register (WTC) to 0 to select the watch timer output.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

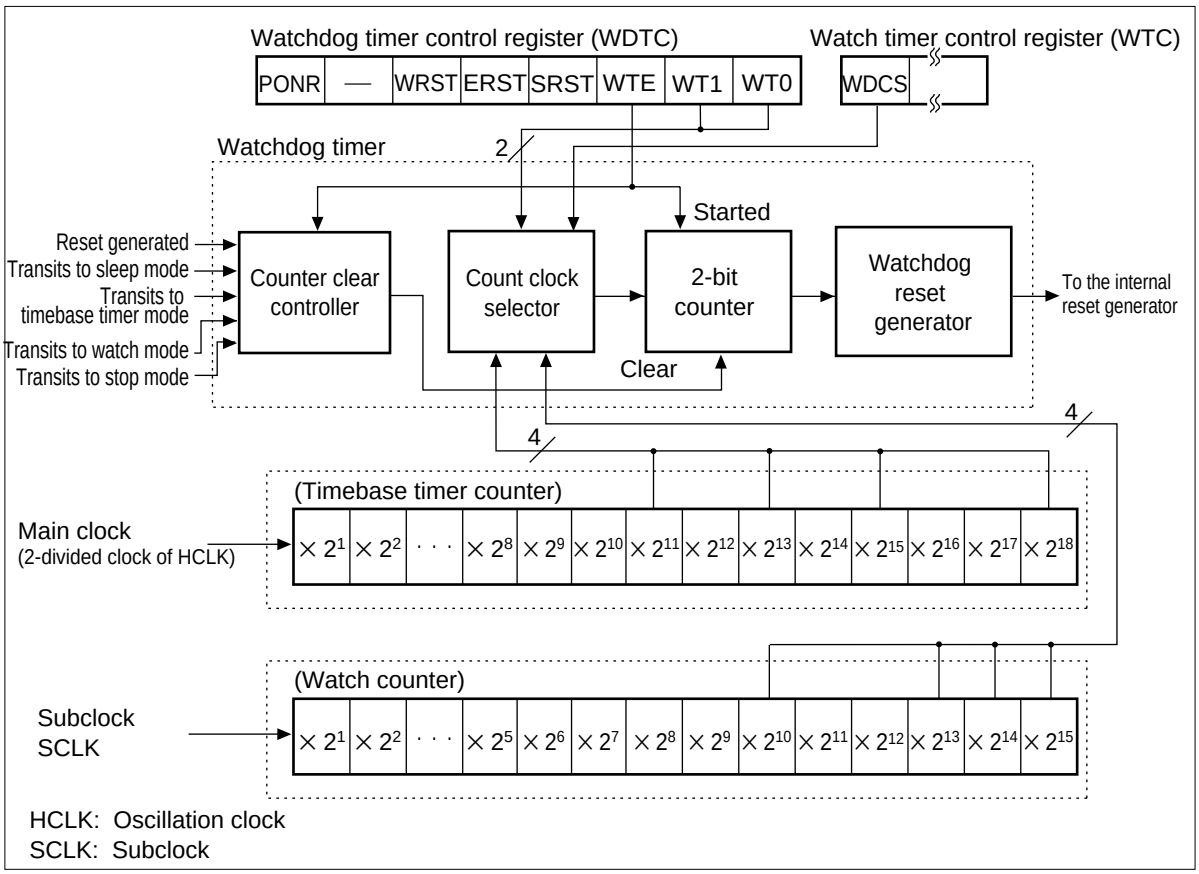
6.2 Configuration of Watchdog Timer

The watchdog timer consists of the following blocks:

- **Count clock selector**
- **Watchdog timer counter (2-bit counter)**
- **Watchdog reset generator**
- **Counter clear controller**
- **Watchdog timer control register (WDTC)**

■ Block Diagram of Watchdog Timer

Figure 6.2-1 Block Diagram of Watchdog Timer



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Count clock selector

  The count clock selector selects the timebase timer output or watch timer output as a count clock input to the watchdog timer. Each timer output has four time intervals that can be set.

- Watchdog timer counter (2-bit counter)

  The watchdog timer counter is a 2-bit up counter that uses the timebase timer output or watch timer output as a count clock. The clock source output destination is set by the watchdog clock select bit in the watch timer control register (WTC: WDCS).

- Watchdog reset generator

  The watchdog reset generator generates a reset signal when the watchdog timer overflows (carrying).

- Counter clear controller

  The counter clear controller clears the watchdog timer counter.

- Watchdog timer control register (WDTC)

  The watchdog timer control register starts and clears the watchdog timer, sets the interval time, and holds reset factors.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

6.3 Watchdog Timer Registers

This section explains the registers used for setting the watchdog timer.

■ List of Registers and Reset Values of Watchdog Timer

Figure 6.3-1 List of Registers and Reset Values of Watchdog Timer

bit	7	6	5	4	3	2	1	0
Watchdog timer control register (WDTC)	X	X	X	X	X	1	1	1

X: Undefined

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

6.3.1 Watchdog Timer Control Register (WDTC)

The watchdog timer control register starts and clears the watchdog timer, sets the interval time, and holds reset factors.

■ Watchdog Timer Control Register (WDTC)

Figure 6.3-2 Watchdog Timer Control Register (WDTC)

7	6	5	4	3	2	1	0
	/						
R	—	R	R	R	W	W	W

Reset value
XXXXX111B

bit 1 bit 0

WT1	WT0	Interval time select bits (Timebase timer output selection)		
		Interval time		Clock cycle
		Minimum	Maximum	
0	0	Approx. 3.58 ms	Approx. 4.61 ms	$2^{14} \pm 2^{11}$/HCLK
0	1	Approx. 14.33 ms	Approx. 18.3 ms	$2^{16} \pm 2^{13}$/HCLK
1	0	Approx. 57.23 ms	Approx. 73.73 ms	$2^{18} \pm 2^{15}$/HCLK
1	1	Approx. 458.75 ms	Approx. 589.82 ms	$2^{21} \pm 2^{18}$/HCLK

HCLK: Oscillation clock
The parenthesized values are interval time when operates at HCLK 4MHz.

bit 1 bit 0

WT1	WT0	Interval time select bits (Watch timer output selection)		
		Interval time		Clock cycle
		Minimum	Maximum	
0	0	Approx. 0.457 s	Approx. 0.576 s	$2^{12} \pm 2^{9}$/SCLK
0	1	Approx. 3.584 s	Approx. 4.608 s	$2^{15} \pm 2^{12}$/SCLK
1	0	Approx. 7.168 s	Approx. 9.216 s	$2^{16} \pm 2^{13}$/SCLK
1	1	Approx. 14.336 s	Approx. 18.432 s	$2^{17} \pm 2^{14}$/SCLK

SCLK: Subclock
The parenthesized values are interval time when operates at SCLK 8.192kHz.

bit 2

WTE	Watchdog timer control bit	
0	First write after reset: starts the watchdog timer	Second or subsequent write after reset: clears of the watchdog timer
1	No effect	

bit 7 bit 5 bit 4 bit 3

Reset factor bits				Reset factor
PONR	WRST	ERST	SRST	
1	×	×	×	Power-on reset
*	1	*	*	Watchdog reset
*	*	1	*	External reset ("L" level input into $\overline{\text{RST}}$ pin)
*	*	*	1	Software reset (Write "1" to RST bit)

R : Read only
W : Write only
* : Holds the previous status
× : Undefined

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 6.3-1 Function of Watching Timer Control Register (WDTC)

Bit name		Function
bit 0, bit 1	WT1, WT0: Interval time select bits	These bits set the interval time of the watchdog timer. The time interval when the watch timer is used as the clock source to the watchdog timer (watchdog clock select bit WDCS = 0) is different from when the main clock mode or the PLL clock mode is selected as the clock mode and the WDCS bit in the watch timer control register (WTC) is set to 1 as shown in Figure 6.3-2 according to the settings of the WTC register. • Only data when the watchdog timer is started is enabled. • Write data after the watchdog timer is started is ignored. • These are write-only bits.
bit 2	WTE: Watchdog timer control bit	This bit starts or clears the watchdog timer. **When set to 0 (first time after reset):** The watchdog timer is started. **When set to 0 (second or subsequent after reset):** The watchdog timer is cleared.
bit6	Unused bits	**Read:** The value is undefined. **Write:** No effect
bit 3 to bit 7	PONR, WRST, ERST, SRST: Reset factor bits	These bits indicate reset factors. • When a reset occurs, the bit corresponding to the reset factor is set to 1. After a reset, the reset factor can be checked by reading the watchdog timer control register (WDTC). • These bits are cleared after the watchdog timer control register (WDTC) is read. **Note:** No bit value other than the PONR bit after power-on reset is assured. If the PONR bit is set at read, other bit values should be ignored.

et4U.com DataShee

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

6.4 Explanation of Operation of Watchdog Timer

After starting, when the watchdog timer reaches the set interval time without the counter being cleared, a watchdog reset occurs.

■ Operation of Watchdog Timer

The operation of the watchdog timer requires the settings shown in Figure 6.4-1.

Figure 6.4-1 Setting of Watchdog Timer

Watchdog timer control register (WDTC)

bit 7	6	5	4	3	2	1	bit 0
PONR	-	WRST	ERST	SRST	WTE	WT1	WT0
					0	◎	◎

Watch timer control register (WTC)

bit 7	6	5	4	3	2	1	bit 0
WDCS	SCE	WTIE	WTOF	WTR	WTC2	WTC1	WTC0
◎							

◎: Used bit
0: Set "0"

DataSheet4U.com

- Selecting clock input source
  - The timebase timer or watch timer can be selected as the clock input source of the count clock to the watchdog timer. When the watchdog clock select bit (WTC: WDCS) is set to 1, the timebase timer is selected. When the bit is set to 0, the watch timer is selected. After a reset, the bit returns to 1.
  - During operation in the subclock mode, set the WDCS bit to 0 to select the watch timer.
- Setting interval time
  - Set the interval time select bits (WDTS: WT1, WT0) to select the interval time for the watchdog timer.
  - Set the interval time concurrently with starting the watchdog timer. Writing to the bit is ignored after the watchdog timer is started.
- Starting watchdog timer
  - When 0 is written to the watchdog timer control bit (WDTC: WTE) after a reset, the watchdog timer is started and starts incrementing.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Clearing watchdog timer

- When 0 is written once again to the watchdog timer control bit (WDTC: WTE) within the interval time after starting the watchdog timer, the watchdog timer is cleared. If the watchdog timer is not cleared within the interval time, it overflows and the CPU is reset.
- A reset, or transitions to the standby modes (sleep mode, stop mode, watch mode, timebase timer mode) clear the watchdog timer.
- During operation in the timebase timer mode or watch mode, the watchdog timer counter is cleared. However, the watchdog timer remains in the activation state.
- Figure 6.4-2 shows the relationship between the clear timing and the interval time of the watchdog timer. The interval time varies with the timing of clearing the watchdog timer.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

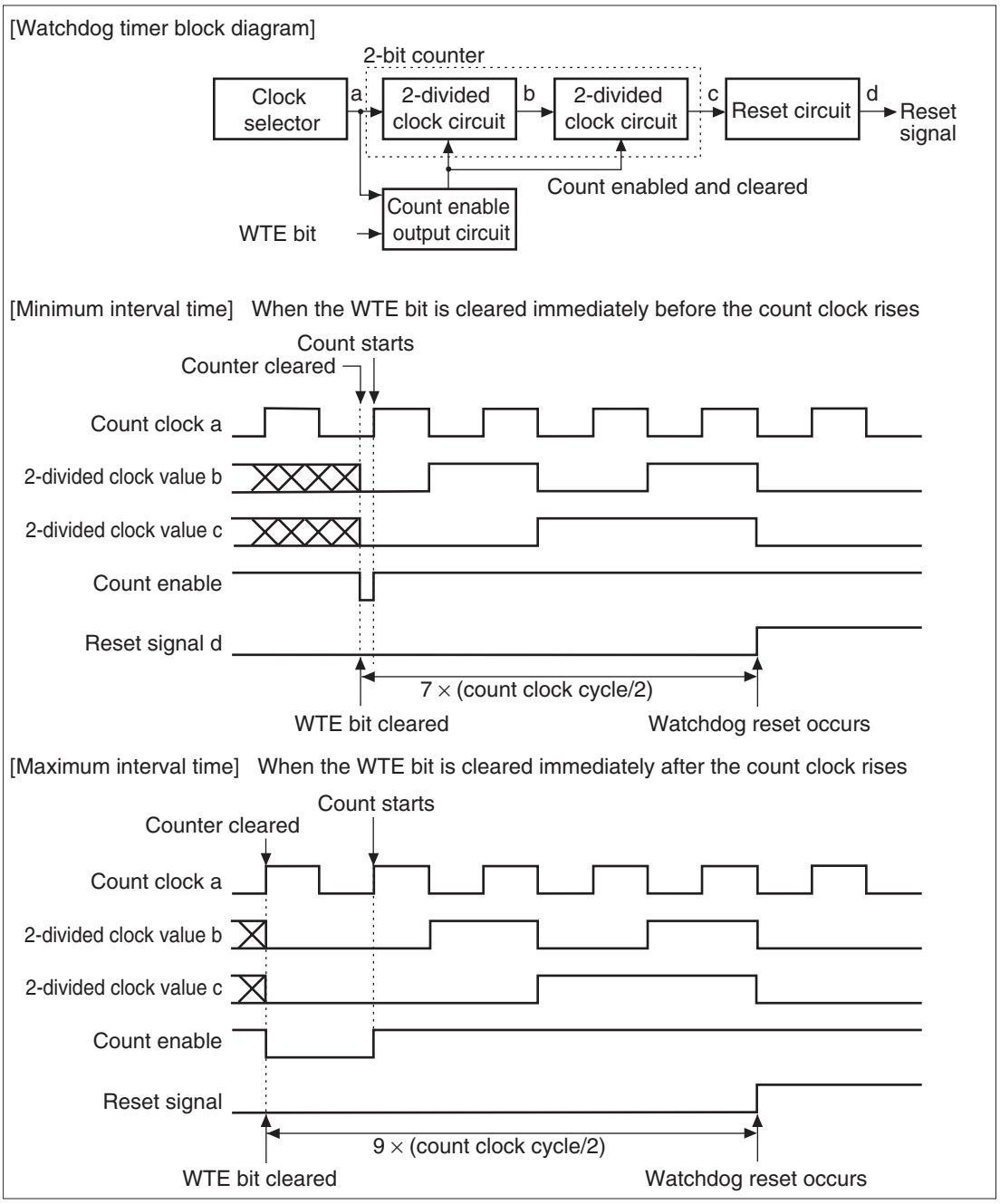
www.DataSheet4U.com

● Checking reset factors

The reset factor bits in the watchdog timer control register (WDTC: PONR, WRST, ERST, SRST) can be read after a reset to check the reset factors.

Reference: For details of reset factor bits, see "3.6 Reset".

Figure 6.4-2 Relationship between Clear Timing and Interval Time of Watchdog Timer



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

6.5 Precautions when Using Watchdog Timer

Take the following precautions when using the watchdog timer.

■ Precautions when Using Watchdog Timer

● Stopping watchdog timer

The watchdog timer is stopped by all the reset sources.

● Interval time

- The interval time uses the carry signal of the timebase timer or watch timer as a count clock. If the timebase timer or watch timer is cleared, the interval time of the watchdog timer may become long.
  The timebase timer is also cleared by writing zero to the timebase timer counter clear bit (TBR) in the timebase timer control register (TBTC); transition from main clock mode to PLL clock mode; transition from subclock mode to main clock mode; and transition from subclock mode to PLL clock mode.
- Set the interval time concurrently with starting the watchdog timer. Setting the interval time except starting the watchdog timer is ignored.

● Precautions when creating program

When clearing the watchdog timer repeatedly in the main loop, set a shorter processing time for the main loop including interrupt processing than the interval time of watchdog timer.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

6.6 Program Examples of Watchdog Timer

Program example of watchdog timer is given below:

■ Program Example of Watchdog Timer

● Processing specification

- The watchdog timer is cleared each time in loop of the main program.
- The main program must be executed once within the minimum interval time of the watchdog timer.

● Coding example

```
WDTC    EQU  0000A8H           ; Watchdog timer control register
WTE     EQU  WDTC:2            ; Watchdog control bit
;
;-----Main program-------------------------------------------------------
CODE        CSEG
START:                         ; Stack pointer (SP), already initialized
        MOV  I:WDTC,#00000011B ; Watchdog timer started
                               ; Interval time of 2^21 + 2^18cycles selected
LOOP:
        CLRB I:WTE             ; Watchdog timer cleared
         :
        Processing by user
         :
        BRA  LOOP
;-----Vector setting------------------------------------------------------
VECT    CSEG ABS=0FFH
        ORG  00FFDCH           ; Reset vector set
        DSL  START
        DB   00H               ; Set to single-chip mode
VECT    ENDS
        END  START
```

et4U.com
DataShee
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 7

16-BIT INPUT/OUTPUT TIMER

This chapter explains the function and operation of 16-bit input/output timer.

7.1 Overview of 16-bit Input/Output Timer

7.2 Block Diagram of 16-bit Input/Output Timer

7.3 Configuration of 16-bit Input/Output Timer

7.4 Interrupts of 16-bit Input/Output Timer

7.5 Explanation of Operation of 16-bit Free-run Timer

7.6 Explanation of Operation of Input Capture

7.7 Precautions when Using 16-bit Input/Output Timer

7.8 Program Example of 16-bit Input/Output Timer

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

7.1 Overview of 16-bit Input/Output Timer

The 16-bit input/output timer is a complex module that consists of a 16-bit free-run timer (x 1 unit) and an input capture (x 2 units/4 input pins). The clock cycle of an input signal and a pulse width can be measured based on the 16-bit free-run timer.

■ Configuration of 16-bit Input/Output Timer

The 16-bit input/output timer consists of the following modules:

- 16-bit free-run timer (× 1 unit)
- Input capture (× 2 units with 2 input pins each)

■ Functions of 16-bit Input/Output Timer

● Functions of 16-bit free-run timer

The 16-bit free-run timer consists of a 16-bit up counter, a timer counter control status register, and a prescaler. The 16-bit up counter increments in synchronization with the division ratio of the machine clock.

- Count clock is selected from eight machine clock division ratios.
  Count clock : ϕ, ϕ/2, ϕ/4, ϕ/8, ϕ/16, ϕ/32, ϕ/64, ϕ/128
- An overflow in the count value generates an interrupt.
- Interrupt generation starts the extended intelligent I/O service (EI^2OS).
- Either a reset or software reset by the timer count clear bit (TCCS: CLR) clears the count value of the 16-bit free-run timer to "0000_H".
- The count value of the 16-bit free-run timer is output to the input capture and can be used as the base time for capture operation.

● Functions of input capture

When the input capture detects the edge of the external signal input to the input pins, it stores the count value of the 16-bit free-run timer in the input capture data registers. The input capture consists of the input capture data registers corresponding to four input pins, an input capture control status register, and an edge detection circuit.

- The detected edge can be selected from among the rising edge, falling edge, and both edges.
- Detecting the edge of the input signal generates an interrupt request to the CPU.
- Interrupt generation starts the EI^2OS.
- Four input pins and four input capture data registers of the input capture can be used to measure up to four events.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

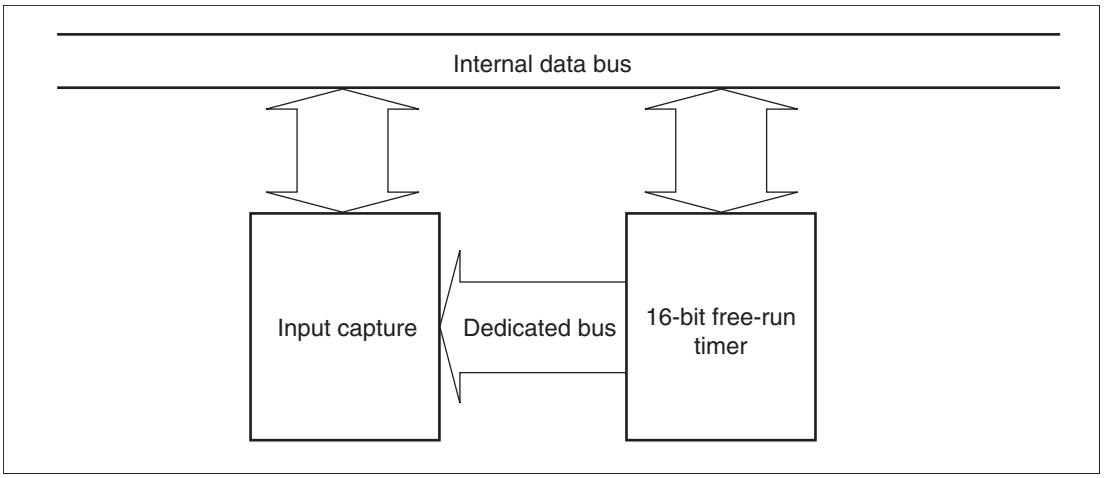
7.2 Block Diagram of 16-bit Input/Output Timer

The 16-bit input/output timer consists of the following modules:

- **16-bit free-run timer**
- **Input capture**

■ Block Diagram of 16-bit Input/Output Timer

Figure 7.2-1 Block Diagram of 16-bit Input/Output Timer



● 16-bit free-run timer

The count value of the 16-bit free-run timer can be use as the base time for the input capture.

● Input capture

The input capture detects the rising edge, falling edge, or both edges of the external signal input to the input pins to retain the count value of the 16-bit free-run timer. Detecting the edge of the input signal generates an interrupt.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

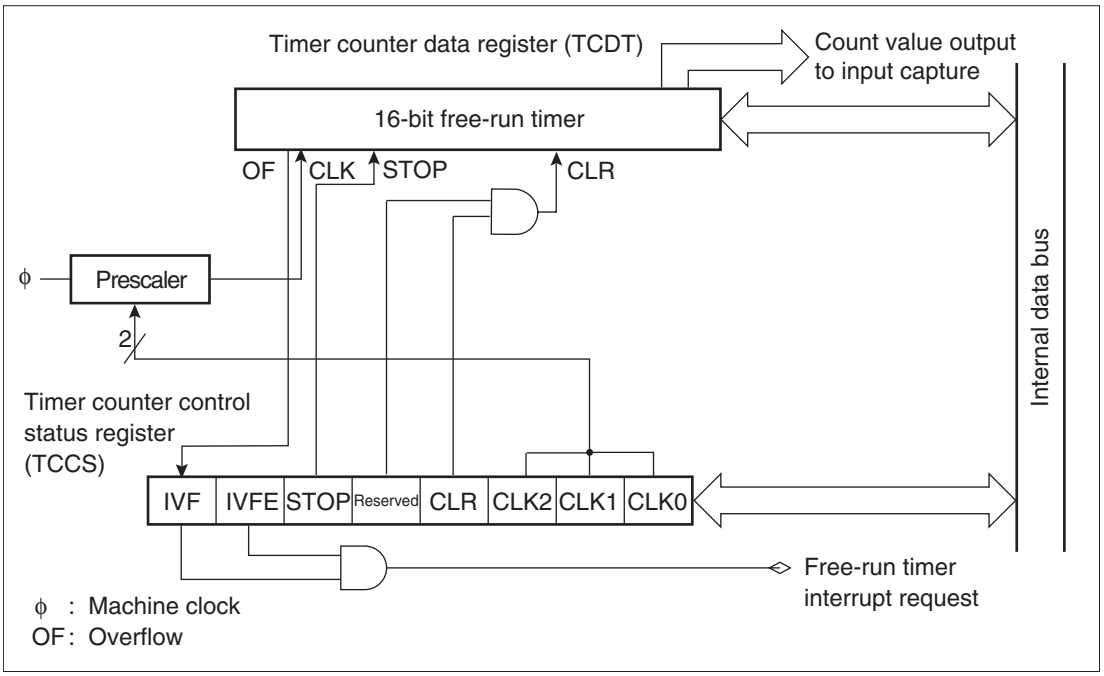
7.2.1 Block Diagram of 16-bit Free-run Timer

The 16-bit free-run timer consists of the following blocks:

- **Prescaler**
- **Timer counter data register (TCDT)**
- **Timer counter control status register (TCCS)**

■ Block Diagram of 16-bit Free-run Timer

Figure 7.2-2 Block Diagram of 16-bit Free-run Timer



■ Details of Pins in Block Diagram

The 16-bit input/output timer has one 16-bit free-run timer.

The interrupt request number of the 16-bit free-run timer is as follows:

Interrupt request number: 19 (13_H)

● Prescaler

The prescaler divides the frequency of machine clock to supply a count clock to the 16-bit up counter. Any of eight machine clock division ratios are selected by setting the timer counter control status register (TCCS).

● Timer counter data register (TCDT)

The timer counter data register (TCDT) is a 16-bit up counter. At read, the current count value of the 16-bit free-run timer can be read. Writing while the counter is stopped enables any count value to be set.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Timer counter control status register (TCCS)

The timer counter control status register (TCCS) selects the division ratio of the machine clock, clears the count value by software, enables or disables the count operation, checks and clears the overflow generation flag, and enables or disables interrupts.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

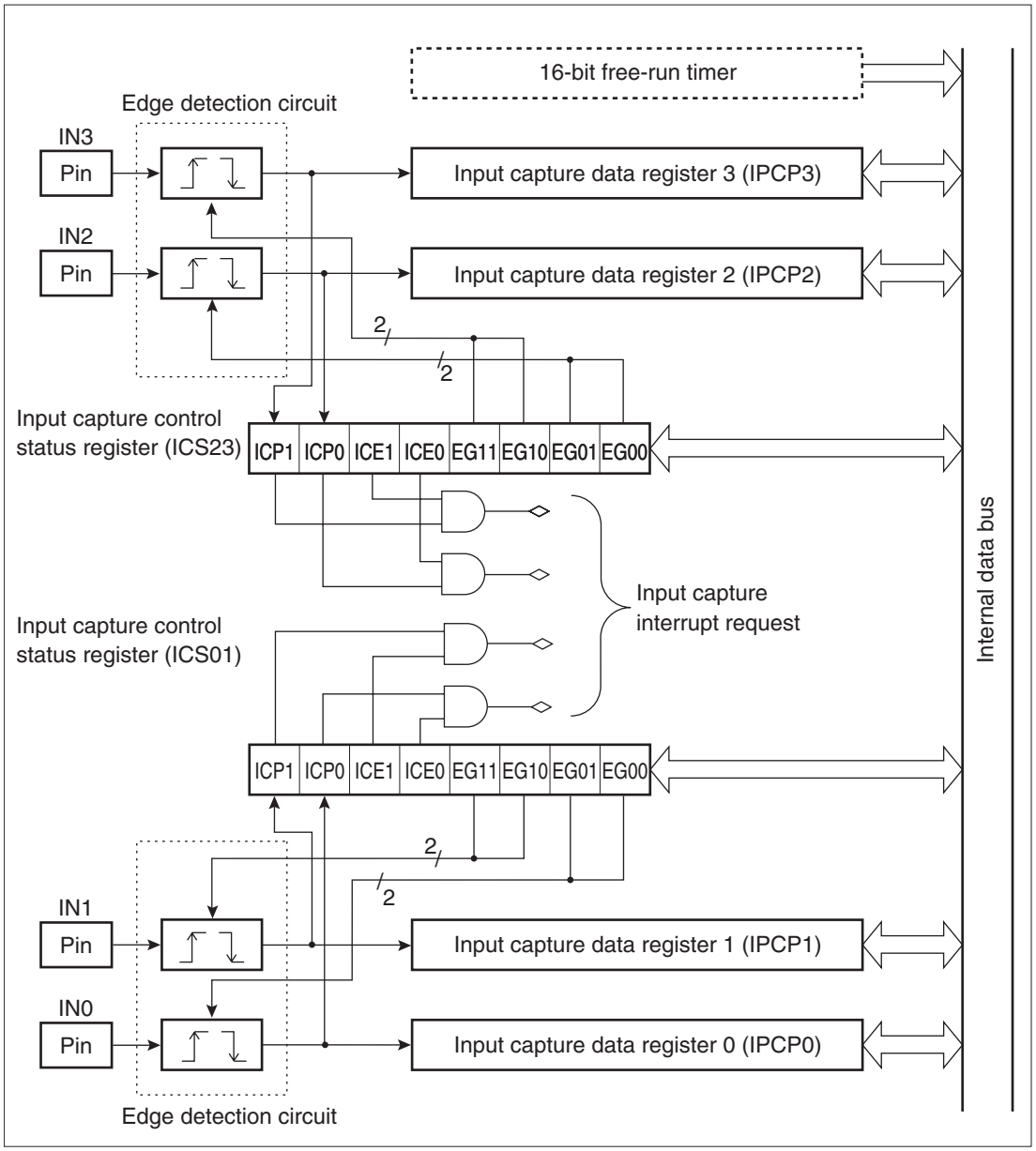
7.2.2 Block Diagram of Input Capture

The input capture consist of the following blocks:

- **Input capture data registers (IPCP0 to IPCP3)**
- **Input capture control status registers (ICS01, ICS23)**
- **Edge detection circuit**

■ Block Diagram of Input Capture

Figure 7.2-3 Block Diagram of Input Capture



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Details of Pins in Block Diagram

The 16-bit input/output timer has four input capture input pins.

The actual pin names and interrupt request numbers of the input capture are shown in Table 7.2-1.

Table 7.2-1 Pins and Interrupt Request Numbers of 16-bit Input/Output Timer

Input Pin	Actual Pin Name	Interrupt Request Number
IN0	P10/IN0	#23 (17_H)
IN1	P11/IN1	#25 (19_H)
IN2	P12/IN2	#30 ($1E_H$)
IN3	P13/IN3	

● Input capture data registers 0 to 3 (IPCP0 to IPCP3)

The counter value of the 16-bit free-run timer actually read when the edge of the external signal input to the input pins (IN0 to IN3) is detected is stored in the input capture data registers (IPCP0 to IPCP3) corresponding to the input pins (IN0 to IN3) to which the signal is input.

● Input capture control status registers (ICS01, ICS23)

The input capture control status registers (ICS01, ICS23) start and stop the capture operation of each input capture, check and clear the valid edge detection flag when the edge is detected, and enable or disable an interrupt. The ICS01 register sets the input capture corresponding to the input pins IN0 and IN1, and the ICS23 register sets the input capture corresponding to the input pins IN2 and IN3.

● Edge detector

The edge detection circuit detects the edge of the external signal input to the input pins. The detected edge can be selected from the rising edge, falling edge, and both edges.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

7.3 Configuration of 16-bit Input/Output Timer

This section explains the pins, registers, and interrupt factors of the 16-bit input/output timer.

■ Pins of 16-bit Input/Output Timer

The pins of the 16-bit input/output timer serve as general-purpose I/O ports. Table 7.3-1 shows the pin functions and the pin settings required to use the 16-bit input/output timer.

Table 7.3-1 Pins of 16-bit Input/Output Timer

Pin Name	Pin Function	Pin Setting Required for Use of 16-bit Input/Output Timer
IN0	General-purpose I/O port, capture input	Set as input port in port direction register (DDR).
IN1	General-purpose I/O port, capture input	Set as input port in port direction register (DDR).
IN2	General-purpose I/O port, capture input	Set as input port in port direction register (DDR).
IN3	General-purpose I/O port, capture input	Set as input port in port direction register (DDR).

et4U.com DataShee

■ Block Diagram of Pins for 16-bit Input/Output Timer

For the block diagram of the pins, see "CHAPTER 4 I/O PORT".

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ List of Registers and Reset Values of 16-bit Input/Output Timer

Figure 7.3-1 List of Registers and Reset Values of 16-bit Input/Output Timer

	bit	7	6	5	4	3	2	1	0
Timer counter control status register (TCCS)		0	0	0	0	0	0	0	0
	bit	15	14	13	12	11	10	9	8
Timer counter data register (High) (TCDT: H)		0	0	0	0	0	0	0	0
	bit	7	6	5	4	3	2	1	0
Timer counter data register (Low) (TCDT: L)		0	0	0	0	0	0	0	0
	bit	7	6	5	4	3	2	1	0
Input capture control status register (ICS01)		0	0	0	0	0	0	0	0
	bit	15	14	13	12	11	10	9	8
Input capture data register 0 (High) (IPCP0: H)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
Input capture data register 0 (Low) (IPCP0: L)		X	X	X	X	X	X	X	X
	bit	15	14	13	12	11	10	9	8
Input capture data register 1 (High) (IPCP1: H)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
Input capture data register 1 (Low) (IPCP1: L)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
Input capture control status register (ICS23)		0	0	0	0	0	0	0	0
	bit	15	14	13	12	11	10	9	8
Input capture data register 2 (High) (IPCP2: H)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
Input capture data register 2 (Low) (IPCP2: L)		X	X	X	X	X	X	X	X
	bit	15	14	13	12	11	10	9	8
Input capture data register 3 (High) (IPCP3: H)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
Input capture data register 3 (Low) (IPCP3: L)		X	X	X	X	X	X	X	X

X: Undefined

DataSheet4U.com
www.DataSheet4U.com

DataSh4U.com

www.DataSheet4U.com

■ Generation of Interrupt Request from 16-bit Input/Output Timer

The 16-bit input/output timer can generate an interrupt request as a result of the following factors:

- Overflow in 16-bit free-run timer

  In the 16-bit input/output timer, when the 16-bit free-run timer overflows, the overflow generation flag bit in the timer counter control status register (TCCS: IVF) is set to 1. When an overflow interrupt is enabled (TCCS: IVFE = 1), an interrupt request is generated.

- Edge detection by capture function

  When the edge of the external signal input to the input pins (IN0 to IN3) is detected, the input capture valid edge detection flag bit in the input capture control status register (ICS: ICP) corresponding to the input pin as the edge is detected is set to 1. When the input capture interrupt corresponding to the channel generating an interrupt request is enabled (ICS: ICE), an interrupt request is generated.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

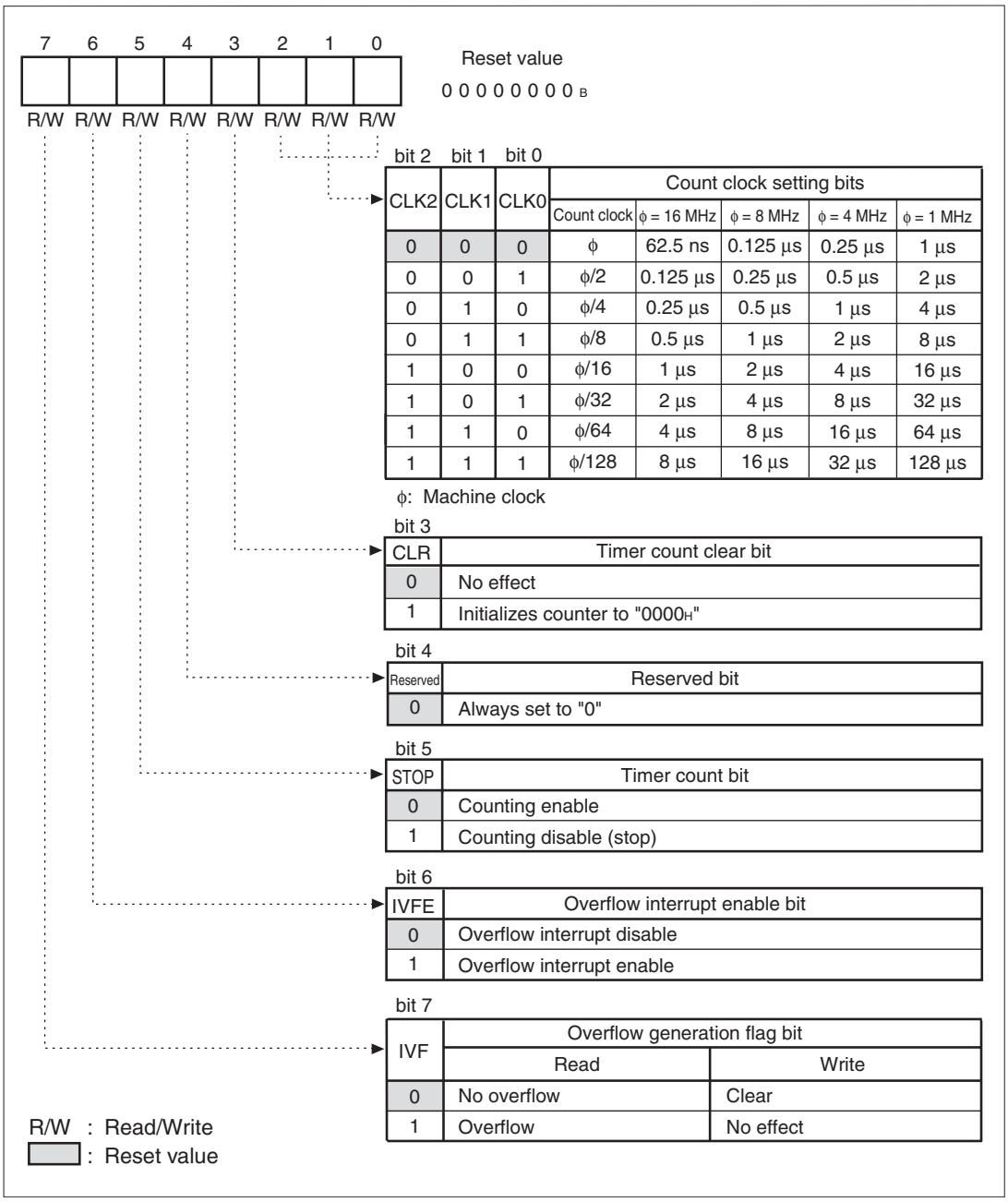
www.DataSheet4U.com

7.3.1 Timer Counter Control Status Register (TCCS)

The timer counter control status register (TCCS) selects the count clock and conditions for clearing the counter, clears the counter, enables the count operation or interrupt, and checks the interrupt request flag.

■ Timer Counter Control Status Register (TCCS)

Figure 7.3-2 Timer Counter Control Status Register (TCCS)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 7.3-2 Functions of Timer Counter Control Status Register (TCCS)

Bit Name		Function
bit 0 bit 1 bit 2	CLK2, CLK1, CLK0: Count clock select bits	These bits set the count clock to the 16-bit free-run time. **Note:** 1. Set the count clock after stopping the count operation (STOP = 1). 2. When rewriting the count clock, write 1 to the timer count clear bit (CLR) and clear the counter value.
bit 3	CLR: Timer count clear bit	This bit clears the counter value of the 16-bit free-run timer. **When set to 1:** Clears timer counter data register (TCDT) to "0000_H" **When set to 0:** No effect **Read:** 0 is always read. • When the counter value changes, the CLR bit is cleared. • When clearing the counter value while stopping the count operation, write "0000_H" to the timer counter data register (TCDT).
bit 4	Reserved: Reserved bit	Always set this bit to 0.
bit 5	STOP: Timer count bit	This bit enables or disables (stops) the count operation of the 16-bit free-run timer. **When set to 0:** Enables count operation The 16-bit timer counter data register (TCDT) starts incrementing in synchronization with the count clock selected by the count clock select bits (CLK1 and CLK0). **When set to 1:** Stops count operation
bit 6	IVFE: Overflow interrupt enable bit	This bit enables or disables an interrupt request generated when the 16-bit free-run timer overflows. **When set to 0:** No interrupt request generated at overflow (IVF = 1) **When set to 1:** Generates interrupt request at overflow (IVF = 1)
bit 7	IVF: Overflow generation flag bit	This bit indicates that the 16-bit free-run timer has overflowed. • If the 16-bit free-run timer overflows or mode setting causes a compare match with the compare register 0 to clear the counter, this bit is set to 1. • When an overflow occurs (IVF=1) with an overflow interrupt enabled (IVFE = 1), an interrupt request is generated. **When set to 0:** Clears bit **When set to 1:** No effect **When EI^2OS started:** Bit cleared **Read by read modify write instructions:** 1 is always read.

DataSheet4U.com

DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

www.DataSheet4U.com

7.3.2 Timer Counter Data Register (TCDT)

The timer counter data register (TCDT) is a 16-bit up counter. At read, the register value being counted is read. At write, while the counter is stopped, any counter value can be set.

■ Timer Counter Data Register (TCDT)

Figure 7.3-3 Timer Counter Data Register (TCDT)

	bit 15	bit 14	bit 13	bit 12	bit 11	bit 10	bit 9	bit 8	Reset value
Timer counter data register (TCDT): High	T15	T14	T13	T12	T11	T10	T9	T8	00000000_B
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

	bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0	Reset value
Timer counter data register (TCDT): Low	SCM	MCM	WS1	WS0	SCS	MCS	CS1	CS0	00000000_B
	R	R	R/W	R/W	R/W	R/W	R/W	R/W	

R/W: Read/ Write

■ Count Operation of Timer Counter Data Register (TCDT)

- When the timer counter data register (TCDT) is read during the count operation, the counter value of the 16-bit free-run timer is read.
- When the counter value of the timer counter data register (TCDT) increments from "$FFFF_H$" to "0000_H", an overflow occurs and the overflow generation flag bit (TCCS: IVF) is also set to 1.
- When an overflow occurs (TCCS: IVF = 1) with an overflow interrupt enabled (TCCS: IVFE = 1), an overflow interrupt request is generated.
- The counter value of the timer counter data register (TCDT) is retained while the count operation is stopped.
- When stopping the count operation of the timer counter data register (TCDT), write 1 to the timer count operation bit (TCCS: STOP).
- When the count operation stops (TCCS: STOP = 1), the counter value of the timer counter data register (TCDT) can be set to any value.

● Factors clearing timer counter data register

The timer counter data register (TCDT) is cleared to "0000_H" by the following factors:

of the following events, the overflow clears the register in synchronization with the count clock and each of the other events clears the register on occurrence of that event.

- Reset
- Writing 1 to the timer count clear bit (TCCS: CLR) (possible even during count operation)
- Writing "0000_H" to timer counter data register (TCDT) while count operation stopped
- Overflow in 16-bit free-run timer

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Note:	Always use a word instruction (MOVW) to set the timer counter data register (TCDT).

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

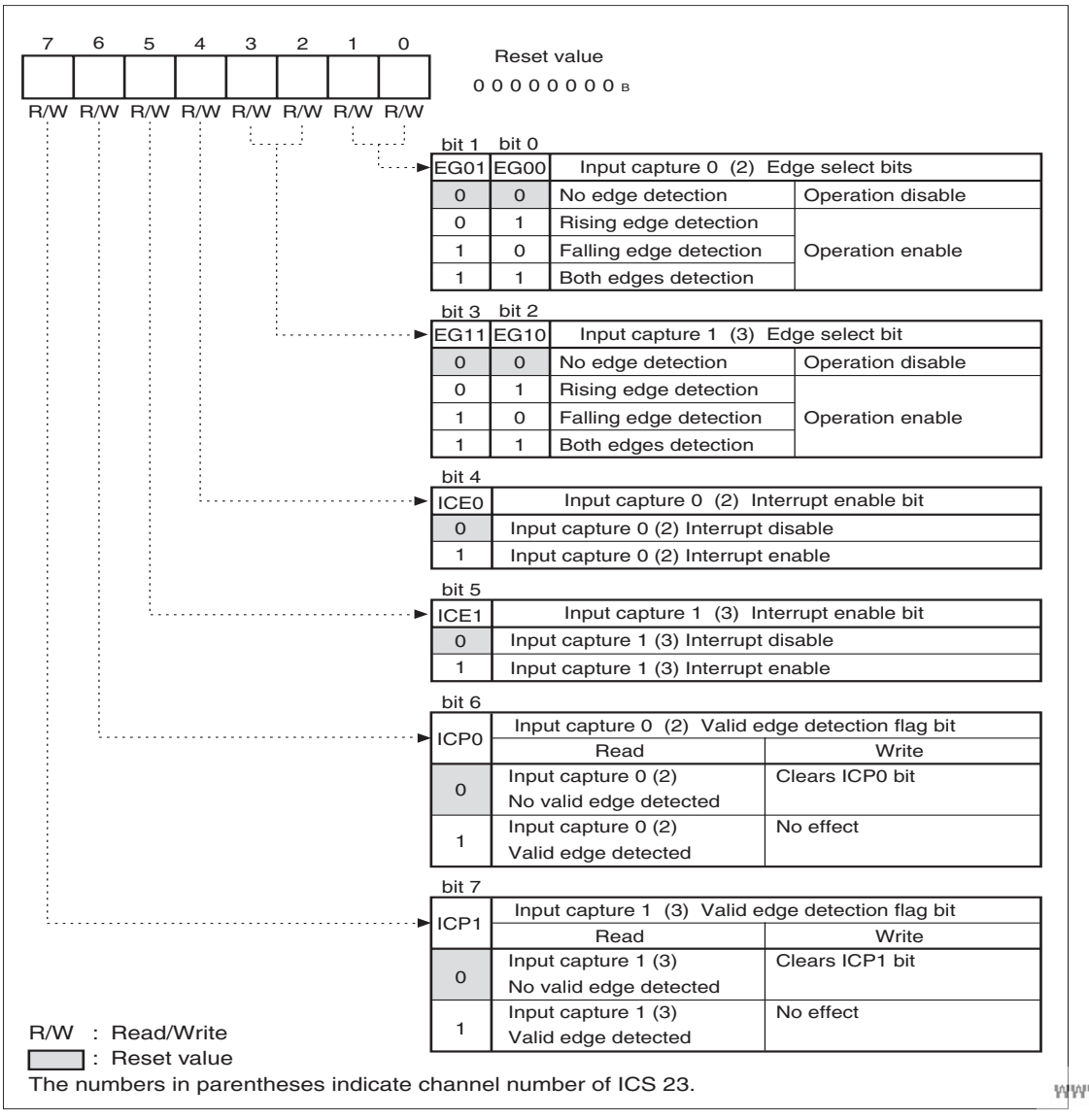
7.3.3 Input Capture Control Status Registers (ICS01 and ICS23)

The input capture control status registers sets the operation of input captures. The ICS01 register sets the operation of input captures 0 and 1 and the ICS23 sets the operation of input captures 2 and 3.
The input capture control status registers provides the following settings:

- **Selecting the edge to be detected**
- **Enabling or disabling an interrupt when the edge is detected**
- **Checking and clearing the valid edge detection flag when the edge is detected**

■ Input Capture Control Status Registers (ICS01 and ICS23)

Figure 7.3-4 Input Capture Control Status Registers (ICS01 and ICS23)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 7.3-3 Functions of Input Capture Control Status Register (ICS01) (1/2)

Bit Name		Function
bit0 bit1	EG01, CEG00: Input capture 0 edge select bits	These bits enable or disable the operation of input capture 0. The edge detected by input capture 0 is selected when the operation of input capture 0 is enabled. **EG01, EG00 = "00_B":** The operation of input capture 0 is disabled and no edge is detected. **EG01, EG00 "00_B":** The operation of input capture 0 is enabled and the edge is detected.
bit 2 bit 3	EG11, EG10: Input capture 1 edge select bits	These bits enable or disable the operation of input capture 1. The edge detected by input capture 1 is selected when the operation of input capture 1 is enabled. **EG01, EG00 = "00_B":** The operation of input capture 1 is disabled and no edge is detected. **EG01, EG00 "00_B":** The operation of input capture 1 is enabled and the edge is detected.
bit 4	ICE0: Input capture 0 interrupt enable bit	This bit enables or disables an interrupt when the edge is detected by input capture 0. **When set to 0:** No interrupt is generated even when the valid edge is detected by input capture 0. **When set to 1:** An interrupt is generated when the valid edge is detected by input capture 0.
bit 5	ICE1: Input capture 1 interrupt enable bit	This bit enables or disables an interrupt when the edge is detected by input capture 1. **When set to 0:** No interrupt is generated even when the edge is detected by input capture 1. **When set to 1:** An interrupt is generated when the edge is detected by input capture 1.
bit 6	ICP0: Input capture 0 valid edge detection flag bit	This bit indicates the edge detection by input capture 0. • When the valid edge selected by the input capture 0 edge select bits (EG01, EG00) is detected, the ICP0 bit is set to 1. • When the valid edge is detected by input capture 0 (ICP0 = 1) when an interrupt due to the edge detection by input capture 0 is enabled (ICE0 = 1), an interrupt is generated. **When set to 0:** The bit is cleared. **When set to 1:** No effect **When EI^2OS started:** The bit is cleared. **Read by read modify write instructions:** 1 is always read.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 7.3-3 Functions of Input Capture Control Status Register (ICS01) (2/2)

Bit Name		Function
bit 7	ICP1: Input capture 1 valid edge detection flag bit	This bit indicates the edge detection by input capture 1. • When the valid edge selected by the input capture 1 edge select bits (EG11, EG10) is detected, the ICP1 bit is set to 1. • When the valid edge is detected by input capture 1 (ICP1 = 1) when an interrupt due to the edge detection by input capture 1 is enabled (ICE1 = 1), an interrupt is generated. **When set to 0:** The bit is cleared. **When set to 1:** No effect **When EI^2OS started:** The bit is cleared. **Read by read modify write instructions:** 1 is always read.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

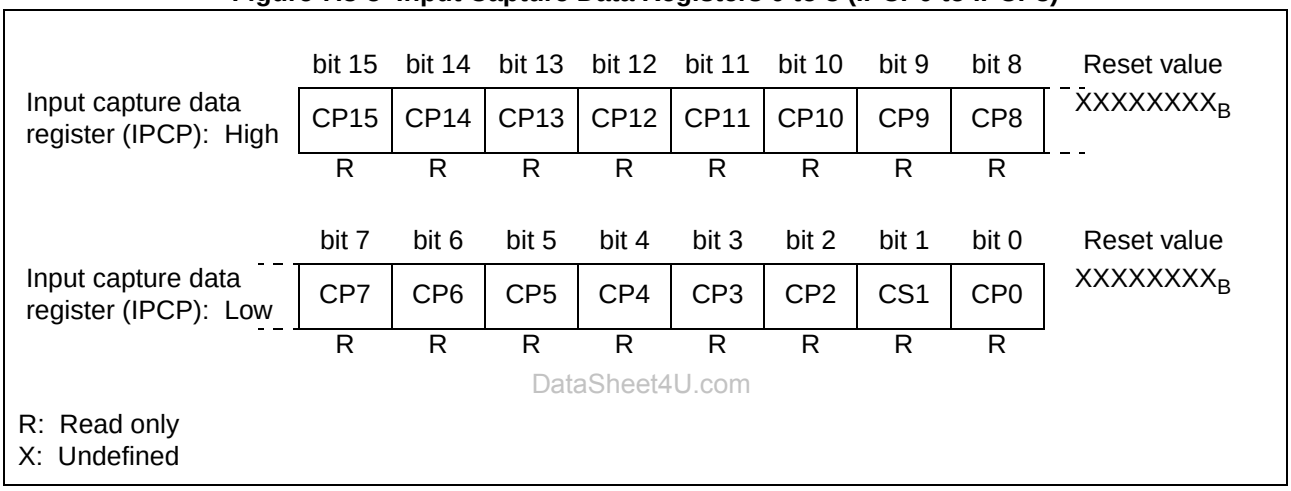
www.DataSheet4U.com

7.3.4 Input Capture Data Registers 0 to 3 (IPCP0 to IPCP3)

The input capture data registers 0 to 3 (IPCP0 to IPCP3) store the counter value of the 16-bit free-run timer read in the timing with the edge detection by the input capture. The counter value of the 16-bit free-run timer is stored in the input capture data registers (IPCP0 to IPCP3) corresponding to the input pins (IN0 to IN3) to which an external signal is input.

■ Input Capture Data Registers 0 to 3 (IPCP0 to IPCP3)

Figure 7.3-5 Input Capture Data Registers 0 to 3 (IPCP0 to IPCP3)



DataSheet4U.com

■ Operation of Input Capture Data Registers 0 to 3 (IPCP0 to IPCP3)

- At the same time that the edges of signals input from the input pins (IN0 to IN3) of the 16-bit input/output timer are detected, the counter value of the 16-bit free-run timer is stored in the input capture data registers 0 to 3 (IPCP0 to IPCP3) corresponding to the input pins (IN0 to IN3).

Note: Always use a word instruction (MOVW) to read the input capture data registers 0 to 3 (IPCP0 to IPCP3).

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

7.4 Interrupts of 16-bit Input/Output Timer

The interrupt factors of the 16-bit input/output timer include an overflow in the 16-bit free-run timer and edge detection by the input capture. Interrupt generation starts EI^{2}OS.

■ Interrupt Control Bits and Interrupt Factors of 16-bit Input/Output Timer

Table 7.4-1 shows the interrupt control bits and interrupt factors of the 16-bit input/output timer.

Table 7.4-1 Interrupt Control Bits and Interrupt Factors of 16-bit Input/Output Timer

Interrupt Name	Overflow Interrupt	Input Capture Interrupt			
Interrupt factor	Overflow in counter value of 16-bit free-run timer	Valid edge input to input pins (IN0 to IN3) of input capture			
		IN0	IN1	IN2	IN3
Interrupt request flag bit	TCCS: IVF	ICS01: ICP0	ICS01: ICP1	ICS23: ICP0	ICS23: ICP1
Interrupt enable bit	TCCS: IVFE	ICS01: ICE0	ICS01: ICE1	ICS23: ICE0	ICS23: ICF

et4U.com DataShee

● 16-bit free-run timer interrupt

- When the counter value of the timer counter data register (TCDT) increments from "FFFF$_H$" to "0000$_H$", an overflow occurs and the overflow generation flag bit (TCCS: IVF) is set simultaneously to 1.
- When an overflow occurs (TCCS: IVF = 1) with an overflow interrupt enabled (TCCS: IVFE = 1), an overflow interrupt is generated.

DataSheet4U.com

● Input capture interrupt

- When the valid edge selected by the input capture edge select bit (ICS: EG) is detected, the input capture interrupt request flag bits (ICS01, ICS23: ICP1, ICP0) corresponding to the input pins (IN0 to IN3) are set to 1.
- When the valid edge is detected by the input captures corresponding to the input pins (IN0 to IN3) with the input capture interrupts corresponding to the input pins (IN0 to IN3) enabled, an input capture interrupt is generated.

■ Correspondence between 16-bit Input/Output Timer Interrupt and EI^{2}OS

For details of the interrupt number, interrupt control register, and interrupt vector address, see "3.5 Interrupt".

■ 16-bit Input/Output Timer Interrupts and EI^{2}OS Function

The 16-bit input/output timer corresponds to the EI^{2}OS function. The generation of enabled interrupt starts the EI^{2}OS. However, it is necessary to disable generation of interrupt requests by resources sharing the interrupt control register (ICR) with the 16-bit input/output timer.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

7.5 Explanation of Operation of 16-bit Free-run Timer

After a reset, the 16-bit free-run timer starts incrementing from "0000_H". When the counter value is incremented from "$FFFF_H$" to "0000_H", an overflow occurs.

■ Setting of 16-bit Free-run Timer

Operation of the 16-bit free-run timer requires the setting shown in Figure 7.5-1.

Figure 7.5-1 Setting of 16-bit Free-run Timer

	bit15	14	13	12	11	10	9	bit8	bit7	6	5	4	3	2	1	bit0
TCCS									IVF	IVFE	STOP	Reserved	CLR	CLK2	CLK1	CLK0
									0	◎	0	0	◎	◎	◎	◎
TCDT	Counter value of 16-bit free-run timer															

◎: Used bit
0: Set 0
Reserved: Always set to "0"

et4U.com DataShee

■ Operation of 16-bit Free-run Timer

DataSheet4U.com

- After a reset, the 16-bit free-run timer starts incrementing from "0000_H" in synchronization with the count clock selected by the count clock select bits (TCCS: CLK2, CLK1, CLK0).
- When the counter value of the timer counter data register (TCDT) is incremented from "$FFFF_H$" to "0000_H", an overflow occurs. When an overflow occurs, the overflow generation flag bit (TCCS: IVF) is set to 1 and the 16-bit free-run timer starts incrementing again from "0000_H".
- When an overflow occurs (TCCS: IVF = 1) with an overflow interrupt enabled (TCCS: IVFE = 1), an overflow interrupt is generated.
- When stopping the count operation of the timer counter data register (TCDT), write 1 to the timer count bit (TCCS: STOP).
- Set the counter value in the timer counter data register (TCDT) after stopping the count operation of the 16-bit free-run timer. After completing setting of the counter value, enable the count operation of the 16-bit free-run timer (TCCS: STOP = 0).

DataSheet4U.com www.DataSheet4U.com

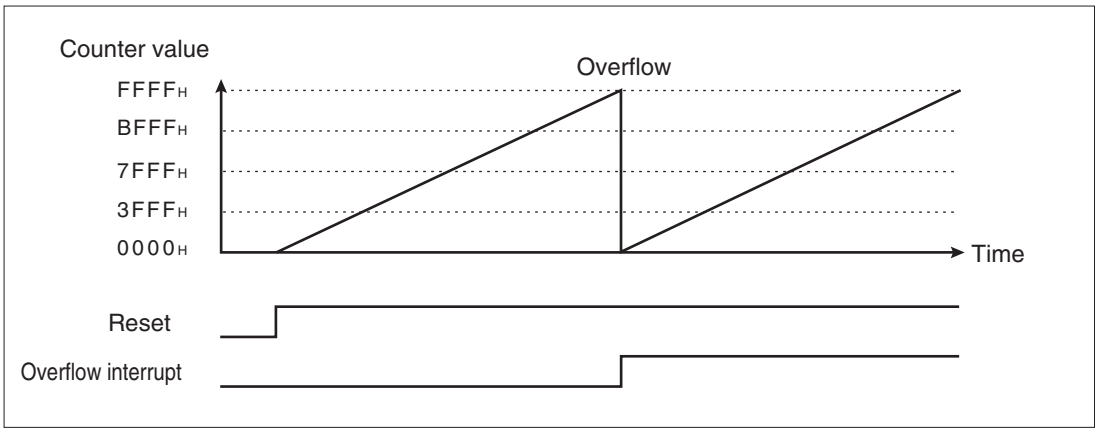
DataSheet4U.com

www.DataSheet4U.com

■ Operation Timing of 16-bit Free-run Timer

Figure 7.5-2 shows counter clearing at an overflow.

Figure 7.5-2 Counter Clearing at an Overflow



et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

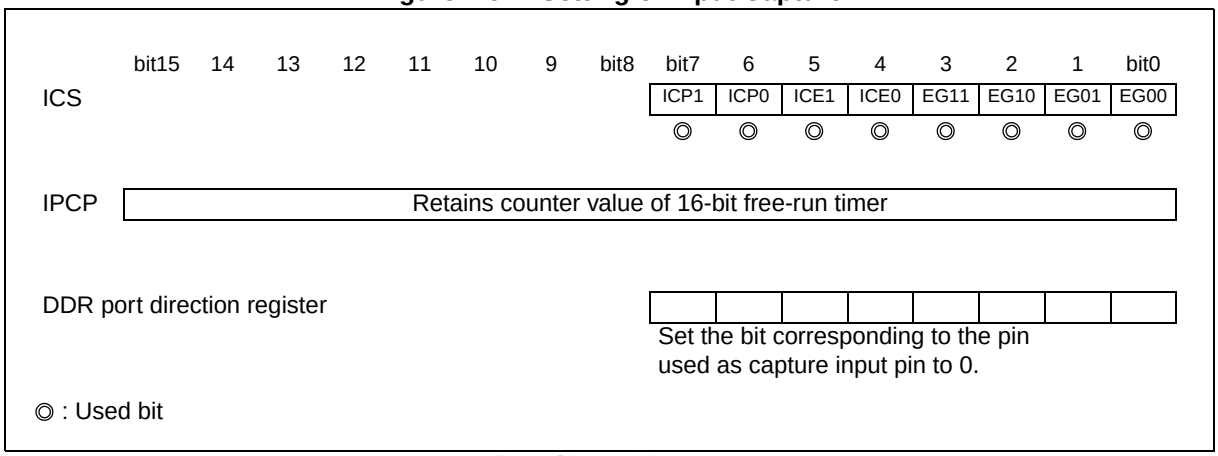
7.6 Explanation of Operation of Input Capture

When the input capture detects the edge of the external signal input to the input pin, it stores the counter value of the 16-bit free-run timer in the input capture data register.

■ Setting of Input Capture

Operation of the input capture requires the setting shown in Figure 7.6-1.

Figure 7.6-1 Setting of Input Capture



DataSheet4U.com

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Operation of Input Capture

- When the valid edges of the external signals input to the input pins (IN0 to IN3) are detected, the input capture valid edge detection flag bit (ICS: ICP) corresponding to the input pin is set to 1. At the same time, the counter value of the 16-bit free-run timer is stored in the input capture data registers (IPCP) corresponding to the input pins (IN0 to IN3).
- The edge to be detected can be selected from the rising edge, falling edge and both edges by setting the input capture edge select bit in the input capture control status register (ICS: EG).
- When the effective edge is detected by the input captures corresponding to the input pins (IN0 to IN3) when the input captures corresponding to the input pins (IN0 to IN3) are enabled for interrupts, an input capture interrupt is generated.
- The input capture valid edge detection flag bit (ICS: ICP) is set when the valid edge is detected, regardless of the interrupt enable settings (ICS01, ICS23: ICE1, ICE0).
- Table 7.6-1 shows the correspondence between the input pins and input captures.

Table 7.6-1 Correspondence between Input Pins and Input Captures

Input Pin	Interrupt Request Flag Bit of Input Capture	Interrupt Output Enable Bit of Input Capture	Input Capture Data Register
IN0	ICS01: ICP0	ICS01: ICE0	IPCP0
IN1	ICS01: ICP1	ICS01: ICE1	IPCP1
IN2	ICS23: ICP0	ICS23: ICE0	IPCP2
IN3	ICS23: ICP1	ICS23: ICE1	IPCP3

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Operation Timing of Input Capture

Figure 7.6-2 shows the timing of reading the counter value of the 16-bit free-run timer.

Figure 7.6-2 Timing of Reading Counter Value of Input Capture

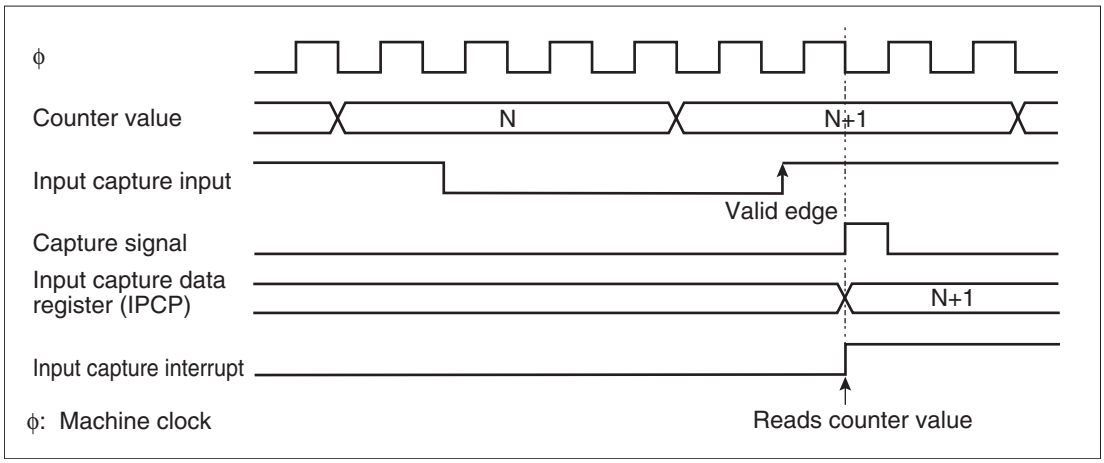
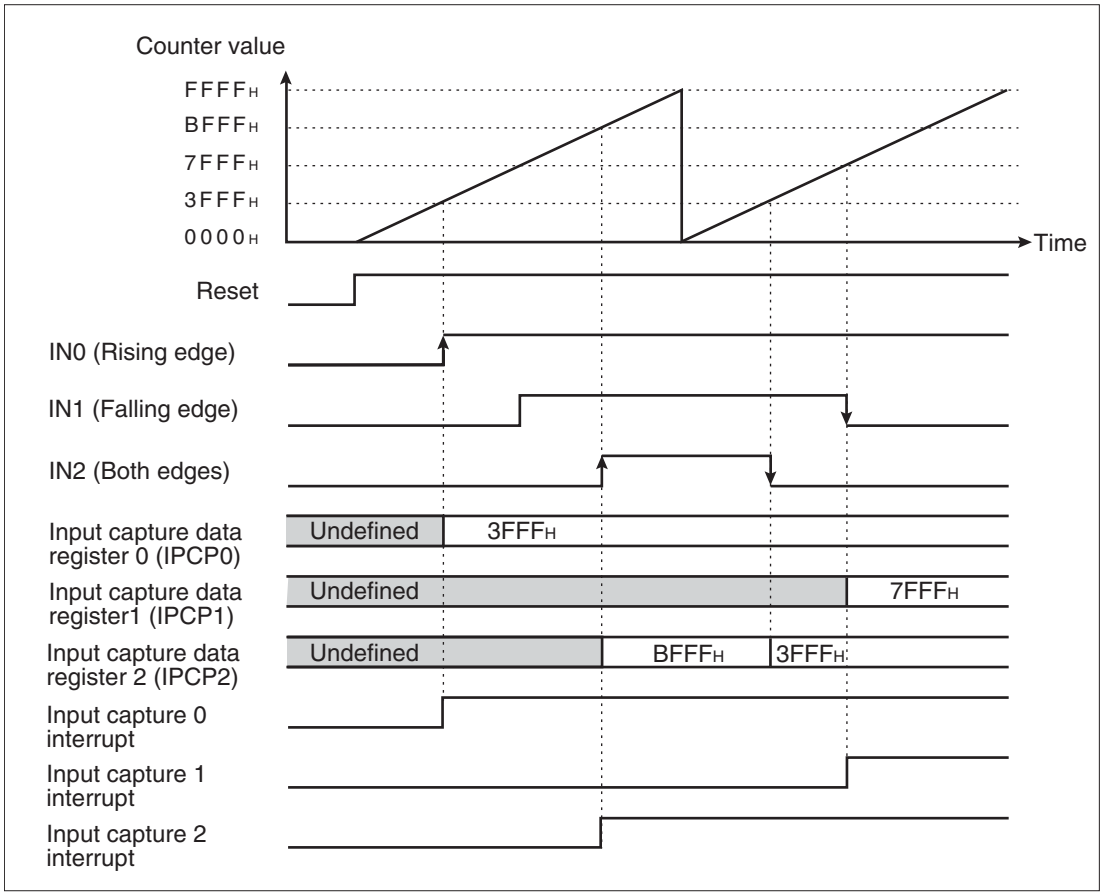


Figure 7.6-3 shows the timing of the capture operation depending on the edge type.

Figure 7.6-3 Timing of Capture Operation Depending on Edge Type



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

7.7 Precautions when Using 16-bit Input/Output Timer

This section explains the precautions when using the 16-bit input/output timer.

■ Precautions when 16-bit Input/Output Timer

● Precautions when setting 16-bit free-run timer

- Do not change the count clock select bits (TCCS: CLK2, CLK1, CLK0) during the count operation (TCCS: STOP = 0).
- The counter value of the 16-bit free-run timer is cleared to "0000_H" by reset. The 16-bit free-run timer can be set by writing any count value to the timer counter data register (TCDT) while the count operation is stopped (TCCS: STOP = 1).
- Always use a word instruction (MOVW) to set the timer counter data register (TCDT).

● Precautions on interrupts

- When an overflow interrupt or an input capture interrupt is enabled, clear only the set bit of the overflow generation flag bit or the input capture valid edge detection flag bit. For example, when clearing the flag bit for the factor that accepted an interrupt, avoid unconditional clearing of the interrupt request flag bits other than those for the factor accepting the interrupt, otherwise another input capture interrupt may be generated.
- If the interrupt request flag bits in the 16-bit input/output timer (TCCS: IVF, ICS01, ICS23: ICP1, ICP0) are set to 1 and interrupts corresponding to the set interrupt request flag bits are enabled (TCCS: IVFE = 1, ICS01, ICS23: ICE1 = 1, ICE0 = 1), it is impossible to return from interrupt processing. Always clear the interrupt request flag bits. When using the EI^2OS, the set interrupt request flag bits are cleared automatically when the EI^2OS is started.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

7.8 Program Example of 16-bit Input/Output Timer

This section gives a program example of the 16-bit input/output timer.

■ Processing of Program for Measuring Cycle Using Input Capture

- The cycle of a signal input to the IN0 pin is measured.
- The 16-bit free-run timer and input capture 0 are used.
- The rising edge is selected as the edge to be detected.
- The machine clock (ϕ) is 16 MHz and the count clock is ϕ/4 (0.25 µs).
- The overflow interrupt and input capture interrupt of input capture 0 are used.
- The overflow interrupt of the 16-bit free-run timer is counted beforehand and used for the cycle calculation.
- The cycle can be determined from the following equation:

Cycle = (overflow count × 10000_H+ nth IPCP0 value - (n-1)th IPCP0 value) × count clock cycle

= (overflow count × 10000_H+ nth IPCP0 value - (n-1)th IPCP0 value) × 0.25µs

● Coding example

```
DDR1    EQU  000011H            ; Port direction register
TCCS    EQU  000058H            ; Timer counter control status register
TCDT    EQU  000056H            ; Timer counter data register
ICS01   EQU  000054H            ; Input capture control status register 01
IPCP0   EQU  000050H            ; Input capture data register 0
IVFE    EQU  TCCS:5             ; Overflow interrupt enable bit
ICP0    EQU  ICS01:6            ; Input capture 0 interrupt request flag bit
ICR04   EQU  0000B4H            ; 16-bit free-run timer interrupt control register
ICR06   EQU  0000B6H            ; 16-bit input capture interrupt control  register
DATA    DSEG ABS=00H
        ORG  0100H
OV_CNT  RW   1                  ; Overflow counter
DATA    ENDS
;-----Main program-------------------------------------------------------------
CODE    CSEG ABS=0FFH
START:
                                ; Stack pointer (SP)
                                ; already initialized
         :
        AND  CCR,#0BFH          ; Interrupt disabled
        MOV  I:ICR04,#00H       ; Interrupt level 0 (highest)
        MOV  I:ICR06,#00H       ; Interrupt level 0 (highest)
        MOV  I:DDR1,#00000000B  ; Pin set as input
```

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com


```
        MOV  I:TCCS,#00110100B  ; Count operation enabled, counter cleared,
                                ; Overflow, interrupt enabled
                                ; Count clock of ϕ/4 selected
        MOV  I:ICS01,#00010001B ; INO pin selected
                                ; IPCP0 set to rising edge
                                ; IPCP1 set to no edge detection
                                ; Each interrupt request flag cleared
                                ; Input capture interrupt request enabled
        MOV  ILM,#07            ; Interrupt mask level set and interrupt enabled
        OR   CCR,#40H           ; Interrupt enabled
         :
;-----Interrupt program-----------------------------------------------------
WARI1   CLRB I:ICP0             ; Input capture 0 interrupt request
                                ; flag cleared
         :
        User Processing (such as cycle calculation)
         :
        MOV  A,0                ; Overflow because of next cycle measurement
                                ; Counter cleared
        MOV   D:OV_CNT,A
        RETI                    ; Return from interrupt
WARI2   CLRB I:IVFE             ; Overflow interrupt request flag cleared
        INC  D:OV_CNT           ; Overflow counter incremented by one
        RETI
CODE    ENDS
;-----Vector setting--------------------------------------------------------
VECT    CSEG ABS=0FFH
        ORG  0FFA0              ; Vector set to interrupt number #23 (17H)
        DSL  WARI1              ; Input capture 0 interrupt
        ORG  0FFB0              ; Vector set to interrupt number #19 (13H)
        DSL  WARI2              ; Overflow interrupt
        ORG  0FFDCH             ; Reset vector set
        DSL  START
        DB   00H                ; Set to single-chip mode
VECT    ENDS
        END  START
```

et4U.com
DataShee
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 8

16-BIT RELOAD TIMER

This chapter explains the functions and the operations of 16-bit reload timer.

8.1 Overview of 16-bit Reload Timer
8.2 Block Diagram of 16-bit Reload Timer
8.3 Configuration of 16-bit Reload Timer
8.4 Interrupts of 16-bit Reload Timer
8.5 Explanation of Operation of 16-bit Reload Timer
8.6 Precautions when Using 16-bit Reload Timer
8.7 Program Example of 16-bit Reload Timer

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

8.1 Overview of 16-bit Reload Timer

The 16-bit reload timer has the following functions:

- **The count clock can be selected from three internal clocks and external event clocks.**
- **A software trigger or external trigger can be selected as the start trigger.**
- **If the 16-bit timer register (TMR) underflows, an interrupt can be generated to the CPU. The 16-bit reload timer can be used as an interval timer by using an interrupt.**
- **If the TMR underflows, either the one-shot mode for stopping the TMR count operation, or the reload mode for reloading the value of the 16-bit reload register (TMRLR) to the TMR to continue the TMR count operation can be selected.**
- **The 16-bit reload timer corresponds to the EI^{2}OS.**
- **The MB90385 series has two channels of 16-bit reload timers.**

■ Operation Modes of 16-bit Reload Timer

Table 8.1-1 indicates the operation modes of the 16-bit reload timer.

Table 8.1-1 Operation Modes of 16-bit Reload Timer

Count Clock	Start Trigger	Operation Performed upon Underflow
Internal clock mode	Software trigger External trigger	One-shot mode Reload mode
Event count mode	Software trigger	One-shot mode Reload mode

■ Internal Clock Mode

- When the count clock select bits in the timer control status register (TMCSR: CSL1, CSL0) are set to "00_B", "01_B" or "10_B", the 16-bit reload timer is set in the internal clock mode.
- In the internal clock mode, the 16-bit reload timer decrements in synchronization with the internal clock.
- The count clock select bits in the timer control status register (TMCSR: CSL1, CSL0) can be used to select three count clock cycles.
- The start trigger sets the edge detection for a software trigger or an external trigger.

■ Event Count Mode

- When the count clock select bits in the timer control status register (TMCSR: CSL1, CSL0) are set to "11_B", the 16-bit reload timer is set to the event count mode.
- In the event count mode, the 16-bit reload timer decrements in synchronization with the edge detection of the external event clock input to the TIN pin.
- A software trigger is selected as the start trigger.
- The 16-bit reload timer can be used as an interval timer by using a fixed cycle of the external clock.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Operation at Underflow

When the start trigger is input, the value set in the 16-bit reload register (TMRLR) is reloaded to the 16-bit timer register, starts decrementing in synchronization with the count clock. When the 16-bit timer register (TMR) is decremented from "0000_H" to "$FFFF_H$", an underflow occurs.

- When an underflow occurs with an underflow interrupt enabled (TMCSR: INTE = 1), an underflow interrupt is generated.
- The 16-bit reload timer operation when an underflow occurs is set by the reload select bit in the timer control status register (TMCSR: RELD).

[One-shot mode (TMCSR: RELD = 0)]

When an underflow occurs, the TMR count operation is stopped. When the next start trigger is input, the value set in the TMRLR is reloaded in the TMR, starting the TMR count operation.

- In the one-shot mode, during the TMR count operation, a High-level or Low-level rectangular wave is output from the TOT pin.
- The pin output level select bit in the timer control status register (TMCSR: OUTL) can be set to select the level (High or Low) of the rectangular wave.

[Reload mode (TMCSR: RELD = 1)]

When an underflow occurs, the value set in the TMRLR is reloaded to the TMR, continuing the TMR count operation.

- In the reload mode, a toggle wave inverting the output level of the TOT pin is output each time an underflow occurs during the TMR count operation.
- The pin output level select bit in the timer control status register (TMCSR: OUTL) can be set to select the level (High or Low) of a toggle wave.
- The 16-bit reload timer can be used as an interval timer by using an underflow interrupt.

Table 8.1-2 Interval Time of 16-bit Reload Timer

Count Clock	Count Clock Cycle	Interval Time
Internal clock mode	2^1T (0.125 µs)	0.125 µs to 8.192 ms
	2^3T (0.5 µs)	0.5 µs to 32.768 ms
	2^5T (2.0 µs)	2.0 µs to 131.1 ms
Event count mode	2^3T or more	0.5 µs

T: Machine cycle
The values in interval time and the parenthesized values are provided when the machine clock operates at 16 MHz.

Reference: The 16-bit reload timer 1 can be used as the clock input source of the UART1 and the start trigger of the A/D converter.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

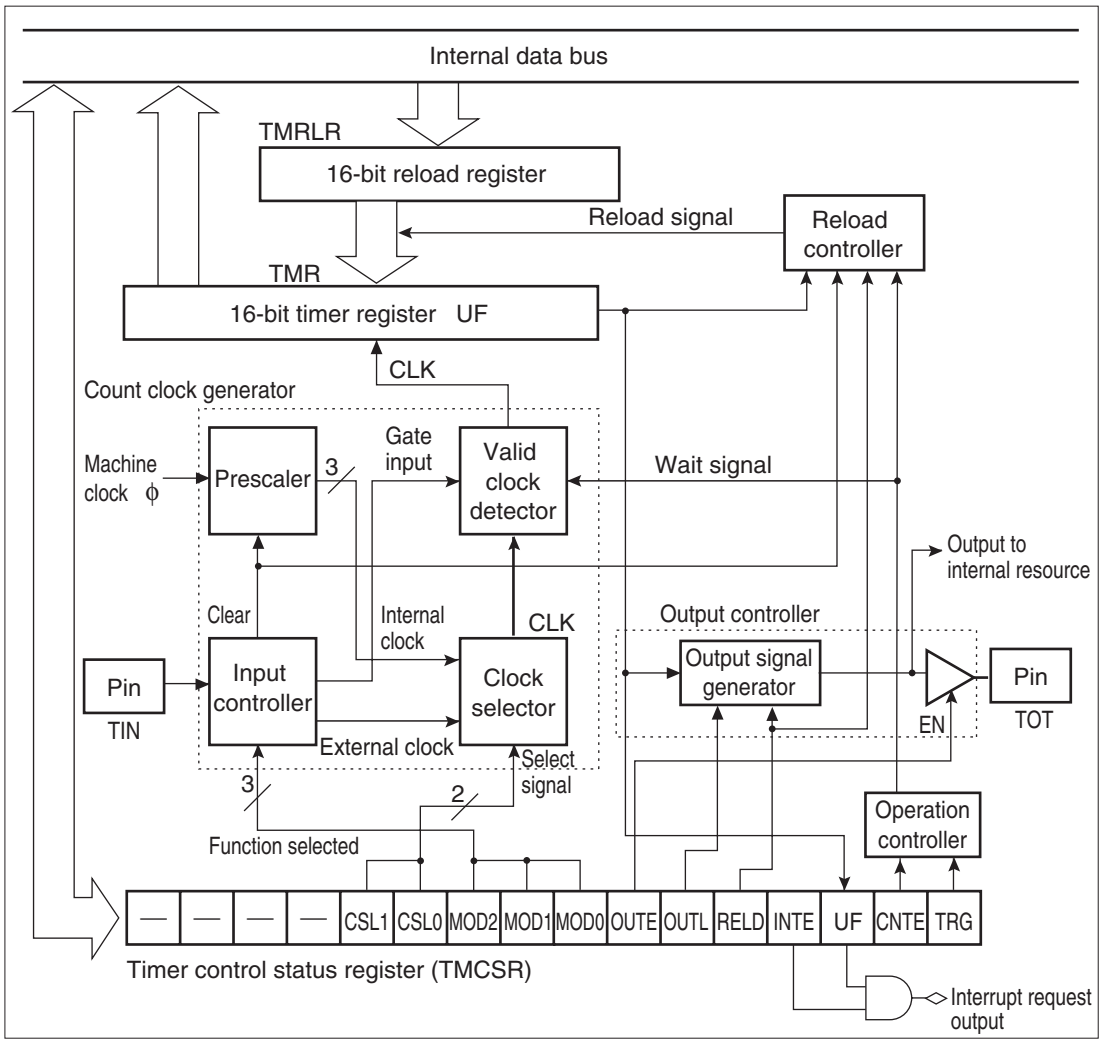
8.2 Block Diagram of 16-bit Reload Timer

The 16-bit reload timers 0 and 1 are composed of the following seven blocks:

- **Count clock generator**
- **Reload controller**
- **Output controller**
- **Operation controller**
- **16-bit timer register (TMR)**
- **16-bit reload register (TMRLR)**
- **Timer control status register (TMCSR)**

■ Block Diagram of 16-bit Reload Timer

Figure 8.2-1 Block Diagram of 16-bit Reload Timer



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Details of pins in block diagram

There are two channels for 16-bit reload timer.

The actual pin names, outputs to resources, and interrupt request numbers for each channel are as follows:

16-bit reload timer 0:

TIN pin: P20/TIN0

TOT pin: P21/TOT0

Interrupt request number: #17 (11_H)

16-bit reload timer 1:

TIN pin: P22/TIN1

TOT pin: P23/TOT1

Output to resources: Clock input source of UART1 and start trigger of A/D converter

Interrupt request number: #36 (24_H)

● Count clock generator

The count clock generator generates a count clock supplied to the 16-bit timer register (TMR) on the basis of the machine clock or external event clock.

● Reload controller

When the 16-bit reload timer starts operation or the TMR underflows, the reload controller reloads the value set in the 16-bit reload register (TMRLR) to the TMR.

● Output controller

The output controller inverts and enables or disables the output of the TOT pin at underflow.

● Operation controller

The operation controller starts or stops the 16-bit reload timer.

● 16-bit timer register (TMR)

The 16-bit timer register (TMR) is a 16-bit down counter. At read, the value being counted is read.

● 16-bit reload register (TMRLR)

The 16-bit reload register (TMRLR) sets the interval time of the 16-bit reload timer. When the 16-bit reload timer starts operation or the 16-bit timer register (TMR) underflows, the value set in the TMRLR is reloaded to the TMR.

● Timer control status register (TMCSR)

The timer control status register (TMCSR) selects the operation mode, sets the operation conditions, selects the start trigger, performs a start using the software trigger, selects the reload operation mode, enables or disables an interrupt request, sets the output level of the TOT pin, and sets the TOT output pin.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

8.3 Configuration of 16-bit Reload Timer

This section explains the pins, registers, and interrupt factors of the 16-bit reload timer.

■ Pins of 16-bit Reload Timer

The pins of the 16-bit reload timer serve as general-purpose I/O ports. Table 8.3-1 shows the pin functions and the pin settings required to use the 16-bit reload timer.

Table 8.3-1 Pins of 16-bit Reload Timer

Pin Name	Pin Function	Pin Setting Required for Use in 16-bit Reload Timer
TIN0	General-purpose I/O port, 16-bit reload timer input	Set as input port in port direction register (DDR).
TOT0	General-purpose I/O port, 16-bit reload timer output	Set timer output enable (TMCSR0: OUTE = 1).
TIN1	General-purpose I/O port, 16-bit reload timer input	Set as input port in port direction register (DDR).
TOT1	General-purpose I/O port, 16-bit reload timer output	Set timer output enable (TMCSR1: OUTE = 1).

et4U.com DataShee DataSheet4U.com

■ Block Diagram for Pins of 16-bit Reload Timer

For details of the block diagram for pins, see "CHAPTER 4 I/O PORT".

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ List of Registers and Reset Values of 16-bit Reload Timer

● Registers of 16-bit reload timer 0

Figure 8.3-1 List of Registers and Reset Values of 16-bit Reload Timer 0

	bit	15	14	13	12	11	10	9	8
Timer control status register (High) (TMCSR0)		X	X	X	X	0	0	0	0
	bit	7	6	5	4	3	2	1	0
Timer control status register (Low) (TMCSR0)		0	0	0	0	0	0	0	0
	bit	15	14	13	12	11	10	9	8
16-bit timer register (High) (TMR0)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
16-bit timer register (Low) (TMR0)		X	X	X	X	X	X	X	X
	bit	15	14	13	12	11	10	9	8
16-bit reload register (High) (TMRLR0)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
16-bit reload register (Low) (TMRLR0)		X	X	X	X	X	X	X	X

X: Undefined

DataSheet4U.com

● Registers of 16-bit reload timer 1

Figure 8.3-2 List of Registers and Reset Values of 16-bit Reload Timer 1

	bit	15	14	13	12	11	10	9	8
Timer control status register (High) (TMCSR1)		X	X	X	X	0	0	0	0
	bit	7	6	5	4	3	2	1	0
Timer control status register (Low) (TMCSR1)		0	0	0	0	0	0	0	0
	bit	15	14	13	12	11	10	9	8
16-bit timer register (High) (TMR1)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
16-bit timer register (Low) (TMR1)		X	X	X	X	X	X	X	X
	bit	15	14	13	12	11	10	9	8
16-bit reload register (High) (TMRLR1)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
16-bit reload register (Low) (TMRLR1)		X	X	X	X	X	X	X	X

X: Undefined

DataSheet4U.com www.DataSheet4U.com

DataSh4U.com

www.DataSheet4U.com

■ Generation of Interrupt Request from 16-bit Reload Timer

When the 16-bit reload timer is started and the count value of the 16-bit timer register is decremented from "0000_H" to "$FFFF_H$", an underflow occurs. When an underflow occurs, the UF bit in the timer control status register is set to 1 (TMCSR: UF). If an underflow interrupt is enabled (TMCSR: INTE = 1), an interrupt request is generated.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

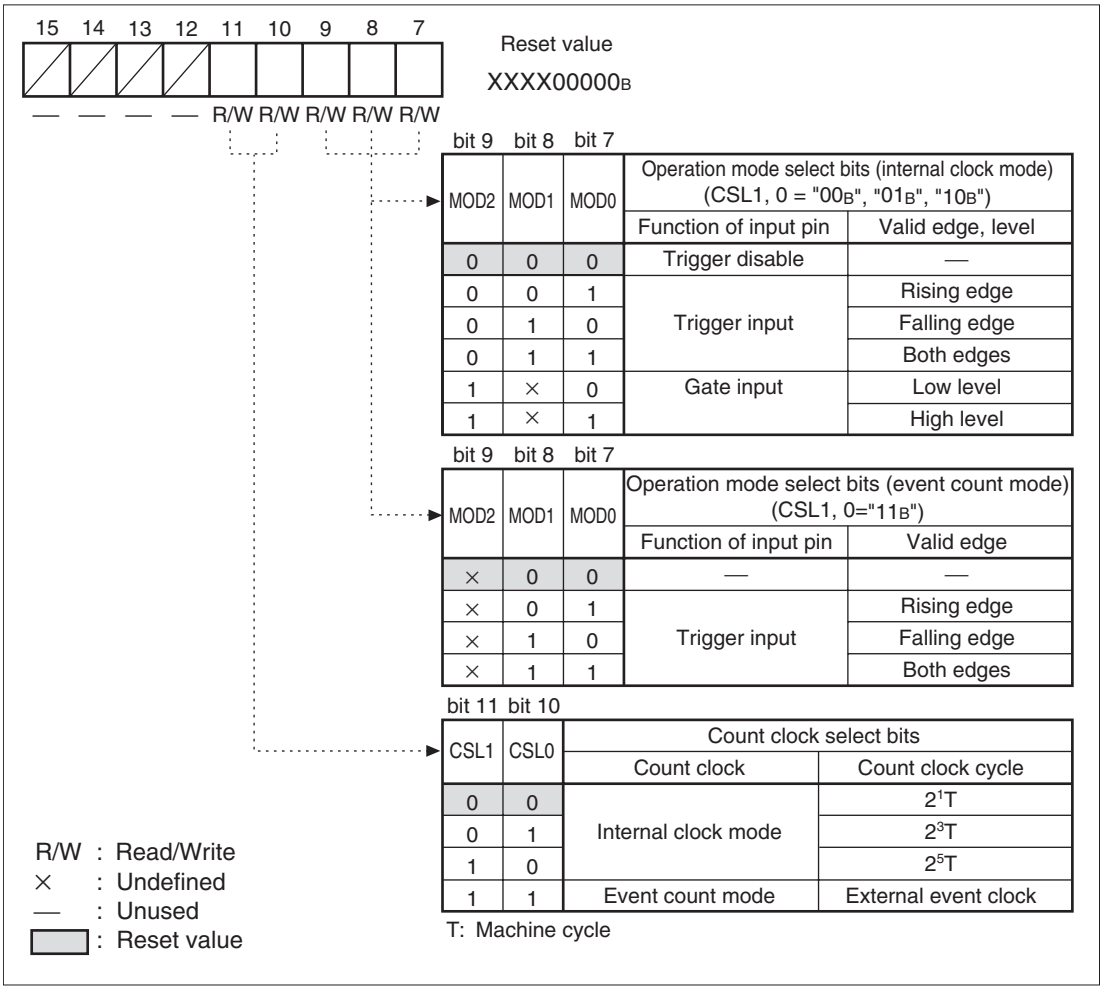
www.DataSheet4U.com

8.3.1 Timer Control Status Registers (High) (TMCSR0: H, TMCSR1: H)

The timer control status registers (High) (TMCSR0: H, TMCSR1: H) set the operation mode and count clock.
This section also explains the bit 7 in the timer control status registers (Low) (TMCSR0: L, TMCSR1: L).

■ Timer Control Status Registers (High) (TMCSR0: H, TMCSR1: H)

Figure 8.3-3 Timer Control Status Registers (High) (TMCSR0: H, TMCSR1: H)



bit 9 bit 8 bit 7

MOD2	MOD1	MOD0	Operation mode select bits (internal clock mode) (CSL1, 0 = "00B", "01B", "10B")	
			Function of input pin	Valid edge, level
0	0	0	Trigger disable	—
0	0	1	Trigger input	Rising edge
0	1	0		Falling edge
0	1	1		Both edges
1	×	0	Gate input	Low level
1	×	1		High level

bit 9 bit 8 bit 7

MOD2	MOD1	MOD0	Operation mode select bits (event count mode) (CSL1, 0="11B")	
			Function of input pin	Valid edge
×	0	0	—	—
×	0	1	Trigger input	Rising edge
×	1	0		Falling edge
×	1	1		Both edges

bit 11 bit 10

CSL1	CSL0	Count clock select bits	
		Count clock	Count clock cycle
0	0	Internal clock mode	2^1T
0	1		2^3T
1	0		2^5T
1	1	Event count mode	External event clock

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 8.3-2 Functions of Timer Control Status Registers (High) (TMCSR0: H, TMCSR1: H)

Bit Name		Function
bit 7 to bit 9	MOD2, MOD1, MOD0: Operation mode select bits	These bits set the operation conditions of the 16-bit reload timer. **[Internal clock mode]** The MOD2 bit is used to select the function of the input pin. **When MOD2 bit set to 0:** The input pin functions as a trigger input. The MOD1 and MOD0 bits are used to select the edge to be detected. When the edge is detected, the value set in the 16-bit reload register (TMRLR) is reloaded in the 16-bit timer register (TMR), starting the count operation of the TMR. **When MOD2 set to 1:** The input pin functions as a gate input. The MOD1 bit is not used. The MOD0 bit is used to select the signal level (High or Low) to be detected. The count operation of the 16-bit timer register (TMR) is performed only when the signal level is input. **[Event count mode]** The MOD2 bit is not used. An external event clock is input from the input pin. The MOD1 and MOD0 bits are used to select the edge to be detected.
bit 10 bit 11	CSL1, CSL0: Count clock select bits	These bits select the count clock of the 16-bit reload timer. **When set to anything other than "11_B":** This bits count by the internal clock (internal clock mode). **When set to "11_B":** The edge of the external event clock is counted (event count mode)
bit 12 to bit 15	Unused bits	**Read:** The value is undefined. **Write:** No effect

DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

8.3.2 Timer Control Status Registers (Low) (TMCSR0: L, TMCSR1: L)

The timer control status registers (Low) (TMCSR0: L, TMCSR1: L) enables or disables the timer operation, checks the generation of a software trigger or an underflow, enables or disables an underflow interrupt, selects the reload mode, and sets the output of the TOT pin.

■ Timer Control Status Registers (Low) (TMCSR0: L, TMCSR1: L)

Figure 8.3-4 Timer Control Status Registers (Low) (TMCSR0: L, TMCSR1: L)

7	6	5	4	3	2	1	0
*							
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value
00000000B

bit 0

TRG	Software trigger bit
0	No effect
1	After reloading, starts counting

bit 1

CNTE	Timer operation enable bit
0	Timer operation disable
1	Timer operation enable (start trigger wait)

bit 2

UF	Underflow generaiton flag bit	
	Read	Write
0	No underflow	Clears UF bit
1	Underflow	No effect

bit 3

INTE	Underflow interrupt enable bit
0	Underflow interrupt disable
1	Underflow interrupt enable

bit 4

RELD	Reload select bit
0	One-shot mode
1	Reload mode

bit 5

OUTL	TOT pin output level select bit	
	One-shot mode (RELD=0)	Reload mode (RELD=1)
0	High rectangular wave output during counting	Low toggle output at starting reload timer
1	Low rectangular wave output during counting	High toggle output at starting reload timer

bit 6

OUTE	TOT pin output enable bit		
	Pin function	Register and pin corresponding to each channel	
		TMCSR0	TMCSR1
0	General-purpose I/O port	General-purpose I/O port	General-purpose I/O port
1	TOT output	TOT0	TOT1

R/W : Read/Write
□ : Reset value
* : For MOD0 (bit 7), see section 8.3.1 "Timer Control Status Registers (High) (TMCSR0: H, TMCSR1: H)".

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 8.3-3 Timer Control Status Registers (Low) (TMCSR0: L, TMCSR1: L)

Bit Name		Function
bit 0	TRG: Software trigger bit	This bit starts the 16-bit reload timer by software. The software trigger function works only when the timer operation is enabled (CNTE = 1). **When set to 0:** Disabled. The state remains unchanged. **When set to 1:** Reloads value set in 16-bit reload register (TMRLR) to 16-bit timer register (TMR), starting TMR count operation **Read:** 0 is always read.
bit 1	CNTE: Timer operation enable bit	This bit enables or disables the operation of the 16-bit reload timer. **When set to1:** 16-bit reload timer enters start trigger wait state. When the start trigger is input, the timer register restarts count operation. **When set to 0:** Stops count operation
bit 2	UF: Underflow generation flag bit	This bit indicates that the TMR underflows. **When set to 0:** Clears this bit **When set to 1:** No effect **Read by read modify write instructions:** 1 is always read.
bit 3	INTE: Underflow interrupt enable bit	This bit enables or disables an underflow interrupt. When an underflow occurs (TMCSR: UF = 1) with an underflow interrupt enabled (TMCSR: INTE = 1), an interrupt request is generated.
bit 4	RELD: Reload select bit	This bit sets the reload operation at underflow. **When set to 1:** At underflow, reloads value set in TMRLR to TMR, continuing count operation (reload mode) **When set to 0:** At underflow, stops count operation (one-shot mode)
bit 5	OUTL: TOT Pin output level select bit	This bit sets the output level of the output pin of the 16-bit reload timer. **<One-shot mode (RELD = 0)>** **When set to 0:** Outputs High-level rectangular wave during TMR count operation **When set to 1:** Outputs Low-level rectangular wave during TMR count operation **<Reload mode (RELD = 1)>** **When set to 0:** Outputs Low-level toggle wave when 16-bit reload timer started **When set to 1:** Outputs High-level toggle wave when 16-bit reload timer started
bit 6	OUTE: TOT Output enable bit	This bit sets the function of the TOT pin of the 16-bit reload timer. **When set to 0:** Functions as general-purpose I/O port **When set to 1:** Functions as TOT pin of 16-bit reload timer

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

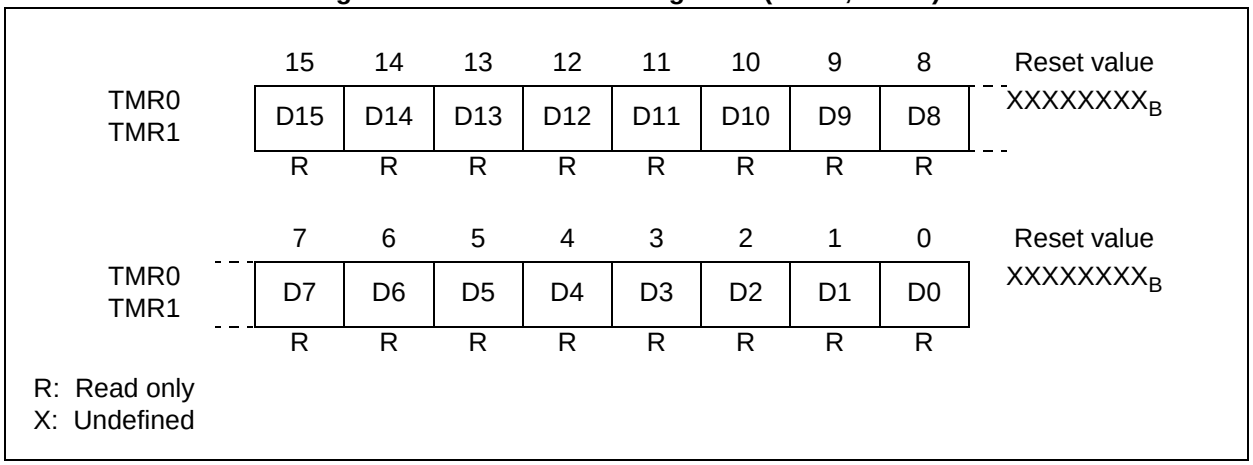
www.DataSheet4U.com

8.3.3 16-bit Timer Registers (TMR0, TMR1)

The 16-bit timer registers (TMR0, TMR1) are 16-bit down counters. At read, the value being counted is read.

■ 16-bit Timer Registers (TMR0, TMR1)

Figure 8.3-5 16-bit Timer Registers (TMR0, TMR1)



When the timer operation is enabled (TMCSR: CNTE = 1) and the start trigger is input, the value set in the 16-bit reload register (TMRLR) is reloaded to the 16-bit timer register (TMR), starting the TMR count operation.

When the timer operation is disabled (TMCSR: CNTE = 0), the TMR value is retained.

When the TMR value is counted down from "0000_H" to "$FFFF_H$" during the TMR count operation, an underflow occurs.

[Reload mode]

When the TMR underflows, the value set in the TMRLR is reloaded to the TMR, starting the TMR count operation.

[One-shot mode]

When the TMR underflows, the TMR count operation is stopped, entering the start trigger input wait state. The TMR value is retained to "$FFFF_H$".

Notes:
- The TMR can be read during the TMR count operation. However, always use the word instruction (MOVW).
- The TMR and the TMRLR are assigned to the same address. At write, the set value can be written to the TMRLR without affecting the TMR. At read, the TMR value being counted can be read.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

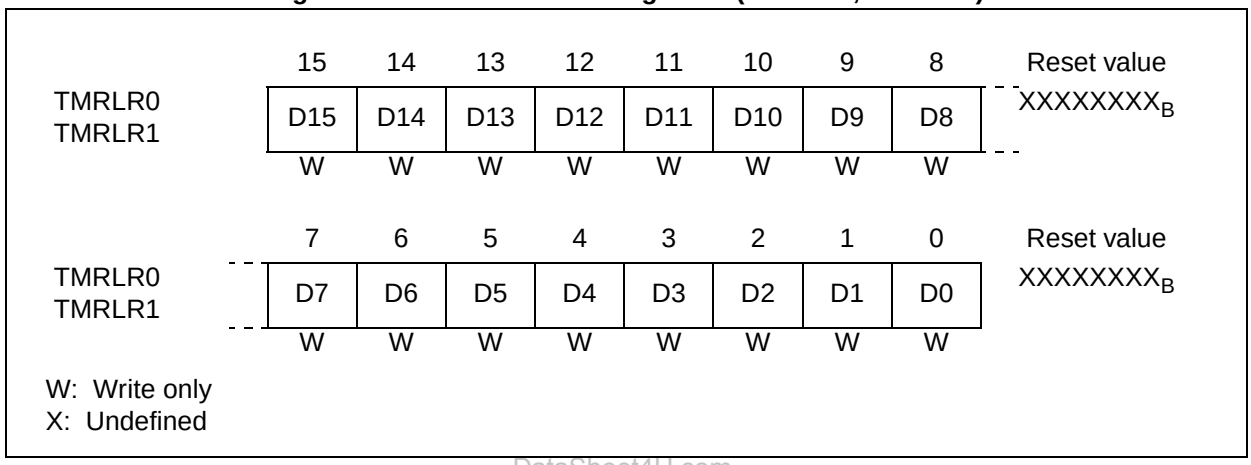
www.DataSheet4U.com

8.3.4 16-bit Reload Registers (TMRLR0, TMRLR1)

The 16-bit reload registers (TMRLR0, TMRLR1) set the value to be reloaded to the 16-bit timer register (TMR). When the start trigger is input, the value set in the 16-bit reload registers (TMRLR0, TMRLR1) is reloaded to the TMR, starting the TMR count operation.

■ 16-bit Reload Registers (TMRLR0, TMRLR1)

Figure 8.3-6 16-bit Reload Registers (TMRLR0, TMRLR1)



DataSheet4U.com

Set the 16-bit reload registers (TMRLR0, TMRLR1) after disabling the timer operation (TMCSR: CNTE = 0). After completing setting of the 16-bit reload registers (TMRLR0, TMRLR1), enable the timer operation (TMCSR: CNTE = 1).

When the start trigger is input, the value set in the TMRLR is reloaded to the TMR, starting the TMR count operation.

Notes:
- Perform a write to the TMRLR after disabling the operation of the 16-bit reload timer (TMCSR: CNTE = 0). Always use the word instruction (MOVW).
- The TMRLR and the TMR are assigned to the same address. At write, the set value can be written to the TMRLR without affecting the TMR. At read, the TMR value being counted is read.
- Instructions, such as the INC/DEC instruction, which provide the read modify write (RMW) operation cannot be used.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

8.4 Interrupts of 16-bit Reload Timer

The 16-bit reload timer generates an interrupt request when the 16-bit timer register (TMR) underflows.

■ Interrupts of 16-bit Reload Timer

When the value of the TMR is decremented from "0000_H" to "$FFFF_H$" during the TMR count operation, an underflow occurs. When an underflow occurs, the underflow generation flag bit in the timer control status register (TMCSR: UF) is set to l. When an underflow interrupt is enabled (TMCSR: INTE = 1), an interrupt request is generated.

Table 8.4-1 Interrupt Control Bits and Interrupt Factors of 16-bit Reload Timer

	16-bit Reload Timer 0	16-bit Reload Timer 1
Interrupt request flag bit	TMCSR0: UF	TMCSR1: UF
Interrupt request enable bit	TMCSR0: INTE	TMCSR1: INTE
Interrupt factor	Underflow in TMR0	Underflow in TMR1

et4U.com

■ Correspondence between 16-bit Reload Timer Interrupt and EI^{2}OS

DataShee

For details of the interrupt number, interrupt control register, and interrupt vector address, see "3.5 Interrupt".

DataSheet4U.com

■ EI^{2}OS Function of 16-bit Reload Timer

The 16-bit reload timer corresponds to the EI^{2}OS function. An underflow in the TMR starts the EI^{2}OS.

The EI^{2}OS is available only when other resources sharing the interrupt control register (ICR) do not use interrupts. When using the EI^{2}OS in the 16-bit reload timers 0 and 1, it is necessary to disable generation of interrupt requests by resources sharing the interrupt control register (ICR) with the 16-bit reload timers 0 and 1.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

8.5 Explanation of Operation of 16-bit Reload Timer

This section explains the setting of the 16-bit reload timer and the operation state of the counter.

■ Setting of 16-bit Reload Timer

● Setting of internal clock mode

Counting the internal clock requires the setting shown in Figure 8.5-1.

Figure 8.5-1 Setting of Internal Clock Mode

	bit15	14	13	12	11	10	9	bit8	bit7	6	5	4	3	2	1	bit0
TMCSR	—	—	—	—	CSL1	CSL0	MOD2	MOD1	MOD0	OUTE	OUTL	RELD	INTE	UF	CNTE	TRG
					Except "11_B"		◎	◎	◎	◎	◎	◎	◎	◎	1	◎

TMRLR	Sets a reload value to 16-bit timer register

◎ : Used bit
1: Set 1

DataSheet4U.com

● Setting of event count mode

Inputting an external event to operate the 16-bit reload timer requires the setting shown in Figure 8.5-2.

Figure 8.5-2 Setting of Event Count Mode

	bit 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	bit 0
TMCSR	—	—	—	—	CSL1	CSL0	MOD2	MOD1	MOD0	OUTE	OUTL	RELD	INTE	UF	CNTE	TRG
					1	1	◎	◎	◎	◎	◎	◎	◎	◎	1	◎

TMRLR	Sets a reload value to 16-bit timer register

Set the bit of DDR (port direction register) corresponding to the pin to be used as TIN pin to "0".

◎ : Used bit
1 : Set 1

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

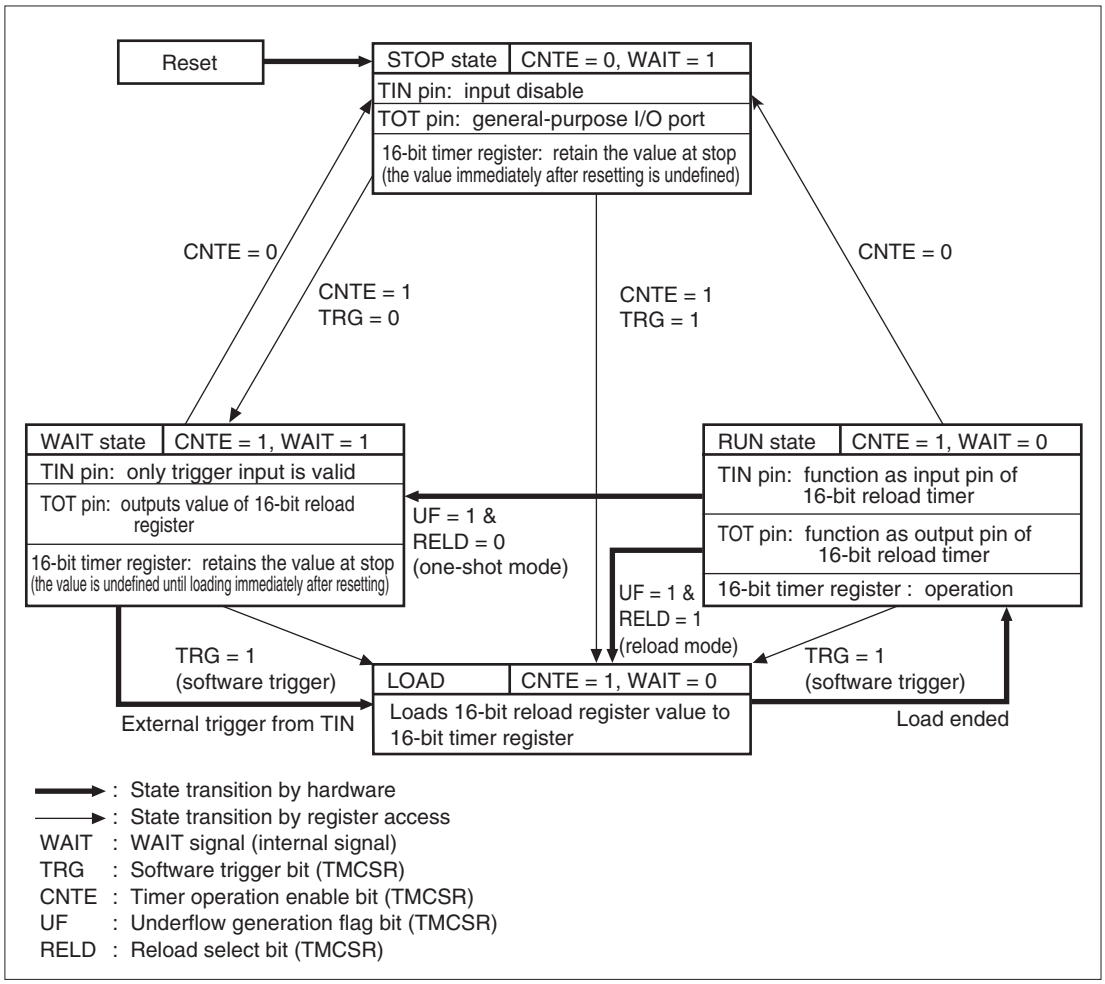
www.DataSheet4U.com

■ Operating State of 16-bit Timer Register

The operating state of the 16-bit timer register is determined by the timer operation enable bit in the timer control status register (TMCSR: CNTE) and the WAIT signal. The operating states include the stop state, start trigger input wait state (WAIT state), and RUN state.

Figure 8.5-3 shows the state transition diagram for the 16-bit timer registers.

Figure 8.5-3 State Transition Diagram



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

8.5.1 Operation in Internal Clock Mode

In the internal clock mode, three operation modes can be selected by setting the operation mode select bits in the timer control status register (TMCSR: MOD2 to MOD0). When the operation mode and reload mode are set, a rectangular wave or a toggle wave is output from the TOT pin.

■ Setting of Internal Clock Mode

- By setting the count clock select bits (CSL1, CSL0) in the timer control status register to "00_B", "01_B" or "10_B", the 16-bit reload timer (TMRLR) is set to the internal clock mode.
- In the internal clock mode, the 16-bit timer register (TMR) decrements in synchronization with the internal clock.
- In the internal clock mode, three count clock cycles can be selected by setting the count clock select bits in the timer control status register (TMCSR: CSL1, CSL0).

[Setting a reload value to TMR]

After the 16-bit reload timer is started, the value set in the TMRLR is reloaded to the TMR.

1. Disables the timer operation (TMCSR: CNTE = 0).
2. Sets a reload value to the TMR in the TMRLR.
3. Enables the timer operation (TMCSR: CNTE = 1).

et4U.com DataShee DataSheet4U.com

Note:	It takes 1 machine cycle (time) to reload the value set in the TMRLR to the TMR after the start trigger is input.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Operation as 16-bit Timer Register Underflows

When the value of the 16-bit timer register (TMR) is decremented from "0000_H" to "$FFFF_H$" during the TMR count operation, an underflow occurs.

- When an underflow occurs, the underflow generation flag bit in the timer control status register (TMCSR: UF) is set to 1.
- When the underflow interrupt enable bit in the timer control status register (TMCSR: INTE) is set to 1, an underflow interrupt is generated.
- The reload operation when an underflow occurs is set by the reload select bit in the timer control status register (TMCSR: RELD).

[One-shot mode (TMCSR: RELD = 0)]

When an underflow occurs, the count operation of the TMR is stopped, entering the start trigger input wait state. When the next start trigger is input, the TMR count operation is restarted.

In the one-shot mode, a rectangular wave is output from the TOT pin during the TMR count operation. The pin output level select bit in the timer control status register (TMCSR: OUTL) can be set to select the level (High or Low) of a rectangular wave.

[Reload mode (TMCSR: RELD = 1)]

When an underflow occurs, the value set in the 16-bit reload timer register (TMRLR) is reloaded to the TMR, continuing the TMR count operation.

In the reload mode, a toggle wave inverting the output level of the TOT pin is output each time an underflow occurs during the TMR count operation. The pin output level select bit in the timer control status register (TMCSR: OUTL) can be set to select the level (High or Low) of a toggle wave as the 16-bit reload timer is started.

et4U.com DataShee DataSheet4U.com

■ Operation in Internal Clock Mode

In the internal clock mode, the operation mode select bits in the timer control status register (TMCSR: MOD2 to MOD0) can be used to select the operation mode. Disable the timer operation by setting the timer operation enable bit in the timer control status register (TMCSR: CNTE).

[Software trigger mode (MOD2 to MOD0 ="000_B")]

If the software trigger mode is set, start the 16-bit reload timer by setting the software trigger bit in the timer control status register (TMCSR: TRG) to 1. When the 16-bit reload timer is started, the value set in the TMRLR is reloaded to the TMR, starting the TMR count operation.

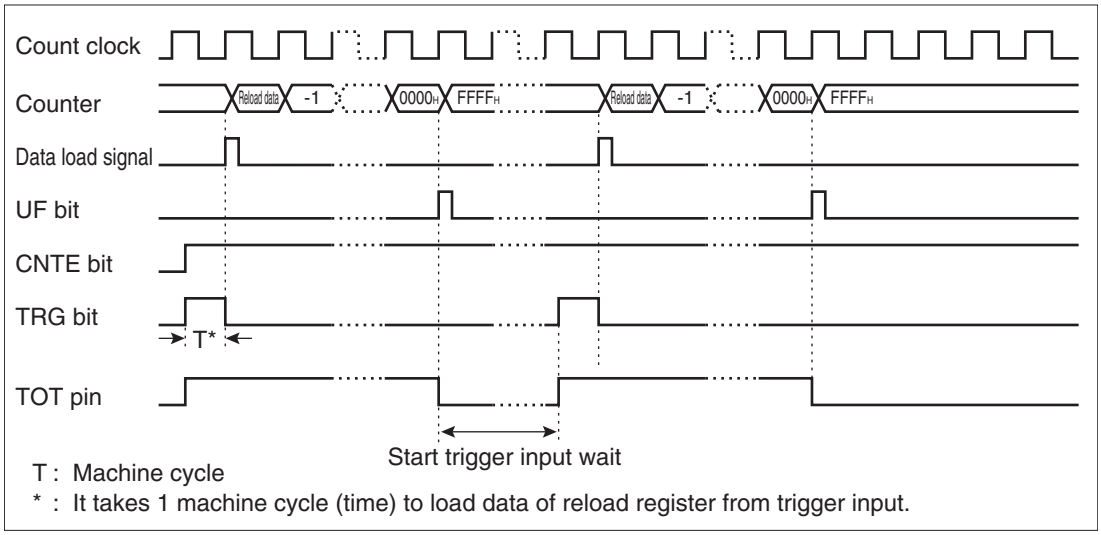
Note:	When both the timer operation enable bit in the timer control status register (TMCSR: CNTE) and the software trigger bit in the timer control status register (TMCSR: TRG) are set to 1, the 16-bit reload timer and the count operation of the TMR are started simultaneously.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

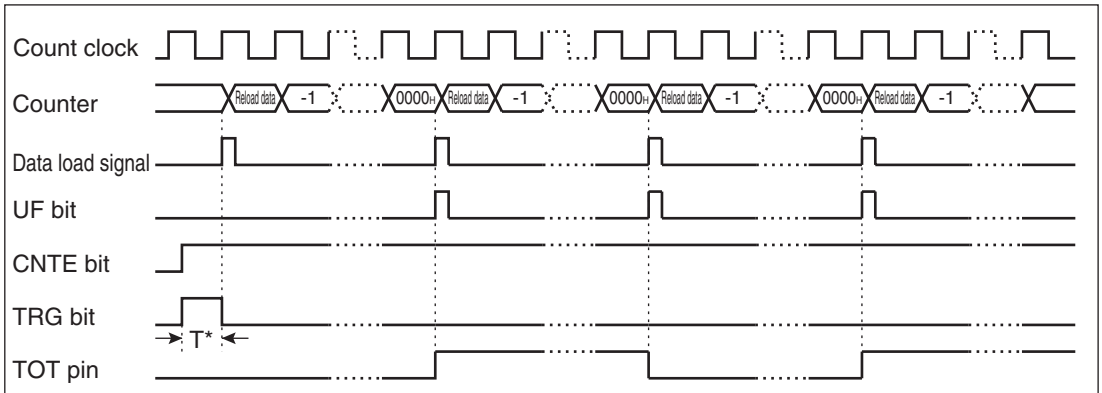
www.DataSheet4U.com

Figure 8.5-4 Count Operation in Software Trigger Mode (One-shot Mode)



T : Machine cycle
* : It takes 1 machine cycle (time) to load data of reload register from trigger input.

Figure 8.5-5 Count Operation in Software Trigger Mode (Reload Mode)



T : Machine cycle
* : It takes 1 machine cycle (time) to load data of reload register from trigger input.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

[External trigger mode (MOD2 to MOD0 =" 001_B", "010_B", "011_B")]

When the external trigger mode is set, the 16-bit reload timer is started by inputting the external valid edge to the TIN pin. When the 16-bit reload timer is started, the value set in the 16-bit reload register (TMRLR) is reloaded to the 16-bit timer register (TMR), starting the TMR count operation.

- By setting the operation mode select bits in the timer control status register (TMCSR: MOD2 to MOD0), the detected edge can be selected from the rising edge, falling edge, and both edges.

Note:	The trigger pulse width of the edge to be input to the TIN pin should be 2 machine cycles (time) or more.

Figure 8.5-6 Count Operation in External Trigger Mode (One-shot Mode)

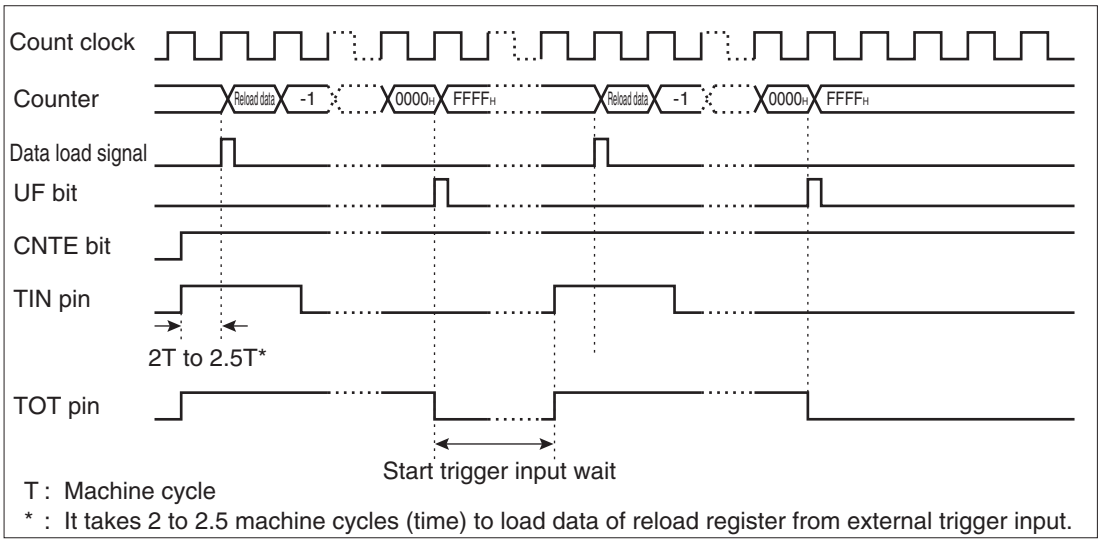
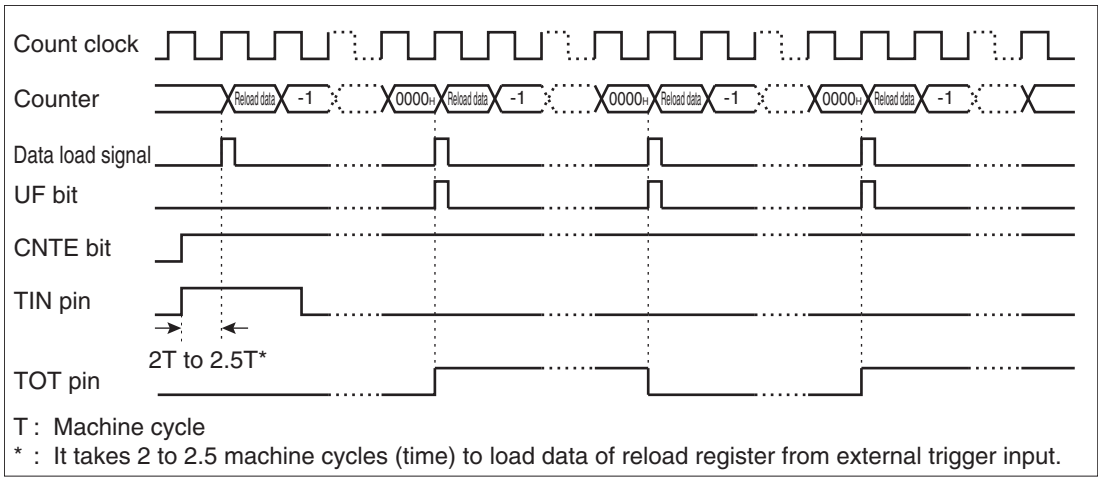


Figure 8.5-7 Count Operation in External Trigger Mode (Reload Mode)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

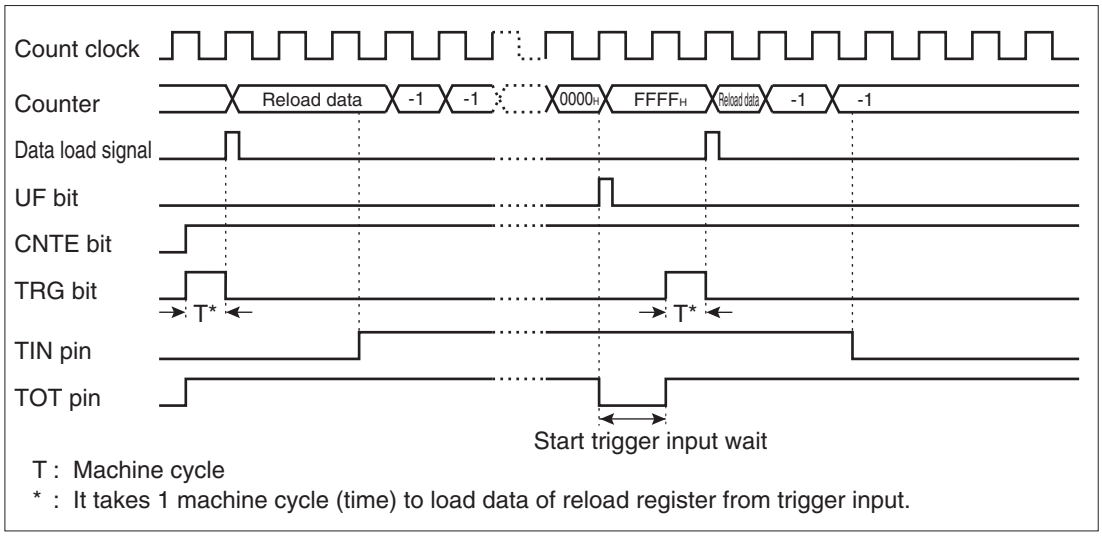
www.DataSheet4U.com

[External gate input mode (MOD2 to MOD0 = "1x0$_B$", "1x1$_B$")]

When the external gate input mode is set, start the 16-bit reload timer by setting the software trigger bit in the timer control status register (TMCSR: TRG) to 1. When the 16-bit reload timer is started, the value set in the 16-bit reload register (TMRLR) is reloaded to the 16-bit timer register (TMR).

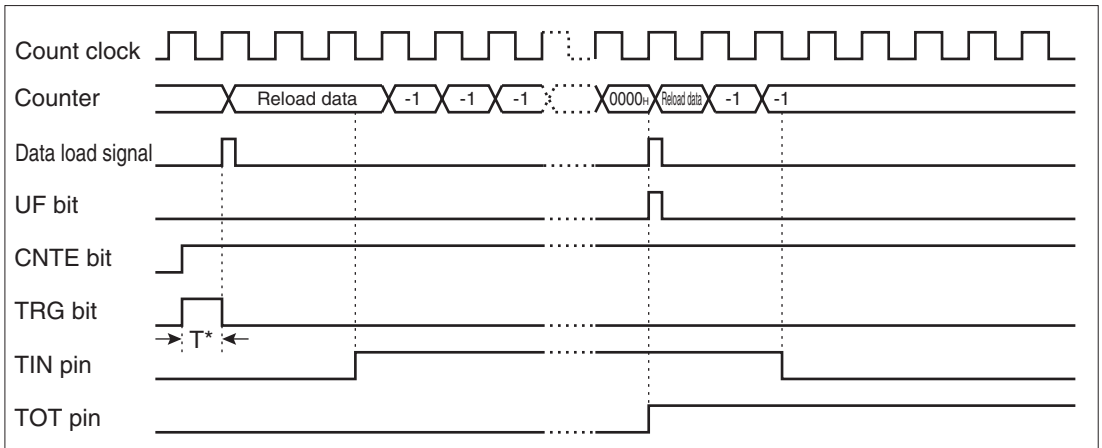
- After the 16-bit reload timer is started, the count operation of the TMR is performed while the set gate input level is input to the TIN pin.
- The gate input level (High or Low) can be selected by setting the operation mode select bits in the timer control status register (TMCSR: MOD2 to MOD0).

Figure 8.5-8 Count Operation in External Gate Input Mode (One-shot Mode)



T : Machine cycle
* : It takes 1 machine cycle (time) to load data of reload register from trigger input.

Figure 8.5-9 Count Operation in External Gate Input Mode (Reload Mode)



T : Machine cycle
* : It takes 1 machine cycle (time) to load data of reload register from trigger input.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

8.5.2 Operation in Event Count Mode

In the event count mode, after the 16-bit reload timer is started, the edge of the signal input to the TIN pin is detected to perform the count operation of the 16-bit timer register (TMR). When the operation mode and the reload mode are set, a rectangular wave or a toggle wave is output from the TOT pin.

■ Setting of Event Count Mode

- The 16-bit reload timer is placed in the event count mode by setting the count clock select bits in the timer control status register (TMCSR: CSL1, CSL0) to "11_B".
- In the event count mode, the TMR decrements in synchronization with the edge detection of the external event clock input to the TIN pin.

[Setting initial value of counter]

After the 16-bit reload timer is started, the value set in the TMRLR is reloaded to the TMR.

1. Disables the operation of the 16-bit reload timer (TMCSR: CNTE = 0).
2. Sets a reload value to the TMR in the TMRLR.
3. Enables the operation of the 16-bit reload timer (TMCSR: CNTE = 1).

et4U.com DataShee

Note: It takes 1 machine cycle (time) to load the value set in the TMRLR to the TMR after the start trigger is input.

DataSheet4U.com

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Operation as 16-bit Timer Register Underflows

When the value of the 16-bit timer register (TMR) is decremented from "0000_H" to "$FFFF_H$" during the TMR count operation, an underflow occurs.

- When an underflow occurs, the underflow generation flag bit in the timer control status register (TMCSR: UF) is set to 1.
- When the underflow interrupt enable bit in the timer control status register (TMCSR: INTE) is set to 1, an underflow interrupt is generated.
- The reload operation when an underflow occurs is set by the reload select bit in the timer control status register (TMCSR: RELD).

[One-shot mode (TMCSR: RELD = 0)]

When an underflow occurs, the TMR count operation is stopped, entering the start trigger input wait state. When the next start trigger is input, the TMR count operation is restarted.

In the one-shot mode, a rectangular wave is output from the TOT pin during the TMR count operation. The pin output level select bit in the timer control status register (TMCSR: OUTL) can be set to select the level (High or Low) of the rectangular wave.

[Reload mode (TMCSR: RELD = 1)]

When an underflow occurs, the value set in the TMRLR is reloaded to the TMR, continuing the TMR count operation.

In the reload mode, a toggle wave inverting the output level of the TOT pin is output each time an underflow occurs during the TMR count operation. The pin output level select bit in the timer control status register (TMCSR: OUTL) can be set to select the level (High or Low) of the toggle wave when the 16-bit reload timer is started.

et4U.com DataShee DataSheet4U.com

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Operation in Event Count Mode

The operation of the 16-bit reload timer is enabled by setting the timer operation enable bit in the timer control status register (TMCSR: CNTE) to 1. When the software trigger bit in the timer control status register (TMCSR: TRG) is set to 1, the 16-bit reload timer is started. When the 16-bit reload timer is started, the value set in the 16-bit reload register (TMRLR) is reloaded to the 16-bit timer register (TMR), starting the TMR count operation. After the 16-bit reload timer is started, the edge of the external event clock input to the TIN pin is detected to perform the TMR count operation.

- By setting the operation mode select bits in the timer control status register (TMCSR: MOD2 to MOD0), the detected edge can be selected from the rising edge, falling edge, and both edges.

Note: The level width of external event clock to be input to the TIN pin should be 4 machine cycles (time) or more.

Figure 8.5-10 Count Operation in Event Count Mode (One-shot Mode)

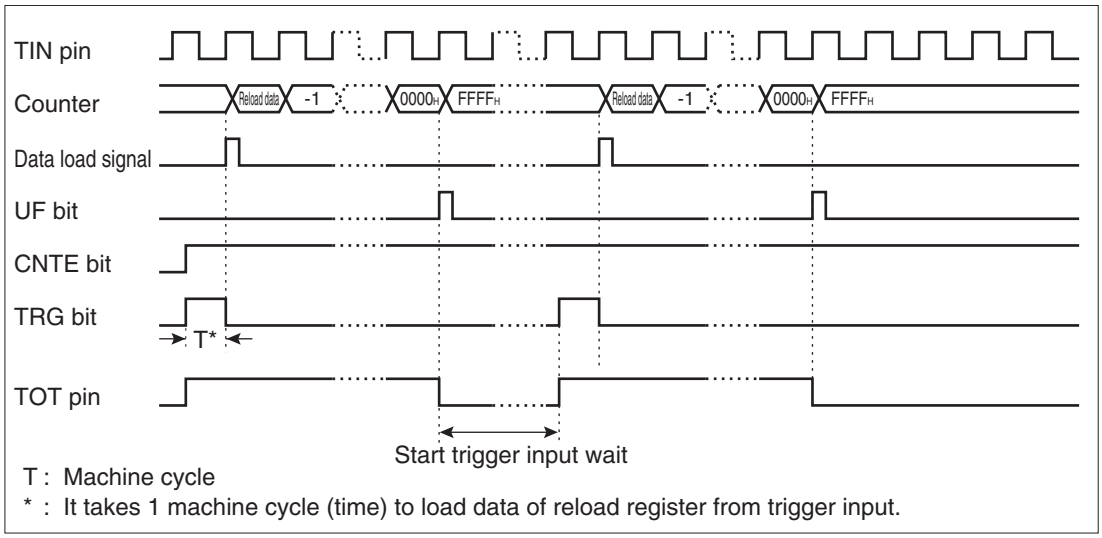
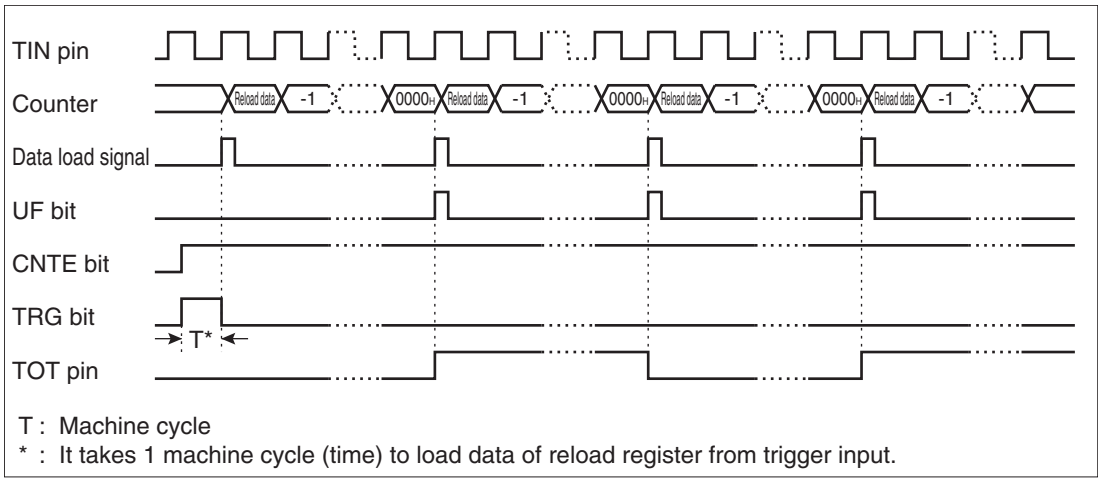


Figure 8.5-11 Count Operation in Event Count Mode (Reload Mode)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

8.6 Precautions when Using 16-bit Reload Timer

This section explains the precautions when using the 16-bit reload timer.

■ Precautions when Using 16-bit Reload Timer

● Precautions when setting by program

- Set the 16-bit reload register (TMRLR) after disabling the timer operation (TMCSR: CNTE = 0)
- The 16-bit timer register (TMR) can be read during the TMR count operation. However, always use the word instruction (MOVW).
- Change the CSL1 and CSL0 bits in the TMCSR after disabling the timer operation (TMCSR: CNTE = 0).

● Precautions on interrupt

- When the UF bit in the TMCSR is set to 1 and the underflow interrupt output is enabled (TMCSR: INTE = 1), it is impossible to return from interrupt processing. Always clear the UF bit. However, when the EI^2OS is used, the UF bit is cleared automatically.
- When using the EI^2OS in the 16-bit reload timer, it is necessary to disable generation of interrupt requests by resources that share the interrupt control register (ICR) with the 16-bit reload timer.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

8.7 Program Example of 16-bit Reload Timer

This section gives a program example of the 16-bit reload timer operated in the internal clock mode and the event count mode:

■ Program Example in Internal Clock Mode

● Processing specification

- The 25-ms interval timer interrupt is generated by the 16-bit reload timer 0.
- The repeated interrupts are generated in the reload mode.
- The timer is started using the software trigger instead of the external trigger input.
- EI^2OS is not used.
- The machine clock is 16 MHz; the count clock is 2 µs.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Coding example

```
ICR03   EQU  0000B3H          ; Interrupt control register for 16-bit reload timer
TMCSR0  EQU  000066H          ; Timer control status register
TMR0    EQU  003900H          ; 16-bit timer register
TMRLR0  EQU  003900H          ; 16-bit reload register
UF0     EQU  TMCSR0:2         ; Interrupt request flag bit
CNTE0   EQU  TMCSR0:1         ; Counter operation enable bit
TRG0    EQU  TMCSR0:0         ; Software trigger bit
;-----Main program-----------------------------------------------------------
CODE    CSEG
;        :                    ; Stack pointer (SP), already initialized
        AND  CCR,#0BFH        ; Interrupts disabled
        MOV  I:ICR03,#00H     ; Interrupt level 0 (highest)
        CLRB I:CNTE0          ; Counter suspended
        MOVW I:TMRLR0,#30D4H  ; Data set for 25-ms timer
        MOVW I:TMCSR0,#0000100000011011B
                              ; Operation of interval timer, clock = 2 ms.
                              ; External trigger disabled, external output disabled
                              ; Reload mode selected, interrupt enabled
                              ; Interrupt flag cleared, count started
        MOV  ILM,#07H         ; ILM in PS set to level 7
        OR   CCR,#40H         ; Interrupts enabled
LOOP:
         :
        Processing by user
         :
        BRA  LOOP
;-----Interrupt program-------------------------------------------------------
WARI:
        CLR  I:UF0            ; Interrupt request flag cleared
         :
        Processing by user
         :
        RETI                  ; Return from interrupt

CODE    ENDS
;-----Vector setting----------------------------------------------------------
VECT    CSEG ABS=0FFH
        ORG  00FFB8H          ; Vector set to interrupt #17 (11H)
        DSL  WARI
        ORG  00FFDCH          ; Reset vector set
        DSL  START
        DB   00H              ; Set to single-chip mode
VECT    ENDS
        END  START
```

et4U.com
DataShee
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Program Example in Event Count Mode

● Processing specification

- An interrupt is generated when rising edges of the pulse input to the external event input pin are counted 10000 times by the 16-bit reload timer 0.
- Operation is performed in the one-shot mode.
- The rising edge is selected for the external trigger input.
- EI^2OS is not used.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Coding example

```
ICR03   EQU  0000B3H          ; Interrupt control register for 16-bit reload timer
TMCSR0  EQU  000066H          ; Timer control status register
TMR0    EQU  003900H          ; 16-bit timer register
TMRLR0  EQU  003900H          ; 16-bit reload register
DDR2    EQU  000012H          ; Port data register
UF0     EQU  TMCSR0:2         ; Interrupt request flag bit
CNTE0   EQU  TMCSR0:1         ; Counter operation enable bit
TRG0    EQU  TMCSR0:0         ; Software trigger bit
;-----Main program-----------------------------------------------------------
CODE        CSEG
;        :                    ; Stack pointer (SP), already initialized
        AND  CCR,#0BFH        ; Interrupts disabled
        MOV  I:ICR03,#00H     ; Interrupt level 0 (highest)
        MOV  I:DDR2,00H       ; Sets P20/TIN0 pin to input
        CLRB I:CNTE0          ; Counter suspended
        MOVW I:TMRLR0,#2710H; Reload value set to 10000 times
        MOVW I:TMCSR0,#0000110000001011B
                              ; Counter operation, external trigger,
                              ; rising edge, and external output disabled
                              ; One-shot mode selected, interrupt enabled
                              ; Interrupt flag cleared, count started
      MOV   ILM,#07H          ; ILM in PS set to level 7
      OR    CCR,#40H          ; Interrupts enabled
LOOP:
         :
        Processing by user
         :
        BRA   LOOP
;-----Interrupt program------------------------------------------------------
WARI:
        CLR  I:UF0            ; Interrupt request flag cleared
         :
        Processing by user
         :
        RETI                  ; Return from interrupt

CODE    ENDS
;-----Vector setting---------------------------------------------------------
VECT    CSEG ABS=0FFH
        ORG  00FFB8H          ; Vector set to interrupt #17 (11H)
        DSL  WARI
        ORG  00FFDCH          ; Reset vector set
        DSL  START
        DB   00H              ; Set to single-chip mode
VECT    ENDS
        END  START
```

et4U.com
DataShee
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 9
WATCH TIMER

This section describes the functions and operations of the watch timer.

9.1 Overview of Watch Timer
9.2 Block Diagram of Watch Timer
9.3 Configuration of Watch Timer
9.4 Watch Timer Interrupt
9.5 Explanation of Operation of Watch Timer
9.6 Program Example of Watch Timer

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

9.1 Overview of Watch Timer

The watch timer is a 15-bit free-run counter that increments in synchronization with the subclock.

- **8 interval times can be selected and an interrupt request can be generated for each interval time.**
- **An operation clock can be supplied to the oscillation stabilization wait time timer of the subclock and the watchdog timer.**
- **The subclock is always used as a count clock regardless of the settings of the clock select register (CKSCR).**

■ Interval Timer Function

- When the watch timer reaches the interval time set by the interval time select bits (WTC: WTC2 to WTC0), the bit corresponding to the interval time of the watch timer counter overflows (carries) and the overflow flag bit is set (WTC: WTOF = 1).
- When the overflow flag bit is set (WTC: WTOF = 1) with interrupt enabled when an overflow occurs (WTC: WTIE = 1), an interrupt request is generated.
- The interval time of the watch timer can be selected from 8 types shown in Table 9.1-1.

et4U.com DataShee

Table 9.1-1 Interval Times of Watch Timer

DataSheet4U.com

Subclock Cycle	Interval Time
SCLK (122 μs)	2^8/SCLK (31.25 ms)
	2^9/SCLK (62.5 ms)
	2^{10}/SCLK (125 ms)
	2^{11}/SCLK (250 ms)
	2^{12}/SCLK (500 ms)
	2^{13}/SCLK (1.0 s)
	2^{14}/SCLK (2.0 s)
	2^{15}/SCLK (4.0 s)

SCLK: Subclock frequency
The parenthesized values are provided when the subclock operates at 8.192 kHz.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Cycle of Clock Supply

The watch timer supplies an operation clock to the oscillation stabilization wait time timer of the subclock and the watchdog timer. Table 9.1-2 shows the cycles of clocks supplied from the watch timer.

Table 9.1-2 Cycle of Clock Supply from Watch Timer

Where to Supply Clock	Clock Cycle
Timer for oscillation stabilization wait time of subclock	2^{14}/SCLK (2.000 s)
Watchdog timer	2^{10}/SCLK (125 ms)
	2^{13}/SCLK (1.000 s)
	2^{14}/SCLK (2.000 s)
	2^{15}/SCLK (4.000 s)

SCLK: Subclock frequency
The parenthesized values are provided when the subclock operates at 8.192 kHz.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

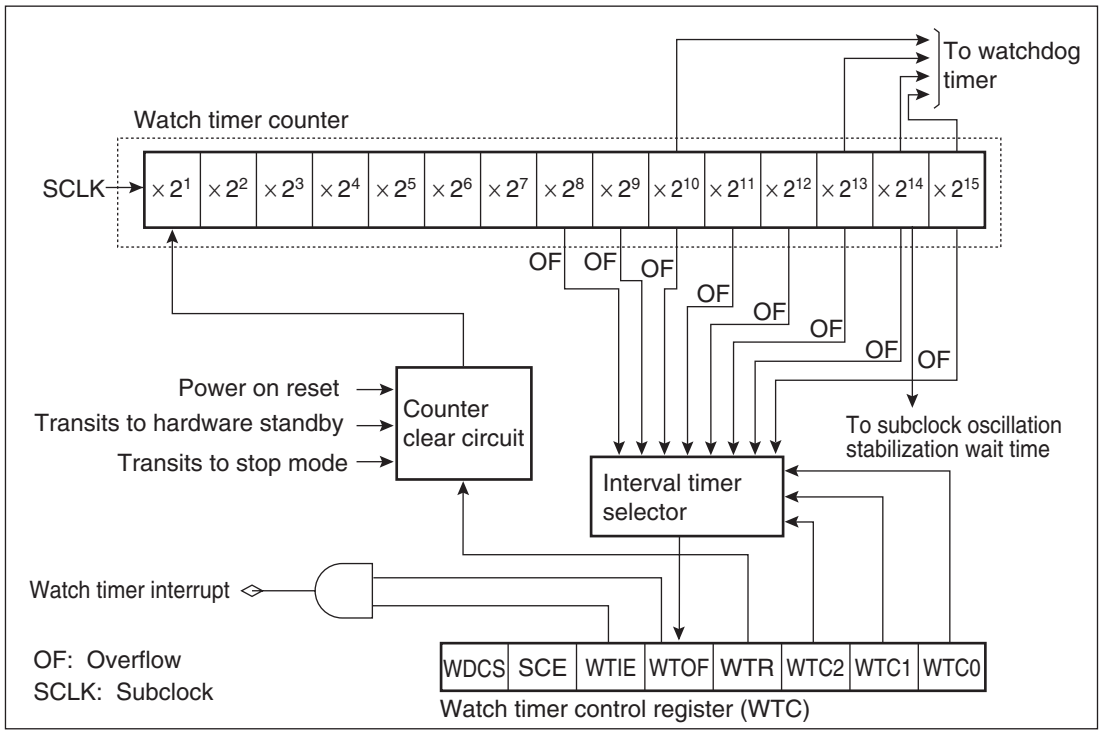
9.2 Block Diagram of Watch Timer

The watch timer consists of the following blocks:

- **Watch timer counter**
- **Counter clear circuit**
- **Interval timer selector**
- **Watch timer control register (WTC)**

■ Block Diagram of Watch Timer

Figure 9.2-1 Block Diagram of Watch Timer



The actual interrupt request number of the watch timer is as follows:

Interrupt request number: #28 ($1C_H$)

● Watch timer counter

The watch timer counter is a 15-bit up counter that uses the subclock (SCLK) as a count clock.

● Counter clear circuit

The counter-clear circuit clears the watch timer counter.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Interval timer selector

The interval timer selector sets the overflow flag bit when the watch timer counter reaches the interval time set in the watch timer control register (WTC).

● Watch timer control register (WTC)

The watch timer control register (WTC) selects the interval time, clears the watch timer counter, enables or disables an interrupt, checks the overflow (carries) state, and clears the overflow flag bit.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

9.3 Configuration of Watch Timer

This section explains the registers and interrupt factors of the watch timer.

■ List of Registers and Reset Values of Watch Timer

Figure 9.3-1 List of Registers and Reset Values of Watch Timer

	bit	7	6	5	4	3	2	1	0
16-bit reload register (Low) (TMRLR1)		1	X	0	0	0	0	0	0

X: Undefined

■ Generation of Interrupt Request from Watch Timer

- When the interval time set by the interval time select bits (WTC: WTC2 to WTC0) is reached, the overflow flag bit (WTC: WTOF) is set to 1.
- When the overflow flag bit is set (WTC: WTOF = 1) with interrupt enabled when the watch timer counter overflows (carries) (WTC: WTIE = 1), an interrupt request is generated.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSh4U.com

www.DataSheet4U.com

9.3.1 Watch Timer Control Register (WTC)

This section explains the functions of the watch timer control register (WTC).

■ Watch Timer Control Register (WTC)

Figure 9.3-2 Watch Timer Control Register (WTC)

7	6	5	4	3	2	1	0
R/W	R	R/W	R/W	R/W	R/W	R/W	R/W

Reset value
1X001000B

bit 2 bit 1 bit 0

WTC2	WTC1	WTC0	Interval time select bits
0	0	0	2^8/SCLK (31.25ms)
0	0	1	2^9/SCLK (62.5ms)
0	1	0	2^{10}/SCLK (125ms)
0	1	1	2^{11}/SCLK (250ms)
1	0	0	2^{12}/SCLK (500ms)
1	0	1	2^{13}/SCLK (1.0s)
1	1	0	2^{14}/SCLK (2.0s)
1	1	1	2^{15}/SCLK (4.0s)

bit 3

WTR	Watch timer clear bit	
	Read	Write
0	—	Clears watch timer counter
1	"1" always read	No effect

bit 4

WTOF	Overflow flag bit	
	Read	Write
0	No overflow of the bit corresponding to set interval time	Clears WTOF bit
1	Overflow of the bit corresponding to set interval time	No effect

bit 5

WTIE	Overflow interrupt enable bit
0	Interrupt request disable
1	Interrupt request enable

bit 6

SCE	Oscillation stabilization wait time end bit
0	Oscillation stabilization wait state
1	Oscillation stabilization wait time end

bit 7

WDCS	Watchdog clock select bit (input clock of watchdog timer)	
	Main or PLL clock mode	Subclock mode
0	Watch timer	Set "0"
1	Timebase timer	

R/W : Read/Write
R : Read only
X : Undefined
SCLK : Subclock
□ : Reset value
The parenthesized values are provided when subclock operates at 8.192 kHz.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 9.3-1 Functions of Watch Timer Control Register (WTC)

Bit Name		Function
bit 2 to bit 0	WTC2, WTC1, WTC0: Interval time select bits	These bits set the interval time of the watch timer. • When the interval time set by the WTC2 to WTC0 bits is reached, the corresponding bit of the watch timer counter overflows (carries) and the overflow flag bit is set (WTC: WTOF = 1). • To set the WTC2 to WTC0 bits, set the WTOF bit to 0.
bit 3	WTR: Watch timer clear bit	This bit clears the watch timer counter. **When set to 0:** Clears watch timer counter to "0000_H" **When set to 1:** No effect **Read:** 1 is always read.
bit 4	WTOF: Overflow flag bit	This bit is set to 1 when the counter value of the watch timer reaches the value set by the interval time select bit. When an overflow (carries) occurs (WTOF = 1) with interrupt request enabled (WTIE = 1), an interrupt request is generated. **When set to 0:** Clears this bit **When set to 1:** No effect • The overflow flag bit is set to 1 when the bit of the watch timer counter corresponding to the interval time set by the interval time select bits (WTC2 to WTC0) overflows (carries).
bit 5	WTIE: Overflow interrupt enable bit	This bit enables or disables generation of an interrupt request when the watch timer counter overflows (carries). **When set to 0:** Interrupt request not generated even at overflow (WTOF = 1) **When set to 1:** Interrupt request generated at overflow (WTOF = 1)
bit 6	SCE: Oscillation stabilization wait time end bit	This bit indicates that the oscillation stabilization wait time of the subclock ends. **When cleared to 0:** Subclock in oscillation stabilization wait state **When set to 1:** Subclock oscillation stabilization wait time ends • The oscillation stabilization wait time of the subclock is fixed at 2^{14}/SCLK (SCLK: subclock frequency).
bit 7	WDCS: Watchdog clock select bit	This bit selects the operation clock of the watchdog timer. **<Main clock mode or PLL clock mode>** **When set to 0:** Selects output of watch timer as operation clock of watchdog timer. **When set to 1:** Selects output of timebase timer as operation clock of watchdog timer. **<Subclock mode>** Always set this bit to 0 to select the output of the watch timer. **Note:** The watch timer and the timebase timer operate asynchronously. When the WDCS bit is changed from 0 to 1, the watchdog timer may run fast. The watchdog timer must be cleared before and after changing the WDCS bit.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

9.4 Watch Timer Interrupt

When the interval time is reached with the watch timer interrupt enabled, the overflow flag bit is set to 1 and an interrupt request is generated.

■ Watch Timer Interrupt

Table 9.4-1 shows the interrupt control bits and interrupt factors of the watch timer.

Table 9.4-1 Interrupt Control Bits of Watch Timer

	Watch Timer
Interrupt factor	Interval time of watch timer counter
Interrupt request flag bit	WTC: WTOF (overflow flag bit)
Interrupt factor enable bit	WTC: WTIE

- When the value set by the interval time select bits (WTC2 to WTC0) in the watch timer control register (WTC) is reached, the overflow flag bit in the WTC register is set to 1 (WTC: WTOF = 1).
- When the overflow flag bit is set (WTC: WTOF = 1) with the watch timer interrupt enabled (WTC: WTIE = 1), an interrupt request is generated.
- At interrupt processing, set the WTOF bit to 0 and cancel the interrupt request.

et4U.com DataShee DataSheet4U.com

■ Watch Timer Interrupt and EI^{2}OS Function

- The watch timer does not correspond to the EI^{2}OS function.
- For details of the interrupt number, interrupt control register, and interrupt vector address, see Section "3.5 Interrupt".

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

9.5 Explanation of Operation of Watch Timer

The watch timer operates as an interval timer or an oscillation stabilization wait time timer of subclock. It also supplies an operation clock to the watchdog timer.

■ Watch Timer Counter

The watch timer counter continues incrementing in synchronization with the subclock (SCLK) while the subclock (SCLK) is operating.

● Clearing watch timer counter

The watch timer counter is cleared to "0000_H" when:

- A power-on reset occurs.
- The mode transits to the stop mode.
- The watch timer clear bit (WTR) in the watch timer control register (WTC) is set to 0.

Note: When the watch timer counter is cleared, the interrupts of the watchdog timer and interval timer that use the output of the watch timer counter are affected.
To clear the watch timer by writing zero to the watch timer clear bit (WTR) in the watch timer control register (WTC), set the overflow interrupt enable bit (WTIE) to "0" and set the watch timer to interrupt inhibited state. Before permitting an interrupt, clear the interrupt request issued by writing zero to the overflow flag bit (WTOF) in the WTC register.

■ Interval Timer Function

The watch timer can be used as an interval timer by generating an interrupt at each interval time.

● Settings when using watch timer as interval timer

Operating the watch timer as an interval timer requires the settings shown in Figure 9.5-1.

Figure 9.5-1 Setting of Watch Timer

	bit7	6	5	4	3	2	1	bit0
WTC	WDCS	SCE	WTIE	WTOF	WTR	WTC2	WTC1	WTC0
	X	X	○	○	○	○	○	○

○: Used bit
X: Undefined

- When the value set by the interval time select bits (WTC1, WTC0) in the watch timer control register (WTC) is reached, the overflow flag bit in the WTC register is set to 1 (WTC: WTOF = 1).
- When the overflow flag bit is set (WTC: WTOF = 1) with the overflow interrupt of the watch timer counter enabled (WTC: WTIE = 1), an interrupt request is generated.
- The overflow flag bit (WTC: WTOF) is set when the interval time is reached at the starting point of the timing at which the watch timer is finally cleared.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Clearing overflow flag bit (WTC: WTOF)

When the mode is switched to the stop mode, the watch timer is used as an oscillation stabilization wait time timer of subclock. The WTOF bit is cleared concurrently with mode switching.

■ Setting Operation Clock of Watchdog Timer

The watchdog clock select bit (WDCS) in the watch timer control register (WTC) can be used to set the clock input source of the watchdog timer.

When using the subclock as the machine clock, always set the WDCS bit to 0 and select the output of the watch timer.

■ Oscillation Stabilization Wait Time Timer of Subclock

When the watch timer returns from the power-on reset and the stop mode, it functions as an oscillation stabilization wait time timer of subclock.

- The subclock oscillation stabilization wait time is fixed at 2^{14}/SCLK (SCLK: subclock frequency).

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

9.6 Program Example of Watch Timer

This section gives a program example of the watch timer.

■ Program Example of Watch Timer

● Processing specifications

An interval interrupt at 2^{13}/SCLK (SCLK: subclock) is generated repeatedly. The interval time is approximately 1.0s (when subclock operates at 8.192 kHz).

● Coding example

```
ICR07   EQU  0000B7H           ; Interrupt control register
WTC     EQU  0000AAH           ; Watch timer control register
WTOF    EQU  WTC:4             ; Overflow flag bit
;
;-----Main program-----------------------------------------------------------
CODE    CSEG
START:
;        :                     ; Stack pointer (SP) already initialized
        AND  CCR,#0BFH         ; Interrupt disabled
        MOV  I:ICR07,#00H      ; Interrupt level 0 (highest)
        MOV  I:WTC, #10100101B ; Interrupt enabled
                               ; Overflow flag bit cleared
                               ; Watch timer counter cleared
                               ; 2^13/SCLK (approx. 1.0 s)
        MOV  ILM,#07H          ; ILM in PS set to level 7
        OR   CCR,#40H          ; Interrupt enabled
LOOP:
         .
        Processing by user
         .
        BRA   LOOP
;-----Interrupt program-------------------------------------------------------
WARI:
        CLRB I:WTOF            ; Overflow flag cleared
         .
        Processing by user
         .
        RETI                   ; Return from interrupt processing
CODE    ENDS
;-----Vector setting----------------------------------------------------------
VECT    CSEG ABS=0FFH
        ORG  00FF8CH           ; Vector set to interrupt #28 (1CH)
        DSL  WARI
        ORG  00FFDCH           ; Reset vector set
        DSL  START
        DB   00H               ; Set to single-chip mode
VECT    ENDS
        END  START
```

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 10

8-/16-BIT PPG TIMER

This section describes the functions and operations of the 8-/16-bit PPG timer.

10.1 Overview of 8-/16-bit PPG Timer
10.2 Block Diagram of 8-/16-bit PPG Timer
10.3 Configuration of 8-/16-bit PPG Timer
10.4 Interrupts of 8-/16-bit PPG Timer
10.5 Explanation of Operation of 8-/16-bit PPG Timer
10.6 Precautions when Using 8-/16-bit PPG Timer

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

10.1 Overview of 8-/16-bit PPG Timer

The 8-/16-bit PPG timer is a reload timer module with two channels (PPG0 and PPG1) that outputs a pulse in any cycle and at any duty ratio. A combination of two channels provides:

- **8-bit PPG output 2-channel independent operation mode**
- **16-bit PPG output operation mode**
- **8 + 8-bit PPG output operation mode**

The MB90385 series has two 8-/16-bit PPG timers. This section explains the functions of PPG0/1. PPG2/3 has the same functions as PPG0/1.

■ Functions of 8-/16-bit PPG Timer

The 8-/16-bit PPG timer consists of four 8-bit reload registers (PRLH0, PRLL0, PRLH1, and PRLL1) and two PPG down counters (PCNT0 and PCNT1).

- Individual setting of High and Low widths in output pulse enables an output pulse of any cycle and duty ratio.
- The count clock can be selected from six internal clocks.
- The 8-/16-bit PPG timer can be used as an interval timer by generating an interrupt request at each interval time.
- An external circuit enables the 8-/16-bit PPG timer to be used as a D/A converter.

■ Operation Modes of 8-/16-bit PPG Timer

● 8-bit PPG output 2-channel independent operation mode

The 8-bit PPG output 2-channel independent operation mode causes the 2-channel modules (PPG0 and PPG1) to operate as each independent 8-bit PPG timer.

Table 10.1-1 shows the interval times in the 8-bit PPG output 2-channel independent operation mode.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 10.1-1 Interval Times in 8-bit PPG Output 2-channel Independent Operation Mode

Count Clock Cycle	PPG0, PPG1	
	Interval Time	Output Pulse Time
1/ϕ (62.5 ns)	1/ϕ to 2^8/ϕ	2/ϕ to 2^9/ϕ
2/ϕ (125 ns)	2/ϕ to 2^9/ϕ	2^2/ϕ to 2^{10}/ϕ
2^2/ϕ (250 ns)	2^2/ϕ to 2^{10}/ϕ	2^3/ϕ to 2^{11}/ϕ
2^3/ϕ (500 ns)	2^3/ϕ to 2^{11}/ϕ	2^4/ϕ to 2^{12}/ϕ
2^4/ϕ (1 µs)	2^4/ϕ to 2^{12}/ϕ	2^5/ϕ to 2^{13}/ϕ
2^9/HCLK (128 µs)	2^9/HCLK to 2^{17}/HCLK	2^{10}/HCLK to 2^{18}/HCLK

HCLK: Oscillation clock
ϕ: Machine clock frequency
The parenthesized values are provided when the oscillation clock operates at 4 MHz and the machine clock operates at 16 MHz.

● 16-bit PPG output operation mode

The 16-bit PPG output operation mode concatenates the 2-channel modules (PPG0 and PPG1) to operate as a 16-bit 1-channel PPG timer.

DataSheet4U.com

Table 10.1-2 shows the interval times in this mode.

Table 10.1-2 Interval Times in 16-bit PPG Output Operation Mode

Count clock cycle	Interval time	Output pulse time
1/ϕ (62.5 ns)	1/ϕ to 2^{16}/ ϕ	2/ϕ to 2^{17}/ϕ
2/ϕ (125 ns)	2/ϕ to 2^{17}/ϕ	2^2/ϕ to 2^{18}/ϕ
2^2/ϕ (250 ns)	2^2/ϕ to 2^{18}/ϕ	2^3/ϕ to 2^{19}/ϕ
2^3/ϕ (500 ns)	2^3/ϕ to 2^{19}/ϕ	2^4/ϕ to 2^{20}/ϕ
2^4/ϕ (1 µs)	2^4/ϕ to 2^{20}/ϕ	2^5/ϕ to 2^{21}/ϕ
2^9/HCLK (128 µs)	2^9/HCLK to 2^{25}/HCLK	2^{10}/HCLK to 2^{26}/HCLK

HCLK: Oscillation clock
ϕ: Machine clock frequency
The parenthesized values are provided when the oscillation clock operates at 4 MHz and the machine clock operates at 16 MHz.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● 8 + 8-bit PPG output operation mode

The 8 + 8-bit PPG output operation mode causes the PPG0 of the 2-channel modules (PPG0 and PPG1) to operate as an 8-bit prescaler and the underflow output of the PPG0 to operate as the count clock of the PPG1.

Table 10.1-3 shows the interval times in this mode.

Table 10.1-3 Interval Times in 8+8-bit PPG Output Operation Mode

Count Clock Cycle	PPG0		PPG1	
	Interval Time	Output Pulse Time	Interval Time	Output Pulse Time
1/ϕ (62.5 ns)	1/ϕ to 2^8/ϕ	2/ϕ to 2^9/ϕ	1/ϕ to 2^{16}/ϕ	2/ϕ to 2^{17}/ϕ
2/ϕ (125 ns)	2/ϕ to 2^9/ϕ	2^2/ϕ to 2^{10}/ϕ	2/ϕ to 2^{17}/ϕ	2^2/ϕ to 2^{18}/ϕ
2^2/ϕ (250 ns)	2^2/ϕ to 2^{10}/ϕ	2^3/ϕ to 2^{11}/ϕ	2^2/ϕ to 2^{18}/ϕ	2^3/ϕ to 2^{19}/ϕ
2^3/ϕ (500 ns)	2^3/ϕ to 2^{11}/ϕ	2^4/ϕ to 2^{12}/ϕ	2^3/ϕ to 2^{19}/ϕ	2^4/ϕ to 2^{20}/ϕ
2^4/ϕ (1 µs)	2^4/ϕ to 2^{12}/ϕ	2^5/ϕ to 2^{13}/ϕ	2^4/ϕ to 2^{20}/ϕ	2^5/ϕ to 2^{21}/ϕ
2^9/HCLK (128 µs)	2^9/HCLK to 2^{17}/HCLK	2^{10}/HCLK to 2^{18}/HCLK	2^9/HCLK to 2^{25}/HCLK	2^{10}/HCLK to 2^{26}/HCLK

HCLK: Oscillation clock
ϕ: Machine clock frequency
The parenthesized values are provided when the oscillation clock operates at 4 MHz and the machine clock operates at 16 MHz.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

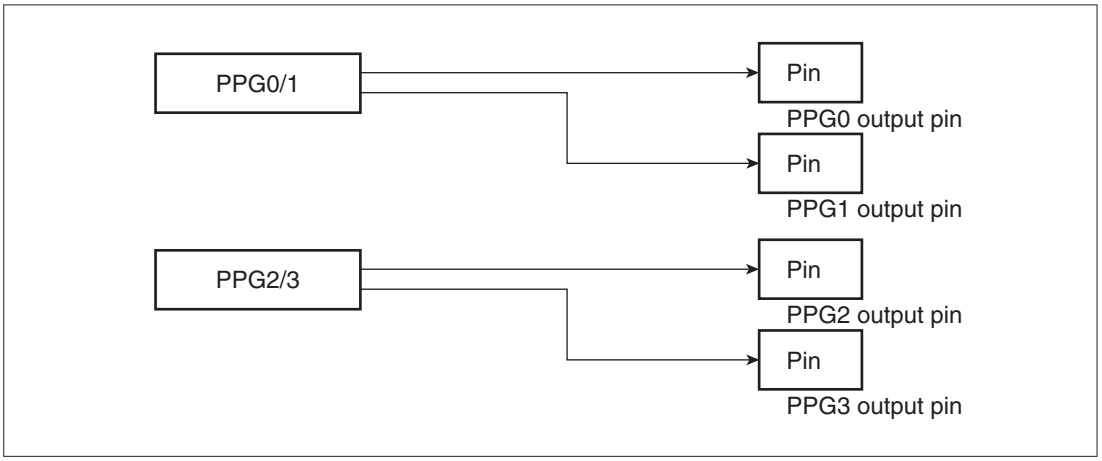
10.2 Block Diagram of 8-/16-bit PPG Timer

The MB90385 series contains two 8/16-bit PPG timer (each with two channels).
One 8-/16-bit PPG timer consists of 8-bit PPG timers with two channels.
This section shows the block diagrams for the 8-/16-bit PPG timer 0 and 8-/16-bit PPG timer 1.
The PPG2 has the same function as the PPG0, and PPG3 has the same function as PPG1.

■ Channels and PPG Pins of PPG Timers

Figure 10.2-1 shows the relationship between the channels and the PPG pins of the 8-/16-bit PPG timers in the MB90385 series.

Figure 10.2-1 Channels and PPG Pins of PPG Timers



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

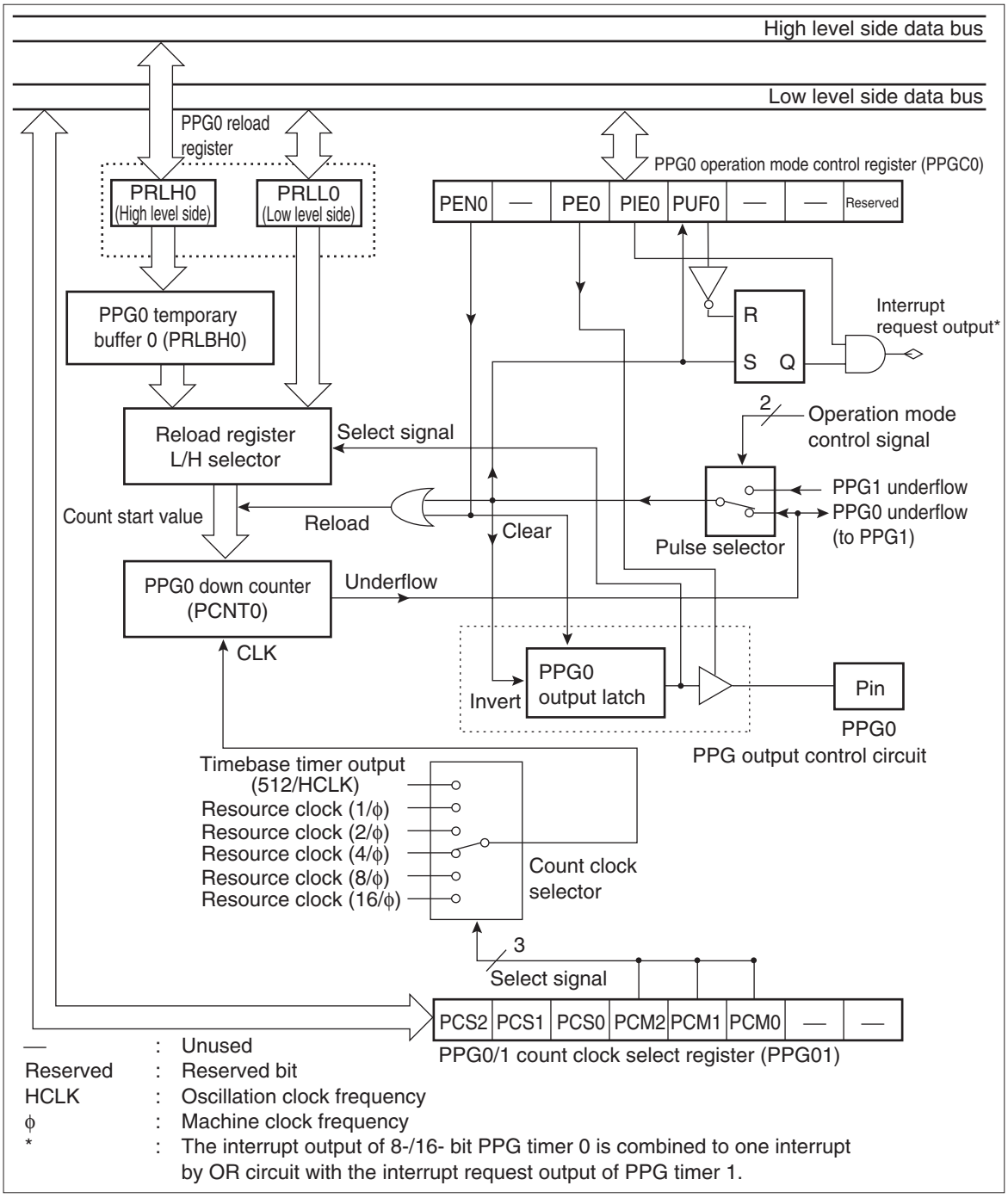
www.DataSheet4U.com

10.2.1 Block Diagram for 8-/16-bit PPG Timer 0

The 8-/16-bit PPG timer 0 consists of the following blocks.

■ Block Diagram of 8-/16-bit PPG Timer 0

Figure 10.2-2 Block Diagram of 8-/16-bit PPG Timer 0



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Details of pins in block diagram

Table 10.2-1 lists the actual pin names and interrupt request numbers of the 8-/16-bit PPG timer.

Table 10.2-1 Pins and Interrupt Request Numbers in Block Diagram

Channel	Output Pin	Interrupt Request Number
PPG0	P14/PPG0	#22 (16_H)
PPG1	P15/PPG1	
PPG2	P16/PPG2	#26 ($1A_H$)
PPG3	P17/PPG3	

- PPG operation mode control register 0 (PPGC0)

This register enables or disables operation of the 8-/16-bit PPG timer 0, the pin output, and an underflow interrupt. It also indicates the occurrence of an underflow.

- PPG0/1 count clock select register (PPG01)

This register sets the count clock of the 8-/16-bit PPG timer 0.

- PPG0 reload registers (PRLH0 and PRLL0)

These registers set the High width or Low width of the output pulse. The values set in these registers are reloaded to the PPG0 down counter (PCNT0) when the 8-/16-bit PPG timer 0 is started.

- PPG0 down counter (PCNT0)

This counter is an 8-bit down counter that alternately reloads the values set in the PPG0 reload registers (PRLH0 and PRLL0) to decrement. When an underflow occurs, the pin output is inverted. This counter is concatenated for use as a single-channel 16-bit PPG down counter.

- PPG0 temporary buffer (PRLBH0)

This buffer prevents deviation of the output pulse width caused at writing to the PPG reload registers (PRLH0 and PRLL0). This buffer stores the PRLH0 value temporarily and enables it in synchronization with the timing of writing to the PRLL0.

- Reload register L/H selector

This selector detects the current pin output level to select which register value, Low reload register (PRLL0) or High reload register (PRLH0), should be reloaded to the PPG0 down counter.

- Count clock selector

This selector selects the count clock to be input to the PPG0 down counter from five frequency-divided clocks of the machine clock or the frequency-divided clocks of the timebase timer.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- PPG output control circuit

  This circuit inverts the pin output level and the output when an underflow occurs.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

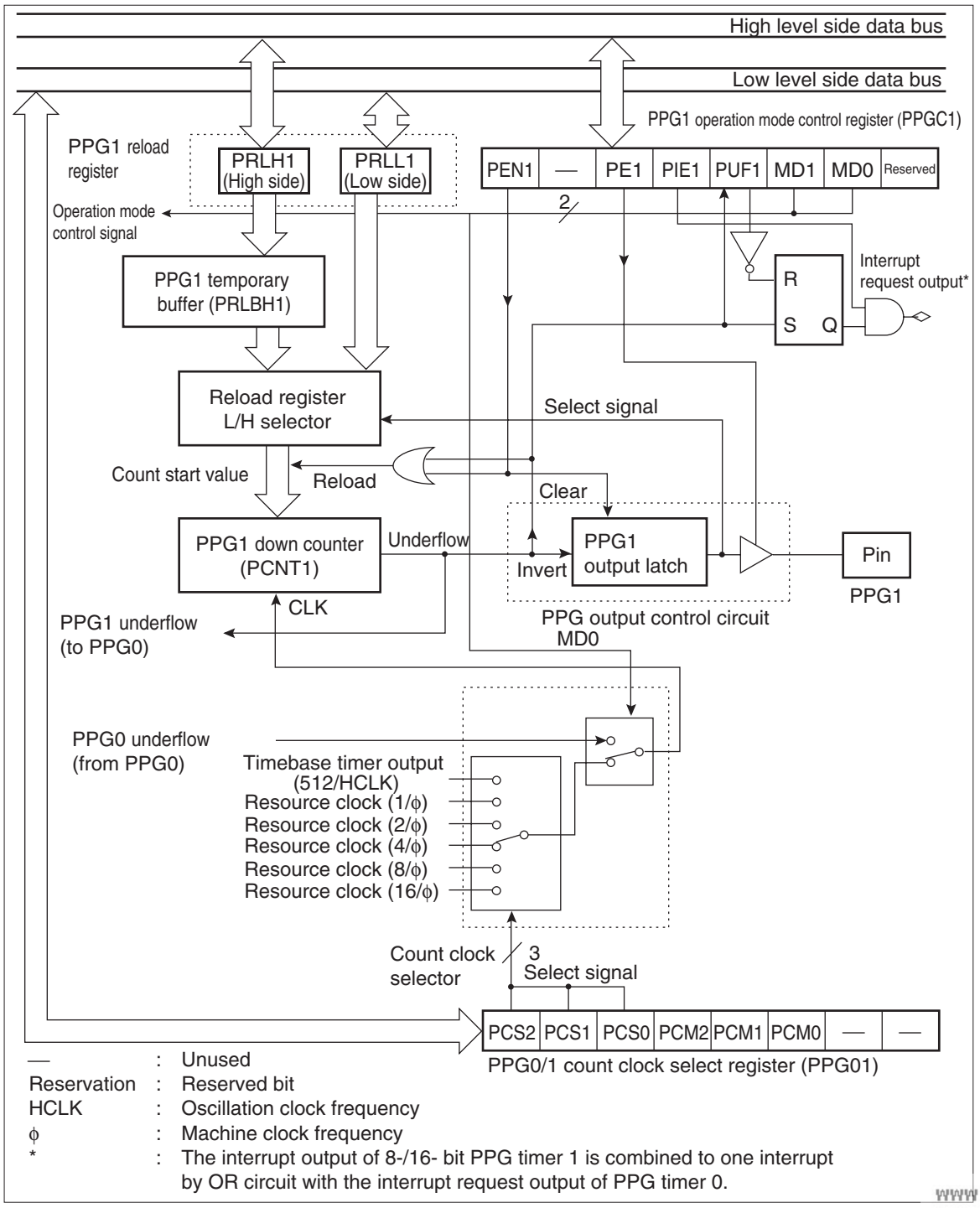
www.DataSheet4U.com

10.2.2 Block Diagram of 8-/16-bit PPG Timer 1

The 8-/16-bit PPG timer 1 consists of the following blocks.

■ Block Diagram of 8-/16-bit PPG Timer 1

Figure 10.2-3 Block Diagram of 8-/16-bit PPG Timer 1



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Details of pins in block diagram

Table 10.2-2 lists the actual pin names and interrupt request numbers of the 8-/16-bit PPG timer.

Table 10.2-2 Pins and Interrupt Request Numbers in Block Diagram

Channel	Output Pin	Interrupt Request Number
PPG0	P14/PPG0	#22 (16_H)
PPG1	P15/PPG1	
PPG2	P16/PPG2	#26 ($1A_H$)
PPG3	P17/PPG3	

- PPG operation mode control register 1 (PPGC1)

This register sets the operation mode of the 8-/16-bit PPG timer, enables or disables the operation of the 8-/16-bit PPG timer 1, the pin output and an underflow interrupt, and also indicates the generation of an underflow.

- PPG2/3 count clock select register (PPG23)

This register sets the count clock of the 8-/16-bit PPG timer 1.

et4U.com DataShee

- PPG1 reload registers (PRLH1 and PRLL1)

DataSheet4U.com

These registers set the High width or Low width of the output pulse. The values set in these registers are reloaded to the PPG1 down counter (PCNT1) when the 8-/16-bit PPG timer 1 is started.

- PPG1 down counter (PCNT1)

This counter is an 8-bit down counter that alternately reloads the values set in the PPG1 reload registers (PRLH1 and PRLL1) to decrement. When an underflow occurs, the pin output is inverted. The 2-channel PPG down counters (PPG0 and PPG1) can also be connected for use as a single-channel 16-bit PPG down counter.

- PPG1 temporary buffer (PRLBH1)

This buffer prevents deviation of the output pulse width caused at writing to the PPG reload registers (PRLH1 and PRLL1). It stores the PRLH1 value temporarily and enables it in synchronization with the timing of writing to the PRLL1.

- Reload register L/H selector

This selector detects the current pin output level to select which register value, Low reload register (PRLL1) or High reload register (PRLH1), should be reloaded to the PPG1 down counter.

- Count clock selector

This selector selects the count clock to be input to the PPG1 down counter from five frequency-divided clocks of the machine clock or the frequency-divided clocks of the timebase timer.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- PPG output control circuit

  This circuit inverts the pin output level and the output when an underflow occurs.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

10.3 Configuration of 8-/16-bit PPG Timer

This section explains the pins, registers and interrupt factors of the 8-/16-bit PPG timer.

■ Pins of 8-/16-bit PPG Timer

The pins of the 8-/16-bit PPG timer serve as general-purpose I/O ports. Table 10.3-1 indicates the pin functions and pin settings required to use the 8-/16-bit PPG timer.

Table 10.3-1 Pins of 8-/16-bit PPG Timer

Channel	Pin Name	Pin Function	Pin Setting Required for Use of 8-/16-bit PPG Timer
PPG0	PPG0 output pin	General-purpose I/O port, PPG0 output pin	Set PPG0 pin output to "enabled" (PPGC0: PE=1)
PPG1	PPG1 output pin	General-purpose I/O port, PPG1 output pin	Set PPG1 pin output to "enabled" (PPGC1: PE1=1)
PPG2	PPG2 output pin	General-purpose I/O port, PPG2 output pin	Set PPG2 pin output to "enabled" (PPGC2: PE0=1)
PPG3	PPG3 output pin	General-purpose I/O port, PPG3 output pin	Set PPG3 pin output to "enabled" (PPGC3: PE1=1)

et4U.com DataShee DataSheet4U.com

■ Block Diagram of 8-/16-bit PPG Timer Pins

See "CHAPTER 4 I/O PORT" for the pin block diagram.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ List of Registers and Reset Values of 8-/16-bit PPG Timer

Figure 10.3-1 List of Registers and Reset Values of 8-/16-bit PPG Timer

	bit	15	14	13	12	11	10	9	8
PPG0 operation mode control register: H (PPGC1)		0	0	0	0	0	0	0	0
	bit	7	6	5	4	3	2	1	0
PPG0 OPERATION MODE CONTROL REGISTER: L (PPGC0)		0	0	0	0	0	0	0	0
	bit	7	6	5	4	3	2	1	0
PPG0/1 COUNT CLOCK SELECT REGISTER (PPG01)		0	0	0	0	0	0	0	0
	bit	15	14	13	12	11	10	9	8
PPG0 reload register: H (PRLH0)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
PPG0 reload register: L (PRLL0)		X	X	X	X	X	X	X	X
	bit	15	14	13	12	11	10	9	8
PPG1 reload register: H (PRLH1)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
PPG1 reload register: L (PRLL1)		X	X	X	X	X	X	X	X

X: Undefined

DataSheet4U.com

■ Generation of Interrupt Request from 8-/16-bit PPG Timer

In the 8-/16-bit PPG timer, the underflow generation flag bits in the PPG operation mode control registers (PPGC0: PUF0, PPGC1: PUF1) are set to 1 when an underflow occurs. If the underflow interrupts of channels causing an underflow are enabled (PPGC0: PIE0=1, PPGC1: PIE1=1), an underflow interrupt request is generated to the interrupt controller.

DataSheet4U.com
www.DataSheet4U.com

DataSh4U.com

www.DataSheet4U.com

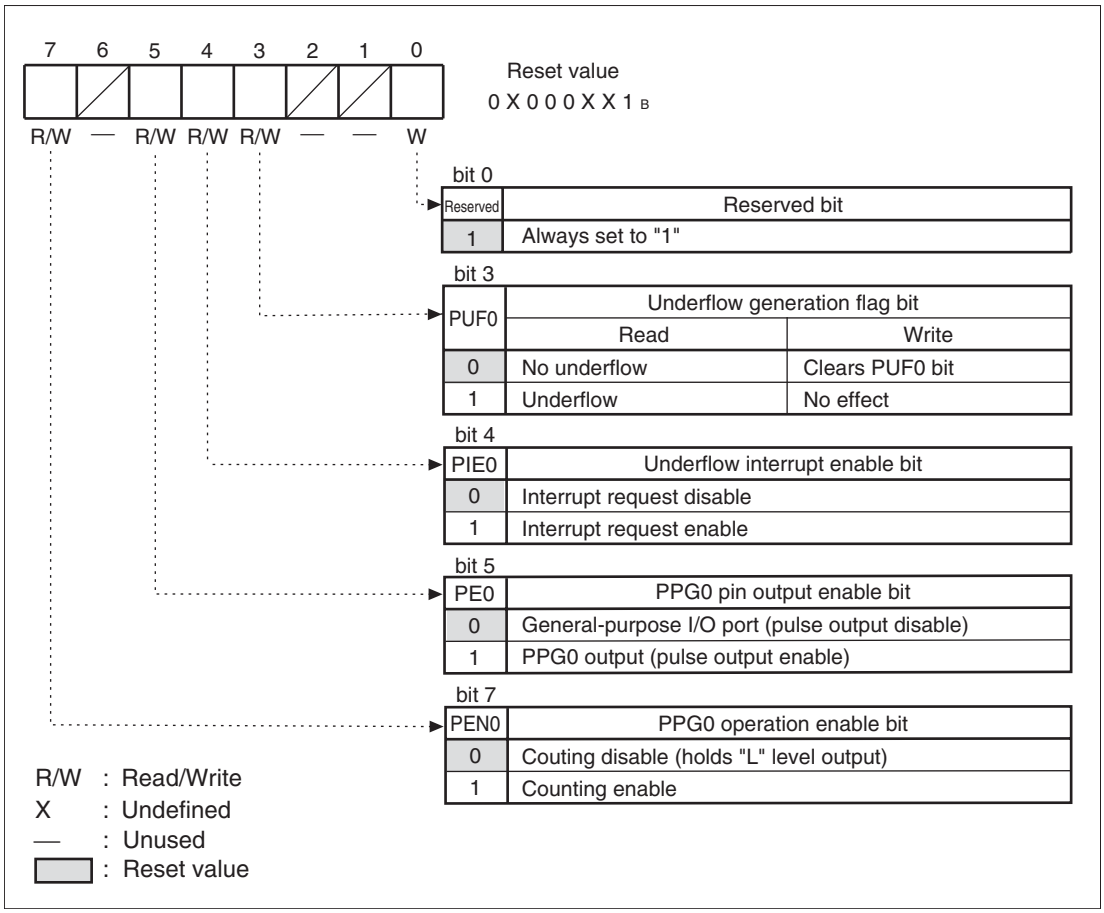
10.3.1 PPG0 Operation Mode Control Register (PPGC0)

The PPG0 operation mode control register (PPGC0) provides the following settings:

- **Enabling or disabling operation of 8-/16-bit PPG timer**
- **Switching between pin functions (enabling or disabling pulse output)**
- **Enabling or disabling underflow interrupt**
- **Setting underflow interrupt request flag**

■ PPG0 Operation Mode Control Register (PPGC0)

Figure 10.3-2 PPG0 Operation Mode Control Register (PPGC0)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 10.3-2 Functions of PPG0 Operation Mode Control Register (PPGC0)

Bit Name		Function
bit 0	Reserved: Reserved bit	Always set this bit to 1.
bit 1 bit 2	Unused bits	**Read:** The value is undefined. **Write:** No effect
bit 3	PUF0: Underflow generation flag bit	**8-bit PPG output 2-channel independent operation mode, 8+8-bit PPG output operation mode:** When the value of the PPG0 down counter is decremented from "00_H" to "FF_H", an underflow occurs (PUF0 = 1). **16-bit PPG output operation mode:** When the values of the PPG0 and PPG1 down counters are decremented from "0000_H" to" $FFFF_H$", an underflow occurs (PUF0 = 1). • When an underflow occurs (PUF0 = 1) with an underflow interrupt enabled (PIE0 = 1), an interrupt request is generated. **When set to 0:** Clears this bit **When set to 1:** No effect **Read by read modify write instructions:** 1 read
bit 4	PIE0: Underflow interrupt enable bit	This bit enables or disables an interrupt. **When set to 0:** No interrupt request generated even at underflow (PUF0 = 1). **When set to 1:** Interrupt request generated at underflow (PUF0 = 1)
bit 5	PE0: PPG0 pin output enable bit	This bit switches between PPG0 pin functions and enables or disables the pulse output. **When set to 0:** PPG0 pin functions as general-purpose I/O port. The pulse output is disabled. **When set to 1:** PPG0 pin functions as PPG0 output pin. The pulse output is enabled.
bit 6	Unused bit	**Read:** The value is undefined. **Write:** No effect
bit 7	PEN0: PPG0 operation enable bit	This bit enables or disables the count operation of the 8-/16-bit PPG timer 0. **When set to 0:** Count operation disabled **When set to 1:** Count operation enabled • When the count operation is disabled (PEN0 = 0), the output is held at a Low level.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

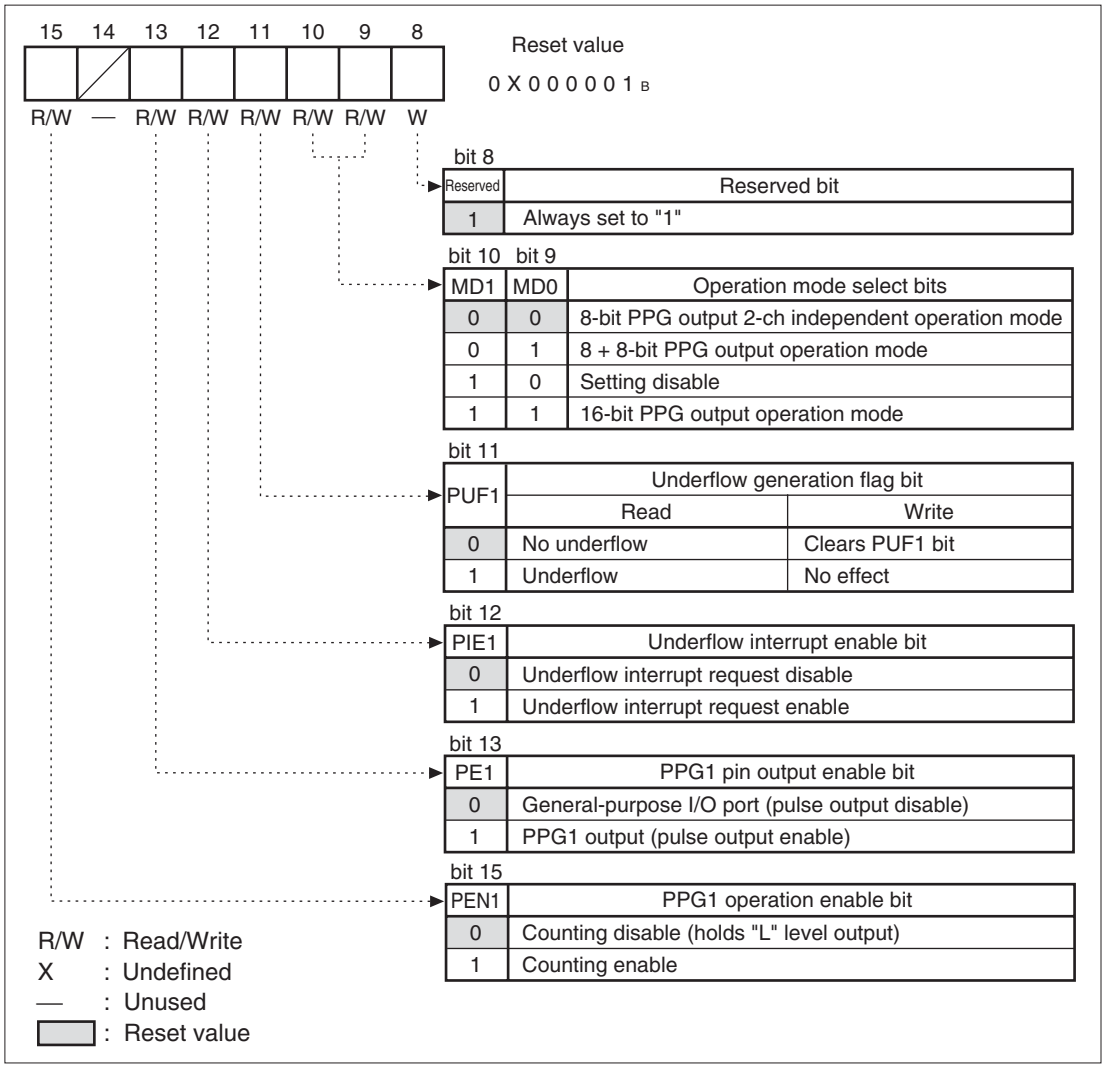
10.3.2 PPG1 Operation Mode Control Register (PPGC1)

The PPG1 operation mode control register (PPGC1) provides the following settings:

- **Enabling or disabling operation of 8-/16-bit PPG timer**
- **Switching between pin functions (enabling or disabling pulse output)**
- **Enabling or disabling underflow interrupt**
- **Setting underflow interrupt request flag**
- **Setting the operation mode of the 8-/16-bit PPG timer**

■ PPG1 Operation Mode Control Register (PPGC1)

Figure 10.3-3 PPG1 Operation Mode Control Register (PPGC1)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 10.3-3 Functions of PPG1 Operation Mode Control Register (PPGC1)

Bit Name		Function
bit 8	Reserved: Reserved bit	Always set this bit to 1.
bit 9 bit 10	MD1, MD0: Operation mode select bits	These bits set the operation mode of the 8-/16-bit PPG timer. **[Any mode other than 8-bit PPG output 2-channel independent operation mode]** • Use a word instruction to set the PPG operation enable bits (PEN0 and PEN1) at one time. • Do not set operation of only one of the two channels (PEN1 = 0/PEN0 = 1 or PEN1 = 1/PEN0 = 0). **Note:** Do not set the MD1 and MD0 bits to "10_B".
bit 11	PUF1: Underflow generation flag bit	**8-bit PPG output 2-channel independent operation mode, 8+8-bit PPG output operation mode:** When the value of the PPG1 down counter is decremented from "00_H" to "FF_H", an underflow occurs (PUF1 = 1). **16-bit PPG output operation mode:** When the values of the PPG0 and PPG1 down counters are decremented from "0000_H" to "FFFF $_H$", an underflow occurs (PUF1 = 1). • When an underflow occurs (PUF1 = 1) with an underflow interrupt enabled (PIE1 = 1), an interrupt request is generated. **When set to 0:** Clears this bit **When set to 1:** No effect **Read by read modify write instructions:** 1 is read.
bit 12	PIE1: Underflow interrupt enable bit	This bit enables or disables an interrupt. **When set to 0:** No interrupt request is generated even at underflow (PUF1 = 1) **When set to 1:** Interrupt request is generated at underflow (PUF1 = 1)
bit 13	PE1: PPG1 Pin output enable bit	This bit switches between PPG1 pin functions and enables or disables the pulse output. **When set to 0:** PPG1 pin functions as general-purpose I/O port. The pulse output is disabled. **When set to 1:** PPG1 pin functions as PPG1 output pin. The pulse output is enabled.
bit 14	Unused bit	**Read:** The value is undefined. **Write:** No effect
bit 15	PEN1: PPG1 operation enable bit	This bit enables or disables the count operation of the 8-/16-bit PPG timer 1. **When set to 0:** Count operation disabled **When set to 1:** Count operation enabled • When the count operation is disabled (PEN1 = 0), the output is held at a Low level.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

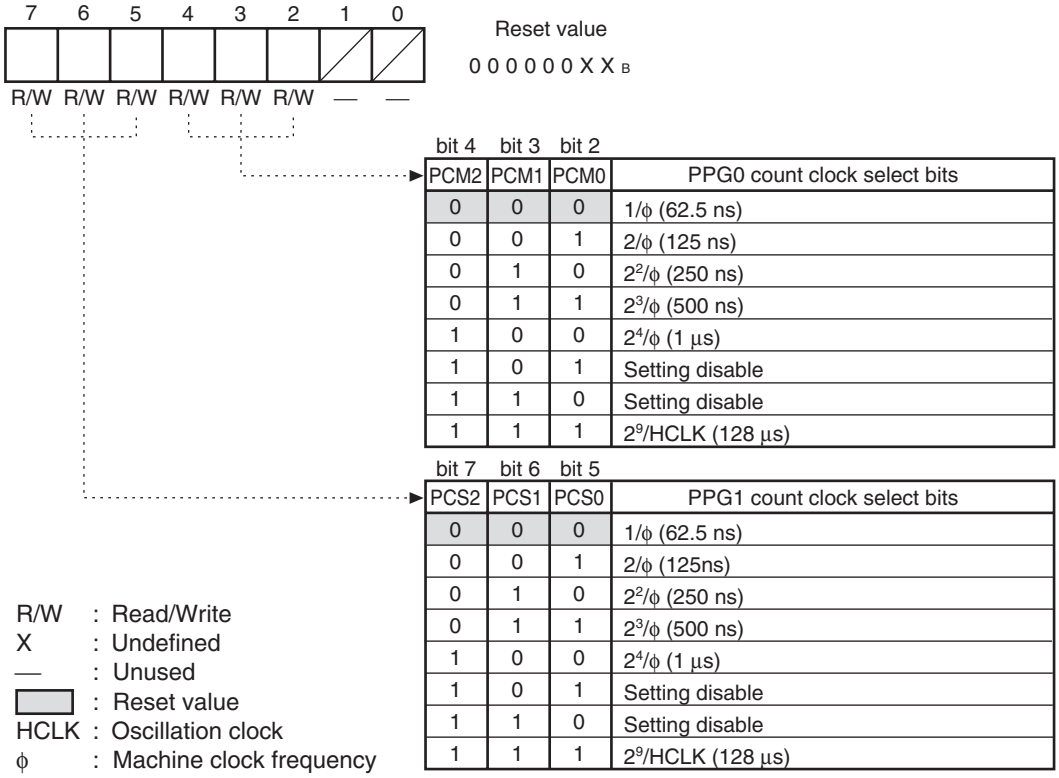
www.DataSheet4U.com

10.3.3 PPG0/1 Count Clock Select Register (PPG01)

The PPG0/1 count clock select register (PPG01) selects the count clock of the 8-/16-bit PPG timer.

■ PPG0/1 Count Clock Select Register (PPG01)

Figure 10.3-4 PPG0/1 Count Clock Select Register (PPG01)



bit 4	bit 3	bit 2	PPG0 count clock select bits
PCM2	PCM1	PCM0	
0	0	0	1/ϕ (62.5 ns)
0	0	1	2/ϕ (125 ns)
0	1	0	2^2/ϕ (250 ns)
0	1	1	2^3/ϕ (500 ns)
1	0	0	2^4/ϕ (1 µs)
1	0	1	Setting disable
1	1	0	Setting disable
1	1	1	2^9/HCLK (128 µs)

bit 7	bit 6	bit 5	PPG1 count clock select bits
PCS2	PCS1	PCS0	
0	0	0	1/ϕ (62.5 ns)
0	0	1	2/ϕ (125ns)
0	1	0	2^2/ϕ (250 ns)
0	1	1	2^3/ϕ (500 ns)
1	0	0	2^4/ϕ (1 µs)
1	0	1	Setting disable
1	1	0	Setting disable
1	1	1	2^9/HCLK (128 µs)

R/W : Read/Write
X : Undefined
— : Unused
□ : Reset value
HCLK : Oscillation clock
ϕ : Machine clock frequency

The parenthesized values are provided when the oscillation clock operates at 4 MHz and the machine clock operates at 16 MHz.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 10.3-4 Functions of PPG0/1 Count Clock Select Register (PPG01)

Bit Name		Function
bit 0 bit 1	Unused bits	**Read:** The value is undefined. **Write:** No effect
bit 2 to bit 4	PCM2 to PCM0: PPG0 count clock select bit	These bits set the count clock of the 8-/16-bit PPG timer 0. • The count clock can be selected from five frequency-divided clocks of the machine clock and the frequency-divided clocks of the timebase timer.
bit 5 to bit 7	PCS2 to PCS0: PPG1 count clock select bits	These bits set the count clock of the 8-/16-bit PPG timer 1. • The count clock can be selected from five frequency-divided clocks of the machine clock and the frequency-divided clocks of the timebase timer. • The settings of the PPG1 count clock select bits (PCS2 to PCS0) are enabled only in the 8-bit PPG output 2-channel independent mode (PPGC1: MD1, MD0 = "00_B").

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

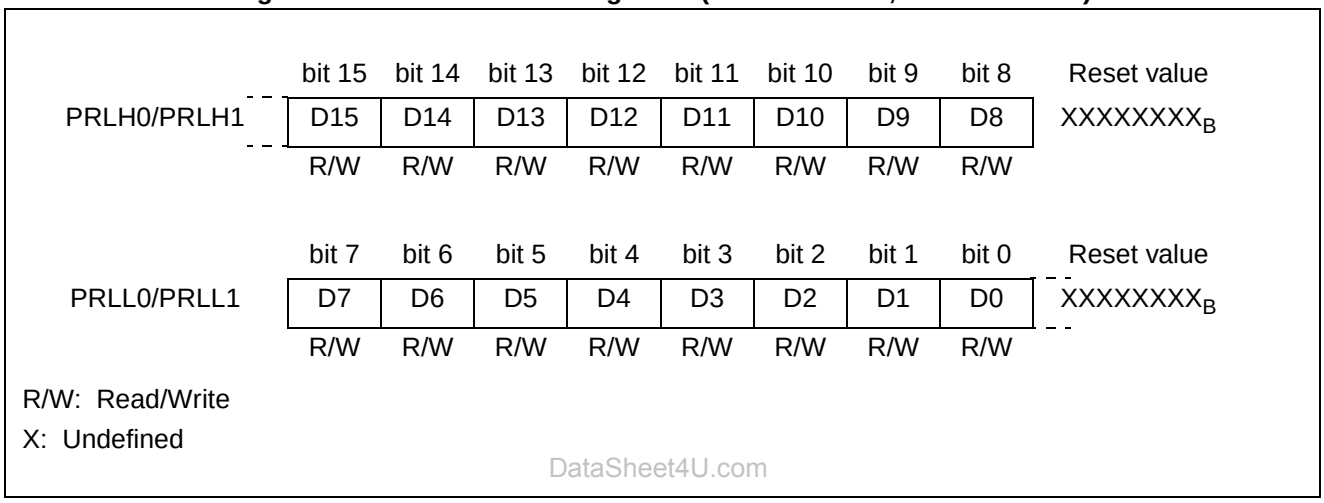
www.DataSheet4U.com

10.3.4 PPG Reload Registers (PRLL0/PRLH0, PRLL1/PRLH1)

The value (reload value) from which the PPG down counter starts counting is set in the PPG reload registers, which are an 8-bit register at Low level and an 8-bit register at High level.

■ PPG Reload Registers (PRLL0/PRLH0, PRLL1/PRLH1)

Figure 10.3-5 PPG Reload Registers (PRLL0/PRLH0, PRLL1/PRLH1)



DataSheet4U.com

Table 10.3-5 indicates the functions of the PPG reload registers.

Table 10.3-5 Functions of PPG Reload Registers

Function	8-/16-bit PPG Timer 0	8-/16-bit PPG Timer 1
Retains reload value on Low-level side	PRLL0	PRLL1
Retains reload value on High-level side	PRLH0	PRLH1

Notes:
- In the 16-bit PPG output operation mode (PPGC1: MD1, MD0 = "11_B"), use a long-word instruction to set the PPG reload registers or the word instruction to set the PPG0 and PPG1 in this order.
- In the 8 + 8-bit PPG output operation mode (PPGC1: MD1, MD0 = "01_B"), set the same value in both the Low-level and High-level PPG reload registers (PRTLL0/PRLH0) of the 8-/16-bit PPG timer 0. Setting a different value in the Low-level and High-level PPG reload registers may cause the 8-/16-bit PPG timer 1 to have different PPG output waveforms at each clock cycle.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

10.4 Interrupts of 8-/16-bit PPG Timer

The 8-/16-bit PPG timer can generate an interrupt request when the PPG down counter underflows. It corresponds to the EI^{2}OS.

■ Interrupts of 8-/16-bit PPG Timer

Table 10.4-1 shows the interrupt control bits and interrupt factor of the 8-/16-bit PPG timer.

Table 10.4-1 Interrupt Control Bits of 8-/16-bit PPG Timer

	PPG0	PPG1
Interrupt request flag bit	PPPGC0: PUF0	PPGC1: PUF1
Interrupt request enable bit	PPGC0: PIE0	PPGC1: PIE1
Interrupt factor	Underflow in PPG0 down counter	Underflow in PPG1 down counter

[8-bit PPG output 2-channel independent operation mode or 8 + 8-bit PPG output operation mode]

- In the 8-bit PPG output 2-channel independent operation mode or the 8 + 8-bit PPG output operation mode, the PPG0 and PPG1 timers can generate an interrupt independently.
- When the value of the PPG0 or PPG1 down counter is decremented from "00_H" to "FF_H", an underflow occurs. When an underflow occurs, the underflow generation flag bit in the channel causing an underflow is set (PPGC0: PUF0 = 1 or PPGC1: PUF1 = 1).
- If an interrupt request from the channel that causes an underflow is enabled (PPGC0: PIE0 = 1 or PPGC1: PIE1 = 1), an interrupt request is generated.

[16-bit PPG output operation mode]

- In the 16-bit PPG output operation mode, when the values of the PPG0 and PPG1 down counters are decremented from "0000_H" to "$FFFF_H$", an underflow occurs. When an underflow occurs, the underflow generation flag bits in the two channels are set at one time (PPGC0: PUF0 = 1 and PPGC1: PUF1 = 1).
- When an underflow occurs with either of the two channel of the interrupt requests enabled (PPGC0: PIE1 = 0, PPGC1: PIE1 = 1 or PPGC0: PIE1 = 1, PPGC1: PIE1 = 0), an interrupt request is generated.
- To prevent duplication of interrupt requests, disable either of the two channel of the underflow interrupt enable bits (PPGC0: PIE1 = 0, PPGC1: PIE1 = 1 or PPGC0: PIE1 = 1, PPGC1: PIE1 = 0).
- When the two channels of the underflow generation flag bits are set (PPGC0: PUF0 = 1 and PPGC1: PUF1 = 1), clear the two channels at the same time.

■ Correspondence between 8-/16-bit PPG Timer Interrupt and EI^{2}OS

For details of the interrupt number, interrupt control register, and interrupt vector address, see "3.5 Interrupt".

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ 8-/16-bit PPG Timer Interrupt and EI^{2}OS Function

The 8-/16-bit PPG timer corresponds to the EI^{2}OS function. Generation of an enabled interrupt factor starts the EI^{2}OS. However, it is necessary to disable generation of interrupt requests by resources sharing the interrupt control register (ICR) with the 8-/16-bit PPG timer.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

10.5 Explanation of Operation of 8-/16-bit PPG Timer

The 8-/16-bit PPG timer outputs a pulse width at any frequency and at any duty ratio continuously.

■ Operation of 8-/16-bit PPG Timer

● Output operation of 8-/16-bit PPG timer

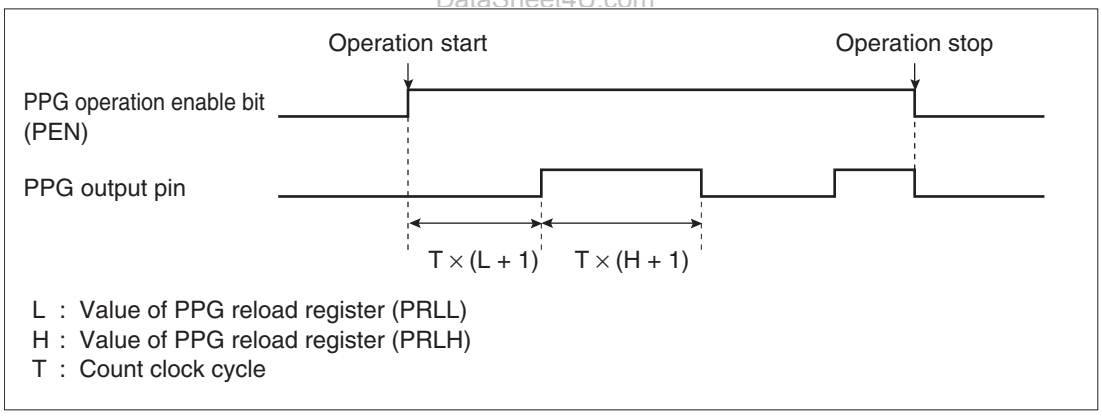
- The 8-/16-bit PPG timer has two (Low-level and High-level) 8-bit reload registers (PRLL0/PRLH0 and PRLL1/PRLH1) for each channel.
- The values set in the 8-bit reload registers (PRLL0/PRLH0 and PRLL1/PRLH1) are reloaded alternately to the PPG down counters (PCNT0 and PCNT1).
- After reloading the values in the PPG down counters, decrementing is performed in synchronization with the count clocks set by the PPG count clock select bits (PPG01: PCM2 to PCM0 and PCS1 and PCS0).
- If the values set in the reload registers are reloaded to the PPG down counters when an underflow occurs, the pin output is inverted.

Figure 10.5-1 shows the output waveform of the 8-/16-bit PPG timer.

et4U.com DataShee

Figure 10.5-1 Output Waveform of 8-/16-bit PPG Timer

DataSheet4U.com



● Operation modes of 8-/16-bit PPG timer

As long as the operation of the 8-/16-bit PPG timer is enabled (PPGC0: PEN0 = 1, PPGC1: PEN1 = 1), a pulse waveform is output continuously from the PPG output pin. A pulse width of any frequency and duty ratio can be set.

The pulse output of the 8-/16-bit PPG timer is not stopped until operation of the 8-/16-bit PPG timer is stopped (PPGC0: PEN0 = 0, PPGC1: PEN1 = 0).

- 8-bit PPG output 2-channel independent operation mode
- 16-bit PPG output operation mode
- 8 + 8-bit PPG output operation mode

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

10.5.1 8-bit PPG Output 2-channel Independent Operation Mode

In the 8-bit PPG output 2-channel independent operation mode, the 8-/16-bit PPG timer is set as an 8-bit PPG timer with two independent channels. PPG output operation and interrupt request generation can be performed independently for each channel.

■ Setting for 8-bit PPG Output 2-channel Independent Operation Mode

Operating the 8-/16-bit PPG timer in the 8-bit PPG output 2-channel independent operation mode requires the setting shown in Figure 10.5-2.

Figure 10.5-2 Setting for 8-bit PPG Output 2-channel Independent Operation Mode

	bit15	14	13	12	11	10	9	bit8	bit7	6	5	4	3	2	1	bit0
PPGC1/PPGC0	PEN1	–	PE1	PIE1	PUF1	MD1	MD0	Reserved	PEN0	–	PE0	PIE0	PUF0	–	–	Reserved
	1		◎	◎	◎	0	0	1	1		◎	◎	◎			1
PPG01	(Reserved area)								PCS2	PCS1	PCS0	PCM2	PCM1	PCM0	–	–
									◎	◎	◎	◎	◎	◎		
PRLH0/PRLL0	PPG0 Set High level side reload values.								PPG0 Set Low level side reload values.							
PRLH1/PRLL1	PPG1 Set High level side reload values.								PPG1 Set Low level side reload values.							

◎ : Used bit
–: Unused bit
1: Set 1
0: Set 0

Note: Use the word instruction to set both High-level and Low-level PPG reload registers (PRLL0/PRLH0 and PRLL1/PRLH1) at the same time.

DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Operation in 8-bit PPG output 2-channel independent operation mode

- The 8-bit PPG timer with two channels performs an independent PPG operation.
- When the pin output is enabled (PPGC0: PE0 = 1, PPGC1: PE1 = 1), the PPG0 pulse wave is output from the PPG0 pin and the PPG1 pulse wave is output from the PPG1 pin.
- When the reload value is set in the PPG reload registers (PRLL0/PRLH0 and PRLL1/PRLH1) to enable the operation of the PPG timer (PPGC0: PEN0 = 1, PPGC1: PEN1 = 1), the PPG down counter of the enabled channel starts counting.
- To stop the count operation of the PPG down counter, disable the operation of the PPG timer of the channel to be stopped (PPGC0: PEN0 = 0, PPGC1: PEN1 = 0). The count operation of the PPG down counter is stopped and the output of the PPG output pin is held at a Low level.
- When the PPG down counter of each channel underflows, the reload values set in the PPG reload registers (PRLL0/PRLH0 and PRLL1/PRLH1) are reloaded to the PPG down counter that underflows.
- When an underflow occurs, the underflow generation flag bit in the channel that causes an underflow is set (PPGC0: PUF0 = 1, PPGC1: PUF1 = 1). If an interrupt request is enabled at the channel that causes an underflow (PPGC0: PIE0 = 1, PPGC1: PIE1 = 1), the interrupt request is generated.

● Output waveform in 8-bit PPG output 2-channel independent operation mode

- The High and Low pulse widths to be output are determined by adding 1 to the value in the PPG reload register and multiplying it by the count clock cycle. For example, if the value in the PPG reload register is "00_H", the pulse width has one count clock cycle, and if the value is "FF_H", the pulse width has 256 count clock cycles.

et4U.com DataShee DataSheet4U.com

The equations for calculating the pulse width are shown below:

P_L= T x (L + 1)

P_H= T x (H + 1)

P_L: Low width of output pulse

P_H: High width of output pulse

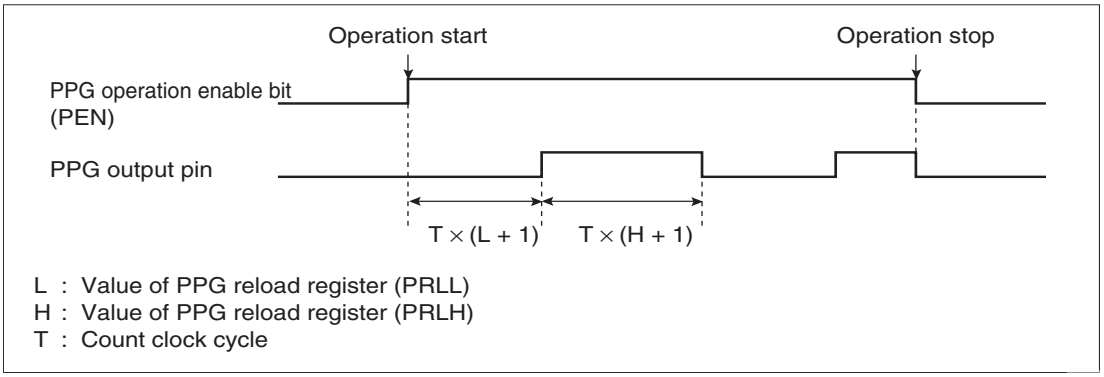
L: Values of 8 bits in PPG reload register (PRLL0 or PRLL1)

H: Values of 8 bits in PPG reload register (PRLH0 or PRLH1)

T: Count clock cycle

Figure 10.5-3 shows the output waveform in the 8-bit PPG output 2-channel independent operation mode.

Figure 10.5-3 Output Waveform in 8-bit PPG Output 2-channel Independent Operation Mode



DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

10.5.2 16-bit PPG Output Operation Mode

In the 16-bit PPG output operation mode, the 8-/16-bit PPG timer is set as a 16-bit PPG timer with one channel.

■ Setting for 16-bit PPG Output Operation Mode

Operating the 8-/16-bit PPG timer in the 16-bit PPG output operation mode requires the setting shown in Figure 10.5-4.

Figure 10.5-4 Setting for 16-bit PPG Output Operation Mode

	bit15	14	13	12	11	10	9	bit8	bit7	6	5	4	3	2	1	bit0
PPGC1/PPGC0	PEN1	–	PE1	PIE1	PUF1	MD1	MD0	Reserved	PEN0	–	PE0	PIE0	PUF0	–	–	Reserved
	1		◎	◎	◎	1	1	1	1		◎	◎	◎			1
PPG01	(Reserved area)								PCS2	PCS1	PCS0	PCM2	PCM1	PCM0	–	–
									X	X	X	◎	◎	◎		

PRLH0/PRLL0	PPG0 Set high level side reload values of lower 8 bits.	PPG0 Set low level side reload values of lower 8 bits.
PRLH1/PRLL1	PPG1 Set high level side reload values of upper 8 bits.	PPG1 Set low level side reload values of upper 8 bits.

◎: Used bit
X: Undefined bit
–: Unused bit
1: Set 1
0: Set 0

DataSheet4U.com

Note: Use a long-word instruction to set the values in the PPG reload registers or a word instruction to set the PPG0 and PPG1 (PRLL0 --> PRLL1 or PRLH0 --> PRLH1) in this order.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Operation in 16-bit PPG output operation mode

- When either PPG0 pin output or PPG1 pin output is enabled (PPGC0: PE0 = 1, PPGC1: PE1 = 1), the same pulse wave is output from both the PPG0 and PPG1 pins.
- When the reload value is set in the PPG reload registers (PRLL0/PRLH0 and PRLL1/PRLH1) to enable operation of the PPG timer (PPGC0: PEN0 = 1 and PPGC1: PEN1 = 1) simultaneously, the PPG down counters start counting as 16-bit down counters (PCNT0 + PCNT1).
- To stop the count operation of the PPG down counters, disable the operation of the PPG timers of both channels (PPGC0: PEN0 = 0 and PPGC1: PEN1 = 0) simultaneously. The count operation of the PPG down counters is stopped and the output of the PPG output pin is held at a Low level.
- If the PPG1 down counter underflows, the reload values set in the PPG0 and PPG1 reload registers (PRLL0/PRLH0 and PRLL1/PRLH1) are reloaded simultaneously to the PPG down counters (PCNT0 + PCNT1).
- When an underflow occurs, the underflow generation flag bits in both channels are set simultaneously (PPGC0: PUF0 = 1, PPGC1: PUF1 = 1). If an interrupt request is enabled at either channel (PPGC0: PIE0 = 1, PPGC1: PIE1 = 1), an interrupt request is generated.

Notes:
- In the 16-bit PPG output operation mode, the underflow generation flag bits in the two channels are set simultaneously when an underflow occurs (PPGC0: PUF0 = 1 and PPGC1: PUF1 = 1). To prevent duplication of interrupt requests, disable either of the underflow interrupt enable bits in the two channels (PPGC0: PIE0 = 0, PPGC1: PIE1 = 1 or PPGC0: PIE0 = 1, PPGC1: PIE1 = 0).
- If the underflow generation flag bits in the two channels are set (PPGC0: PUF0 = 0 and PPGC1: PUF1 = 0), clear the two channels at the same time.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Output waveform in 16-bit PPG output operation mode

- The High and Low pulse widths to be output are determined by adding 1 to the value in the PPG reload register and multiplying it by the count clock cycle. For example, if the value in the PPG reload register is "0000_H", the pulse width has one count clock cycle, and if the value is "$FFFF_H$", the pulse width has 65,536 count clock cycles.

The equations for calculating the pulse width are shown below:

PL= T x (L + 1)

PH= T x (H + 1)

PL: Low width of output pulse

PH: High width of output pulse

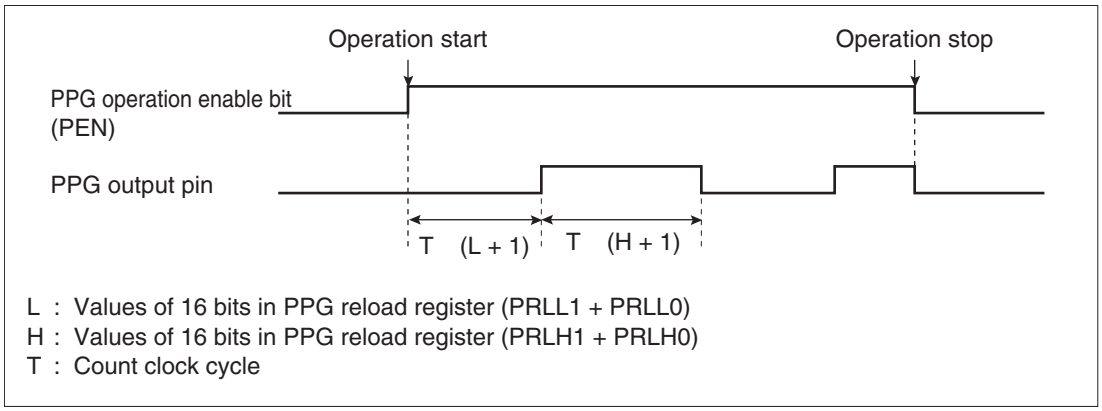
L: Values of 16 bits in PPG reload register (PRLL0 + PRLL1)

H: Values of 16 bits in PPG reload register (PRLH0 + PRLH1)

T: Count clock cycle

Figure 10.5-5 shows the output waveform in the 16-bit PPG output operation mode.

Figure 10.5-5 Output Waveform in 16-bit PPG Output Operation Mode



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

10.5.3 8+8-bit PPG Output Operation Mode

In the 8+8-bit PPG output operation mode, the 8-/16-bit PPG timer is set as an 8-bit PPG timer. The PPG0 operates as an 8-bit prescaler and the PPG1 operates using the PPG output of the PPG0 as a clock source.

■ Setting for 8+8-bit PPG Output Operation Mode

Operating the 8-/16-bit PPG timer in the 8+8-bit PPG output operation mode requires the setting shown in Figure 10.5-6.

Figure 10.5-6 Setting for 8+8-bit PPG Output Operation Mode

	bit15	14	13	12	11	10	9	bit8	bit7	6	5	4	3	2	1	bit0
PPGC1/PPGC0	PEN1	–	PE1	PIE1	PUF1	MD1	MD0	Reserved	PEN0	–	PE0	PIE0	PUF0	–	–	Reserved
	1		◎	◎	◎	0	1	1	1		◎	◎	◎			1
PPG01	(Reserved area)								PCS2	PCS1	PCS0	PCM2	PCM1	PCM0	–	–
									X	X	X	◎	◎	◎		
PRLH0/PRLL0	PPG0 Set High level side reload values.								PPG0 Set Low level side reload values.							
PRLH1/PRLL1	PPG1 Set High level side reload values.								PPG1 Set Low level side reload values.							

◎ : Used bit
X: Undefined bit
–: Unused bit
1: Set 1
0: Set 0

Note: Use the word instruction to set both High-level and Low-level PPG reload registers (PRLL0/PRLH0 and PRLL1/PRLH1) at the same time.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Operation in 8+8-bit PPG output operation mode

- The PPG0 operates as the prescaler of the PPG1 timer and the PPG1 operates using the PPG0 output as a count clock.
- When pin output is enabled (PPGC0: PE0 = 1, PPGC1: PE1 = 1), the PPG0 pulse wave is output from the PPG0 pin and the PPG1 pulse wave is output form the PPG1 pin.
- When the reload value is set in the PPG reload registers (PRLL0/PRLH0, PRLL1/PRLH1) to enable operation of the PPG timer (PPGC0: PEN0 = 1 and PPGC1: PEN1 = 1), the PPG down counter starts counting.
- To stop the count operation of the PPG down counters, disable the operation of the PPG timers of both channels (PPGC0: PEN0 = 0 and PPGC1: PEN1 = 0) at the same time. The count operation of the PPG down counters is stopped and the output of the PPG output pin is held at a Low level.
- If the PPG down counter of each channel underflows, the reload values set in the PPG reload registers (PRLL0/PRLH0, PRLL1/PRLH1) are reloaded to the PPG down counter that underflows.
- When an underflow occurs, the underflow generation flag bit in the channel that causes an underflow (PPGC0: PUF0 = 1, PPGC1: PUF1 = 1) is set. If an interrupt request is enabled at the channel that causes an underflow (PPGC0: PIE0 = 1, PPGC1: PIE1 = 1), an interrupt request is generated.

Notes:
- Do not operate PPG1 (PPGC1: PEN1 = 1) when PPG0 is stopped (PPGC0: PEN0 = 0).
- It is recommended to set the same value in both Low-level and High-level PPG reload registers (PRLL0/PRLH0, PRLL1/PRLH1).

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Output waveform in 8+8-bit PPG output operation mode

- The High and Low pulse widths to be output are determined by adding 1 to the value in the PPG reload register and multiplying it by the count clock cycle.

The equations for calculating the pulse width are shown below:

PL = T x (L_0+ 1) x (L_1+ 1)

PH = T x (H_0+ 1) x (H_1+ 1)

PL: Low width of output pulse of PPG1 pin

PH: High width of output pulse of PPG1 pin

L_0: Values of 8 bits in PPG reload register (PRLL0)

H_0: Values of 8 bits in PPG reload register (PRLH0)

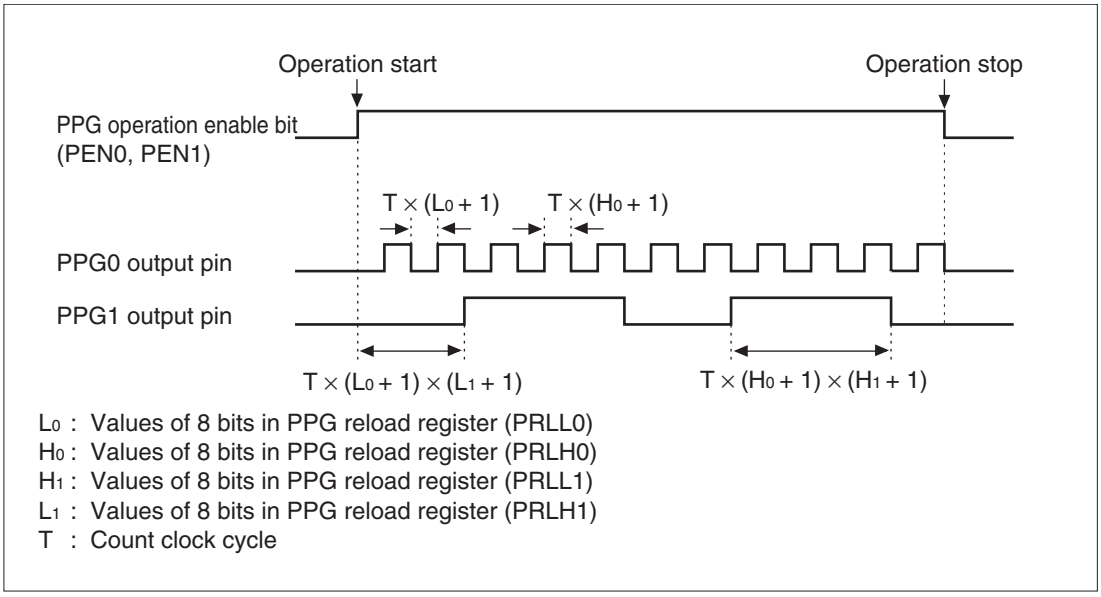
L_1: Values of 8 bits in PPG reload register (PRLL1)

H_1: Values of 8 bits in PPG reload register (PRLH1)

T: Count clock cycle

Figure 10.5-7 shows the output waveform in the 8+8-bit PPG output operation mode.

Figure 10.5-7 Output Waveform in 8+8-bit PPG Output Operation Mode



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

10.6 Precautions when Using 8-/16-bit PPG Timer

This section explains the precautions when using the 8-/16-bit PPG timer.

■ Precautions when Using 8-/16-bit PPG Timer

● Effect on 8-/16-bit PPG timer when using timebase timer output

- If the output signal of the timebase timer is used as the input signal for the count clock of the 8-/16-bit PPG timer (PPG01: PCM2 to PCM0 = "111_B", PCS2 to PCS0 = "111_B"), deviation may occur in the first count cycle in which the PPG timer is started by trigger input or in the count cycle immediately after the PPG timer is stopped.
- When the timebase timer counter is cleared during the count operation of the PPG down counter, deviation may occur in the count cycle.

● Setting of PPG reload registers when using 8-bit PPG timer

- The Low-level and High-level pulse widths are determined at the timing of reloading the values in the Low-level PPG reload registers (PRLL0, PRLL1) to the PPG down counter.
- If the 8-bit PPG timer is used in the 8-bit PPG output 2-channel independent operation mode or the 8 + 8-bit PPG output operation mode, use a word instruction to set both High-level and Low-level PPG reload registers (PRLL0/PRLH0, PRLL1/PRLH1) at the same time.

et4U.com DataShee DataSheet4U.com

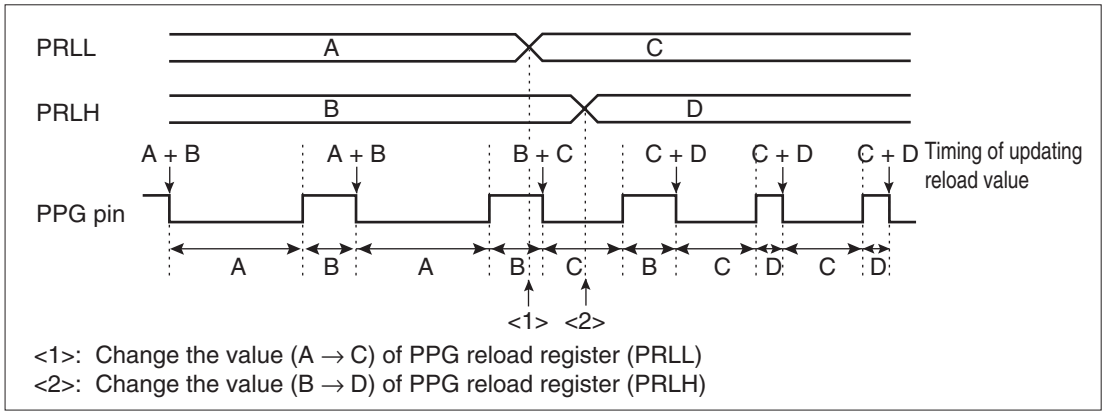
Using a byte instruction may cause an unexpected pulse to be generated.

[Example of rewriting PPG reload registers using byte instruction]

Immediately before the signal level of the PPG pin switches from High to Low, if the value in the High-level PPG reload register (PRLH) is rewritten after the value in the Low-level PPG reload register (PRLL) is rewritten using the byte instruction, a Low-level pulse width is generated after rewriting and a High-level pulse width is generated before rewriting.

Figure 10.6-1 shows the waveform as the values in the PPG reload registers are rewritten using the byte instruction.

Figure 10.6-1 Waveform when Values in PPG Reload Registers Rewritten Using Byte Instruction



DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Setting of PPG reload registers when using 16-bit PPG timer

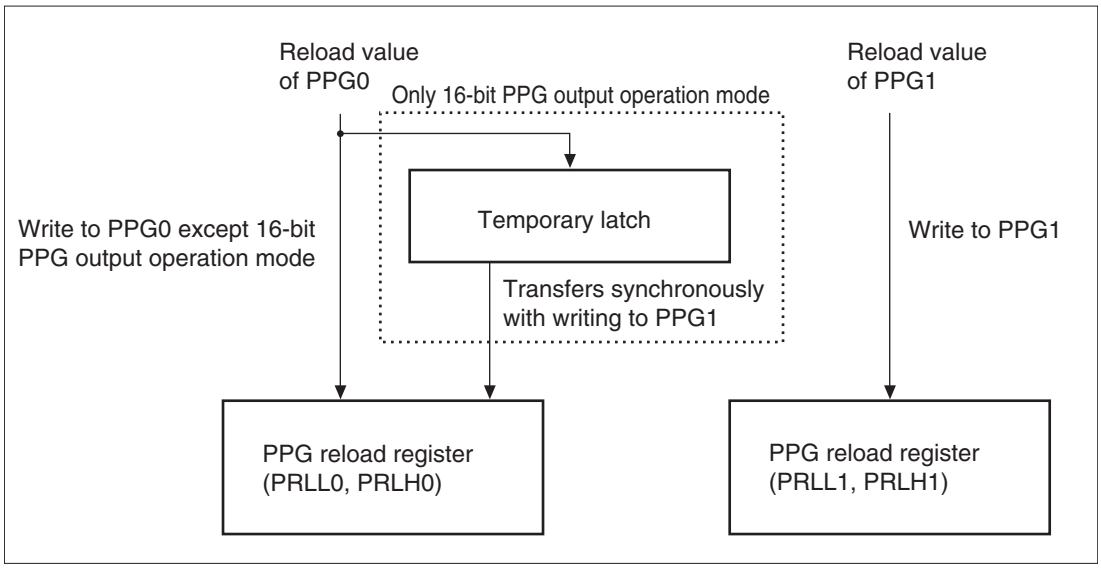
- Use a long-word instruction to set the PPG reload registers (PRLL0/PRLH0, PRLL1/PRLH1) or a word instruction to set the PPG0 and PPG1 (PRLL0 --> PRLL1 or PRLH0 --> PRLH1) in this order.

[Reload timing in 16-bit PPG output operation mode]

In the 16-bit PPG output operation mode, the reload values written to the PPG0 reload registers (PRLL0/PRLH0) are written temporarily to the temporary latch, written to the PPG1 reload registers (PRLL1/PRLH1), and then transferred to the PPG0 reload registers (PRLL0/PRLH0). Therefore, when setting the reload value in the PPG1 reload registers (PRLL1/PRLH1), it is necessary to set the reload value in the PPG0 reload registers (PRLL0/PRLH0) simultaneously or set the reload value in the PPG0 reload registers (PRLL0/PRLH0) before setting it in the PPG1 reload registers (PRLL1/PRLH1).

Figure 10.6-2 shows the reload timing in the 16-bit PPG output operation mode.

Figure 10.6-2 Reload Timing in 16-bit PPG Output Operation Mode



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 11

DELAYED INTERRUPT GENERATION MODULE

This chapter explains the functions and operations of the delayed interrupt generation module.

11.1 Overview of Delayed Interrupt Generation Module
11.2 Block Diagram of Delayed Interrupt Generation Module
11.3 Configuration of Delayed Interrupt Generation Module
11.4 Explanation of Operation of Delayed Interrupt Generation Module
11.5 Precautions when Using Delayed Interrupt Generation Module
11.6 Program Example of Delayed Interrupt Generation Module

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

11.1 Overview of Delayed Interrupt Generation Module

The delayed interrupt generation module generates the interrupt for task switching. The hardware interrupt request can be generated/cancelled by software.

■ Overview of Delayed Interrupt Generation Module

By using the delayed interrupt generation module, a hardware interrupt request can be generated or cancelled by software.

Table 11.1-1 shows the overview of the delayed interrupt generation module.

Table 11.1-1 Overview of Delayed Interrupt Generate Module

	Function and Control
Interrupt factor	An interrupt request is generated by setting the R0 bit in the delayed interrupt request generate/cancel register to 1 (DIRR: R0 = 1). An interrupt request is cancelled by setting the R0 bit in the delayed interrupt request generate/cancel register to 0 (DIRR: R0 = 0).
Interrupt number	#42 ($2A_H$)
Interrupt control	An interrupt is not enabled by the DIRR register.
Interrupt flag	The interrupt flag is held in the R0 bit in the DIRR register.
EI^2OS	The DIRR register does not correspond to the EI^2OS.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

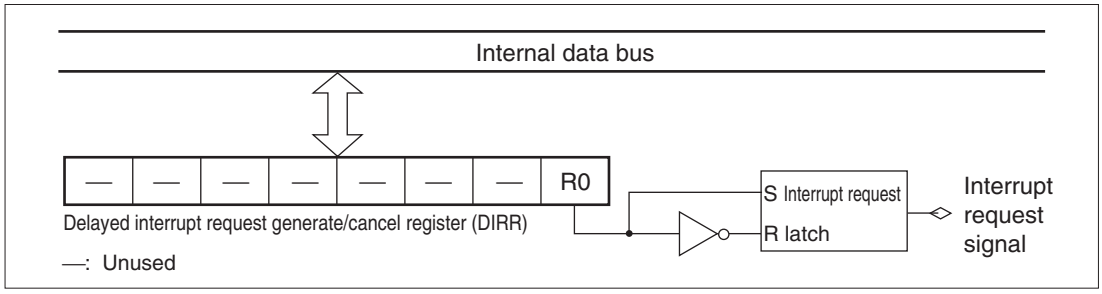
11.2 Block Diagram of Delayed Interrupt Generation Module

The delayed interrupt generation module consists of the following blocks:

- **Interrupt request latch**
- **Delayed interrupt request generate/cancel register (DIRR)**

■ Block Diagram of Delayed Interrupt Generation Module

Figure 11.2-1 Block Diagram of Delayed Interrupt Generation Module



● Interrupt request latch

This latch keeps the settings (delayed interrupt request generation or cancellation) of the delayed interrupt request generate/cancel register (DIRR).

● Delayed interrupt request generate/cancel register (DIRR)

This register generates or cancels a delayed interrupt request.

■ Interrupt Number

The interrupt number used in the delayed interrupt generation module is as follows:

Interrupt number #42 ($2A_H$)

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

11.3 Configuration of Delayed Interrupt Generation Module

This section lists registers and reset values in the delayed interrupt generation module.

■ List of Registers and Reset Values in Delayed Interrupt Generation Module

Figure 11.3-1 List of Registers and Reset Values in Delayed Interrupt Generation Module

bit	15	14	13	12	11	10	9	8
Delayed interrupt request generate/ cancel register (DIRR)	X	X	X	X	X	X	X	0

X: Undefined

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSh4U.com

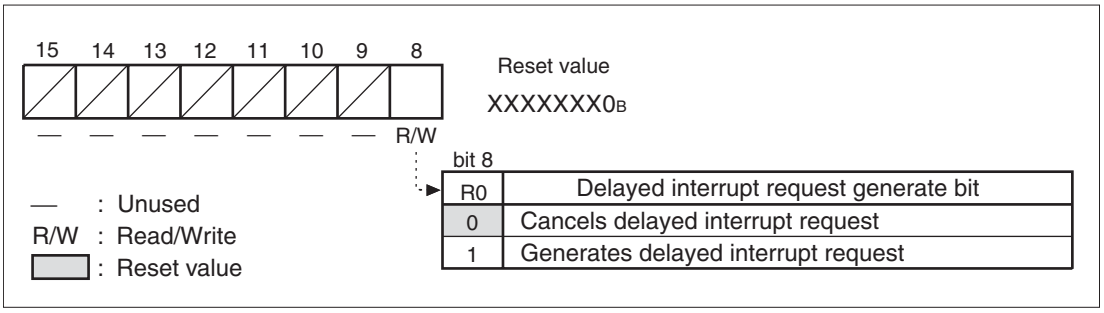
www.DataSheet4U.com

11.3.1 Delayed Interrupt Request Generate/Cancel Register (DIRR)

The delayed interrupt request generate/cancel register (DIRR) generates or cancels a delayed interrupt request.

■ Delayed Interrupt Request Generate/Cancel Register (DIRR)

Figure 11.3-2 Delayed Interrupt Request Generate/Cancel Register (DIRR)



et4U.com DataShee

Table 11.3-1 Functions of Delayed Interrupt Request Generate/Cancel Register (DIRR)

DataSheet4U.com

Bit Name		Function
bit 8	R0: Delayed interrupt request generate bit	This bit generates or cancels a delayed interrupt request. **When set to 0:** Cancels delayed interrupt request **When set to 1:** Generates delayed interrupt request
bit 9 to bit 15	Unused bits	**Read:** The value is undefined **Write:** No effect

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

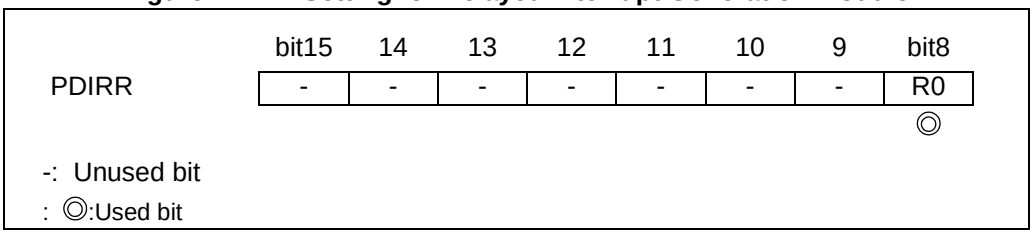
11.4 Explanation of Operation of Delayed Interrupt Generation Module

The delayed interrupt generation module has a function for generating or canceling an interrupt request by software.

■ Explanation of Operation of Delayed Interrupt Generation Module

Using the delayed interrupt generation module requires the setting shown in Figure 11.4-1.

Figure 11.4-1 Setting for Delayed Interrupt Generation Module



When the R0 bit in the delayed interrupt request generate/cancel register (DIRR) is set to 1 (DIRR: R0 = 1), an interrupt request is generated. There is no interrupt request enable bit.

et4U.com DataShee

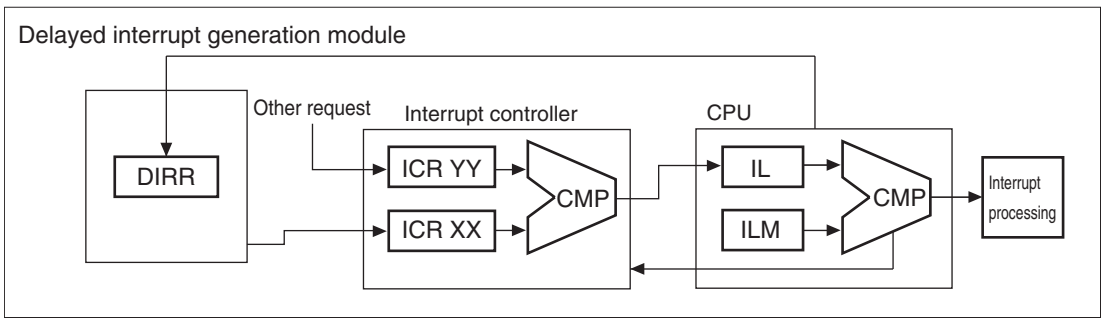
● Operation of delayed interrupt generation module

DataSheet4U.com

- When the R0 bit in the delayed interrupt request generate/cancel register (DIRR) is set to 1, the interrupt request latch is set to 1 and an interrupt request is generated to the interrupt controller.
- When an interrupt request is preferred to other requests by the interrupt controller, the interrupt request is generated to the CPU.
- When the level of an interrupt request (ICR: IL) is preferred to that of the interrupt level mask bit (ILM) in the processor status (PS), the CPU delays interrupt processing until completion of execution of the current instruction.
- At interrupt processing, the user program sets the R0 bit to 0, cancels the interrupt request, and changes the task.

Figure 11.4-2 shows the operation of the delayed interrupt generation module.

Figure 11.4-2 Operation of Delayed Interrupt Generation Module



DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

11.5 Precautions when Using Delayed Interrupt Generation Module

This section explains the precautions when using the delayed interrupt generation module.

■ Precautions when Using Delayed Interrupt Generation Module

- The interrupt processing is restarted at return from interrupt processing without setting the R0 bit in the delayed interrupt request generate/cancel register (DIRR) to 0 within the interrupt processing routine.
- Unlike software interrupts, interrupts in the delayed interrupt generation module are delayed.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

11.6 Program Example of Delayed Interrupt Generation Module

This section gives a program example of the delayed interrupt generation module.

■ Program Example of Delayed Interrupt Generation Module

● Processing specifications

The main program writes 1 to the R0 bit in the delayed interrupt request generate/cancel register (DIRR), generates a delayed interrupt request, and changes the task.

● Coding example

```
ICR15   EQU  0000BFH          ; Interrupt control register
DIRR    EQU  00009FH          ; Delayed interrupt request generate/cancel register
DIRR_R0 EQU  DIRR:0           ; Delayed interrupt request generate bit
;-----Main program-----------------------------------------------------------
CODE    CSEG
START:                        ; Stack pointer (SP) already initialized
        AND  CCR,#0BFH        ; Interrupt disabled
        MOV  I:ICR15,#00H     ; Interrupt level 0 (highest)
        MOV  ILM,#07H         ; ILM in PS set to level 7
        OR   CCR, #40H        ; Interrupt enabled
        SETB I:DIRR_R0        ; Delayed interrupt request generated
LOOP    MOV  A,#00H           ; Infinite loop
        MOV  A,#01H
        BRA  LOOP
;-----Interrupt program-------------------------------------------------------
WARI:
        CLRB I:DIRR_R0        ; Interrupt request flag cleared
         :
;       Processing by user
;        :
        RETI                  ; Return from interrupt
CODE    ENDS
;-----Vector setting----------------------------------------------------------
VECT    CSEG ABS=0FFH
        ORG  00FF54H          ; Vector set to interrupt  #42 (2AH)
        DSL  WARI
        ORG  0FFDCH           ; Reset vector set
        DSL  START
        DB   00H              ; Set to single-chip mode
VECT    ENDS
        END  START
```

et4U.com
DataShee
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 12

DTP/EXTERNAL INTERRUPT

This chapter explains the functions and operations of DTP/external interrupt.

12.1 Overview of DTP/External Interrupt
12.2 Block Diagram of DTP/External Interrupt
12.3 Configuration of DTP/External Interrupt
12.4 Explanation of Operation of DTP/External Interrupt
12.5 Precautions when Using DTP/External Interrupt
12.6 Program Example of DTP/External Interrupt Circuit

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

12.1 Overview of DTP/External Interrupt

The DTP/external interrupt sends interrupt requests from external peripheral devices or data transfer requests to the CPU to generate an external interrupt request, or starts the EI^{2}OS. RX input of CAN controller can be used as external interrupt input.

■ DTP/External Interrupt Function

The interrupt request inputted to external interrupt input pins (INT7 to INT4) and RX input from external peripheral devices generates an external interrupt request, or starts the EI^{2}OS as an interrupt request from peripheral function.

If the EI^{2}OS is disabled in the interrupt control register (ICR: ISE = 0), the external interrupt function is enabled, branching to interrupt processing.

If the EI^{2}OS is enabled (ICR: ISE = 1), the DTP function is enabled and automatic data transfer is performed, branching to interrupt processing after the completion of data transfer for the specified number of times.

Table 12.1-1 shows an overview of the DTP/external interrupt.

Table 12.1-1 Overview of DTP/External Interrupt

	External Interrupt	DTP Function
Input pin	5 pins (RX, INT4 to INT7)	
Interrupt factor	The interrupt factor is set in unit of pins using the detection level setting registers (ELVR).	
	Input of High level, Low level, rising edge, or falling edge	Input of High level or Low level
Interrupt number	#15 ($0F_H$), #24 (18_H), #27 ($1B_H$)	
Interrupt control	The interrupt request output is enabled/disabled using the DTP/external interrupt enable register (ENIR).	
Interrupt flag	The interrupt factor is held using the DTP/external interrupt factor register (EIRR)	
Processing selection	The EI^{2}OS is disabled. (ICR: ISE=0)	The EI^{2}OS is enabled. (ICR: ISE=1)
Processing contents	A branch is caused to the external interrupt processing.	EI^{2}OS performs auto data transfer and completes the specified number of timer for data transfers, causing a branch to the interrupt processing.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

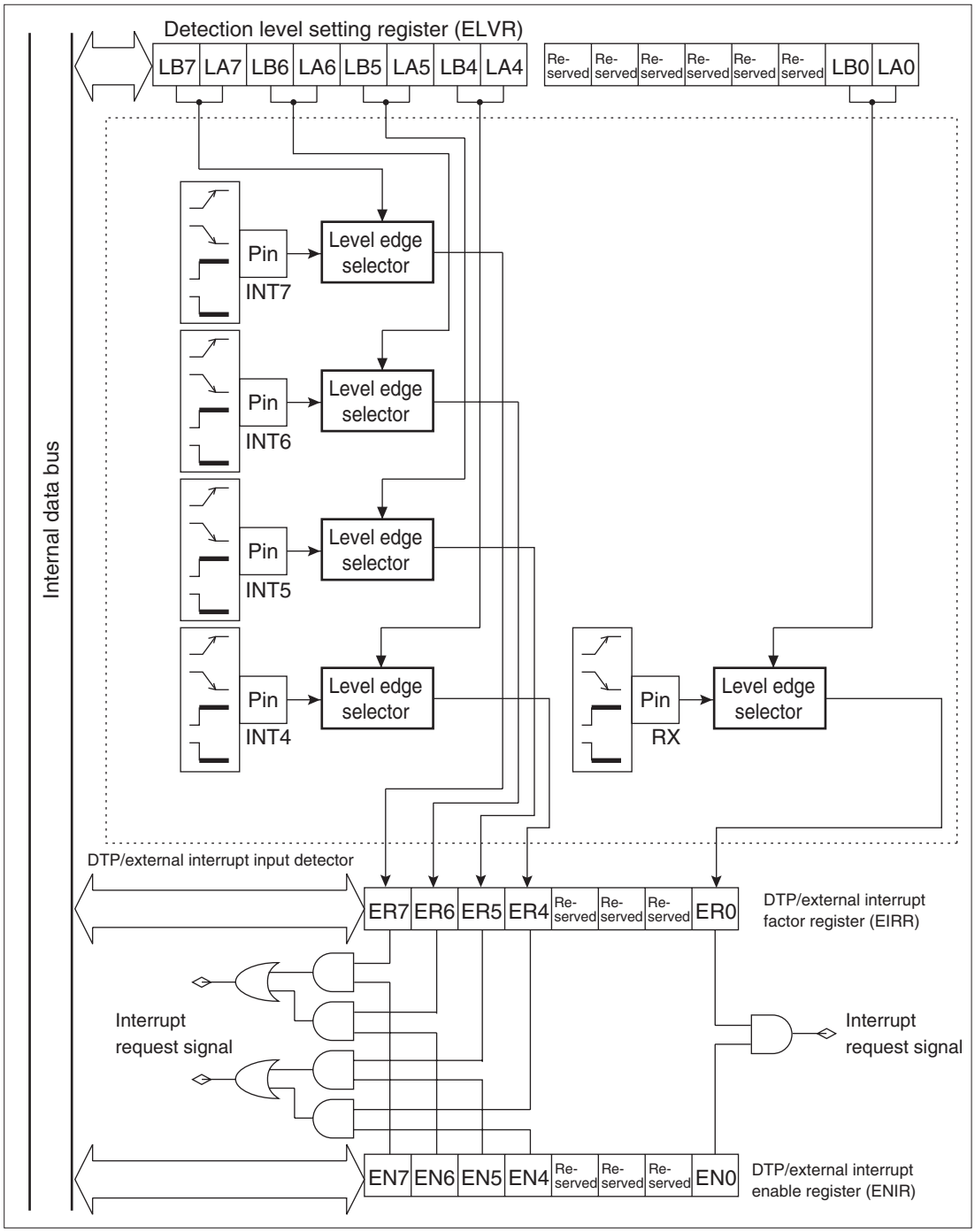
www.DataSheet4U.com

12.2 Block Diagram of DTP/External Interrupt

The block diagram of the DTP/external interrupt is shown below.

■ Block Diagram of DTP/External Interrupt

Figure 12.2-1 Block Diagram of DTP/External Interrupt



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- DTP/external interrupt input detector

  This circuit detects interrupt requests or data transfer requests generated from external peripheral devices.

  The interrupt request flag bit corresponding to the pin whose level or edge set by the detection level setting register (ELVR) is detected is set to 1 (EIRR: ER).

- Detection level setting register (ELVR)

  This register sets the level or edge of input signals from external peripheral devices that cause DTP/external interrupt factors.

- DTP/external interrupt factor register (EIRR)

  This register holds DTP/external interrupt factors.

  If an enable signal is input to the DTP/external interrupt pin, the corresponding DTP/external interrupt request flag bit is set to 1.

- DTP/external interrupt enable register (ENIR)

  This register enables or disables DTP/external interrupt requests from external peripheral devices.

■ Details of Pins and Interrupt Numbers

Table 12.2-1 shows the pins and interrupt numbers used in the DTP/external interrupt.

Table 12.2-1 Pins and Interrupt Numbers Used by DTP/External Interrupt

Pin	Channel	Interrupt Number
P44/RX	RX	#15 ($0F_H$)
P24/INT4	4	#24 (18_H)
P25/INT5	5	
P26/INT6	6	#27 ($1B_H$)
P27/INT7	7	

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

12.3 Configuration of DTP/External Interrupt

This section lists and details the pins, interrupt factors, and registers in the DTP/ external interrupt.

■ Pins of DTP/External Interrupt

The pins used by the DTP/external interrupt serve as general-purpose I/0 ports.

Table 12.3-1 lists the pin functions and the pin setting required for use in the DTP/external interrupt

Table 12.3-1 Pins of DTP/External Interrupt

Pin Name	Pin Function	Pin Settings Required for Use in DTP/External Interrupt
P44/RX	General-purpose I/O ports, CAN reception input	Set as input ports in port direction register (DDR)
P24/INT4	General-purpose I/O ports, DTP external interrupt inputs	
P25/INT5		
P26/INT6		
P27/INT7		

DataSheet4U.com

■ Block Diagram of Pins

See "CHAPTER 4 I/O PORT" for the block diagram of pins.

■ List of Registers and Reset Values in DTP/External Interrupt

Figure 12.3-1 List of Registers and Reset Values in DTP/External Interrupt

	bit	15	14	13	12	11	10	9	8
DTP/external interrupt factor register (EIRR)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
DTP/external interrupt enable register (ENIR)		0	0	0	0	0	0	0	0
	bit	15	14	13	12	11	10	9	8
Detection level setting register: High (ELVR)		0	0	0	0	0	0	0	0
	bit	7	6	5	4	3	2	1	0
Detection level setting register: Low (ELVR)		0	0	0	0	0	0	0	0

X: Undefined

DataSheet4U.com www.DataSheet4U.com

DataSh4U.com

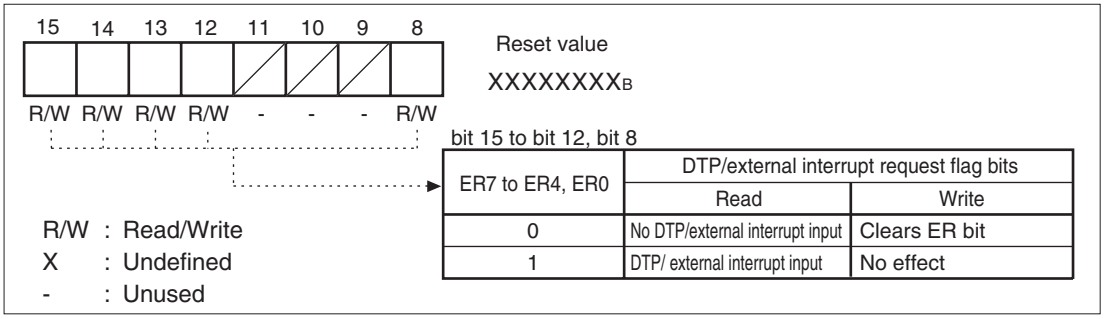
www.DataSheet4U.com

12.3.1 DTP/External Interrupt Factor Register (EIRR)

The DTP/external interrupt factor register (EIRR) holds DTP/external interrupt factors. When a valid signal is input to the DTP/external interrupt pin and the RX pin, the corresponding interrupt request flag bit is set to 1.

■ DTP/External Interrupt Factor Register (EIRR)

Figure 12.3-2 DTP/External Interrupt Factor Register (EIRR)



et4U.com DataShee

Table 12.3-2 Function of DTP/External Interrupt Factor Register (EIRR)

DataSheet4U.com

Bit Name		Function
bit 8, bit 12 to bit 15	ER7 to ER4, ER0: DTP/External interrupt request flag bits	These bits are set to 1 when the edges or level signals set by the detection condition select bits in the detection level setting register (ELVR: LB, LA) are input to the DTP/external interrupt pins and RX pin. **When set to 1:** When the DTP/external interrupt request enable bit (ENIR: EN) is set to 1, an interrupt request is generated to the corresponding DTP/ external interrupt channel. **When set to 0:** Cleared **When set to 1:** No effect **Note:** Reading by read-modify-write type instructions always reads "1". If more than one DTP/external interrupt request is enabled (ENIR: EN = 1), clear only the bit in the channel that accepts an interrupt (EIRR: ER = 0). No other bits must be cleared unconditionally. **Reference:** When the EI^{2}OS is started, the interrupt request flag bit is automatically cleared after the completion of data transfer (EIRR: ER = 0)
bit 9 to bit11	Unused bit	**Read:** The value is undefined **Write:** No effect

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

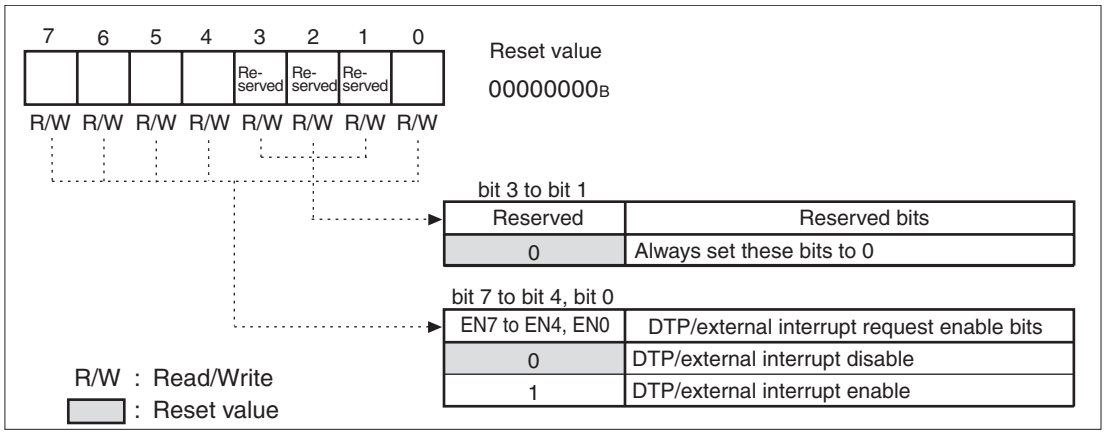
www.DataSheet4U.com

12.3.2 DTP/External Interrupt Enable Register (ENIR)

The DTP/external interrupt enable register (ENIR) enables/disables the DTP/external interrupt request for external interrupt pins (INT7 to INT4) and the RX pin respectively.

■ DTP/External Interrupt Enable Register (ENIR)

Figure 12.3-3 DTP/External Interrupt Enable Register (ENIR)



DataSheet4U.com

Table 12.3-3 Functions of DTP/External Interrupt Enable Register (ENIR)

Bit Name		Function
bit 0, bit 4 to bit 7	EN7 to EN4, EN0: DTP/external interrupt request enable bits	The DTP/external interrupt enable register (ENIR) enables/disables the DTP/external interrupt request for DTP/external interrupt pins (INT7 to INT4) and the RX pin. If the DTP/external interrupt request enable bit (ENIR: EN) and the DTP/external interrupt request flag bit (EIRR: ER) are set to 1, the interrupt request is generated to the corresponding DTP/external interrupt pins or the RX pin. **Reference:** The state of the DTP/external interrupt pin and the RX pin can be read directly using the port data register irrespective of the setting of the DTP/external interrupt request enable bit.

Table 12.3-4 Correspondence among DTP/External Interrupt Pins, DTP/External Interrupt Request Flag Bits, and DTP/External Interrupt Request Enable Bits

DTP/External Interrupt Pins	DTP/External Interrupt Request Flag Bits	DTP/External Interrupt Request Enable Bits
RX	ER0	EN0
INT4	ER4	EN4
INT5	ER5	EN5
INT6	ER6	EN6
INT7	ER7	EN7

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

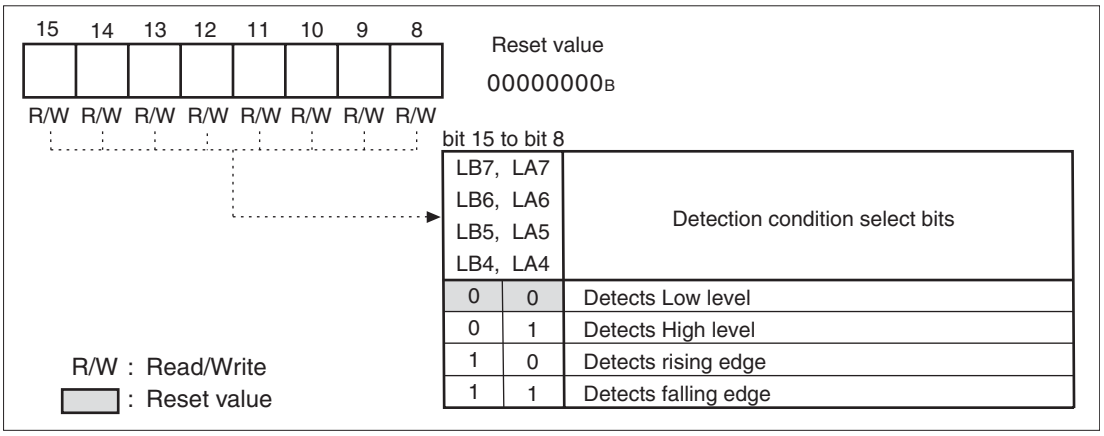
www.DataSheet4U.com

12.3.3 Detection Level Setting Register (ELVR) (High)

The detection level setting register (High) sets the levels or edges of input signals that cause interrupt factors in INT7 to INT4 of the DTP/external interrupt pins.

■ Detection Level Setting Register (ELVR) (High)

Figure 12.3-4 Detection Level Setting Register (ELVR) (High)



DataSheet4U.com

Table 12.3-5 Functions of Detection Level Setting Register (ELVR) (High)

Bit Name		Function
bit 8 to bit 15	LB7, LA7 to LB4, LA4: Detection condition select bits	These bits set the levels or edges of input signals from external peripheral devices that cause interrupt factors in the DTP/external interrupt pins. • Two levels or two edges are selectable for external interrupts, and two levels are selectable for the EI^{2}OS. **Reference:** When the set detection signal is input to the DTP/external interrupt pins, the DTP/external interrupt request flag bits are set to 1 even if DTP/external interrupt requests are disabled (ENIR: EN = 0).

Table 12.3-6 Correspondence between Detection Level Setting Register (ELVR) (High) and Channels

DTP/External Interrupt Pin	Bit Name
INT4	LB4, LA4
INT5	LB5, LA5
INT6	LB6, LA6
INT7	LB7, LA7

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

12.3.4 Detection Level Setting Register (ELVR) (Low)

The detection level setting register (ELVR) (Low) sets the levels or edges of input signals that cause interrupt factors in the RX pin.

■ Detection Level Setting Register (ELVR) (Low)

Figure 12.3-5 Detection Level Setting Register (ELVR) (Low)

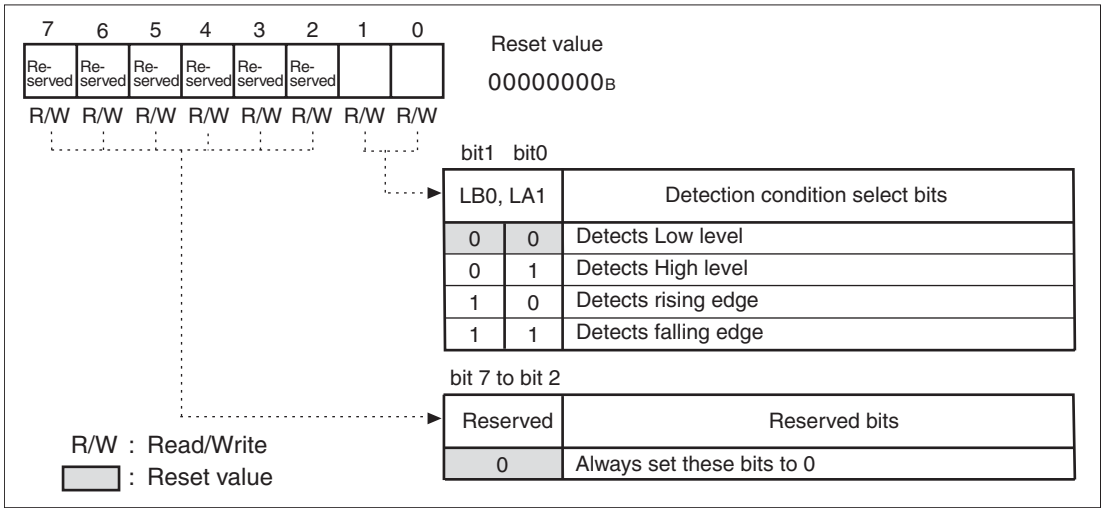


Table 12.3-7 Functions of Detection Level Setting Register (ELVR) (Low)

Bit Name		Function
bit 0 to bit 1	LB3, LA0: Detection condition select bits	These bits set the levels or edges of input signals from external peripheral devices that cause interrupt factors in the RX pin. • Two levels or two edges are selectable for external interrupts, and two levels are selectable for the EI^{2}OS. **Reference:** When the set detection signal is input to the RX pin, the DTP/external interrupt request flag bits are set to 1 even if DTP/external interrupt requests are disabled (ENIR: EN = 0).

Table 12.3-8 Correspondence between Detection Level Setting Register (ELVR) (Low) and Channels

DTP/External Interrupt Pin	Bit Name
RX	LB0, LA0

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

12.4 Explanation of Operation of DTP/External Interrupt

The DTP/external interrupt has an external interrupt function and a DTP function. The setting and operation of each function are explained.

■ Setting of DTP/External Interrupt

Using the DTP/external interrupt requires, the setting shown in Figure 12.4-1.

Figure 12.4-1 Setting of DTP/External Interrupt

	bit15	14	13	12	11	10	9	bit8	bit7	6	5	4	3	2	1	bit0
ICR interrupt control register	ICS3	ICS2	ICS1	ICS0	ISE	IL2	IL1	IL0	ICS3	ICS2	ICS1	ICS0	ISE	IL2	IL1	IL0
External interrupt	-	-	-	-	0	◎	◎	◎	-	-	-	-	0	◎	◎	◎
DTP	◎	◎	◎	◎	1	◎	◎	◎	◎	◎	◎	◎	1	◎	◎	◎
EIRR/ENIR	ER7	ER6	ER5	ER4	-	-	-	ER0	EN7	EN6	EN5	EN4	Re-served	Re-served	Re-served	EN0
	◎	◎	◎	◎	-	-	-	◎	○	○	○	○	0	0	0	○
ELVR	LB7	LA7	LB6	LA6	LB5	LA5	LB4	LA4	Re-served	Re-served	Re-served	Re-served	Re-served	Re-served	LB0	LA0
	◎	◎	◎	◎	◎	◎	◎	◎	0	0	0	0	0	0	◎	◎
DDR port direction register	⊠	⊠	⊠	⊠	⊠	⊠	⊠	⊠								

Set the bit corresponding to pin used for DTP/external interrupt input to 0.

–: Unused bit
◎: Used bit
○: Set the bit corresponding to used pin to 1
0: Set 0
1: Set 1

● Setting procedure

To use the DTP/external interrupt, set each register by using the following procedure:

1. Set the interrupt request enable bit corresponding to the DTP/external interrupt channel to be used to 0 (ENIR: EN).
2. Use the detection condition select bit corresponding to the DTP/external interrupt pin and the RX pin to be used to set the edge or level to be detected (ELVR: LA, LB).
3. Set the interrupt request flag bit corresponding to the DTP/external interrupt channel to be used to 0 (EIRR: ER).
4. Set the interrupt request enable bit corresponding to the DTP/external interrupt channel to be used to 1 (ENIR: EN).

- When setting the registers for the DTP/external interrupt, the external interrupt request must be disabled in advance (ENIR: EN = 0).
- When enabling the DTP/external interrupt request (ENIR: EN = 1), the corresponding DTP/external interrupt request flag bit must be cleared in advance (EIRR: ER = 0). These actions prevent the mistaken interrupt request from occurring when setting the register.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Selecting of DTP or external interrupt function

Whether the DTP function or the external interrupt function is executed depends on the setting of the EI^{2}OS enable bit in the corresponding interrupt control register (ICR: ISE).

If the ISE bit is set to 1, the EI^{2}OS is enabled and the DTP function is executed.

If the ISE bit is set to 0, the EI^{2}OS is disabled and the external interrupt function is executed.

Notes:
- All interrupt requests assigned to one interrupt control register have the same interrupt levels (IL2 to IL0).
- If two or more interrupt requests are assigned to one interrupt control register and the EI^{2}OS is used in one of them, other interrupt requests cannot be used.

■ DTP/External Interrupt Operation

The control bits and the interrupt factors for the DTP/external interrupt are shown in Table 12.4-1.

Table 12.4-1 Control Bits and Interrupt Factors for DTP/External Interrupt

	DTP/External Interrupt
Interrupt request flag bit	EIRR: ER7 to ER4, ER0
Interrupt request enable bit	ENIR: EN7 to EN4, EN0
Interrupt factor	Input of valid edge/level to INT7 to INT4, RX pins

et4U.com DataShee DataSheet4U.com

If the interrupt request from the DTP/external interrupt is output to the interrupt controller and the EI^{2}OS enable bit in the interrupt control register (ICR: ISE) is set to 0, the interrupt processing is executed. This bit is set to 1, the EI^{2}OS is executed.

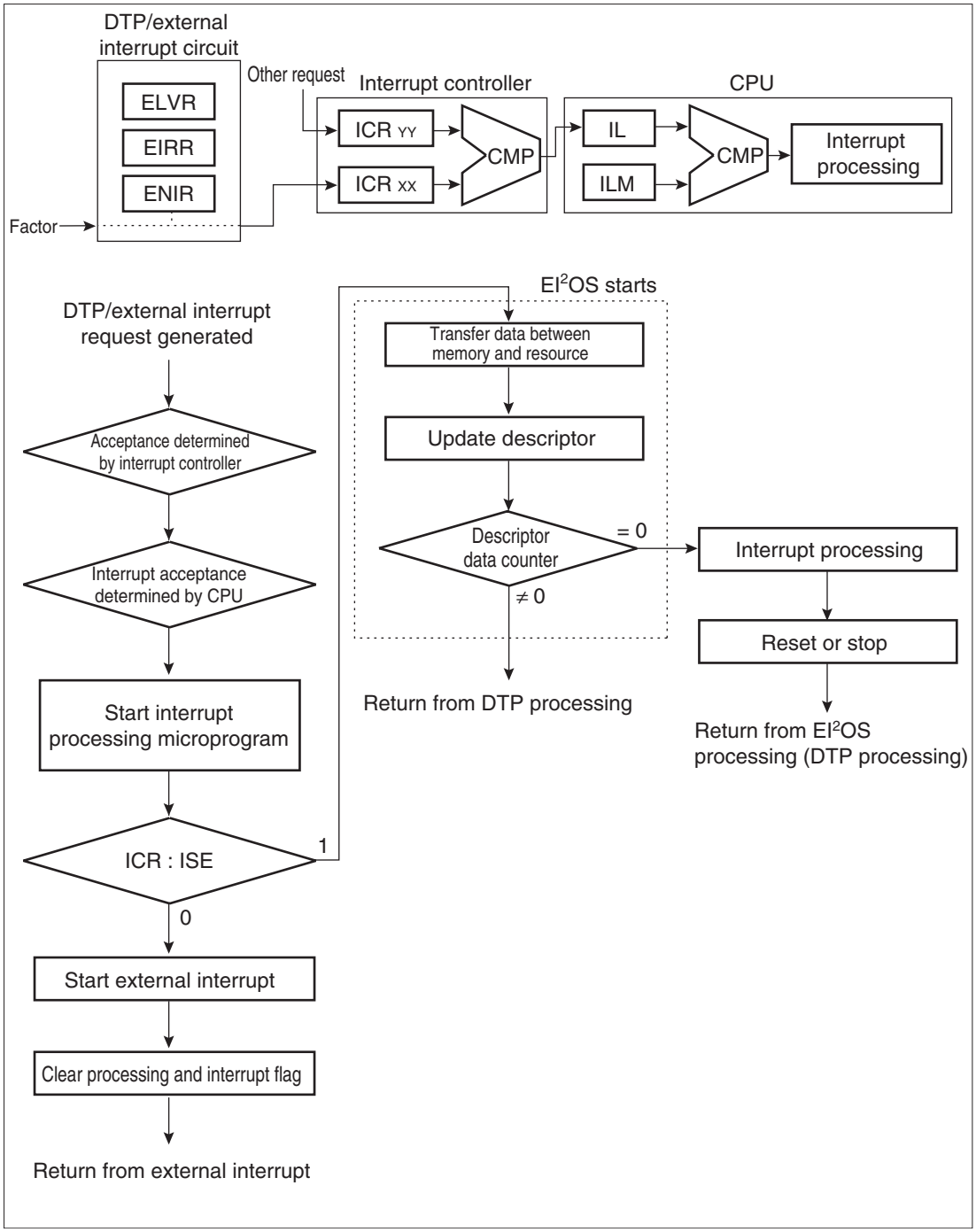
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 12.4-2 shows the operation of the DTP/external interrupt.

Figure 12.4-2 Operation of DTP/External Interrupt



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

12.4.1 External Interrupt Function

The DTP/external interrupt has an external interrupt function for generating an interrupt request by detecting the signal (edge or level) in the DTP/external interrupt pin and the RX pin.

■ External Interrupt Function

- When the signal (edge or level) set in the detection level setting register is detected in the DTP/external interrupt pin and the RX pin, the interrupt request flag bit in the DTP/external interrupt factor register (EIRR: ER) is set to 1.
- If the interrupt request enable bit in the DTP/external interrupt enable register is enabled (ENIR: EN = 1) and the interrupt request flag bit set to 1, the interrupt is implemented to the interrupt controller.
- If an interrupt request is preferred to other interrupt request by the interrupt controller, the interrupt request is generated.
- If the level of an interrupt request (ICR: IL) is higher than that of the interrupt level mask bit (TLM) in the processor status (PS) and the interrupt enable bit is enabled (PS: CCR: I = 1), the CPU performs interrupt processing after completion of the current instruction execution and branches to interrupt processing.
- At interrupt processing, set the corresponding DTP/external interrupt request flag bit to 0 and clear the DTP/external interrupt request.

et4U.com DataShee
DataSheet4U.com

Notes:
- When the DTP/external interrupt start factor is generated, the DTP/external interrupt request flag bit (EIRR: ER) is set to 1, regardless of the setting of the DTP/external interrupt request enable bit (ENIR: EN).
- When the interrupt processing is started, clear the DTP/external interrupt request flag bit that caused the start factor. Control cannot be returned from the interrupt while the DTP/external interrupt request flag bit is set to 1. When clearing, do not clear any flag bit other than the accepted DTP/external interrupt factor.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

12.4.2 DTP Function

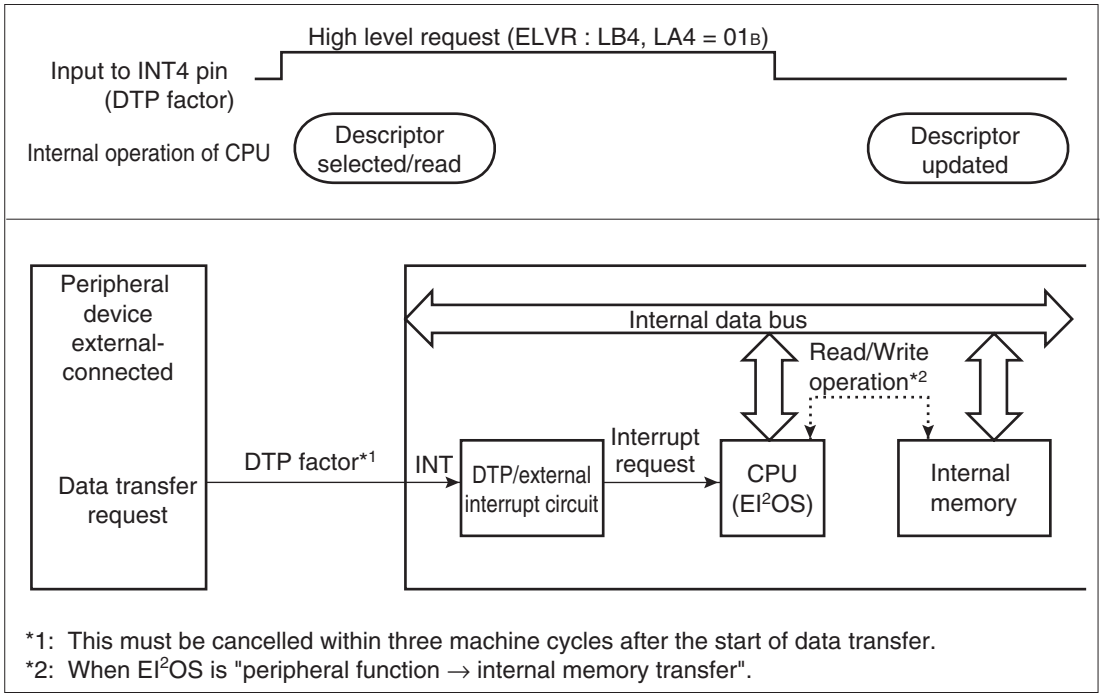
The DTP/external interrupt has the DTP function that detects the signal of the external peripheral device from the DTP/external interrupt pin and the RX pin to start the EI^{2}OS.

■ DTP Function

The DTP function detects the signal level set by the detection level setting register of the DTP/external interrupt function to start the EI^{2}OS.

- When the EI^{2}OS operation is already enabled (ICR: ISE = 1) at the point when the interrupt request is accepted by the CPU, the DTP function starts the EI^{2}OS and starts data transfer.
- When transfer of one data item is completed, the descriptor is updated and the DTP/external interrupt request flag bit is cleared to prepare for the next request from the DTP/external interrupt pin and the RX pin.
- When the EI^{2}OS completes transfer of all the data, control branches to the interrupt processing.

Figure 12.4-3 Example of Interface with External Peripheral Device



*1: This must be cancelled within three machine cycles after the start of data transfer.
*2: When EI^{2}OS is "peripheral function → internal memory transfer".

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

12.5 Precautions when Using DTP/External Interrupt

This section explains the precautions when using the DTP/external interrupt.

■ Precautions when Using DTP/External Interrupt

● Condition of external-connected peripheral device when DTP function is used

- When using the DTP function, the peripheral device must automatically clear a data transfer request when data transfer is performed.
- Inactivate the transfer request signal within three machine cycles after starting data transfer. If the transfer request signal remains active, the DTP/external interrupt regards the transfer request signal as a generation of next transfer request.

● External interrupt input polarity

- When the edge detection is set in the detection level setting register, the pulse width for edge detection must be at least three machine cycles.
- When a level causing an interrupt factor is input with level detection set in the detection level setting register, the interrupt request flag bit (EIRR:ER) of the DTP/external interrupt factor register is set to 1 and the factor is held as shown in Figure 12.5-1.

With the factor held in the interrupt request flag bit (EIRR:ER), the request to the interrupt controller remains active if the interrupt request is enabled (ENIR: EN = 1) even after the DTP/external interrupt factor is cancelled. To cancel the request to the interrupt controller, clear the interrupt request flag bit (EIRR:ER) as shown in Figure 12.5-2.

Figure 12.5-1 Clearing Interrupt Request Flag Bit (EIRR:ER) when Level Set

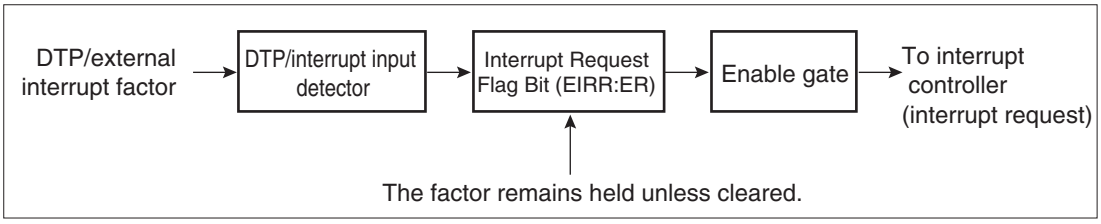
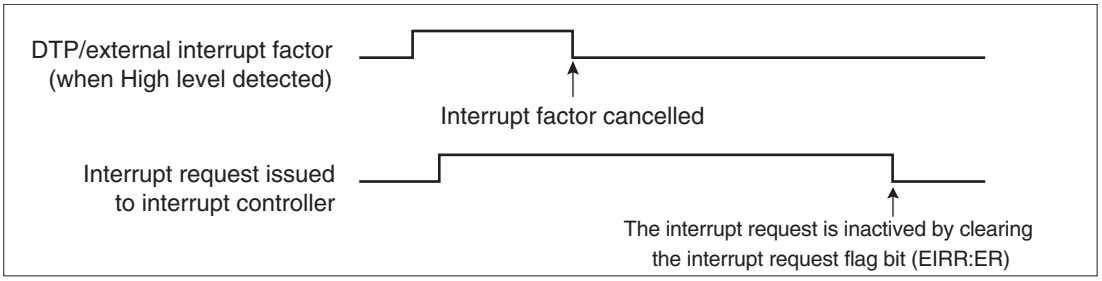


Figure 12.5-2 DTP/External Interrupt Factor and Interrupt Request Generated when Interrupt Request Enabled



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Precautions on interrupts

- When the DTP/external interrupt is used as the external interrupt function, no return from interrupt processing can be made with the DTP/external interrupt request flag bit set to 1 (EIRR: ER) and the DTP/external interrupt request set to "enabled" (ENIR: EN = 1). Always set the DTP/external interrupt request flag bit to 0 (EIRR: ER) at interrupt processing.
- When the level detection is set in the detection level setting register and the level that becomes the interrupt factor remains input, the DTP/external interrupt request flag bit is reset immediately even when cleared (EIRR: ER = 0). Disable the DTP/external interrupt request output as needed (ENIR: EN = 0), or cancel the interrupt factor itself.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

12.6 Program Example of DTP/External Interrupt Circuit

This section gives a program example of the DTP/external interrupt function.

■ Program Example of DTP/External Interrupt Function

● Processing specifications

An external interrupt is generated by detecting the rising edge of the pulse input to the INT4 pin.

● Coding example

```
ICR06   EQU  0000B6H           ; DTP/external interrupt control register
DDR2    EQU  000012H           ; Port 2 direction register
ENIR    EQU  000030H           ; DTP/external interrupt enable register
EIRR    EQU  000031H           ; DTP/external interrupt factor register
ELVRL   EQU  000032H           ; Detection level setting register: L
ELVRH   EQU  000033H           ; Detection level setting register: H
ER0     EQU  EIRR:0            ; INT4 Interrupt request flag bit
EN0     EQU  ENIR:0            ; INT4 Interrupt request enable bit
;
;-----Main program-----------------------------------------------------------
CODE    CSEG
START:                          ; Stack pointer (SP) already initialized
        MOV  I:DDR2,#00000000B  ; DDR2 set to input port
        AND  CCR,#0BFH          ; Interrupts disabled
        MOV  I:ICR06,#00H       ; Interrupt level 0 (highest)
        CLRB I:ER4              ; INT4 disabled using ENIR
        MOV  I:ELVRL,#00000010B ; Rising edge selected for INT4
        CLRB I:ER4              ; INT4 interrupt request flag
                                ; cleared using EIRR
        SETB I:EN4              ; INT4 interrupt request enabled using ENIR
        MOV  ILM, #07H          ; ILM in PS set to level 7
        OR   CCR, #40H          ; Interrupts enabled
LOOP:
         .
        Processing by user
         .
        BRA  LOOP
;-----Interrupt program------------------------------------------------------
WARI:
        CLRB I:ER4              ; Interrupt request flag cleared
         .
        Processing by user
         .
        RETI                    ; Return from interrupt processing
CODE    ENDS
;-----Vector setting---------------------------------------------------------
VECT    CSEG ABS=0FFH
        ORG  00FFC0H            ; Vector set to interrupt number #15 (0F$_H$)
        DSL  WARI
        ORG  00FFDCH            ; Reset vector set
        DSL  START
        DB   00H                ; Set to single-chip mode
VECT    ENDS
        END  START
```

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Program Example of DTP Function

● Processing specification

- Channel 0 of the EI^{2}OS is started by detecting the High level of the signal input to the INT4 pin.
- RAM data is output to port 1 by performing DTP processing (EI^{2}OS).

● Coding example

```
ICR06   EQU  0000B6H          ; DTP/external interrupt control register
DDR1    EQU  000011H          ; Port 1 direction register
DDR5    EQU  000015H          ; Port 5 direction register
ENIR    EQU  000030H          ; DTP/external interrupt enable register
EIRR    EQU  000031H          ; DTP/external interrupt factor register
ELVRL   EQU  000032H          ; Detection level setting register: L
ELVRH   EQU  000033H          ; Detection level setting register: H
ER4     EQU  EIRR:0           ; INT4 interrupt request flag bit
EN4     EQU  ENIR:0           ; INT4 interrupt request enable bit
;
BAPL    EQU  000100H          ; Buffer address pointer lower
BAPM    EQU  000101H          ; Buffer address pointer middle
BAPH    EQU  000102H          ; Buffer address pointer higher
ISCS    EQU  000103H          ; EI2OS status register
IOAL    EQU  000104H          ; I/O address register lower
IOAH    EQU  000105H          ; I/O address register higher
DCTL    EQU  000106H          ; Data counter lower
DCTH    EQU  000107H          ; Data counter higher
;
;-----Main program-----------------------------------------------------------
CODE    CSEG
START:                        ; Stack pointer (SP) already initialized
        MOV  I:DDR1,#11111111B ; DDR1 set to output port
        MOV  I:DDR5,#00000000B ; DDR5 set to input port
        AND  CCR,#0BFH        ; Interrupts disabled
        MOV  I:ICR06,#08H     ; Interrupt level 0 (highest) EI2OS
                              ; Channel 0
;
; Deta bank register (DTB) = 00H
;
        MOV  BAPL,#00H        ; Address for storing output data set
        MOV  BAPM,#06H        ; (600H to 60AH used)
        MOV  BAPH,#00H
        MOV  ISCS,#12H        ; Byte transfer, buffer address + 1
                              ; I/O address fixed,
                              ; transfer from memory to I/O
        MOV  IOAL,#00H        ; Port 0 (PDR0) set as
        MOV  IOAH,#00H        ; transfer destination address pointer
        MOV  DCTL,#0AH        ; Transfer count set to 10
        MOV  DCTH,#00H
;
        CLRB I:EN4            ; INT4 disabled using ENIR
        MOV  I:ELVRL,#00010000B; H level detection set for INT4
        CLRB I:ER4            ; INT4 interrupt request flag cleared using EIRR
        SETB I:EN4            ; INT4 interrupt request enabled using ENIR
        MOV  ILM,#07H         ; ILM in PS set to level 7
        OR   CCR,#40H         ; Interrupts enabled
```

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com


```
LOOP:
         .
        Processing by user
         .
        BRA   LOOP
;-----Interrupt program-------------------------------------------------------
WARI:
        CLRB   I:ER4           ; INT4 interrupt request flag cleared
         .
        Processing by user
         .
        RETI                   ; Return from interrupt processing
CODE    ENDS
;-----Vector setting----------------------------------------------------------
VECT    CSEG ABS = 0FFH
        ORG  00FF9CH           ; Vector set to interrupt number #24 (18H)
        DSL  WARI
        ORG  00FFDCH           ; Reset vector set
        DSL  START
        DB   00H               ; Set to single-chip mode set
VECT    ENDS
        END  START
```

et4U.com

DataShee

DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 13

8-/10-BIT A/D CONVERTER

This chapter explains the functions and operation of 8-/10-bit A/D converter.

13.1 Overview of 8-/10-bit A/D Converter
13.2 Block Diagram of 8-/10-bit A/D Converter
13.3 Configuration of 8-/10-bit A/D Converter
13.4 Interrupt of 8-/10-bit A/D Converter
13.5 Explanation of Operation of 8-/10-bit A/D Converter
13.6 Precautions when Using 8-/10-bit A/D Converter

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.1 Overview of 8-/10-bit A/D Converter

The 8-/10-bit A/D converter converts the analog input voltage to a 8- or 10-bit digital value by using the RC sequential-comparison converter system.

- **An input signal can be selected from the input signals of the analog input pins for 8 channels.**
- **The start trigger can be selected from a software trigger, internal timer output, and an external trigger.**

■ Function of 8-/10-bit A/D Converter

The 8-/10-bit A/D converter converts the analog voltage (input voltage) input to the analog input pin into an 8- or 10-bit digital value (A/D conversion).

The 8-/10-bit A/D converter has the following functions:

- A/D conversion time is a minimum of 6.12 µs* per channel including sampling time.
- Sampling time is a minimum of 2.0 µs per channel.*
- RC sequential-comparison converter system with sample & hold circuit
- Setting of 8-bit or 10-bit resolution enabled
- Analog input pin can be used up to 8 channels.
- Generates interrupt request by storing A/D conversion results in A/D data register
- Starts EI^{2}OS if interrupt request generated. Use of the EI^{2}OS prevents data loss even at continuous conversion.
- Selects start trigger from software trigger, internal timer output, and external trigger (falling edge)

*: When the machine clock frequency operates at 16 MHz

■ Conversion Modes of 8-/10-bit A/D Converter

There are conversion modes of 8-/10-bit A/D converter as shown below:

Table 13.1-1 Conversion Modes of 8-/10-bit A/D Converter

Conversion Mode	Description
Single conversion mode	A/D conversion is performed sequentially from the start channel to the end channel. When A/D conversion for the end channel is terminated, it stops.
Continuous conversion mode	A/D conversion is performed sequentially from the start channel to the end channel. When A/D conversion for the end channel is terminated, it is continued after returning to the start channel.
Pause-conversion mode	A/D conversion is performed sequentially from the start channel to the end channel. When A/D conversion for the end channel is terminated, A/D conversion and pause are repeated after returning to the start channel.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

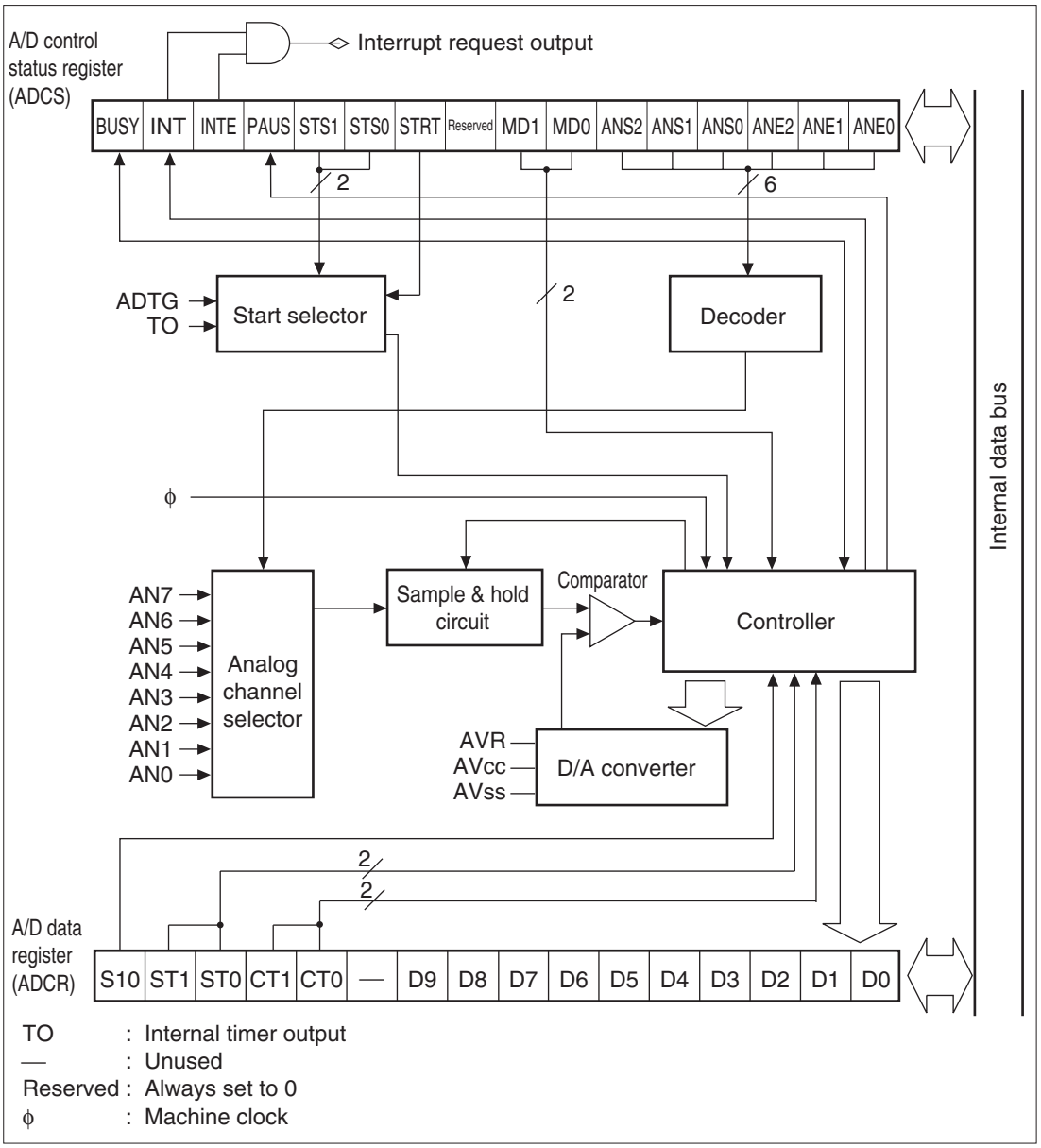
www.DataSheet4U.com

13.2 Block Diagram of 8-/10-bit A/D Converter

The 8-/10-bit A/D converter consists of following blocks.

■ Block Diagram of 8-/10-bit A/D Converter

Figure 13.2-1 Block Diagram of 8-/10-bit A/D Converter



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Details of pins in block diagram

Table 13.2-1 shows the actual pin names and interrupt request numbers of the 8-/10-bit A/D converter

Table 13.2-1 Pins and Interrupt Request Numbers in Block Diagram

Pin Name/Interrupt Request Number in Block Diagram		Actual Pin Name/Interrupt Request Number
ADTG	Trigger input pin	P37/ADTG
TO	Internal timer output	TO (16-bit reload timer, 16-bit free-run timer)
AN0	Analog input pin ch 0	P50/AN0
AN1	Analog input pin ch 1	P51/AN1
AN2	Analog input pin ch 2	P52/AN2
AN3	Analog input pin ch 3	P53/AN3
AN4	Analog input pin ch 4	P54/AN4
AN5	Analog input pin ch 5	P55/AN5
AN6	Analog input pin ch 6	P56/AN6
AN7	Analog input pin ch 7	P57/AN7
AVR	Vref+ Input pin	AVR
AV_{CC}	V_{CC} Input pin	AV_{CC}
AV_{SS}	V_{SS} Input pin	AV_{SS}
Interrupt request output		#18 (12_H)

- A/D control status registers (ADCS)

This register starts the A/D conversion function by software, selects the start trigger for the A/D conversion function, selects the conversion mode, enables or disables an interrupt request, checks and clears the interrupt request flag, temporarily stops A/D conversion and checks the state during conversion, and sets the start and end channels for A/D conversion.

- A/D data registers (ADCR)

This register stores the A/D conversion results, and selects the comparison time, sampling time, and resolution of A/D conversion.

- Start selector

This selector selects the trigger to start A/D conversion. An internal timer output or external pin input can be set as the start trigger.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Decoder

This decoder sets the A/D conversion start channel select bits and the A/D conversion end channel select bits in the A/D control status register (ADCS: ANS2 to ANS0 and ANE2 to ANE0) to select the analog input pin to be used for A/D conversion.

● Analog channel selector

This selector selects the pin to be used for A/D conversion from the 8-channel analog input pins by receiving a signal from the decoder.

● Sample & hold circuit

This circuit holds the input voltage selected by the analog channel selector. By holding the input voltage immediately after A/D conversion is started, A/D conversion is performed without being affected by the fluctuation of the input voltage during A/D conversion.

● D/A converter

This converter generates the reference voltage which is compared with the input voltage held in the sample & hold circuit.

● Comparator

This comparator compares the D/A converter output voltage with input voltage held in the sample & hold circuit to determine the mount of voltage.

DataSheet4U.com

● Controller

This circuit determines the A/D conversion value by receiving the signal indicating the amount of voltage determined by the comparator. When the A/D conversion results are determined, the result data is stored in the A/D data register. If an interrupt request is enabled, an interrupt is generated.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.3 Configuration of 8-/10-bit A/D Converter

This section explains the pins, registers, and interrupt factors of the A/D converter.

■ Pins of 8-/10-bit A/D Converter

The pins of the 8-/10-bit A/D converter serve as general-purpose I/O ports. Table 13.3-1 shows the pin functions and the setting required for use of the 8-/10-bit A/D converter.

Table 13.3-1 Pins of 8-/10-bit A/D Converter

Function Used	Pin Name	Pin Function	Setting Required for Use of 8-/10-bit A/D Converter
Trigger input	ADTG	General-purpose I/O port, external trigger input	Set as input port in port direction register (DDR).
Channel 0	AN0	General-purpose I/O ports, analog inputs	Set as input ports in port direction register (DDR). Input of analog signal enabled (ADER: ADE7 to ADE0 = 11111111_B)
Channel 1	AN1		
Channel 2	AN2		
Channel 3	AN3		
Channel 4	AN4		
Channel 5	AN5		
Channel 6	AN6		
Channel 7	AN7		

DataSheet4U.com

Reference: See CHAPTER 4 "I/O PORT" for the block diagram of pins.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ List of Registers and Reset Values of 8-/10-bit A/D Converter

Figure 13.3-1 Register and Reset Value of 8-/10-bit A/D Converter

	bit	7	6	5	4	3	2	1	0
A/D control status register (High) (ADCS: H)		0	0	0	0	0	0	0	0
	bit	15	14	13	12	11	10	9	8
A/D control status register (Low) (ADCS: L)		0	0	0	0	0	0	0	0
	bit	7	6	5	4	3	2	1	0
A/D data register (High) (ADCR: H)		0	0	1	0	1	X	X	X
	bit	15	14	13	12	11	10	9	8
A/D data register (Low) (ADCR: L)		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
Analog input enable register (ADER)		1	1	1	1	1	1	1	1

X: Undefined

■ Generation of Interrupt from 8-/10-bit A/D Converter

In the 8-/10-bit A/D converter, when the A/D conversion results are stored in the A/D data register (ADCR), the interrupt request flag bit in the A/D control status register (ADCS: INT) is set to 1. When an interrupt request is enabled (ADCS: INTE = 1), an interrupt is generated.

DataSheet4U.com
www.DataSheet4U.com

DataSh4U.com

www.DataSheet4U.com

13.3.1 A/D Control Status Register (High) (ADCS: H)

The A/D control status register (High) (ADCS: H) provides the following settings:

- **Starting A/D conversion function by software**
- **Selecting start trigger for A/D conversion**
- **Storing A/D conversion results in A/D data register to enable or disable interrupt request**
- **Storing A/D conversion results in A/D data register to check and clear interrupt request flag**
- **Pausing A/D conversion and checking state during conversion**

■ A/D Control Status Register (High) (ADCS: H)

Figure 13.3-2 A/D Control Status Register (High) (ADCS: H)

15	14	13	12	11	10	9	8
R/W	R/W	R/W	R/W	R/W	R/W	W	R/W

Reset value
00000000B

bit 8

Reserved	Reserved bit
0	Always set to "0"

bit 9

STRT	A/D conversion software start bit
0	Does not start A/D conversion
1	Starts A/D conversion

bit 11 bit 10

STS1	STS0	A/D conversion start trigger select bit
0	0	Starts software
0	1	Starts software or external trigger
1	0	Starts software or internal timer
1	1	Starts software, external trigger, or internal timer

bit 12

PAUS	Pause flag bit (This bit is enabled only when EI^{2}OS is used)
0	A/D conversion does not pause
1	A/D conversion pauses

bit 13

INTE	Interrupt request enable bit
0	Interrupt request disable
1	Interrupt request enable

bit 14

INT	Interrupt request flag bit	
	Read	Write
0	A/D conversion not terminated	Clear to "0"
1	A/D conversion terminated	No effect

bit 15

BUSY	A/D conversion-on flag bit	
	Read	Write
0	A/D conversion terminated (inactive state)	Terminates A/D conversion forcibly
1	A/D conversion in operation	No effect

R/W : Read/Write
□ : Reset value

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 13.3-2 Function of Each Bit of A/D Control Status Register (High) (ADCS: H)

Bit Name		Function
bit 8	Reserved: Reserved bit	Always set this bit to 0.
bit 9	STRT: A/D conversion software start bit	This bit starts the 8-/10-bit A/D converter by software. **When set to 1:** Starts 8-/10-bit A/D converter • If A/D conversion pauses in the pause-conversion mode, it is resumed by writing 1 to the STRT bit. **When set to 0:** Invalid. The state remains unchanged. **Read:** The byte/word command reads "1". The read-modify-write series commands read "0". **Note:** Do not perform forcible termination (BUSY = 0) and software start (STRT = 1) of the 8-/10-bit A/D converter simultaneously.
bit 10 bit 11	STS1, STS0: A/D conversion start trigger select bits	These bits select the trigger to start the 8-/10-bit A/D converter. If two or more start triggers are set (STS1, STS0 "00_B"), the 8-/10-bit A/D converter is started by the first-generated start trigger. **Note:** Start trigger setting should be changed when the operation of resource generating a start trigger is stopped.
bit 12	PAUS: Pause flag bit	This bit indicates the A/D conversion operating state when the EI^{2}OS function is used. • The PAUS bit is enabled only when the EI^{2}OS function is used. • A/D conversion pauses while the A/D conversion results are transferred from the A/D data register (ADCR) to memory. When A/D conversion pauses, the PAUS bit is set to 1. • After transfer of the A/D conversion results to memory, the 8-/10-bit A/D converter automatically resumes A/D conversion. When A/D conversion is started, the PAUS bit is cleared to 0.
bit 13	INTE: Interrupt request enable bit	This bit enables or disables output of an interrupt request. • When the interrupt request flag bit is set (INT=1) with an interrupt request enabled (INTE = 1), an interrupt request is generated. **Note:** Always set this bit to 1 when the EI^{2}OS function is used.
bit 14	INT: Interrupt request flag bit	This bit indicates that an interrupt request is generated. • When A/D conversion is terminated and its results are stored in the A/D data register (ADCR), the INT bit is set to 1. • When the interrupt request flag bit is set (INT = 1) with an interrupt request enabled (INTE = 1), an interrupt request is generated. **When set to 0:** Cleared **When set to 1:** No effect When EI^{2}OS function started: Cleared **Note:** To clear the INT bit, write 0 when the 8-/10-bit A/D converter is stopped.
bit 15	BUSY: A/D conversion-on flag bit	This bit forcibly terminates the 8-/10-bit A/D converter. When read, this bit indicates whether the 8-/10-bit A/D converter is operating or stopped. **When set to 0:** Forcibly terminates 8-/10-bit A/D converter **When set to 1:** No effect **Read:** 1 is read when the 8-/10-bit A/D converter is operating and 0 is written when the converter is stopped. **Note:** Do not perform forcibly termination (BUSY=0) and software start (STRT=1) of the 8-/10-bit A/D converter simultaneously.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.3.2 A/D Control Status Register (Low) (ADCS: L)

The A/D control status register (Low) (ADCS: L) provides the following settings:

- **Selecting A/D conversion mode**
- **Selecting start channel and end channel of A/D conversion**

■ A/D Control Status Register (Low) (ADCS: L)

Figure 13.3-3 A/D Control Status Register Low (ADCS: L)

7	6	5	4	3	2	1	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Reset value
00000000B

bit 2 bit 1 bit 0

ANE2	ANE1	ANE0	A/D conversion end channel select bits
0	0	0	AN0 pin
0	0	1	AN1 pin
0	1	0	AN2 pin
0	1	1	AN3 pin
1	0	0	AN4 pin
1	0	1	AN5 pin
1	1	0	AN6 pin
1	1	1	AN7 pin

bit 5 bit 4 bit 3

ANS2	ANS1	ANS0	A/D conversion start channel select bits		
			Inactive state	Read during a conversion	Read during a pause in stop conversion mode
0	0	0	AN0 pin	Channel number currently being converted	Channel number just previously converted
0	0	1	AN1 pin		
0	1	0	AN2 pin		
0	1	1	AN3 pin		
1	0	0	AN4 pin		
1	0	1	AN5 pin		
1	1	0	AN6 pin		
1	1	1	AN7 pin		

bit 7 bit 6

MD1	MD0	A/D conversion mode select bits
0	0	Single conversion mode 1 (restartable during conversion)
0	1	Single conversion mode 2 (not-restartable during conversion)
1	0	Continuous conversion mode (not-restartable during conversion)
1	1	Pause-conversion mode (not-restartable during conversion)

R/W : Read/Write
: Reset value

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 13.3-3 Function of Each Bit of A/D Control Status Register (Low) (ADCS: L) (1/2)

Bit Name		Function
bit 0 to bit 2	ANE2 to ANE0: A/D conversion end channel select bits	These bits set the channel at which A/D conversion is terminated. **Start channel < end channel:** A/D conversion starts at channel set by A/D conversion start channel select bits (ANS2 to ANS0) and terminates channel set by A/D conversion end channel select bits (ANE2 to ANE0) **Start channel = end channel:** A/D conversion is performed only for one channel set by A/D converter end (= start) channel select bits (ANE2 to ANE0 = ANS2 to ANS0). **Start channel > end channel:** A/D conversion is performed from channel set by A/D conversion start channel select bits (ANS2 to ANS0) to AN7, and from AN0 to channel set by A/D conversion end channel select bits (ANE2 to ANE0). **Continuous conversion mode and pause-conversion mode:** When A/D conversion terminated at the channel set by the A/D conversion end channel select bits (ANE2 to ANE0), it returns to the channel set by the A/D conversion start channel select bits (ANS2 to ANS0). **Note:** Do not set the A/D conversion end channel select bits (ANE2 to ANE0) during A/D conversion.
bit 3 to bit 5	ANS2 to ANS0: A/D conversion start channel select bits	These bits set the channel at which A/D conversion start. At read, the channel number under A/D conversion or A/D-converted immediately before A/D conversion pauses can be checked. And before A/D conversion starts, the previous conversion channel number will be read even if these bits have been already set to the new value. These bits are initialized to "000_B" at reset. **Start channel < end channel:** A/D conversion starts at channel set by A/D conversion start channel select bits (ANS2 to ANS0) and terminates at channel set by A/D conversion end channel select bits (ANE2 to ANE0). **Start channel = end channel:** A/D conversion is performed only for one channel set by A/D conversion (= end) channel select bits (ANS2 to ANS0 = ANE2 to ANE0). **Start channel > end channel:** A/D conversion is performed from channel set by A/D conversion start channel select bits (ANS2 to ANS0) to AN7, and from AN0 to channel set by A/D conversion end channel select bits (ANE2 to ANE0). **Continuous conversion mode and pause-conversion mode:** When A/D conversion terminates at the channel set by the A/D conversion end channel select bits (ANE2 to ANE0), it returns to the channel set by the A/D conversion start channel select bits (ANS2 to ANS0). **Read (During A/D conversion):** The channel numbers (7 to 0) under A/D conversion are read. **Read (During Pause-conversion mode and temporary stop):** At read during a pause, the channel number A/D-converted immediately before a pause is read. **Note:** Do not set the A/D conversion start channel bits (ANS2 to ANS0) during A/D conversion.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 13.3-3 Function of Each Bit of A/D Control Status Register (Low) (ADCS: L) (2/2)

Bit Name		Function
bit 6 bit 7	MD1, MD0: A/D conversion mode select bits	These bits set the A/D conversion mode. **Single conversion mode 1:** • The analog inputs from the start channel (ADCS: ANS2 to ANS0) to the end channel (ADCS: ANE2 to ANE0) are A/D-converted continuously. • The A/D conversion pauses after A/D conversion for the end channel. • This mode can be restarted during A/D conversion. **Single conversion mode 2:** • The analog inputs from the start channel (ADCS: ANS2 to ANS0) to the end channel (ADCS: ANE2 to ANE0) are A/D-converted continuously. • The A/D conversion after A/D conversion for the end channel. • This mode cannot be restarted during A/D conversion. **Continuous conversion mode:** • The analog inputs from the start channel (ADCS: ANS2 to ANS0) to the end channel (ADCS: ANE2 to ANE0) are A/D-converted continuously. • When A/D conversion for the end channel is terminated, it is continued after returning to the analog input for the start channel. • To terminate A/D conversion forcibly, write 0 to the A/D conversion-on flag bit in the A/D control status register (ADCS: BUSY). • This mode cannot be restarted during A/D conversion. **Pause conversion mode:** • A/D conversion for the start channel (ADCS: ANS2 to ANS0) starts. The A/D conversion pauses at termination of A/D conversion for a channel. When the start trigger is input while A/D conversion pauses, A/D conversion for the next channel is started. • The A/D conversion pauses at the termination of A/D conversion for the end channel. When the start trigger is input while A/D conversion pauses, A/D conversion is continued after returning to the analog input for the start channel. • To terminate A/D conversion forcibly, write 0 to the A/D conversion-on flag bit in the A/D control status register (ADCS: BUSY). • This mode cannot be restarted during A/D conversion. **Note:** When the conversion mode is set to "not restartable" (MD1, MD0 "00_B"), it cannot be restarted with any start triggers (software trigger, internal timer, and external trigger) during A/D conversion.

DataSheet4U.com

DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

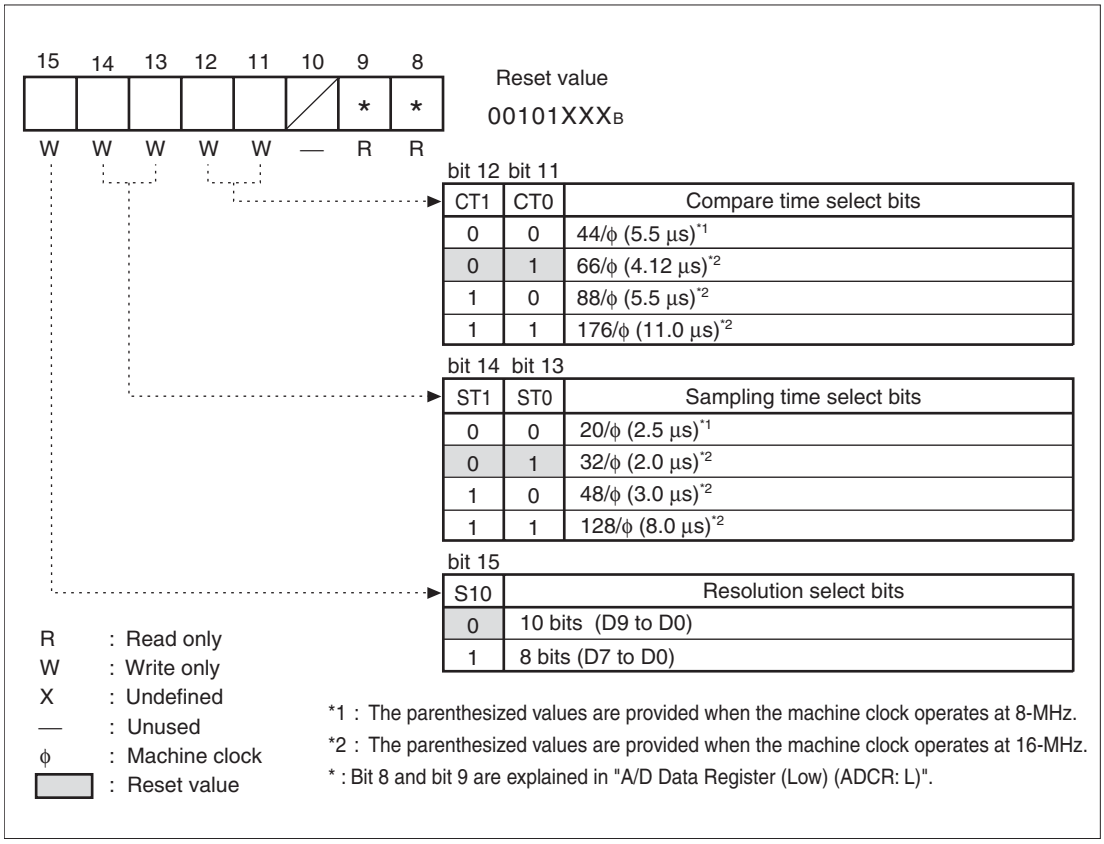
www.DataSheet4U.com

13.3.3 A/D Data Register (High) (ADCR: H)

The higher five bits in the A/D data register (ADCR: H) select the compare time, sampling time and resolution of A/D conversion.
Bits 9 and 8 in the A/D data register (ADCR) are explained in Section "13.3.4 A/D Data Register (Low) (ADCR: L)".

■ A/D Data Register (High) (ADCR: H)

Figure 13.3-4 A/D Data Register (High) (ADCR: H)



CT1	CT0	Compare time select bits
0	0	44/ϕ (5.5 μs)*1
0	1	66/ϕ (4.12 μs)*2
1	0	88/ϕ (5.5 μs)*2
1	1	176/ϕ (11.0 μs)*2

ST1	ST0	Sampling time select bits
0	0	20/ϕ (2.5 μs)*1
0	1	32/ϕ (2.0 μs)*2
1	0	48/ϕ (3.0 μs)*2
1	1	128/ϕ (8.0 μs)*2

S10	Resolution select bits
0	10 bits (D9 to D0)
1	8 bits (D7 to D0)

*1 : The parenthesized values are provided when the machine clock operates at 8-MHz.
*2 : The parenthesized values are provided when the machine clock operates at 16-MHz.
* : Bit 8 and bit 9 are explained in "A/D Data Register (Low) (ADCR: L)".

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 13.3-4 Functions of A/D Data Register (High) (ADCR: H)

Bit Name		Function
bit 11 bit 12	CT1, CT0: Compare time select bits	These bits set the A/D conversion compare time. • These bits set the time required from when analog input is A/D-converted until it is stored in the data bits (D9 to D0). **Note:** The setting of CT1 and CT0 = "00_B" is based on operation at 8 MHz. Setting based on operation at 16 MHz does not assure normal operation. When these bits are read, "00_B" is read.
bit 13 bit 14	ST1, ST0: Sampling time select bits	These bits set the A/D conversion sampling time. • These bits set the time required from when A/D conversion starts until the input analog voltage is sampled and held by the sample & hold circuit. **Note:** The setting of ST1 and ST0 = "00_B" is based on operation at 8 MHz. Setting based on operation at 16 MHz does not assure normal operation. When these bits are read, "00_B" is read.
bit 15	S10: Resolution select bit	This bit selects the A/D conversion resolution. **When set to 0:** Sets A/D conversion resolution in 10 bits from A/D conversion data bits D9 to D0. **When set to 1:** Sets A/D conversion resolution in 8 bits from A/D conversion data bits D7 to D0. **Note:** Change the S10 bit in the pausing state before starting A/D conversion. Changing the S10 bit after A/D conversion starts disables the A/D conversion results stored in the A/D conversion data bits (D9 to D0).

et4U.com DataShee DataSheet4U.com

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.3.4 A/D Data Register (Low) (ADCR: L)

**The A/D data register (Low) (ADCR: L) stores the A/D conversion results.
Bits 8 and 9 in the A/D data register (ADCR) are explained in this section.**

■ A/D Data Register (Low) (ADCR: L)

Figure 13.3-5 A/D Data Register (Low) (ADCR: L)

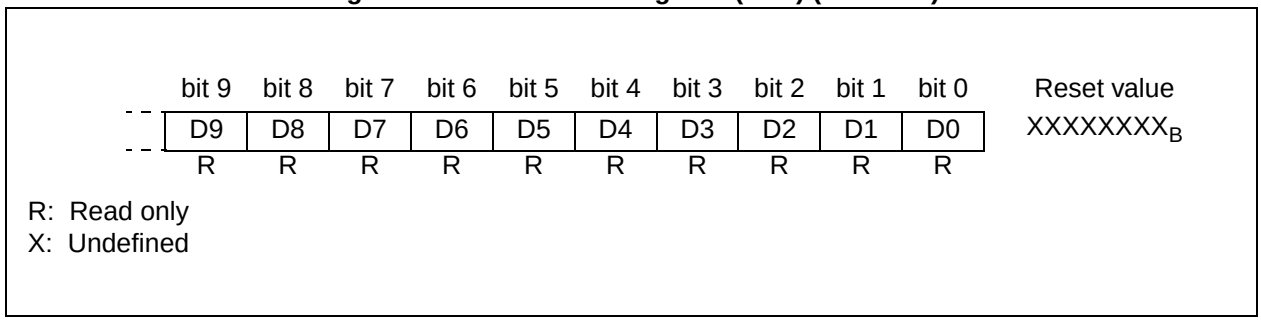


Table 13.3-5 Functions of A/D Data Register (Low) (ADCR: L)

Bit Name		Function
bit 0 to bit 9	D9 to D0: A/D conversion data bits	These bits store the A/D conversion results. **When resolution set in 10 bits (S10 = 0):** Conversion data is stored in the 10 bits from D9 to D0. **When resolution set in 8 bits (S10 = 1):** Conversion data is stored in the 8 bits from D7 to D0. **Note:** Use a word instruction (MOVW) to read the A/D conversion results stored in the A/D conversion data bits (D9 to D0).

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

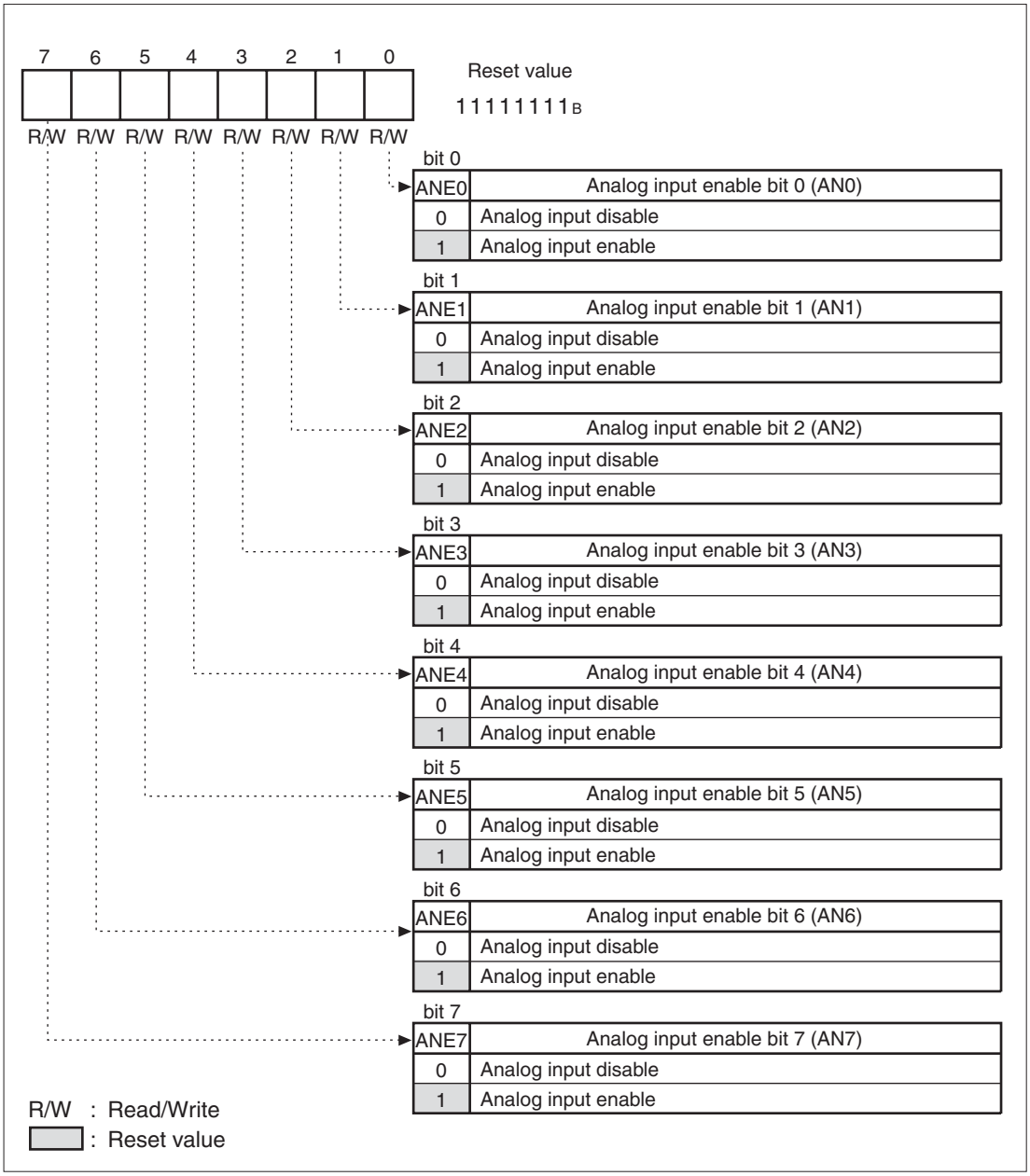
www.DataSheet4U.com

13.3.5 Analog Input Enable Register (ADER)

The analog input enable register (ADER) enables or disables the analog input pins to be used in the 8-/10-bit A/D converter.

■ Analog Input Enable Register (ADER)

Figure 13.3-6 Analog Input Enable Register (ADER)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 13.3-6 Functions of Analog Input Enable Register (ADER)

Bit Name		Function
bit 0 to bit 7	ADE7 to ADE0: Analog input enable bits	These bits enable or disable the analog input of the pin to be used for A/D conversion. **When set to 0:** Disables analog input **When set to 1:** Enables analog input

Notes:
- The analog input pins serve as a general-purpose I/O port of the port 5. When using the pin as an analog input pin, switch the pin to analog input pin according to the setting of the port 5 direction register (DDR5) and the analog input enable register (ADER).
- When using the pin as an analog input pin, write 0 to the bit in the port 5 direction register (DDR5) corresponding to the pin to be used and turn off the output transistor. Also write 1 to the bit in the analog input enable register (ADER) corresponding to the pin to be used and set the pin to analog input.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.4 Interrupt of 8-/10-bit A/D Converter

When A/D conversion is terminated and its results are stored in the A/D data register (ADCR), the 8-/10-bit A/D converter generates an interrupt request. The EI^{2}OS function can be used.

■ Interrupt of A/D Converter

When A/D conversion of the analog input voltage is terminated and its results are stored in the A/D data register (ADCR), the interrupt request flag bit in the A/D control status register (ADCS: INT) is set to 1. When the interrupt request flag bit is set (ADCS: INT = 1) with an interrupt request output enabled (ADCS: INTE = 1), an interrupt request is generated.

■ 8-/10-bit A/D Converter Interrupt and EI^{2}OS

See Section "3.5 Interrupt" for details of the interrupt number, interrupt control register, and interrupt vector address.

■ EI^{2}OS Function of 8-/10-bit A/D Converter

In the 8-/10-bit A/D converter, the EI^{2}OS function can be used to transfer the A/D conversion results from the A/D data register (ADCR) to memory. If the EI^{2}OS function is used, the A/D-converted data protection function is activated to cause A/D conversion to pause during memory transfer and prevent data loss as A/D conversion is performed continuously.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.5 Explanation of Operation of 8-/10-bit A/D Converter

The 8-/10-bit A/D converter has the following A/D conversion modes. Set each mode according to the setting of the A/D conversion mode select bits in the A/D control status register (ADCS: MD1, MD0).

- **Single conversion mode (restartable/not-restartable during A/D conversion)**
- **Continuous conversion mode (not-restartable during A/D conversion)**
- **Pause conversion mode (not-restartable during A/D conversion)**

■ Single Conversion Mode (ADCS: MD1, MD0 = "00_B" or "01_B")

- When the start trigger is input, the analog inputs from the start channel (ADCS: ANS2 to ANS0) to the end channel (ADCS: ANE2 to ANE0) are A/D-converted continuously.
- The A/D conversion stops at the termination of the A/D conversion for the end channel.
- To terminate A/D conversion forcibly, write 0 to the A/D conversion-on flag bit in the A/D control status register (ADCS: BUSY).
- When the A/D conversion mode select bits (MD1, MD0) are set to "00_B", this mode can be restarted during A/D conversion. If the bits are set to 01_B, this mode cannot be restarted during A/D conversion.

■ Continuous Conversion Mode (ADCS: MD1, MD0 = "10_B")

- When the start trigger is input, the analog inputs from the start channel (ADCS: ANS2 to ANS0) to the end channel (ADCS: ANE2 to ANE0) are A/D-converted continuously.
- When A/D conversion for the end channel is terminated, it is continued after returning to the analog input for the start channel.
- To terminate A/D conversion forcibly, write 0 to the A/D conversion-on flag bit in the A/D control status register (ADCS: BUSY).
- This mode cannot be restarted during A/D conversion.

■ Pause-conversion Mode (ADCS: MD1, MD0 = "11_B")

- When the start trigger is input, A/D conversion starts for the start channel (ADCS: ANS2 to ANS0). The A/D conversion pauses at the termination of A/D conversion for one channel. When the start trigger is input while A/D conversion pauses, A/D conversion is performed for the next channel.
- The A/D conversion pauses at termination of A/D conversion for the end channel. When the start trigger is input while A/D conversion pauses, A/D conversion is continued after returning to the analog input for the start channel.
- To terminate A/D conversion forcibly, write 0 to the A/D conversion-on flag bit in the A/D control status register (ADCS: BUSY).
- This mode cannot be restarted during A/D conversion.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.5.1 Single Conversion Mode

In the Single conversion mode, A/D conversion is performed sequentially from the start channel to the end channel. The A/D conversion stops at the termination of A/D conversion for the end channel.

■ Setting of Single Conversion Mode

Operating the 8-/10-bit A/D converter in the Single conversion mode requires the setting shown in Figure 13.5-1.

Figure 13.5-1 Setting of Single Conversion Mode

	bit15	14	13	12	11	10	9	bit 8	bit 7	6	5	4	3	2	1	bit0
ADCS	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	Re-served	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0
	◎	◎	◎	◎	◎	◎	◎	0	0	◎	◎	◎	◎	◎	◎	◎
ADCR	S10	ST1	ST0	CT1	CT0	–	D9 to D0(Converted data stored)									
	◎	◎	◎	◎	◎											
ADER									○	○	○	○	○	○	○	○

–: Unused
◎: Used bit
○: Set the bit corresponding to pin to be used as analog input pin to 1.
0: Set 0

DataSheet4U.com

■ Operation of Single Conversion Mode

- When the start trigger is input, A/D conversion starts from the channel set by the A/D conversion start channel select bits (ANS2 to ANS0) and is performed continuously up to the channel set by the A/D conversion end channel select bits (ANE2 to ANE0).
- The A/D conversion stops at the termination of the A/D conversion for the channel set by the A/D conversion end channel select bits (ANE2 to ANE0).
- To terminate A/D conversion forcibly, write 0 to the A/D conversion-on flag bit in the A/D control status register (ADCS: BUSY).
- When the A/D conversion mode select bits (MD1, MD0) are set to "00_B", this mode can be restarted during A/D conversion. If the bits are set to "01_B", this mode cannot be restarted during A/D conversion.

[When start and end channels are the same]

- If the start and end channels have the same channel number (ADCS: ANS2 to ANS0 = ADCS: ANE2 to ANE0), only one A/D conversion for one channel set as the start channel (= end channel) is performed and terminated.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

[Conversion order in single conversion mode]

Table 13.5-1 gives an example of the conversion order in the single conversion mode.

Table 13.5-1 Conversion Order in Single Conversion Mode

Start Channel	End Channel	Conversion Order
AN0 pin (ADCS: ANS = "000_B")	AN3 pin (ADCS: ANE = "011_B")	AN0 --> AN1 --> AN2 --> AN3 --> End
AN6 pin (ADCS: ANS = "110_B")	AN2 pin (ADCS: ANE = "010_B")	AN6 --> AN7 --> AN0 --> AN1 --> AN2 --> End
AN3 pin (ADCS: ANS = "011_B")	AN3 pin (ADCS: ANE = "011_B")	AN3 --> End

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.5.2 Continuous Conversion Mode

In the continuous conversion mode, A/D conversion is performed sequentially from the start channel to the end channel.
When A/D conversion for the end channel is terminated, it is continued after returning to the start channel.

■ Setting of Continuous Conversion mode

Operating the 8-/10-bit A/D converter in the continuous conversion mode requires the setting shown in Figure 13.5-2.

Figure 13.5-2 Setting of Continuous Conversion Mode

	bit15	14	13	12	11	10	9	bit 8	bit 7	6	5	4	3	2	1	bit0
ADCS	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	Re-served	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0
	◎	◎	◎	◎	◎	◎	◎	0	1	0	◎	◎	◎	◎	◎	◎

ADCR	S10	ST1	ST0	CT1	CT0	–	D9 to D0 (Converted data stored)
	◎	◎	◎	◎	◎		

ADER								
	○	○	○	○	○	○	○	○

DataSheet4U.com

–: Unused
◎: Used bit
○: Set the bit corresponding to pin to be used as analog input pin to 1.
1: Set 1
0: Set 1

■ Operation of Continuous Conversion Mode

- When the start trigger is input, A/D conversion starts from the channel set by the A/D conversion start channel select bits (ANS2 to ANS0) and is performed continuously up to the channel set by the A/D conversion end channel select bits (ANE2 to ANE0).
- When A/D conversion for the channel set by the A/D conversion end channel select bits (ANE2 to ANE0) is terminated, it is continued after returning to the channel set by the A/D conversion start channel select bits (ANS2 to ANS0).
- To terminate A/D conversion forcibly, write 0 to the A/D conversion-on flag bit in the A/D control status register (ADCS: BUSY).
- This mode cannot be restarted during A/D conversion.

[When start and end channels are the same]

- If the start and end channels have the same channel number (ADCS: ANS2 to ANS0 = ADCS: ANE2 to ANE0), A/D conversion for one channel set as the start channel (= end channel) is repeated.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

[Conversion order in continuous conversion mode]

Table 13.5-2 gives an example of the conversion order in the continuous conversion mode.

Table 13.5-2 Conversion Order in Continuous Conversion Mode

Start Channel	End Channel	Conversion Order
AN0 pin (ADCS: ANS = "000_B")	AN3 pin (ADCS: ANE = "011_B")	AN0 --> AN1 --> AN2 --> AN3 --> AN0 --> Repeat
AN6 pin (ADCS: ANS = "110_B")	AN2 pin (ADCS: ANE = "010_B")	AN6 --> AN7 --> AN0 --> AN1 --> AN2 --> AN6 --> Repeat
AN3 pin (ADCS: ANS = "011_B")	AN3 pin (ADCS: ANE = "011_B")	AN3 --> AN3 --> Repeat

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.5.3 Pause-conversion Mode

In the pause-conversion mode, A/D conversion starts and pauses repeatedly for each channel. When the start trigger is input after the A/D conversion pauses at the termination of the A/D conversion for the end channel, A/D conversion is continued after returning to the start channel.

■ Setting of Pause-conversion Mode

Operating the 8-/10-bit A/D converter in the pause-conversion mode requires the setting shown in Figure 13.5-3.

Figure 13.5-3 Setting of Pause-conversion Mode

	bit15	14	13	12	11	10	9	bit 8	bit 7	6	5	4	3	2	1	bit0
ADCS	BUSY	INT	INTE	PAUS	STS1	STS0	STRT	Re-served	MD1	MD0	ANS2	ANS1	ANS0	ANE2	ANE1	ANE0
	◎	◎	◎	◎	◎	◎	◎	0	1	1	◎	◎	◎	◎	◎	◎

ADCR	S10	ST1	ST0	CT1	CT0	–	D9 to D0 (Converted data stored)
	◎	◎	◎	◎	◎		

ADER								
	○	○	○	○	○	○	○	○

DataSheet4U.com

–: Unused
◎: Used bit
○: Set the bit corresponding to pin to be used as analog input pin to 1.
1: Set 1
0: Set 1

■ Operation of Pause-conversion Mode

- When the start trigger is input, A/D conversion starts at the channel set by the A/D conversion start channel select bits (ANS2 to ANS0). The A/D conversion pauses at the termination of the A/D conversion for one channel. When the start trigger is input while A/D conversion pauses, A/D conversion for the next channel is performed.
- The A/D conversion pauses at the termination of the A/D conversion for the channel set by the A/D conversion end channel select bits (ANE2 to ANE0). When the start trigger is input while A/D conversion pauses, A/D conversion is continued after returning to the channel set by the A/D conversion start channel select bits (ANS2 to ANS0).
- To restart this mode while A/D conversion pauses, input the start trigger set by the A/D start trigger select bits in the A/D control status register (ADCS: STS1, STS0).
- To terminate A/D conversion forcibly, write 0 to the A/D conversion-on flag bit in the A/D control status register (ADCS: BUSY).
- This mode cannot be restarted during A/D conversion.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

[When start and end channels are the same]

- If the start and end channels have the same channel number (ADCS: ANS2 to ANS0 = ADCS: ANE2 to ANE0), A/D conversion for one channel set as the start channel (= end channel), and pause are repeated.

[Conversion order in pause-conversion mode]

Table 13.5-3 gives an example of the conversion order in the pause-conversion mode.

Table 13.5-3 Conversion Order in Pause-conversion Mode

Start Channel	End Channel	Conversion Order
AN0 pin (ADCS: ANS = "000_B")	AN3 pin (ADCS: ANE="011_B")	AN0 --> Stop, Start --> AN1 --> Stop, Start --> AN2 --> Stop, Start --> AN3 --> Stop, Start --> AN0 --> Repeat
AN6 pin (ADCS: ANS = "110_B")	AN2 pin (ADCS: ANE="010_B")	AN6 --> Stop, Start --> AN7 --> Stop, Start --> AN0 --> Stop, Start --> AN1 --> Stop, Start --> AN2 --> Stop, Start --> AN6 --> Repeat
AN3 pin (ADCS: ANS = "011_B")	AN3 pin (ADCS: ANE="011_B")	AN3 --> Stop, Start --> AN3 --> Stop, Start --> Repeat

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.5.4 Conversion Using EI^{2}OS Function

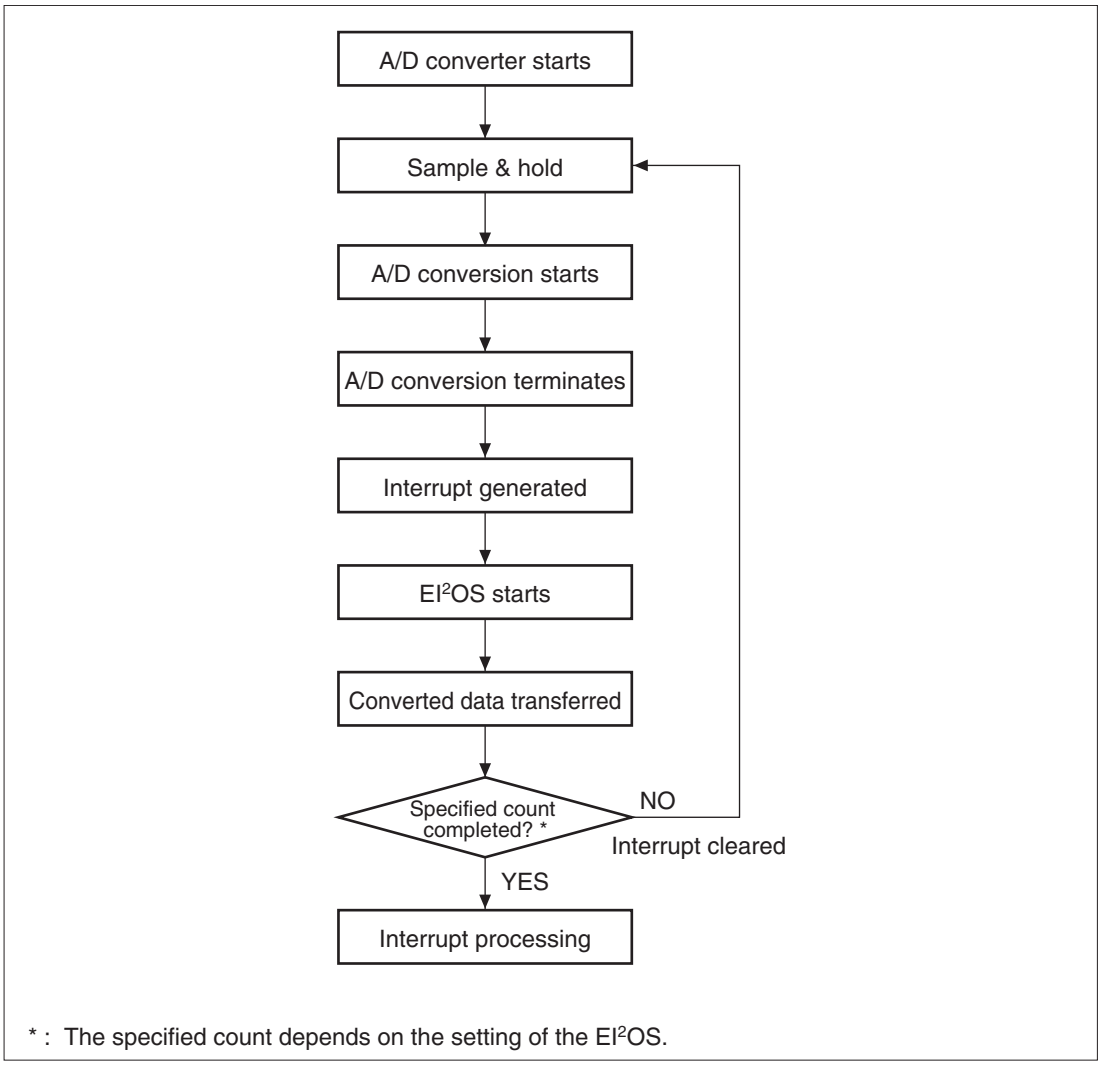
The 8-/10-bit A/D converter can transfer the A/D conversion result to memory by using the EI^{2}OS function.

■ Conversion Using EI^{2}OS

The use of the EI^{2}OS enables the A/D-converted data protection function to transfer multiple data to memory without the loss of converted data even if A/D conversion is performed continuously.

The conversion flow when the EI^{2}OS is used is shown in Figure 13.5-4.

Figure 13.5-4 Flow of Conversion when Using EI^{2}OS



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.5.5 A/D-converted Data Protection Function

A/D conversion with the output an interrupt request enabled activates the A/D conversion data protection function.

■ A/D-converted Data Protection Function in 8-/10-bit A/D Converter

The 8-/10-bit A/D converter has only one A/D data register (ADCR) where A/D-converted data is stored. When the A/D conversion results are determined after the termination of A/D conversion, data in the A/D data register is rewritten. Therefore, the A/D conversion results may be lost if the A/D conversion results already stored are not read before data in the A/D data register is rewritten. The A/D-converted data protection function in the 8-/10-bit A/D converter is activated to prevent data loss. This function automatically causes A/D conversion to pause when an interrupt request is generated (ADCS: INT = 1) with an interrupt request enabled (ADCS: INTE = 1).

● A/D-converted data protection function when EI^{2}OS not used

- When the A/D conversion results are stored in the A/D data register (ADCR) after the analog input is A/D-converted, the interrupt request flag bit in the A/D control status register (ADCS: INT) is set to 1.
- A/D conversion pauses for data protection while the interrupt request flag bit in the A/D control status register (ADCS: INT) is set.
- When the INT bit is set with an interrupt request from the A/D control status register enabled (ADCS: INTE = 1), an interrupt request is generated. When the INT bit is cleared by the generated interrupt processing, the pause of A/D conversion is cancelled.

● A/D-converted data protection function when EI^{2}OS used

- A/D conversion pauses for data protection while the EI^{2}OS function is used to transfer the A/D conversion results to memory from the A/D data register after A/D conversion. When A/D conversion pauses, the pause flag bit in the A/D control status register (ADCS: PAUS) is set to 1.
- When the transfer of the A/D conversion results to memory by the EI^{2}OS function is terminated, the pause of A/D conversion is cancelled and the pause flag bit (ADCS: PAUS) is cleared to 0. If A/D conversion is performed continuously, it is restarted.

● Processing flow of A/D conversion data protection function when EI^{2}OS used

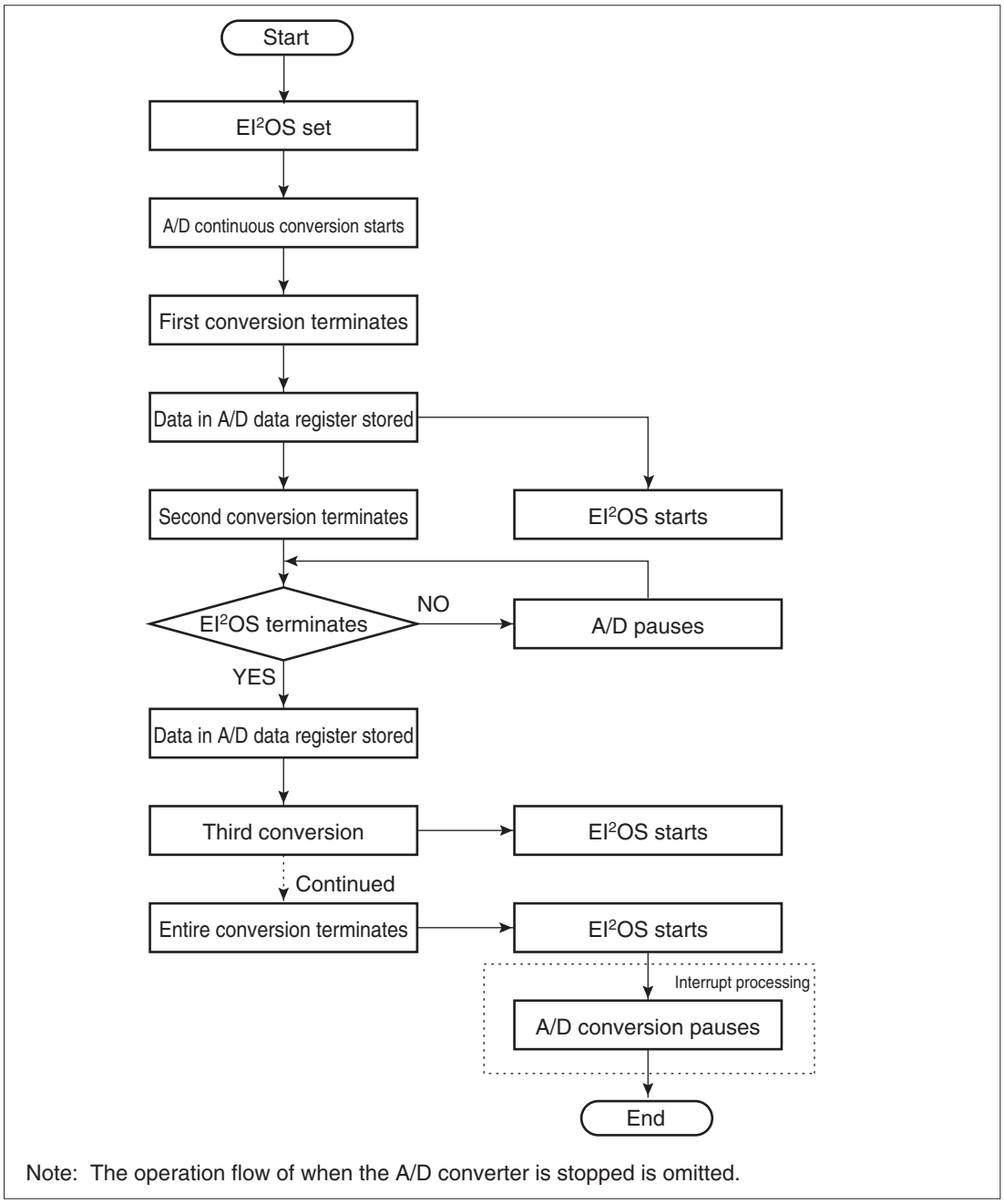
Figure 13.5-5 shows the processing flow of the A/D conversion data protection function when the EI^{2}OS is used.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 13.5-5 Processing Flow of A/D Conversion Data Protection Function when Using EI^{2}OS



Note: The operation flow of when the A/D converter is stopped is omitted.

Notes:
- The A/D conversion data protection function is activated only when an interrupt request is enabled. Set the interrupt request enable bit in the A/D control status register (ADCS: INTE) to 1.
- When the EI^{2}OS function is used to transfer the A/D conversion results to memory, do not disable output of an interrupt request. If output of an interrupt request is disabled during a pause of A/D conversion (ADCS: INTE = 0), A/D conversion may be restarted to rewrite data being transferred.
- When the EI^{2}OS function is used to transfer the A/D conversion results to memory, do not restart. Restarting during a pause of A/D conversion may cause loss of the A/D conversion results.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

13.6 Precautions when Using 8-/10-bit A/D Converter

Precautions when using the 8-/10-bit A/D converter are given below:

■ Precautions when Using 8-/10-bit A/D Converter

● Analog input pin

- The analog input pins serve as general-purpose I/O ports of port 5. When using the pin as an analog input pin, switch the pin to "analog input pin" according to the setting of the port 5 direction register (DDR5) and the analog input enable register (ADER).
- When using the pin as an analog input pin, write 0 to the bit in the DDR5 corresponding to the pin to be used and turn off the output transistor. Also write 1 to the bit in the ADER corresponding to the pin to be used and set the pin to "analog input enable."
- When an intermediate-level signal is input with the pin set as a general-purpose I/O port, the input leakage current flows in the gate. When using the pin as an analog input pin, always set the pin to "analog input enable".

● Precaution when starting by internal timer or external trigger

- The input value at which the 8-/10-bit A/D converter is started by the internal timer output or external trigger should be set to "inactive" (High for external trigger). Holding the input value for the start trigger active may cause the 8-/10-bit A/D converter to start concurrently with the setting of the A/D start trigger select bits in the A/D control status register (ADCS: STS1, STS0).

● Procedure of 8-/10-bit A/D converter and analog input power-on

- Always apply a power to the A/D converter power (AV_{CC}, AVR) and the analog input (AN0 to AN7) after or concurrently with the digital power (V_{CC})-on.
- Always turn off the A/D converter power and the analog input before or concurrently with the digital power (V_{CC})-down. Note that AVR should not exceed AV_{CC} at power on or power down. (There is no problem to turn on or off the analog power and digital power concurrently.)

● Power supply voltage of 8-/10-bit A/D converter

- To prevent latch up, note that the 8-/10-bit A/D converter power (AV_{CC}) should not exceed the digital power (V_{CC}) voltage.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 14 UART1

This chapter explains the function and operation of the UART.

14.1 Overview of UART1
14.2 Block Diagram of UART1
14.3 Configuration of UART1
14.4 Interrupt of UART1
14.5 Baud Rate of UART1
14.6 Explanation of Operation of UART1
14.7 Precautions when Using UART1
14.8 Program Example for UART1

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.1 Overview of UART1

The UART1 is a general-purpose serial-data communication interface for synchronous or asynchronous communication with external devices.

- **Incorporates a bidirectional communication function (clock synchronous and asynchronous modes)**
- **Incorporates a master/slave type communication function (in multiprocessor mode: only master)**
- **Capable of generating an interrupt request at completion of transmit completion and receive completion, and at detection of a receive error**
- **Supports expansion intelligent I/O service (EI^{2}OS)**

■ Function of UART1

● Function of UART1

The UART1 is a general-purpose serial-data communication interface, which transmits/receives serial data with external devices. UART1 has functions listed in Table 14.1-1.

Table 14.1-1 Function of UART1

	Function
Data buffer	Full-duplicate double-buffer
Transfer mode	• Synchronous to clock (without start bit/stop bit and parity bit) • Asynchronous (start-stop synchronization to clock)
Baud rate	• Dedicated baud-rate generator (The baud rate can be selected from among eight types.) • Any baud rate can be set by external clock. • A clock supplied from the internal clock (16-bit reload timer 1) can be used.
Data length	• 7 bits (for asynchronous normal mode only) • 8 bits
Signal type	NRZ (Non Return to Zero) type
Detection of receive error	• Framing error • Overrun error • Parity error (not supported for operation mode 1)
Interrupt request	• Receive interrupt (receive, detection of receive error) • Transmit interrupt (transmit) • Both the transmission and reception support EI^{2}OS.
Master/slave type communication function (asynchronous multiprocessor mode)	This function enables communications between 1 (only use master) and n (slave) (This function is used only as the master side)

Note: At the clock synchronous transfer, the UART only transfers data, not affixing the start and stop bits.

DataSheet4U.com

www.DataSheet4U.com

Table 14.1-2 Operation Mode of UART1

Operation Mode		Data Length		Synchronous/ Asynchronous	Length of Stop Bit
		With Parity	No Parity		
0	Asynchronous mode (Normal mode)	7 or 8 bits		Asynchronous	1 bit or 2 bits *2
1	Multiprocessor mode	8 + 1 *1	–	Asynchronous	
2	Synchronous mode	8	–	Synchronous	None

–: Setting disabled
*1: +1 is the address/data select bit (SCR1 register bit 11: A/D) used for controlling communications.
*2: During reception, only one bit can be detected as the stop bit.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

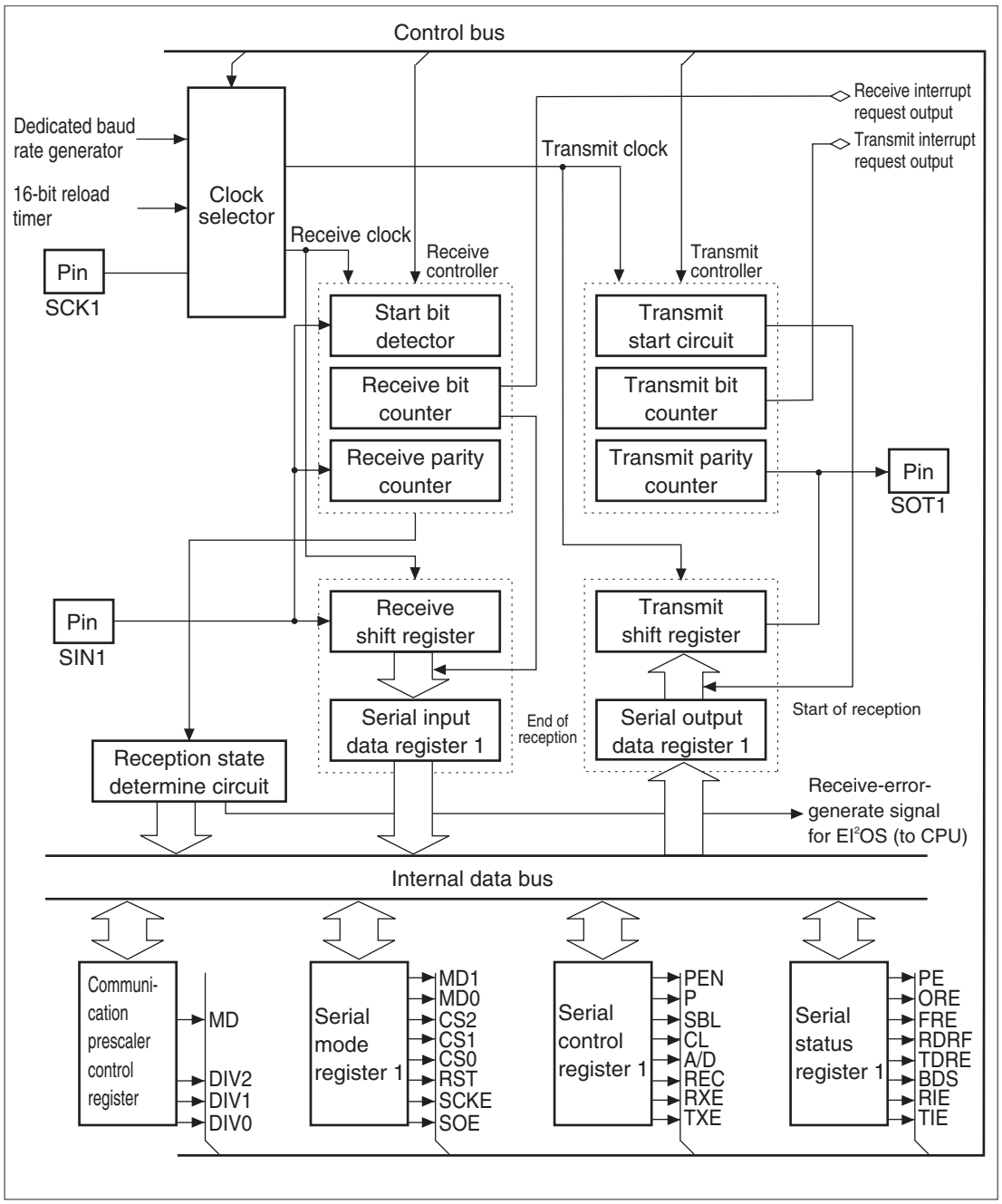
www.DataSheet4U.com

14.2 Block Diagram of UART1

The UART1 consists of the following block.

■ Block Diagram of UART1

Figure 14.2-1 Block Diagram of UART1



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Details of pins, etc., in block diagram

The actual pin names and interrupt request numbers used in the UART1 are as follows:

SIN1 pin: P40/SIN1

SCK1 pin: P41/SCK1

SOT1 pin: P42/SOT1

Transmit interrupt number 1: #38 (26_H)

Receive interrupt number 1: #37 (25_H)

- Clock selector

The clock selector selects the transmit/receive clock from the dedicated baud rate generator, external input clock, and internal clock (clock supplied from 16-bit reload timer).

- Receive controller

The receive controller is composed of receive bit counter, start bit detector and receive parity counter. The receive bit counter counts the receive data, and outputs a receive interrupt request when reception of one piece of data is completed according to the specified data length.

The start bit detector detects the start bit from the serial input signal and writes the received data to the serial input data register (SIDR1), on a bit-by-bit shift basis in accordance with the transfer rate.

The receive parity counter detects parity bit of the receive data.

et4U.com DataShee

- Transmit controller

DataSheet4U.com

The transmit controller is composed of the transmit bit counter, transmit start circuit, and transmit parity counter. The transmit bit counter counts the transmit data, and outputs a transmit interrupt request when transmission of one piece of data is completed according to the set data length. The transmit start circuit starts transmission when serial output data register (SODR1) is written. The transmit parity counter generates the parity bit of the data transferred when parity is provided.

- Receive shift register

The receive shift register writes the receive data input from the SIN pin while shifting bit-by-bit, and when the data reception is completed, it transfers the receive data to the serial input data register (SIDR1).

- Transmit shift register

Data written to SODR1 is transferred to the transmit shift register itself, and then the data is output to the SOT pin while shifting bit-by-bit.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Serial mode register 1 (SMR1)

  This register:

  - Selects operation mode
  - Selects clock input source (baud rate)
  - Sets dedicated baud rate generator
  - Selects clock speed (clock division value) when using dedicated baud rate generator
  - Enables or disables output of serial data and clock pins
  - Initialize UART

- Serial control register 1 (SCR1)

  This register:

  - Sets availability of parity
  - Selects type of parity
  - Sets stop bit length
  - Sets data length
  - Selects frame data format in operation mode 1 (asynchronous multiprocessor mode)
  - Clears error flag
  - Enables or disables transmitting
  - Enables or disables receiving

et4U.com DataShee
DataSheet4U.com

- Serial status register 1 (SSR1)

  The status register checks the transmission/reception state and error state and sets enabling/disabling of the transmit/receive interrupt request.

- Serial input data register 1 (SIDR1)

  The serial input data register retains the receive data. The serial input data is converted and then stored in this register.

- Serial output data register 1 (SODR1)

  The serial output data register sets the transmit data. Data written to this register is serial-converted and then output.

- Communication prescaler control register (CDCR)

  The control register sets the baud rate of the baud rate generator, which sets the start/stop of the communication prescaler and the division rate of machine clock.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.3 Configuration of UART1

The UART1 pins, interrupt factors, register list and details are shown.

■ UART1 Pin

The pins used in the UART1 serve as general-purpose I/O port.

Table 14.3-1 indicates the pin functions and the setting necessary for use in the UART1.

Table 14.3-1 UART1 Pin

Pin Name	Pin Function	Setting Necessary for Use in UART1
SOT1	General-purpose I/O port, serial data output	Set to output enable. (SMR1 register bit 0: SOE=1)
SCK1	General-purpose I/O port, serial clock input/output	In clock input, set pin as input port in port direction register (DDR).
		In clock output, set to output enable. (SMR1 register bit 1: SCKE=1)
SIN1	General-purpose I/O port, serial data input	Set pin as input port in DDR.

et4U.com DataShee

■ Block Diagram of Pins of UART1

DataSheet4U.com

See "CHAPTER 4 I/O PORT" for the block diagram of pins.

■ List of Registers in UART1

Figure 14.3-1 List of Registers and Reset Values in UART1

	bit	15	14	13	12	11	10	9	8
Serial control register 1 (SCR1)		0	0	0	0	0	1	0	0
	bit	7	6	5	4	3	2	1	0
Serial mode register 1 (SMR1)		0	0	0	0	0	0	0	0
	bit	15	14	13	12	11	10	9	8
Serial status register 1 (SSR1)		0	0	0	0	1	0	0	0
	bit	7	6	5	4	3	2	1	0
Serial input data register 1 (SIDR1)/ Serial output data register 1 (SODR1)		X	X	X	X	X	X	X	X
	bit	15	14	13	12	11	10	9	8
Communication prescaler control register 1 (CDCR1)		X	X	X	X	X	X	X	X

Note: Function as SIDR1 when reading, function as SODR1 when writing

X: Undefined

DataSheet4U.com www.DataSheet4U.com

DataSh4U.com

www.DataSheet4U.com

■ Interrupt Request Generation by UART1

● Receive interrupt

- When receive data is loaded to the serial input data register (SIDR1), the receive data load flag bit (bit 12: RDRF) in the serial status register (SSR1) is set to 1. When a receive interrupt is enabled (bit 9: RIE = 1), a receive interrupt request is generated to the interrupt controller.
- When either a framing error, overrun error, or parity error occurs, the framing error flag bit (bit 13: FRE), the overrun error flag bit (bit 14: ORE), or parity error flag bit (bit 15: PE) in the serial status register (SSR1) are set to 1 according to the error occurred. When a receive interrupt is enabled (bit 9: RIE = 1), a receive interrupt is requested to the interrupt controller.

● Transmit interrupt

When transmit data is transferred from the serial output data register (SODR1) to the transmit shift register, the transmit data empty flag bit (bit 11: TDRE) in the serial status register (SSR1) is set to 1. If a transmit interrupt is enabled (bit 8: TIE = 1), a transmit interrupt is requested to the interrupt controller.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

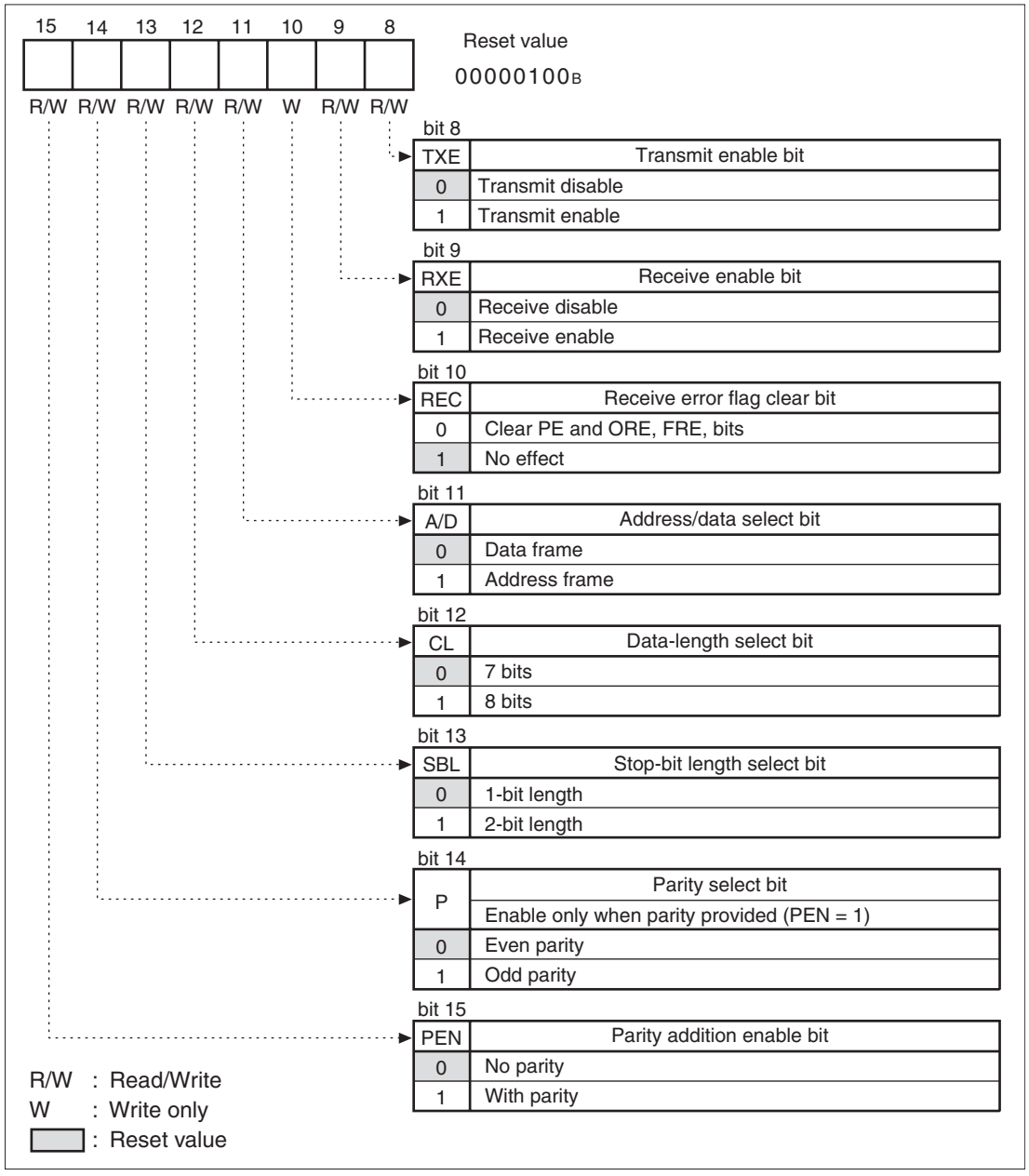
www.DataSheet4U.com

14.3.1 Serial Control Register 1 (SCR1)

The serial control register 1 (SCR1) performs the following: setting parity bit, selecting stop bit length and data length, selecting frame data format in operation mode 1 (asynchronous multiprocessor mode), clearing receive error flag, and enabling/ disabling of transmitting/receiving.

■ Serial Control Register 1 (SCR1)

Figure 14.3-2 Serial Control Register 1 (SCR1)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 14.3-2 Function of Serial Control Register 1 (SCR1)

Bit Name		Function
bit 8	TXE: Transmit enable bit	Enable or disable the UART1 for sending. **When set to 0:** Transmission disabled **When set to 1:** Transmission enabled **Note:** When transmitting is disabled during transmitting, transmitting stops after the data in the serial output data register (SODR1) being transmitted is completed. To set this bit to 0, after writing data to SODR1, wait for a time of 1/16th of the baud rate in the asynchronous mode and for a time equal to or more than the baud rate in the synchronous mode.
bit 9	RXE: Receive enable bit	Enable or disable the UART1 for receiving. **When set to 0:** Reception disabled **When set to 1:** Reception enabled **Note:** When receiving is disabled during receiving, receiving stops after the data being received is stored in the serial input data register.
bit 10	REC: Receive error flag clear bit	Clear the receive error flags (bit 15 to 13: PE, ORE and FRE) of the serial status register (SSR1) to 0. **When set to 0:** Clears PE, ORE and FRE bits **When set to 1:** No effect **When read:** 1 always read **Note:** When a receive interrupt is enabled (bit 9: RIE = 1), set the bit10: REC bit to 0 only when any one of the PE, ORE and FRE bits is set to 1.
bit 11	A/D: Address/data select bit	In operation mode 1 (asynchronous multiprocessor mode), set the data format of the frame to be transmitted/received. **When bit set to 0:** Data frame set **When bit set to 1:** Address frame set
bit 12	CL: Data-length select bit	Specify the length of send and receive data. **Note:** A data length of 7 bits can be selected only in operation mode 0 (asynchronous normal mode). In operation modes 1 and 2 (asynchronous multiprocessor mode, Clock synchronous mode), be sure to set a data length of 8 bits.
bit 13	SBL: Stop-bit length select bit	Set the length of the stop bit (frame end mark of send data) in operation modes 0 and 1 (multiprocessor mode, synchronous mode). **Note:** At receiving, only the first bit of the stop bit is always detected.
bit 14	P: Parity select bit	Select either odd or even parity when with parity (PEN = 1) is set.
bit 15	PEN: Parity addition enable bit	Specify whether to add (at sending) and detect (at receiving) a parity bit. **Note:** A parity bit is not added in operation modes 1 and 2 (Multiprocessor mode, Synchronous mode). Be sure to set this bit to 0.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

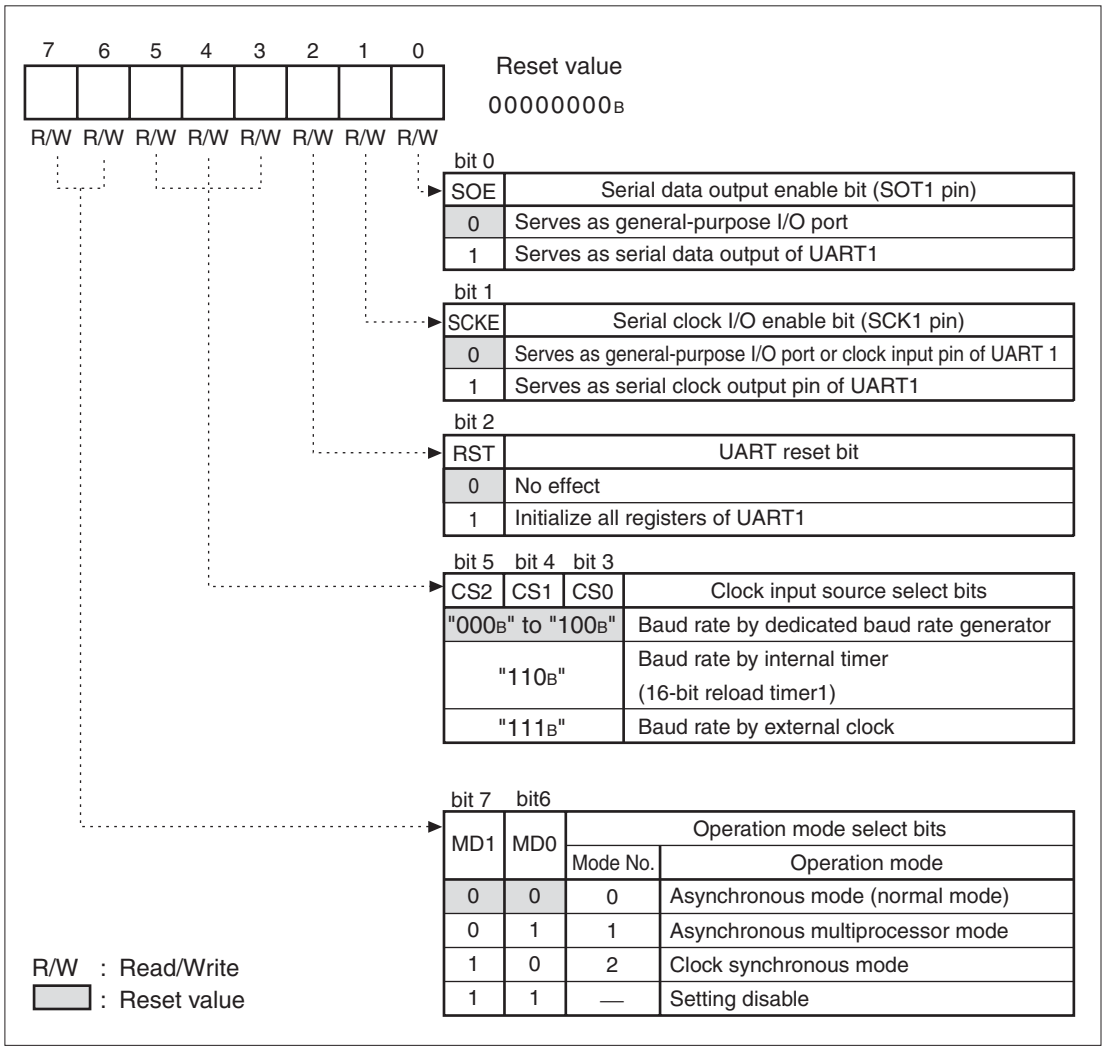
www.DataSheet4U.com

14.3.2 Serial Mode Register 1 (SMR1)

The serial mode register 1 (SMR1) performs selecting operation mode, selecting baud rate clock, and disabling/enabling of output of serial data and clock to pin.

■ Serial Mode Register 1 (SMR1)

Figure 14.3-3 Serial Mode Register 1 (SMR1)



bit 0

SOE	Serial data output enable bit (SOT1 pin)
0	Serves as general-purpose I/O port
1	Serves as serial data output of UART1

bit 1

SCKE	Serial clock I/O enable bit (SCK1 pin)
0	Serves as general-purpose I/O port or clock input pin of UART 1
1	Serves as serial clock output pin of UART1

bit 2

RST	UART reset bit
0	No effect
1	Initialize all registers of UART1

bit 5 bit 4 bit 3

CS2	CS1	CS0	Clock input source select bits
"000B" to "100B"			Baud rate by dedicated baud rate generator
"110B"			Baud rate by internal timer (16-bit reload timer1)
"111B"			Baud rate by external clock

bit 7 bit6

MD1	MD0	Operation mode select bits	
		Mode No.	Operation mode
0	0	0	Asynchronous mode (normal mode)
0	1	1	Asynchronous multiprocessor mode
1	0	2	Clock synchronous mode
1	1	—	Setting disable

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 14.3-3 Function of Serial Mode Register 1 (SMR1)

Bit Name		Function
bit 0	SOE: Serial-data output enable bit	Enable or disable output of serial data. **When set to 0:** General-purpose I/O port set **When set to 1:** Serial data output pin set
bit 1	SCKE: Serial clock I/O enable bit	Switch between input and output of the serial clock. **When set to 0:** General-purpose I/O port or serial clock input pin set **When set to 1:** Serial clock output pin set **Notes:** 1. When using the SCK1 pin as the serial clock input, set the pin to the input port using the port direction register (DDR). Also select the external clock (bit 5 to 3: CS2 to CS0 = "111_B") using the clock input source select bit. 2. When using the SCK pin as the serial clock output, set the clock input source select bit to anything other than the external clock (bit 5 to 3: CS2 to CS0 = anything other than "111_B").
bit 2	RST: UART Reset bit	This bit resets all registers in the UART1. **When set to 0:** No effect on operation **When set to 1:** Resets all registers in UART1
bit 3 to bit 5	CS0 to CS2: Clock input source select bits	Set the clock input source for the baud rate. • Select the external clock (SCK1 pin), internal timer (16-bit reload timer), or dedicated baud rate generator as the clock input source. • Set the baud rate when selecting the dedicated baud rate generator.
bit 6 and bit 7	MD0, MD1: Operation mode select bits	Select the UART1 operation mode. **Notes:** 1. In operation mode 1 (asynchronous multiprocessor mode), only the master can be used for master/slave communication. In operation mode 1, the address/data bit on bit 9 cannot be received, so the slave cannot be used. 2. In operation mode 1 (asynchronous multiprocessor mode), the parity check function cannot be used, set the parity addition enable bit to no parity (SCR1 register bit 15: PEN = 0).

Note: When 0 is written to the RST bit of Serial Mode Register, the UART interruption should be prohibited. To prohibit the interruption, take one of the following procedures:

1. Before writing 0 to the RST bit, clear I flag to prohibit all interrupt factors.
2. Before writing 0 to the RST bit, prohibit the UART interruption with the ILM register.
3. When 0 is written to the RST bit, writing should be performed at the UART interruption level or the level with higher priority than the UART interruption.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

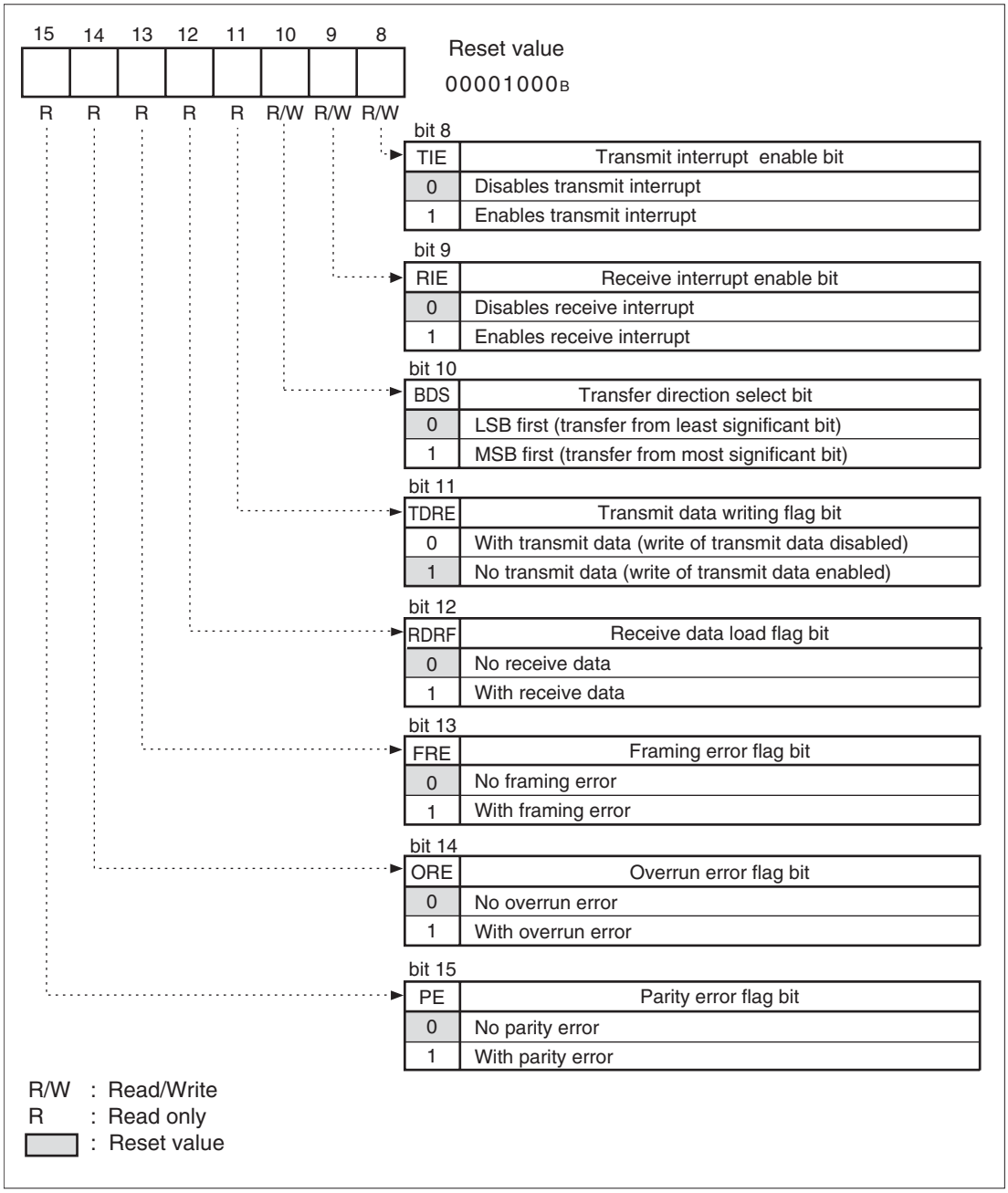
www.DataSheet4U.com

14.3.3 Serial Status Register 1 (SSR1)

The serial status register 1 (SSR1) checks the transmission/reception status and error status and enables/disables interrupts.

■ Serial Status Register 1 (SSR1)

Figure 14.3-4 Serial Status Register 1 (SSR1)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 14.3-4 Function of Serial Status Register 1 (SSR1)

Bit Name		Function
bit 8	TIE: Transmit interrupt enable bit	Enable or disable send interrupt. **When set to 1:** A receive interrupt request is issued when data written to the serial output data register 1 (SODR1) is sent to the transmit shift register (bit 11: TDRE = 1).
bit 9	RIE: Receive interrupt enable bit	Enable or disable receive data. **When set to 1:** A receive interrupt request is issued when receive data is loaded to the serial input data register 1 (SIDR1) (bit 12: RDRF = 1) or when a receive error occurs (bit 15: PE = 1, bit 14: ORE = 1, or bit 13: FRE = 1).
bit 10	BDS: Transfer direction select bit	This bit sets the direction of serial data transfer. **When set to 0:** Transfers data from least significant bit (LSB first) **When set to 1:** Transfers data from most significant bit (MSB first) **Note:** At reading and writing data from and to the serial data register, data is written to the serial output data register (SODR1) and then the transfer direction select bit (BDS) is rewritten to switch between the upper bits and the lower bits of data. In this case, the written data becomes invalid.
bit 11	TDRE: Transmit data write flag bit	Show the status of the serial output data register 1. • This bit is cleared to 0 when send data is written to the serial output register 1(SODR1). • This bit is set to 1 when data is loaded to the send shift register and transmission starts. • When a transmission interrupt is enabled (bit 8: TIE = 1), a transmit interrupt request is issued when data written to the serial output data register 1(SODR1) is transmitted to the transmit shift register (bit 11: TDRE=1).
bit 12	RDRF: Receive data load flag bit	Show the status of the serial input data register 1 (SIDR1). • This bit is set to 1 when receive data is loaded to the serial input register 1 (SIDR1). • This bit is cleared to 0 when data is read from the SIDR1. • When a receive interrupt is enabled (bit 9: RIE = 1), a receive interrupt request is issued when receive data is loaded to the serial input data register 1 (SIDR1).
bit 13	FRE: Framing error flag bit	Detect a framing error in receive data. • This bit is set to 1 when a framing error occurs. • This bit is cleared when 0 is written to the receive error flag clear bit (SCR1 register bit 10: REC). • When a receive interrupt is enabled (bit 9: RIE = 1), a receive interrupt request is issued when a framing error occurs. • When the framing error flag bit is set (bit 13: FRE = 1), data in the serial input data register 1 (SIDR1) is invalid.
bit 14	ORE: Overrun error flag bit	Detect an overrun error in receiving. • This bit is set to 1 when an overrun error occurs. • This bit is cleared when 0 is written to the receive error flag clear bit (SCR1 register bit 10: REC). • When a receive interrupt is enabled (bit 9: RIE = 1), a receive interrupt request is issued when an overrun error occurs. • When the overrun error flag bit is set (bit 14: ORE = 1), data in the serial input data register 1 (SIDR1) is invalid.
bit 15	PE: Parity error flag bit	Detect a parity error in receiving. • This bit is set to 1 when a parity error occurs. • This bit is cleared when 0 is written to the receive error flag clear bit (SCR1 register bit 10: REC). • When a receive interrupt is enabled (bit 9: RIE = 1), a receive interrupt request is issued when a parity error occurs. • When the parity error flag bit is set (bit 15: PE = 1), data in the serial input data register 1 (SIDR1) is invalid.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.3.4 Serial Input Data Register 1 (SIDR1) and Serial Output Data Register 1 (SODR1)

The serial input data register (SIDR1) and serial output data register (SODR1) are allocated to the same address. At read, the register functions as SIDR1. At write, the register functions as SODR1.

■ Serial Input Data Register 1 (SIDR1)

Figure 14.3-5 Serial Input Data Register 1 (SIDR1)

	bit 7	6	5	4	3	2	1	bit 0	Reset value
	D7	D6	D5	D4	D3	D2	D1	D0	$XXXXXXXX_B$
	R	R	R	R	R	R	R	R	

R: Read only
X: Undefined

SIDR1 is a data buffer register for receiving serial data.

- The serial data signal transmitted to the serial data input pin (SIN1) is converted by the shift register and stored in SIDR1.
- When the data length is 7 bits, the upper one bit (SIDR1: D7) becomes invalid.
- When receive data is stored in the serial input data register 1 (SIDR1), the receive data load flag bit (SSR1 register bit 12: RDRF) is set to 1. When a receive interrupt is enabled (SSR1 register bit 9: RIE = 1), a receive interrupt request is issued to the interrupt controller.
- Read SIDR1 when the receive data load flag bit (SSR1 register bit 12: RDRF) is set to 1. The receive data load flag bit (SSR1 register bit 12: RDRF) is cleared to 0 automatically when SIDR1 is read.
- When a receive error occurs (any one of SSR1 register bit 15, 14, 13: PE, ORE and FRE is 1), the receive data in SIDR1 becomes invalid.

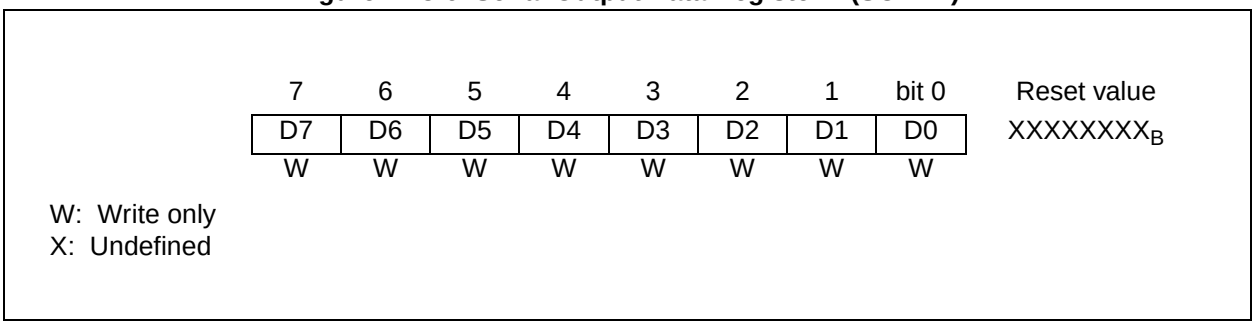
DataSheet4U.com
www.DataSheet4U.com

DataSh4U.com

www.DataSheet4U.com

■ Serial Output Data Register 1 (SODR1)

Figure 14.3-6 Serial Output Data Register 1 (SODR1)



The serial output data register 1 (SODR1) is a data buffer register for transmitting serial data.

- When data to be transmitted is written to SODR1 when transmission is enabled (SCR1 register bit 8: TXE = 1), it is transferred to the transmit shift register, converted to serial data, and transmit from the serial data output pin (SOT1).
- When the data length is 7 bits, the upper one bit (SODR1 register bit 7: D7) becomes invalid.
- The transmit data write flag (SSR1 register bit 11: TDRE) is cleared to 0 when send data is written to SODR1.
- The transmit data write flag is set to 1 at completion of data transfer to the transmit shift register.
- When the transmit data write flag (SSR1 register bit 11: TDRE) is 1, the next transmit data can be written. When a transmit interrupt is enabled (SSR1 register bit 8: TIE=1), a transmit interrupt occurs to the interrupt controller. The next transmit bit data should be written with the transmit data write flag (SCR1 register bit 11: TDRE) at 1.

et4U.com DataShee DataSheet4U.com

Note:	Serial output data register is a write-only register and serial input data register is a read-only register. However, since they are allocated to the same address, the write and read values are different. Therefore, do not use instructions that perform read-modify-write (RMW) operation such as INC and DEC instructions.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

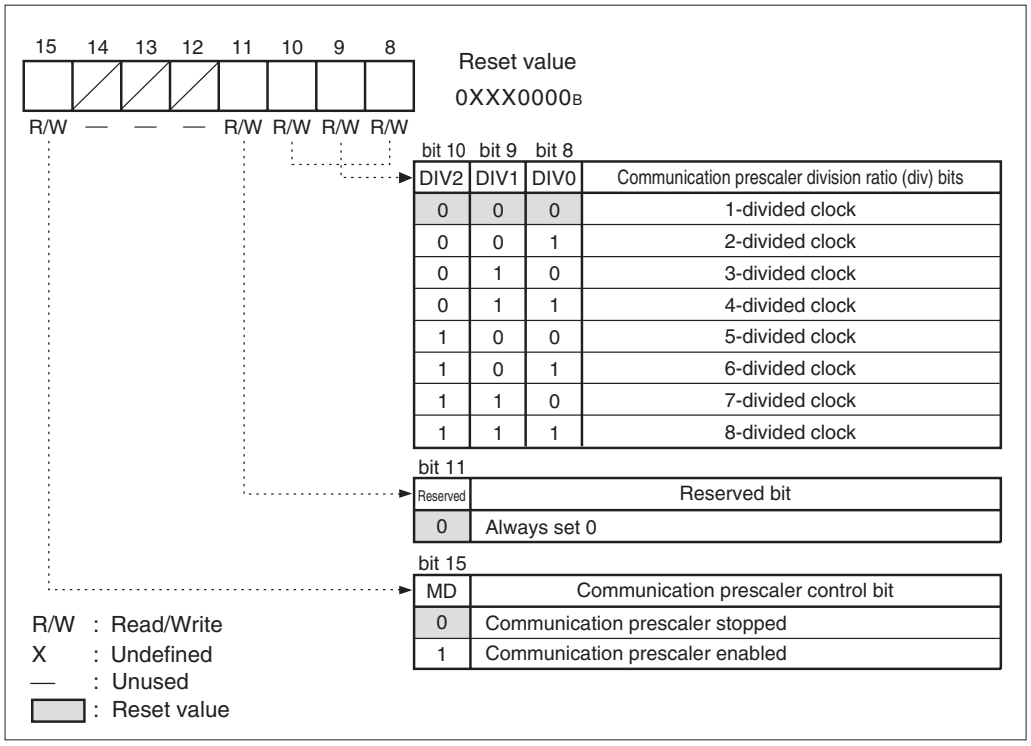
14.3.5 Communication Prescaler Control Register 1 (CDCR1)

The communication prescaler control register 1 (CDCR1) is used to set the baud rate of the dedicated baud rate generator for the UART1.

- **Starts/stop the communication prescaler**
- **Sets the division ratio for machine clock**

■ Communication Prescaler Control Register 1 (CDCR1)

Figure 14.3-7 Communication Prescaler Control Register 1 (CDCR1)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 14.3-5 Functions of Communication Prescaler Control Register 1 (CDCR1)

Bit Name		Function
bit 8 to bit 10	DIV0 to DIV2: Communication prescaler division ratio bits	• These bits set the machine clock division ratio. **Note:** When changing the division ratio, the time of at least 2 cycles division of the division clock should be allowed before the next communication is started in order to stabilize the clock frequency.
bit 11	Reserved: Reserved bit	Be sure to set this bit to 0.
bit 12 to bit 14	Unused bits	**Read:** The value is not fixed. **Write:** No effect
bit 15	MD: Communication prescaler control bit	This bit enables or disables the communication prescaler. **When set to 0:** Stops communication prescaler **When set to 1:** Operates communication prescaler

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.4 Interrupt of UART1

The UART1 has a receive and a transmit interrupts, and the following factors can issue interrupt requests.

- **Receive data is loaded to the serial input data register 1 (SIDR1).**
- **A receive error (parity error, overrun error, framing error) occurs.**
- **When send data transferred from the serial output data register 1 (SODR1) to transmit shift register**

Also, each of these interrupt factors supports the expansion intelligent I/O service (EI^{2}OS).

■ Interrupt of UART1

The UART1 interrupt control bits and interrupt factors are shown in Table 14.4-1.

Table 14.4-1 UART1 Interrupt Control Bit and Interrupt Factor

Trans-mission/ Reception	Interrupt-request Flag Bit	Operation Mode			Interrupt Factor	Interrupt Factor Enable Bit	Clear of the Interrupt-request Flag
		0	1	2			
Reception	SSR1: RDRF	O	O	O	Receive data loaded into serial input data register 1 (SIDR1)	SSR1: RIE	Reading receive data
	SSR1:ORE	O	O	O	Overrun error		Writing 0 to receive error flag clear bit (SCR1 register bit 10: REC)
	SSR1:FRE	O	O	X	Framing error		
	SSR1:PE	O	X	X	Parity error		
Transmission	SSR1: TDRE	O	O	O	Transfer of transmit data completed from serial output data register 1 (SODR1)	SSR1: TIE	Writing transmit data

O: Available
X: Not available

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Receive interrupt

When a receive interrupt is enabled (SSR1 register bit 9: RIE = 1), a receive interrupt request is issued to the interrupt controller at completion of data receiving (SSR1 register bit 12: RDRF = 1) or when any one of the overrun error (SSR1 register bit 14: ORE = 1), framing error (SSR 1 register bit 13: FRE = 1), and parity error (SSR 1 register bit 15: PE = 1) occurs.

The receive data load flag (SSR1 register bit 12: RDRF) is cleared to 0 automatically when the serial input data register 1 (SIDR1) is read. Each receive error flag (SSR1 register bit 15, 14, 13: PE, ORE, FRE) is cleared to 0 when 0 is written to the receive error flag clear bit (SCR1 register bit 10: REC).

Note:	If a receive error (parity error, overrun error, framing error) occurs, correct the error as necessary, and then write 0 to the receive error flag clear bit (SCR1 register bit 10: REC) to clear each receive error flag.

- Transmit interrupt

When send data is transmitted from the serial output data register 1 (SODR1) to the transmit shift register, the transmit data write flag bit (SSR1 register bit 11: TDRE) is set to 1.

When a transmit interrupt is enabled (SSR1 register bit 8: TIE = 1), a send interrupt request is issued to the interrupt controller.

■ Interrupt Related to UART1 and EI^{2}OS

See Section "3.5 Interrupt" for the interrupt number, interrupt control register, and interrupt vector addresses.

et4U.com DataShee

■ EI^{2}OS Function of UART1

DataSheet4U.com

The UART1 supports EI^{2}OS. Consequently, EI^{2}OS can be started separately for receive interrupts and transmit interrupts.

- At reception:

The EI^{2}OS can be used regardless of the state of other resources.

- At transmission:

Since the interrupt control registers (ICR13, 14) are shared with transmit interrupts of UART1, EI^{2}OS can be started only when UART1 receive interrupts are not used.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.4.1 Generation of Receive Interrupt and Timing of Flag Set

Interrupts at receiving include the receive completion (SSR1 register bit 12: RDRF), and the receive error (SSR1 register bit 15, 14, 13: PE, ORE, FRE).

■ Generation of Receive Interrupt and Timing of Flag Set

● Receive data load flag and each receive error flag sets

When data is received, it is stored in the serial input data register 1 (SIDR1) when the stop bit is detected (in operation modes 0 and 1: Asynchronous normal mode, Asynchronous multiprocessor mode) or when the last bit of receive data (SIDR1 register bit 7: D7) is detected (in operation mode 2: Clock synchronous mode). When a receive error occurs, the error flags (SSR1 register bit 15, 14, 13: PE, ORE, FRE) and receive data load flag (SSR1 register bit 12: RDRF) are set. In each operation mode, the received data in the serial input data register 1 (SIDR1) is invalid if either error flag is set.

Operation mode 0 (Asynchronous normal mode)

The receive data load flag bit (SSR1 register bit 12: RDRF) is set when the stop bit is detected. The error flags (SSR1 register bit 15, 14, 13: PE, ORE, FRE) are set when a receive error occurs.

Operation mode 1 (Asynchronous multiprocessor mode)

The receive data load flag bit (SSR1 register bit 12: RDRF) is set when the stop bit is detected. The error flags (SSR1 register bit 14, 13: ORE, FRE) are set when a receive error occurs. A parity error (SSR1 register bit 15: PE) cannot be detected.

Operation mode 2 (Clock synchronous mode)

The receive data load flag bit (SSR1 register bit 12: RDRF) is set to 1 when the last bit of receive data (SIDR1 register bit 7: D7) is detected. The error flags (SSR1 register bit 14: ORE) are set when a receive error occurs. A parity error (SSR1 register bit 15: PE) and framing error (SSR1 register bit 13: FRE) cannot be detected.

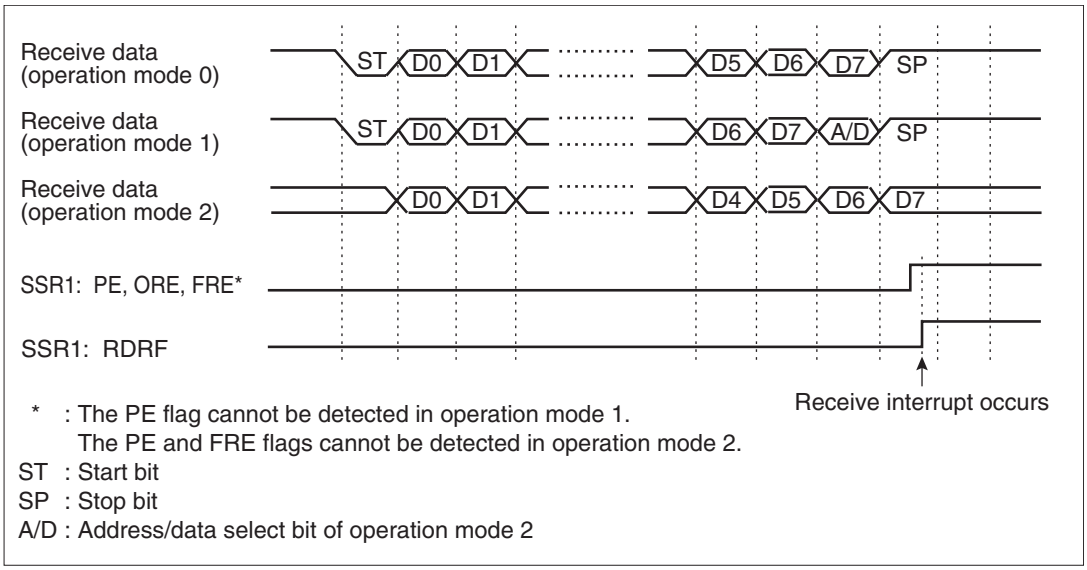
Reception and timing of flag set are shown in Figure 14.4-1.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 14.4-1 Reception and Timing of Flag Set



* : The PE flag cannot be detected in operation mode 1.
The PE and FRE flags cannot be detected in operation mode 2.
ST : Start bit
SP : Stop bit
A/D : Address/data select bit of operation mode 2

● Timing of receive interrupt request generation

With a receive interrupt enabled (SSR1 register bit 9: RIE = 1), when any one of the receive data load flag (SSR1 register bit 12: RDRF), parity error flag (SSR1 register bit 15: PE), overrun error flag (SSR1 register bit 14: ORE) and framing error flag (SSR1 register bit 13: FRE) is set, reception interrupt is requested to interrupt controller.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.4.2 Generation of Transmit Interrupt and Timing of Flag Set

At transmission, the interrupt is generated in the state which the succeeding data can be written to the serial output data register 1 (SODR1).

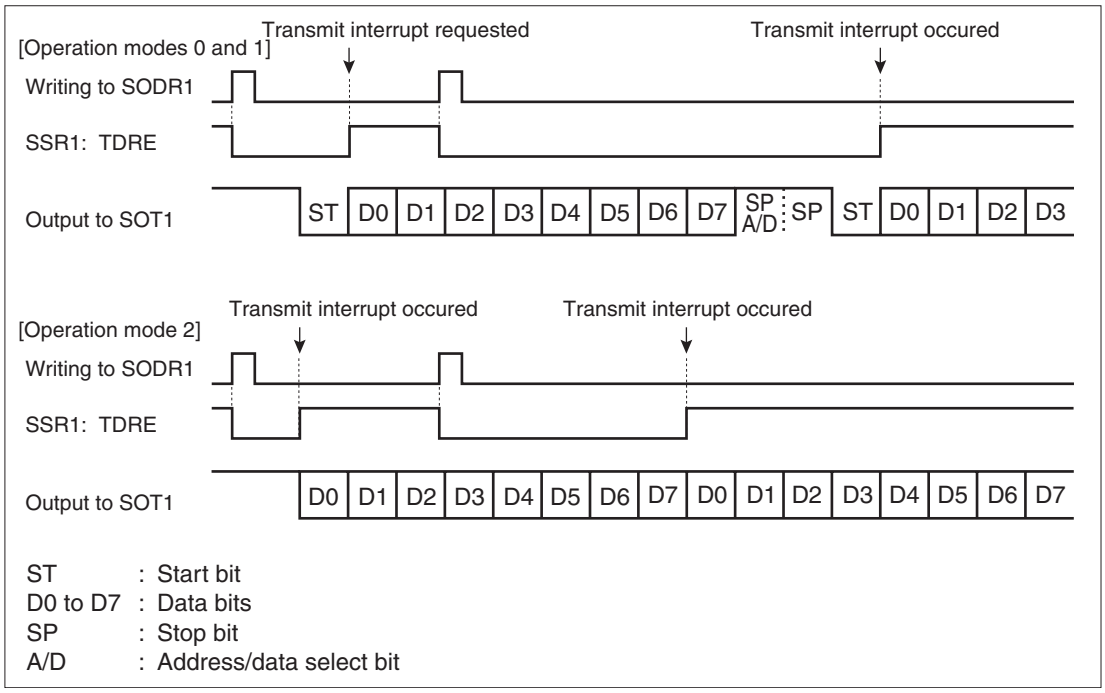
■ Generation of Transmit Interrupt and Timing of Flag Set

● Set and clear of transmit data empty flag bit

The send data write flag bit (SSR1 register bit 11: TDRE) is set when the send data written to the serial output data register 1 (SODR1) is loaded to the send shift register and the next data is ready for writing. The send data write flag bit (SSR1 register bit 11: TDRE) is cleared to 0 when the next send data is written to the serial output data register 1 (SODR1).

Transmission and timing of flag set are shown in Figure 14.4-2.

Figure 14.4-2 Transmission and Timing of Flag Set



● Timing of transmit interrupt request

When a transmit interrupt is enabled (SSR1 register bit 8: TIE = 1), a send interrupt request is issued to interrupt controller when the transmit data write flag bit (SSR1 register bit 11: TDRE) is set.

Note: When sending is disabled during sending (SCR1 register bit 8: TXE=0: and also in operation mode 1 (asynchronous multiprocessor mode), receiving disabled (also including bit 9: RXE), the send data write flag bit is set (SSR1 register bit 11: TDRF=1) and UART 1 communications are disabled after the shift operation of the send shift register stops. The send data written to the serial output data register 1 before the transmission stops (SODR1) is sent.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.5 Baud Rate of UART1

One of the following can be selected as the UART1 transmit/receive clock.

- **Dedicated baud rate generator**
- **Internal clock (16-bit reload timer output)**
- **External clock (clock input to SCK1 pin)**

■ Select of UART1 Baud Rate

The UART1 baud rate select circuit comprises as shown in Figure 14.5-1. The clock input source can be selected from among the following three types:

● Baud rate by dedicated baud rate generator

- When using the dedicated baud rate generator incorporated into UART1 as a clock input source, set the CS2 to CS0 bits in the serial mode register (SMR1) bit 5 to 3 to "000_B" to "101_B" according to the baud rate. The baud rate can be selected from six types.

● Baud rate by internal timer

- When using the internal clock supplied from the 16-bit reload timer as a clock input source, set the CS2 to CS0 bits in SMR1 bit 5 to 3 to "110_B".
- The baud rate is the value at which the clock supplied from the 16-bit reload timer as it is in the clock synchronous mode, and the value at which the frequency of the supplied clock is divided by 16 in the clock asynchronous mode.
- Any baud rate can be selected according to the setting values of the 16-bit reload timer.

● Baud rate by external clock

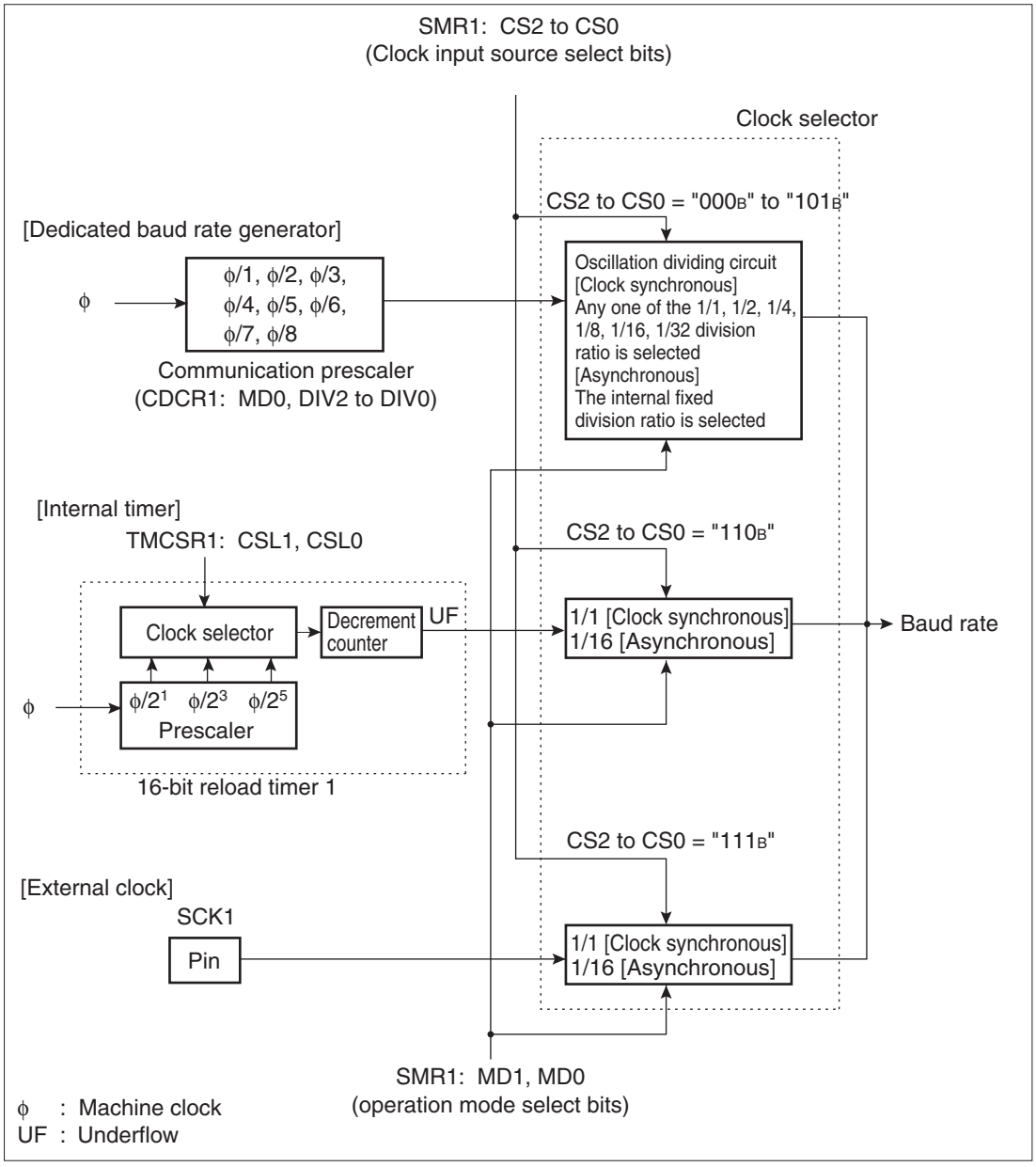
- When using the external clock supplied from the clock input pin (SCK1) in the UART1 as the clock input source, set the CS2 to CS0 bits in SMR1 bit 5 to 3 to "111_B".
- The baud rate is the value at which the external clock is supplied in the clock synchronous mode and the value at which the frequency of the input clock is divided by 16 in the clock asynchronous mode.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 14.5-1 UART1 Baud Rate Selector



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.5.1 Baud Rate by Dedicated Baud Rate Generator

The baud rate that can be set when the output clock of the dedicated baud rate generator is selected as the transmit/receive clock of the UART1 is shown.

■ Baud Rate by Dedicated Baud Rate Generator

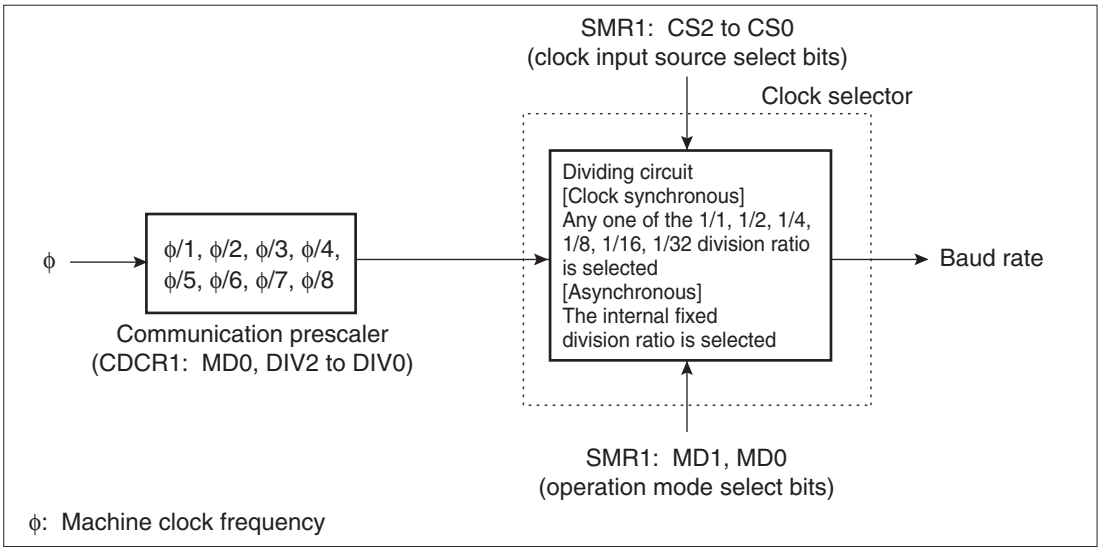
The baud rate based on the dedicated baud rate generator is set by setting the clock input source select bits in the serial mode register (SMR1 register bit 5 to 3: CS2 to CS0) to "000_B" to "101_B".

When generating a transmit/receive clock using the dedicated baud rate generator, the division ratio for the clock input source selected by the clock selector is selected to determine the baud rate after the machine clock frequency is divided by the communication prescaler.

The division ratio at which the machine clock frequency is divided by the communication prescaler is the same for the clock synchronous and asynchronous modes. The division ratio at which the baud rate is determined is different for the clock synchronous and asynchronous modes.

Figure 14.5-2 shows the baud rate selector based on the dedicated baud rate generator.

Figure 14.5-2 Baud Rate Selector Based on Dedicated Baud Rate Generator



● Calculation expression for baud rate

Baud rate in asynchronous mode = ϕ x div x (division ratio of transfer clock in asynchronous mode)

Baud rate in clock synchronous mode = ϕ x div x (division ratio of transfer clock in clock synchronous mode)

ϕ: Machine clock frequency

div: Division ratio based on communication prescaler

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Division ratio based on communication prescaler (common between asynchronous and clock synchronous modes)

The division ratio of the machine clock is set by the division ratio select bits in the communication prescaler control register (CDCR1 register bit 10 to 8: DIV2 to DIV0).

Table 14.5-1 Division Ratio Based on Communication Prescaler

MD	DIV2	DIV1	DIV0	div
0	–	–	–	Stop
1	0	0	0	1
1	0	0	1	2
1	0	1	0	3
1	0	1	1	4
1	1	0	0	5
1	1	0	1	6
1	1	1	0	7
1	1	1	1	8

div: Division ratio based on communication prescaler

et4U.com DataShee

DataSheet4U.com

- Baud rate (asynchronous mode)

The baud rate in the asynchronous mode is generated using output clock of the communication prescaler. The division ratio is set by the clock input source select bits (SMR1 register bit 5 to 3: CS2 to CS0).

Table 14.5-2 Baud Rate (Asynchronous Mode)

CS2	CS1	CS0	Asynchronous Mode (Start/Stop Synchronous)	Calculation
0	0	0	76,923 bps	(ϕ/div) / (8 x 13 x 2)
0	0	1	38,461 bps	(ϕ/div) / (8 x 13 x 4)
0	1	0	19,230 bps	(ϕ/div) / (8 x 13 x 8)
0	1	1	9,615 bps	(ϕ/div) / (8 x 13 x 16)
1	0	0	500 kbps	(ϕ/div) / (8 x 2 x 2)
1	0	1	250 kbps	(ϕ/div) / (8 x 2 x 4)

ϕ: Machine clock frequency
div: Division ratio based on communication prescaler

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Baud rate (synchronous mode)

The baud rate in the synchronous mode is generated by dividing the output clock of the communication prescaler by 1, 2, 4, 8, 16 and 32. Set the division ratio using the clock input source select bits (bits 5 to 3 in SMR1 register: CS2 to CS0).

Table 14.5-3 Baud Rate (Clock Synchronous)

CS2	CS1	CS0	CLK Synchronous	Calculation
0	0	0	2 Mbps	(ϕ/div) / 1
0	0	1	1 Mbps	(ϕ/div) / 2
0	1	0	500 kbps	(ϕ/div) / 4
0	1	1	250 kbps	(ϕ/div) / 8
1	0	0	125 kbps	(ϕ/div) / 16
1	0	1	62.5 kbps	(ϕ/div) / 32

ϕ: Machine clock frequency
div: Division ratio based on communication prescaler

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.5.2 Baud Rate by Internal Timer (16-bit Reload Timer)

The setting when selecting the internal clock supplied from the 16-bit reload timer 1 as the clock input source of the UART1 and the baud rate calculation are shown below.

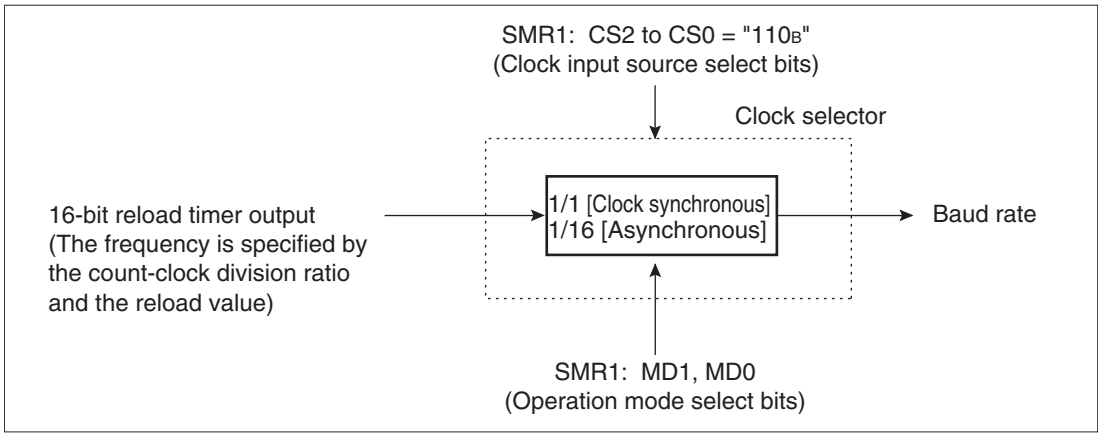
■ Baud Rate by Internal Timer (16-bit Reload Timer Output)

The baud rate based on the internal timer (16-bit reload timer output) is set by setting the clock input source select bits (SMR1 register bit 5 to 3: CS2 to CS0) to "110_B". Any baud rate can be set by selecting the division ratio of the count clock and the reload value of the 16-bit reload timer freely.

Figure 14.5-3 shows the baud rate selector based on the internal timer.

- If the internal timer (16-bit reload timer) is selected as a clock input source (SMR1 register bit 5 to 3: CS2 to CS0), the 16-bit reload timer output pin (TOT) is connected internally and does not need to be connected externally to the external clock input pin (SCK).
- The 16-bit reload timer output pin (TOT) can be used as a general-purpose I/O port when it is not being used in other way.

Figure 14.5-3 Baud Rate Selector by Internal Timer (16-bit Reload Timer Output)



● Calculation expression for baud rate

$$\text{Asynchronous baud rate} = \frac{\phi / N}{16 \times 2 \times (n+1)} \text{ bps}$$

$$\text{Clock synchronous} = \frac{\phi / N}{2 \times (n+1)} \text{ bps}$$

ϕ: machine clock frequency

N: division ratio based on communication prescaler for 16-bit reload timer (2^1, 2^3, 2^5)

n: reload value for 16-bit reload timer (0 to 65,535)

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Example of setting baud rates and reload register setting values (machine clock frequency: 7.3728 MHz)

Table 14.5-4 Baud Rate and Reload Value

Baud Rate (bps)	Reload Value			
	Clock Asynchronous (start-stop synchronization)		Clock Synchronous	
	N = 2^1(machine clock 2-divided)	N = 2^3(machine clock 8-divided)	N = 2^1(machine clock 2-divided)	N = 2^3(machine clock 8-divided)
38,400	2	–	47	11
19,200	5	–	95	23
9,600	11	2	191	47
4,800	23	5	383	95
2,400	47	11	767	191
1,200	95	23	1,535	383
600	191	47	3,071	767
300	383	95	6,143	1,535

N: Division ratio based on communication prescaler for 16-bit reload timer
–: Setting disabled

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.5.3 Baud Rate by External Clock

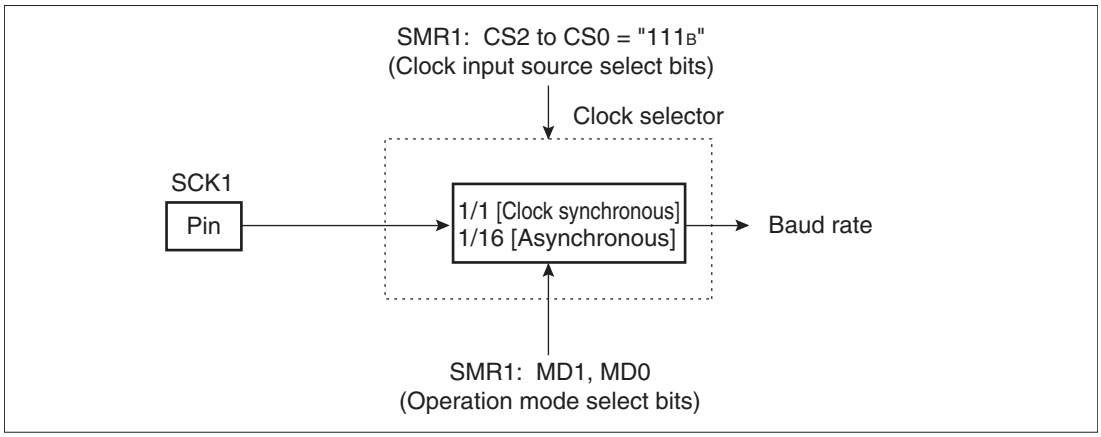
This section explains the setting when selecting the external clock as the transmit/ receive clock of the UART1.

■ Baud Rate by External Clock

To select a baud rate by the external clock input, the following settings are essential:

- Set the clock input source select bits in the serial mode register (SMR1 register bit 5 to 3: CS2 to CS0) to "111_B".
- Set the SCK1 pin as the input port in the port direction register (DDR).
- Set the serial clock I/O enable bit (SMR1 register bit 1: SCKE) to 0.
- Set the baud rate on the basis of the external clock input from the SCK1 pin. Since the internal division ratio is fixed, the external input clock must be changed in changing the baud rate.

Figure 14.5-4 Baud Rate Selector by External Clock



● Expressions to obtain baud rate

Asynchronous baud rate = f/16 bps

Clock synchronous baud rate = f bps

f: External clock frequency (2 MHz max.)

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.6 Explanation of Operation of UART1

The UART1 has master/slave type connection communication function (operation mode 1: asynchronous multiprocessor mode) in addition to bidirectional serial communication function (operation modes 0 and 2: asynchronous normal mode and clock synchronous mode).

■ Operation of UART1

● Operation mode

The UART1 has three types of operation modes, they can set the inter-CPU connection mode or data communication mode.

Table 14.6-1 shows operation mode of UART1.

Table 14.6-1 Operation Mode of UART 1

Operation Mode		Data Length		Synchronous/ Asynchronous	Length of Stop Bit
		No Parity	With Parity		
0	Normal mode	7 bits or 8 bits		Asynchronous	1 bit or 2 bits *2
1	Multiprocessor mode	8 + 1 *1	–	Asynchronous	
2	Clock synchronous mode	8	–	Synchronous	None

–: Setting disabled
*1: +1 is the address/data select bit (SCR1 register bit 11: A/D) used for controlling communications.
*2: During reception, only one bit can be detected as the stop bit.

Note: The UART1 operation mode 1 (asynchronous multiprocessor mode) is only used as the master in the master/slave type connection.

● Inter-CPU connection mode

Either 1-to-1 connection or master/slave type connection can be selected for the inter-CPU connection mode. In both cases, the data length, parity, synchronous or asynchronous mode, etc., must be the same for all CPUs. The operation modes are selected as follows.

- For the 1-to-1 connection, the same operation mode (either operation mode 0 or 2: normal mode, clock synchronous mode) must be adopted for the two CPUs. For the asynchronous mode, select operation mode 1: asynchronous multiprocessor mode (SMR1 register bit 7, 6: MD1, MD0 = "00_B"): for the synchronous mode select operation mode 2: clock synchronous mode (SMR1 register bit 7, 6: MD1, MD0 = "10_B").
- For the master/slave type connection, operation mode 1: asynchronous multiprocessor mode (SMR1 register bit 7, 6: MD1, MD0 = "01_B" is set; select operation mode 1 (asynchronous multiprocessor mode) and use it as the master. For this connection, select no parity and 8-bit data length.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Synchronous/asynchronous

For the operation modes, either the asynchronous mode (start-stop synchronization) or the clock-synchronous mode can be selected.

● Signal mode

The UART1 can only handle the NRZ (Non Return to Zero) data format.

● Start of transmission/reception

- Transmission starts when the transmission enable bit of the serial control register (SCR1 register bit 8: TXE) is set to 1.
- Reception starts when the reception enable bit of the serial control register (SCR1 register bit 9: RXE) is set to 1.

● Stop of transmission/reception

- Transmission stops when the transmission enable bit of the serial control register (SCR1 register bit 8: TXE) is set to 0.
- Reception stops when the reception enable bit of the serial control register (SCR1 register bit 9: RXE) is set to 0.

et4U.com

● Stop during transmission/reception

DataShee

DataSheet4U.com

- When reception is disabled during receiving (during data input to reception shift register) (SCR1 register bit 9: RXE = 0), it stops after reception of the frame being received is completed and the receive data is stored to the serial input data register 1 (SIDR1).
- When transmission is disabled during transmission (during data output from the transmission shift register) (SCR1 register bit 8: TXE = 0), it stops after transmission of one frame to the transmission shift register from the serial output data register 1 (SODR1) is completed.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.6.1 Operation in Asynchronous Mode (Operation Mode 0 or 1)

When the UART 1 is used in operation mode 0 (asynchronous normal mode) or operation mode 1 (asynchronous multiprocessor mode), the asynchronous transfer mode is selected.

■ Operation in Asynchronous Mode

● Format of transmit/receive data

Transmission and reception always start with the start bit (Low level); transmission and reception are performed at the specified data bit length on LSB first basis and end with the stop bit (High level).

- In operation mode 0 (Asynchronous normal mode), the data length can be set to 7 or 8 bits. Use of the parity bit can be specified.
- In operation mode 1 (Asynchronous multiprocessor mode), the data length is fixed to 8 bits. There is no parity bit. The address/data bit (SCR1 register bit 11: A/D) is added to bit 9.

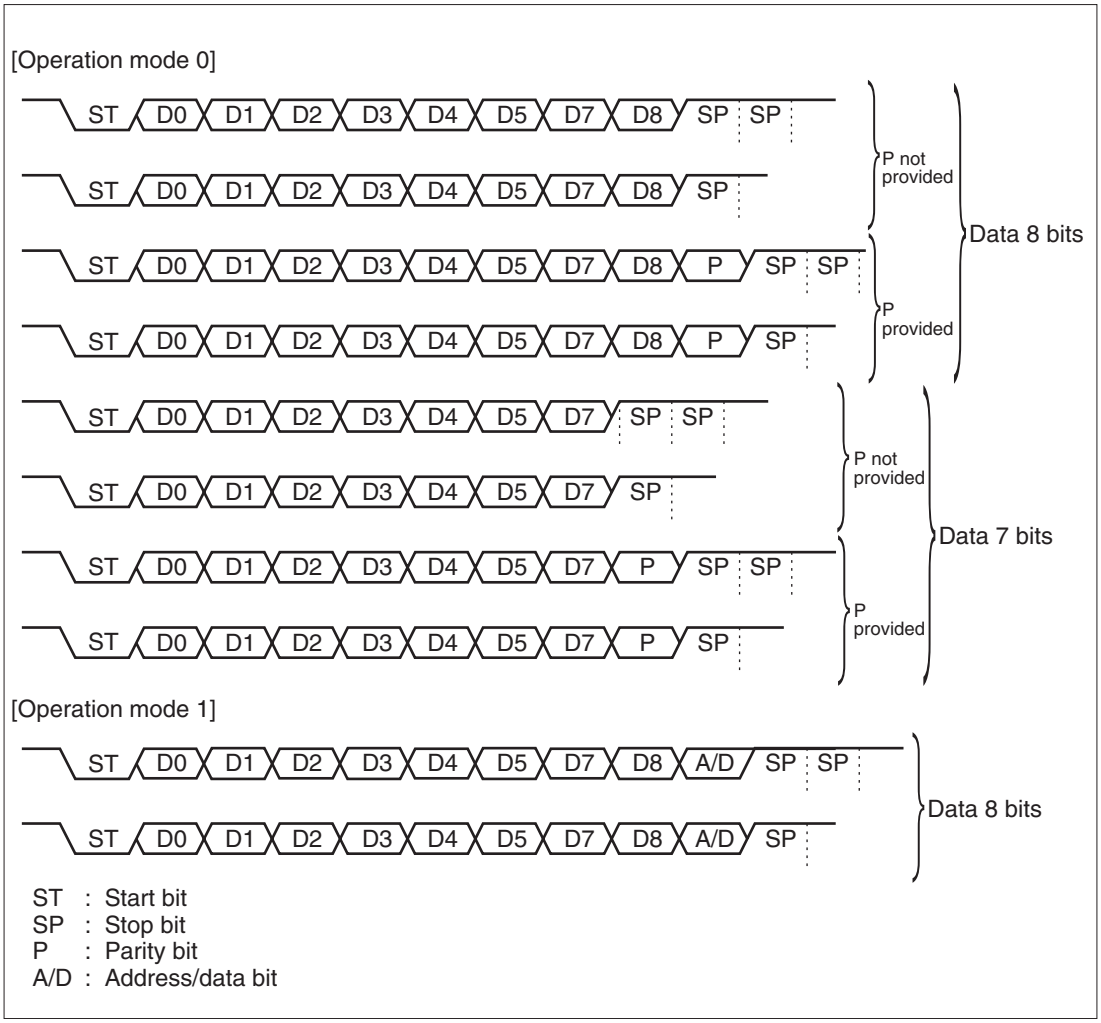
Figure 14.6-1 shows the transmit/receive data format in the asynchronous mode.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 14.6-1 Format of Transmit/Receive Data (Operation Mode 0 or 1)



● Transmission

- Transmit data is written to the serial output data register 1 (SODR1) with the transmit data write flag bit (SSR1 register bit 11: TDRE) set to 1.
- Transmission starts when transmit data is written and the transmit enable bit of the serial control register (SCR1 register bit 8: TXE) is set to 1.
- The transmit data write flag bit (SSR1 register bit 11: TDRE) is cleared to 0 temporarily when transmit data is written to SODR1.
- The transmit data write flag bit (SSR1 register bit 11: TDRE) is set to 1 again once the transmit data is written to the send shift register from the serial output data register 1 (SODR1).
- When the transmit interrupt enable bit (SSR1 register bit 8: TIE) is set to 1, a send interrupt request is issued once the send data write flag bit (SSR1 register bit 11: TDRE) is set to 1. The succeeding send data can be written to the serial output data register 1 (SODR1) at interrupt processing.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Reception

- When reception is enabled (SCR1 register bit 9: RXE = 1), receiving is always performed.
- When the start bit of receive data is detected, the serial input data register 1 (SIDR1) receives one frame of data and stores data to the serial input data register 1 (SIDR1) according to the data format specified in the serial control register 1 (SCR1).
- At completion of receiving one frame of data, the receive data load flag bit (SSR1 register bit 12: RDRF) is set to 1.
- When the status of the error flag of the serial status register 1 (SSR1) is checked to find normal reception at the completion of one frame of data, read the receive data from the serial input data register 1 (SIDR1). When a receive error occurs, perform error processing.
- The receive data load flag bit (SSR1 register bit 12: RDRF) is cleared to 0 when receive data is read.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

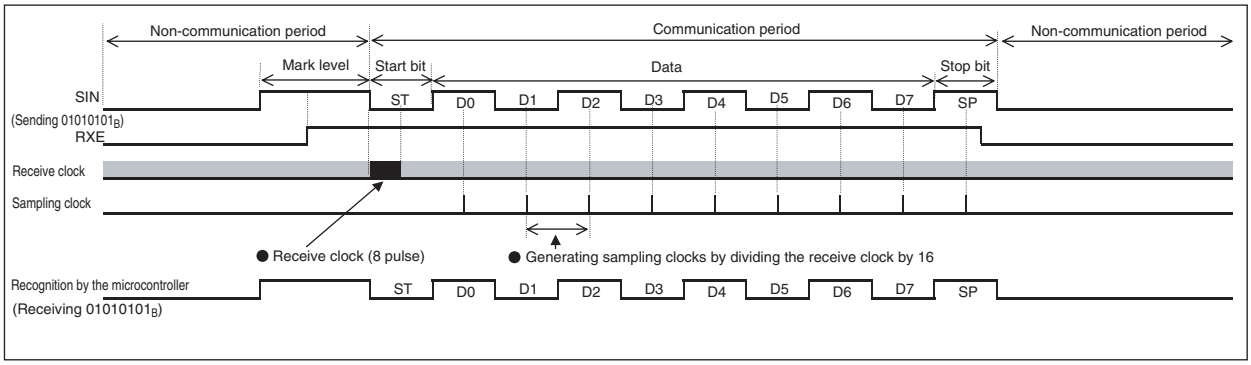
www.DataSheet4U.com

● Detecting the start bit

Implement the following settings to detect the start bit:

- Set the communication line level to H (attach the mark level) before the communication period.
- Specify reception permission (RXE = H) while the communication line level is H (mark level).
- Do not specify reception permission (RXE = H) for periods other than the communication period (without mark level). Otherwise, data is not received correctly.
- After the stop bit is detected (the RDRF flag is set to 1), specify reception inhibition (RXE = L) while the communication line level is H (mark level).

Figure 14.6-2 Example of normal operation

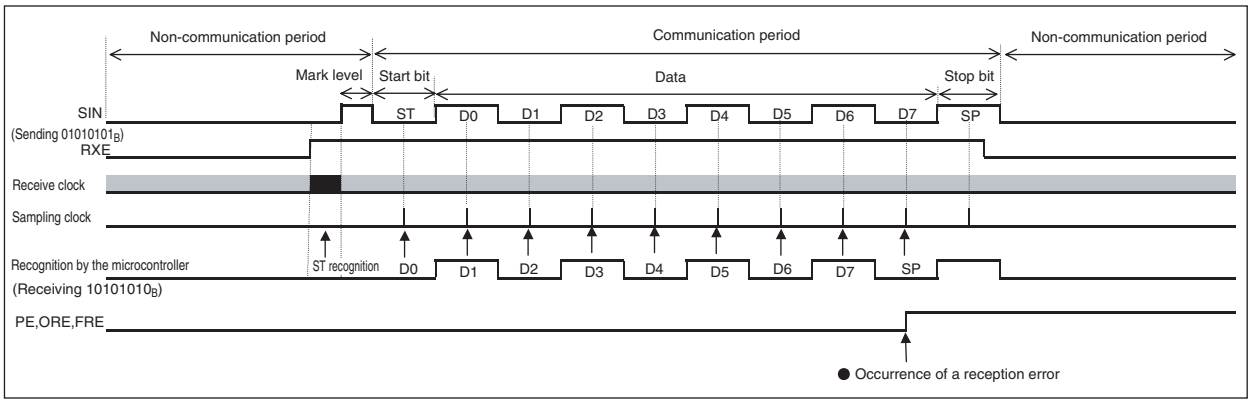


et4U.com
DataShee
DataSheet4U.com

Note that specifying reception permission at the timing shown below obstructs the correct recognition of the input data (SIN) by the microcontroller.

- Example of operation if reception permission (RXE = H) is specified while the communication line level is L.

Figure 14.6-3 Example of abnormal operation



● Stop bit

During transmission, one bit or two bits can be selected. However, the receive side always detects only the first bit.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Error detection

- In operation mode 0 (asynchronous normal mode), parity, overrun, and frame errors can be detected.
- In operation mode 1 (asynchronous multiprocessor mode), overrun and frame errors can be detected, but parity errors cannot be detected.

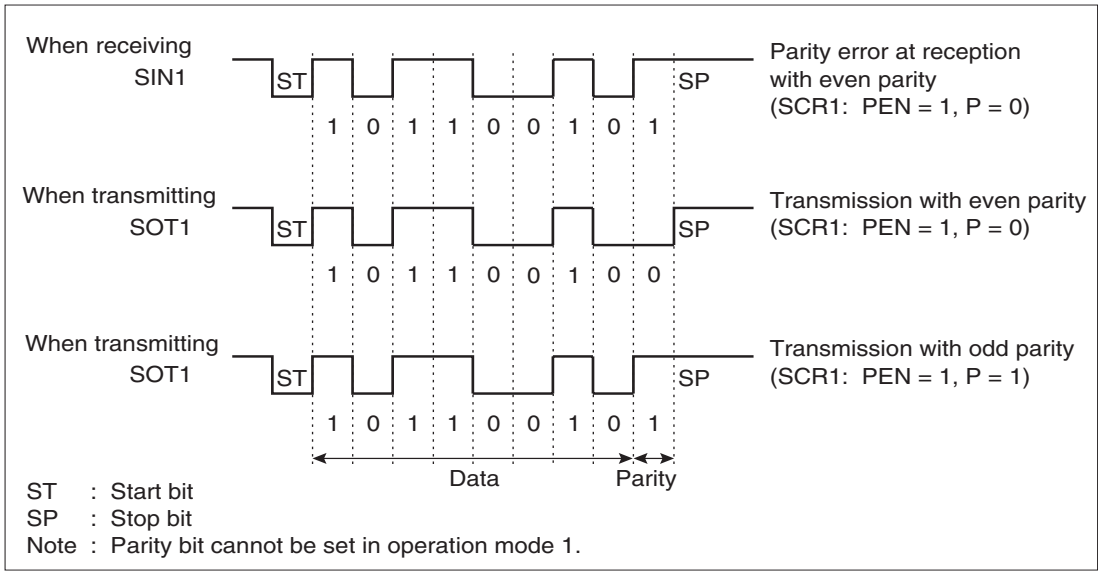
● Parity bit

A parity bit can be set only in operation mode 0 (asynchronous normal mode). The parity addition enable bit (SCR1 register bit 15: PEN) is used to specify whether there is parity or not, and the parity select bit (SCR1 register bit 14: P) is used to select odd or even parity.

There is no parity bit in operation modes 1 (asynchronous multiprocessor mode).

The transmit/receive data when the parity bit enabled are shown in Figure 14.6-4.

Figure 14.6-4 Transmit/Receive Data when Parity Bit Enabled



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.6.2 Operation in Clock Synchronous Mode (Operation Mode 2)

When the UART1 is used in operation mode 2, the transfer mode is clock synchronous.

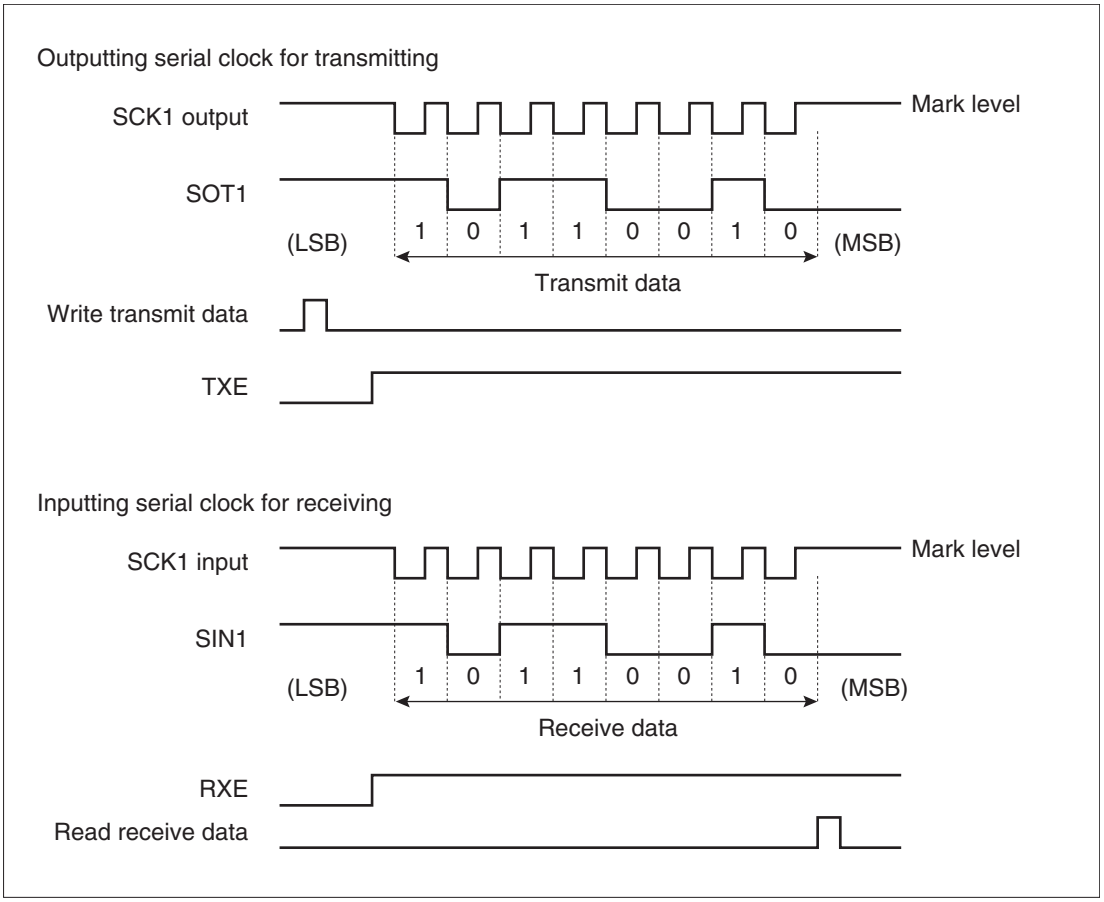
■ Operation in Clock Synchronous Mode (Operation Mode 2)

● Format of transmit/receive data

In the clock synchronous mode, 8-bit data is transmitted/received on LSB-first, and the start and stop bits are not added.

Figure 14.6-5 shows the transmit/receive data format for the clock synchronous mode.

Figure 14.6-5 Format of Transmit/Receive Data (Operation Mode 2)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Clock supply

In the clock synchronous mode, count of clocks equal to the transmit and receive bits count must be supplied.

- When the internal clock (dedicated baud rate generator or internal timer) has already selected (SMR1 register bit 5 to 3: CS2 to CS0 = "000_B" to "101_B" or "110_B") and data is transmitted, the synchronous clock for data reception is generated automatically.
- When the external clock has already selected (SMR1 register bit 5 to 3: CS2 to CS0 = "111_B"), the clock for exact one byte must be supplied from outside after ensuring that data is present (SSR1 register bit 11: TDRE = 0) in the serial output data register 1 (SODR1). Also, before and after transmitting, always return to the mark level (High level).

● Error detection

Only overrun errors can be detected; parity and framing errors cannot be detected.

● Setting of register

Table 14.6-2 shows the setting of the control register in transmitting serial data from the transmitting end to the receiving end using the clock synchronous mode (operation mode 2).

Table 14.6-2 Setting of Control Register

Register Name	Bit Name	Setting	
		Transmit End (output serial clock)	Receive End (input serial clock)
Serial mode register 1 (SMR1)	MD1, MD0	Set clock synchronous mode (MD1, MD0 = "10_B").	
	CS2, CS1, CS0	Set clock input source. • Dedicated baud rate generator (CS2 to CS0 = "000_B" to "101_B") • Internal timer (CS2 to CS0 = "110_B")	Set clock input source. • External clock (CS2 to CS0 = "111_B")
	SCKE	Set serial clock output (SCKE = 1).	Set serial clock input (SCKE = 0).
	SOE	Set serial data output pin (SOE = 1).	Set general-purpose I/O port (SOE = 0).
Serial control register 1 (SCR1)	PEN	Do not add parity bit (PEN = 0).	
	CL	8-bit data length (CL = 1)	
	REC	Initialize error flag (REC = 0).	
	TXE	Enable transmitting (TXE = 1).	Disable transmitting (TXE = 0).
	RXE	Disable receiving (RXE = 0).	Enable receiving (RXE = 1).
Serial status register 1 (SSR1)	TIE	Enable transmitting interrupt (TIE = 1)	Disable transmitting interrupt (TIE = 0)
	RIE	Disable receiving interrupt (RIE = 0).	Enable receiving interrupt (RIE = 1).

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Starting communications

When send data is written to the serial output data register 1 (SODR1), communication is started. When starting communication only in receiving, it is always necessary to write dummy send data to the serial output data register 1 (SODR1).

● Terminating communications

After transmitting and receiving of one frame of data, the receive data load flag bit (SSR1 register bit 12: RDRF) is set to 1. When data is received, check the overrun error flag bit (SSR1 register bit 14: ORE) to ensure that the communication has performed normally.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.6.3 Bidirectional Communication Function (Operation Modes 0 and 2)

In operation modes 0 and 2 (asynchronous normal mode, clock synchronous mode), normal serial bidirectional communications using 1-to-1 connection can be performed. For operation mode 0 (asynchronous normal mode), the asynchronous mode is used; for operation mode 2 (clock synchronous mode), the clock synchronous mode is used.

■ Bidirectional Communication Function

To operate the UART1 in the operation mode 0, 2 (asynchronous normal mode, clock synchronous mode), shown in Figure 14.6-6 is required.

Figure 14.6-6 Setting of Operation Modes 0, 2 (Asynchronous Normal Mode and Clock Synchronous Mode) for UART1

	bit15	14	13	12	11	10	9	bit8	bit7	6	5	4	3	2	1	bit0
SCR1, SMR1	PEN	P	SBL	CL	AD	REC	RXE	TXE	MD1	MD0	CS2	CS1	CS0	Reserved	SCKE	SOE
Operation mode 0 →	◎	◎	◎	◎	X	0	◎	◎	0	0	◎	◎	◎	0	◎	◎
Operation mode 2 →	0	X	X	1	X	0	◎	◎	1	0	◎	◎	◎	0	◎	◎

	PE	ORE	FRE	RDRF	TDRE	–	RIE	TIE	Setting of transmit data (at write) / Retention of receive data (at read)
SSR1, SIDR1/SODR1									
Operation mode 0 →	◎	◎	◎	◎	◎		◎	◎	
Operation mode 2 →	X	◎	X	◎	◎		◎	◎	

DDR Port direction register

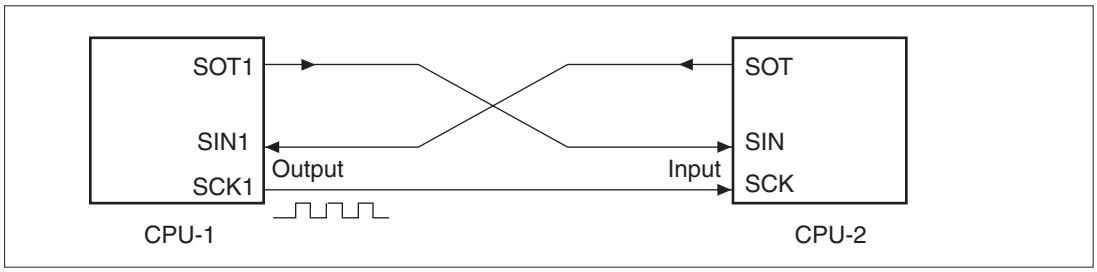
Set the bit to 0 corresponding to pin used as SIN1 and SCK1 input pins.

–: Unused bit
◎: Used bit
X: Undefined bit
1: Set 1
0: Set 0

● Inter-CPU connect

Connect the two CPUs as shown in Figure 14.6-7.

Figure 14.6-7 Example of Bidirectional Communication Connect for UART1



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

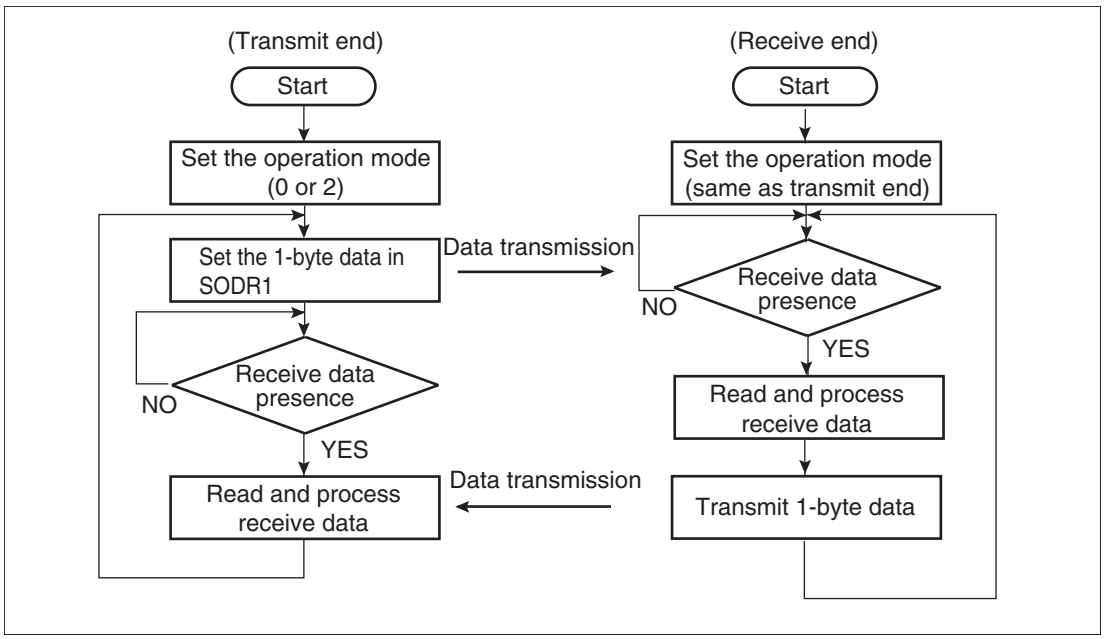
www.DataSheet4U.com

● Communication procedure

Communications start at any timing from the transmitting end when transmit data is provided. At the transmitting end, set transmit data in the serial output data register (SODR1) and set the transmitting enable bit in the serial control register (SCR1 register bit 8: TXE) to 1 to start transmitting.

Figure 14.6-8 gives an example of transferring receive data to the transmitting end to inform the transmitting end of normal reception.

Figure 14.6-8 Flowchart for Bidirectional Communication



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.6.4 Master/Slave Type Communication Function (Multiprocessor Mode)

Operation mode 1 (asynchronous multiprocessor mode) enables communications by the master/slave type connection of more than one CPU. Only the master CPU functions.

■ Master/Slave Type Communication Function

To operate the UART1 in operation mode 1 (asynchronous multiprocessor mode), the setting shown in Figure 14.6-9 is required.

Figure 14.6-9 Setting of Operation Mode 1 (Asynchronous Multiprocessor Mode) for UART1

	bit15	14	13	12	11	10	9	bit8	bit7	6	5	4	3	2	1	bit0
SCR1, SMR1	PEN	P	SBL	CL	AD	REC	RXE	TXE	MD1	MD0	CS2	CS1	CS0	Reserved	SCKE	SOE
	0	X	◎	1	◎	0	◎	◎	0	1	◎	◎	◎	0	0	◎

SSR1, SIDR1/SODR1	PE	ORE	FRE	RDRF	TDRE	–	RIE	TIE	Setting of transmit data (at write) / Retention of receive data (at read)
	X	◎	◎	◎	◎		◎	◎	

DDR Port direction register

Set the bit to 0 corresponding to pin used as SIN1 and SCK1 input pins.

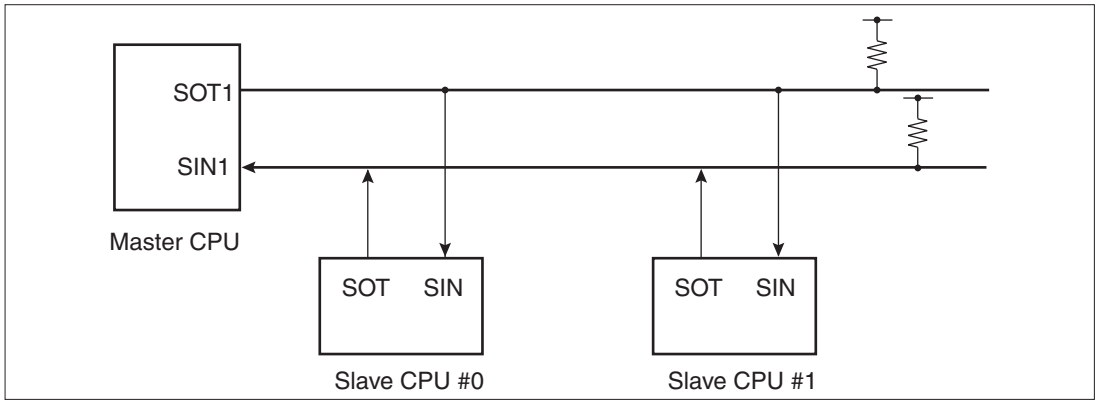
–: Unused bit
◎: Used bit
X: Undefined bit
1: Set 1
0: Set 0

DataSheet4U.com

● Inter-CPU connect

One master CPU and more than one slave CPU are connected to two common communication lines to compose the communication system. The UART1 can be used only as the master CPU.

Figure 14.6-10 Example of Master/Slave Type Communication Connect for UART1



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Function select

At master/slave type communication, select the operation mode and data transfer type.

Since the parity check function cannot be used in operation mode 1 (asynchronous multiprocessor mode), set the parity add enable bit (SCR1 register bit 15: PEN) to 0.

Table 14.6-3 Select of Master/Slave Type Communication Function

	Operation Mode		Data	Parity	Synchronous System	Stop Bit
	Master CPU	Slave CPU				
Address transmit/ receive	Operation mode 1	–	A/D = 1 + 8-bit address	Not provided	Asynchronous	1 bit or 2 bits
Data transmit/ receive			A/D = 0 + 8-bit data			

● Communication procedure

Communications start when the master CPU transmits address data.

The address data is data with the A/D bit set to 1. The address/data select bit (SCR1 register bit 11: A/D) is added to select the slave CPU that the master CPU communicates with. When the program identifies address data and finds a match with the allocated address, each slave CPU starts communications with the master CPU.

Figure 14.6-11 shows the flowchart for master/slave type communications.

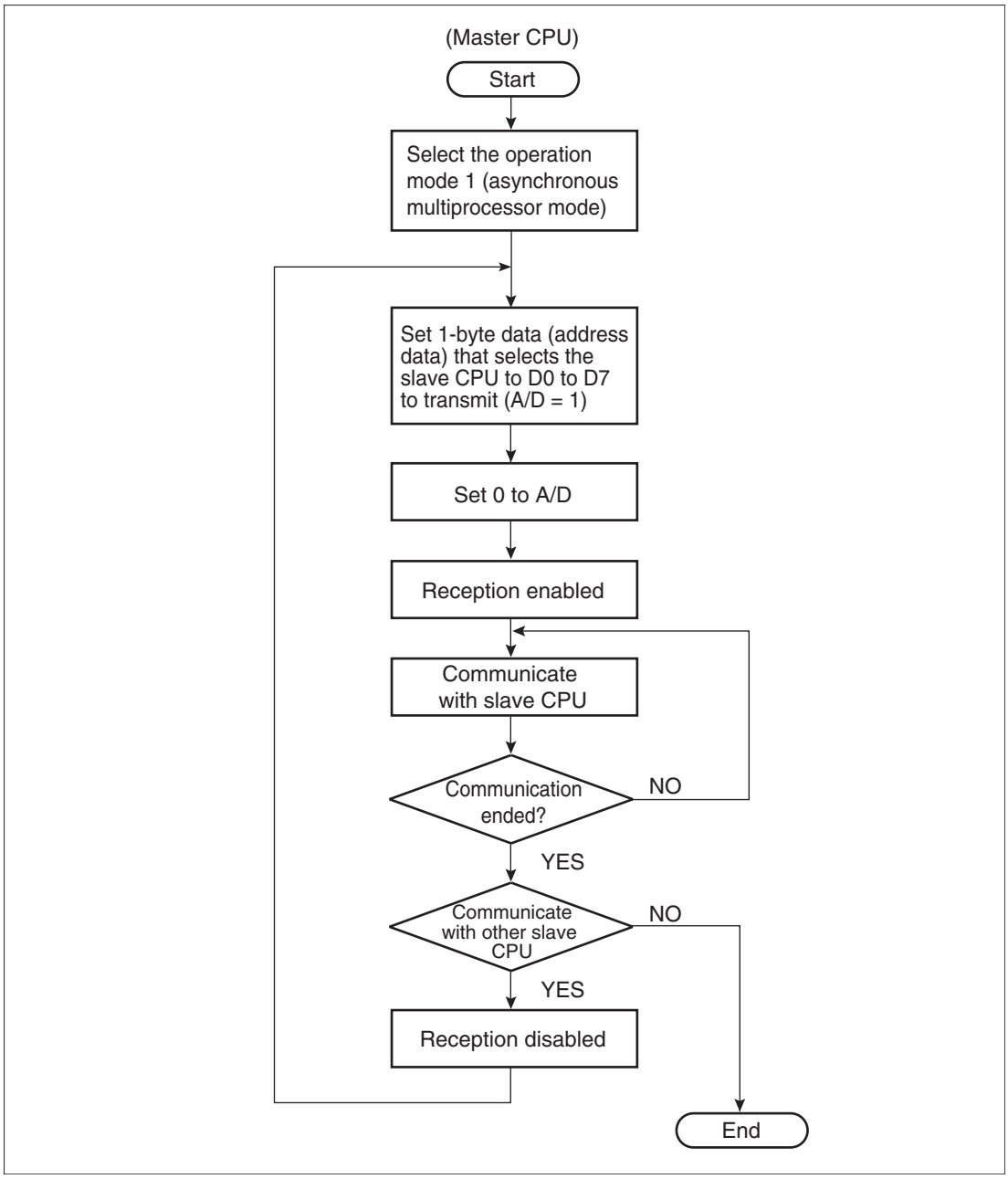
DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 14.6-11 Flowchart for Master/Slave Type Communications



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.7 Precautions when Using UART1

Use of the UART1 requires the following cautions.

■ Precautions when Using UART1

● Enabling sending and receiving

The send enable bit (SCR1 register bit 8: TXE) and receive enable bit (SCR1 register bit 9: RXE) are provided for sending and receiving.

- In the initial state after reset, both sending and receiving are disabled (SCR1 register bit 8: TXE = 0, bit 9: RXE = 0). Therefore, it is necessary to enable sending and receiving.
- Sending and receiving are disabled to stop (SCR1 register bit 8: TXE = 0, bit 9: RXE = 0).

● Setting operation mode

Set the operation mode after disabling sending and receiving (SCR1 register bit 8: TXE = 0, bit 9: RXE = 0). When the operation mode is changed during sending and receiving, the sent and received data is not assured.

et4U.com

● Clock synchronous mode

DataShee

Operation mode 2 (clock synchronous mode) is set as the clock synchronous mode. Send and receive data do not have the start and stop bits.

DataSheet4U.com

● Timing of enabling send interrupt

The initial value after reset of the send data write enable flag bit (SSR1 register bit 11: TDRE) is set at 1 (no send data, send data write enabled). Therefore, the send interrupt is enabled (SSR1 register bit 8: TIE = 1) and a send interrupt request is issued simultaneously. Always prepare send data and enable a send interrupt (SSR1 register bit 8: TIE = 1).

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

14.8 Program Example for UART1

This section gives a program example for the UART1.

■ Program Example for UART1

● Processing

The bidirectional communication function (normal mode) of the UART1 is used to perform serial transmission/reception.

- Set operation mode 0, asynchronous mode (normal), 8-bit data length, 2-bit stop bit length, and no parity.
- Use the P40/SIN1 and P42/SOT1 pins for communications.
- Use the dedicated baud rate generator to set the baud rate to approximately 9600 bps.
- Transmit the character 13_H from the SOT1 pin and receive it at an interrupt.
- Assume the machine clock (ϕ) 16 MHz.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Coding example

```
ICR13   EQU  0000BDH           ; UART Transmit/receive interrupt control register
DDR1    EQU  000011H           ; Port 1 data direction register
CDCR1   EQU  00001BH           ; Communication prescaler register 1
SMR1    EQU  000024H           ; Mode control register 1
SCR1    EQU  000025H           ; Control register 1
SIDR1   EQU  000026H           ; Input data register 1
SODR1   EQU  000026H           ; Output data register 1
SSR1    EQU  000027H           ; Status register 1
REC     EQU  SCR1:2            ; Receive error flag clear bit
;-----Main program-------------------------------------------------------------
CODE    CSEG ABS=0FFH
START:
;        :                     ; Assume stack pointer (SP) already reset
        AND  CCR,#0BFH         ; Disable interrupt
        MOV  I:ICR13,#00H      ; Interrupt level 0 (highest priority)
        MOV  I:DDR1,#00000000B ; Set SIN1 as input pin.
        MOV  I:CDCR1,#080H     ; Enable communication prescaler
        MOV  I:SMR1,#00010001B ; Operation mode 0 (asynchronous)
                               ; Use dedicated baud rate generator (9615 bps)
                               ; Disable clock pulse output and enable data output
        MOV  I:SCR1,#00010011B ; Without N parity.  2-bit stop bit
                               ; Clear 8-bit data bit and receive error flag
                               ; Enable transmitting/receiving
        MOV  I:SSR1,#00000010B ; Disable transmit interrupt and enable receive
                               ; interrupt
        MOV  I:SODR1,#13H      ; Write send data
        MOV  ILM,#07H          ; Set ILM of PS to level 7
        OR   CCR,#40H          ; Enable interrupt
LOOP:   MOV  A,#00H            ; Infinite loop
        MOV  A,#01H
        BRA  LOOP
;-----Interruption program-----------------------------------------------------
WARI:
        MOV  A,SIDR1           ; Read receive data
        CLRB I:REC             ; Clear receive interrupt request flag
;        :
;       Processing by user
;        :
        RETI                   ; Return from interrupt
CODE    ENDS
;-----Vector setting-----------------------------------------------------------
VECT    CSEG ABS=0FFH
        ORG  0FF68H            ; Set interrupt #37 (25H) vector
        DSL  WARI
        ORG  0FFDCH            ; Set reset vector
        DSL  START
        DB   00H               ; Set single-chip mode
VECT    ENDS
```

et4U.com
DataShee
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 15

CAN CONTROLLER

This chapter explains the functions and operations of the CAN controller.

15.1 Overview of CAN Controller
15.2 Block Diagram of CAN Controller
15.3 Configuration of CAN Controller
15.4 Interrupts of CAN Controller
15.5 Explanation of Operation of CAN Controller
15.6 Precautions when Using CAN Controller
15.7 Program Example of CAN Controller

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.1 Overview of CAN Controller

The CAN (controller area network) is a serial communication protocol conformed to CAN Ver. 2.0A and Ver. 2.0B. Transmitting and receiving can be performed in the standard frame format and the extended frame format.

■ Overview of CAN Controller

- The CAN controller format conforms to CAN Ver. 2.0A and Ver. 2.0B.
- Transmitting and receiving can be performed in the standard frame format and the extended frame format.
- Data frames can be transmitted automatically by remote frames receiving.
- The baud rate ranges from 10 kpbs to 1 Mbps (at 16-MHz machine clock frequency).

Table 15.1-1 Data Transfer Baud Rate

Machine Clock	Baud Rate
16 MHz	1 Mbps
12 MHz	1 Mbps
8 MHz	1 Mbps
4 MHz	500 Kbps
2 MHz	250 Kbps

et4U.com DataSheet4U.com DataShee

- The CAN controller equips eight transmit/receive message buffers.
- The standard frame format provides transmitting and receiving with 11-bit ID and the extended frame format 29-bit ID.
- Message data can be set from 0 byte to 8 bytes.
- Message buffer configuration can be performed at a multilevel.
- The CAN controller has two acceptance mask registers. These registers can set masks independently for the receive message ID.
- The two acceptance mask registers can receive in the format of standard frame and extended frame.
- Four masks can be set at all bit comparison and masking, and partially at acceptance mask registers 0 and 1.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

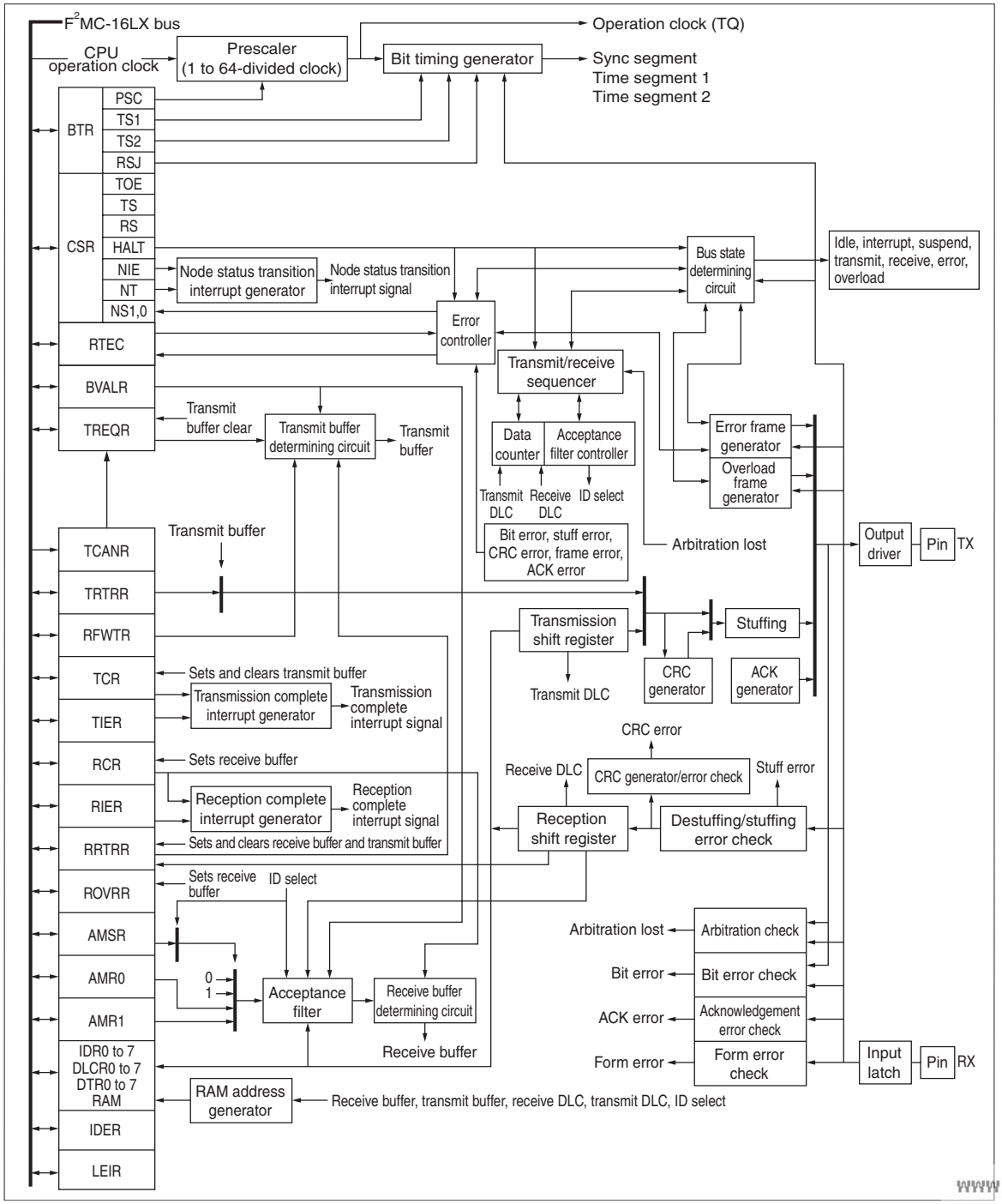
www.DataSheet4U.com

15.2 Block Diagram of CAN Controller

The CAN controller consists of two types of registers; one controls the CAN controller and the other controls each message buffer.

■ Block Diagram of CAN Controller

Figure 15.2-1 Block Diagram of CAN Controller



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

The pin names in the block diagram are as follows:

TX pin: P43/TX

RX pin: P44/RX

- Bit timing register (BTR)

  This register sets the division ratio at which CAN bit timing is generated.

- Control status register (CSR)

  This register controls the operation of the CAN controller. It indicates the state of transmitting/receiving and the CAN bus, controls interrupts, and controls the bus halt and indicates its state.

- Receive/transmit error counter register (RTEC)

  This register indicates the number of times transmit and receive errors have occurred. It counts up when an error occurs in transmitting and receiving messages and counts down when transmitting and receiving are performed normally.

- Message buffer validating register (BVALR)

  This register enables or disables a specified message buffer, and also indicates the enabled/disabled status.

- IDE register (IDER)

  This register sets the frame format of each message buffer. It sets the standard frame format or extended frame format.

- Transmit request register (TREQR)

  This register sets a transmit request to each message buffer.

- Transmit cancel register (TCANR)

  This register cancels transmit requests held in each message buffer.

- Transmit RTR register (TRTRR)

  This register selects a frame format transmitted to each message buffer. It selects the data frame or remote frame.

- Remote frame receive waiting register (RFWTR)

  This register sets the condition for transmitting start when a transmit request of the data frame is set.

- Transmit complete register (TCR)

  The bit is set which is corresponding to the number of the message buffer that completes message transmitting.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Transmit complete interrupt enable register (TIER)

  This register controls the generation of an interrupt request when each message buffer completes transmitting. When an interrupt is enabled, an interrupt request is generated when transmitting is completed.

- Receive complete register (RCR)

  This register sets the bit corresponding to the number of the message buffer that completes receiving message.

- Receive complete interrupt enable register (RIER)

  This register controls output of an interrupt request when each message buffer completes receiving. If output of an interrupt request is enabled, an interrupt request is output at completion of receiving.

- Receive RTR register (RRTRR)

  When a remote frame is stored in a message buffer, the bit corresponding to the number of the message buffer is set.

- Receive overrun register (ROVRR)

  This register sets the bit corresponding to the number of the message buffer that overruns when the message is received.

et4U.com DataShee

DataSheet4U.com

- Acceptance mask select register (AMSR)

  This register sets the method for masking the receive message for each message buffer.

- Acceptance mask registers (AMR0 and AMR1)

  These registers set a mask with the ID for filtering the message to be received.

- Last event indication register (LEIR)

  This register indicates the operating state that last occurred. It indicates that either node status transition, transmitting completion, or receiving completion occurred.

- Prescaler

  The prescaler generates a bit timing clock at a frequency of 1/1 to 1/64 of the system clock.

  It sets the operation clock (TQ).

- Bit timing generator

  This generator detects a bit timing clock signal to generate a sync segment and time segments 1 and 2.

- Node status transition interrupt generator

  This generates a node status transition interrupt signal when the node status transits.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Bus state identification circuit

  This circuit identifies the CAN bus state from the bus halt bit (CSR: HALT) and the signal from the error frame generator.

- Acceptance filter

  This filter compares the receive message ID with the acceptance code to select the message to be received.

- Transmit message buffers/receive message buffers

  There are 8 message buffers to store the message to be transmitted and received.

- CRC generator/ACK generator

  This circuit generates a CRC field or an ACK field when a data frame or remote frame is transmitted.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.3 Configuration of CAN Controller

This section explains the pins and, related registers, interrupt factors of the CAN controller.

■ Pins of CAN Controller

Table 15.3-1 Pins of CAN Controller

Pin Name	Pin Function	Setting of Pin Used in CAN Controller
TX	Transmit output pin General-purpose I/O port	Specify TX pin as transmit output pin (when TOE bit in CSR register set to 1)
RX	Receive input pin General-purpose I/O port	Specify RX pin as receive input pin (when bit 4 in DDR4 register set to 0)

■ Block Diagram for Pins of CAN Controller

See "CHAPTER 4 I/O PORT" for details of the block diagram of pins.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ CAN Controller Registers

Figure 15.3-1, Figure 15.3-2 and Figure 15.3-3 list the registers of the CAN controller.

Figure 15.3-1 Registers of CAN Controller (Control Registers)

CAN controller control register

bit 15 ····· bit 8	bit 7 ····· bit 0	Reset value
Reserved area*	BVALR (Message buffer enable register)	00000000_B
Reserved area*	TREQR (Transmission request register)	00000000_B
Reserved area*	TCANR (Transmission cancel register)	00000000_B
Reserved area*	TCR (Transmission complete register)	00000000_B
Reserved area*	RCR (Reception complete register)	00000000_B
Reserved area*	RRTRR (Reception RTR register)	00000000_B
Reserved area*	ROVRR (Reception overrun register)	00000000_B
Reserved area*	RIER (Reception complete interrupt enable register)	00000000_B

bit 15 ····· bit 8	bit 7 ····· bit 0	Reset value	
CSR (Control status register)		$00XXX000_B$	$0XXXX001_B$
Reserved area*	LEIR (Last event indicate register)		$000XX000_B$
RTEC (Receive/transmit error counter)		00000000_B	00000000_B
BTR (Bit timing register)		$X1111111_B$	11111111_B
Reserved area*	IDER (IDE register)		$XXXXXXXX_B$
Reserved area*	TRTRR(Transmission RTR register)		00000000_B
Reserved area*	RFWTR (Remote frame receive waiting register)		$XXXXXXXX_B$
Reserved area*	TIER (Transmission complete interrupt enable register)		00000000_B
AMSR (Acceptance mask select register)		$XXXXXXXX_B$	$XXXXXXXX_B$
Reserved area*			
AMR0 (Acceptance mask register 0)		$XXXXXXXX_B$ $XXXXXXXX_B$	$XXXXXXXX_B$ $XXXXXXXX_B$
AMR1 (Acceptance mask register 1)		$XXXXXXXX_B$ $XXXXXXXX_B$	$XXXXXXXX_B$ $XXXXXXXX_B$

*: Reserved area cannot be used because address is used in the system.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 15.3-2 Registers of CAN Controller (ID Register and DLC Register)

Message buffer (ID register)

bit 15 ····· bit 8 bit 7 ····· bit 0	Reset value	
RAM (General-purpose RAM) (16 bytes)	$XXXXXXXX_B$ ~	~ $XXXXXXXX_B$
IDR0 (ID register 0)	$XXXXXXXX_B$ $XXXXXXXX_B$	$XXXXXXXX_B$ $XXXXXXXX_B$
IDR1 (ID register 1)	$XXXXXXXX_B$ $XXXXXXXX_B$	$XXXXXXXX_B$ $XXXXXXXX_B$
IDR2 (ID register 2)	$XXXXXXXX_B$ $XXXXXXXX_B$	$XXXXXXXX_B$ $XXXXXXXX_B$
IDR3 (ID register 3)	$XXXXXXXX_B$ $XXXXXXXX_B$	$XXXXXXXX_B$ $XXXXXXXX_B$
IDR4 (ID register 4)	$XXXXXXXX_B$ $XXXXXXXX_B$	$XXXXXXXX_B$ $XXXXXXXX_B$
IDR5 (ID register 5)	$XXXXXXXX_B$ $XXXXXXXX_B$	$XXXXXXXX_B$ $XXXXXXXX_B$
IDR6 (ID register 6)	$XXXXXXXX_B$ $XXXXXXXX_B$	$XXXXXXXX_B$ $XXXXXXXX_B$
IDR7 (ID register 7)	$XXXXXXXX_B$ $XXXXXXXX_B$	$XXXXXXXX_B$ $XXXXXXXX_B$

Message buffer (DLC register)

bit 15 ····· bit 8	bit 7 ····· bit 0	Reset value
Reserved area*	DLC0 (DLC register 0)	$XXXXXXXX_B$
Reserved area*	DLC1 (DLC register 1)	$XXXXXXXX_B$
Reserved area*	DLC2 (DLC register 2)	$XXXXXXXX_B$
Reserved area*	DLC3 (DLC register 3)	$XXXXXXXX_B$
Reserved area*	DLC4 (DLC register 4)	$XXXXXXXX_B$
Reserved area*	DLC5 (DLC register 5)	$XXXXXXXX_B$
Reserved area*	DLC6 (DLC register 6)	$XXXXXXXX_B$
Reserved area*	DLC7 (DLC register 7)	$XXXXXXXX_B$

*: Reserved area cannot be used because address is used in the system.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 15.3-3 Registers of CAN Controller (DTR Register)

Message buffer (DTR register)

bit 15 ····· bit 8	bit 7 ····· bit 0	Reset value
DTR0 (Data register 0) (8 bytes)		$XXXXXXXX_B$ ~ $XXXXXXXX_B$
DTR1 (Data register 1) (8 bytes)		$XXXXXXXX_B$ ~ $XXXXXXXX_B$
DTR2 (Data register 2) (8 bytes)		$XXXXXXXX_B$ ~ $XXXXXXXX_B$
DTR3 (Data register 3) (8 bytes)		$XXXXXXXX_B$ ~ $XXXXXXXX_B$
DTR4 (Data register 4) (8 bytes)		$XXXXXXXX_B$ ~ $XXXXXXXX_B$
DTR5 (Data register 5) (8 bytes)		$XXXXXXXX_B$ ~ $XXXXXXXX_B$
DTR6 (Data register 6) (8 bytes)		$XXXXXXXX_B$ ~ $XXXXXXXX_B$
DTR7 (Data register 7) (8 bytes)		$XXXXXXXX_B$ ~ $XXXXXXXX_B$
Reserved area* (128 bytes)		

*: Reserved area cannot be used because address is used in the system.

et4U.com DataShee DataSheet4U.com

■ Generation of Interrupt Request by CAN Controller

The CAN controller has a transmit complete interrupt, receive complete interrupt, and node status interrupt.

Each interrupt request is generated as follows:

- When a transmit complete interrupt is enabled for the message buffer (x) (TIER: TIEx = 1), the TCx bit in the transmit complete register is set to 1 and a transmit complete interrupt request is generated after a completion of message transmitting.
- When a receive complete interrupt is enabled for the message buffer (x) (RIER: RIEx = 1), the RCx bit in the receive complete register is set to 1 and a receive complete interrupt request is generated after a completion of message receiving.
- When a node status transition interrupt is enabled (CSR: NIE = 1), the NT bit in the CAN status register is set to 1 and a node status transition interrupt request is generated after the node status transits.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

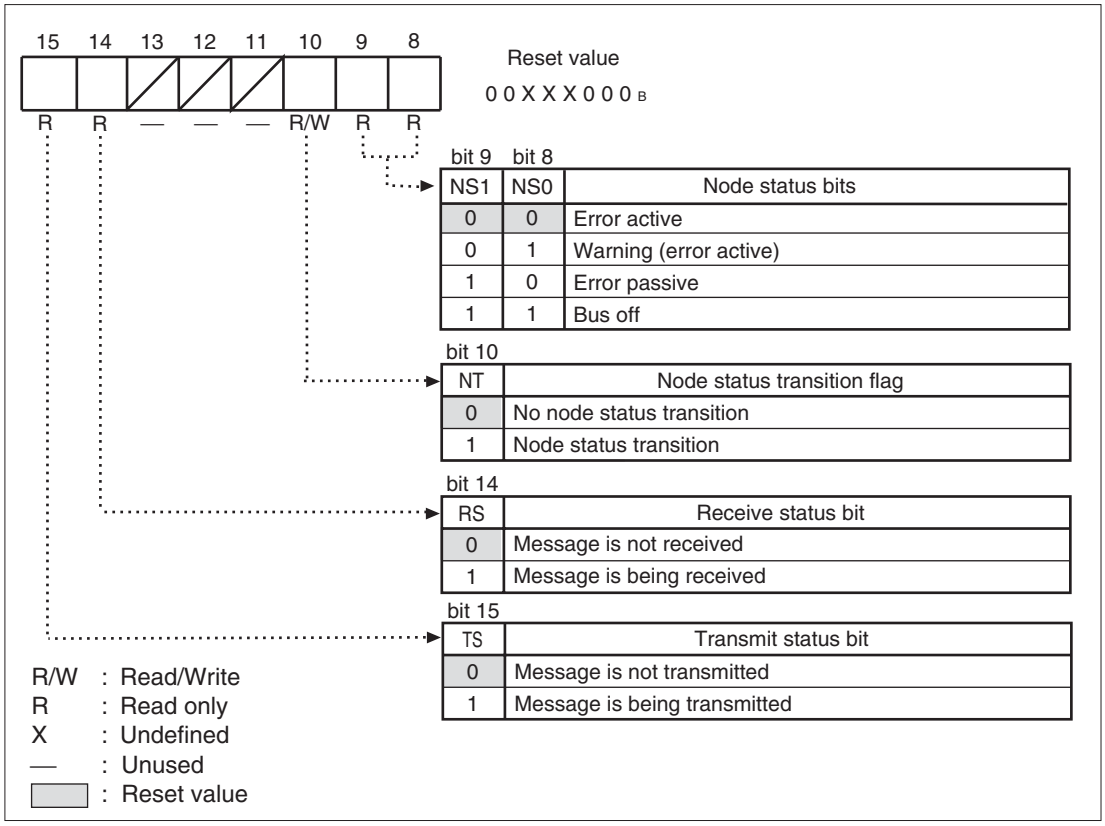
www.DataSheet4U.com

15.3.1 Control Status Register (High) (CSR: H)

The control status register (CSR) controls operation of the CAN controller. The control status register (High) (CSR: H) transmits and receives the message and indicates the node status.

■ Control Status Register (High) (CSR: H)

Figure 15.3-4 Control Status Register (High) (CSR: H)



Note: It is prohibited to execute a bit operation (read-modify-write) instruction on the lower 8 bits of control status register (CSR). Only in the case of HALT bits unchanged, use any bit operation instructions without problems (initialization of the macro instructions, etc.).

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-2 Functions of Control Status Register (High) (CSR: H)

Bit Name		Function
bit 8 bit 9	NS1, NS0: Node status bits	The combination of the NS1 and NS0 bits indicates the current node status. **00_B:** Error active **01_B:** Warning (error active) **10_B:** Error passive **11_B:** Bus off **Note:** Warning is included in error active in the CAN specifications as a node status.
bit 10	NT: Node status transition flag bit	This bit indicates that the node status transits. **When node status transits:** Bit set to 1 1. Error active (00_B) --> Warning (01_B) 2. Warning (01_B) --> Error Passive (10_B) 3. Error Passive (10_B) --> Bus off (11_B) 4. Bus off (11_B) --> Error active (00_B) (The parenthesized values are those for the NS1 and NS0 bits.) **When set to 0:** Clears this bit. **When set to 1:** Disables bit setting **Read using read modify write instructions:** 1 always read
bit 11 to bit 13	Unused bits	**Read:** Value not fixed **Write:** No effect
bit 14	RS: Receive status bit	This bit indicates whether the message is being received. **Message being received:** Bit set to 1 • For example, if the message is on the bus, even during message transmitting, this bit is set to 1 regardless of whether the receive message passes the acceptance filter. **Error frame or overload frame on bus:** Bit set to 0 • When the RS bit is 0, the bus halt state (HALT = 1), bus intermission state and bus idle state are also included.
bit 15	TS: Transmit status bit	This bit indicates whether the message is being transmitted. **Message being transmitted:** Bit set to 1 **Error frame or overload frame being transmitted:** Bit set to 0

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

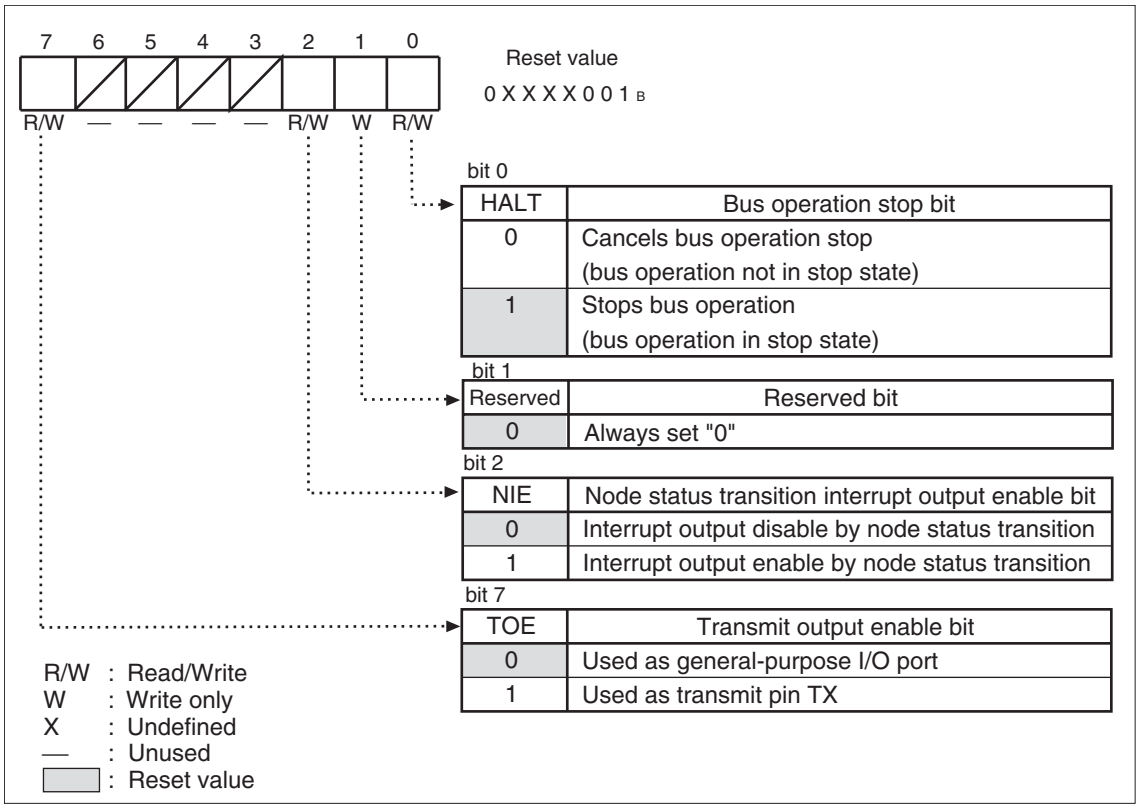
www.DataSheet4U.com

15.3.2 Control Status Register (Low) (CSR: L)

The control status register (CSR) controls operation of the CAN controller. The control status register (Low) (CSR: L) enables and disables transmit interrupt and node status transition interrupt, controls bus halt and indicates the node status.

■ Control Status Register (Low) (CSR: L)

Figure 15.3-5 Control Status Register (Low) (CSR: L)



Note:	It is prohibited to execute a bit operation (read-modify-write) instruction on the lower 8 bits of control status register (CSR). Only in the case of HALT bits unchanged, use any bit operation instructions without problems (initialization of the macro instructions, etc.).

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-3 Functions of Control Status Register (Low) (CSR: L) (1/2)

Bit Name		Function

bit 0	HALT:
	Bus halt bit
		This bit controls the bus halt. The halt state of the bus can be checked by reading this bit.
		Writing to this bit
		0: Cancels bus operation stop
		1: Sets bus operation stop
		Reading this bit
		0: Bus operation not in stop state
		1: Bus operation in stop state
		Note: When write 0 to this bit during the node status is Bus Off, ensure that 1 is written to this bit.
		Example program:
		switch (IO_CANCT0.CSR.bit.NS)
		{
		case 0 : /* error active */
		break;
		case 1 : /* warning */
		break;
		case 2 : /* error passive */
		break;
		default : /* bus off */
		for (i=0; (i <= 500) && (IO_CANCT0.CSR.bit.
		HALT == 0); i++);
		IO_CANCT0.CSR.word = 0x0084; /* HALT = 0 */
		break;
		}
		* : The variable "i" is used for fail-safe.
		[Conditions for canceling bus halt]
		• The state in which the bus is halted by a hardware reset or by writing 1 to the HALT bit is cancelled after 0 is written to the HALT bit and an 11-bit High level (receive) is input continuously to the receive input pin (RX).
		• The state in the bus off is cancelled after 0 is written to the HALT bit and an 11-bit High level (receive) is input continuously 128 times to the receive input pin (RX).
		• The values of the transmit and receive error counters are both returned to 0 and the node status transits to error active.
		• When write 0 to HALT bit during the node status is Bus Off, ensure that 1 is written to this bit.
		[State in which bus halted]
		• Transmitting and receiving are not performed.
		• A High level (receive) is output to the transmit output pin (TX).
		• Values of other register and error counter remain unchanged.
		Note: Set the bit timing register (BTR) after halting the bus.

bit 1	Reserved: Reserved bit	Always set this bit to 0. **Read:** 0 is always read.
bit 2	NIE: Node status transition interrupt output enable bit	This bit controls generation of a node status transition interrupt when the node status transits (CSR: NT = 1). **When set to 0:** Disables interrupt generation **When set to 1:** Enables interrupt generation
bit 3 to bit 6	Unused bits	**Read:** Value not fixed **Write:** No effect

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-3 Functions of Control Status Register (Low) (CSR: L) (2/2)

Bit Name		Function
bit 7	TOE: Transmit output enable bit	This bit switches between the general-purpose I/O port and the transmit pin (TX). **When set to 0:** Functions as general-purpose I/O port **When set to 1:** Functions as transmit pin (TX)

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

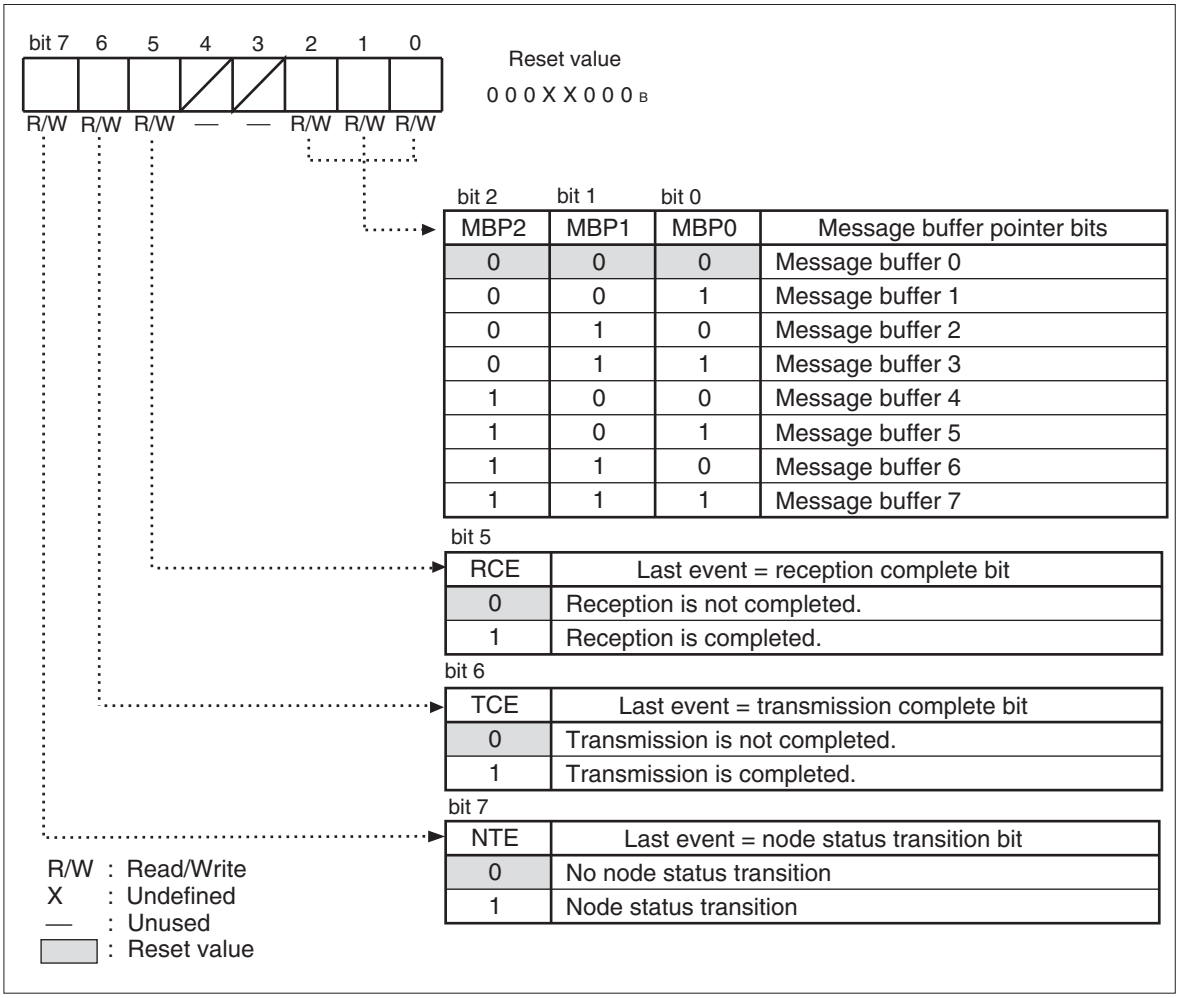
www.DataSheet4U.com

15.3.3 Last Event Indicate Register (LEIR)

This register indicates the state of the last event.

■ Last Event Indicate Register (LEIR)

Figure 15.3-6 Last Event Indicate Register (LEIR)



Note: When any of the node status transition bit (NTE), transmission complete bit (TCE), and reception complete bit (RCE) corresponding to the last event is set to 1, other bits are set to 0.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-4 Functions of Last Event Indicate Register (LEIR)

Bit Name		Function
bit 0 to bit 2	MBP2 to MBP0: Message buffer pointer bits	These bits indicate the number (x) of the message buffer where the last event occurs which is corresponding to each message buffer pointer bit. **Receiving completed:** Indicates number (x) of message buffer that completes receiving message **Transmitting completed:** Indicates number (x) of message buffer that completes transmitting message **Node status transition:** The values of the MBP2 to MBP0 bits are invalid. **When set to 0:** Cleared **When set to 1:** No effect **Read by read modify write instruction:** 1 always read
bit 3 bit 4	Unused bits	**Read:** Value not fixed **Write:** No effect on operation
bit 5	RCE: Last event reception complete bit	This bit indicates that receiving the last event is completed. **Receiving of last event completed:** Sets bit to 1 when RCx bit in reception complete register set (RCR: RCx = 1) • Nothing is related to the setting of the reception complete interrupt enable register (RIER). • The number (x) of the message buffer that completes receiving the message is indicated as the last event in the MBP2 to MBP0 bits. **When set to 0:** Cleared **When set to 1:** No effect **Read using read modify write instructions:** 1 always read
bit 6	TCE: Last event transmission complete bit	This bit indicates that the transmitting the last event is completed. **Transmitting of last event completed:** Sets bit to 1 when TCx bit in transmission complete register set (TCR: TCx = 1) • Nothing is related to the setting of the transmission complete interrupt enable register (TIER). • The number (x) of the message buffer that completes receiving the message is indicated as the last event in the MBP2 to MBP0 bits. **When set to 0:** Cleared **When set to 1:** No effect **Read using read modify write instruction:** 1 always read
bit 7	NTE: Last event node status transition bit	This bit indicates that the last event refers to the node status transition. **Last event referring to node status transition:** Sets bit to 1 when NTx bit in control status register set (CSR: NTx = 1) • The NTE bit is set to 1 at the same time that the TCx in the transmission complete register (TCR) is set. • Nothing is related to the setting of the NIE bit in the control status register (CSR). **When set to 0:** Cleared **When set to 1:** No effect **Read by read modify write instruction:** 1 always read

Note: When the last event indicate register (LEIR) is accessed in interrupt processing of the CAN controller, the event causing the interrupt does not always match the event indicated by the last event indicate register (LEIR). Other event may occur before the last event indicate register (LEIR) is accessed in interrupt processing after an interrupt request is generated.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

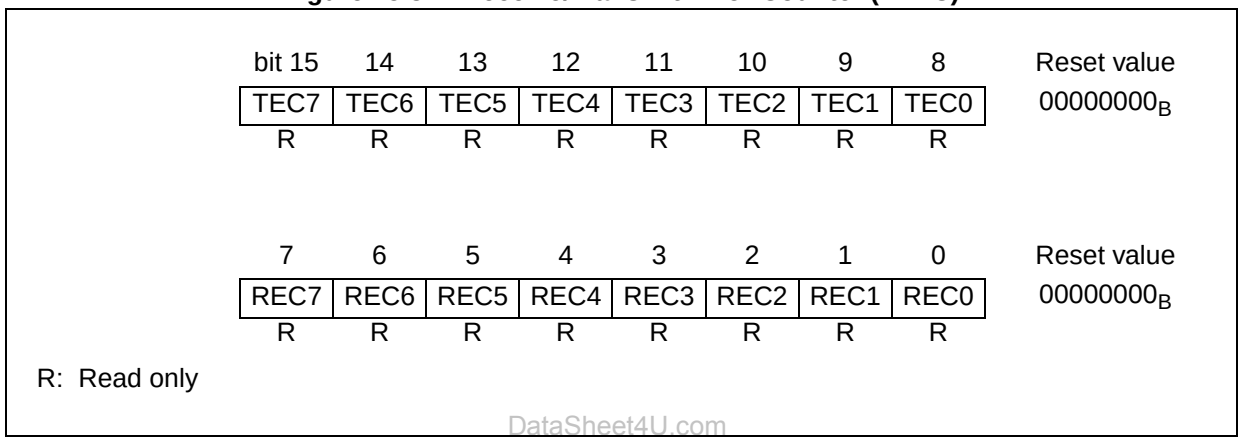
www.DataSheet4U.com

15.3.4 Receive/Transmit Error Counter (RTEC)

The receive/transmit error counter (RTEC) indicates the number of times an error occurs at transmitting and receiving the message. It counts up when transmit or receive errors occurs and counts down when transmitting and receiving are performed normally.

■ Receive/Transmit Error Counter (RTEC)

Figure 15.3-7 Receive/Transmit Error Counter (RTEC)



DataSheet4U.com

Table 15.3-5 Functions of Receive/Transmit Error Counter (RTEC)

Bit Name		Function
bit 0 to bit 7	REC7 to REC0: Receive error counter bits	**Receive error counter value = 96 or more:** Node status transits to warning (CSR: NS1, NS0 = 01_B) **Receive error counter value = 128 or more:** Node status transits to error passive (CSR: NS1, NS0 = 10_B) **Receive error counter value = 256 or more:** Stops counting up. The node status remains with error passive (CSR: NS1, NS0="10_B").
bit 8 to bit 15	TEC7 to TEC0: Transmit error counter bits	**Transmit error counter value = 96 or more :** Node status transits to warning (CSR: NS1, NS0 = 01_B) **Transmit error counter value = 128 or more :** Node status transits to error passive (CSR: NS1, NS0 = 10_B) **Transmit error counter value = 256 or more :** Stops counting up. The node status transits to bus off (CSR: NS1, NS0 = 11_B).

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Node Status Transition due to Error Occurrence

In the CAN controller, the node status transits according to the error count of the receive/transmit error counter (RTEC). Figure 15.3-8 shows the node status transition.

Figure 15.3-8 Node Status Transition

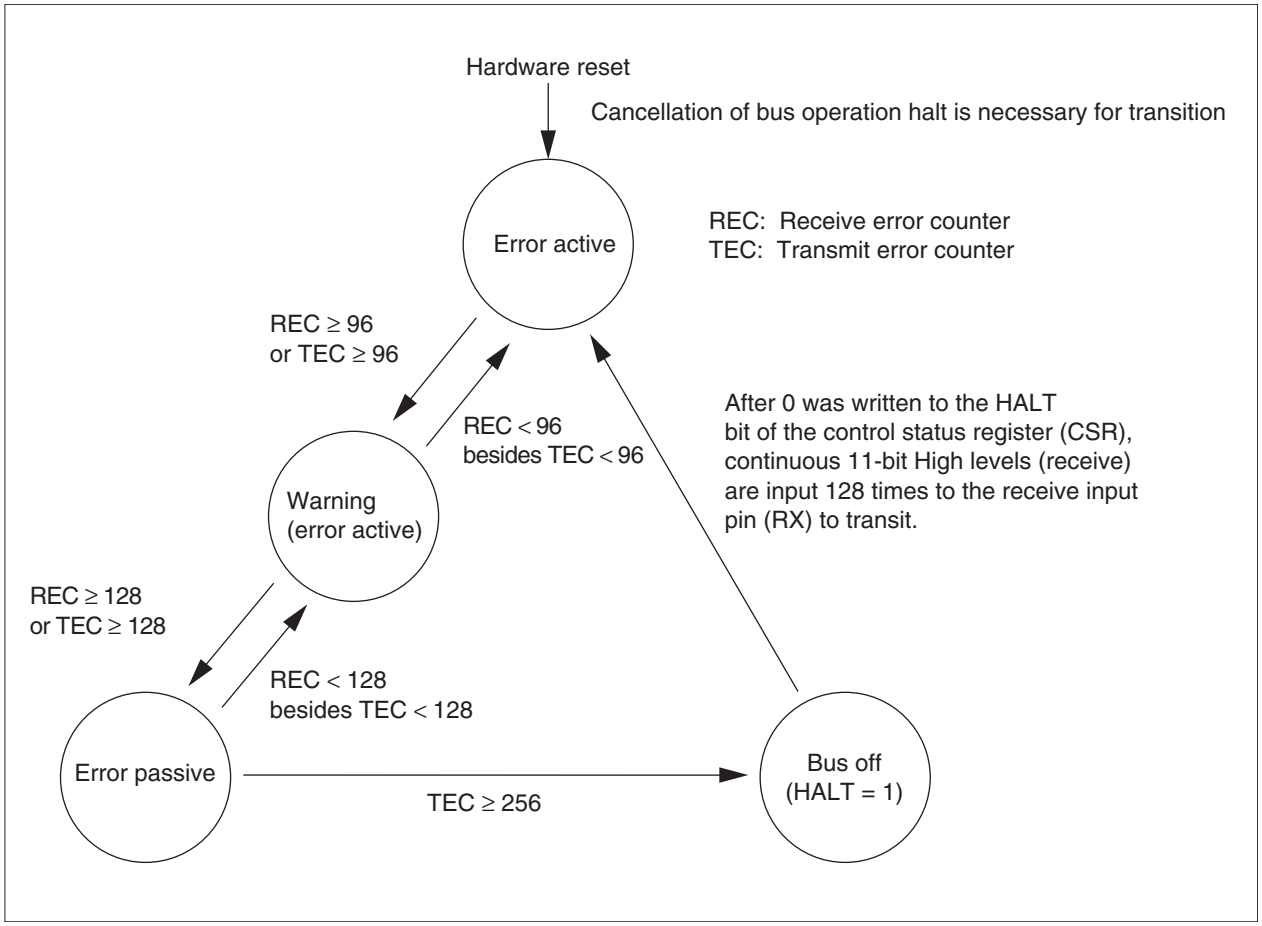


Table 15.3-6 Node Status

Node Status	State of CAN Bus
Error active	Normal state
Warning	A bus fault occurs
Error passive	
Bus off	Communications are disabled. The CAN controller is completely isolated from the CAN bus. (To return to the normal state, perform the steps in the above figure.)

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

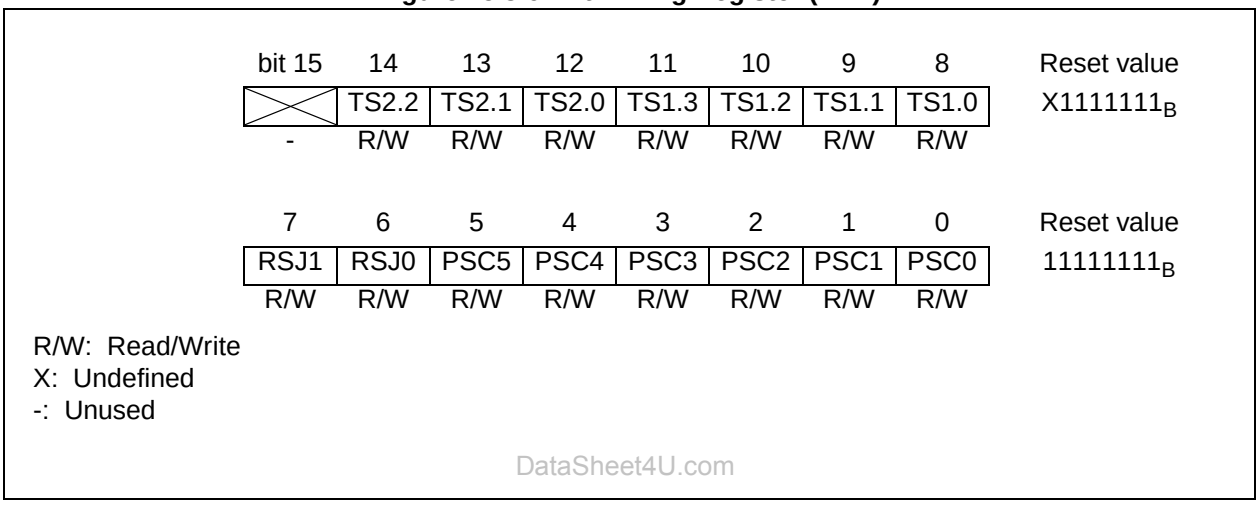
www.DataSheet4U.com

15.3.5 Bit Timing Register (BTR)

The bit timing register (BTR) sets the prescaler and bit timing after halting the bus (CSR: HALT = 1).

■ Bit Timing Register (BTR)

Figure 15.3-9 Bit Timing Register (BTR)



DataSheet4U.com

Table 15.3-7 Functions of Bit Timing Register (BTR)

Bit Name		Function
bit 0 to bit 5	PSC5 to PSC0: Prescaler setting bits 5 to 0	• These bits divide the frequency of the system clock to determine the time quantum (TQ) of the CAN controller.
bit 6 to bit 7	RSJ1, RSJ0: Resynchronous jump width setting bits 1, 0	• These bits set the resynchronous jump width (RSJW).
bit 8 to bit 11	TS1.3 to TS1.0: Time segment 1 setting bits 3 to 0	• These bits set the time of time segment 1 (TSEG1). Time segment 1 is equivalent to propagation segment (PROP_SEG) and phase buffer segment 1 (PHASE_SEG1) based on CAN specifications.
bit 12 to bit 14	TS2.2 to TS2.0: Time segment 2 setting bits 2 to 0	• These bits set the time of time segment 2 (TSEG2). Time segment 2 is equivalent to phase buffer segment 2 (PHASE_SEG2) based on CAN specifications.

Note: Set the bit timing register (BTR) after halting the bus (CSR: HALT = 1). After setting the bit timing register (BTR), write 0 to the HALT bit in the control status register to cancel the bus halt.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

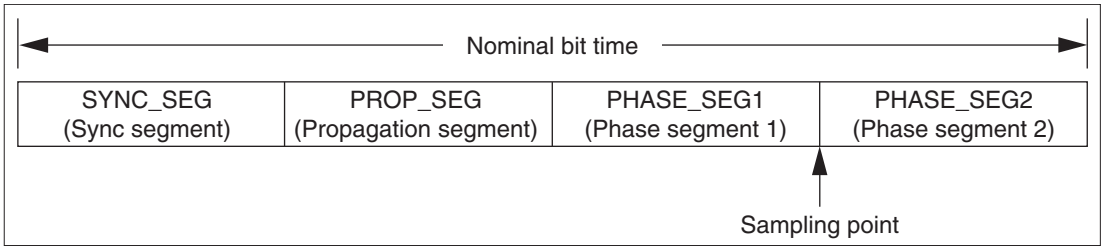
www.DataSheet4U.com

■ Definition of Bit Timing Segment

Bit timing is set in the bit timing register (BTR). Figure 15.3-10 and Figure 15.3-11 show the segments of the nominal bit time (one bit of time within message) and bit timing register (BTR).

● Bit time segments of general CAN specifications

Figure 15.3-10 Bit Time Segments of General CAN Specifications



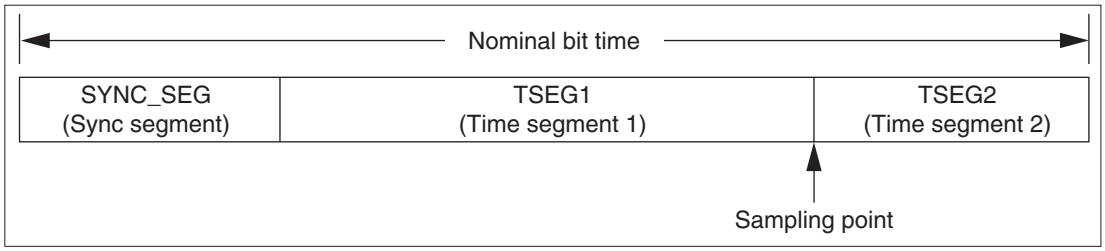
- SYNC_SEG (sync segment): Synchronization is performed to shorten or prolong the bit time.
- PROP_SEG (propagation segment): The physical delay among networks is adjusted.
- PHASE_SEG (phase segment): The phase shift due to oscillation errors is adjusted.

● Bit time segments of Fujitsu CAN controller

The propagation segment (PROP_SEG) and phase segment 1 (PHASE_SEG1) are used as the time segment 1 (TSEG1). The phase segment 2 (PHASE_SEG2) is used as the time segment 2 (TSEG2).

DataSheet4U.com

Figure 15.3-11 Bit Time Segments of CAN Controller



- TSEG1 = PROP_SEG + PHASE_SEG1
- TSEG2 = PHASE_SEG2

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Calculation of Bit Timing

Figure 15.3-12 and Figure 15.3-13 show the calculation example of bit timing, respectively, assuming input clock (CLK), time quantum (TQ), bit time (BT), synchronous segment (SYNC_SEG), time segments 1 and 2 (TSEG1, TSEG2), resynchronous jump width (RSJW), and frequency division (PSC).

Figure 15.3-12 Calculation of Bit Timing

· TQ = (PSC + 1) × CLK
· BT = SYNC_SEG + TSEG1 + TSEG2
= (1 + (TS1 + 1) + (TS2 + 1)) × TQ
= (3 + TS1 + TS2) × TQ
· RSJW = (RSJ + 1) × TQ

For each segment, the following conditions should be met.

· When PSC is 1 to 63 (2 to 64-divided clock)
TSEG1 ≥ 2TQ
TSEG1 ≥ RSJW
TSEG2 ≥ 2TQ
TSEG2 ≥ RSJW

· When PSC is 0 (1-divided clock)
TSEG1 ≥ 5TQ
TSEG2 ≥ 2TQ
TSEG2 ≥ RSJW

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 15.3-13 Calculation Example of Bit Timing

Examle: When 1TQ is 1/20 bit timing at 100 kbps (1/100 kbps/20)
Condition: (resynchronous jump width is 4TQ, delay time is 50 µs)

(1) Calculations of time quantum (TQ) [TQ = (PSC + 1) × CLK]

(unit: µs)

	Frequency division of input clock (PSC+1)															
Machine clock (CLK)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
4 MHz (0.25 µs)	0.25	0.5	0.75	1	1.25	1.5	1.75	2	2.25	2.5	2.75	3	3.25	3.5	3.75	
8 MHz (0.125 µs)	0.13	0.25	0.38	0.5	0.63	0.75	0.88	1	1.13	1.25	1.38	1.5	1.63	1.75	1.88	
16 MHz (0.0625 µs)	0.06	0.13	0.19	0.25	0.31	0.38	0.44	0.5	0.56	0.63	0.69	0.75	0.81	0.88	0.94	

(4) Conditions of bit timing (BT) [BT ≥ 8TQ]

(unit: µs)

Machine clock (CLK)	8TQ															
4 MHz (0.25 µs)	2	4	6	8	10	12	14	16	18	20	22	24	26	28	30	
8 MHz (0.125 µs)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	
16 MHz (0.0625 µs)	0.5	1	1.5	2	2.5	3	3.5	4	4.5	5	5.5	6	6.5	7	7.5	

(3) Setting of resynchronous jump width (when resynchronous jump width is 4TQ)

RSJ+1 (frequency division of TQ)	1	2	3	4	(unit: TQ)
RSJW = (RSJ + 1) × TQ	0.5	1	1.5	2	(unit: µs)

(5) Conditions of TSEG2

RSJW = (RSJ + 1) × TQ	1	2	3	4	(unit: TQ)
TSEG2 ≥ RSJW	1	2	3	4	(unit: TQ)
TSEG2 ≥ RSJW	0.5	1	1.5	2	(unit: µs)

(6) Conditions of TSEG1

TSEG1 ≥ Delay time + RSJW (Assuming that delay time is 50ns) × 2 + 4TQ ≥ 5TQ

TSEG1

5 (unit: TQ)

(2) Calculation of bit time (BT) based on the above setting and conditions

BT = SYNC_SEG + TSEG1 + TSEG2
= (1 + (TS1 + 1) + (TS2+1)) × TQ
= (3 + TS1 + TS2) × TQ

(unit: kbps)

TSEG1 + 1 \ TSEG2 + 1	1	2	3	4	5	6	7	8
1	667	500	400	333	286	250	222	200
2	500	400	333	286	250	222	200	182
3	400	333	286	250	222	200	182	167
4	333	286	250	222	200	182	167	154
5	286	250	222	200	182	167	154	143
6	250	222	200	182	167	154	143	133
7	222	200	182	167	154	143	133	125
8	200	182	167	154	143	133	125	118
9	182	167	154	143	133	125	118	111
10	167	154	143	133	125	118	111	105
11	154	143	133	125	118	111	105	100
12	143	133	125	118	111	105	100	95.2
13	133	125	118	111	105	100	95.2	90.9
14	125	118	111	105	100	95.2	90.9	87
15	118	111	105	100	95.2	90.9	87	83.3
16	111	105	100	95.2	90.9	87	83.3	80
				(1)	(2)	(3)	(4)	(5)

Calculation of sampling point

	SYNC_SEG + (TSEG1 + 1)	TSEG2 + 1	Sampling point
(1)	16	4	80%
(2)	15	5	75%
(3)	14	6	70%
(4)	13	7	65%
(5)	12	8	60%

SYNC_SEG	TSEG 1 + 1	TSEG2 + 1

↑ Sampling point

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

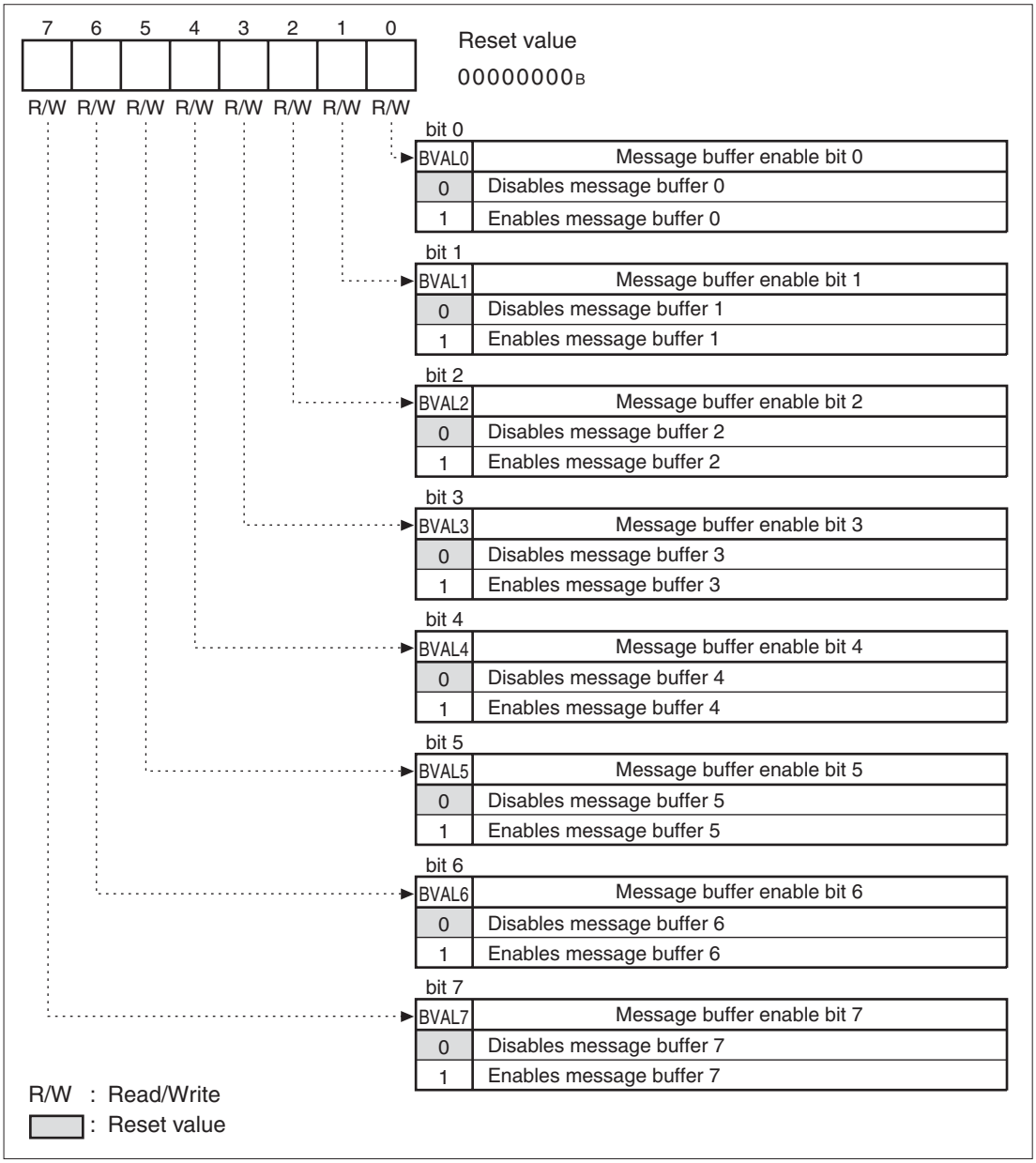
www.DataSheet4U.com

15.3.6 Message Buffer Valid Register (BVALR)

The message buffer valid register (BVALR) enables or disables the message buffers and indicates their status.

■ Message Buffer Valid Register (BVALR)

Figure 15.3-14 Message Buffer Valid Register (BVALR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-8 Functions of Message Buffer Enable Register

Bit Name		Function
bit 0 to bit 7	BVAL7 to BVAL0: Message buffer enable bits 7 to 0	These bits enable or disable transmitting and receiving of the message to and from the message buffer (x). **When set to 0:** No message can be transmitted and received to and from the message buffer (x). **When set to 1:** A message can be transmitted and received to and from the message buffer (x). **[Message buffer disabled (BVALx = 0)]** **During transmitting:** Transmitting and receiving are disabled after message transmitting is completed or a transmit error is terminated. **During receiving:** Transmitting and receiving are disabled immediately. When the received message is stored in the message buffer, transmitting and receiving are disabled after the message is stored. **Note:** The read modify write instructions are disabled until the BVALx bit is actually set to 0 after 0 is written to the bit.

Note: To invalidate the message buffer (by setting the BVALR: BVAL bit to 0) while CAN Controller is participating in CAN communication (the read value of the CSR: HALT bit is 0 and CAN Controller is ready to receive or transmit messages), follow the cautions in Section "15.6 Precautions when Using CAN Controller".

DataSheet4U.com
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

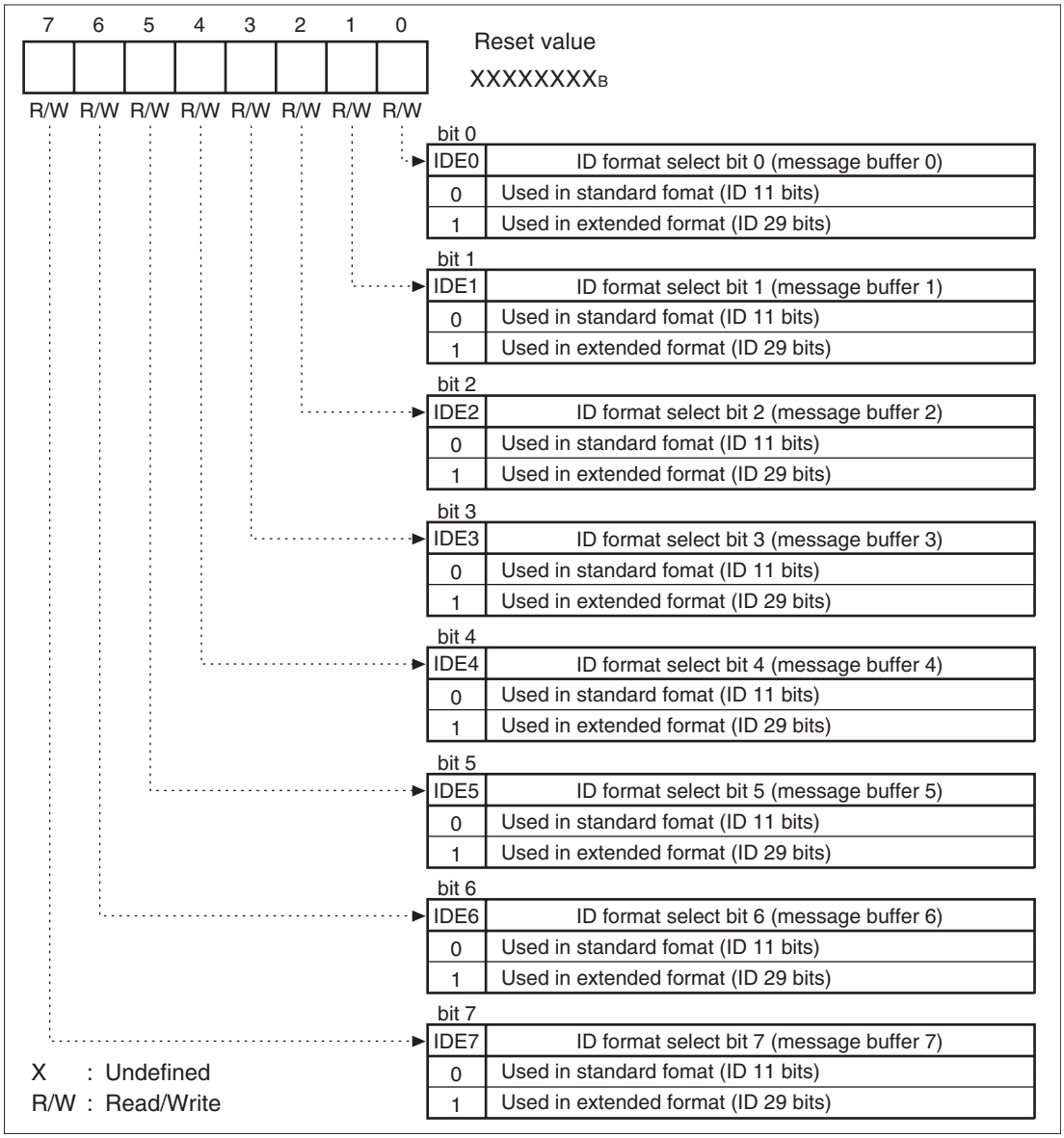
www.DataSheet4U.com

15.3.7 IDE Register (IDER)

The IDE register (IDER) sets the frame format of the message buffer used during transmitting and receiving. Transmitting and receiving are enabled in the standard frame format (ID11 bits) and the extended frame format (ID29 bits).

■ IDE Register (IDER)

Figure 15.3-15 IDE Register (IDER)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-9 Functions of IDE Register (IDER)

Bit Name		Function
bit 0 to bit 7	IDE7 to IDE0: ID Format select bits 7 to 0	These bits set the ID format of the message buffer (x). **When set to 0:** Uses message buffer (x) in standard format (ID11 bits) **When set to 1:** Uses message buffer (x) in extended format (ID29 bits) **Note:** The IDE register (IDER) should be set after having the message buffer (x) disabled (BVALR: BVALx = 0). Setting the IDE register (IDER) with the message buffer (x) being enabled may store message unnecessary received.

Note: To invalidate the message buffer (by setting the BVALR: BVAL bit to 0) while CAN Controller is participating in CAN communication (the read value of the CSR: HALT bit is 0 and CAN Controller is ready to receive or transmit messages), follow the cautions in Section "15.6 Precautions when Using CAN Controller".

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

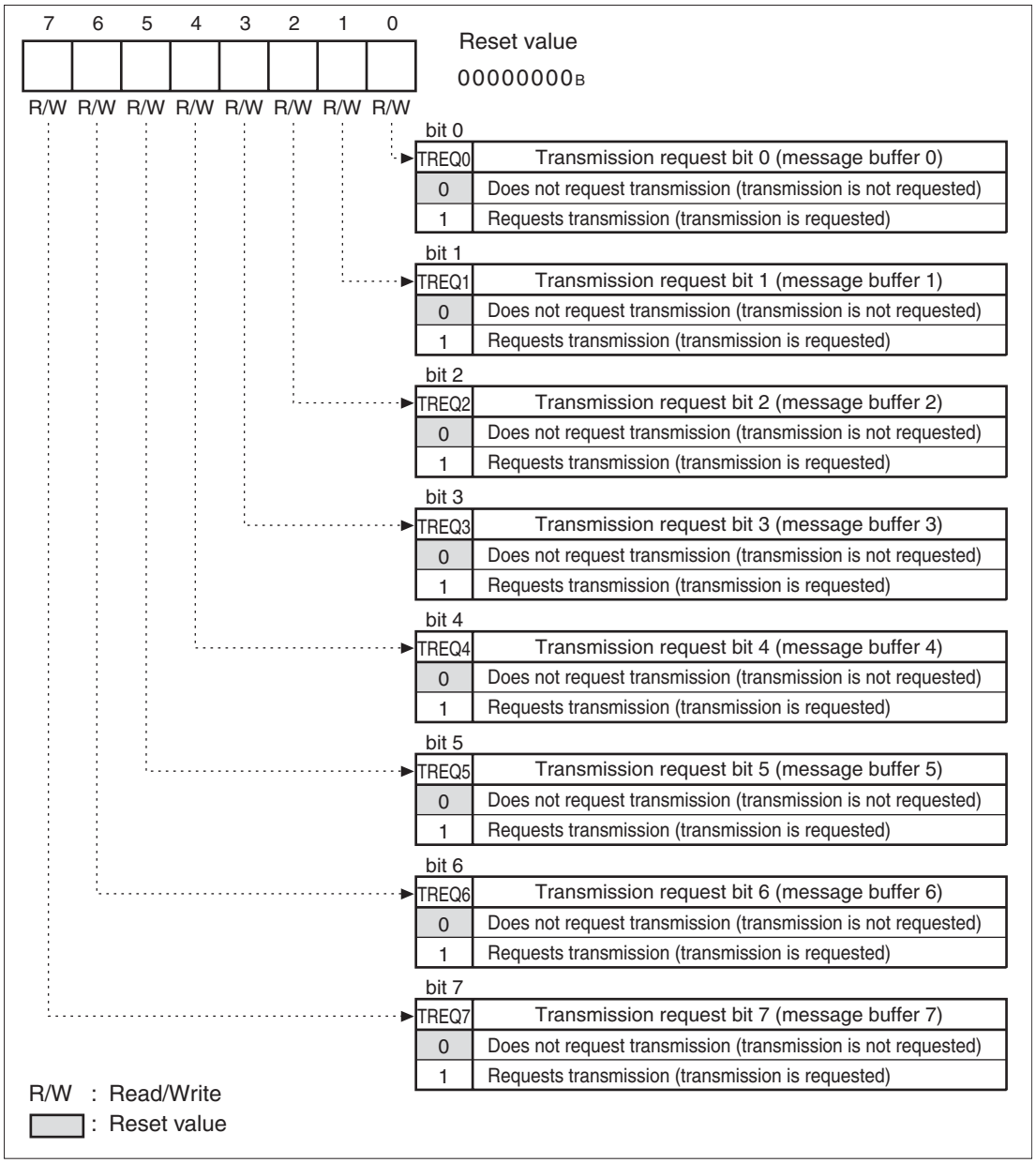
www.DataSheet4U.com

15.3.8 Transmission Request Register (TREQR)

The transmission request register (TREQR) sets a transmit request for each message buffer and indicates its status.

■ Transmission Request Register (TREQR)

Figure 15.3-16 Transmission Request Register (TREQR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-10 Functions of Transmission Request Register (TREQR)

Bit Name		Function
bit 0 to bit7	TREQ7 to TREQ0: Transmission request bits 7 to 0	These bits starts transmitting for the message buffer (x). **When set to 0:** No effect **When set to 1:** Starts transmitting for message buffer (x) • If more than one transmit complete bit is set (TREQx = 1), transmitting is started with the lower number of the message buffer (x) that accepts the transmit request. • These bits remain 1s during the transmit being requested and are cleared to 0 when transmitting is completed or the transfer request is cancelled. • Clearing a transmit request when transmitting is completed (TREQx = 0) overrides setting of the transmit request bit when 0 is written (TREQx = 1) if both occur at the same time. **Read by read modify write instruction:** 1 always read **[Setting of remote frame receive wait bit (RFWTR: RFWTx)]** **RFWTx bit = 0:** Starts transmitting immediately even if RRTRx bit in receive RTR register = 1 **RFWTx bit = 1:** Starts transmitting after remote frame received.

References:
- See "15.3.10 Remote Frame Receiving Wait Register (RFWTR)" for details of the remote frame receive wait register (RFWTR).
- See "15.3.15 Reception RTR Register (RRTRR)" for details of the receive RTR register (RRTRR).
- See "15.3.11 Transmission Cancel Register (TCANR)" and "15.5.1 Transmission" for details about the transmit cancellation.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

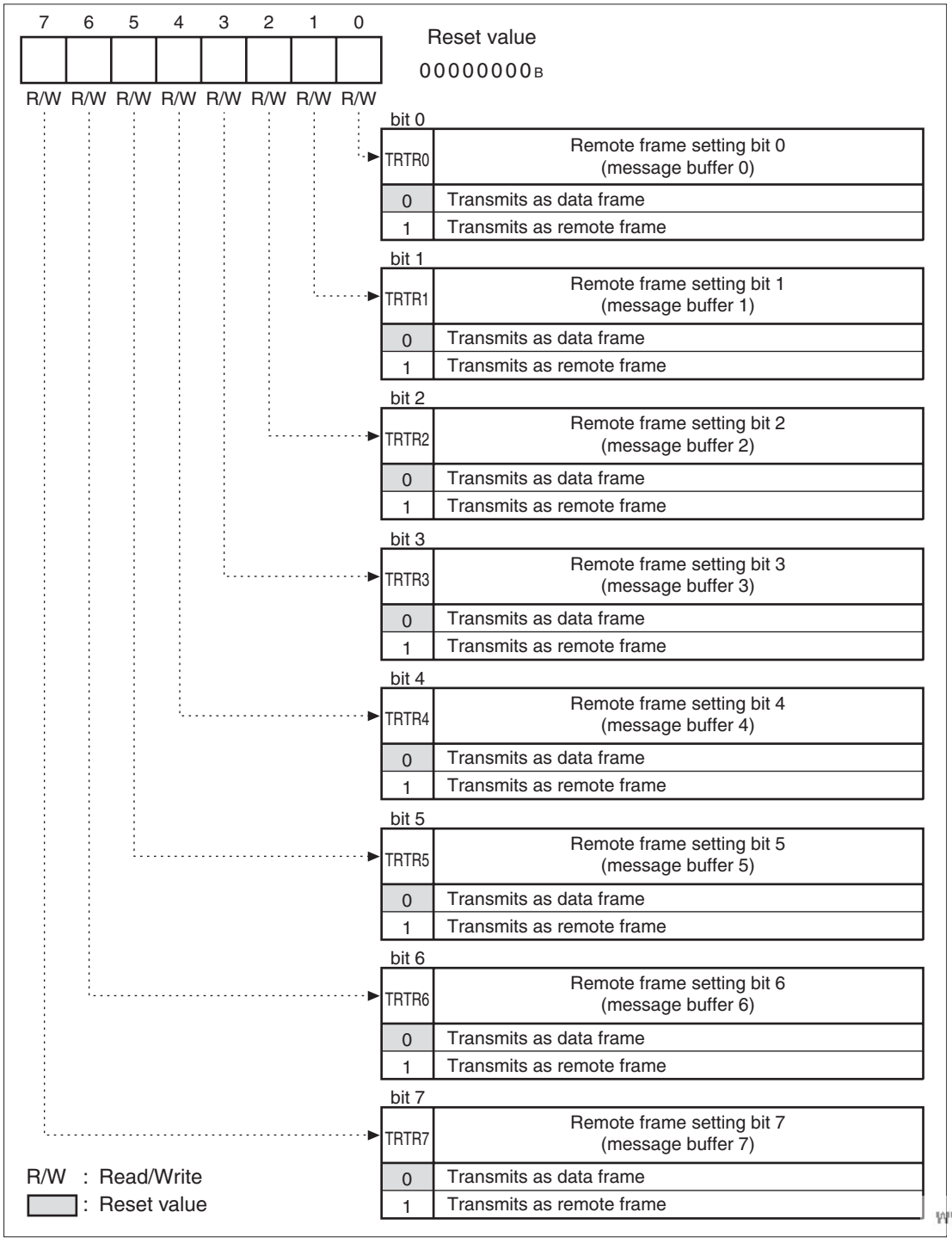
www.DataSheet4U.com

15.3.9 Transmission RTR Register (TRTRR)

This register sets the frame format of transmit message for the message buffers.

■ Transmission RTR Register (TRTRR)

Figure 15.3-17 Transmission RTR Register (TRTRR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- When 0 is written to each bit in the transmit RTR register (TRTRR), the data frame format is set. When 1 is written to each bit, the remote frame format is set.

Table 15.3-11 Functions of Transmission RTR Register (TRTRR)

Bit Name		Function
bit 0 to bit 7	TRTR7 to TRTR0: Remote frame setting bits 7 to 0	These bits set the transfer format of the message buffer (x) for transmitting or receiving. **When set to 0:** Sets data frame format **When set to 1:** Sets remote frame format

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

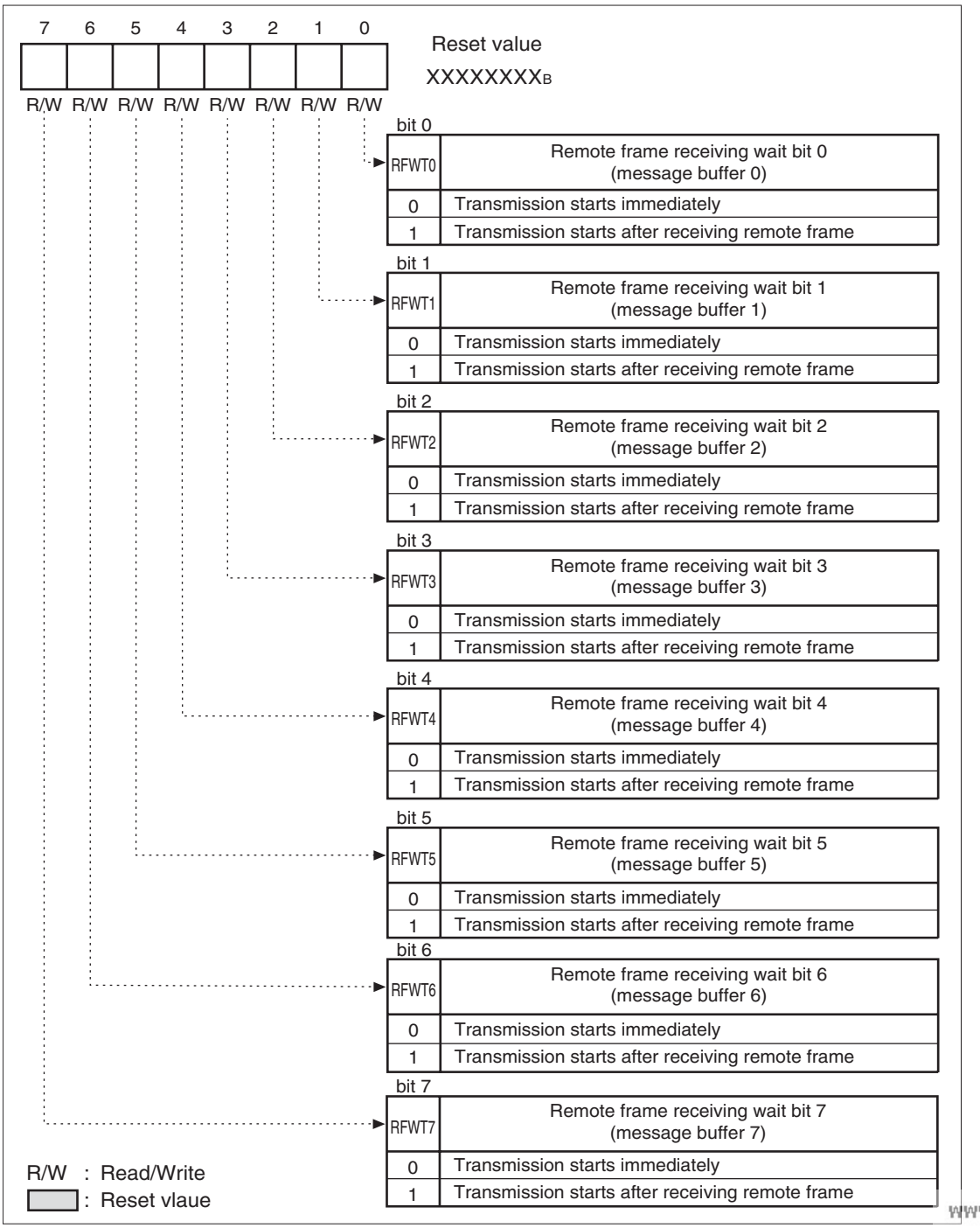
www.DataSheet4U.com

15.3.10 Remote Frame Receiving Wait Register (RFWTR)

Remote frame receiving wait register (RFWTR) sets whether this register waits remote frame receiving when transmission request of data frame is set.

■ Remote Frame Receiving Wait Register (RFWTR)

Figure 15.3-18 Remote Frame Receive Wait Register (RFWTR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-12 Functions of Remote Frame Receiving Wait Register (RFWTR)

Bit Name		Function
bit 0 to bit 7	RFWT7 to RFWT0: Remote frame receiving wait bits 7 to 0	These bits set whether to wait for reception of a remote frame for the message buffer (x) for which a request to transmit a data frame is set. **When set to 0:** Starts transmitting immediately for message buffer (x) for which a request to transmit data frame is set • Transmitting is started immediately even if the receive RTR register is already set in the message buffer (x) (RRTRR: RRTRx = 1). **When set to 1:** Starts transmitting after remote frame is received in message buffer (x) in which a request to transmit a data frame is set. **Note:** When transmitting a remote frame, do not write 1 to the RFWTx bit.

References:
- See "15.3.8 Transmission Request Register (TREQR)" for details of the transmission request register (TREQR).
- See "15.3.9 Transmission RTR Register (TRTRR)" for details of the transmission RTR register (TRTRR).
- See "15.3.15 Reception RTR Register (RRTRR)" for details of the receive RTR register (RRTRR).

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

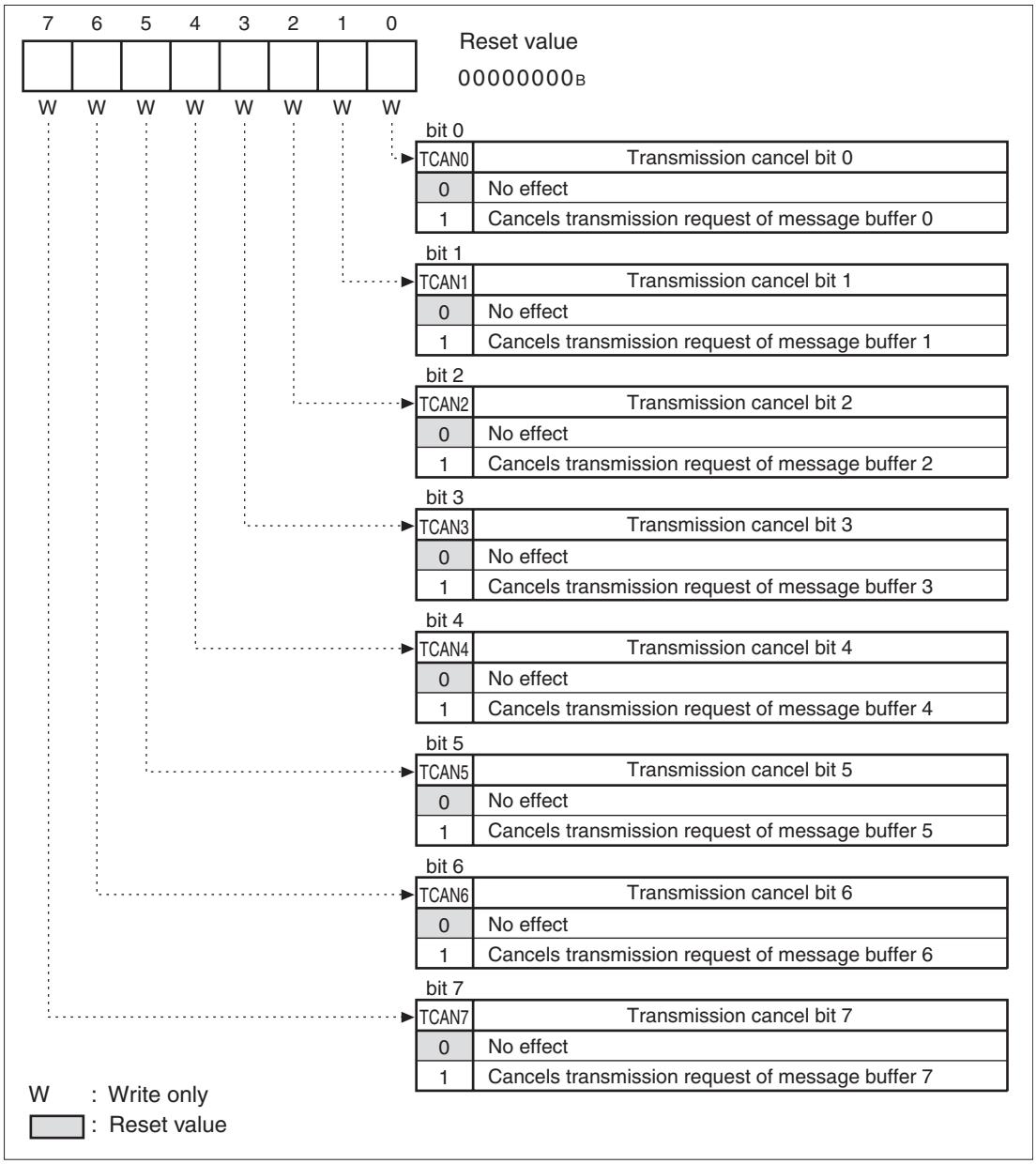
www.DataSheet4U.com

15.3.11 Transmission Cancel Register (TCANR)

The transmission cancel register (TCANR) sets cancellation of a transmission request for the message buffer in the transmit wait state.

■ Transmission Cancel Register (TCANR)

Figure 15.3-19 Transmission Cancel Register (TCANR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-13 Functions of Transmission Cancel Register (TCANR)

Bit Name		Function
bit 0 to bit 7	TCAN7 to TCAN0: Transmission cancel bits 7 to 0	These bits cancel a transmission request for the message buffer (x) in the transmit wait state. **When set to 0:** No effect **When set to 1:** Cancels transmission request for message buffer (x) • When a transmission request is cancelled by setting 1 to the TCANx bit, the TREQx bit corresponding to the message buffer (x) is cleared (TREQx = 0) for which transmission request is cancelled. **Read:** 0 always read **Note:** The transmission cancel register (TCANR) is a write-only register.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

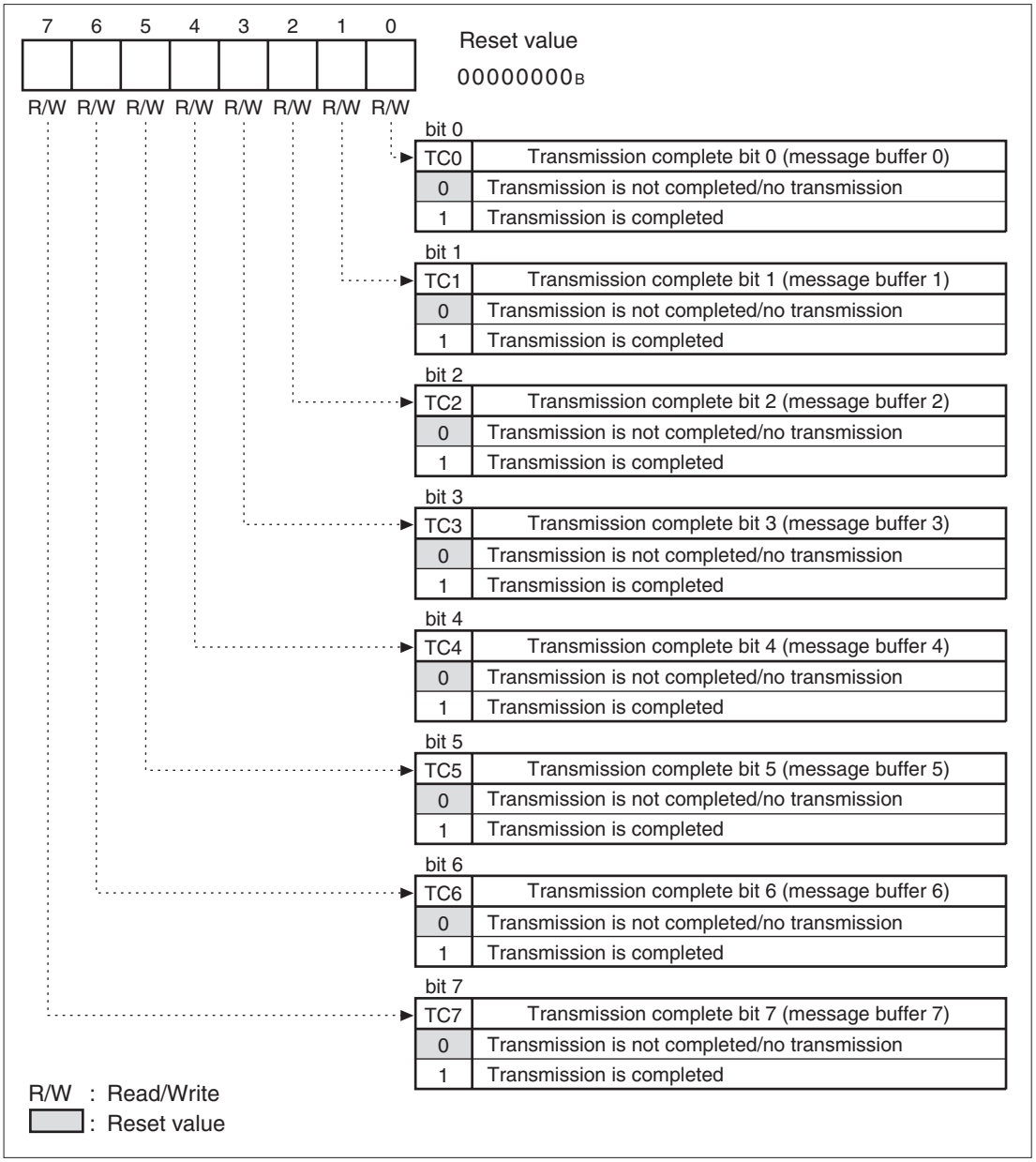
www.DataSheet4U.com

15.3.12 Transmission Complete Register (TCR)

The transmission complete register (TCR) indicates whether transmitting a data from the message buffer completes. When an output of interrupt request is enabled at completing transmitting, an interrupt request is output when transmitting is completed.

■ Transmission Complete Register (TCR)

Figure 15.3-20 Transmission Complete Register (TCR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-14 Functions of Transmission Complete Register (TCR)

Bit Name		Function
bit 0 to bit 7	TC7 to TC0: Transmission complete bits 7 to 0	These bits indicate whether the message buffer (x) completes transmitting message. **When message transmitting completed:** 1 is set to the TCx bit corresponding to the message buffer (x) that completes transmitting. **When set to 0:** Clears bits if transmitting already completed **When set to 1:** No effect **Read by read modify write instruction:** 1 always read • Setting the TCx bit when transmitting is completed (TCx = 1) overrides clearing of the TCx bit when 0 is written (TCx = 0) if both occur at the same time. • When the TREQx bit in the transmit request register (TREQR) is set (TREQR: TREQx = 1), the TCx bit is cleared (TCx = 0). **[Generation of transmission complete interrupt]** • If the transmit complete interrupt enable register (TIER) is set (TIER: TIEx = 1), a transmit complete interrupt is generated when transmitting is completed (TCR: TCx = 1).

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

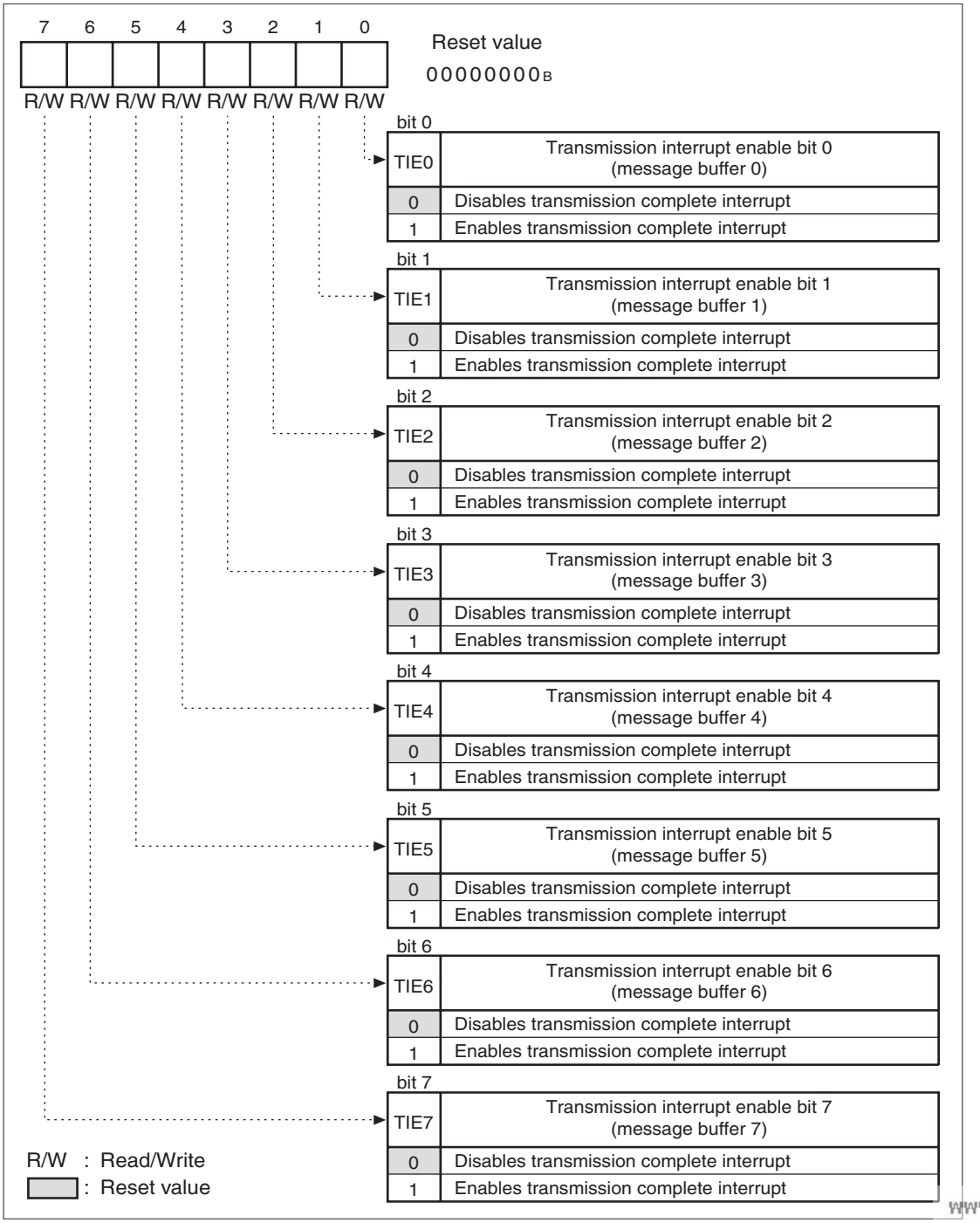
www.DataSheet4U.com

15.3.13 Transmission Complete Interrupt Enable Register (TIER)

The transmission complete interrupt enable register (TIER) enables or disables a transmit complete interrupt for each message buffer.

■ Transmission Complete Interrupt Enable Register (TIER)

Figure 15.3-21 Transmission complete Interrupt Enable Register (TIER)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-15 Functions of Transmission Complete Interrupt Enable Register (TIER)

Bit Name		Function
bit 0 to bit 7	TIE7 to TIE0: Transmission complete interrupt enable bits 7 to 0	These bits enable or disable a transmission complete interrupt for the message buffer (x). **When set to 0:** Disables transmit complete interrupt for message buffer (x) **When set to 1:** Enables transmit complete interrupt for message buffer (x)

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

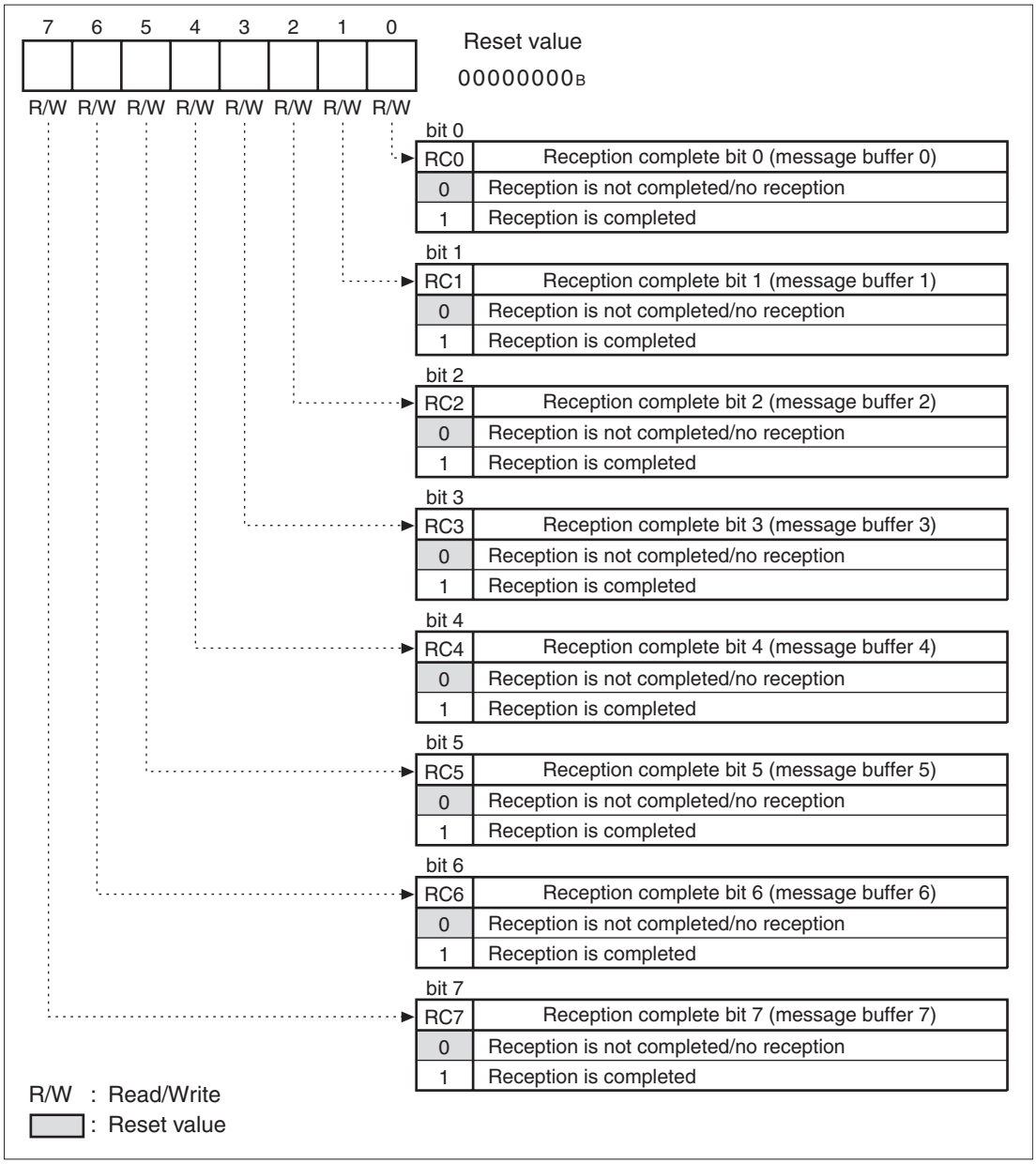
www.DataSheet4U.com

15.3.14 Reception Complete Register (RCR)

The reception complete register (RCR) indicates whether the reception a data to the message buffer (x) completes receiving. When an interrupt is enabled at completion of receiving, an interrupt request is generated.

■ Reception Complete Register (RCR)

Figure 15.3-22 Reception Complete Register (RCR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-16 Functions of Reception Complete Register (RCR)

Bit Name		Function
bit 0 to bit 7	RC7 to RC0: Reception complete bits 7 to 0	These bits indicate whether the message buffer (x) completes message receiving. **When message receiving completed:** 1 is set to the RCx bit corresponding to the message buffer (x) that completes receiving. **When set to 0:** Clears bits when receiving already completed **When set to 1:** No effect **Read by read modify write instruction:** 1 always read • Setting the RCx bit when receiving is completed (RCx = 1) overrides clearing of the RCx bit when 0 is written (RCx = 0) if both occur at the same time. **[Generation of reception complete interrupt]** • If the reception complete enable register is set (RIER: RIEx = 1), a reception complete interrupt is generated when receiving is completed. **Note:** To clear the reception complete register (RCR), read the received message after the completion of receiving and write 0.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

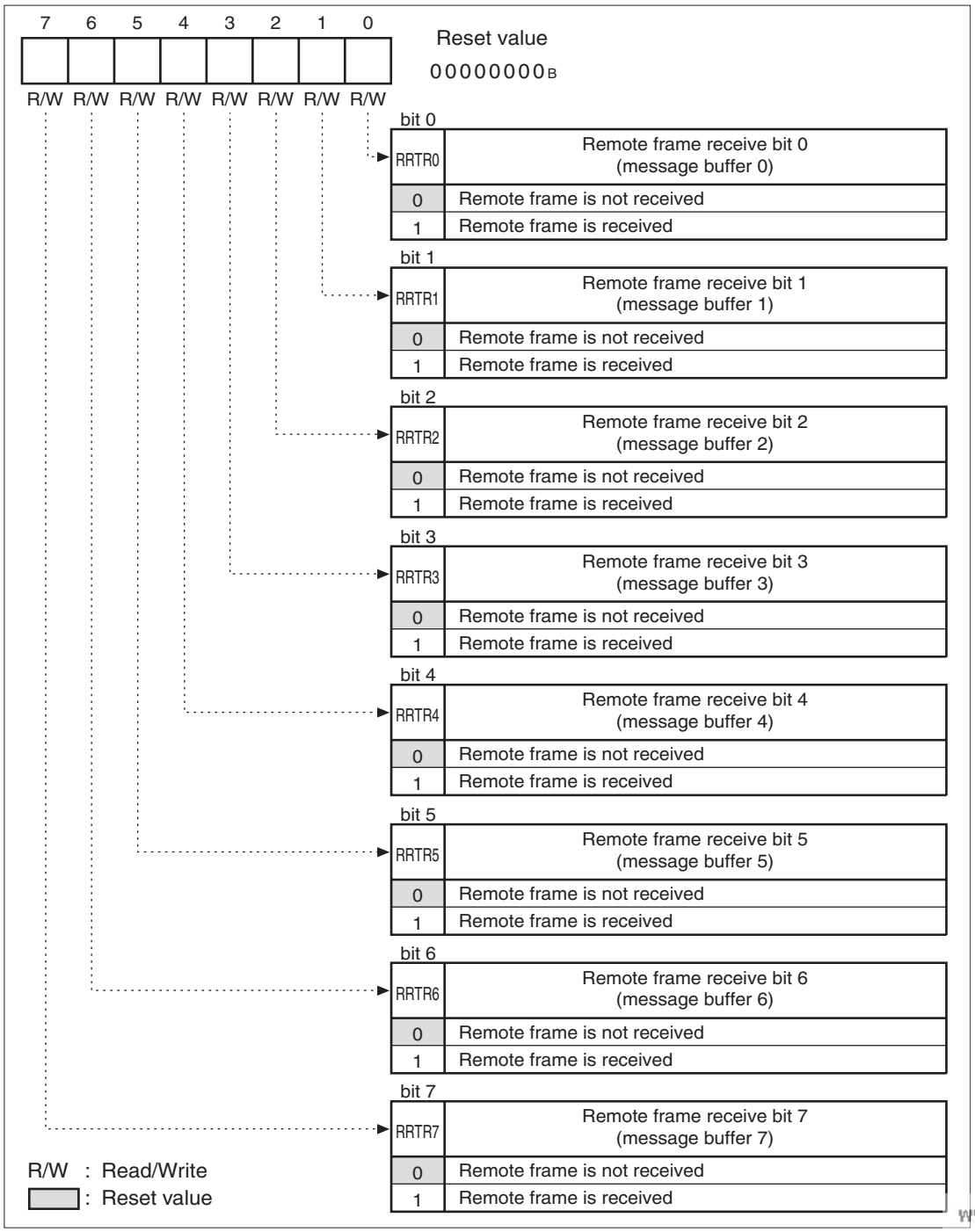
www.DataSheet4U.com

15.3.15 Reception RTR Register (RRTRR)

The reception RTR register (RRTRR) indicates that the remote frame is stored in the message buffer.

■ Reception RTR Register (RRTRR)

Figure 15.3-23 Reception RTR Register (RRTRR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-17 Functions of Reception RTR Register (RRTRR)

Bit Name		Function
bit 0 to bit 7	RRTR7 to RRTR0: Remote frame receive bits 7 to 0	These bits indicate that the message buffer (x) receives a remote frame. **When remote frame is received:** 1 is set to the RRTRx bit corresponding to the message buffer (x) that receives a remote frame. **When set to 0:** Cleared when receiving completed **When set to 1:** No effect • Setting the RRTRx bit when a remote frame is received (RRTRx = 1) overrides clearing of the RRTRx bit when 0 is written (RRTRx = 0) if both occur at the same time. • The RRTRx bit corresponding to the message buffer (x) that receives a data frame is cleared (RRTRx = 0). • If message transmitting is completed (TCR: TCx = 1), the RRTRx bit corresponding to the message buffer (x) that transmits the message is cleared (RRTRx = 0). **Read by read modify write instruction:** 1 always read

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

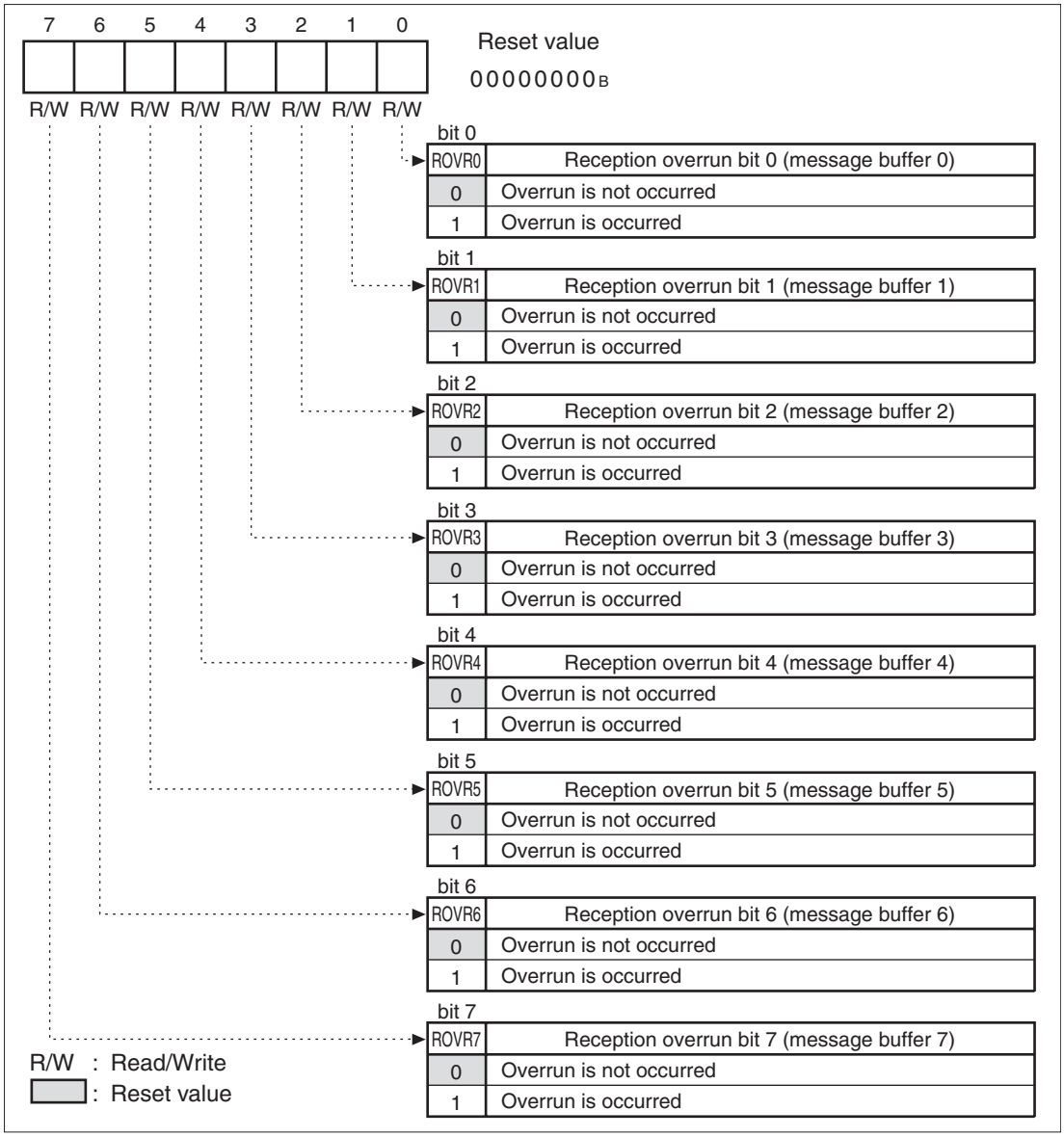
www.DataSheet4U.com

15.3.16 Reception Overrun Register (ROVRR)

The reception overrun register (ROVRR) indicates that an overrun occurs (the corresponding message buffer is in the receive complete state.) at storing the received message in the message buffer.

■ Reception Overrun Register (ROVRR)

Figure 15.3-24 Reception Overrun Register (ROVRR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-18 Functions of Reception Overrun Register (ROVRR)

Bit Name		Function
bit 0 to bit 7	ROVR7 to ROVR0: Reception overrun bits 7 to 0	These bits indicate that an overrun occurs at storing the received message in the message buffer that had completed receiving. **At overrun:** 1 is set to the ROVRx bit corresponding to the message buffer (x) where an overrun occurs. **When set to 0:** Cleared when 0 is set to after reception overrun occurred **When set to 1:** No effect **Read by read modify write instruction:** 1 always read • Setting the ROVRx bit when an overrun occurs (ROVRx = 1) overrides clearing of the ROVRx bit when 0 is written (ROVRx = 0) if both occur at the same time.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

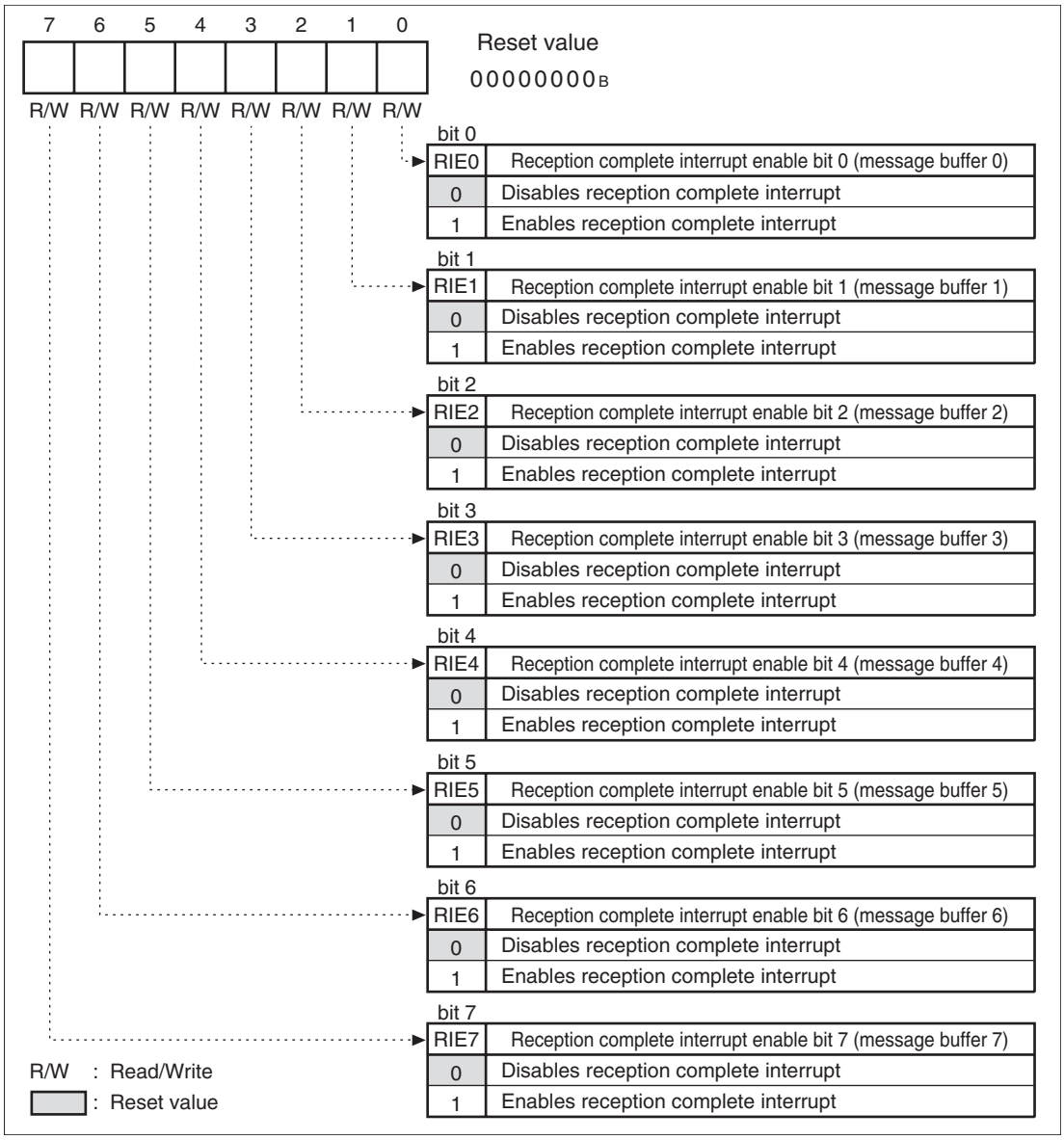
www.DataSheet4U.com

15.3.17 Reception Complete Interrupt Enable Register (RIER)

The reception complete interrupt enable register (RIER) enables or disables a reception complete interrupt for each message buffer.

■ Reception Complete Interrupt Enable Register (RIER)

Figure 15.3-25 Reception Complete Interrupt Enable Register (RIER)



bit		
bit 0	RIE0	Reception complete interrupt enable bit 0 (message buffer 0)
	0	Disables reception complete interrupt
	1	Enables reception complete interrupt
bit 1	RIE1	Reception complete interrupt enable bit 1 (message buffer 1)
	0	Disables reception complete interrupt
	1	Enables reception complete interrupt
bit 2	RIE2	Reception complete interrupt enable bit 2 (message buffer 2)
	0	Disables reception complete interrupt
	1	Enables reception complete interrupt
bit 3	RIE3	Reception complete interrupt enable bit 3 (message buffer 3)
	0	Disables reception complete interrupt
	1	Enables reception complete interrupt
bit 4	RIE4	Reception complete interrupt enable bit 4 (message buffer 4)
	0	Disables reception complete interrupt
	1	Enables reception complete interrupt
bit 5	RIE5	Reception complete interrupt enable bit 5 (message buffer 5)
	0	Disables reception complete interrupt
	1	Enables reception complete interrupt
bit 6	RIE6	Reception complete interrupt enable bit 6 (message buffer 6)
	0	Disables reception complete interrupt
	1	Enables reception complete interrupt
bit 7	RIE7	Reception complete interrupt enable bit 7 (message buffer 7)
	0	Disables reception complete interrupt
	1	Enables reception complete interrupt

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-19 Functions of Reception Complete Interrupt Enable Register (RIER)

Bit Name		Function
bit 0 to bit 7	RIE7 to RIE0: Reception complete interrupt enable bits 7 to 0	These bits enable or disable a reception complete interrupt for the message buffer (x). **When set to 0:** Disables reception complete interrupt for message buffer (x) **When set to 1:** Enables reception complete interrupt for message buffer (x)

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.3.18 Acceptance Mask Select Register (AMSR)

The acceptance mask select register (AMSR) selects the mask (acceptance mask) format for comparison between the identifier (ID) of the received message and the message buffer.

■ Acceptance Mask Select Register (AMSR)

Figure 15.3-26 Acceptance Mask Select Register (AMSR)

bit 15	14	13	12	11	10	9	8	Reset value
AMS7.1	AMS7.0	AMS6.1	AMS6.0	AMS5.1	AMS5.0	AMS4.1	AMS4.0	$XXXXXXXX_B$
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

7	6	5	4	3	2	1	0	Reset value
AMS3.1	AMS3.0	AMS2.1	AMS2.0	AMS1.1	AMS1.0	AMS0.1	AMS0.0	$XXXXXXXX_B$
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	

AMSx.1	AMSx.0	Acceptance mask select bits (x = 7 to 0)
0	0	Full-bit comparison
0	1	Full-bit mask
1	0	Uses acceptance mask register 0 (AMR0)
1	1	Uses acceptance mask register 1 (AMR1)

x (7 to 0) is message buffer's number (x).

X: Undefined
R/W: Read/Write

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-20 Functions of Acceptance Mask Select Register (AMSR)

Bit Name		Function
bit 0 to bit 15	AMS7.0 to AMS0.0, AMS7.1 to AMS0.1: Acceptance mask select bits 7.0 to 0.0, 7.1 to 0.0	These bits select the mask (acceptance mask) format for comparison between the received message ID and message buffer ID (IDR) for the message buffer (x). No comparison with masked bits is made. **Full-bit comparison:** All bits are compared in collating the setting values of the ID register (IDR) with the received message ID. **Full-bit masking:** All bits for the setting values of the ID register (IDR) and the received message ID are masked. **Using acceptance mask register 0 (or 1):** The acceptance mask register 0 or 1 (AMR0 or AMR1) is used as an acceptance mask filter. At collating the setting values of the ID register (IDR) with the received message ID, only the bits set to 0 and corresponding to the AMx bit in the acceptance mask register are compared and the bits set to 1 and corresponding to the AMx bit are masked. • If the AMSx.1 and AMSx.0 bits are set to 10_B or 11_B, always set the acceptance mask register (AMR0 or AMR1) to be used, too. **Note:** The acceptance mask select register (AMSR) should be set after disabling the message buffer (x) to be set (BVALR: BVALx = 0). Setting the acceptance mask select register (AMSR) with the message buffer (x) enabled may store a message unnecessary received.

Note: To invalidate the message buffer (by setting the BVALR: BVAL bit to 0) while CAN Controller is participating in CAN communication (the read value of the CSR: HALT bit is 0 and CAN Controller is ready to receive or transmit messages), follow the cautions in Section "15.6 Precautions when Using CAN Controller".

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.3.19 Acceptance Mask Register (AMR)

The CAN controller has two acceptance mask registers (AMR0 and AMR1). Both of them can be used in the standard frame format (ID11 bits, AM28 to AM18) and the extended frame format (ID29 bits, AM28 to AM0).

■ Acceptance Mask Register (AMR)

Figure 15.3-27 Acceptance Mask Register (AMR)

	bit 7	6	5	4	3	2	1	0	Reset value
BYTE0	AM28	AM27	AM26	AM25	AM24	AM23	AM22	AM21	$XXXXXXXX_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	bit 15	14	13	12	11	10	9	8	Reset value
BYTE1	AM20	AM19	AM18	AM17	AM16	AM15	AM14	AM13	$XXXXXXXX_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	bit 7	6	5	4	3	2	1	0	Reset value
BYTE2	AM12	AM11	AM10	AM9	AM8	AM7	AM6	AM5	$XXXXXXXX_B$
	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
	15	14	13	12	11	10	9	8	Reset value
BYTE3	AM4	AM3	AM2	AM1	AM0				$XXXXXXXX_B$
	R/W	R/W	R/W	R/W	R/W	–	–	–	

R/W: Read/Write
X : Undefined
- : Unused
▒: Used bits in the standard frame format

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-21 Functions of Acceptance Mask Register (AMSR)

Bit Name		Function
bit 0 to bit 7	AM21 to AM 28: Acceptance mask bits 28 to 21 (BYTE0)	These bits set whether to compare or mask each bit at collating the acceptance code set in the ID register (IDR: IDx) with the received message ID. • If the AMSx.1 or AMSx.0 bits of acceptance mask select registers are set to 10_B or 11_B, always set the acceptance mask register (AMR0 or AMR1) to be used, too. **Standard frame format (IDER: IDEx = 0):** 11 bits from AM28 to AM18 are used. **Extended frame format (IDER: IDEx = 1):** 29 bits from AM28 to AM0 are used. **When AMx bit set to 0 (compare):** The bits corresponding to the AMx bit set to 0 are compared at collating the acceptance code set in the ID register (IDR: IDx) with the received message ID. **When AMx bit set to 1 (mask):** The bits corresponding to the AMx bit set to 1 are masked at collating the acceptance code set in the ID register (IDR: IDx) with the received message ID. **Note:** The acceptance mask select register (AMR0, AMR1) should be set after disabling the message buffer (x) to be set (BVALR: BVALx = 0). Setting the acceptance mask select register (AMR) with the message buffer (x) enabled may store a message unnecessary received.
bit 8 to bit 15	AM20 to AM 13: Acceptance mask bits 20 to 13 (BYTE1)	
bit 0 to bit 7	AM12 to AM 5: Acceptance mask bits 12 to 5 (BYTE2)	
bit 11 to bit 15	AM4 to AM 0: Acceptance mask bits 4 to 0 (BYTE3)	

Note:	To invalidate the message buffer (by setting the BVALR: BVAL bit to 0) while CAN Controller is participating in CAN communication (the read value of the CSR: HALT bit is 0 and CAN Controller is ready to receive or transmit messages), follow the cautions in Section "15.6 Precautions when Using CAN Controller".

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.3.20 Message Buffers

The message buffers consist of ID register, DLC register, and data register are used for transmission/reception of the message.

■ Message Buffers

- There are 8 message buffers.
- One message buffer (x) (x = 0 to 7) consists of an ID register (IDRx), DLC register (DLCRx), and data register (DTRx).
- The message buffer (x) is used to transmit and receive messages.
- Higher priority is given to smaller number message buffer.
  - At transmitting, if a transmit request is generated to more than one message buffer, transmitting is started from the message buffer with the smallest number.
  - At receiving, if the received message ID passes the acceptance filter (which compares received message ID with message buffer ID after acceptance masking) set in more than one message buffer, a received message is stored in the message buffer with the smallest number.
- If the same acceptance filter is set in more than one message buffer, it can be used as multiple message buffers. This provides sufficient time to perform receiving.

et4U.com DataShee

Notes:
- Write by words to the message buffer area and general-purpose RAM area. At writing by bytes, undefined data is written to the upper bytes and writing to the upper bytes is ignored when writing to the lower bytes is performed.
- The message buffer (x) area disabled by the message buffer enable register (BVALR: BVALx = 0) can be used as a general-purpose RAM area. However, during transmitting or receiving, it may take up to 64 machine cycles to access the message buffer area and general-purpose RAM area.

References:
- See "15.5.1 Transmission".
- See "15.5.2 Reception".
- See "15.5.4 Setting Multiple Message Receiving" for details of the configuration of the multiple message buffer.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

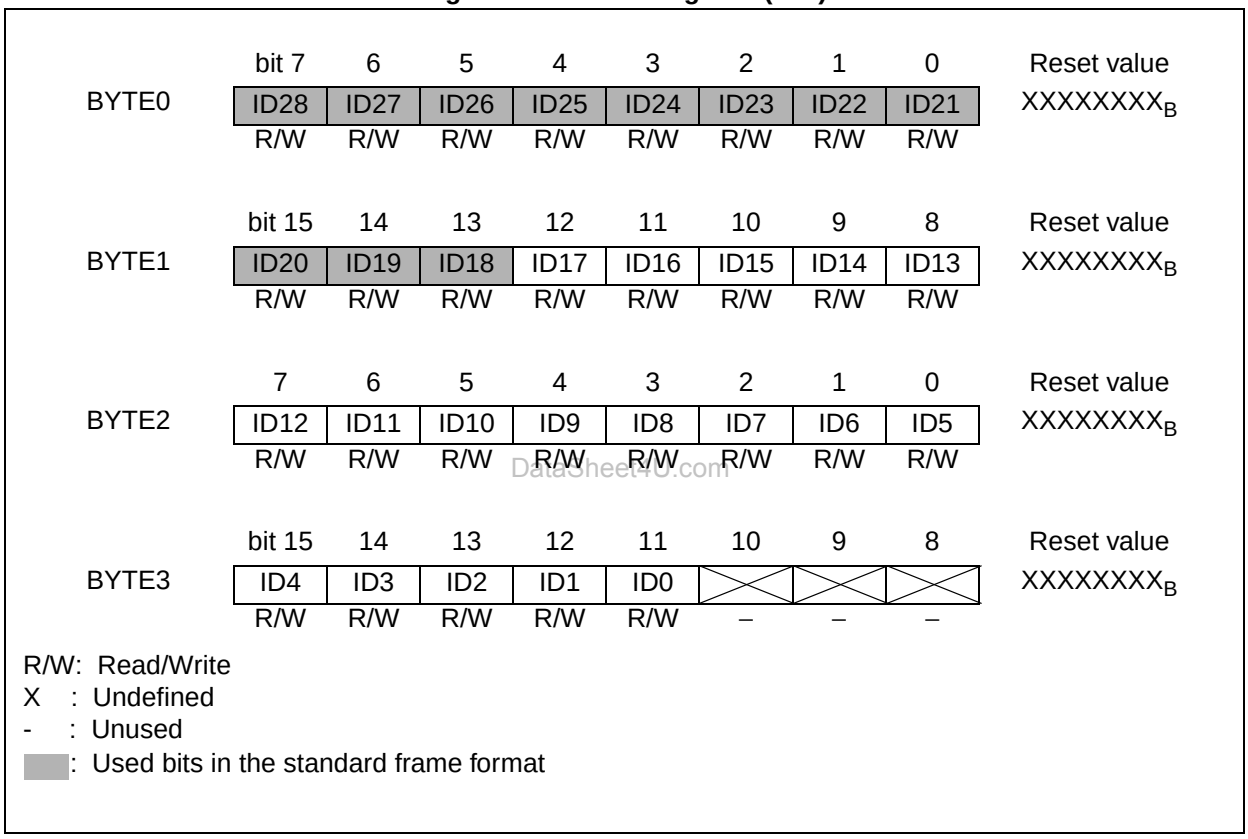
www.DataSheet4U.com

15.3.21 ID Register (IDRx, x = 7 to 0)

The ID register (IDR) sets the ID of the message buffer used for transmitting and receiving. In the standard frame format, 11 bits from ID28 to ID18 are used, and in the extended frame format, 29 bits from ID28 to ID0 are used.

■ ID Register (IDR)

Figure 15.3-28 ID Register (IDR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 15.3-22 Functions of ID Register (IDR)

Bit Name		Function
bit 0 to bit 7	ID28 to ID21: ID bits 28 to 21 (BYTE0)	These bits set the acceptance code or transmit message ID to be collated with the received message ID. **Standard frame format (IDER: IDEx = 0):** 11 bits from ID28 to ID18 are used. • The old messages left in the receive shift register are stored in ID17 to ID0. This will not affect the operation. • All received message IDs are stored even if specific bits are masked. **Extended frame format (IDER: IDEx = 1):** 29 bits from ID28 to ID0 are used. **Notes:** • When using the standard frame format (IDER: IDEx = 0), the bits from ID28 to ID22 cannot be all set to 1. • When setting the ID register (IDR), perform writing by words. Writing by bytes is disabled. • The ID register (IDR) should be set after disabling the message buffer (x) to be set (BVALR: BVALx = 0). Setting the ID register (IDR) with the message buffer (x) enabled may store a message unnecessary received.
bit 8 to bit 15	ID20 to ID13: ID bits 20 to 13 (BYTE1)	
bit 0 to bit 7	ID12 to ID5: ID bits 28 to 21 (BYTE2)	
bit 11 to bit 15	ID4 to ID 0: ID bits 4 to 0 (BYTE3)	

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Setting example of ID register (IDR)

Table 15.3-23 gives a setting example of the ID register (IDR) in the standard and extended frame formats.

Table 15.3-23 Example of ID Setting in Standard and Extended Frame Formats

Standard Frame Format			
ID (Dec)	ID (Hex)	BYTE0	BYTE1
1	1	00	20
2	2	00	40
3	3	00	60
4	4	00	80
5	5	00	A0
6	6	00	C0
7	7	00	E0
8	8	01	00
9	9	01	20
10	A	01	40
⋮		⋮	
30	1E	03	C0
31	1F	03	E0
32	20	04	00
⋮		⋮	
100	64	0C	80
101	65	0C	A0
⋮		⋮	
200	C8	19	00
⋮		⋮	
2043	7FB	FF	60
2044	7FC	FF	80
2045	7FD	FF	A0
2046	7FE	FF	C0
2047	7FF	FF	E0

Extended Frame Format					
ID (Dec)	ID (Hex)	BYTE0	BYTE1	BYTE2	BYTE3
1	1	00	00	00	08
2	2	00	00	00	10
3	3	00	00	00	18
4	4	00	00	00	20
5	5	00	00	00	28
6	6	00	00	00	30
7	7	00	00	00	38
8	8	00	00	00	40
9	9	00	00	00	48
10	A	00	00	00	50
⋮		⋮			
30	1E	00	00	00	F0
31	1F	00	00	00	F8
32	20	00	00	01	00
⋮		⋮			
100	64	00	00	03	20
101	65	00	00	03	28
⋮		⋮			
200	C8	00	00	06	40
⋮		⋮			
2043	7FB	00	00	3F	D8
2044	7FC	00	00	3F	E0
2045	7FD	00	00	3F	E8
2046	7FE	00	00	3F	F0
2047	7FF	00	00	3F	F8
⋮		⋮			
8190	1FFE	00	00	FF	F0
8191	1FFF	00	00	FF	F8
8192	2000	00	01	00	00
⋮		⋮			
536870905	1FFFFFF9	FF	FF	FC	80
536870906	1FFFFFFA	FF	FF	FD	00
536870907	1FFFFFFB	FF	FF	FD	80
536870908	1FFFFFFC	FF	FF	FE	00
536870909	1FFFFFFD	FF	FF	FE	80
536870910	1FFFFFFE	FF	FF	FF	00
536870911	1FFFFFFF	FF	FF	FF	80

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

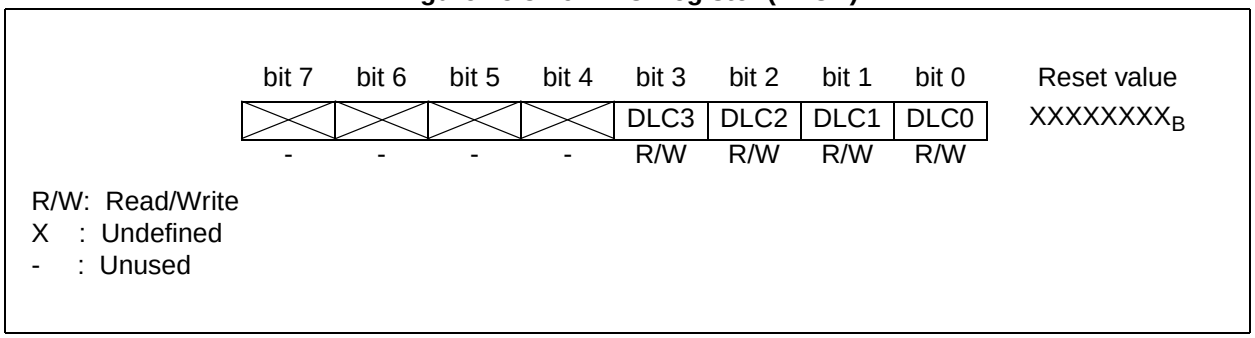
www.DataSheet4U.com

15.3.22 DLC Register (DLCR)

The DLC register (DLCR) sets the data length of the message to be transmitted or received.

■ DLC Register (DLCR)

Figure 15.3-29 DLC Register (DLCR)



et4U.com

Table 15.3-24 Functions of DLC Register (DLCR)

DataSheet4U.com

Bit Name		Function
bit 0 to bit 3	DLC3 to DLC0: Data length setting bits	These bits set the data length (byte count) of the message to be transmitted or received. **When data frame transmitted:** The data length (byte count) of the transmit message is set. **When remote frame transmitted:** The data length (byte count) of the request message is set. **When data frame received:** The data length (byte count) of the received message is stored. **When remote frame received:** The data length (byte count) of the request message is stored. **Notes:** • The data length should be set within the range of 0 to 8 bytes. • When setting the DLC register (DLCR), write by words. Writing by bytes is disabled.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

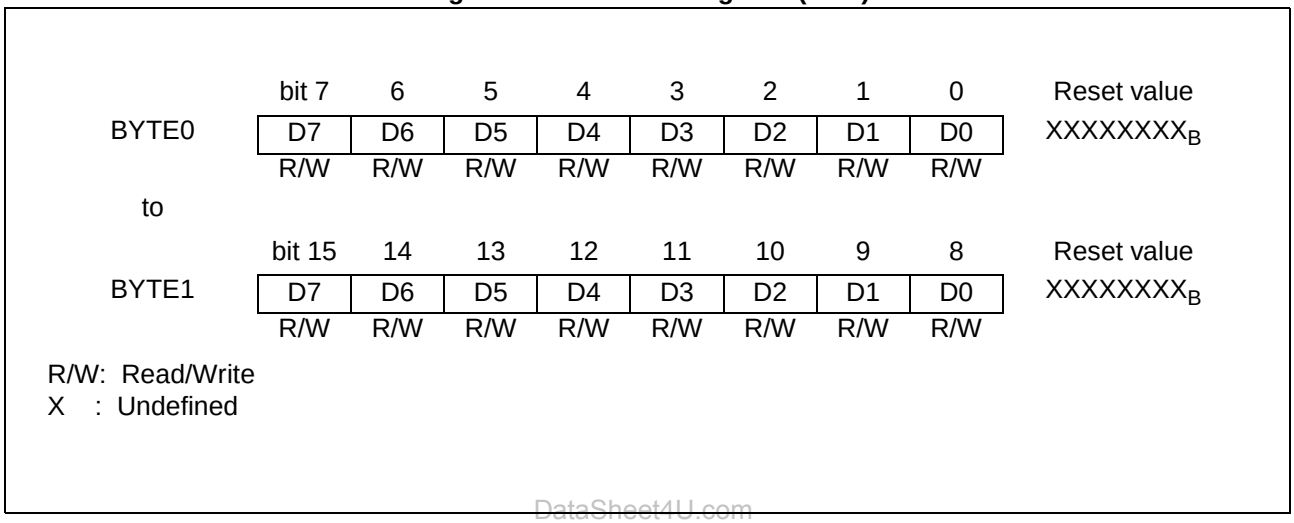
www.DataSheet4U.com

15.3.23 Data Register (DTR)

The data register (DTR) sets the messages at transmitting or receiving a data frame. The data length can be set from 0 to 8 bytes.

■ Data Register (DTR)

Figure 15.3-30 Data Register (DTR)



et4U.com DataShee DataSheet4U.com

Table 15.3-25 Functions of Data Register (DTR)

Bit Name		Function
bit 0 to bit 15	D7 to D0 (BYTE7 to BYTE0): Data bits 7 to 0	• The data register (DTRx) is used only for transmitting or receiving a data frame, and is not used for a remote frame. • The transmit message is set up to 8 bytes. The message is transmitted on an MSB-first basis starting with the small message buffer number (BYTE0 to BYTE7). • The received message is stored on an MSB-first basis starting with the small message buffer number (BYTE0 to BYTE7). • If the received message is less than 8 bytes, undefined data is stored in the rest of the bytes of the data register (DTRx). However this does not affect the operation. **Note:** When setting the data register (DTR), write by words. Writing by bytes is disabled.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.4 Interrupts of CAN Controller

The CAN controller has a transmit complete interrupt, receive complete interrupt and node state transition interrupt, and can generate interrupts when;

- **The transmission complete bit (TCR: TCx) is set.**
- **The reception complete bit (RCR: RCx) is set.**
- **The node status transition flag (CSR: NT) is set.**

■ Interrupts of CAN Controller

Table 15.4-1 shows the interrupt control bits and interrupt factors of the CAN controller.

Table 15.4-1 Interrupt Control Bits and Interrupt Factors of CAN Controller

Transmit/ Receive	Interrupt Flag Bit	Interrupt Factor	Interrupt Enable Bit	Clearing of Interrupt Request Flag
Transmit	Transmission complete bit TCR: TCx=1	Message transmitting complete	Transmission complete interrupt enable bit TIER: TIEx = 1	Setting transmission request bit (TREQR: TREQx = 1) Writing 0 to transmission complete bit (TCR: TCx)
Receive	Reception complete bit RCR: RCx=1	Message receiving complete	Reception complete interrupt enable bit RIER: RIEx=1	Writing 0 to reception complete bit (RCR: RCx)
Transmit	Node status transition flag CSR: NT=1	Node status transition	Node status transition interrupt enable bit CSR: NIE=1	Writing 0 to node status transition flag (CSR: NT)

● Transmission complete interrupt

When message transmitting is completed, 1 is set to the TCx bit in the transmission complete register (TCR). When a transmission complete interrupt is enabled (TIER: TIEx = 1) and when TCx = 1, a transmission complete interrupt is generated. When a transmission request to the message buffer is set (TREQR: TREQx = 1), the TCx bit in the transmission complete register (TCR) is automatically cleared to 0. When 0 is written to the TCx bit in the transmission complete register (TCR) after the completion of message transmitting (TCR: TCx = 1), the TCx bit is cleared.

● Reception complete interrupt

When message receiving is completed, 1 is set to the RCx bit in the receive complete register (RCR). When a reception complete interrupt is enabled (RIER: RIEx = 1) and when RCx = 1, a reception complete interrupt is generated. When 0 is written to the RCx bit in the reception complete register (RCR) after the completion of message receiving (RCR: RCx = 1), the RCx bit is cleared.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Node status transition interrupt

  When the node status of the CAN controller changes, 1 is set to the NT bit in the control status register (CSR). If a node status transition interrupt is enabled (CSR: NIE = 1) when NT = 1, a node status transition interrupt is generated. When 0 is written to the NT bit in the control status register the NT bit is cleared.

■ Registers and Vector Tables Related to Interrupt of CAN Controller

See "3.5 Interrupt" for details of the interrupts.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.5 Explanation of Operation of CAN Controller

This section explains the procedures for transmitting and receiving messages and the setting of bit timing, frame format, ID and acceptance filter.

■ Explanation of Operation of CAN Controller

The following sections provide more details of the operation of CAN controller.

- Transmission of message (See Section "15.5.1 Transmission")
- Reception of message (See Section "15.5.2 Reception")
- Procedures for transmission/reception of message (See Section "15.5.3 Procedures for Transmitting and Receiving")
- Reception of multiple message (See Section "15.5.4 Setting Multiple Message Receiving")

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

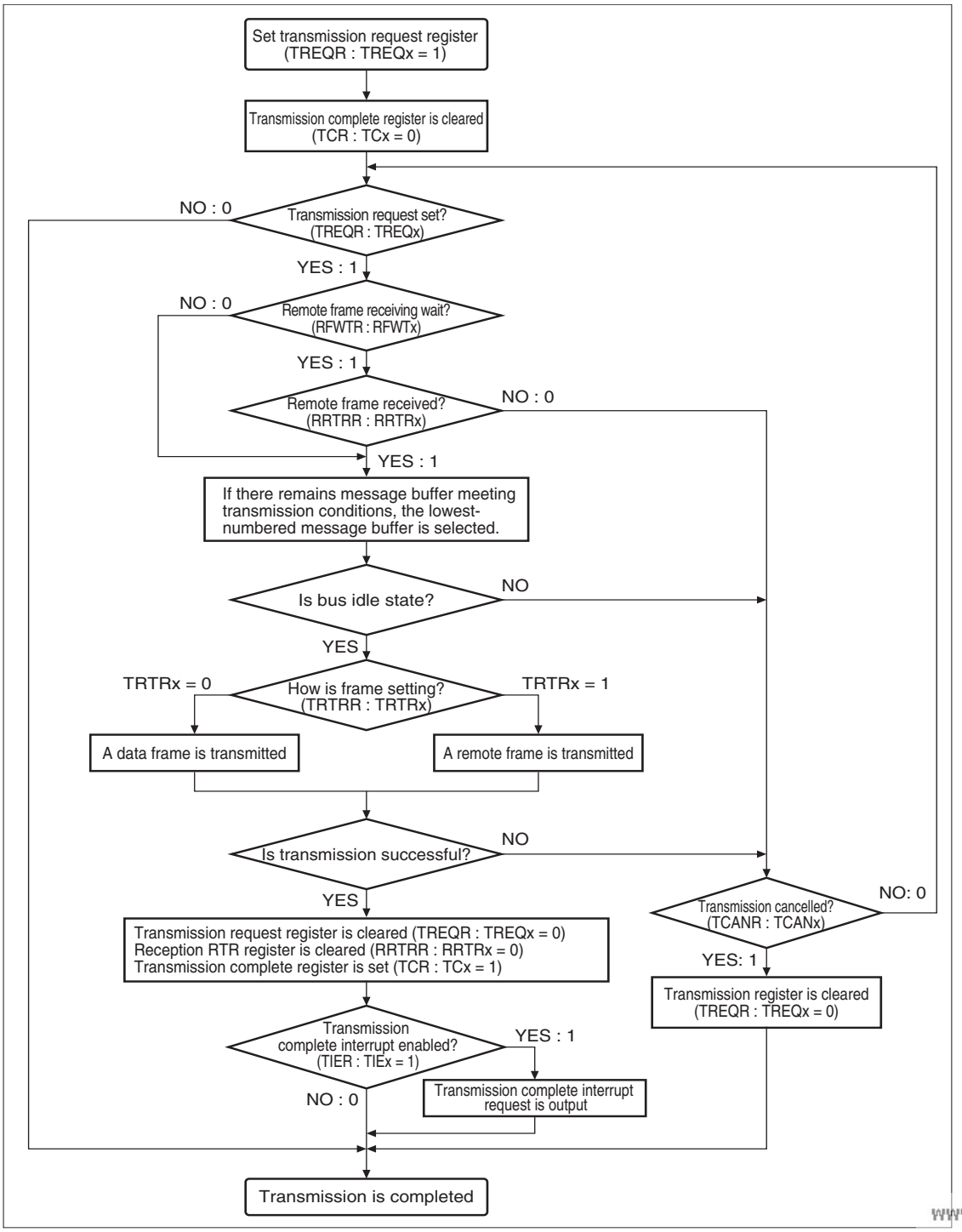
www.DataSheet4U.com

15.5.1 Transmission

Figure 15.5-1 shows a transmission flowchart.

■ Transmission

Figure 15.5-1 Transmission Flowchart



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Starting transmitting

Setting of transmission request

To start transmitting, set the TREQx bit in the transmission request register to 1 which is corresponding to the message buffer (x) that transmits the message. When the TREQx bit is set, the transmission complete register is cleared (TCR: TCx = 0).

Presence or absence of remote frame receive wait

If the RFWTx bit in the remote frame receive wait register is set, transmitting is started after a remote frame is received (RRTRR: RRTRx = 1).

If the remote frame receive wait register does not wait for receiving of a remote frame (RFWTR: RFWTx = 0), transmitting is started immediately after the transmission request bit is set (TREQR: TREQx = 1).

● Performing transmitting

Transmission request set in more than one message buffer

When a transmission request is set in more than one message buffer (TREQR: TREQx = 1), transmitting is performed starting with the small-numbered message buffer (x = 7 to 0).

Transmitting to CAN bus

Transmitting message to the CAN bus from the transmit output pin (TX) is started when the CAN bus is idle.

Arbitration

Arbitration is performed when a message buffer conflicts with transmitting from other CAN controllers on the CAN bus. If arbitration fails or an error occurs during transmitting, retransmitting is performed automatically until it succeeds after waiting until the bus goes idle again.

Selection of frame format

When 0 is set to the TRTRx bit in the transmit RTR register, a data frame is transmitted. When 1 is set to the bit, a remote frame is transmitted.

● Canceling transmit request

Cancellation by transmission cancel register (TCANR)

During transmitting message, the transmission request set in the message buffer that is not transmitted (held) can be cancelled by setting 1 in the transmission cancel register (TCANR).

When the transmission request is completely cancelled (TCANR: TCANx = 1), the transmission request register is cleared (TREQx = 1).

Cancellation by receiving message

The message buffer can receive the message even during requesting a transmitting. However, the transmission request is cancelled under the following conditions:

Request to transmit data frame:

When a data frame is received, the transmission request is cancelled. When a remote frame is received, the transmission request is not cancelled.

Request to transmit remote frame:

The transmission request is cancelled even if either a data frame or remote frame is received.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

- Completing transmitting

Success of transmitting

When transmitting is terminated normally, the TCx bit in the transmission complete register is set. The transmission request register and receive RTR register (TREQR: TREQx = 0, RRTRR: RRTRx = 0) are cleared.

Generation of transmission interrupt

When the TIEx bit in the transmission complete interrupt enable register is set, an interrupt request is generated when transmitting is completed (TCR: TCx = 1).

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

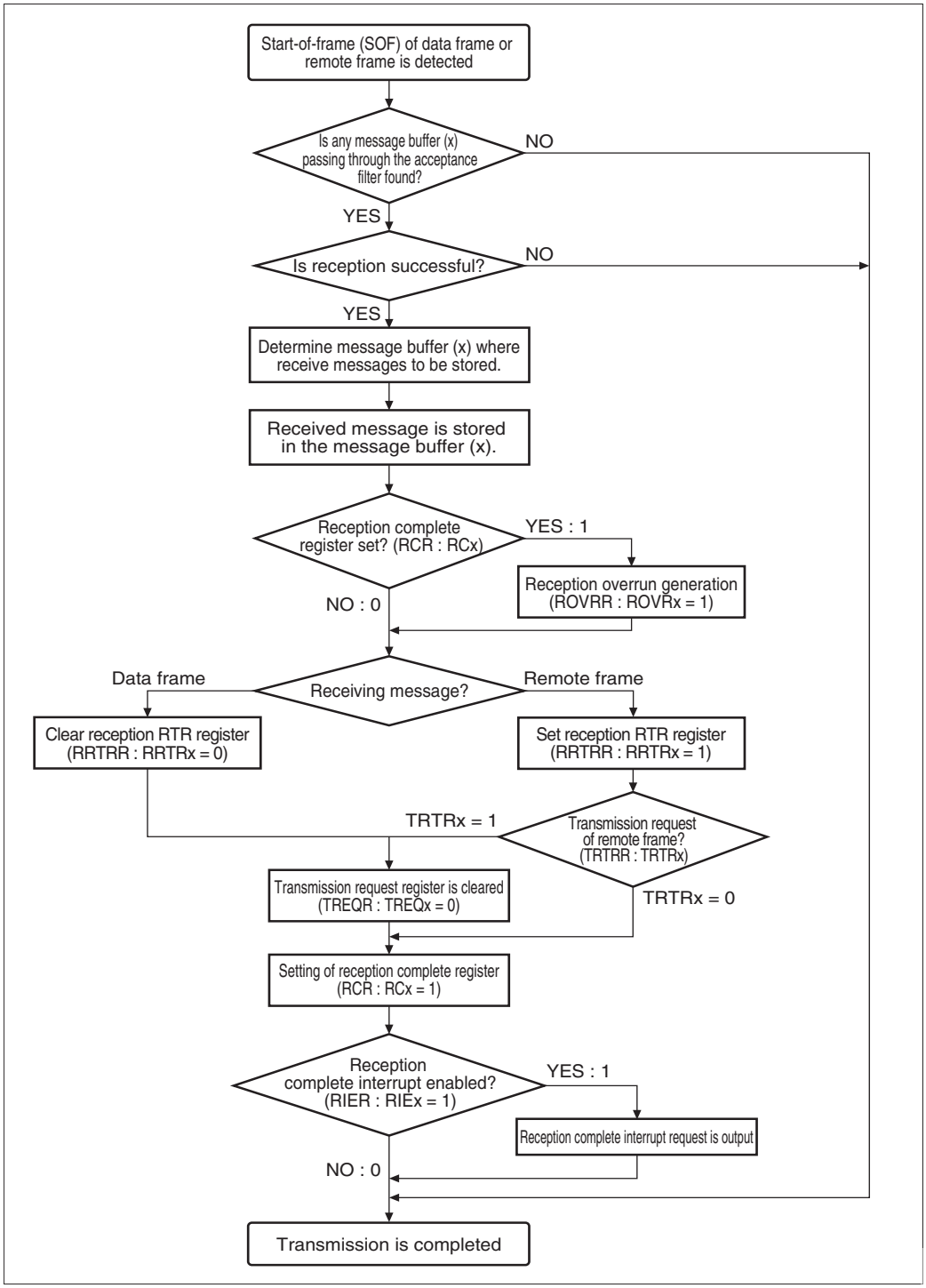
www.DataSheet4U.com

15.5.2 Reception

Figure 15.5-2 shows a reception flowchart.

■ Reception

Figure 15.5-2 Reception Flowchart



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Starting receiving

Receiving is started when the start-of-frame (SOF) of a data frame or remote frame is detected on the CAN bus.

● Acceptance filter

The received message in the standard frame format is compared with the message buffer (x) set in the standard frame format (IDER: IDEx = 0). The received message in the extended frame format is compared with the message buffer (x) set in the extended frame format (IDER: IDEx = 1).

Passing through acceptance filter

If all bits set to "compare" in the acceptance mask are matched after comparison between the received message ID and acceptance code (IDR: IDx), the received message passes the acceptance filter in the message buffer (x).

● Storing received message

If receiving message is successful, the received message is stored in the message buffer (x) that has the ID that had passed the acceptance filter.

Data frame received

The received message is stored in the ID register (IDR) and DLC register (DLCR), data register (DTR). If the received message is less than 8 bytes, undefined data is stored in the rest of the bytes in the data register (DTR).

et4U.com DataShee

Remote frame received

DataSheet4U.com

The received message is stored in the ID register (IDR) and DLC register (DLCR). The data register (DTR) remains unchanged.

More than one message buffer

If there is more than one message buffer with the ID that had passed the acceptance filter, the message buffer (x) where the received message is stored is determined under the following conditions:

- Higher priority is given to the message buffer with a smaller number (x = 0 to 7). The priority of message buffer 0 is the highest and 7 is the lowest.
- The received message is stored in preference to the message buffer that has not been completed receiving (RCR: RCx = 0).
- If the bit in the acceptance mask select register is set to "full-bit comparison" (AMSx.1 = 0, 0 = 00_B), the received message is stored in the corresponding message buffer (x), regardless of the setting value of the reception complete register (RCR: RCx).
- If there is more than one message buffer that has not been completed receiving, or if there is more than one message buffer with the AMSx.1 and AMSx.0 bits in the acceptance mask select register set to 00_B (full-bit comparison), the received message is stored in the message buffer with the smallest number (x).
- If there is no message buffer that satisfies the above conditions, the received message is stored in the message buffer with the lowest number (x).
- The message buffers should be arranged in order of ascending number (x) as follows;
  - Smallest number (x): Acceptance mask set to "full-bit comparison"
  - Middle number (x): Acceptance mask registers 0 and 1 used
  - Largest number (x): Acceptance mask set to "full-bit masking"

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

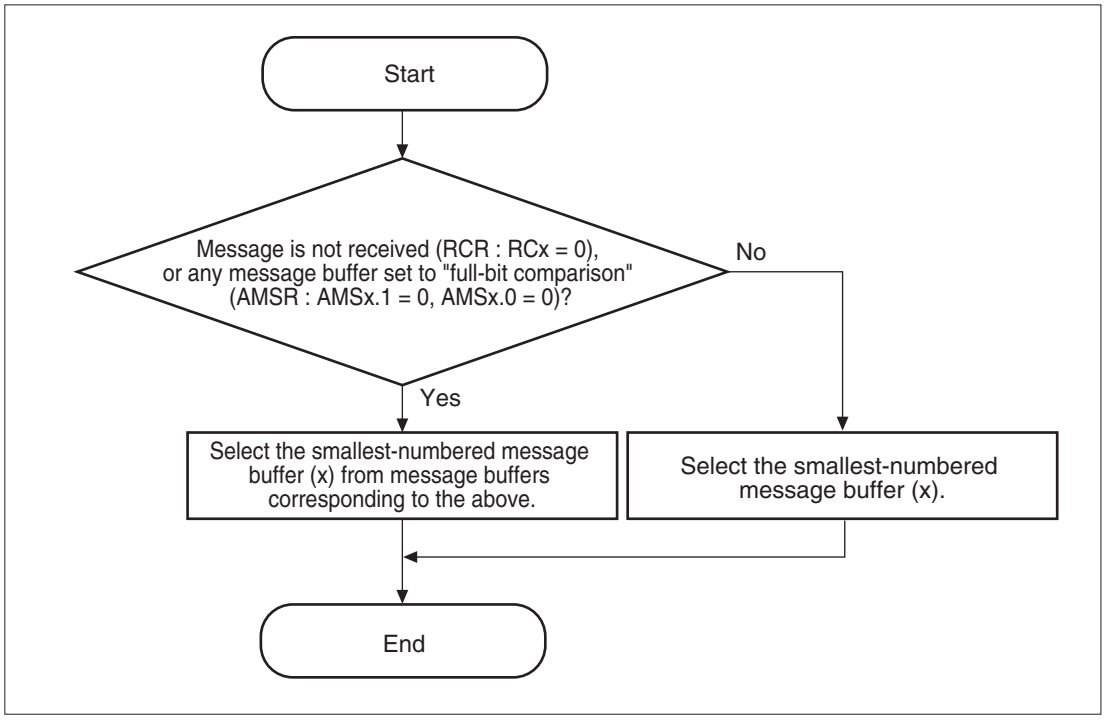
www.DataSheet4U.com

● Setting of acceptance mask select register

Table 15.5-1 Setting of Acceptance Mask Select Register

AMSx. 1	AMSx. 0	Acceptance Mask (x = 7 to 0)
0	0	Full-bit comparison is performed.
0	1	Full-bit masking is performed.
1	0	Acceptance mask register 0 (AMR0) is used.
1	1	Acceptance mask register 1 (AMR1) is used.

Figure 15.5-3 Flowchart of Determining Message Buffer that Stores Received Message



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Receive overrun

When another received message is stored in the message buffer that has completed receiving (RCR: RCx = 1), a receive overrun occurs. When a receive overrun occurs, 1 is set to the ROVRx bit in the receive overrun register corresponding to the number of the message buffer (x) where the receive overrun occurs.

● Processing for reception of data frame and remote frame

Processing for reception of data frame

- The reception RTR register is cleared (RRTRR: RRTRx = 0).
- The transmission request register is cleared (TREQR: TREQx = 0) immediately before the received message is stored. A transmission request to the message buffer (x) that does not perform transmitting is cancelled.

Note:	Either the request to transmit a data frame or a remote frame is cancelled.

Processing for reception of remote frame

- The reception RTR register is set (RRTRR: RRTRx = 1).
- If the transmission RTR register is set (TRTRR: TRTRx = 1), the transmission request register is cleared (TREQx = 0). The request to transmit a remote frame to the message buffer (x) that does not perform transmitting is cancelled.

et4U.com

Note:	The request to transmit a data frame is not cancelled. For details about how to cancel a transmit request, see "Canceling transmit request" in Section "15.5.1 Transmission".

DataShee

DataSheet4U.com

● Completing receiving

When the received message is stored, the reception complete register is set (RCR: RCx = 1). If the reception complete interrupt enable register is set (RIER: RIEx = 1), an interrupt is generated when receiving is completed (RCR: RCx = 1).

Note:	The CAN controller cannot receive any message transmitted by itself.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.5.3 Procedures for Transmitting and Receiving

The section explains the procedure for transmission/reception of message.

■ Presetting

● Setting of bit timing

- Set the bit timing register (BTR) after halting the bus operation (CSR: HALT = 1).

● Setting of frame format

- Set the frame format used in the message buffer (x). When using the standard frame format, set the IDEx bit in the IDE register (IDER) to 0. When using the extended frame format, set the IDEx bit to 1.

● Setting of ID

- Set the ID of the message buffer (x) to the ID28 to ID0 bits in the ID register (IDR). In the standard frame format, it does not have to set the ID17 to ID0 bits. The ID of the message buffer (x) is used as the transmit message ID at transmitting and as the acceptance code at receiving.
- Set the ID after disabling the message buffer (x) (BVALR: BVALx = 0). Setting the ID with the message buffer (x) enabled may store a message unnecessary received.

et4U.com DataShee

DataSheet4U.com

● Setting of acceptance filter

- The acceptance filter used in the message buffer (x) is set by a combination of the acceptance code and acceptance mask. Set the acceptance filter after disabling the message buffer (x) (BVALR: BVALx = 0). Setting the acceptance filter with the message buffer (x) enabled may store a message unnecessary received.
- The acceptance mask used for each message buffer (x) is selected by the acceptance mask select register (AMSR). When using the acceptance mask registers (AMR0 and AMR1), set the acceptance mask register (AMR0.1), too.
- Set the acceptance mask so that a transmission request will not be cancelled by storing a message unnecessary received.

DataSheet4U.com www.DataSheet4U.com

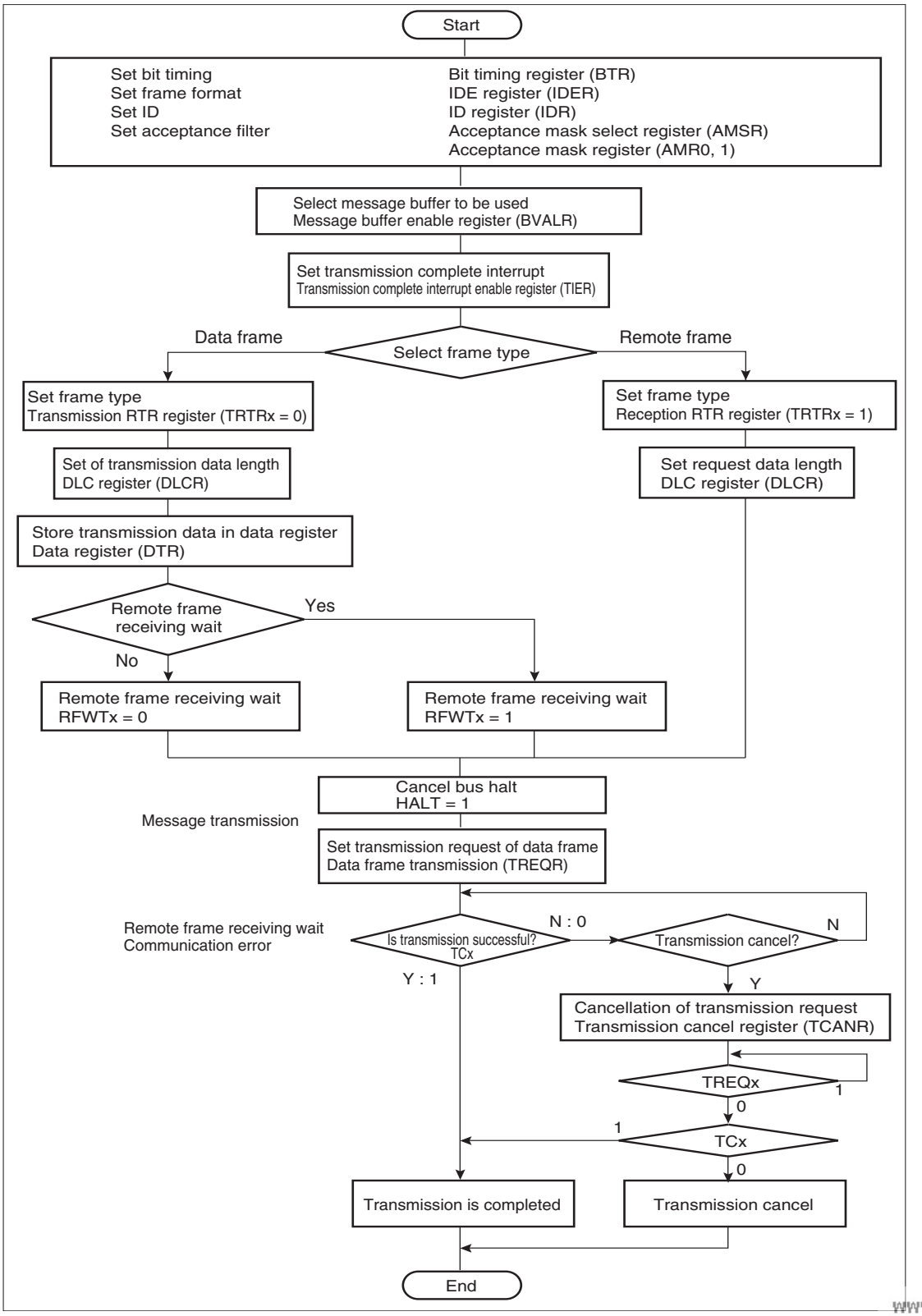
DataSheet4U.com

www.DataSheet4U.com

■ Procedure for Transmitting message Buffer (x)

Figure 15.5-4 shows a procedure for the transmit setting.

Figure 15.5-4 Flowchart of Procedure for Transmit Setting



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Procedure for transmission message buffer (x)

After completion of presetting, set the message buffer (x) enabled (BVALR: BVALx =1) by message buffer enable register.

● Setting transmit data length code

- Set the transmit data length code (byte count) to the DLC3 to DLC0 bits in the DLC register (DLCR).
- When transmitting a data frame (TRTRR: TRTRx = 0), set the data length of the transmit message.
- When transmitting a remote frame (TRTRR: TRTRx = 1), set the data length (byte count) of the message to be requested.

Note:	Setting other than "0000_B" to "1000_B" (0 to 8 bytes) is prohibited.

● Setting transmit data (only for transmission of data frame)

When transmitting a data frame (TRTRR: TRTRx = 0), set the data of byte count to be transmitted in the data register (DTR).

Note:	Rewrite transmit data after setting the TREQx bit in the transmit request register to 0. There is no need to set the bit disabled in the message buffer enable register (BVALR: BVALx = 0). When the bit is set to disabled, no remote frame can be received.

et4U.com DataShee

● Setting transmission RTR register (TRTRR)

DataSheet4U.com

- When transmitting a data frame, set the TRTRx bit in the transmission RTR register to 0.
- When transmitting a remote frame, set the TRTRx bit in the transmission RTR register to 1.

● Setting conditions for starting transmitting (only in transmitting data frame)

- When setting the request to transmit a data frame (TREQR: TREQx = 1 and TRTRR: TRTRx = 0) and starting transmitting immediately, set the RFWTx bit in the remote frame wait register to 0.
- When setting the request to transmit a data frame (TREQR: TREQx = 1 and TRTRR: TRTRx = 0) and starting transmitting after waiting until a remote frame is received (RRTRR: RRTRx = 1), set the RFWTx bit in the remote frame wait register to 1.

Note:	When the RFWT bit in the remote frame wait register is set to 1, no remote frame can be transmitted.

● Setting transmission complete interrupt

- When enabling an interrupt when transmitting is completed (TCR: TCx = 1), set the TIEx bit in the transmit complete enable register to 1.
- When disabling an interrupt when transmitting is completed (TCR: TCx = 1), set the TIEx bit in the transmission complete enable register to 0.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Canceling bus halt

After the completion of setting bit timing and transmitting, write 0 to the HALT bit in the control status register (CSR: HALT) to cancel the bus halt.

● Setting transmission request

To set a transmission request, set the TREQx bit in the transmission request register to 1.

● Canceling transmission request

- To cancel the transmission request held in the message buffer (x), write 1 to the TCANx bit in the transmission cancel register.
- Check the TREQx bit in the transmission request register (TREQR). When the TREQx bit is 0, transmission cancel is terminated or transmitting is completed. After that, check the TCx bit in the transmission complete register (TCR). If the TCx bit is 0, transmission cancel is terminated and if the TCx bit is 1, transmitting is completed.

● Processing when transmitting completed

- When transmitting is successful, 1 is set to the TCx bit in the transmit complete register (TCR).
- When a transmission complete interrupt is enabled (TIER: TIEx = 1), an interrupt is generated.
- After checking the completion of transmitting, write 0 to the TCx bit in the transmission complete register (TCR) to clear the transmission complete register (TCR). When the transmission complete register (TCR) is cleared, the transmission complete interrupt is cancelled.
- When the message is received or stored, the held transmission requests are cancelled as follows:
  - When a data frame is received, the request to transmit a data frame is cancelled.
  - When a data frame is received, the request to transmit a remote frame is cancelled.
  - When a remote frame is received, the request to transmit a remote frame is cancelled.

When a remote frame is received or stored, the request to transmit a data frame is not cancelled but the data in the ID register and DLC register are rewritten to the data of the received remote frame. Therefore, the data in the ID register and DLC register for the data frame to be transmitted are replaced by data in the received remote frame.

DataSheet4U.com
www.DataSheet4U.com

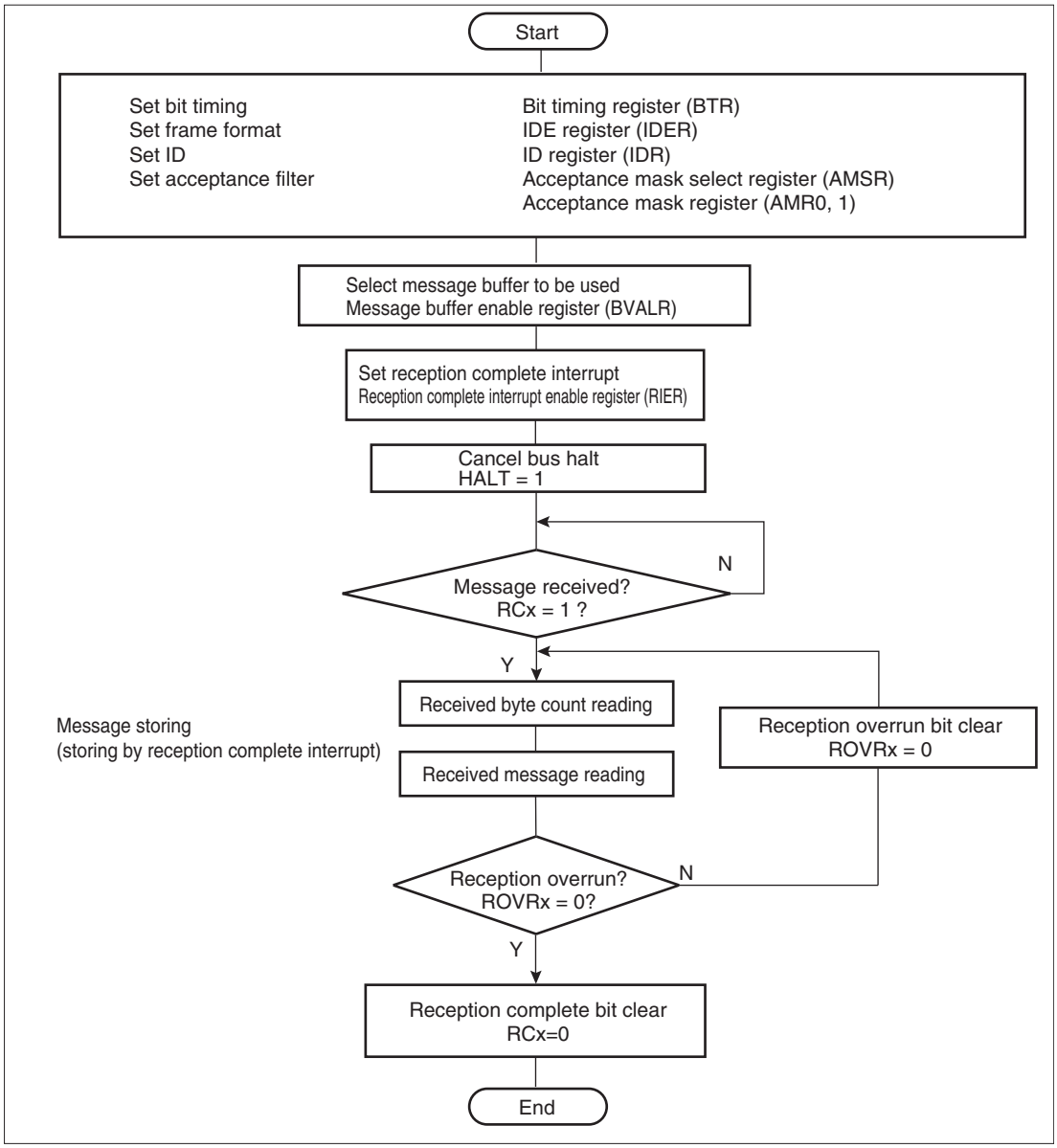
DataSheet4U.com

www.DataSheet4U.com

■ Procedure for Receiving Message Buffer (x)

Figure 15.5-5 shows the procedure for the receiving setting.

Figure 15.5-5 Flowchart of Procedure for Receive Setting



● Procedure for receiving message buffer (x)

After presetting, perform the following setting:

● Setting reception complete interrupt

- To generate a reception complete interrupt, set the RIEx bit in the reception complete interrupt enable register (RIER) to 1.
- To disable a reception complete interrupt (RCR: RCx = 1), set the RIEx bit to 0.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Starting receiving

To start receiving after the completion of setting, set the BVALx bit in the message buffer enable register (BVALR) to 1 and enable the message buffer (x).

● Canceling bus halt

After the completion of setting bit timing and transmitting, write 0 to the HALT bit in the control status register (CSR: HALT) to cancel the bus halt.

● Processing when receiving completed

- If reception is successful after passing through the acceptance filter, the received message is stored in the message buffer (x), 1 is set to the RCx of the reception complete register (RCR). For data frame reception, RRTRx bit of the remote request receive register (RRTRR) is cleared to 0. For remote frame reception, 1 is set to the RRTRx bit.
- If a reception interrupt is enabled (RIEx of the reception interrupt enable register (RIER) is 1), an interrupt is generated.
- Process the received message after checking the completion of receiving (RCR: RCx = 1).
- Check the ROVRx bit in the receive overrun register (ROVRR) after the completion of processing the received message.
  - If the ROVRx bit is set to 0, the received message is enabled. When 0 is written to the RCx bit (a reception complete interrupt is also cancelled), receiving is terminated.
  - If the ROVRx bit is set to 1, a receive overrun occurs and the new message may overwrite the received message. When a receive overrun occurs, write 0 to the ROVRx bit and then process the received message again.

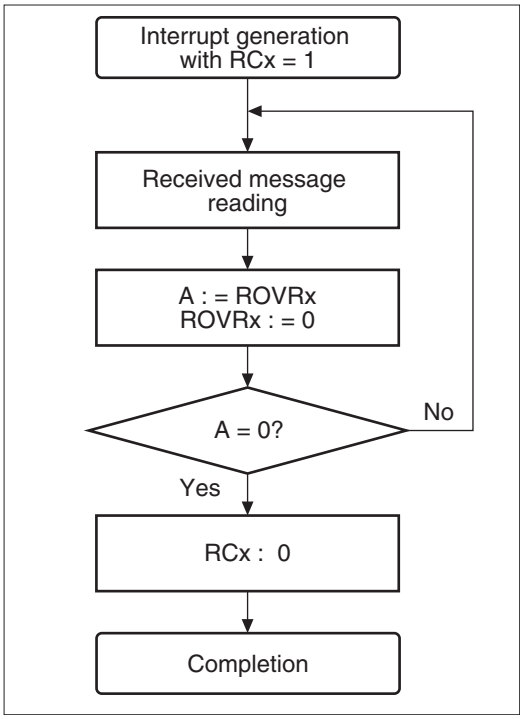
et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 15.5-6 shows an example of reception interrupt processing.

Figure 15.5-6 Example of Reception Interrupt Processing



et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.5.4 Setting Multiple Message Receiving

When there is insufficient time to receive messages such as frequently received messages or messages with different IDs, more than one message buffer can be combined to a multiple message buffer to give the CPU sufficient time to process received messages.
To configure multiple message buffers, perform the same setting of acceptance filter of the message buffers to be combined.

■ Setting Configuration of Multiple Message Buffer

When four messages in the standard frame format are received with doing the acceptance filter of message buffers 5, 6 and 7 on the same settings, the multiple message buffer operates as shown in the figure.

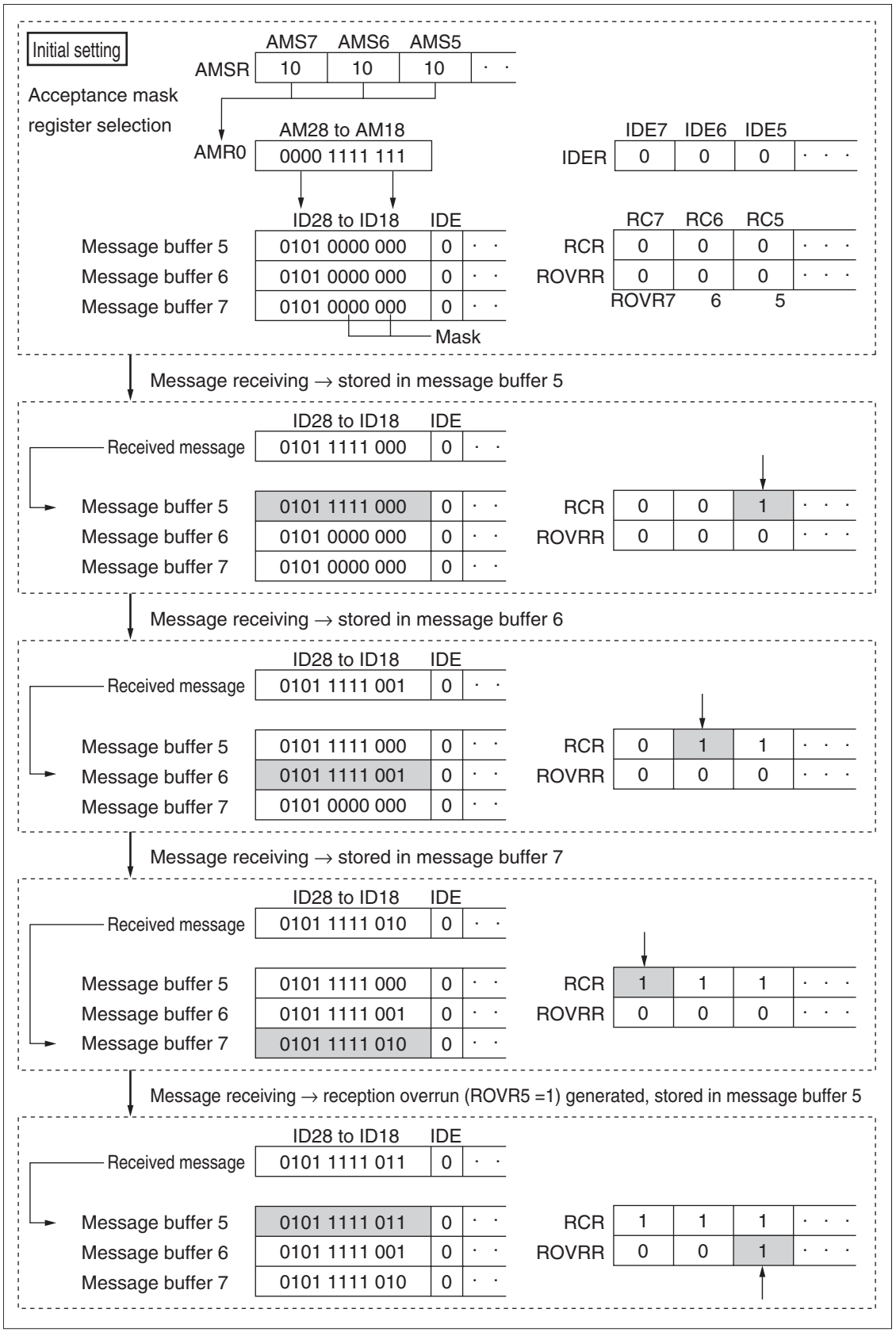
Note:	When the acceptance mask select register is set to "full-bit comparison" (AMSR: AMSx.1, AMSx.0 = 00_B), do not set the same acceptance code. When the register is set to "full-bit comparison", the messages are always stored in the message buffer with the smaller number, so the message buffers cannot be formed into a multiple message buffer.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 15.5-7 Example of Operation of Multiple Message Buffer



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.6 Precautions when Using CAN Controller

Use of the CAN Controller requires the following cautions.

■ Caution for Disabling Message Buffers by BVAL bits

The use of BVAL bits may affect malfunction of CAN Controller when messages buffers are set disabled while CAN Controller is participating in CAN communication (read value of HALT bit is 0 and CAN Controller is ready to receive or transmit messages). This section shows the work around of this malfunction.

● Condition

When following two conditions occur at the same time, CAN Controller will not perform to receive or transmit messages normally.

- CAN Controller is participating in the CAN communication. (i.e. The read value of HALT bit is 0 and CAN Controller is ready to receive or transmit messages)
- Message buffers are read or written when the message buffers are disabled by BVAL bits.

● Work around

et4U.com DataShee

Operation for re-configuring receiving message buffers

DataSheet4U.com

While CAN Controller is participating in CAN communication (the read value of HALT bit is 0 and CAN Controller is ready to receive or transmit messages), it is necessary to following one from the two operations described below to re-configure message buffers by ID, AMS and AMR0/1 register-settings.

- Use of HALT bit
  - Write 1 to HALT bit and read it back for checking the result is 1. Then change the settings for ID/ AMS/AMR0/1 registers.
- No Use of Message Buffer 0
  - Don't use the message buffer 0. In other words, disable message buffer (BVAL0=0), prohibit receive interrupt (RIE0=0) and do not request transmission (TREQ0=0).

Operation for processing received message

Don't use the receiving prohibition by BVAL bit to avoid over-written of next message. Use the ROVR bit for checking if over-write has been performed. For details, refer to section "15.3.16 Reception Overrun Register (ROVRR)" and "15.5.3 Procedures for Transmitting and Receiving".

Operation for suppressing transmission request

Don't use BVAL bit for suppressing transmission request, use TCAN bit instead of it.

Operation for composing transmission message

For composing a transmission message, it is necessary to disable the message buffer by BVAL bit to change contents of ID and IDE registers. In this case, BVAL bit should reset (BVAL=0) after checking if TREQ bit is 0 or after completion of the previous message transmission (TC=1).

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

15.7 Program Example of CAN Controller

This section shows the program example of CAN controller.

■ Program Example of CAN Transmission and Reception

● Processing specifications

- Set message buffer 5 of CAN to data frame transmit mode and message buffer 0 to data frame receive mode.
- Setting of frame format: Standard frame format
- Setting of ID: message buffer 0 ID = 1, message buffer 5 ID = 5
- Baud rate: 100 Kbps (machine clock = 16 MHz)
- Acceptance mask selection: full-bit comparison
- After entering the bus mode (HALT = 0), data $A0A0_H$ is transmitted.
- A transmission request is made within the transmission complete interrupt routine (TREQx=1) to transmit the same data (When TREQx is set to start sending, the transmission complete interrupt bit is cleared).
- The reception complete interrupt bit is cleared within the reception interrupt routine.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Coding example

```
          :
          :
          :
;//Setting of data format (CAN initialization)
        MOVW BTR,#05CC7H        ; Setting  baud rate 100 Kbps
        MOVW IDER, #0000H       ; (Machine clock = 16 MHz)
                                ; Setting of frame format
                                ; (0:  Standard, 1:Expanded)
        MOVW IDR51,#0A000H      ; Setting of data frame 5 ID (ID = 5)
        MOVW IDR01,#2000H       ; Setting of data frame 0 ID (ID = 1)
        MOVW AMSR,#0000H        ; Acceptance mask select register
                                ; (full-bit comparison)
        MOVW BVALR,#021H        ; Message buffers 5 and 0 enabled
   ;//Transmit setting
        MOVW DLCR5,#02H         ; Setting of transmission data length
                                ; (00H:  0-byte length, 08H:  8-byte length)
        MOVW RFWTR,#0000H       ; Remote frame receive wait register
        MOVW TRTRR,#0000H       ; Transmission RTR register (0:  Data frame
                                ; transmission, 1:  Remote frame transmission)
        MOVW TIER,#0020H        ; Transmission complete interrupt enable register
   ;//Reception setting
        MOVW RIER,#0001H        ; Reception complete interrupt enable register
;//Bus operation start
        MOV  CSR0,#80H          ; Control status register (HALT=0)
sthlt   BBS  CSR0:0,sthlt       ; Wait until HALT=0
   ;//Transmission data set
        MOVW DTR5,#0A0A0H       ; Write A0A0H to data register of message buffer 5.
        MOVW TREQR,#0020H       ; Transmission request register
                                ; (1:  Transmission start, 0:  Transmission stop)
          :
          :
          :
;//Reception complete interrupt
CANRX
        MOVW RCR,#0000H         ; Reception complete register
            RETI
;//Transmission complete interrupt
CANTX
        MOVW TREQR,#0020H       ; Transmission request register
                                ; (1: Transmission start, 0: Transmission stop)
        RETI
```

et4U.com
DataShee
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 16

8/16 ADDRESS MATCH DETECTION FUNCTION

This chapter explains the address match detection function and its operation.

16.1 Overview of Address Match Detection Function
16.2 Block Diagram of Address Match Detection Function
16.3 Configuration of Address Match Detection Function
16.4 Explanation of Operation of Address Match Detection Function
16.5 Program Example of Address Match Detection Function

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

16.1 Overview of Address Match Detection Function

If the address of the instruction to be processed next to the instruction currently processed by the program matches the address set in the detect address setting registers, the address match detection function forcibly replaces the next instruction to be processed by the program with the INT9 instruction to branch to the interrupt processing program. Since the address match detection function can use the INT9 interrupt for instruction processing, the program can be corrected by patch processing.

■ Overview of Address Match Detection Function

- The address of the instruction to be processed next to the instruction currently processed by the program is always held in the address latch through the internal bus. The address match detection function always compares the value of the address held in the address latch with that of the address set in the detect address setting registers. When these compared values match, the next instruction to be processed by the CPU is forcibly replaced by the INT9 instruction, and the interrupt processing program is executed.
- There are two detect address setting registers (PADR0 and PADR1), each of which has an interrupt enable bit. The generation of an interrupt due to a match between the address held in the address latch and the address set in the detect address setting registers can be enabled and disabled for each register.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

16.2 Block Diagram of Address Match Detection Function

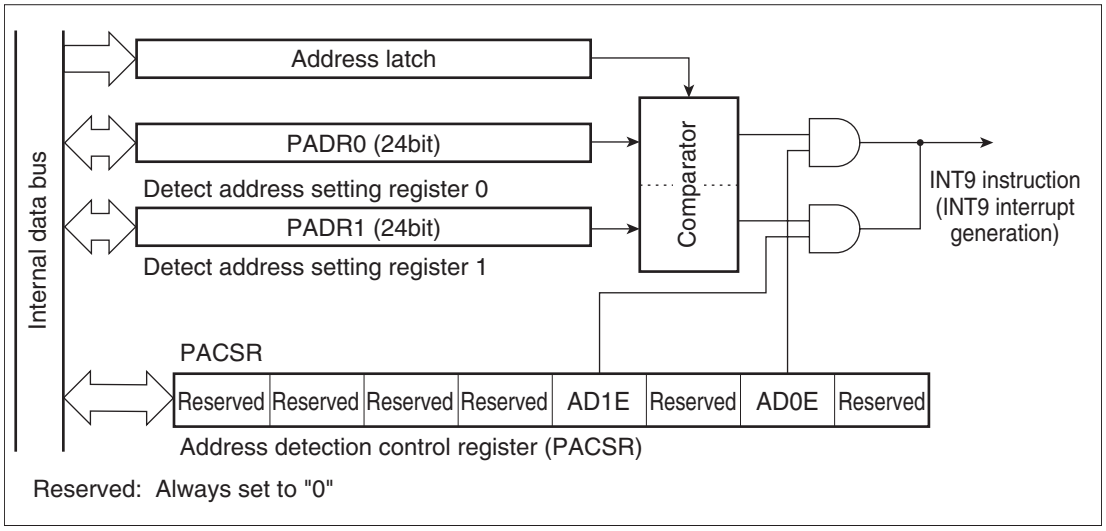
The address match detection module consists of the following blocks:

- **Address latch**
- **Address detection control register (PACSR)**
- **Detect address setting registers**

■ Block Diagram of Address Match Detection Function

Figure 16.2-1 shows the block diagram of the address match detection function.

Figure 16.2-1 Block Diagram of the Address Match Detection Function



● Address latch

The address latch stores the value of the address output to the internal data bus.

● Address detection control register (PACSR)

The address detection control register enables or disables output of an interrupt at an address match.

● Detect address setting registers (PADR0, PADR1)

The detect address setting registers set the address that is compared with the value of the address latch.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

16.3 Configuration of Address Match Detection Function

This section details the registers used by the address match detection function.

■ List of Registers and Reset Values of Address Match Detection Function

Figure 16.3-1 List of Registers and Reset Values of Address Match Detection Function

	bit	7	6	5	4	3	2	1	0
Address detection control register (PACSR)		0	0	0	0	0	0	0	0
	bit	7	6	5	4	3	2	1	0
Detect address setting register 0 (PADR0) : High		X	X	X	X	X	X	X	X
	bit	15	14	13	12	11	10	9	8
Detect address setting register 0 (PADR0) : Middle		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
Detect address setting register 0 (PADR0) : Low		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
Detect address setting register 1 (PADR1) : High		X	X	X	X	X	X	X	X
	bit	15	14	13	12	11	10	9	8
Detect address setting register 1 (PADR1) : Middle		X	X	X	X	X	X	X	X
	bit	7	6	5	4	3	2	1	0
Detect address setting register 1 (PADR1) : Low		X	X	X	X	X	X	X	X

X: Undefined

DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

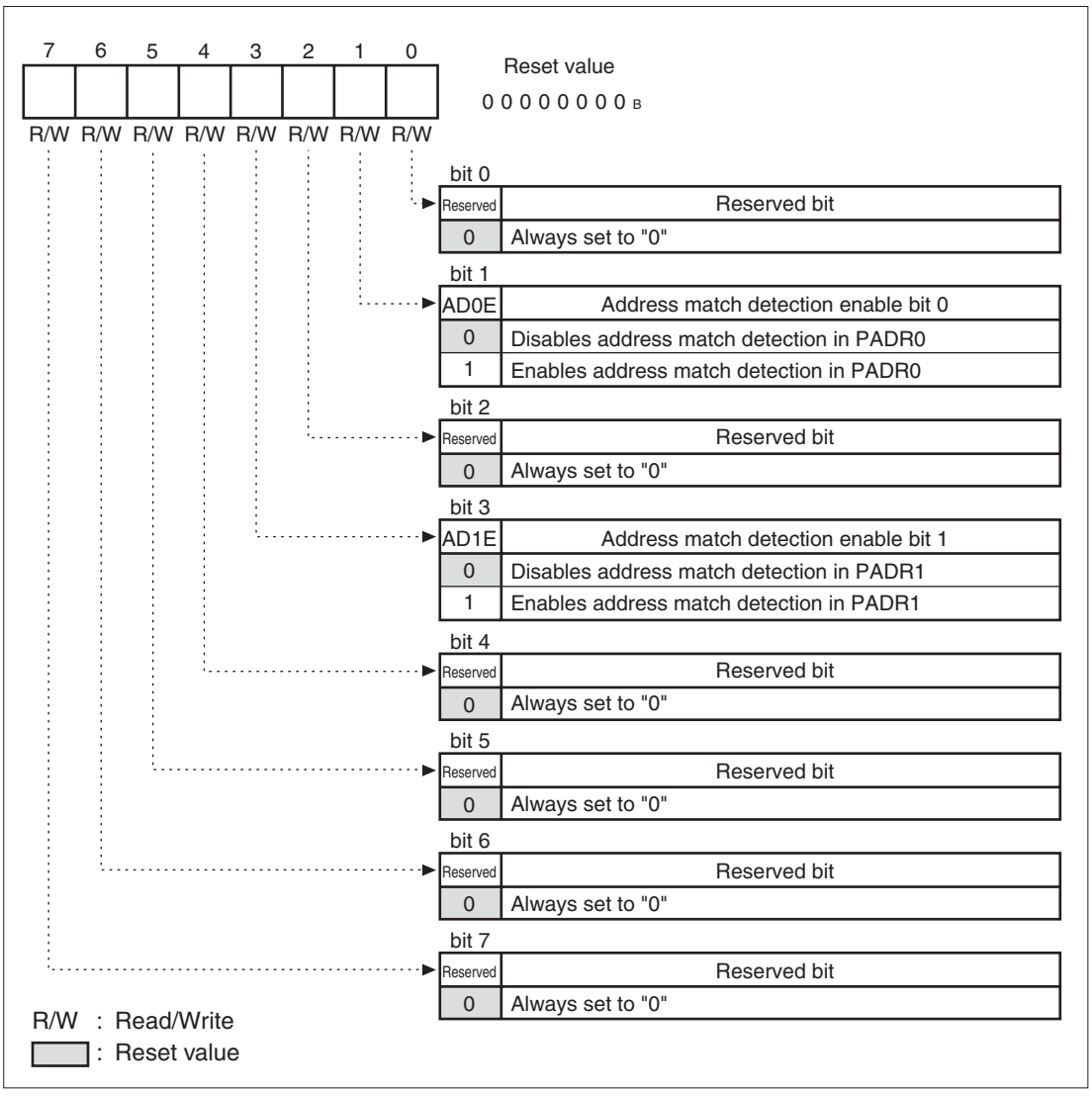
www.DataSheet4U.com

16.3.1 Address Detection Control Register (PACSR)

The address detection control register (PACSR) enables or disables output of an interrupt at an address match. When an address match is detected when output of an interrupt at an address match is enabled, the INT9 interrupt is generated.

■ Address Detection Control Register (PACSR)

Figure 16.3-2 Address Detection Control Register (PACSR)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 16.3-1 Functions of Address Detection Control Register (PACSR)

Bit Name		Function
bit 0	Reserved: reserved bit	Always set to 0.
bit 1	AD0E: Address match detection enable bit 0	The address match detection operation with the detect address setting register 0 (PADR0) is enabled or disabled. **When set to 0:** Disables the address match detection operation. **When set to 1:** Enables the address match detection operation. • When the value of detect address setting register 0 (PADR0) matches with the value of address latch at enabling the address match detect operation (AD0E = 1), the INT9 instruction is immediately executed.
bit 2	Reserved: reserved bit	Always set to 0.
bit 3	AD1E: Address match detection enable bit 1	The address match detection operation with the detect address setting register 1 (PADR1) is enabled or disabled. **When set to 0:** Disables the address match detection operation. **When set to 1:** Enables the address match detection operation. • When the value of detect address setting registers 1 (PADR1) matches with the value of address latch at enabling the address match detection operation (AD1E = 1), the INT9 instruction is immediately executed.
bit 4 to bit 7	Reserved: reserved bit	Always set to 0.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

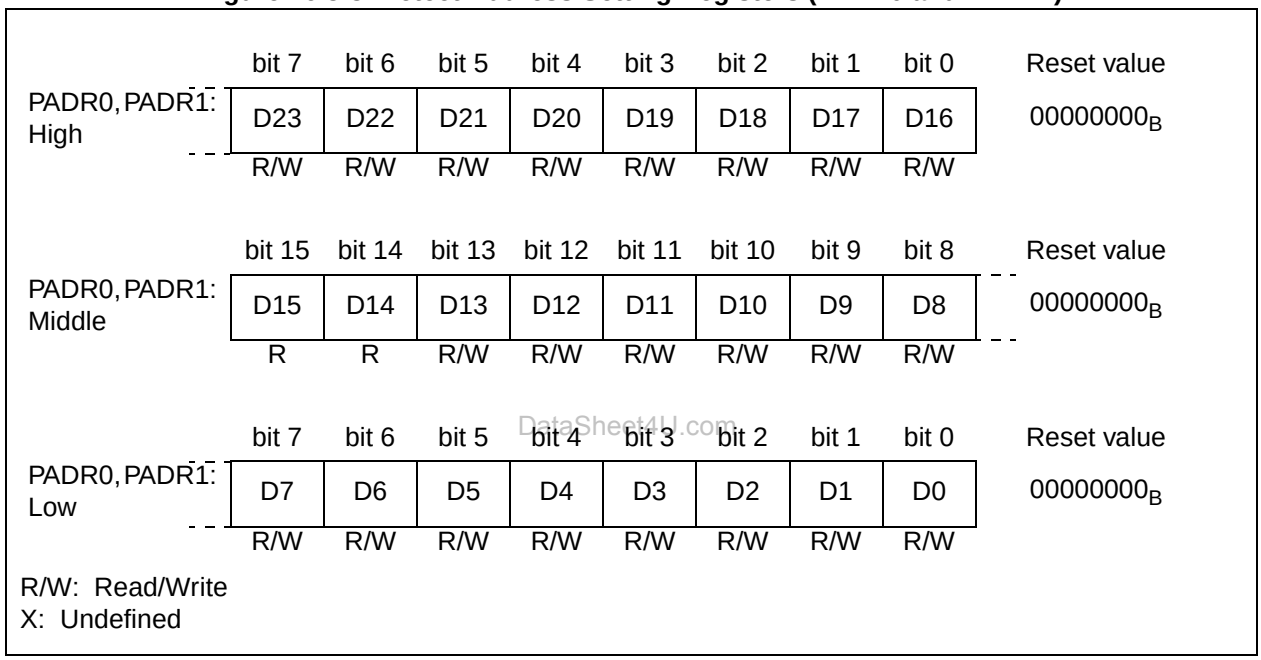
www.DataSheet4U.com

16.3.2 Detect Address Setting Registers (PADR0 and PADR1)

The value of an address to be detected is set in the detect address setting registers. When the address of the instruction processed by the program matches the address set in the detect address setting registers, the next instruction is forcibly replaced by the INT9 instruction, and the interrupt processing program is executed.

■ Detect Address Setting Registers (PADR0 and PADR1)

Figure 16.3-3 Detect Address Setting Registers (PADR0 and PADR1)



DataSheet4U.com

DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

www.DataSheet4U.com

■ Functions of Detect Address Setting Registers

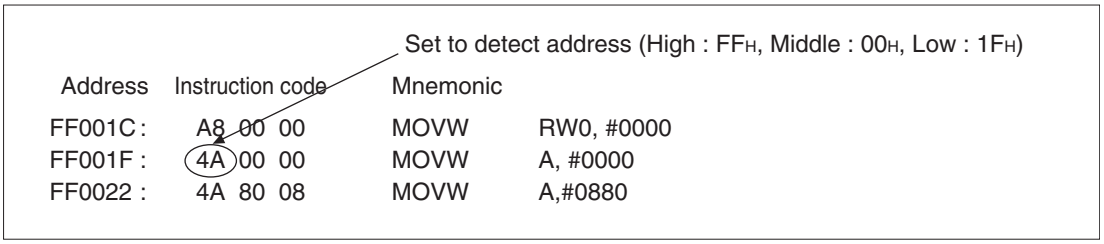
- There are two detect address setting registers (PADR0 and PADR1) that consist of a high byte (bank), middle byte, and low byte, totaling 24 bits.

Table 16.3-2 Address Setting of Detect Address Setting Registers

Register Name	Interrupt Output Enable	Address Setting	
Detect address setting register 0 (PADR0)	PACSR: AD0E	High	Set the upper 8 bits of detect address 0 (bank).
		Middle	Set the middle 8 bits of detect address 0.
		Low	Set the lower 8 bits of detect address 0.
Detect address setting register 1 (PADR1)	PACSR: AD1E	High	Set the upper 8 bits of detect address 1 (bank).
		Middle	Set the middle 8 bits of detect address 1.
		Low	Set the lower 8 bits of detect address 1.

- In the detect address setting registers (PADR0 and PADR1), starting address (first byte) of instruction to be replaced by INT9 instruction should be set.

Figure 16.3-4 Setting of Starting Address of Instruction Code to be Replaced by INT9



Notes:
- When an address of other than the first byte is set to the detect address setting register (PADR0 and PADR1), the instruction code is not replaced by INT9 instruction and a program of an interrupt processing is not be performed. When the address is set to the second byte or subsequent, the address set by the instruction code is replaced by "01" (INT9 instruction code) and, which may cause malfunction.
- The detect address setting registers (PADR0 and PADR1) should be set after disabling the address match detection (PACSR: AD0E = 0 or AD1E = 0) of corresponding address match control registers. If the detect address setting registers are changed without disabling the address match detection, the address match detection function will work immediately after an address match occurs during writing address, which may cause malfunction.
- The address match detection function can be used only for addresses of the internal ROM area. If addresses of the external memory area are set, the address match detection function will not work and the INT9 instruction will not be executed.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

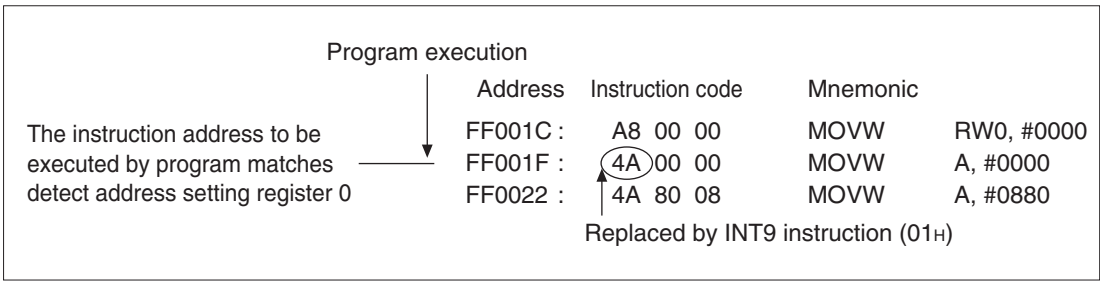
16.4 Explanation of Operation of Address Match Detection Function

If the addresses of the instructions executed in the program match those set in the detection address setting registers (PADR0 and PADR1), the address match detection function will replace the first instruction code with the INT9 instruction (01_H) to branch to the interrupt processing program.

■ Operation of Address Match Detection Function

Figure 16.4-1 shows the operation of the address match detection function when the detect addresses are set and an address match is detected.

Figure 16.4-1 Operation of Address Match Detection Function



■ Setting Detect Address

1. Disable the detection address setting register 0 (PADR0) where the detect address is set for address match detection (PACSR: AD0E = 0).
2. Set the detect address in the detection address setting register 0 (PADR0). Set "FF_H" at the higher bits of the detection address setting register 0 (PADR0), "00_H" at the middle bits, and "$1F_H$" at the lower bits.
3. Enable the detect address setting register 0 (PADR0) where the detect address is set for address match detection (PACSR: AD0E = 1).

■ Program Execution

1. If the address of the instruction to be executed in the program matches the set detect address, the first instruction code at the matched address is replaced by the INT9 instruction code (01_H).
2. INT9 instruction is executed. INT9 interrupt is generated and then interrupt processing program is executed.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

16.4.1 Example of using Address Match Detection Function

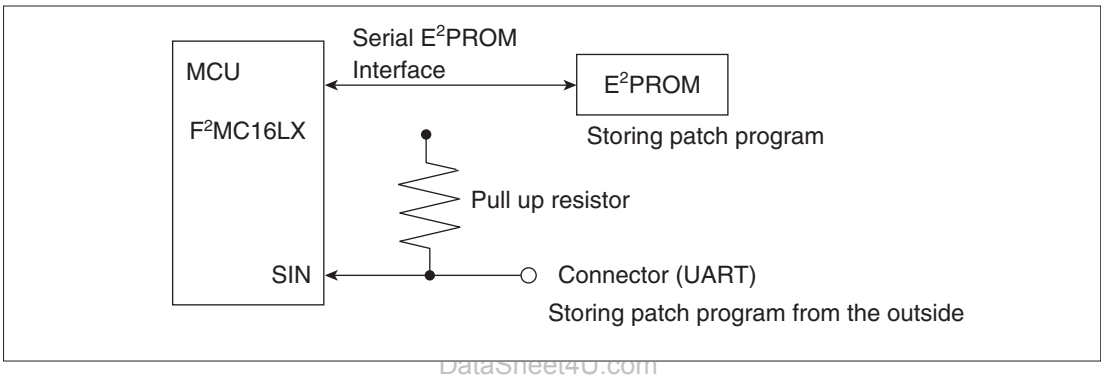
This section gives an example of patch processing for program correction using the address match detection function.

■ System Configuration and E^{2}PROM Memory Map

● System configuration

Figure 16.4-2 gives an example of the system configuration using the address match detection function.

Figure 16.4-2 Example of System Configuration using Address Match Detection Function



DataSheet4U.com
www.DataSheet4U.com

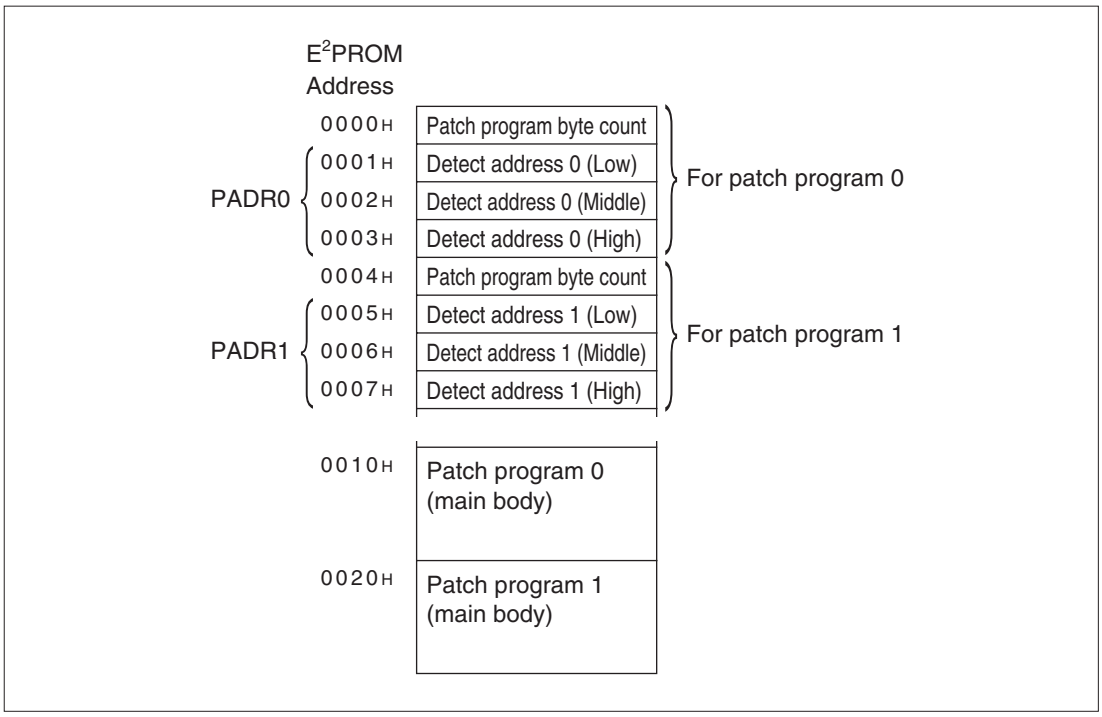
DataSheet4U.com

www.DataSheet4U.com

■ E²PROM Memory Map

Figure 16.4-3 shows the allocation of the patch program and data at storing the patch program in E²PROM.

Figure 16.4-3 Allocation of E²PROM Patch Program and Data



et4U.com

DataShee

● Patch program byte count

The total byte count of the patch program (main body) is stored. If the byte count is "00_H", it indicates that no patch program is provided.

● Detect address (24 bits)

The address where the instruction code is replaced by the INT9 instruction code due to program error is stored. This address is set in the detection address setting registers (PADR0 and PADR1).

● Patch program (main body)

The program executed by the INT9 interrupt processing when the program address matches the detect address is stored. Patch program 0 is allocated from any predetermined address. Patch program 1 is allocated from the address indicating <starting address of patch program 0 + total byte count of patch program 0>.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Setting and Operating State

● Initialization

- E^2PROM data are all cleared to "00_H".

● Occurrence of program error

- By using the connector (UART), information about the patch program is transmitted to the MCU (F^2MC-16LX) from the outside according to the allocation of the E^2PROM patch program and data.
- The MCU (F^2MC-16LX) stores the information received from outside in the E^2PROM.

● Reset sequence

- After reset, the MCU (F^2MC-16LX) reads the byte count of the E^2PROM patch program to check the presence or absence of the correction program.
- If the byte count of the patch program is not "00_H", the higher, middle and lower bits at detect addresses 0 and 1 are read and set in the detection address setting registers 0 and 1 (PADR0 and PADR1). The patch program (main body) is read according to the byte count of the patch program and written to RAM in the MCU (F^2MC-16LX).
- The patch program (main body) is allocated to the address where the patch program is executed in the INT9 interrupt processing by the address match detection function.
- Address match detection is enabled (PACSR: AD0E = 1, AD1E = 1)

et4U.com DataShee DataSheet4U.com

● INT9 Interrupt processing

- Interrupt processing is performed by the INT9 instruction. The MB90385 series has no interrupt request flag by address match detection. Therefore, if the stack information in the program counter is discarded, the detect address cannot be checked. When checking the detect address, check the value of program counter stacked in the interrupt processing routine.
- The patch program is executed, branching to the normal program.

DataSheet4U.com www.DataSheet4U.com

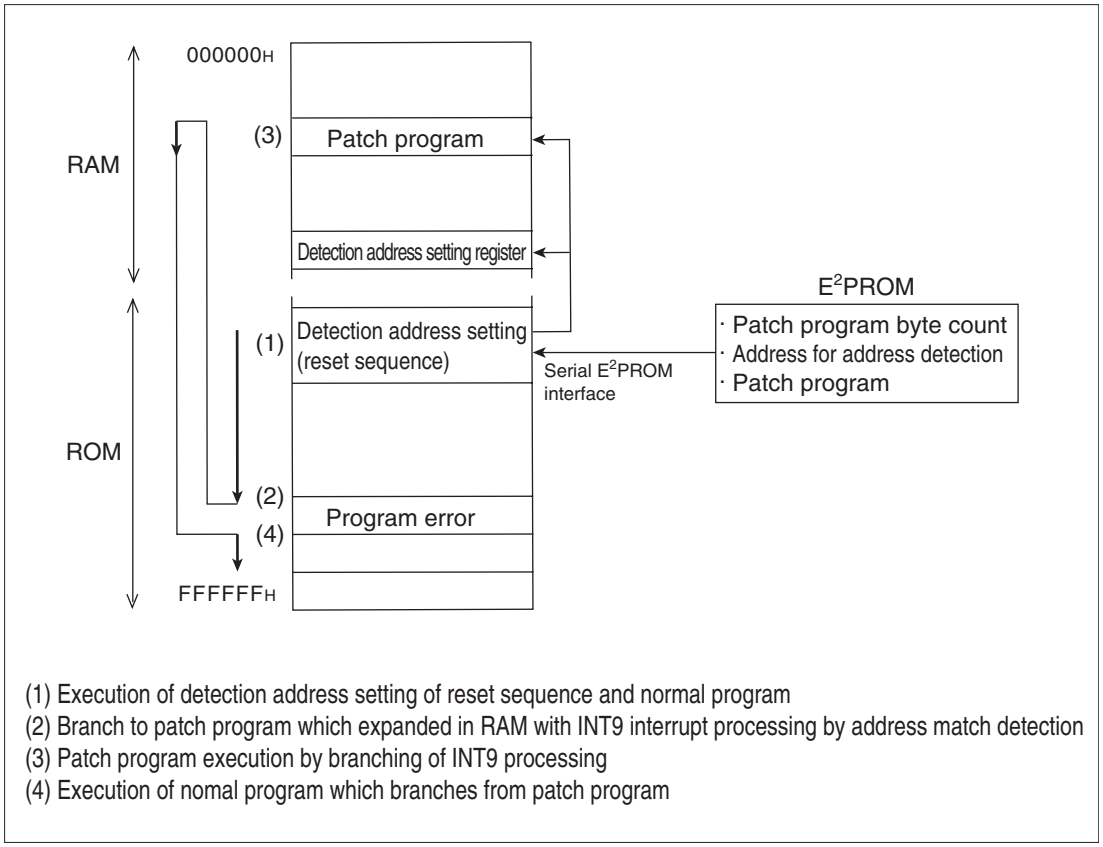
DataSheet4U.com

www.DataSheet4U.com

■ Operation of Address Match Detection Function at Storing Patch Program in E^2PROM

Figure 16.4-4 shows the operation of the address match detection function at storing the patch program in E^2PROM.

Figure 16.4-4 Operation of Address Match Detection Function at Storing Patch Program in E^2PROM



(1) Execution of detection address setting of reset sequence and normal program
(2) Branch to patch program which expanded in RAM with INT9 interrupt processing by address match detection
(3) Patch program execution by branching of INT9 processing
(4) Execution of nomal program which branches from patch program

DataSheet4U.com
www.DataSheet4U.com

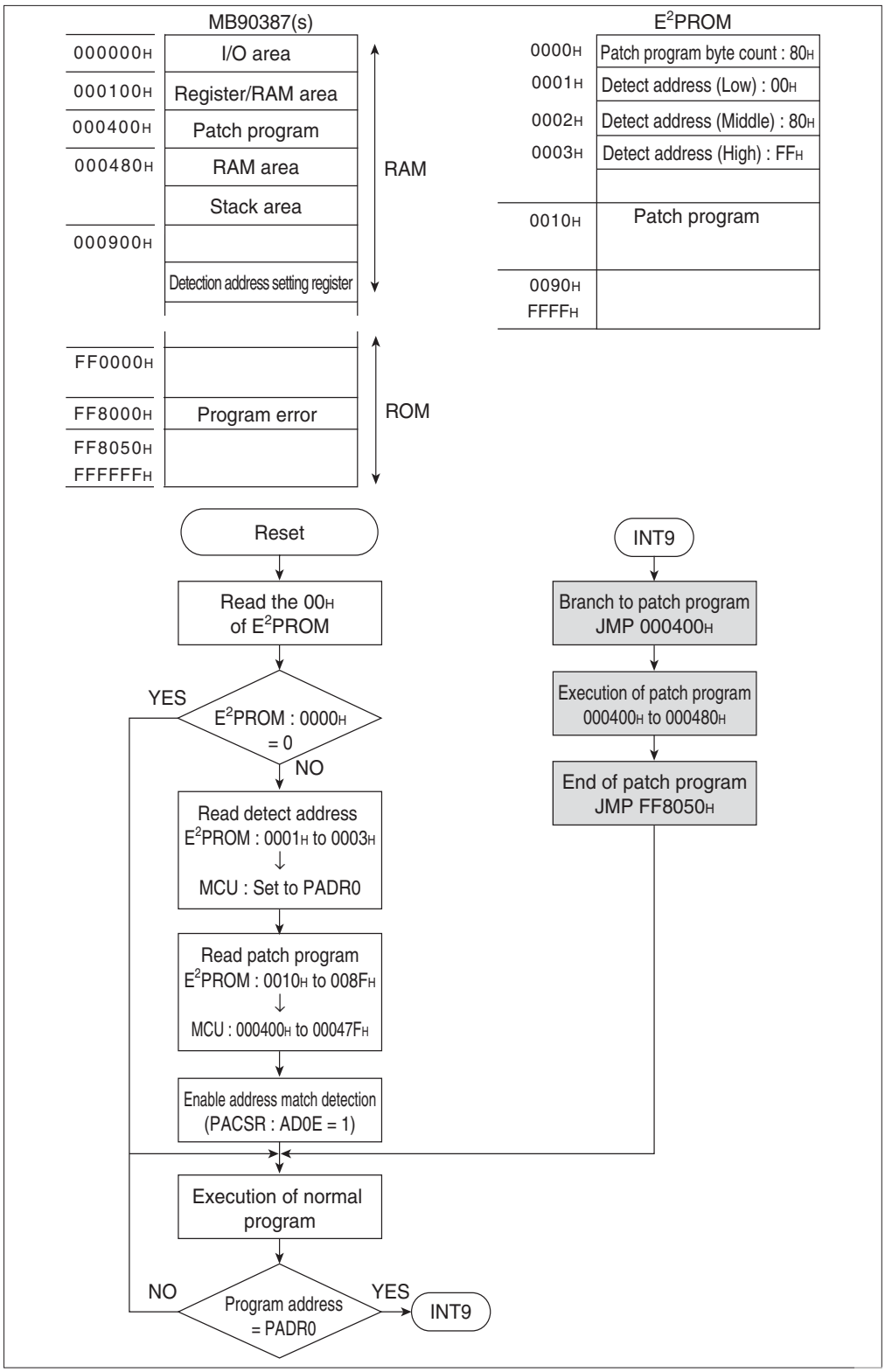
DataSheet4U.com

www.DataSheet4U.com

■ Flow of Patch Processing

Figure 16.4-5 shows the flow of patch processing using the address match detection function.

Figure 16.4-5 Flow of Patch Processing



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

16.5 Program Example of Address Match Detection Function

This section gives a program example for the address match detection function.

■ Program Example for Address Match Detection Function

● Processing specifications

If the address of the instruction to be executed by the program matches the address set in the detection address setting register (PADR0), the INT9 instruction is executed.

● Coding example

```
PACSR   EQU  00009EH            ; Address detection control register
PADRL   EQU  001FF0H            ; Detection address setting register 0 (Low)
PADRM   EQU  001FF1H            ; Detection address setting register 0 (Middle)
PADRH   EQU  001FF2H            ; Detection address setting register 0 (High)
;
;-----Main program-----------------------------------------------------------
CODE    CSEG
START:
                                ; Stack pointer (SP), etc.,
                                ; already reset
        MOV  PADRL,#00H         ; Set address detection register 0 (Low)
        MOV  PADRM,#00H         ; Set address detection register 0 (Middle)
        MOV  PADRH,#00H         ; Set address detection register 0 (High)
;
        MOV  I:PACSR,#00000010B ; Enable address match
         :
        processing by user
         :
LOOP:
         :
        processing by user
         :
        BAR  LOOP
;-----Interrupt program-------------------------------------------------------
WARI:
         :
        processing by user
         :
        BETI                    ; Return from interrupt processing
CODE    ENDS
;-----Vector setting----------------------------------------------------------
VECT    CSEG ABS=0FFH
        ORG  00FFDCH
        DSL  WARI
        ORG  00FFDCH            ; Set reset vector
        DSL  START
        DB   00H                ; Set to single-chip mode
VECT    ENDS
        END  START
```

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 17

ROM MIRRORING FUNCTION SELECT MODULE

This chapter describes the functions and operations of the ROM mirroring function select module.

17.1 Overview of ROM Mirroring Function Select Module

17.2 ROM Mirroring Function Select Register (ROMM)

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

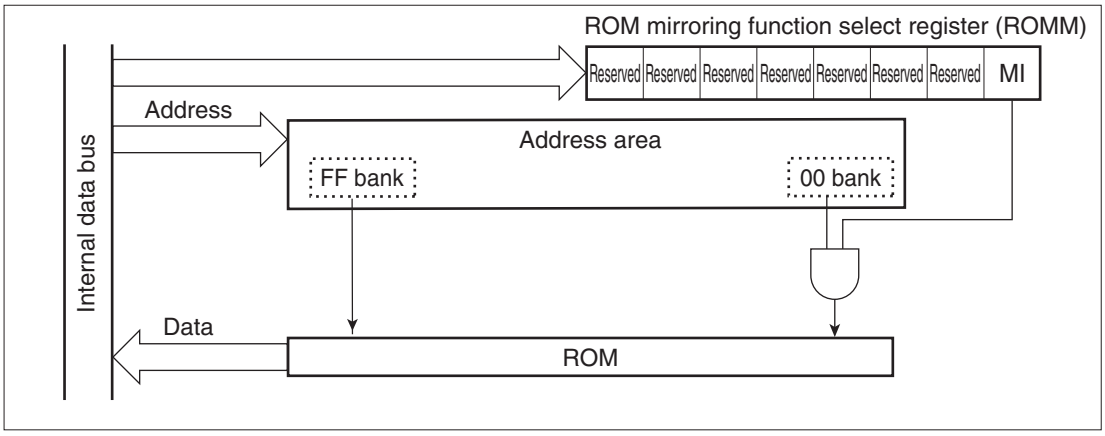
www.DataSheet4U.com

17.1 Overview of ROM Mirroring Function Select Module

The ROM mirroring function select module provides a setting so that ROM data in the FF bank can be read by access to the 00 bank.

■ Block Diagram of ROM Mirroring Function Select Module

Figure 17.1-1 Block Diagram of ROM Mirroring Function Select Module



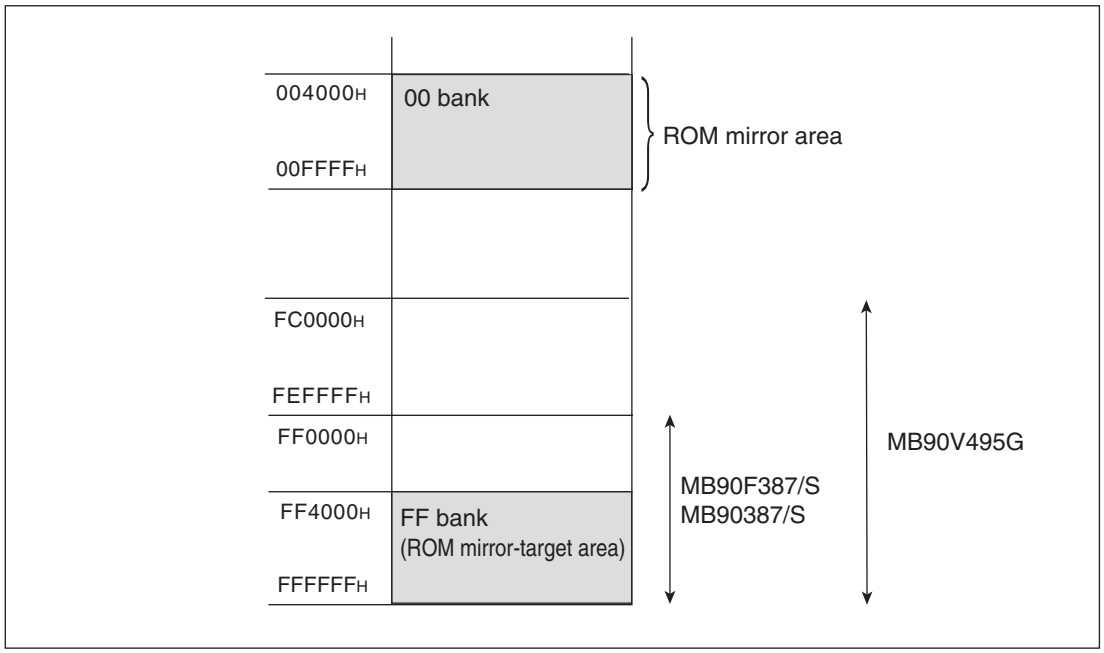
et4U.com
DataShee

■ Access to FF Bank by ROM Mirroring Function

DataSheet4U.com

Figure 17.1-2 shows the location in memory when ROM mirroring function allows access to the 00 bank to read ROM data in the FF bank.

Figure 17.1-2 Access to FF Bank by ROM Mirroring Function



DataSheet4U.com
www.DataSheet4U.com

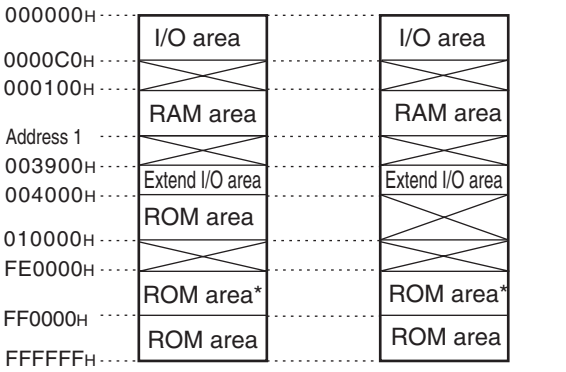
DataSheet4U.com

www.DataSheet4U.com

■ Memory Space when ROM Mirroring Function Enabled/Disabled

Figure 17.1-3 shows the availability of access to memory space when the ROM mirroring function is enabled or disabled.

Figure 17.1-3 Memory Space when ROM Mirroring Function Enabled/Disabled (in Single Chip Mode)



Product type	MB90V495G	MB90F387/S	MB90387/S
Address 1	001900H	000900H	000900H

: Internal access memory

: Access disabled

∗ : When the area from "$FE0000_H$" to "$FEFFFF_H$" of MB90387/S or MB90F387/S is read out, the area "$FF0000_H$" to "$FFFFFF_H$" can be read.

■ List of Registers and Reset Values of ROM Mirroring Function Select Module

Figure 17.1-4 List of Registers and Reset Values of ROM Mirroring Function Select Module

bit	15	14	13	12	11	10	9	8
ROM mirroring function select register (ROMM)	X	X	X	X	X	X	X	1

X: Undefined

DataSheet4U.com
www.DataSheet4U.com

DataSh4U.com

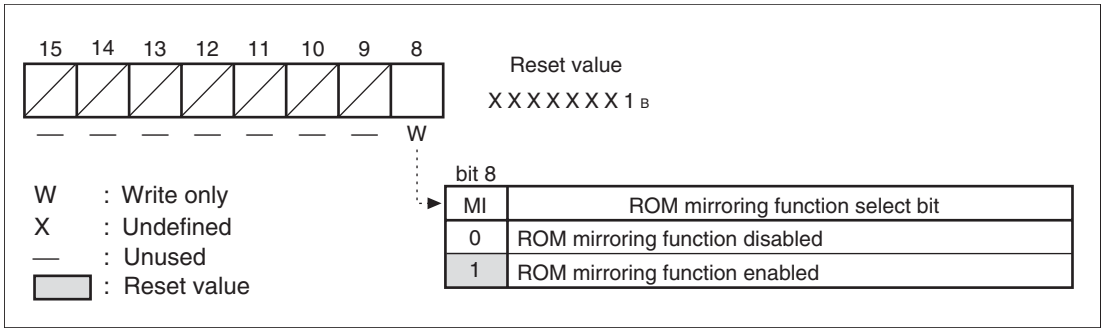
www.DataSheet4U.com

17.2 ROM Mirroring Function Select Register (ROMM)

The ROM mirroring function select register (ROMM) enables or disables the ROM mirroring function. When the ROM mirroring function is enabled, ROM data in the FF bank can be read by access to the 00 bank.

■ ROM Mirror Function Select Register (ROMM)

Figure 17.2-1 ROM Mirroring Function Select Register (ROMM)



et4U.com DataShee

Table 17.2-1 Functions of ROM Mirroring Function Select Register (ROMM)

DataSheet4U.com

Bit Name		Function
bit 8	MI: ROM mirroring function select bit	This bit enables or disables the ROM mirroring function. **When set to 0:** Disables ROM mirroring function **When set to 1:** Enables ROM mirroring function • When the ROM mirroring function is enabled (MI = 1), data at ROM addresses "$FF4000_H$" to "$FFFFFF_H$" can be read by accessing addresses "004000_H" to "$00FFFF_H$"
bit 9 to bit 15	Unused bits	**Read:** Value undefined Be sure to set these bits to 0.

Note: While the ROM area at addresses "004000_H" to "$00FFFF_H$" is being used, access to the ROM mirroring function select register (ROMM) is prohibited.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 18

512 KBIT FLASH MEMORY

This chapter describes the function and operation of 512 Kbit flash memory.

18.1 Overview of 512 Kbit Flash Memory
18.2 Registers and Sector Configuration of Flash Memory
18.3 Flash Memory Control Status Register (FMCS)
18.4 How to Start Automatic Algorithm of Flash Memory
18.5 Check the Execution State of Automatic Algorithm
18.6 Details of Programming/Erasing Flash Memory
18.7 Program Example of 512 Kbit Flash Memory

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.1 Overview of 512 Kbit Flash Memory

There are three ways of programming and erasing flash memory as follows:
1. Programming and erasing using parallel writer
2. Programming and erasing using serial writer
3. Programming and erasing by executing program
This chapter describes the above "3. Programming and Erasing by Executing Program".

■ Overview of 512 Kbit Flash Memory

512 Kbit flash memory is placed in the FF_Hbanks on the CPU memory map. The function of the flash memory I/F circuit provides read access and program access from the CPU to flash memory.

Programming and erasing flash memory are enabled by an instruction from the CPU via the flash memory I/F circuit. This allows reprogramming in the mounted state under CPU control and improvement of programming data efficiency.

■ Features of 512 Kbit Flash Memory

- 128 Kwords x 8 bits/64 Kwords x 16 bits (16 K + 8 K + 8 K + 32 K) sector configuration
- Uses automatic program algorithm (Embedded Algorithm™: the same manner as MBM29LV200
- Erase pause/restart function
- Detects completion of writing/erasing using data polling or toggle bit functions
- Detects completion of writing/erasing by CPU interrupts
- Sector erase function (any combination of sectors)
- Programming/erase count 10,000 (min.)
- Flash read cycle time (min.): 2 machine cycles
- Sector protection function
- Temporary sector protection cancel function
- Extend sector protection function

Embedded Algorithm™ is a registered trademark of Advanced Micro Devices, Inc.

Note: The function for reading the manufacture code and device code is unprovided.
These codes cannot be accessed by any command.

■ Programming and Erasing Flash Memory

- Programming and erasing flash memory cannot be performed at one time.
- Programming or erasing flash memory can be performed by copying the program in flash memory to RAM and executing the program copied in RAM.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.2 Registers and Sector Configuration of Flash Memory

This section explains the registers and the sector configuration of flash memory.

■ List of Registers and Reset Values of Flash Memory

Figure 18.2-1 List of Registers and Reset Values of Flash Memory

	bit	7	6	5	4	3	2	1	0
Flash memory control status register (FMCS)		0	0	0	X	0	0	0	0

X: Undefined

■ Sector Configuration of 512 Kbit Flash Memory

Figure 18.2-2 shows the sector configuration of 512 Kbit flash memory. The upper and lower addresses of each sector are given in the figure.

● Sector configuration

For access from the CPU, the FF bank register has SA0 to SA3.

et4U.com DataShee

Figure 18.2-2 Sector Configuration of 512 Kbit Flash Memory

DataSheet4U.com

Flash memory	CPU address	Writer address*
SA0 (32 Kbytes)	FF0000H	70000H
	FF7FFFH	77FFFH
SA1 (8 Kbytes)	FF8000H	78000H
	FF9FFFH	79FFFH
SA2 (8 Kbytes)	FFA000H	7A000H
	FFBFFFH	7BFFFH
SA3 (16 Kbytes)	FFC000H	7C000H
	FFFFFFH	7FFFFH

*: The writer address is equivalent to the CPU address when data is programmed to flash memory by a parallel writer. This address is where programming and erasing are performed by a general-purpose writer.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

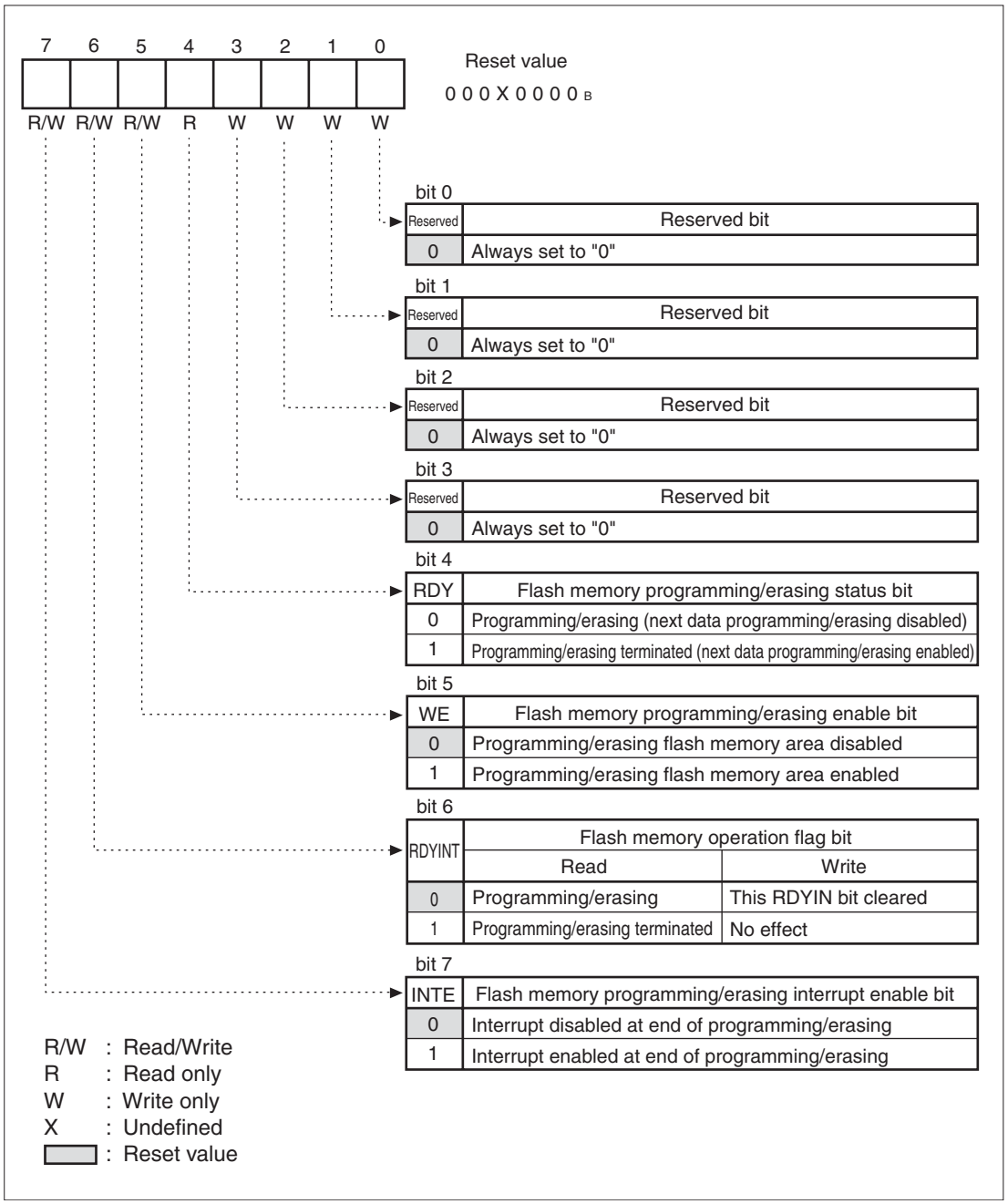
www.DataSheet4U.com

18.3 Flash Memory Control Status Register (FMCS)

The flash memory control status register (FMCS) functions are shown in Figure 18.3-1.

■ Flash Memory Control Status Register (FMCS)

Figure 18.3-1 Flash Memory Control Status Register (FMCS)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 18.3-1 Functions of Flash Memory Control Status Register (FMCS)

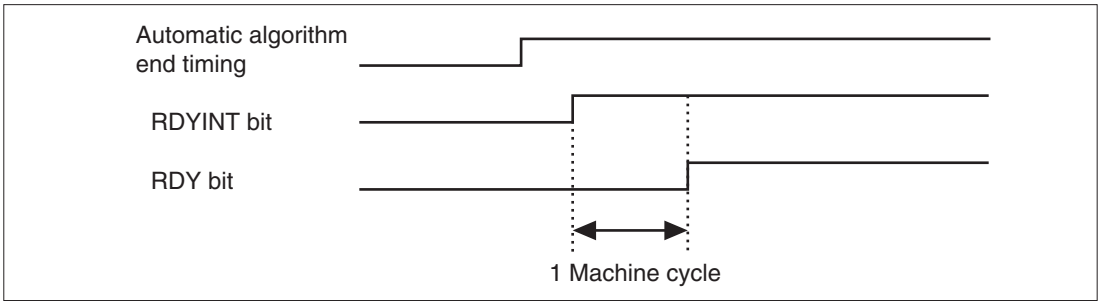
Bit Name		Function
bit 0 to bit 3	Reserved: Reserved bits	Always set these bits to 0.
bit 4	RDY: Flash memory programming/erasing status bit	This bit shows the programming/erasing status of flash memory. • If the RDY bit is 0, programming/erasing flash memory is disabled. • The suspend commands, such as the read/reset command and sector erasing pause, can be accepted even if the RDY bit is 0. The RDY bit is set to 1 when programming/erasing is completed.
bit 5	WE: Flash memory programming/erasing enable bit	This bit enables or disables the programming/erasing of flash memory area. The WE bit should be set before starting the command to program/erase flash memory. **When set to 0:** No program/erase signal is generated even if the command to program/erase the FF bank is input. **When set to 1:** Programming/erasing flash memory is enabled after inputting program/erase command to the FF bank. • When not performing programming/erasing, the WE bit should be set to 0 so as not to accidentally program or erase flash memory.
bit 6	RDYINT: Flash memory operation flag bit	This bit shows the operating state of flash memory. If programming/erasing flash memory is terminated, the RDYINT bit is set to 1 in timing of termination of the automatic flash memory algorithm. • If the RDYINT bit is set to 1 when an interrupt as programming/erasing flash memory is terminated is enabled (FMCS: INTE = 1), an interrupt is requested. • If the RDYINT bit is 0, programming/erasing flash memory is disabled. **When set to 0:** Cleared. **When set to 1:** Unaffected. If the read-modify-write (RMW) instructions are used, 1 is always read.
bit 7	INTE: Flash memory programming/erasing interrupt enable bit	This bit enables or disables an interrupt as programming/erasing flash memory is terminated. **When set to 1:** If the flash memory operation flag bit is set to 1 (FMCS: RDYINT = 1), an interrupt is requested.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Note: The flash memory operation flag bit (RDYINT) and flash memory programming/erasing status bit (RDY) do not change simultaneously. A program should be created so that either RDYINT bit or RDY bit can identify the termination of programming/erasing.



et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.4 How to Start Automatic Algorithm of Flash Memory

There are four commands for starting the automatic algorithm of flash memory: read/reset, write, chip erase. The sector erase command controls suspension and resumption of sector erase.

■ Command Sequence Table

Table 18.4-1 list the commands used in programming/erasing flash memory.

All data is written to command registers by byte access but should be written by word access in the normal mode. Upper data bytes are ignored.

Table 18.4-1 Command Sequence Table

Command Sequence	Bus Write Access	Write Cycle of First Bus		Write Cycle of Second Bus		Write Cycle of Third Bus		Write Cycle of Fourth Bus		Write Cycle of Fifth Bus		Write Cycle of Sixth Bus	
		Address	Data	Address	Data	Address	Data	Address	Data	Address	Data	Address	Data
Read/ Reset*	1	FFXXXX	XXF0	–	–	–	–	–	–	–	–	–	–
Read/ Reset*	4	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XXF0	RA	RD	–	–	–	–
Write program	4	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XXA0	PA (even)	PD (word)	–	–	–	–
Chip erase	6	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XX80	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XX10
Sector erase	6	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XX80	FFAAAA	XXAA	FF5554	XX55	SA (even)	XX30
Sector erase suspend		Input of address"FFXXXX"Data ($xxB0_H$) suspends sector erasing.											
Sector erase resume		Input address"FFXXXX"Data ($xx30_H$) suspends and resume sector erasing.											
Auto Select	3	FFAAAA	XXAA	FF5554	XX55	FFAAAA	XX90	–	–	–	–	–	–

Notes:
- Addresses in the table are the values in the CPU memory map. All addresses and data are hexadecimal values, where "x" is any value.
- RA: Read address
- PA: Program address. Only even addresses can be specified.
- SA: Sector address (See "18.2 Registers and Sector Configuration of Flash Memory")
- RD: Read data
- PD: Program data. Only word data can be specified.

*: Two kinds of read/reset commands can reset flash memory to the read mode.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Auto Select in Table 18.4-2 is the command to check the state of sector protection. The addresses must be set as indicated below together with the command in Table 18.4-2.

Table 18.4-2 Address Setting for Auto Select

	AQ13 to AQ15	AQ7	AQ2	AQ1	AQ0	DQ7 to DQ0
Sector protection	Sector address	L	H	L	L	CODE *

*: The output at the protected sector address is "01_H".
The output at the unprotected sector address is "00_H".

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.5 Check the Execution State of Automatic Algorithm

Since the programming/erasing flow is controlled by the automatic algorithm, hardware sequence can check the internal operating state of flash memory.

■ Hardware Sequence Flags

● Overview of hardware sequence flag

The hardware sequence flag consists of the following 5-bit outputs:

- Data polling flag (DQ7)
- Toggle bit flag (DQ6)
- Timing limit over flag (DQ5)
- Sector erasing timer flag (DQ3)
- Toggle bit 2 flag (DQ2)

These flags can be used to check whether programming, chip and sector erasing, and erase code writing are enabled.

The hardware sequence flags can be referred by setting command sequences and performing read access to the address of a target sector in flash memory. Table 18.5-1 gives the bit allocation of the hardware sequence flags.

DataSheet4U.com

Table 18.5-1 Bit Allocation of Hardware Sequence Flags

Bit No.	7	6	5	4	3	2	1	0
Hardware sequence flag	DQ7	DQ6	DQ5	–	DQ3	DQ2	–	–

- To identify whether automatic programming/chip and sector erasing is in execution or terminated, check the hardware sequence flag or the flash memory programming/erasing status bit in the flash memory control status register (FMCS: RDY). Programming/erasing is terminated, returning to the read/reset state.
- To create a programming/erasing program, use the DQ7, DQ6, DQ5, DQ3 and DQ2 flags to check that automatic programming/erasing is terminated and read data.
- The hardware sequence flags can also be used to check whether the second and later sector erase code writing is enabled.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Explanation of hardware sequence flag

Table 18.5-2 lists the functions of the hardware sequence flag.

Table 18.5-2 List of Hardware Sequence Flag Functions

State		DQ7	DQ6	DQ5	DQ3	DQ2
State change in normal operation	Programming --> Completed (when program address specified)	$\overline{DQ7}$ --> DATA:7	Toggle --> DATA:6	0 --> DATA:5	0 --> DATA:3	1 --> DATA:2
	Chip and sector erasing --> Completed	0 --> 1	Toggle --> Stop	0 --> 1	1	Toggle --> Stop
	Sector erasing wait --> Started	0	Toggle	0	0 --> 1	Toggle
	Erasing --> Sector erasing suspended (Sector being erased)	0 --> 1	Toggle --> 1	0	1 --> 0	Toggle
	Sector erasing suspended --> Resumed (Sector being erased)	1 --> 0	1 --> Toggle	0	0 --> 1	Toggle
	Sector erasing being suspended (Sector not being erased)	DATA:7	DATA:6	DATA:5	DATA:3	DATA:2
Abnormal operation	Programming	$\overline{DQ7}$	Toggle	1	0	1
	Chip and sector erasing	0	Toggle	1	1	*

*: If the DQ5 flag is 1 (timing limit over), the DQ2 flag performs the toggle operation for continuous reading from the programming/erasing sector but does not perform the toggle operation for reading from other sectors.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.5.1 Data Polling Flag (DQ7)

The data polling flag (DQ7) is mainly used to notify that the automatic algorithm is executing or has been completed using the data polling function.

■ Data Polling Flag (DQ7)

Table 18.5-3 and Table 18.5-4 give the state transition of the data polling flag.

Table 18.5-3 State Transition of Data Polling Flag (State Change at Normal Operation)

Operating State	Programming --> Completed	Chip and Sector Erasing --> Completed	Wait for Sector Erasing --> Started	Sector Erasing --> Erasing Suspended (Sector being Erased)	Sector Erasing Suspended --> Resume (Sector being Erased)	Sector Erasing being Suspended (Sector not being Erased)
DQ7	$\overline{DQ7}$ --> DATA:7	0 --> 1	0	0 --> 1	1 --> 0	DATA:7

Table 18.5-4 State Transition of Data Polling Flag (State Change at Abnormal Operation)

Operating State	Programming	Chip and Sector Erasing
DQ7	$\overline{DQ7}$	0

et4U.com DataShee DataSheet4U.com

- At programming
  - Read access during execution of the auto-programming algorithm causes flash memory to output the reversed data of bit 7 last written.
  - Read access at the end of the auto-programming algorithm causes flash memory to output the value of bit 7 at the address to which read access was performed.
- At chip/sector erasing
  - During executing chip and sector erasing algorithms, when read access is made to the currently being erasing sector, bit 7 of flash memory outputs 0. When chip erasing/sector erasing is terminated, bit 7 of flash memory outputs 1.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● At sector erasing suspension

- Read access during sector erasing suspension causes flash memory to output 1 if the address specified by the address signal belongs to the sector being erased. Flash memory outputs bit 7 (DATA: 7) of the read value at the address specified by the signal address if the address specified by the address signal does not belong to the sector being erased.
- Referring this flag together with the toggle bit flag (DQ6) permits a decision on whether flash memory is in the erase suspended state and which sector is being erased.

Note:	Read access to the specified address while the automatic algorithm starts is ignored. Data reading is enabled after 1 is set to the data polling flag (DQ7). Data reading after the end of the automatic algorithm should be performed following read access after completion of data polling has been checked.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.5.2 Toggle Bit Flag (DQ6)

The toggle bit flag (DQ6) is a hardware sequence flag used to notify that the automatic algorithm is being executed or in the end state using the toggle bit function.

■ Toggle Bit Flag (DQ6)

Table 18.5-5 and Table 18.5-6 give the state transition of the toggle bit flag.

Table 18.5-5 State Transition of Toggle Bit Flag (State Change at Normal Operation)

Operating State	Programming --> Completed	Chip and Sector Erasing --> Erasing Completed	Wait for Sector Erasing --> Erasing Started	Sector Erasing --> Erasing Suspended (Sector being Erased)	Sector Erasing Suspended --> Resume (Sector being Erased)	Sector Erasing Suspended (Sector not being Erased)
DQ6	Toggle --> DATA:6	Toggle --> Stop	Toggle	Toggle --> 1	1 --> Toggle	DATA:6

Table 18.5-6 State Transition of Toggle Bit Flag (State Change at Abnormal Operation)

Operating State	Programming	Chip and Sector Erasing
DQ6	Toggle	Toggle

et4U.com DataSheet4U.com DataShee

- At programming and chip/sector erasing

  - If a continuous read access is made during the execution of the automatic algorithm for programming and chip erasing/sector erasing, flash memory toggle-outputs 1 and 0 alternately every reading.
  - If a continuous read access is made after the completion of the automatic algorithm for programming and chip erasing/sector erasing, flash memory outputs bit 6 (DATA: 6) for the read value of the read address every reading.

- At sector erasing suspension

  If a read access is made in the sector erasing suspension state, flash memory outputs 1 when the read address is the sector being erased and bit 6 (DATA: 6) for the read value of the read address when the read address is not the sector being erased.

Reference: If the sector for programming is reprogram-protected, the toggle bit flag (DQ6) produces a toggle output for approximately 2 µs, and then terminates it without reprogramming data.
If all sectors for erasing are reprogram-protected, the toggle bit flag (DQ6) produces a toggle output for approximately 100 µs, and then returns to the read/reset state without reprogramming data.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.5.3 Timing Limit Over Flag (DQ5)

The timing limit over flag (DQ5) is a hardware sequence flag that notifies flash memory that the execution of the automatic algorithm has exceeded a prescribed time (the time required for programming/erasing).

■ Timing Limit Over Flag (DQ5)

Table 18.5-7 and Table 18.5-8 give the state transition of the timing limit over flag.

Table 18.5-7 State Transition of Timing Limit Over Flag (State Change at Normal Operation)

Operating State	Programming --> Completed	Chip and Sector Erasing --> Completed	Wait for Sector Erasing --> Started	Sector Erasing --> Erasing Suspended (Sector being Erased)	Sector Erasing Suspended --> Resume (Sector being Erased)	Sector Erasing being Suspended (Sector not being Erased)
DQ5	0 --> DATA:5	0 --> 1	0	0	0	DATA:5

Table 18.5-8 State Transition of Timing Limit Over Flag (State Change at Abnormal Operation)

Operating State	Programming	Chip and Sector Erasing
DQ5	1	1

et4U.com DataShee DataSheet4U.com

- At programming and chip erasing/sector erasing

  - If a read access made after starting the automatic algorithm for programming or chip erasing/sector erasing is within a prescribed time (the time required for programming/erasing), the timing limit over flag (DQ5) outputs 0. If it exceeds the prescribed time, the timing limit over flag (DQ5) outputs 1.
  - The timing limit over flag (DQ5) can be used to identify the success or failure of programming/erasing, regardless of whether the automatic algorithm is in progress or terminated. If the automatic algorithm by the data polling or the toggle bit function is in execution when the timing limit over flag (DQ5) outputs 1, programming can be identified as a failure.
  - For example, when 1 is set to the flash memory address with 1 set the flash memory, programming fails. In this case, the flash memory will be locked and the automatic algorithm will not complete. Therefore, no valid data is output from the data polling flag (DQ7). Also, the toggle bit flag (DQ6) does not stop the toggle operation and exceeds the time limit, causing the timing limit over flag (DQ5) to output 1. This state means that the flash memory is not being used correctly; it does not mean that the flash memory is faulty. When this state occurs, execute the reset command.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.5.4 Sector Erase Timer Flag (DQ3)

The sector erase timer flag (DQ3) is a hardware sequence flag used to notify during the period of waiting for sector erasing after the sector erase command has started.

■ Sector Erase Timer Flag (DQ3)

Table 18.5-9 and Table 18.5-10 give the state transition of the sector erase timer flag.

Table 18.5-9 State Transition of Sector Erase Timer Flag (State Change at Normal Operation)

Operating State	Programming --> Completed	Chip and Sector Erasing --> Completed	Wait for Sector Erasing --> Started	Sector Erasing --> Erasing Suspended (Sector being Erased)	Sector Erasing Suspended --> Resume (Sector being Erased)	Sector Erasing being Suspended (Sector not being Erased)
DQ3	0 --> DATA:3	1	0 --> 1	1 --> 0	0 --> 1	DATA:3

Table 18.5-10 State Transition of Sector Erase Timer Flag (State Change at Abnormal Operation)

Operating State	Programming	Chip and Sector Erasing
DQ3	0	1

et4U.com DataShee DataSheet4U.com

- At sector erasing
  - If a read access made after starting the sector erase command is within a sector erasing wait period, the sector erasing timer flag (DQ3) outputs 0. If it exceeds the period, the sector erasing timer flag (DQ3) outputs 1.
  - If the sector erasing timer flag (DQ3) is 1, indicating that the automatic algorithm for sector erasing by the data polling or toggle bit function is in progress (DQ = 0; DQ6 produces a toggle output), sector erasing is performed. If any command other than the sector erasing suspension is set, it is ignored until sector erasing is terminated.
  - If the sector erasing timer flag (DQ3) is 0, flash memory can accept the sector erase command. To program the sector erase command, check that the sector erasing timer flag (DQ3) is 0. If the flag is 1, flash memory may not accept the sector erase command of suspending.

- At sector erasing suspension
  - Read access during sector erasing suspension causes flash memory to output 1, if the read address is the sector being erased. Flash memory outputs bit 3 (DATA: 3) for the read value of the read address when the read address is not the sector being erased.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.5.5 Toggle Bit 2 Flag (DQ2)

The toggle bit 2 flag (DQ2) is a hardware sequence flag that notifies flash memory that sector erasing is being suspended using the toggle bit function.

■ Toggle Bit Flag (DQ2)

Table 18.5-11 and Table 18.5-12 give the state transition of the toggle bit flag.

Table 18.5-11 State Transition of Toggle Bit Flag (State Change at Normal Operation)

Operating State	Programming --> Completed	Chip and Sector Erasing --> Completed	Wait for Sector Erasing --> Started	Sector Erasing --> Erasing Suspended (Sector being Erased)	Sector Erasing Suspended --> Resume Sector being Erased)	Sector Erasing being Suspended (Sector not being Erased)
DQ2	1 --> DATA:2	Toggle --> Stop	Toggle	Toggle	Toggle	DATA:2

Table 18.5-12 State Transition of Toggle Bit Flag (State Change at Abnormal Operation)

Operating State	Programming	Chip and Sector Erasing
DQ2	1	*

*: If the DQ5 flag is 1 (timing limit over), the DQ2 flag performs the toggle operation for continuous reading from the programming/erasing sector but does not perform the toggle operation for reading from other sectors.

- At sector erasing
  - If a continuous read access is made during the execution of the automatic algorithm for chip erasing/ sector erasing, flash memory toggle-outputs 1 and 0 alternately every reading.
  - If a continuous read access is made after the completion of the algorithm for chip erasing/sector erasing, flash memory outputs bit 2 (DATA: 2) for the read value of the read address every reading.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● At sector erasing suspension

- If a read access is made in the sector erasing suspension state, flash memory outputs 1 and 0 alternately when the read address is the sector being erased and bit 2 (DATA: 2) for the read value of the read address when the read address is not the sector being erased.
- If programming is performed in the sector erasing suspension state, flash memory outputs 1 when a continuous read access is started with the sector that is not in the erasing suspension state.
- The toggle bit 2 flag (DQ2) is used together with the toggle bit flag (DQ6) to detect that sector erasing is suspended (the DQ2 flag performs the toggle operation but the DQ6 flag does not).
- If a read access from the sector being erased is made, the toggle bit 2 flag (DQ2) performs the toggle operation, so it can also be used to detect the sector being erased.

Reference:	If all sectors for erasing are reprogram-protected, the toggle bit flag (DQ2) produces a toggle output for approximately 100 µs, and then returns to the read/reset state without reprogramming data.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.6 Details of Programming/Erasing Flash Memory

This section explains the procedure for inputting commands starting the automatic algorithm, and for read/reset of flash memory, programming, chip erasing, sector erasing, sector erasing suspension and sector erasing resumption.

■ Detailed Explanation of Programming and Erasing Flash Memory

Automatic algorithm can be started by programming the command sequence of read/reset, programming, chip erasing, sector erasing, sector erasing suspension and erasing resumption from CPU to flash memory. Programming flash memory from the CPU should always be performed continuously. The termination of the automatic algorithm can be checked by the data polling function. After normal termination, it returns to the read/reset state.

Each operation is explained in the following order.

- Read/reset state
- Data programming
- All data erasing (chip all erase)
- Any data erasing (sector erase)
- Sector erasing suspension
- Sector erasing resumption

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.6.1 Read/Reset State in Flash Memory

This section explains the procedure for inputting the read/reset command to place flash memory in the read/reset state.

■ Read/Reset State in Flash Memory

- Flash memory can be placed in the read/reset state by continuously transmitting the read/reset command in the command sequence table from CPU to flash memory.
- There are two kinds of read/reset commands: one is executed at one time bus operation, and the other is executed at three times bus operations; the command sequence of both is essentially the same.
- Since the read/reset state is the initial state for flash memory, flash memory always enters this state after power-on and at the normal termination of command. The read/reset state is also described as the wait state for command input.
- In the read/reset state, a read access to flash memory enables data to be read. As is the case with mask ROM, a program access from the CPU can be made. A read access to flash memory does not require the read/reset command. If the command is not terminated normally, use the read/reset command to initialize the automatic algorithm.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.6.2 Data Programming to Flash Memory

This section explains the procedure for inputting the program command to program data to flash memory.

■ Data Programming to Flash Memory

- In order to start the data programming automatic algorithm, continuously transmit the program command in the command sequence table from CPU to flash memory.
- At completion of data programming to a target address in the fourth cycle, the automatic algorithm starts automatic programming.

● How to specify address

- The only even addresses can be specified for the programming address specified by programming data cycle. Specifying odd addresses prevents correct writing. Writing to even addresses must be performed in word data units.
- Programming is possible in any address order or even beyond sector boundaries. However, execution of one programming command, permits programming of only one word for data.

et4U.com

● Notes on data programming

DataShee

- Data 0 cannot be returned to data 1 by programming. When data 0 is programmed to data 1, the data polling algorithm (DQ7) or toggling (DQ6) is not terminated and the flash memory is considered faulty; the timing limit over flag (DQ5) is determined as an error.
- When data is read in the read/reset state, the bit data remains 0. To return the bit data to 1 from 0, erase flash memory data.
- All commands are ignored during automatic programming. If a hardware reset occurs during programming, data being programmed to addresses are not assured.

■ Data Programming Procedure

- Figure 18.6-1 gives an example of the procedure for programming data into flash memory. The hardware sequence flags can be used to check the operating state of the automatic algorithm in flash memory. The data polling flag (DQ7) is used for checking the completion of programming to flash memory in this example.
- Flag check data should be read from the address where data was last written.
- Because the data polling flag (DQ7) and the timing limit over flag (DQ5) change at the same time, the data polling flag (DQ7) must be checked even when the timing limit over flag (DQ5) is 1.
- Similarly, since the toggle bit flag (DQ6) stops toggling at the same time the timing limit over flag (DQ5) changes to 1, toggle bit flag (DQ6) must be checked.

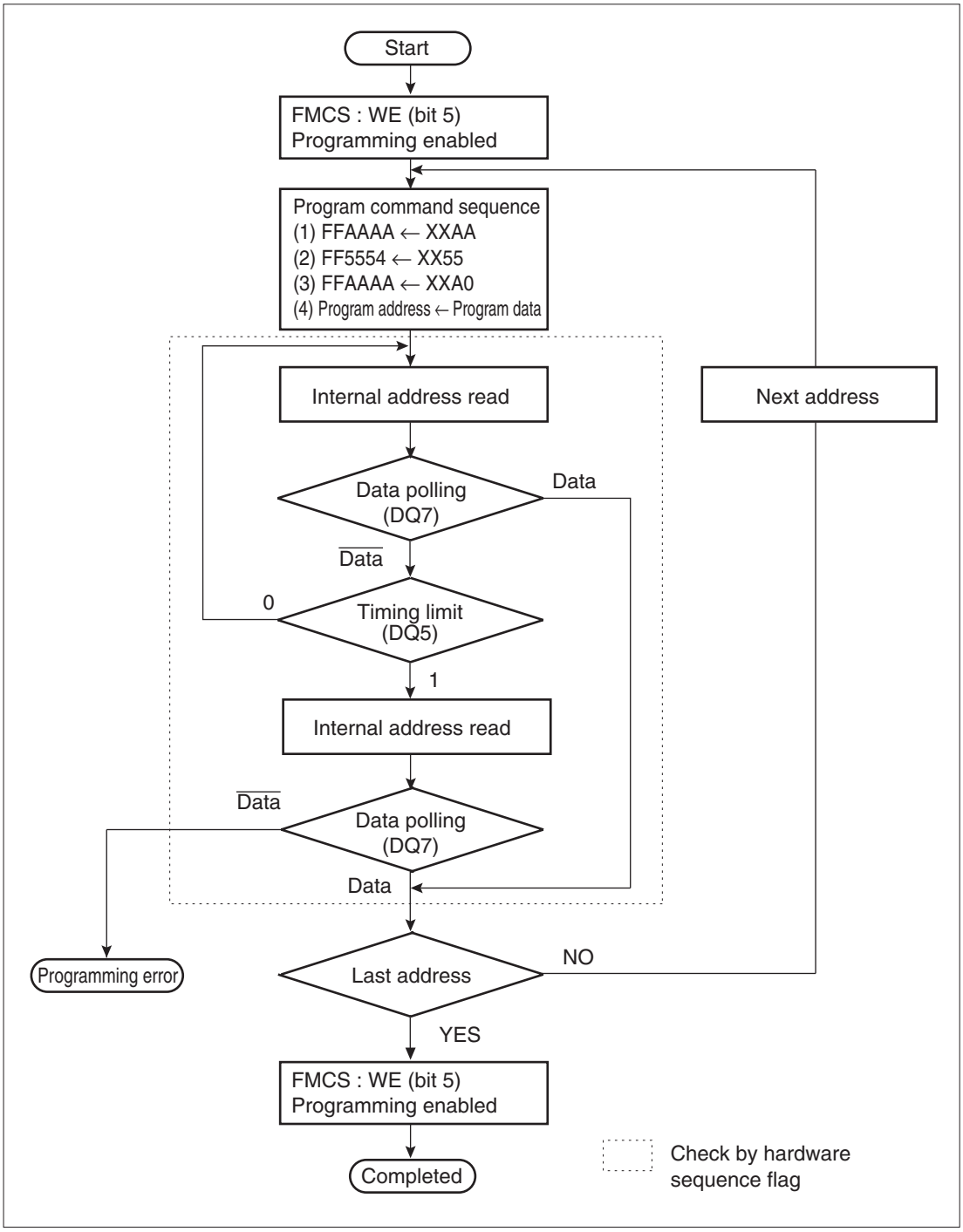
DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 18.6-1 Example of Data Programming Procedure



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.6.3 Data Erase from Flash Memory (Chip Erase)

This section explains the procedure for inputting the chip erase command to erase all data from flash memory.

■ All Data Erase from Flash Memory (Chip Erase)

- All data can be erased from flash memory by continuously transmitting the chip erase command in the command sequence table from CPU to flash memory.
- The chip erase command is executed in six bus operations. Chip erasing is started at completion of the sixth programming cycle.
- Before chip erasing, the user need not perform programming to flash memory. During execution of the automatic erasing algorithm, flash memory automatically programs 0 before erasing all cells.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.6.4 Erasing Any Data in Flash Memory (Sector Erasing)

This section explains the procedure for inputting the sector erase command to erase any data in flash memory. Sector-by-sector erasing is enabled and multiple sectors can be specified at a time.

■ Erasing Any Data in Flash Memory (Sector Erasing)

Any sector in flash memory can be erased by continuously transmitting the sector erase command in the command sequence table from CPU to flash memory.

● How to specify sector

- The sector erase command is executed in six bus operations. By setting the address on the sixth cycle in the even address in the target sector and programming the sector erase code (30_H) to data, a 50 µs sector erasing wait is started
- When erasing more than one sector, the sector erase code (30_H) is programmed to the sector address to be erased, following the above.

● Notes on specifying multiple sectors

et4U.com DataShee

- Sector erasing is started after a 50 µs period waiting for sector erasing is completed after the last sector erase code has been programmed.

DataSheet4U.com

- That is, when erasing more than one sector simultaneously, the address of erase sector and the sector erase code must be input within 50 µs. If the sector erase code is input 50 µs or later, it cannot be accepted.
- Whether continuous programming of the sector erase code is enabled can be checked by the sector erase timer flag (DQ3).
- In this case, the address from which the sector erase timer is flag (DQ3) read should correspond to the sector to be erased.

■ Erasing Procedure for Flash Memory Sectors

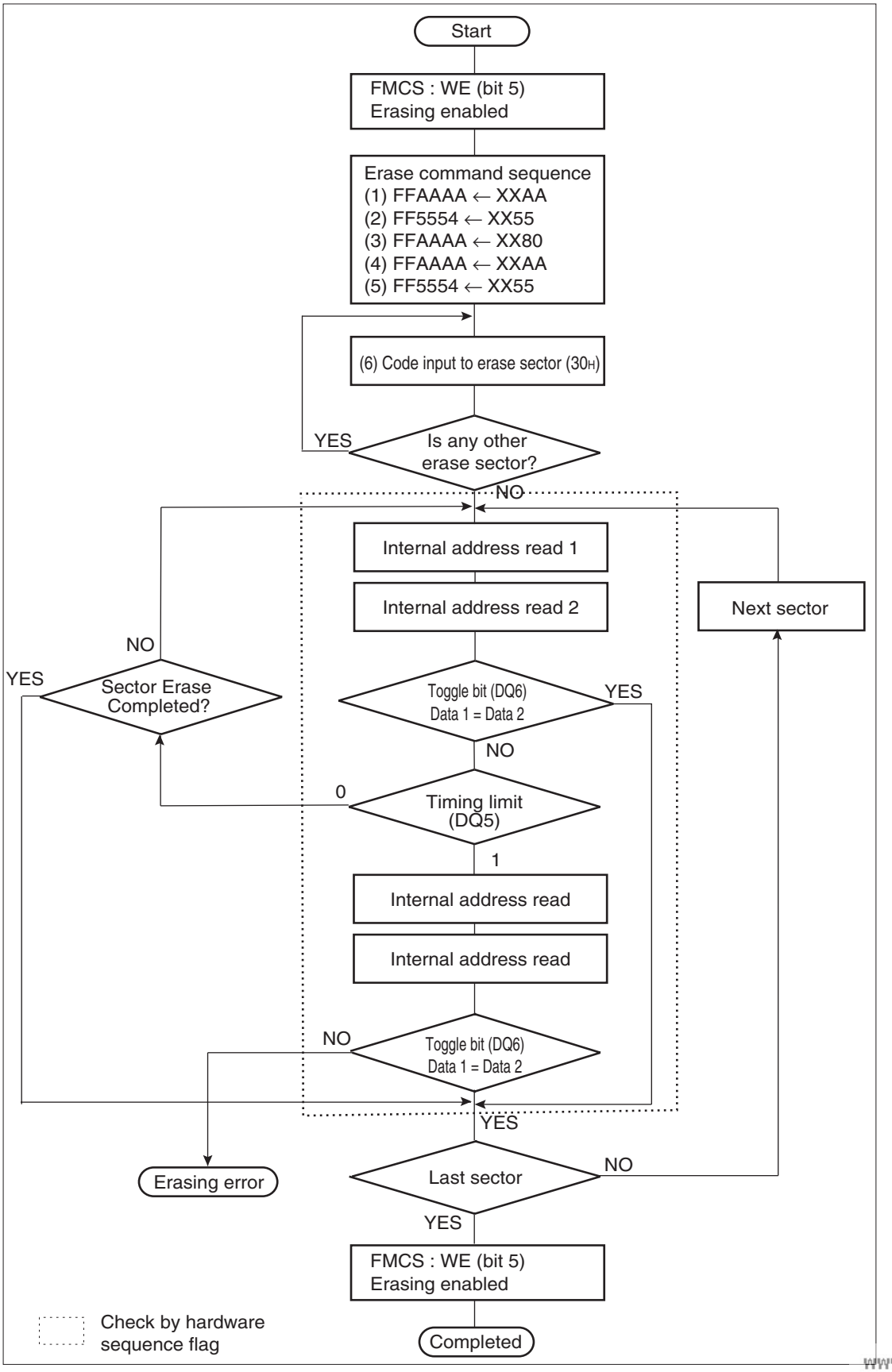
- The state of the automatic algorithm in the flash memory can be determined using the hardware sequence flag. Figure 18.6-2 gives an example of the flash memory sector erase procedure. In this example, the toggle bit flag (DQ6) is used to check that erase ends.
- DQ6 terminates toggling concurrently with the change of the timing limit over flag (DQ5) to 1, so the DQ6 must be checked even when DQ5 is 1.
- Similarly, the data polling flag (DQ7) changes concurrently with the transition of the DQ5, so DQ7 must be checked.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure 18.6-2 Example of Sector Erasing Procedure



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.6.5 Sector Erase Suspension in Flash Memory

This section explains the procedure for inputting the sector erase suspend command to suspend sector erasing. Data can be read from the sector not being erased.

■ Sector Erase Suspension in Flash Memory

- To cause flash memory sector erasing to suspend, continuously transmit the sector erasing suspend command in the command sequence table from CPU to flash memory.
- The sector erasing suspend command suspends the sector erase currently being performed, enabling data read from a sector that is currently not being erased. Only read can be performed when this command is suspended; programming cannot be performed.
- This command is only enabled during the sector erasing period including the erasing wait time; it is ignored during the chip erasing period or during programming.
- The sector erasing suspend command is executed when the sector erasing suspend code ($B0_H$) is programmed. Arbitrary address in flash memory should be set for address. If the sector erasing suspend command is executed during sector erasing pause, the command input again is ignored.
- When the sector erasing suspend command is input during the sector erasing wait period, the sector erase wait state ends immediately, the erasing is interrupted, and the erase stop state occurs.
- When the erase suspend command is input during the sector erasing after the sector erase wait period, the erase suspend state occurs after 15 μs max.

et4U.com DataShee DataSheet4U.com

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.6.6 Sector Erase Resumption in Flash Memory

This section explains the procedure for inputting the sector erase resume command to resume erasing of the suspended flash memory sector.

■ Erase Resumption in Flash Memory

- Suspended sector erasing can be resumed by continuously transmitting the sector erase resume command in the command sequence table from CPU to flash memory.
- The sector erase resume command resumes sector erasing suspended by the sector erase suspend command. This command is executed by writing the erase resume code (30_H). In this case, any address in the flash memory area is specified.
- Inputting the sector erase resume command during sector erasing is ignored.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

18.7 Program Example of 512 Kbit Flash Memory

A program example of the 512 Kbit flash memory is given below.

■ Program Example of 512 Kbit Flash Memory

```
                NAME    FLASHWE
                TITLE   FLASHWE
;-------------------------------------------------------------------------------
; 512 Kbit FLASH Sample Program
; 1: Transfer program in FLASH (address FFBC00 H,
;    sector SA2) to RAM (address 000700H).
; 2: Execute program on RAM.
; 3: Program PDR1 value to FLASH (address FF0000H, sector SA0).
; 4: Read programmed value (address FF0000H, sector SA0) and output to PDR2.
; 5: Erase programmed sector (SA0).
; 6: Output check that data is erased.
; Conditions
;    - Count of bytes transferred to RAM:  100H (256 bytes)
;    - Completion of programming and erasing checked by:
;            Timing limit over flag (DQ5)
;            Toggle bit flag (DQ6)
;            RDY (FMCS)
;    -  Action taken at error
;            Output H to P00 to P07.
;            Issue reset command.
;-------------------------------------------------------------------------------
;
RESOUS  IOSEG ABS=00           ; Definition of "RESOUS" I/O segment
        ORG   0000H
PDR0    RB    1
PDR1    RB    1
PDR2    RB    1
PDR3    RB    1
        ORG   0010H
DDR0    RB    1
DDR1    RB    1
DDR2    RB    1
DDR3    RB    1
        ORG   00A1H
CKSCR   RB    1
        ORG   00AEH
FMCS    RB    1
        ORG   006FH
ROMM    RB    1
RESOUS  ENDS
;
```

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com


```
SSTA    SSEG
        RW    0127H
STA_T   RW    1
SSTA    ENDS
;
DATA    DSEG  ABS=0FFH          ; FLASH command address
        ORG   5554H
COMADR2 ORG   1
        ORG   0AAAAH
COMADR1 ORG   1
DATA    ENDS
;-------------------------------------------------------------------------------
; Main program (SA1)
;-------------------------------------------------------------------------------
CODE    CSEG
START:
        ;-----------------------------------------------------------------------
        ; Initialize
        ;-----------------------------------------------------------------------
        MOV   CKSCR,#0BAH       ; Set to 3-multiplying count
        MOV   RP,#0
        MOV   A,#!STA_T
        MOV   SSB,A
        MOVW  A,#STA_T
        MOVW  SP, A
        MOV   ROMM,#00H         ; Mirror OFF
        MOV   PDR0,#00H         ; For error check
        MOV   DDR0,#0FFH
        MOV   PDR1,#00H         ; Data input port
        MOV   DDR1,#00H
        MOV   PDR2,#00H         ; Data output port
        MOV   DDR2,#0FFH
        ;-----------------------------------------------------------------------
        ; Transfer FLASH programming/erasing program (FFBC00H) to RAM
        ; (address 700H)
        ;-----------------------------------------------------------------------
        MOVW  A,#0700H          ; Transfer destination RAM area
        MOVW  A,#0BC00H         ; Transfer source address
                                ; (position where program exist)
        MOVW  RW0,#100H         ; Count of bytes to be transferred
        MOVS  ADB,PCB           ; Transfer 100H from FFBC00H to 000700H
        CALLP 000700H           ; Jump to address where transferred program exists

        ;-----------------------------------------------------------------------
        ; Data output
        ;-----------------------------------------------------------------------
OUT     MOV   A,#0FFH
        MOV   ADB,A
        MOVW  RW2,#0000H
        MOVW  A,@RW2+00
        MOV   PDR2,A
END     JMP   *
CODE    ENDS
```

et4U.com DataShee
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com


```
;-------------------------------------------------------------------------------
Flash programming/erasing program (SA2)
;-------------------------------------------------------------------------------
RAMPRG  CSEG  ABS=0FFH
        ORG   0BC00H
        ;-----------------------------------------------------------------------
        ; Initialize
        ;-----------------------------------------------------------------------
        MOVW  RW0,#0500H       ; RW0: RAM space for storage of input data
                               ; 00:0500 to
        MOVW  RW2,#0000H       ; RW2: Flash memory programming address
                               ; FD:0000 to
        MOV   A,#00H           ; DTB change
        MOV   DTB,A            ; Specify bank for @RW0
        MOV   A,#0FFH          ; ADB change 1
        MOV   ADB,A            ; Specify bank for program mode specifying address
        MOV   PDR3,#00H        ; Initialize switch
        MOV   DDR3,#00H
;
WAIT1   BBC   PDR3:0,WAIT1     ; PDR3: 0  with High level, start programming
;
;-------------------------------------------------------------------------------
; Program (SA0)
;-------------------------------------------------------------------------------
        MOV   A,PDR1
        MOVW  @RW0+00,A        ; Save PDR1 data in RAM.
        MOV   FMCS,#20H        ; Set program mode.
        MOVW  ADB:COMADR1, #00AAH
                               ; Flash program command 1
        MOVW  ADB:COMADR2, #0055H
                               ; Flash program command 2
        MOVW  ADB:COMADR1, #00A0H
                              ; Flash program command 3
;
        MOVW  A, @RW0+00       ; Program input data (RW0) to flash memory (RW2).
;
        MOVW  @RW2+00, A
WRITE   ; Waiting time check
        ;-----------------------------------------------------------------------
        ;ERROR occurs when the time limit over check flag is set and toggling.
        ;-----------------------------------------------------------------------
        MOVW  A,@RW2+00
        AND   A,#20H           ; DQ5 time limit check
        BZ    NTOW             ; Time limit over
        MOVW  A,@RW2+00        ; AH
        MOVW  A,@RW2+00        ; AL
        XORW  A                ; XOR of AH and AL (1 if value is invalid)
        AND   A,#40H           ; Is DQ6 toggle bit?
        BNZ   ERROR            ; If no, go to ERROR.
        ;-----------------------------------------------------------------------
        ;Programming end check (FMCS-RDY)
        ;-----------------------------------------------------------------------
NTOW    MOVW  A,FMCS
        AND   A,#10H           ; Extract RDY bit (bit 4) of FMCS.
        BZ    WRITE            ; Is programming ended?
        MOV   FMCS,#00H        ; Cancel program mode.
```

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com


```
        ;-------------------------------------------------------------------------
        ;Program data output
        ;-------------------------------------------------------------------------
        MOVW  RW2,#0000H       ; Output program data
        MOVW  A, @RW2+00
        MOV   PDR2,A
;
WAIT2   BBC   PDR3:1,WAIT2     ; PDR3:1 With "H", start sector erasing.
;
;-----------------------------------------------------------------------------
; Sector erasing (SA0)
;-----------------------------------------------------------------------------
        MOV   @RW2+00,#0000H     ; Initialize address
        MOV   FMCS,#20H          ; Set erase mode
        MOVW  ADB:COMADR1,#00AAH ; Flash command 1
        MOVW  ADB:COMADR2,#0055H ; Flash command 2
        MOVW  ADB:COMADR1,#0080H ; Flash command 3
        MOVW  ADB:COMADR1,#00AAH ; Flash command 4
        MOVW  ADB:COMADR2,#0055H ; Flash command 5
        MOV   @RW2+00,#0030H     ; Issue erase command to sector to be erased 6.
ELS     ; Waiting time check
        ;-------------------------------------------------------------------------
        ;ERROR occurs when time limit over check flag is set and toggling is underway.
;-----------------------------------------------------------------------------
        MOVW  A,@RW2+00
        AND   A,#20H           ; DQ5 time limit check
        BZ    NOTE             ; Time limit over
        MOVW  A,@RW2+00        ; During AH programming, "H/L" is output
        MOVW  A,@RW2+00        ; alternately every time AL is read from DQ6
        XORW  A                ; XOR of AH and AL (1 if DQ6 value invalid,
                               ;  indicating programming underway)
        AND   A, #40H          ; Is DQ6 toggle bit "H"?
        BNZ   ERROR            ; If "H", go to ERROR
        ;-------------------------------------------------------------------------
        ;Erasing end check (FMCS-RDY)
        ;-------------------------------------------------------------------------
NTOE    MOVW  A,FMCS           ;
        AND   A,#10H           ; Extract RDY bit (bit 4) of FMCS
        BZ    ELS              ; Is erasing ended?
        MOV   FMCS,#00H        ; Cancel FLASH erase mode
        RETP                   ; Return to main program
;-----------------------------------------------------------------------------
;Error
;-----------------------------------------------------------------------------
ERROR   MOV   ADB:COMADR1,#0F0H ; Reset command (read enabled)
        MOV   FMCS,#00H         ; Cancel FLASH mode
        MOV   PDR0,#0FFH        ; Check error processing
        RETP                    ; Return to main program
RAMPRG  ENDS
;-------------------------------------------------------------------
VECT    CSEG  ABS=0FFH
        ORG   0FFDCH
        DSL   START
        DB    00H
VECT    ENDS
;
        ENDS  START
```

et4U.com DataShee

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

CHAPTER 19

FLASH SERIAL PROGRAMMING CONNECTION

This chapter describes an example of serial programming connection using the flash microcontroller programmer made by Yokogawa Digital Computer Corporation.

DataSheet4U.com

19.1 Basic Configuration of Serial Programming Connection for F^2MC-16LX MB90F387/S

19.2 Connection Example in Single-chip Mode (User Power Supply)

19.3 Connection Example in Single-chip Mode (Writer Power Supply)

19.4 Example of Minimum Connection to Flash Microcontroller Programmer (User Power Supply)

19.5 Example of Minimum Connection to Flash Microcontroller Programmer (Writer Power Supply)

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

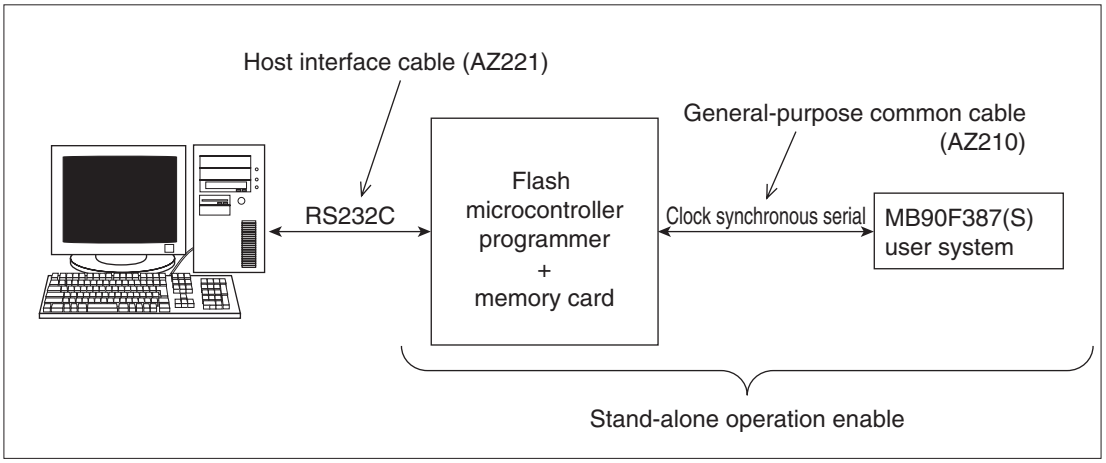
19.1 Basic Configuration of Serial Programming Connection for F²MC-16LX MB90F387/S

The MB90F387/S supports the serial on-board programming of flash ROM (Fujitsu standard). The specification for serial on-board programming are explained below.

■ Basic Configuration of Serial Programming Connection for MB90F387/S

The flash microcontroller programmer made by Yokogawa Digital Computer Corporation is used for Fujitsu standard serial on-board programming.

Figure 19.1-1 Basic Configuration of Serial Programming Connection



Note:	Inquire of Yokogawa Digital Computer Corporation for details about the functions and operations of the AF220/AF210/AF120/AF110 flash microcontroller programmer, general-purpose common cable for connection (AZ210), and connectors

Table 19.1-1 Pins Used for Fujitsu Standard Serial On-board Programming (1/2)

Pin	Function	Supplementary Information
MD2, MD1, MD0	Mode pins	Writing 1 to MD2, 1 to MD1 and 0 to MD0 sets the flash serial program mode.
X0, X1	Oscillation pins	In the flash serial program mode, the internal operating clock of the CPU has a frequency one time that of the PLL clock, so the internal operating clock frequency is the same as the oscillation clock frequency. Since the oscillation clock frequency serves as an internal operation clock, the oscillator used for serial programming have frequencies from 1 MHz to 16 MHz.
P30, P31	Programming program start pins	Input a Low level to P30 and a High level to P31.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table 19.1-1 Pins Used for Fujitsu Standard Serial On-board Programming (2/2)

Pin	Function	Supplementary Information
$\overline{RST}$	Reset pin	–
SIN1	Serial data input pin	UART is used in clock synchronous mode.
SOT1	Serial data output pin	
SCK1	Serial clock input pin	
C	C pin	This pin is a capacitance pin for stabilizing voltage. Connect the ceramic capacitor approx. 0.1 µF externally.
V_{CC}	Power supply voltage pin	If the program voltage (5 V 10%) is supplied from the user system, the flash microcontroller programmer need not be connected.
V_{SS}	GND pin	GND pin is common to the ground of the flash microcontroller programmer.

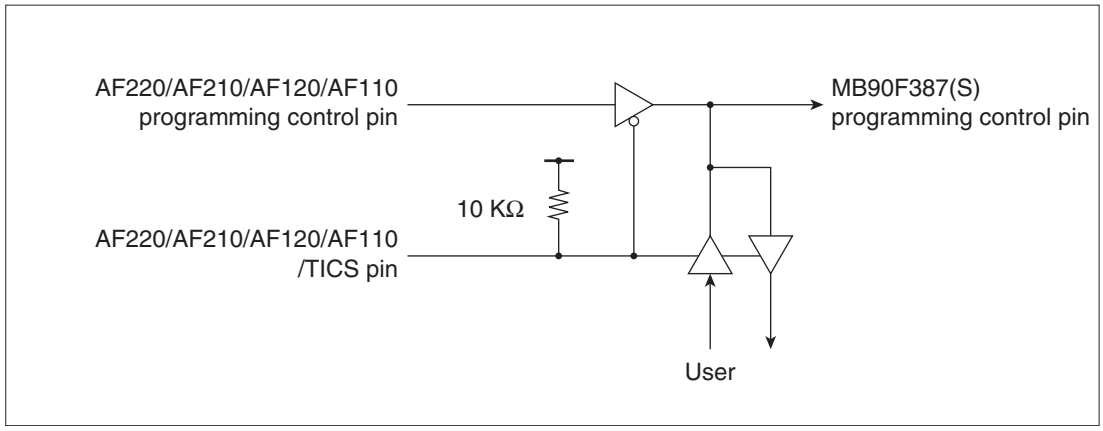
Note: Even if the P30, SIN1, SOT1, and SCK1 pins are used for the user system, the controller shown in the figure below is required. The TICS signal of the flash microcontroller programmer can be used to disconnect the user circuit during serial programming.

See the following serial programming connection examples given in Sections "19.2 Connection Example in Single-chip Mode (User Power Supply)" to "19.5 Example of Minimum Connection to Flash Microcontroller Programmer (Writer Power Supply)".

- Connection example in single-chip mode (user power supply)
- Connection example in single-chip mode (writer power supply)
- Example of minimum connection with flash microcontroller programmer (user power supply)
- Example of minimum connection with flash microcontroller programmer (writer power supply)

et4U.com DataShee DataSheet4U.com

Figure 19.1-2 Control circuit



DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Oscillation Clock Frequency and Serial Clock Input Frequency

The inputable serial clock frequency for the MB90F387/S can be determined by the following expression. Therefore, change the serial clock input frequency according to the setting of the programmer of the flash microcontroller on the basis of the oscillation clock frequency.

Inputable serial clock frequency = 0.125 x oscillation clock frequency

Table 19.1-2 Maximum Serial Clock Frequency

Oscillation Clock Frequency	Maximum serial clock frequency that can be input for the microcomputer	Maximum serial clock frequency that can be set with AF220/AF210/ AF120/AF110	Maximum serial clock frequency that can be set with AF200
4 MHz	500 kHz	500 kHz	500 kHz
8 MHz	1 MHz	850 kHz	500 kHz
16 MHz	2 MHz	1.25 MHz	500 kHz

■ Flash Microcontroller Programmer System Configuration (Made by Yokogawa Digital Computer Corporation)

Table 19.1-3 Flash Microcontroller Programmer System Configuration (Made by Yokogawa Digital Computer Corporation)

Model		Function
Unit	AF220/AC4P	Model with internal Ethernet interface /100 V to 220 V power adapter
	AF210/AC4P	Standard model /100 V to 220 V power adapter
	AF120/AC4P	Single key internal Ethernet interface model /100 V to 220 V power adapter
	AF110/AC4P	Single key model /100 V to 220 V power adapter
AZ221		PC/AT RS232C cable for writer
AZ220		Standard target probe (a) length: 1 m
FF201		Control module for Fujitsu F^2MC-16LX flash microcontroller
/P2		2MB PC Card (Option) FLASH memory corresponding Max. 128 KB
/P4		4MB PC Card (Option) FLASH memory corresponding Max. 512 KB

Note: The AF200 flash microcontroller programmer is an end product but is made available using the control module FF201. Examples of serial programming connections can correspond to those in the next section.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

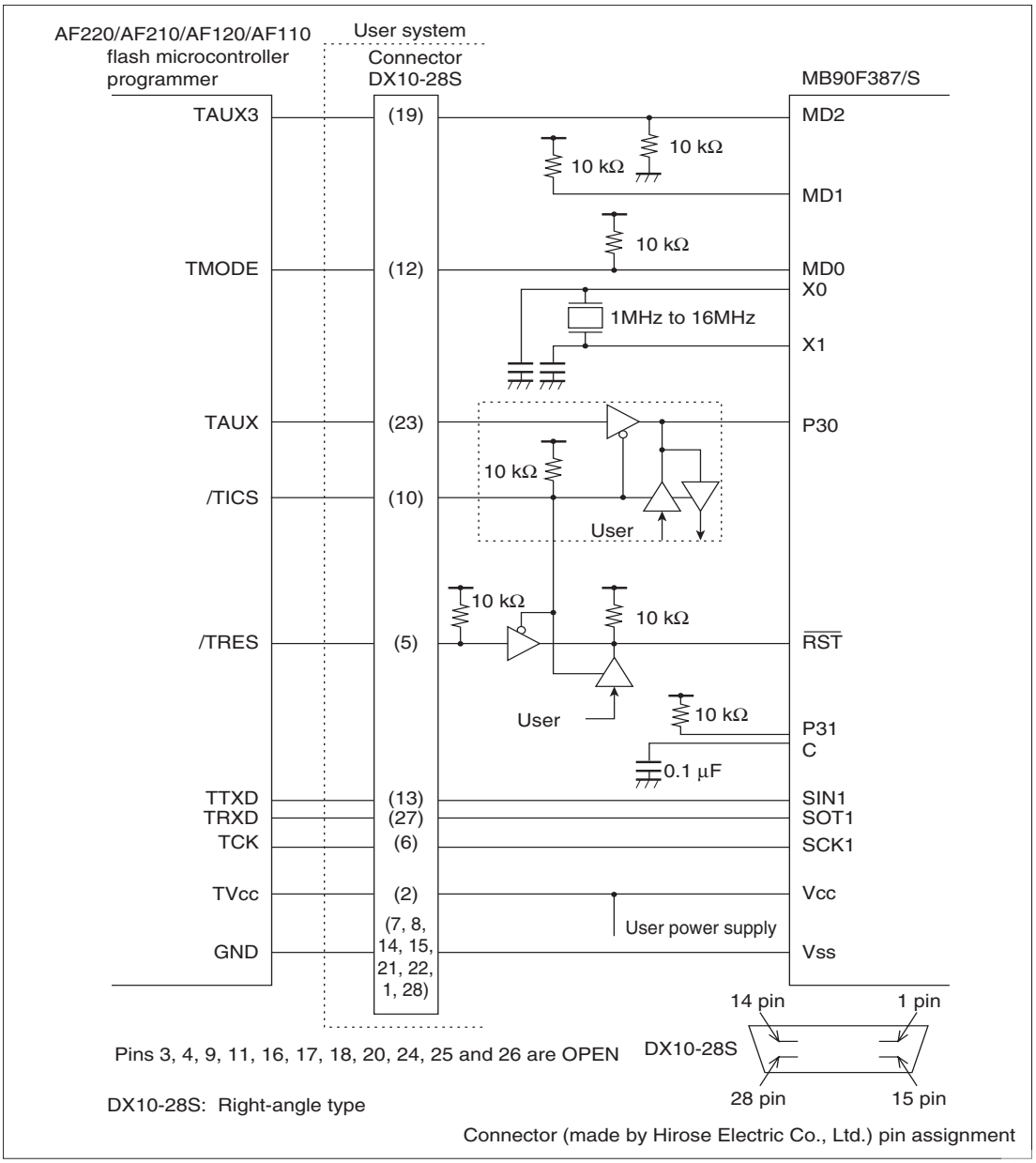
www.DataSheet4U.com

19.2 Connection Example in Single-chip Mode (User Power Supply)

When 1 is input to the mode pin MD2 of the user system placed in single-chip mode and 0 to the mode pin MD0 from the TAUX and TMODE pins of the AF220/AF210/AF120/AF110, the system enters the flash memory serial programming mode. A connection example using the user power supply is given below.

■ Connection Example in Single-chip Mode (User Power Supply Used)

Figure 19.2-1 Example of Serial Programming Connection for MB90F387/S (User Power Supply Used)



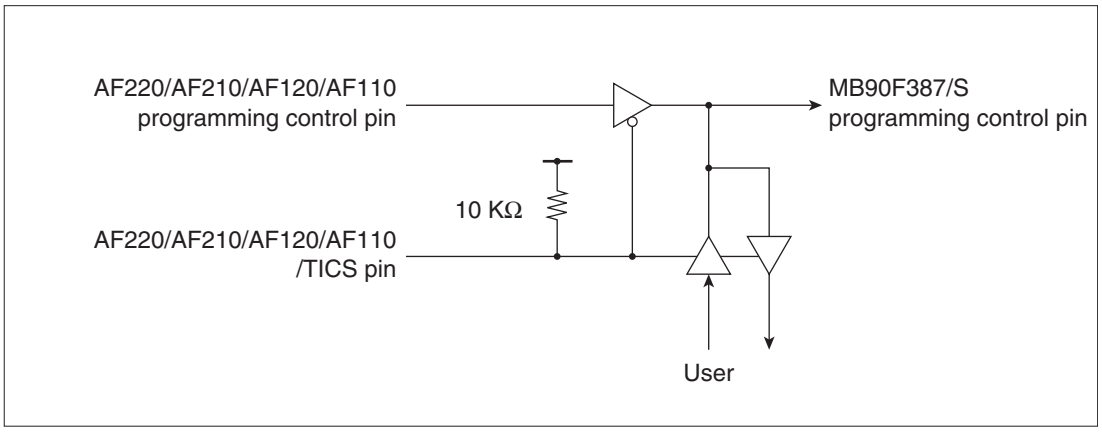
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Notes:
- Even if the SIN1, SOT1, and SCK1 pins are used for the user system, the controller shown in the figure below is required in the same as P30. The /TICS signal of the flash microcontroller programmer can be used to disconnect the user circuit during serial programming.
- Connect the AF220/AF210/AF120/AF110 while the user power is off.

Figure 19.2-2 Control circuit



et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

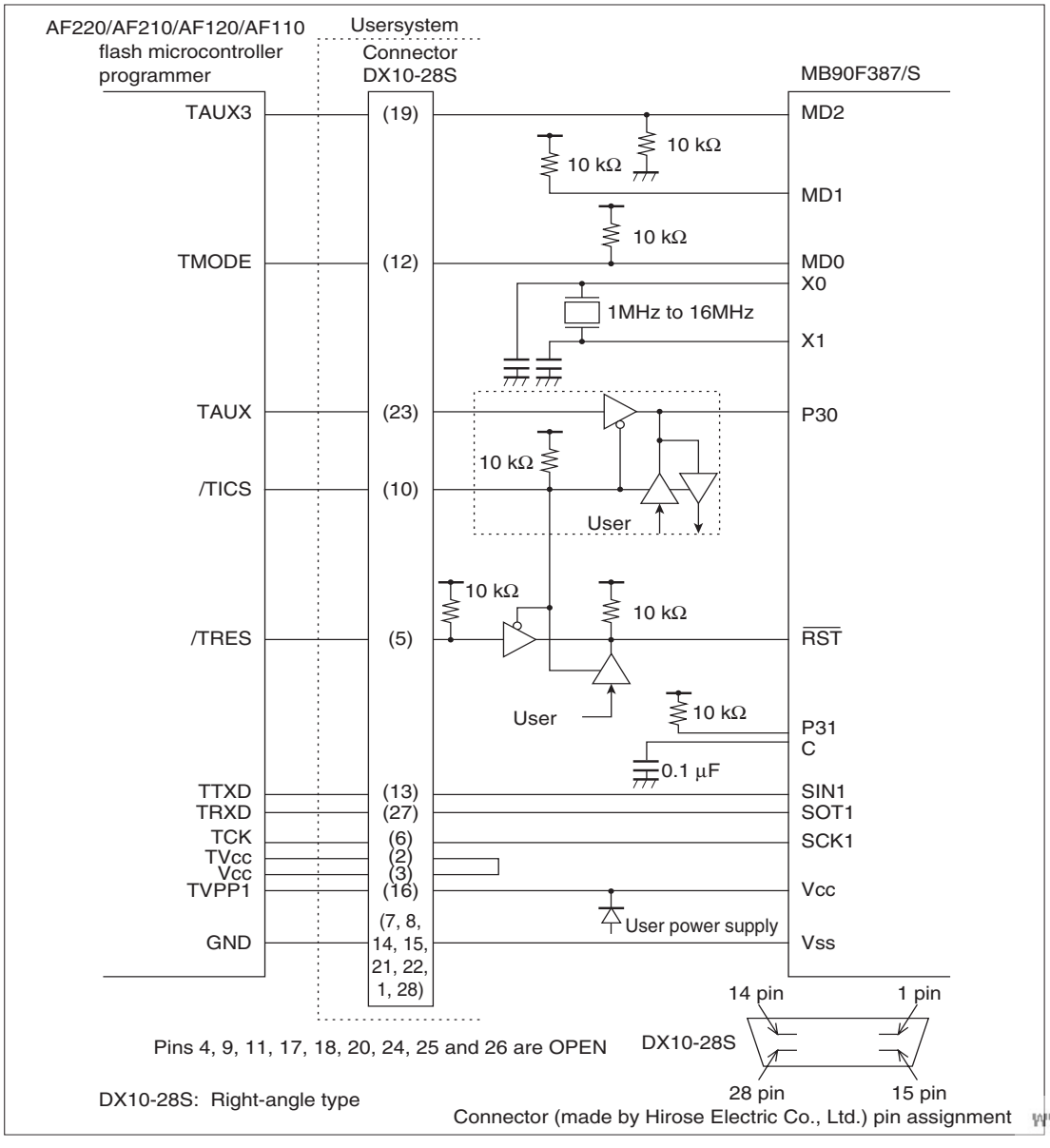
www.DataSheet4U.com

19.3 Connection Example in Single-chip Mode (Writer Power Supply)

When 1 is input to the mode pin MD2 of the user system placed in single-chip mode and 0 to the mode pin MD0 from the TAUX and TMODE pins of the AF220/AF210/AF210/ AF120/AF110, the system enters the flash memory serial programming mode.
A connection example using the writer power supply is given below.

■ Connection Example in Single-chip Mode (Power Supplied from Flash Microcontroller Programmer)

Figure 19.3-1 Example of Serial Programming Connection for MB90F387/S (Power Supplied from Flash Microcontroller Programmer)



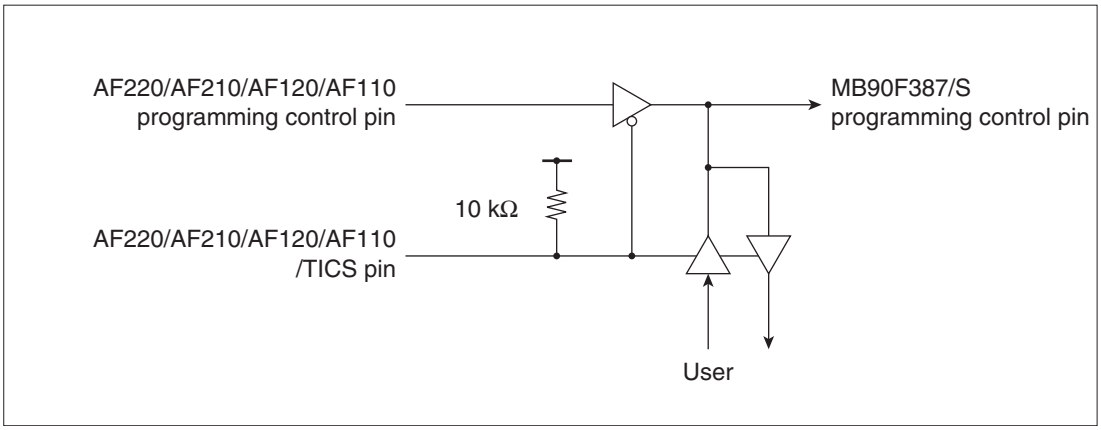
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Notes:
- Even if the SIN1, SOT1, and SCK1 pins are used for the user system, the controller shown in the figure below is required in the same as P30 (Figure 19.3-2). The /TICS signal of the flash microcontroller programmer can be used to disconnect the user circuit during serial programming
- Connect the AF220/AF210/AF120/AF110 while the user power is off.
- When supplying programming power from AF220/AF210/AF120/AF110, do not short-circuit the programming power and user power.

Figure 19.3-2 Control circuit



et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

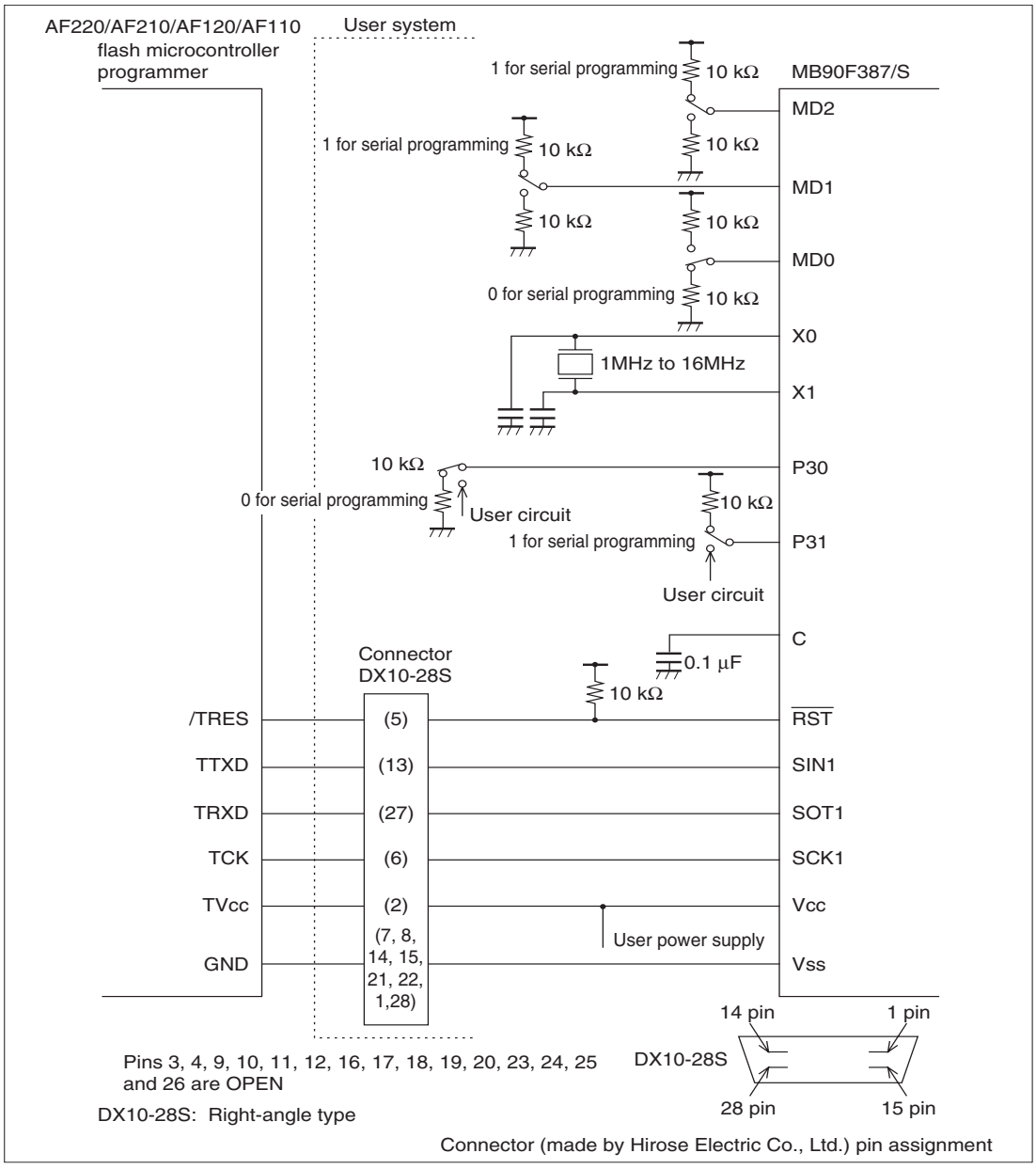
www.DataSheet4U.com

19.4 Example of Minimum Connection to Flash Microcontroller Programmer (User Power Supply)

When each pin is set as shown below at programming to flash memory, there is no need for connections between MD2, MD0, P30 and the flash microcontroller programmer.

■ Example of Minimum Connection to Flash Microcontroller Programmer (User Power Supply Used)

Figure 19.4-1 Example of Minimum Connection to the Flash Microcontroller Programmer (User Power Supply Used)



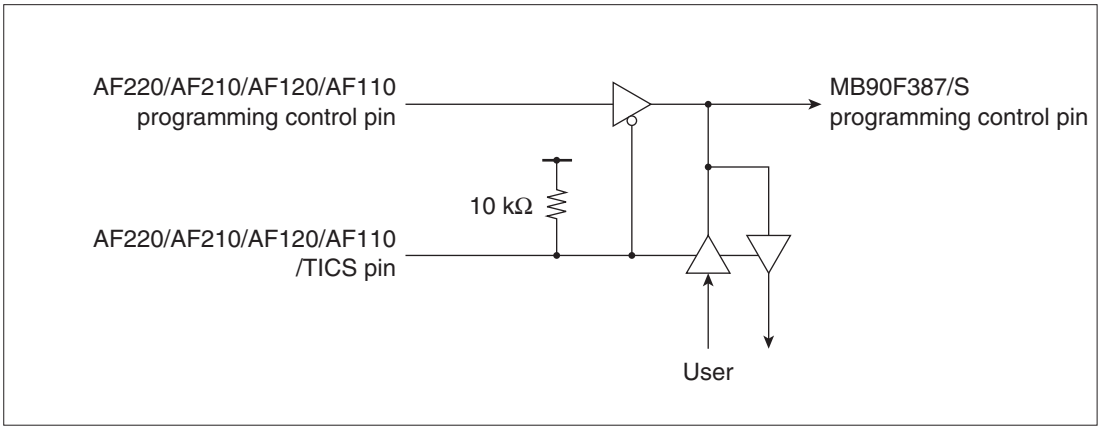
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Notes:
- Even if the SIN1, SOT1, and SCK1 pins are used for the user system, the controller shown in the figure below is required. The /TICS signal of the flash microcontroller programmer can be used to disconnect the user circuit during serial programming.
- Connect the AF220/AF210/AF120/AF110 while the user power is off.

Figure 19.4-2 Control circuit



et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

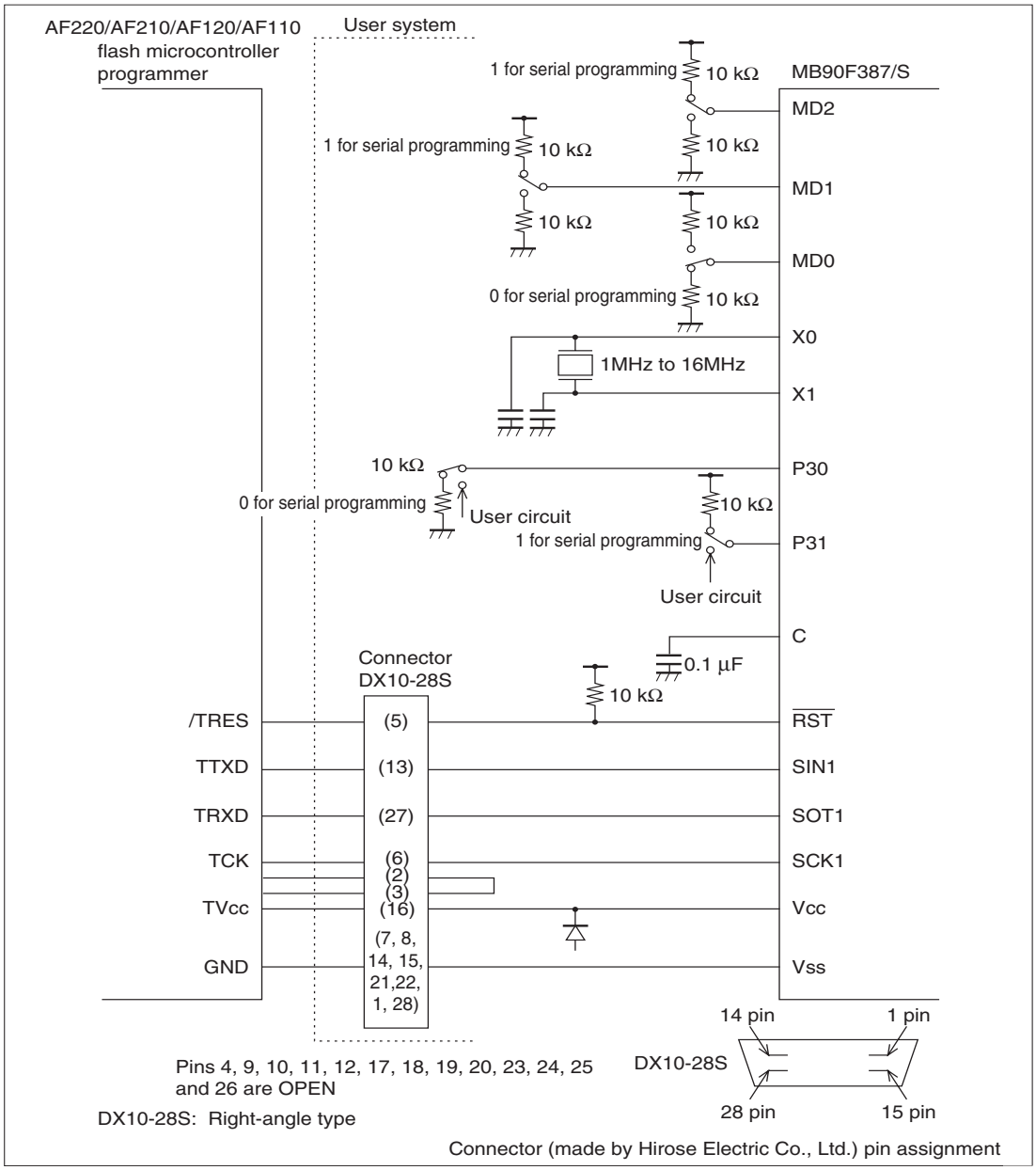
www.DataSheet4U.com

19.5 Example of Minimum Connection to Flash Microcontroller Programmer (Writer Power Supply)

When each pin is set as shown below at programming to flash memory, there is no need for connections between MD2, MD0, P30 and the flash microcontroller programmer.

■ Example of Minimum Connection to Flash Microcontroller Programmer (Power Supplied from Flash Microcontroller Programmer)

Figure 19.5-1 Example of Minimum Connection to the Flash Microcontroller Programmer (Power Supplied from Flash Microcontroller Programmer)



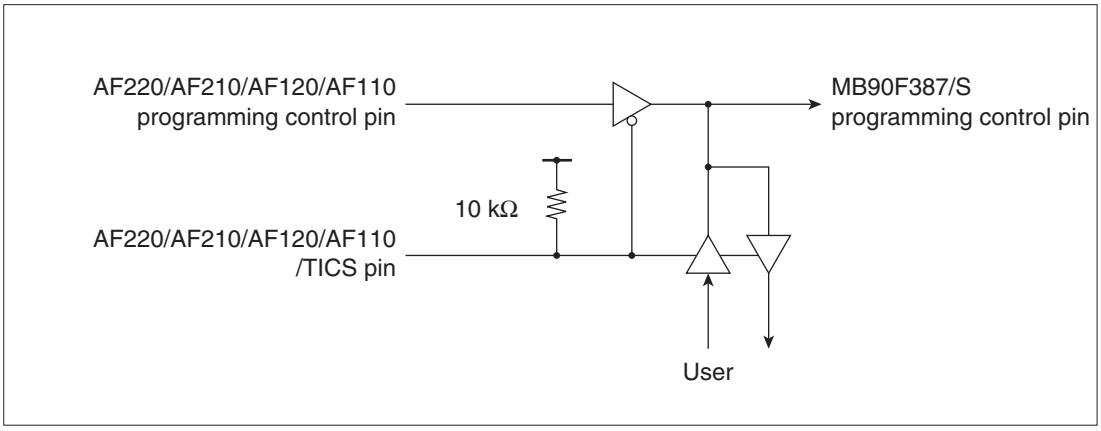
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Notes:
- Even if the SIN1, SOT1, and SCK1 pins are used for the user system, the controller shown in the figure below is required. The /TICS signal of the flash microcontroller programmer can be used to disconnect the user circuit during serial programming.
- Connect the AF220/AF210/AF120/AF110 while the user power is off.
- When supplying programming power from AF220/AF210/AF120/AF110, do not short-circuit the programming power and user power.

Figure 19.5-2 Control circuit



et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

APPENDIX

The appendices provide the I/O map and outline of instructions.

APPENDIX A Instructions
APPENDIX B Register Index
APPENDIX C Pin Function Index
APPENDIX D Interrupt Vector Index

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

APPENDIX A Instructions

Appendix A describes the instructions used by the F^{2}MC-16LX.

A.1 Instruction Types
A.2 Addressing
A.3 Direct Addressing
A.4 Indirect Addressing
A.5 Execution Cycle Count
A.6 Effective Address Field
A.7 How to Read the Instruction List
A.8 F^{2}MC-16LX Instruction List
A.9 Instruction Map

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

A.1 Instruction Types

The F^{2}MC-16LX supports 351 types of instructions. Addressing is enabled by using an effective address field of each instruction or using the instruction code itself.

■ Instruction Types

The F^{2}MC-16LX supports the following 351 types of instructions:

- 41 transfer instructions (byte)
- 38 transfer instructions (word or long word)
- 42 addition/subtraction instructions (byte, word, or long word)
- 12 increment/decrement instructions (byte, word, or long word)
- 11 comparison instructions (byte, word, or long word)
- 11 unsigned multiplication/division instructions (word or long word)
- 11 signed multiplication/division instructions (word or long word)
- 39 logic instructions (byte or word)
- 6 logic instructions (long word)
- 6 sign inversion instructions (byte or word)
- 1 normalization instruction (long word)
- 18 shift instructions (byte, word, or long word)
- 50 branch instructions
- 6 accumulator operation instructions (byte or word)
- 28 other control instructions (byte, word, or long word)
- 21 bit operation instructions
- 10 string instructions

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

A.2 Addressing

With the F²MC-16LX, the address format is determined by the instruction effective address field or the instruction code itself (implied). When the address format is determined by the instruction code itself, specify an address in accordance with the instruction code used. Some instructions permit the user to select several types of addressing.

■ Addressing

The F²MC-16LX supports the following 23 types of addressing:

- Immediate (#imm)
- Register direct
- Direct branch address (addr16)
- Physical direct branch address (addr24)
- I/O direct (io)
- Abbreviated direct address (dir)
- Direct address (addr16)
- I/O direct bit address (io:bp)
- Abbreviated direct bit address (dir:bp)
- Direct bit address (addr16:bp)
- Vector address (#vct)
- Register indirect (@RWj j = 0 to 3)
- Register indirect with post increment (@RWj+ j = 0 to 3)
- Register indirect with displacement (@RWi + disp8 i = 0 to 7, @RWj+ disp16 j = 0 to 3)
- Long register indirect with displacement (@RLi + disp8 i = 0 to 3)
- Program counter indirect with displacement (@PC + disp16)
- Register indirect with base index (@RW0 + RW7, @RW1 + RW7)
- Program counter relative branch address (rel)
- Register list (rlst)
- Accumulator indirect (@A)
- Accumulator indirect branch address (@A)
- Indirectly-specified branch address (@ear)
- Indirectly-specified branch address (@eam)

et4U.com DataShee DataSheet4U.com DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Effective Address Field

Table A.2-1 lists the address formats specified by the effective address field.

Table A.2-1 Effective Address Field

Code	Representation			Address format	Default bank
00	R0	RW0	RL0	Register direct: Individual parts correspond to the byte, word, and long word types in order from the left.	None
01	R1	RW1	(RL0)		
02	R2	RW2	RL1		
03	R3	RW3	(RL1)		
04	R4	RW4	RL2		
05	R5	RW5	(RL2)		
06	R6	RW6	RL3		
07	R7	RW7	(RL3)		
08	@RW0			Register indirect	DTB
09	@RW1				DTB
0A	@RW2				ADB
0B	@RW3				SPB
0C	@RW0+			Register indirect with post increment	DTB
0D	@RW1+				DTB
0E	@RW2+				ADB
0F	@RW3+				SPB
10	@RW0+disp8			Register indirect with 8-bit displacement	DTB
11	@RW1+disp8				DTB
12	@RW2+disp8				ADB
13	@RW3+disp8				SPB
14	@RW4+disp8			Register indirect with 8-bit displacement	DTB
15	@RW5+disp8				DTB
16	@RW6+disp8				ADB
17	@RW7+disp8				SPB
18	@RW0+disp16			Register indirect with 16-bit displacement	DTB
19	@RW1+disp16				DTB
1A	@RW2+disp16				ADB
1B	@RW3+disp16				SPB
1C	@RW0+RW7			Register indirect with index	DTB
1D	@RW1+RW7			Register indirect with index	DTB
1E	@PC+disp16			PC indirect with 16-bit displacement	PCB
1F	addr16			Direct address	DTB

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

A.3 Direct Addressing

An operand value, register, or address is specified explicitly in direct addressing mode.

■ Direct Addressing

● Immediate addressing (#imm)

Specify an operand value explicitly (#imm4/ #imm8/ #imm16/ #imm32).

Figure A.3-1 Example of immediate addressing (#imm)

MOVW A, #01212H (This instruction stores the operand value in A.)

Before execution	A	2233 4455	
After execution	A	4455 1212	(Some instructions transfer AL to AH.)

● Register direct addressing

Specify a register explicitly as an operand. Table A.3-1 lists the registers that can be specified. Figure A.3-2 shows an example of register direct addressing.

Table A.3-1 Direct Addressing Registers

General-purpose register	Byte	R0, R1, R2, R3, R4, R5, R6, R7
	Word	RW0, RW1, RW2, RW3, RW4, R5W, RW6, RW7
	Long word	RL0, RL1, RL2, RL3
Special-purpose register	Accumulator	A, AL
	Pointer	SP *
	Bank	PCB, DTB, USB, SSB, ADB
	Page	DPR
	Control	PS, CCR, RP, ILM

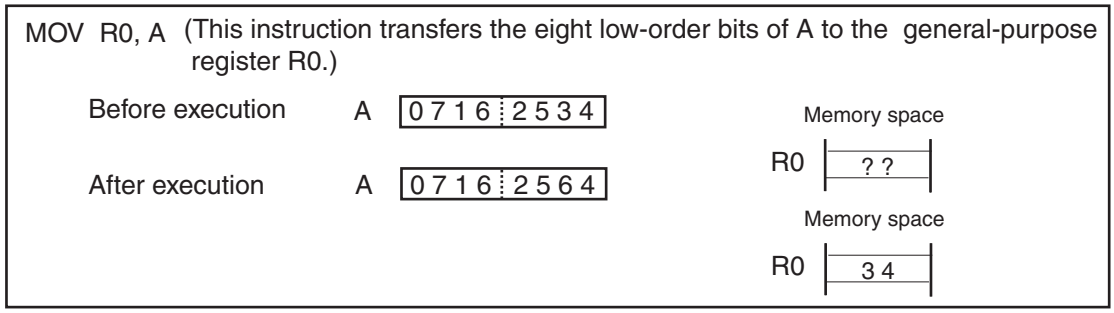
*: One of the user stack pointer (USP) and system stack pointer (SSP) is selected and used depending on the value of the S flag bit in the condition code register (CCR). For branch instructions, the program counter (PC) is not specified in an instruction operand but is specified implicitly.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

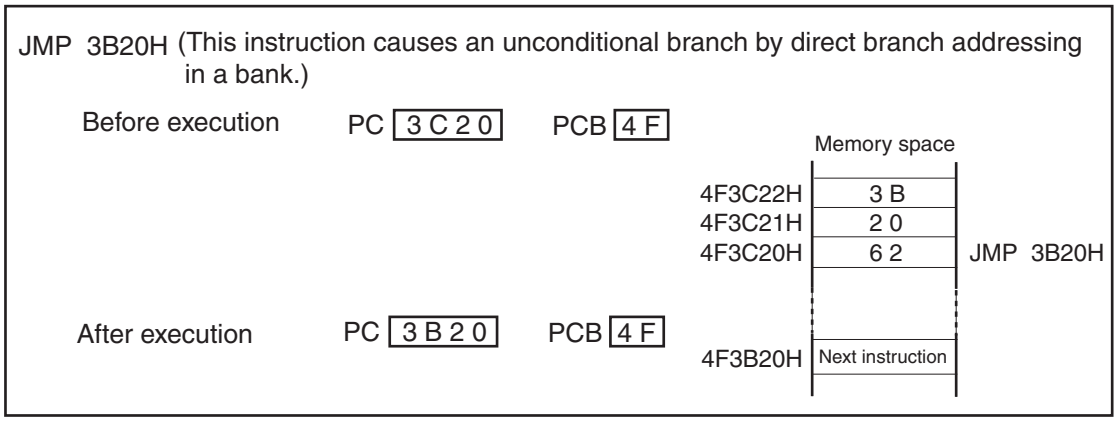
Figure A.3-2 Example of Register Direct Addressing



● Direct branch addressing (addr16)

Specify an offset explicitly for the branch destination address. The size of the offset is 16 bits, which indicates the branch destination in the logical address space. Direct branch addressing is used for an unconditional branch, subroutine call, or software interrupt instruction. Bits 23 to 16 of the address are specified by the program bank register (PCB).

Figure A.3-3 Example of Direct Branch Addressing (addr16)



DataSheet4U.com
www.DataSheet4U.com

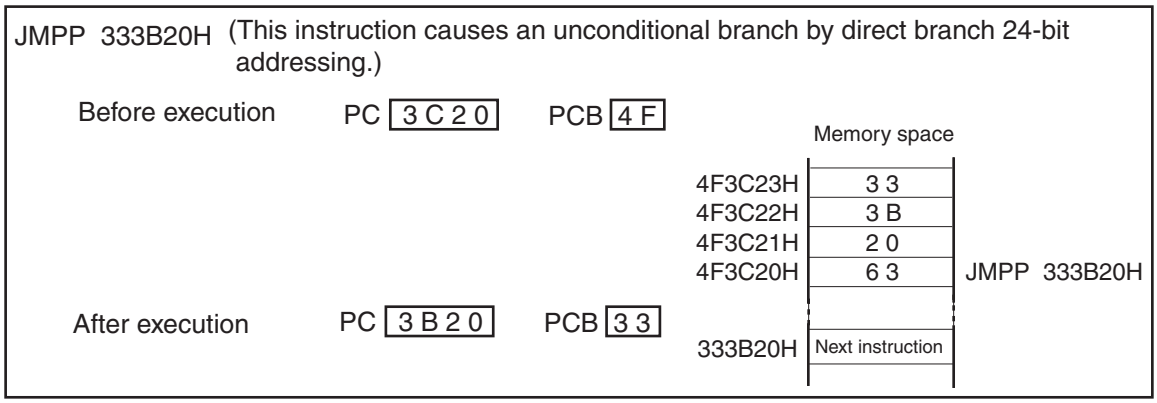
DataSheet4U.com

www.DataSheet4U.com

● Physical direct branch addressing (addr24)

Specify an offset explicitly for the branch destination address. The size of the offset is 24 bits. Physical direct branch addressing is used for unconditional branch, subroutine call, or software interrupt instruction.

Figure A.3-4 Example of Direct Branch Addressing (addr24)

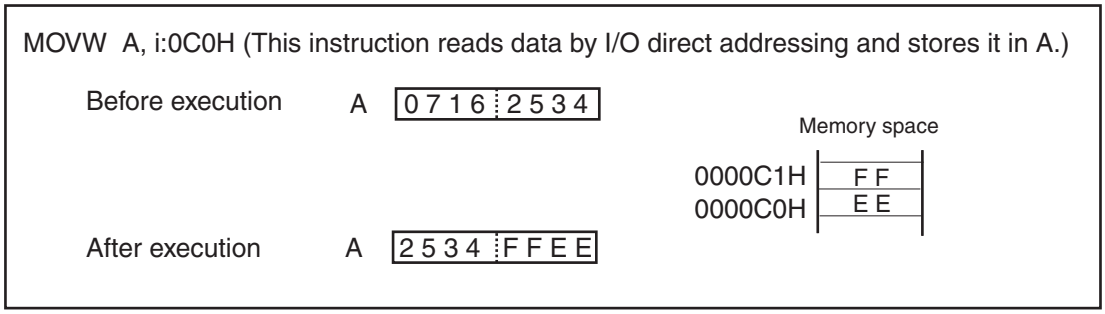


● I/O direct addressing (io)

Specify an 8-bit offset explicitly for the memory address in an operand. The I/O address space in the physical address space from 000000_H to $0000FF_H$ is accessed regardless of the data bank register (DTB) and direct page register (DPR). A bank select prefix for bank addressing is invalid if specified before an instruction using I/O direct addressing.

DataSheet4U.com

Figure A.3-5 Example of I/O Direct Addressing (io)



DataSheet4U.com
www.DataSheet4U.com

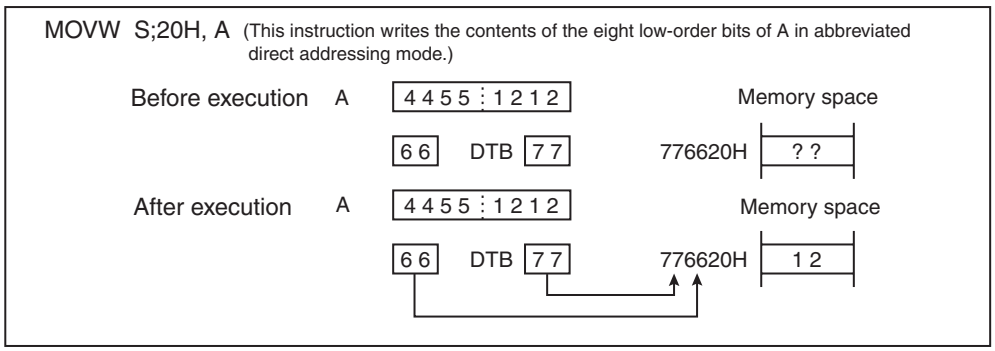
DataSheet4U.com

www.DataSheet4U.com

● Abbreviated direct addressing (dir)

Specify the eight low-order bits of a memory address explicitly in an operand. Address bits 8 to 15 are specified by the direct page register (DPR). Address bits 16 to 23 are specified by the data bank register (DTB).

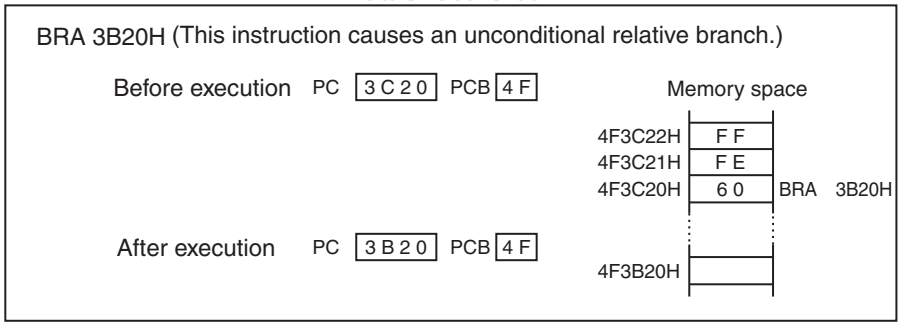
Figure A.3-6 Example of Abbreviated Direct Addressing (dir)



● Direct addressing (addr16)

Specify the 16 low-order bits of a memory address explicitly in an operand. Address bits 16 to 23 are specified by the data bank register (DTB). A prefix instruction for access space addressing is invalid for this mode of addressing.

Figure A.3-7 Example of Direct Addressing (addr16)



DataSheet4U.com
www.DataSheet4U.com

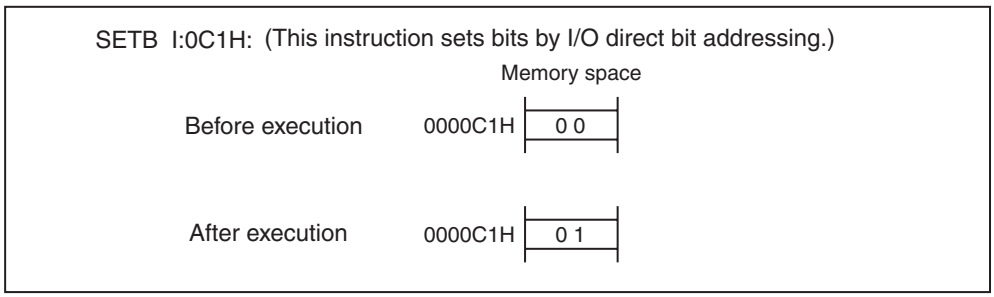
DataSheet4U.com

www.DataSheet4U.com

● I/O direct bit addressing (io:bp)

Specify bits in physical addresses 000000_H to $0000FF_H$ explicitly. Bit positions are indicated by ":bp", where the larger number indicates the most significant bit (MSB) and the lower number indicates the least significant bit (LSB).

Figure A.3-8 Example of I/O Direct Bit Addressing (io:bp)



● Abbreviated direct bit addressing (dir:bp)

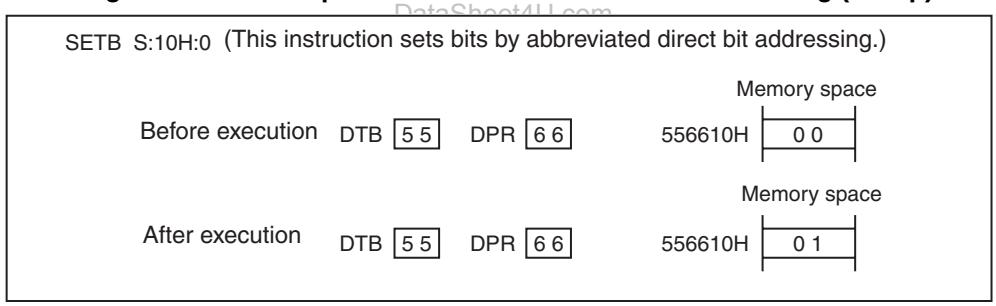
Specify the eight low-order bits of a memory address explicitly in an operand. Address bits 8 to 15 are specified by the direct page register (DPR). Address bits 16 to 23 are specified by the data bank register (DTB). Bit positions are indicated by ":bp", where the larger number indicates the most significant bit (MSB) and the lower number indicates the least significant bit (LSB).

et4U.com

DataShee

Figure A.3-9 Example of Abbreviated Direct Bit Addressing (dir:bp)

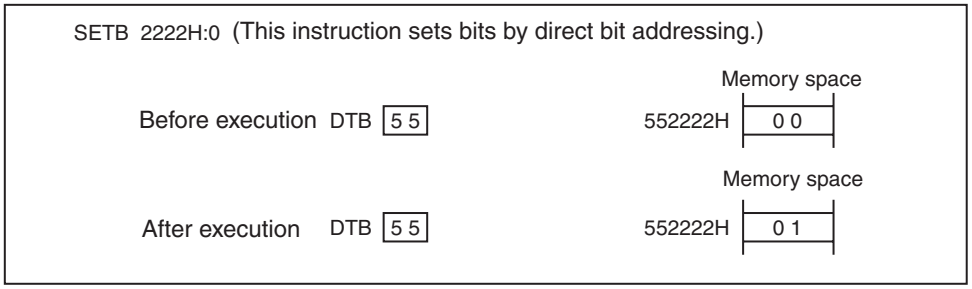
DataSheet4U.com



● Direct bit addressing (addr16:bp)

Specify arbitrary bits in 64 kilobytes explicitly. Address bits 16 to 23 are specified by the data bank register (DTB). Bit positions are indicated by ":bp", where the larger number indicates the most significant bit (MSB) and the lower number indicates the least significant bit (LSB).

Figure A.3-10 Example of Direct Bit addressing (addr16:bp)



DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

● Vector Addressing (#vct)

Specify vector data in an operand to indicate the branch destination address. There are two sizes for vector numbers: 4 bits and 8 bits. Vector addressing is used for a subroutine call or software interrupt instruction.

Figure A.3-11 Example of Vector Addressing (#vct)

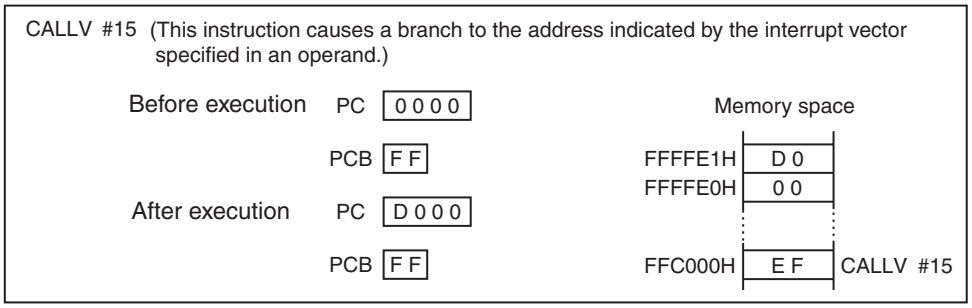


Table A.3-2 CALLV Vector List

Instruction	Vector address L	Vector address H
CALLV #0	$XXFFFE_H$	$XXFFFF_H$
CALLV #1	$XXFFFC_H$	$XXFFFD_H$
CALLV #2	$XXFFFA_H$	$XXFFFB_H$
CALLV #3	$XXFFF8_H$	$XXFFF9_H$
CALLV #4	$XXFFF6_H$	$XXFFF7_H$
CALLV #5	$XXFFF4_H$	$XXFFF5_H$
CALLV #6	$XXFFF2_H$	$XXFFF3_H$
CALLV #7	$XXFFF0_H$	$XXFFF1_H$
CALLV #8	$XXFFEE_H$	$XXFFEF_H$
CALLV #9	$XXFFEC_H$	$XXFFED_H$
CALLV #10	$XXFFEA_H$	$XXFFEB_H$
CALLV #11	$XXFFE8_H$	$XXFFE9_H$
CALLV #12	$XXFFE6_H$	$XXFFE7_H$
CALLV #13	$XXFFE4_H$	$XXFFE5_H$
CALLV #14	$XXFFE2_H$	$XXFFE3_H$
CALLV #15	$XXFFE0_H$	$XXFFE1_H$

Note: A PCB register value is set in XX.

Note: When the program bank register (PCB) is FF_H, the vector area overlaps the vector area of INT #vct8 (#0 to #7). Use vector addressing carefully (see Table A.3-2).

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

A.4 Indirect Addressing

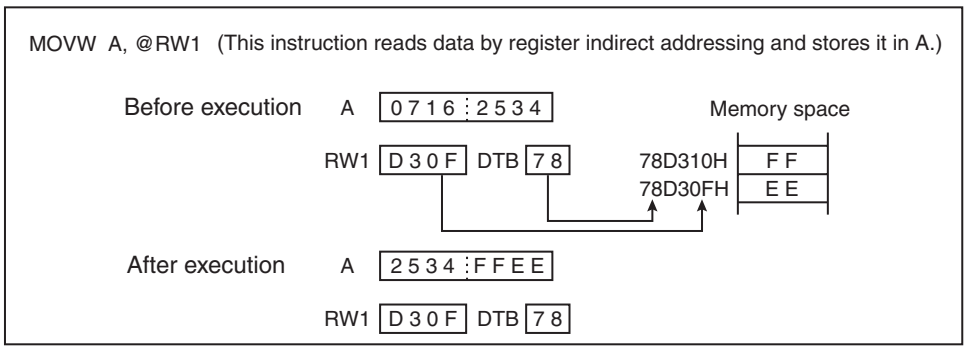
In indirect addressing mode, an address is specified indirectly by the address data of an operand.

■ Indirect Addressing

● Register indirect addressing (@RWj j = 0 to 3)

Memory is accessed using the contents of general-purpose register RWj as an address. Address bits 16 to 23 are indicated by the data bank register (DTB) when RW0 or RW1 is used, system stack bank register (SSB) or user stack bank register (USB) when RW3 is used, or additional data bank register (ADB) when RW2 is used.

Figure A.4-1 Example of Register Indirect Addressing (@RWj j = 0 to 3)



● Register indirect addressing with post increment (@RWj+ j = 0 to 3)

Memory is accessed using the contents of general-purpose register RWj as an address. After operand operation, RWj is incremented by the operand size (1 for a byte, 2 for a word, or 4 for a long word). Address bits 16 to 23 are indicated by the data bank register (DTB) when RW0 or RW1 is used, system stack bank register (SSB) or user stack bank register (USB) when RW3 is used, or additional data bank register (ADB) when RW2 is used.

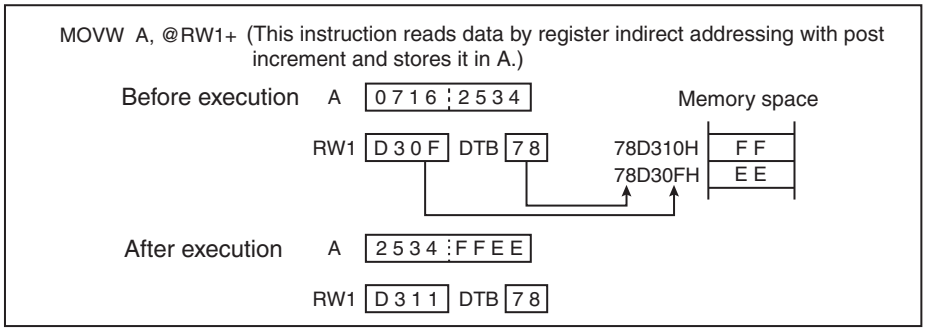
If the post increment results in the address of the register that specifies the increment, the incremented value is referenced after that. In this case, if the next instruction is a write instruction, priority is given to writing by an instruction and, therefore, the register that would be incremented becomes write data.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

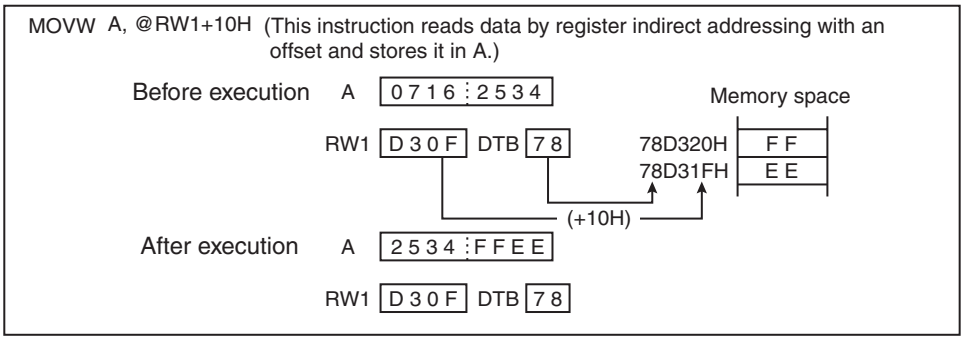
Figure A.4-2 Example of Register Indirect Addressing with Post Increment (@RWj + j = 0 to 3)



● Register indirect addressing with offset (@RWi + disp8 i = 0 to 7, @RWj + disp16 j = 0 to 3)

Memory is accessed using the address obtained by adding an offset to the contents of general-purpose register RWj. Two types of offset, byte and word offsets, are used. They are added as signed numeric values. Address bits 16 to 23 are indicated by the data bank register (DTB) when RW0, RW1, RW4, or RW5 is used, system stack bank register (SSB) or user stack bank register (USB) when RW3 or RW7 is used, or additional data bank register (ADB) when RW2 or RW6 is used.

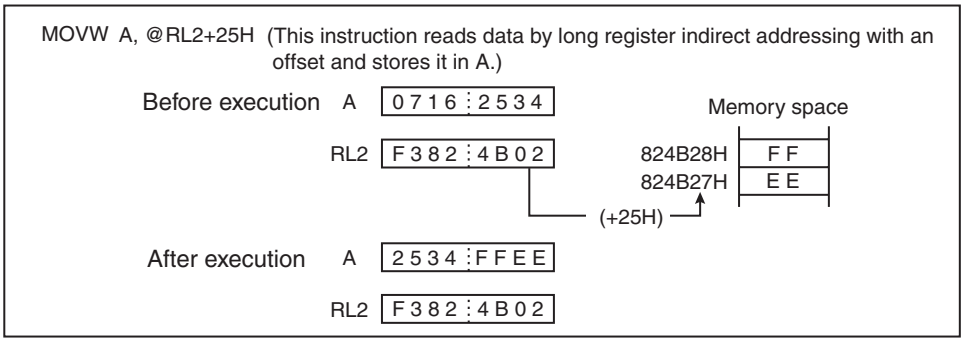
Figure A.4-3 Example of Register Indirect Addressing with Offset (@RWi + disp8 i = 0 to 7, @RWj + disp16 j = 0 to 3)



● Long register indirect addressing with offset (@RLi + disp8 i = 0 to 3)

Memory is accessed using the address that is the 24 low-order bits obtained by adding an offset to the contents of general-purpose register RLi. The offset is 8-bits long and is added as a signed numeric value.

Figure A.4-4 Example of Long Register Indirect Addressing with Offset (@RLi + disp8 i = 0 to 3)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

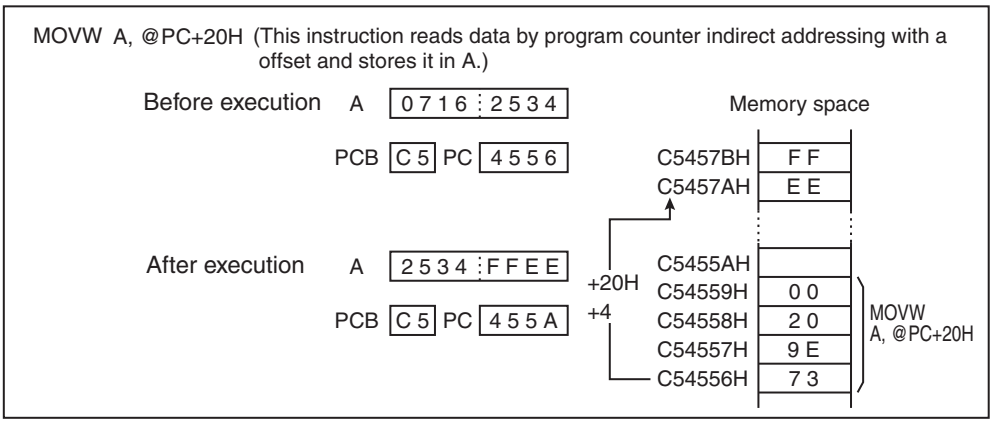
www.DataSheet4U.com

● Program counter indirect addressing with offset (@PC + disp16)

Memory is accessed using the address indicated by (instruction address + 4 + disp16). The offset is one word long. Address bits 16 to 23 are specified by the program bank register (PCB). Note that the operand address of each of the following instructions is not deemed to be (next instruction address + disp16):

- DBNZ eam, rel
- DWBNZ eam, rel
- CBNE eam, #imm8, rel
- CWBNE eam, #imm16, rel
- MOV eam, #imm8
- MOVW eam, #imm16

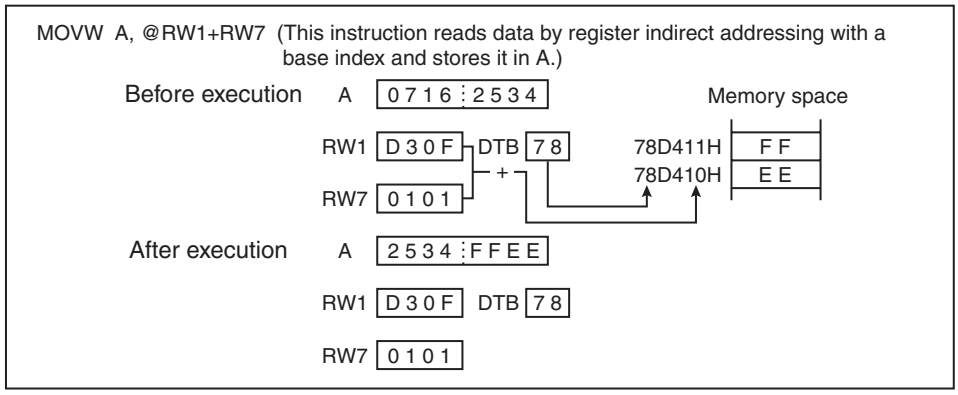
Figure A.4-5 Example of Program Counter Indirect Addressing with Offset (@PC + disp16)



● Register indirect addressing with base index (@RW0 + RW7, @RW1 + RW7)

Memory is accessed using the address determined by adding RW0 or RW1 to the contents of general-purpose register RW7. Address bits 16 to 23 are indicated by the data bank register (DTB).

Figure A.4-6 Example of Register Indirect Addressing with Base Index (@RW0 + RW7, @RW1 + RW7)



DataSheet4U.com

www.DataSheet4U.com

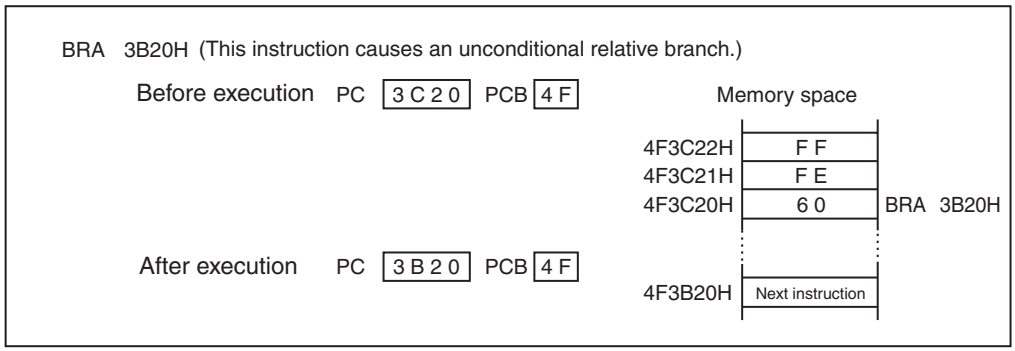
DataSheet4U.com

www.DataSheet4U.com

● Program counter relative branch addressing (rel)

The address of the branch destination is a value determined by adding an 8-bit offset to the program counter (PC) value. If the result of addition exceeds 16 bits, bank register incrementing or decrementing is not performed and the excess part is ignored, and therefore the address is contained within a 64-kilobyte bank. This addressing is used for both conditional and unconditional branch instructions. Address bits 16 to 23 are indicated by the program bank register (PCB).

Figure A.4-7 Example of Program Counter Relative Branch Addressing (rel)



● Register list (rlst)

Specify a register to be pushed onto or popped from a stack.

Figure A.4-8 Configuration of the Register List

MSB							LSB
RW7	RW6	RW5	RW4	RW3	RW2	RW1	RW0

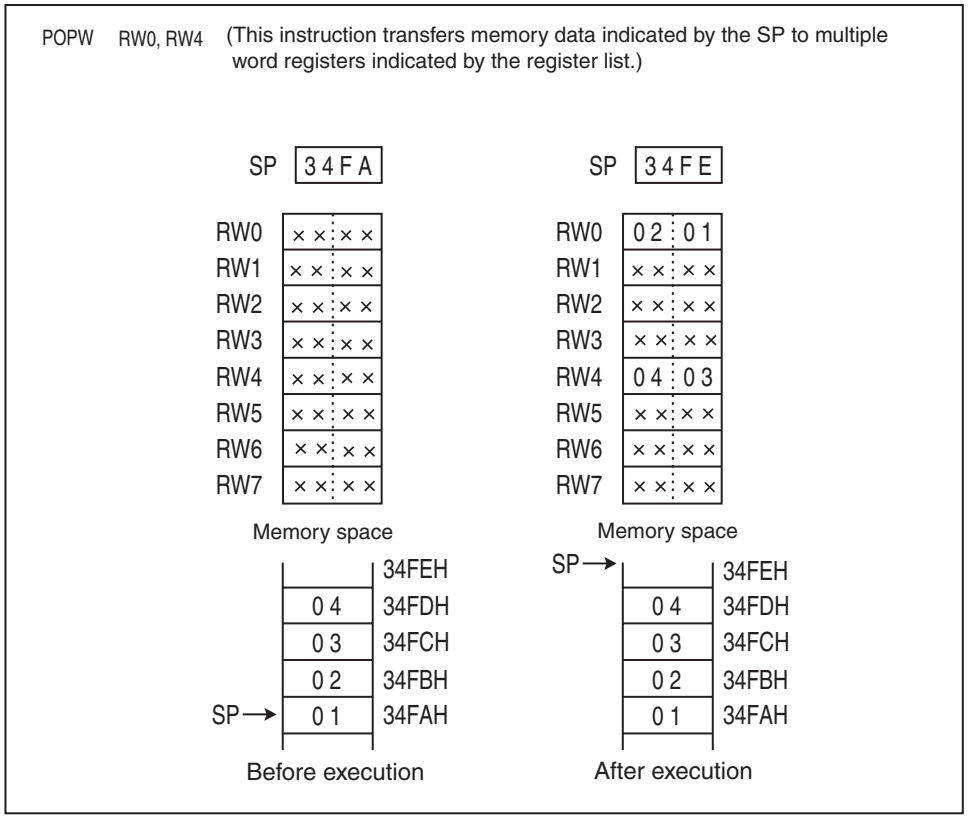
A register is selected when the corresponding bit is 1 and deselected when the bit is 0.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

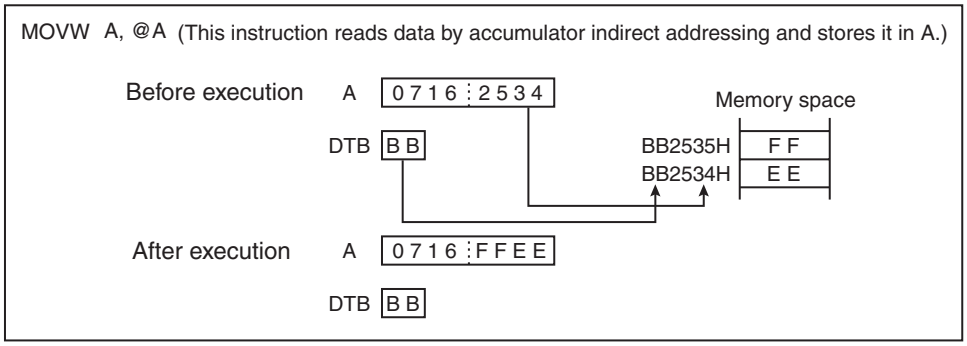
Figure A.4-9 Example of Register List (rlist)



● Accumulator indirect addressing (@A)

Memory is accessed using the address indicated by the contents of the low-order bytes (16 bits) of the accumulator (AL). Address bits 16 to 23 are specified by a mnemonic in the data bank register (DTB).

Figure A.4-10 Example of Accumulator Indirect Addressing (@A)



DataSheet4U.com

www.DataSheet4U.com

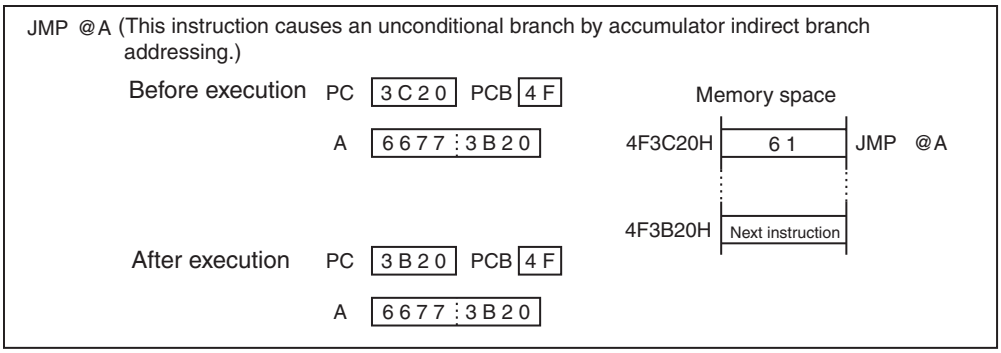
DataSheet4U.com

www.DataSheet4U.com

● Accumulator indirect branch addressing (@A)

The address of the branch destination is the content (16 bits) of the low-order bytes (AL) of the accumulator. It indicates the branch destination in the bank address space. Address bits 16 to 23 are specified by the program bank register (PCB). For the Jump Context (JCTX) instruction, however, address bits 16 to 23 are specified by the data bank register (DTB). This addressing is used for unconditional branch instructions.

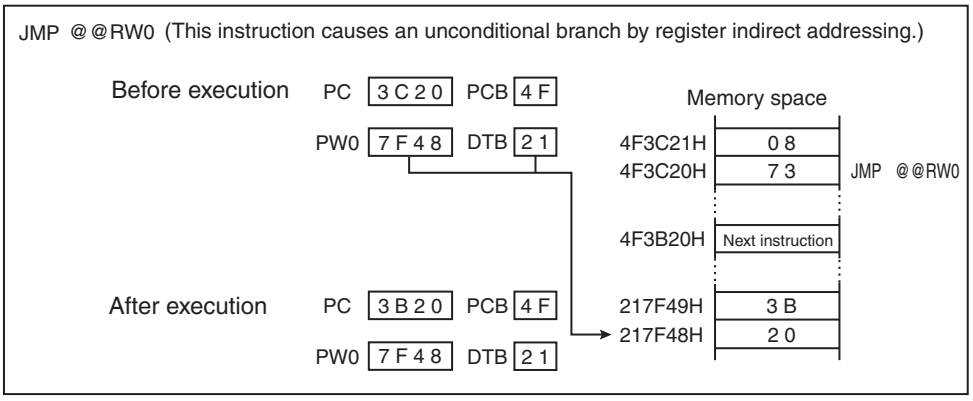
Figure A.4-11 Example of Accumulator Indirect Branch Addressing (@A)



● Indirect specification branch addressing (@ear)

The address of the branch destination is the word data at the address indicated by ear.

Figure A.4-12 Example of Indirect Specification Branch Addressing (@ear)



DataSheet4U.com
www.DataSheet4U.com

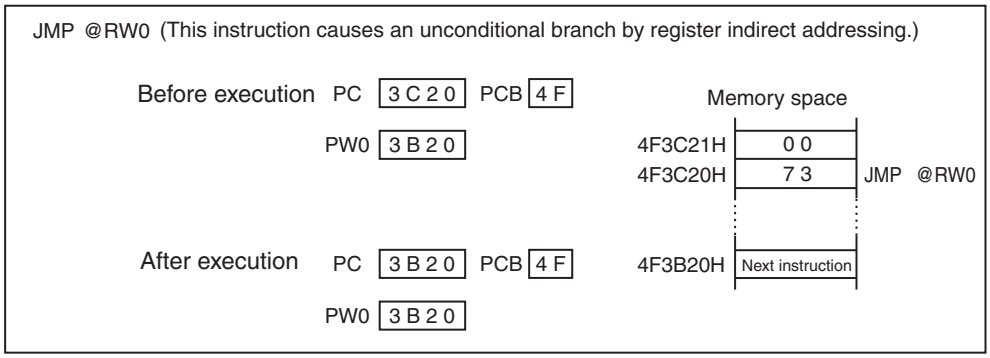
DataSheet4U.com

www.DataSheet4U.com

● Indirect specification branch addressing (@eam)

The address of the branch destination is the word data at the address indicated by eam.

Figure A.4-13 Example of Indirect Specification Branch Addressing (@eam)



et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

A.5 Execution Cycle Count

The number of cycles required for instruction execution (execution cycle count) is obtained by adding the number of cycles required for each instruction, "correction value" determined by the condition, and the number of cycles for instruction fetch.

■ Execution Cycle Count

The number of cycles required for instruction execution (execution cycle count) is obtained by adding the number of cycles required for each instruction, "correction value" determined by the condition, and the number of cycles for instruction fetch. In the mode of fetching an instruction from memory such as internal ROM connected to a 16-bit bus, the program fetches the instruction being executed in word increments. Therefore, intervening in data access increases the execution cycle count.

Similarly, in the mode of fetching an instruction from memory connected to an 8-bit external bus, the program fetches every byte of an instruction being executed. Therefore, intervening in data access increases the execution cycle count. In CPU intermittent operation mode, access to a general-purpose register, internal ROM, internal RAM, internal I/O, or external data bus causes the clock to the CPU to halt for the cycle count specified by the CG0 and CG1 bits of the low power consumption mode control register. Therefore, for the cycle count required for instruction execution in CPU intermittent operation mode, add the "access count x cycle count for the halt" as a correction value to the normal execution count.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

■ Calculating the Execution Cycle Count

Table A.5-1 lists execution cycle counts and Table A.5-2 and Table A.5-3 summarize correction value data.

Table A.5-1 Execution Cycle Counts in Each Addressing Mode

Code	Operand	(a) *	Register access count in each addressing mode
		Execution cycle count in each addressing mode	
00 to 07	Ri Rwi RLi	See the instruction list.	See the instruction list.
08 to 0B	@RWj	2	1
0C to 0F	@RWj+	4	2
10 to 17	@RWi+disp8	2	1
18 to 1B	@RWi+disp16	2	1
1C 1D 1E 1F	@RW0+RW7 @RW1+RW7 @PC+disp16 addr16	4 4 2 1	2 2 0 0

et4U.com
DataShee
DataSheet4U.com

* : (a) is used for ～ (cycle count) and B (correction value) in "A.8 F^{2}MC-16LX Instruction List".

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.5-2 Cycle Count Correction Values for Counting Execution Cycles

Operand	(b) byte (*)		(c) word (*)		(d) long (*)	
	Cycle count	Access count	Cycle count	Access count	Cycle count	Access count
Internal register	+0	1	+0	1	+0	2
Internal memory Even address	+0	1	+0	1	+0	2
Internal memory Odd address	+0	1	+2	2	+4	4
External data bus 16-bit even address	+1	1	+1	1	+2	2
External data bus 16-bit odd address	+1	1	+4	2	+8	4
External data bus 8-bits	+1	1	+4	2	+8	4

* : (b), (c), and (d) are used for ∼ (cycle count) and B (correction value) in "A.8 F^2MC-16LX Instruction List".

DataSheet4U.com

Note: When an external data bus is used, the cycle counts during which an instruction is made to wait by ready input or automatic ready must also be added.

Table A.5-3 Cycle Count Correction Values for Counting Instruction Fetch Cycles

Instruction	Byte boundary	Word boundary
Internal memory	-	+2
External data bus 16-bits	-	+3
External data bus 8-bits	+3	-

Notes:
- When an external data bus is used, the cycle counts during which an instruction is made to wait by ready input or automatic ready must also be added.
- Actually, instruction execution is not delayed by every instruction fetch. Therefore, use the correction values to calculate the worst case.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

A.6 Effective Address Field

Table A.6-1 shows the effective address field.

■ Effective Address Field

Table A.6-1 Effective Address Field

Code	Representation			Address format	Byte count of extended address part *
00	R0	RW0	RL0	Register direct: Individual parts correspond to the byte, word, and long word types in order from the left.	-
01	R1	RW1	(RL0)		
02	R2	RW2	RL1		
03	R3	RW3	(RL1)		
04	R4	RW4	RL2		
05	R5	RW5	(RL2)		
06	R6	RW6	RL3		
07	R7	RW7	(RL3)		
08	@RW0			Register indirect	0
09	@RW1				
0A	@RW2				
0B	@RW3				
0C	@RW0+			Register indirect with post increment	0
0D	@RW1+				
0E	@RW2+				
0F	@RW3+				
10	@RW0+disp8			Register indirect with 8-bit displacement	1
11	@RW1+disp8				
12	@RW2+disp8				
13	@RW3+disp8				
14	@RW4+disp8				
15	@RW5+disp8				
16	@RW6+disp8				
17	@RW7+disp8				
18	@RW0+disp16			Register indirect with 16-bit displacement	2
19	@RW1+disp16				
1A	@RW2+disp16				
1B	@RW3+disp16				
1C	@RW0+RW7			Register indirect with index	0
1D	@RW1+RW7			Register indirect with index	0
1E	@PC+disp16			PC indirect with 16-bit displacement	2
1F	addr16			Direct address	2

*: Each byte count of the extended address part applies to + in the # (byte count) column in "A.8 F^{2}MC-16LX Instruction List".

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

A.7 How to Read the Instruction List

Table A.7-1 describes the items used in the F^{2}MC-16LX Instruction List, and Table A.7-2 describes the symbols used in the same list.

■ Description of instruction presentation items and symbols

Table A.7-1 Description of Items in the Instruction List (1/2)

Item	Description
Mnemonic	Uppercase, symbol: Represented as is in the assembler. Lowercase: Rewritten in the assembler. Number of following lowercase: Indicates bit length in the instruction.
#	Indicates the number of bytes.
~	Indicates the number of cycles. See Table A.2-1 for the alphabetical letters in items.
RG	Indicates the number of times a register access is performed during instruction execution. The number is used to calculate the correction value for CPU intermittent operation.
B	Indicates the correction value used to calculate the actual number of cycles during instruction execution. The actual number of cycles during instruction execution can be determined by adding the value in the ~ column to this value.
Operation	Indicates the instruction operation.
LH	Indicates the special operation for bits 15 to 08 of the accumulator. Z: Transfers 0. X: Transfers after sign extension. -: No transfer
AH	Indicates the special operation for the 16 high-order bits of the accumulator. *: Transfers from AL to AH. -: No transfer Z: Transfers 00 to AH. X: Transfers 00H or FFH to AH after AL sign extension.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.7-1 Description of Items in the Instruction List (2/2)

Item	Description
I	Each indicates the state of each flag:I (interrupt enable), S (stack), T (sticky bit), N (negative), Z (zero), V (overflow), C (carry). *: Changes upon instruction execution. -: No change Z: Set upon instruction execution. X: Reset upon instruction execution.
S	
T	
N	
Z	
V	
C	
RMW	Indicates whether the instruction is a Read Modify Write instruction (reading data from memory by the I instruction and writing the result to memory). *: Read Modify Write instruction -: Not Read Modify Write instruction Note: Cannot be used for an address that has different meanings between read and write operations.

Table A.7-2 Explanation on Symbols in the Instruction List (1/2)

Symbol	Explanation
A	The bit length used varies depending on the 32-bit accumulator instruction. Byte: Low-order 8 bits of byte AL Word: 16 bits of word AL Long word: 32 bits of AL and AH
AH	16 high-order bits of A
AL	16 low-order bits of A
SP	Stack pointer (USP or SSP)
PC	Program counter
PCB	Program bank register
DTB	Data bank register
ADB	Additional data bank register
SSB	System stack bank register
USB	User stack bank register
SPB	Current stack bank register (SSB or USB)
DPR	Direct page register
brg1	DTB, ADB, SSB, USB, DPR, PCB, SPB
brg2	DTB, ADB, SSB, USB, DPR, SPB

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.7-2 Explanation on Symbols in the Instruction List (2/2)

Symbol	Explanation
Ri	R0, R1, R2, R3, R4, R5, R6, R7
RWi	RW0, RW1, RW2, RW3, RW4, RW5, RW6, RW7
RWj	RW0, RW1, RW2, RW3
RLi	RL0, RL1, RL2, RL3
dir	Abbreviated direct addressing
addr16	Direct addressing
addr24	Physical direct addressing
ad24 0-15	Bits 0 to 15 of addr24
ad24 16-23	Bits 16 to 23 of addr24
io	I/O area (000000_H to $0000FF_H$)
#imm4	4-bit immediate data
#imm8	8-bit immediate data
#imm16	16-bit immediate data
#imm32	32-bit immediate data
ext (imm8)	16-bit data obtained by sign extension of 8-bit immediate data
disp8	8-bit displacement
disp16	16-bit displacement
bp	Bit offset
vct4	Vector number (0 to 15)
vct8	Vector number (0 to 255)
( ) b	Bit address
rel	PC relative branch
ear	Effective addressing (code 00 to 07)
eam	Effective addressing (code 08 to 1F)
rlst	Register list

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

A.8 F²MC-16LX Instruction List

Table A.8-1 to Table A.9-19 list the instructions used by the F²MC-16LX.

■ F²MC-16LX Instruction List

Table A.8-1 41 Transfer Instructions (byte)

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOV A,dir	2	3	0	(b)	byte (A) <-- (dir)	Z	*	-	-	*	*	-	-	-	-
MOV A,addr16	3	4	0	(b)	byte (A) <-- (addr16)	Z	*	-	-	*	*	-	-	-	-
MOV A,Ri	1	2	1	0	byte (A) <-- (Ri)	Z	*	-	-	*	*	-	-	-	-
MOV A,ear	2	2	1	0	byte (A) <-- (ear)	Z	*	-	-	*	*	-	-	-	-
MOV A,eam	2+	3 + (a)	0	(b)	byte (A) <-- (eam)	Z	*	-	-	*	*	-	-	-	-
MOV A,io	2	3	0	(b)	byte (A) <-- (io)	Z	*	-	-	*	*	-	-	-	-
MOV A,#imm8	2	2	0	0	byte (A) <-- imm8	Z	*	-	-	*	*	-	-	-	-
MOV A,@A	2	3	0	(b)	byte (A) <-- ((A))	Z	-	-	-	*	*	-	-	-	-
MOV A,@RLi+disp8	3	10	2	(b)	byte (A) <-- ((RLi)+disp8)	Z	*	-	-	*	*	-	-	-	-
MOVN A,#imm4	1	1	0	0	byte (A) <-- imm4	Z	*	-	-	R	*	-	-	-	-
MOVX A,dir	2	3	0	(b)	byte (A) <-- (dir)	X	*	-	-	-	*	*	-	-	-
MOVX A,addr16	3	4	0	(b)	byte (A) <-- (addr16)	X	*	-	-	-	*	*	-	-	-
MOVX A,Ri	2	2	1	0	byte (A) <-- (Ri)	X	*	-	-	-	*	*	-	-	-
MOVX A,ear	2	2	1	0	byte (A) <-- (ear)	X	*	-	-	-	*	*	-	-	-
MOVX A,eam	2+	3 + (a)	0	(b)	byte (A) <-- (eam)	X	*	-	-	-	*	*	-	-	-
MOVX A,io	2	3	0	(b)	byte (A) <-- (io)	X	*	-	-	-	*	*	-	-	-
MOVX A,#imm8	2	2	0	0	byte (A) <-- imm8	X	*	-	-	-	*	*	-	-	-
MOVX A,@A	2	3	0	(b)	byte (A) <-- ((A))	X	-	-	-	-	*	*	-	-	-
MOVX A,@RWi+disp8	2	5	1	(b)	byte (A) <-- ((RWi)+disp8)	X	*	-	-	-	*	*	-	-	-
MOVX A,@RLi+disp8	3	10	2	(b)	byte (A) <-- ((RLi)+disp8	X	*	-	-	-	*	*	-	-	-
MOV dir,A	2	3	0	(b)	byte (dir) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOV addr16,A	3	4	0	(b)	byte (addr16) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOV Ri,A	1	2	1	0	byte (Ri) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOV ear,A	2	2	1	0	byte (ear) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOV eam,A	2+	3 + (a)	0	(b)	byte (eam) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOV io,A	2	3	0	(b)	byte (io) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOV @RLi+disp8,A	3	10	2	(b)	byte ((RLi)+disp8) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOV Ri,ear	2	3	2	0	byte (Ri) <-- (ear)	-	-	-	-	-	*	*	-	-	-
MOV Ri,eam	2+	4 + (a)	1	(b)	byte (Ri) <-- (eam)	-	-	-	-	-	*	*	-	-	-
MOV ear,Ri	2	4	2	0	byte (ear) <-- (Ri)	-	-	-	-	-	*	*	-	-	-
MOV eam,Ri	2+	5 + (a)	1	(b)	byte (eam) <-- (Ri)	-	-	-	-	-	*	*	-	-	-
MOV Ri,#imm8	2	2	1	0	byte (Ri) <-- imm8	-	-	-	-	-	*	*	-	-	-
MOV io,#imm8	3	5	0	(b)	byte (io) <-- imm8	-	-	-	-	-	-	-	-	-	-
MOV dir,#imm8	3	5	0	(b)	byte (dir) <-- imm8	-	-	-	-	-	-	-	-	-	-
MOV ear,#imm8	3	2	1	0	byte (ear) <-- imm8	-	-	-	-	-	*	*	-	-	-
MOV eam,#imm8	3+	4 + (a)	0	(b)	byte (eam) <-- imm8	-	-	-	-	-	-	-	-	-	-
MOV @AL,AH / MOV @A,T	2	3	0	(b)	byte ((A)) <-- (AH)	-	-	-	-	-	*	*	-	-	-
XCH A,ear	2	4	2	0	byte (A) <--> (ear)	Z	-	-	-	-	-	-	-	-	-
XCH A,eam	2+	5 + (a)	0	2 x (b)	byte (A) <--> (eam)	Z	-	-	-	-	-	-	-	-	-
XCH Ri,ear	2	7	4	0	byte (Ri) <--> (ear)	-	-	-	-	-	-	-	-	-	-
XCH Ri,eam	2+	9 + (a)	2	2 x (b)	byte (Ri) <--> (eam)	-	-	-	-	-	-	-	-	-	-

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-2 38 Transfer Instructions (byte)

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVW A,dir	2	3	0	(c)	word (A) <-- (dir)	-	*	-	-	-	*	*	-	-	-
MOVW A,addr16	3	4	0	(c)	word (A) <-- (addr16)	-	*	-	-	-	*	*	-	-	-
MOVW A,SP	3	1	0	0	word (A) <-- (SP)	-	*	-	-	-	*	*	-	-	-
MOVW A,RWi	1	2	1	0	word (A) <-- (RWi)	-	*	-	-	-	*	*	-	-	-
MOVW A,ear	2	2	1	0	word (A) <-- (ear)	-	*	-	-	-	*	*	-	-	-
MOVW A,eam	2+	3 + (a)	0	(c)	word (A) <-- (eam)	-	*	-	-	-	*	*	-	-	-
MOVW A,io	2	3	0	(c)	word (A) <-- (io)	-	*	-	-	-	*	*	-	-	-
MOVW A,@A	2	3	0	(c)	word (A) <-- ((A))	-	-	-	-	-	*	*	-	-	-
MOVW A,#imm16	3	2	2	0	word (A) <-- imm16	-	*	-	-	-	*	*	-	-	-
MOVW A,@RWi+disp8	2	5	1	(c)	word (A) <-- ((RWi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW A,@RLi+disp8	3	10	2	(c)	word (A) <-- ((RLi)+disp8)	-	*	-	-	-	*	*	-	-	-
MOVW dir,A	2	3	0	(c)	word (dir) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOVW addr16,A	3	4	0	(c)	word (addr16) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOVW SP,A	1	1	0	0	word (SP) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,A	1	2	1	0	word (RWi) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOVW ear,A	2	2	1	0	word (ear) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOVW eam,A	2+	3 + (a)	0	(c)	word (eam) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOVW io,A	2	3	0	(c)	word (io) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOVW @RWi+disp8,A	2	5	1	(c)	word ((RWi)+disp8) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOVW @RLi+disp8,A	3	10	2	(c)	word ((RLi)+disp8) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,ear	2	3	2	0	word (RWi) <-- (ear)	-	-	-	-	-	*	*	-	-	-
MOVW	2+	4 + (a)	1	(c)	word (RWi) <-- (eam)	-	-	-	-	-	*	*	-	-	-
MOVW ear,Rwi	2	4	2	0	word (ear) <-- (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW eam,Rwi	2+	5 + (a)	1	(c)	word (eam) <-- (RWi)	-	-	-	-	-	*	*	-	-	-
MOVW RWi,#imm16	3	2	1	0	word (RWi) <-- imm16	-	-	-	-	-	*	*	-	-	-
MOVW io,#imm16	4	5	0	(c)	word (io) <-- imm16	-	-	-	-	-	-	-	-	-	-
MOVW ear,#imm16	4	2	1	0	word (ear) <-- imm16	-	-	-	-	-	*	*	-	-	-
MOVW eam,#imm16	4+	4 + (a)	0	(c)	word (eam) <-- imm16	-	-	-	-	-	-	-	-	-	-
MOVW @AL,AH / MOVW @A,T	2	3	0	(c)	word ((A)) <-- (AH)	-	-	-	-	-	*	*	-	-	-
XCHW A,ear	2	4	2	0	word (A) <--> (ear)	-	-	-	-	-	-	-	-	-	-
XCHW A,eam	2+	5 + (a)	0	2 x (c)	word (A) <-- >(eam)	-	-	-	-	-	-	-	-	-	-
XCHW RWi, ear	2	7	4	0	word (RWi) <--> (ear)	-	-	-	-	-	-	-	-	-	-
XCHW RWi, eam	2+	9 + (a)	2	2 x (c)	word (RWi) <--> (eam)	-	-	-	-	-	-	-	-	-	-
MOVL A,ear	2	4	2	0	long (A) <-- (ear)	-	-	-	-	-	*	*	-	-	-
MOVL A,eam	2+	5 + (a)	0	(d)	long (A) <-- (eam)	-	-	-	-	-	*	*	-	-	-
MOVL A,#imm32	5	3	0	0	long (A) <-- imm32	-	-	-	-	-	*	*	-	-	-
MOVL ear,A	2	4	2	0	long (ear1) <-- (A)	-	-	-	-	-	*	*	-	-	-
MOVL eam,A	2+	5 + (a)	0	(d)	long(eam1) <-- (A)	-	-	-	-	-	*	*	-	-	-

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-3 42 Addition/subtraction Instructions (byte, word, long word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
ADD	A,#imm8	2	2	0	0	byte (A) <-- (A) + imm8	Z	-	-	-	-	*	*	*	*	-
ADD	A,dir	2	5	0	(b)	byte (A) <-- (A) + (dir)	Z	-	-	-	-	*	*	*	*	-
ADD	A,ear	2	3	1	0	byte (A) <-- (A) + (ear)	Z	-	-	-	-	*	*	*	*	-
ADD	A,eam	2+	4 + (a)	0	(b)	byte (A) <-- (A) + (eam)	Z	-	-	-	-	*	*	*	*	-
ADD	ear,A	2	3	2	0	byte (ear) <-- (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADD	eam,A	2+	5 + (a)	0	2 x (b)	byte (eam) <-- (eam) + (A)	Z	-	-	-	-	*	*	*	*	*
ADDC	A	1	2	0	0	byte (A) <-- (AH) + (AL) + (C)	Z	-	-	-	-	*	*	*	*	-
ADDC	A,ear	2	3	1	0	byte (A) <-- (A) + (ear)+ (C)	Z	-	-	-	-	*	*	*	*	-
ADDC	A,eam	2+	4 + (a)	0	(b)	byte (A) <-- (A) + (eam)+ (C)	Z	-	-	-	-	*	*	*	*	-
ADDDC	A	1	3	0	0	byte (A) <-- (AH) + (AL) + (C) (decimal)	Z	-	-	-	-	*	*	*	*	-
SUB	A,#imm8	2	2	0	0	byte (A) <-- (A) - imm8	Z	-	-	-	-	*	*	*	*	-
SUB	A,dir	2	5	0	(b)	byte (A) <-- (A) - (dir)	Z	-	-	-	-	*	*	*	*	-
SUB	A,ear	2	3	1	0	byte (A) <-- (A) - (ear)	Z	-	-	-	-	*	*	*	*	-
SUB	A,eam	2+	4 + (a)	0	(b)	byte (A) <-- (A) - (eam)	Z	-	-	-	-	*	*	*	*	-
SUB	ear,A	2	3	2	0	byte (ear) <-- (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUB	eam,A	2+	5 + (a)	0	2 x (b)	byte (eam) <-- (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBC	A	1	2	0	0	byte (A) <-- (AH) - (AL) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC	A,ear	2	3	1	0	byte (A) <-- (A) - (ear) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBC	A,eam	2+	4 + (a)	0	(b)	byte (A) <-- (A) - (eam) - (C)	Z	-	-	-	-	*	*	*	*	-
SUBDC	A	1	3	0	0	byte (A) <-- (AH) - (AL) - (C) (decimal)	Z	-	-	-	-	*	*	*	*	-
ADDW	A	1	2	0	0	word (A) <-- (AH) + (AL)	-	-	-	-	-	*	*	*	*	-
ADDW	A,ear	2	3	1	0	word (A) <-- (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDW	A,eam	2+	4+(a)	0	(c)	word (A) <-- (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDW	A,#imm16	3	2	0	0	word (A) <-- (A) + imm16	-	-	-	-	-	*	*	*	*	-
ADDW	ear,A	2	3	2	0	word (ear) <-- (ear) + (A)	-	-	-	-	-	*	*	*	*	-
ADDW	eam,A	2+	5+(a)	0	2 x (c)	word (eam) <-- (eam) + (A)	-	-	-	-	-	*	*	*	*	*
ADDCW	A,ear	2	3	1	0	word (A) <-- (A) + (ear) + (C)	-	-	-	-	-	*	*	*	*	-
ADDCW	A,eam	2+	4+(a)	0	(c)	word (A) <-- (A) + (eam) + (C)	-	-	-	-	-	*	*	*	*	-
SUBW	A	1	2	0	0	word (A) <-- (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
SUBW	A,ear	2	3	1	0	word (A) <-- (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBW	A,eam	2+	4+(a)	0	(c)	word (A) <-- (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBW	A,#imm16	3	2	0	0	word (A) <-- (A) - imm16	-	-	-	-	-	*	*	*	*	-
SUBW	ear,A	2	3	2	0	word (ear) <-- (ear) - (A)	-	-	-	-	-	*	*	*	*	-
SUBW	eam,A	2+	5+(a)	0	2 x (c)	word (eam) <-- (eam) - (A)	-	-	-	-	-	*	*	*	*	*
SUBCW	A,ear	2	3	1	0	word (A) <-- (A) - (ear) - (C)	-	-	-	-	-	*	*	*	*	-
SUBCW	A,eam	2+	4+(a)	0	(c)	word (A) <-- (A) - (eam) - (C)	-	-	-	-	-	*	*	*	*	-
ADDL	A,ear	2	6	2	0	long (A) <-- (A) + (ear)	-	-	-	-	-	*	*	*	*	-
ADDL	A,eam	2+	7+(a)	0	(d)	long (A) <-- (A) + (eam)	-	-	-	-	-	*	*	*	*	-
ADDL	A,#imm32	5	4	0	0	long (A) <-- (A) + imm32	-	-	-	-	-	*	*	*	*	-
SUBL	A,ear	2	6	2	0	long (A) <-- (A) - (ear)	-	-	-	-	-	*	*	*	*	-
SUBL	A,eam	2+	7+(a)	0	(d)	long (A) <-- (A) - (eam)	-	-	-	-	-	*	*	*	*	-
SUBL	A,#imm32	5	4	0	0	long (A) <-- (A) - imm32	-	-	-	-	-	*	*	*	*	-

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-4 12 Increment/decrement Instructions (byte, word, long word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
INC	ear	2	3	2	0	byte (ear) <-- (ear) + 1	-	-	-	-	-	*	*	*	-	-
INC	eam	2+	5+(a)	0	2 x (b)	byte (eam) <-- (eam) + 1	-	-	-	-	-	*	*	*	-	*
DEC	ear	2	3	2	0	byte (ear) <-- (ear) - 1	-	-	-	-	-	*	*	*	-	-
DEC	eam	2+	5+(a)	0	2 x (b)	byte (eam) <-- (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCW	ear	2	3	2	0	word (ear) <-- (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCW	eam	2+	5+(a)	0	2 x (c)	word (eam) <-- (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECW	ear	2	3	2	0	word (ear) <-- (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECW	eam	2+	5+(a)	0	2 x (c)	word (eam) <-- (eam) - 1	-	-	-	-	-	*	*	*	-	*
INCL	ear	2	7	4	0	long (ear) <-- (ear) + 1	-	-	-	-	-	*	*	*	-	-
INCL	eam	2+	9+(a)	0	2 x (d)	long (eam) <-- (eam) + 1	-	-	-	-	-	*	*	*	-	*
DECL	ear	2	7	4	0	long (ear) <-- (ear) - 1	-	-	-	-	-	*	*	*	-	-
DECL	eam	2+	9+(a)	0	2 x (d)	long (eam) <-- (eam) - 1	-	-	-	-	-	*	*	*	-	*

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

Table A.8-5 11 Compare Instructions (byte, word, long word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
CMP	A	1	1	0	0	byte (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
CMP	A,ear	2	2	1	0	byte (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMP	A,eam	2+	3+(a)	0	(b)	byte (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMP	A,#imm8	2	2	0	0	byte (A) - imm8	-	-	-	-	-	*	*	*	*	-
CMPW	A	1	1	0	0	word (AH) - (AL)	-	-	-	-	-	*	*	*	*	-
CMPW	A,ear	2	2	1	0	word (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMPW	A,eam	2+	3+(a)	0	(c)	word (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMPW	A,#imm16	3	2	0	0	word (A) - imm16	-	-	-	-	-	*	*	*	*	-
CMPL	A,ear	2	6	2	0	long (A) - (ear)	-	-	-	-	-	*	*	*	*	-
CMPL	A,eam	2+	7+(a)	0	(d)	long (A) - (eam)	-	-	-	-	-	*	*	*	*	-
CMPL	A,#imm32	5	3	0	0	long (A) - imm32	-	-	-	-	-	*	*	*	*	-

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-6 11 Unsigned multiplication/division instructions (word, long word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
DIVU	A	1	*1	0	0	word (AH) / byte (AL) quotient --> byte (AL) remainder --> byte (AH)	-	-	-	-	-	-	-	*	*	-
DIVU	A,ear	2	*2	1	0	word (A) / byte (ear) quotient --> byte (A) remainder --> byte (ear)	-	-	-	-	-	-	-	*	*	-
DIVU	A,eam	2+	*3	0	*6	word (A) / byte (eam) quotient --> byte (A) remainder --> byte (eam)	-	-	-	-	-	-	-	*	*	-
DIVUW	A,ear	2	*4	1	0	long (A) / word (ear) quotient --> word(A) remainder --> word(ear)	-	-	-	-	-	-	-	*	*	-
DIVUW	A,eam	2+	*5	0	*7	long (A) / word (eam) quotient --> word(A) remainder --> word(eam)	-	-	-	-	-	-	-	*	*	-
MULU	A	1	*8	0	0	byte (AH) * byte (AL) --> word (A)	-	-	-	-	-	-	-	-	-	-
MULU	A,ear	2	*9	1	0	byte (A) * byte (ear) --> word (A)	-	-	-	-	-	-	-	-	-	-
MULU	A,eam	2+	*10	0	(b)	byte (A) * byte (eam) --> word (A)	-	-	-	-	-	-	-	-	-	-
MULUW	A	1	*11	0	0	word (AH) * word (AL) --> Long (A)	-	-	-	-	-	-	-	-	-	-
MULUW	A,ear	2	*12	1	0	word (A) * word (ear) --> Long (A)	-	-	-	-	-	-	-	-	-	-
MULUW	A,eam	2+	*13	0	(c)	word (A) * word (eam) --> Long (A)	-	-	-	-	-	-	-	-	-	-

*1: 3: Division by 0 7: Overflow 15: Normal
*2: 4: Division by 0 8: Overflow 16: Normal
*3: 6+(a): Division by 0 9+(a): Overflow 19+(a): Normal
*4: 4: Division by 0 7: Overflow 22: Normal
*5: 6+(a): Division by 0 8+(a): Overflow 26+(a): Normal
*6: (b): Division by 0 or overflow 2 x (b): Normal
*7: (c): Division by 0 or overflow 2 x (c): Normal
*8: 3: Byte (AH) is 0. 7: Byte (AH) is not 0.
*9: 4: Byte (ear) is 0. 8: Byte (ear) is not 0.
*10: 5+(a): Byte (eam) is 0, 9+(a): Byte (eam) is not 0.
*11: 3: Word(AH) is 0. 11: Word (AH) is not 0.
*12: 4: Word(ear) is 0. 12: Word (ear) is not 0.
*13: 5+(a): Word (eam) is 0. 13+(a): Word (eam) is not 0.

et4U.com
DataShee
DataSheet4U.com

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-7 11 Signed Multiplication/Division Instructions (word, long word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
DIV	A	2	*1	0	0	word (AH) / byte (AL) quotient --> byte (AL) remainder --> byte (AH)	Z	-	-	-	-	-	-	*	*	-
DIV	A,ear	2	*2	1	0	word (A) / byte (ear) quotient --> byte (A) remainder --> byte (ear)	Z	-	-	-	-	-	-	*	*	-
DIV	A,eam	2+	*3	0	*6	word (A) / byte (eam) quotient --> byte (A) remainder --> byte (eam)	Z	-	-	-	-	-	-	*	*	-
DIVW	A,ear	2	*4	1	0	long (A) / word (ear) quotient --> word(A) remainder --> word(ear)	-	-	-	-	-	-	-	*	*	-
DIVW	A,eam	2+	*5	0	*7	long (A) / word (eam) quotient --> word(A) remainder --> word(eam)	-	-	-	-	-	-	-	*	*	-
MUL	A	2	*8	0	0	byte (AH) * byte (AL) --> word (A)	-	-	-	-	-	-	-	-	-	-
MUL	A,ear	2	*9	1	0	byte (A) * byte (ear) --> word (A)	-	-	-	-	-	-	-	-	-	-
MUL	A,eam	2+	*10	0	(b)	byte (A) * byte (eam) --> word (A)	-	-	-	-	-	-	-	-	-	-
MULW	A	2	*11	0	0	word (AH) * word (AL) --> Long (A)	-	-	-	-	-	-	-	-	-	-
MULW	A,ear	2	*12	1	0	word (A) * word (ear) --> Long (A)	-	-	-	-	-	-	-	-	-	-
MULW	A,eam	2+	*13	0	(c)	word (A) * word (eam) --> Long (A)	-	-	-	-	-	-	-	-	-	-

*1: 3: Division by 0, 8 or 18: Overflow, 18: Normal
*2: 4: Division by 0, 11 or 22: Overflow, 23: Normal
*3: 5+(a): Division by 0, 12+(a) or 23+(a): Overflow, 24+(a): Normal
*4: When dividend is positive; 4: Division by 0, 12 or 30: Overflow, 31: Normal
When dividend is negative; 4: Division by 0, 12 or 31: Overflow, 32: Normal
*5: When dividend is positive; 5+(a): Division by 0, 12+(a) or 31+(a): Overflow, 32+(a): Normal
When dividend is negative; 5+(a): Division by 0, 12+(a) or 32+(a): Overflow, 33+(a): Normal
*6: (b): Division by 0 or overflow, 2 x (b): Normal
*7: (c): Division by 0 or overflow, 2 x (c): Normal
*8: 3: Byte (AH) is 0, 12: result is positive, 13: result is negative
*9: 4: Byte (ear) is 0, 13: result is positive, 14: result is negative
*10: 5+(a): Byte (eam) is 0, 14+(a): result is positive, 15+(a): result is negative
*11: 3: Word(AH) is 0, 16: result is positive, 19: result is negative
*12: 4: Word(ear) is 0, 17: result is positive, 20: result is negative
*13: 5+(a): Word(eam) is 0, 18+(a): result is positive, 21+(a): result is negative

Notes:
- The execution cycle count found when an overflow occurs in a DIV or DIVW instruction may be a pre-operation count or a post-operation count depending on the detection timing.
- When an overflow occurs with DIV or DIVW instruction, the contents of the AL are destroyed.
- See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-8 39 Logic 1 Instructions (byte, word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
AND	A,#imm8	2	2	0	0	byte (A) <-- (A) and imm8	-	-	-	-	-	*	*	R	-	-
AND	A,ear	2	3	1	0	byte (A) <-- (A) and (ear)	-	-	-	-	-	*	*	R	-	-
AND	A,eam	2+	4+(a)	0	(b)	byte (A) <-- (A) and (eam)	-	-	-	-	-	*	*	R	-	-
AND	ear,A	2	3	2	0	byte (ear) <-- (ear) and (A)	-	-	-	-	-	*	*	R	-	-
AND	eam,A	2+	5+(a)	0	2 x (b)	byte (eam) <-- (eam) and (A)	-	-	-	-	-	*	*	R	-	*
OR	A,#imm8	2	2	0	0	byte (A) <-- (A) or imm8	-	-	-	-	-	*	*	R	-	-
OR	A,ear	2	3	1	0	byte (A) <-- (A) or (ear)	-	-	-	-	-	*	*	R	-	-
OR	A,eam	2+	4+(a)	0	(b)	byte (A) <-- (A) or (eam)	-	-	-	-	-	*	*	R	-	-
OR	ear,A	2	3	2	0	byte (ear) <-- (ear) or (A)	-	-	-	-	-	*	*	R	-	-
OR	eam,A	2+	5+(a)	0	2 x (b)	byte (eam) <-- (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XOR	A,#imm8	2	2	0	0	byte (A) <-- (A) xor imm8	-	-	-	-	-	*	*	R	-	-
XOR	A,ear	2	3	1	0	byte (A) <-- (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XOR	A,eam	2+	4+(a)	0	(b)	byte (A) <-- (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XOR	ear,A	2	3	2	0	byte (ear) <-- (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XOR	eam,A	2+	5+(a)	0	2 x (b)	byte (eam) <-- (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOT	A	1	2	0	0	byte (A) <-- not (A)	-	-	-	-	-	*	*	R	-	-
NOT	ear	2	3	2	0	byte (ear) <-- not (ear)	-	-	-	-	-	*	*	R	-	-
NOT	eam	2+	5+(a)	0	2 x (b)	byte (eam) <-- not (eam)	-	-	-	-	-	*	*	R	-	*
ANDW	A	1	2	0	0	word (A) <-- (AH) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW	A,#imm16	3	2	0	0	word (A) <-- (A) and imm16	-	-	-	-	-	*	*	R	-	-
ANDW	A,ear	2	3	1	0	word (A) <-- (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDW	A,eam	2+	4+(a)	0	(c)	word (A) <-- (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ANDW	ear,A	2	3	2	0	word (ear) <-- (ear) and (A)	-	-	-	-	-	*	*	R	-	-
ANDW	eam,A	2+	5+(a)	0	2 x (c)	word (eam) <-- (eam) and (A)	-	-	-	-	-	*	*	R	-	*
ORW	A	1	2	0	0	word (A) <-- (AH) or (A)	-	-	-	-	-	*	*	R	-	-
ORW	A,#imm16	3	2	0	0	word (A) <-- (A) or imm16	-	-	-	-	-	*	*	R	-	-
ORW	A,ear	2	3	1	0	word (A) <-- (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORW	A,eam	2+	4+(a)	0	(c)	word (A) <-- (A) or (eam)	-	-	-	-	-	*	*	R	-	-
ORW	ear,A	2	3	2	0	word (ear) <-- (ear) or (A)	-	-	-	-	-	*	*	R	-	-
ORW	eam,A	2+	5+(a)	0	2 x (c)	word (eam) <-- (eam) or (A)	-	-	-	-	-	*	*	R	-	*
XORW	A	1	2	0	0	word (A) <-- (AH) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW	A,#imm16	3	2	0	0	word (A) <-- (A) xor imm16	-	-	-	-	-	*	*	R	-	-
XORW	A,ear	2	3	1	0	word (A) <-- (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORW	A,eam	2+	4+(a)	0	(c)	word (A) <-- (A) xor (eam)	-	-	-	-	-	*	*	R	-	-
XORW	ear,A	2	3	2	0	word (ear) <-- (ear) xor (A)	-	-	-	-	-	*	*	R	-	-
XORW	eam,A	2+	5+(a)	0	2 x (c)	word (eam) <-- (eam) xor (A)	-	-	-	-	-	*	*	R	-	*
NOTW	A	1	2	0	0	word (A) <-- not (A)	-	-	-	-	-	*	*	R	-	-
NOTW	ear	2	3	2	0	word (ear) <-- not (ear)	-	-	-	-	-	*	*	R	-	-
NOTW	eam	2+	5+(a)	0	2 x (c)	word (eam) <-- not (eam)	-	-	-	-	-	*	*	R	-	*

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-9 Six Logic 2 Instructions (long word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
ANDL	A,ear	2	6	2	0	long (A) <-- (A) and (ear)	-	-	-	-	-	*	*	R	-	-
ANDL	A,eam	2+	7+(a)	0	(d)	long (A) <-- (A) and (eam)	-	-	-	-	-	*	*	R	-	-
ORL	A,ear	2	6	2	0	long (A) <-- (A) or (ear)	-	-	-	-	-	*	*	R	-	-
ORL	A,eam	2+	7+(a)	0	(d)	long (A) <-- (A) or (eam)	-	-	-	-	-	*	*	R	-	-
XORL	A,ear	2	6	2	0	long (A) <-- (A) xor (ear)	-	-	-	-	-	*	*	R	-	-
XORL	A,eam	2+	7+(a)	0	(d)	long (A) <-- (A) xor (eam)	-	-	-	-	-	*	*	R	-	-

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-10 Six Sign Inversion Instructions (byte, word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
NEG	A	1	2	0	0	byte (A) <-- 0 - (A)	X	-	-	-	-	*	*	*	*	-
NEG	ear	2	3	2	0	byte (ear) <-- 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEG	eam	2+	5+(a)	0	2 x (b)	byte (eam) <-- 0 - (eam)	-	-	-	-	-	*	*	*	*	*
NEGW	A	1	2	0	0	word (A) <-- 0 - (A)	-	-	-	-	-	*	*	*	*	-
NEGW	ear	2	3	2	0	word (ear) <-- 0 - (ear)	-	-	-	-	-	*	*	*	*	-
NEGW	eam	2+	5+(a)	0	2 x (c)	word (eam) <-- 0 - (eam)	-	-	-	-	-	*	*	*	*	*

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

Table A.8-11 One Normalization Instruction (long word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
NRML	A,R0	2	*1	1	0	long (A) <-- Shifts to the position where '1' is set for the first time. byte (RD) <-- Shift count at that time	-	-	-	-	-	-	*	-	-	-

*1: 4 when all accumulators have a value of 0; otherwise, 6+(R0)

et4U.com DataShee
DataSheet4U.com
DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-12 18 Shift Instructions (byte, word, long word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
RORC	A	2	2	0	0	byte (A) <-- With right rotation carry	-	-	-	-	-	*	*	-	*	-
ROLC	A	2	2	0	0	byte (A) <-- With left rotation carry	-	-	-	-	-	*	*	-	*	-
RORC	ear	2	3	2	0	byte (ear) <-- With right rotation carry	-	-	-	-	-	*	*	-	*	-
RORC	eam	2+	5+(a)	0	2 x (b)	byte (eam) <-- With right rotation carry	-	-	-	-	-	*	*	-	*	*
ROLC	ear	2	3	2	0	byte (ear) <-- With left rotation carry	-	-	-	-	-	*	*	-	*	-
ROLC	eam	2+	5+(a)	0	2 x (b)	byte (eam) <-- With left rotation carry	-	-	-	-	-	*	*	-	*	*
ASR	A,R0	2	*1	1	0	byte (A) <-- Arithmetic right shift (A, 1 bit)	-	-	-	-	-	*	*	-	*	-
LSR	A,R0	2	*1	1	0	byte (A) <-- Logical right barrel shift (A, R0)	-	-	-	-	-	*	*	-	*	-
LSL	A,R0	2	*1	1	0	byte (A) <-- Logical left barrel shift (A, R0)	-	-	-	-	-	*	*	-	*	-
ASRW	A	1	2	0	0	word (A) <-- Arithmetic right shift (A, 1 bit)	-	-	-	-	*	*	*	-	*	-
LSRW	A/SHRW A	1	2	0	0	word (A) <-- Logical right shift (A, 1 bit)	-	-	-	-	*	R	*	-	*	-
LSLW	A/SHLW A	1	2	0	0	word (A) <-- Logical left shift (A, 1 bit)	-	-	-	-	-	*	*	-	*	-
ASRW	A,R0	2	*1	1	0	word (A) <-- Arithmetic right barrel shift (A, R0)	-	-	-	-	*	*	*	-	*	-
LSRW	A,R0	2	*1	1	0	word (A) <-- Logical right barrel shift (A, R0)	-	-	-	-	*	*	*	-	*	-
LSLW	A,R0	2	*1	1	0	word (A) <-- Logical left barrel shift (A, R0)	-	-	-	-	-	*	*	-	*	-
ASRL	A,R0	2	*2	1	0	long (A) <-- Arithmetic right barrel shift (A, R0)	-	-	-	-	*	*	*	-	*	-
LSRL	A,R0	2	*2	1	0	long (A) <-- Logical right barrel shift (A, R0)	-	-	-	-	*	*	*	-	*	-
LSLL	A,R0	2	*2	1	0	long (A) <-- Logical left barrel shift (A, R0)	-	-	-	-	-	*	*	-	*	-

*1: 6 when R0 is 0; otherwise, 5 + (R0)
*2: 6 when R0 is 0; otherwise, 6 + (R0)

DataSheet4U.com

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-13 31 Branch 1 Instructions

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
BZ/BEQ	rel	2	*1	0	0	Branch on (Z) = 1	-	-	-	-	-	-	-	-	-	-
BNZ/BNE	rel	2	*1	0	0	Branch on (Z) = 0	-	-	-	-	-	-	-	-	-	-
BC/BLO	rel	2	*1	0	0	Branch on (C) = 1	-	-	-	-	-	-	-	-	-	-
BNC/BHS	rel	2	*1	0	0	Branch on (C) = 0	-	-	-	-	-	-	-	-	-	-
BN	rel	2	*1	0	0	Branch on (N) = 1	-	-	-	-	-	-	-	-	-	-
BP	rel	2	*1	0	0	Branch on (N) = 0	-	-	-	-	-	-	-	-	-	-
BV	rel	2	*1	0	0	Branch on (V) = 1	-	-	-	-	-	-	-	-	-	-
BNV	rel	2	*1	0	0	Branch on (V) = 0	-	-	-	-	-	-	-	-	-	-
BT	rel	2	*1	0	0	Branch on (T) = 1	-	-	-	-	-	-	-	-	-	-
BNT	rel	2	*1	0	0	Branch on (T) = 0	-	-	-	-	-	-	-	-	-	-
BLT	rel	2	*1	0	0	Branch on (V) nor (N) = 1	-	-	-	-	-	-	-	-	-	-
BGE	rel	2	*1	0	0	Branch on (V) nor (N) = 0	-	-	-	-	-	-	-	-	-	-
BLE	rel	2	*1	0	0	Branch on ((V) xor (N)) or (Z) = 1	-	-	-	-	-	-	-	-	-	-
BGT	rel	2	*1	0	0	Branch on ((V) xor (N)) or (Z) = 0	-	-	-	-	-	-	-	-	-	-
BLS	rel	2	*1	0	0	Branch on (C) or (Z) = 1	-	-	-	-	-	-	-	-	-	-
BHI	rel	2	*1	0	0	Branch on (C) or (Z) = 0	-	-	-	-	-	-	-	-	-	-
BRA	rel	2	*1	0	0	Unconditional branch	-	-	-	-	-	-	-	-	-	-
JMP	@A	1	2	0	0	word (PC) <-- (A)	-	-	-	-	-	-	-	-	-	-
JMP	addr16	3	3	0	0	word (PC) <-- addr16	-	-	-	-	-	-	-	-	-	-
JMP	@ear	2	3	1	0	word (PC) <-- (ear)	-	-	-	-	-	-	-	-	-	-
JMP	@eam	2+	4+(a)	0	(c)	word (PC) <-- (eam)	-	-	-	-	-	-	-	-	-	-
JMPP	@ear *3	2	5	2	0	word (PC) <-- (ear), (PCB) <-- (ear+2)	-	-	-	-	-	-	-	-	-	-
JMPP	@eam *3	2+	6+(a)	0	(d)	word (PC) <-- (eam), (PCB) <-- (eam+2)	-	-	-	-	-	-	-	-	-	-
JMPP	addr24	4	4	0	0	word(PC) <-- ad24 0-15,(PCB) <-- ad24 16-23	-	-	-	-	-	-	-	-	-	-
CALL	@ear *4	2	6	1	(c)	word (PC) <-- (ear)	-	-	-	-	-	-	-	-	-	-
CALL	addr16 *5	2+	7+(a)	0	2 x (c)	word (PC) <-- (eam)	-	-	-	-	-	-	-	-	-	-
CALL	@eam *4	3	6	0	(c)	word (PC) <-- addr16	-	-	-	-	-	-	-	-	-	-
CALLV	#vct4 *5	1	7	0	2 x (c)	Vector call instruction	-	-	-	-	-	-	-	-	-	-
CALLP	@ear *6	2	10	2	2 x (c)	word(PC) <-- (ear)0-15,(PCB) <-- (ear)16-23	-	-	-	-	-	-	-	-	-	-
CALLP	@eam *6	2+	11+(a)	0	*2	word(PC) <-- (eam)0-15,(PCB) <-- (eam)16-23	-	-	-	-	-	-	-	-	-	-
CALLP	addr24 *7	4	10	0	2 x (c)	word(PC) <-- addr0-15, (PCB) <-- addr16-23	-	-	-	-	-	-	-	-	-	-

*1: 4 when a branch is made; otherwise, 3
*2: 3 x (c) + (b)
*3: Read (word) of branch destination address
*4: W: Save to stack (word) R: Read (word) of branch destination address
*5: Save to stack (word)
*6: W: Save to stack (long word), R: Read (long word) of branch destination address
*7: Save to stack (long word)

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-14 19 Branch 2 Instructions

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
CBNE	A,#imm8,rel	3	*1	0	0	Branch on byte (A) not equal to imm8	-	-	-	-	-	*	*	*	*	-
CWBNE	A,#imm16,rel	4	*1	0	0	Branch on word (A) not equal to imm16	-	-	-	-	-	*	*	*	*	-
CBNE	ear,#imm8,rel	4	*2	1	0	Branch on byte (ear) not equal to imm8	-	-	-	-	-	*	*	*	*	-
CBNE	eam,#imm8,rel *9	4+	*3	0	(b)	Branch on byte (eam) not equal to imm8	-	-	-	-	-	*	*	*	*	-
CWBNE	ear,#imm16,rel	5	*4	1	0	Branch on word (ear) not equal to imm16	-	-	-	-	-	*	*	*	*	-
CWBNE	eam,#imm16,rel*9	5+	*3	0	(c)	Branch on word (eam) not equal to imm16	-	-	-	-	-	*	*	*	*	-
DBNZ	ear,rel	3	*5	2	0	Branch on byte (ear) = (ear) - 1, (ear)not equal to 0	-	-	-	-	-	*	*	*	-	-
DBNZ	eam,rel	3+	*6	2	2 x (b)	Branch on byte (eam) = (eam) - 1, (eam) not equal to 0	-	-	-	-	-	*	*	*	-	*
DWBNZ	ear,rel	3	*5	2	0	Branch on word (ear) = (ear) - 1, (ear) not equal to 0	-	-	-	-	-	*	*	*	-	-
DWBNZ	eam,rel	3+	*6	2	2 x (c)	Branch on word (eam) = (eam) - 1, (eam) not equal to 0	-	-	-	-	-	*	*	*	-	*
INT	#vct8	2	20	0	8 x (c)	Software interrupt	-	-	R	S	-	-	-	-	-	-
INT	addr16	3	16	0	6 x (c)	Software interrupt	-	-	R	S	-	-	-	-	-	-
INTP	addr24	4	17	0	6 x (c)	Software interrupt	-	-	R	S	-	-	-	-	-	-
INT9		1	20	0	8 x (c)	Software interrupt	-	-	R	S	-	-	-	-	-	-
RETI		1	*8	0	*7	Return from interrupt	-	-	*	*	*	*	*	*	*	-
LINK	#imm8	2	6	0	(c)	Saves the old frame pointer in the stack upon entering the function, then sets the new frame pointer and reserves the local pointer area.	-	-	-	-	-	-	-	-	-	-
UNLINK		1	5	0	(c)	Recovers the old frame pointer from the stack upon exiting the function.	-	-	-	-	-	-	-	-	-	-
RET	*10	1	4	0	(c)	Return from subroutine	-	-	-	-	-	-	-	-	-	-
RETP	*11	1	6	0	(d)	Return from subroutine	-	-	-	-	-	-	-	-	-	-

*1: 5 when a branch is made; otherwise, 4
*2: 13 when a branch is made; otherwise, 12
*3: 7+(a) when a branch is made; otherwise, 6+(a)
*4: 8 when a branch is made; otherwise, 7
*5: 7 when a branch is made; otherwise, 6
*6: 8+(a) when a branch is made; otherwise, 7+(a)
*7: 3 x (b) + 2 x (c) when jumping to the next interruption request; 6 x (c) when returning from the current interruption
*8: 15 when jumping to the next interruption request; 17 when returning from the current interruption
*9: Do not use RWj+ addressing mode with a CBNE or CWBNE instruction.
*10: Return from stack (word)
*11: Return from stack (long word)

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-15 28 Other Control Instructions (byte, word, long word)

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
PUSHW	A	1	4	0	(c)	word (SP) <-- (SP) - 2 , ((SP)) <-- (A)	-	-	-	-	-	-	-	-	-	-
PUSHW	AH	1	4	0	(c)	word (SP) <-- (SP) - 2 , ((SP)) <-- (AH)	-	-	-	-	-	-	-	-	-	-
PUSHW	PS	1	4	0	(c)	word (SP) <-- (SP) - 2 , ((SP)) <-- (PS)	-	-	-	-	-	-	-	-	-	-
PUSHW	rlst	2	*3	*5	*4	(SP) <-- (SP) - 2n , ((SP)) <-- (rlst)	-	-	-	-	-	-	-	-	-	-
POPW	A	1	3	0	(c)	word (A) <-- ((SP)) , (SP) <-- (SP) + 2	-	*	-	-	-	-	-	-	-	-
POPW	AH	1	3	0	(c)	word (AH) <-- ((SP)) , (SP) <-- (SP) + 2	-	-	-	-	-	-	-	-	-	-
POPW	PS	1	4	0	(c)	word (PS) <-- ((SP)) , (SP) <-- (SP) + 2	-	-	*	*	*	*	*	*	*	-
POPW	rlst	2	*2	*5	*4	(rlst) <-- ((SP)) , (SP) <-- (SP)	-	-	-	-	-	-	-	-	-	-
JCTX	@A	1	14	0	6 x (c)	Context switch instruction	-	-	*	*	*	*	*	*	*	-
AND	CCR,#imm8	2	3	0	0	byte (CCR) <-- (CCR) and imm8	-	-	*	*	*	*	*	*	*	-
OR	CCR,#imm8	2	3	0	0	byte(CCR) <-- (CCR) or imm8	-	-	*	*	*	*	*	*	*	-
MOV	RP,#imm8	2	2	0	0	byte (RP) <-- imm8	-	-	-	-	-	-	-	-	-	-
MOV	ILM,#imm8	2	2	0	0	byte (ILM) <-- imm8	-	-	-	-	-	-	-	-	-	-
MOVEA	RWi,ear	2	3	1	0	word (RWi) <-- ear	-	-	-	-	-	-	-	-	-	-
MOVEA	RWi,eam	2+	2+(a)	1	0	word (RWi) <-- eam	-	-	-	-	-	-	-	-	-	-
MOVEA	A,ear	2	1	0	0	word (A) <-- ear	-	*	-	-	-	-	-	-	-	-
MOVEA	A,eam	2+	1+(a)	0	0	word (A) <-- eam	-	*	-	-	-	-	-	-	-	-
ADDSP	#imm8	2	3	0	0	word (SP) <-- ext(imm8)	-	-	-	-	-	-	-	-	-	-
ADDSP	#imm16	3	3	0	0	word (SP) <-- imm16	-	-	-	-	-	-	-	-	-	-
MOV	A,brg1	2	*1	0	0	byte (A) <-- (brg1)	Z	*	-	-	-	*	*	-	-	-
MOV	brg2,A	2	1	0	0	byte (brg2) <-- (A)	-	-	-	-	-	*	*	-	-	-
NOP		1	1	0	0	No operation	-	-	-	-	-	-	-	-	-	-
ADB		1	1	0	0	Prefix code for AD space access	-	-	-	-	-	-	-	-	-	-
DTB		1	1	0	0	Prefix code for DT space access	-	-	-	-	-	-	-	-	-	-
PCB		1	1	0	0	Prefix code for PC space access	-	-	-	-	-	-	-	-	-	-
SPB		1	1	0	0	Prefix code for SP space access	-	-	-	-	-	-	-	-	-	-
NCC		1	1	0	0	Prefix code for flag no-change	-	-	-	-	-	-	-	-	-	-
CMR		1	1	0	0	Prefix code for common register bank	-	-	-	-	-	-	-	-	-	-

*1: PCB, ADB, SSB, USB, SPB: 1, DTB, DPR: 2
*2: 7 + 3 x (POP count) + 2 x (POP last register number), 7 when RLST = 0 (no transfer register)
*3: 29 + 3 x (PUSH count) - 3 x (PUSH last register number), 8 when RLST = 0 (no transfer register)
*4: (POP count) x (c) or (PUSH count) x (c)
*5: (POP count) or (PUSH count)

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-16 21 Bit Operand Instructions

Mnemonic		#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVB	A,dir:bp	3	5	0	(b)	byte (A) <-- (dir:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB	A,addr16:bp	4	5	0	(b)	byte (A) <-- (addr16:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB	A,io:bp	3	4	0	(b)	byte (A) <-- (io:bp)b	Z	*	-	-	-	*	*	-	-	-
MOVB	dir:bp,A	3	7	0	2 x (b)	bit (dir:bp)b <-- (A)	-	-	-	-	-	*	*	-	-	*
MOVB	addr16:bp,A	4	7	0	2 x (b)	bit (addr16:bp)b <-- (A)	-	-	-	-	-	*	*	-	-	*
MOVB	io:bp,A	3	6	0	2 x (b)	bit (io:bp)b <-- (A)	-	-	-	-	-	*	*	-	-	*
SETB	dir:bp	3	7	0	2 x (b)	bit (dir:bp)b <-- 1	-	-	-	-	-	-	-	-	-	*
SETB	addr16:bp	4	7	0	2 x (b)	bit (addr16:bp)b <-- 1	-	-	-	-	-	-	-	-	-	*
SETB	io:bp	3	7	0	2 x (b)	bit (io:bp)b <-- 1	-	-	-	-	-	-	-	-	-	*
CLRB	dir:bp	3	7	0	2 x (b)	bit (dir:bp)b <-- 0	-	-	-	-	-	-	-	-	-	*
CLRB	addr16:bp	4	7	0	2 x (b)	bit (addr16:bp)b <-- 0	-	-	-	-	-	-	-	-	-	*
CLRB	io:bp	3	7	0	2 x (b)	bit (io:bp)b <-- 0	-	-	-	-	-	-	-	-	-	*
BBC	dir:bp,rel	4	*1	0	(b)	Branch on (dir:bp) b = 0	-	-	-	-	-	-	*	-	-	-
BBC	addr16:bp,rel	5	*1	0	(b)	Branch on (addr16:bp) b = 0	-	-	-	-	-	-	*	-	-	-
BBC	io:bp,rel	4	*2	0	(b)	Branch on (io:bp) b = 0	-	-	-	-	-	-	*	-	-	-
BBS	dir:bp,rel	4	*1	0	(b)	Branch on (dir:bp) b = 1	-	-	-	-	-	-	*	-	-	-
BBS	addr16:bp,rel	5	*1	0	(b)	Branch on (addr16:bp) b = 1	-	-	-	-	-	-	*	-	-	-
BBS	io:bp,rel	4	*1	0	(b)	Branch on (io:bp) b = 1	-	-	-	-	-	-	*	-	-	-
SBBS	addr16:bp,rel	5	*3	0	2 x (b)	Branch on (addr16:bp) b = 1, bit = 1	-	-	-	-	-	-	*	-	-	*
WBTS	io:bp	3	*4	0	*5	Waits until (io:bp) b = 1	-	-	-	-	-	-	-	-	-	-
WBTC	io:bp	3	*4	0	*5	Waits until (io:bp) b = 0	-	-	-	-	-	-	-	-	-	-

*1: 8 when a branch is made; otherwise, 7
*2: 7 when a branch is made; otherwise, 6
*3: 10 when the condition is met; otherwise 9
*4: Undefined count
*5: Until the condition is met(dir:bp)b

DataSheet4U.com

Note: See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

Table A.8-17 Six Accumulator Operation Instructions (byte, word)

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
SWAP	1	3	0	0	byte (A)0-7 <--> (A)8-15	-	-	-	-	-	-	-	-	-	-
SWAPW	1	2	0	0	word (AH) <--> (AL)	-	*	-	-	-	-	-	-	-	-
EXT	1	1	0	0	Byte sign extension	X	-	-	-	-	*	*	-	-	-
EXTW	1	2	0	0	Word sign extension	-	X	-	-	-	*	*	-	-	-
ZEXT	1	1	0	0	Byte zero extension	Z	-	-	-	-	R	*	-	-	-
ZEXTW	1	1	0	0	Word zero extensionbyte	-	z	-	-	-	R	*	-	-	-

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.8-18 Ten String Instructions

Mnemonic	#	~	RG	B	Operation	LH	AH	I	S	T	N	Z	V	C	RMW
MOVS / MOVSI	2	*2	*5	*3	byte transfer @AH+ <-- @AL+, counter = RW0	-	-	-	-	-	-	-	-	-	-
MOVSD	2	*2	*5	*3	byte transfer @AH- <-- @AL-, counter = RW0	-	-	-	-	-	-	-	-	-	-
SCEQ / SCEQI	2	*1	*5	*4	byte search @AH+ <-- AL, counter RW0	-	-	-	-	-	*	*	*	*	-
SCEQD	2	*1	*5	*4	byte search @AH- <-- AL, counter RW0	-	-	-	-	-	*	*	*	*	-
FILS / FILSI	2	6m+6	*5	*3	byte fill @AH+ <-- AL, counter RW0	-	-	-	-	-	*	*	-	-	-
MOVSW / MOVSWI	2	*2	*5	*6	word transfer @AH+ <-- @AL+, counter = RW0	-	-	-	-	-	-	-	-	-	-
MOVSWD	2	*2	*5	*6	word transfer @AH- <-- @AL-, counter = RW0	-	-	-	-	-	-	-	-	-	-
SCWEQ / SCWEQI	2	*1	*5	*7	word search @AH+ - AL, counter = RW0	-	-	-	-	-	*	*	*	*	-
SCWEQD	2	*1	*5	*7	word search @AH- - AL, counter = RW0	-	-	-	-	-	*	*	*	*	-
FILSW / FILSWI	2	6m+6	*5	*6	word fill @AH+ <-- AL, counter = RW0	-	-	-	-	-	*	*	-	-	-

*1: 5 when RW0 is 0, 4 + 7 x (RW0) when the counter expires, or 7n + 5 when a match occurs
*2: 5 when RW0 is 0; otherwise, 4 + 8 x (RW0)
*3: (b) x (RW0) + (b) x (RW0) When the source and destination access different areas, calculate the (b) item individually.
*4: (b) x n
*5: 2 x (RW0)
*6: (c) x (RW0) + (c) x (RW0) When the source and destination access different areas, calculate the (c) item individually.
*7: (c) x n

et4U.com
DataShee
DataSheet4U.com

Note: m: RW0 value (counter value), n: Loop count
See Table A.5-1 and Table A.5-2 for information on (a) to (d) in the table.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

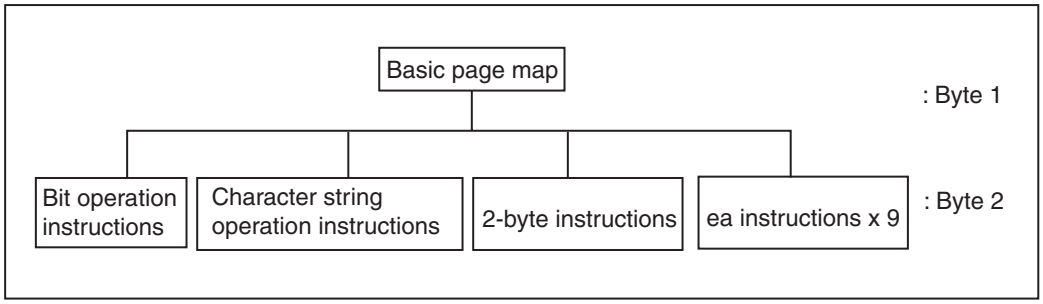
www.DataSheet4U.com

A.9 Instruction Map

Each F^{2}MC-16LX instruction code consists of 1 or 2 bytes. Therefore, the instruction map consists of multiple pages. Table A.9-2 to Table A.9-21 summarize the F^{2}MC-16LX instruction map.

■ Structure of Instruction Map

Figure A.9-1 Structure of Instruction Map



et4U.com
DataShee

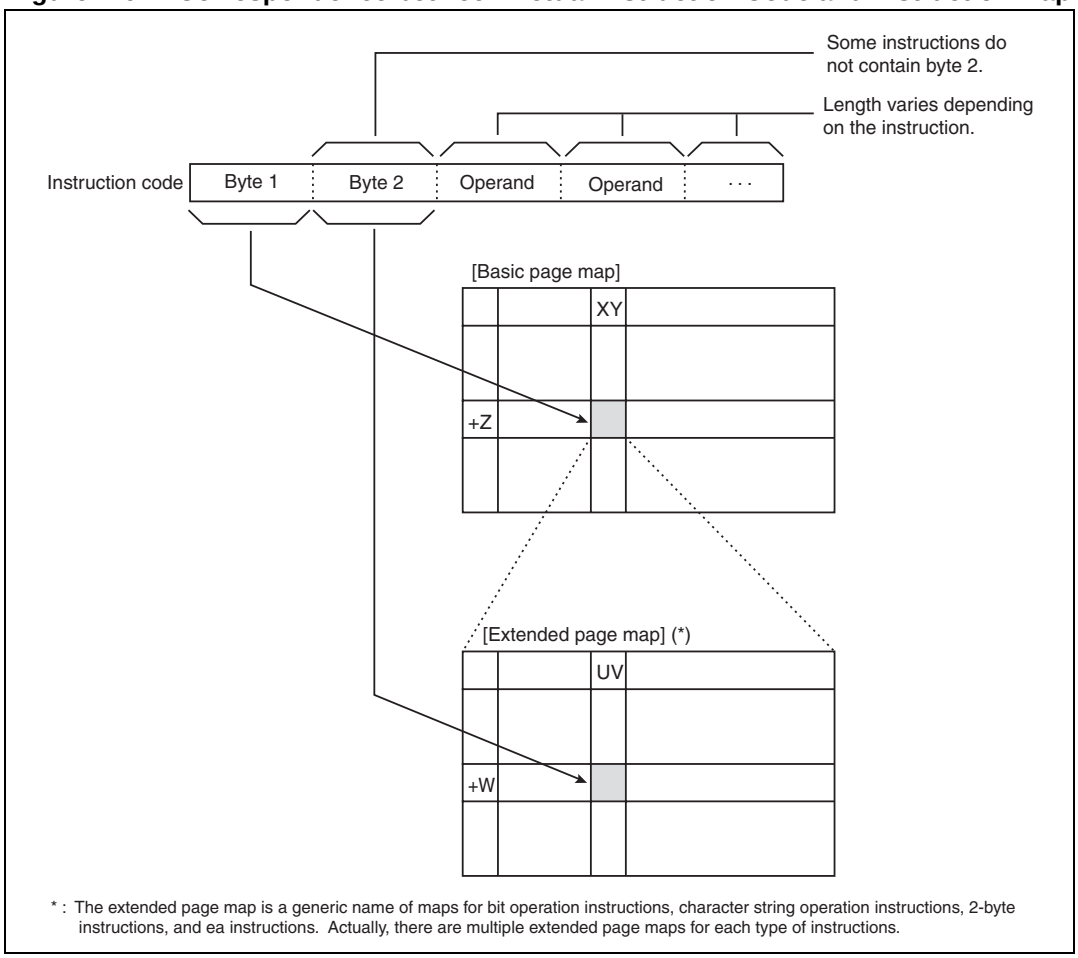
An instruction such as the NOP instruction that ends in one byte is completed within the basic page. An instruction such as the MOVS instruction that requires two bytes recognizes the existence of byte 2 when it references byte 1, and can check the following one byte by referencing the map for byte 2. Figure A.9-2 shows the correspondence between an actual instruction code and instruction map.

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Figure A.9-2 Correspondence between Actual Instruction Code and Instruction Map



An example of an instruction code is shown in Table A.9-1.

Table A.9-1 Example of an Instruction Code

Instruction	Byte 1 (from basic page map)	Byte 2 (from extended page map)
NOP	00 + 0 = 00	-
AND A, #8	30 + 4 = 34	-
MOV A, ADB	60 + F = 6F	00 + 0 = 00
@RW2+d8, #8rel	70 + 0 = 70	F0 + 2 = F2

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-2 Basic Page Map

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	NOP	CMR	ADD A,dir	ADD A,#8	MOV A,dir	MOV A,io	BRA rel	ea instruction 1	MOV A,Ri	MOV Ri,A	MOV Ri,#8	MOVX A,Ri	MOVX A, @RWi+d8	MOVN A,#4	CALLV #4	BZ /BEQ rel
+ 1	INT9	NCC	SUB A,dir	SUB A,#8	MOV dir,A	MOV io,A	JMP @A	ea instruction 2								BNZ/BNE rel
+ 2	ADDDC A	SUBDC A	ADDC A	SUBC A	MOV A,#8	MOV A,addr16	JMP addr16	ea instruction 3								BC /BLO rel
+ 3	NEG A	JCTX @A	CMP A	CMP A,#8	MOVX A,#8	MOV addr16,A	JMPP addr24	ea instruction 4								BNC /BHS rel
+ 4	PCB	EXT	AND CCR,#8	AND A,#8	MOV dir,#8	MOV io,#8	CALL addr16	ea instruction 5								BN rel
+ 5	DTB	ZEXT	OR CCR,#8	OR A,#8	MOVX A,dir	MOVX A,io	CALLP addr24	ea instruction 6								BP rel
+ 6	ADB	SWAP	DIVU A	XOR A,#8	MOVW A,SP	MOVW io,#16	RETP	ea instruction 7								BV rel
+ 7	SPB	ADDSP #8	MULU A	NOT A	MOVW SP,A	MOVX A,addr16	RET	ea instruction 8								BNV rel
+ 8	LINK #imm8	ADDL A,#32	ADDW A	ADDW A,#16	MOVW A,dir	MOVW A,io	INT #vct8	ea instruction 9	MOVW A,RWi	MOVW RWi,A	MOVW RWi,#16	MOVW A, @RWi+d8	MOVW @R Wi+d8,A			BT rel
+ 9	UNLINK	SUBL A,#32	SUBW A	SUBW A,#16	MOVW dir,A	MOVW io,A	INT addr16	MOVEA RWi,ea								BNT rel
+ A	MOV RP,#8	MOV ILM,#8	CBNE A, #8,rel	CWBNE A, #16,rel	MOVW A,#16	MOVW A,addr16	INTP addr24	MOV Ri,ea								BLT rel
+ B	NEGW	CMPL A,#32	CMPW A	CMPW A,#16	MOVL A,#32	MOVW addr16,A	RETI	MOVW RWi,ea								BGE rel
+ C	LSLW A	EXTW	ANDW A	ANDW A,#16	PUSHW A	POPW A	Bit operation instruction	MOV ea,Ri								BLE rel
+ D		ZEXTW	ORW A	ORW A,#16	PUSHW AH	POPW AH		MOVW ea,RWi								BGT rel
+ E	ASRW A	SWAPW	XORW A	XORW A,#16	PUSHW PS	POPW PS	Character string operation instruction	XCH Ri,ea								BLS rel
+ F	LSRW A	ADDSP #16	MULUW A	NOTW A	PUSHW rlst	POPW rlst	2-byte instruction	XCHW RWi,ea								BHI rel

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-3 Bit Operation Instruction Map (first byte = $6C_H$)

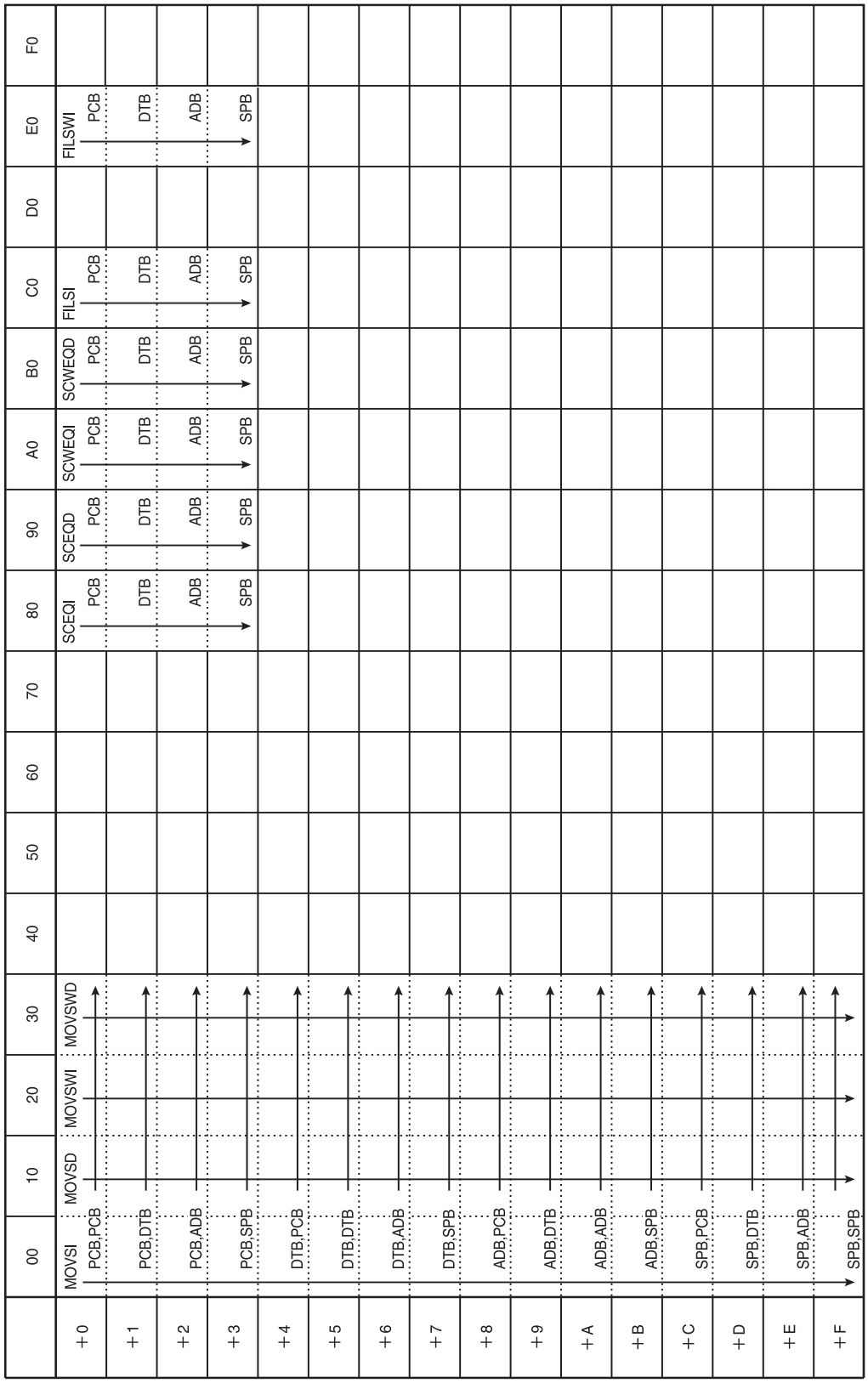
	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	MOVB A,io:bp		MOVB io:bp,A		CLRB io:bp		SETB io:bp		BBC io :bp,rel		BBS io :bp,rel		WBTS io:bp		WBTC io:bp	
+ 1																
+ 2																
+ 3																
+ 4																
+ 5																
+ 6																
+ 7																
+ 8	MOVB A,dir:bp	MOVB A,a ddr16:bp	MOVB dir:bp,A	MOVB addr16:bp,A	CLRB dir:bp	CLRB a ddr16:bp	SETB dir:bp	SETB a ddr16:bp	BBC dir:bp,rel	BBC ad16 :bp,rel	BBS dir:bp,rel	BBS ad16 :bp,rel				SBBS a ddr16:bp
+ 9																
+ A																
+ B																
+ C																
+ D																
+ E																
+ F																

DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-4 Character String Operation Instruction Map (first byte = $6E_H$)



DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-5 2-byte Instruction Map (first byte = $6F_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+0	MOV A,DTB	MOV DTB,A	MOVX A, @RL0+d8	MOV @RL0 +d8,A	MOV A, @RL0+d8											
+1	MOV A,ADB	MOV ADB,A														
+2	MOV A,SSB	MOV SSB,A	MOVX A, @RL1+d8	MOV @RL1 +d8,A	MOV A, @RL1+d8											
+3	MOV A,USB	MOV USB,A														
+4	MOV A,DPR	MOV DPR,A	MOVX A, @RL2+d8	MOV @RL2 +d8,A	MOV A, @RL2+d8											
+5	MOV A,@A	MOV @AL,AH														
+6	MOV A,PCB	MOVX A,@A	MOVX A, @RL3+d8	MOV @RL3 +d8,A	MOV A, @RL3+d8											
+7	ROLC A	RORC A														
+8				MOVW @RL 0+d8,A	MOVW A, @RL0+d8		MUL A									
+9							MULW A									
+A				MOVW @RL 1+d8,A	MOVW A, @RL1+d8		DIVU A									
+B																
+C	LSLW A,R0	LSLL A,R0	LSL A,R0	MOVW @RL 2+d8,A	MOVW A, @RL2+d8											
+D	MOVW A,@A	MOVW @AL,AH	NRML A,R0													
+E	ASRW A,R0	ASRL A,R0	ASR A,R0	MOVW @RL 3+d8,A	MOVW A, @RL3+d8											
+F	LSRW A,R0	LSRL A,R0	LSR A,R0													

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-6 ea Instruction 1 (first byte = 70_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
					CWBNE ,↓	CWBNE ,↓									CBNE ,↓	CBNE ,↓
+ 0	ADDL A, A,RL0	ADDL A, @RW0+d8	SUBL A,RL0	SUBL A, @RW0+d8	RW0, #16,rel	@RW0+d8, #16,rel	CMPL A,RL0	CMPL A, @RW0+d8	ANDL A,RL0	ANDL A, @RW0+d8	ORL A,RL0	ORL A, @RW0+d8	XORL A,RL0	XORL A, @RW0+d8	R0, #8,rel	@RW0+d8, #8,rel
+ 1	ADDL A, A,RL0	ADDL A, @RW1+d8	SUBL A,RL0	SUBL A, @RW1+d8	RW1, #16,rel	@RW1+d8, #16,rel	CMPL A,RL0	CMPL A, @RW1+d8	ANDL A,RL0	ANDL A, @RW1+d8	ORL A,RL0	ORL A, @RW1+d8	XORL A,RL0	XORL A, @RW1+d8	R1, #8,rel	@RW1+d8, #8,rel
+ 2	ADDL A, A,RL1	ADDL A, @RW2+d8	SUBL A,RL1	SUBL A, @RW2+d8	RW2, #16,rel	@RW2+d8, #16,rel	CMPL A,RL1	CMPL A, @RW2+d8	ANDL A,RL1	ANDL A, @RW2+d8	ORL A,RL1	ORL A, @RW2+d8	XORL A,RL1	XORL A, @RW2+d8	R2, #8,rel	@RW2+d8, #8,rel
+ 3	ADDL A, A,RL1	ADDL A, @RW3+d8	SUBL A,RL1	SUBL A, @RW3+d8	RW3, #16,rel	@RW3+d8, #16,rel	CMPL A,RL1	CMPL A, @RW3+d8	ANDL A,RL1	ANDL A, @RW3+d8	ORL A,RL1	ORL A, @RW3+d8	XORL A,RL1	XORL A, @RW3+d8	R3, #8,rel	@RW3+d8, #8,rel
+ 4	ADDL A, A,RL2	ADDL A, @RW4+d8	SUBL A,RL2	SUBL A, @RW4+d8	RW4, #16,rel	@RW4+d8, #16,rel	CMPL A,RL2	CMPL A, @RW4+d8	ANDL A,RL2	ANDL A, @RW4+d8	ORL A,RL2	ORL A, @RW4+d8	XORL A,RL2	XORL A, @RW4+d8	R4, #8,rel	@RW4+d8, #8,rel
+ 5	ADDL A, A,RL2	ADDL A, @RW5+d8	SUBL A,RL2	SUBL A, @RW5+d8	RW5, #16,rel	@RW5+d8, #16,rel	CMPL A,RL2	CMPL A, @RW5+d8	ANDL A,RL2	ANDL A, @RW5+d8	ORL A,RL2	ORL A, @RW5+d8	XORL A,RL2	XORL A, @RW5+d8	R5, #8,rel	@RW5+d8, #8,rel
+ 6	ADDL A, A,RL3	ADDL A, @RW6+d8	SUBL A,RL3	SUBL A, @RW6+d8	RW6, #16,rel	@RW6+d8, #16,rel	CMPL A,RL3	CMPL A, @RW6+d8	ANDL A,RL3	ANDL A, @RW6+d8	ORL A,RL3	ORL A, @RW6+d8	XORL A,RL3	XORL A, @RW6+d8	R6, #8,rel	@RW6+d8, #8,rel
+ 7	ADDL A, A,RL3	ADDL A, @RW7+d8	SUBL A,RL3	SUBL A, @RW7+d8	RW7, #16,rel	@RW7+d8, #16,rel	CMPL A,RL3	CMPL A, @RW7+d8	ANDL A,RL3	ANDL A, @RW7+d8	ORL A,RL3	ORL A, @RW7+d8	XORL A,RL3	XORL A, @RW7+d8	R7, #8,rel	@RW7+d8, #8,rel
+ 8	ADDL A, A,@RW0	ADDL A, @RW0+d16	SUBL A,@RW0	SUBL A, @RW0+d16	@RW0, #16,rel	@RW0+d16 , ,#16,rel	CMPL A,@RW0	CMPL A, @RW0+d16	ANDL A,@RW0	ANDL A, @RW0+d16	ORL A,@RW0	ORL A, @RW0+d16	XORL A,@RW0	XORL A, @RW0+d16	@RW0, #8,rel	@RW0+d16 ,#8,rel
+ 9	ADDL A, A,@RW1	ADDL A, @RW1+d16	SUBL A,@RW1	SUBL A, @RW1+d16	@RW1, #16,rel	@RW1+d16 , ,#16,rel	CMPL A,@RW1	CMPL A, @RW1+d16	ANDL A,@RW1	ANDL A, @RW1+d16	ORL A,@RW1	ORL A, @RW1+d16	XORL A,@RW1	XORL A, @RW1+d16	@RW1, #8,rel	@RW1+d16 ,#8,rel
+ A	ADDL A, A,@RW2	ADDL A, @RW2+d16	SUBL A,@RW2	SUBL A, @RW2+d16	@RW2, #16,rel	@RW2+d16 , ,#16,rel	CMPL A,@RW2	CMPL A, @RW2+d16	ANDL A,@RW2	ANDL A, @RW2+d16	ORL A,@RW2	ORL A, @RW2+d16	XORL A,@RW2	XORL A, @RW2+d16	@RW2, #8,rel	@RW2+d16 ,#8,rel
+ B	ADDL A, A,@RW3	ADDL A, @RW3+d16	SUBL A,@RW3	SUBL A, @RW3+d16	@RW3, #16,rel	@RW3+d16 , ,#16,rel	CMPL A,@RW3	CMPL A, @RW3+d16	ANDL A,@RW3	ANDL A, @RW3+d16	ORL A,@RW3	ORL A, @RW3+d16	XORL A,@RW3	XORL A, @RW3+d16	@RW3, #8,rel	@RW3+d16 ,#8,rel
+ C	ADDL A, A,@RW0+	ADDL A, @RW0+RW7	SUBL A,@RW0+	SUBL A, @RW0+RW7	Use prohibited	@RW0+RW7 , ,#16,rel	CMPL A,@RW0+	CMPL A, @RW0+RW7	ANDL A,@RW0+	ANDL A, @RW0+RW7	ORL A,@RW0+	ORL A, @RW0+RW7	XORL A,@RW0+	XORL A, @RW0+RW7	Use prohibited	@RW0+RW7 ,#8,rel
+ D	ADDL A, A,@RW1+	ADDL A, @RW1+RW7	SUBL A,@RW1+	SUBL A, @RW1+RW7	Use prohibited	@RW1+RW7 , ,#16,rel	CMPL A,@RW1+	CMPL A, @RW1+RW7	ANDL A,@RW1+	ANDL A, @RW1+RW7	ORL A,@RW1+	ORL A, @RW1+RW7	XORL A,@RW1+	XORL A, @RW1+RW7	Use prohibited	@RW1+RW7 ,#8,rel
+ E	ADDL A, A,@RW2+	ADDL A, @PC+d16	SUBL A,@RW2+	SUBL A, @PC+d16	Use prohibited	@PC+d16, #16,rel	CMPL A,@RW2+	CMPL A, @PC+d16	ANDL A,@RW2+	ANDL A, @PC+d16	ORL A,@RW2+	ORL A, @PC+d16	XORL A,@RW2+	XORL A, @PC+d16	Use prohibited	@PC+d16, #8,rel
+ F	ADDL A,@RW3+	ADDL A, addr16	SUBL A,@RW3+	SUBL A, addr16	Use prohibited	addr16, #16,rel	CMPL A,@RW3+	CMPL A, addr16	ANDL A,@RW3+	ANDL A, addr16	ORL A,@RW3+	ORL A, addr16	XORL A,@RW3+	XORL A, addr16	Use prohibited	addr16, #8,rel

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-7 ea Instruction 2 (first byte = 71_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	JMPP @RL0	JMPP @@RW0+d8	CALLP @RL0	CALLP @@RW0+d8	INCL RL0	INCL @RW0+d8	DECL RL0	DECL @RW0+d8	MOVL A,RL0	MOVL A, @RW0+d8	MOVL RL0,A	MOVL A, @RW0+d8	MOV R0,#8	MOV A, @RW0+d8	MOVEA A,RW0	MOVEA A, @RW0+d8
+ 1	JMPP @RL0	JMPP @@RW1+d8	CALLP @RL0	CALLP @@RW1+d8	INCL RL0	INCL @RW1+d8	DECL RL0	DECL @RW1+d8	MOVL A,RL0	MOVL A, @RW1+d8	MOVL RL0,A	MOVL A, @RW1+d8	MOV R1,#8	MOV A, @RW1+d8	MOVEA A,RW1	MOVEA A, @RW1+d8
+ 2	JMPP @RL1	JMPP @@RW2+d8	CALLP @RL1	CALLP @@RW2+d8	INCL RL1	INCL @RW2+d8	DECL RL1	DECL @RW2+d8	MOVL A,RL1	MOVL A, @RW2+d8	MOVL RL1,A	MOVL A, @RW2+d8	MOV R2,#8	MOV A, @RW2+d8	MOVEA A,RW2	MOVEA A, @RW2+d8
+ 3	JMPP @RL1	JMPP @@RW3+d8	CALLP @RL1	CALLP @@RW3+d8	INCL RL1	INCL @RW3+d8	DECL RL1	DECL @RW3+d8	MOVL A,RL1	MOVL A, @RW3+d8	MOVL RL1,A	MOVL A, @RW3+d8	MOV R3,#8	MOV A, @RW3+d8	MOVEA A,RW3	MOVEA A, @RW3+d8
+ 4	JMPP @RL2	JMPP @@RW4+d8	CALLP @RL2	CALLP @@RW4+d8	INCL RL2	INCL @RW4+d8	DECL RL2	DECL @RW4+d8	MOVL A,RL2	MOVL A, @RW4+d8	MOVL RL2,A	MOVL A, @RW4+d8	MOV R4,#8	MOV A, @RW4+d8	MOVEA A,RW4	MOVEA A, @RW4+d8
+ 5	JMPP @RL2	JMPP @@RW5+d8	CALLP @RL2	CALLP @@RW5+d8	INCL RL2	INCL @RW5+d8	DECL RL2	DECL @RW5+d8	MOVL A,RL2	MOVL A, @RW5+d8	MOVL RL2,A	MOVL A, @RW5+d8	MOV R5,#8	MOV A, @RW5+d8	MOVEA A,RW5	MOVEA A, @RW5+d8
+ 6	JMPP @RL3	JMPP @@RW6+d8	CALLP @RL3	CALLP @@RW6+d8	INCL RL3	INCL @RW6+d8	DECL RL3	DECL @RW6+d8	MOVL A,RL3	MOVL A, @RW6+d8	MOVL RL3,A	MOVL A, @RW6+d8	MOV R6,#8	MOV A, @RW6+d8	MOVEA A,RW6	MOVEA A, @RW6+d8
+ 7	JMPP @RL3	JMPP @@RW7+d8	CALLP @RL3	CALLP @@RW7+d8	INCL RL3	INCL @RW7+d8	DECL RL3	DECL @RW7+d8	MOVL A,RL3	MOVL A, @RW7+d8	MOVL RL3,A	MOVL A, @RW7+d8	MOV R7,#8	MOV A, @RW7+d8	MOVEA A,RW7	MOVEA A, @RW7+d8
+ 8	JMPP @@RW0	JMPP @@RW0+d16	CALLP @@RW0	CALLP @@RW0+d16	INCL @RW0	INCL @RW0+d16	DECL @RW0	DECL @RW0+d16	MOVL A,@RW0	MOVL A, @RW0+d16	MOVL @RW0,A	MOVL A, @RW0+d16	MOV @RW0,#8	MOV @RW 0+d16,#8	MOVEA A,@RW0	MOVEA A, @RW0+d16
+ 9	JMPP @@RW1	JMPP @@RW1+d16	CALLP @@RW1	CALLP @@RW1+d16	INCL @RW1	INCL @RW1+d16	DECL @RW1	DECL @RW1+d16	MOVL A,@RW1	MOVL A, @RW1+d16	MOVL @RW1,A	MOVL A, @RW1+d16	MOV @RW1,#8	MOV @RW 1+d16,#8	MOVEA A,@RW1	MOVEA A, @RW1+d16
+ A	JMPP @@RW2	JMPP @@RW2+d16	CALLP @@RW2	CALLP @@RW2+d16	INCL @RW2	INCL @RW2+d16	DECL @RW2	DECL @RW2+d16	MOVL A,@RW2	MOVL A, @RW2+d16	MOVL @RW2,A	MOVL A, @RW2+d16	MOV @RW2,#8	MOV @RW 2+d16,#8	MOVEA A,@RW2	MOVEA A, @RW2+d16
+ B	JMPP @@RW3	JMPP @@RW3+d16	CALLP @@RW3	CALLP @@RW3+d16	INCL @RW3	INCL @RW3+d16	DECL @RW3	DECL @RW3+d16	MOVL A,@RW3	MOVL A, @RW3+d16	MOVL @RW3,A	MOVL A, @RW3+d16	MOV @RW3,#8	MOV @RW 3+d16,#8	MOVEA A,@RW3	MOVEA A, @RW3+d16
+ C	JMPP @@RW0+	JMPP @@RW0+RW7	CALLP @@RW0+	CALLP @@RW0+RW7	INCL @RW0+	INCL @RW0+RW7	DECL @RW0+	DECL @RW0+RW7	MOVL A,@RW0+	MOVL A, @RW0+RW7	MOVL @RW0+,A	MOVL A, @RW0+RW7	MOV @RW0+,#8	MOV @RW 0+RW7,#8	MOVEA A,@RW0+	MOVEA A, @RW0+RW7
+ D	JMPP @@RW1+	JMPP @@RW1+RW7	CALLP @@RW1+	CALLP @@RW1+RW7	INCL @RW1+	INCL @RW1+RW7	DECL @RW1+	DECL @RW1+RW7	MOVL A,@RW1+	MOVL A, @RW1+RW7	MOVL @RW1+,A	MOVL A, @RW1+RW7	MOV @RW1+,#8	MOV @RW 1+RW7,#8	MOVEA A,@RW1+	MOVEA A, @RW1+RW7
+ E	JMPP @@RW2+	JMPP @@PC+d16	CALLP @@RW2+	CALLP @@PC+d16	INCL @RW2+	INCL @PC+d16	DECL @RW2+	DECL @PC+d16	MOVL A,@RW2+	MOVL A, @PC+d16	MOVL @RW2+,A	MOVL A, @P	MOV @RW2+,#8	MOV @P C+d16,#8	MOVEA A,@RW2+	MOVEA A, @PC+d16
+ F	JMPP @@RW3+	JMPP @addr16	CALLP @@RW3+	CALLP @addr16	INCL @RW3+	INCL addr16	DECL @RW3+	DECL addr16	MOVL A,@RW3+	MOVL A, addr16	MOVL @RW3+,A	MOVL a addr16,A	MOV @RW3+,#8	MOV a ddr16,#8	MOVEA A,@RW3+	MOVEA A, addr16

et4U.com
DataShee
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-8 ea Instruction 3 (first byte = 72_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	ROLC R0	ROLC @RW0+d8	RORC R0	RORC @RW0+d8	INC R0	INC @RW0+d8	DEC R0	DEC @RW0+d8	MOV A,R0	MOV A, @RW0+d8	MOV R0,A	MOV A, W0+d8,A	MOVX A,R0	MOVX A, @RW0+d8	XCH A,R0	XCH A, @RW0+d8
+ 1	ROLC R1	ROLC @RW1+d8	RORC R1	RORC @RW1+d8	INC R1	INC @RW1+d8	DEC R1	DEC @RW1+d8	MOV A,R1	MOV A, @RW1+d8	MOV R1,A	MOV A, W1+d8,A	MOVX A,R1	MOVX A, @RW1+d8	XCH A,R1	XCH A, @RW1+d8
+ 2	ROLC R2	ROLC @RW2+d8	RORC R2	RORC @RW2+d8	INC R2	INC @RW2+d8	DEC R2	DEC @RW2+d8	MOV A,R2	MOV A, @RW2+d8	MOV R2,A	MOV A, W2+d8,A	MOVX A,R2	MOVX A, @RW2+d8	XCH A,R2	XCH A, @RW2+d8
+ 3	ROLC R3	ROLC @RW3+d8	RORC R3	RORC @RW3+d8	INC R3	INC @RW3+d8	DEC R3	DEC @RW3+d8	MOV A,R3	MOV A, @RW3+d8	MOV R3,A	MOV A, W3+d8,A	MOVX A,R3	MOVX A, @RW3+d8	XCH A,R3	XCH A, @RW3+d8
+ 4	ROLC R4	ROLC @RW4+d8	RORC R4	RORC @RW4+d8	INC R4	INC @RW4+d8	DEC R4	DEC @RW4+d8	MOV A,R4	MOV A, @RW4+d8	MOV R4,A	MOV A, W4+d8,A	MOVX A,R4	MOVX A, @RW4+d8	XCH A,R4	XCH A, @RW4+d8
+ 5	ROLC R5	ROLC @RW5+d8	RORC R5	RORC @RW5+d8	INC R5	INC @RW5+d8	DEC R5	DEC @RW5+d8	MOV A,R5	MOV A, @RW5+d8	MOV R5,A	MOV A, W5+d8,A	MOVX A,R5	MOVX A, @RW5+d8	XCH A,R5	XCH A, @RW5+d8
+ 6	ROLC R6	ROLC @RW6+d8	RORC R6	RORC @RW6+d8	INC R6	INC @RW6+d8	DEC R6	DEC @RW6+d8	MOV A,R6	MOV A, @RW6+d8	MOV R6,A	MOV A, W6+d8,A	MOVX A,R6	MOVX A, @RW6+d8	XCH A,R6	XCH A, @RW6+d8
+ 7	ROLC R7	ROLC @RW7+d8	RORC R7	RORC @RW7+d8	INC R7	INC @RW7+d8	DEC R7	DEC @RW7+d8	MOV A,R7	MOV A, @RW7+d8	MOV R7,A	MOV A, W7+d8,A	MOVX A,R7	MOVX A, @RW7+d8	XCH A,R7	XCH A, @RW7+d8
+ 8	ROLC @RW0	ROLC @RW0+d16	RORC @RW0	RORC @RW0+d16	INC @RW0	INC @RW0+d16	DEC @RW0	DEC @RW0+d16	MOV A,@RW0	MOV A, @RW0+d16	MOV @RW0,A	MOV A, W0+d16,A	MOVX A,@RW0	MOVX A, @RW0+d16	XCH A,@RW0	XCH A, @RW0+d16
+ 9	ROLC @RW1	ROLC @RW1+d16	RORC @RW1	RORC @RW1+d16	INC @RW1	INC @RW1+d16	DEC @RW1	DEC @RW1+d16	MOV A,@RW1	MOV A, @RW1+d16	MOV @RW1,A	MOV A, W1+d16,A	MOVX A,@RW1	MOVX A, @RW1+d16	XCH A,@RW1	XCH A, @RW1+d16
+ A	ROLC @RW2	ROLC @RW2+d16	RORC @RW2	RORC @RW2+d16	INC @RW2	INC @RW2+d16	DEC @RW2	DEC @RW2+d16	MOV A,@RW2	MOV A, @RW2+d16	MOV @RW2,A	MOV A, W2+d16,A	MOVX A,@RW2	MOVX A, @RW2+d16	XCH A,@RW2	XCH A, @RW2+d16
+ B	ROLC @RW3	ROLC @RW3+d16	RORC @RW3	RORC @RW3+d16	INC @RW3	INC @RW3+d16	DEC @RW3	DEC @RW3+d16	MOV A,@RW3	MOV A, @RW3+d16	MOV @RW3,A	MOV A, W3+d16,A	MOVX A,@RW3	MOVX A, @RW3+d16	XCH A,@RW3	XCH A, @RW3+d16
+ C	ROLC @RW0+	ROLC @RW0+RW7	RORC @RW0+	RORC @RW0+RW7	INC @RW0+	INC @RW0+RW7	DEC @RW0+	DEC @RW0+RW7	MOV A,@RW0+	MOV A, @RW0+RW7	MOV @RW0+,A	MOV A, W0+RW7,A	MOVX A,@RW0+	MOVX A, @RW0+RW7	XCH A,@RW0+	XCH A, @RW0+RW7
+ D	ROLC @RW1+	ROLC @RW1+RW7	RORC @RW1+	RORC @RW1+RW7	INC @RW1+	INC @RW1+RW7	DEC @RW1+	DEC @RW1+RW7	MOV A,@RW1+	MOV A, @RW1+RW7	MOV @RW1+,A	MOV A, W1+RW7,A	MOVX A,@RW1+	MOVX A, @RW1+RW7	XCH A,@RW1+	XCH A, @RW1+RW7
+ E	ROLC @RW2+	ROLC @PC+d16	RORC @RW2+	RORC @PC+d16	INC @RW2+	INC @PC+d16	DEC @RW2+	DEC @PC+d16	MOV A,@RW2+	MOV A, @PC+d16	MOV @RW2+,A	MOV A, C+d16,A	MOVX A,@RW2+	MOVX A, @PC+d16	XCH A,@RW2+	XCH A, @PC+d16
+ F	ROLC @RW3+	ROLC addr16	RORC @RW3+	RORC addr16	INC @RW3+	INC addr16	DEC @RW3+	DEC addr16	MOV A,@RW3+	MOV A, addr16	MOV @RW3+,A	MOV addr16,A	MOVX A,@RW3+	MOVX A, addr16	XCH A,@RW3+	XCH A, addr16

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-9 ea Instruction 4 (first byte = 73_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	JMP @RW0	JMP @@RW0+d8	CALL @RW0	CALL @@RW0+d8	INCW RW0	INCW @RW0+d8	DECW RW0	DECW @RW0+d8	MOVW A,RW0	MOVW A, @RW0+d8	MOVW RW0,A	MOVW @R W0+d8,A	MOVW RW0,#16	MOVW @RW 0+d8,#16	XCHW A,RW0	XCHW A, @RW0+d8
+ 1	JMP @RW1	JMP @@RW1+d8	CALL @RW1	CALL @@RW1+d8	INCW RW1	INCW @RW1+d8	DECW RW1	DECW @RW1+d8	MOVW A,RW1	MOVW A, @RW1+d8	MOVW RW1,A	MOVW @R W1+d8,A	MOVW RW1,#16	MOVW @RW 1+d8,#16	XCHW A,RW1	XCHW A, @RW1+d8
+ 2	JMP @RW2	JMP @@RW2+d8	CALL @RW2	CALL @@RW2+d8	INCW RW2	INCW @RW2+d8	DECW RW2	DECW @RW2+d8	MOVW A,RW2	MOVW A, @RW2+d8	MOVW RW2,A	MOVW @R W2+d8,A	MOVW RW2,#16	MOVW @RW 2+d8,#16	XCHW A,RW2	XCHW A, @RW2+d8
+ 3	JMP @RW3	JMP @@RW3+d8	CALL @RW3	CALL @@RW3+d8	INCW RW3	INCW @RW3+d8	DECW RW3	DECW @RW3+d8	MOVW A,RW3	MOVW A, @RW3+d8	MOVW RW3,A	MOVW @R W3+d8,A	MOVW RW3,#16	MOVW @RW 3+d8,#16	XCHW A,RW3	XCHW A, @RW3+d8
+ 4	JMP @RW4	JMP @@RW4+d8	CALL @RW4	CALL @@RW4+d8	INCW RW4	INCW @RW4+d8	DECW RW4	DECW @RW4+d8	MOVW A,RW4	MOVW A, @RW4+d8	MOVW RW4,A	MOVW @R W4+d8,A	MOVW RW4,#16	MOVW @RW 4+d8,#16	XCHW A,RW4	XCHW A, @RW4+d8
+ 5	JMP @RW5	JMP @@RW5+d8	CALL @RW5	CALL @@RW5+d8	INCW RW5	INCW @RW5+d8	DECW RW5	DECW @RW5+d8	MOVW A,RW5	MOVW A, @RW5+d8	MOVW RW5,A	MOVW @R W5+d8,A	MOVW RW5,#16	MOVW @RW 5+d8,#16	XCHW A,RW5	XCHW A, @RW5+d8
+ 6	JMP @RW6	JMP @@RW6+d8	CALL @RW6	CALL @@RW6+d8	INCW RW6	INCW @RW6+d8	DECW RW6	DECW @RW6+d8	MOVW A,RW6	MOVW A, @RW6+d8	MOVW RW6,A	MOVW @R W6+d8,A	MOVW RW6,#16	MOVW @RW 6+d8,#16	XCHW A,RW6	XCHW A, @RW6+d8
+ 7	JMP @RW7	JMP @@RW7+d8	CALL @RW7	CALL @@RW7+d8	INCW RW7	INCW @RW7+d8	DECW RW7	DECW @RW7+d8	MOVW A,RW7	MOVW A, @RW7+d8	MOVW RW7,A	MOVW @R W7+d8,A	MOVW RW7,#16	MOVW @RW 7+d8,#16	XCHW A,RW7	XCHW A, @RW7+d8
+ 8	JMP @@RW0	JMP @@RW0+d16	CALL @@RW0	CALL @@RW0+d16	INCW @RW0	INCW @RW0+d16	DECW @RW0	DECW @RW0+d16	MOVW A,@RW0	MOVW A, @RW0+d16	MOVW @RW0,A	MOVW @R W0+d16,A	MOVW @RW0,#16	MOVW@RW0 +d16,#16	XCHW A,@RW0	XCHW A, @RW0+d16
+ 9	JMP @@RW1	JMP @@RW1+d16	CALL @@RW1	CALL @@RW1+d16	INCW @RW1	INCW @RW1+d16	DECW @RW1	DECW @RW1+d16	MOVW A,@RW1	MOVW A, @RW1+d16	MOVW @RW1,A	MOVW @R W1+d16,A	MOVW @RW1,#16	MOVW@RW1 +d16,#16	XCHW A,@RW1	XCHW A, @RW1+d16
+ A	JMP @@RW2	JMP @@RW2+d16	CALL @@RW2	CALL @@RW2+d16	INCW @RW2	INCW @RW2+d16	DECW @RW2	DECW @RW2+d16	MOVW A,@RW2	MOVW A, @RW2+d16	MOVW @RW2,A	MOVW @R W2+d16,A	MOVW @RW2,#16	MOVW@RW2 +d16,#16	XCHW A,@RW2	XCHW A, @RW2+d16
+ B	JMP @@RW3	JMP @@RW3+d16	CALL @@RW3	CALL @@RW3+d16	INCW @RW3	INCW @RW3+d16	DECW @RW3	DECW @RW3+d16	MOVW A,@RW3	MOVW A, @RW3+d16	MOVW @RW3,A	MOVW @R W3+d16,A	MOVW @RW3,#16	MOVW@RW3 +d16,#16	XCHW A,@RW3	XCHW A, @RW3+d16
+ C	JMP @@RW0+	JMP @@RW0+RW7	CALL @@RW0+	CALL @@RW0+RW7	INCW @RW0+	INCW @RW0+RW7	DECW @RW0+	DECW @RW0+RW7	MOVW A,@RW0+	MOVW A, @RW0+RW7	MOVW @RW0+ A	MOVW @R W0+RW7,A	MOVW @ RW0+,#16	MOVW@RW0 +RW7,#16	XCHW A,@RW0+	XCHW A, @RW0+RW7
+ D	JMP @@RW1+	JMP @@RW1+RW7	CALL @@RW1+	CALL @@RW1+RW7	INCW @RW1+	INCW @RW1+RW7	DECW @RW1+	DECW @RW1+RW7	MOVW A,@RW1+	MOVW A, @RW1+RW7	MOVW @RW1+ A	MOVW @R W1+RW7,A	MOVW @ RW1+,#16	MOVW@RW1 +RW7,#16	XCHW A,@RW1+	XCHW A, @RW1+RW7
+ E	JMP @@RW2+	JMP @@PC+d16	CALL @@RW2+	CALL @@PC+d16	INCW @RW2+	INCW @PC+d16	DECW @RW2+	DECW @PC+d16	MOVW A,@RW2+	MOVW A, @PC+d16	MOVW @RW2+ A	MOVW @P C+d16,A	MOVW @ RW2+,#16	MOVW @PC +d16,#16	XCHW A,@RW2+	XCHW A, @PC+d16
+ F	JMP @@RW3+	JMP @addr16	CALL @@RW3+	CALL @addr16	INCW @RW3+	INCW addr16	DECW @RW3+	DECW addr16	MOVW A,@RW3+	MOVW A, addr16	MOVW @RW3+ A	MOVW addr16,A	MOVW @ RW3+,#16	MOVW ad dr16,#16	XCHW A,@RW3+	XCHW A, addr16

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-10 ea Instruction 5 (first byte = 74_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	ADD A,A,R0	ADD A,@RW0+d8	SUB A,A,R0	SUB A,@RW0+d8	ADDC A,A,R0	ADDC A,@RW0+d8	CMP A,R0	CMP A,@RW0+d8	AND A,A,R0	AND A,@RW0+d8	OR A,A,R0	OR A,@RW0+d8	XOR A,A,R0	XOR A,@RW0+d8	DBNZ R0,r	DBNZ @RW0+d8,r
+ 1	ADD A,A,R1	ADD A,@RW1+d8	SUB A,A,R1	SUB A,@RW1+d8	ADDC A,A,R1	ADDC A,@RW1+d8	CMP A,R1	CMP A,@RW1+d8	AND A,A,R1	AND A,@RW1+d8	OR A,A,R1	OR A,@RW1+d8	XOR A,A,R1	XOR A,@RW1+d8	DBNZ R1,r	DBNZ @RW1+d8,r
+ 2	ADD A,A,R2	ADD A,@RW2+d8	SUB A,A,R2	SUB A,@RW2+d8	ADDC A,A,R2	ADDC A,@RW2+d8	CMP A,R2	CMP A,@RW2+d8	AND A,A,R2	AND A,@RW2+d8	OR A,A,R2	OR A,@RW2+d8	XOR A,A,R2	XOR A,@RW2+d8	DBNZ R2,r	DBNZ @RW2+d8,r
+ 3	ADD A,A,R3	ADD A,@RW3+d8	SUB A,A,R3	SUB A,@RW3+d8	ADDC A,A,R3	ADDC A,@RW3+d8	CMP A,R3	CMP A,@RW3+d8	AND A,A,R3	AND A,@RW3+d8	OR A,A,R3	OR A,@RW3+d8	XOR A,A,R3	XOR A,@RW3+d8	DBNZ R3,r	DBNZ @RW3+d8,r
+ 4	ADD A,A,R4	ADD A,@RW4+d8	SUB A,A,R4	SUB A,@RW4+d8	ADDC A,A,R4	ADDC A,@RW4+d8	CMP A,R4	CMP A,@RW4+d8	AND A,A,R4	AND A,@RW4+d8	OR A,A,R4	OR A,@RW4+d8	XOR A,A,R4	XOR A,@RW4+d8	DBNZ R4,r	DBNZ @RW4+d8,r
+ 5	ADD A,A,R5	ADD A,@RW5+d8	SUB A,A,R5	SUB A,@RW5+d8	ADDC A,A,R5	ADDC A,@RW5+d8	CMP A,R5	CMP A,@RW5+d8	AND A,A,R5	AND A,@RW5+d8	OR A,A,R5	OR A,@RW5+d8	XOR A,A,R5	XOR A,@RW5+d8	DBNZ R5,r	DBNZ @RW5+d8,r
+ 6	ADD A,A,R6	ADD A,@RW6+d8	SUB A,A,R6	SUB A,@RW6+d8	ADDC A,A,R6	ADDC A,@RW6+d8	CMP A,R6	CMP A,@RW6+d8	AND A,A,R6	AND A,@RW6+d8	OR A,A,R6	OR A,@RW6+d8	XOR A,A,R6	XOR A,@RW6+d8	DBNZ R6,r	DBNZ @RW6+d8,r
+ 7	ADD A,A,R7	ADD A,@RW7+d8	SUB A,A,R7	SUB A,@RW7+d8	ADDC A,A,R7	ADDC A,@RW7+d8	CMP A,R7	CMP A,@RW7+d8	AND A,A,R7	AND A,@RW7+d8	OR A,A,R7	OR A,@RW7+d8	XOR A,A,R7	XOR A,@RW7+d8	DBNZ R7,r	DBNZ @RW7+d8,r
+ 8	ADD A,@RW0	ADD A,@RW0+d16	SUB A,@RW0	SUB A,@RW0+d16	ADDC A,@RW0	ADDC A,@RW0+d16	CMP A,@RW0	CMP A,@RW0+d16	AND A,@RW0	AND A,@RW0+d16	OR A,@RW0	OR A,@RW0+d16	XOR A,@RW0	XOR A,@RW0+d16	DBNZ @RW0,r	DBNZ @RW0+d16,r
+ 9	ADD A,@RW1	ADD A,@RW1+d16	SUB A,@RW1	SUB A,@RW1+d16	ADDC A,@RW1	ADDC A,@RW1+d16	CMP A,@RW1	CMP A,@RW1+d16	AND A,@RW1	AND A,@RW1+d16	OR A,@RW1	OR A,@RW1+d16	XOR A,@RW1	XOR A,@RW1+d16	DBNZ @RW1,r	DBNZ @RW1+d16,r
+ A	ADD A,@RW2	ADD A,@RW2+d16	SUB A,@RW2	SUB A,@RW2+d16	ADDC A,@RW2	ADDC A,@RW2+d16	CMP A,@RW2	CMP A,@RW2+d16	AND A,@RW2	AND A,@RW2+d16	OR A,@RW2	OR A,@RW2+d16	XOR A,@RW2	XOR A,@RW2+d16	DBNZ @RW2,r	DBNZ @RW2+d16,r
+ B	ADD A,@RW3	ADD A,@RW3+d16	SUB A,@RW3	SUB A,@RW3+d16	ADDC A,@RW3	ADDC A,@RW3+d16	CMP A,@RW3	CMP A,@RW3+d16	AND A,@RW3	AND A,@RW3+d16	OR A,@RW3	OR A,@RW3+d16	XOR A,@RW3	XOR A,@RW3+d16	DBNZ @RW3,r	DBNZ @RW3+d16,r
+ C	ADD A,@RW0+	ADD A,@RW0+RW7	SUB A,@RW0+	SUB A,@RW0+RW7	ADDC A,@RW0+	ADDC A,@RW0+RW7	CMP A,@RW0+	CMP A,@RW0+RW7	AND A,@RW0+	AND A,@RW0+RW7	OR A,@RW0+	OR A,@RW0+RW7	XOR A,@RW0+	XOR A,@RW0+RW7	DBNZ @RW0+,r	DBNZ @RW0+RW7,r
+ D	ADD A,@RW1+	ADD A,@RW1+RW7	SUB A,@RW1+	SUB A,@RW1+RW7	ADDC A,@RW1+	ADDC A,@RW1+RW7	CMP A,@RW1+	CMP A,@RW1+RW7	AND A,@RW1+	AND A,@RW1+RW7	OR A,@RW1+	OR A,@RW1+RW7	XOR A,@RW1+	XOR A,@RW1+RW7	DBNZ @RW1+,r	DBNZ @RW1+RW7,r
+ E	ADD A,@RW2+	ADD A,@PC+d16	SUB A,@RW2+	SUB A,@PC+d16	ADDC A,@RW2+	ADDC A,@PC+d16	CMP A,@RW2+	CMP A,@PC+d16	AND A,@RW2+	AND A,@PC+d16	OR A,@RW2+	OR A,@PC+d16	XOR A,@RW2+	XOR A,@PC+d16	DBNZ @RW2+,r	DBNZ @PC+d16,r
+ F	ADD A,@RW3+	ADD A,addr16	SUB A,@RW3+	SUB A,addr16	ADDC A,@RW3+	ADDC A,addr16	CMP A,@RW3+	CMP A,addr16	AND A,@RW3+	AND A,addr16	OR A,@RW3+	OR A,addr16	XOR A,@RW3+	XOR A,addr16	DBNZ @RW3+,r	DBNZ addr16,r

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-11 ea Instruction 6 (first byte = 75_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	ADD R0,A	ADD @R W0+d8,A	SUB R0,A	SUB @R W0+d8,A	SUBC A,R0	SUBC A, @RW0+d8	NEG R0	NEG @RW0+d8	AND R0,A	AND @R W0+d8,A	OR R0,A	OR @R W0+d8,A	XOR R0,A	XOR @R W0+d8,A	NOT R0	NOT @RW0+d8
+ 1	ADD R1,A	ADD @R W1+d8,A	SUB R1,A	SUB @R W1+d8,A	SUBC A,R1	SUBC A, @RW1+d8	NEG R1	NEG @RW1+d8	AND R1,A	AND @R W1+d8,A	OR R1,A	OR @R W1+d8,A	XOR R1,A	XOR @R W1+d8,A	NOT R1	NOT @RW1+d8
+ 2	ADD R2,A	ADD @R W2+d8,A	SUB R2,A	SUB @R W2+d8,A	SUBC A,R2	SUBC A, @RW2+d8	NEG R2	NEG @RW2+d8	AND R2,A	AND @R W2+d8,A	OR R2,A	OR @R W2+d8,A	XOR R2,A	XOR @R W2+d8,A	NOT R2	NOT @RW2+d8
+ 3	ADD R3,A	ADD @R W3+d8,A	SUB R3,A	SUB @R W3+d8,A	SUBC A,R3	SUBC A, @RW3+d8	NEG R3	NEG @RW3+d8	AND R3,A	AND @R W3+d8,A	OR R3,A	OR @R W3+d8,A	XOR R3,A	XOR @R W3+d8,A	NOT R3	NOT @RW3+d8
+ 4	ADD R4,A	ADD @R W4+d8,A	SUB R4,A	SUB @R W4+d8,A	SUBC A,R4	SUBC A, @RW4+d8	NEG R4	NEG @RW4+d8	AND R4,A	AND @R W4+d8,A	OR R4,A	OR @R W4+d8,A	XOR R4,A	XOR @R W4+d8,A	NOT R4	NOT @RW4+d8
+ 5	ADD R5,A	ADD @R W5+d8,A	SUB R5,A	SUB @R W5+d8,A	SUBC A,R5	SUBC A, @RW5+d8	NEG R5	NEG @RW5+d8	AND R5,A	AND @R W5+d8,A	OR R5,A	OR @R W5+d8,A	XOR R5,A	XOR @R W5+d8,A	NOT R5	NOT @RW5+d8
+ 6	ADD R6,A	ADD @R W6+d8,A	SUB R6,A	SUB @R W6+d8,A	SUBC A,R6	SUBC A, @RW6+d8	NEG R6	NEG @RW6+d8	AND R6,A	AND @R W6+d8,A	OR R6,A	OR @R W6+d8,A	XOR R6,A	XOR @R W6+d8,A	NOT R6	NOT @RW6+d8
+ 7	ADD R7,A	ADD @R W7+d8,A	SUB R7,A	SUB @R W7+d8,A	SUBC A,R7	SUBC A, @RW7+d8	NEG R7	NEG @RW7+d8	AND R7,A	AND @R W7+d8,A	OR R7,A	OR @R W7+d8,A	XOR R7,A	XOR @R W7+d8,A	NOT R7	NOT @RW7+d8
+ 8	ADD @RW0,A	ADD @R W0+d16,A	SUB @RW0,A	SUB @R W0+d16,A	SUBC A,@RW0	SUBC A, @RW0+d16	NEG @RW0	NEG @RW0+d16	AND @RW0,A	AND @R W0+d16,A	OR @RW0,A	OR @R W0+d16,A	XOR @RW0,A	XOR @R W0+d16,A	NOT @RW0	NOT @RW0+d16
+ 9	ADD @RW1,A	ADD @R W1+d16,A	SUB @RW1,A	SUB @R W1+d16,A	SUBC A,@RW1	SUBC A, @RW1+d16	NEG @RW1	NEG @RW1+d16	AND @RW1,A	AND @R W1+d16,A	OR @RW1,A	OR @R W1+d16,A	XOR @RW1,A	XOR @R W1+d16,A	NOT @RW1	NOT @RW1+d16
+ A	ADD @RW2,A	ADD @R W2+d16,A	SUB @RW2,A	SUB @R W2+d16,A	SUBC A,@RW2	SUBC A, @RW2+d16	NEG @RW2	NEG @RW2+d16	AND @RW2,A	AND @R W2+d16,A	OR @RW2,A	OR @R W2+d16,A	XOR @RW2,A	XOR @R W2+d16,A	NOT @RW2	NOT @RW2+d16
+ B	ADD @RW3,A	ADD @R W3+d16,A	SUB @RW3,A	SUB @R W3+d16,A	SUBC A,@RW3	SUBC A, @RW3+d16	NEG @RW3	NEG @RW3+d16	AND @RW3,A	AND @R W3+d16,A	OR @RW3,A	OR @R W3+d16,A	XOR @RW3,A	XOR @R W3+d16,A	NOT @RW3	NOT @RW3+d16
+ C	ADD @RW0+,A	ADD @R W0+RW7,A	SUB @RW0+,A	SUB @R W0+RW7,A	SUBC A,@RW0+	SUBC A, @RW0+RW7	NEG @RW0+	NEG @RW0+RW7	AND @RW0+,A	AND @R W0+RW7,A	OR @RW0+,A	OR @R W0+RW7,A	XOR @RW0+,A	XOR @R W0+RW7,A	NOT @RW0+	NOT @RW0+RW7
+ D	ADD @RW1+,A	ADD @R W1+RW7,A	SUB @RW1+,A	SUB @R W1+RW7,A	SUBC A,@RW1+	SUBC A, @RW1+RW7	NEG @RW1+	NEG @RW1+RW7	AND @RW1+,A	AND @R W1+RW7,A	OR @RW1+,A	OR @R W1+RW7,A	XOR @RW1+,A	XOR @R W1+RW7,A	NOT @RW1+	NOT @RW1+RW7
+ E	ADD @RW2+,A	ADD @P C+d16,A	SUB @RW2+,A	SUB @P C+d16,A	SUBC A,@RW2+	SUBC A, @PC+d16	NEG @RW2+	NEG @PC+d16	AND @RW2+,A	AND @P C+d16,A	OR @RW2+,A	OR @P C+d16,A	XOR @RW2+,A	XOR @P C+d16,A	NOT @RW2+	NOT @PC+d16
+ F	ADD @RW3+,A	ADD addr16,A	SUB @RW3+,A	SUB addr16,A	SUBC A,@RW3+	SUBC A, addr16	NEG @RW3+	NEG addr16	AND @RW3+,A	AND addr16,A	OR @RW3+,A	OR addr16,A	XOR @RW3+,A	XOR addr16,A	NOT @RW3+	NOT addr16

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-12 ea Instruction 7 (first byte = 76_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	ADDW A,A,RW0	ADDW A,@RW0+d8	SUBW A,A,RW0	SUBW A,@RW0+d8	ADDCW A,A,RW0	ADDCW A,@RW0+d8	CMPW A,A,RW0	CMPW A,@RW0+d8	ANDW A,A,RW0	ANDW A,@RW0+d8	ORW A,A,RW0	ORW A,@RW0+d8	XORW A,A,RW0	XORW A,@RW0+d8	DWBNZ RW0,r	DWBNZ @RW0+d8,r
+ 1	ADDW A,A,RW1	ADDW A,@RW1+d8	SUBW A,A,RW1	SUBW A,@RW1+d8	ADDCW A,A,RW1	ADDCW A,@RW1+d8	CMPW A,A,RW1	CMPW A,@RW1+d8	ANDW A,A,RW1	ANDW A,@RW1+d8	ORW A,A,RW1	ORW A,@RW1+d8	XORW A,A,RW1	XORW A,@RW1+d8	DWBNZ RW1,r	DWBNZ @RW1+d8,r
+ 2	ADDW A,A,RW2	ADDW A,@RW2+d8	SUBW A,A,RW2	SUBW A,@RW2+d8	ADDCW A,A,RW2	ADDCW A,@RW2+d8	CMPW A,A,RW2	CMPW A,@RW2+d8	ANDW A,A,RW2	ANDW A,@RW2+d8	ORW A,A,RW2	ORW A,@RW2+d8	XORW A,A,RW2	XORW A,@RW2+d8	DWBNZ RW2,r	DWBNZ @RW2+d8,r
+ 3	ADDW A,A,RW3	ADDW A,@RW3+d8	SUBW A,A,RW3	SUBW A,@RW3+d8	ADDCW A,A,RW3	ADDCW A,@RW3+d8	CMPW A,A,RW3	CMPW A,@RW3+d8	ANDW A,A,RW3	ANDW A,@RW3+d8	ORW A,A,RW3	ORW A,@RW3+d8	XORW A,A,RW3	XORW A,@RW3+d8	DWBNZ RW3,r	DWBNZ @RW3+d8,r
+ 4	ADDW A,A,RW4	ADDW A,@RW4+d8	SUBW A,A,RW4	SUBW A,@RW4+d8	ADDCW A,A,RW4	ADDCW A,@RW4+d8	CMPW A,A,RW4	CMPW A,@RW4+d8	ANDW A,A,RW4	ANDW A,@RW4+d8	ORW A,A,RW4	ORW A,@RW4+d8	XORW A,A,RW4	XORW A,@RW4+d8	DWBNZ RW4,r	DWBNZ @RW4+d8,r
+ 5	ADDW A,A,RW5	ADDW A,@RW5+d8	SUBW A,A,RW5	SUBW A,@RW5+d8	ADDCW A,A,RW5	ADDCW A,@RW5+d8	CMPW A,A,RW5	CMPW A,@RW5+d8	ANDW A,A,RW5	ANDW A,@RW5+d8	ORW A,A,RW5	ORW A,@RW5+d8	XORW A,A,RW5	XORW A,@RW5+d8	DWBNZ RW5,r	DWBNZ @RW5+d8,r
+ 6	ADDW A,A,RW6	ADDW A,@RW6+d8	SUBW A,A,RW6	SUBW A,@RW6+d8	ADDCW A,A,RW6	ADDCW A,@RW6+d8	CMPW A,A,RW6	CMPW A,@RW6+d8	ANDW A,A,RW6	ANDW A,@RW6+d8	ORW A,A,RW6	ORW A,@RW6+d8	XORW A,A,RW6	XORW A,@RW6+d8	DWBNZ RW6,r	DWBNZ @RW6+d8,r
+ 7	ADDW A,A,RW7	ADDW A,@RW7+d8	SUBW A,A,RW7	SUBW A,@RW7+d8	ADDCW A,A,RW7	ADDCW A,@RW7+d8	CMPW A,A,RW7	CMPW A,@RW7+d8	ANDW A,A,RW7	ANDW A,@RW7+d8	ORW A,A,RW7	ORW A,@RW7+d8	XORW A,A,RW7	XORW A,@RW7+d8	DWBNZ RW7,r	DWBNZ @RW7+d8,r
+ 8	ADDW A,A,@RW0	ADDW A,@RW0+d16	SUBW A,A,@RW0	SUBW A,@RW0+d16	ADDCW A,A,@RW0	ADDCW A,@RW0+d16	CMPW A,A,@RW0	CMPW A,@RW0+d16	ANDW A,A,@RW0	ANDW A,@RW0+d16	ORW A,A,@RW0	ORW A,@RW0+d16	XORW A,A,@RW0	XORW A,@RW0+d16	DWBNZ @RW0,r	DWBNZ @R W0+d16,r
+ 9	ADDW A,A,@RW1	ADDW A,@RW1+d16	SUBW A,A,@RW1	SUBW A,@RW1+d16	ADDCW A,A,@RW1	ADDCW A,@RW1+d16	CMPW A,A,@RW1	CMPW A,@RW1+d16	ANDW A,A,@RW1	ANDW A,@RW1+d16	ORW A,A,@RW1	ORW A,@RW1+d16	XORW A,A,@RW1	XORW A,@RW1+d16	DWBNZ @RW1,r	DWBNZ @R W1+d16,r
+ A	ADDW A,A,@RW2	ADDW A,@RW2+d16	SUBW A,A,@RW2	SUBW A,@RW2+d16	ADDCW A,A,@RW2	ADDCW A,@RW2+d16	CMPW A,A,@RW2	CMPW A,@RW2+d16	ANDW A,A,@RW2	ANDW A,@RW2+d16	ORW A,A,@RW2	ORW A,@RW2+d16	XORW A,A,@RW2	XORW A,@RW2+d16	DWBNZ @RW2,r	DWBNZ @R W2+d16,r
+ B	ADDW A,A,@RW3	ADDW A,@RW3+d16	SUBW A,A,@RW3	SUBW A,@RW3+d16	ADDCW A,A,@RW3	ADDCW A,@RW3+d16	CMPW A,A,@RW3	CMPW A,@RW3+d16	ANDW A,A,@RW3	ANDW A,@RW3+d16	ORW A,A,@RW3	ORW A,@RW3+d16	XORW A,A,@RW3	XORW A,@RW3+d16	DWBNZ @RW3,r	DWBNZ @R W3+d16,r
+ C	ADDW A,A,@RW0+	ADDW A,@RW0+RW7	SUBW A,A,@RW0+	SUBW A,@RW0+RW7	ADDCW A,A,@RW0+	ADDCW A,@RW0+RW7	CMPW A,A,@RW0+	CMPW A,@RW0+RW7	ANDW A,A,@RW0+	ANDW A,@RW0+RW7	ORW A,A,@RW0+	ORW A,@RW0+RW7	XORW A,A,@RW0+	XORW A,@RW0+RW7	DWBNZ @RW0+,r	DWBNZ @R W0+RW7,r
+ D	ADDW A,A,@RW1+	ADDW A,@RW1+RW7	SUBW A,A,@RW1+	SUBW A,@RW1+RW7	ADDCW A,A,@RW1+	ADDCW A,@RW1+RW7	CMPW A,A,@RW1+	CMPW A,@RW1+RW7	ANDW A,A,@RW1+	ANDW A,@RW1+RW7	ORW A,A,@RW1+	ORW A,@RW1+RW7	XORW A,A,@RW1+	XORW A,@RW1+RW7	DWBNZ @RW1+,r	DWBNZ @R W1+RW7,r
+ E	ADDW A,A,@RW2+	ADDW A,@PC+d16	SUBW A,A,@RW2+	SUBW A,@PC+d16	ADDCW A,A,@RW2+	ADDCW A,@PC+d16	CMPW A,A,@RW2+	CMPW A,@PC+d16	ANDW A,A,@RW2+	ANDW A,@PC+d16	ORW A,A,@RW2+	ORW A,@PC+d16	XORW A,A,@RW2+	XORW A,@PC+d16	DWBNZ @RW2+,r	DWBNZ @R PC+d16,r
+ F	ADDW A,A,@RW3+	ADDW A,addr16	SUBW A,A,@RW3+	SUBW A,addr16	ADDCW A,A,@RW3+	ADDCW A,addr16	CMPW A,A,@RW3+	CMPW A,addr16	ANDW A,A,@RW3+	ANDW A,addr16	ORW A,A,@RW3+	ORW A,addr16	XORW A,A,@RW3+	XORW A,addr16	DWBNZ @RW3+,r	DWBNZ addr16,r

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-13 ea Instruction 8 (first byte = 77_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	RW0,A	W0+d8,A	RW0,A	W0+d8,A	A,RW0	@RW0+d8	RW0	@RW0+d8	RW0,A	W0+d8,A	RW0,A	W0+d8,A	RW0,A	W0+d8,A	RW0	@RW0+d8
+ 1	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	RW1,A	W1+d8,A	RW1,A	W1+d8,A	A,RW1	@RW1+d8	RW1	@RW1+d8	RW1,A	W1+d8,A	RW1,A	W1+d8,A	RW1,A	W1+d8,A	RW1	@RW1+d8
+ 2	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	RW2,A	W2+d8,A	RW2,A	W2+d8,A	A,RW2	@RW2+d8	RW2	@RW2+d8	RW2,A	W2+d8,A	RW2,A	W2+d8,A	RW2,A	W2+d8,A	RW2	@RW2+d8
+ 3	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	RW3,A	W3+d8,A	RW3,A	W3+d8,A	A,RW3	@RW3+d8	RW3	@RW3+d8	RW3,A	W3+d8,A	RW3,A	W3+d8,A	RW3,A	W3+d8,A	RW3	@RW3+d8
+ 4	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	RW4,A	W4+d8,A	RW4,A	W4+d8,A	A,RW4	@RW4+d8	RW4	@RW4+d8	RW4,A	W4+d8,A	RW4,A	W4+d8,A	RW4,A	W4+d8,A	RW4	@RW4+d8
+ 5	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	RW5,A	W5+d8,A	RW5,A	W5+d8,A	A,RW5	@RW5+d8	RW5	@RW5+d8	RW5,A	W5+d8,A	RW5,A	W5+d8,A	RW5,A	W5+d8,A	RW5	@RW5+d8
+ 6	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	RW6,A	W6+d8,A	RW6,A	W6+d8,A	A,RW6	@RW6+d8	RW6	@RW6+d8	RW6,A	W6+d8,A	RW6,A	W6+d8,A	RW6,A	W6+d8,A	RW6	@RW6+d8
+ 7	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	RW7,A	W7+d8,A	RW7,A	W7+d8,A	A,RW7	@RW7+d8	RW7	@RW7+d8	RW7,A	W7+d8,A	RW7,A	W7+d8,A	RW7,A	W7+d8,A	RW7	@RW7+d8
+ 8	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	@RW0,A	W0+d16,A	@RW0,A	W0+d16,A	A,@RW0	@RW0+d16	@RW0	@RW0+d16	@RW0,A	W0+d16,A	@RW0,A	W0+d16,A	@RW0,A	W0+d16,A	@RW0	@RW0+d16
+ 9	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	@RW1,A	W1+d16,A	@RW1,A	W1+d16,A	A,@RW1	@RW1+d16	@RW1	@RW1+d16	@RW1,A	W1+d16,A	@RW1,A	W1+d16,A	@RW1,A	W1+d16,A	@RW1	@RW1+d16
+ A	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	@RW2,A	W2+d16,A	@RW2,A	W2+d16,A	A,@RW2	@RW2+d16	@RW2	@RW2+d16	@RW2,A	W2+d16,A	@RW2,A	W2+d16,A	@RW2,A	W2+d16,A	@RW2	@RW2+d16
+ B	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	@RW3,A	W3+d16,A	@RW3,A	W3+d16,A	A,@RW3	@RW3+d16	@RW3	@RW3+d16	@RW3,A	W3+d16,A	@RW3,A	W3+d16,A	@RW3,A	W3+d16,A	@RW3	@RW3+d16
+ C	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	@RW0+,A	W0+RW7,A	@RW0+,A	W0+RW7,A	A,@RW0+	@RW0+RW7	@RW0+	@RW0+RW7	@RW0+,A	W0+RW7,A	@RW0+,A	W0+RW7,A	@RW0+,A	W0+RW7,A	@RW0+	@RW0+RW7
+ D	ADDW	ADDW @R	SUBW	SUBW @R	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW @R	ORW	ORW @R	XORW	XORW @R	NOTW	NOTW
	@RW1+,A	W1+RW7,A	@RW1+,A	W1+RW7,A	A,@RW1+	@RW1+RW7	@RW1+	@RW1+RW7	@RW1+,A	W1+RW7,A	@RW1+,A	W1+RW7,A	@RW1+,A	W1+RW7,A	@RW1+	@RW1+RW7
+ E	ADDW	ADDW @P	SUBW	SUBW @P	SUBCW	SUBCW A,	NEGW	@PC+d16	ANDW	ANDW @P	ORW	ORW @P	XORW	XORW @P	NOTW	NOTW
	@RW2+,A	C+d16,A	@RW2+,A	C+d16,A	A,@RW2+	@PC+d16	@RW2+	@PC+d16	@RW2+,A	C+d16,A	@RW2+,A	C+d16,A	@RW2+,A	C+d16,A	@RW2+	@PC+d16
+ F	ADDW	ADDW	SUBW	SUBW	SUBCW	SUBCW A,	NEGW	NEGW	ANDW	ANDW	ORW	ORW	XORW	XORW	NOTW	NOTW
	@RW3+,A	addr16,A	@RW3+,A	addr16,A	A,@RW3+	addr16	@RW3+	addr16	@RW3+,A	addr16,A	@RW3+,A	addr16,A	@RW3+,A	addr16,A	@RW3+	addr16

et4U.com

DataShee

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-14 ea Instruction 9 (first byte = 78_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	MULU A,A,R0	MULU A,@RW0+d8	MULUW A,A,R0	MULUW A,@RW0+d8	MUL A,A,R0	MUL A,@RW0+d8	MULW A,A,R0	MULW A,@RW0+d8	DIVU A,A,R0	DIVU A,@RW0+d8	DIVUW A,A,R0	DIVUW A,@RW0+d8	DIV A,A,R0	DIV A,@RW0+d8	DIVW A,A,R0	DIVW A,@RW0+d8
+ 1	MULU A,A,R1	MULU A,@RW1+d8	MULUW A,A,R1	MULUW A,@RW1+d8	MUL A,A,R1	MUL A,@RW1+d8	MULW A,A,R1	MULW A,@RW1+d8	DIVU A,A,R1	DIVU A,@RW1+d8	DIVUW A,A,R1	DIVUW A,@RW1+d8	DIV A,A,R1	DIV A,@RW1+d8	DIVW A,A,R1	DIVW A,@RW1+d8
+ 2	MULU A,A,R2	MULU A,@RW2+d8	MULUW A,A,R2	MULUW A,@RW2+d8	MUL A,A,R2	MUL A,@RW2+d8	MULW A,A,R2	MULW A,@RW2+d8	DIVU A,A,R2	DIVU A,@RW2+d8	DIVUW A,A,R2	DIVUW A,@RW2+d8	DIV A,A,R2	DIV A,@RW2+d8	DIVW A,A,R2	DIVW A,@RW2+d8
+ 3	MULU A,A,R3	MULU A,@RW3+d8	MULUW A,A,R3	MULUW A,@RW3+d8	MUL A,A,R3	MUL A,@RW3+d8	MULW A,A,R3	MULW A,@RW3+d8	DIVU A,A,R3	DIVU A,@RW3+d8	DIVUW A,A,R3	DIVUW A,@RW3+d8	DIV A,A,R3	DIV A,@RW3+d8	DIVW A,A,R3	DIVW A,@RW3+d8
+ 4	MULU A,A,R4	MULU A,@RW4+d8	MULUW A,A,R4	MULUW A,@RW4+d8	MUL A,A,R4	MUL A,@RW4+d8	MULW A,A,R4	MULW A,@RW4+d8	DIVU A,A,R4	DIVU A,@RW4+d8	DIVUW A,A,R4	DIVUW A,@RW4+d8	DIV A,A,R4	DIV A,@RW4+d8	DIVW A,A,R4	DIVW A,@RW4+d8
+ 5	MULU A,A,R5	MULU A,@RW5+d8	MULUW A,A,R5	MULUW A,@RW5+d8	MUL A,A,R5	MUL A,@RW5+d8	MULW A,A,R5	MULW A,@RW5+d8	DIVU A,A,R5	DIVU A,@RW5+d8	DIVUW A,A,R5	DIVUW A,@RW5+d8	DIV A,A,R5	DIV A,@RW5+d8	DIVW A,A,R5	DIVW A,@RW5+d8
+ 6	MULU A,A,R6	MULU A,@RW6+d8	MULUW A,A,R6	MULUW A,@RW6+d8	MUL A,A,R6	MUL A,@RW6+d8	MULW A,A,R6	MULW A,@RW6+d8	DIVU A,A,R6	DIVU A,@RW6+d8	DIVUW A,A,R6	DIVUW A,@RW6+d8	DIV A,A,R6	DIV A,@RW6+d8	DIVW A,A,R6	DIVW A,@RW6+d8
+ 7	MULU A,A,R7	MULU A,@RW7+d8	MULUW A,A,R7	MULUW A,@RW7+d8	MUL A,A,R7	MUL A,@RW7+d8	MULW A,A,R7	MULW A,@RW7+d8	DIVU A,A,R7	DIVU A,@RW7+d8	DIVUW A,A,R7	DIVUW A,@RW7+d8	DIV A,A,R7	DIV A,@RW7+d8	DIVW A,A,R7	DIVW A,@RW7+d8
+ 8	MULU A,A,@RW0	MULU A,@RW0+d16	MULUW A,A,@RW0	MULUW A,@RW0+d16	MUL A,A,@RW0	MUL A,@RW0+d16	MULW A,A,@RW0	MULW A,@RW0+d16	DIVU A,A,@RW0	DIVU A,@RW0+d16	DIVUW A,A,@RW0	DIVUW A,@RW0+d16	DIV A,A,@RW0	DIV A,@RW0+d16	DIVW A,A,@RW0	DIVW A,@RW0+d16
+ 9	MULU A,A,@RW1	MULU A,@RW1+d16	MULUW A,A,@RW1	MULUW A,@RW1+d16	MUL A,A,@RW1	MUL A,@RW1+d16	MULW A,A,@RW1	MULW A,@RW1+d16	DIVU A,A,@RW1	DIVU A,@RW1+d16	DIVUW A,A,@RW1	DIVUW A,@RW1+d16	DIV A,A,@RW1	DIV A,@RW1+d16	DIVW A,A,@RW1	DIVW A,@RW1+d16
+ A	MULU A,A,@RW2	MULU A,@RW2+d16	MULUW A,A,@RW2	MULUW A,@RW2+d16	MUL A,A,@RW2	MUL A,@RW2+d16	MULW A,A,@RW2	MULW A,@RW2+d16	DIVU A,A,@RW2	DIVU A,@RW2+d16	DIVUW A,A,@RW2	DIVUW A,@RW2+d16	DIV A,A,@RW2	DIV A,@RW2+d16	DIVW A,A,@RW2	DIVW A,@RW2+d16
+ B	MULU A,A,@RW3	MULU A,@RW3+d16	MULUW A,A,@RW3	MULUW A,@RW3+d16	MUL A,A,@RW3	MUL A,@RW3+d16	MULW A,A,@RW3	MULW A,@RW3+d16	DIVU A,A,@RW3	DIVU A,@RW3+d16	DIVUW A,A,@RW3	DIVUW A,@RW3+d16	DIV A,A,@RW3	DIV A,@RW3+d16	DIVW A,A,@RW3	DIVW A,@RW3+d16
+ C	MULU A,A,@RW0+	MULU A,@RW0+RW7	MULUW A,A,@RW0+	MULUW A,@RW0+RW7	MUL A,A,@RW0+	MUL A,@RW0+RW7	MULW A,A,@RW0+	MULW A,@RW0+RW7	DIVU A,A,@RW0+	DIVU A,@RW0+RW7	DIVUW A,A,@RW0+	DIVUW A,@RW0+RW7	DIV A,A,@RW0+	DIV A,@RW0+RW7	DIVW A,A,@RW0+	DIVW A,@RW0+RW7
+ D	MULU A,A,@RW1+	MULU A,@RW1+RW7	MULUW A,A,@RW1+	MULUW A,@RW1+RW7	MUL A,A,@RW1+	MUL A,@RW1+RW7	MULW A,A,@RW1+	MULW A,@RW1+RW7	DIVU A,A,@RW1+	DIVU A,@RW1+RW7	DIVUW A,A,@RW1+	DIVUW A,@RW1+RW7	DIV A,A,@RW1+	DIV A,@RW1+RW7	DIVW A,A,@RW1+	DIVW A,@RW1+RW7
+ E	MULU A,A,@RW2+	MULU A,@PC+d16	MULUW A,A,@RW2+	MULUW A,@PC+d16	MUL A,A,@RW2+	MUL A,@PC+d16	MULW A,A,@RW2+	MULW A,@PC+d16	DIVU A,A,@RW2+	DIVU A,@PC+d16	DIVUW A,A,@RW2+	DIVUW A,@PC+d16	DIV A,A,@RW2+	DIV A,@PC+d16	DIVW A,A,@RW2+	DIVW A,@PC+d16
+ F	MULU A,A,@RW3+	MULU A,addr16	MULUW A,A,@RW3+	MULUW A,addr16	MUL A,A,@RW3+	MUL A,addr16	MULW A,A,@RW3+	MULW A,addr16	DIVU A,A,@RW3+	DIVU A,addr16	DIVUW A,A,@RW3+	DIVUW A,addr16	DIV A,A,@RW3+	DIV A,addr16	DIVW A,A,@RW3+	DIVW A,addr16

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-15 MOVEA RWi, ea Instruction (first byte = 79_H)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	MOVEA RW0,RW0	MOVEA RW0 ,@RW0+d8	MOVEA RW1,@RW0	MOVEA RW1 ,@RW0+d8	MOVEA RW2,RW0	MOVEA RW2 ,@RW0+d8	MOVEA RW3,RW0	MOVEA RW3 ,@RW0+d8	MOVEA RW4,RW0	MOVEA RW4 ,@RW0+d8	MOVEA R5,RW0	MOVEA RW5 ,@RW0+d8	MOVEA R6,RW0	MOVEA RW6 ,@RW0+d8	MOVEA R7,RW0	MOVEA RW7 ,@RW0+d8
+ 1	MOVEA RW0,RW1	MOVEA RW0 ,@RW1+d8	MOVEA RW1,RW1	MOVEA RW1 ,@RW1+d8	MOVEA RW2,RW1	MOVEA RW2 ,@RW1+d8	MOVEA RW3,RW1	MOVEA RW3 ,@RW1+d8	MOVEA RW4,RW1	MOVEA RW4 ,@RW1+d8	MOVEA R5,RW1	MOVEA RW5 ,@RW1+d8	MOVEA R6,RW1	MOVEA RW6 ,@RW1+d8	MOVEA R7,RW1	MOVEA RW7 ,@RW1+d8
+ 2	MOVEA RW0,RW2	MOVEA RW0 ,@RW2+d8	MOVEA RW1,RW2	MOVEA RW1 ,@RW2+d8	MOVEA RW2,RW2	MOVEA RW2 ,@RW2+d8	MOVEA RW3,RW2	MOVEA RW3 ,@RW2+d8	MOVEA RW4,RW2	MOVEA RW4 ,@RW2+d8	MOVEA R5,RW2	MOVEA RW5 ,@RW2+d8	MOVEA R6,RW2	MOVEA RW6 ,@RW2+d8	MOVEA R7,RW2	MOVEA RW7 ,@RW2+d8
+ 3	MOVEA RW0,RW3	MOVEA RW0 ,@RW3+d8	MOVEA RW1,RW3	MOVEA RW1 ,@RW3+d8	MOVEA RW2,RW3	MOVEA RW2 ,@RW3+d8	MOVEA RW3,RW3	MOVEA RW3 ,@RW3+d8	MOVEA RW4,RW3	MOVEA RW4 ,@RW3+d8	MOVEA R5,RW3	MOVEA RW5 ,@RW3+d8	MOVEA R6,RW3	MOVEA RW6 ,@RW3+d8	MOVEA R7,RW3	MOVEA RW7 ,@RW3+d8
+ 4	MOVEA RW0,RW4	MOVEA RW0 ,@RW4+d8	MOVEA RW1,RW4	MOVEA RW1 ,@RW4+d8	MOVEA RW2,RW4	MOVEA RW2 ,@RW4+d8	MOVEA RW3,RW4	MOVEA RW3 ,@RW4+d8	MOVEA RW4,RW4	MOVEA RW4 ,@RW4+d8	MOVEA R5,RW4	MOVEA RW5 ,@RW4+d8	MOVEA R6,RW4	MOVEA RW6 ,@RW4+d8	MOVEA R7,RW4	MOVEA RW7 ,@RW4+d8
+ 5	MOVEA RW0,RW5	MOVEA RW0 ,@RW5+d8	MOVEA RW1,RW5	MOVEA RW1 ,@RW5+d8	MOVEA RW2,RW5	MOVEA RW2 ,@RW5+d8	MOVEA RW3,RW5	MOVEA RW3 ,@RW5+d8	MOVEA RW4,RW5	MOVEA RW4 ,@RW5+d8	MOVEA R5,RW5	MOVEA RW5 ,@RW5+d8	MOVEA R6,RW5	MOVEA RW6 ,@RW5+d8	MOVEA R7,RW5	MOVEA RW7 ,@RW5+d8
+ 6	MOVEA RW0,RW6	MOVEA RW0 ,@RW6+d8	MOVEA RW1,RW6	MOVEA RW1 ,@RW6+d8	MOVEA RW2,RW6	MOVEA RW2 ,@RW6+d8	MOVEA RW3,RW6	MOVEA RW3 ,@RW6+d8	MOVEA RW4,RW6	MOVEA RW4 ,@RW6+d8	MOVEA R5,RW6	MOVEA RW5 ,@RW6+d8	MOVEA R6,RW6	MOVEA RW6 ,@RW6+d8	MOVEA R7,RW6	MOVEA RW7 ,@RW6+d8
+ 7	MOVEA RW0,RW7	MOVEA RW0 ,@RW7+d8	MOVEA RW1,RW7	MOVEA RW1 ,@RW7+d8	MOVEA RW2,RW7	MOVEA RW2 ,@RW7+d8	MOVEA RW3,RW7	MOVEA RW3 ,@RW7+d8	MOVEA RW4,RW7	MOVEA RW4 ,@RW7+d8	MOVEA R5,RW7	MOVEA RW5 ,@RW7+d8	MOVEA R6,RW7	MOVEA RW6 ,@RW7+d8	MOVEA R7,RW7	MOVEA RW7 ,@RW7+d8
+ 8	MOVEA R0,@RW0	MOVEA RW0 ,@RW0+d16	MOVEA RW1,@RW0	MOVEA RW1 ,@RW0+d16	MOVEA RW2,@RW0	MOVEA RW2 ,@RW0+d16	MOVEA RW3,@RW0	MOVEA RW3 ,@RW0+d16	MOVEA RW4,@RW0	MOVEA RW4 ,@RW0+d16	MOVEA R5,@RW0	MOVEA RW5 ,@RW0+d16	MOVEA R6,@RW0	MOVEA RW6 ,@RW0+d16	MOVEA R7,@RW0	MOVEA RW7 ,@RW0+d16
+ 9	MOVEA R0,@RW1	MOVEA RW0 ,@RW1+d16	MOVEA RW1,@RW1	MOVEA RW1 ,@RW1+d16	MOVEA RW2,@RW1	MOVEA RW2 ,@RW1+d16	MOVEA RW3,@RW1	MOVEA RW3 ,@RW1+d16	MOVEA RW4,@RW1	MOVEA RW4 ,@RW1+d16	MOVEA R5,@RW1	MOVEA RW5 ,@RW1+d16	MOVEA R6,@RW1	MOVEA RW6 ,@RW1+d16	MOVEA R7,@RW1	MOVEA RW7 ,@RW1+d16
+ A	MOVEA R0,@RW2	MOVEA RW0 ,@RW2+d16	MOVEA RW1,@RW2	MOVEA RW1 ,@RW2+d16	MOVEA RW2,@RW2	MOVEA RW2 ,@RW2+d16	MOVEA RW3,@RW2	MOVEA RW3 ,@RW2+d16	MOVEA RW4,@RW2	MOVEA RW4 ,@RW2+d16	MOVEA R5,@RW2	MOVEA RW5 ,@RW2+d16	MOVEA R6,@RW2	MOVEA RW6 ,@RW2+d16	MOVEA R7,@RW2	MOVEA RW7 ,@RW2+d16
+ B	MOVEA R0,@RW3	MOVEA RW0 ,@RW3+d16	MOVEA RW1,@RW3	MOVEA RW1 ,@RW3+d16	MOVEA RW2,@RW3	MOVEA RW2 ,@RW3+d16	MOVEA RW3,@RW3	MOVEA RW3 ,@RW3+d16	MOVEA RW4,@RW3	MOVEA RW4 ,@RW3+d16	MOVEA R5,@RW3	MOVEA RW5 ,@RW3+d16	MOVEA R6,@RW3	MOVEA RW6 ,@RW3+d16	MOVEA R7,@RW3	MOVEA RW7 ,@RW3+d16
+ C	MOVEA R0,@RW0+	MOVEA RW0 ,@RW0+RW7	MOVEA R1,@RW0+	MOVEA RW1 ,@RW0+RW7	MOVEA R2,@RW0+	MOVEA RW2 ,@RW0+RW7	MOVEA R3,@RW0+	MOVEA RW3 ,@RW0+RW7	MOVEA R4,@RW0+	MOVEA RW4 ,@RW0+RW7	MOVEA R5,@RW0+	MOVEA RW5 ,@RW0+RW7	MOVEA R6,@RW0+	MOVEA RW6 ,@RW0+RW7	MOVEA R7,@RW0+	MOVEA RW7 ,@RW0+RW7
+ D	MOVEA R0,@RW1+	MOVEA RW0 ,@RW1+RW7	MOVEA R1,@RW1+	MOVEA RW1 ,@RW1+RW7	MOVEA R2,@RW1+	MOVEA RW2 ,@RW1+RW7	MOVEA R3,@RW1+	MOVEA RW3 ,@RW1+RW7	MOVEA R4,@RW1+	MOVEA RW4 ,@RW1+RW7	MOVEA R5,@RW1+	MOVEA RW5 ,@RW1+RW7	MOVEA R6,@RW1+	MOVEA RW6 ,@RW1+RW7	MOVEA R7,@RW1+	MOVEA RW7 ,@RW1+RW7
+ E	MOVEA R0,@RW2+	MOVEA RW0 ,@PC+d16	MOVEA R1,@RW2+	MOVEA RW1 ,@PC+d16	MOVEA R2,@RW2+	MOVEA RW2 ,@PC+d16	MOVEA R3,@RW2+	MOVEA RW3 ,@PC+d16	MOVEA R4,@RW2+	MOVEA RW4 ,@PC+d16	MOVEA R5,@RW2+	MOVEA RW5 ,@PC+d16	MOVEA R6,@RW2+	MOVEA RW6 ,@PC+d16	MOVEA R7,@RW2+	MOVEA RW7 ,@PC+d16
+ F	MOVEA R0,@RW3+	MOVEA RW0 ,addr16	MOVEA R1,@RW3+	MOVEA RW1 ,addr16	MOVEA R2,@RW3+	MOVEA RW2 ,addr16	MOVEA R3,@RW3+	MOVEA RW3 ,addr16	MOVEA R4,@RW3+	MOVEA RW4 ,addr16	MOVEA R5,@RW3+	MOVEA RW5 ,addr16	MOVEA R6,@RW3+	MOVEA RW6 ,addr16	MOVEA R7,@RW3+	MOVEA RW7 ,addr16

et4U.com
DataShee
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-16 MOV Ri, ea Instruction (first byte = $7A_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	MOV R0, R0,R0	MOV R0, @RW0+d8	MOV R1,R0	MOV R1, @RW0+d8	MOV R2,R0	MOV R2, @RW0+d8	MOV R3,R0	MOV R3, @RW0+d8	MOV R4,R0	MOV R4, @RW0+d8	MOV R5,R0	MOV R5, @RW0+d8	MOV R6,R0	MOV R6, @RW0+d8	MOV R7,R0	MOV R7, @RW0+d8
+ 1	MOV R0,R1	MOV R0, @RW1+d8	MOV R1,R1	MOV R1, @RW1+d8	MOV R2,R1	MOV R2, @RW1+d8	MOV R3,R1	MOV R3, @RW1+d8	MOV R4,R1	MOV R4, @RW1+d8	MOV R5,R1	MOV R5, @RW1+d8	MOV R6,R1	MOV R6, @RW1+d8	MOV R7,R1	MOV R7, @RW1+d8
+ 2	MOV R0,R2	MOV R0, @RW2+d8	MOV R1,R2	MOV R1, @RW2+d8	MOV R2,R2	MOV R2, @RW2+d8	MOV R3,R2	MOV R3, @RW2+d8	MOV R4,R2	MOV R4, @RW2+d8	MOV R5,R2	MOV R5, @RW2+d8	MOV R6,R2	MOV R6, @RW2+d8	MOV R7,R2	MOV R7, @RW2+d8
+ 3	MOV R0,R3	MOV R0, @RW3+d8	MOV R1,R3	MOV R1, @RW3+d8	MOV R2,R3	MOV R2, @RW3+d8	MOV R3,R3	MOV R3, @RW3+d8	MOV R4,R3	MOV R4, @RW3+d8	MOV R5,R3	MOV R5, @RW3+d8	MOV R6,R3	MOV R6, @RW3+d8	MOV R7,R3	MOV R7, @RW3+d8
+ 4	MOV R0,R4	MOV R0, @RW4+d8	MOV R1,R4	MOV R1, @RW4+d8	MOV R2,R4	MOV R2, @RW4+d8	MOV R3,R4	MOV R3, @RW4+d8	MOV R4,R4	MOV R4, @RW4+d8	MOV R5,R4	MOV R5, @RW4+d8	MOV R6,R4	MOV R6, @RW4+d8	MOV R7,R4	MOV R7, @RW4+d8
+ 5	MOV R0,R5	MOV R0, @RW5+d8	MOV R1,R5	MOV R1, @RW5+d8	MOV R2,R5	MOV R2, @RW5+d8	MOV R3,R5	MOV R3, @RW5+d8	MOV R4,R5	MOV R4, @RW5+d8	MOV R5,R5	MOV R5, @RW5+d8	MOV R6,R5	MOV R6, @RW5+d8	MOV R7,R5	MOV R7, @RW5+d8
+ 6	MOV R0,R6	MOV R0, @RW6+d8	MOV R1,R6	MOV R1, @RW6+d8	MOV R2,R6	MOV R2, @RW6+d8	MOV R3,R6	MOV R3, @RW6+d8	MOV R4,R6	MOV R4, @RW6+d8	MOV R5,R6	MOV R5, @RW6+d8	MOV R6,R6	MOV R6, @RW6+d8	MOV R7,R6	MOV R7, @RW6+d8
+ 7	MOV R0,R7	MOV R0, @RW7+d8	MOV R1,R7	MOV R1, @RW7+d8	MOV R2,R7	MOV R2, @RW7+d8	MOV R3,R7	MOV R3, @RW7+d8	MOV R4,R7	MOV R4, @RW7+d8	MOV R5,R7	MOV R5, @RW7+d8	MOV R6,R7	MOV R6, @RW7+d8	MOV R7,R7	MOV R7, @RW7+d8
+ 8	MOV R0,@RW0	MOV R0, @RW0+d16	MOV R1,@RW0	MOV R1, @RW0+d16	MOV R2,@RW0	MOV R2, @RW0+d16	MOV R3,@RW0	MOV R3, @RW0+d16	MOV R4,@RW0	MOV R4, @RW0+d16	MOV R5,@RW0	MOV R5, @RW0+d16	MOV R6,@RW0	MOV R6, @RW0+d16	MOV R7,@RW0	MOV R7, @RW0+d16
+ 9	MOV R0,@RW1	MOV R0, @RW1+d16	MOV R1,@RW1	MOV R1, @RW1+d16	MOV R2,@RW1	MOV R2, @RW1+d16	MOV R3,@RW1	MOV R3, @RW1+d16	MOV R4,@RW1	MOV R4, @RW1+d16	MOV R5,@RW1	MOV R5, @RW1+d16	MOV R6,@RW1	MOV R6, @RW1+d16	MOV R7,@RW1	MOV R7, @RW1+d16
+ A	MOV R0,@RW2	MOV R0, @RW2+d16	MOV R1,@RW2	MOV R1, @RW2+d16	MOV R2,@RW2	MOV R2, @RW2+d16	MOV R3,@RW2	MOV R3, @RW2+d16	MOV R4,@RW2	MOV R4, @RW2+d16	MOV R5,@RW2	MOV R5, @RW2+d16	MOV R6,@RW2	MOV R6, @RW2+d16	MOV R7,@RW2	MOV R7, @RW2+d16
+ B	MOV R0,@RW3	MOV R0, @RW3+d16	MOV R1,@RW3	MOV R1, @RW3+d16	MOV R2,@RW3	MOV R2, @RW3+d16	MOV R3,@RW3	MOV R3, @RW3+d16	MOV R4,@RW3	MOV R4, @RW3+d16	MOV R5,@RW3	MOV R5, @RW3+d16	MOV R6,@RW3	MOV R6, @RW3+d16	MOV R7,@RW3	MOV R7, @RW3+d16
+ C	MOV R0, @RW0+	MOV R0, @RW0+RW7	MOV R1, @RW0+	MOV R1, @RW0+RW7	MOV R2, @RW0+	MOV R2, @RW0+RW7	MOV R3, @RW0+	MOV R3, @RW0+RW7	MOV R4, @RW0+	MOV R4, @RW0+RW7	MOV R5, @RW0+	MOV R5, @RW0+RW7	MOV R6, @RW0+	MOV R6, @RW0+RW7	MOV R7, @RW0+	MOV R7, @RW0+RW7
+ D	MOV R0, @RW1+	MOV R0, @RW1+RW7	MOV R1, @RW1+	MOV R1, @RW1+RW7	MOV R2, @RW1+	MOV R2, @RW1+RW7	MOV R3, @RW1+	MOV R3, @RW1+RW7	MOV R4, @RW1+	MOV R4, @RW1+RW7	MOV R5, @RW1+	MOV R5, @RW1+RW7	MOV R6, @RW1+	MOV R6, @RW1+RW7	MOV R7, @RW1+	MOV R7, @RW1+RW7
+ E	MOV R0, @RW2+	MOV R0, @PC+d16	MOV R1, @RW2+	MOV R1, @PC+d16	MOV R2, @RW2+	MOV R2, @PC+d16	MOV R3, @RW2+	MOV R3, @PC+d16	MOV R4, @RW2+	MOV R4, @PC+d16	MOV R5, @RW2+	MOV R5, @PC+d16	MOV R6, @RW2+	MOV R6, @PC+d16	MOV R7, @RW2+	MOV R7, @PC+d16
+ F	MOV R0, @RW3+	MOV R0, addr16	MOV R1, @RW3+	MOV R1, addr16	MOV R2, @RW3+	MOV R2, addr16	MOV R3, @RW3+	MOV R3, addr16	MOV R4, @RW3+	MOV R4, addr16	MOV R5, @RW3+	MOV R5, addr16	MOV R6, @RW3+	MOV R6, addr16	MOV R7, @RW3+	MOV R7, addr16

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-17 MOVW RWi, ea Instruction (first byte = $7B_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	MOVW RW0,RW0	MOVW RW0 ,@RW0+d8	MOVW RW1,RW0	MOVW RW1 ,@RW0+d8	MOVW RW2,RW0	MOVW RW2 ,@RW0+d8	MOVW RW3,RW0	MOVW RW3 ,@RW0+d8	MOVW RW4,RW0	MOVW RW4 ,@RW0+d8	MOVW RW5,RW0	MOVW RW5 ,@RW0+d8	MOVW RW6,RW0	MOVW RW6 ,@RW0+d8	MOVW RW7,RW0	MOVW RW7 ,@RW0+d8
+ 1	MOVW RW0,RW1	MOVW RW0 ,@RW1+d8	MOVW RW1,RW1	MOVW RW1 ,@RW1+d8	MOVW RW2,RW1	MOVW RW2 ,@RW1+d8	MOVW RW3,RW1	MOVW RW3 ,@RW1+d8	MOVW RW4,RW1	MOVW RW4 ,@RW1+d8	MOVW RW5,RW1	MOVW RW5 ,@RW1+d8	MOVW RW6,RW1	MOVW RW6 ,@RW1+d8	MOVW RW7,RW1	MOVW RW7 ,@RW1+d8
+ 2	MOVW RW0,RW2	MOVW RW0 ,@RW2+d8	MOVW RW1,RW2	MOVW RW1 ,@RW2+d8	MOVW RW2,RW2	MOVW RW2 ,@RW2+d8	MOVW RW3,RW2	MOVW RW3 ,@RW2+d8	MOVW RW4,RW2	MOVW RW4 ,@RW2+d8	MOVW RW5,RW2	MOVW RW5 ,@RW2+d8	MOVW RW6,RW2	MOVW RW6 ,@RW2+d8	MOVW RW7,RW2	MOVW RW7 ,@RW2+d8
+ 3	MOVW RW0,RW3	MOVW RW0 ,@RW3+d8	MOVW RW1,RW3	MOVW RW1 ,@RW3+d8	MOVW RW2,RW3	MOVW RW2 ,@RW3+d8	MOVW RW3,RW3	MOVW RW3 ,@RW3+d8	MOVW RW4,RW3	MOVW RW4 ,@RW3+d8	MOVW RW5,RW3	MOVW RW5 ,@RW3+d8	MOVW RW6,RW3	MOVW RW6 ,@RW3+d8	MOVW RW7,RW3	MOVW RW7 ,@RW3+d8
+ 4	MOVW RW0,RW4	MOVW RW0 ,@RW4+d8	MOVW RW1,RW4	MOVW RW1 ,@RW4+d8	MOVW RW2,RW4	MOVW RW2 ,@RW4+d8	MOVW RW3,RW4	MOVW RW3 ,@RW4+d8	MOVW RW4,RW4	MOVW RW4 ,@RW4+d8	MOVW RW5,RW4	MOVW RW5 ,@RW4+d8	MOVW RW6,RW4	MOVW RW6 ,@ RW4+d8	MOVW RW7,RW4	MOVW RW7 ,@RW4+d8
+ 5	MOVW RW0,RW5	MOVW RW0 ,@RW5+d8	MOVW RW1,RW5	MOVW RW1 ,@RW5+d8	MOVW RW2,RW5	MOVW RW2 ,@RW5+d8	MOVW RW3,RW5	MOVW RW3 ,@RW5+d8	MOVW RW4,RW5	MOVW RW4 ,@RW5+d8	MOVW RW5,RW5	MOVW RW5 ,@RW5+d8	MOVW RW6,RW5	MOVW RW6 ,@RW5+d8	MOVW RW7,RW5	MOVW RW7 ,@RW5+d8
+ 6	MOVW RW0,RW6	MOVW RW0 ,@RW6+d8	MOVW RW1,RW6	MOVW RW1 ,@RW6+d8	MOVW RW2,RW6	MOVW RW2 ,@RW6+d8	MOVW RW3,RW6	MOVW RW3 ,@RW6+d8	MOVW RW4,RW6	MOVW RW4 ,@RW6+d8	MOVW RW5,RW6	MOVW RW5 ,@RW6+d8	MOVW RW6,RW6	MOVW RW6 ,@RW6+d8	MOVW RW7,RW6	MOVW RW7 ,@RW6+d8
+ 7	MOVW RW0,RW7	MOVW RW0 ,@RW7+d8	MOVW RW1,RW7	MOVW RW1 ,@RW7+d8	MOVW RW2,RW7	MOVW RW2 ,@RW7+d8	MOVW RW3,RW7	MOVW RW3 ,@RW7+d8	MOVW RW4,RW7	MOVW RW4 ,@RW7+d8	MOVW RW5,RW7	MOVW RW5 ,@RW7+d8	MOVW RW6,RW7	MOVW RW6 ,@RW7+d8	MOVW RW7,RW7	MOVW RW7 ,@RW7+d8
+ 8	MOVW RW0,@RW0	MOVW RW0, @RW0+d16	MOVW RW1,@RW0	MOVW RW1, @RW0+d16	MOVW RW2,@RW0	MOVW RW2, @RW0+d16	MOVW RW3,@RW0	MOVW RW3, @RW0+d16	MOVW RW4,@RW0	MOVW RW4, @RW0+d16	MOVW RW5,@RW0	MOVW RW5, @RW0+d16	MOVW RW6,@RW0	MOVW RW6, @RW0+d16	MOVW RW7,@RW0	MOVW RW7, @RW0+d16
+ 9	MOVW RW0,@RW1	MOVW RW0, @RW1+d16	MOVW RW1,@RW1	MOVW RW1, @RW1+d16	MOVW RW2,@RW1	MOVW RW2, @RW1+d16	MOVW RW3,@RW1	MOVW RW3, @RW1+d16	MOVW RW4,@RW1	MOVW RW4, @RW1+d16	MOVW RW5,@RW1	MOVW RW5, @RW1+d16	MOVW RW6,@RW1	MOVW RW6, @RW1+d16	MOVW RW7,@RW1	MOVW RW7, @RW1+d16
+ A	MOVW RW0,@RW2	MOVW RW0, @RW2+d16	MOVW RW1,@RW2	MOVW RW1, @RW2+d16	MOVW RW2,@RW2	MOVW RW2, @RW2+d16	MOVW RW3,@RW2	MOVW RW3, @RW2+d16	MOVW RW4,@RW2	MOVW RW4, @RW2+d16	MOVW RW5,@RW2	MOVW RW5, @RW2+d16	MOVW RW6,@RW2	MOVW RW6, @RW2+d16	MOVW RW7,@RW2	MOVW RW7, @RW2+d16
+ B	MOVW RW0,@RW3	MOVW RW0, @RW3+d16	MOVW RW1,@RW3	MOVW RW1, @RW3+d16	MOVW RW2,@RW3	MOVW RW2, @RW3+d16	MOVW RW3,@RW3	MOVW RW3, @RW3+d16	MOVW RW4,@RW3	MOVW RW4, @RW3+d16	MOVW RW5,@RW3	MOVW RW5, @RW3+d16	MOVW RW6,@RW3	MOVW RW6, @RW3+d16	MOVW RW7,@RW3	MOVW RW7, @RW3+d16
+ C	W0,@RW0+	MOVW RW0, @RW0+RW7	W1,@RW0+	MOVW RW1, @RW0+RW7	W2,@RW0+	MOVW RW2, @RW0+RW7	W3,@RW0+	MOVW RW3, @RW0+RW7	W4,@RW0+	MOVW RW4, @RW0+RW7	W5,@RW0+	MOVW RW5, @RW0+RW7	W6,@RW0+	MOVW RW6, @RW0+RW7	W7,@RW0+	MOVW RW7, @RW0+RW7
+ D	W0,@RW1+	MOVW RW0, @RW1+RW7	W1,@RW1+	MOVW RW1, @RW1+RW7	W2,@RW1+	MOVW RW2, @RW1+RW7	W3,@RW1+	MOVW RW3, @RW1+RW7	W4,@RW1+	MOVW RW4, @RW1+RW7	W5,@RW1+	MOVW RW5, @RW1+RW7	W6,@RW1+	MOVW RW6, @RW1+RW7	W7,@RW1+	MOVW RW7, @RW1+RW7
+ E	W0,@RW2+	MOVW RW0 ,@PC+d16	W1,@RW2+	MOVW RW1 ,@PC+d16	W2,@RW2+	MOVW RW2 ,@PC+d16	W3,@RW2+	MOVW RW3 ,@PC+d16	W4,@RW2+	MOVW RW4 ,@PC+d16	W5,@RW2+	MOVW RW5 ,@PC+d16	W6,@RW2+	MOVW RW6 ,@PC+d16	W7,@RW2+	MOVW RW7 ,@PC+d16
+ F	W0,@RW3+	MOVW RW 0,addr16	W1,@RW3+	MOVW RW 1,addr16	W2,@RW3+	MOVW RW 2,addr16	W3,@RW3+	MOVW RW 3,addr16	W4,@RW3+	MOVW R 4,addr16	W5,@RW3+	MOVW R 5,addr16	W6,@RW3+	MOVW R 6,addr16	W7,@RW3+	MOVW RW 7,addr16

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-18 MOV ea, Ri Instruction (first byte = $7C_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	MOV R0,R0	MOV @R W0+d8,R0	MOV R0,R1	MOV @R W0+d8,R1	MOV R0,R2	MOV @R W0+d8,R2	MOV R0,R3	MOV @R W0+d8,R3	MOV R0,R4	MOV @R W0+d8,R4	MOV R0,R5	MOV @R W0+d8,R5	MOV R0,R6	MOV @R W0+d8,R6	MOV R0,R7	MOV @R W0+d8,R7
+ 1	MOV R1,R0	MOV @R W1+d8,R0	MOV R1,R1	MOV @R W1+d8,R1	MOV R1,R2	MOV @R W1+d8,R2	MOV R1,R3	MOV @R W1+d8,R3	MOV R1,R4	MOV @R W1+d8,R4	MOV R1,R5	MOV @R W1+d8,R5	MOV R1,R6	MOV @R W1+d8,R6	MOV R1,R7	MOV @R W1+d8,R7
+ 2	MOV R2,R0	MOV @R W2+d8,R0	MOV R2,R1	MOV @R W2+d8,R1	MOV R2,R2	MOV @R W2+d8,R2	MOV R2,R3	MOV @R W2+d8,R3	MOV R2,R4	MOV @R W2+d8,R4	MOV R2,R5	MOV @R W2+d8,R5	MOV R2,R6	MOV @R W2+d8,R6	MOV R2,R7	MOV @R W2+d8,R7
+ 3	MOV R3,R0	MOV @R W3+d8,R0	MOV R3,R1	MOV @R W3+d8,R1	MOV R3,R2	MOV @R W3+d8,R2	MOV R3,R3	MOV @R W3+d8,R3	MOV R3,R4	MOV @R W3+d8,R4	MOV R3,R5	MOV @R W3+d8,R5	MOV R3,R6	MOV @R W3+d8,R6	MOV R3,R7	MOV @R W3+d8,R7
+ 4	MOV R4,R0	MOV @R W4+d8,R0	MOV R4,R1	MOV @R W4+d8,R1	MOV R4,R2	MOV @R W4+d8,R2	MOV R4,R3	MOV @R W4+d8,R3	MOV R4,R4	MOV @R W4+d8,R4	MOV R4,R5	MOV @R W4+d8,R5	MOV R4,R6	MOV @R W4+d8,R6	MOV R4,R7	MOV @R W4+d8,R7
+ 5	MOV R5,R0	MOV @R W5+d8,R0	MOV R5,R1	MOV @R W5+d8,R1	MOV R5,R2	MOV @R W5+d8,R2	MOV R5,R3	MOV @R W5+d8,R3	MOV R5,R4	MOV @R W5+d8,R4	MOV R5,R5	MOV @R W5+d8,R5	MOV R5,R6	MOV @R W5+d8,R6	MOV R5,R7	MOV @R W5+d8,R7
+ 6	MOV R6,R0	MOV @R W6+d8,R0	MOV R6,R1	MOV @R W6+d8,R1	MOV R6,R2	MOV @R W6+d8,R2	MOV R6,R3	MOV @R W6+d8,R3	MOV R6,R4	MOV @R W6+d8,R4	MOV R6,R5	MOV @R W6+d8,R5	MOV R6,R6	MOV @R W6+d8,R6	MOV R6,R7	MOV @R W6+d8,R7
+ 7	MOV R7,R0	MOV @R W7+d8,R0	MOV R7,R1	MOV @R W7+d8,R1	MOV R7,R2	MOV @R W7+d8,R2	MOV R7,R3	MOV @R W7+d8,R3	MOV R7,R4	MOV @R W7+d8,R4	MOV R7,R5	MOV @R W7+d8,R5	MOV R7,R6	MOV @R W7+d8,R6	MOV R7,R7	MOV @R W7+d8,R7
+ 8	MOV @RW0,R0	MOV @RW 0+d16,R0	MOV @RW0,R1	MOV @RW 0+d16,R1	MOV @RW0,R2	MOV @RW 0+d16,R2	MOV @RW0,R3	MOV @RW 0+d16,R3	MOV @RW0,R4	MOV @RW 0+d16,R4	MOV @RW0,R5	MOV @RW 0+d16,R5	MOV @RW0,R6	MOV @RW 0+d16,R6	MOV @RW0,R7	MOV @RW 0+d16,R7
+ 9	MOV @RW1,R0	MOV @RW 1+d16,R0	MOV @RW1,R1	MOV @RW 1+d16,R1	MOV @RW1,R2	MOV @RW 1+d16,R2	MOV @RW1,R3	MOV @RW 1+d16,R3	MOV @RW1,R4	MOV @RW 1+d16,R4	MOV @RW1,R5	MOV @RW 1+d16,R5	MOV @RW1,R6	MOV @RW 1+d16,R6	MOV @RW1,R7	MOV @RW 1+d16,R7
+ A	MOV @RW2,R0	MOV @RW 2+d16,R0	MOV @RW2,R1	MOV @RW 2+d16,R1	MOV @RW2,R2	MOV @RW 2+d16,R2	MOV @RW2,R3	MOV @RW 2+d16,R3	MOV @RW2,R4	MOV @RW 2+d16,R4	MOV @RW2,R5	MOV @RW 2+d16,R5	MOV @RW2,R6	MOV @RW 2+d16,R6	MOV @RW2,R7	MOV @RW 2+d16,R7
+ B	MOV @RW3,R0	MOV @RW 3+d16,R0	MOV @RW3,R1	MOV @RW 3+d16,R1	MOV @RW3,R2	MOV @RW 3+d16,R2	MOV @RW3,R3	MOV @RW 3+d16,R3	MOV @RW3,R4	MOV @RW 3+d16,R4	MOV @RW3,R5	MOV @RW 3+d16,R5	MOV @RW3,R6	MOV @RW 3+d16,R6	MOV @RW3,R7	MOV @RW 3+d16,R7
+ C	MOV @RW0+,R0	MOV @RW 0+RW7,R0	MOV @RW0+,R1	MOV @RW 0+RW7,R1	MOV @RW0+,R2	MOV @RW 0+RW7,R2	MOV @RW0+,R3	MOV @RW 0+RW7,R3	MOV @RW0+,R4	MOV @RW 0+RW7,R4	MOV @RW0+,R5	MOV @RW 0+RW7,R5	MOV @RW0+,R6	MOV @RW 0+RW7,R6	MOV @RW0+,R7	MOV @RW 0+RW7,R7
+ D	MOV @RW1+,R0	MOV @RW 1+RW7,R0	MOV @RW1+,R1	MOV @RW 1+RW7,R1	MOV @RW1+,R2	MOV @RW 1+RW7,R2	MOV @RW1+,R3	MOV @RW 1+RW7,R3	MOV @RW1+,R4	MOV @RW 1+RW7,R4	MOV @RW1+,R5	MOV @RW 1+RW7,R5	MOV @RW1+,R6	MOV @RW 1+RW7,R6	MOV @RW1+,R7	MOV @RW 1+RW7,R7
+ E	MOV @RW2+,R0	MOV P C+d16,R0	MOV @RW2+,R1	MOV P C+d16,R1	MOV @RW2+,R2	MOV P C+d16,R2	MOV @RW2+,R3	MOV P C+d16,R3	MOV @RW2+,R4	MOV P C+d16,R4	MOV @RW2+,R5	MOV P C+d16,R5	MOV @RW2+,R6	MOV P C+d16,R6	MOV @RW2+,R7	MOV P C+d16,R7
+ F	MOV @RW3+,R0	MOV a ddr16,R0	MOV @RW3+,R1	MOV a ddr16,R1	MOV @RW3+,R2	MOV a ddr16,R2	MOV @RW3+,R3	MOV a ddr16,R3	MOV @RW3+,R4	MOV a ddr16,R4	MOV @RW3+,R5	MOV a ddr16,R5	MOV @RW3+,R6	MOV a ddr16,R6	MOV @RW3+,R7	MOV a ddr16,R7

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-19 MOVW ea, Rwi Instruction (first byte = $7D_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	MOVW RW0,RW0	MOVW @RW 0+d8,RW0	MOVW RW0,RW1	MOVW @RW 0+d8,RW1	MOVW RW0,RW2	MOVW @RW 0+d8,RW2	MOVW RW0,RW3	MOVW @RW 0+d8,RW3	MOVW RW0,RW4	MOVW @RW 0+d8,RW4	MOVW RW0,RW5	MOVW @RW 0+d8,RW5	MOVW RW0,RW6	MOVW @RW 0+d8,RW6	MOVW RW0,RW7	MOVW @RW 0+d8,RW7
+ 1	MOVW RW1,RW0	MOVW @RW 1+d8,RW0	MOVW RW1,RW1	MOVW @RW 1+d8,RW1	MOVW RW1,RW2	MOVW @RW 1+d8,RW2	MOVW RW1,RW3	MOVW @RW 1+d8,RW3	MOVW RW1,RW4	MOVW @RW 1+d8,RW4	MOVW RW1,RW5	MOVW @RW 1+d8,RW5	MOVW RW1,RW6	MOVW @RW 1+d8,RW6	MOVW RW1,RW7	MOVW @RW 1+d8,RW7
+ 2	MOVW RW2,RW0	MOVW @RW 2+d8,RW0	MOVW RW2,RW1	MOVW @RW 2+d8,RW1	MOVW RW2,RW2	MOVW @RW 2+d8,RW2	MOVW RW2,RW3	MOVW @RW 2+d8,RW3	MOVW RW2,RW4	MOVW @RW 2+d8,RW4	MOVW RW2,RW5	MOVW @RW 2+d8,RW5	MOVW RW2,RW6	MOVW @RW 2+d8,RW6	MOVW RW2,RW7	MOVW @RW 2+d8,RW7
+ 3	MOVW RW3,RW0	MOVW @RW 3+d8,RW0	MOVW RW3,RW1	MOVW @RW 3+d8,RW1	MOVW RW3,RW2	MOVW @RW 3+d8,RW2	MOVW RW3,RW3	MOVW @RW 3+d8,RW3	MOVW RW3,RW4	MOVW @RW 3+d8,RW4	MOVW RW3,RW5	MOVW @RW 3+d8,RW5	MOVW RW3,RW6	MOVW @RW 3+d8,RW6	MOVW RW3,RW7	MOVW @RW 3+d8,RW7
+ 4	MOVW RW4,RW0	MOVW @RW 4+d8,RW0	MOVW RW4,RW1	MOVW @RW 4+d8,RW1	MOVW RW4,RW2	MOVW @RW 4+d8,RW2	MOVW RW4,RW3	MOVW @RW 4+d8,RW3	MOVW RW4,RW4	MOVW @RW 4+d8,RW4	MOVW RW4,RW5	MOVW @RW 4+d8,RW5	MOVW RW4,RW6	MOVW @RW 4+d8,RW6	MOVW RW4,RW7	MOVW @RW 4+d8,RW7
+ 5	MOVW RW5,RW0	MOVW @RW 5+d8,RW0	MOVW RW5,RW1	MOVW @RW 5+d8,RW1	MOVW RW5,RW2	MOVW @RW 5+d8,RW2	MOVW RW5,RW3	MOVW @RW 5+d8,RW3	MOVW RW5,RW4	MOVW @RW 5+d8,RW4	MOVW RW5,RW5	MOVW @RW 5+d8,RW5	MOVW RW5,RW6	MOVW @RW 5+d8,RW6	MOVW RW5,RW7	MOVW @RW 5+d8,RW7
+ 6	MOVW RW6,RW0	MOVW @RW 6+d8,RW0	MOVW RW6,RW1	MOVW @RW 6+d8,RW1	MOVW RW6,RW2	MOVW @RW 6+d8,RW2	MOVW RW6,RW3	MOVW @RW 6+d8,RW3	MOVW RW6,RW4	MOVW @RW 6+d8,RW4	MOVW RW6,RW5	MOVW @RW 6+d8,RW5	MOVW RW6,RW6	MOVW @RW 6+d8,RW6	MOVW RW6,RW7	MOVW @RW 6+d8,RW7
+ 7	MOVW RW7,RW0	MOVW @RW 7+d8,RW0	MOVW RW7,RW1	MOVW @RW 7+d8,RW1	MOVW RW7,RW2	MOVW @RW 7+d8,RW2	MOVW RW7,RW3	MOVW @RW 7+d8,RW3	WOVW RW7,RW4	MOVW @RW 7+d8,RW4	WOVW RW7,RW5	MOVW @RW 7+d8,RW5	WOVW RW7,RW6	MOVW @RW 7+d8,RW6	WOVW RW7,RW7	MOVW @RW 7+d8,RW7
+ 8	MOVW @RW0,RW0	MOVW @RW0 +d16,RW0	MOVW @RW0,RW1	MOVW @RW0 +d16,RW1	MOVW @RW0,RW2	MOVW @RW0 +d16,RW2	MOVW @RW0,RW3	MOVW @RW0 +d16,RW3	MOVW @RW0,RW4	MOVW @RW0 +d16,RW4	MOVW @RW0,RW5	MOVW @RW0 +d16,RW5	MOVW @RW0,RW6	MOVW @RW0 +d16,RW6	MOVW @RW0,RW7	MOVW @RW0 +d16,RW7
+ 9	MOVW @RW1,RW0	MOVW @RW1 +d16,RW0	MOVW @RW1,RW1	MOVW @RW1 +d16,RW1	MOVW @RW1,RW2	MOVW @RW1 +d16,RW2	MOVW @RW1,RW3	MOVW @RW1 +d16,RW3	MOVW @RW1,RW4	MOVW @RW1 +d16,RW4	MOVW @RW1,RW5	MOVW @RW1 +d16,RW5	MOVW @RW1,RW6	MOVW @RW1 +d16,RW6	MOVW @RW1,RW7	MOVW @RW1 +d16,RW7
+ A	MOVW @RW2,RW0	MOVW @RW2 +d16,RW0	MOVW @RW2,RW1	MOVW @RW2 +d16,RW1	MOVW @RW2,RW2	MOVW @RW2 +d16,RW2	MOVW @RW2,RW3	MOVW @RW2 +d16,RW3	MOVW @RW2,RW4	MOVW @RW2 +d16,RW4	MOVW @RW2,RW5	MOVW @RW2 +d16,RW5	MOVW @RW2,RW6	MOVW @RW2 +d16,RW6	MOVW @RW2,RW7	MOVW @RW2 +d16,RW7
+ B	MOVW @RW3,RW0	MOVW @RW3 +d16,RW0	MOVW @RW3,RW1	MOVW @RW3 +d16,RW1	MOVW @RW3,RW2	MOVW @RW3 +d16,RW2	MOVW @RW3,RW3	MOVW @RW3 +d16,RW3	MOVW @RW3,RW4	MOVW @RW3 +d16,RW4	MOVW @RW3,RW5	MOVW @RW3 +d16,RW5	MOVW @RW3,RW6	MOVW @RW3 +d16,RW6	MOVW @RW3,RW7	MOVW @RW3 +d16,RW7
+ C	MOVW @ RW0+,RW0	MOVW @ +RW7,RW0	MOVW @ RW0+,RW1	MOVW @ +RW7,RW1	MOVW @ RW0+,RW2	MOVW @ +RW7,RW2	MOVW @ RW0+,RW3	MOVW @ +RW7,RW3	MOVW @ RW0+,RW4	MOVW @ +RW7,RW4	MOVW @ RW0+,RW5	MOVW @ +RW7,RW5	MOVW @ RW0+,RW6	MOVW @ +RW7,RW6	MOVW @ RW0+,RW7	MOVW @ +RW7,RW7
+ D	MOVW @ RW1+,RW0	MOVW @ +RW7,RW0	MOVW @ RW1+,RW1	MOVW @ +RW7,RW1	MOVW @ RW1+,RW2	MOVW @ +RW7,RW2	MOVW @ RW1+,RW3	MOVW @ +RW7,RW3	MOVW @ RW1+,RW4	MOVW @ +RW7,RW4	MOVW @ RW1+,RW5	MOVW @ +RW7,RW5	MOVW @ RW1+,RW6	MOVW @ +RW7,RW6	MOVW @ RW1+,RW7	MOVW @ +RW7,RW7
+ E	MOVW @ RW2+,RW0	MOVW @PC+ d16,RW0	MOVW @ RW2+,RW1	MOVW @PC+ d16,RW1	MOVW @ RW2+,RW2	MOVW @PC+ d16,RW2	MOVW @ RW2+,RW3	MOVW @PC+ d16,RW3	MOVW @ RW2+,RW4	MOVW @PC+ d16,RW4	MOVW @ RW2+,RW5	MOVW @PC+ d16,RW5	MOVW @ RW2+,RW6	MOVW @PC+ d16,RW6	MOVW @ RW2+,RW7	MOVW @PC+ d16,RW7
+ F	MOVW @ RW3+,RW0	MOVW addr 16,RW0	MOVW @ RW3+,RW1	MOVW addr 16,RW1	MOVW @ RW3+,RW2	MOVW addr 16,RW2	MOVW @ RW3+,RW3	MOVW addr 16,RW3	MOVW @ RW3+,RW4	MOVW addr 16,RW4	MOVW @ RW3+,RW5	MOVW addr 16,RW5	MOVW @ RW3+,RW6	MOVW addr 16,RW6	MOVW @ RW3+,RW7	MOVW addr 16,RW7

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-20 XCH Ri, ea Instruction (first byte = $7E_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	XCH R0, R0,R0	XCH R0, @RW0+d8	XCH R1, R1,R0	XCH R1, @RW0+d8	XCH R2, R2,R0	XCH R2, @RW0+d8	XCH R3, R3,R0	XCH R3, @RW0+d8	XCH R4, R4,R0	XCH R4, @RW0+d8	XCH R5, R5,R0	XCH R5, @RW0+d8	XCH R6, R6,R0	XCH R6, @RW0+d8	XCH R7, R7,R0	XCH R7, @RW0+d8
+ 1	XCH R0, R0,R1	XCH R0, @RW1+d8	XCH R1, R1,R1	XCH R1, @RW1+d8	XCH R2, R2,R1	XCH R2, @RW1+d8	XCH R3, R3,R1	XCH R3, @RW1+d8	XCH R4, R4,R1	XCH R4, @RW1+d8	XCH R5, R5,R1	XCH R5, @RW1+d8	XCH R6, R6,R1	XCH R6, @RW1+d8	XCH R7, R7,R1	XCH R7, @RW1+d8
+ 2	XCH R0, R0,R2	XCH R0, @RW2+d8	XCH R1, R1,R2	XCH R1, @RW2+d8	XCH R2, R2,R2	XCH R2, @RW2+d8	XCH R3, R3,R2	XCH R3, @RW2+d8	XCH R4, R4,R2	XCH R4, @RW2+d8	XCH R5, R5,R2	XCH R5, @RW2+d8	XCH R6, R6,R2	XCH R6, @RW2+d8	XCH R7, R7,R2	XCH R7, @RW2+d8
+ 3	XCH R0, R0,R3	XCH R0, @RW3+d8	XCH R1, R1,R3	XCH R1, @RW3+d8	XCH R2, R2,R3	XCH R2, @RW3+d8	XCH R3, R3,R3	XCH R3, @RW3+d8	XCH R4, R4,R3	XCH R4, @RW3+d8	XCH R5, R5,R3	XCH R5, @RW3+d8	XCH R6, R6,R3	XCH R6, @RW3+d8	XCH R7, R7,R3	XCH R7, @RW3+d8
+ 4	XCH R0, R0,R4	XCH R0, @RW4+d8	XCH R1, R1,R4	XCH R1, @RW4+d8	XCH R2, R2,R4	XCH R2, @RW4+d8	XCH R3, R3,R4	XCH R3, @RW4+d8	XCH R4, R4,R4	XCH R4, @RW4+d8	XCH R5, R5,R4	XCH R5, @RW4+d8	XCH R6, R6,R4	XCH R6, @RW4+d8	XCH R7, R7,R4	XCH R7, @RW4+d8
+ 5	XCH R0, R0,R5	XCH R0, @RW5+d8	XCH R1, R1,R5	XCH R1, @RW5+d8	XCH R2, R2,R5	XCH R2, @RW5+d8	XCH R3, R3,R5	XCH R3, @RW5+d8	XCH R4, R4,R5	XCH R4, @RW5+d8	XCH R5, R5,R5	XCH R5, @RW5+d8	XCH R6, R6,R5	XCH R6, @RW5+d8	XCH R7, R7,R5	XCH R7, @RW5+d8
+ 6	XCH R0, R0,R6	XCH R0, @RW6+d8	XCH R1, R1,R6	XCH R1, @RW6+d8	XCH R2, R2,R6	XCH R2, @RW6+d8	XCH R3, R3,R6	XCH R3, @RW6+d8	XCH R4, R4,R6	XCH R4, @RW6+d8	XCH R5, R5,R6	XCH R5, @RW6+d8	XCH R6, R6,R6	XCH R6, @RW6+d8	XCH R7, R7,R6	XCH R7, @RW6+d8
+ 7	XCH R0, R0,R7	XCH R0, @RW7+d8	XCH R1, R1,R7	XCH R1, @RW7+d8	XCH R2, R2,R7	XCH R2, @RW7+d8	XCH R3, R3,R7	XCH R3, @RW7+d8	XCH R4, R4,R7	XCH R4, @RW7+d8	XCH R5, R5,R7	XCH R5, @RW7+d8	XCH R6, R6,R7	XCH R6, @RW7+d8	XCH R7, R7,R7	XCH R7, @RW7+d8
+ 8	XCH R0, @RW0	XCH R0, @RW0+d16	XCH R1, @RW0	XCH R1, @RW0+d16	XCH R2, @RW0	XCH R2, @RW0+d16	XCH R3, @RW0	XCH R3, @RW0+d16	XCH R4, @RW0	XCH R4, @RW0+d16	XCH R5, @RW0	XCH R5, @RW0+d16	XCH R6, @RW0	XCH R6, @RW0+d16	XCH R7, @RW0	XCH R7, @RW0+d16
+ 9	XCH R0, @RW1	XCH R0, @RW1+d16	XCH R1, @RW1	XCH R1, @RW1+d16	XCH R2, @RW1	XCH R2, @RW1+d16	XCH R3, @RW1	XCH R3, @RW1+d16	XCH R4, @RW1	XCH R4, @RW1+d16	XCH R5, @RW1	XCH R5, @RW1+d16	XCH R6, @RW1	XCH R6, @RW1+d16	XCH R7, @RW1	XCH R7, @RW1+d16
+ A	XCH R0, @RW2	XCH R0, W2+d16,A	XCH R1, @RW2	XCH R1, W2+d16,A	XCH R2, @RW2	XCH R2, W2+d16,A	XCH R3, @RW2	XCH R3, W2+d16,A	XCH R4, @RW2	XCH R4, W2+d16,A	XCH R5, @RW2	XCH R5, W2+d16,A	XCH R6, @RW2	XCH R6, W2+d16,A	XCH R7, @RW2	XCH R7, W2+d16,A
+ B	XCH R0, @RW3	XCH R0, @RW3+d16	XCH R1, @RW3	XCH R1, @RW3+d16	XCH R2, @RW3	XCH R2, @RW3+d16	XCH R3, @RW3	XCH R3, @RW3+d16	XCH R4, @RW3	XCH R4, @RW3+d16	XCH R5, @RW3	XCH R5, @RW3+d16	XCH R6, @RW3	XCH R6, @RW3+d16	XCH R7, @RW3	XCH R7, @RW3+d16
+ C	XCH R0, @RW0+	XCH R0, @RW0+RW7	XCH R1, @RW0+	XCH R1, @RW0+RW7	XCH R2, @RW0+	XCH R2, @RW0+RW7	XCH R3, @RW0+	XCH R3, @RW0+RW7	XCH R4, @RW0+	XCH R4, @RW0+RW7	XCH R5, @RW0+	XCH R5, @RW0+RW7	XCH R6, @RW0+	XCH R6, @RW0+RW7	XCH R7, @RW0+	XCH R7, @RW0+RW7
+ D	XCH R0, @RW1+	XCH R0, @RW1+RW7	XCH R1, @RW1+	XCH R1, @RW1+RW7	XCH R2, @RW1+	XCH R2, @RW1+RW7	XCH R3, @RW1+	XCH R3, @RW1+RW7	XCH R4, @RW1+	XCH R4, @RW1+RW7	XCH R5, @RW1+	XCH R5, @RW1+RW7	XCH R6, @RW1+	XCH R6, @RW1+RW7	XCH R7, @RW1+	XCH R7, @RW1+RW7
+ E	XCH R0, @RW2+	XCH R0, @PC+d16	XCH R1, @RW2+	XCH R1, @PC+d16	XCH R2, @RW2+	XCH R2, @PC+d16	XCH R3, @RW2+	XCH R3, @PC+d16	XCH R4, @RW2+	XCH R4, @PC+d16	XCH R5, @RW2+	XCH R5, @PC+d16	XCH R6, @RW2+	XCH R6, @PC+d16	XCH R7, @RW2+	XCH R7, @PC+d16
+ F	XCH R0, @RW3+	XCH R0, addr16	XCH R1, @RW3+	XCH R1, addr16	XCH R2, @RW3+	XCH R2, addr16	XCH R3, @RW3+	XCH R3, addr16	XCH R4, @RW3+	XCH R4, addr16	XCH R5, @RW3+	XCH R5, addr16	XCH R6, @RW3+	XCH R6, addr16	XCH R7, @RW3+	XCH R7, addr16

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table A.9-21 XCHW RWi, ea Instruction (first byte = $7F_H$)

	00	10	20	30	40	50	60	70	80	90	A0	B0	C0	D0	E0	F0
+ 0	XCHW RW0,RW0	XCHW RW0,@RW0+d8	XCHW RW1,RW0	XCHW RW1,@RW0+d8	XCHW RW2,RW0	XCHW RW2,@RW0+d8	XCHW RW3,RW0	XCHW RW3,@RW0+d8	XCHW RW4,RW0	XCHW RW4,@RW0+d8	XCHW RW5,RW0	XCHW RW5,@RW0+d8	XCHW RW6,RW0	XCHW RW6,@RW0+d8	XCHW RW7,RW0	XCHW RW7,@RW0+d8
+ 1	XCHW RW0,RW1	XCHW RW0,@RW1+d8	XCHW RW1,RW1	XCHW RW1,@RW1+d8	XCHW RW2,RW1	XCHW RW2,@RW1+d8	XCHW RW3,RW1	XCHW RW3,@RW1+d8	XCHW RW4,RW1	XCHW RW4,@RW1+d8	XCHW RW5,RW1	XCHW RW5,@RW1+d8	XCHW RW6,RW1	XCHW RW6,@RW1+d8	XCHW RW7,RW1	XCHW RW7,@RW1+d8
+ 2	XCHW RW0,RW2	XCHW RW0,@RW2+d8	XCHW RW1,RW2	XCHW RW1,@RW2+d8	XCHW RW2,RW2	XCHW RW2,@RW2+d8	XCHW RW3,RW2	XCHW RW3,@RW2+d8	XCHW RW4,RW2	XCHW RW4,@RW2+d8	XCHW RW5,RW2	XCHW RW5,@RW2+d8	XCHW RW6,RW2	XCHW RW6,@RW2+d8	XCHW RW7,RW2	XCHW RW7,@RW2+d8
+ 3	XCHW RW0,RW3	XCHW RW0,@RW3+d8	XCHW RW1,RW3	XCHW RW1,@RW3+d8	XCHW RW2,RW3	XCHW RW2,@RW3+d8	XCHW RW3,RW3	XCHW RW3,@RW3+d8	XCHW RW4,RW3	XCHW RW4,@RW3+d8	XCHW RW5,RW3	XCHW RW5,@RW3+d8	XCHW RW6,RW3	XCHW RW6,@RW3+d8	XCHW RW7,RW3	XCHW RW7,@RW3+d8
+ 4	XCHW RW0,RW4	XCHW RW0,@RW4+d8	XCHW RW1,RW4	XCHW RW1,@RW4+d8	XCHW RW2,RW4	XCHW RW2,@RW4+d8	XCHW RW3,RW4	XCHW RW3,@RW4+d8	XCHW RW4,RW4	XCHW RW4,@RW4+d8	XCHW RW5,RW4	XCHW RW5,@RW4+d8	XCHW RW6,RW4	XCHW RW6,@RW4+d8	XCHW RW7,RW4	XCHW RW7,@RW4+d8
+ 5	XCHW RW0,RW5	XCHW RW0,@RW5+d8	XCHW RW1,RW5	XCHW RW1,@RW5+d8	XCHW RW2,RW5	XCHW RW2,@RW5+d8	XCHW RW3,RW5	XCHW RW3,@RW5+d8	XCHW RW4,RW5	XCHW RW4,@RW5+d8	XCHW RW5,RW5	XCHW RW5,@RW5+d8	XCHW RW6,RW5	XCHW RW6,@RW5+d8	XCHW RW7,RW5	XCHW RW7,@RW5+d8
+ 6	XCHW RW0,RW6	XCHW RW0,@RW6+d8	XCHW RW1,RW6	XCHW RW1,@RW6+d8	XCHW RW2,RW6	XCHW RW2,@RW6+d8	XCHW RW3,RW6	XCHW RW3,@RW6+d8	XCHW RW4,RW6	XCHW RW4,@RW6+d8	XCHW RW5,RW6	XCHW RW5,@RW6+d8	XCHW RW6,RW6	XCHW RW6,@RW6+d8	XCHW RW7,RW6	XCHW RW7,@RW6+d8
+ 7	XCHW RW0,RW7	XCHW RW0,@RW7+d8	XCHW RW1,RW7	XCHW RW1,@RW7+d8	XCHW RW2,RW7	XCHW RW2,@RW7+d8	XCHW RW3,RW7	XCHW RW3,@RW7+d8	XCHW RW4,RW7	XCHW RW4,@RW7+d8	XCHW RW5,RW7	XCHW RW5,@RW7+d8	XCHW RW6,RW7	XCHW RW6,@RW7+d8	XCHW RW7,RW7	XCHW RW7,@RW7+d8
+ 8	XCHW RW0,@RW0	XCHW RW0,@RW0+d16	XCHW RW1,@RW0	XCHW RW1,@RW0+d16	XCHW RW2,@RW0	XCHW RW2,@RW0+d16	XCHW RW3,@RW0	XCHW RW3,@RW0+d16	XCHW RW4,@RW0	XCHW RW4,@RW0+d16	XCHW RW5,@RW0	XCHW RW5,@RW0+d16	XCHW RW6,@RW0	XCHW RW6,@RW0+d16	XCHW RW7,@RW0	XCHW RW7,@RW0+d16
+ 9	XCHW RW0,@RW1	XCHW RW0,@RW1+d16	XCHW RW1,@RW1	XCHW RW1,@RW1+d16	XCHW RW2,@RW1	XCHW RW2,@RW1+d16	XCHW RW3,@RW1	XCHW RW3,@RW1+d16	XCHW RW4,@RW1	XCHW RW4,@RW1+d16	XCHW RW5,@RW1	XCHW RW5,@RW1+d16	XCHW RW6,@RW1	XCHW RW6,@RW1+d16	XCHW RW7,@RW1	XCHW RW7,@RW1+d16
+ A	XCHW RW0,@RW2	XCHW RW0,@RW2+d16	XCHW RW1,@RW2	XCHW RW1,@RW2+d16	XCHW RW2,@RW2	XCHW RW2,@RW2+d16	XCHW RW3,@RW2	XCHW RW3,@RW2+d16	XCHW RW4,@RW2	XCHW RW4,@RW2+d16	XCHW RW5,@RW2	XCHW RW5,@RW2+d16	XCHW RW6,@RW2	XCHW RW6,@RW2+d16	XCHW RW7,@RW2	XCHW RW7,@RW2+d16
+ B	XCHW RW0,@RW3	XCHW RW0,@RW3+d16	XCHW RW1,@RW3	XCHW RW1,@RW3+d16	XCHW RW2,@RW3	XCHW RW2,@RW3+d16	XCHW RW3,@RW3	XCHW RW3,@RW3+d16	XCHW RW4,@RW3	XCHW RW4,@RW3+d16	XCHW RW5,@RW3	XCHW RW5,@RW3+d16	XCHW RW6,@RW3	XCHW RW6,@RW3+d16	XCHW RW7,@RW3	XCHW RW7,@RW3+d16
+ C	XCHW R W0,@RW0+	XCHW RW0,@RW0+RW7	XCHW R W1,@RW0+	XCHW RW1,@RW0+RW7	XCHW R W2,@RW0+	XCHW RW2,@RW0+RW7	XCHW R W3,@RW0+	XCHW RW3,@RW0+RW7	XCHW R W4,@RW0+	XCHW RW4,@RW0+RW7	XCHW R W5,@RW0+	XCHW RW5,@RW0+RW7	XCHW R W6,@RW0+	XCHW RW6,@RW0+RW7	XCHW R W7,@RW0+	XCHW RW7,@RW0+RW7
+ D	XCHW R WO,@RW1+	XCHW RW0,@RW1+RW7	XCHW R W1,@RW1+	XCHW RW1,@RW1+RW7	XCHW R W2,@RW1+	XCHW RW2,@RW1+RW7	XCHW R W3,@RW1+	XCHW RW3,@RW1+RW7	XCHW R W4,@RW1+	XCHW RW4,@RW1+RW7	XCHW R W5,@RW1+	XCHW RW5,@RW1+RW7	XCHW R W6,@RW1+	XCHW RW6,@RW1+RW7	XCHW R W7,@RW1+	XCHW RW7,@RW1+RW7
+ E	XCHW R WO,@RW2+	XCHW RW0,@PC+d16	XCHW R W1,@RW2+	XCHW RW1,@PC+d16	XCHW R W2,@RW2+	XCHW RW2,@PC+d16	XCHW R W3,@RW2+	XCHW RW3,@PC+d16	XCHW R W4,@RW2+	XCHW RW4,@PC+d16	XCHW R W5,@RW2+	XCHW RW5,@PC+d16	XCHW R W6,@RW2+	XCHW RW6,@PC+d16	XCHW R W7,@RW2+	XCHW RW7,@PC+d16
+ F	XCHW R WO,@RW3+	XCHW RW0, addr16	XCHW R W1,@RW3+	XCHW RW1, addr16	XCHW R W2,@RW3+	XCHW RW2, addr16	XCHW R W3,@RW3+	XCHW RW3, addr16	XCHW R W4,@RW3+	XCHW RW4, addr16	XCHW R W5,@RW3+	XCHW RW5, addr16	XCHW R W6,@RW3+	XCHW RW6, addr16	XCHW R W7,@RW3+	XCHW RW7, addr16

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

APPENDIX B Register Index

■ Register Index

Table B-1 Register Index (1/9)

Address	Abbreviation of Register Name	Register Name	Reset Value	Resource Name	Page Number
000000_H	(Reserved area) *				
000001_H	PDR1	Port 1 data register	$XXXXXXXX_B$	Port 1	163
000002_H	PDR2	Port 2 data register	$XXXXXXXX_B$	Port 2	168
000003_H	PDR3	Port 3 data register	$XXXXXXXX_B$	Port 3	173
000004_H	PDR4	Port 4 data register	$XXXXXXXX_B$	Port 4	178
000005_H	PDR5	Port5 data register	$XXXXXXXX_B$	Port 5	183
000006_H to 000010_H	(Reserved area) *				
000011_H	DDR1	Port 1 direction register	00000000_B	Port 1	163
000012_H	DDR2	Port 2 direction register	00000000_B	Port 2	168
000013_H	DDR3	Port 3 direction register	$000X0000_B$	Port 3	173
000014_H	DDR4	Port 4 direction register	$XXX00000_B$	Port 4	178
000015_H	DDR5	Port 5 direction register	00000000_B	Port 5	183
000016_H to $00001A_H$	(Reserved area) *				
$00001B_H$	ADER	Analog input enable register	11111111_B	8/10-bit A/D converter	362
$00001C_H$ to 000025_H	(Reserved area) *				
000026_H	SMR1	Serial mode register 1	00000000_B	UART1	387
000027_H	SCR1	Serial control register 1	00000100_B		385
000028_H	SIDR1/SODR1	Serial input data register 1/ serial output data register 1	$XXXXXXXX_B$		391/392
000029_H	SSR1	Serial status register 1	00001000_B		389
$00002A_H$	(Reserved area) *				
$00002B_H$	CDCR1	Communication prescaler control register 1	$0XXX0000_B$	UART1	393

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table B-1 Register Index (2/9)

Address	Abbreviation of Register Name	Register Name	Reset Value	Resource Name	Page Number
$00002C_H$ to $00002F_H$	(Reserved area) *				
000030_H	ENIR	DTP/external interrupt enable register	00000000_B	DTP/external interrupt	333
000031_H	EIRR	DTP/external interrupt factor register	$XXXXXXXX_B$		332
000032_H	ELVR	Detection level setting register	00000000_B		335
000033_H			00000000_B		334
000034_H	ADCS	A/D control status register	00000000_B	8/10-bit A/D converter	356
000035_H			00000000_B		354
000036_H	ADCR	A/D data register	$XXXXXXXX_B$		361
000037_H			$00101XXX_B$		359
000038_H to $00003F_H$	(Reserved area) *				
000040_H	PPGC0	PPG0 operation mode control register	$0X000XX1_B$	8/16-bit PPG PPG timer 0/1	298
000041_H	PPGC1	PPG1 operation mode control register	$0X000001_B$		300
000042_H	PPG01	PPG0/1 count clock select register	$000000XX_B$		302
000043_H	(Reserved area) *				
000044_H	PPGC2	PPG2 operation mode control register	$0X000XX1_B$	8/16-bit PPG timer 2/3	298
000045_H	PPGC3	PPG3 operation mode control register	$0X000001_B$		300
000046_H	PPG23	PPG2/3 count clock select register	$000000XX_B$		302
000047_H to $00004F_H$	(Reserved area) *				

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table B-1 Register Index (3/9)

Address	Abbreviation of Register Name	Register Name	Reset Value	Resource Name	Page Number
000050_H	IPCP0	Input capture data register 0	$XXXXXXXX_B$	16-bit I/O timers	232
000051_H			$XXXXXXXX_B$		
000052_H	IPCP1	Input capture data register 1	$XXXXXXXX_B$		232
000053_H			$XXXXXXXX_B$		
000054_H	ICS01	Input capture control status register	00000000_B		229
000055_H	ICS23		00000000_B		
000056_H	TCDT	Timer counter data register	00000000_B		227
000057_H			00000000_B		
000058_H	TCCS	Timer counter control status register	00000000_B		225
000059_H	(Reserved area) *				
$00005A_H$	IPCP2	Input capture data register 2	$XXXXXXXX_B$	16-bit I/O timers	232
$00005B_H$			$XXXXXXXX_B$		
$00005C_H$	IPCP3	Input capture data register 3	$XXXXXXXX_B$		232
$00005D_H$			$XXXXXXXX_B$		
$00005E_H$ to 000065_H	(Reserved area) *				
000066_H	TMCSR0	Timer control status register	00000000_B	16-bit reload timer 0	253
000067_H			$XXXX0000_B$		251
000068_H	TMCSR1		00000000_B	16-bit reload timer 1	253
000069_H			$XXXX0000_B$		251
$00006A_H$ to $00006E_H$	(Reserved area) *				
$00006F_H$	ROMM	ROM mirroring function select register	$XXXXXXX1_B$	ROM mirroring function select module	526
000070_H to $00007F_H$	(Reserved area) *				
000080_H	BVALR	Message buffer valid register	00000000_B	CAN controller	450
000081_H	(Reserved area) *				
000082_H	TREQR	Transmission request register	00000000_B	CAN controller	454

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table B-1 Register Index (4/9)

Address	Abbreviation of Register Name	Register Name	Reset Value	Resource Name	Page Number
000083_H	(Reserved area) *				
000084_H	TCANR	Transmission cancel register	00000000_B	CAN controller	460
000085_H	(Reserved area) *				
000086_H	TCR	Transmission complete register	00000000_B	CAN controller	462
000087_H	(Reserved area) *				
000088_H	RCR	Reception complete register	00000000_B	CAN controller	466
000089_H	(Reserved area) *				
$00008A_H$	RRTRR	Reception RTR register	00000000_B	CAN controller	468
$00008B_H$	(Reserved area) *				
$00008C_H$	ROVRR	Reception overrun register	00000000_B	CAN controller	470
$00008D_H$	(Reserved area) *				
$00008E_H$	RIER	Reception complete interrupt enable register	00000000_B	CAN controller	472
$00008F_H$ to $00009D_H$	(Reserved area) *				
$00009E_H$	PACSR	Address detection control register	00000000_B	Address match detecting function	511
$00009F_H$	DIRR	Delayed interrupt request generate/cancel register	$XXXXXXX0_B$	Delayed interrupt generation module	323
$0000A0_H$	LPMCR	Low-power consumption mode control register	00011000_B	Low-power consumption mode	130
$0000A1_H$	CKSCR	Clock select register	11111100_B	Clock	115
$0000A2_H$ to $0000A7_H$	(Reserved area) *				
$0000A8_H$	WDTC	Watchdog timer control register	$XXXXX111_B$	Watchdog timer	208
$0000A9_H$	TBTC	Timebase timer control register	$1XX00100_B$	Timebase timer	193
$0000AA_H$	WTC	Watch timer control register	$1X001000_B$	Watch timer	279
$0000AB_H$ to $0000AD_H$	(Reserved area) *				
$0000AE_H$	FMCS	Flash memory control status register	$000X0000_B$	512-Kbit flash memory	530
$0000AF_H$	(Reserved area) *				

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table B-1 Register Index (5/9)

Address	Abbreviation of Register Name	Register Name	Reset Value	Resource Name	Page Number
$0000B0_H$	ICR00	Interrupt control register 00	00000111_B	Interrupt controller	66
$0000B1_H$	ICR01	Interrupt control register 01	00000111_B		
$0000B2_H$	ICR02	Interrupt control register 02	00000111_B		
$0000B3_H$	ICR03	Interrupt control register 03	00000111_B		
$0000B4_H$	ICR04	Interrupt control register 04	00000111_B		
$0000B5_H$	ICR05	Interrupt control register 05	00000111_B		
$0000B6_H$	ICR06	Interrupt control register 06	00000111_B		
$0000B7_H$	ICR07	Interrupt control register 07	00000111_B		
$0000B8_H$	ICR08	Interrupt control register 08	00000111_B		
$0000B9_H$	ICR09	Interrupt control register 09	00000111_B		
$0000BA_H$	ICR10	Interrupt control register 10	00000111_B		
$0000BB_H$	ICR11	Interrupt control register 11	00000111_B		
$0000BC_H$	ICR12	Interrupt control register 12	00000111_B		
$0000BD_H$	ICR13	Interrupt control register 13	00000111_B		
$0000BE_H$	ICR14	Interrupt control register 14	00000111_B		
$0000BF_H$	ICR15	Interrupt control register 15	00000111_B		
$0000C0_H$ to $0000FF_H$	(Reserved area) *				
$001FF0_H$	PADR0	Detect address setting register 0 (Low)	$XXXXXXXX_B$	Address match detecting function	513
$001FF1_H$		Detect address setting register 0 (Middle)	$XXXXXXXX_B$		
$001FF2_H$		Detect address setting register 0 (High)	$XXXXXXXX_B$		
$001FF3_H$	PADR1	Detect address setting register 1 (Low)	$XXXXXXXX_B$		513
$001FF4_H$		Detect address setting register 1 (Middle)	$XXXXXXXX_B$		
$001FF5_H$		Detect address setting register 1 (High)	$XXXXXXXX_B$		
003900_H	TMR0/TMRLR0	16-bit timer register 0/ 16-bit reload register 0	$XXXXXXXX_B$	16-bit reload timer 0	255/256
003901_H			$XXXXXXXX_B$		

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table B-1 Register Index (6/9)

Address	Abbreviation of Register Name	Register Name	Reset Value	Resource Name	Page Number
003902_H	TMR1/TMRLR1	16-bit timer register 1/ 16-bit reload register 1	$XXXXXXXX_B$	16-bit reload timer 1	255/256
003903_H			$XXXXXXXX_B$		
003904_H to $00390F_H$	(Reserved area) *				
003910_H	PRLL0	PPG0 reload register L	$XXXXXXXX_B$	8/16-bit PPG timer	304
003911_H	PRLH0	PPG0 reload register H	$XXXXXXXX_B$		304
003912_H	PRLL1	PPG1 reload register L	$XXXXXXXX_B$		304
003913_H	PRLH1	PPG1 reload register H	$XXXXXXXX_B$		304
003914_H	PRLL2	PPG2 reload register L	$XXXXXXXX_B$		304
003915_H	PRLH2	PPG2 reload register H	$XXXXXXXX_B$		304
003916_H	PRLL3	PPG3 reload register L	$XXXXXXXX_B$		304
003917_H	PRLH3	PPG3 reload register H	$XXXXXXXX_B$		304
003918_H to $00392F_H$	(Reserved area) *				
003930_H to $003BFF_H$	(Reserved area) *				
$003C00_H$ to $003C0F_H$	RAM (general-purpose RAM)				
$003C10_H$ to $003C13_H$	IDR0	ID register 0	$XXXXXXXX_B$ to $XXXXXXXX_B$	CAN controller	479
$003C14_H$ to $003C17_H$	IDR1	ID register 1	$XXXXXXXX_B$ to $XXXXXXXX_B$		479
$003C18_H$ to $003C1B_H$	IDR2	ID register 2	$XXXXXXXX_B$ to $XXXXXXXX_B$		479
$003C1C_H$ to $003C1F_H$	IDR3	ID register 3	$XXXXXXXX_B$ to $XXXXXXXX_B$		479
$003C20_H$ to $003C23_H$	IDR4	ID register 4	$XXXXXXXX_B$ to $XXXXXXXX_B$		479

DataSheet4U.com
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table B-1 Register Index (7/9)

Address	Abbreviation of Register Name	Register Name	Reset Value	Resource Name	Page Number
$003C24_H$ to $003C27_H$	IDR5	ID register 5	$XXXXXXXX_B$ to $XXXXXXXX_B$	CAN controller	479
$003C28_H$ to $003C2B_H$	IDR6	ID register 6	$XXXXXXXX_B$ to $XXXXXXXX_B$		479
$003C2C_H$ to $003C2F_H$	IDR7	ID register 7	$XXXXXXXX_B$ to $XXXXXXXX_B$		479
$003C30_H$ $003C31_H$	DLCR0	DLC register 0	$XXXXXXXX_B$ $XXXXXXXX_B$		482
$003C32_H$ $003C33_H$	DLCR1	DLC register 1	$XXXXXXXX_B$ $XXXXXXXX_B$		482
$003C34_H$ $003C35_H$	DLCR2	DLC register 2	$XXXXXXXX_B$ $XXXXXXXX_B$		482
$003C36_H$ $003C37_H$	DLCR3	DLC register 3	$XXXXXXXX_B$ $XXXXXXXX_B$		482
$003C38_H$ $003C39_H$	DLCR4	DLC register 4	$XXXXXXXX_B$ $XXXXXXXX_B$		482
$003C3A_H$ $003C3B_H$	DLCR5	DLC register 5	$XXXXXXXX_B$ $XXXXXXXX_B$		482
$003C3C_H$ $003C3D_H$	DLCR6	DLC register 6	$XXXXXXXX_B$ $XXXXXXXX_B$		482
$003C3E_H$ $003C3F_H$	DLCR7	DLC register 7	$XXXXXXXX_B$ $XXXXXXXX_B$		482
$003C40_H$ to $003C47_H$	DTR0	Data register 0	$XXXXXXXX_B$ to $XXXXXXXX_B$		483
$003C48_H$ to $003C4F_H$	DTR1	Data register 1	$XXXXXXXX_B$ to $XXXXXXXX_B$		483
$003C50_H$ to $003C57_H$	DTR2	Data register 2	$XXXXXXXX_B$ to $XXXXXXXX_B$		483
$003C58_H$ to $003C5F_H$	DTR3	Data register 3	$XXXXXXXX_B$ to $XXXXXXXX_B$		483

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table B-1 Register Index (8/9)

Address	Abbreviation of Register Name	Register Name	Reset Value	Resource Name	Page Number
$003C60_H$ to $003C67_H$	DTR4	Data register 4	$XXXXXXXX_B$ to $XXXXXXXX_B$	CAN controller	483
$003C68_H$ to $003C6F_H$	DTR5	Data register 5	$XXXXXXXX_B$ to $XXXXXXXX_B$		483
$003C70_H$ to $003C77_H$	DTR6	Data register 6	$XXXXXXXX_B$ to $XXXXXXXX_B$		483
$003C78_H$ to $003C7F_H$	DTR7	Data register 7	$XXXXXXXX_B$ to $XXXXXXXX_B$		483
$003C80_H$ to $003CFF_H$	(Reserved area) *				
$003D00_H$ $003D01_H$	CSR	Control status register	$0XXXX001_B$ $00XXX000_B$	CAN controller	437/439
$003D02_H$	LEIR	Last event indicate register	$000XX000_B$		442
$003D03_H$	(Reserved area) *				
$003D04_H$ $003D05_H$	RTEC	Receive/transmit error counter	00000000_B 00000000_B	CAN controller	444
$003D06_H$ $003D07_H$	BTR	Bit timing register	11111111_B $X1111111_B$		446
$003D08_H$	IDER	IDE register	$XXXXXXXX_B$		452
$003D09_H$	(Reserved area) *				
$003D0A_H$	TRTRR	Transmission RTR register	00000000_B	CAN controller	456
$003D0B_H$	(Reserved area) *				
$003D0C_H$	RFWTR	Remote frame receiving wait register	$XXXXXXXX_B$	CAN controller	458
$003D0D_H$	(Reserved area) *				
$003D0E_H$	TIER	Transmission complete interrupt enable register	00000000_B	CAN controller	464
$003D0F_H$	(Reserved area) *				
$003D10_H$ $003D11_H$	AMSR	Acceptance mask select register	$XXXXXXXX_B$ to $XXXXXXXX_B$	CAN controller	474

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table B-1 Register Index (9/9)

Address	Abbreviation of Register Name	Register Name	Reset Value	Resource Name	Page Number
$003D12_H$ $003D13_H$	(Reserved area) *				
$003D14_H$ to $003D17_H$	AMR0	Acceptance mask register 0	$XXXXXXXX_B$ to $XXXXXXXX_B$	CAN controller	476
$003D18_H$ to $003D1B_H$	AMR1	Acceptance mask register 1	$XXXXXXXX_B$ to $XXXXXXXX_B$		476
$003D1C_H$ to $003DFF_H$	(Reserved area) *				
$003E00_H$ to $003EFF_H$	(Reserved area) *				
$003FF0_H$ to $003FFF_H$	(Reserved area) *				

et4U.com DataShee

Explanation of reset value

DataSheet4U.com

0: The reset value of this bit is 0.
1: The reset value of this bit is 1.
X: The reset value of this bit is unfixed.

* : Do not write the data to "(Reserved area)". If the data is read from "(Reserved area)", it is undefined values.

DataSheet4U.com www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

APPENDIX C Pin Function Index

■ Pin Function Index

Table C-1 Pin Function Index (1/2)

Pin Number M05	Pin Name	Circuit Type	Function	Page Number for Function Explanation	Page Number for Block Diagram
1	AV_{CC}	–	V_{CC} input pin for A/D converter	350	349
2	AVR	–	Vref + input pin for A/D converter	350	349
3 to 10	P50 to P57	E	General-purpose I/O ports	181	182
	AN0 to AN7		Analog input pins for A/D converter	350	349
11	P37	D	General-purpose I/O port	171	172
	ADTG		External trigger input pin for A/D converter	352	349
12	P20	D	General-purpose I/O port	166	167
	TIN0		Event input pin for reload timer 0	248	246
13	P21	D	General-purpose I/O port	166	167
	TOT0		Event output pin for reload timer 0	248	246
14	P22	D	General-purpose I/O port	166	167
	TIN1		Event input pin for reload timer 1	248	246
15	P23	D	General-purpose I/O ports	166	167
	TOT1		Event output pin for reload timer 1	248	246
16 to 19	P24 to P27	D	General-purpose I/O ports	166	167
	INT4 to INT7		External interrupt input pins	331	329
20	MD2	F	Operation mode select input pin	150	–
21	MD1	C	Operation mode select input pin	150	–
22	MD0	C	Operation mode select input pin	150	–
23	$\overline{RST}$	B	External reset input pin	103	103
24	V_{CC}		Power (5 V) input pin	–	–
25	V_{SS}	–	Power (0 V) input pin	–	–
26	C	–	Power stabilization capacitance pin	–	–
27	X0	A	High-speed oscillation pin	109	112
28	X1	A	High-speed oscillation pin	109	112
29 to 32	P10 to P13	D	General-purpose I/O ports	161	162
	IN0 to IN3		Trigger input pins for input capture channels 0 to 3	222	220
33 to 36	P14 to P17	D	General-purpose I/O ports	161	162
	PPG0 to PPG3		Output pins for PPG timers 01 and 23	296	290/293

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table C-1 Pin Function Index (2/2)

Pin Number M05	Pin Name	Circuit Type	Function	Page Number for Function Explanation	Page Number for Block Diagram
37	P40	D	General-purpose I/O port	176	177
	SIN1		Serial data input pin for UART1	383	380
38	P41	D	General-purpose I/O port	176	177
	SCK1		Serial clock input/output pin for UART1	383	380
39	P42	D	General-purpose I/O port	176	177
	SOT1		Serial data output pin for UART1	383	380
40	P43	D	General-purpose I/O port	176	177
	TX		Transmit output pin for CAN controller	433	429
41	P44	D	General-purpose I/O port	176	177
	RX		Receive input pin for CAN controller	433	429
42 to 45	P30 to P33	D	General-purpose I/O port	171	172
46	X0A*	A	Low-speed oscillation pin	109	112
	P35*		General-purpose I/O port	171	172
47	X1A*	A	Low-speed oscillation pin	109	112
	P36*		General-purpose I/O port	171	172
48	AV_{SS}	–	V_{SS} input pin for A/D converter	350	349

*:MB90387, MB90F387: X1A, X0A
MB90387S, MB90F387S: P36, P35

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

APPENDIX D Interrupt Vector Index

■ Interrupt Vector Index

Table D-1 Interrupt Vector Index (1/2)

Interrupt Number	Interrupt Factor	Interrupt Control		Address in Vector Table			Page Number
		ICR	Address	Low	Middle	High	
#08	Reset	–	–	$FFFFDC_H$	$FFFFDD_H$	$FFFFDE_H$	99
#09	INT9 instruction	–	–	$FFFFD8_H$	$FFFFD9_H$	$FFFFDA_H$	508
#10	Exception processing	–	–	$FFFFD4_H$	$FFFFD5_H$	$FFFFD6_H$	91
#11	CAN controller receive completion	ICR00	$0000B0_H$	$FFFFD0_H$	$FFFFD1_H$	$FFFFD2_H$	484
#12	CAN controller receive completion/ node status transition			$FFFFCC_H$	$FFFFCD_H$	$FFFFCE_H$	484
#13	Reserved	ICR01	$0000B1_H$	$FFFFC8_H$	$FFFFC9_H$	$FFFFCA_H$	–
#14	Reserved			$FFFFC4_H$	$FFFFC5_H$	$FFFFC6_H$	–
#15	CAN wake-up	ICR02	$0000B2_H$	$FFFFC0_H$	$FFFFC1_H$	$FFFFC2_H$	62
#16	Timebase timer			$FFFFBC_H$	$FFFFBD_H$	$FFFFBE_H$	195
#17	16-bit reload timer 0	ICR03	$0000B3_H$	$FFFFB8_H$	$FFFFB9_H$	$FFFFBA_H$	257
#18	8/10-bit A/D converter			$FFFFB4_H$	$FFFFB5_H$	$FFFFB6_H$	364
#19	16-bit free-run timer overflow	ICR04	$0000B4_H$	$FFFFB0_H$	$FFFFB1_H$	$FFFFB2_H$	233
#20	Reserved			$FFFFAC_H$	$FFFFAD_H$	$FFFFAE_H$	–
#21	Reserved	ICR05	$0000B5_H$	$FFFFA8_H$	$FFFFA9_H$	$FFFFAA_H$	–
#22	PPG timer channel 0/1 underflow			$FFFFA4_H$	$FFFFA5_H$	$FFFFA6_H$	305
#23	Input capture 0 fetched	ICR06	$0000B6_H$	$FFFFA0_H$	$FFFFA1_H$	$FFFFA2_H$	233
#24	External interrupt 4 (INT4)/ external interrupt 5 (INT5)			$FFFF9C_H$	$FFFF9D_H$	$FFFF9E_H$	328
#25	Input capture 1 fetched	ICR07	$0000B7_H$	$FFFF98_H$	$FFFF99_H$	$FFFF9A_H$	233
#26	PPG timer channel 2/3 underflow			$FFFF94_H$	$FFFF95_H$	$FFFF96_H$	305
#27	External interrupt 6 (INT6)/ external interrupt 7 (INT7)	ICR08	$0000B8_H$	$FFFF90_H$	$FFFF91_H$	$FFFF92_H$	328
#28	Watch timer			$FFFF8C_H$	$FFFF8D_H$	$FFFF8E_H$	281

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Table D-1 Interrupt Vector Index (2/2)

Interrupt Number	Interrupt Factor	Interrupt Control		Address in Vector Table			Page Number
		ICR	Address	Low	Middle	High	
#29	Reserved	ICR09	$0000B9_H$	$FFFF88_H$	$FFFF89_H$	$FFFF8A_H$	–
#30	Input capture 2 fetched Input capture 3 fetched			$FFFF84_H$	$FFFF85_H$	$FFFF86_H$	233
#31	Reserved	ICR10	$0000BA_H$	$FFFF80_H$	$FFFF81_H$	$FFFF82_H$	–
#32	Reserved			$FFFF7C_H$	$FFFF7D_H$	$FFFF7E_H$	–
#33	Reserved	ICR11	$0000BB_H$	$FFFF78_H$	$FFFF79_H$	$FFFF7A_H$	–
#34	Reserved			$FFFF74_H$	$FFFF75_H$	$FFFF76_H$	–
#35	Reserved	ICR12	$0000BC_H$	$FFFF70_H$	$FFFF71_H$	$FFFF72_H$	–
#36	16-bit reload timer 1			$FFFF6C_H$	$FFFF6D_H$	$FFFF6E_H$	257
#37	UART1 receive	ICR13	$0000BD_H$	$FFFF68_H$	$FFFF69_H$	$FFFF6A_H$	395
#38	UART1 transmit			$FFFF64_H$	$FFFF65_H$	$FFFF66_H$	395
#39	Reserved	ICR14	$0000BE_H$	$FFFF60_H$	$FFFF61_H$	$FFFF62_H$	–
#40	Reserved			$FFFF5C_H$	$FFFF5D_H$	$FFFF5E_H$	–
#41	Flash memory	ICR15	$0000BF_H$	$FFFF58_H$	$FFFF59_H$	$FFFF5A_H$	528
#42	Delayed interrupt generation module			$FFFF54_H$	$FFFF55_H$	$FFFF56_H$	320

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

APPENDIX

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

INDEX

The index follows on the next page.
This is listed in alphabetic order.

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Index

Numerics

16-bit Free-run Timer
- Block Diagram of 16-bit Free-run Timer 218
- Operation of 16-bit Free-run Timer 234
- Operation Timing of 16-bit Free-run Timer..... 235
- Setting of 16-bit Free-run Timer 234

16-bit Input/Output Timer
- 16-bit Input/Output Timer Interrupts and EI^{2}OS Function 233
- Block Diagram of 16-bit Input/Output Timer 217
- Block Diagram of Pins for 16-bit Input/Output Timer 222
- Configuration of 16-bit Input/Output Timer 216
- Correspondence between 16-bit Input/Output Timer Interrupt and EI^{2}OS..... 233
- Functions of 16-bit Input/Output Timer 216
- Generation of Interrupt Request from 16-bit Input/Output Timer 224
- Interrupt Control Bits and Interrupt Factors of 16-bit Input/Output Timer 233
- List of Registers and Reset Values of 16-bit Input/Output Timer 223
- Pins of 16-bit Input/Output Timer 222
- Precautions when 16-bit Input/Output Timer..... 239

16-bit PPG Output Operation Mode
- Setting for 16-bit PPG Output Operation Mode 310

16-bit Reload Registers
- 16-bit Reload Registers (TMRLR0,TMRLR1) 256

16-bit Reload Timer
- Baud Rate by Internal Timer (16-bit Reload Timer Output) 405
- Block Diagram for Pins of 16-bit Reload Timer 248
- Block Diagram of 16-bit Reload Timer..... 246
- Correspondence between 16-bit Reload Timer Interrupt and EI^{2}OS..... 257
- EI^{2}OS Function of 16-bit Reload Timer..... 257
- Generation of Interrupt Request from 16-bit Reload Timer 250
- Interrupts of 16-bit Reload Timer..... 257
- List of Registers and Reset Values of 16-bit Reload Timer 249
- Operation Modes of 16-bit Reload Timer..... 244
- Pins of 16-bit Reload Timer 248
- Precautions when Using 16-bit Reload Timer 268
- Setting of 16-bit Reload Timer 258

16-bit Timer Register
- 16-bit Timer Registers (TMR0,TMR1) 255
- Operating State of 16-bit Timer Register 259
- Operation as 16-bit Timer Register Underflows 261, 266

24-bit Operand
- Linear Addressing by Specifying 24-bit Operand 28

2-channel Independent Operation Mode
- Setting for 8-bit PPG Output 2-channel Independent Operation Mode.....308

32-bit Register
- Addressing by Indirect-specifying 32-bit Register 28

512 Kbit Flash Memory
- Features of 512 Kbit Flash Memory.....528
- Overview of 512 Kbit Flash Memory.....528
- Program Example of 512 Kbit Flash Memory.....553
- Sector Configuration of 512 Kbit Flash Memory 529

8+8-bit PPG Output Operation Mode
- Setting for 8+8-bit PPG Output Operation Mode 313

8-/10-bit A/D Converter
- 8-/10-bit A/D Converter Interrupt and EI^{2}OS 364
- A/D-converted Data Protection Function in 8-/10-bit A/D Converter.....373
- Block Diagram of 8-/10-bit A/D Converter.....349
- Conversion Modes of 8-/10-bit A/D Converter 348
- EI^{2}OS Function of 8-/10-bit A/D Converter.....364
- Function of 8-/10-bit A/D Converter.....348
- Generation of Interrupt from 8-/10-bit A/D Converter 353
- List of Registers and Reset Values of 8-/10-bit A/D Converter.....353
- Pins of 8-/10-bit A/D Converter 352
- Precautions when Using 8-/10-bit A/D Converter 375

8-/16-bit PPG Timer
- 8-/16-bit PPG Timer Interrupt and EI^{2}OS Function 306
- Block Diagram of 8-/16-bit PPG Timer 0 290
- Block Diagram of 8-/16-bit PPG Timer 1 293
- Block Diagram of 8-/16-bit PPG Timer Pins.....296
- Correspondence between 8-/16-bit PPG Timer Interrupt and EI^{2}OS 305
- Functions of 8-/16-bit PPG Timer.....286

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Generation of Interrupt Request from 8-/16-bit PPG Timer 297
Interrupts of 8-/16-bit PPG Timer 305
List of Registers and Reset Values of 8-/16-bit PPG Timer 297
Operation Modes of 8-/16-bit PPG Timer 286
Operation of 8-/16-bit PPG Timer 307
Pins of 8-/16-bit PPG Timer 296
Precautions when Using 8-/16-bit PPG Timer 316

8-bit PPG Output
Setting for 8-bit PPG Output 2-channel Independent Operation Mode 308

A

A
Accumulator (A) 36

A/D Control Status Register
A/D Control Status Register (High) (ADCS: H) 354
A/D Control Status Register (Low) (ADCS: L) 356

A/D Converter
8-/10-bit A/D Converter Interrupt and EI^{2}OS 364
A/D-converted Data Protection Function in 8-/10-bit A/D Converter 373
Block Diagram of 8-/10-bit A/D Converter 349
Conversion Modes of 8-/10-bit A/D Converter 348
EI^{2}OS Function of 8-/10-bit A/D Converter 364
Function of 8-/10-bit A/D Converter 348
Generation of Interrupt from 8-/10-bit A/D Converter 353
Interrupt of A/D Converter 364
List of Registers and Reset Values of 8-/10-bit A/D Converter 353
Pins of 8-/10-bit A/D Converter 352
Precautions when Using 8-/10-bit A/D Converter 375

A/D Data Register
A/D Data Register (High) (ADCR: H) 359
A/D Data Register (Low) (ADCR: L) 361

A/D-converted Data
A/D-converted Data Protection Function in 8-/10-bit A/D Converter 373

Acceptance Mask Register
Acceptance Mask Register (AMR) 476

Acceptance Mask Select Register
Acceptance Mask Select Register (AMSR) 474

Access Space
Bank Registers and Access Space 29

Accumulator
Accumulator (A) 36

ADB
Additional Bank Register (ADB) 49
Bank Select Prefix (PCB,DTB,ADB,SPB) 53

ADCR
A/D Data Register (High) (ADCR: H) 359
A/D Data Register (Low) (ADCR: L) 361

ADCS
A/D Control Status Register (High) (ADCS: H) 354
A/D Control Status Register (Low) (ADCS: L) 356
Continuous Conversion Mode (ADCS: MD1,MD0= "10_B") 365

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Pause-conversion Mode (ADCS: MD1,MD0= "11_B") 365
Single Conversion Mode (ADCS: MD1,MD0= "00_B" or "01_B") 365
Additional Bank Register
Additional Bank Register (ADB) 49
Address
Effective Address Field..... 573, 590
Address Detection Control Register
Address Detection Control Register (PACSR) 511
Address Match Detection Function
Block Diagram of Address Match Detection Function 509
List of Registers and Reset Values of Address Match Detection Function 510
Operation of Address Match Detection Function 515
Operation of Address Match Detection Function at Storing Patch Program in E^2PROM 519
Overview of Address Match Detection Function 508
Program Example for Address Match Detection Function 521
Addressing
Addressing 572
Addressing by Indirect-specifying 32-bit Register 28
Bank Addressing and Default Space..... 30
Direct Addressing..... 574
Indirect Addressing 580
Linear Addressing and Bank Addressing 27
ADER
Analog Input Enable Register (ADER)..... 362
All Data Erase
All Data Erase from Flash Memory (Chip Erase) 548
AMR
Acceptance Mask Register (AMR)..... 476
AMSR
Acceptance Mask Select Register (AMSR) 474
Analog Input Enable Register
Analog Input Enable Register (ADER)..... 362
Array
Array of Prefix Codes..... 58
Asynchronous Mode
Operation in Asynchronous Mode..... 410

B

Bank
Access to FF Bank by ROM Mirroring Function 524
Register Bank 51
Setting of Each Bank and Data Access.....49
Bank Addressing
Bank Addressing and Default Space 30
Linear Addressing and Bank Addressing.....27
Bank Registers
Bank Registers and Access Space.....29
Bank Select Prefix
Bank Select Prefix (PCB,DTB,ADB,SPB) 53
BAP
Buffer Address Pointer (BAP).....86
Basic Configuration
Basic Configuration of Serial Programming Connection for MB90F387/S 558
Baud Rate
Baud Rate by Dedicated Baud Rate Generator 402
Baud Rate by External Clock 407
Baud Rate by Internal Timer (16-bit Reload Timer Output).....405
Select of UART1 Baud Rate 400
Bidirectional Communication
Bidirectional Communication Function.....418
Bit Timing
Calculation of Bit Timing 448
Bit Timing Register
Bit Timing Register (BTR) 446
Bit Timing Segment
Definition of Bit Timing Segment 447
Block Diagram
Block Diagram for Pins of 16-bit Reload Timer 248
Block Diagram for Pins of CAN Controller 433
Block Diagram of 16-bit Free-run Timer.....218
Block Diagram of 16-bit Input/Output Timer.....217
Block Diagram of 16-bit Reload Timer 246
Block Diagram of 8-/10-bit A/D Converter.....349
Block Diagram of 8-/16-bit PPG Timer 0 290
Block Diagram of 8-/16-bit PPG Timer 1 293
Block Diagram of 8-/16-bit PPG Timer Pins.....296
Block Diagram of Address Match Detection Function 509
Block Diagram of CAN Controller 429
Block Diagram of Clock Generation Section 112
Block Diagram of Delayed Interrupt Generation Module 321
Block Diagram of DTP/External Interrupt 329
Block Diagram of External Reset Pin.....103
Block Diagram of Input Capture.....220
Block Diagram of Low-power Consumption Circuit 127
Block Diagram of MB90385 Series 8
Block Diagram of Pins 331
Block Diagram of Pins for 16-bit Input/Output Timer 222

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Block Diagram of Pins of Port 2 (General-purpose I/O Port).....167
Block Diagram of Pins of Port 3.....172
Block Diagram of Pins of Port 4.....177
Block Diagram of Pins of Port 5.....182
Block Diagram of Pins of UART1.....383
Block Diagram of Port 1 Pins (in Single Chip Mode).....162
Block Diagram of ROM Mirroring Function Select Module.....524
Block Diagram of Timebase Timer.....190
Block Diagram of UART1.....380
Block Diagram of Watch Timer.....276
Block Diagram of Watchdog Timer.....205
Details of Pins in Block Diagram.....218, 221

BTR
Bit Timing Register (BTR).....446

Buffer Address Pointer
Buffer Address Pointer (BAP).....86

Bus Mode
Bus Mode.....154

BVAL
Caution for Disabling Message Buffers by BVAL bits.....503

BVALR
Message Buffer Valid Register (BVALR).....450

C

CAN
Program Example of CAN Transmission and Reception.....504

CAN Controller
Block Diagram for Pins of CAN Controller.....433
Block Diagram of CAN Controller.....429
CAN Controller Registers.....434
Explanation of Operation of CAN Controller.....486
Generation of Interrupt Request by CAN Controller.....436
Interrupts of CAN Controller.....484
Overview of CAN Controller.....428
Pins of CAN Controller.....433
Registers and Vector Tables Related to Interrupt of CAN Controller.....485

CCR
Configuration of Condition Code Register (CCR).....43

CDCR
Communication Prescaler Control Register 1 (CDCR1).....393

Channels
Channels and PPG Pins of PPG Timers.....289

Chip Erase
All Data Erase from Flash Memory (Chip Erase).....548

Circuit
Block Diagram of Low-power Consumption Circuit.....127

CKSCR
Configuration of Clock Select Register (CKSCR).....115

Clock
Baud Rate by External Clock.....407
Block Diagram of Clock Generation Section.....112
Clock.....109
Clock Supply Map.....110
Connection of Oscillator and External Clock.....123
Machine Clock.....119
Oscillation Clock Frequency and Serial Clock Input Frequency.....560
Register in Clock Generation Section and List of Reset Values.....114
Setting Operation Clock of Watchdog Timer.....283
Supply of Operation Clock.....199

Clock Generation
Block Diagram of Clock Generation Section.....112
Register in Clock Generation Section and List of Reset Values.....114

Clock Mode
Clock Mode.....118, 125
Transition of Clock Mode.....118, 147

Clock Select Register
Configuration of Clock Select Register (CKSCR).....115

Clock Supply
Clock Supply.....189
Cycle of Clock Supply.....275

Clock Synchronous Mode
Operation in Clock Synchronous Mode (Operation Mode 2).....415

CMR
Common Register Bank Prefix (CMR).....55

Command Sequence
Command Sequence Table.....533

Common Register Bank Prefix
Common Register Bank Prefix (CMR).....55

Communication
Bidirectional Communication Function.....418
Master/Slave Type Communication Function.....420

Communication Prescaler Control Register
Communication Prescaler Control Register 1 (CDCR1).....393

Condition Code Register
Configuration of Condition Code Register (CCR).....43

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Connection
Example of Minimum Connection to Flash Microcontroller Programmer (Power Supplied from Flash Microcontroller Programmer)..... 567
Example of Minimum Connection to Flash Microcontroller Programmer (User Power Supply Used) 565
Connection Example
Connection Example in Single-chip Mode (Power Supplied from Flash Microcontroller Programmer)..... 563
Connection Example in Single-chip Mode (User Power Supply Used) 561
Consumption
Block Diagram of Low-power Consumption Circuit 127
CPU Operation Modes and Current Consumption 124
Continuous Conversion Mode
Continuous Conversion Mode (ADCS: MD1,MD0= "10_B") 365
Operation of Continuous Conversion Mode 368
Continuous Conversion mode
Setting of Continuous Conversion mode..... 368
Control Status Register
Control Status Register (High) (CSR: H) 437
Control Status Register (Low) (CSR: L) 439
Conversion
Conversion Using EI^{2}OS 372
Conversion Mode
Continuous Conversion Mode (ADCS: MD1,MD0= "10_B") 365
Conversion Modes of 8-/10-bit A/D Converter 348
Operation of Continuous Conversion Mode 368
Operation of Pause-conversion Mode 370
Operation of Single Conversion Mode..... 366
Pause-conversion Mode (ADCS: MD1,MD0= "11_B") 365
Setting of Continuous Conversion mode..... 368
Setting of Pause-conversion Mode 370
Setting of Single Conversion Mode..... 366
Single Conversion Mode (ADCS: MD1,MD0= "00_B" or "01_B")..... 365
Count Clock Select Register
PPG0/1 Count Clock Select Register (PPG01) 302
CPU
CPU and Resources for MB90385 Series..... 6
CPU Intermittent Operation Mode
CPU Intermittent Operation Mode 125
Operation in CPU Intermittent Operation Mode 133
CPU Operation Modes
CPU Operation Modes and Current Consumption 124
CSR
Control Status Register (High) (CSR: H).....437
Control Status Register (Low) (CSR: L).....439
Current Consumption
CPU Operation Modes and Current Consumption 124
Cycle
Cycle of Clock Supply.....275
Processing of Program for Measuring Cycle Using Input Capture.....240

D

Data Access
Setting of Each Bank and Data Access.....49
Data Bank Register
Data Bank Register (DTB).....49
Data Counter
Data Counter (DCT).....84
Data Polling Flag
Data Polling Flag (DQ7).....537
Data Programming
Data Programming Procedure 546
Data Programming to Flash Memory 546
Data Register
Data Register (DTR) 483
DCT
Data Counter (DCT).....84
Dedicated Baud Rate Generator
Baud Rate by Dedicated Baud Rate Generator 402
Dedicated Registers
Configuration of Dedicated Registers.....33
Dedicated Registers and General-purpose Register 35
Default Space
Bank Addressing and Default Space 30
Delayed Interrupt Generation Module
Block Diagram of Delayed Interrupt Generation Module 321
Explanation of Operation of Delayed Interrupt Generation Module 324
List of Registers and Reset Values in Delayed Interrupt Generation Module.....322
Overview of Delayed Interrupt Generation Module 320
Precautions when Using Delayed Interrupt Generation Module 325
Program Example of Delayed Interrupt Generation Module 326

DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

www.DataSheet4U.com

Delayed Interrupt Request Generate/Cancel Register
Delayed Interrupt Request Generate/Cancel Register (DIRR).....323
Descriptor
Configuration of EI^{2}OS Descriptor (ISD).....82
Detect Address
Setting Detect Address 515
Detect Address Setting Registers
Detect Address Setting Registers (PADR0 and PADR1).....513
Functions of Detect Address Setting Registers 514
Detection Level Setting Register
Detection Level Setting Register (ELVR) (High) 334
Detection Level Setting Register (ELVR) (Low) 335
Direct Addressing
Direct Addressing 574
Direct Page Register
Direct Page Register (DPR) 48
DIRR
Delayed Interrupt Request Generate/Cancel Register (DIRR).....323
Disabling Message Buffers
Caution for Disabling Message Buffers by BVAL bits 503
DLC Register
DLC Register (DLCR).....482
DLCR
DLC Register (DLCR).....482
DPR
Direct Page Register (DPR) 48
DQ2
Toggle Bit Flag (DQ2) 542
DQ3
Sector Erase Timer Flag (DQ3) 541
DQ5
Timing Limit Over Flag (DQ5) 540
DQ6
Toggle Bit Flag (DQ6) 539
DQ7
Data Polling Flag (DQ7).....537
DTB
Bank Select Prefix (PCB,DTB,ADB,SPB).....53
Data Bank Register (DTB).....49
DTP Function
DTP Function 340
Program Example of DTP Function 344
DTP/External Interrupt
Block Diagram of DTP/External Interrupt 329
DTP/External Interrupt Operation.....337
List of Registers and Reset Values in DTP/External Interrupt 331
Pins of DTP/External Interrupt.....331
Precautions when Using DTP/External Interrupt 341
Program Example of DTP/External Interrupt Function 343
Setting of DTP/External Interrupt.....336
DTP/External Interrupt Enable Register
DTP/External Interrupt Enable Register (ENIR) 333
DTP/External Interrupt Factor Register
DTP/External Interrupt Factor Register (EIRR) 332
DTP/External Interrupt Function
DTP/External Interrupt Function 328
DTR
Data Register (DTR) 483

E

E^{2}PROM
E^{2}PROM Memory Map.....517
System Configuration and E^{2}PROM Memory Map 516
Effective Address
Effective Address Field.....573, 590
EI^{2}OS
16-bit Input/Output Timer Interrupts and EI^{2}OS Function 233
8-/10-bit A/D Converter Interrupt and EI^{2}OS 364
8-/16-bit PPG Timer Interrupt and EI^{2}OS Function 306
Conversion Using EI^{2}OS.....372
Correspondence between 16-bit Input/Output Timer Interrupt and EI^{2}OS 233
Correspondence between 16-bit Reload Timer Interrupt and EI^{2}OS 257
Correspondence between 8-/16-bit PPG Timer Interrupt and EI^{2}OS 305
Correspondence between Timebase Timer Interrupt and EI^{2}OS.....195
EI^{2}OS.....80
EI^{2}OS Function of 16-bit Reload Timer.....257
EI^{2}OS Function of 8-/10-bit A/D Converter 364
EI^{2}OS Function of UART1.....396
EI^{2}OS Processing Time (time for one transfer) 89
Interrupt Related to UART1 and EI^{2}OS 396
Operation of EI^{2}OS.....81, 87
Procedure for Use of EI^{2}OS.....88
Program Example of EI^{2}OS.....97
Watch Timer Interrupt and EI^{2}OS Function 281
EI^{2}OS Descriptor
Configuration of EI^{2}OS Descriptor (ISD).....82
EI^{2}OS Status Register
EI^{2}OS Status Register (ISCS) 85

DataSheet4U.com

www.DataSheet4U.com

EIRR
DTP/External Interrupt Factor Register (EIRR) 332
ELVR
Detection Level Setting Register (ELVR) (High) 334
Detection Level Setting Register (ELVR) (Low) 335
ENIR
DTP/External Interrupt Enable Register (ENIR) 333
Erase
All Data Erase from Flash Memory (Chip Erase) 548
Sector Erase Suspension in Flash Memory 551
Erase Resumption
Erase Resumption in Flash Memory 552
Erasing
Detailed Explanation of Programming and Erasing Flash Memory 544
Erasing Any Data in Flash Memory (Sector Erasing) 549
Programming and Erasing Flash Memory 528
Erasing Procedure
Erasing Procedure for Flash Memory Sectors 549
Error
Node Status Transition due to Error Occurrence 445
Event Count Mode
Event Count Mode 244
Operation in Event Count Mode 267
Program Example in Event Count Mode 271
Setting of Event Count Mode 265
Exception Processing
Exception Processing 91
Execution Cycle Count
Calculating the Execution Cycle Count 588
Execution Cycle Count 587
Extended I/O
Extended I/O Area 23
External Clock
Baud Rate by External Clock 407
Connection of Oscillator and External Clock 123
External Interrupt
External Interrupt Function 339
External Reset
Block Diagram of External Reset Pin 103

F

F^{2}MC-16LX
F^{2}MC-16LX Instruction List 594
Factor
Correspondence of Reset Factor Bit and Reset Factor 107
Notes on Reset Factor Bit 107
Fetch
Mode Fetch 105
FF Bank
Access to FF Bank by ROM Mirroring Function 524
Flag Change Inhibit Prefix
Flag Change Inhibit Prefix (NCC) 56
Flag Set
Generation of Receive Interrupt and Timing of Flag Set 397
Generation of Transmit Interrupt and Timing of Flag Set 399
Flags
Hardware Sequence Flags 535
Flash Memory
All Data Erase from Flash Memory (Chip Erase) 548
Data Programming to Flash Memory 546
Detailed Explanation of Programming and Erasing Flash Memory 544
Erase Resumption in Flash Memory 552
Erasing Any Data in Flash Memory (Sector Erasing) 549
Erasing Procedure for Flash Memory Sectors 549
Features of 512 Kbit Flash Memory 528
List of Registers and Reset Values of Flash Memory 529
Overview of 512 Kbit Flash Memory 528
Program Example of 512 Kbit Flash Memory 553
Programming and Erasing Flash Memory 528
Read/Reset State in Flash Memory 545
Sector Configuration of 512 Kbit Flash Memory 529
Sector Erase Suspension in Flash Memory 551
Flash Memory Control Status Register
Flash Memory Control Status Register (FMCS) 530
Flash Microcontroller Programmer
Connection Example in Single-chip Mode (Power Supplied from Flash Microcontroller Programmer) 563
Example of Minimum Connection to Flash Microcontroller Programmer (Power Supplied from Flash Microcontroller Programmer) 567
Example of Minimum Connection to Flash Microcontroller Programmer (User Power Supply Used) 565

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Flash Microcontroller Programmer System
Flash Microcontroller Programmer System Configuration (Made by Yokogawa Digital Computer Corporation).....560
FMCS
Flash Memory Control Status Register (FMCS) 530
FPT-48P-M26
Package Dimension of FPT-48P-M26 10
Pin Assignment (FPT-48P-M26) 9
Free-run Timer
Block Diagram of 16-bit Free-run Timer 218
Operation of 16-bit Free-run Timer.....234
Operation Timing of 16-bit Free-run Timer 235
Setting of 16-bit Free-run Timer.....234
Frequency
Oscillation Clock Frequency and Serial Clock Input Frequency 560

G

General
Operation of Port 3 (General - purpose I/O port) 174
General-purpose I/O Port
Block Diagram of Pins of Port 2 (General-purpose I/O Port).....167
Operation of Port 2 (General-purpose I/O Port) 169
General-purpose Register
Configuration of General-purpose Register.....50
Dedicated Registers and General-purpose Register 35
General-purpose Register Area and Register Bank Pointer.....45
Generator
Baud Rate by Dedicated Baud Rate Generator 402

H

Handling Devices
Precautions when Handling Devices 18
Hardware Interrupt
Hardware Interrupt.....71
Hardware Interrupt Inhibition.....72
Mechanism of Hardware Interrupt 72
Operation of Hardware Interrupt.....75
Procedure for Use of Hardware Interrupt.....76
Return from Hardware Interrupt 74
Start of Hardware Interrupt 74
Hardware Sequence Flags
Hardware Sequence Flags.....535

I

I/O
I/O Area 23
I/O Address Pointer
I/O Address Pointer (IOA).....84
I/O Circuit
I/O Circuit 14
I/O Port
Block Diagram of Pins of Port 2 (General-purpose I/O Port) 167
I/O Port Function.....158
Operation of Port 2 (General-purpose I/O Port) 169
Operation of Port 3 (General - purpose I/O port) 174
Registers of I/O Ports.....160
ICR
Bit Configuration of Interrupt Control Register (ICR) 68
Interrupt Control Register (ICR00 to ICR15).....66
ICS
Input Capture Control Status Registers (ICS01 and ICS23) 229
ID Register
ID Register (IDR).....479
IDE Register
IDE Register (IDER).....452
IDER
IDE Register (IDER).....452
IDR
ID Register (IDR).....479
ILM
Interrupt Level Mask Register (ILM).....46
Image Access
Image Access to Internal ROM 25
Independent Operation Mode
Setting for 8-bit PPG Output 2-channel Independent Operation Mode.....308
Index
Interrupt Vector Index.....642
Pin Function Index.....640
Register Index 631
Indirect Addressing
Indirect Addressing.....580
Indirect-specifying
Addressing by Indirect-specifying 32-bit Register 28
Input Capture
Block Diagram of Input Capture 220
Operation of Input Capture 237
Operation Timing of Input Capture.....238
Processing of Program for Measuring Cycle Using Input Capture.....240
Setting of Input Capture 236

DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

www.DataSheet4U.com

INDEX

Input Capture Control Status Registers
Input Capture Control Status Registers (ICS01 and ICS23) 229
Input Capture Data Registers
Input Capture Data Registers 0 to 3 (IPCP0 to IPCP3) 232
Operation of Input Capture Data Registers 0 to 3 (IPCP0 to IPCP3) 232
Input/Output Pins
State of Input/Output Pins (Single-chip Mode) 145
Input/Output Timer
16-bit Input/Output Timer Interrupts and EI^{2}OS Function..... 233
Block Diagram of 16-bit Input/Output Timer 217
Block Diagram of Pins for 16-bit Input/Output Timer 222
Configuration of 16-bit Input/Output Timer 216
Correspondence between 16-bit Input/Output Timer Interrupt and EI^{2}OS..... 233
Functions of 16-bit Input/Output Timer 216
Generation of Interrupt Request from 16-bit Input/Output Timer 224
Interrupt Control Bits and Interrupt Factors of 16-bit Input/Output Timer 233
List of Registers and Reset Values of 16-bit Input/Output Timer 223
Pins of 16-bit Input/Output Timer 222
Precautions when 16-bit Input/Output Timer 239
Instruction
Description of instruction presentation items and symbols..... 591
F^{2}MC-16LX Instruction List 594
Prefix Code and Interrupt Inhibit Instruction..... 57
Instruction Map
Structure of Instruction Map..... 609
Instruction Types
Instruction Types..... 571
Internal Clock Mode
Internal Clock Mode..... 244
Operation in Internal Clock Mode..... 261
Program Example in Internal Clock Mode 269
Setting of Internal Clock Mode..... 260
Internal ROM
Image Access to Internal ROM..... 25
Internal Timer
Baud Rate by Internal Timer (16-bit Reload Timer Output) 405
Interrupt
16-bit Input/Output Timer Interrupts and EI^{2}OS Function..... 233
8-/10-bit A/D Converter Interrupt and EI^{2}OS 364
8-/16-bit PPG Timer Interrupt and EI^{2}OS Function 306
Block Diagram of DTP/External Interrupt 329
Cancellation of Standby Mode by Interrupt 146
Correspondence between 16-bit Input/Output Timer Interrupt and EI^{2}OS..... 233
Correspondence between 16-bit Reload Timer Interrupt and EI^{2}OS 257
Correspondence between 8-/16-bit PPG Timer Interrupt and EI^{2}OS 305
Correspondence between Timebase Timer Interrupt and EI^{2}OS..... 195
Details of Pins and Interrupt Numbers..... 330
DTP/External Interrupt Function 328
DTP/External Interrupt Operation..... 337
External Interrupt Function 339
Generation of Interrupt from 8-/10-bit A/D Converter 353
Generation of Receive Interrupt and Timing of Flag Set..... 397
Generation of Transmit Interrupt and Timing of Flag Set..... 399
Hardware Interrupt..... 71
Hardware Interrupt Inhibition..... 72
Interrupt Control Bits and Interrupt Factors of 16-bit Input/Output Timer..... 233
Interrupt Number 321
Interrupts of 16-bit Reload Timer 257
Interrupts of 8-/16-bit PPG Timer..... 305
Interrupt of A/D Converter..... 364
Interrupts of CAN Controller 484
Interrupt of UART1 395
Interrupt Operation..... 60
Interrupt Related to UART1 and EI^{2}OS 396
List of Registers and Reset Values in DTP/External Interrupt 331
Mechanism of Hardware Interrupt 72
Multiple Interrupts 77
Operation of Hardware Interrupt..... 75
Pins of DTP/External Interrupt 331
Precautions when Using DTP/External Interrupt 341
Procedure for Use of Hardware Interrupt..... 76
Program Example of DTP/External Interrupt Function 343
Registers and Vector Tables Related to Interrupt of CAN Controller 485
Return from Hardware Interrupt 74
Setting of DTP/External Interrupt 336
Start and Operation of Software Interrupt 79
Start of Hardware Interrupt 74
Timebase Timer Interrupt 195
Type and Function of Interrupt..... 59
Watch Timer Interrupt..... 281
Watch Timer Interrupt and EI^{2}OS Function..... 281

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Interrupt Control Register
Bit Configuration of Interrupt Control Register (ICR) 68
Function of Interrupt Control Register 69
Interrupt Control Register (ICR00 to ICR15) 66
Interrupt Control Register List 64
Interrupt Factor,Interrupt Vector, and Interrupt Control Register 62
Interrupt Factor
Interrupt Control Bits and Interrupt Factors of 16-bit Input/Output Timer 233
Interrupt Factor,Interrupt Vector, and Interrupt Control Register 62
Interrupt Inhibit Instruction
Prefix Code and Interrupt Inhibit Instruction 57
Interrupt Level Mask Register
Interrupt Level Mask Register (ILM) 46
Interrupt Number
Interrupt Number 321
Interrupt Processing
Program Example of Interrupt Processing 95
Stack Operation at Return from Interrupt Processing 94
Stack Operation at Starting Interrupt Processing 94
Time Required to Start Interrupt Processing 92
Interrupt Request
Generation of Interrupt Request by CAN Controller 436
Generation of Interrupt Request from 16-bit Input/Output Timer 224
Generation of Interrupt Request from 16-bit Reload Timer 250
Generation of Interrupt Request from 8-/16-bit PPG Timer 297
Generation of Interrupt Request from Timebase Timer 192
Generation of Interrupt Request from Watch Timer 278
Interrupt Request Generation by UART1 384
Interrupt Vector
Interrupt Factor,Interrupt Vector, and Interrupt Control Register 62
Interrupt Vector 61
Interrupt Vector Index
Interrupt Vector Index 642
Interval Timer
Functions of Interval Timer 188
Interval Timer Function 196, 274, 282
IOA
I/O Address Pointer (IOA) 84
IPCP
Input Capture Data Registers 0 to 3 (IPCP0 to IPCP3) 232
Operation of Input Capture Data Registers 0 to 3 (IPCP0 to IPCP3) 232
ISCS
EI^{2}OS Status Register (ISCS) 85
ISD
Configuration of EI^{2}OS Descriptor (ISD) 82

L

Last Event Indicate Register
Last Event Indicate Register (LEIR) 442
LEIR
Last Event Indicate Register (LEIR) 442
Linear Addressing
Linear Addressing and Bank Addressing 27
Linear Addressing by Specifying 24-bit Operand 28
Lineup
Product Lineup for MB90385 Series 5
List
List of Registers and Reset Values of 16-bit Reload Timer 249
Low-power Consumption
Block Diagram of Low-power Consumption Circuit 127
Low-Power Consumption Mode Control Register
Low-power Consumption Mode Control Register (LPMCR) 130
Low-power Consumption Mode Control Register and Reset Values 129
Notes on Accessing the Low-Power Consumption Mode Control Register (LPMCR) to Enter the Standby Mode 148
LPMCR
Low-power Consumption Mode Control Register (LPMCR) 130
Notes on Accessing the Low-Power Consumption Mode Control Register (LPMCR) to Enter the Standby Mode 148

M

Machine Clock
Machine Clock 119
Master/Slave Type Communication
Master/Slave Type Communication Function 420
MB90385 Series
Block Diagram of MB90385 Series 8
CPU and Resources for MB90385 Series 6
Features of MB90385 Series 2
Memory Map for MB90385 Series 24
Product Lineup for MB90385 Series 5

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

MB90F387/S
Basic Configuration of Serial Programming Connection for MB90F387/S 558
MD
Continuous Conversion Mode (ADCS: MD1,MD0= "10_B") 365
Pause-conversion Mode (ADCS: MD1,MD0= "11_B") 365
Setting of Mode Pins (MD2 to MD0) 150
Single Conversion Mode (ADCS: MD1,MD0= "00_B" or "01_B") 365
Memory Access Mode
Selection of Memory Access Mode 155
Memory Map
E^2PROM Memory Map 517
Memory Map 26
Memory Map for MB90385 Series 24
System Configuration and E^2PROM Memory Map 516
Memory Space
Memory Space 22
Memory Space when ROM Mirroring Function Enabled/Disabled 525
Message Buffer
Caution for Disabling Message Buffers by BVAL bits 503
Message Buffers 478
Procedure for Receiving Message Buffer (x) 498
Procedure for Transmitting message Buffer (x) 495
Setting Configuration of Multiple Message Buffer 501
Message Buffer Valid Register
Message Buffer Valid Register (BVALR) 450
Minimum Connection
Example of Minimum Connection to Flash Microcontroller Programmer (Power Supplied from Flash Microcontroller Programmer) 567
Example of Minimum Connection to Flash Microcontroller Programmer (User Power Supply Used) 565
Mode
Block Diagram of Port 1 Pins (in Single Chip Mode) 162
Bus Mode 154
Cancellation of Standby Mode by Interrupt 146
Classification of Modes 149
Clock Mode 118, 125
Connection Example in Single-chip Mode (Power Supplied from Flash Microcontroller Programmer) 563
Connection Example in Single-chip Mode (User Power Supply Used) 561
Continuous Conversion Mode (ADCS: MD1,MD0= "10_B") 365
Conversion Modes of 8-/10-bit A/D Converter 348
CPU Intermittent Operation Mode 125
CPU Operation Modes and Current Consumption 124
Notes on Accessing the Low-Power Consumption Mode Control Register (LPMCR) to Enter the Standby Mode 148
Event Count Mode 244
Function of Registers for Port 1 (in Single Chip Mode) 163
Internal Clock Mode 244
Mode Pin 104
Note on Cancelling Standby Mode 146
Notes on the Transition to Standby Mode 146
Operating State in Each Standby Mode 134
Operation in Asynchronous Mode 410
Operation in Clock Synchronous Mode (Operation Mode 2) 415
Operation in CPU Intermittent Operation Mode 133
Operation in Event Count Mode 267
Operation in Internal Clock Mode 261
Operation Mode 149
Operation Modes of 16-bit Reload Timer 244
Operation Modes of 8-/16-bit PPG Timer 286
Operation of Continuous Conversion Mode 368
Operation of Pause-conversion Mode 370
Operation of Port 1 (in Single Chip Mode) 164
Operation of Single Conversion Mode 366
Oscillation Stabilization Wait Time in Standby Mode 104
Pause-conversion Mode (ADCS: MD1,MD0= "11_B") 365
Program Example in Event Count Mode 271
Program Example in Internal Clock Mode 269
Registers for Port 1 (in Single Chip Mode) 162
Return from Sleep Mode 136
Return from Stop Mode 142
Return from Timebase Timer Mode 140
Return from Watch mode 138
Selection of Memory Access Mode 155
Setting for 16-bit PPG Output Operation Mode 310
Setting for 8+8-bit PPG Output Operation Mode 313
Setting for 8-bit PPG Output 2-channel Independent Operation Mode 308
Setting of Continuous Conversion mode 368
Setting of Event Count Mode 265
Setting of Internal Clock Mode 260
Setting of Pause-conversion Mode 370
Setting of Single Conversion Mode 366
Single Conversion Mode (ADCS: MD1,MD0= "00_B" or "01_B") 365

DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

www.DataSheet4U.com

Standby Mode 125
State of Input/Output Pins (Single-chip Mode) 145
Stop Mode.....141
Transition of Clock Mode 118, 147
Transition to Sleep Mode.....135
Transition to Standby Mode 146
Transition to Timebase Timer Mode 139
Transition to Watch mode.....137

Mode Data
Mode Data 152
Setting Mode Data 153
State of Pins after Mode Data Read.....108

Mode Fetch
Mode Fetch 105

Mode Pins
Setting Mode Pins.....151
Setting of Mode Pins (MD2 to MD0).....150

Multi-byte Data
Access to Multi-byte Data.....32
Storage of Multi-byte Data in Stack 32
Store of Multi-byte Data in RAM 31

Multi-byte Length
Storage of Multi-byte Length Operand.....31

Multiple Interrupts
Multiple Interrupts 77

Multiple Message Buffer
Setting Configuration of Multiple Message Buffer 501

Multiplication Rate
Selection of PLL Clock Multiplication Rate 119

N

NCC
Flag Change Inhibit Prefix (NCC) 56

Node Status Transition
Node Status Transition due to Error Occurrence 445

O

Operand
Linear Addressing by Specifying 24-bit Operand 28
Storage of Multi-byte Length Operand.....31

Operating State
Setting and Operating State.....518

Operation Clock
Supply of Operation Clock.....199

Operation Mode
CPU Intermittent Operation Mode 125
CPU Operation Modes and Current Consumption 124
Operation in CPU Intermittent Operation Mode 133
Operation Mode 149
Operation Modes of 16-bit Reload Timer.....244
Setting for 8-bit PPG Output 2-channel Independent Operation Mode.....308

Operation Mode Control Register
PPG0 Operation Mode Control Register (PPGC0) 298
PPG1 Operation Mode Control Register (PPGC1) 300

Oscillation Clock Frequency
Oscillation Clock Frequency and Serial Clock Input Frequency.....560

Oscillation Stabilization Wait Time
Operation as Oscillation Stabilization Wait Time Timer 197
Operation During Oscillation Stabilization Wait Time 122
Oscillation Stabilization Wait Time 147
Oscillation Stabilization Wait Time in Standby Mode 104
Oscillation Stabilization Wait Time Timer of Subclock.....283
Reset Sources and Oscillation Stabilization Wait Time 101

Oscillator
Connection of Oscillator and External Clock.....123

P

Package Dimension
Package Dimension of FPT-48P-M26.....10

PACSR
Address Detection Control Register (PACSR) 511

PADR
Detect Address Setting Registers (PADR0 and PADR1).....513

Patch
Operation of Address Match Detection Function at Storing Patch Program in E^2PROM 519

Patch Processing
Flow of Patch Processing 520

Pause-conversion Mode
Operation of Pause-conversion Mode 370
Pause-conversion Mode (ADCS: MD1,MD0= "11_B") 365
Setting of Pause-conversion Mode 370

PC
Program Counter (PC).....47

PCB
Bank Select Prefix (PCB,DTB,ADB,SPB) 53
Program Bank Register (PCB) 49

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

INDEX

Pin Assignment
Pin Assignment (FPT-48P-M26)..... 9
Pin Assignment of Port 1 161
Pin Assignment of Port 2 166
Pin Assignment of Port 3 171
Pin Assignment of Port 4 176
Pin Assignment of Port 5 181
Pin Description
Pin Description 11
Pin Function Index
Pin Function Index 640
PLL Clock Multiplication Rate
Selection of PLL Clock Multiplication Rate..... 119
Port 1
Block Diagram of Port 1 Pins (in Single Chip Mode) 162
Configuration of Port 1 161
Function of Registers for Port 1 (in Single Chip Mode) 163
Operation of Port 1 (in Single Chip Mode)..... 164
Pin Assignment of Port 1 161
Registers for Port 1 (in Single Chip Mode) 162
Port 2
Block Diagram of Pins of Port 2 (General-purpose I/O Port)..... 167
Configuration of Port 2 166
Function of Registers for Port 2 168
Operation of Port 2 (General-purpose I/O Port) 169
Pin Assignment of Port 2 166
Registers for Port 2..... 167
Port 3
Block Diagram of Pins of Port 3 172
Configuration of Port 3 171
Function of Registers for Port 3 173
Operation of Port 3 (General - purpose I/O port) 174
Pin Assignment of Port 3 171
Registers for Port 3..... 172
Port 4
Block Diagram of Pins of Port 4 177
Configuration of Port 4 176
Function of Registers for Port 4 178
Operation of Port 4..... 179
Pin Assignment of Port 4 176
Registers for Port 4..... 177
Port 5
Block Diagram of Pins of Port 5 182
Configuration of Port 5 181
Function of Registers for Port 5 183
Operation of Port 5..... 185
Pin Assignment of Port 5 181
Registers for Port 5..... 182
Power Supply
Connection Example in Single-chip Mode (User Power Supply Used).....561
Connection Example in Single-chip Mode (Power Supplied from Flash Microcontroller Programmer) 563
Example of Minimum Connection to Flash Microcontroller Programmer (Power Supplied from Flash Microcontroller Programmer) 567
Example of Minimum Connection to Flash Microcontroller Programmer (User Power Supply Used).....565
PPG
Channels and PPG Pins of PPG Timers.....289
PPG0 Operation Mode Control Register (PPGC0) 298
PPG0/1 Count Clock Select Register (PPG01) 302
PPG1 Operation Mode Control Register (PPGC1) 300
PPG Output
Setting for 8-bit PPG Output 2-channel Independent Operation Mode 308
PPG Output Operation Mode
Setting for 16-bit PPG Output Operation Mode 310
Setting for 8+8-bit PPG Output Operation Mode 313
PPG Reload Registers
PPG Reload Registers (PRLL0/PRLH0,PRLL1/PRLH1).....304
PPG Timer
8-/16-bit PPG Timer Interrupt and EI^{2}OS Function 306
Block Diagram of 8-/16-bit PPG Timer 0 290
Block Diagram of 8-/16-bit PPG Timer 1 293
Block Diagram of 8-/16-bit PPG Timer Pins 296
Channels and PPG Pins of PPG Timers.....289
Correspondence between 8-/16-bit PPG Timer Interrupt and EI^{2}OS 305
Functions of 8-/16-bit PPG Timer.....286
Generation of Interrupt Request from 8-/16-bit PPG Timer.....297
Interrupts of 8-/16-bit PPG Timer.....305
List of Registers and Reset Values of 8-/16-bit PPG Timer.....297
Operation Modes of 8-/16-bit PPG Timer.....286
Operation of 8-/16-bit PPG Timer 307
Pins of 8-/16-bit PPG Timer.....296
Precautions when Using 8-/16-bit PPG Timer 316
PPGC
PPG0 Operation Mode Control Register (PPGC0) 298

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

PPG1 Operation Mode Control Register (PPGC1) 300
Prefix
Bank Select Prefix (PCB,DTB,ADB,SPB)..... 53
Common Register Bank Prefix (CMR)..... 55
Flag Change Inhibit Prefix (NCC) 56
Prefix Code
Array of Prefix Codes..... 58
Prefix Code 52
Prefix Code and Interrupt Inhibit Instruction..... 57
Presentation items
Description of instruction presentation items and symbols 591
Presetting
Presetting 494
PRLH
PPG Reload Registers (PRLL0/PRLH0,PRLL1/PRLH1) 304
PRLL
PPG Reload Registers (PRLL0/PRLH0,PRLL1/PRLH1) 304
Processing
Exception Processing 91
Program Example of Interrupt Processing..... 95
Stack Operation at Return from Interrupt Processing 94
Stack Operation at Starting Interrupt Processing 94
Time Required to Start Interrupt Processing 92
Processing Time
EI^{2}OS Processing Time (time for one transfer) 89
Processor Status
Configuration of Processor Status (PS)..... 42
Product Lineup
Product Lineup for MB90385 Series 5
Program
Operation of Address Match Detection Function at Storing Patch Program in E^{2}PROM 519
Processing of Program for Measuring Cycle Using Input Capture 240
Program Execution..... 515
Program Bank Register
Program Bank Register (PCB)..... 49
Program Counter
Program Counter (PC)..... 47
Program Example
Program Example for Address Match Detection Function..... 521
Program Example for UART1..... 424
Program Example in Event Count Mode 271
Program Example in Internal Clock Mode..... 269
Program Example of 512 Kbit Flash Memory 553
Program Example of CAN Transmission and Reception..... 504
Program Example of Delayed Interrupt Generation Module 326
Program Example of DTP Function..... 344
Program Example of DTP/External Interrupt Function 343
Program Example of EI^{2}OS 97
Program Example of Interrupt Processing 95
Program Example of Timebase Timer..... 201
Program Example of Watch Timer 284
Program Example of Watchdog Timer..... 214
Programming
Data Programming Procedure 546
Data Programming to Flash Memory 546
Detailed Explanation of Programming and Erasing Flash Memory 544
Programming and Erasing Flash Memory 528
Protection
A/D-converted Data Protection Function in 8-/10-bit A/D Converter..... 373
PS
Configuration of Processor Status (PS) 42

R

RAM
RAM Area 23
Store of Multi-byte Data in RAM..... 31
RCR
Reception Complete Register (RCR) 466
Read
Read/Reset State in Flash Memory 545
Receive Interrupt
Generation of Receive Interrupt and Timing of Flag Set 397
Receive/Transmit Error Counter
Receive/Transmit Error Counter (RTEC) 444
Receiving
Procedure for Receiving Message Buffer (x) 498
Reception
Program Example of CAN Transmission and Reception..... 504
Reception..... 490
Reception Complete Interrupt Enable Register
Reception Complete Interrupt Enable Register (RIER) 472
Reception Complete Register
Reception Complete Register (RCR) 466
Reception Overrun Register
Reception Overrun Register (ROVRR) 470
Reception RTR Register
Reception RTR Register (RRTRR)..... 468

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Register Bank
Register Bank 51
Register Bank Pointer
General-purpose Register Area and Register Bank Pointer 45
Register Bank Pointer (RP) 45
Register Index
Register Index 631
Reload Registers
16-bit Reload Registers (TMRLR0,TMRLR1) 256
Reload Timer
Baud Rate by Internal Timer (16-bit Reload Timer Output) 405
Block Diagram for Pins of 16-bit Reload Timer 248
Block Diagram of 16-bit Reload Timer 246
Correspondence between 16-bit Reload Timer Interrupt and EI^{2}OS 257
EI^{2}OS Function of 16-bit Reload Timer 257
Generation of Interrupt Request from 16-bit Reload Timer 250
Interrupts of 16-bit Reload Timer 257
List of Registers and Reset Values of 16-bit Reload Timer 249
Operation Modes of 16-bit Reload Timer 244
Pins of 16-bit Reload Timer 248
Precautions when Using 16-bit Reload Timer 268
Setting of 16-bit Reload Timer 258
Remote Frame Receiving Wait Register
Remote Frame Receiving Wait Register (RFWTR) 458
Reset
Block Diagram of External Reset Pin 103
Flowchart of Reset Operation 104
List of Registers and Reset Values in Delayed Interrupt Generation Module 322
List of Registers and Reset Values in DTP/External Interrupt 331
List of Registers and Reset Values of 16-bit Input/Output Timer 223
List of Registers and Reset Values of 16-bit Reload Timer 249
List of Registers and Reset Values of 8-/10-bit A/D Converter 353
List of Registers and Reset Values of 8-/16-bit PPG Timer 297
List of Registers and Reset Values of Address Match Detection Function 510
List of Registers and Reset Values of Flash Memory 529
List of Registers and Reset Values of ROM Mirroring Function Select Module 525
List of Registers and Reset Values of Timebase Timer 192
List of Registers and Reset Values of Watch Timer 278
List of Registers and Reset Values of Watchdog Timer 207
Low-power Consumption Mode Control Register and Reset Values 129
Read/Reset State in Flash Memory 545
Register in Clock Generation Section and List of Reset Values 114
Reset Factor Bit 106
Reset Factors 99
Reset Sources and Oscillation Stabilization Wait Time 101
State of Pins at Reset 108
Reset Factor
Correspondence of Reset Factor Bit and Reset Factor 107
Notes on Reset Factor Bit 107
Reset Factor Bit 106
Reset Factors 99
Reset Sources
Reset Sources and Oscillation Stabilization Wait Time 101
Reset State
Read/Reset State in Flash Memory 545
Resources
CPU and Resources for MB90385 Series 6
RFWTR
Remote Frame Receiving Wait Register (RFWTR) 458
RIER
Reception Complete Interrupt Enable Register (RIER) 472
ROM
ROM Area 23
ROM Mirror Function Select Register
ROM Mirror Function Select Register (ROMM) 526
ROM Mirroring Function
Access to FF Bank by ROM Mirroring Function 524
Memory Space when ROM Mirroring Function Enabled/Disabled 525
ROM Mirroring Function Select Module
Block Diagram of ROM Mirroring Function Select Module 524
List of Registers and Reset Values of ROM Mirroring Function Select Module 525
ROMM
ROM Mirror Function Select Register (ROMM) 526

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

ROVRR
Reception Overrun Register (ROVRR) 470
RP
Register Bank Pointer (RP) 45
RRTRR
Reception RTR Register (RRTRR) 468
RTEC
Receive/Transmit Error Counter (RTEC) 444

S

SCR
Serial Control Register 1 (SCR1).....385
Sector
Erasing Procedure for Flash Memory Sectors 549
Sector Configuration of 512 Kbit Flash Memory 529
Sector Configuration
Sector Configuration of 512 Kbit Flash Memory 529
Sector Erase Suspension
Sector Erase Suspension in Flash Memory 551
Sector Erase Timer Flag
Sector Erase Timer Flag (DQ3) 541
Sector Erasing
Erasing Any Data in Flash Memory (Sector Erasing) 549
Segment
Definition of Bit Timing Segment.....447
Serial Clock Input Frequency
Oscillation Clock Frequency and Serial Clock Input Frequency 560
Serial Control Register
Serial Control Register 1 (SCR1).....385
Serial Input Data Register
Serial Input Data Register 1 (SIDR1) 391
Serial Mode Register
Serial Mode Register 1 (SMR1).....387
Serial Output Data Register
Serial Output Data Register 1 (SODR1) 392
Serial Programming Connection
Basic Configuration of Serial Programming Connection for MB90F387/S 558
Serial Status Register
Serial Status Register 1 (SSR1) 389
Setting
Setting of Internal Clock Mode.....260
SIDR
Serial Input Data Register 1 (SIDR1) 391
Single Chip Mode
Block Diagram of Port 1 Pins (in Single Chip Mode) 162
Connection Example in Single-chip Mode (Power Supplied from Flash Microcontroller Programmer) 563
Connection Example in Single-chip Mode (User Power Supply Used).....561
Function of Registers for Port 1 (in Single Chip Mode) 163
Operation of Port 1 (in Single Chip Mode) 164
Registers for Port 1 (in Single Chip Mode).....162
State of Input/Output Pins (Single-chip Mode) 145
Single Conversion Mode
Operation of Single Conversion Mode 366
Setting of Single Conversion Mode 366
Single Conversion Mode (ADCS: MD1,MD0= "00_B" or "01_B") 365
Sleep Mode
Return from Sleep Mode 136
Transition to Sleep Mode 135
SMR
Serial Mode Register 1 (SMR1) 387
SODR
Serial Output Data Register 1 (SODR1).....392
Software Interrupt
Return from Software Interrupt 79
Start and Operation of Software Interrupt.....79
SPB
Bank Select Prefix (PCB,DTB,ADB,SPB) 53
SSB
User Stack Bank Register (USB) and System Stack Bank Register (SSB) 49
SSP
System Stack Pointer (SSP) 40
SSR
Serial Status Register 1 (SSR1).....389
Stack
Stack Area 41
Stack Operation at Return from Interrupt Processing 94
Stack Operation at Starting Interrupt Processing 94
Stack Selection.....39
Storage of Multi-byte Data in Stack.....32
Standby Mode
Cancellation of Standby Mode by Interrupt 146
Notes on Accessing the Low-Power Consumption Mode Control Register (LPMCR) to Enter the Standby Mode.....148
Note on Cancelling Standby Mode 146
Notes on the Transition to Standby Mode.....146
Operating State in Each Standby Mode 134
Oscillation Stabilization Wait Time in Standby Mode 104
Standby Mode 125

DataSheet4U.com
www.DataSheet4U.com
DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Transition to Standby Mode 146
Start Interrupt Processing
Time Required to Start Interrupt Processing..... 92
Starting Interrupt Processing
Stack Operation at Starting Interrupt Processing 94
State
Read/Reset State in Flash Memory..... 545
State Transition Diagram
State Transition Diagram 144
Status Transition
Node Status Transition due to Error Occurrence 445
Stop Mode
Return from Stop Mode 142
Stop Mode..... 141
Storing Patch Program
Operation of Address Match Detection Function at Storing Patch Program in E^{2}PROM 519
Subclock
Oscillation Stabilization Wait Time Timer of Subclock 283
Supply Map
Clock Supply Map..... 110
Symbols
Description of instruction presentation items and symbols..... 591
Synchronous Mode
Operation in Clock Synchronous Mode (Operation Mode 2) 415
System Configuration
System Configuration and E^{2}PROM Memory Map 516
System Stack Bank Register
User Stack Bank Register (USB) and System Stack Bank Register (SSB) 49
System Stack Pointer
System Stack Pointer (SSP)..... 40

T

TBTC
Timebase Timer Control Register (TBTC)..... 193
TCANR
Transmission Cancel Register (TCANR) 460
TCCS
Timer Counter Control Status Register (TCCS) 225
TCDT
Count Operation of Timer Counter Data Register (TCDT) 227
Timer Counter Data Register (TCDT) 227
TCR
Transmission Complete Register (TCR).....462
TIER
Transmission Complete Interrupt Enable Register (TIER).....464
Timebase Timer
Block Diagram of Timebase Timer.....190
Correspondence between Timebase Timer Interrupt and EI^{2}OS.....195
Generation of Interrupt Request from Timebase Timer 192
List of Registers and Reset Values of Timebase Timer 192
Precautions when Using Timebase Timer.....200
Program Example of Timebase Timer.....201
Timebase Timer Interrupt 195
Timebase Timer Control Register
Timebase Timer Control Register (TBTC) 193
Timebase Timer Mode
Return from Timebase Timer Mode.....140
Transition to Timebase Timer Mode 139
Timer Control Status Registers
Timer Control Status Registers (High) (TMCSR0: H,TMCSR1: H) 251
Timer Control Status Registers (Low) (TMCSR0: L,TMCSR1: L).....253
Timer Counter Control Status Register
Timer Counter Control Status Register (TCCS) 225
Timer Counter Data Register
Count Operation of Timer Counter Data Register (TCDT).....227
Timer Counter Data Register (TCDT).....227
Timer Registers
16-bit Timer Registers (TMR0,TMR1) 255
Timing Limit Over Flag
Timing Limit Over Flag (DQ5) 540
TMCSR
Timer Control Status Registers (High) (TMCSR0: H,TMCSR1: H) 251
Timer Control Status Registers (Low) (TMCSR0: L,TMCSR1: L).....253
TMR
16-bit Timer Registers (TMR0,TMR1) 255
TMRLR
16-bit Reload Registers (TMRLR0,TMRLR1) 256
Toggle Bit Flag
Toggle Bit Flag (DQ2) 542
Toggle Bit Flag (DQ6) 539
Transfer
EI^{2}OS Processing Time (time for one transfer) 89

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

Transition
Node Status Transition due to Error Occurrence 445
Notes on the Transition to Standby Mode 146
State Transition Diagram 144
Transition of Clock Mode 147
Transition to Sleep Mode 135
Transition to Standby Mode 146
Transition to Timebase Timer Mode 139
Transition to Watch mode 137
Transmission
Program Example of CAN Transmission and Reception 504
Transmission 487
Transmission Cancel Register
Transmission Cancel Register (TCANR) 460
Transmission Complete Interrupt Enable Register
Transmission Complete Interrupt Enable Register (TIER) 464
Transmission Complete Register
Transmission Complete Register (TCR) 462
Transmission Request Register
Transmission Request Register (TREQR) 454
Transmission RTR Register
Transmission RTR Register (TRTRR) 456
Transmit Interrupt
Generation of Transmit Interrupt and Timing of Flag Set 399
Transmitting
Procedure for Transmitting message Buffer (x) 495
TREQR
Transmission Request Register (TREQR) 454
TRTRR
Transmission RTR Register (TRTRR) 456

U

UART1
Block Diagram of Pins of UART1 383
Block Diagram of UART1 380
EI^{2}OS Function of UART1 396
Function of UART1 378
Interrupt of UART1 395
Interrupt Related to UART1 and EI^{2}OS 396
Interrupt Request Generation by UART1 384
List of Registers in UART1 383
Operation of UART1 408
Precautions when Using UART1 423
Program Example for UART1 424
Select of UART1 Baud Rate 400
UART1 Pin 383
Underflow
Operation at Underflow 245
Underflows
Operation as 16-bit Timer Register Underflows 261, 266
USB
User Stack Bank Register (USB) and System Stack Bank Register (SSB) 49
User Power Supply
Example of Minimum Connection to Flash Microcontroller Programmer (User Power Supply Used) 565
Connection Example in Single-chip Mode (User Power Supply Used) 561
User Stack Bank Register
User Stack Bank Register (USB) and System Stack Bank Register (SSB) 49
User Stack Pointer
User Stack Pointer (USP) 40
USP
User Stack Pointer (USP) 40

V

Vector Tables
Registers and Vector Tables Related to Interrupt of CAN Controller 485

W

Watch mode
Return from Watch mode 138
Transition to Watch mode 137
Watch Timer
Block Diagram of Watch Timer 276
Generation of Interrupt Request from Watch Timer 278
List of Registers and Reset Values of Watch Timer 278
Program Example of Watch Timer 284
Watch Timer Counter 282
Watch Timer Interrupt 281
Watch Timer Interrupt and EI^{2}OS Function 281
Watch Timer Control Register
Watch Timer Control Register (WTC) 279
Watchdog Timer
Block Diagram of Watchdog Timer 205
Functions of Watchdog Timer 204
List of Registers and Reset Values of Watchdog Timer 207
Operation of Watchdog Timer 210
Precautions when Using Watchdog Timer 213
Program Example of Watchdog Timer 214
Setting Operation Clock of Watchdog Timer 283

DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

INDEX

Watchdog Timer Control Register
Watchdog Timer Control Register (WDTC)..... 208
WDTC
Watchdog Timer Control Register (WDTC)..... 208
WTC
Watch Timer Control Register (WTC)..... 279

et4U.com
DataShee
DataSheet4U.com
DataSheet4U.com
www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com

DataShee

DataSheet4U.com

CM44-10118-2E

FUJITSU SEMICONDUCTOR • CONTROLLER MANUAL

F^2MC^{TM}-16LX
16-BIT MICROCONTROLLER
MB90385 Series
HARDWARE MANUAL

November 2005 the second edition

Published **FUJITSU LIMITED** Electronic Devices

Edited Business Promotion Dept.

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

et4U.com

DataSheet4U.com

DataSheet4U.com

www.DataSheet4U.com

DataSheet4U.com