Hitachi 16-Bit Single-Chip Microcomputer

H8S/2639 Series,
H8S/2638 Series,
H8S/2636 Series

Hardware Manual

HITACHI

ADE-602-189B
Rev. 3.0
11/20/01
Hitachi, Ltd.

Hitachi semiconductor

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

Preface

This LSI has the internal 32-bit H8S/2600 CPU and includes a variety of peripheral functions necessary for a system configuration. It serves as a high-performance microcomputer.

The built-in peripheral devices include a 16-bit timer pulse unit (TPU), a programmable pulse generator (PPG), a watchdog timer unit (WDT), a serial communication interface (SCI), an A/D converter, a motor control PWM timer (PWM), a PC brake controller and I/O ports. It also has an internal data transfer controller (DTC), which performs high-speed data transfer without using the CPU, thus enabling the use of the LSI as an embedded microcomputer in various advanced control systems. Two types of internal ROM are available: flash memory (F-ZTAT™*) and mask ROM. The LSI can be used flexibly in a wide range of applications from applied equipment with varied specifications and early production models to full-scale mass-produced products.

Note: * F-ZTAT™ is a trademark of Hitachi, Ltd.

Target Users: This manual was written for users who will be using the H8S/2636, H8S/2638 and H8S/2639 in the design of application systems. Members of this audience are expected to understand the fundamentals of electrical circuits, logical circuits, and microcomputers.

Objective: This manual was written to explain the hardware functions and electrical characteristics of the H8S/2636, H8S/2638 and H8S/2639 to the above audience. Refer to the H8S/2600 Series, H8S/2000 Series Programming Manual for a detailed description of the instruction set.

Notes on reading this manual:

- In order to understand the overall functions of the chip
  Read the manual according to the contents. This manual can be roughly categorized into parts on the CPU, system control functions, peripheral functions and electrical characteristics.
- In order to understand the details of the CPU's functions
  Read the H8S/2600 Series, H8S/2000 Series Programming Manual.
- In order to understand the details of a register when its name is known
  The addresses, bits, and initial values of the registers are summarized in Appendix B, Internal I/O Registers.
  Example: Bit order: The MSB is on the left and the LSB is on the right.

Related Manuals: The latest versions of all related manuals are available from our web site. Please ensure you have the latest versions of all documents you require.
http://www.hitachisemiconductor.com/

H8S/2636, H8S/2638, H8S/2639 manuals:

Manual Title	ADE No.
H8S/2636, H8S/2638, H8S/2639 Hardware Manual	This manual
H8S/2600 Series, H8S/2000 Series Programming Manual	ADE-602-083

Users manuals for development tools:

Manual Title	ADE No.
C/C++ Complier, Assembler, Optimized Linkage Editor User's Manual	ADE-702-247
Simulator Debugger (for Windows) Users Manual	ADE-702-037
Hitachi Embedded Workshop Users Manual	ADE-702-201

Application Notes:

Manual Title	ADE No.
H8S Series Technical Q & A	ADE-502-059

Main Revisions and Additions in this Edition

Page	Item	Revisions (See Manual for Details)
	—	Content added and amended due to addition of mask ROM version and H8S/2639F-ZTAT
1	1.1 Overview	Description of Overview amended
4		Table 1.1 Overview Input clock frequency of clock pulse generator amended and product lineup amended
6	1.2 Internal Block Diagram	Figure 1-1 (a) Internal Block Diagram of H8S/2636 Note amended and added
7		Figure 1-1 (b) Internal Block Diagram of H8S/2638 and H8S/2639 Note amended and added
8	1.3.1 Pin Arrangement	Figure 1-2 Pin Arrangement of HD64F2636F (FP-128B: Top View) Note amended and added
9		Figure 1-3 Pin Arrangement of H8S/2638 Series (FP-128B: Top View) Note amended and added
10		Figure 1-4 Pin Arrangement H8S/2639 Series (FP-128B: Top View) Added
15	1.3.2 Pin Functions in Each Operating Mode	Table 1-2 Pin Functions in Each Operating Mode Note amended
16	1.3.3 Pin Functions	Table 1-3 Pin Functions Name of VCL pin changed to On-chip step-down power supply pin Note amended and added
21	1.4 Differences between H8S/2636, H8S/2638, and H8S/2639	Replaced
61	2.8.1 Overview	Figure 2-14 Processing States Note amended

Page	Item	Revisions (See Manual for Details)
82, 83	3.5 Address Map in Each Operating Mode	Figure 3-1 Memory Map in Each Operating Mode in the H8S/2636 Figure 3-2 Memory Map in Each Operating Mode in the H8S/2638 and H8S/2639 “External area” amended to “Address space”
107 to 109	5.3.3 Interrupt Exception Handling Vector Table	Table 5-4 Interrupt Sources, Vector Addresses, and Interrupt Priorities “Reserved” amended to “Reserved for system use”
120	5.5.5 IRQ Interrupts	Section 5.5.5 added
122	5.6.3 Operation	Notes on use amended
210	9.1 Overview	Table 9-1 Port Functions Port F SYNCI deleted
239	9.6.3 Pin Functions	Added
266	9.11 Port F 9.11.1 Overview	“IIC input pin* (SYNCI)” and Note deleted from description Figure 9-21 Port F Pin Functions SYNCI and Note deleted
270	9.11.3 Pin Functions	Table 9-19 Port F Pin Functions SYNCI and Note deleted
396	12.1 Overview 12.1.2 Block Diagram	Figure 12-1 (a) Block Diagram of WDT0 *2 and note added
403	12.2 Register Descriptions 12.2.2 Timer Control/Status Register (TCSR)	WDT0 Input Clock Select *2 and note added
407	12.2.4 Notes on Register Access	Writing to RSTCSR description amended Figure 12-3 Format of Data Written to RSTCSR (WDT0) RSTS bits deleted from “Writing to RSTE and RSTS bits”
527	15.2.4 I^2C Bus Mode Register (ICMR)	Bits 5 to 3—Serial Clock Select (CKS2 to CKS0): Transfer rate ø = 25 MHz deleted
544 to 546	15.3.2 Master Transmit Operation	Replaced
547 to 549	15.3.3 Master Receive Operation	Replaced
553	15.3.5 Slave Transmit Operation	Figure 15-10 Example of Slave Transmit Mode Operation Timing (MLS = 0) IRIC waveform [3] amended to [4]

Page	Item	Revisions (See Manual for Details)
563	15.4 Usage Notes	Table 15-7 Permissible SCL Rise Time (t_{Sr}) Values ø = 25 MHz time indication [ns] deleted
565		Table 15-8 I2C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf}) ø = 25 MHz time indication (at maximum transfer rate) [ns] deleted
579	16.2.2 General Status Register (GSR)	Description amended Bits 7 to 4 description amended
588	16.2.7 Transmit Acknowledge Register (TXACK)	TXACK bit figure amended Note added
589	16.2.8 Abort Acknowledge Register (ABACK)	ABACK bit figure amended Note added
590	16.2.9 Receive Complete Register (RXPR)	RXPR bit figure amended Note added
591	16.2.10 Remote Request Register (RFPR)	RFPR bit figure amended Note added
592	16.2.11 Interrupt Register (IRR)	IRR bit figure amended Note added Bit 15 table amended
634	16.5 Usage Notes	Note 4 Error counters description amended Note 8 added
651	17.4.3 Input Sampling and A/D Conversion Time	Table 17-4 A/D Conversion Time (Single Mode) A/D conversion time CKS0 = 0 min. value when CKS1 = 0 amended
652	17.6 Usage Notes	Setting Range of Analog Power Supply and Other Pins (2) description amended
676	19.2.6 PWM Duty Registers 1A, 1C, 1E, 1G (PWDTR1A, 1C, 1E, 1G)	Bits 15 to 13, 12, 11 and 10 description amended Bits 9 to 0 description added
678	19.2.7 PWM Buffer Registers 1A, 1C, 1E, 1G (PWBFR1A, 1C, 1E, 1G)	Bits 12, 9 to 0 description amended

Page	Item	Revisions (See Manual for Details)
679	19.2.8 PWM Duty Registers 2A to 2H (PWDTR2A to PWDTR2H)	Bits 15 to 10 description amended Bits 13 and 11 deleted Bit 12 deleted Bits 9 to 0 description added
680	19.2.9 PWM Buffer Registers 2A to 2D (PWBFR2A to PWBFR2D)	Bits 9 to 0 description amended
691 to 694	—	21A.1 Overview, 21A.2 Register Descriptions, 21A.3 Operation Added
710	21A.8.6 Flash Memory Power Control Register (FLPWCR)	Bit 7 description added
747	21A.15.1 Notes on Power-Down States	Note 2 added
755 to 758	—	21B.1 Overview, 21B.2 Register Descriptions, 21B.3 Operation Added
774	21B.8.6 Flash Memory Power Control Register (FLPWCR)	Bit 7 description added
783	21B.10.1 Program Mode	Replaced
784	21B.10.2 Program-Verify Mode	Replaced
788	21B.10.3 Erase Mode	Replaced
	21B.10.4 Erase-Verify Mode	Replaced
797	21B.14.1 Socket Adapter and Memory Map	Table 21B-14 Type of Socket Adapter Product name amended
809	21B.14.9 Notes on Power-Down States	Note 2 added
823, 824	22A.3.2 External Clock Input	Added
829 to 838	22B Clock Pulse Generator (H8S/2639 Series)	Added
840	23A.1 Overview	Table 23A-1 LSI Internal States in Each Mode PBC amended
847	23A.2.3 Low-Power Control Register (LPWRCR)	Description of bit 3 deleted

Page	Item	Revisions (See Manual for Details)
860	23B.1 Overview	Table 23B-1 LSI Internal States in Each Mode PBC amended
861		Table 23B-2 LSI Internal States in Each Mode (H8S/2639 Series) Added
870	23B.2.3 Low-Power Control Register (LPWRCR)	Bit 5 description added
886	23B.12 ϕ Clock Output Disabling Function	Table 23B-7 ϕ Pin State in Each Processing State Subactive mode amended
887	23B.13 Usage Notes	Section 23B.13 added
889 to 904	24.1 H8S/2636F-ZTAT Electrical Characteristics	— Preliminary — deleted
889	24.1.1 Absolute Maximum Ratings	Table 24-1 Absolute Maximum Ratings Power supply voltage symbol and Input voltage (OSC1, OSC2) ratings amended
890	24.1.2 Power Supply Voltage and Operating Frequency Range	Figure 24-1 Power Supply Voltage and Operating Ranges added Note added
893	24.1.3 DC Characteristics	Table 24-2 DC Characteristics Max. current dissipation value in standby mode amended from 2.0 to 20
900	24.1.4 AC Characteristics	Table 24-7 Timing of On-Chip Supporting Modules Transmit data setup time and transmit data hold time for HCAN amended to receive data setup time and receive data hold time, respectively
901	24.1.5 A/D Conversion Characteristics	Table 24-8 A/D Conversion Characteristics Max. values for nonlinearity error, offset error, and full-scale error amended
905 to 922	24.2 H8S/2638 Series Electrical Characteristics	— Preliminary — deleted
905	24.2.1 Absolute Maximum Ratings	Table 24-11 Absolute Maximum Ratings Power supply voltage symbol and Input voltage (OSC1, OSC2) ratings amended
906	24.2.2 Power Supply Voltage and Operating Frequency Range	Figure 24-3 Power Supply Voltage and Operating Ranges+ added Note added

Page	Item	Revisions (See Manual for Details)
908	24.2.3 DC Characteristics	Table 24-12 DC Characteristics Output high voltage amended
909		Table 24-12 DC Characteristics Max. current dissipation value in standby mode amended from 2.0 to 20
911		Table 24-14 Bus Drive Characteristics Added
917	24.2.4 AC Characteristics	Table 24-18 Timing of On-Chip Supporting Modules Transmit data setup time and transmit data hold time for HCAN amended to receive data setup time and receive data hold time, respectively
918		Table 24-19 I^2C Bus Timing [Option] Conditions amended
919	24.2.5 A/D Conversion Characteristics	Table 24-20 A/D Conversion Characteristics Max. values for nonlinearity error, offset error, and full-scale error amended
921, 922	24.2.7 Flash Memory Characteristics	Replaced
923 to 940	24.3 H8S/2639 Series Electrical Characteristics	Added
1044	B.1 Address	Bit 3 RFCUT in address H'FDEC 3 added and notes amended
1198, 1201	B.2 Functions	PWBFR1A—PWM Buffer Register 1A PWBFR1C—PWM Buffer Register 1C PWBFR1E—PWM Buffer Register 1E PWBFR1G—PWM Buffer Register 1G Output terminal select and duty amended PWBFR2A—PWM Buffer Register 2A PWBFR2B—PWM Buffer Register 2B PWBFR2C—PWM Buffer Register 2C PWBFR2D—PWM Buffer Register 1D Duty amended
1295		I^2C Bus Mode Register Transfer clock select transfer route ø = 25 MHz deleted

Page	Item	Revisions (See Manual for Details)
1327	C.5 Port A Block Diagram	Figure C-5 (a) Port A Block Diagram (Pin PA0) Amended
1328 to 1330		Figure C-5 (b) - (d) Port A Block Diagram (Pin PA1) - (Pin PA3) Added
1335	C.10 Port F Block Diagram	Figure C.10 (a) Port F Block Diagram (Pin PF0) Amended
1347	F Product Code Lineup	Added
1348	G Package Dimensions	Amended

HITACHI

Contents

Section 1 Overview 1
1.1 Overview 1
1.2 Internal Block Diagram 6
1.3 Pin Description 8
1.3.1 Pin Arrangement 8
1.3.2 Pin Functions in Each Operating Mode 11
1.3.3 Pin Functions 16
1.4 Differences between H8S/2636, H8S/2638, and H8S/2639 21

Section 2 CPU 23
2.1 Overview 23
2.1.1 Features 23
2.1.2 Differences between H8S/2600 CPU and H8S/2000 CPU 24
2.1.3 Differences from H8/300 CPU 25
2.1.4 Differences from H8/300H CPU 25
2.2 CPU Operating Modes 26
2.3 Address Space 31
2.4 Register Configuration 32
2.4.1 Overview 32
2.4.2 General Registers 33
2.4.3 Control Registers 34
2.4.4 Initial Register Values 36
2.5 Data Formats 37
2.5.1 General Register Data Formats 37
2.5.2 Memory Data Formats 39
2.6 Instruction Set 40
2.6.1 Overview 40
2.6.2 Instructions and Addressing Modes 41
2.6.3 Table of Instructions Classified by Function 43
2.6.4 Basic Instruction Formats 52
2.7 Addressing Modes and Effective Address Calculation 54
2.7.1 Addressing Mode 54
2.7.2 Effective Address Calculation 57
2.8 Processing States 61
2.8.1 Overview 61
2.8.2 Reset State 62
2.8.3 Exception-Handling State 63
2.8.4 Program Execution State 66
2.8.5 Bus-Released State 66

2.8.6 Power-Down State 66
2.9 Basic Timing 67
2.9.1 Overview 67
2.9.2 On-Chip Memory (ROM, RAM) 67
2.9.3 On-Chip Supporting Module Access Timing 69
2.9.4 On-Chip HCAN Module Access Timing 71
2.9.5 Port H and J Register Access Timing 72
2.9.6 External Address Space Access Timing 74
2.10 Usage Note 74
2.10.1 TAS Instruction 74
2.10.2 STM/LDM Instructions 74
2.10.3 Caution to observe when using bit manipulation instructions 74

Section 3 MCU Operating Modes 75
3.1 Overview 75
3.1.1 Operating Mode Selection 75
3.1.2 Register Configuration 76
3.2 Register Descriptions 76
3.2.1 Mode Control Register (MDCR) 76
3.2.2 System Control Register (SYSCR) 77
3.2.3 Pin Function Control Register (PFCR) 78
3.3 Operating Mode Descriptions 80
3.3.1 Mode 4 80
3.3.2 Mode 5 80
3.3.3 Mode 6 80
3.3.4 Mode 7 80
3.4 Pin Functions in Each Operating Mode 81
3.5 Address Map in Each Operating Mode 81

Section 4 Exception Handling 85
4.1 Overview 85
4.1.1 Exception Handling Types and Priority 85
4.1.2 Exception Handling Operation 86
4.1.3 Exception Vector Table 86
4.2 Reset 88
4.2.1 Overview 88
4.2.2 Reset Sequence 88
4.2.3 Interrupts after Reset 90
4.2.4 State of On-Chip Supporting Modules after Reset Release 91
4.3 Traces 91
4.4 Interrupts 92
4.5 Trap Instruction 93
4.6 Stack Status after Exception Handling 94

4.7 Notes on Use of the Stack 95

Section 5 Interrupt Controller 97
5.1 Overview 97
5.1.1 Features 97
5.1.2 Block Diagram 98
5.1.3 Pin Configuration 99
5.1.4 Register Configuration 99
5.2 Register Descriptions 100
5.2.1 System Control Register (SYSCR) 100
5.2.2 Interrupt Priority Registers A to H, J to M (IPRA to IPRH, IPRJ to IPRM) 101
5.2.3 IRQ Enable Register (IER) 102
5.2.4 IRQ Sense Control Registers H and L (ISCRH, ISCRL) 103
5.2.5 IRQ Status Register (ISR) 104
5.3 Interrupt Sources 105
5.3.1 External Interrupts 105
5.3.2 Internal Interrupts 106
5.3.3 Interrupt Exception Handling Vector Table 106
5.4 Interrupt Operation 110
5.4.1 Interrupt Control Modes and Interrupt Operation 110
5.4.2 Interrupt Control Mode 0 113
5.4.3 Interrupt Control Mode 2 115
5.4.4 Interrupt Exception Handling Sequence 117
5.4.5 Interrupt Response Times 118
5.5 Usage Notes 119
5.5.1 Contention between Interrupt Generation and Disabling 119
5.5.2 Instructions that Disable Interrupts 120
5.5.3 Times when Interrupts are Disabled 120
5.5.4 Interrupts during Execution of EEPMOV Instruction 120
5.5.5 IRQ Interrupts 120
5.6 DTC Activation by Interrupt 121
5.6.1 Overview 121
5.6.2 Block Diagram 121
5.6.3 Operation 122

Section 6 PC Break Controller (PBC) 123
6.1 Overview 123
6.1.1 Features 123
6.1.2 Block Diagram 124
6.1.3 Register Configuration 125
6.2 Register Descriptions 125
6.2.1 Break Address Register A (BARA) 125
6.2.2 Break Address Register B (BARB) 126

6.2.3 Break Control Register A (BCRA) 126
6.2.4 Break Control Register B (BCRB) 128
6.2.5 Module Stop Control Register C (MSTPCRC) 128
6.3 Operation 129
6.3.1 PC Break Interrupt Due to Instruction Fetch 129
6.3.2 PC Break Interrupt Due to Data Access 129
6.3.3 Notes on PC Break Interrupt Handling 130
6.3.4 Operation in Transitions to Power-Down Modes 130
6.3.5 PC Break Operation in Continuous Data Transfer 131
6.3.6 When Instruction Execution is Delayed by One State 132
6.3.7 Additional Notes 133

Section 7 Bus Controller 135
7.1 Overview 135
7.1.1 Features 135
7.1.2 Block Diagram 136
7.1.3 Pin Configuration 137
7.1.4 Register Configuration 137
7.2 Register Descriptions 138
7.2.1 Bus Width Control Register (ABWCR) 138
7.2.2 Access State Control Register (ASTCR) 138
7.2.3 Wait Control Registers H and L (WCRH, WCRL) 140
7.2.4 Bus Control Register H (BCRH) 144
7.2.5 Bus Control Register L (BCRL) 145
7.2.6 Pin Function Control Register (PFCR) 146
7.3 Overview of Bus Control 148
7.3.1 Area Partitioning 148
7.3.2 Bus Specifications 149
7.3.3 Memory Interfaces 150
7.3.4 Interface Specifications for Each Area 151
7.4 Basic Bus Interface 152
7.4.1 Overview 152
7.4.2 Data Size and Data Alignment 152
7.4.3 Valid Strobes 154
7.4.4 Basic Timing 155
7.4.5 Wait Control 163
7.5 Burst ROM Interface 165
7.5.1 Overview 165
7.5.2 Basic Timing 165
7.5.3 Wait Control 167
7.6 Idle Cycle 168
7.6.1 Operation 168
7.6.2 Pin States During Idle Cycles 170

7.7 Write Data Buffer Function 172
7.8 Bus Arbitration 173
7.8.1 Overview 173
7.8.2 Operation 173
7.8.3 Bus Transfer Timing 173
7.9 Resets and the Bus Controller 174

Section 8 Data Transfer Controller (DTC) 175
8.1 Overview 175
8.1.1 Features 175
8.1.2 Block Diagram 176
8.1.3 Register Configuration 177
8.2 Register Descriptions 178
8.2.1 DTC Mode Register A (MRA) 178
8.2.2 DTC Mode Register B (MRB) 180
8.2.3 DTC Source Address Register (SAR) 181
8.2.4 DTC Destination Address Register (DAR) 181
8.2.5 DTC Transfer Count Register A (CRA) 181
8.2.6 DTC Transfer Count Register B (CRB) 182
8.2.7 DTC Enable Registers (DTCER) 182
8.2.8 DTC Vector Register (DTVECR) 183
8.2.9 Module Stop Control Register A (MSTPCRA) 184
8.3 Operation 186
8.3.1 Overview 186
8.3.2 Activation Sources 188
8.3.3 DTC Vector Table 189
8.3.4 Location of Register Information in Address Space 193
8.3.5 Normal Mode 194
8.3.6 Repeat Mode 195
8.3.7 Block Transfer Mode 196
8.3.8 Chain Transfer 198
8.3.9 Operation Timing 199
8.3.10 Number of DTC Execution States 200
8.3.11 Procedures for Using DTC 202
8.3.12 Examples of Use of the DTC 203
8.4 Interrupts 206
8.5 Usage Notes 206

Section 9 I/O Ports 207
9.1 Overview 207
9.2 Port 1 211
9.2.1 Overview 211
9.2.2 Register Configuration 212

9.2.3 Pin Functions 214
9.3 Port 3 226
9.3.1 Overview 226
9.3.2 Register Configuration 226
9.3.3 Pin Functions 229
9.4 Port 4 231
9.4.1 Overview 231
9.4.2 Register Configuration 232
9.4.3 Pin Functions 232
9.5 Port 9 233
9.5.1 Overview 233
9.5.2 Register Configuration 234
9.5.3 Pin Functions 234
9.6 Port A 235
9.6.1 Overview 235
9.6.2 Register Configuration 236
9.6.3 Pin Functions 239
9.6.4 Pin Functions 241
9.6.5 MOS Input Pull-Up Function 242
9.7 Port B 243
9.7.1 Overview 243
9.7.2 Register Configuration 244
9.7.3 Pin Functions 247
9.7.4 MOS Input Pull-Up Function 248
9.8 Port C 249
9.8.1 Overview 249
9.8.2 Register Configuration 250
9.8.3 Pin Functions for Each Mode 253
9.8.4 MOS Input Pull-Up Function 255
9.9 Port D 256
9.9.1 Overview 256
9.9.2 Register Configuration 257
9.9.3 Pin Functions 259
9.9.4 MOS Input Pull-Up Function 260
9.10 Port E 261
9.10.1 Overview 261
9.10.2 Register Configuration 262
9.10.3 Pin Functions 264
9.10.4 MOS Input Pull-Up Function 265
9.11 Port F 266
9.11.1 Overview 266
9.11.2 Register Configuration 267
9.11.3 Pin Functions 269

9.12 Port H 271
9.12.1 Overview 271
9.12.2 Register Configuration 271
9.12.3 Pin Functions 273
9.13 Port J 274
9.13.1 Overview 274
9.13.2 Register Configuration 274
9.13.3 Pin Functions 276

Section 10 16-Bit Timer Pulse Unit (TPU) 277
10.1 Overview 277
10.1.1 Features 277
10.1.2 Block Diagram 281
10.1.3 Pin Configuration 282
10.1.4 Register Configuration 284
10.2 Register Descriptions 286
10.2.1 Timer Control Register (TCR) 286
10.2.2 Timer Mode Register (TMDR) 291
10.2.3 Timer I/O Control Register (TIOR) 293
10.2.4 Timer Interrupt Enable Register (TIER) 306
10.2.5 Timer Status Register (TSR) 309
10.2.6 Timer Counter (TCNT) 313
10.2.7 Timer General Register (TGR) 314
10.2.8 Timer Start Register (TSTR) 315
10.2.9 Timer Synchro Register (TSYR) 316
10.2.10 Module Stop Control Register A (MSTPCRA) 317
10.3 Interface to Bus Master 318
10.3.1 16-Bit Registers 318
10.3.2 8-Bit Registers 318
10.4 Operation 320
10.4.1 Overview 320
10.4.2 Basic Functions 321
10.4.3 Synchronous Operation 327
10.4.4 Buffer Operation 329
10.4.5 Cascaded Operation 333
10.4.6 PWM Modes 335
10.4.7 Phase Counting Mode 340
10.5 Interrupts 347
10.5.1 Interrupt Sources and Priorities 347
10.5.2 DTC Activation 349
10.5.3 A/D Converter Activation 349
10.6 Operation Timing 350
10.6.1 Input/Output Timing 350

10.6.2 Interrupt Signal Timing 354
10.7 Usage Notes 358

Section 11 Programmable Pulse Generator (PPG) 369
11.1 Overview 369
11.1.1 Features 369
11.1.2 Block Diagram 370
11.1.3 Pin Configuration 371
11.1.4 Registers 372
11.2 Register Descriptions 373
11.2.1 Next Data Enable Registers H and L (NDERH, NDERL) 373
11.2.2 Output Data Registers H and L (PODRH, PODRL) 374
11.2.3 Next Data Registers H and L (NDRH, NDRL) 375
11.2.4 Notes on NDR Access 375
11.2.5 PPG Output Control Register (PCR) 377
11.2.6 PPG Output Mode Register (PMR) 379
11.2.7 Port 1 Data Direction Register (P1DDR) 382
11.2.8 Module Stop Control Register A (MSTPCRA) 382
11.3 Operation 383
11.3.1 Overview 383
11.3.2 Output Timing 384
11.3.3 Normal Pulse Output 385
11.3.4 Non-Overlapping Pulse Output 387
11.3.5 Inverted Pulse Output 390
11.3.6 Pulse Output Triggered by Input Capture 391
11.4 Usage Notes 392

Section 12 Watchdog Timer 395
12.1 Overview 395
12.1.1 Features 395
12.1.2 Block Diagram 396
12.1.3 Pin Configuration 398
12.1.4 Register Configuration 398
12.2 Register Descriptions 399
12.2.1 Timer Counter (TCNT) 399
12.2.2 Timer Control/Status Register (TCSR) 400
12.2.3 Reset Control/Status Register (RSTCSR) 405
12.2.4 Notes on Register Access 406
12.3 Operation 408
12.3.1 Watchdog Timer Operation 408
12.3.2 Interval Timer Operation 410
12.3.3 Timing of Setting Overflow Flag (OVF) 410
12.3.4 Timing of Setting of Watchdog Timer Overflow Flag (WOVF) 411

12.4 Interrupts 412
12.5 Usage Notes 412
12.5.1 Contention between Timer Counter (TCNT) Write and Increment 412
12.5.2 Changing Value of PSS* and CKS2 to CKS0 413
12.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode 413
12.5.4 Internal Reset in Watchdog Timer Mode 413
12.5.5 OVF Flag Clearing in Interval Timer Mode 413

Section 13 Serial Communication Interface (SCI) 415
13.1 Overview 415
13.1.1 Features 415
13.1.2 Block Diagram 417
13.1.3 Pin Configuration 418
13.1.4 Register Configuration 419
13.2 Register Descriptions 420
13.2.1 Receive Shift Register (RSR) 420
13.2.2 Receive Data Register (RDR) 420
13.2.3 Transmit Shift Register (TSR) 421
13.2.4 Transmit Data Register (TDR) 421
13.2.5 Serial Mode Register (SMR) 422
13.2.6 Serial Control Register (SCR) 425
13.2.7 Serial Status Register (SSR) 429
13.2.8 Bit Rate Register (BRR) 433
13.2.9 Smart Card Mode Register (SCMR) 440
13.2.10 Module Stop Control Register B (MSTPCRB) 441
13.3 Operation 443
13.3.1 Overview 443
13.3.2 Operation in Asynchronous Mode 445
13.3.3 Multiprocessor Communication Function 456
13.3.4 Operation in Clocked Synchronous Mode 464
13.4 SCI Interrupts 472
13.5 Usage Notes 473

Section 14 Smart Card Interface 483
14.1 Overview 483
14.1.1 Features 483
14.1.2 Block Diagram 484
14.1.3 Pin Configuration 485
14.1.4 Register Configuration 486
14.2 Register Descriptions 487
14.2.1 Smart Card Mode Register (SCMR) 487
14.2.2 Serial Status Register (SSR) 489
14.2.3 Serial Mode Register (SMR) 491

14.2.4 Serial Control Register (SCR) 493
14.3 Operation 494
14.3.1 Overview 494
14.3.2 Pin Connections 494
14.3.3 Data Format 496
14.3.4 Register Settings 498
14.3.5 Clock 500
14.3.6 Data Transfer Operations 502
14.3.7 Operation in GSM Mode 509
14.3.8 Operation in Block Transfer Mode 510
14.4 Usage Notes 511

Section 15 I^2C Bus Interface [Option] (Only for the H8S/2638 and H8S/2639) 515
15.1 Overview 515
15.1.1 Features 515
15.1.2 Block Diagram 516
15.1.3 Input/Output Pins 518
15.1.4 Register Configuration 519
15.2 Register Descriptions 520
15.2.1 I^2C Bus Data Register (ICDR) 520
15.2.2 Slave Address Register (SAR) 523
15.2.3 Second Slave Address Register (SARX) 524
15.2.4 I^2C Bus Mode Register (ICMR) 525
15.2.5 I^2C Bus Control Register (ICCR) 528
15.2.6 I^2C Bus Status Register (ICSR) 535
15.2.7 Serial Control Register X (SCRX) 540
15.2.8 DDC Switch Register (DDCSWR) 541
15.2.9 Module Stop Control Register B (MSTPCRB) 542
15.3 Operation 543
15.3.1 I^2C Bus Data Format 543
15.3.2 Master Transmit Operation 544
15.3.3 Master Receive Operation 547
15.3.4 Slave Receive Operation 550
15.3.5 Slave Transmit Operation 552
15.3.6 IRIC Setting Timing and SCL Control 554
15.3.7 Operation Using the DTC 555
15.3.8 Noise Canceler 556
15.3.9 Sample Flowcharts 556
15.3.10 Initialization of Internal State 560
15.4 Usage Notes 562

Section 16 Hitachi Controller Area Network (HCAN).....571
16.1 Overview..... 571
16.1.1 Features..... 571
16.1.2 Block Diagram..... 572
16.1.3 Pin Configuration..... 573
16.1.4 Register Configuration..... 574
16.2 Register Descriptions..... 578
16.2.1 Master Control Register (MCR)..... 578
16.2.2 General Status Register (GSR)..... 579
16.2.3 Bit Configuration Register (BCR)..... 581
16.2.4 Mailbox Configuration Register (MBCR)..... 585
16.2.5 Transmit Wait Register (TXPR)..... 586
16.2.6 Transmit Wait Cancel Register (TXCR)..... 587
16.2.7 Transmit Acknowledge Register (TXACK)..... 588
16.2.8 Abort Acknowledge Register (ABACK)..... 589
16.2.9 Receive Complete Register (RXPR)..... 590
16.2.10 Remote Request Register (RFPR)..... 591
16.2.11 Interrupt Register (IRR)..... 592
16.2.12 Mailbox Interrupt Mask Register (MBIMR)..... 596
16.2.13 Interrupt Mask Register (IMR)..... 597
16.2.14 Receive Error Counter (REC)..... 599
16.2.15 Transmit Error Counter (TEC)..... 600
16.2.16 Unread Message Status Register (UMSR)..... 600
16.2.17 Local Acceptance Filter Masks (LAFML, LAFMH)..... 601
16.2.18 Message Control (MC0 to MC15)..... 603
16.2.19 Message Data (MD0 to MD15)..... 607
16.2.20 Module Stop Control Register C (MSTPCRC)..... 608
16.3 Operation..... 610
16.3.1 Hardware and Software Resets..... 610
16.3.2 Initialization after Hardware Reset..... 613
16.3.3 Transmit Mode..... 616
16.3.4 Receive Mode..... 622
16.3.5 HCAN Sleep Mode..... 628
16.3.6 HCAN Halt Mode..... 630
16.3.7 Interrupt Interface..... 630
16.3.8 DTC Interface..... 632
16.4 CAN Bus Interface..... 633
16.5 Usage Notes..... 634

Section 17 A/D Converter.....635
17.1 Overview..... 635
17.1.1 Features..... 635

17.1.2 Block Diagram 636
17.1.3 Pin Configuration 637
17.1.4 Register Configuration 638
17.2 Register Descriptions 639
17.2.1 A/D Data Registers A to D (ADDRA to ADDRD) 639
17.2.2 A/D Control/Status Register (ADCSR) 640
17.2.3 A/D Control Register (ADCR) 643
17.2.4 Module Stop Control Register A (MSTPCRA) 644
17.3 Interface to Bus Master 645
17.4 Operation 646
17.4.1 Single Mode (SCAN = 0) 646
17.4.2 Scan Mode (SCAN = 1) 648
17.4.3 Input Sampling and A/D Conversion Time 650
17.4.4 External Trigger Input Timing 651
17.5 Interrupts 652
17.6 Usage Notes 652

Section 18 D/A Converter 659
18.1 Overview 659
18.1.1 Features 659
18.1.2 Block Diagram 659
18.1.3 Input and Output Pins 661
18.1.4 Register Configuration 661
18.2 Register Descriptions 662
18.2.1 D/A Data Registers 0, 1 (DADR0, DADR1) 662
18.2.2 D/A Control Register 01 (DACR01) 662
18.2.3 Module Stop Control Register A (MSTPCRA) 664
18.3 Operation 665

Section 19 Motor Control PWM Timer 667
19.1 Overview 667
19.1.1 Features 667
19.1.2 Block Diagram 668
19.1.3 Pin Configuration 670
19.1.4 Register Configuration 671
19.2 Register Descriptions 672
19.2.1 PWM Control Registers 1 and 2 (PWCR1, PWCR2) 672
19.2.2 PWM Output Control Registers 1 and 2 (PWOCR1, PWOCR2) 673
19.2.3 PWM Polarity Registers 1 and 2 (PWPR1, PWPR2) 674
19.2.4 PWM Counters 1 and 2 (PWCNT1, PWCNT2) 675
19.2.5 PWM Cycle Registers 1 and 2 (PWCYR1, PWCYR2) 675
19.2.6 PWM Duty Registers 1A, 1C, 1E, 1G (PWDTR1A, 1C, 1E, 1G) 676
19.2.7 PWM Buffer Registers 1A, 1C, 1E, 1G (PWBFR1A, 1C, 1E, 1G) 678

19.2.8 PWM Duty Registers 2A to 2H (PWDTR2A to PWDTR2H) ... 679
19.2.9 PWM Buffer Registers 2A to 2D (PWBFR2A to PWBFR2D) ... 680
19.2.10 Module Stop Control Register D (MSTPCRD) ... 681
19.3 Bus Master Interface ... 682
19.3.1 16-Bit Data Registers ... 682
19.3.2 8-Bit Data Registers ... 682
19.4 Operation ... 683
19.4.1 PWM Channel 1 Operation ... 683
19.4.2 PWM Channel 2 Operation ... 684
19.5 Usage Note ... 685

Section 20 RAM ... 687
20.1 Overview ... 687
20.1.1 Block Diagram ... 687
20.1.2 Register Configuration ... 688
20.2 Register Descriptions ... 689
20.2.1 System Control Register (SYSCR) ... 689
20.3 Operation ... 689
20.4 Usage Notes ... 689

Section 21A ROM ... 691
21A.1 Overview ... 691
21A.1.1 Block Diagram ... 691
21A.1.2 Register Configuration ... 691
21A.2 Register Descriptions ... 692
21A.2.1 Mode Control Register (MDCR) ... 692
21A.3 Operation ... 692
21A.4 Flash Memory Overview ... 695
21A.5 Overview ... 696
21A.5.1 Block Diagram ... 696
21A.5.2 Mode Transitions ... 697
21A.5.3 On-Board Programming Modes ... 698
21A.5.4 Flash Memory Emulation in RAM ... 700
21A.5.5 Differences between Boot Mode and User Program Mode ... 701
21A.5.6 Block Configuration ... 702
21A.6 Pin Configuration ... 703
21A.7 Register Configuration ... 704
21A.8 Register Descriptions ... 704
21A.8.1 Flash Memory Control Register 1 (FLMCR1) ... 704
21A.8.2 Flash Memory Control Register 2 (FLMCR2) ... 707
21A.8.3 Erase Block Register 1 (EBR1) ... 708
21A.8.4 Erase Block Register 2 (EBR2) ... 708
21A.8.5 RAM Emulation Register (RAMER) ... 709

21A.8.6 Flash Memory Power Control Register (FLPWCR) 710
21A.9 On-Board Programming Modes 711
21A.9.1 Boot Mode 711
21A.9.2 User Program Mode 716
21A.10 Flash Memory Programming/Erasing 718
21A.10.1 Program Mode 720
21A.10.2 Program-Verify Mode 721
21A.10.3 Erase Mode 725
21A.10.4 Erase-Verify Mode 725
21A.11 Protection 727
21A.11.1 Hardware Protection 727
21A.11.2 Software Protection 728
21A.11.3 Error Protection 729
21A.12 Flash Memory Emulation in RAM 731
21A.13 Interrupt Handling when Programming/Erasing Flash Memory 733
21A.14 Flash Memory Programmer Mode 734
21A.14.1 Socket Adapter and Memory Map 734
21A.14.2 Programmer Mode Operation 735
21A.14.3 Memory Read Mode 736
21A.14.4 Auto-Program Mode 740
21A.14.5 Auto-Erase Mode 742
21A.14.6 Status Read Mode 744
21A.14.7 Status Polling 745
21A.14.8 Programmer Mode Transition Time 745
21A.14.9 Notes on Memory Programming 746
21A.15 Flash Memory and Power-Down States 747
21A.15.1 Notes on Power-Down States 747
21A.16 Flash Memory Programming and Erasing Precautions 748
21A.17 Note on Switching from F-ZTAT Version to Mask ROM Version 753

Section 21B ROM 755
21B.1 Overview 755
21B.1.1 Block Diagram 755
21B.1.2 Register Configuration 755
21B.2 Register Descriptions 756
21B.2.1 Mode Control Register (MDCR) 756
21B.3 Operation 756
21B.4 Features 759
21B.5 Overview 760
21B.5.1 Block Diagram 760
21B.5.2 Mode Transitions 761
21B.5.3 On-Board Programming Modes 762
21B.5.4 Flash Memory Emulation in RAM 764

21B.5.5 Differences between Boot Mode and User Program Mode 765
21B.5.6 Block Configuration 766
21B.6 Pin Configuration 766
21B.7 Register Configuration 767
21B.8 Register Descriptions 767
21B.8.1 Flash Memory Control Register 1 (FLMCR1) 767
21B.8.2 Flash Memory Control Register 2 (FLMCR2) 770
21B.8.3 Erase Block Register 1 (EBR1) 771
21B.8.4 Erase Block Register 2 (EBR2) 771
21B.8.5 RAM Emulation Register (RAMER) 772
21B.8.6 Flash Memory Power Control Register (FLPWCR) 773
21B.9 On-Board Programming Modes 774
21B.9.1 Boot Mode 774
21B.9.2 User Program Mode 779
21B.10 Programming/Erasing Flash Memory 781
21B.10.1 Program Mode 783
21B.10.2 Program-Verify Mode 784
21B.10.3 Erase Mode 788
21B.10.4 Erase-Verify Mode 788
21B.11 Protection 790
21B.11.1 Hardware Protection 790
21B.11.2 Software Protection 791
21B.11.3 Error Protection 792
21B.12 Flash Memory Emulation in RAM 794
21B.13 Interrupt Handling when Programming/Erasing Flash Memory 796
21B.14 Flash Memory Programmer Mode 797
21B.14.1 Socket Adapter and Memory Map 797
21B.14.2 Programmer Mode Operation 798
21B.14.3 Memory Read Mode 799
21B.14.4 Auto-Program Mode 803
21B.14.5 Auto-Erase Mode 805
21B.14.6 Status Read Mode 807
21B.14.7 Status Polling 808
21B.14.8 Programmer Mode Transition Time 808
21B.14.9 Notes on Memory Programming 809
21B.15 Flash Memory and Power-Down States 810
21B.15.1 Notes on Power-Down States 810
21B.16 Flash Memory Programming and Erasing Precautions 811
21B.17 Note on Switching from F-ZTAT Version to Mask ROM Version 816

Section 22A Clock Pulse Generator (H8S/2636, H8S/2638) 817
22A.1 Overview 817
22A.1.1 Block Diagram 817

22A.1.2 Register Configuration 818
22A.2 Register Descriptions 818
22A.2.1 System Clock Control Register (SCKCR) 818
22A.2.2 Low-Power Control Register (LPWRCR) 819
22A.3 Oscillator 820
22A.3.1 Connecting a Crystal Resonator 820
22A.3.2 External Clock Input 823
22A.4 PLL Circuit 825
22A.5 Medium-Speed Clock Divider 826
22A.6 Bus Master Clock Selection Circuit 826
22A.7 Subclock Oscillator 826
22A.8 Subclock Waveform Generation Circuit 827
22A.9 Note on Crystal Resonator 827

Section 22B Clock Pulse Generator (H8S/2639 Series) 829
22B.1 Overview 829
22B.1.1 Block Diagram 829
22B.1.2 Register Configuration 830
22B.2 Register Descriptions 830
22B.2.1 System Clock Control Register (SCKCR) 830
22B.2.2 Low-Power Control Register (LPWRCR) 831
22B.3 Oscillator 832
22B.3.1 Connecting a Crystal Resonator 832
22B.3.2 External Clock Input 835
22B.4 PLL Circuit 837
22B.5 Medium-Speed Clock Divider 838
22B.6 Bus Master Clock Selection Circuit 838
22B.7 Subclock Divider 838
22B.8 Note on Crystal Resonator 838

Section 23A Power-Down Modes [HD64F2636F, HD64F2638F, HD6432636F, HD6432638F] 839
23A.1 Overview 839
23A.1.1 Register Configuration 843
23A.2 Register Descriptions 844
23A.2.1 Standby Control Register (SBYCR) 844
23A.2.2 System Clock Control Register (SCKCR) 845
23A.2.3 Low-Power Control Register (LPWRCR) 847
23A.2.4 Timer Control/Status Register (TCSR) 847
23A.2.5 Module Stop Control Register (MSTPCR) 848
23A.3 Medium-Speed Mode 850
23A.4 Sleep Mode 851
23A.4.1 Sleep Mode 851

23A.4.2 Exiting Sleep Mode 851
23A.5 Module Stop Mode 852
23A.5.1 Module Stop Mode 852
23A.5.2 Usage Notes 854
23A.6 Software Standby Mode 854
23A.6.1 Software Standby Mode 854
23A.6.2 Clearing Software Standby Mode 854
23A.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode 855
23A.6.4 Software Standby Mode Application Example 856
23A.6.5 Usage Notes 857
23A.7 Hardware Standby Mode 857
23A.7.1 Hardware Standby Mode 857
23A.7.2 Hardware Standby Mode Timing 858
23A.8 ø Clock Output Disabling Function 858

Section 23B Power-Down Modes [HD64F2636UF, HD6432636UF, HD64F2638UF, HD6432638UF, HD64F2638WF, HD6432638WF, HD64F2639UF, HD6432639UF, HD64F2639WF, HD6432639WF] 859
23B.1 Overview 859
23B.1.1 Register Configuration 864
23B.2 Register Descriptions 865
23B.2.1 Standby Control Register (SBYCR) 865
23B.2.2 System Clock Control Register (SCKCR) 867
23B.2.3 Low-Power Control Register (LPWRCR) 868
23B.2.4 Timer Control/Status Register (TCSR) 871
23B.2.5 Module Stop Control Register (MSTPCR) 872
23B.3 Medium-Speed Mode 873
23B.4 Sleep Mode 874
23B.4.1 Sleep Mode 874
23B.4.2 Exiting Sleep Mode 874
23B.5 Module Stop Mode 875
23B.5.1 Module Stop Mode 875
23B.5.2 Usage Notes 877
23B.6 Software Standby Mode 877
23B.6.1 Software Standby Mode 877
23B.6.2 Clearing Software Standby Mode 877
23B.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode 878
23B.6.4 Software Standby Mode Application Example 879
23B.6.5 Usage Notes 880
23B.7 Hardware Standby Mode 880

23B.7.1 Hardware Standby Mode 880
23B.7.2 Hardware Standby Mode Timing 881
23B.8 Watch Mode (U-Mask, W-Mask Version Only) 881
23B.8.1 Watch Mode 881
23B.8.2 Exiting Watch Mode 882
23B.8.3 Notes 882
23B.9 Sub-Sleep Mode (U-Mask, W-Mask Version Only) 883
23B.9.1 Sub-Sleep Mode 883
23B.9.2 Exiting Sub-Sleep Mode 883
23B.10 Sub-Active Mode (U-Mask, W-Mask Version Only) 884
23B.10.1 Sub-Active Mode 884
23B.10.2 Exiting Sub-Active Mode 884
23B.11 Direct Transitions (U-Mask, W-Mask Version Only) 885
23B.11.1 Overview of Direct Transitions 885
23B.12 ø Clock Output Disabling Function 886
23B.13 Usage Notes 887

Section 24 Electrical Characteristics 889
24.1 H8S/2636 Series Electrical Characteristics 889
24.1.1 Absolute Maximum Ratings 889
24.1.2 Power Supply Voltage and Operating Frequency Range 890
24.1.3 DC Characteristics 891
24.1.4 AC Characteristics 895
24.1.5 A/D Conversion Characteristics 901
24.1.6 D/A Conversion Characteristics 902
24.1.7 Flash Memory Characteristics 903
24.2 H8S/2638 Series Electrical Characteristics 905
24.2.1 Absolute Maximum Ratings 905
24.2.2 Power Supply Voltage and Operating Frequency Range 906
24.2.3 DC Characteristics 907
24.2.4 AC Characteristics 912
24.2.5 A/D Conversion Characteristics 919
24.2.6 D/A Conversion Characteristics 920
24.2.7 Flash Memory Characteristics 921
24.3 H8S/2639 Series Electrical Characteristics 923
24.3.1 Absolute Maximum Ratings 923
24.3.2 Power Supply Voltage and Operating Frequency Range 924
24.3.3 DC Characteristics 925
24.3.4 AC Characteristics 930
24.3.5 A/D Conversion Characteristics 937
24.3.6 D/A Conversion Characteristics 938
24.3.7 Flash Memory Characteristics 939
24.4 Operation Timing 941

24.4.1 Clock Timing 941
24.4.2 Control Signal Timing 942
24.4.3 Bus Timing 943
24.4.4 On-Chip Supporting Module Timing 946
24.5 Usage Note 950

Appendix A Instruction Set 951
A.1 Instruction List 951
A.2 Instruction Codes 975
A.3 Operation Code Map 990
A.4 Number of States Required for Instruction Execution 994
A.5 Bus States During Instruction Execution 1008
A.6 Condition Code Modification 1022

Appendix B Internal I/O Register 1028
B.1 Address 1028
B.2 Functions 1050

Appendix C I/O Port Block Diagrams 1313
C.1 Port 1 Block Diagrams 1313
C.2 Port 3 Block Diagrams 1319
C.3 Port 4 Block Diagram 1325
C.4 Port 9 Block Diagram 1326
C.5 Port A Block Diagram 1327
C.6 Port B Block Diagram 1331
C.7 Port C Block Diagram 1332
C.8 Port D Block Diagram 1333
C.9 Port E Block Diagram 1334
C.10 Port F Block Diagrams 1335
C.11 Port G Block Diagram 1341
C.12 Port J Block Diagram 1342

Appendix D Pin States 1343
D.1 Port States in Each Mode 1343

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode 1346

Appendix F Product Code Lineup 1347

Appendix G Package Dimensions 1348

Section 1 Overview

1.1 Overview

The H8S/2636 and H8S/2638 and H8S/2639 are microcomputers (MCUs: microcomputer units), built around the H8S/2600 CPU, employing Hitachi's proprietary architecture, and equipped with peripheral functions on-chip.

The H8S/2600 CPU has an internal 32-bit architecture, is provided with sixteen 16-bit general registers and a concise, optimized instruction set designed for high-speed operation, and can address a 16-Mbyte linear address space. The instruction set is upward-compatible with H8/300 and H8/300H CPU instructions at the object-code level, facilitating migration from the H8/300, H8/300L, or H8/300H Series.

On-chip peripheral functions required for system configuration include data transfer controller (DTC) bus masters, ROM and RAM memory, a16-bit timer-pulse unit (TPU), programmable pulse generator (PPG), motor control PWM timer (PWM) watchdog timer (WDT), serial communication interface (SCI), A/D converter, D/A converter, Hitachi controller area network (HCAN) and I/O ports. An I^2C bus interface (IIC) is available as an option in the H8S/2638 and H8S/2639.

On-chip ROM is available as 128-kbyte and 256-kbyte flash memory (F-ZTAT™* version), and as 128-kbyte and 256-kbyte mask ROM. ROM is connected to the CPU via a 16-bit data bus, enabling both byte and word data to be accessed in one state. Instruction fetching has been speeded up, and processing speed increased.

Four operating modes, modes 4 to 7, are provided, and there is a choice of single-chip mode or external expansion mode.

Subclock (32 kHz oscillation) functions are available in the U-mask and W-mask versions only. These functions cannot be used with the other versions.

The features of the H8S/2636, H8S/2638 and H8S/2639 are shown in table 1-1.

Note: * F-ZTAT™ is a trademark of Hitachi, Ltd.

Table 1-1 Overview

Item	Specification
CPU	• General-register machine — Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers) • High-speed operation suitable for realtime control — Maximum clock rate: 20 MHz — High-speed arithmetic operations 8/16/32-bit register-register add/subtract : 50 ns 16 × 16-bit register-register multiply : 200 ns 16 × 16 + 42-bit multiply and accumulate : 200 ns 32 ÷ 16-bit register-register divide : 1000 ns • Instruction set suitable for high-speed operation — Sixty-nine basic instructions — 8/16/32-bit move/arithmetic and logic instructions — Unsigned/signed multiply and divide instructions — Multiply-and accumulate instruction — Powerful bit-manipulation instructions • CPU operating modes — Advanced mode: 16-Mbyte address space
Bus controller	• Address space divided into 8 areas, with bus specifications settable independently for each area • Choice of 8-bit or 16-bit access space for each area • 2-state or 3-state access space can be designated for each area • Number of program wait states can be set for each area • Direct connection to burst ROM supported
PC break controller	• Supports debugging functions by means of PC break interrupts • Two break channels
Data transfer controller (DTC)	• Can be activated by internal interrupt or software • Multiple transfers or multiple types of transfer possible for one activation source • Transfer possible in repeat mode, block transfer mode, etc. • Request can be sent to CPU for interrupt that activated DTC
16-bit timer-pulse unit (TPU)	• 6-channel 16-bit timer on-chip • Pulse I/O processing capability for up to 16 pins' • Automatic 2-phase encoder count capability
Programmable pulse generator (PPG)	• Maximum 8-bit pulse output possible with TPU as time base • Output trigger selectable in 4-bit groups • Non-overlap margin can be set • Direct output or inverse output setting possible

Item	Specification
Watchdog timer (WDT) 2 channels	• Watchdog timer or interval timer selectable • Operation using sub-clock supported (WDT1 only)*
Motor control PWM timer (PWM)	• Maximum of 16 10-bit PWM outputs • Eight outpus with two channels each built in • Duty settable between 0% and 100% • Automatic transfer of buffer register data supported • Settable to any one of 5 operating speeds
Serial communication interface (SCI) 3 channels (SCI0 to SCI2)	• Asynchronous mode or synchronous mode selectable • Multiprocessor communication function • Smart card interface function
Hitachi controller area network (HCAN) 2 channels	• CAN: Ver. 2.0B compliant • Buffer size: 15 transmit/receive messages, transmit only one message • Filtering of receive messages
A/D converter	• Resolution: 10 bits • Input: 12 channels • High-speed conversion: 13.3 µs minimum conversion time (at 20 MHz operation) • Single or scan mode selectable • Sample and hold circuit • A/D conversion can be activated by external trigger or timer trigger
D/A converter	• Resolution: 8 bits • Output: 2 channels
I/O ports	• 72 I/O pins, 12 input-only pins
Memory	• Flash memory or mask ROM • High-speed static RAM
Interrupt controller	• Seven external interrupt pins (NMI, $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$) • 49 internal interrupt sources • Eight priority levels settable
Power-down states	• Medium-speed mode • Sleep mode • Module-stop mode • Software standby mode • Hardware standby mode • Sub-clock operation* (sub-active mode, sub-sleep mode, watch mode)

Product Name	ROM	RAM
H8S/2636	128 kbytes	4 kbytes
H8S/2638	256 kbytes	16 kbytes
H8S/2639		

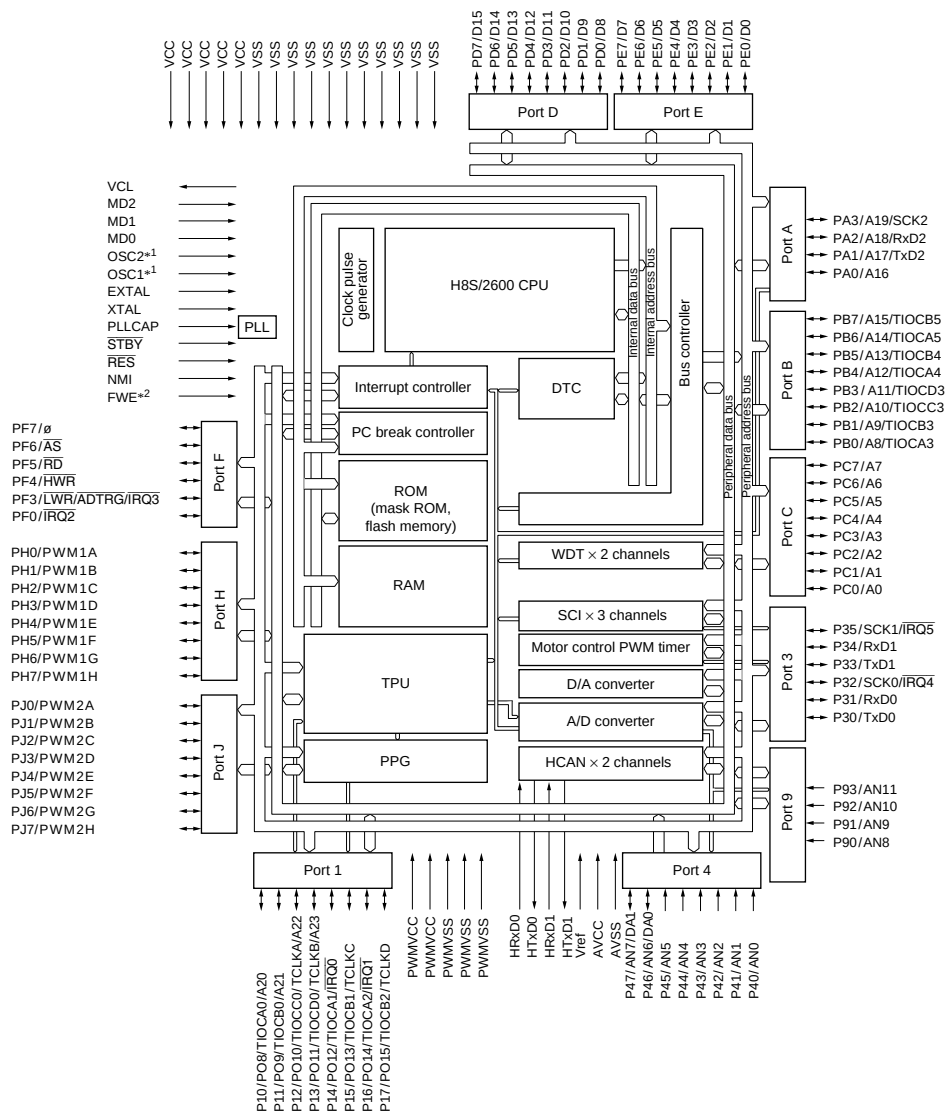
Item	Specification				
Operating modes	Four MCU operating modes				
		CPU		External Data Bus	
	Mode	Operating Mode	Description	On-Chip ROM	Initial Value	Maximum Value
	4	Advanced	On-chip ROM disabled expansion mode	Disabled	16 bits	16 bits
	5		On-chip ROM disabled expansion mode	Disabled	8 bits	16 bits
	6		On-chip ROM enabled expansion mode	Enabled	8 bits	16 bits
	7		Single-chip mode	Enabled	—	—
Clock pulse generator	• On-chip PLL circuit (×1, ×2, ×4) • Input clock frequency: 4 to 20 MHz					
I^2C bus interface (IIC) ×2 channel (Option) (Only for the H8S/2638 and H8S/2639)	• Conforms to the I^2C bus interface type advocated by Philips • Single master mode/slave mode • Possible to determine arbitration lost conditions • Supports two slave addresses					
Packages	• 128-pin plastic QFP (FP-128B)					

Item	Specification					
Product lineup	Model Name		Subclock (32 kHz Oscillation) Functions	I^2C bus interface	ROM/ RAM (Bytes)	Packages
	Mask ROM Version	F-ZTAT Version				
	HD6432636F*	HD64F2636F	No	—	128 k/ 4 k	FP-128B
	HD6432636UF* (U-Mask Version)	HD64F2636UF (U-Mask Version)	Yes	—		
	HD6432638F*	HD64F2638F	No	No	256 k/ 16 k	
	HD6432638UF* (U-Mask Version)	HD64F2638UF (U-Mask Version)	Yes	No		
	HD6432638WF* (W-Mask Version)	HD64F2638WF (W-Mask Version)	Yes	Yes		
	HD6432639UF* (U-Mask Version)	HD64F2639UF (U-Mask Version)	Yes	No		
	HD6432639WF* (W-Mask Version)	HD64F2639WF (W-Mask Version)	Yes	Yes		
	Note: * Under planning					

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are not available in the HD64F2636F and HD64F2638F, but are available in the HD64F2636UF and HD64F2638UF (U-mask version).

1.2 Internal Block Diagram

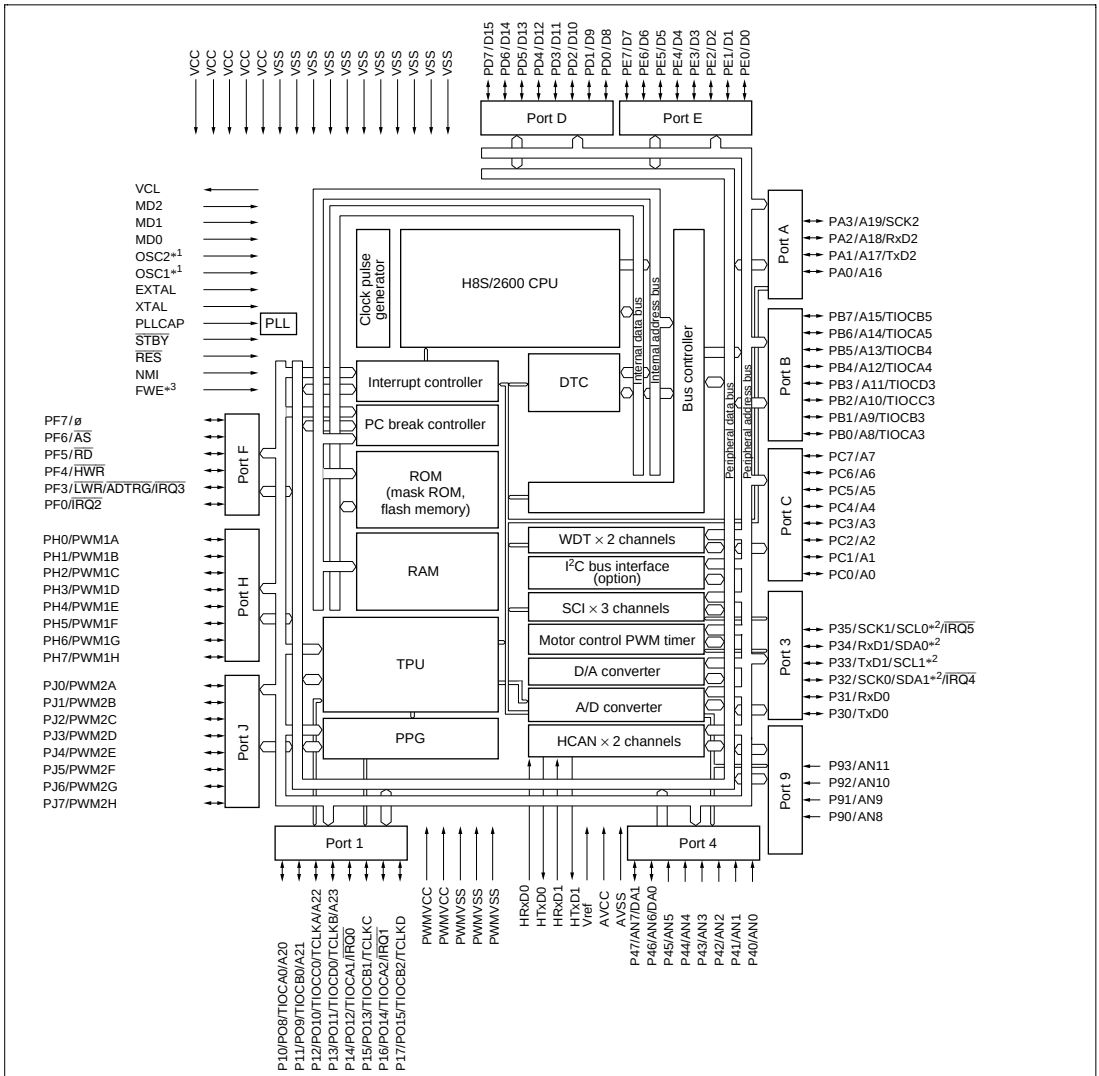
Figure 1-1 (a) shows an internal block diagram of the H8S/2636.



Notes: *1 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask version. These functions cannot be used with the other versions.
See section 22A.7, Subclock Oscillator, for the method of fixing pins OSC1 and OSC2.
*2 The FWE pin only applies to the flash memory version.

Figure 1-1 (a) Internal Block Diagram of H8S/2636

Figure 1-1 (b) shows an internal block diagram of the H8S/2638 and H8S/2639.



Notes: *1 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions. These functions cannot be used with the other versions.
See section 22A.7, Subclock Oscillator, for the method of fixing pins OSC1 and OSC2.
The H8S/2639 has no OSC1 and OSC2 pins.

*2 These pins are used for the I2C bus interface.
The I2C bus interface is available as an option in the H8S/2638 and H8S/2639.

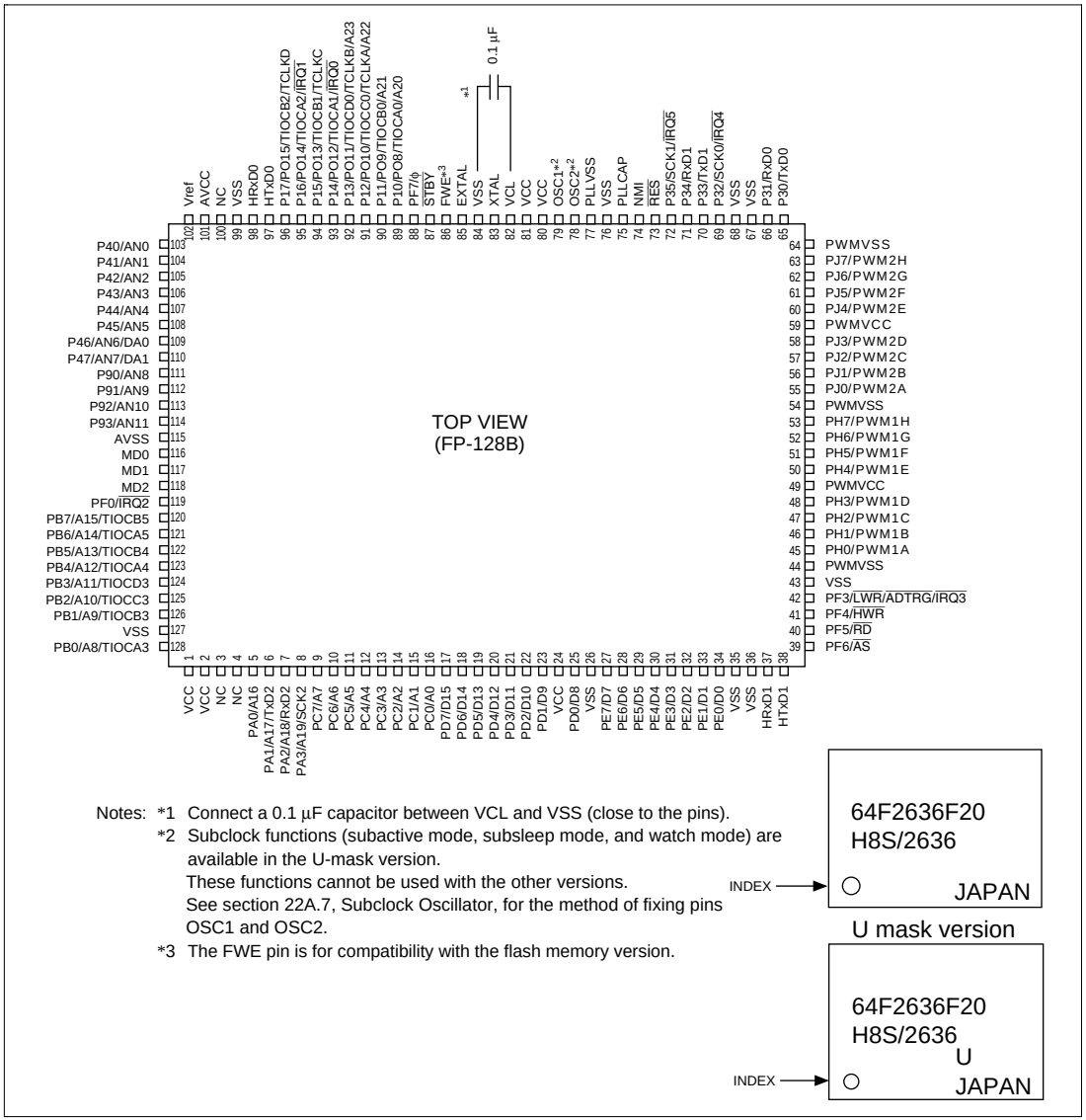
*3 The FWE pin is for compatibility with the flash memory version.

Figure 1-1 (b) Internal Block Diagram of H8S/2638 and H8S/2639

1.3 Pin Description

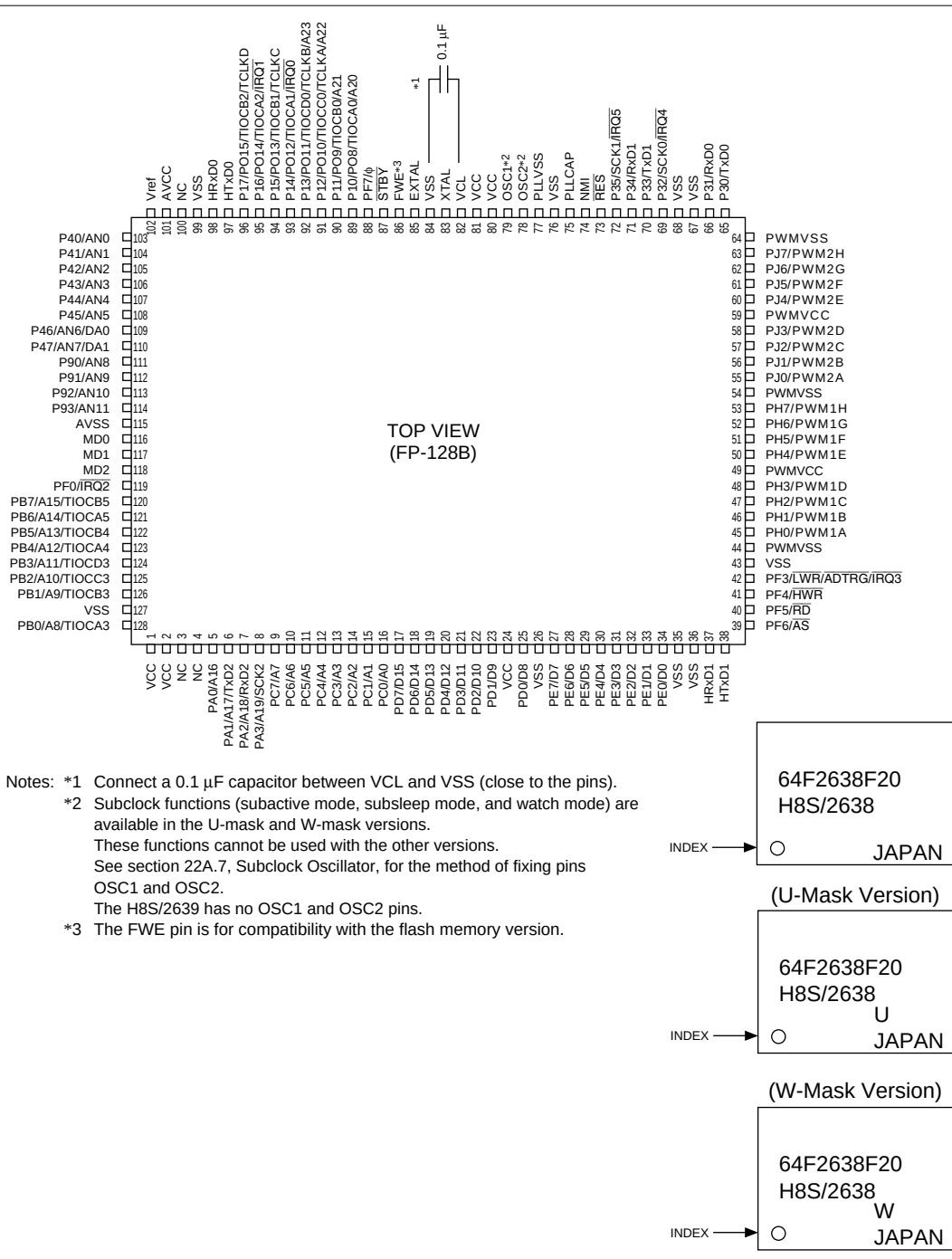
1.3.1 Pin Arrangement

Figure 1-2 shows the pin arrangement of the H8S/2636, Figure 1-3 shows the pin arrangement of the H8S/2638, and Figure 1-4 shows the pin arrangement of the H8S/2639.



Notes: *1 Connect a 0.1 µF capacitor between VCL and VSS (close to the pins).
*2 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask version.
These functions cannot be used with the other versions.
See section 22A.7, Subclock Oscillator, for the method of fixing pins OSC1 and OSC2.
*3 The FWE pin is for compatibility with the flash memory version.

Figure 1-2 Pin Arrangement of HD64F2636F (FP-128B: Top View)



Notes: *1 Connect a 0.1 µF capacitor between VCL and VSS (close to the pins).

*2 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions.
These functions cannot be used with the other versions.
See section 22A.7, Subclock Oscillator, for the method of fixing pins OSC1 and OSC2.
The H8S/2639 has no OSC1 and OSC2 pins.

*3 The FWE pin is for compatibility with the flash memory version.

Figure 1-3 Pin Arrangement of H8S/2638 Series (FP-128B: Top View)

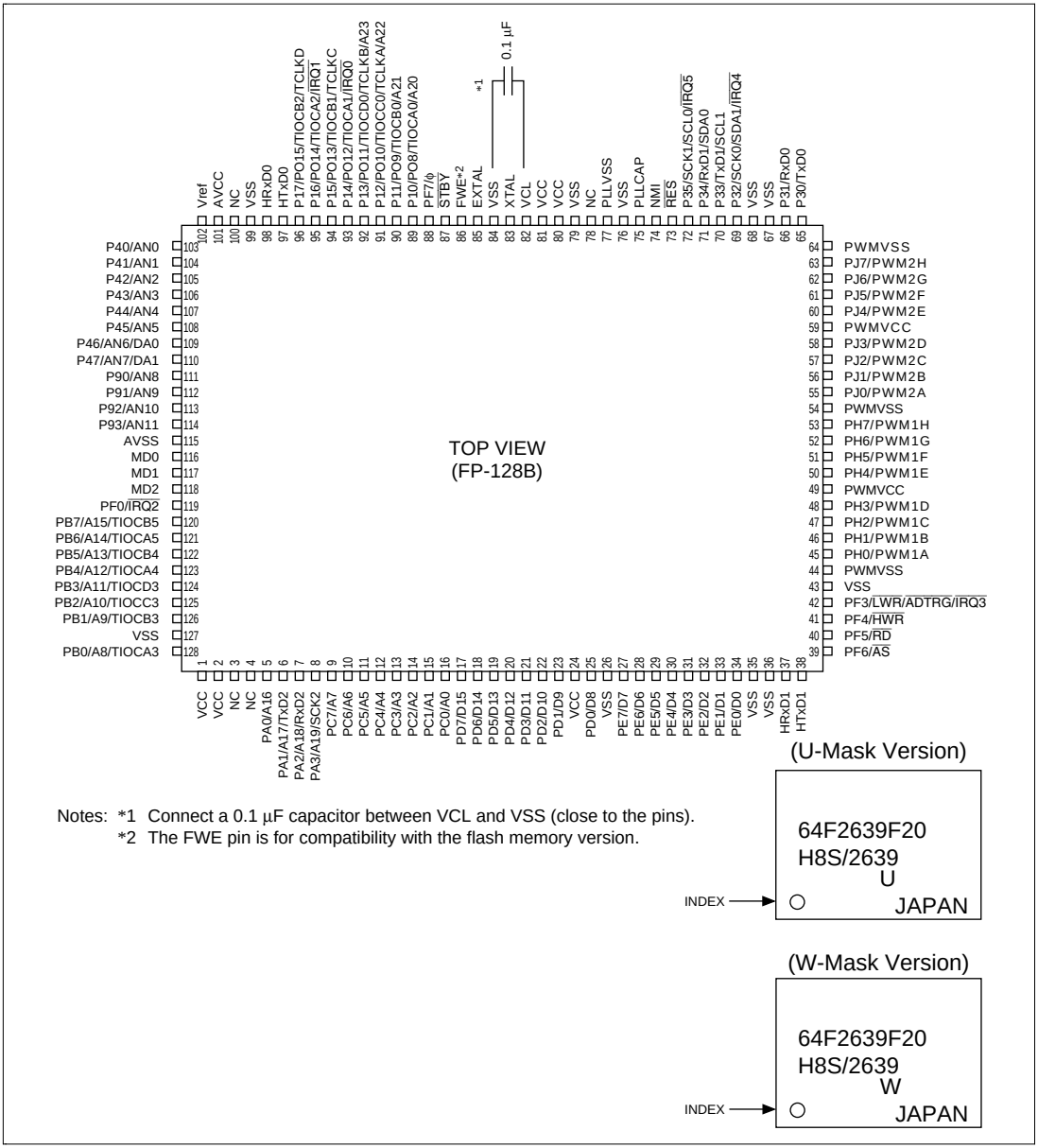


Figure 1-4 Pin Arrangement H8S/2639 Series (FP-128B: Top View)

1.3.2 Pin Functions in Each Operating Mode

Table 1-2 shows the pin functions of the H8S/2636, H8S/2638 in each of the operating modes.

Table 1-2 Pin Functions in Each Operating Mode

Pin No.	Pin Name			
FP-128B	Mode 4	Mode 5	Mode 6	Mode 7
1	VCC	VCC	VCC	VCC
2	VCC	VCC	VCC	VCC
3	NC	NC	NC	NC
4	NC	NC	NC	NC
5	A16	A16	PA0/A16	PA0
6	A17	A17	PA1/A17/TxD2	PA1/TxD2
7	A18	A18	PA2/A18/RxD2	PA2/RxD2
8	A19	A19	PA3/A19/SCK2	PA3/SCK2
9	A7	A7	PC7/A7	PC7
10	A6	A6	PC6/A6	PC6
11	A5	A5	PC5/A5	PC5
12	A4	A4	PC4/A4	PC4
13	A3	A3	PC3/A3	PC3
14	A2	A2	PC2/A2	PC2
15	A1	A1	PC1/A1	PC1
16	A0	A0	PC0/A0	PC0
17	D15	D15	D15	PD7
18	D14	D14	D14	PD6
19	D13	D13	D13	PD5
20	D12	D12	D12	PD4
21	D11	D11	D11	PD3
22	D10	D10	D10	PD2
23	D9	D9	D9	PD1
24	VCC	VCC	VCC	VCC
25	D8	D8	D8	PD0
26	VSS	VSS	VSS	VSS

Pin No.	Pin Name			
FP-128B	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
27	D7	PE7/D7	PE7/D7	PE7
28	D6	PE6/D6	PE6/D6	PE6
29	D5	PE5/D5	PE5/D5	PE5
30	D4	PE4/D4	PE4/D4	PE4
31	D3	PE3/D3	PE3/D3	PE3
32	D2	PE2/D2	PE2/D2	PE2
33	D1	PE1/D1	PE1/D1	PE1
34	D0	PE0/D0	PE0/D0	PE0
35	VSS	VSS	VSS	VSS
36	VSS	VSS	VSS	VSS
37	HRxD1	HRxD1	HRxD1	HRxD1
38	HTxD1	HTxD1	HTxD1	HTxD1
39	$\overline{AS}$	$\overline{AS}$	$\overline{AS}$	PF6
40	$\overline{RD}$	$\overline{RD}$	$\overline{RD}$	PF5
41	$\overline{HWR}$	$\overline{HWR}$	$\overline{HWR}$	PF4
42	$\overline{LWR}$	PF3/ $\overline{LWR}$ /$\overline{ADTRG}$/$\overline{IRQ3}$	PF3/ $\overline{LWR}$ /$\overline{ADTRG}$/$\overline{IRQ3}$	PF3 /$\overline{ADTRG}$/$\overline{IRQ3}$
43	VSS	VSS	VSS	VSS
44	PWMVSS	PWMVSS	PWMVSS	PWMVSS
45	PH0/PWM1A	PH0/PWM1A	PH0/PWM1A	PH0/PWM1A
46	PH1/PWM1B	PH1/PWM1B	PH1/PWM1B	PH1/PWM1B
47	PH2/PWM1C	PH2/PWM1C	PH2/PWM1C	PH2/PWM1C
48	PH3/PWM1D	PH3/PWM1D	PH3/PWM1D	PH3/PWM1D
49	PWMVCC	PWMVCC	PWMVCC	PWMVCC
50	PH4/PWM1E	PH4/PWM1E	PH4/PWM1E	PH4/PWM1E
51	PH5/PWM1F	PH5/PWM1F	PH5/PWM1F	PH5/PWM1F
52	PH6/PWM1G	PH6/PWM1G	PH6/PWM1G	PH6/PWM1G
53	PH7/PWM1H	PH7/PWM1H	PH7/PWM1H	PH7/PWM1H
54	PWMVSS	PWMVSS	PWMVSS	PWMVSS
55	PJ0/PWM2A	PJ0/PWM2A	PJ0/PWM2A	PJ0/PWM2A
56	PJ1/PWM2B	PJ1/PWM2B	PJ1/PWM2B	PJ1/PWM2B

Pin No.	Pin Name			
FP-128B	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
57	PJ2/PWM2C	PJ2/PWM2C	PJ2/PWM2C	PJ2/PWM2C
58	PJ3/PWM2D	PJ3/PWM2D	PJ3/PWM2D	PJ3/PWM2D
59	PWMVCC	PWMVCC	PWMVCC	PWMVCC
60	PJ4/PWM2E	PJ4/PWM2E	PJ4/PWM2E	PJ4/PWM2E
61	PJ5/PWM2F	PJ5/PWM2F	PJ5/PWM2F	PJ5/PWM2F
62	PJ6/PWM2G	PJ6/PWM2G	PJ6/PWM2G	PJ6/PWM2G
63	PJ7/PWM2H	PJ7/PWM2H	PJ7/PWM2H	PJ7/PWM2H
64	PWMVSS	PWMVSS	PWMVSS	PWMVSS
65	P30/TxD0	P30/TxD0	P30/TxD0	P30/TxD0
66	P31/RxD0	P31/RxD0	P31/RxD0	P31/RxD0
67	VSS	VSS	VSS	VSS
68	VSS	VSS	VSS	VSS
69	P32/SCK0/SDA1*[2]/ $\overline{\text{IRQ4}}$	P32/SCK0/SDA1*[2]/ $\overline{\text{IRQ4}}$	P32/SCK0/SDA1*[2]/ $\overline{\text{IRQ4}}$	P32/SCK0/SDA1*[2]/ $\overline{\text{IRQ4}}$
70	P33/TxD1/SCL1*[2]	P33/TxD1/SCL1*[2]	P33/TxD1/SCL1*[2]	P33/TxD1/SCL1*[2]
71	P34/RxD1/SDA0*[2]	P34/RxD1/SDA0*[2]	P34/RxD1/SDA0*[2]	P34/RxD1/SDA0*[2]
72	P35/SCK1/SCL0*[2]/ $\overline{\text{IRQ5}}$	P35/SCK1/SCL0*[2]/ $\overline{\text{IRQ5}}$	P35/SCK1/SCL0*[2]/ $\overline{\text{IRQ5}}$	P35/SCK1/SCL0*[2]/ $\overline{\text{IRQ5}}$
73	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$	$\overline{\text{RES}}$
74	NMI	NMI	NMI	NMI
75	PLLCAP	PLLCAP	PLLCAP	PLLCAP
76	VSS	VSS	VSS	VSS
77	PLLVSS	PLLVSS	PLLVSS	PLLVSS
78	OSC2*[1]	OSC2*[1]	OSC2*[1]	OSC2*[1]
79	OSC1*[1]	OSC1*[1]	OSC1*[1]	OSC1*[1]
80	VCC	VCC	VCC	VCC
81	VCC	VCC	VCC	VCC
82	VCL	VCL	VCL	VCL
83	XTAL	XTAL	XTAL	XTAL
84	VSS	VSS	VSS	VSS
85	EXTAL	EXTAL	EXTAL	EXTAL
86	FWE	FWE	FWE	FWE

Pin No.	Pin Name			
FP-128B	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
87	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$	$\overline{\text{STBY}}$
88	PF7/ø	PF7/ø	PF7/ø	PF7/ø
89	A20	A20	P10/PO8/TIOCA0/A20	P10/PO8/TIOCA0
90	P11/PO9/TIOCB0/A21	P11/PO9/TIOCB0/A21	P11/PO9/TIOCB0/A21	P11/PO9/TIOCB0
91	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA/A22	P12/PO10/TIOCC0/ TCLKA
92	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB/A23	P13/PO11/TIOCD0/ TCLKB
93	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$	P14/PO12/TIOCA1/ $\overline{\text{IRQ0}}$
94	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC	P15/PO13/TIOCB1/ TCLKC
95	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$	P16/PO14/TIOCA2/ $\overline{\text{IRQ1}}$
96	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD	P17/PO15/TIOCB2/ TCLKD
97	HTxD0	HTxD0	HTxD0	HTxD0
98	HRxD0	HRxD0	HRxD0	HRxD0
99	VSS	VSS	VSS	VSS
100	NC	NC	NC	NC
101	AVCC	AVCC	AVCC	AVCC
102	Vref	Vref	Vref	Vref
103	P40/AN0	P40/AN0	P40/AN0	P40/AN0
104	P41/AN1	P41/AN1	P41/AN1	P41/AN1
105	P42/AN2	P42/AN2	P42/AN2	P42/AN2
106	P43/AN3	P43/AN3	P43/AN3	P43/AN3
107	P44/AN4	P44/AN4	P44/AN4	P44/AN4
108	P45/AN5	P45/AN5	P45/AN5	P45/AN5
109	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0	P46/AN6/DA0
110	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1	P47/AN7/DA1
111	P90/AN8	P90/AN8	P90/AN8	P90/AN8
112	P91/AN9	P91/AN9	P91/AN9	P91/AN9

Pin No.	Pin Name			
FP-128B	**Mode 4**	**Mode 5**	**Mode 6**	**Mode 7**
113	P92/AN10	P92/AN10	P92/AN10	P92/AN10
114	P93/AN11	P93/AN11	P93/AN11	P93/AN11
115	AVSS	AVSS	AVSS	AVSS
116	MD0	MD0	MD0	MD0
117	MD1	MD1	MD1	MD1
118	MD2	MD2	MD2	MD2
119	PF0/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{IRQ2}}$	PF0/$\overline{\text{IRQ2}}$
120	A15	A15	PB7/A15/TIOCB5	PB7/TIOCB5
121	A14	A14	PB6/A14/TIOCA5	PB6/TIOCA5
122	A13	A13	PB5/A13/TIOCB4	PB5/TIOCB4
123	A12	A12	PB4/A12/TIOCA4	PB4/TIOCA4
124	A11	A11	PB3/A11/TIOCD3	PB3/TIOCD3
125	A10	A10	PB2/A10/TIOCC3	PB2/TIOCC3
126	A9	A9	PB1/A9/TIOCB3	PB1/TIOCB3
127	VSS	VSS	VSS	VSS
128	A8	A8	PB0/A8/TIOCA3	PB0/TIOCA3

Notes: NC pins should be connected to VSS or left open.

*1 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions.

These functions cannot be used with the other versions.

See section 22A.7, Subclock Oscillator, for the method of fixing pins OSC1 and OSC2.

The H8S/2639 has no OSC1 and OSC2 pins.

*2 These pins are used for the I^2C bus interface.

The I^2C bus interface (IIC) is available as an option in the H8S/2638 and H8S/2639.

1.3.3 Pin Functions

Table 1-3 outlines the pin functions of the H8S/2636.

Table 1-3 Pin Functions

Type	Symbol	I/O	Name and Function
Power	VCC	Input	Power supply: For connection to the power supply. All VCC pins should be connected to the system power supply.
	VSS	Input	Ground: For connection to ground (0 V). All VSS pins should be connected to the system power supply (0 V).
	VCL	Output	On-chip step-down power supply pin: The VCL pin need not be connected to the power supply. Connect this pin to VSS via a 0.1 µF capacitor (placed close to the pins).
Clock	PLLVSS	Input	PLL ground: Ground for on-chip PLL oscillator.
	PLLCAP	Input	PLL capacitance: External capacitance pin for on-chip PLL oscillator.
	XTAL	Input	Connects to a crystal oscillator. See section 22AB, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator.
	EXTAL	Input	Connects to a crystal oscillator. See section 22AB, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator.
	OSC1*[1]	Input	Subclock: Connects to a 32.768 kHz crystal oscillator. See section 22A, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator.*
	OSC2*[1]	Input	Subclock: Connects to a 32.768 kHz crystal oscillator. See section 22A, Clock Pulse Generator, for typical connection diagrams for a crystal oscillator.*
	ø	Output	System clock: Supplies the system clock to an external device.
HCAN	HTxD0, HTxD1	Output	HCAN transmit data. Pin for CAN bus transmission.
	HRxD0, HRxD1	Input	HCAN receive data. Pin for CAN bus reception.

Type	Symbol	I/O	Name and Function
Operating mode control	MD2 to MD0	Input	Mode pins: These pins set the operating mode. The relation between the settings of pins MD2 to MD0 and the operating mode is shown below. These pins should not be changed while the H8S/2636 is operating.

MD2	MD1	MD0	Operating Mode
0	0	0	—
		1	—
	1	0	—
		1	—
1	0	0	Mode 4
		1	Mode 5
	1	0	Mode 6
		1	Mode 7

System control	$\overline{\text{RES}}$	Input	Reset input: When this pin is driven low, the chip is reset.
	$\overline{\text{STBY}}$	Input	Standby: When this pin is driven low, a transition is made to hardware standby mode.
	FWE*[2]	Input	Flash write enable: Pin for flash memory use (in planning stage).
Interrupts	NMI	Input	Nonmaskable interrupt: Requests a nonmaskable interrupt. When this pin is not used, it should be fixed high.
	$\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$	Input	Interrupt request 5 to 0: These pins request a maskable interrupt.
Address bus	A23 to A0	Output	Address bus: These pins output an address.
Data bus	D15 to D0	I/O	Data bus: These pins constitute a bidirectional data bus.

Type	Symbol	I/O	Name and Function
Bus control	$\overline{AS}$	Output	Address strobe: When this pin is low, it indicates that address output on the address bus is enabled.
	$\overline{RD}$	Output	Read: When this pin is low, it indicates that the external address space can be read.
	$\overline{HWR}$	Output	High write: A strobe signal that writes to external space and indicates that the upper half (D15 to D8) of the data bus is enabled.
	$\overline{LWR}$	Output	Low write: A strobe signal that writes to external space and indicates that the lower half (D7 to D0) of the data bus is enabled.
16-bit timer-pulse unit (TPU)	TCLKD to TCLKA	Input	Clock input D to A: These pins input an external clock.
	TIOCA0, TIOCB0, TIOCC0, TIOCD0	I/O	Input capture/ output compare match A0 to D0: The TGR0A to TGR0D input capture input or output compare output, or PWM output pins.
	TIOCA1, TIOCB1	I/O	Input capture/ output compare match A1 and B1: The TGR1A and TGR1B input capture input or output compare output, or PWM output pins.
	TIOCA2, TIOCB2	I/O	Input capture/ output compare match A2 and B2: The TGR2A and TGR2B input capture input or output compare output, or PWM output pins.
	TIOCA3, TIOCB3, TIOCC3, TIOCD3	I/O	Input capture/ output compare match A3 to D3: The TGR3A to TGR3D input capture input or output compare output, or PWM output pins.
	TIOCA4, TIOCB4	I/O	Input capture/output compare match A4 and B4: The TGR4A and TGR4B input capture input or output compare output, or PWM output pins.
	TIOCA5, TIOCB5	I/O	Input capture/output compare match A5 and B5: The TGR5A and TGR5B input capture input or output compare output, or PWM output pins.
Programmable pulse generator (PPG)	PO15 to PO8	Output	Pulse output 15 to 8: Pulse output pins.

Type	Symbol	I/O	Name and Function
Serial communication interface (SCI)/ Smart Card interface	TxD2, TxD1, TxD0	Output	Transmit data (channel 0, 1, 2): Data output pins.
	RxD2, RxD1, RxD0	Input	Receive data (channel 0, 1, 2): Data input pins.
	SCK2, SCK1, SCK0	I/O	Serial clock (channel 0, 1, 2): Clock I/O pins. The SCK0 output type is NMOS push-pull.
A/D converter	AN11 to AN0	Input	Analog 11 to 0: Analog input pins.
	$\overline{\text{ADTRG}}$	Input	A/D conversion external trigger input: Pin for input of an external trigger to start A/D conversion.
D/A converter	DA1, DA0	Output	Analog output: Analog output pins for D/A converter.
A/D converter, D/A converter	AVCC	Input	A/D converter and D/A converter power supply pin When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+5 V).
	AVSS	Input	Ground pin for A/D converter and D/A converter Connect to system power supply (0 V).
	Vref	Input	A/D converter and D/A converter reference voltage input pin When the A/D converter and D/A converter are not used, this pin should be connected to the system power supply (+5 V).
I/O ports	P17 to P10	I/O	Port 1: An 8-bit I/O port. Input or output can be designated for each bit by means of the port 1 data direction register (P1DDR).
	P35 to P30	I/O	Port 3: A 6-bit I/O port. Input or output can be designated for each bit by means of the port 3 data direction register (P3DDR).
	P47 to P40	Input	Port 4: An 8-bit input port.
	P93 to P90	Input	Port 9: A 4-bit input port.
	PA3 to PA0	I/O	Port A: A 4-bit I/O port. Input or output can be designated for each bit by means of the port A data direction register (PADDR).

Type	Symbol	I/O	Name and Function
I/O ports	PB7 to PB0	I/O	Port B: An 8-bit I/O port. Input or output can be designated for each bit by means of the port B data direction register (PBDDR).
	PC7 to PC0	I/O	Port C: An 8-bit I/O port. Input or output can be designated for each bit by means of the port C data direction register (PCDDR).
	PD7 to PD0	I/O	Port D: An 8-bit I/O port. Input or output can be designated for each bit by means of the port D data direction register (PDDDR).
	PE7 to PE0	I/O	Port E: An 8-bit I/O port. Input or output can be designated for each bit by means of the port E data direction register (PEDDR).
	PF7 to PF3, PF0	I/O	Port F: A 6-bit I/O port. Input or output can be designated for each bit by means of the port F data direction register (PFDDR).
	PH7 to PH0	I/O	Port H: An 8-bit I/O port. Input or output can be designated for each bit by means of the port B data direction register (PHDDR).
	PJ7 to PJ0	I/O	Port J: An 8-bit I/O port. Input or output can be designated for each bit by means of the port J data direction register (PJDDR).
Motor control PWM	PWM1A to PWM1H	Output	PWM output: Motor control PWM channel 1 output pins
	PWM2A to PWM2H	Output	PWM output: Motor control PWM channel 2 output pins
	PWMVCC	Input	PWM Power Supply: Power supply pin for motor-control PWM. Connect to the system power supply (+5 V) when the motor-control function is not used.
	PWMVSS	Input	PWM Ground: Ground pin for motor-control PWM. Connect to the system power supply (0 V).
I^2C bus interface (IIC) (Option)	SCL0, SCL1	I/O	I^2C clock input/output (Channel 0/1): I^2C clock input/output pins that have bus-driving capability. The output of SCL0 is an NMOS open-drain type.
(Only for the H8S/2638)	SDA0, SDA1	I/O	I^2C data input/output (Channel 0/1): I^2C data input/output pins that have bus-driving capability. The output of SDA0 is an NMOS open-drain type.

Note: *1 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions.

These functions cannot be used with the other versions.

See section 22A.7, Subclock Oscillator, for the method of fixing pins OSC1 and OSC2.

The H8S/2639 has no OSC1 and OSC2 pins.

*2 The FWE pin is functional only in the flash memory version. In the mask ROM version, the FWE pin must be left open.

1.4 Differences between H8S/2636, H8S/2638, and H8S/2639

There are four versions of the H8S/2636, including ROM and U-mask options; there are six four versions of the H8S/2638, including ROM, U-mask, and W-mask options; and there are four versions of the H8S/2639, including ROM, U-mask, and W-mask options.

The specifications of these products are compared in Table 1-4 below.

Table 1.4 Comparison of Product Specifications

Product type	H8S/2636				H8S/2638					
	H8S/2636		H8S/2636 U-Mask Version		H8S/2638		H8S/2638 U-Mask version		H8S/2638 W-Mask Version	
Model	HD64F 2636F	HD643 2636F*	HD64F 2636UF	HD6432 636UF*	HD64F 2638F	HD6432 638F*	HD64F 2638UF	HD6432 638UF*	HD64F 2638WF	HD6432 638WF*
Product specifications	128 kB on-chip flash memory	128 kB mask ROM	On-chip flash memory Subclock function	128 kB mask ROM	256 kB on-chip flash memory	256 kB mask ROM	256 kB on-chip flash memory Subclock function	256 kB mask ROM	256 kB on-chip flash memory Subclock function I²C bus interface	256 kB mask ROM
RAM size	4 kbytes				16 kbytes					
ROM size	128 kbytes				256 kbytes					
Flash memory	See section 21A, ROM.				See section 21B, ROM.					
I²C bus interface	—								See section 15, I²C Bus Interface	
Power-down modes	See section 23A, Power-Down Modes				See section 23B, Power-Down Modes					
Electrical characteristics	See section 24, Electrical Characteristics				See section 24, Electrical Characteristics					

Note: * Under planning

Section 2 CPU

2.1 Overview

The H8S/2600 CPU is a high-speed central processing unit with an internal 32-bit architecture that is upward-compatible with the H8/300 and H8/300H CPUs. The H8S/2600 CPU has sixteen 16-bit general registers, can address a 16-Mbyte (architecturally 4-Gbyte) linear address space, and is ideal for realtime control.

2.1.1 Features

The H8S/2600 CPU has the following features.

- Upward-compatible with H8/300 and H8/300H CPUs
  - Can execute H8/300 and H8/300H object programs
- General-register architecture
  - Sixteen 16-bit general registers (also usable as sixteen 8-bit registers or eight 32-bit registers)
- Sixty-nine basic instructions
  - 8/16/32-bit arithmetic and logic instructions
  - Multiply and divide instructions
  - Powerful bit-manipulation instructions
  - Multiply-and-accumulate instruction
- Eight addressing modes
  - Register direct [Rn]
  - Register indirect [@ERn]
  - Register indirect with displacement [@(d:16,ERn) or @(d:32,ERn)]
  - Register indirect with post-increment or pre-decrement [@ERn+ or @–ERn]
  - Absolute address [@aa:8, @aa:16, @aa:24, or @aa:32]
  - Immediate [#xx:8, #xx:16, or #xx:32]
  - Program-counter relative [@(d:8,PC) or @(d:16,PC)]
  - Memory indirect [@@aa:8]
- 16-Mbyte address space
  - Program: 16 Mbytes
  - Data: 16 Mbytes (4 Gbytes architecturally)

- High-speed operation
  - — All frequently-used instructions execute in one or two states
  - — Maximum clock rate : 20 MHz
  - — 8/16/32-bit register-register add/subtract : 50 ns
  - — 8 × 8-bit register-register multiply : 150 ns
  - — 16 ÷ 8-bit register-register divide : 600 ns
  - — 16 × 16-bit register-register multiply : 200 ns
  - — 32 ÷ 16-bit register-register divide : 1000 ns

- Two CPU operating modes
  - — Normal mode*
  - — Advanced mode

Note: * Not available in the H8S/2636.

- Power-down state
  - — Transition to power-down state by SLEEP instruction
  - — CPU clock speed selection

2.1.2 Differences between H8S/2600 CPU and H8S/2000 CPU

The differences between the H8S/2600 CPU and the H8S/2000 CPU are as shown below.

- Register configuration

  The MAC register is supported only by the H8S/2600 CPU.
- Basic instructions

  The four instructions MAC, CLRMAC, LDMAC, and STMAC are supported only by the H8S/2600 CPU.
- Number of execution states

  The number of execution states of the MULXU and MULXS instructions is different in each CPU.

		Execution States	
Instruction	**Mnemonic**	**H8S/2600**	**H8S/2000**
MULXU	MULXU.B Rs, Rd	3	12
	MULXU.W Rs, ERd	4	20
MULXS	MULXS.B Rs, Rd	4	13
	MULXS.W Rs, ERd	5	21

In addition, there are differences in address space, CCR and EXR register functions, power-down modes, etc., depending on the model.

2.1.3 Differences from H8/300 CPU

In comparison to the H8/300 CPU, the H8S/2600 CPU has the following enhancements.

- More general registers and control registers
  - — Eight 16-bit expanded registers, and one 8-bit and two 32-bit control registers, have been added.
- Expanded address space
  - — Normal mode* supports the same 64-kbyte address space as the H8/300 CPU.
  - — Advanced mode supports a maximum 16-Mbyte address space.

Note: * Not available in the H8S/2636.

- Enhanced addressing
  - — The addressing modes have been enhanced to make effective use of the 16-Mbyte address space.
- Enhanced instructions
  - — Addressing modes of bit-manipulation instructions have been enhanced.
  - — Signed multiply and divide instructions have been added.
  - — A multiply-and-accumulate instruction has been added.
  - — Two-bit shift instructions have been added.
  - — Instructions for saving and restoring multiple registers have been added.
  - — A test and set instruction has been added.
- Higher speed
  - — Basic instructions execute twice as fast.

2.1.4 Differences from H8/300H CPU

In comparison to the H8/300H CPU, the H8S/2600 CPU has the following enhancements.

- Additional control register
  - — One 8-bit and two 32-bit control registers have been added.
- Enhanced instructions
  - — Addressing modes of bit-manipulation instructions have been enhanced.
  - — A multiply-and-accumulate instruction has been added.

— Two-bit shift instructions have been added.
— Instructions for saving and restoring multiple registers have been added.
— A test and set instruction has been added.

- Higher speed
  — Basic instructions execute twice as fast.

2.2 CPU Operating Modes

The H8S/2600 CPU has two operating modes: normal and advanced. Normal mode* supports a maximum 64-kbyte address space. Advanced mode supports a maximum 16-Mbyte total address space (architecturally a maximum 16-Mbyte program area and a maximum of 4 Gbytes for program and data areas combined). The mode is selected by the mode pins of the microcontroller.

Note: * Not available in the H8S/2636 and H8S/2638.

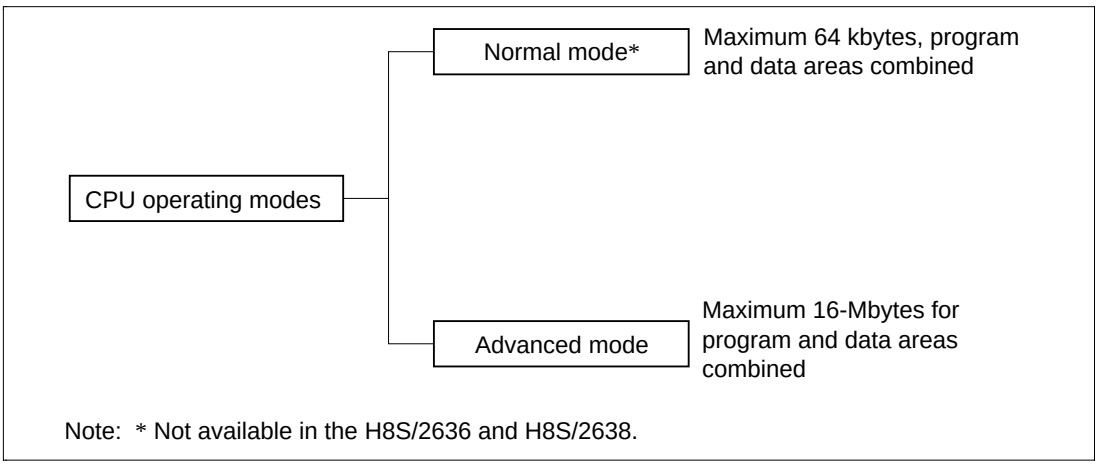


Figure 2-1 CPU Operating Modes

(1) Normal Mode (Not Available in the H8S/2636 and H8S/2638)

The exception vector table and stack have the same structure as in the H8/300 CPU.

Address Space: A maximum address space of 64 kbytes can be accessed.

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers. When En is used as a 16-bit register it can contain any value, even when the corresponding general register (Rn) is used as an address register. If the general register is referenced in the register indirect addressing mode with pre-decrement (@–Rn) or post-increment (@Rn+) and a carry or borrow occurs, however, the value in the corresponding extended register (En) will be affected.

Instruction Set: All instructions and addressing modes can be used. Only the lower 16 bits of effective addresses (EA) are valid.

Exception Vector Table and Memory Indirect Branch Addresses: In normal mode the top area starting at H'0000 is allocated to the exception vector table. One branch address is stored per 16 bits (figure 2-2). The exception vector table differs depending on the microcontroller. For details of the exception vector table, see section 4, Exception Handling.

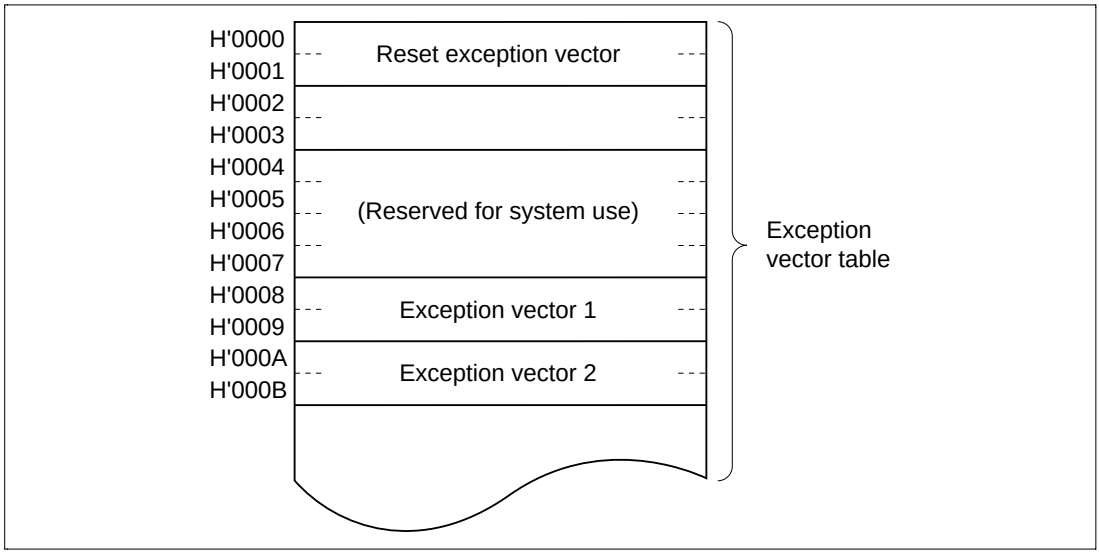
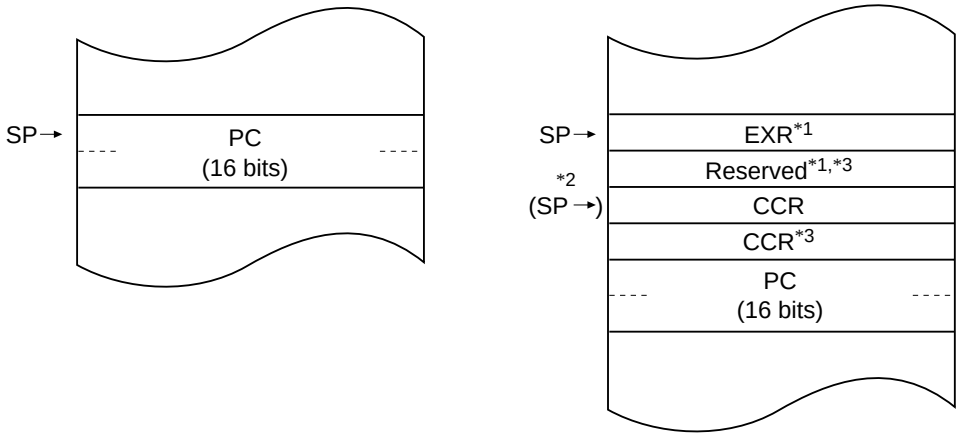


Figure 2-2 Exception Vector Table (Normal Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In normal mode the operand is a 16-bit word operand, providing a 16-bit branch address. Branch addresses can be stored in the top area from H'0000 to H'00FF. Note that this area is also used for the exception vector table.

Stack Structure: When the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2-3. When EXR is invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.



(a) Subroutine Branch **(b) Exception Handling**

Notes: *1 When EXR is not used it is not stored on the stack.
*2 SP when EXR is not used.
*3 Ignored when returning.

Figure 2-3 Stack Structure in Normal Mode

(2) Advanced Mode

Address Space: Linear access is provided to a 16-Mbyte maximum address space (architecturally a maximum 16-Mbyte program area and a maximum 4-Gbyte data area, with a maximum of 4 Gbytes for program and data areas combined).

Extended Registers (En): The extended registers (E0 to E7) can be used as 16-bit registers, or as the upper 16-bit segments of 32-bit registers or address registers.

Instruction Set: All instructions and addressing modes can be used.

Exception Vector Table and Memory Indirect Branch Addresses: In advanced mode the top area starting at H'00000000 is allocated to the exception vector table in units of 32 bits. In each 32 bits, the upper 8 bits are ignored and a branch address is stored in the lower 24 bits (figure 2-4). For details of the exception vector table, see section 4, Exception Handling.

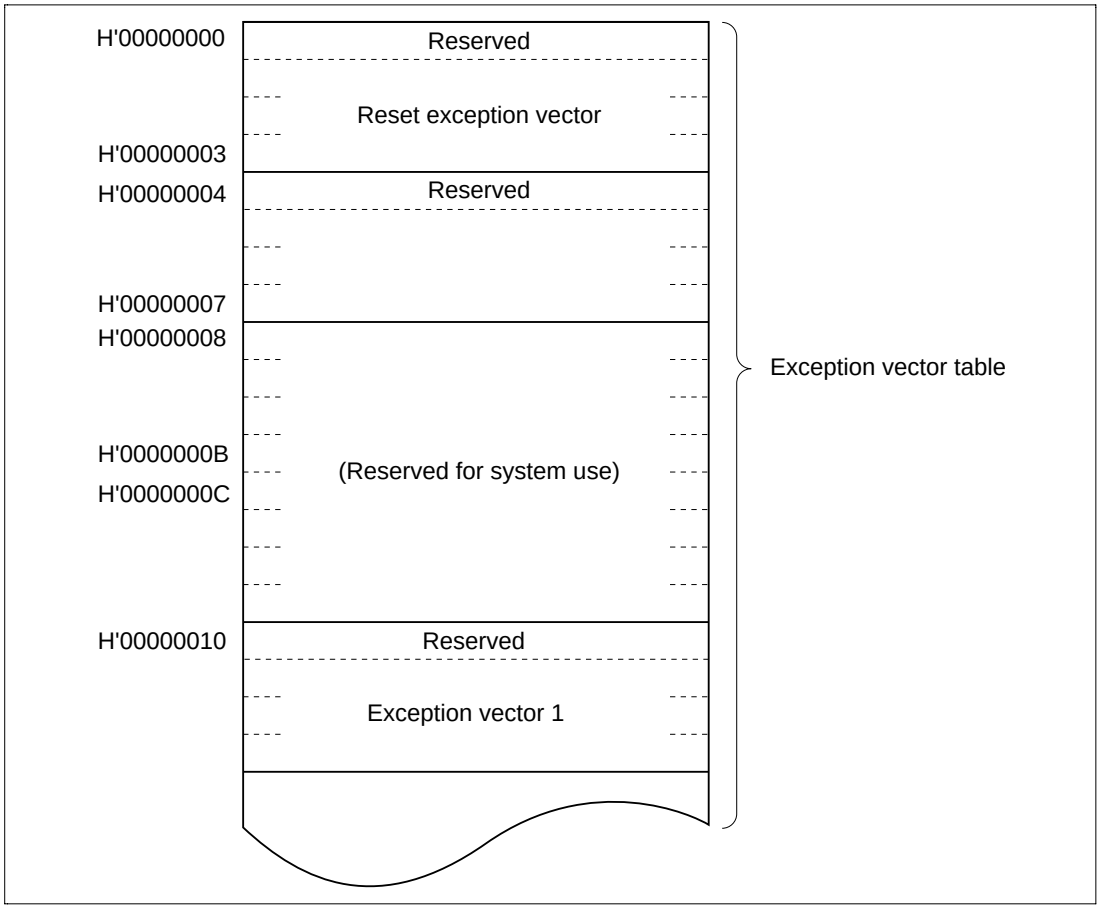
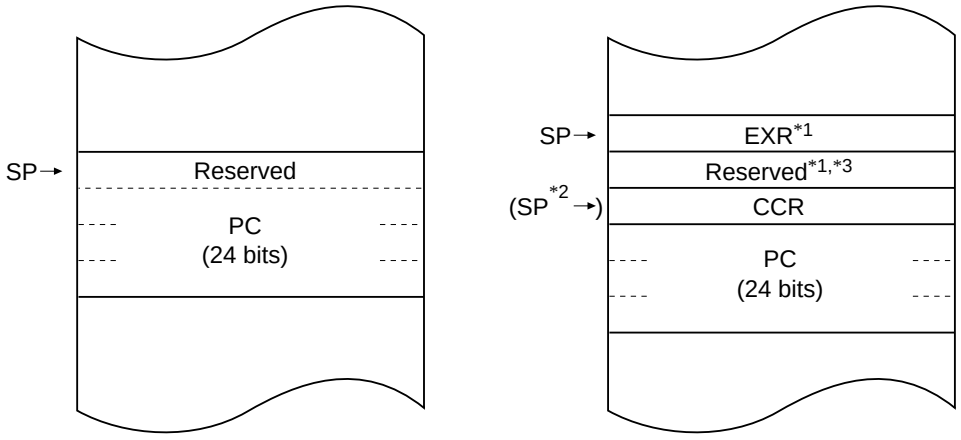


Figure 2-4 Exception Vector Table (Advanced Mode)

The memory indirect addressing mode (@@aa:8) employed in the JMP and JSR instructions uses an 8-bit absolute address included in the instruction code to specify a memory operand that contains a branch address. In advanced mode the operand is a 32-bit longword operand, providing a 32-bit branch address. The upper 8 bits of these 32 bits are a reserved area that is regarded as H'00. Branch addresses can be stored in the area from H'00000000 to H'000000FF. Note that the first part of this range is also the exception vector table.

Stack Structure: In advanced mode, when the program counter (PC) is pushed onto the stack in a subroutine call, and the PC, condition-code register (CCR), and extended control register (EXR) are pushed onto the stack in exception handling, they are stored as shown in figure 2-5. When EXR is invalid, it is not pushed onto the stack. For details, see section 4, Exception Handling.



(a) Subroutine Branch **(b) Exception Handling**

Notes: *1 When EXR is not used it is not stored on the stack.
*2 SP when EXR is not used.
*3 Ignored when returning.

Figure 2-5 Stack Structure in Advanced Mode

2.3 Address Space

Figure 2-6 shows a memory map of the H8S/2600 CPU. The H8S/2600 CPU provides linear access to a maximum 64-kbyte address space in normal mode, and a maximum 16-Mbyte (architecturally 4-Gbyte) address space in advanced mode.

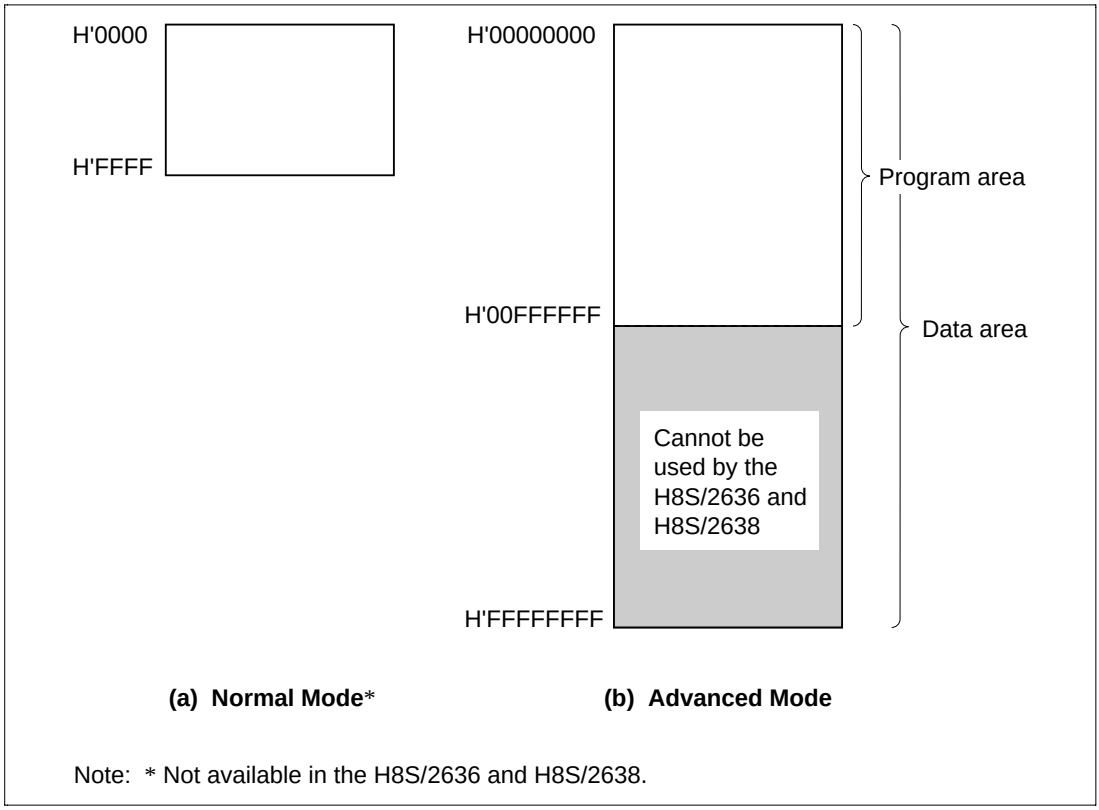


Figure 2-6 Memory Map

2.4 Register Configuration

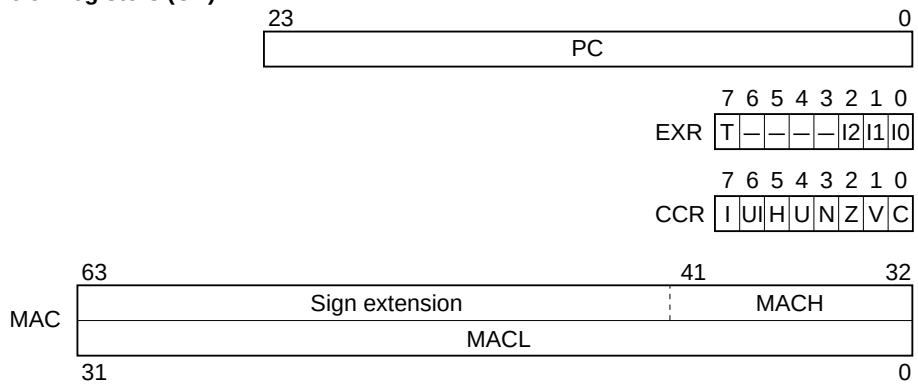
2.4.1 Overview

The CPU has the internal registers shown in figure 2-7. There are two types of registers: general registers and control registers.

General Registers (Rn) and Extended Registers (En)

	15 — 07	07 — 0	
ER0	E0	R0H	R0L
ER1	E1	R1H	R1L
ER2	E2	R2H	R2L
ER3	E3	R3H	R3L
ER4	E4	R4H	R4L
ER5	E5	R5H	R5L
ER6	E6	R6H	R6L
ER7 (SP)	E7	R7H	R7L

Control Registers (CR)



Legend

SP:	Stack pointer	H:	Half-carry flag
PC:	Program counter	U:	User bit
EXR:	Extended control register	N:	Negative flag
T:	Trace bit	Z:	Zero flag
I2 to I0:	Interrupt mask bits	V:	Overflow flag
CCR:	Condition-code register	C:	Carry flag
I:	Interrupt mask bit	MAC:	Multiply-accumulate register
UI:	User bit or interrupt mask bit*		

Note: * Cannot be used as an interrupt mask bit in the H8S/2636 and H8S/2638.

Figure 2-7 CPU Registers

2.4.2 General Registers

The CPU has eight 32-bit general registers. These general registers are all functionally alike and can be used as both address registers and data registers. When a general register is used as a data register, it can be accessed as a 32-bit, 16-bit, or 8-bit register. When the general registers are used as 32-bit registers or address registers, they are designated by the letters ER (ER0 to ER7).

The ER registers divide into 16-bit general registers designated by the letters E (E0 to E7) and R (R0 to R7). These registers are functionally equivalent, providing a maximum sixteen 16-bit registers. The E registers (E0 to E7) are also referred to as extended registers.

The R registers divide into 8-bit general registers designated by the letters RH (R0H to R7H) and RL (R0L to R7L). These registers are functionally equivalent, providing a maximum sixteen 8-bit registers.

Figure 2-8 illustrates the usage of the general registers. The usage of each register can be selected independently.

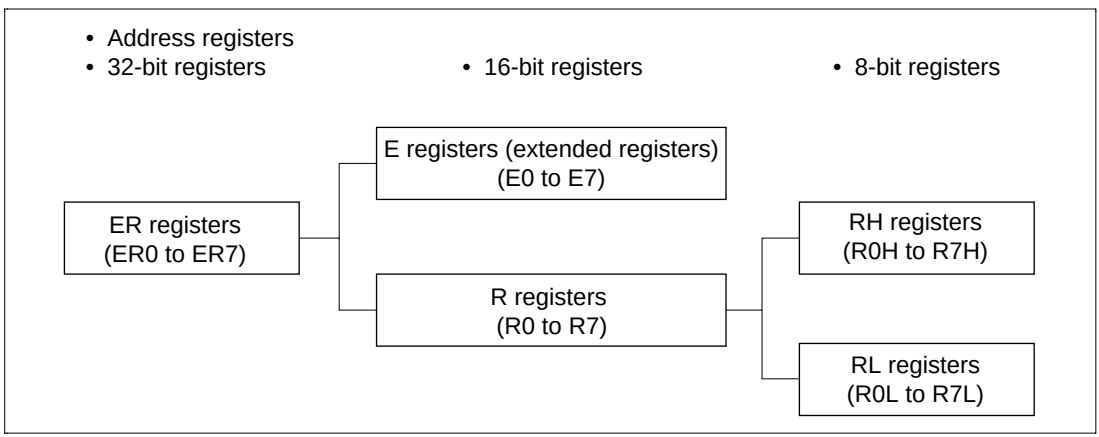


Figure 2-8 Usage of General Registers

General register ER7 has the function of stack pointer (SP) in addition to its general-register function, and is used implicitly in exception handling and subroutine calls. Figure 2-9 shows the stack.

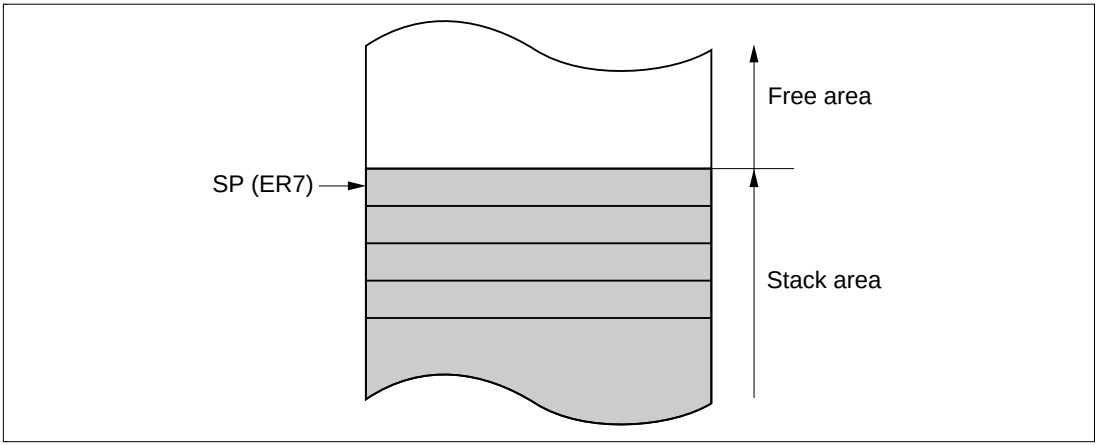


Figure 2-9 Stack

2.4.3 Control Registers

The control registers are the 24-bit program counter (PC), 8-bit extended control register (EXR), 8-bit condition-code register (CCR), and 64-bit multiply-accumulate register (MAC).

(1) Program Counter (PC): This 24-bit counter indicates the address of the next instruction the CPU will execute. The length of all CPU instructions is 2 bytes (one word), so the least significant PC bit is ignored. (When an instruction is fetched, the least significant PC bit is regarded as 0.)

(2) Extended Control Register (EXR): This 8-bit register contains the trace bit (T) and three interrupt mask bits (I2 to I0).

Bit 7—Trace Bit (T): Selects trace mode. When this bit is cleared to 0, instructions are executed in sequence. When this bit is set to 1, a trace exception is generated each time an instruction is executed.

Bits 6 to 3—Reserved: These bits are reserved. They are always read as 1.

Bits 2 to 0—Interrupt Mask Bits (I2 to I0): These bits designate the interrupt mask level (0 to 7). For details, refer to section 5, Interrupt Controller.

Operations can be performed on the EXR bits by the LDC, STC, ANDC, ORC, and XORC instructions. All interrupts, including NMI, are disabled for three states after one of these instructions is executed, except for STC.

(3) Condition-Code Register (CCR): This 8-bit register contains internal CPU status information, including an interrupt mask bit (I) and half-carry (H), negative (N), zero (Z), overflow (V), and carry (C) flags.

Bit 7—Interrupt Mask Bit (I): Masks interrupts other than NMI when set to 1. (NMI is accepted regardless of the I bit setting.) The I bit is set to 1 by hardware at the start of an exception-handling sequence. For details, refer to section 5, Interrupt Controller.

Bit 6—User Bit or Interrupt Mask Bit (UI): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions. This bit can also be used as an interrupt mask bit. For details, refer to section 5, Interrupt Controller.

Bit 5—Half-Carry Flag (H): When the ADD.B, ADDX.B, SUB.B, SUBX.B, CMP.B, or NEG.B instruction is executed, this flag is set to 1 if there is a carry or borrow at bit 3, and cleared to 0 otherwise. When the ADD.W, SUB.W, CMP.W, or NEG.W instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 11, and cleared to 0 otherwise. When the ADD.L, SUB.L, CMP.L, or NEG.L instruction is executed, the H flag is set to 1 if there is a carry or borrow at bit 27, and cleared to 0 otherwise.

Bit 4—User Bit (U): Can be written and read by software using the LDC, STC, ANDC, ORC, and XORC instructions.

Bit 3—Negative Flag (N): Stores the value of the most significant bit (sign bit) of data.

Bit 2—Zero Flag (Z): Set to 1 to indicate zero data, and cleared to 0 to indicate non-zero data.

Bit 1—Overflow Flag (V): Set to 1 when an arithmetic overflow occurs, and cleared to 0 at other times.

Bit 0—Carry Flag (C): Set to 1 when a carry occurs, and cleared to 0 otherwise. Used by:

- Add instructions, to indicate a carry
- Subtract instructions, to indicate a borrow
- Shift and rotate instructions, to store the value shifted out of the end bit

The carry flag is also used as a bit accumulator by bit manipulation instructions.

Some instructions leave some or all of the flag bits unchanged. For the action of each instruction on the flag bits, refer to Appendix A.1, List of Instructions.

Operations can be performed on the CCR bits by the LDC, STC, ANDC, ORC, and XORC instructions. The N, Z, V, and C flags are used as branching conditions for conditional branch (Bcc) instructions.

(4) Multiply-Accumulate Register (MAC): This 64-bit register stores the results of multiply-and-accumulate operations. It consists of two 32-bit registers denoted MACH and MACL. The lower 10 bits of MACH are valid; the upper bits are a sign extension.

2.4.4 Initial Register Values

Reset exception handling loads the CPU's program counter (PC) from the vector table, clears the trace bit in EXR to 0, and sets the interrupt mask bits in CCR and EXR to 1. The other CCR bits and the general registers are not initialized. In particular, the stack pointer (ER7) is not initialized. The stack pointer should therefore be initialized by an MOV.L instruction executed immediately after a reset.

2.5 Data Formats

The CPU can process 1-bit, 4-bit (BCD), 8-bit (byte), 16-bit (word), and 32-bit (longword) data. Bit-manipulation instructions operate on 1-bit data by accessing bit n (n = 0, 1, 2, …, 7) of byte operand data. The DAA and DAS decimal-adjust instructions treat byte data as two digits of 4-bit BCD data.

2.5.1 General Register Data Formats

Figure 2-10 shows the data formats in general registers.

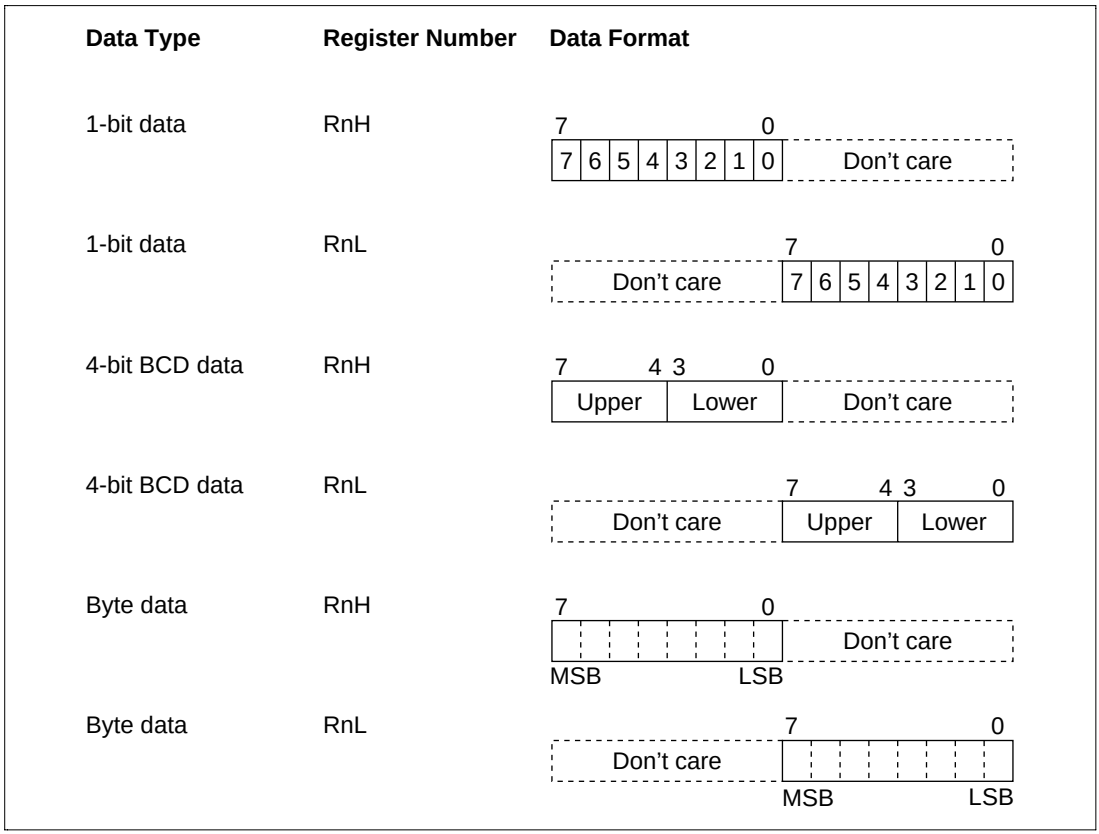
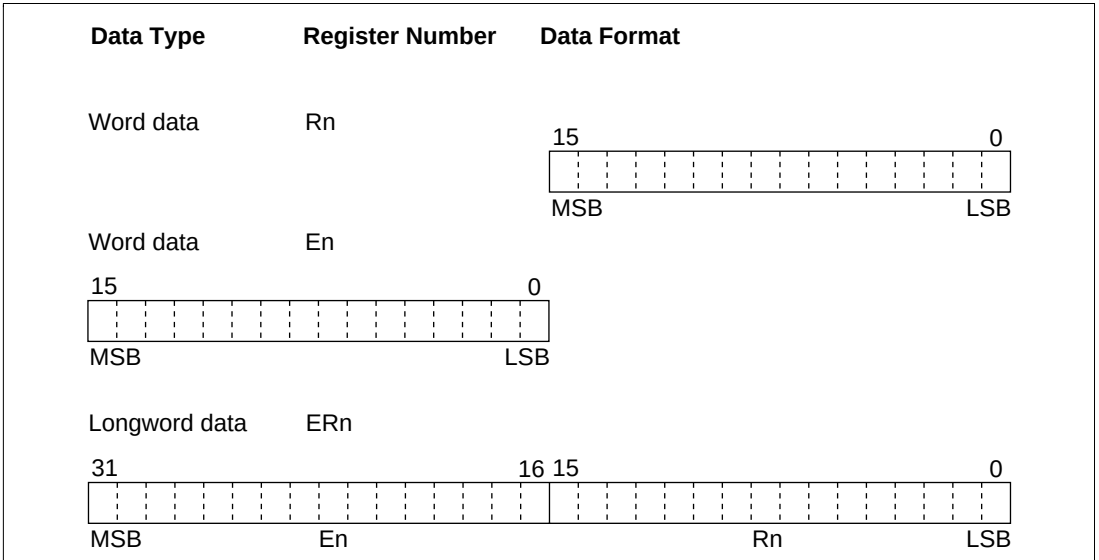


Figure 2-10 General Register Data Formats



Legend

ERn: General register ER
En: General register E
Rn: General register R
RnH: General register RH
RnL: General register RL
MSB: Most significant bit
LSB: Least significant bit

Figure 2-10 General Register Data Formats (cont)

2.5.2 Memory Data Formats

Figure 2-11 shows the data formats in memory. The CPU can access word data and longword data in memory, but word or longword data must begin at an even address. If an attempt is made to access word or longword data at an odd address, no address error occurs but the least significant bit of the address is regarded as 0, so the access starts at the preceding address. This also applies to instruction fetches.

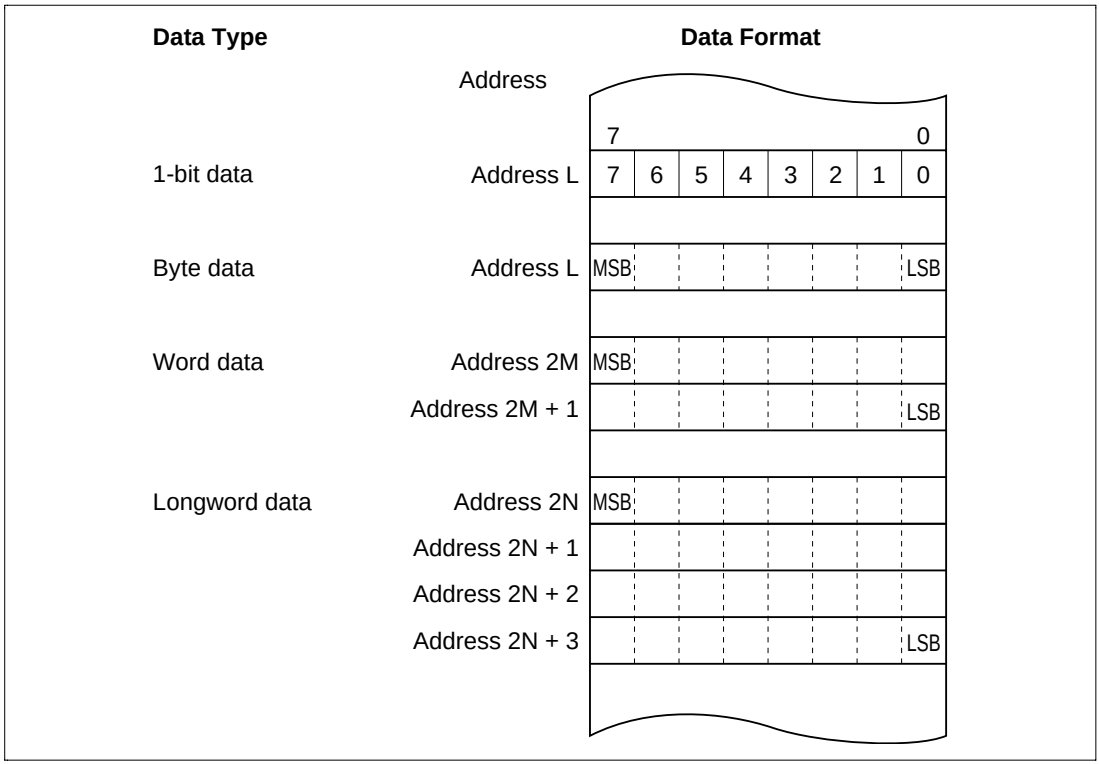


Figure 2-11 Memory Data Formats

When ER7 is used as an address register to access the stack, the operand size should be word size or longword size.

2.6 Instruction Set

2.6.1 Overview

The H8S/2600 CPU has 69 types of instructions. The instructions are classified by function in table 2-1.

Table 2-1 Instruction Classification

Function	Instructions	Size	Types
Data transfer	MOV	BWL	5
	POP*[1], PUSH*[1]	WL	
	LDM*[5], STM*[5]	L	
	MOVFPE*[3], MOVTPE*[3]	B	
Arithmetic operations	ADD, SUB, CMP, NEG	BWL	23
	ADDX, SUBX, DAA, DAS	B	
	INC, DEC	BWL	
	ADDS, SUBS	L	
	MULXU, DIVXU, MULXS, DIVXS	BW	
	EXTU, EXTS	WL	
	TAS*[4]	B	
	MAC, LDMAC, STMAC, CLRMAC	—	
Logic operations	AND, OR, XOR, NOT	BWL	4
Shift	SHAL, SHAR, SHLL, SHLR, ROTL, ROTR, ROTXL, ROTXR	BWL	8
Bit manipulation	BSET, BCLR, BNOT, BTST, BLD, BILD, BST, BIST, BAND, BIAND, BOR, BIOR, BXOR, BIXOR	B	14
Branch	Bcc*[2], JMP, BSR, JSR, RTS	—	5
System control	TRAPA, RTE, SLEEP, LDC, STC, ANDC, ORC, XORC, NOP	—	9
Block data transfer	EEPMOV	—	1

Notes: B-byte size; W-word size; L-longword size.

*1 POP.W Rn and PUSH.W Rn are identical to MOV.W @SP+, Rn and MOV.W Rn, @-SP. POP.L ERn and PUSH.L ERn are identical to MOV.L @SP+, ERn and MOV.L ERn, @-SP.

*2 Bcc is the general name for conditional branch instructions.

*3 Not available in the H8S/2636 and H8S/2638.

*4 Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

*5 Only registers ER0 to ER6 should be used when using the STM/LDM instruction.

2.6.2 Instructions and Addressing Modes

Table 2-2 indicates the combinations of instructions and addressing modes that the H8S/2600 CPU can use.

Table 2-2 Combinations of Instructions and Addressing Modes

Function	Instruction	Addressing Modes														
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:32,ERn)	@–ERn/@ERn+	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC)	@@aa:8	—
Data transfer	MOV	BWL	BWL	BWL	BWL	BWL	BWL	B	BWL	—	BWL	—	—	—	—
	POP, PUSH	—	—	—	—	—	—	—	—	—	—	—	—	—	WL
	LDM*3, STM*3	—	—	—	—	—	—	—	—	—	—	—	—	—	L
	MOVFPE*1, MOVTPE*1	—	—	—	—	—	—	—	B	—	—	—	—	—	—
Arithmetic operations	ADD, CMP	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	SUB	WL	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	ADDX, SUBX	B	B	—	—	—	—	—	—	—	—	—	—	—	—
	ADDS, SUBS	—	L	—	—	—	—	—	—	—	—	—	—	—	—
	INC, DEC	—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	DAA, DAS	—	B	—	—	—	—	—	—	—	—	—	—	—	—
	MULXU, DIVXU	—	BW	—	—	—	—	—	—	—	—	—	—	—	—
	MULXS, DIVXS	—	BW	—	—	—	—	—	—	—	—	—	—	—	—
	NEG	—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	EXTU, EXTS	—	WL	—	—	—	—	—	—	—	—	—	—	—	—
	TAS*2	—	—	B	—	—	—	—	—	—	—	—	—	—	—
	MAC	—	—	—	—	—	○	—	—	—	—	—	—	—	—
	CLRMAC	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	LDMAC, STMAC	—	L	—	—	—	—	—	—	—	—	—	—	—	—

Function	Instruction	Addressing Modes													
		#xx	Rn	@ERn	@(d:16,ERn)	@(d:32,ERn)	@–ERn/@ERn+	@aa:8	@aa:16	@aa:24	@aa:32	@(d:8,PC)	@(d:16,PC)	@@aa:8	—
Logic operations	AND, OR, XOR	BWL	BWL	—	—	—	—	—	—	—	—	—	—	—	—
	NOT	—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
Shift		—	BWL	—	—	—	—	—	—	—	—	—	—	—	—
Bit manipulation		—	B	B	—	—	—	B	B	—	B	—	—	—	—
Branch	Bcc, BSR	—	—	—	—	—	—	—	—	—	—	○	○	—	—
	JMP, JSR	—	—	—	—	—	—	—	—	○	—	—	—	○	—
	RTS	—	—	—	—	—	—	—	—	—	—	—	—	—	○
System control	TRAPA	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	RTE	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	SLEEP	—	—	—	—	—	—	—	—	—	—	—	—	—	○
	LDC	B	B	W	W	W	W	—	W	—	W	—	—	—	—
	STC	—	B	W	W	W	W	—	W	—	W	—	—	—	—
	ANDC, ORC, XORC	B	—	—	—	—	—	—	—	—	—	—	—	—	—
	NOP	—	—	—	—	—	—	—	—	—	—	—	—	—	○
Block data transfer		—	—	—	—	—	—	—	—	—	—	—	—	—	BW

Legend

B: Byte

W: Word

L: Longword

Note: *1 Not available in the H8S/2636 and H8S/2638.

*2 Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

*3 Only registers ER0 to ER6 should be used when using the STM/LDM instruction.

2.6.3 Table of Instructions Classified by Function

Table 2-3 summarizes the instructions in each functional category. The notation used in table 2-3 is defined below.

Operation Notation

Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Addition
–	Subtraction
×	Multiplication
÷	Division
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Move
¬	NOT (logical complement)
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

Table 2-3 Instructions Classified by Function

Type	Instruction	Size*[1]	Function
Data transfer	MOV	B/W/L	(EAs) → Rd, Rs → (EAd) Moves data between two general registers or between a general register and memory, or moves immediate data to a general register.
	MOVFPE	B	Cannot be used in the H8S/2636 and H8S/2638.
	MOVTPE	B	Cannot be used in the H8S/2636 and H8S/2638.
	POP	W/L	@SP+ → Rn Pops a register from the stack. POP.W Rn is identical to MOV.W @SP+, Rn. POP.L ERn is identical to MOV.L @SP+, ERn.
	PUSH	W/L	Rn → @–SP Pushes a register onto the stack. PUSH.W Rn is identical to MOV.W Rn, @–SP. PUSH.L ERn is identical to MOV.L ERn, @–SP.
	LDM*[2]	L	@SP+ → Rn (register list) Pops two or more general registers from the stack.
	STM*[2]	L	Rn (register list) → @–SP Pushes two or more general registers onto the stack.

Type	Instruction	Size*[1]	Function
Arithmetic operations	ADD SUB	B/W/L	Rd ± Rs → Rd, Rd ± #IMM → Rd Performs addition or subtraction on data in two general registers, or on immediate data and data in a general register. (Immediate byte data cannot be subtracted from byte data in a general register. Use the SUBX or ADD instruction.)
	ADDX SUBX	B	Rd ± Rs ± C → Rd, Rd ± #IMM ± C → Rd Performs addition or subtraction with carry or borrow on byte data in two general registers, or on immediate data and data in a general register.
	INC DEC	B/W/L	Rd ± 1 → Rd, Rd ± 2 → Rd Increments or decrements a general register by 1 or 2. (Byte operands can be incremented or decremented by 1 only.)
	ADDS SUBS	L	Rd ± 1 → Rd, Rd ± 2 → Rd, Rd ± 4 → Rd Adds or subtracts the value 1, 2, or 4 to or from data in a 32-bit register.
	DAA DAS	B	Rd decimal adjust → Rd Decimal-adjusts an addition or subtraction result in a general register by referring to the CCR to produce 4-bit BCD data.
	MULXU	B/W	Rd × Rs → Rd Performs unsigned multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
	MULXS	B/W	Rd × Rs → Rd Performs signed multiplication on data in two general registers: either 8 bits × 8 bits → 16 bits or 16 bits × 16 bits → 32 bits.
	DIVXU	B/W	Rd ÷ Rs → Rd Performs unsigned division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.

Type	Instruction	Size*[1]	Function
Arithmetic operations	DIVXS	B/W	Rd ÷ Rs → Rd Performs signed division on data in two general registers: either 16 bits ÷ 8 bits → 8-bit quotient and 8-bit remainder or 32 bits ÷ 16 bits → 16-bit quotient and 16-bit remainder.
	CMP	B/W/L	Rd – Rs, Rd – #IMM Compares data in a general register with data in another general register or with immediate data, and sets CCR bits according to the result.
	NEG	B/W/L	0 – Rd → Rd Takes the two's complement (arithmetic complement) of data in a general register.
	EXTU	W/L	Rd (zero extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by padding with zeros on the left.
	EXTS	W/L	Rd (sign extension) → Rd Extends the lower 8 bits of a 16-bit register to word size, or the lower 16 bits of a 32-bit register to longword size, by extending the sign bit.
	TAS	B	@ERd – 0, 1 → (<bit 7> of @ERd)*[3] Tests memory contents, and sets the most significant bit (bit 7) to 1.
	MAC	—	(EAs) × (EAd) + MAC → MAC Performs signed multiplication on memory contents and adds the result to the multiply-accumulate register. The following operations can be performed: 16 bits × 16 bits + 32 bits → 32 bits, saturating 16 bits × 16 bits + 42 bits → 42 bits, non-saturating
	CLRMAC	—	0 → MAC Clears the multiply-accumulate register to zero.
	LDMAC STMAC	L	Rs → MAC, MAC → Rd Transfers data between a general register and a multiply-accumulate register.

Type	Instruction	Size*[1]	Function
Logic operations	AND	B/W/L	Rd $\wedge$ Rs $\rightarrow$ Rd, Rd $\wedge$ #IMM $\rightarrow$ Rd Performs a logical AND operation on a general register and another general register or immediate data.
	OR	B/W/L	Rd $\vee$ Rs $\rightarrow$ Rd, Rd $\vee$ #IMM $\rightarrow$ Rd Performs a logical OR operation on a general register and another general register or immediate data.
	XOR	B/W/L	Rd $\oplus$ Rs $\rightarrow$ Rd, Rd $\oplus$ #IMM $\rightarrow$ Rd Performs a logical exclusive OR operation on a general register and another general register or immediate data.
	NOT	B/W/L	$\neg$ (Rd) $\rightarrow$ (Rd) Takes the one's complement of general register contents.
Shift operations	SHAL SHAR	B/W/L	Rd (shift) $\rightarrow$ Rd Performs an arithmetic shift on general register contents. 1-bit or 2-bit shift is possible.
	SHLL SHLR	B/W/L	Rd (shift) $\rightarrow$ Rd Performs a logical shift on general register contents. 1-bit or 2-bit shift is possible.
	ROTL ROTR	B/W/L	Rd (rotate) $\rightarrow$ Rd Rotates general register contents. 1-bit or 2-bit rotation is possible.
	ROTXL ROTXR	B/W/L	Rd (rotate) $\rightarrow$ Rd Rotates general register contents through the carry flag. 1-bit or 2-bit rotation is possible.

Type	Instruction	Size*[1]	Function
Bit-manipulation instructions	BSET	B	1 → (<bit-No.> of <EAd>) Sets a specified bit in a general register or memory operand to 1. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BCLR	B	0 → (<bit-No.> of <EAd>) Clears a specified bit in a general register or memory operand to 0. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BNOT	B	¬ (<bit-No.> of <EAd>) → (<bit-No.> of <EAd>) Inverts a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BTST	B	¬ (<bit-No.> of <EAd>) → Z Tests a specified bit in a general register or memory operand and sets or clears the Z flag accordingly. The bit number is specified by 3-bit immediate data or the lower three bits of a general register.
	BAND	B	C ∧ (<bit-No.> of <EAd>) → C ANDs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIAND	B	C ∧ [¬ (<bit-No.> of <EAd>)] → C ANDs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
	BOR	B	C ∨ (<bit-No.> of <EAd>) → C ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIOR	B	C ∨ ¬ (<bit-No.> of <EAd>) → C ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.

Type	Instruction	Size*[1]	Function
Bit-manipulation instructions	BXOR	B	C ⊕ (<bit-No.> of <EAd>) → C Exclusive-ORs the carry flag with a specified bit in a general register or memory operand and stores the result in the carry flag.
	BIXOR	B	C ⊕ ¬ [(<bit-No.> of <EAd>)] → C Exclusive-ORs the carry flag with the inverse of a specified bit in a general register or memory operand and stores the result in the carry flag. The bit number is specified by 3-bit immediate data.
	BLD	B	(<bit-No.> of <EAd>) → C Transfers a specified bit in a general register or memory operand to the carry flag.
	BILD	B	¬ (<bit-No.> of <EAd>) → C Transfers the inverse of a specified bit in a general register or memory operand to the carry flag. The bit number is specified by 3-bit immediate data.
	BST	B	C → (<bit-No.> of <EAd>) Transfers the carry flag value to a specified bit in a general register or memory operand.
	BIST	B	¬ C → (<bit-No.> of <EAd>) Transfers the inverse of the carry flag value to a specified bit in a general register or memory operand. The bit number is specified by 3-bit immediate data.

Type	Instruction	Size*[1]	Function
Branch instructions	Bcc	—	Branches to a specified address if a specified condition is true. The branching conditions are listed below.

Mnemonic	Description	Condition
BRA(BT)	Always (true)	Always
BRN(BF)	Never (false)	Never
BHI	High	$C \vee Z = 0$
BLS	Low or same	$C \vee Z = 1$
BCC(BHS)	Carry clear (high or same)	$C = 0$
BCS(BLO)	Carry set (low)	$C = 1$
BNE	Not equal	$Z = 0$
BEQ	Equal	$Z = 1$
BVC	Overflow clear	$V = 0$
BVS	Overflow set	$V = 1$
BPL	Plus	$N = 0$
BMI	Minus	$N = 1$
BGE	Greater or equal	$N \oplus V = 0$
BLT	Less than	$N \oplus V = 1$
BGT	Greater than	$Z \vee (N \oplus V) = 0$
BLE	Less or equal	$Z \vee (N \oplus V) = 1$

	JMP	—	Branches unconditionally to a specified address.
	BSR	—	Branches to a subroutine at a specified address.
	JSR	—	Branches to a subroutine at a specified address.
	RTS	—	Returns from a subroutine

Type	Instruction	Size*[1]	Function
System control instructions	TRAPA	—	Starts trap-instruction exception handling.
	RTE	—	Returns from an exception-handling routine.
	SLEEP	—	Causes a transition to a power-down state.
	LDC	B/W	(EAs) → CCR, (EAs) → EXR Moves the source operand contents or immediate data to CCR or EXR. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
	STC	B/W	CCR → (EAd), EXR → (EAd) Transfers CCR or EXR contents to a general register or memory. Although CCR and EXR are 8-bit registers, word-size transfers are performed between them and memory. The upper 8 bits are valid.
	ANDC	B	CCR ∧ #IMM → CCR, EXR ∧ #IMM → EXR Logically ANDs the CCR or EXR contents with immediate data.
	ORC	B	CCR ∨ #IMM → CCR, EXR ∨ #IMM → EXR Logically ORs the CCR or EXR contents with immediate data.
	XORC	B	CCR ⊕ #IMM → CCR, EXR ⊕ #IMM → EXR Logically exclusive-ORs the CCR or EXR contents with immediate data.
	NOP	—	PC + 2 → PC Only increments the program counter.

Type	Instruction	Size*[1]	Function
Block data transfer instruction	EEPMOV.B	—	if R4L ≠ 0 then Repeat @ER5+ → @ER6+ R4L–1 → R4L Until R4L = 0 else next;
	EEPMOV.W	—	if R4 ≠ 0 then Repeat @ER5+ → @ER6+ R4–1 → R4 Until R4 = 0 else next;
			Transfers a data block according to parameters set in general registers R4L or R4, ER5, and ER6. R4L or R4: size of block (bytes) ER5: starting source address ER6: starting destination address Execution of the next instruction begins as soon as the transfer is completed.

Notes: *1 Size refers to the operand size.
B: Byte
W: Word
L: Longword
*2 Only registers ER0 to ER6 should be used when using the STM/LDM instruction.
*3 Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

2.6.4 Basic Instruction Formats

The CPU instructions consist of 2-byte (1-word) units. An instruction consists of an operation field (op field), a register field (r field), an effective address extension (EA field), and a condition field (cc).

(1) Operation Field: Indicates the function of the instruction, the addressing mode, and the operation to be carried out on the operand. The operation field always includes the first four bits of the instruction. Some instructions have two operation fields.

(2) Register Field: Specifies a general register. Address registers are specified by 3 bits, data registers by 3 bits or 4 bits. Some instructions have two register fields. Some have no register field.

(3) Effective Address Extension: Eight, 16, or 32 bits specifying immediate data, an absolute address, or a displacement.

(4) Condition Field: Specifies the branching condition of Bcc instructions.

Figure 2-12 shows examples of instruction formats.

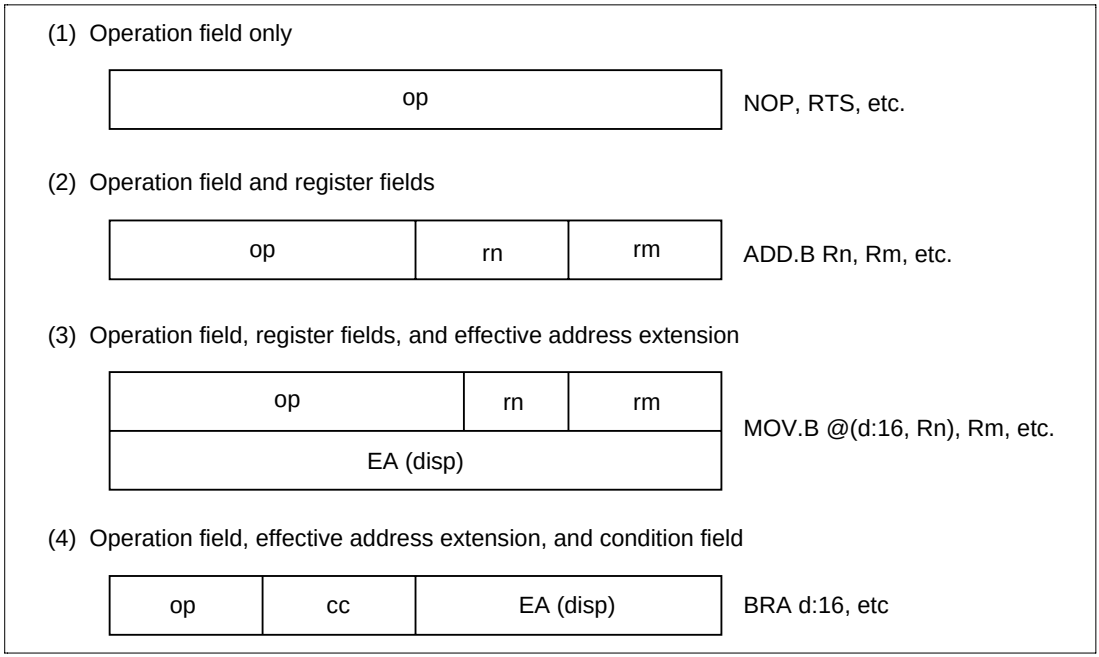


Figure 2-12 Instruction Formats (Examples)

2.7 Addressing Modes and Effective Address Calculation

2.7.1 Addressing Mode

The CPU supports the eight addressing modes listed in table 2-4. Each instruction uses a subset of these addressing modes. Arithmetic and logic instructions can use the register direct and immediate modes. Data transfer instructions can use all addressing modes except program-counter relative and memory indirect. Bit manipulation instructions use register direct, register indirect, or absolute addressing mode to specify an operand, and register direct (BSET, BCLR, BNOT, and BTST instructions) or immediate (3-bit) addressing mode to specify a bit number in the operand.

Table 2-4 Addressing Modes

No.	Addressing Mode	Symbol
1	Register direct	Rn
2	Register indirect	@ERn
3	Register indirect with displacement	@(d:16,ERn)/@(d:32,ERn)
4	Register indirect with post-increment Register indirect with pre-decrement	@ERn+ @–ERn
5	Absolute address	@aa:8/@aa:16/@aa:24/@aa:32
6	Immediate	#xx:8/#xx:16/#xx:32
7	Program-counter relative	@(d:8,PC)/@(d:16,PC)
8	Memory indirect	@@aa:8

(1) Register Direct—Rn: The register field of the instruction specifies an 8-, 16-, or 32-bit general register containing the operand. R0H to R7H and R0L to R7L can be specified as 8-bit registers. R0 to R7 and E0 to E7 can be specified as 16-bit registers. ER0 to ER7 can be specified as 32-bit registers.

(2) Register Indirect—@ERn: The register field of the instruction code specifies an address register (ERn) which contains the address of the operand on memory. If the address is a program instruction address, the lower 24 bits are valid and the upper 8 bits are all assumed to be 0 (H'00).

(3) Register Indirect with Displacement—@(d:16, ERn) or @(d:32, ERn): A 16-bit or 32-bit displacement contained in the instruction is added to an address register (ERn) specified by the register field of the instruction, and the sum gives the address of a memory operand. A 16-bit displacement is sign-extended when added.

(4) Register Indirect with Post-Increment or Pre-Decrement—@ERn+ or @-ERn:

- Register indirect with post-increment—@ERn+

  The register field of the instruction code specifies an address register (ERn) which contains the address of a memory operand. After the operand is accessed, 1, 2, or 4 is added to the address register contents and the sum is stored in the address register. The value added is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, the register value should be even.

- Register indirect with pre-decrement—@-ERn

  The value 1, 2, or 4 is subtracted from an address register (ERn) specified by the register field in the instruction code, and the result becomes the address of a memory operand. The result is also stored in the address register. The value subtracted is 1 for byte access, 2 for word transfer instruction, or 4 for longword transfer instruction. For word or longword transfer instruction, the register value should be even.

(5) Absolute Address—@aa:8, @aa:16, @aa:24, or @aa:32: The instruction code contains the absolute address of a memory operand. The absolute address may be 8 bits long (@aa:8), 16 bits long (@aa:16), 24 bits long (@aa:24), or 32 bits long (@aa:32).

To access data, the absolute address should be 8 bits (@aa:8), 16 bits (@aa:16), or 32 bits (@aa:32) long. For an 8-bit absolute address, the upper 24 bits are all assumed to be 1 (H'FFFF). For a 16-bit absolute address the upper 16 bits are a sign extension. A 32-bit absolute address can access the entire address space.

A 24-bit absolute address (@aa:24) indicates the address of a program instruction. The upper 8 bits are all assumed to be 0 (H'00).

Table 2-5 indicates the accessible absolute address ranges.

Table 2-5 Absolute Address Access Ranges

Absolute Address		Normal Mode*	Advanced Mode
Data address	8 bits (@aa:8)	H'FF00 to H'FFFF	H'FFFF00 to H'FFFFFF
	16 bits (@aa:16)	H'0000 to H'FFFF	H'000000 to H'007FFF, H'FF8000 to H'FFFFFF
	32 bits (@aa:32)		H'000000 to H'FFFFFF
Program instruction address	24 bits (@aa:24)		

Note: * Not available in the H8S/2636 and H8S/2638.

(6) Immediate—#xx:8, #xx:16, or #xx:32: The instruction contains 8-bit (#xx:8), 16-bit (#xx:16), or 32-bit (#xx:32) immediate data as an operand.

The ADDS, SUBS, INC, and DEC instructions contain immediate data implicitly. Some bit manipulation instructions contain 3-bit immediate data in the instruction code, specifying a bit number. The TRAPA instruction contains 2-bit immediate data in its instruction code, specifying a vector address.

(7) Program-Counter Relative—@(d:8, PC) or @(d:16, PC): This mode is used in the Bcc and BSR instructions. An 8-bit or 16-bit displacement contained in the instruction is sign-extended and added to the 24-bit PC contents to generate a branch address. Only the lower 24 bits of this branch address are valid; the upper 8 bits are all assumed to be 0 (H'00). The PC value to which the displacement is added is the address of the first byte of the next instruction, so the possible branching range is –126 to +128 bytes (–63 to +64 words) or –32766 to +32768 bytes (–16383 to +16384 words) from the branch instruction. The resulting value should be an even number.

(8) Memory Indirect—@@aa:8: This mode can be used by the JMP and JSR instructions. The instruction code contains an 8-bit absolute address specifying a memory operand. This memory operand contains a branch address. The upper bits of the absolute address are all assumed to be 0, so the address range is 0 to 255 (H'0000 to H'00FF in normal mode*, H'000000 to H'0000FF in advanced mode). In normal mode* the memory operand is a word operand and the branch address is 16 bits long. In advanced mode the memory operand is a longword operand, the first byte of which is assumed to be all 0 (H'00).

Note that the first part of the address range is also the exception vector area. For further details, refer to section 4, Exception Handling.

Note: * Not available in the H8S/2636 and H8S/2638.

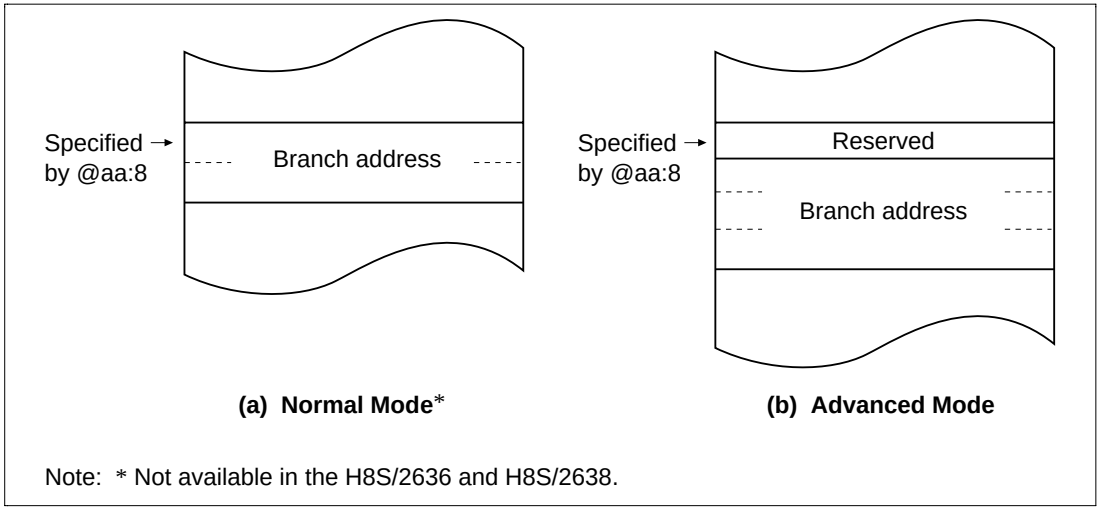


Figure 2-13 Branch Address Specification in Memory Indirect Mode

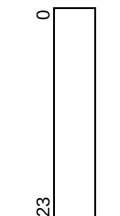
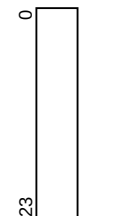
If an odd address is specified in word or longword memory access, or as a branch address, the least significant bit is regarded as 0, causing data to be accessed or instruction code to be fetched at the address preceding the specified address. (For further information, see section 2.5.2, Memory Data Formats.)

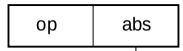
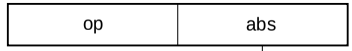
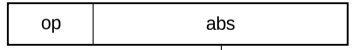
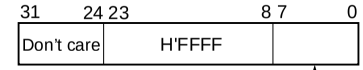
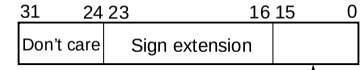
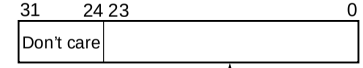
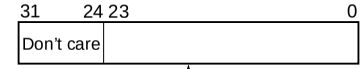
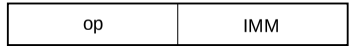
2.7.2 Effective Address Calculation

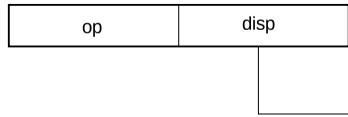
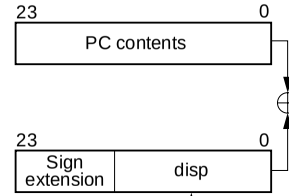
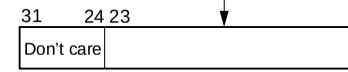
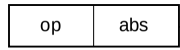
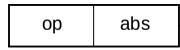
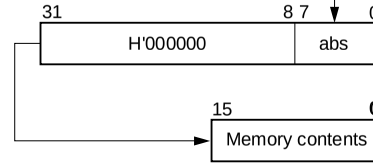
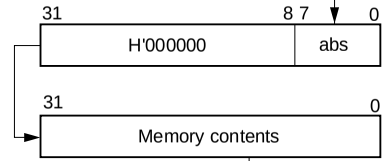
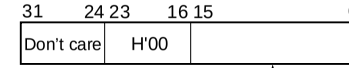
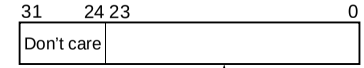
Table 2-6 indicates how effective addresses are calculated in each addressing mode. In normal mode* the upper 8 bits of the effective address are ignored in order to generate a 16-bit address.

Note: * Not available in the H8S/2636 and H8S/2638.

Table 2.6 Effective Address Calculation

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
1	Register direct (Rn) op | rm | rn	Operand is general register contents.	
2	Register indirect (@ERn) op | r		
3	Register indirect with displacement @(d:16, ERn) or @(d:32, ERn) op | r | disp		
4	Register indirect with post-increment or pre-decrement • Register indirect with post-increment @ERn+ op | r • Register indirect with pre-decrement @–ERn op | r		

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
5	Absolute address @aa:8  @aa:16  @aa:24  @aa:32 		   
6	Immediate #xx:8/#xx:16/#xx:32  		Operand is immediate data.

No.	Addressing Mode and Instruction Format	Effective Address Calculation	Effective Address (EA)
7	Program-counter relative @(d:8, PC)/@(d:16, PC) 		
8	Memory indirect @@aa:8 • Normal mode*  • Advanced mode 	 	 

Note: * Not available in the H8S/2636 and H8S/2638.

2.8 Processing States

2.8.1 Overview

The CPU has five main processing states: the reset state, exception handling state, program execution state, bus-released state, and power-down state. Figure 2-14 shows a diagram of the processing states. Figure 2-15 indicates the state transitions.

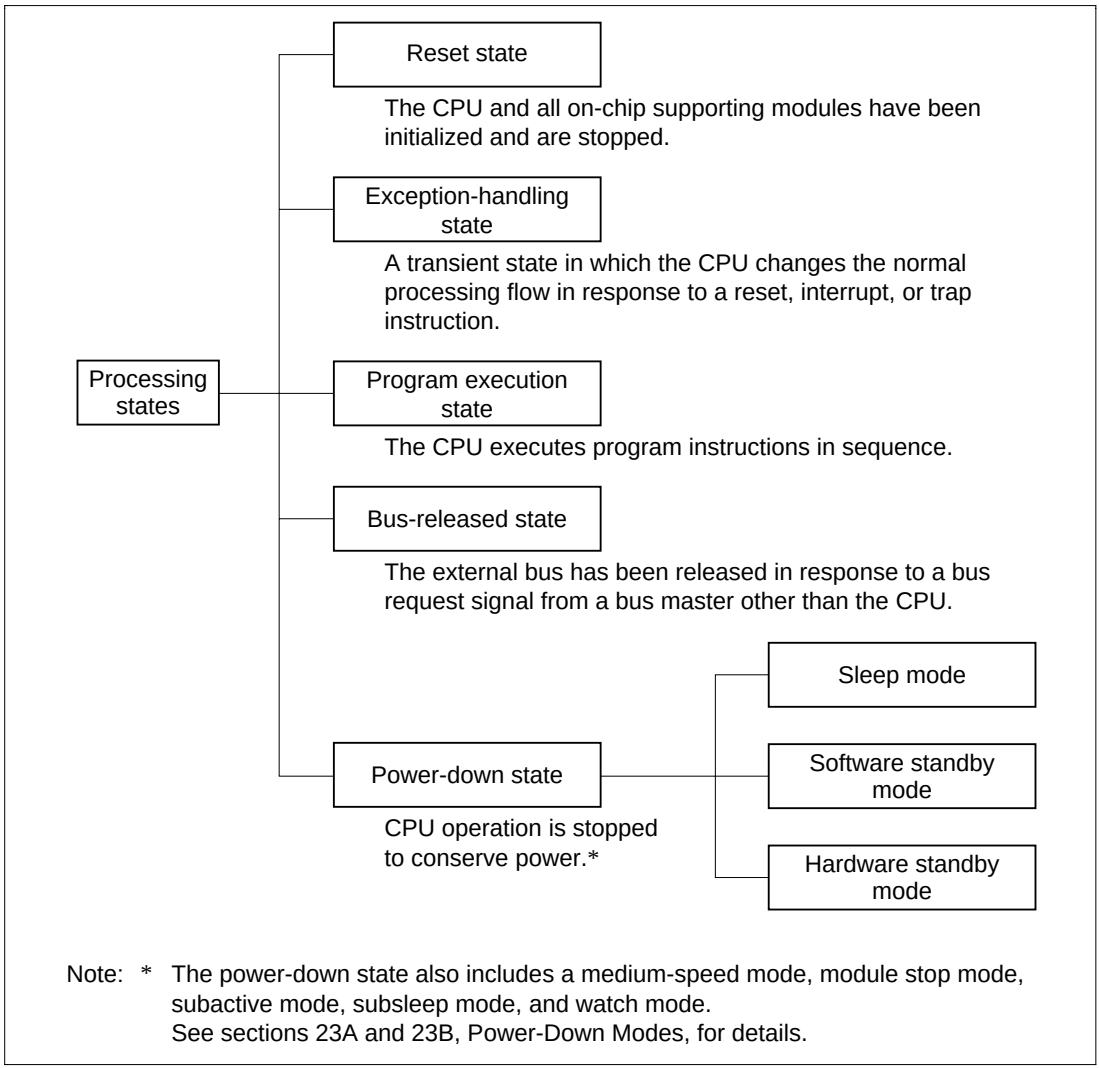
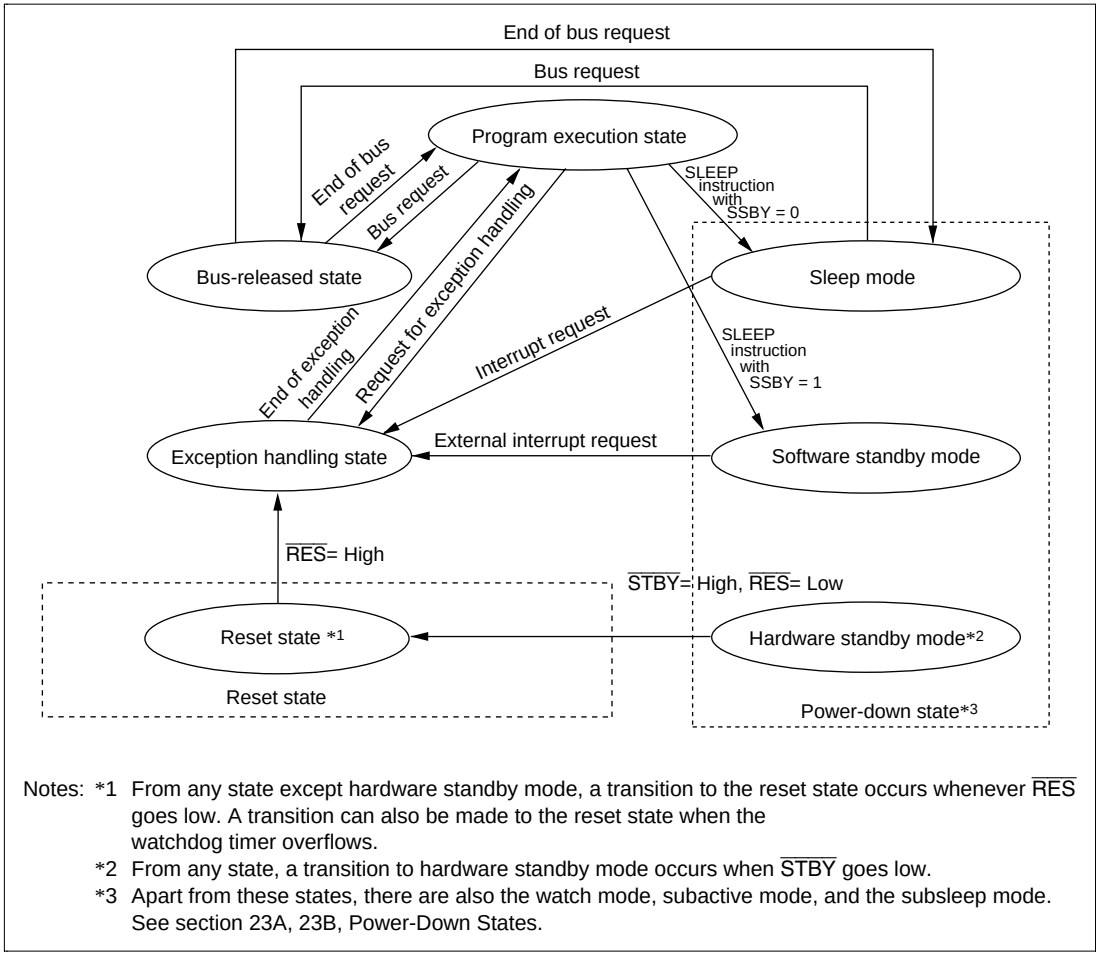


Figure 2-14 Processing States



Notes: *1 From any state except hardware standby mode, a transition to the reset state occurs whenever $\overline{RES}$ goes low. A transition can also be made to the reset state when the watchdog timer overflows.

*2 From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ goes low.

*3 Apart from these states, there are also the watch mode, subactive mode, and the subsleep mode. See section 23A, 23B, Power-Down States.

Figure 2-15 State Transitions

2.8.2 Reset State

When the $\overline{RES}$ goes low, all current processing stops and the CPU enters the reset state. In reset state all interrupts are disenabled.

Reset exception handling starts when the $\overline{RES}$ signal changes from low to high.

The reset state can also be entered by a watchdog timer overflow. For details, refer to section 12 Watchdog Timer.

2.8.3 Exception-Handling State

The exception-handling state is a transient state that occurs when the CPU alters the normal processing flow due to a reset, interrupt, or trap instruction. The CPU fetches a start address (vector) from the exception vector table and branches to that address.

(1) Types of Exception Handling and Their Priority

Exception handling is performed for traces, resets, interrupts, and trap instructions. Table 2-7 indicates the types of exception handling and their priority. Trap instruction exception handling is always accepted, in the program execution state.

Exception handling and the stack structure depend on the interrupt control mode set in SYSCR.

Table 2-7 Exception Handling Types and Priority

Priority	Type of Exception	Detection Timing	Start of Exception Handling
High ↑	Reset	Synchronized with clock	Exception handling starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or when the watchdog timer overflows.
	Trace	End of instruction execution or end of exception-handling sequence*1	When the trace (T) bit is set to 1, the trace starts at the end of the current instruction or current exception-handling sequence
	Interrupt	End of instruction execution or end of exception-handling sequence*2	When an interrupt is requested, exception handling starts at the end of the current instruction or current exception-handling sequence
	Trap instruction	When TRAPA instruction is executed	Exception handling starts when a trap (TRAPA) instruction is executed*3
Low			

Notes: *1 Traces are enabled only in interrupt control mode 2. Trace exception-handling is not executed at the end of the RTE instruction.

*2 Interrupts are not detected at the end of the ANDC, ORC, XORC, and LDC instructions, or immediately after reset exception handling.

*3 Trap instruction exception handling is always accepted, in the program execution state.

(2) Reset Exception Handling

After the $\overline{\text{RES}}$ pin has gone low and the reset state has been entered, when $\overline{\text{RES}}$ goes high again, reset exception handling starts. The CPU enters the reset state when the $\overline{\text{RES}}$ is low. When reset exception handling starts the CPU fetches a start address (vector) from the exception vector table and starts program execution from that address. All interrupts, including NMI, are disabled during reset exception handling and after it ends.

(3) Traces

Traces are enabled only in interrupt control mode 2. Trace mode is entered when the T bit of EXR is set to 1. When trace mode is established, trace exception handling starts at the end of each instruction.

At the end of a trace exception-handling sequence, the T bit of EXR is cleared to 0 and trace mode is cleared. Interrupt masks are not affected.

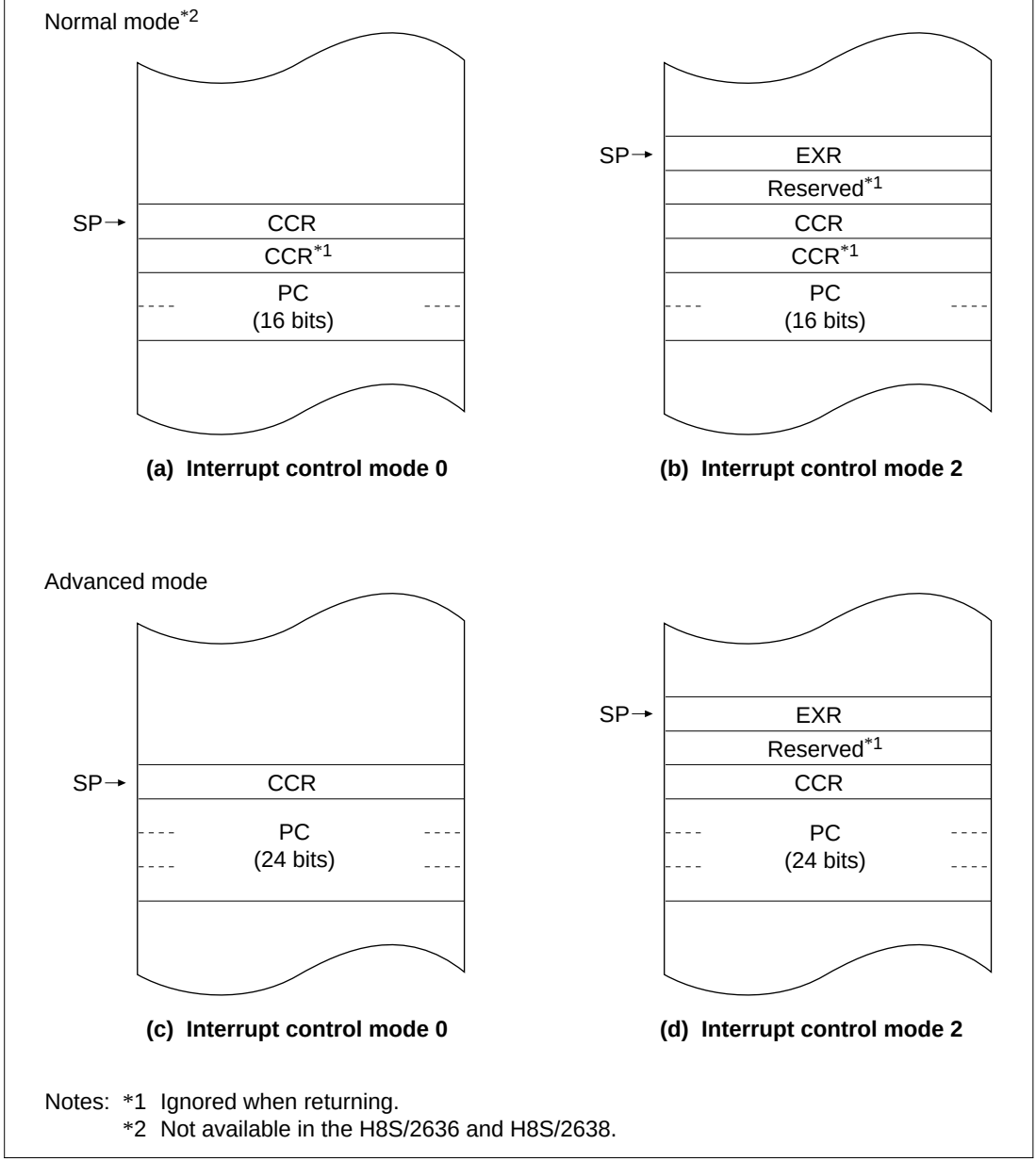
The T bit saved on the stack retains its value of 1, and when the RTE instruction is executed to return from the trace exception-handling routine, trace mode is entered again. Trace exception-handling is not executed at the end of the RTE instruction.

Trace mode is not entered in interrupt control mode 0, regardless of the state of the T bit.

(4) Interrupt Exception Handling and Trap Instruction Exception Handling

When interrupt or trap-instruction exception handling begins, the CPU references the stack pointer (ER7) and pushes the program counter and other control registers onto the stack. Next, the CPU alters the settings of the interrupt mask bits in the control registers. Then the CPU fetches a start address (vector) from the exception vector table and program execution starts from that start address.

Figure 2-16 shows the stack after exception handling ends.



Notes: *1 Ignored when returning.
*2 Not available in the H8S/2636 and H8S/2638.

Figure 2-16 Stack Structure after Exception Handling (Examples)

2.8.4 Program Execution State

In this state the CPU executes program instructions in sequence.

2.8.5 Bus-Released State

This is a state in which the bus has been released in response to a bus request from a bus master other than the CPU. While the bus is released, the CPU halts operations.

Bus masters other than the CPU is data transfer controller (DTC).

For further details, refer to section 7, Bus Controller.

2.8.6 Power-Down State

The power-down state includes both modes in which the CPU stops operating and modes in which the CPU does not stop. There are five modes in which the CPU stops operating: sleep mode, software standby mode, hardware standby mode, subsleep mode, and watch mode. There are also three other power-down modes: medium-speed mode, module stop mode, and subactive mode. In medium-speed mode the CPU and other bus masters operate on a medium-speed clock. Module stop mode permits halting of the operation of individual modules, other than the CPU. Subactive mode, subsleep mode, and watch mode are power-down states using subclock input. For details, refer to section 23A, 23B, Power-Down State.

(1) Sleep Mode: A transition to sleep mode is made if the SLEEP instruction is executed while the software standby bit (SSBY) in the standby control register (SBYCR) is cleared to 0. In sleep mode, CPU operations stop immediately after execution of the SLEEP instruction. The contents of CPU registers are retained.

(2) Software Standby Mode: A transition to software standby mode is made if the SLEEP instruction is executed while the SSBY bit in SBYCR is set to 1, the LSON bit in LPWRCR is set to 0, and the PSS bit in TCSR (WDT1) is set to 0. In software standby mode, the CPU and clock halt and all MCU operations stop. As long as a specified voltage is supplied, the contents of CPU registers and on-chip RAM are retained. The I/O ports also remain in their existing states.

(3) Hardware Standby Mode: A transition to hardware standby mode is made when the $\overline{\text{STBY}}$ pin goes low. In hardware standby mode, the CPU and clock halt and all MCU operations stop. The on-chip supporting modules are reset, but as long as a specified voltage is supplied, on-chip RAM contents are retained.

2.9 Basic Timing

2.9.1 Overview

The H8S/2600 CPU is driven by a system clock, denoted by the symbol ø. The period from one rising edge of ø to the next is referred to as a "state." The memory cycle or bus cycle consists of one, two, or three states. Different methods are used to access on-chip memory, on-chip supporting modules, and the external address space.

2.9.2 On-Chip Memory (ROM, RAM)

On-chip memory is accessed in one state. The data bus is 16 bits wide, permitting both byte and word transfer instruction. Figure 2-17 shows the on-chip memory access cycle. Figure 2-18 shows the pin states.

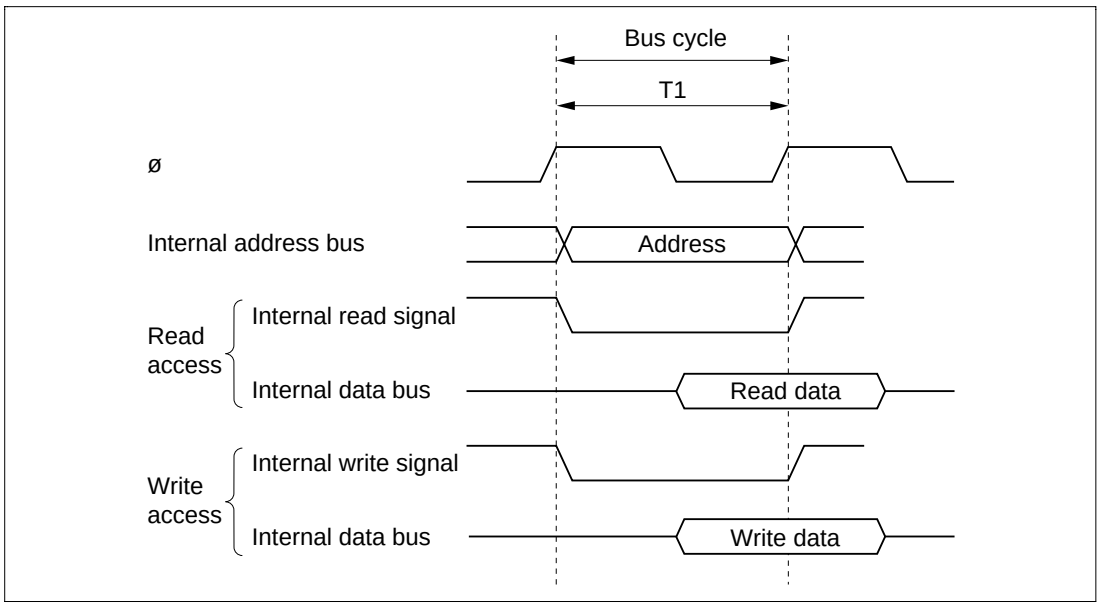


Figure 2-17 On-Chip Memory Access Cycle

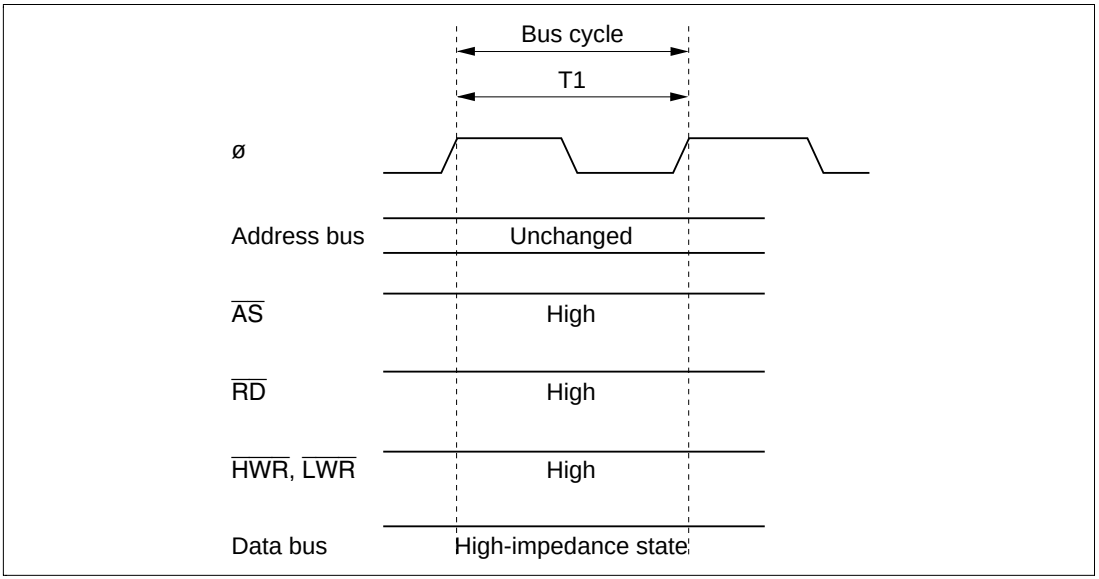


Figure 2-18 Pin States during On-Chip Memory Access

2.9.3 On-Chip Supporting Module Access Timing

The on-chip supporting modules are accessed in two states. The data bus is either 8 bits or 16 bits wide, depending on the particular internal I/O register being accessed. Figure 2-19 shows the access timing for the on-chip supporting modules. Figure 2-20 shows the pin states.

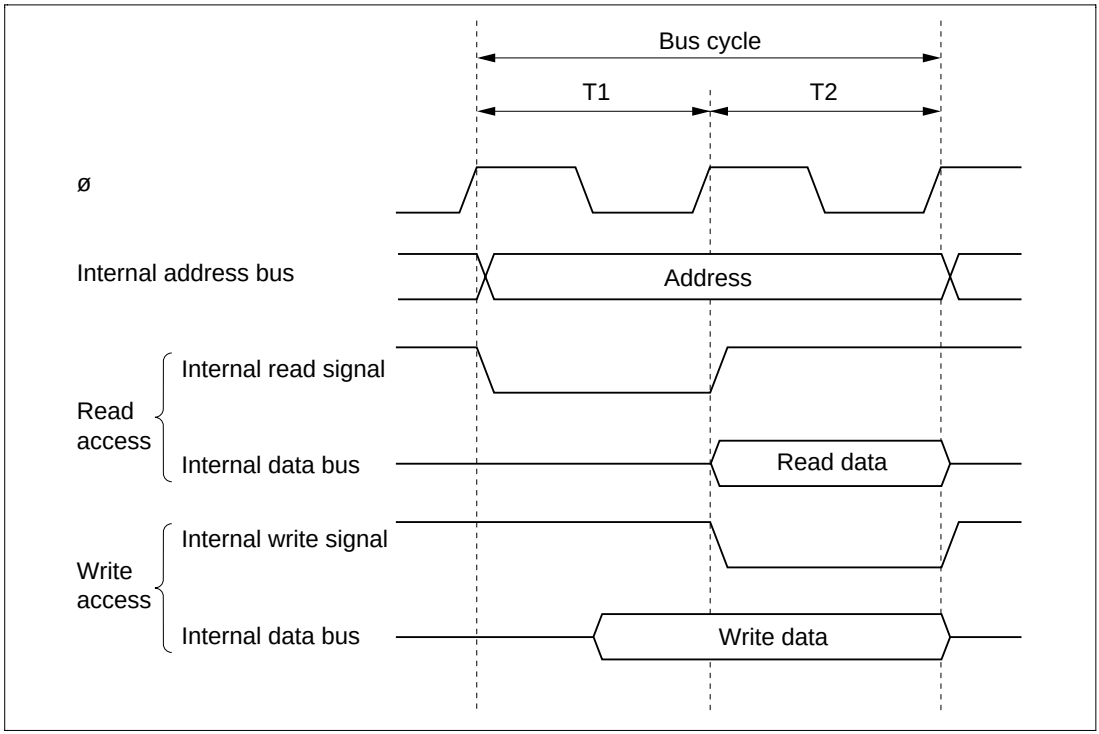


Figure 2-19 On-Chip Supporting Module Access Cycle

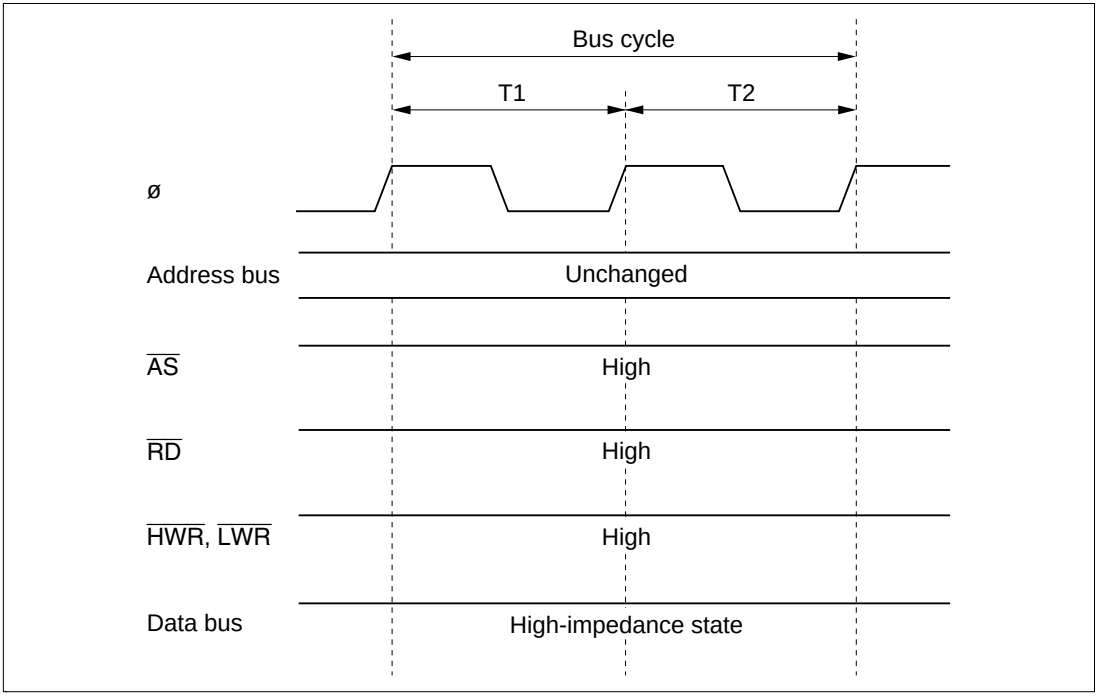


Figure 2-20 Pin States during On-Chip Supporting Module Access

2.9.4 On-Chip HCAN Module Access Timing

On-chip HCAN module access is performed in four states. The data bus width is 16 bits. Wait states can be inserted by means of a wait request from the HCAN. On-chip HCAN module access timing is shown in figures 2-21 and 2-22, and the pin states in figure 2-23.

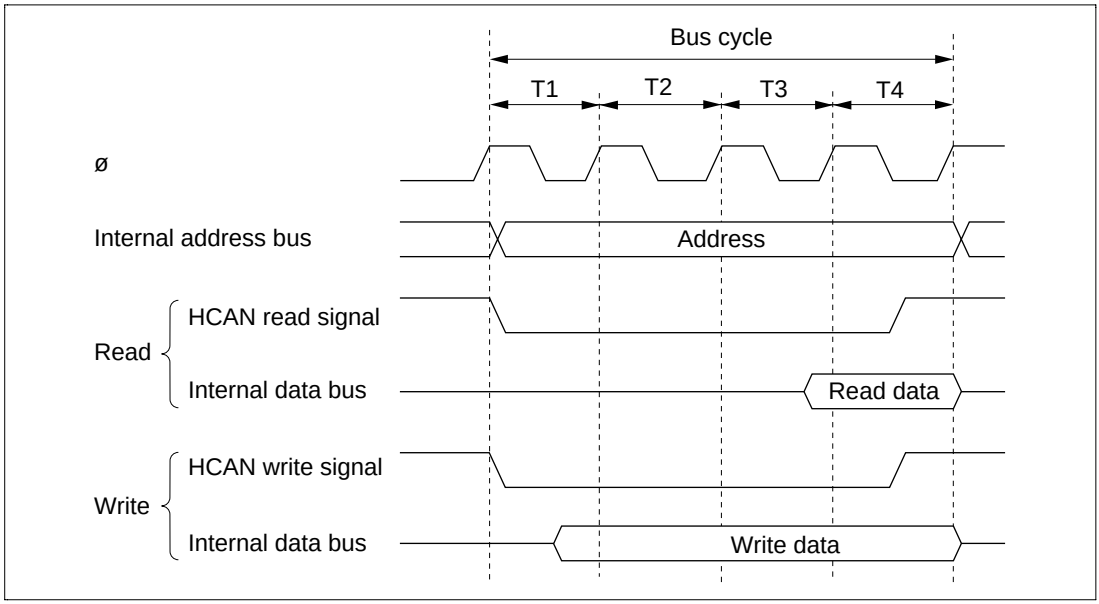


Figure 2-21 On-Chip HCAN Module Access Cycle (No Wait State)

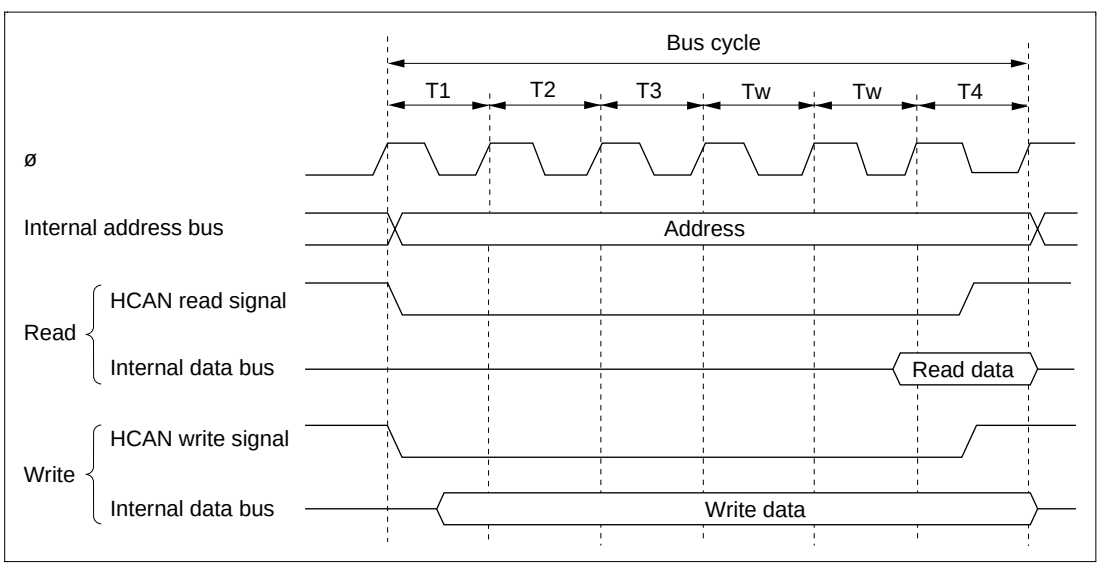


Figure 2-22 On-Chip HCAN Module Access Cycle (Wait States Inserted)

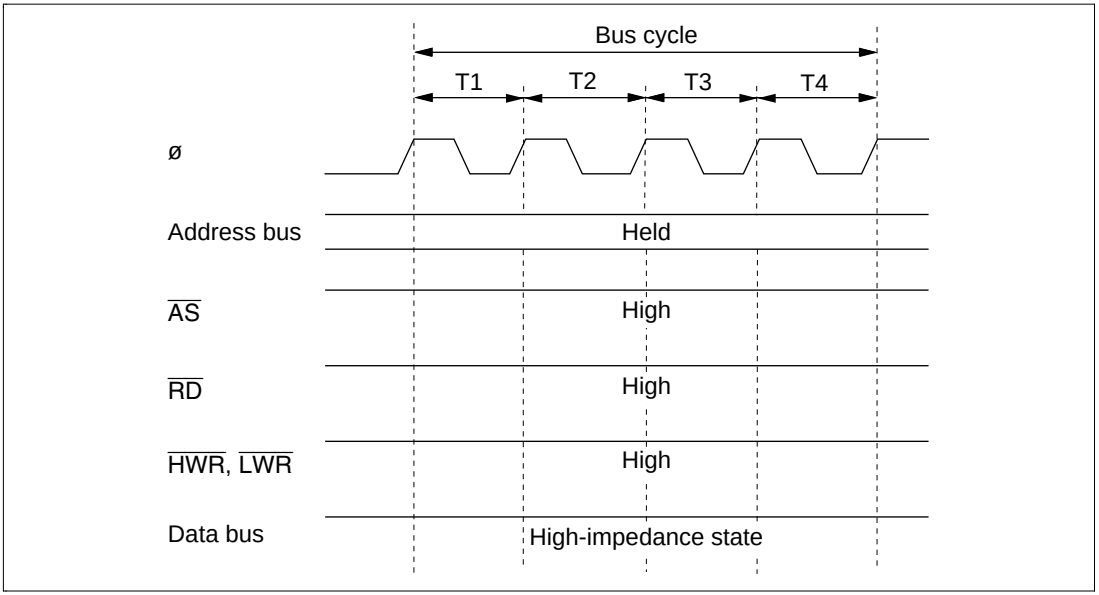


Figure 2-23 Pin States in On-Chip HCAN Module Access

2.9.5 Port H and J Register Access Timing

Accesses to port H and J registers and the on-chip motor control PWM timer module are performed in four states. The data bus width is 8 or 16 bits depending on the internal I/O register.

Access timing for port H and J registers and the on-chip motor control PWM timer module is shown in figure 2-24, and the pin states are shown in figure 2-25.

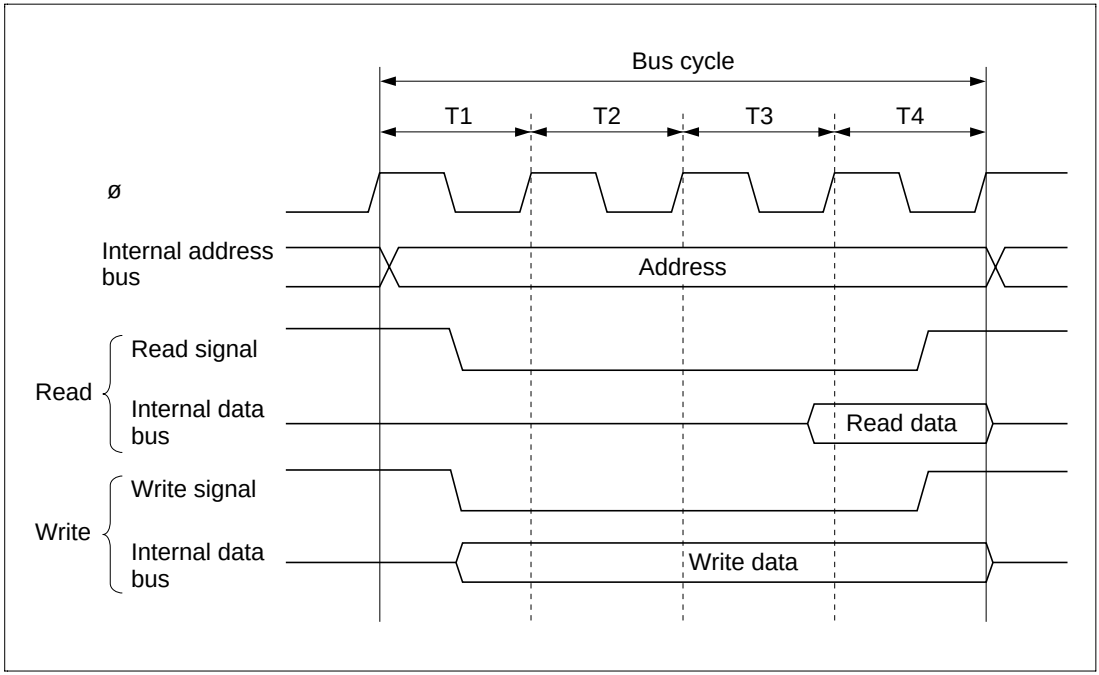


Figure 2-24 Access Cycle for Ports H and J Registers and On-Chip Motor Control PWM Timer Module

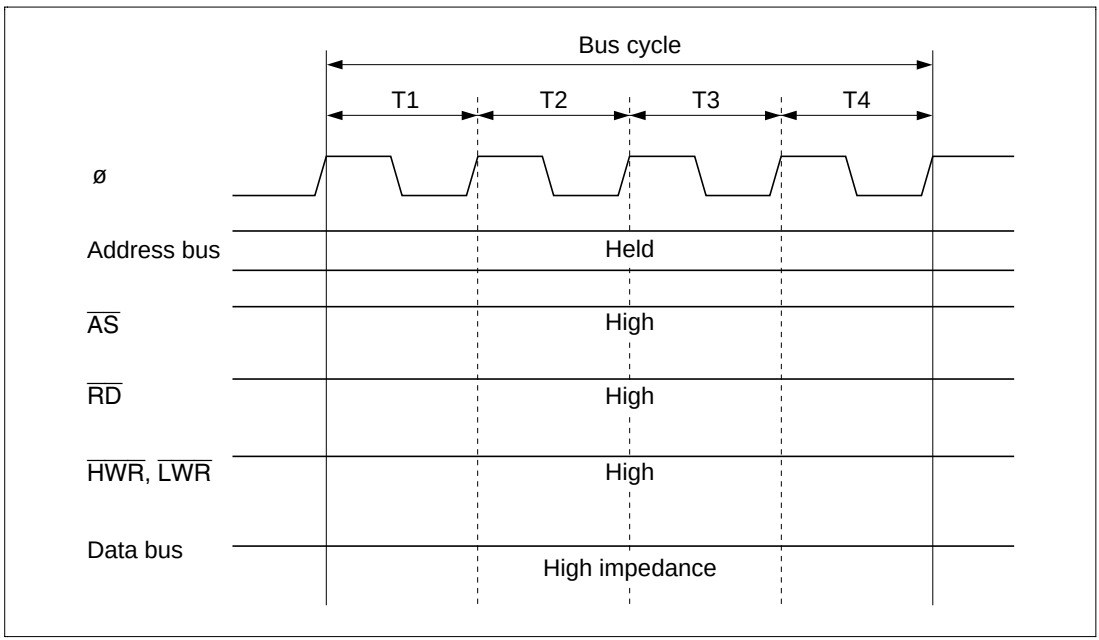


Figure 2-25 Pin States in Access to Ports H and J Registers and On-Chip Motor Control PWM Timer Module

2.9.6 External Address Space Access Timing

The external address space is accessed with an 8-bit or 16-bit data bus width in a two-state or three-state bus cycle. In three-state access, wait states can be inserted. For further details, refer to section 7, Bus Controller.

2.10 Usage Note

2.10.1 TAS Instruction

Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction. The TAS instruction is not generated by the Hitachi H8S and H8/300 series C/C++ compilers. If the TAS instruction is used as a user-defined intrinsic function, ensure that only register ER0, ER1, ER4, or ER5 is used.

2.10.2 STM/LDM Instructions

With STM and LDM instructions, register ER7 cannot be used as a register that can be saved (STM) or restored (LDM) since it is the stack pointer.

The number of registers that can be saved (STM) or restored (LDM) by a single instruction is two, three, or four. The registers that can be used in these cases are as follows.

Two registers: ER0–ER1, ER2–ER3, ER4–ER5
Three registers: ER0–ER2, ER4–ER6
Four registers: ER0–ER3

The Hitachi H8S and H8/300 Series C/C++ compilers do not generate STM/LDM instructions that include ER7.

2.10.3 Caution to observe when using bit manipulation instructions

The BSET, BCLR, BNOT, BST and BIST instructions read data in a unit of byte, then, after bit manipulation, they write data in a unit of byte. Therefore, caution must be exercised when executing any of these instructions for registers and ports that include write-only bits.

The BCLR instruction can be used to clear the flag of an internal I/O register to 0. In that case, if it is clearly known that the pertinent flag is set to 1 in an interrupt processing routine or other processing, there is no need to read the flag in advance.

Section 3 MCU Operating Modes

3.1 Overview

3.1.1 Operating Mode Selection

The H8S/2636 and H8S/2638 have four operating modes (modes 4 to 7). These modes enable selection of the CPU operating mode, enabling/disabling of on-chip ROM, and the initial bus width setting, by setting the mode pins (MD2 to MD0).

Table 3-1 lists the MCU operating modes.

Table 3-1 MCU Operating Mode Selection

MCU Operating Mode	MD2	MD1	MD0	CPU Operating Mode	Description	On-Chip ROM	External Data Bus: Initial Width	External Data Bus: Max. Width
0*	0	0	0	—	—	—	—	—
1*			1	—				
2*		1	0					
3*			1					
4	1	0	0	Advanced	On-chip ROM disabled, expanded mode	Disabled	16 bits	16 bits
5			1				8 bits	16 bits
6		1	0		On-chip ROM enabled, expanded mode	Enabled	8 bits	16 bits
7			1		Single-chip mode		—	—

Note: * Not available in the H8S/2636 and H8S/2638.

The CPU's architecture allows for 4 Gbytes of address space, but the chip actually accesses a maximum of 16 Mbytes.

Modes 4 to 6 are externally expanded modes that allow access to external memory and peripheral devices.

The external expansion modes allow switching between 8-bit and 16-bit bus modes. After program execution starts, an 8-bit or 16-bit address space can be set for each area, depending on the bus controller setting. If 16-bit access is selected for any one area, 16-bit bus mode is set; if 8-bit access is selected for all areas, 8-bit bus mode is set.

Note that the functions of each pin depend on the operating mode.

The H8S/2636 and H8S/2638 can be used only in modes 4 to 7. This means that the mode pins must be set to select one of these modes. Do not change the inputs at the mode pins during operation.

3.1.2 Register Configuration

The H8S/2636 and H8S/2638 have a mode control register (MDCR) that indicates the inputs at the mode pins (MD2 to MD0), and a system control register (SYSCR) that controls the operation of the chip. Table 3-2 summarizes these registers.

Table 3-2 MCU Registers

Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R	Undetermined	H'FDE7
System control register	SYSCR	R/W	H'01	H'FDE5
Pin function control register	PFCR	R/W	H'0D/H'00	H'FDEB

Note: * Lower 16 bits of the address.

3.2 Register Descriptions

3.2.1 Mode Control Register (MDCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	:	1	0	0	0	0	—*	—*	—*
R/W	:	—	—	—	—	—	R	R	R

Note: * Determined by pins MD2 to MD0.

MDCR is an 8-bit read-only register that indicates the current operating mode of the chip.

Bit 7—Reserved: This bit is always read as 1 and cannot be modified.

Bits 6 to 3—Reserved: These bits are always read as 0 and cannot be modified.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to MD2 to MD0. MDS2 to MDS0 are read-only bits, and they cannot be written to. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are cancelled by a reset.

3.2.2 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1	0
		MACS	—	INTM1	INTM0	NMIEG	—	—	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	R/W	R/W	R/W	—	—	R/W

SYSCR is an 8-bit readable-writable register that selects saturating or non-saturating calculation for the MAC instruction, selects the interrupt control mode, selects the detected edge for NMI, and enables or disenables on-chip RAM.

SYSCR is initialized to H'01 by a reset and in hardware standby mode. SYSCR is not initialized in software standby mode.

Bit 7—MAC Saturation (MACS): Selects either saturating or non-saturating calculation for the MAC instruction.

Bit 7		
MACS	**Description**	
0	Non-saturating calculation for MAC instruction	(Initial value)
1	Saturating calculation for MAC instruction	

Bit 6—Reserved: This bit is always read as 0 and cannot be modified.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select the control mode of the interrupt controller. For details of the interrupt control modes, see section 5.4.1, Interrupt Control Modes and Interrupt Operation.

Bit 5	Bit 4	Interrupt		
INTM1	**INTM0**	**Control Mode**	**Description**	
0	0	0	Control of interrupts by I bit	(Initial value)
	1	—	Setting prohibited	
1	0	2	Control of interrupts by I2 to I0 bits and IPR	
	1	—	Setting prohibited	

Bit 3—NMI Edge Select (NMIEG): Selects the valid edge of the NMI interrupt input.

Bit 3

NMIEG	Description	
0	An interrupt is requested at the falling edge of NMI input	(Initial value)
1	An interrupt is requested at the rising edge of NMI input	

Bit 2— Reserved: Only 0 should be written to this bit.

Bit 1—Reserved: This bit is always read as 0 and cannot be modified.

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized when the reset status is released. It is not initialized in software standby mode.

Bit 0

RAME	Description	
0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

Note: When the DTC is used, the RAME bit must not be cleared to 0.

3.2.3 Pin Function Control Register (PFCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	AE3	AE2	AE1	AE0
Initial value	:	0	0	0	0	1/0	1/0	0	1/0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFCR is an 8-bit readable-writeable register that performs address output control in extension modes involving ROM.

PFCR is initialized to H'0D/H'00 by a reset and in the hardware standby mode.

Bits 7 to 4— Reserved: Only 0 should be written to these bits.

Bits 3 to 0—Address Output Enable 3 to 0 (AE3–AE0): These bits select enabling or disabling of address outputs A8 to A23 in ROMless expanded mode and modes with ROM. When a pin is enabled for address output, the address is output regardless of the corresponding DDR setting. When a pin is disabled for address output, it becomes an output port when the corresponding DDR bit is set to 1.

Bit 3	Bit 2	Bit 1	Bit 0	
AE3	**AE2**	**AE1**	**AE0**	**Description**
0	0	0	0	A8–A23 address output disabled (Initial value*)
			1	A8 address output enabled; A9–A23 address output disabled
		1	0	A8, A9 address output enabled; A10–A23 address output disabled
			1	A8–A10 address output enabled; A11–A23 address output disabled
	1	0	0	A8–A11 address output enabled; A12–A23 address output disabled
			1	A8–A12 address output enabled; A13–A23 address output disabled
		1	0	A8–A13 address output enabled; A14–A23 address output disabled
			1	A8–A14 address output enabled; A15–A23 address output disabled
1	0	0	0	A8–A15 address output enabled; A16–A23 address output disabled
			1	A8–A16 address output enabled; A17–A23 address output disabled
		1	0	A8–A17 address output enabled; A18–A23 address output disabled
			1	A8–A18 address output enabled; A19–A23 address output disabled
	1	0	0	A8–A19 address output enabled; A20–A23 address output disabled
			1	A8–A20 address output enabled; A21–A23 address output disabled (Initial value*)
		1	0	A8–A21 address output enabled; A22, A23 address output disabled
			1	A8–A23 address output enabled

Note: * In expanded mode with ROM, bits AE3 to AE0 are initialized to B'0000.

In ROMless expanded mode, bits AE3 to AE0 are initialized to B'1101.

Address pins A0 to A7 are made address outputs by setting the corresponding DDR bits to 1.

3.3 Operating Mode Descriptions

3.3.1 Mode 4

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.

Ports A, B, and C, function as an address bus, ports D and E function as a data bus, and part of port F carries bus control signals.

The initial bus mode after a reset is 16 bits, with 16-bit access to all areas. However, note that if 8-bit access is designated by the bus controller for all areas, the bus mode switches to 8 bits.

3.3.2 Mode 5

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is disabled.

Ports A, B, and C, function as an address bus, ports D and E function as a data bus, and part of port F carries bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, note that if 16-bit access is designated by the bus controller for any area, the bus mode switches to 16 bits and port E becomes a data bus.

3.3.3 Mode 6

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled.

Ports A, B, and C, function as input port pins immediately after a reset. Address output can be performed by setting the corresponding DDR (data direction register) bits to 1.

Port D functions as a data bus, and part of port F carries bus control signals.

The initial bus mode after a reset is 8 bits, with 8-bit access to all areas. However, note that if 16-bit access is designated by the bus controller for any area, the bus mode switches to 16 bits and port E becomes a data bus.

3.3.4 Mode 7

The CPU can access a 16-Mbyte address space in advanced mode. The on-chip ROM is enabled, but external addresses cannot be accessed.

All I/O ports are available for use as input-output ports.

3.4 Pin Functions in Each Operating Mode

The pin functions of ports A to F vary depending on the operating mode. Table 3-3 shows their functions in each operating mode.

Table 3-3 Pin Functions in Each Mode

Port		Mode 4	Mode 5	Mode 6	Mode 7
Port A		A	A	P*/A	P
Port B		A	A	P*/A	P
Port C		A	A	P*/A	P
Port D		D	D	D	P
Port E		P/D*	P*/D	P*/D	P
Port F	PF7	P/C*	P/C*	P/C*	P*/C
	PF6 to PF4	C	C	C	P
	PF3, PF2	P/C*	P*/C	P*/C	
	PF0	P*/C	P*/C	P*/C	

Legend
P: I/O port
A: Address bus output
D: Data bus I/O
C: Control signals, clock I/O
*: After reset

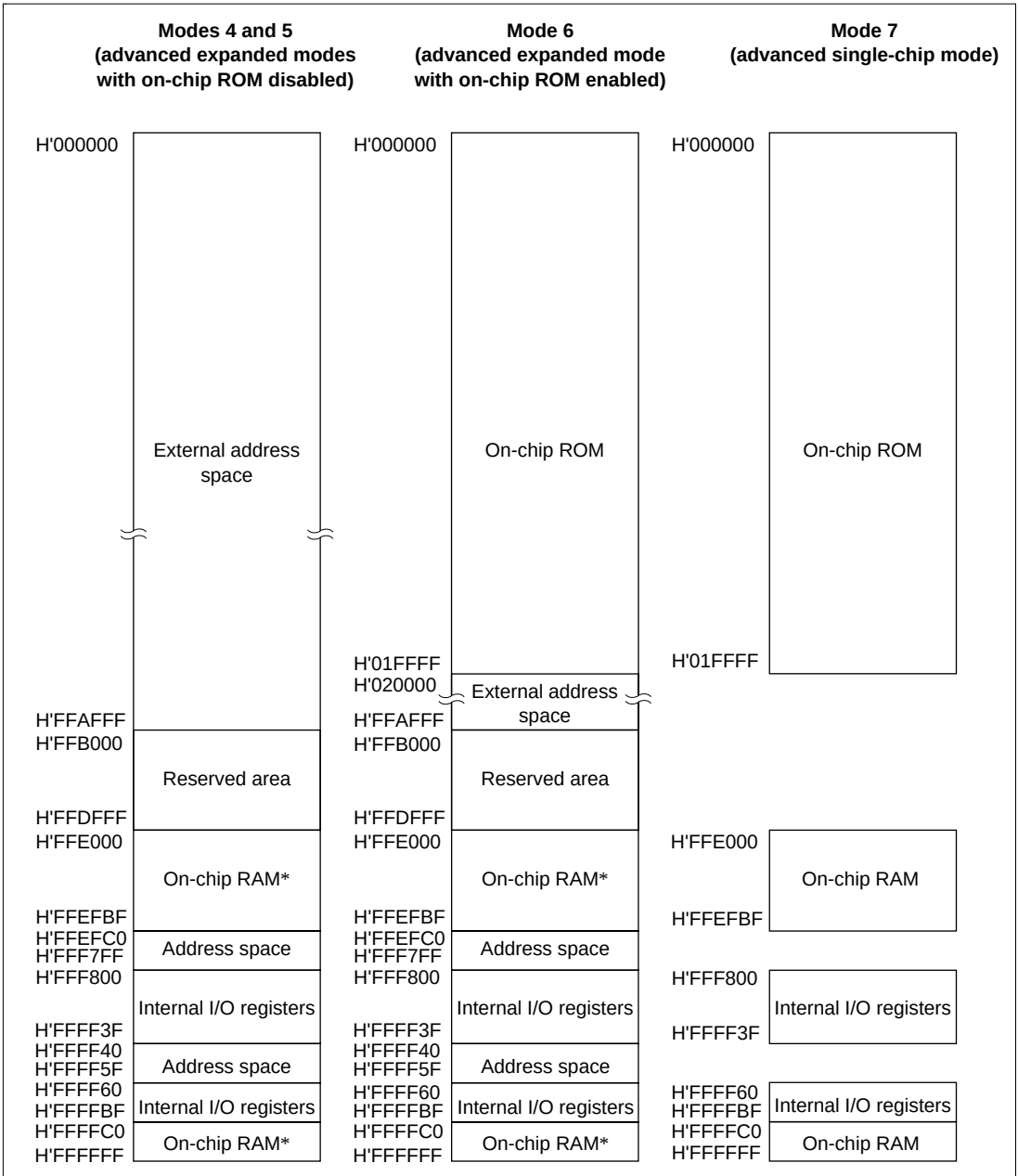
3.5 Address Map in Each Operating Mode

A address map of the H8S/2636 is shown in figure 3-1.

A address map of the H8S/2638 is shown in figure 3-2.

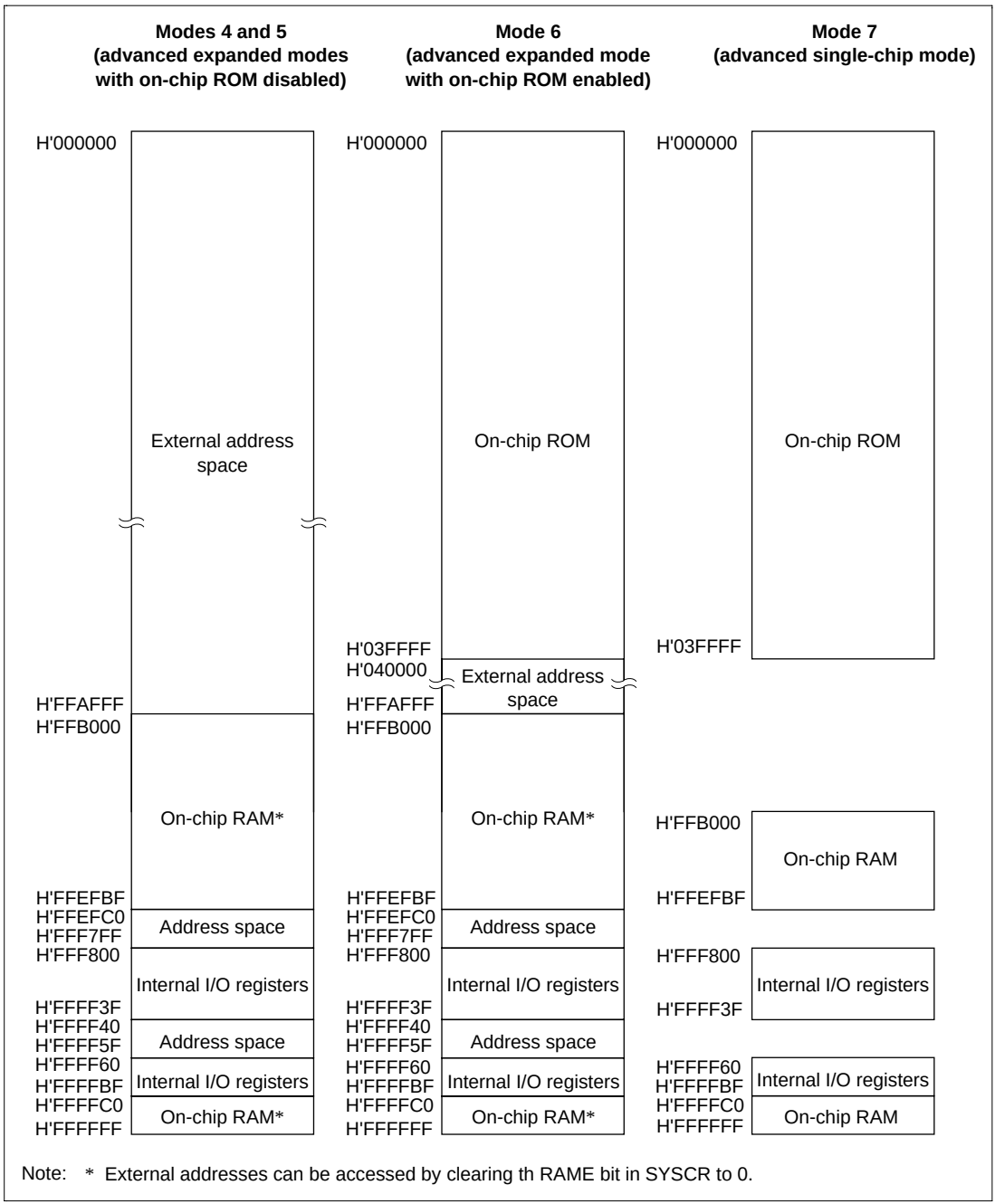
The address space is 16 Mbytes in modes 4 to 7 (advanced modes).

The address space is divided into eight areas for modes 4 to 7. For details, see section 7, Bus Controller.



Note: * External addresses can be accessed by clearing th RAME bit in SYSCR to 0.

Figure 3-1 Memory Map in Each Operating Mode in the H8S/2636



Note: * External addresses can be accessed by clearing th RAME bit in SYSCR to 0.

Figure 3-2 Memory Map in Each Operating Mode in the H8S/2638 and H8S/2639

Section 4 Exception Handling

4.1 Overview

4.1.1 Exception Handling Types and Priority

As table 4-1 indicates, exception handling may be caused by a reset, direct transition*4, trap instruction, or interrupt. Exception handling is prioritized as shown in table 4-1. If two or more exceptions occur simultaneously, they are accepted and processed in order of priority. Trap instruction exceptions are accepted at all times, in the program execution state.

Exception handling sources, the stack structure, and the operation of the CPU vary depending on the interrupt control mode set by the INTM0 and INTM1 bits of SYSCR.

Table 4-1 Exception Types and Priority

Priority	Exception Type	Start of Exception Handling
High ↑	Reset	Starts immediately after a low-to-high transition at the $\overline{\text{RES}}$ pin, or when the watchdog overflows. The CPU enters the reset state when the $\overline{\text{RES}}$ pin is low.
	Trace*1	Starts when execution of the current instruction or exception handling ends, if the trace (T) bit is set to 1
	Direct transition*4	Starts when a direct transition occurs due to execution of a SLEEP instruction.
	Interrupt	Starts when execution of the current instruction or exception handling ends, if an interrupt request has been issued*2
Low	Trap instruction (TRAPA)*3	Started by execution of a trap instruction (TRAPA)

Notes: *1 Traces are enabled only in interrupt control mode 2. Trace exception handling is not executed after execution of an RTE instruction.

*2 Interrupt detection is not performed on completion of ANDC, ORC, XORC, or LDC instruction execution, or on completion of reset exception handling.

*3 Trap instruction exception handling requests are accepted at all times in program execution state.

*4 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only. These functions cannot be used with the other versions.

4.1.2 Exception Handling Operation

Exceptions originate from various sources. Trap instructions and interrupts are handled as follows:

1. The program counter (PC), condition code register (CCR), and extended register (EXR) are pushed onto the stack.
2. The interrupt mask bits are updated. The T bit is cleared to 0.
3. A vector address corresponding to the exception source is generated, and program execution starts from that address.

For a reset exception, steps 2 and 3 above are carried out.

4.1.3 Exception Vector Table

The exception sources are classified as shown in figure 4-1. Different vector addresses are assigned to different exception sources.

Table 4-2 lists the exception sources and their vector addresses.

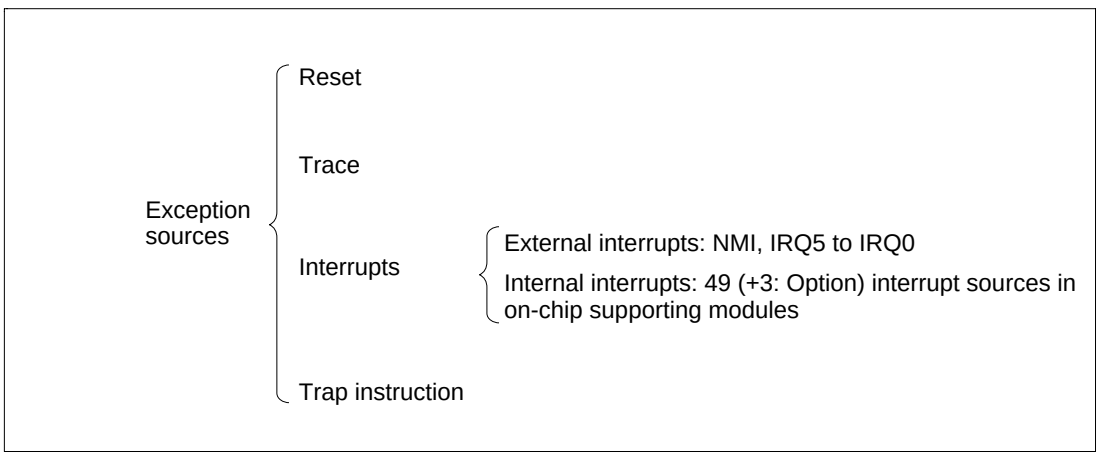


Figure 4-1 Exception Sources

Table 4-2 Exception Vector Table

Exception Source		Vector Number	Vector Address*[1] Advanced Mode
Reset		0	H'0000 to H'0003
Reserved for system use		1	H'0004 to H'0007
		2	H'0008 to H'000B
		3	H'000C to H'000F
		4	H'0010 to H'0013
Trace		5	H'0014 to H'0017
Direct Transition*[3]		6	H'0018 to H'001B
External interrupt	NMI	7	H'001C to H'001F
Trap instruction (4 sources)		8	H'0020 to H'0023
		9	H'0024 to H'0027
		10	H'0028 to H'002B
		11	H'002C to H'002F
Reserved for system use		12	H'0030 to H'0033
		13	H'0034 to H'0037
		14	H'0038 to H'003B
		15	H'003C to H'003F
External interrupt	IRQ0	16	H'0040 to H'0043
	IRQ1	17	H'0044 to H'0047
	IRQ2	18	H'0048 to H'004B
	IRQ3	19	H'004C to H'004F
	IRQ4	20	H'0050 to H'0053
	IRQ5	21	H'0054 to H'0057
Reserved for system use		22	H'0058 to H'005B
		23	H'005C to H'005F
Internal interrupt*[2]		24 | 127	H'0060 to H'0063 | H'01FC to H'01FF

Notes: *1 Lower 16 bits of the address.

*2 For details of internal interrupt vectors, see section 5.3.3, Interrupt Exception Handling Vector Table.

*3 See 23B.11 Direct Transition for details on direct transition.
Subclock functions are available in the U-mask and W-mask versions of the H8S/2636, H8S/2638, and H8S/2639 only.

4.2 Reset

4.2.1 Overview

A reset has the highest exception priority.

When the $\overline{\text{RES}}$ pin goes low, all current operations are stopped, and this LSI enters reset state. A reset initializes the internal state of the CPU and the registers of on-chip supporting modules. Immediately after a reset, interrupt control mode 0 is set.

When the $\overline{\text{RES}}$ pin goes from low to high, reset exception handling starts.

The H8S/2636 can also be reset by overflow of the watchdog timer. For details see section 12, Watchdog Timer.

4.2.2 Reset Sequence

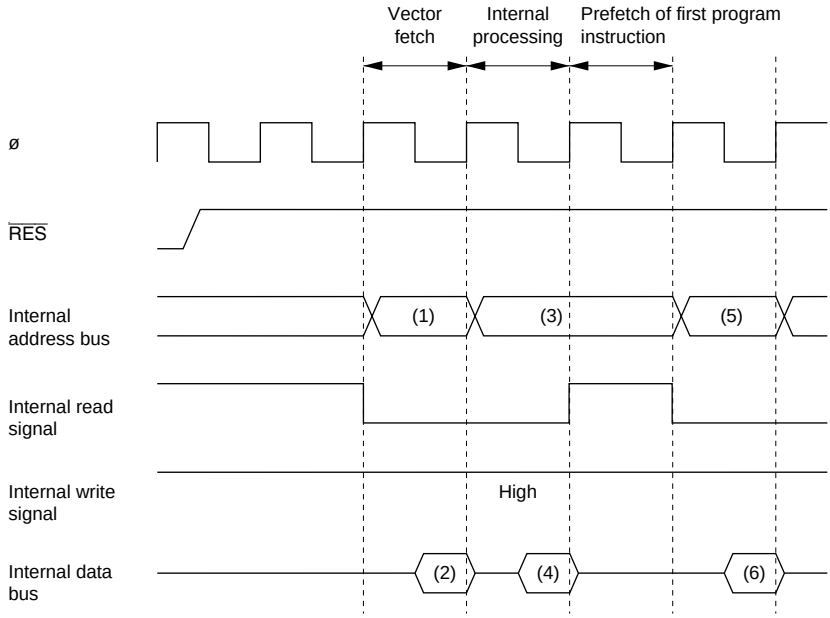
This LSI enters reset state when the $\overline{\text{RES}}$ pin goes low.

To ensure that this LSI is reset, hold the $\overline{\text{RES}}$ pin low for at least 20 ms at power-up. To reset during operation, hold the $\overline{\text{RES}}$ pin low for at least 20 states.

When the $\overline{\text{RES}}$ pin goes high after being held low for the necessary time, this LSI starts reset exception handling as follows.

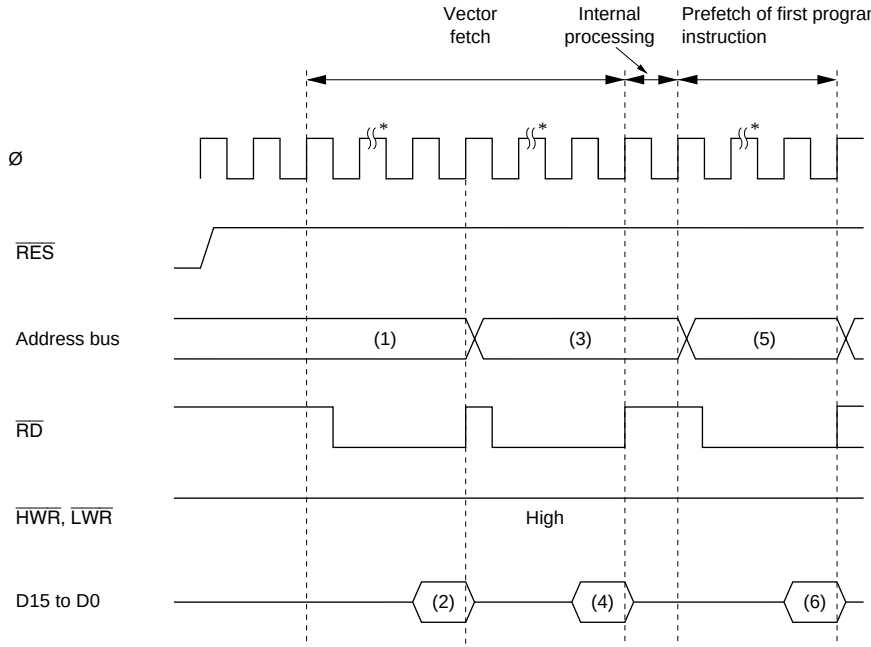
1. The internal state of the CPU and the registers of the on-chip supporting modules are initialized, the T bit is cleared to 0 in EXR, and the I bit is set to 1 in EXR and CCR.
2. The reset exception handling vector address is read and transferred to the PC, and program execution starts from the address indicated by the PC.

Figures 4-2 and 4-3 show examples of the reset sequence.



(1) (3) Reset exception handling vector address (when power-on reset, (1) = H'000000, (3) = H'000002)
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Figure 4-2 Reset Sequence (Modes 6 and 7)



(1) (3) Reset exception handling vector address (when reset, (1) = H'000000, (3) = H'000002)
(2) (4) Start address (contents of reset exception handling vector address)
(5) Start address ((5) = (2) (4))
(6) First program instruction

Note: * 3 program wait states are inserted.

Figure 4-3 Reset Sequence (Mode 4)

4.2.3 Interrupts after Reset

If an interrupt is accepted after a reset but before the stack pointer (SP) is initialized, the PC and CCR will not be saved correctly, leading to a program crash. To prevent this, all interrupt requests, including NMI, are disabled immediately after a reset. Since the first instruction of a program is always executed immediately after the reset state ends, make sure that this instruction initializes the stack pointer (example: MOV.L #xx: 32, SP).

4.2.4 State of On-Chip Supporting Modules after Reset Release

After reset release, MSTPCRA to MSTPCRD are initialized to H'3F, H'FF, H'FF, and B'11******[*], respectively, and all modules except the DTC, enter module stop mode. Consequently, on-chip supporting module registers cannot be read or written to. Register reading and writing is enabled when module stop mode is exited.

Note: * The value of bits 5 to 0 is undefined.

4.3 Traces

Traces are enabled in interrupt control mode 2. Trace mode is not activated in interrupt control mode 0, irrespective of the state of the T bit. For details of interrupt control modes, see section 5, Interrupt Controller.

If the T bit in EXR is set to 1, trace mode is activated. In trace mode, a trace exception occurs on completion of each instruction.

Trace mode is canceled by clearing the T bit in EXR to 0. It is not affected by interrupt masking.

Table 4-3 shows the state of CCR and EXR after execution of trace exception handling.

Interrupts are accepted even within the trace exception handling routine.

The T bit saved on the stack retains its value of 1, and when control is returned from the trace exception handling routine by the RTE instruction, trace mode resumes.

Trace exception handling is not carried out after execution of the RTE instruction.

Table 4-3 Status of CCR and EXR after Trace Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	Trace exception handling cannot be used.			
2	1	—	—	0

Legend

1: Set to 1

0: Cleared to 0

—: Retains value prior to execution.

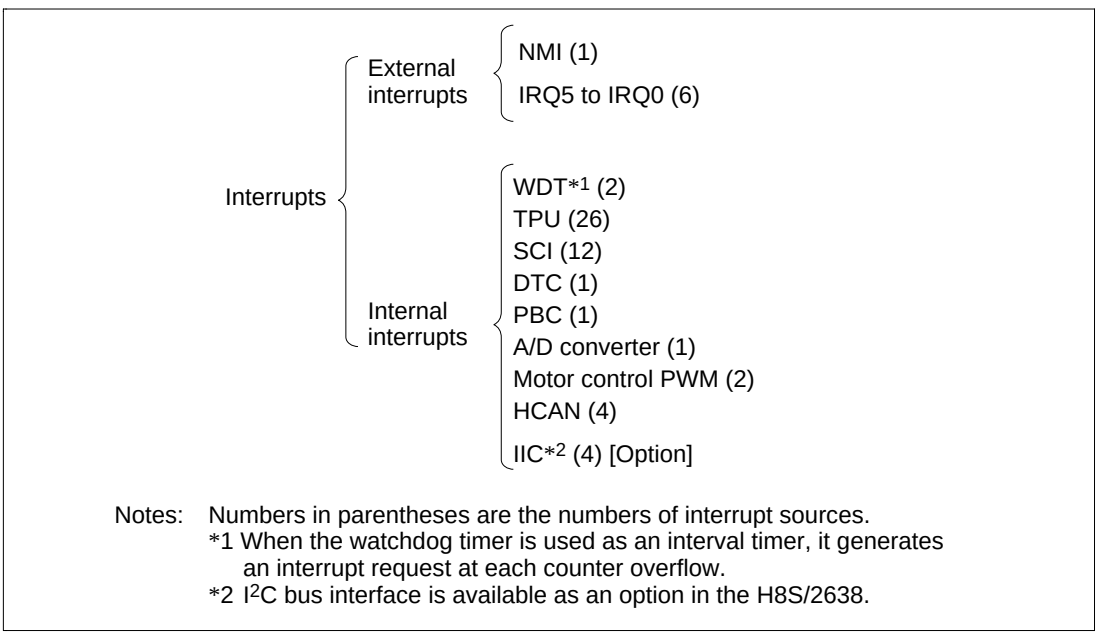
4.4 Interrupts

Interrupt exception handling can be requested by seven external sources (NMI, IRQ5 to IRQ0) and 49 internal sources in the on-chip supporting modules. Figure 4-4 classifies the interrupt sources and the number of interrupts of each type.

The on-chip supporting modules that can request interrupts include the watchdog timer (WDT), 16-bit timer-pulse unit (TPU), serial communication interface (SCI), data transfer controller (DTC), PC break controller (PBC), A/D converter, Hitachi controller area network (HCAN), motor control PWM timer, and I^2C bus interface (IIC). Each interrupt source has a separate vector address.

NMI is the highest-priority interrupt. Interrupts are controlled by the interrupt controller. The interrupt controller has two interrupt control modes and can assign interrupts other than NMI to eight priority/mask levels to enable multiplexed interrupt control.

For details of interrupts, see section 5, Interrupt Controller.



Notes: Numbers in parentheses are the numbers of interrupt sources.
*1 When the watchdog timer is used as an interval timer, it generates an interrupt request at each counter overflow.
*2 I^2C bus interface is available as an option in the H8S/2638.

Figure 4-4 Interrupt Sources and Number of Interrupts

4.5 Trap Instruction

Trap instruction exception handling starts when a TRAPA instruction is executed. Trap instruction exception handling can be executed at all times in the program execution state.

The TRAPA instruction fetches a start address from a vector table entry corresponding to a vector number from 0 to 3, as specified in the instruction code.

Table 4-4 shows the status of CCR and EXR after execution of trap instruction exception handling.

Table 4-4 Status of CCR and EXR after Trap Instruction Exception Handling

	CCR		EXR	
Interrupt Control Mode	**I**	**UI**	**I2 to I0**	**T**
0	1	—	—	—
2	1	—	—	0

Legend
1: Set to 1
0: Cleared to 0
—: Retains value prior to execution.

4.6 Stack Status after Exception Handling

Figure 4-5 shows the stack after completion of trap instruction exception handling and interrupt exception handling.

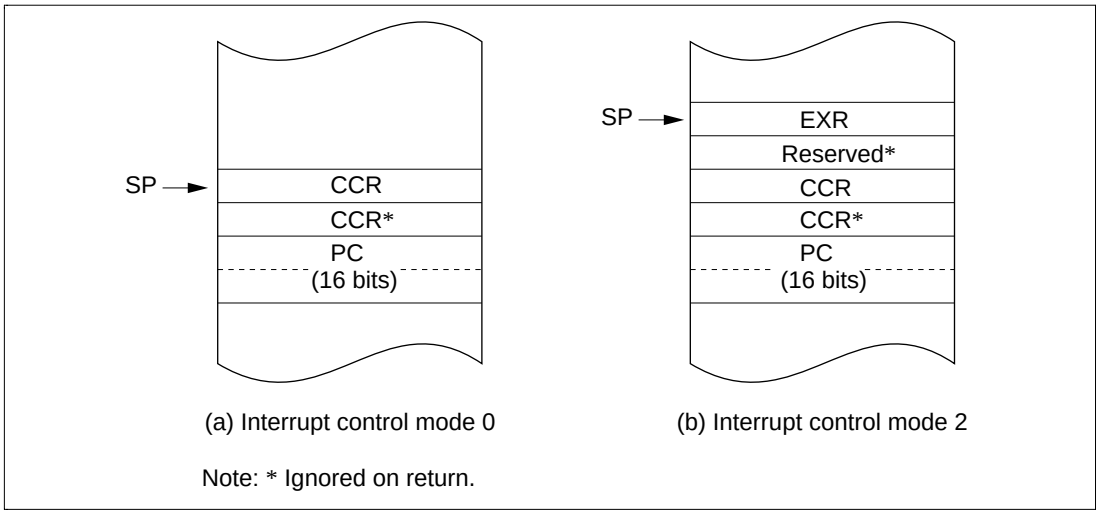


Figure 4-5 (1) Stack Status after Exception Handling (Normal Modes: Not Available in the H8S/2636 and H8S/2638)

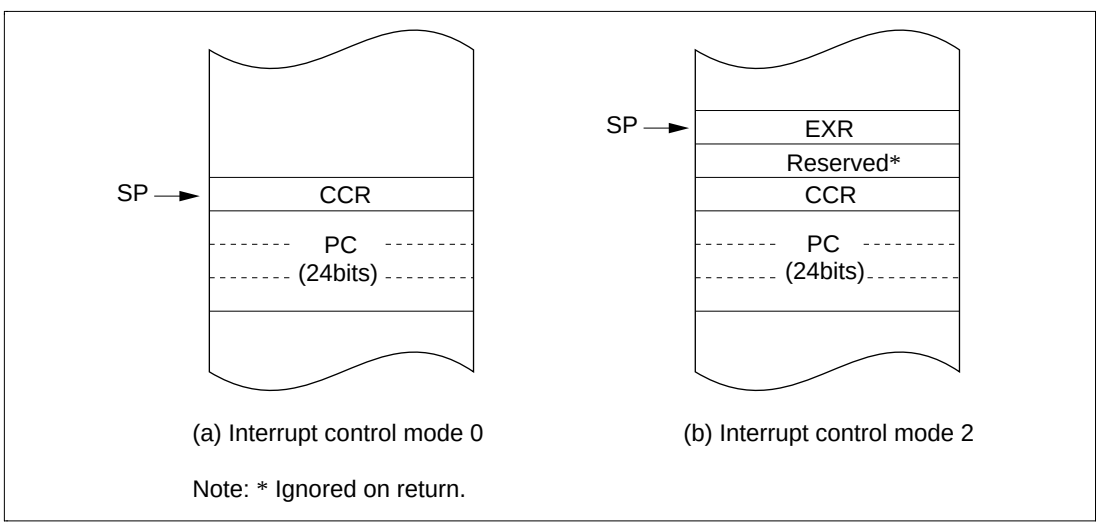


Figure 4-5 (2) Stack Status after Exception Handling (Advanced Modes)

4.7 Notes on Use of the Stack

When accessing word data or longword data, the chip assumes that the lowest address bit is 0. The stack should always be accessed by word transfer instruction or longword transfer instruction, and the value of the stack pointer (SP, ER7) should always be kept even. Use the following instructions to save registers:

```
PUSH.W   Rn    (or MOV.W Rn, @-SP)
PUSH.L   ERn   (or MOV.L ERn, @-SP)
```

Use the following instructions to restore registers:

```
POP.W    Rn    (or MOV.W @SP+, Rn)
POP.L    ERn   (or MOV.L @SP+, ERn)
```

Setting SP to an odd value may lead to a malfunction. Figure 4-6 shows an example of what happens when the SP value is odd.

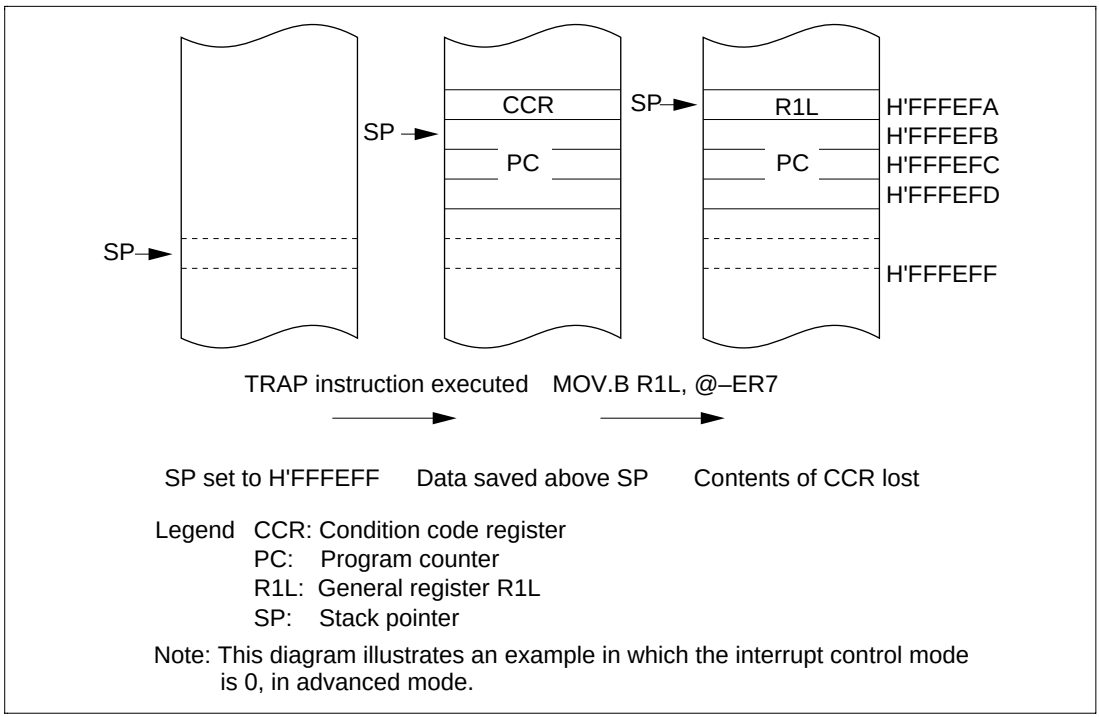


Figure 4-6 Operation when SP Value is Odd

Section 5 Interrupt Controller

5.1 Overview

5.1.1 Features

The H8S/2636 and H8S/2638 control interrupts by means of an interrupt controller. The interrupt controller has the following features:

- Two interrupt control modes
  - — Any of two interrupt control modes can be set by means of the INTM1 and INTM0 bits in the system control register (SYSCR).
- Priorities settable with IPR
  - — An interrupt priority register (IPR) is provided for setting interrupt priorities. Eight priority levels can be set for each module for all interrupts except NMI.
  - — NMI is assigned the highest priority level of 8, and can be accepted at all times.
- Independent vector addresses
  - — All interrupt sources are assigned independent vector addresses, making it unnecessary for the source to be identified in the interrupt handling routine.
- Seven external interrupts
  - — NMI is the highest-priority interrupt, and is accepted at all times. Rising edge or falling edge can be selected for NMI.
  - — Falling edge, rising edge, or both edge detection, or level sensing, can be selected for IRQ5 to IRQ0.
- DTC control
  - — DTC activation is performed by means of interrupts.

5.1.2 Block Diagram

A block diagram of the interrupt controller is shown in Figure 5-1.

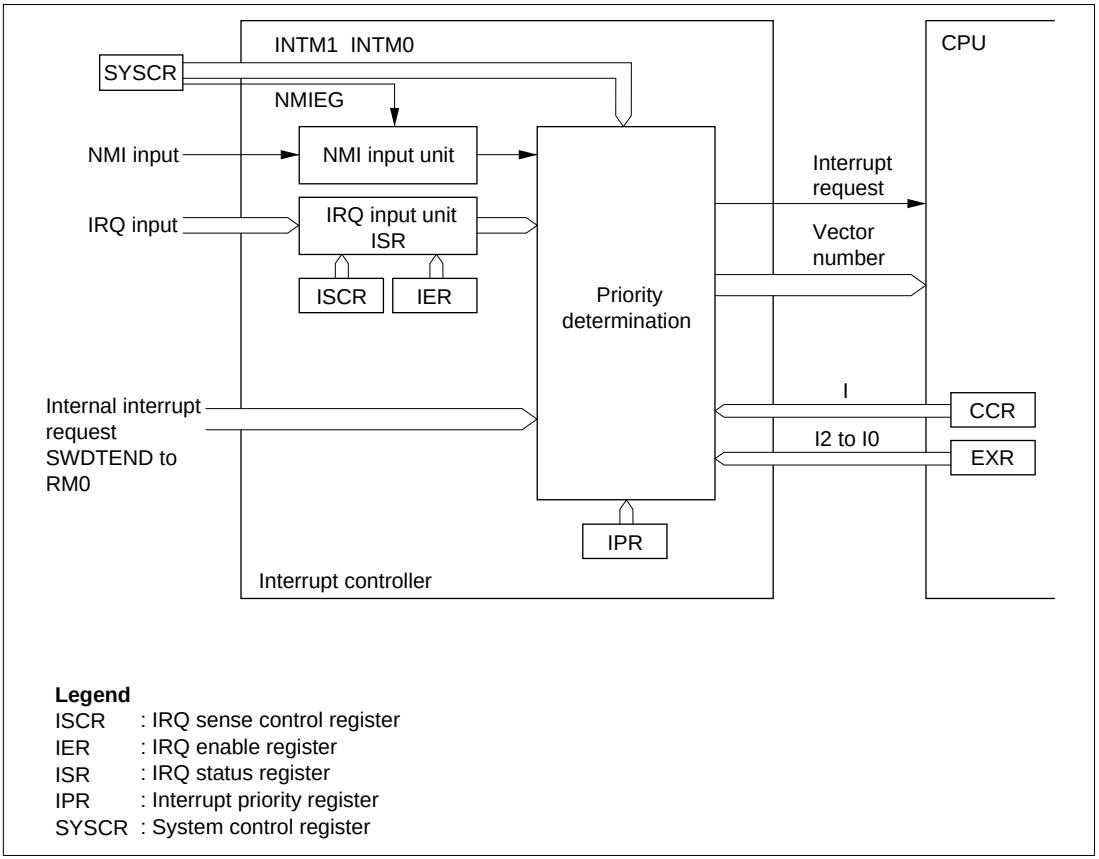


Figure 5-1 Block Diagram of Interrupt Controller

5.1.3 Pin Configuration

Table 5-1 summarizes the pins of the interrupt controller.

Table 5-1 Interrupt Controller Pins

Name	Symbol	I/O	Function
Nonmaskable interrupt	NMI	Input	Nonmaskable external interrupt; rising or falling edge can be selected
External interrupt requests 5 to 0	$\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$	Input	Maskable external interrupts; rising, falling, or both edges, or level sensing, can be selected

5.1.4 Register Configuration

Table 5-2 summarizes the registers of the interrupt controller.

Table 5-2 Interrupt Controller Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
System control register	SYSCR	R/W	H'01	H'FDE5
IRQ sense control register H	ISCRH	R/W	H'00	H'FE12
IRQ sense control register L	ISCRL	R/W	H'00	H'FE13
IRQ enable register	IER	R/W	H'00	H'FE14
IRQ status register	ISR	R/(W)*[2]	H'00	H'FE15
Interrupt priority register A	IPRA	R/W	H'77	H'FEC0
Interrupt priority register B	IPRB	R/W	H'77	H'FEC1
Interrupt priority register C	IPRC	R/W	H'77	H'FEC2
Interrupt priority register D	IPRD	R/W	H'77	H'FEC3
Interrupt priority register E	IPRE	R/W	H'77	H'FEC4
Interrupt priority register F	IPRF	R/W	H'77	H'FEC5
Interrupt priority register G	IPRG	R/W	H'77	H'FEC6
Interrupt priority register H	IPRH	R/W	H'77	H'FEC7
Interrupt priority register J	IPRJ	R/W	H'77	H'FEC9
Interrupt priority register K	IPRK	R/W	H'77	H'FECA
Interrupt priority register L	IPRL	R/W	H'77	H'FECB
Interrupt priority register M	IPRM	R/W	H'77	H'FECC

Notes: *1 Lower 16 bits of the address.
*2 Can only be written with 0 for flag clearing.

5.2 Register Descriptions

5.2.1 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1	0
		MACS	—	INTM1	INTM0	NMIEG	—	—	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—	R/W

SYSCR is an 8-bit readable/writable register that selects the interrupt control mode, and the detected edge for NMI.

Only bits 5 to 3 are described here; for details of the other bits, see section 3.2.2, System Control Register (SYSCR).

SYSCR is initialized to H'01 by a reset and in hardware standby mode. SYSCR is not initialized in software standby mode.

Bits 5 and 4—Interrupt Control Mode 1 and 0 (INTM1, INTM0): These bits select one of two interrupt control modes for the interrupt controller.

Bit 5	Bit 4	Interrupt		
INTM1	**INTM0**	**Control Mode**	**Description**	
0	0	0	Interrupts are controlled by I bit	(Initial value)
	1	—	Setting prohibited	
1	0	2	Interrupts are controlled by bits I2 to I0, and IPR	
	1	—	Setting prohibited	

Bit 3—NMI Edge Select (NMIEG): Selects the input edge for the NMI pin.

Bit 3		
NMIEG	**Description**	
0	Interrupt request generated at falling edge of NMI input	(Initial value)
1	Interrupt request generated at rising edge of NMI input	

5.2.2 Interrupt Priority Registers A to H, J to M (IPRA to IPRH, IPRJ to IPRM)

Bit	:	7	6	5	4	3	2	1	0
		—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0
Initial value	:	0	1	1	1	0	1	1	1
R/W	:	—	R/W	R/W	R/W	—	R/W	R/W	R/W

The IPR registers are twelve 8-bit readable/writable registers that set priorities (levels 7 to 0) for interrupts other than NMI.

The correspondence between IPR settings and interrupt sources is shown in table 5-3.

The IPR registers set a priority (level 7 to 0) for each interrupt source other than NMI.

The IPR registers are initialized to H'77 by a reset and in hardware standby mode.

Bits 7 and 3—Reserved: These bits are always read as 0 and cannot be modified.

Table 5-3 Correspondence between Interrupt Sources and IPR Settings

	Bits	
Register	**6 to 4**	**2 to 0**
IPRA	IRQ0	IRQ1
IPRB	IRQ2	IRQ4
	IRQ3	IRQ5
IPRC	—*1	DTC
IPRD	Watchdog timer 0	—*1
IPRE	PC break	A/D converter, watchdog timer 1
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRJ	—*1	SCI channel 0
IPRK	SCI channel 1	SCI channel 2
IPRL	—*1	IIC (Option)*2
IPRM	PWM channel 1, 2 HCAN channel 1	HCAN channel 0

Notes: *1 Reserved. These bits are always read as 1 and cannot be modified.

*2 I^2C bus interface is available as an option in the H8S/2638.
The IIC bit becomes reserved bit when this optional feature is not used.

As shown in table 5-3, multiple interrupts are assigned to one IPR. Setting a value in the range from H'0 to H'7 in the 3-bit groups of bits 6 to 4 and 2 to 0 sets the priority of the corresponding interrupt. The lowest priority level, level 0, is assigned by setting H'0, and the highest priority level, level 7, by setting H'7.

When interrupt requests are generated, the highest-priority interrupt according to the priority levels set in the IPR registers is selected. This interrupt level is then compared with the interrupt mask level set by the interrupt mask bits (I2 to I0) in the extend register (EXR) in the CPU, and if the priority level of the interrupt is higher than the set mask level, an interrupt request is issued to the CPU.

5.2.3 IRQ Enable Register (IER)

Bit	:	7	6	5	4	3	2	1	0
		—	—	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IER is an 8-bit readable/writable register that controls enabling and disabling of interrupt requests IRQ5 to IRQ0.

IER is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 and 6—Reserved: These bits are always read as 0, and should only be written with 0.

Bits 5 to 0—IRQ5 to IRQ0 Enable (IRQ5E to IRQ0E): These bits select whether IRQ5 to IRQ0 are enabled or disabled.

Bit n IRQnE	Description	
0	IRQn interrupts disabled	(Initial value)
1	IRQn interrupts enabled	

(n = 5 to 0)

5.2.4 IRQ Sense Control Registers H and L (ISCRH, ISCRL)

ISCRH

Bit	:	15	14	13	12	11	10	9	8
		—	—	—	—	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ISCRL

Bit	:	7	6	5	4	3	2	1	0
		IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The ISCR registers are 16-bit readable/writable registers that select rising edge, falling edge, or both edge detection, or level sensing, for the input at pins $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$.

The ISCR registers are initialized to H'0000 by a reset and in hardware standby mode.

Bits 15 to 12—Reserved: These bits are always read as 0, and should only be written with 0.

Bits 11 to 0: IRQ5 Sense Control A and B (IRQ5SCA, IRQ5SCB) to IRQ0 Sense Control A and B (IRQ0SCA, IRQ0SCB)

Bits 11 to 0

IRQ5SCB to IRQ0SCB	IRQ5SCA to IRQ0SCA	Description
0	0	Interrupt request generated at $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ input low level (initial value)
	1	Interrupt request generated at falling edge of $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ input
1	0	Interrupt request generated at rising edge of $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ input
	1	Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ input

5.2.5 IRQ Status Register (ISR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only 0 can be written, to clear the flag.

ISR is an 8-bit readable/writable register that indicates the status of IRQ5 to IRQ0 interrupt requests.

ISR is initialized to H'00 by a reset and in hardware standby mode.

They are not initialized in software standby mode.

Bits 7 and 6—Reserved: These bits are always read as 0.

Bits 5 to 0—IRQ5 to IRQ0 flags (IRQ5F to IRQ0F): These bits indicate the status of IRQ5 to IRQ0 interrupt requests.

Bit n IRQnF	Description
0	[Clearing conditions] (Initial value) • Cleared by reading IRQnF flag when IRQnF = 1, then writing 0 to IRQnF flag • When interrupt exception handling is executed when low-level detection is set (IRQnSCB = IRQnSCA = 0) and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed when falling, rising, or both-edge detection is set (IRQnSCB = 1 or IRQnSCA = 1) • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0
1	[Setting conditions] • When $\overline{\text{IRQn}}$ input goes low when low-level detection is set (IRQnSCB = IRQnSCA = 0) • When a falling edge occurs in $\overline{\text{IRQn}}$ input when falling edge detection is set (IRQnSCB = 0, IRQnSCA = 1) • When a rising edge occurs in $\overline{\text{IRQn}}$ input when rising edge detection is set (IRQnSCB = 1, IRQnSCA = 0) • When a falling or rising edge occurs in $\overline{\text{IRQn}}$ input when both-edge detection is set (IRQnSCB = IRQnSCA = 1)

(n = 5 to 0)

5.3 Interrupt Sources

Interrupt sources comprise external interrupts (NMI and IRQ5 to IRQ0) and internal interrupts (49 sources).

5.3.1 External Interrupts

There are seven external interrupts: NMI and IRQ5 to IRQ0. Of these, NMI and IRQ5 to IRQ0 can be used to restore the chip from software standby mode.

NMI Interrupt: NMI is the highest-priority interrupt, and is always accepted by the CPU regardless of the interrupt control mode or the status of the CPU interrupt mask bits. The NMIEG bit in SYSCR can be used to select whether an interrupt is requested at a rising edge or a falling edge on the NMI pin.

The vector number for NMI interrupt exception handling is 7.

IRQ5 to IRQ0 Interrupts: Interrupts IRQ5 to IRQ0 are requested by an input signal at pins $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$. Interrupts IRQ5 to IRQ0 have the following features:

- Using ISCR, it is possible to select whether an interrupt is generated by a low level, falling edge, rising edge, or both edges, at pins $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$.
- Enabling or disabling of interrupt requests IRQ5 to IRQ0 can be selected with IER.
- The interrupt priority level can be set with IPR.
- The status of interrupt requests IRQ5 to IRQ0 is indicated in ISR. ISR flags can be cleared to 0 by software.

A block diagram of interrupts IRQ5 to IRQ0 is shown in figure 5-2.

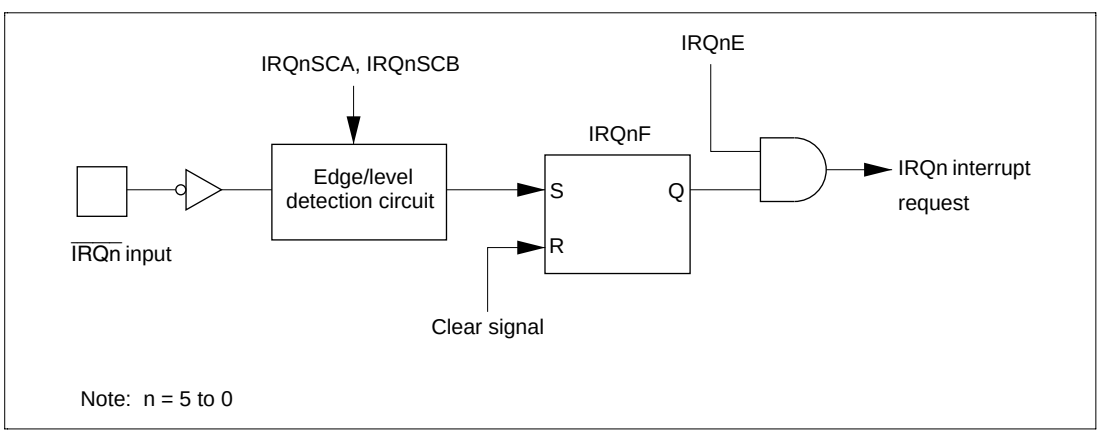


Figure 5-2 Block Diagram of Interrupts IRQ5 to IRQ0

Figure 5-3 shows the timing of setting IRQnF.

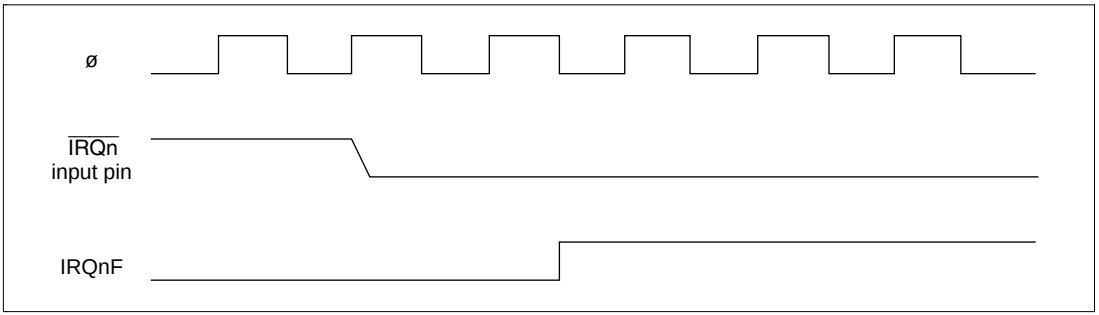


Figure 5-3 Timing of Setting IRQnF

The vector numbers for IRQ5 to IRQ0 interrupt exception handling are 21 to 16.

Detection of IRQ5 to IRQ0 interrupts does not depend on whether the relevant pin has been set for input or output. However, when a pin is used as an external interrupt input pin, do not clear the corresponding DDR to 0 and use the pin as an I/O pin for another function.

5.3.2 Internal Interrupts

There are 49 sources for internal interrupts from on-chip supporting modules.

- For each on-chip supporting module there are flags that indicate the interrupt request status, and enable bits that select enabling or disabling of these interrupts. If both of these are set to 1 for a particular interrupt source, an interrupt request is issued to the interrupt controller.
- The interrupt priority level can be set by means of IPR.
- The DTC can be activated by a TPU, SCI, or other interrupt request. When the DTC is activated by an interrupt, the interrupt control mode and interrupt mask bits are not affected.

5.3.3 Interrupt Exception Handling Vector Table

Table 5-4 shows interrupt exception handling sources, vector addresses, and interrupt priorities. For default priorities, the lower the vector number, the higher the priority.

Priorities among modules can be set by means of the IPR. The situation when two or more modules are set to the same priority, and priorities within a module, are fixed as shown in table 5-4.

Table 5-4 Interrupt Sources, Vector Addresses, and Interrupt Priorities

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*[1] Advanced Mode	IPR	Priority
NMI	External pin	7	H'001C		High
IRQ0		16	H'0040	IPRA6 to 4	
IRQ1		17	H'0044	IPRA2 to 0	
IRQ2 IRQ3		18 19	H'0048 H'004C	IPRB6 to 4	
IRQ4 IRQ5		20 21	H'0050 H'0054	IPRB2 to 0	
Reserved for system use	—	22 23	H'0058 H'005C	—	
SWDTEND (software activation interrupt end)	DTC	24	H'0060	IPRC2 to 0	
WOVI0 (interval timer)	Watchdog timer 0	25	H'0064	IPRD6 to 4	
Reserved for system use	—	26	H'0068	—	
PC break	PC break	27	H'006C	IPRE6 to 4	
ADI (A/D conversion end)	A/D	28	H'0070	IPRE2 to 0	
WOVI1 (interval timer)	Watchdog timer 1	29	H'0074		
Reserved for system use	—	30 31	H'0078 H'007C	—	
TGI0A (TGR0A input capture/compare match)	TPU channel 0	32	H'0080	IPRF6 to 4	
TGI0B (TGR0B input capture/compare match)		33	H'0084		
TGI0C (TGR0C input capture/compare match)		34	H'0088		
TGI0D (TGR0D input capture/compare match)		35	H'008C		
TCI0V (overflow 0)		36	H'0090		
Reserved for system use	—	37 to 39	H'0094 to H'009C	—	Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*[1] Advanced Mode	IPR	Priority
TGI1A (TGR1A input capture/compare match)	TPU channel 1	40	H'00A0	IPRF2 to 0	High
TGI1B (TGR1B input capture/compare match)		41	H'00A4		
TCI1V (overflow 1)		42	H'00A8		
TCI1U (underflow 1)		43	H'00AC		
TGI2A (TGR2A input capture/compare match)	TPU channel 2	44	H'00B0	IPRG6 to 4	
TGI2B (TGR2B input capture/compare match)		45	H'00B4		
TCI2V (overflow 2)		46	H'00B8		
TCI2U (underflow 2)		47	H'00BC		
TGI3A (TGR3A input capture/compare match)	TPU channel 3	48	H'00C0	IPRG2 to 0	
TGI3B (TGR3B input capture/compare match)		49	H'00C4		
TGI3C (TGR3C input capture/compare match)		50	H'00C8		
TGI3D (TGR3D input capture/compare match)		51	H'00CC		
TCI3V (overflow 3)		52	H'00D0		
Reserved for system use	—	53 to 55	H'00D4 to H'00DC	—	
TGI4A (TGR4A input capture/compare match)	TPU channel 4	56	H'00E0	IPRH6 to 4	
TGI4B (TGR4B input capture/compare match)		57	H'00E4		
TCI4V (overflow 4)		58	H'00E8		
TCI4U (underflow 4)		59	H'00EC		
TGI5A (TGR5A input capture/compare match)	TPU channel 5	60	H'00F0	IPRH2 to 0	
TGI5B (TGR5B input capture/compare match)		61	H'00F4		
TCI5V (overflow 5)		62	H'00F8		
TCI5U (underflow 5)		63	H'00FC		Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address*[1] Advanced Mode	IPR	Priority
Reserved for system use	—	64 to 79	H'0100 to H'013C	—	High
ERI0 (receive error 0) RXI0 (reception completed 0) TXI0 (transmit data empty 0) TEI0 (transmission end 0)	SCI channel 0	80 81 82 83	H'0140 H'0144 H'0148 H'014C	IPRJ2 to 0	
ERI1 (receive error 1) RXI1 (reception completed 1) TXI1 (transmit data empty 1) TEI1 (transmission end 1)	SCI channel 1	84 85 86 87	H'0150 H'0154 H'0158 H'015C	IPRK6 to 4	
ERI2 (receive error 2) RXI2 (reception completed 2) TXI2 (transmit data empty 2) TEI2 (transmission end 2)	SCI channel 2	88 89 90 91	H'0160 H'0164 H'0168 H'016C	IPRK2 to 0	
Reserved for system use	—	92 to 99	H'0170 to H'018C	—	
IICI0 (1-byte transmission/ reception completed) DDCSW1 (format switch)	IIC channel 0 (option)*[2]	100 101	H'0190 H'0194	IPRL2 to 0	
IICI1 Reserved for system use	IIC channel 1 (option)*[2]	102 103	H'0198 H'019C		
PWM1	PWM channel 1	104	H'01A0	IPRM6 to 4	
PWM2	PWM channel 2	105	H'01A4		
ERS0, OVR0, RM1, SLE0, RM0	HCAN1	106 107	H'01A8 H'01AC		
ERS0, OVR0, RM1, SLE0, RM0	HCAN0	108 109	H'01B0 H'01B4	IPRM2 to 0	
Reserved for system use	—	110	H'01B8		
		111	H'01BC		Low

Notes: *1 Lower 16 bits of the start address.

*2 IIC is available as an option in the H8S/2638 and H8S/2639 only.

5.4 Interrupt Operation

5.4.1 Interrupt Control Modes and Interrupt Operation

Interrupt operations in the chip differ depending on the interrupt control mode.

NMI interrupts are accepted at all times except in the reset state and the hardware standby state. In the case of IRQ interrupts and on-chip supporting module interrupts, an enable bit is provided for each interrupt. Clearing an enable bit to 0 disables the corresponding interrupt request. Interrupt sources for which the enable bits are set to 1 are controlled by the interrupt controller.

Table 5-5 shows the interrupt control modes.

The interrupt controller performs interrupt control according to the interrupt control mode set by the INTM1 and INTM0 bits in SYSCR, the priorities set in IPR, and the masking state indicated by the I bit in the CPU's CCR, and bits I2 to I0 in EXR.

Table 5-5 Interrupt Control Modes

Interrupt Control Mode	SYSCR		Priority Setting Registers	Interrupt Mask Bits	Description
	INTM1	INTM0			
0	0	0	—	I	Interrupt mask control is performed by the I bit.
—		1	—	—	Setting prohibited
2	1	0	IPR	I2 to I0	8-level interrupt mask control is performed by bits I2 to I0. 8 priority levels can be set with IPR.
—		1	—	—	Setting prohibited

Figure 5-4 shows a block diagram of the priority decision circuit.

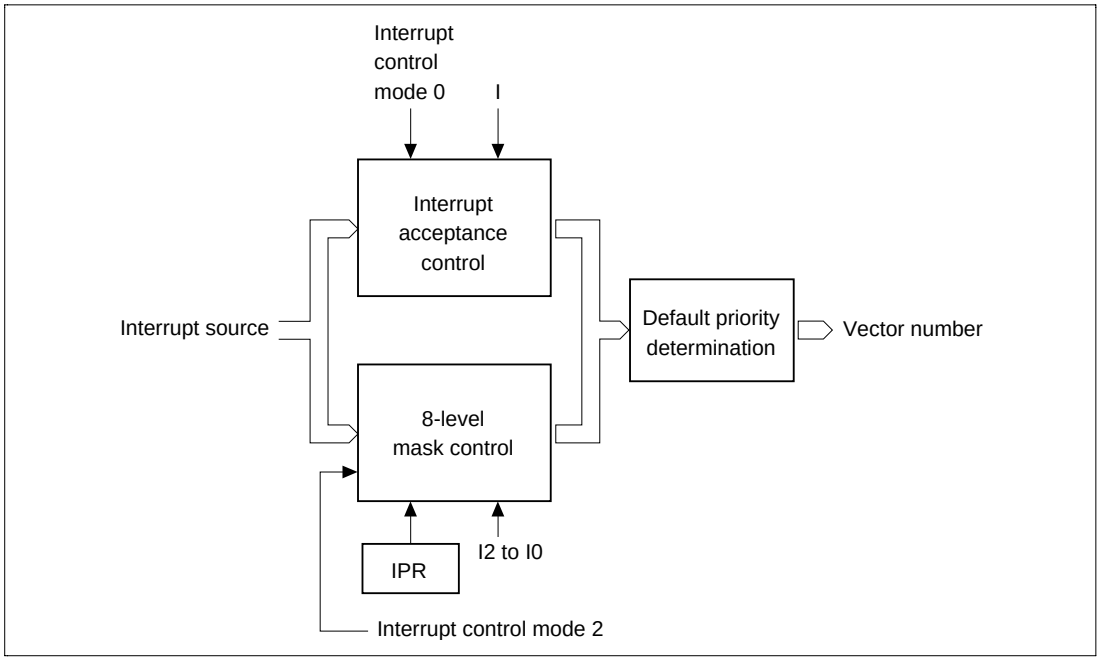


Figure 5-4 Block Diagram of Interrupt Control Operation

(1) Interrupt Acceptance Control

In interrupt control mode 0, interrupt acceptance is controlled by the I bit in CCR.

Table 5-6 shows the interrupts selected in each interrupt control mode.

Table 5-6 Interrupts Selected in Each Interrupt Control Mode (1)

	Interrupt Mask Bits	
Interrupt Control Mode	**I**	**Selected Interrupts**
0	0	All interrupts
	1	NMI interrupts
2	*	All interrupts

Legend
* : Don't care

(2) 8-Level Control

In interrupt control mode 2, 8-level mask level determination is performed for the selected interrupts in interrupt acceptance control according to the interrupt priority level (IPR).

The interrupt source selected is the interrupt with the highest priority level, and whose priority level set in IPR is higher than the mask level.

Table 5-7 Interrupts Selected in Each Interrupt Control Mode (2)

Interrupt Control Mode	Selected Interrupts
0	All interrupts
2	Highest-priority-level (IPR) interrupt whose priority level is greater than the mask level (IPR > I2 to I0).

(3) Default Priority Determination

When an interrupt is selected by 8-level control, its priority is determined and a vector number is generated.

If the same value is set for IPR, acceptance of multiple interrupts is enabled, and so only the interrupt source with the highest priority according to the preset default priorities is selected and has a vector number generated.

Interrupt sources with a lower priority than the accepted interrupt source are held pending.

Table 5-8 shows operations and control signal functions in each interrupt control mode.

Table 5-8 Operations and Control Signal Functions in Each Interrupt Control Mode

Interrupt Control Mode	Setting		Interrupt Acceptance Control		8-Level Control			Default Priority Determination	T (Trace)
	INTM1	INTM0		I		I2 to I0	IPR		
0	0	0	○	IM	X	—	—*2	○	—
2	1	0	X	—*1	○	IM	PR	○	T

Legend
- ○ : Interrupt operation control performed
- X : No operation. (All interrupts enabled)
- IM : Used as interrupt mask bit
- PR : Sets priority.
- — : Not used.

Notes: *1 Set to 1 when interrupt is accepted.
*2 Keep the initial setting.

5.4.2 Interrupt Control Mode 0

Enabling and disabling of IRQ interrupts and on-chip supporting module interrupts can be set by means of the I bit in the CPU's CCR. Interrupts are enabled when the I bit is cleared to 0, and disabled when set to 1.

Figure 5-5 shows a flowchart of the interrupt acceptance operation in this case.

[1] If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.

[2] The I bit is then referenced. If the I bit is cleared to 0, the interrupt request is accepted. If the I bit is set to 1, only an NMI interrupt is accepted, and other interrupt requests are held pending.

[3] Interrupt requests are sent to the interrupt controller, the highest-ranked interrupt according to the priority system is accepted, and other interrupt requests are held pending.

[4] When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.

[5] The PC and CCR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.

[6] Next, the I bit in CCR is set to 1. This masks all interrupts except NMI.

[7] A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.

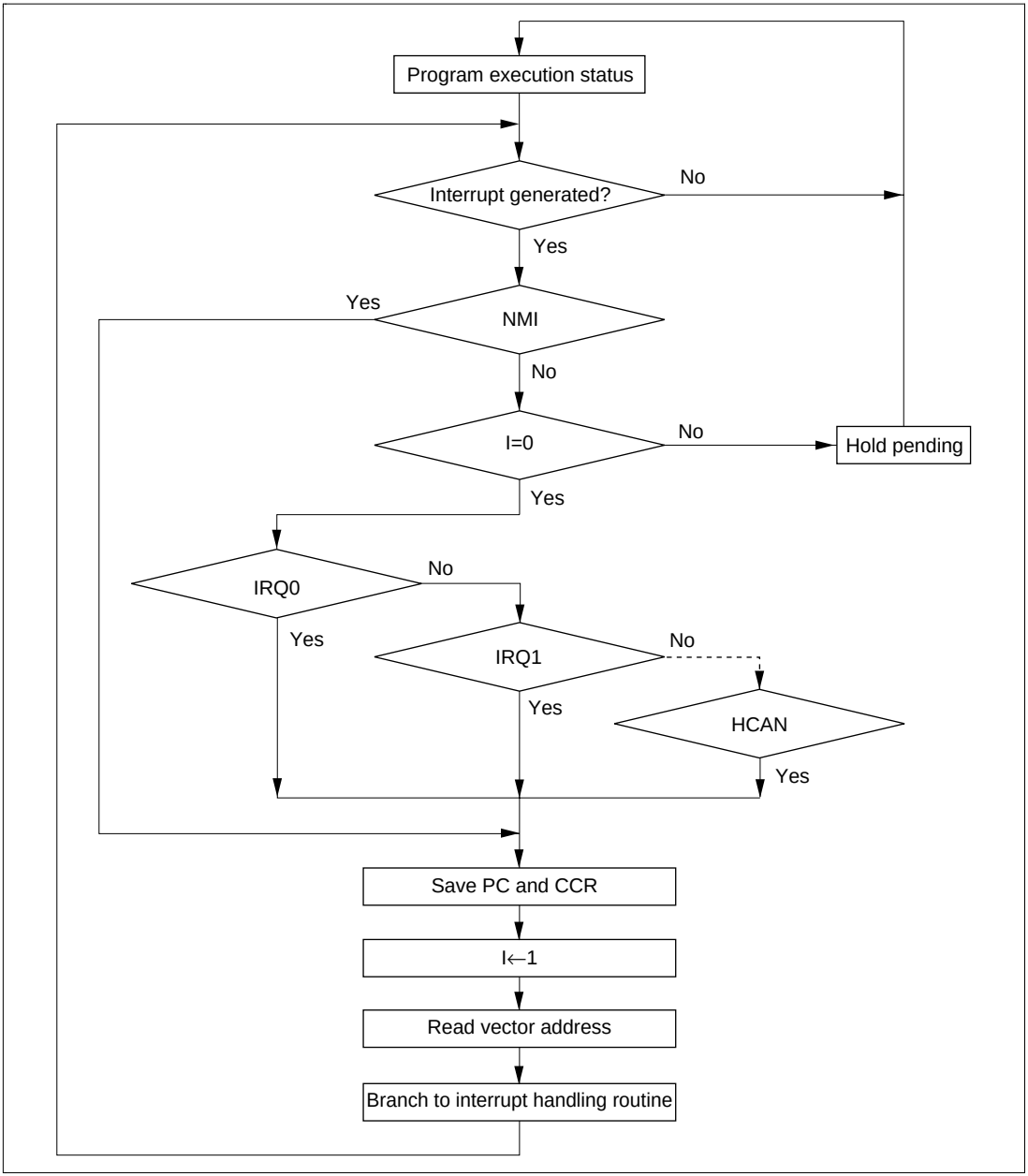


Figure 5-5 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 0

5.4.3 Interrupt Control Mode 2

Eight-level masking is implemented for IRQ interrupts and on-chip supporting module interrupts by comparing the interrupt mask level set by bits I2 to I0 of EXR in the CPU with IPR.

Figure 5-6 shows a flowchart of the interrupt acceptance operation in this case.

[1] If an interrupt source occurs when the corresponding interrupt enable bit is set to 1, an interrupt request is sent to the interrupt controller.

[2] When interrupt requests are sent to the interrupt controller, the interrupt with the highest priority according to the interrupt priority levels set in IPR is selected, and lower-priority interrupt requests are held pending. If a number of interrupt requests with the same priority are generated at the same time, the interrupt request with the highest priority according to the priority system shown in table 5-4 is selected.

[3] Next, the priority of the selected interrupt request is compared with the interrupt mask level set in EXR. An interrupt request with a priority no higher than the mask level set at that time is held pending, and only an interrupt request with a priority higher than the interrupt mask level is accepted.

[4] When an interrupt request is accepted, interrupt exception handling starts after execution of the current instruction has been completed.

[5] The PC, CCR, and EXR are saved to the stack area by interrupt exception handling. The PC saved on the stack shows the address of the first instruction to be executed after returning from the interrupt handling routine.

[6] The T bit in EXR is cleared to 0. The interrupt mask level is rewritten with the priority level of the accepted interrupt.

If the accepted interrupt is NMI, the interrupt mask level is set to H'7.

[7] A vector address is generated for the accepted interrupt, and execution of the interrupt handling routine starts at the address indicated by the contents of that vector address.

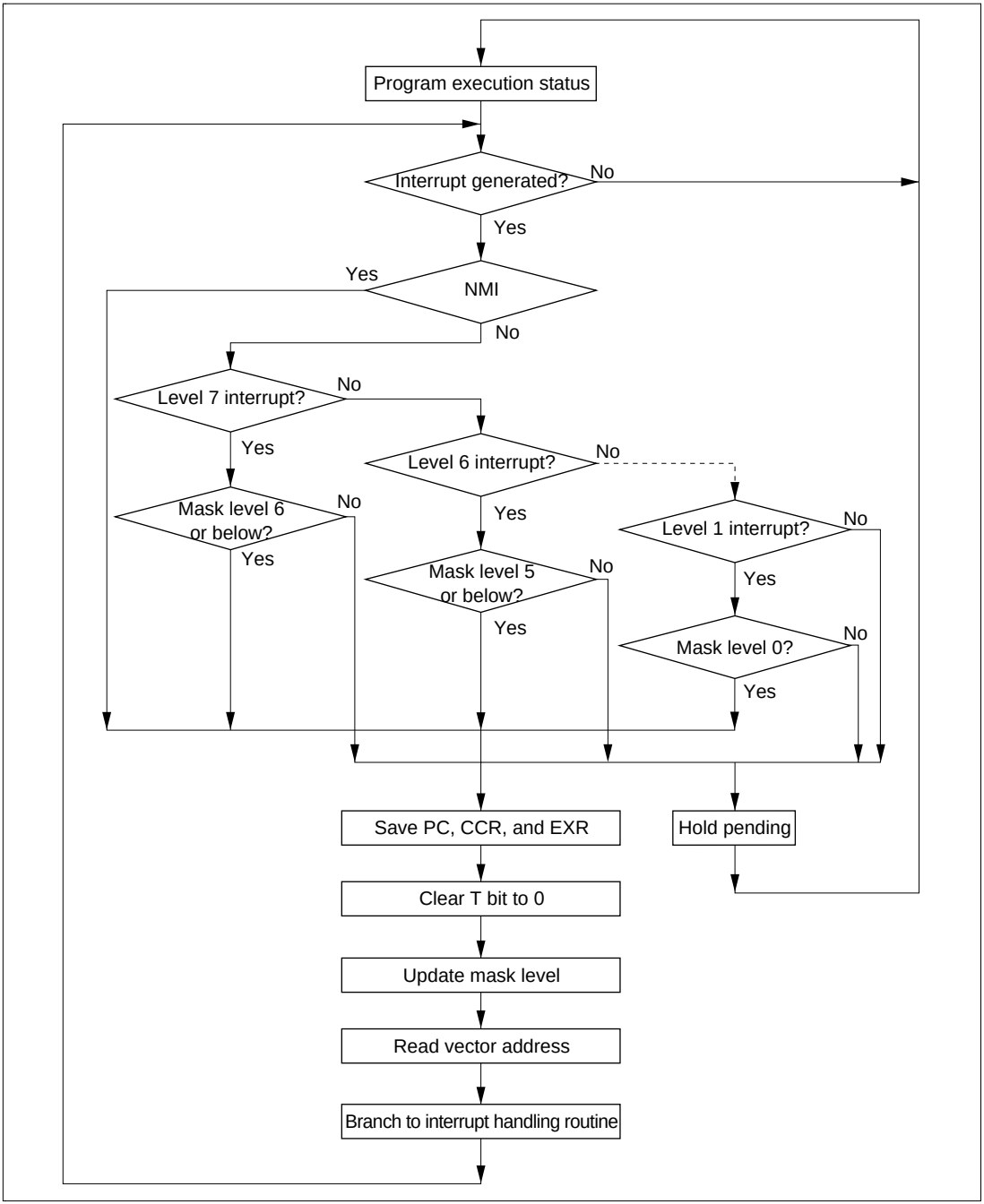


Figure 5-6 Flowchart of Procedure Up to Interrupt Acceptance in Interrupt Control Mode 2

5.4.4 Interrupt Exception Handling Sequence

Figure 5-7 shows the interrupt exception handling sequence. The example shown is for the case where interrupt control mode 0 is set in advanced mode, and the program area and stack area are in on-chip memory.

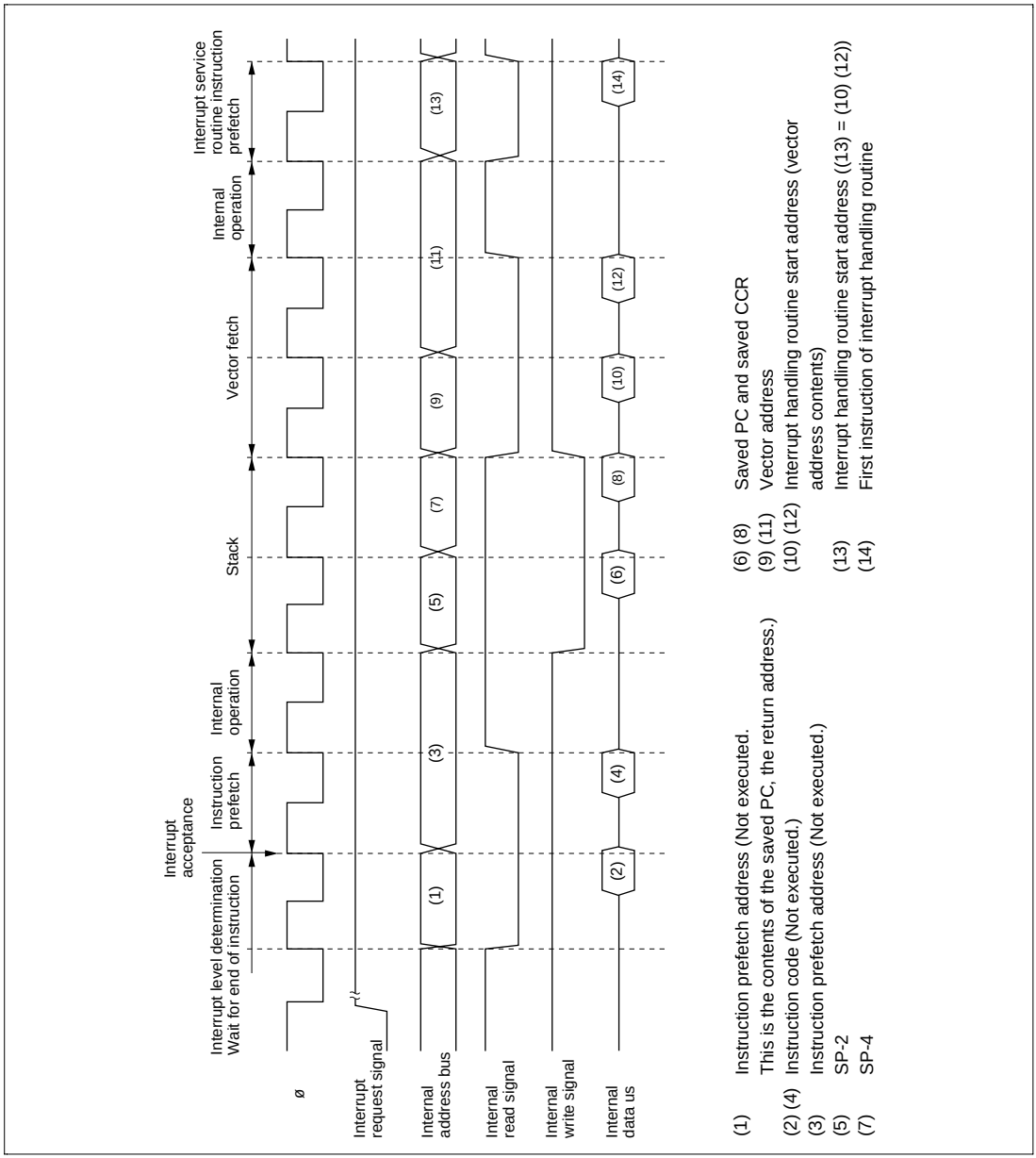


Figure 5-7 Interrupt Exception Handling

5.4.5 Interrupt Response Times

The chip is capable of fast word transfer instruction to on-chip memory, and the program area is provided in on-chip ROM and the stack area in on-chip RAM, enabling high-speed processing.

Table 5-9 shows interrupt response times - the interval between generation of an interrupt request and execution of the first instruction in the interrupt handling routine. The execution status symbols used in table 5-9 are explained in table 5-10.

Table 5-9 Interrupt Response Times

		Normal Mode*5		Advanced Mode	
No.	**Execution Status**	**INTM1 = 0**	**INTM1 = 1**	**INTM1 = 0**	**INTM1 = 1**
1	Interrupt priority determination*1	3	3	3	3
2	Number of wait states until executing instruction ends*2	1 to $(19+2\cdot S_I)$	1 to $(19+2\cdot S_I)$	1 to $(19+2\cdot S_I)$	1 to $(19+2\cdot S_I)$
3	PC, CCR, EXR stack save	$2\cdot S_K$	$3\cdot S_K$	$2\cdot S_K$	$3\cdot S_K$
4	Vector fetch	S_I	S_I	$2\cdot S_I$	$2\cdot S_I$
5	Instruction fetch*3	$2\cdot S_I$	$2\cdot S_I$	$2\cdot S_I$	$2\cdot S_I$
6	Internal processing*4	2	2	2	2
Total (using on-chip memory)		11 to 31	12 to 32	12 to 32	13 to 33

Notes: *1 Two states in case of internal interrupt.
*2 Refers to MULXS and DIVXS instructions.
*3 Prefetch after interrupt acceptance and interrupt handling routine prefetch.
*4 Internal processing after interrupt acceptance and internal processing after vector fetch.
*5 Not available in the H8S/2636 and H8S/2638.

Table 5-10 Number of States in Interrupt Handling Routine Execution Statuses

		Object of Access				
			External Device			
			8 Bit Bus		16 Bit Bus	
Symbol		**Internal Memory**	**2-State Access**	**3-State Access**	**2-State Access**	**3-State Access**
Instruction fetch	S_I	1	4	6+2m	2	3+m
Branch address read	S_J					
Stack manipulation	S_K					

Legend
m: Number of wait states in an external device access.

5.5 Usage Notes

5.5.1 Contention between Interrupt Generation and Disabling

When an interrupt enable bit is cleared to 0 to disable interrupts, the disabling becomes effective after execution of the instruction.

In other words, when an interrupt enable bit is cleared to 0 by an instruction such as BCLR or MOV, if an interrupt is generated during execution of the instruction, the interrupt concerned will still be enabled on completion of the instruction, and so interrupt exception handling for that interrupt will be executed on completion of the instruction. However, if there is an interrupt request of higher priority than that interrupt, interrupt exception handling will be executed for the higher-priority interrupt, and the lower-priority interrupt will be ignored.

The same also applies when an interrupt source flag is cleared to 0.

Figure 5-8 shows an example in which the TCIEV bit in the TPU's TIER register is cleared to 0.

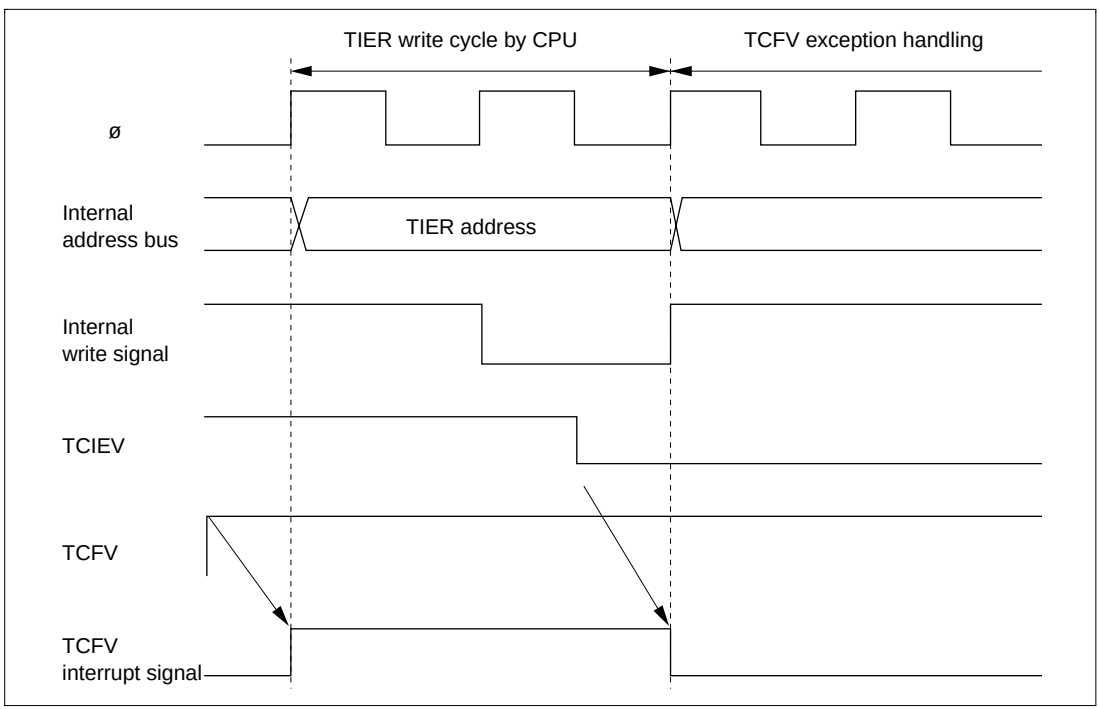


Figure 5-8 Contention between Interrupt Generation and Disabling

The above contention will not occur if an enable bit or interrupt source flag is cleared to 0 while the interrupt is masked.

5.5.2 Instructions that Disable Interrupts

Instructions that disable interrupts are LDC, ANDC, ORC, and XORC. After any of these instructions is executed, all interrupts including NMI are disabled and the next instruction is always executed. When the I bit is set by one of these instructions, the new value becomes valid two states after execution of the instruction ends.

5.5.3 Times when Interrupts are Disabled

There are times when interrupt acceptance is disabled by the interrupt controller.

The interrupt controller disables interrupt acceptance for a 3-state period after the CPU has updated the mask level with an LDC, ANDC, ORC, or XORC instruction.

5.5.4 Interrupts during Execution of EEPMOV Instruction

Interrupt operation differs between the EEPMOV.B instruction and the EEPMOV.W instruction.

With the EEPMOV.B instruction, an interrupt request (including NMI) issued during the transfer is not accepted until the move is completed.

With the EEPMOV.W instruction, if an interrupt request is issued during the transfer, interrupt exception handling starts at a break in the transfer cycle. The PC value saved on the stack in this case is the address of the next instruction.

Therefore, if an interrupt is generated during execution of an EEPMOV.W instruction, the following coding should be used.

```
L1:   EEPMOV.W
      MOV.W     R4,R4
      BNE       L1
```

5.5.5 IRQ Interrupts

When operating by clock input, acceptance of input to an $\overline{\text{IRQ}}$ pin is synchronized with the clock. In software standby mode, the input is accepted asynchronously. For details on the input conditions, see section 24.3.2, Control Signal Timing.

5.6 DTC Activation by Interrupt

5.6.1 Overview

The DTC can be activated by an interrupt. In this case, the following options are available:

- Interrupt request to CPU
- Activation request to DTC
- Selection of a number of the above

For details of interrupt requests that can be used with to activate the DTC, see section 8, Data Transfer Controller.

5.6.2 Block Diagram

Figure 5-9 shows a block diagram of the DTC interrupt controller.

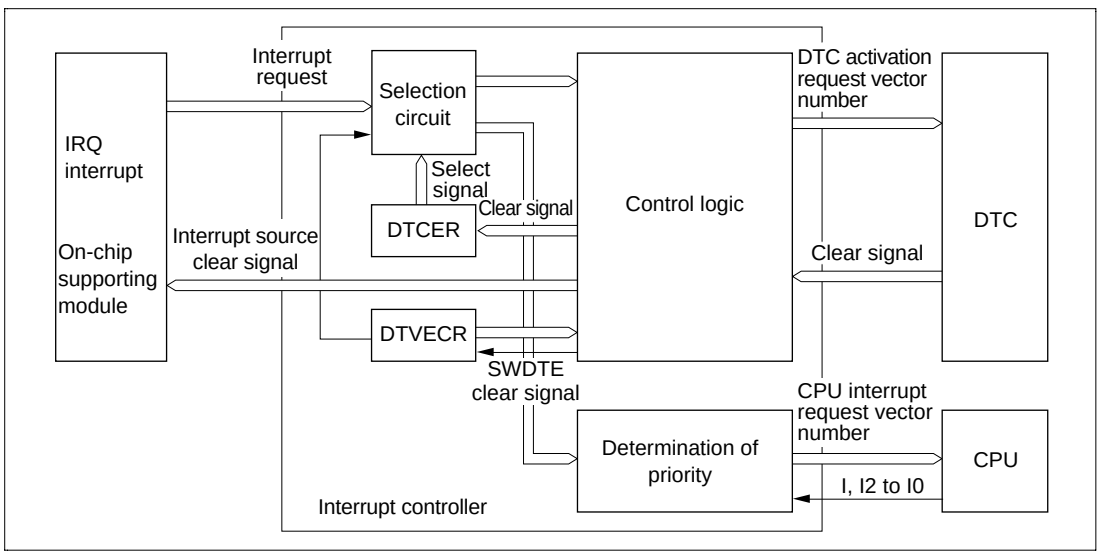


Figure 5-9 Interrupt Control for DTC

5.6.3 Operation

The interrupt controller has three main functions in DTC control.

(1) Selection of Interrupt Source: Interrupt factors are selected as DTC activation request or CPU interrupt request by the DTCE bit of DTCERA to DTCERG of DTC.

By specifying the DISEL bit of the DTC's MRB, it is possible to clear the DTCE bit to 0 after DTC data transfer, and request a CPU interrupt.

If DTC carries out the designate number of data transfers and the transfer counter reads 0, after DTC data transfer, the DTCE bit is also cleared to 0, and a CPU interrupt requested.

(2) Determination of Priority: The DTC activation source is selected in accordance with the default priority order, and is not affected by mask or priority levels. See 8.3.3 DTC Vector Table for the respective priority.

(3) Operation Order: If the same interrupt is selected as a DTC activation source and a CPU interrupt source, the DTC data transfer is performed first, followed by CPU interrupt exception handling.

Table 5-11 shows the interrupt factor clear control and selection of interrupt factors by specification of the DTCE bit of DTCERA to DTCERG of DTC, and the DISEL bit of DTC's MRB.

Table 5-11 Interrupt Source Selection and Clearing Control

Settings			
DTC		**Interrupt Source Selection/Clearing Control**	
DTCE	**DISEL**	**DTC**	**CPU**
0	*	X	Δ
1	0	Δ	X
	1	○	Δ

Legend

Δ : The relevant interrupt is used. Interrupt source clearing is performed.
(The CPU should clear the source flag in the interrupt handling routine.)

○ : The relevant interrupt is used. The interrupt source is not cleared.

X : The relevant bit cannot be used.

* : Don't care

(4) Notes on Use: SCI and A/D converter interrupt sources are cleared when the DTC reads or writes to the prescribed register.

Section 6 PC Break Controller (PBC)

6.1 Overview

The PC break controller (PBC) provides functions that simplify program debugging. Using these functions, it is easy to create a self-monitoring debugger, enabling programs to be debugged with the chip alone, without using an in-circuit emulator. Four break conditions can be set in the PBC: instruction fetch, data read, data write, and data read/write.

6.1.1 Features

The PC break controller has the following features:

- Two break channels (A and B)
- The following can be set as break compare conditions:
  - — 24 address bits
    Bit masking possible
  - — Bus cycle
    Instruction fetch
    Data access: data read, data write, data read/write
  - — Bus master
    Either CPU or CPU/DTC can be selected
- The timing of PC break exception handling after the occurrence of a break condition is as follows:
  - — Immediately before execution of the instruction fetched at the set address (instruction fetch)
  - — Immediately after execution of the instruction that accesses data at the set address (data access)
- Module stop mode can be set
  - — The initial setting is for PBC operation to be halted. Register access is enabled by clearing module stop mode.

6.1.2 Block Diagram

Figure 6-1 shows a block diagram of the PC break controller.

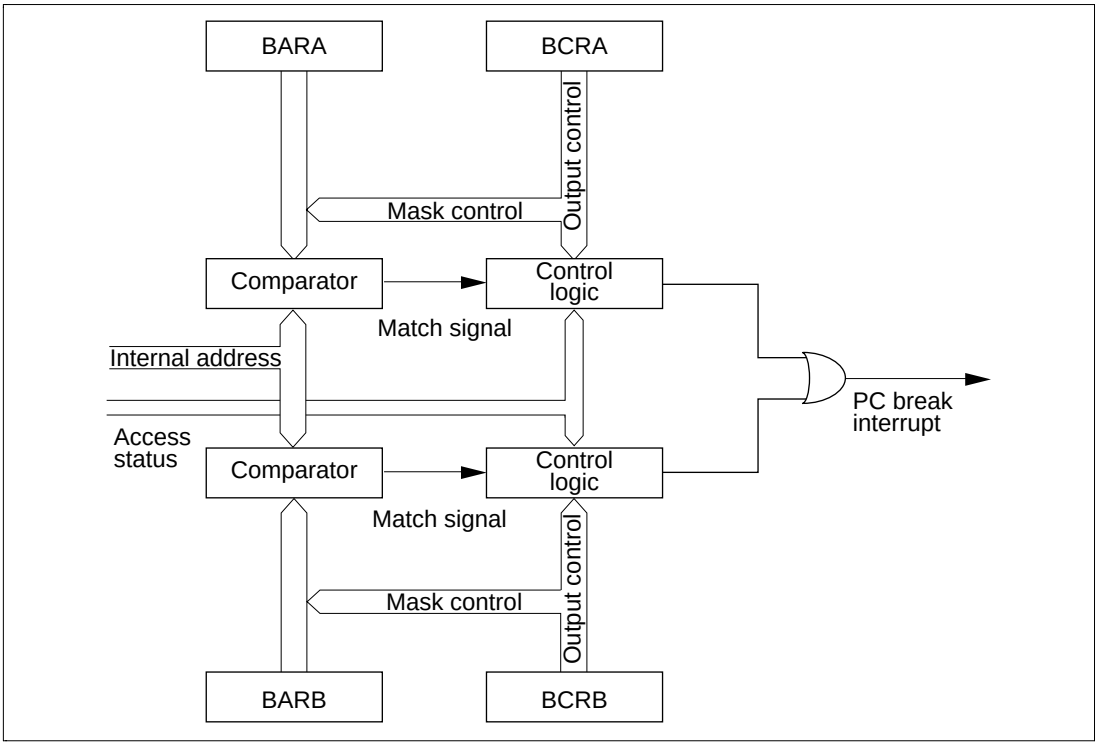


Figure 6-1 Block Diagram of PC Break Controller

6.1.3 Register Configuration

Table 6-1 shows the PC break controller registers.

Table 6-1 PC Break Controller Registers

Name	Abbreviation	R/W	Initial Value Reset	Address[*1]
Break address register A	BARA	R/W	H'XX000000	H'FE00
Break address register B	BARB	R/W	H'XX000000	H'FE04
Break control register A	BCRA	R/(W)[*2]	H'00	H'FE08
Break control register B	BCRB	R/(W)[*2]	H'00	H'FE09
Module stop control register C	MSTPCRC	R/W	H'FF	H'FDEA

Notes: *1 Lower 16 bits of the address.
*2 Only a 0 may be written to this bit to clear the flag.

6.2 Register Descriptions

6.2.1 Break Address Register A (BARA)

Bit	31	…	24	23	22	21	20	19	18	17	16	…	7	6	5	4	3	2	1	0
	—	…	—	BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16	…	BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0
Initial value	Undefined	…	Undefined	0	0	0	0	0	0	0	0	…	0	0	0	0	0	0	0	0
Read/Write	—	…	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	…	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BARA is a 32-bit readable/writable register that specifies the channel A break address.

BAA23 to BAA0 are initialized to H'000000 by a reset and in hardware standby mode.

Bits 31 to 24—Reserved: These bits return an undefined value if read, and cannot be modified.

Bits 23 to 0—Break Address A23 to A0 (BAA23–BAA0): These bits hold the channel A PC break address.

6.2.2 Break Address Register B (BARB)

BARB is the channel B break address register. The bit configuration is the same as for BARA.

6.2.3 Break Control Register A (BCRA)

Bit	7	6	5	4	3	2	1	0
	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only a 0 may be written to this bit to clear the flag.

BCRA is an 8-bit readable/writable register that controls channel A PC breaks. BCRA (1) selects the break condition bus master, (2) specifies bits subject to address comparison masking, and (3) specifies whether the break condition is applied to an instruction fetch or a data access. It also contains a condition match flag.

BCRA is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Condition Match Flag A (CMFA): Set to 1 when a break condition set for channel A is satisfied. This flag is not cleared to 0.

Bit 7

CMFA	Description	
0	[Clearing condition]	
	When 0 is written to CMFA after reading CMFA = 1	(Initial value)
1	[Setting condition]	
	When a condition set for channel A is satisfied	

Bit 6—CPU Cycle/DTC Cycle Select A (CDA): Selects the channel A break condition bus master.

Bit 6

CDA	Description	
0	PC break is performed when CPU is bus master	(Initial value)
1	PC break is performed when CPU or DTC is bus master	

Bits 5 to 3—Break Address Mask Register A2 to A0 (BAMRA2–BAMRA0): These bits specify which bits of the break address (BAA23–BAA0) set in BARA are to be masked.

Bit 5	Bit 4	Bit 3	
BAMRA2	**BAMRA1**	**BAMRA0**	**Description**
0	0	0	All BARA bits are unmasked and included in break conditions (Initial value)
		1	BAA0 (lowest bit) is masked, and not included in break conditions
	1	0	BAA1–0 (lower 2 bits) are masked, and not included in break conditions
		1	BAA2–0 (lower 3 bits) are masked, and not included in break conditions
1	0	0	BAA3–0 (lower 4 bits) are masked, and not included in break conditions
		1	BAA7–0 (lower 8 bits) are masked, and not included in break conditions
	1	0	BAA11–0 (lower 12 bits) are masked, and not included in break conditions
		1	BAA15–0 (lower 16 bits) are masked, and not included in break conditions

Bits 2 and 1—Break Condition Select A (CSELA1, CSELA0): These bits selection an instruction fetch, data read, data write, or data read/write cycle as the channel A break condition.

Bit 2	Bit 1		
CSELA1	**CSELA0**	**Description**	
0	0	Instruction fetch is used as break condition	(Initial value)
	1	Data read cycle is used as break condition	
1	0	Data write cycle is used as break condition	
	1	Data read/write cycle is used as break condition	

Bit 0—Break Interrupt Enable A (BIEA): Enables or disables channel A PC break interrupts.

Bit 0		
BIEA	**Description**	
0	PC break interrupts are disabled	(Initial value)
1	PC break interrupts are enabled	

6.2.4 Break Control Register B (BCRB)

BCRB is the channel B break control register. The bit configuration is the same as for BCRA.

6.2.5 Module Stop Control Register C (MSTPCRC)

Bit	7	6	5	4	3	2	1	0
	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPC4 bit is set to 1, PC break controller operation is stopped at the end of the bus cycle, and module stop mode is entered. Register read/write accesses are not possible in module stop mode. For details, see section 23A.5, 23B.5, Module Stop Mode.

MSTPCRC is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 4—Module Stop (MSTPC4): Specifies the PC break controller module stop mode.

Bit 4

MSTPC4	Description	
0	PC break controller module stop mode is cleared	
1	PC break controller module stop mode is set	(Initial value)

6.3 Operation

The operation flow from break condition setting to PC break interrupt exception handling is shown in sections 6.3.1 and 6.3.2, taking the example of channel A.

6.3.1 PC Break Interrupt Due to Instruction Fetch

(1) Initial settings

— Set the break address in BARA. For a PC break caused by an instruction fetch, set the address of the first instruction byte as the break address.

— Set the break conditions in BCRA.

BCRA bit 6 (CDA): With a PC break caused by an instruction fetch, the bus master must be the CPU. Set 0 to select the CPU.

BCRA bits 5–3 (BAMA2–0): Set the address bits to be masked.

BCRA bits 2–1 (CSELA1–0): Set 00 to specify an instruction fetch as the break condition.

BCRA bit 0 (BIEA): Set to 1 to enable break interrupts.

(2) Satisfaction of break condition

— When the instruction at the set address is fetched, a PC break request is generated immediately before execution of the fetched instruction, and the condition match flag (CMFA) is set.

(3) Interrupt handling

— After priority determination by the interrupt controller, PC break interrupt exception handling is started.

6.3.2 PC Break Interrupt Due to Data Access

(1) Initial settings

— Set the break address in BARA. For a PC break caused by a data access, set the target ROM, RAM, I/O, or external address space address as the break address. Stack operations and branch address reads are included in data accesses.

— Set the break conditions in BCRA.

BCRA bit 6 (CDA): Select the bus master.

BCRA bits 5–3 (BAMA2–0): Set the address bits to be masked.

BCRA bits 2–1 (CSELA1–0): Set 01, 10, or 11 to specify data access as the break condition.

BCRA bit 0 (BIEA): Set to 1 to enable break interrupts.

(2) Satisfaction of break condition

— After execution of the instruction that performs a data access on the set address, a PC break request is generated and the condition match flag (CMFA) is set.

(3) Interrupt handling

— After priority determination by the interrupt controller, PC break interrupt exception handling is started.

6.3.3 Notes on PC Break Interrupt Handling

(1) The PC break interrupt is shared by channels A and B. The channel from which the request was issued must be determined by the interrupt handler.

(2) The CMFA and CMFB flags are not cleared to 0, so 0 must be written to CMFA or CMFB after first reading the flag while it is set to 1. If the flag is left set to 1, another interrupt will be requested after interrupt handling ends.

(3) A PC break interrupt generated when the DTC is the bus master is accepted after the bus has been transferred to the CPU by the bus controller.

6.3.4 Operation in Transitions to Power-Down Modes

The operation when a PC break interrupt is set for an instruction fetch at the address after a SLEEP instruction is shown below.

(1) When the SLEEP instruction causes a transition from high-speed (medium-speed) mode to sleep mode, or from subactive mode* to subsleep mode*:

After execution of the SLEEP instruction, a transition is not made to sleep mode or subsleep mode*, and PC break interrupt handling is executed. After execution of PC break interrupt handling, the instruction at the address after the SLEEP instruction is executed (figure 6-2 (A)).

(2) When the SLEEP instruction causes a transition from high-speed (medium-speed) mode to subactive mode*:

After execution of the SLEEP instruction, a transition is made to subactive mode* via direct transition exception handling. After the transition, PC break interrupt handling is executed, then the instruction at the address after the SLEEP instruction is executed (figure 6-2 (B)).

(3) When the SLEEP instruction causes a transition from subactive mode* to high-speed (medium-speed) mode:

After execution of the SLEEP instruction, and following the clock oscillation settling time, a transition is made to high-speed (medium-speed) mode via direct transition exception handling. After the transition, PC break interrupt handling is executed, then the instruction at the address after the SLEEP instruction is executed (figure 6-2 (C)).

(4) When the SLEEP instruction causes a transition to software standby mode or watch mode*:

After execution of the SLEEP instruction, a transition is made to the respective mode, and PC break interrupt handling is not executed. However, the CMFA or CMFB flag is set (figure 6-2 (D)).

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only of the H8S/2636, H8S/2638, and H8S/2639 only.

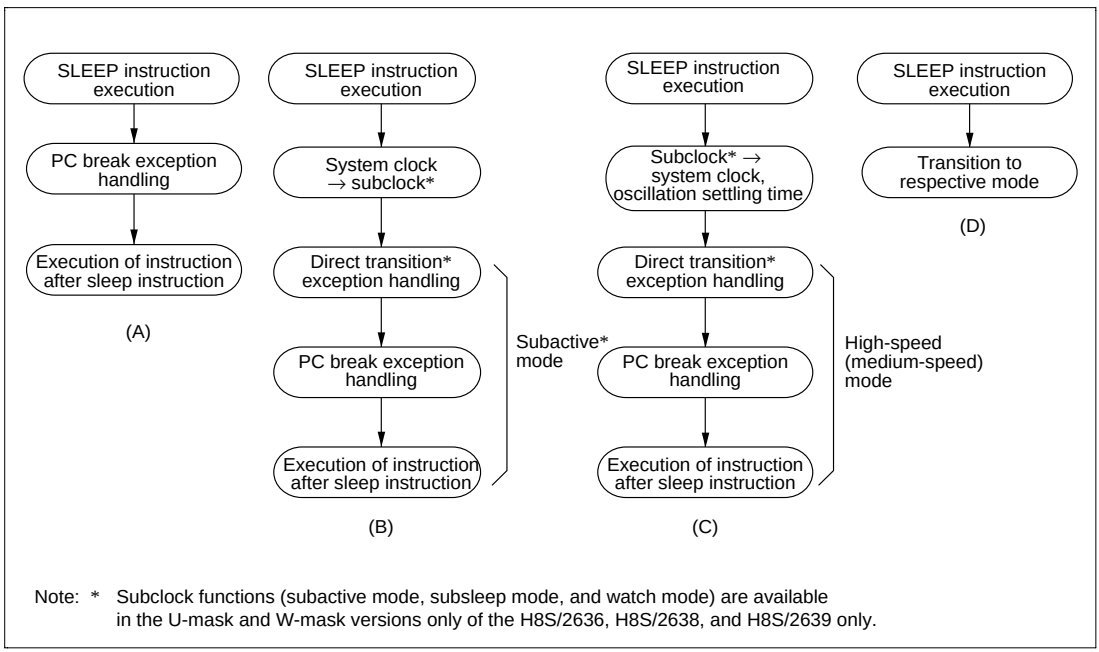


Figure 6-2 Operation in Power-Down Mode Transitions

6.3.5 PC Break Operation in Continuous Data Transfer

If a PC break interrupt is generated when the following operations are being performed, exception handling is executed on completion of the specified transfer.

(1) When a PC break interrupt is generated at the transfer address of an EEPMOV.B instruction:

PC break exception handling is executed after all data transfers have been completed and the EEPMOV.B instruction has ended.

(2) When a PC break interrupt is generated at a DTC transfer address:31

PC break exception handling is executed after the DTC has completed the specified number of data transfers, or after data for which the DISEL bit is set to 1 has been transferred.

6.3.6 When Instruction Execution is Delayed by One State

Caution is required in the following cases, as instruction execution is one state later than usual.

(1) When the PBC is enabled (i.e. when the break interrupt enable bit is set to 1), execution of a one-word branch instruction (Bcc d:8, BSR, JSR, JMP, TRAPA, RTE, or RTS) located in on-chip ROM or RAM is always delayed by one state.

(2) When break interruption by instruction fetch is set, the set address indicates on-chip ROM or RAM space, and that address is used for data access, the instruction that executes the data access is one state later than in normal operation.

(3) When break interruption by instruction fetch is set and a break interrupt is generated, if the executing instruction immediately preceding the set instruction has one of the addressing modes shown below, and that address indicates on-chip ROM or RAM, and that address is used for data access, the instruction will be one state later than in normal operation.

@ERn, @(d:16,ERn), @(d:32,ERn), @-ERn/ERn+, @aa:8, @aa:24, @aa:32, @(d:8,PC), @(d:16,PC), @@aa:8

(4) When break interruption by instruction fetch is set and a break interrupt is generated, if the executing instruction immediately preceding the set instruction is NOP or SLEEP, or has #xx,Rn as its addressing mode, and that instruction is located in on-chip ROM or RAM, the instruction will be one state later than in normal operation.

6.3.7 Additional Notes

(1) When a PC break is set for an instruction fetch at the address following a BSR, JSR, JMP, TRAPA, RTE, or RTS instruction:

Even if the instruction at the address following a BSR, JSR, JMP, TRAPA, RTE, or RTS instruction is fetched, it is not executed, and so a PC break interrupt is not generated by the instruction fetch at the next address.

(2) When the I bit is set by an LDC, ANDC, ORC, or XORC instruction, a PC break interrupt becomes valid two states after the end of the executing instruction. If a PC break interrupt is set for the instruction following one of these instructions, since interrupts, including NMI, are disabled for a 3-state period in the case of LDC, ANDC, ORC, and XORC, the next instruction is always executed. For details, see section 5, Interrupt Controller.

(3) When a PC break is set for an instruction fetch at the address following a Bcc instruction:

A PC break interrupt is generated if the instruction at the next address is executed in accordance with the branch condition, but is not generated if the instruction at the next address is not executed.

(4) When a PC break is set for an instruction fetch at the branch destination address of a Bcc instruction:

A PC break interrupt is generated if the instruction at the branch destination is executed in accordance with the branch condition, but is not generated if the instruction at the branch destination is not executed.

Section 7 Bus Controller

7.1 Overview

The chip has a built-in bus controller (BSC) that manages the external address space divided into eight areas. The bus specifications, such as bus width and number of access states, can be set independently for each area, enabling multiple memories to be connected easily.

The bus controller also has a bus arbitration function, and controls the operation of the internal bus masters: the CPU, and data transfer controller (DTC).

7.1.1 Features

The features of the bus controller are listed below.

- Manages external address space in area units
  - — Manages the external space as 8 areas of 2-Mbytes
  - — Bus specifications can be set independently for each area
  - — Burst ROM interface can be set
- Basic bus interface
  - — 8-bit access or 16-bit access can be selected for each area
  - — 2-state access or 3-state access can be selected for each area
  - — Program wait states can be inserted for each area
- Burst ROM interface
  - — Burst ROM interface can be set for area 0
  - — Choice of 1- or 2-state burst access
- Idle cycle insertion
  - — An idle cycle can be inserted in case of an external read cycle between different areas
  - — An idle cycle can be inserted in case of an external write cycle immediately after an external read cycle
- Write buffer functions
  - — External write cycle and internal access can be executed in parallel
- Bus arbitration function
  - — Includes a bus arbiter that arbitrates bus mastership among the CPU and DTC
- Other features
  - — External bus release function

7.1.2 Block Diagram

Figure 7-1 shows a block diagram of the bus controller.

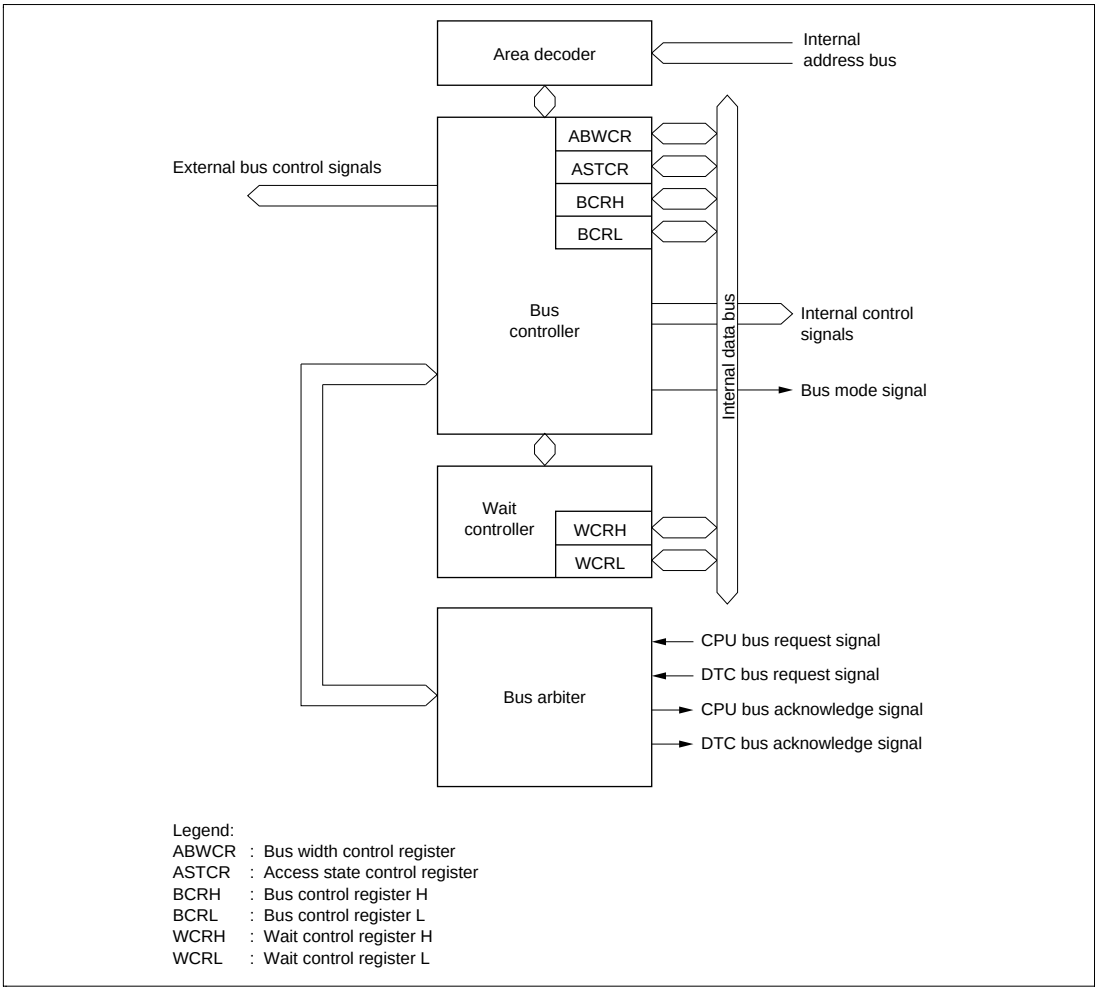


Figure 7-1 Block Diagram of Bus Controller

7.1.3 Pin Configuration

Table 7-1 summarizes the pins of the bus controller.

Table 7-1 Bus Controller Pins

Name	Symbol	I/O	Function
Address strobe	$\overline{AS}$	Output	Strobe signal indicating that address output on address bus is enabled.
Read	$\overline{RD}$	Output	Strobe signal indicating that external space is being read.
High write	$\overline{HWR}$	Output	Strobe signal indicating that external space is to be written, and upper half (D15 to D8) of data bus is enabled.
Low write	$\overline{LWR}$	Output	Strobe signal indicating that external space is to be written, and lower half (D7 to D0) of data bus is enabled.

7.1.4 Register Configuration

Table 7-2 summarizes the registers of the bus controller.

Table 7-2 Bus Controller Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
Bus width control register	ABWCR	R/W	H'FF/H'00*[2]	H'FED0
Access state control register	ASTCR	R/W	H'FF	H'FED1
Wait control register H	WCRH	R/W	H'FF	H'FED2
Wait control register L	WCRL	R/W	H'FF	H'FED3
Bus control register H	BCRH	R/W	H'D0	H'FED4
Bus control register L	BCRL	R/W	H'08	H'FED5
Pin function control register	PFCR	R/W	H'0D/H'00	H'FDEB

Notes: *1 Lower 16 bits of the address.
*2 Determined by the MCU operating mode.

7.2 Register Descriptions

7.2.1 Bus Width Control Register (ABWCR)

Bit	:	7	6	5	4	3	2	1	0
		ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
Modes 5 to 7									
Initial value	:	1	1	1	1	1	1	1	1
RW	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Mode 4									
Initial value	:	0	0	0	0	0	0	0	0
RW	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ABWCR is an 8-bit readable/writable register that designates each area for either 8-bit access or 16-bit access.

ABWCR sets the data bus width for the external memory space. The bus width for on-chip memory and internal I/O registers is fixed regardless of the settings in ABWCR.

After a reset and in hardware standby mode, ABWCR is initialized to H'FF in modes 5, 6, 7, and to H'00 in mode 4. It is not initialized in software standby mode.

Bits 7 to 0—Area 7 to 0 Bus Width Control (ABW7 to ABW0): These bits select whether the corresponding area is to be designated for 8-bit access or 16-bit access.

Bit n	
ABWn	**Description**
0	Area n is designated for 16-bit access
1	Area n is designated for 8-bit access

(n = 7 to 0)

7.2.2 Access State Control Register (ASTCR)

Bit	:	7	6	5	4	3	2	1	0
		AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ASTCR is an 8-bit readable/writable register that designates each area as either a 2-state access space or a 3-state access space.

ASTCR sets the number of access states for the external memory space. The number of access states for on-chip memory and internal I/O registers is fixed regardless of the settings in ASTCR.

ASTCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 to 0—Area 7 to 0 Access State Control (AST7 to AST0): These bits select whether the corresponding area is to be designated as a 2-state access space or a 3-state access space.

Wait state insertion is enabled or disabled at the same time.

Bit n		
ASTn	**Description**	
0	Area n is designated for 2-state access Wait state insertion in area n external space is disabled	
1	Area n is designated for 3-state access Wait state insertion in area n external space is enabled	(Initial value)

(n = 7 to 0)

7.2.3 Wait Control Registers H and L (WCRH, WCRL)

WCRH and WCRL are 8-bit readable/writable registers that select the number of program wait states for each area.

Program waits are not inserted in the case of on-chip memory or internal I/O registers.

WCRH and WCRL are initialized to H'FF by a reset and in hardware standby mode. They are not initialized in software standby mode.

WCRH

Bit	:	7	6	5	4	3	2	1	0
		W71	W70	W61	W60	W51	W50	W41	W40
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Area 7 Wait Control 1 and 0 (W71, W70): These bits select the number of program wait states when area 7 in external space is accessed while the AST7 bit in ASTCR is set to 1.

Bit 7	Bit 6	
W71	**W70**	**Description**
0	0	Program wait not inserted when external space area 7 is accessed
	1	1 program wait state inserted when external space area 7 is accessed
1	0	2 program wait states inserted when external space area 7 is accessed
	1	3 program wait states inserted when external space area 7 is accessed (Initial value)

Bits 5 and 4—Area 6 Wait Control 1 and 0 (W61, W60): These bits select the number of program wait states when area 6 in external space is accessed while the AST6 bit in ASTCR is set to 1.

Bit 5	Bit 4	
W61	**W60**	**Description**
0	0	Program wait not inserted when external space area 6 is accessed
	1	1 program wait state inserted when external space area 6 is accessed
1	0	2 program wait states inserted when external space area 6 is accessed
	1	3 program wait states inserted when external space area 6 is accessed (Initial value)

Bits 3 and 2—Area 5 Wait Control 1 and 0 (W51, W50): These bits select the number of program wait states when area 5 in external space is accessed while the AST5 bit in ASTCR is set to 1.

Bit 3	Bit 2	
W51	**W50**	**Description**
0	0	Program wait not inserted when external space area 5 is accessed
	1	1 program wait state inserted when external space area 5 is accessed
1	0	2 program wait states inserted when external space area 5 is accessed
	1	3 program wait states inserted when external space area 5 is accessed (Initial value)

Bits 1 and 0—Area 4 Wait Control 1 and 0 (W41, W40): These bits select the number of program wait states when area 4 in external space is accessed while the AST4 bit in ASTCR is set to 1.

Bit 1	Bit 0	
W41	**W40**	**Description**
0	0	Program wait not inserted when external space area 4 is accessed
	1	1 program wait state inserted when external space area 4 is accessed
1	0	2 program wait states inserted when external space area 4 is accessed
	1	3 program wait states inserted when external space area 4 is accessed (Initial value)

WCRL

Bit	:	7	6	5	4	3	2	1	0
		W31	W30	W21	W20	W11	W10	W01	W00
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Area 3 Wait Control 1 and 0 (W31, W30): These bits select the number of program wait states when area 3 in external space is accessed while the AST3 bit in ASTCR is set to 1.

Bit 7	Bit 6	
W31	**W30**	**Description**
0	0	Program wait not inserted when external space area 3 is accessed
	1	1 program wait state inserted when external space area 3 is accessed
1	0	2 program wait states inserted when external space area 3 is accessed
	1	3 program wait states inserted when external space area 3 is accessed (Initial value)

Bits 5 and 4—Area 2 Wait Control 1 and 0 (W21, W20): These bits select the number of program wait states when area 2 in external space is accessed while the AST2 bit in ASTCR is set to 1.

Bit 5	Bit 4	
W21	**W20**	**Description**
0	0	Program wait not inserted when external space area 2 is accessed
	1	1 program wait state inserted when external space area 2 is accessed
1	0	2 program wait states inserted when external space area 2 is accessed
	1	3 program wait states inserted when external space area 2 is accessed (Initial value)

Bits 3 and 2—Area 1 Wait Control 1 and 0 (W11, W10): These bits select the number of program wait states when area 1 in external space is accessed while the AST1 bit in ASTCR is set to 1.

Bit 3	Bit 2	
W11	**W10**	**Description**
0	0	Program wait not inserted when external space area 1 is accessed
	1	1 program wait state inserted when external space area 1 is accessed
1	0	2 program wait states inserted when external space area 1 is accessed
	1	3 program wait states inserted when external space area 1 is accessed (Initial value)

Bits 1 and 0—Area 0 Wait Control 1 and 0 (W01, W00): These bits select the number of program wait states when area 0 in external space is accessed while the AST0 bit in ASTCR is set to 1.

Bit 1	Bit 0	
W01	**W00**	**Description**
0	0	Program wait not inserted when external space area 0 is accessed
	1	1 program wait state inserted when external space area 0 is accessed
1	0	2 program wait states inserted when external space area 0 is accessed
	1	3 program wait states inserted when external space area 0 is accessed (Initial value)

7.2.4 Bus Control Register H (BCRH)

Bit	:	7	6	5	4	3	2	1	0
		ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—
Initial value	:	1	1	0	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BCRH is an 8-bit readable/writable register that selects enabling or disabling of idle cycle insertion, and the memory interface for area 2 to 5, and 0.

BCRH is initialized to H'D0 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Idle Cycle Insert 1 (ICIS1): Selects whether or not one idle cycle state is to be inserted between bus cycles when successive external read cycles are performed in different areas.

Bit 7

ICIS1	Description
0	Idle cycle not inserted in case of successive external read cycles in different areas
1	Idle cycle inserted in case of successive external read cycles in different areas (Initial value)

Bit 6—Idle Cycle Insert 0 (ICIS0): Selects whether or not one idle cycle state is to be inserted between bus cycles when successive external read and external write cycles are performed .

Bit 6

ICIS0	Description
0	Idle cycle not inserted in case of successive external read and external write cycles
1	Idle cycle inserted in case of successive external read and external write cycles (Initial value)

Bit 5—Burst ROM Enable (BRSTRM): Selects whether area 0 is used as a burst ROM interface.

Bit 5

BRSTRM	Description
0	Area 0 is basic bus interface (Initial value)
1	Area 0 is burst ROM interface

Bit 4—Burst Cycle Select 1 (BRSTS1): Selects the number of burst cycles for the burst ROM interface.

Bit 4	
BRSTS1	**Description**
0	Burst cycle comprises 1 state
1	Burst cycle comprises 2 states (Initial value)

Bit 3—Burst Cycle Select 0 (BRSTS0): Selects the number of words that can be accessed in a burst ROM interface burst access.

Bit 3	
BRSTS0	**Description**
0	Max. 4 words in burst access (Initial value)
1	Max. 8 words in burst access

Bits 2 to 0—Reserved: Only 0 should be written to these bits.

7.2.5 Bus Control Register L (BCRL)

Bit :	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	WDBE	—
Initial value :	0	0	0	0	1	0	0	0
R/W :	R/W	R/W	—	R/W	R/W	R/W	R/W	R/W

BCRL is an 8-bit readable/writable register that performs selection of the external bus-released state protocol, enabling or disabling of the write data buffer function.

BCRL is initialized to H'08 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 and 6—Reserved: Only 0 should be written to these bits.

Bit 5—Reserved: It is always read as 0. Cannot be written to.

Bit 4—Reserved: Only 0 should be written to this bit.

Bit 3—Reserved: Only 1 should be written to this bit.

Bit 2—Reserved: Only 0 should be written to this bit.

Bit 1—Write Data Buffer Enable (WDBE): This bit selects whether or not to use the write buffer function in the external write cycle.

Bit 1		
WDBE	**Description**	
0	Write data buffer function not used	(Initial value)
1	Write data buffer function used	

Bit 0—Reserved: Only 0 should be written to these bits.

7.2.6 Pin Function Control Register (PFCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	AE3	AE2	AE1	AE0
Initial value	:	0	0	0	0	1/0	1/0	0	1/0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PFCR is an 8-bit read/write register that controls the address output in expanded mode with ROM.

PFCR is initialized to H'0D/H'00 by a reset and in hardware standby mode. It retains its previous state in software standby mode.

Bits 7 to 4—Reserved: Only 0 should be written to these bits.

Bits 3 to 0—Address Output Enable 3 to 0 (AE3–AE0): These bits select enabling or disabling of address outputs A8 to A23 in ROMless expanded mode and modes with ROM. When a pin is enabled for address output, the address is output regardless of the corresponding DDR setting. When a pin is disabled for address output, it becomes an output port when the corresponding DDR bit is set to 1.

Bit 3	Bit 2	Bit 1	Bit 0	
AE3	**AE2**	**AE1**	**AE0**	**Description**
0	0	0	0	A8–A23 address output disabled (Initial value*)
			1	A8 address output enabled; A9–A23 address output disabled
		1	0	A8, A9 address output enabled; A10–A23 address output disabled
			1	A8–A10 address output enabled; A11–A23 address output disabled
	1	0	0	A8–A11 address output enabled; A12–A23 address output disabled
			1	A8–A12 address output enabled; A13–A23 address output disabled
		1	0	A8–A13 address output enabled; A14–A23 address output disabled
			1	A8–A14 address output enabled; A15–A23 address output disabled
1	0	0	0	A8–A15 address output enabled; A16–A23 address output disabled
			1	A8–A16 address output enabled; A17–A23 address output disabled
		1	0	A8–A17 address output enabled; A18–A23 address output disabled
			1	A8–A18 address output enabled; A19–A23 address output disabled
	1	0	0	A8–A19 address output enabled; A20–A23 address output disabled
			1	A8–A20 address output enabled; A21–A23 address output disabled (Initial value*)
		1	0	A8–A21 address output enabled; A22, A23 address output disabled
			1	A8–A23 address output enabled

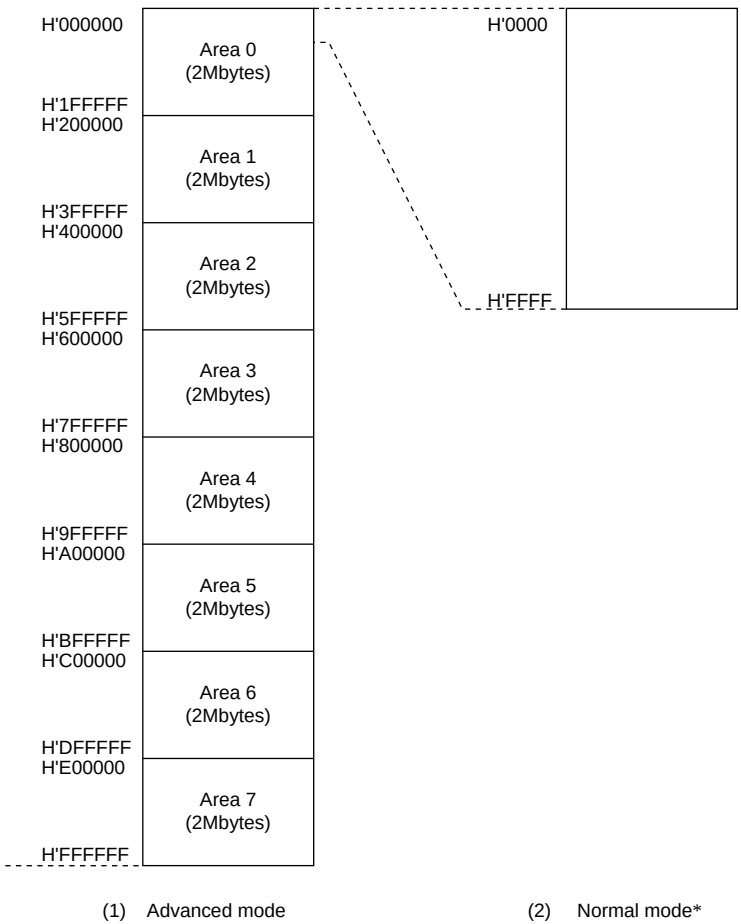
Note: * In expanded mode with ROM, bits AE3 to AE0 are initialized to B'0000.
In ROMless expanded mode, bits AE3 to AE0 are initialized to B'1101.
Address pins A0 to A7 are made address outputs by setting the corresponding DDR bits to 1.

7.3 Overview of Bus Control

7.3.1 Area Partitioning

In advanced mode, the bus controller partitions the 16 Mbytes address space into eight areas, 0 to 7, in 2-Mbyte units, and performs bus control for external space in area units. In normal mode*, it controls a 64-kbyte address space comprising part of area 0. Figure 7-2 shows an outline of the memory map.

Note: * Not available in the H8S/2636 and H8S/2638.



Note: * Not available in the H8S/2636 and H8S/2638.

Figure 7-2 Overview of Area Partitioning

7.3.2 Bus Specifications

The external space bus specifications consist of three elements: bus width, number of access states, and number of program wait states.

The bus width and number of access states for on-chip memory and internal I/O registers are fixed, and are not affected by the bus controller.

Bus Width: A bus width of 8 or 16 bits can be selected with ADWCR. An area for which an 8-bit bus is selected functions as an 8-bit access space, and an area for which a 16-bit bus is selected functions as a16-bit access space.

If all areas are designated for 8-bit access, 8-bit bus mode is set; if any area is designated for 16-bit access, 16-bit bus mode is set. When the burst ROM interface is designated, 16-bit bus mode is always set.

Number of Access States: Two or three access states can be selected with ASTCR. An area for which 2-state access is selected functions as a 2-state access space, and an area for which 3-state access is selected functions as a 3-state access space.

With the burst ROM interface, the number of access states may be determined without regard to ASTCR.

When 2-state access space is designated, wait insertion is disabled.

Number of Program Wait States: When 3-state access space is designated by ASTCR, the number of program wait states to be inserted automatically is selected with WCRH and WCRL. From 0 to 3 program wait states can be selected.

Table 7-3 shows the bus specifications for each basic bus interface area.

Table 7-3 Bus Specifications for Each Area (Basic Bus Interface)

ABWCR	ASTCR	WCRH, WCRL		Bus Specifications (Basic Bus Interface)		
ABWn	ASTn	Wn1	Wn0	Bus Width	Access States	Program Wait States
0	0	—	—	16	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3
1	0	—	—	8	2	0
	1	0	0		3	0
			1			1
		1	0			2
			1			3

7.3.3 Memory Interfaces

The chip's memory interfaces comprise a basic bus interface that allows direct connection or ROM, SRAM, and so on, and a burst ROM interface that allows direct connection of burst ROM. The memory interface can be selected independently for each area.

An area for which the basic bus interface is designated functions as normal space, and an area for which the burst ROM interface is designated functions as burst ROM space.

7.3.4 Interface Specifications for Each Area

The initial state of each area is basic bus interface, 3-state access space. The initial bus width is selected according to the operating mode. The bus specifications described here cover basic items only, and the sections on each memory interface (7.4 and 7.5) should be referred to for further details.

Area 0: Area 0 includes on-chip ROM, and in ROM-disabled expansion mode, all of area 0 is external space. In ROM-enabled expansion mode, the space excluding on-chip ROM is external space.

Either basic bus interface or burst ROM interface can be selected for area 0.

Areas 1 to 6: In external expansion mode, all of areas 1 to 6 is external space.

Only the basic bus interface can be used for areas 1 to 6.

Area 7: Area 7 includes the on-chip RAM and internal I/O registers. In external expansion mode, the space excluding the on-chip RAM and internal I/O registers is external space. The on-chip RAM is enabled when the RAME bit in the system control register (SYSCR) is set to 1; when the RAME bit is cleared to 0, the on-chip RAM is disabled and the corresponding space becomes external space.

Only the basic bus interface can be used for the area 7.

7.4 Basic Bus Interface

7.4.1 Overview

The basic bus interface enables direct connection of ROM, SRAM, and so on.

The bus specifications can be selected with ABWCR, ASTCR, WCRH, and WCRL (see table 7-3).

7.4.2 Data Size and Data Alignment

Data sizes for the CPU and other internal bus masters are byte, word, and longword. The bus controller has a data alignment function, and when accessing external space, controls whether the upper data bus (D15 to D8) or lower data bus (D7 to D0) is used according to the bus specifications for the area being accessed (8-bit access space or 16-bit access space) and the data size.

8-Bit Access Space: Figure 7-3 illustrates data alignment control for the 8-bit access space. With the 8-bit access space, the upper data bus (D15 to D8) is always used for accesses. The amount of data that can be accessed at one time is one byte: a word transfer instruction is performed as two byte accesses, and a longword transfer instruction, as four byte accesses.

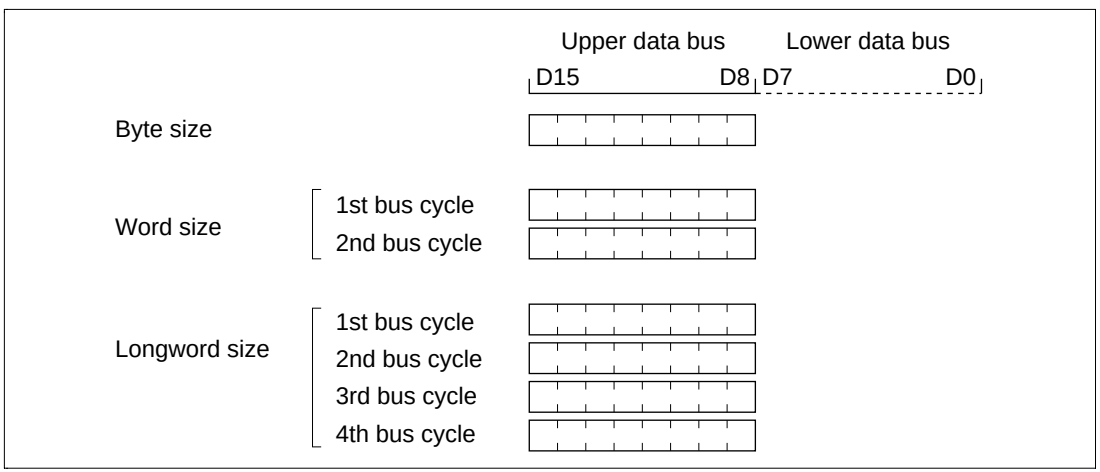


Figure 7-3 Access Sizes and Data Alignment Control (8-Bit Access Space)

16-Bit Access Space: Figure 7-4 illustrates data alignment control for the 16-bit access space. With the 16-bit access space, the upper data bus (D15 to D8) and lower data bus (D7 to D0) are used for accesses. The amount of data that can be accessed at one time is one byte or one word, and a longword transfer instruction is executed as two word transfer instructions.

In byte access, whether the upper or lower data bus is used is determined by whether the address is even or odd. The upper data bus is used for an even address, and the lower data bus for an odd address.

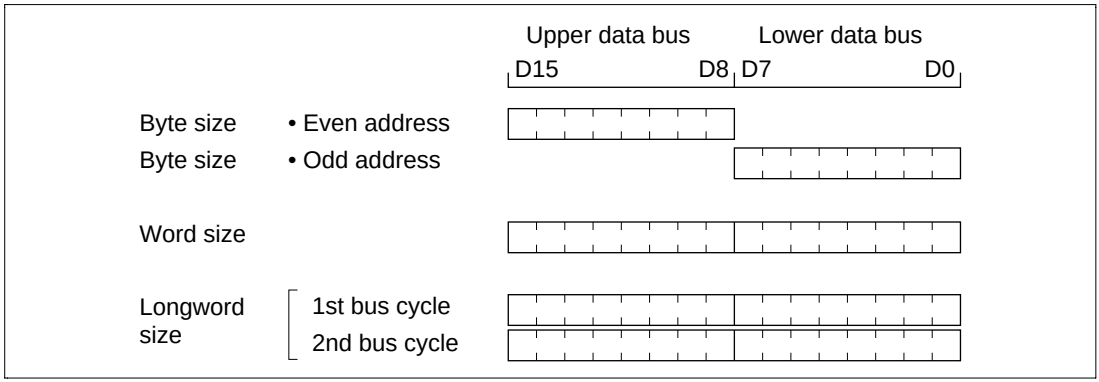


Figure 7-4 Access Sizes and Data Alignment Control (16-Bit Access Space)

7.4.3 Valid Strobes

Table 7-4 shows the data buses used and valid strobes for the access spaces.

In a read, the $\overline{RD}$ signal is valid without discrimination between the upper and lower halves of the data bus.

In a write, the $\overline{HWR}$ signal is valid for the upper half of the data bus, and the $\overline{LWR}$ signal for the lower half.

Table 7-4 Data Buses Used and Valid Strobes

Area	Access Size	Read/ Write	Address	Valid Strobe	Upper Data Bus (D15 to D8)	Lower data bus (D7 to D0)
8-bit access space	Byte	Read	—	$\overline{RD}$	Valid	Invalid
		Write	—	$\overline{HWR}$		Hi-Z
16-bit access space	Byte	Read	Even	$\overline{RD}$	Valid	Invalid
			Odd		Invalid	Valid
		Write	Even	$\overline{HWR}$	Valid	Hi-Z
			Odd	$\overline{LWR}$	Hi-Z	Valid
	Word	Read	—	$\overline{RD}$	Valid	Valid
		Write	—	$\overline{HWR}$, $\overline{LWR}$	Valid	Valid

Note: Hi-Z: High impedance.
Invalid: Input state; input value is ignored.

7.4.4 Basic Timing

8-Bit 2-State Access Space: Figure 7-5 shows the bus timing for an 8-bit 2-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used.

The $\overline{\text{LWR}}$ pin is fixed high. Wait states cannot be inserted.

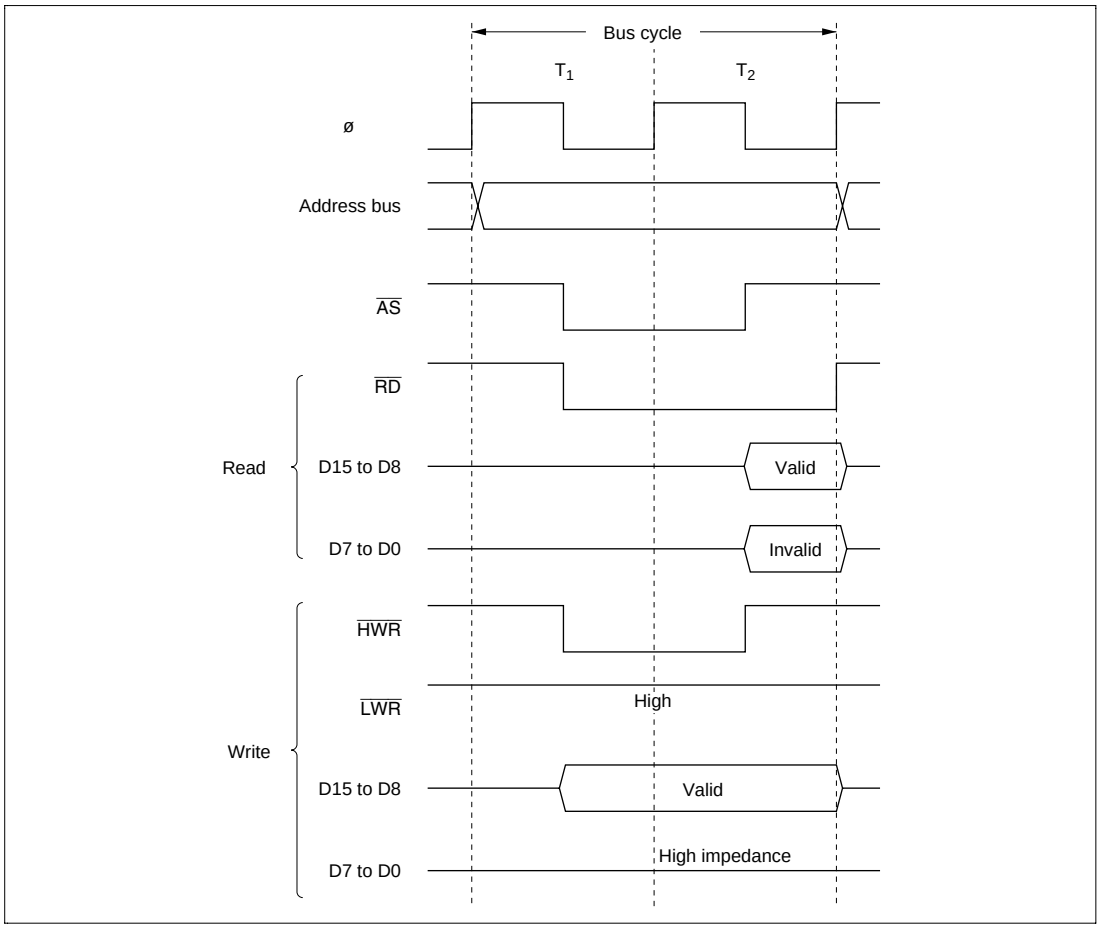


Figure 7-5 Bus Timing for 8-Bit 2-State Access Space

8-Bit 3-State Access Space: Figure 7-6 shows the bus timing for an 8-bit 3-state access space. When an 8-bit access space is accessed, the upper half (D15 to D8) of the data bus is used.

The $\overline{\text{LWR}}$ pin is fixed high. Wait states can be inserted.

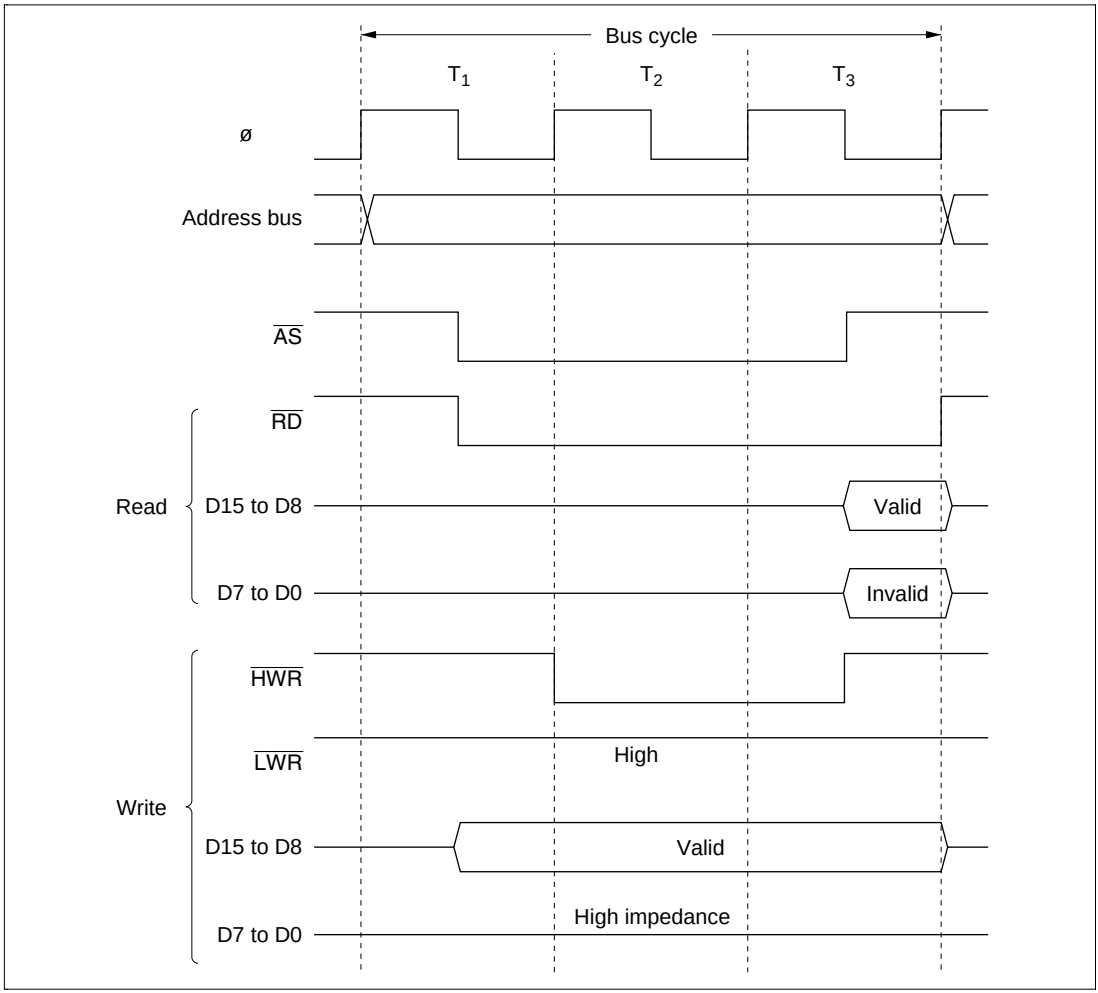


Figure 7-6 Bus Timing for 8-Bit 3-State Access Space

16-Bit 2-State Access Space: Figures 7-7 to 7-9 show bus timings for a 16-bit 2-state access space. When a 16-bit access space is accessed, the upper half (D15 to D8) of the data bus is used for the even address, and the lower half (D7 to D0) for the odd address.

Wait states cannot be inserted.

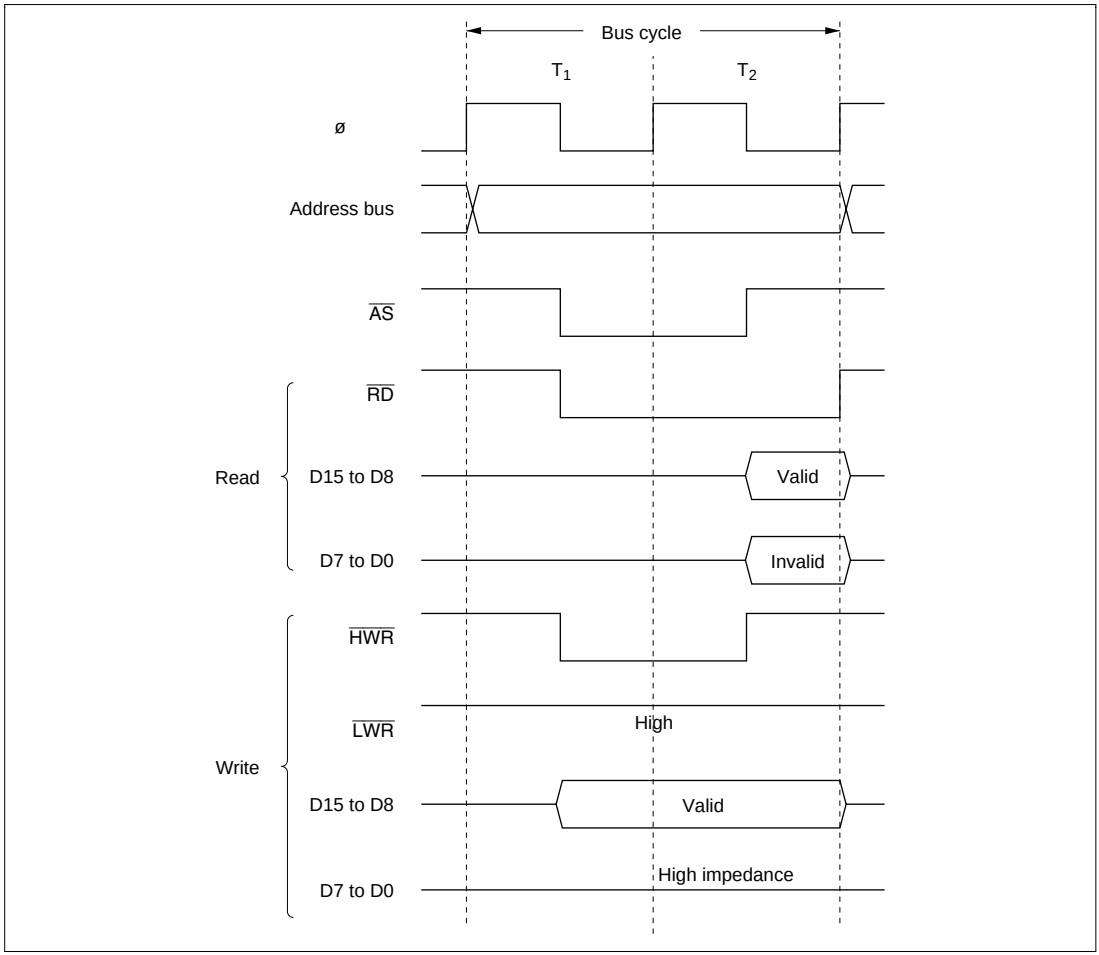


Figure 7-7 Bus Timing for 16-Bit 2-State Access Space (1) (Even Address Byte Access)

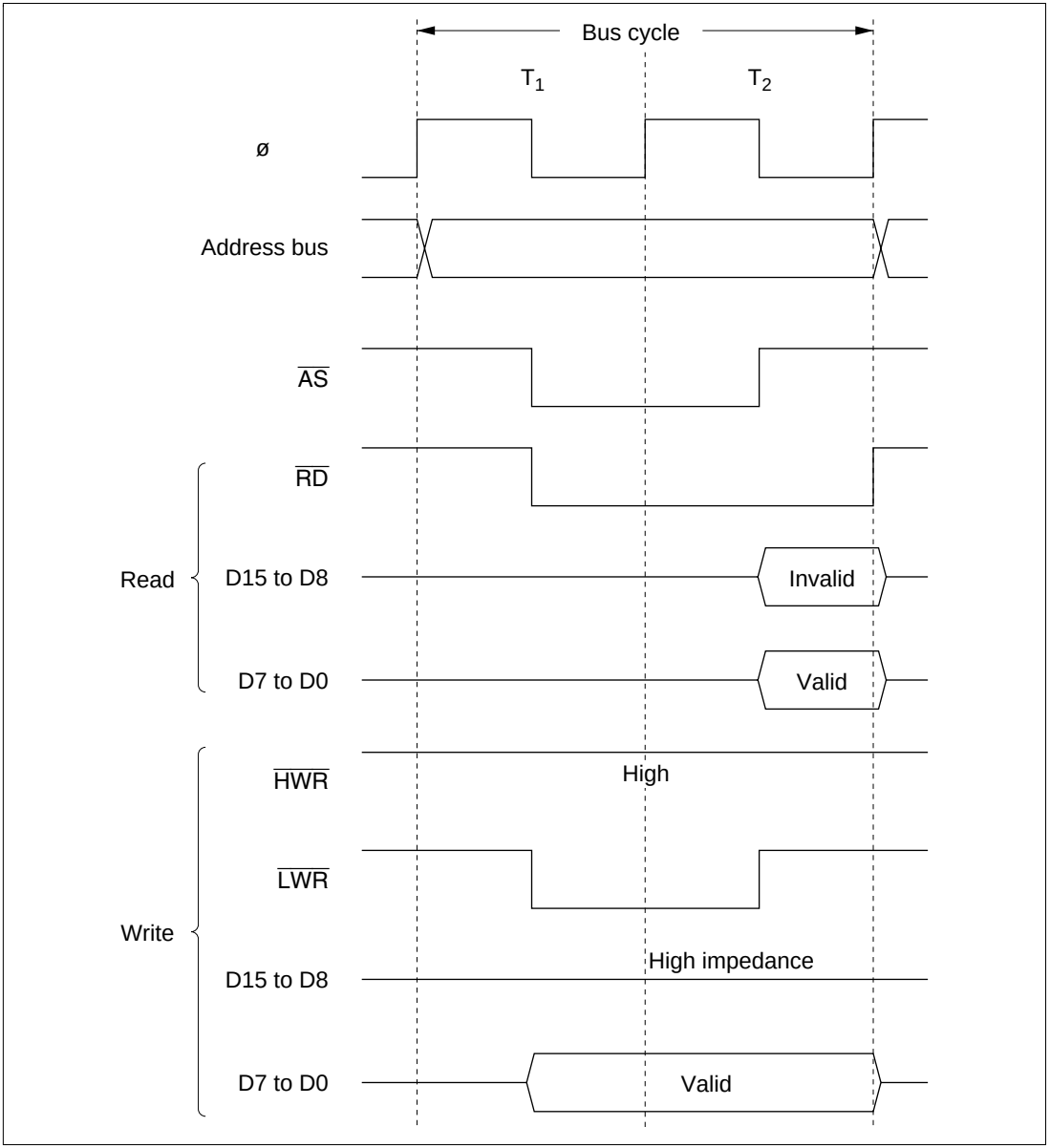


Figure 7-8 Bus Timing for 16-Bit 2-State Access Space (2) (Odd Address Byte Access)

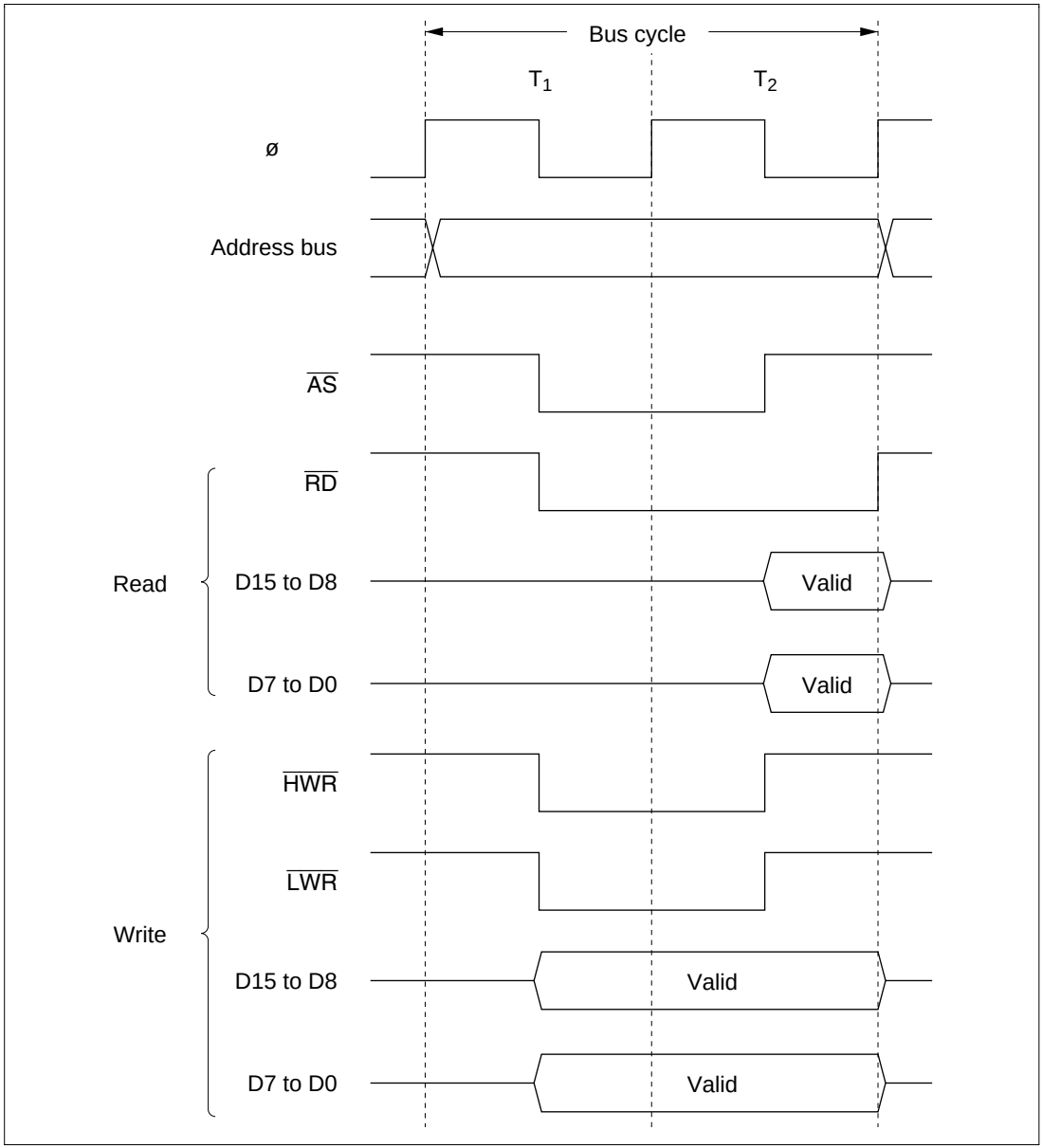


Figure 7-9 Bus Timing for 16-Bit 2-State Access Space (3) (Word Access)

16-Bit 3-State Access Space: Figures 7-10 to 7-12 show bus timings for a 16-bit 3-state access space. When a 16-bit access space is accessed , the upper half (D15 to D8) of the data bus is used for the even address, and the lower half (D7 to D0) for the odd address.

Wait states can be inserted.

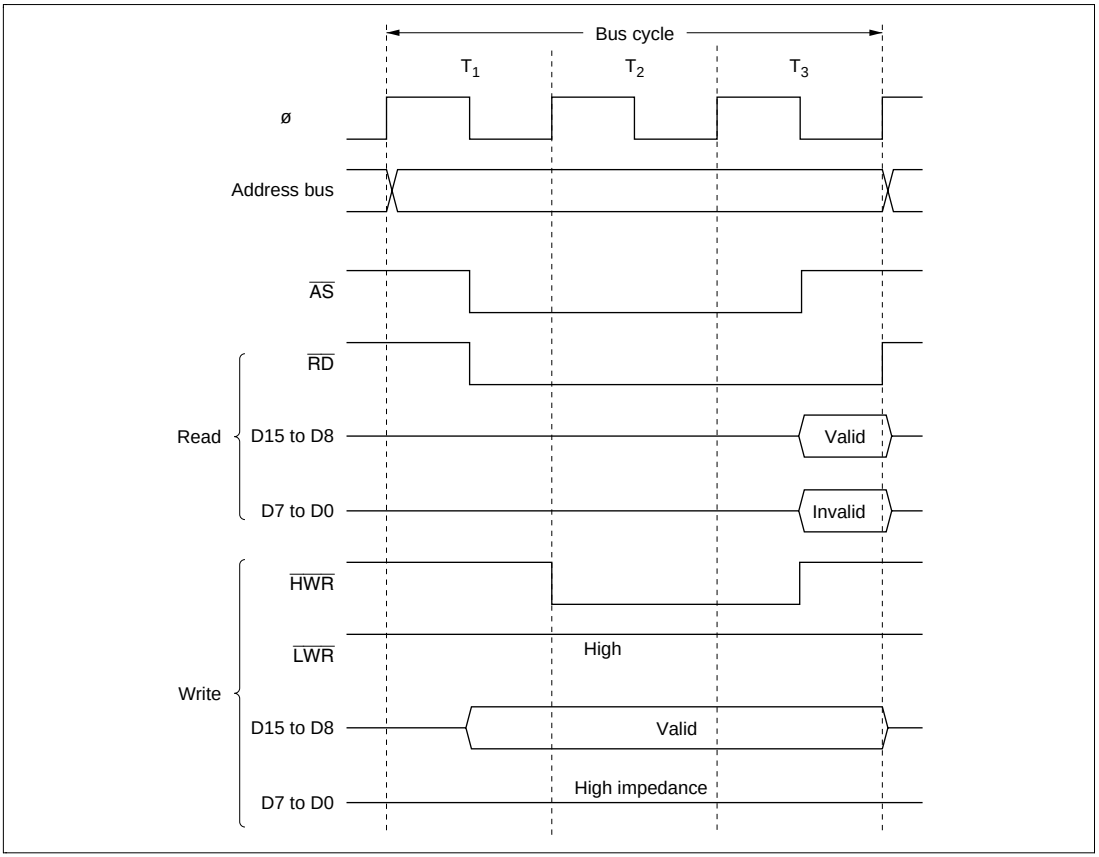


Figure 7-10 Bus Timing for 16-Bit 3-State Access Space (1) (Even Address Byte Access)

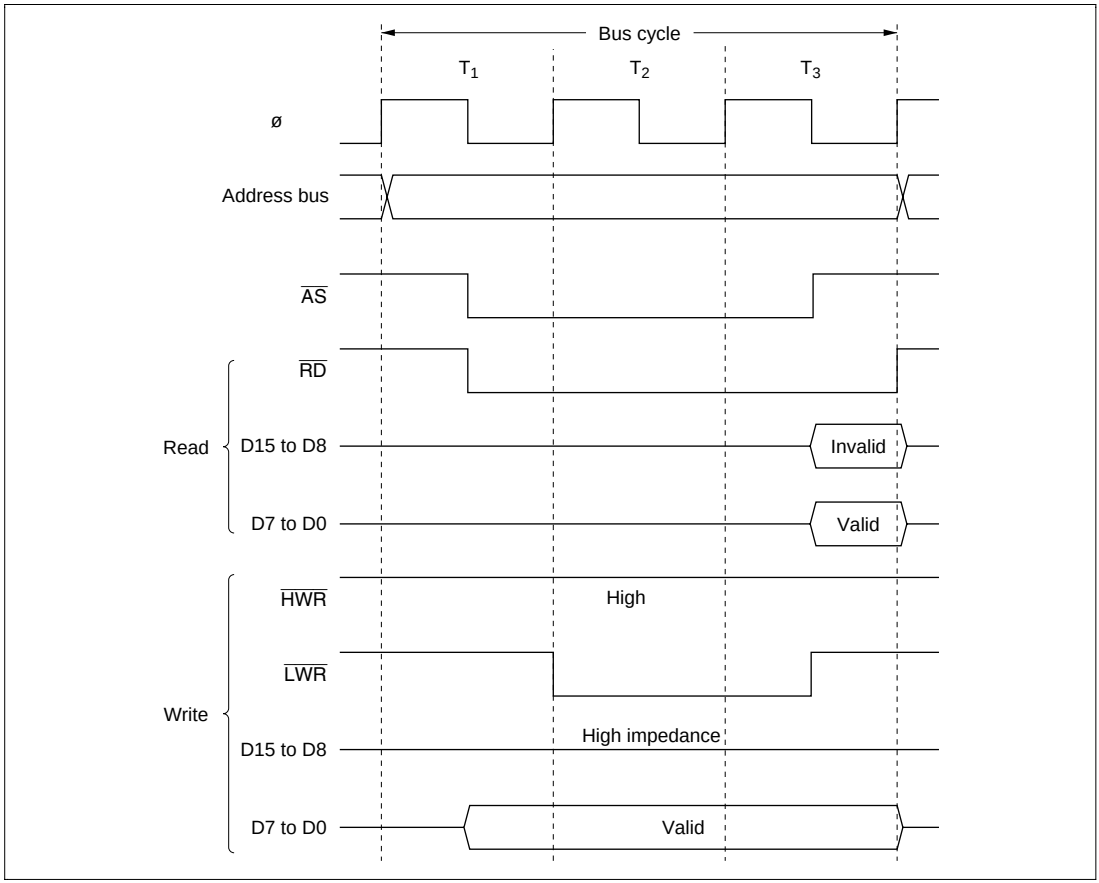


Figure 7-11 Bus Timing for 16-Bit 3-State Access Space (2) (Odd Address Byte Access)

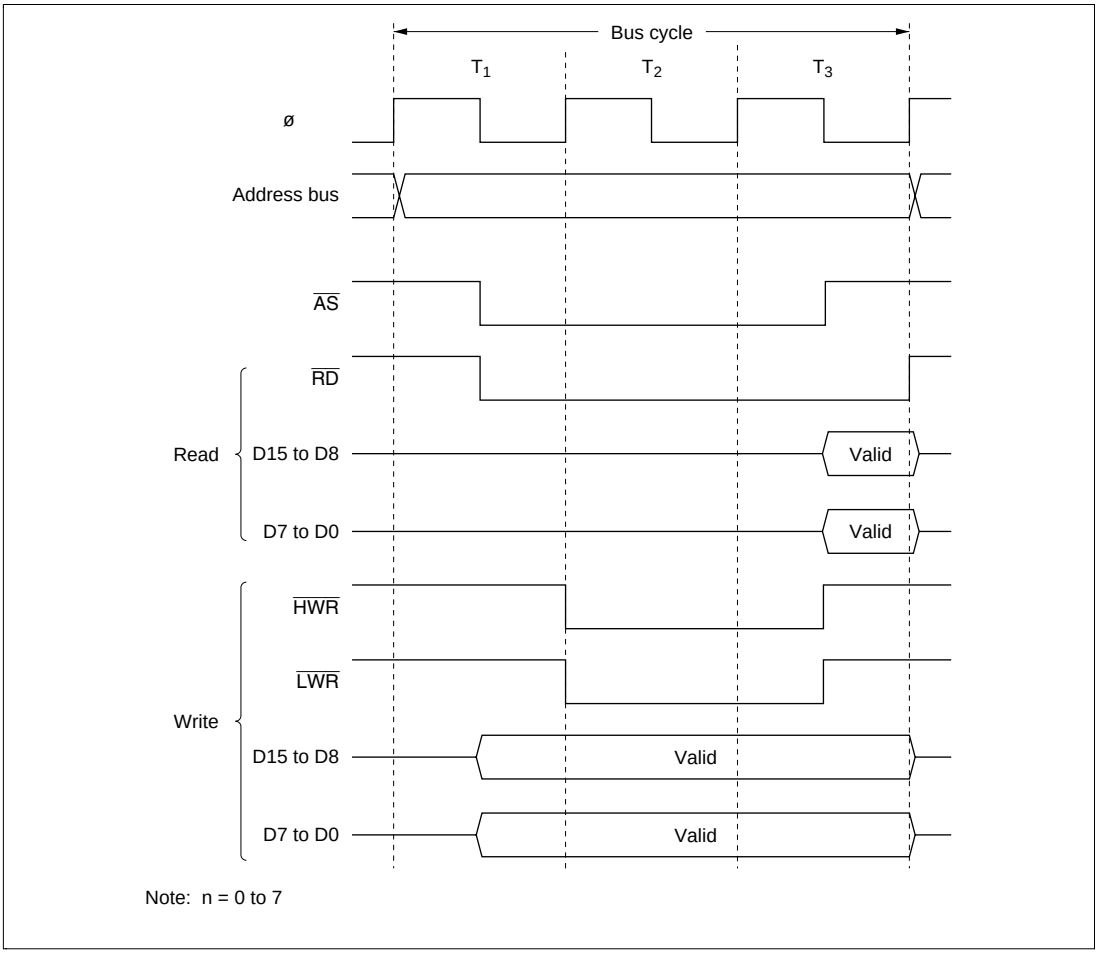


Figure 7-12 Bus Timing for 16-Bit 3-State Access Space (3) (Word Access)

7.4.5 Wait Control

When accessing external space, the chip can extend the bus cycle by inserting one or more wait states (T_w). There are two ways of inserting wait states: program wait insertion.

Program Wait Insertion

From 0 to 3 wait states can be inserted automatically between the T_2 state and T_3 state on an individual area basis in 3-state access space, according to the settings of WCRH and WCRL.

Figure 7-13 shows an example of wait state insertion timing.

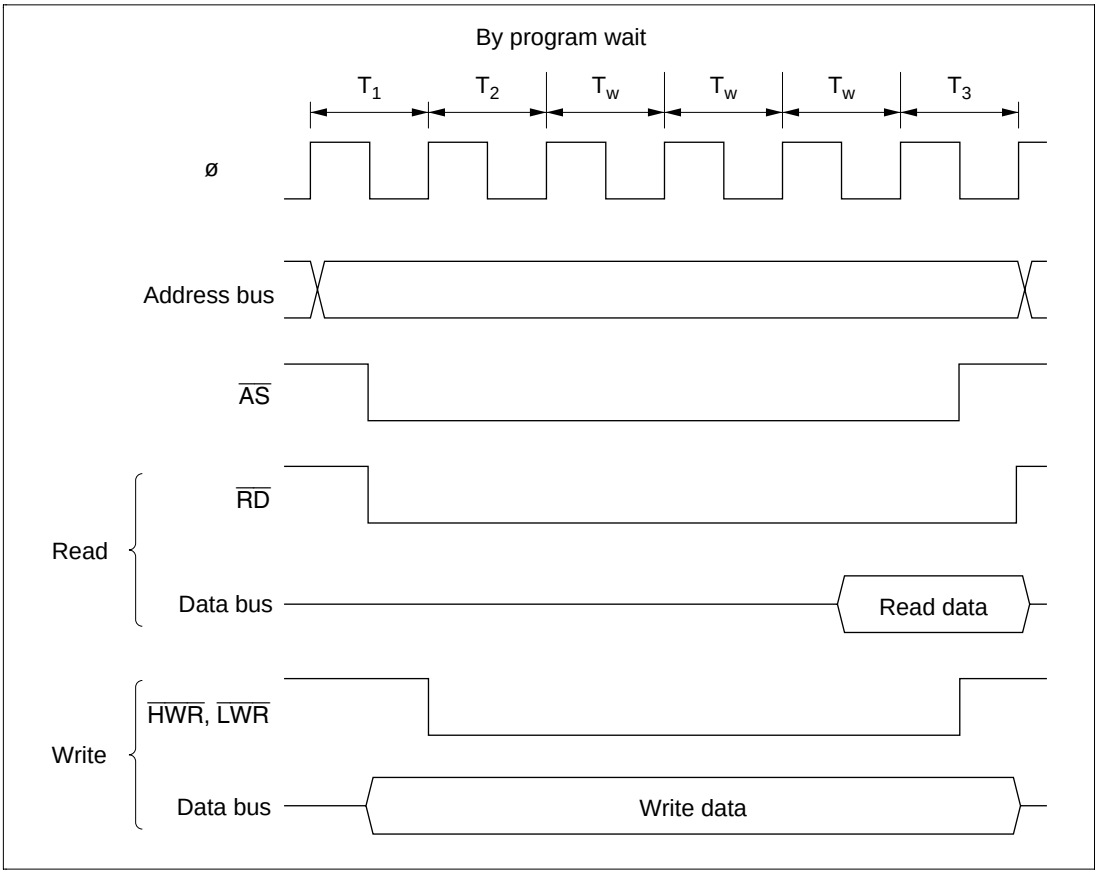


Figure 7-13 Example of Wait State Insertion Timing

The settings after a reset are: 3-state access, 3 program wait state insertion.

7.5 Burst ROM Interface

7.5.1 Overview

In this LSI, the area 0 external space can be set as burst ROM space and burst ROM interfacing performed. Burst ROM space interfacing allows 16-bit ROM capable of burst access to be accessed at high-speed.

The BRSTRM bit of BCRH sets area 0 as burst ROM space. CPU instruction fetches (only) can be performed using a maximum of 4-word or 8-word continuous burst access. 1 state or 2 states can be selected in the case of burst access.

7.5.2 Basic Timing

The AST0 bit of ASTCR sets the number of access states in the initial cycle (full access) of the burst ROM interface. Wait states can be inserted when the AST0 bit is set to 1. The burst cycle can be set for 1 state or 2 sttes by setting the BRSTS1 bit of BCRH. Wait states cannot be inserted. When area 0 is set as burst ROM space, area 0 is a 16-bit access space regardless of the ABW0 bit of ABWCR.

When the BRSTS0 bit of BCRH is cleared to 0, 4-word max. burst access is performed. When the BRSTS0 bit is set to 1, 8-word max. burst access is performed.

Figure 7-14 (a) and (b) shows the basic access timing for the burst ROM space.

Figure 7-14 (a) is an example when both the AST0 and BRSTS1 bits are set to 1.

Figure 7-14 (b) is an example when both the AST0 and BRSTS1 bits are set to 0.

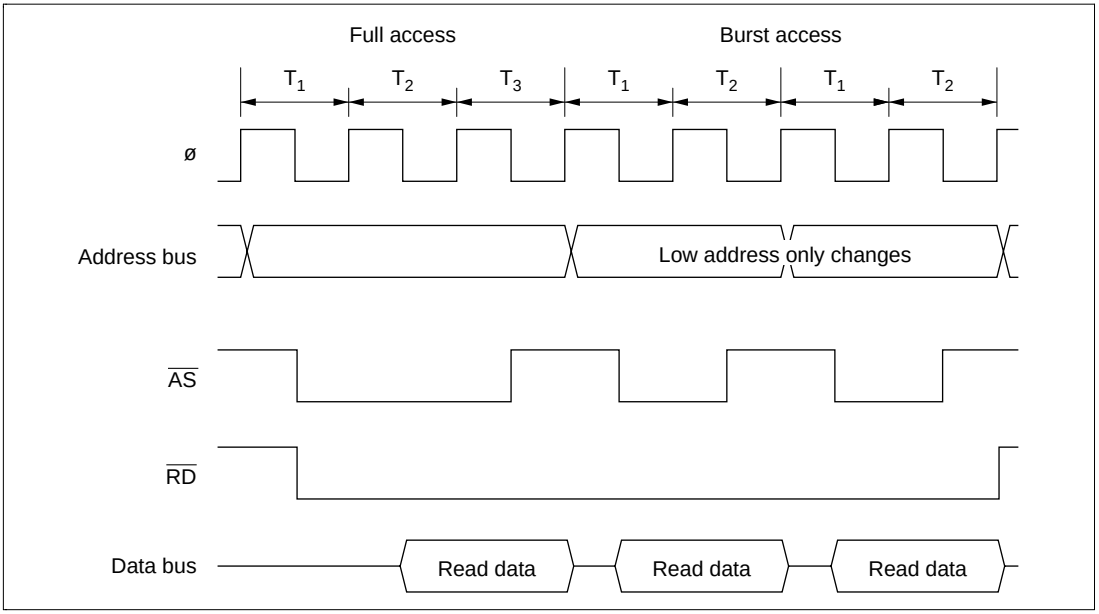


Figure 7-14 (a) Example Burst ROM Access Timing (AST0=BRSTS1=1)

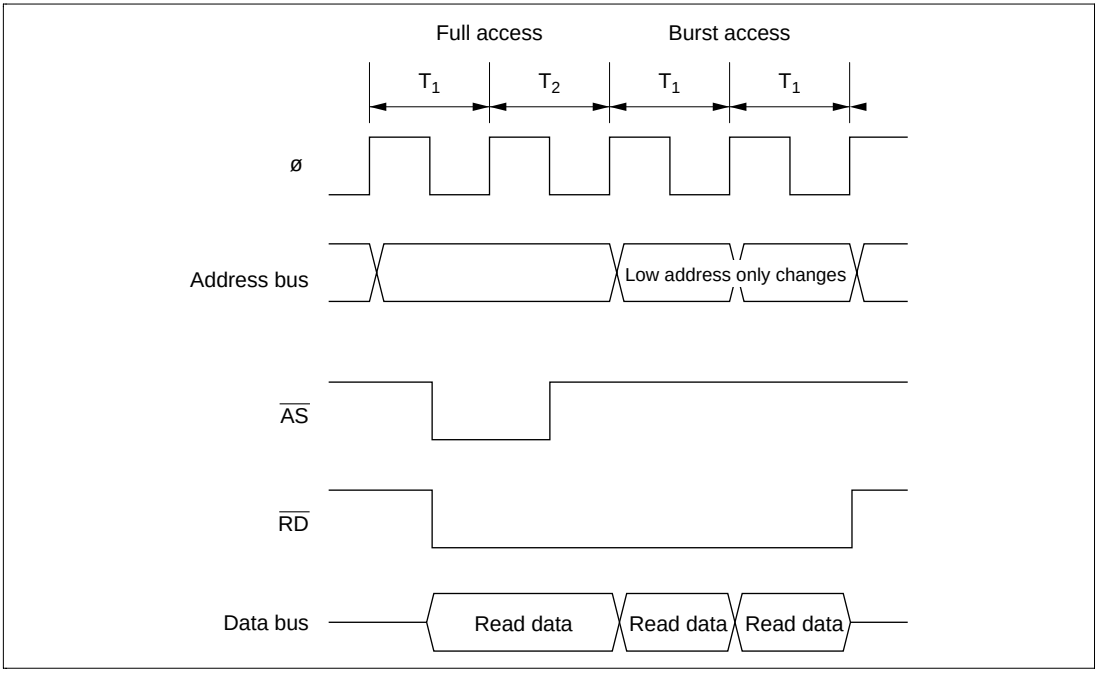


Figure 7-14 (b) Example Burst ROM Access Timing (AST0=BRSTS1=0)

7.5.3 Wait Control

As with the basic bus interface, program waits can be inserted in the burst ROM interface initial cycle (full access). See section 7.4.5, Wait Control.

Wait states cannot be inserted in the burst cycle.

7.6 Idle Cycle

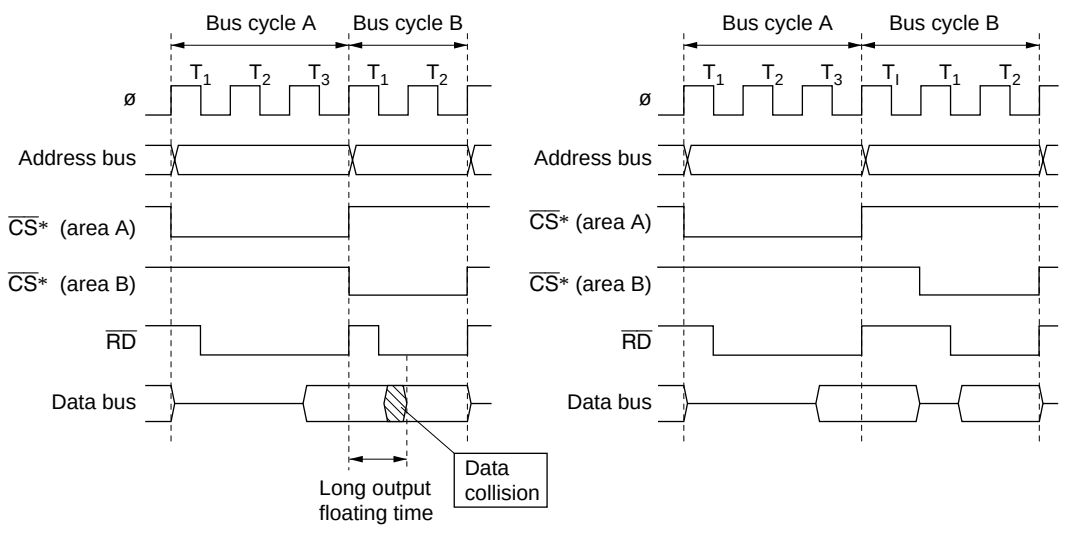
7.6.1 Operation

When the chip accesses external space , it can insert a 1-state idle cycle (T_I) between bus cycles in the following two cases: (1) when read accesses between different areas occur consecutively, and (2) when a write cycle occurs immediately after a read cycle. By inserting an idle cycle it is possible, for example, to avoid data collisions between ROM, with a long output floating time, and high-speed memory, I/O interfaces, and so on.

(1) Consecutive Reads between Different Areas

If consecutive reads between different areas occur while the ICIS1 bit in BCRH is set to 1, an idle cycle is inserted at the start of the second read cycle.

Figure 7-15 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a read cycle from SRAM, each being located in a different area. In (a), an idle cycle is not inserted, and a collision occurs in cycle B between the read data from ROM and that from SRAM. In (b), an idle cycle is inserted, and a data collision is prevented.



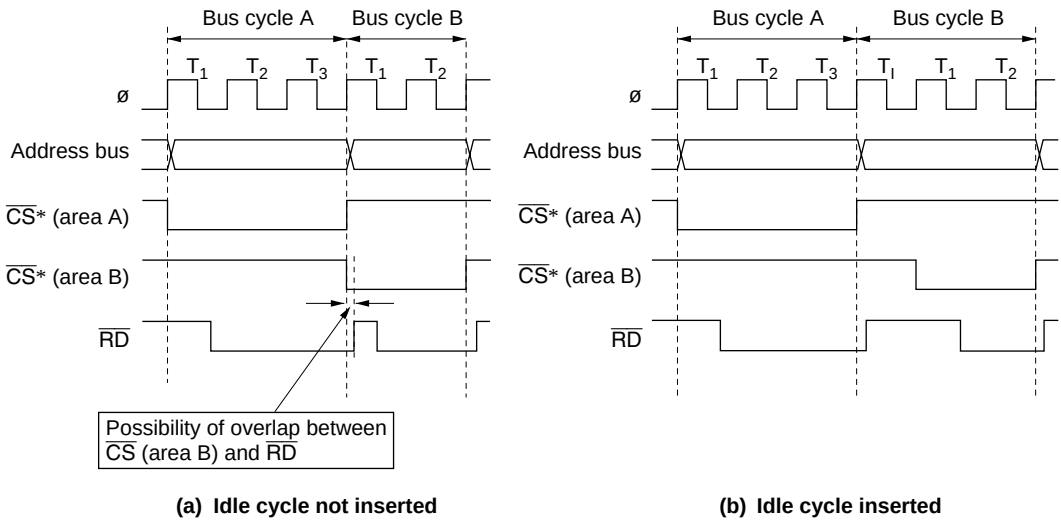
Note: * The $\overline{CS}$ signal is generated externally rather than inside the LSI device.

Figure 7-15 Example of Idle Cycle Operation (1)

(2) Write after Read

If an external write occurs after an external read while the ICIS0 bit in BCRH is set to 1, an idle cycle is inserted at the start of the write cycle.

Figure 7-16 shows an example of the operation in this case. In this example, bus cycle A is a read cycle from ROM with a long output floating time, and bus cycle B is a CPU write cycle. In (a), an idle cycle is not inserted, and a collision occurs in cycle B between the read data from ROM and the CPU write data. In (b), an idle cycle is inserted, and a data collision is prevented.



Note: * The $\overline{CS}$ signal is generated externally rather than inside the LSI device.

Figure 7-16 Example of Idle Cycle Operation (2)

(3) Relationship between Chip Select ($\overline{CS}$*) Signal and Read ($\overline{RD}$) Signal

Depending on the system's load conditions, the $\overline{RD}$ signal may lag behind the $\overline{CS}$ signal*. An example is shown in figure 7-17.

In this case, with the setting for no idle cycle insertion (a), there may be a period of overlap between the bus cycle A $\overline{RD}$ signal and the bus cycle B $\overline{CS}$ signal.

Setting idle cycle insertion, as in (b), however, will prevent any overlap between the $\overline{RD}$ and $\overline{CS}$ signals.

In the initial state after reset release, idle cycle insertion (b) is set.

Note: * The $\overline{CS}$ signal is generated externally rather than inside the LSI device.

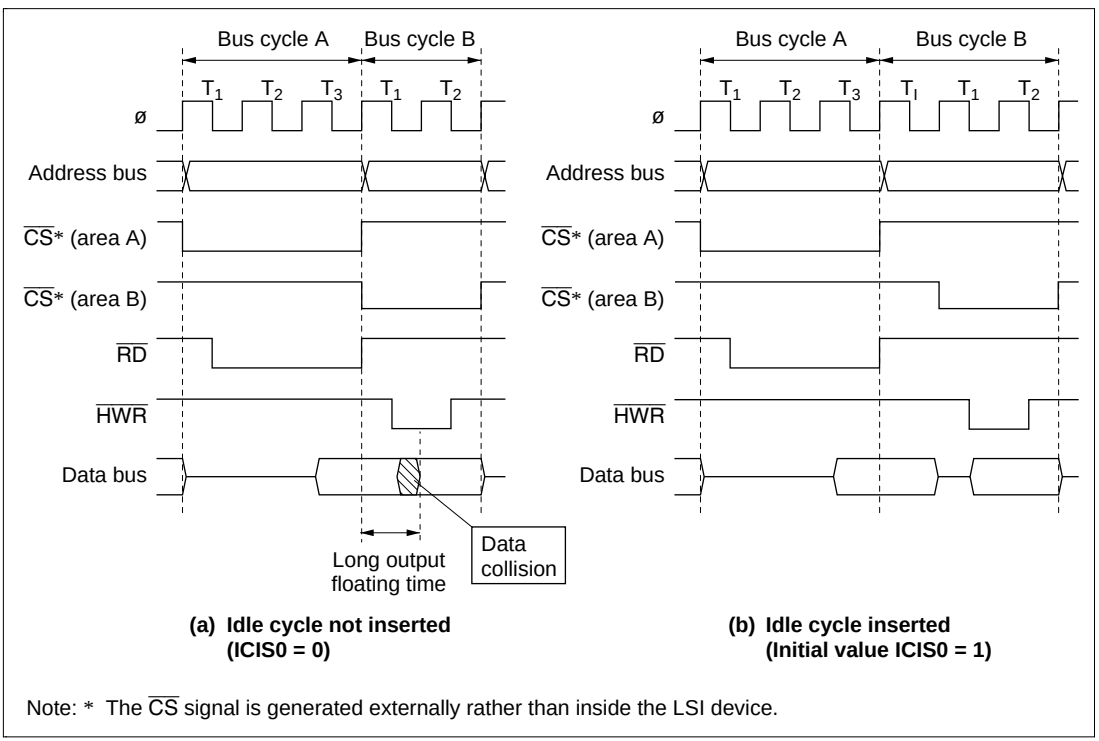


Figure 7-17 Relationship between Chip Select ($\overline{CS}$)* and Read ($\overline{RD}$)

7.6.2 Pin States During Idle Cycles

Table 7-5 shows the pin states during idle cycles.

Table 7-5 Pin States During Idle Cycles

Pins	Pin State
A23 to A0	Content identical to immediately following bus cycle
D15 to D0	High impedance
$\overline{AS}$	High level
$\overline{RD}$	High level
$\overline{HWR}$	High level
$\overline{LWR}$	High level

7.7 Write Data Buffer Function

The chip has a write data buffer function in the external data bus. Using this function enables the write data buffer to be accessed in parallel. The write data buffer function is made available by setting the WDBE bit in BCRL to 1.

Figure 7-18 shows an example of the timing when the write data buffer function is used. When this function is used, if an external write continues for 2 states or longer, and there is an internal access next, only an external write is executed in the first state, but from the next state onward an internal access (on-chip memory or internal I/O register read/write) is executed in parallel with the external write rather than waiting until it ends.

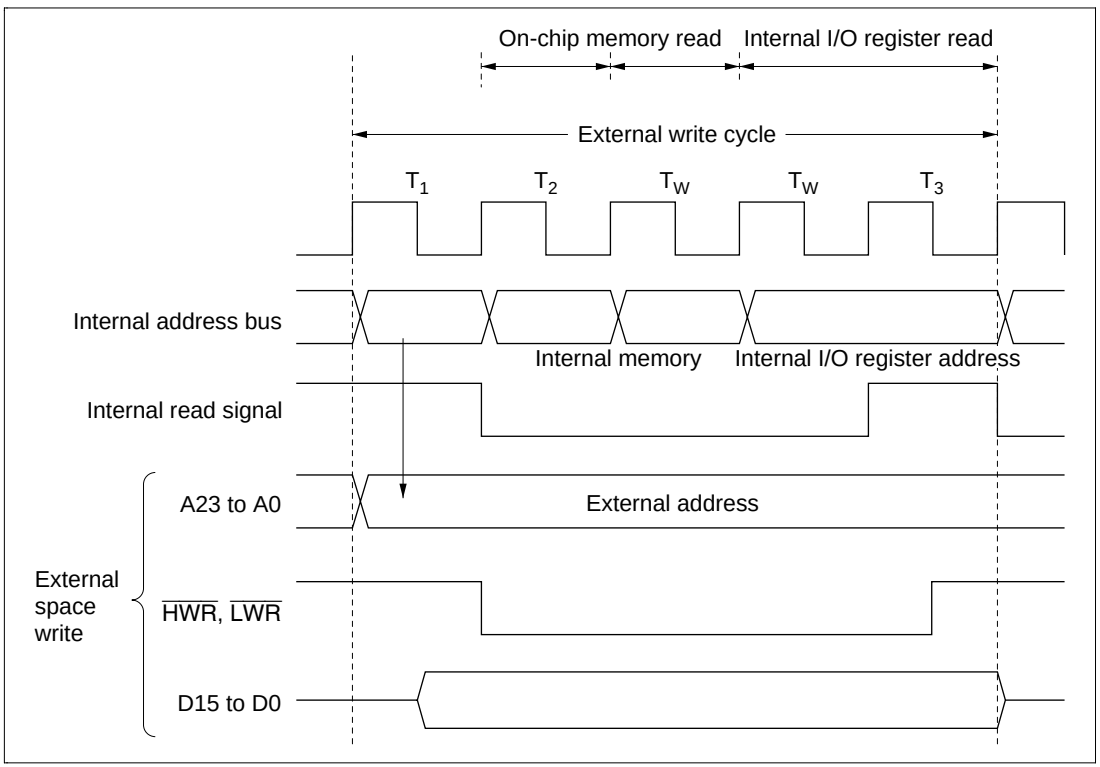


Figure 7-18 Example of Timing when Write Data Buffer Function is Used

7.8 Bus Arbitration

7.8.1 Overview

The chip has a bus arbiter that arbitrates bus master operations.

There are two bus masters, the CPU and DTC which perform read/write operations when they have possession of the bus. Each bus master requests the bus by means of a bus request signal. The bus arbiter determines priorities at the prescribed timing, and permits use of the bus by means of a bus request acknowledge signal. The selected bus master then takes possession of the bus and begins its operation.

7.8.2 Operation

The bus arbiter detects the bus masters' bus request signals, and if the bus is requested, sends a bus request acknowledge signal to the bus master making the request. If there are bus requests from more than one bus master, the bus request acknowledge signal is sent to the one with the highest priority. When a bus master receives the bus request acknowledge signal, it takes possession of the bus until that signal is canceled.

The order of priority of the bus masters is as follows:

(High) DTC > CPU (Low)

7.8.3 Bus Transfer Timing

Even if a bus request is received from a bus master with a higher priority than that of the bus master that has acquired the bus and is currently operating, the bus is not necessarily transferred immediately. There are specific times at which each bus master can relinquish the bus.

CPU: The CPU is the lowest-priority bus master, and if a bus request is received from the DTC, the bus arbiter transfers the bus to the bus master that issued the request. The timing for transfer of the bus is as follows:

- The bus is transferred at a break between bus cycles. However, if a bus cycle is executed in discrete operations, as in the case of a longword-size access, the bus is not transferred between the operations. See Appendix A-5, Bus States During Instruction Execution, for timings at which the bus is not transferred.
- If the CPU is in sleep mode, it transfers the bus immediately.

DTC: The DTC sends the bus arbiter a request for the bus when an activation request is generated.

The DTC can release the bus after a vector read, a register information read (3 states), a single data transfer, or a register information write (3 states). It does not release the bus during a register information read (3 states), a single data transfer, or a register information write (3 states).

7.9 Resets and the Bus Controller

In a reset, the chip, including the bus controller, enters the reset state at that point, and an executing bus cycle is discontinued.

Section 8 Data Transfer Controller (DTC)

8.1 Overview

The chip includes a data transfer controller (DTC). The DTC can be activated by an interrupt or software, to transfer data.

8.1.1 Features

- Transfer possible over any number of channels
  - — Transfer information is stored in memory
  - — One activation source can trigger a number of data transfers (chain transfer)
- Wide range of transfer modes
  - — Normal, repeat, and block transfer modes available
  - — Incrementing, decrementing, and fixing of source and destination addresses can be selected
- Direct specification of 16-Mbyte address space possible
  - — 24-bit transfer source and destination addresses can be specified
- Transfer can be set in byte or word units
- A CPU interrupt can be requested for the interrupt that activated the DTC
  - — An interrupt request can be issued to the CPU after one data transfer ends
  - — An interrupt request can be issued to the CPU after the specified data transfers have completely ended
- Activation by software is possible
- Module stop mode can be set
  - — The initial setting enables DTC registers to be accessed. DTC operation is halted by setting module stop mode.

8.1.2 Block Diagram

Figure 8-1 shows a block diagram of the DTC.

The DTC's register information is stored in the on-chip RAM*. A 32-bit bus connects the DTC to the on-chip RAM (1 kbyte), enabling 32-bit/1-state reading and writing of the DTC register information.

Note: * When the DTC is used, the RAME bit in SYSCR must be set to 1.

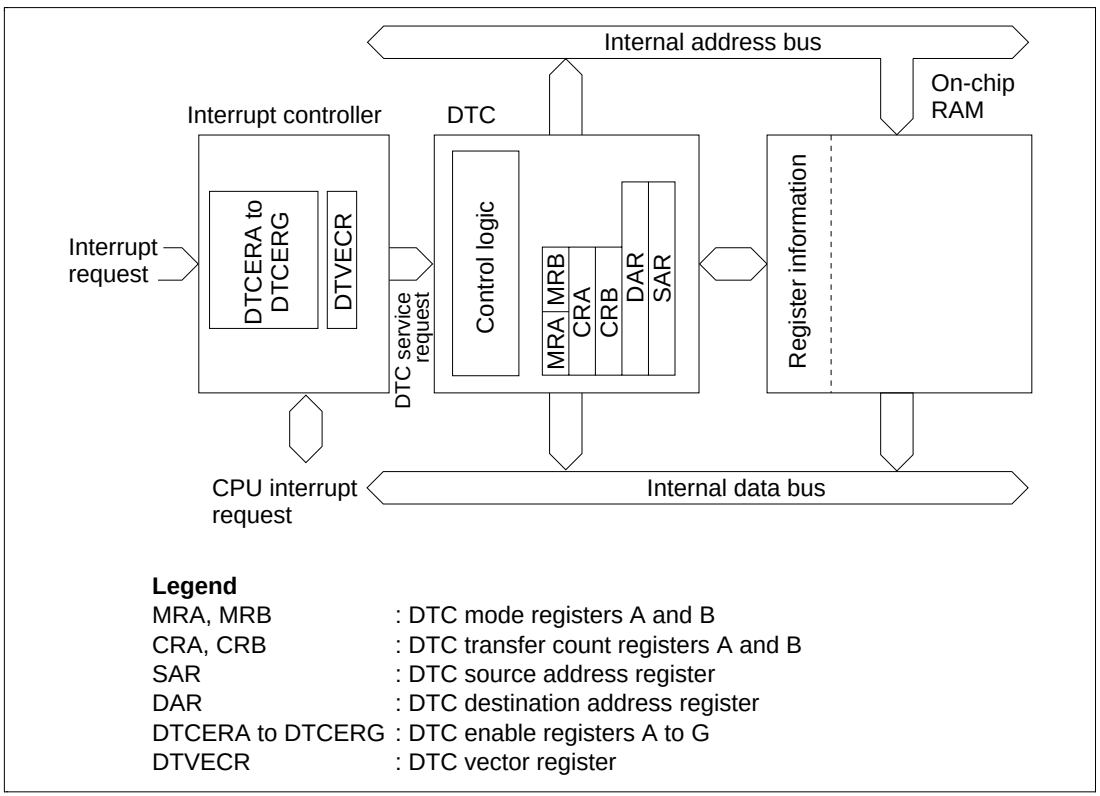


Figure 8-1 Block Diagram of DTC

8.1.3 Register Configuration

Table 8-1 summarizes the DTC registers.

Table 8-1 DTC Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
DTC mode register A	MRA	—*[2]	Undefined	—*[3]
DTC mode register B	MRB	—*[2]	Undefined	—*[3]
DTC source address register	SAR	—*[2]	Undefined	—*[3]
DTC destination address register	DAR	—*[2]	Undefined	—*[3]
DTC transfer count register A	CRA	—*[2]	Undefined	—*[3]
DTC transfer count register B	CRB	—*[2]	Undefined	—*[3]
DTC enable registers	DTCER	R/W	H'00	H'FE16 to H'FE1C
DTC vector register	DTVECR	R/W	H'00	H'FE1F
Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address.

*2 Registers within the DTC cannot be read or written to directly.

*3 Register information is located in on-chip RAM addresses H'EBC0 to H'EFBF. It cannot be located in external memory space. When the DTC is used, do not clear the RAME bit in SYSCR to 0.

8.2 Register Descriptions

8.2.1 DTC Mode Register A (MRA)

Bit	:	7	6	5	4	3	2	1	0
		SM1	SM0	DM1	DM0	MD1	MD2	DTS	Sz
Initial value	:	*	*	*	*	*	*	*	*
R/W	:	—	—	—	—	—	—	—	—

*: Undefined

MRA is an 8-bit register that controls the DTC operating mode.

Bits 7 and 6—Source Address Mode 1 and 0 (SM1, SM0): These bits specify whether SAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 7	Bit 6	
SM1	**SM0**	**Description**
0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	SAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bits 5 and 4—Destination Address Mode 1 and 0 (DM1, DM0): These bits specify whether DAR is to be incremented, decremented, or left fixed after a data transfer.

Bit 5	Bit 4	
DM1	**DM0**	**Description**
0	—	DAR is fixed
1	0	DAR is incremented after a transfer (by +1 when Sz = 0; by +2 when Sz = 1)
	1	DAR is decremented after a transfer (by –1 when Sz = 0; by –2 when Sz = 1)

Bits 3 and 2—DTC Mode (MD1, MD0): These bits specify the DTC transfer mode.

Bit 3	Bit 2	
MD1	**MD0**	**Description**
0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

Bit 1—DTC Transfer Mode Select (DTS): Specifies whether the source side or the destination side is set to be a repeat area or block area, in repeat mode or block transfer mode.

Bit 1	
DTS	**Description**
0	Destination side is repeat area or block area
1	Source side is repeat area or block area

Bit 0—DTC Data Transfer Size (Sz): Specifies the size of data to be transferred.

Bit 0	
Sz	**Description**
0	Byte-size transfer
1	Word-size transfer

8.2.2 DTC Mode Register B (MRB)

Bit	:	7	6	5	4	3	2	1	0
		CHNE	DISEL	—	—	—	—	—	—
Initial value:		*	*	*	*	*	*	*	*
R/W	:	—	—	—	—	—	—	—	—

*: Undefined

MRB is an 8-bit register that controls the DTC operating mode.

Bit 7—DTC Chain Transfer Enable (CHNE): Specifies chain transfer. With chain transfer, a number of data transfers can be performed consecutively in response to a single transfer request.

In data transfer with CHNE set to 1, determination of the end of the specified number of transfers, clearing of the interrupt source flag, and clearing of DTCER is not performed.

Bit 7

CHNE	Description
0	End of DTC data transfer (activation waiting state is entered)
1	DTC chain transfer (new register information is read, then data is transferred)

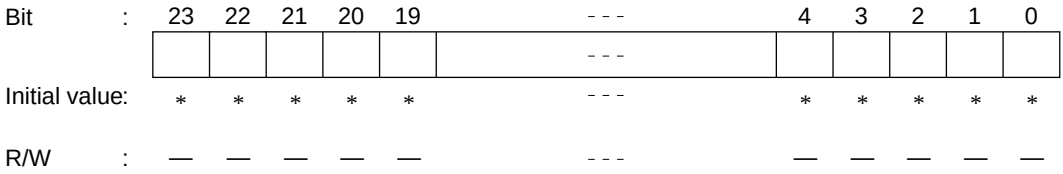
Bit 6—DTC Interrupt Select (DISEL): Specifies whether interrupt requests to the CPU are disabled or enabled after a data transfer.

Bit 6

DISEL	Description
0	After a data transfer ends, the CPU interrupt is disabled unless the transfer counter is 0 (the DTC clears the interrupt source flag of the activating interrupt to 0)
1	After a data transfer ends, the CPU interrupt is enabled (the DTC does not clear the interrupt source flag of the activating interrupt to 0)

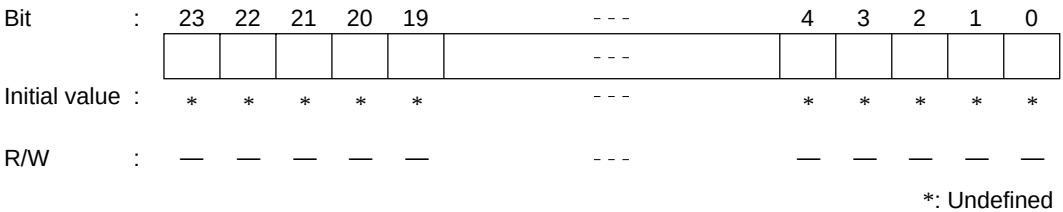
Bits 5 to 0—Reserved: These bits have no effect on DTC operation in the chip, and should always be written with 0.

8.2.3 DTC Source Address Register (SAR)



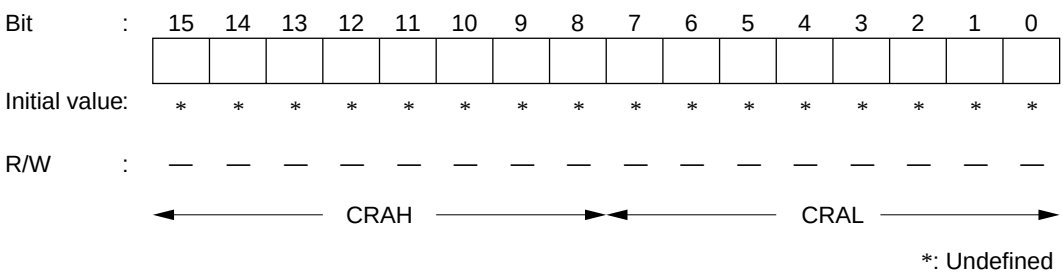
SAR is a 24-bit register that designates the source address of data to be transferred by the DTC. For word-size transfer, specify an even source address.

8.2.4 DTC Destination Address Register (DAR)



DAR is a 24-bit register that designates the destination address of data to be transferred by the DTC. For word-size transfer, specify an even destination address.

8.2.5 DTC Transfer Count Register A (CRA)



CRA is a 16-bit register that designates the number of times data is to be transferred by the DTC.

In normal mode, the entire CRA functions as a 16-bit transfer counter (1 to 65536). It is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

In repeat mode or block transfer mode, the CRA is divided into two parts: the upper 8 bits (CRAH) and the lower 8 bits (CRAL). CRAH holds the number of transfers while CRAL functions as an 8-bit transfer counter (1 to 256). CRAL is decremented by 1 every time data is transferred, and the contents of CRAH are sent when the count reaches H'00. This operation is repeated.

8.2.6 DTC Transfer Count Register B (CRB)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value:		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*
R/W	:	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

*: Undefined

CRB is a 16-bit register that designates the number of times data is to be transferred by the DTC in block transfer mode. It functions as a 16-bit transfer counter (1 to 65536) that is decremented by 1 every time data is transferred, and transfer ends when the count reaches H'0000.

8.2.7 DTC Enable Registers (DTCER)

Bit	:	7	6	5	4	3	2	1	0
		DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The DTC enable registers comprise seven 8-bit readable/writable registers, DTCERA to DTCERG with bits corresponding to the interrupt sources that can control enabling and disabling of DTC activation. These bits enable or disable DTC service for the corresponding interrupt sources.

The DTC enable registers are initialized to H'00 by a reset and in hardware standby mode.

Bit n—DTC Activation Enable (DTCEn)

Bit n DTCEn	Description	
0	DTC activation by this interrupt is disabled [Clearing conditions] • When the DISEL bit is 1 and the data transfer has ended • When the specified number of transfers have ended	(Initial value)
1	DTC activation by this interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified number of transfers have not ended	

(n = 7 to 0)

A DTCE bit can be set for each interrupt source that can activate the DTC. The correspondence between interrupt sources and DTCE bits is shown in table 9-4, together with the vector number generated for each interrupt controller.

For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR for reading and writing. If all interrupts are masked, multiple activation sources can be set at one time by writing data after executing a dummy read on the relevant register.

8.2.8 DTC Vector Register (DTVECR)

Bit	:	7	6	5	4	3	2	1	0
		SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/(W)*1	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Notes: *1 Only 1 can be written to the SWDTE bit.
*2 Bits DTVEC6 to DTVEC0 can be written to when SWDTE = 0.

DTVECR is an 8-bit readable/writable register that enables or disables DTC activation by software, and sets a vector number for the software activation interrupt.

DTVECR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—DTC Software Activation Enable (SWDTE): Enables or disables DTC activation by software.

Bit 7 SWDTE	Description
0	DTC software activation is disabled (Initial value) [Clearing conditions] • When the DISEL bit is 0 and the specified number of transfers have not ended • When 0 s written to the DISEL bit after a software-activated data transfer end interrupt (SWDTEND) request has been sent to the CPU
1	DTC software activation is enabled [Holding conditions] • When the DISEL bit is 1 and data transfer has ended • When the specified number of transfers have ended • During data transfer due to software activation

Bits 6 to 0—DTC Software Activation Vectors 6 to 0 (DTVEC6 to DTVEC0): These bits specify a vector number for DTC software activation.

The vector address is expressed as H'0400 + ((vector number) << 1). <<1 indicates a one-bit left-shift. For example, when DTVEC6 to DTVEC0 = H'10, the vector address is H'0420.

8.2.9 Module Stop Control Register A (MSTPCRA)

Bit	7	6	5	4	3	2	1	0
	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	0	0	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRA is a 8-bit readable/writable register that performs module stop mode control.

When the MSTPA6 bit in MSTPCRA is set to 1, the DTC operation stops at the end of the bus cycle and a transition is made to module stop mode. However, 1 cannot be written in the MSTPA6 bit while the DTC is operating. For details, see section 23A.5, 23B.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 6—Module Stop (MSTPA6): Specifies the DTC module stop mode.

Bit 6

MSTPA6	Description	
0	DTC module stop mode cleared	(Initial value)
1	DTC module stop mode set	

8.3 Operation

8.3.1 Overview

When activated, the DTC reads register information that is already stored in memory and transfers data on the basis of that register information. After the data transfer, it writes updated register information back to memory. Pre-storage of register information in memory makes it possible to transfer data over any required number of channels. Setting the CHNE bit to 1 makes it possible to perform a number of transfers with a single activation.

Figure 8-2 shows a flowchart of DTC operation.

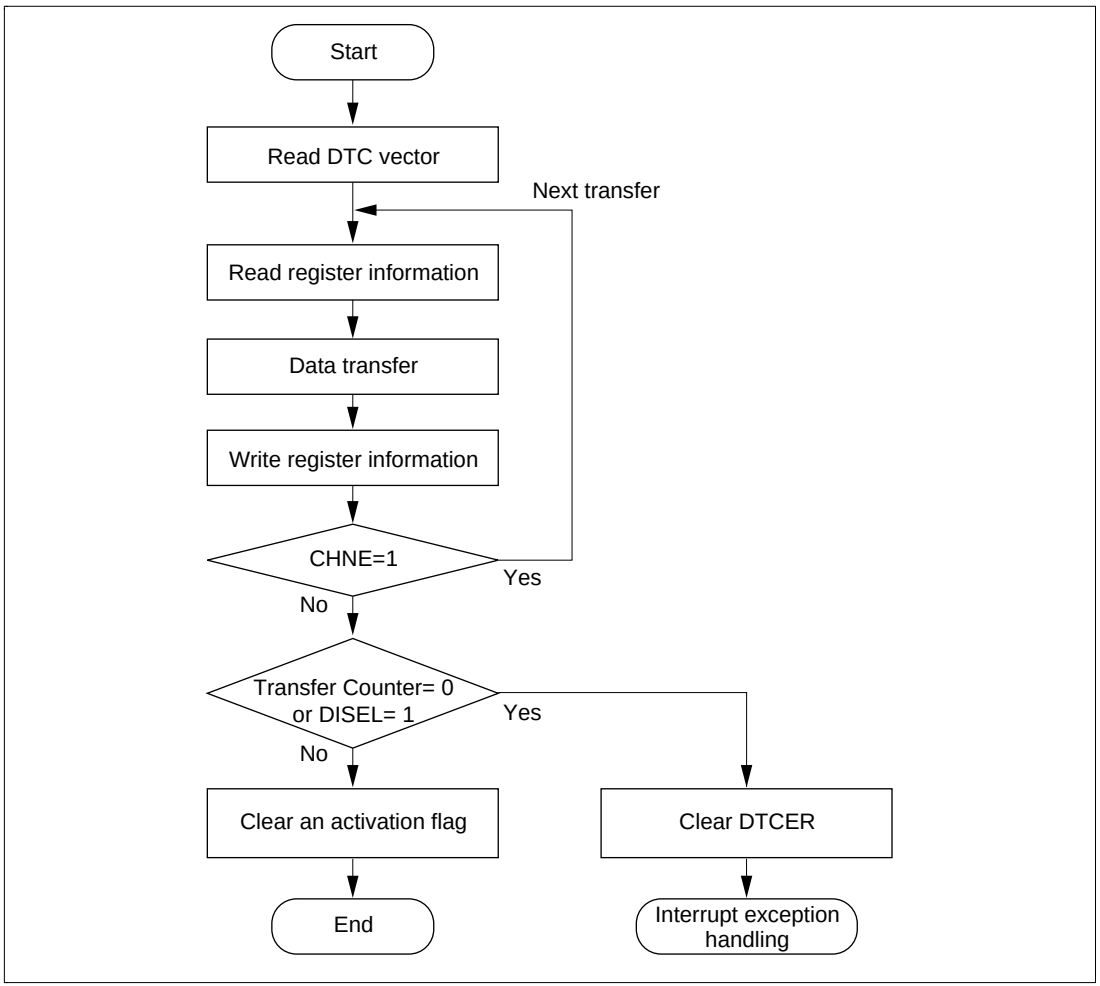


Figure 8-2 Flowchart of DTC Operation

The DTC transfer mode can be normal mode, repeat mode, or block transfer mode.

The 24-bit SAR designates the DTC transfer source address and the 24-bit DAR designates the transfer destination address. After each transfer, SAR and DAR are independently incremented, decremented, or left fixed.

Table 8-2 outlines the functions of the DTC.

Table 8-2 DTC Functions

		Address Registers	
Transfer Mode	**Activation Source**	**Transfer Source**	**Transfer Destination**
• Normal mode — One transfer request transfers one byte or one word — Memory addresses are incremented or decremented by 1 or 2 — Up to 65,536 transfers possible • Repeat mode — One transfer request transfers one byte or one word — Memory addresses are incremented or decremented by 1 or 2 — After the specified number of transfers (1 to 256), the initial state resumes and operation continues • Block transfer mode — One transfer request transfers a block of the specified size — Block size is from 1 to 256 bytes or words — Up to 65,536 transfers possible — A block area can be designated at either the source or destination	• IRQ • TPU TGI • SCI TXI or RXI • A/D converter ADI • Motor control PWM CMI • HCAN RM0 (mail box 0) • Software	24 bits	24 bits

8.3.2 Activation Sources

The DTC operates when activated by an interrupt or by a write to DTVECR by software. An interrupt request can be directed to the CPU or DTC, as designated by the corresponding DTCER bit. An interrupt becomes a DTC activation source when the corresponding bit is set to 1, and a CPU interrupt source when the bit is cleared to 0.

At the end of a data transfer (or the last consecutive transfer in the case of chain transfer), the activation source or corresponding DTCER bit is cleared. Table 8-3 shows activation source and DTCER clearance. The activation source flag, in the case of RXI0, for example, is the RDRF flag of SCI0.

Table 8-3 Activation Source and DTCER Clearance

Activation Source	When the DISEL Bit Is 0 and the Specified Number of Transfers Have Not Ended	When the DISEL Bit Is 1, or when the Specified Number of Transfers Have Ended
Software activation	The SWDTE bit is cleared to 0	The SWDTE bit remains set to 1 An interrupt is issued to the CPU
Interrupt activation	The corresponding DTCER bit remains set to 1 The activation source flag is cleared to 0	The corresponding DTCER bit is cleared to 0 The activation source flag remains set to 1 A request is issued to the CPU for the activation source interrupt

Figure 8-3 shows a block diagram of activation source control. For details see section 5, Interrupt Controller.

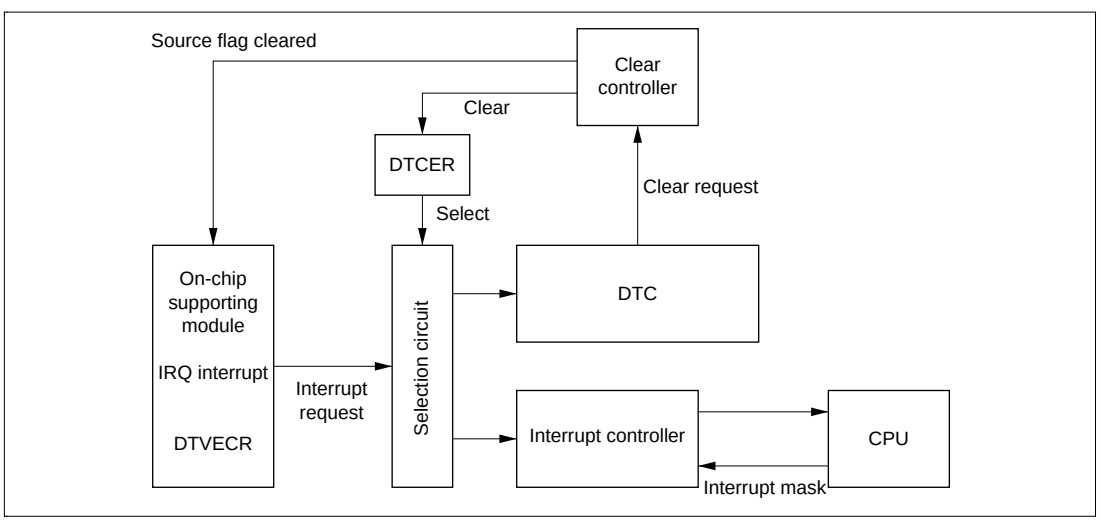


Figure 8-3 Block Diagram of DTC Activation Source Control

When an interrupt has been designated a DTC activation source, existing CPU mask level and interrupt controller priorities have no effect. If there is more than one activation source at the same time, the DTC operates in accordance with the default priorities.

8.3.3 DTC Vector Table

Figure 8-4 shows the correspondence between DTC vector addresses and register information.

Table 8-4 shows the correspondence between activation and vector addresses. When the DTC is activated by software, the vector address is obtained from: H'0400 + (DTVECR[6:0] << 1) (where << 1 indicates a 1-bit left shift). For example, if DTVECR is H'10, the vector address is H'0420.

The DTC reads the start address of the register information from the vector address set for each activation source, and then reads the register information from that start address. The register information can be placed at predetermined addresses in the on-chip RAM. The start address of the register information should be an integral multiple of four.

The configuration of the vector address is the same in both normal* and advanced modes, a 2-byte unit being used in both cases. These two bytes specify the lower bits of the address in the on-chip RAM.

Note: * Not available in the H8S/2636 and H8S/2638.

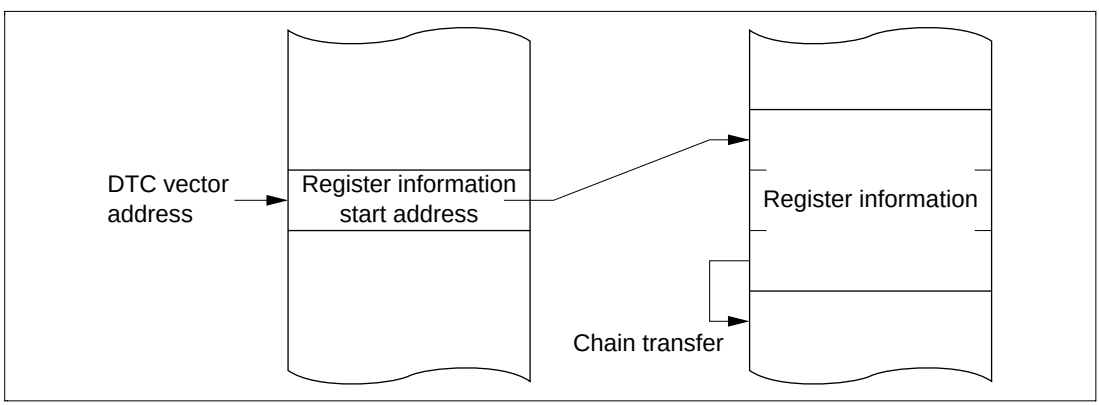


Figure 8-4 Correspondence between DTC Vector Address and Register Information

Table 8-4 Interrupt Sources, DTC Vector Addresses, and Corresponding DTCEs

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE*[1]	Priority
Write to DTVECR	Software	DTVECR	H'0400+ (DTVECR [6:0] <<1)	—	High
IRQ0	External pin	16	H'0420	DTCEA7	
IRQ1		17	H'0422	DTCEA6	
IRQ2		18	H'0424	DTCEA5	
IRQ3		19	H'0426	DTCEA4	
IRQ4		20	H'0428	DTCEA3	
IRQ5		21	H'042A	DTCEA2	
Reserved	—	22 to 27	H'042C to H'0436	—	
ADI (A/D conversion end)	A/D	28	H'0438	DTCEB6	
Reserved	—	29 to 31	H'043A to H'043E	—	
TGI0A (GR0A compare match/ input capture)	TPU channel 0	32	H'0440	DTCEB5	
TGI0B (GR0B compare match/ input capture)		33	H'0442	DTCEB4	
TGI0C (GR0C compare match/ input capture)		34	H'0444	DTCEB3	
TGI0D (GR0D compare match/ input capture)		35	H'0446	DTCEB2	
Reserved	—	36 to 39	H'0448 to H'044E	—	
TGI1A (GR1A compare match/ input capture)	TPU channel 1	40	H'0450	DTCEB1	
TGI1B (GR1B compare match/ input capture)		41	H'0452	DTCEB0	
TGI2A (GR2A compare match/ input capture)	TPU channel 2	44	H'0458	DTCEC7	
TGI2B (GR2B compare match/ input capture)		45	H'045A	DTCEC6	Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE*[1]	Priority
TGI3A (GR3A compare match/ input capture)	TPU channel 3	48	H'0460	DTCEC5	High
TGI3B (GR3B compare match/ input capture)		49	H'0462	DTCEC4	
TGI3C (GR3C compare match/ input capture)		50	H'0464	DTCEC3	
TGI3D (GR3D compare match/ input capture)		51	H'0466	DTCEC2	
Reserved	—	52 to 55	H'0468 to H'046E	—	
TGI4A (GR4A compare match/ input capture)	TPU channel 4	56	H'0470	DTCEC1	
TGI4B (GR4B compare match/ input capture)		57	H'0472	DTCEC0	
Reserved	—	58, 59	H'0474 to H'0476	—	
TGI5A (GR5A compare match/ input capture)	TPU channel 5	60	H'0478	DTCED5	
TGI5B (GR5B compare match/ input capture)		61	H'047A	DTCED4	
Reserved	—	62 to 80	H'047C to H'04A0	—	
RXI0 (reception complete 0)	SCI channel 0	81	H'04A2	DTCEE3	
TXI0 (transmit data empty 0)		82	H'04A4	DTCEE2	
Reserved	—	83, 84	H'04A6 to H'04A8	—	
RXI1 (reception complete 1)	SCI channel 1	85	H'04AA	DTCEE1	
TXI1 (transmit data empty 1)		86	H'04AC	DTCEE0	
Reserved	—	87, 88	H'04AE to H'04B0	—	
RXI2 (reception complete 2)	SCI channel 2	89	H'04B2	DTCEF7	
TXI2 (transmit data empty 2)		90	H'04B4	DTCEF6	
Reserved	—	91 to 97	H'04B6 to H'04C2	—	Low

Interrupt Source	Origin of Interrupt Source	Vector Number	Vector Address	DTCE*[1]	Priority
IICI0 (1-byte transmission/ reception completed)*[2]	IIC channel 0 (option)	100	H'04C8	DTCEF1	High
IICI1 (1-byte transmission/ reception completed)*[2]	IIC channel 1 (option)	102	H'04CC	DTCEF0	
CMI1 (PWCYR1 compare match)	PWM	104	H'04D0	DTCEG7	
CMI2 (PWCYR2 compare match)		105	H'04D2	DTCEG6	
Reserved	—	106	H'04D4	—	
RM0 (HCAN1 mail box 0)	HCAN1	107	H'04D6	DTCEG4	
Reserved	—	108	H'04D8	—	
RM0 (HCAN0 mail box 0)	HCAN0	109	H'04DA	DTCEG2	
Reserved	—	110 to 124	H'04DC to H'04F8	—	Low

Notes: *1 DTCE bits with no corresponding interrupt are reserved, and should be written with 0.

*2 I^2C bus interface is available as an option in the H8S/2638 and H8S/2639. These bits become reserved bits when this optional feature is not used or in the H8S/2636.

8.3.4 Location of Register Information in Address Space

Figure 8-5 shows how the register information should be located in the address space.

Locate the MRA, SAR, MRB, DAR, CRA, and CRB registers, in that order, from the start address of the register information (contents of the vector address). In the case of chain transfer, register information should be located in consecutive areas.

Locate the register information in the on-chip RAM (addresses: H'FFEBC0 to H'FFEFBF).

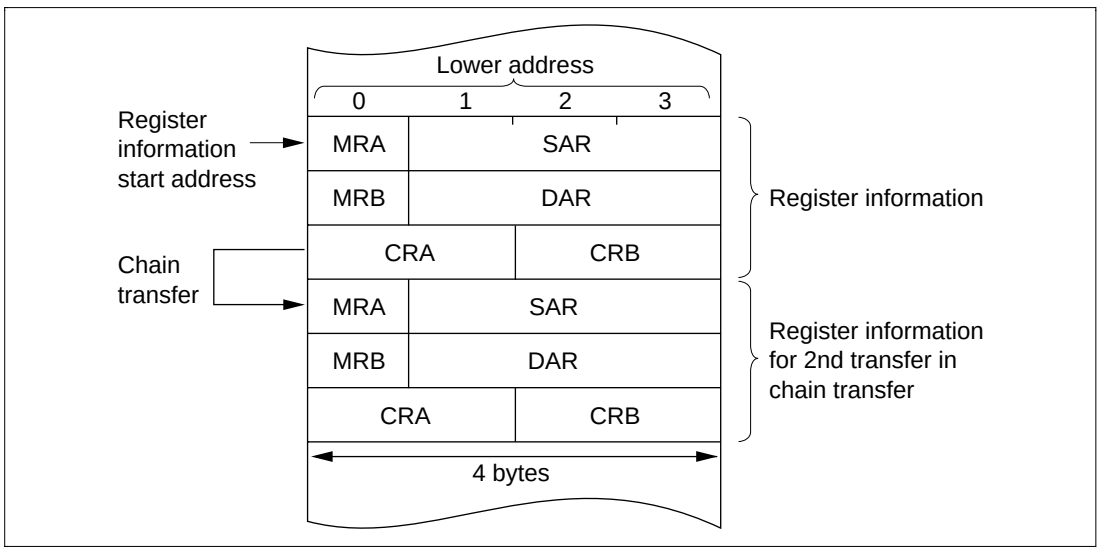


Figure 8-5 Location of Register Information in Address Space

8.3.5 Normal Mode

In normal mode, one operation transfers one byte or one word of data.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt can be requested.

Table 8-5 lists the register information in normal mode and figure 9-6 shows memory mapping in normal mode.

Table 8-5 Register Information in Normal Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register A	CRA	Designates transfer count
DTC transfer count register B	CRB	Not used

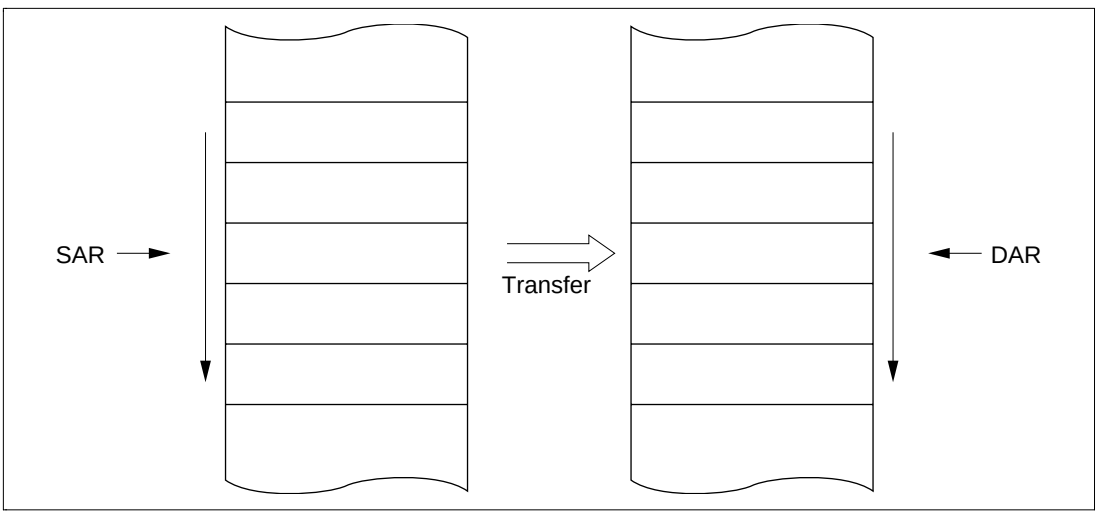


Figure 8-6 Memory Mapping in Normal Mode

8.3.6 Repeat Mode

In repeat mode, one operation transfers one byte or one word of data.

From 1 to 256 transfers can be specified. Once the specified number of transfers have ended, the initial state of the transfer counter and the address register specified as the repeat area is restored, and transfer is repeated. In repeat mode the transfer counter value does not reach H'00, and therefore CPU interrupts cannot be requested when DISEL = 0.

Table 8-6 lists the register information in repeat mode and figure 9-7 shows memory mapping in repeat mode.

Table 8-6 Register Information in Repeat Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds number of transfers
DTC transfer count register AL	CRAL	Designates transfer count
DTC transfer count register B	CRB	Not used

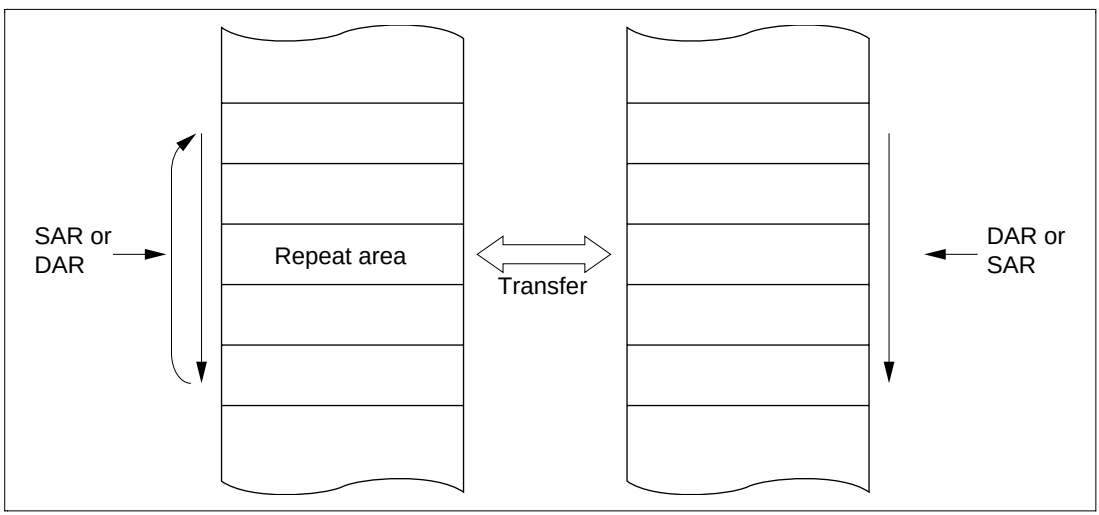


Figure 8-7 Memory Mapping in Repeat Mode

8.3.7 Block Transfer Mode

In block transfer mode, one operation transfers one block of data. Either the transfer source or the transfer destination is designated as a block area.

The block size is 1 to 256. When the transfer of one block ends, the initial state of the block size counter and the address register specified as the block area is restored. The other address register is then incremented, decremented, or left fixed.

From 1 to 65,536 transfers can be specified. Once the specified number of transfers have ended, a CPU interrupt is requested.

Table 8-7 lists the register information in block transfer mode and figure 8-8 shows memory mapping in block transfer mode.

Table 8-7 Register Information in Block Transfer Mode

Name	Abbreviation	Function
DTC source address register	SAR	Designates source address
DTC destination address register	DAR	Designates destination address
DTC transfer count register AH	CRAH	Holds block size
DTC transfer count register AL	CRAL	Designates block size count
DTC transfer count register B	CRB	Transfer count

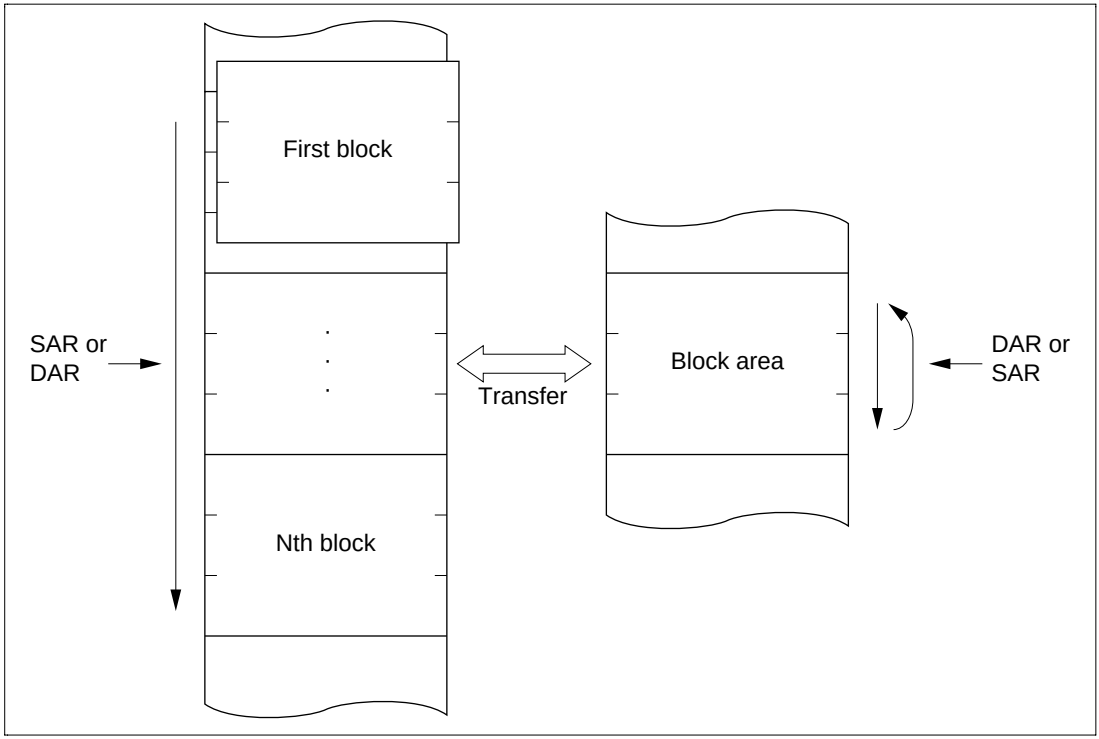


Figure 8-8 Memory Mapping in Block Transfer Mode

8.3.8 Chain Transfer

Setting the CHNE bit to 1 enables a number of data transfers to be performed consectutively in response to a single transfer request. SAR, DAR, CRA, CRB, MRA, and MRB, which define data transfers, can be set independently.

Figure 8-9 shows the memory map for chain transfer.

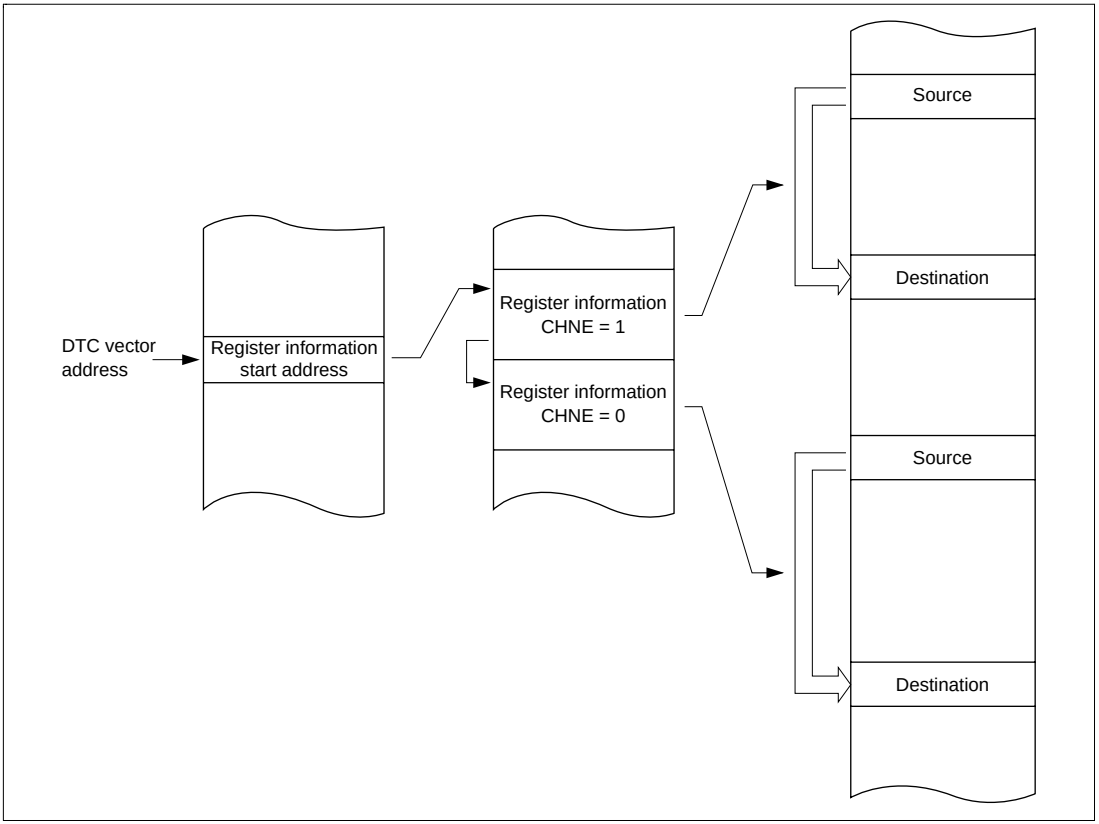


Figure 8-9 Chain Transfer Memory Map

In the case of transfer with CHNE set to 1, an interrupt request to the CPU is not generated at the end of the specified number of transfers or by setting of the DISEL bit to 1, and the interrupt source flag for the activation source is not affected.

8.3.9 Operation Timing

Figures 8-10 to 8-12 show an example of DTC operation timing.

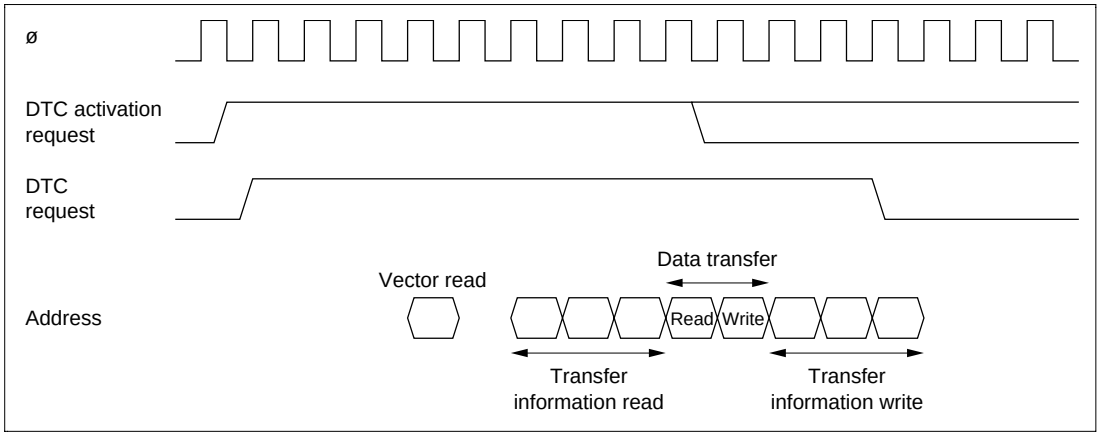


Figure 8-10 DTC Operation Timing (Example in Normal Mode or Repeat Mode)

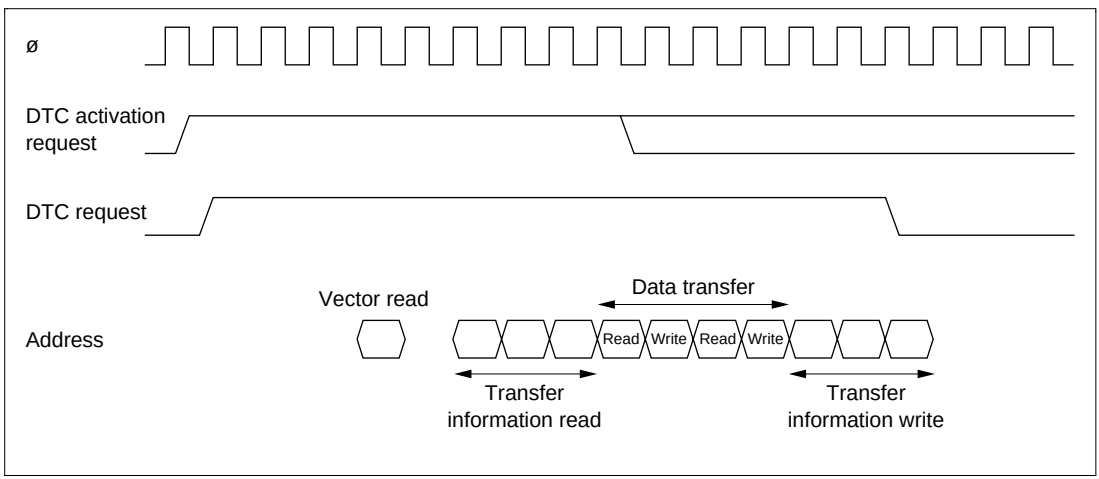


Figure 8-11 DTC Operation Timing (Example of Block Transfer Mode, with Block Size of 2)

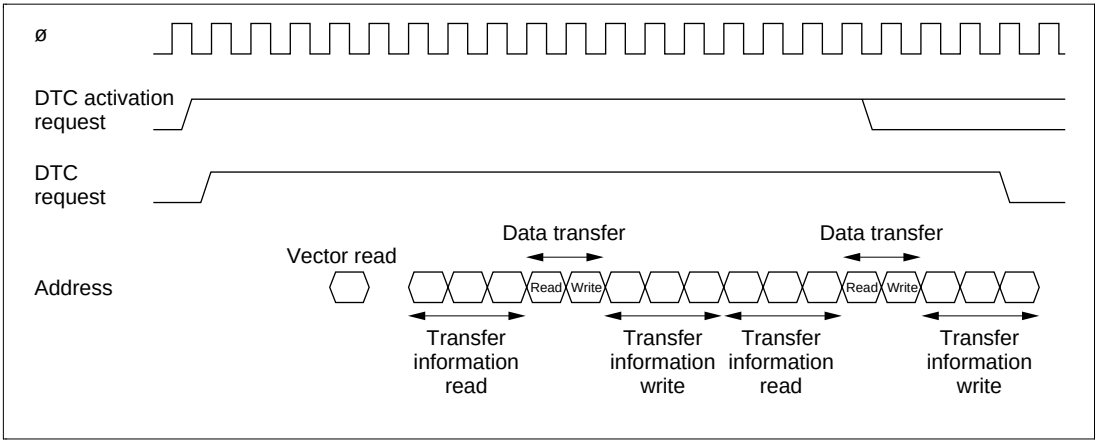


Figure 8-12 DTC Operation Timing (Example of Chain Transfer)

8.3.10 Number of DTC Execution States

Table 8-8 lists execution statuses for a single DTC data transfer, and table 8-9 shows the number of states required for each execution status.

Table 8-8 DTC Execution Statuses

Mode	Vector Read I	Register Information Read/Write J	Data Read K	Data Write L	Internal Operations M
Normal	1	6	1	1	3
Repeat	1	6	1	1	3
Block transfer	1	6	N	N	3

N: Block size (initial setting of CRAH and CRAL)

Table 8-9 Number of States Required for Each Execution Status

Object to be Accessed			On-Chip RAM	On-Chip ROM	On-Chip I/O Registers		External Devices			
Bus width			32	16	8	16	8	8	16	16
Access states			1	1	2	2	2	3	2	3
Execution status	Vector read	S_I	—	1	—	—	4	6+2m	2	3+m
	Register information read/write	S_J	1	—	—	—	—	—	—	—
	Byte data read	S_K	1	1	2	2	2	3+m	2	3+m
	Word data read	S_K	1	1	4	2	4	6+2m	2	3+m
	Byte data write	S_L	1	1	2	2	2	3+m	2	3+m
	Word data write	S_L	1	1	4	2	4	6+2m	2	3+m
	Internal operation	S_M	1	1	1	1	1	1	1	1

The number of execution states is calculated from the formula below. Note that Σ means the sum of all transfers activated by one activation event (the number in which the CHNE bit is set to 1, plus 1).

Number of execution states = $I \cdot S_I + \Sigma (J \cdot S_J + K \cdot S_K + L \cdot S_L) + M \cdot S_M$

For example, when the DTC vector address table is located in on-chip ROM, normal mode is set, and data is transferred from the on-chip ROM to an internal I/O register, the time required for the DTC operation is 13 states. The time from activation to the end of the data write is 10 states.

8.3.11 Procedures for Using DTC

Activation by Interrupt: The procedure for using the DTC with interrupt activation is as follows:

[1] Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.

[2] Set the start address of the register information in the DTC vector address.

[3] Set the corresponding bit in DTCER to 1.

[4] Set the enable bits for the interrupt sources to be used as the activation sources to 1. The DTC is activated when an interrupt used as an activation source is generated.

[5] After the end of one data transfer, or after the specified number of data transfers have ended, the DTCE bit is cleared to 0 and a CPU interrupt is requested. If the DTC is to continue transferring data, set the DTCE bit to 1.

Activation by Software: The procedure for using the DTC with software activation is as follows:

[1] Set the MRA, MRB, SAR, DAR, CRA, and CRB register information in the on-chip RAM.

[2] Set the start address of the register information in the DTC vector address.

[3] Check that the SWDTE bit is 0.

[4] Write 1 to SWDTE bit and the vector number to DTVECR.

[5] Check the vector number written to DTVECR.

[6] After the end of one data transfer, if the DISEL bit is 0 and a CPU interrupt is not requested, the SWDTE bit is cleared to 0. If the DTC is to continue transferring data, set the SWDTE bit to 1. When the DISEL bit is 1, or after the specified number of data transfers have ended, the SWDTE bit is held at 1 and a CPU interrupt is requested.

8.3.12 Examples of Use of the DTC

(1) Normal Mode

An example is shown in which the DTC is used to receive 128 bytes of data via the SCI.

[1] Set MRA to fixed source address (SM1 = SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), normal mode (MD1 = MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one data transfer by one interrupt (CHNE = 0, DISEL = 0). Set the SCI RDR address in SAR, the start address of the RAM area where the data will be received in DAR, and 128 (H'0080) in CRA. CRB can be set to any value.

[2] Set the start address of the register information at the DTC vector address.

[3] Set the corresponding bit in DTCER to 1.

[4] Set the SCI to the appropriate receive mode. Set the RIE bit in SCR to 1 to enable the reception complete (RXI) interrupt. Since the generation of a receive error during the SCI reception operation will disable subsequent reception, the CPU should be enabled to accept receive error interrupts.

[5] Each time reception of one byte of data ends on the SCI, the RDRF flag in SSR is set to 1, an RXI interrupt is generated, and the DTC is activated. The receive data is transferred from RDR to RAM by the DTC. DAR is incremented and CRA is decremented. The RDRF flag is automatically cleared to 0.

[6] When CRA becomes 0 after the 128 data transfers have ended, the RDRF flag is held at 1, the DTCE bit is cleared to 0, and an RXI interrupt request is sent to the CPU. The interrupt handling routine should perform wrap-up processing.

(2) Chain Transfer

An example of DTC chain transfer is shown in which pulse output is performed using the PPG. Chain transfer can be used to perform pulse output data transfer and PPG output trigger cycle updating. Repeat mode transfer to the PPG's NDR is performed in the first half of the chain transfer, and normal mode transfer to the TPU's TGR in the second half. This is because clearing of the activation source and interrupt generation at the end of the specified number of transfers are restricted to the second half of the chain transfer (transfer when CHNE = 0).

[1] Perform settings for transfer to the PPG's NDR. Set MRA to source address incrementing (SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), repeat mode (MD1 = 0, MD0 = 1), and word size (Sz = 1). Set the source side as a repeat area (DTS = 1). Set MRB to chain mode (CHNE = 1, DISEL = 0). Set the data table start address in SAR, the NDRH address in DAR, and the data table size in CRAH and CRAL. CRB can be set to any value.

[2] Perform settings for transfer to the TPU's TGR. Set MRA to source address incrementing (SM1 = 1, SM0 = 0), fixed destination address (DM1 = DM0 = 0), normal mode (MD1 = MD0 = 0), and word size (Sz = 1). Set the data table start address in SAR, the TGRA address in DAR, and the data table size in CRA. CRB can be set to any value.

[3] Locate the TPU transfer register information consecutively after the NDR transfer register information.

[4] Set the start address of the NDR transfer register information to the DTC vector address.

[5] Set the bit corresponding to TGIA in DTCER to 1.

[6] Set TGRA as an output compare register (output disabled) with TIOR, and enable the TGIA interrupt with TIER.

[7] Set the initial output value in PODR, and the next output value in NDR. Set bits in DDR and NDER for which output is to be performed to 1. Using PCR, select the TPU compare match to be used as the output trigger.

[8] Set the CST bit in TSTR to 1, and start the TCNT count operation.

[9] Each time a TGRA compare match occurs, the next output value is transferred to NDR and the set value of the next output trigger period is transferred to TGRA. The activation source TGFA flag is cleared.

[10] When the specified number of transfers are completed (the TPU transfer CRA value is 0), the TGFA flag is held at 1, the DTCE bit is cleared to 0, and a TGIA interrupt request is sent to the CPU. Termination processing should be performed in the interrupt handling routine.

(3) Software Activation

An example is shown in which the DTC is used to transfer a block of 128 bytes of data by means of software activation. The transfer source address is H'1000 and the destination address is H'2000. The vector number is H'60, so the vector address is H'04C0.

[1] Set MRA to incrementing source address (SM1 = 1, SM0 = 0), incrementing destination address (DM1 = 1, DM0 = 0), block transfer mode (MD1 = 1, MD0 = 0), and byte size (Sz = 0). The DTS bit can have any value. Set MRB for one block transfer by one interrupt (CHNE = 0). Set the transfer source address (H'1000) in SAR, the destination address (H'2000) in DAR, and 128 (H'8080) in CRA. Set 1 (H'0001) in CRB.

[2] Set the start address of the register information at the DTC vector address (H'04C0).

[3] Check that the SWDTE bit in DTVECR is 0. Check that there is currently no transfer activated by software.

[4] Write 1 to the SWDTE bit and the vector number (H'60) to DTVECR. The write data is H'E0.

[5] Read DTVECR again and check that it is set to the vector number (H'60). If it is not, this indicates that the write failed. This is presumably because an interrupt occurred between steps 3 and 4 and led to a different software activation. To activate this transfer, go back to step 3.

[6] If the write was successful, the DTC is activated and a block of 128 bytes of data is transferred.

[7] After the transfer, an SWDTEND interrupt occurs. The interrupt handling routine should clear the SWDTE bit to 0 and perform other wrap-up processing.

8.4 Interrupts

An interrupt request is issued to the CPU when the DTC finishes the specified number of data transfers, or a data transfer for which the DISEL bit was set to 1. In the case of interrupt activation, the interrupt set as the activation source is generated. These interrupts to the CPU are subject to CPU mask level and interrupt controller priority level control.

In the case of activation by software, a software activated data transfer end interrupt (SWDTEND) is generated.

When the DISEL bit is 1 and one data transfer has ended, or the specified number of transfers have ended, after data transfer ends, the SWDTE bit is held at 1 and an SWDTEND interrupt is generated. The interrupt handling routine should clear the SWDTE bit to 0.

When the DTC is activated by software, an SWDTEND interrupt is not generated during a data transfer wait or during data transfer even if the SWDTE bit is set to 1.

8.5 Usage Notes

Module Stop: When the MSTPA6 bit in MSTPCRA is set to 1, the DTC clock stops, and the DTC enters the module stop state. However, 1 cannot be written in the MSTPA6 bit while the DTC is operating.

On-Chip RAM: The MRA, MRB, SAR, DAR, CRA, and CRB registers are all located in on-chip RAM. When the DTC is used, the RAME bit in SYSCR must not be cleared to 0.

DTCE Bit Setting: For DTCE bit setting, use bit manipulation instructions such as BSET and BCLR. If all interrupts are masked, multiple activation sources can be set at one time by writing data after executing a dummy read on the relevant register.

Section 9 I/O Ports

9.1 Overview

The chip has 10 I/O ports (ports 1, 3 and A to F, H, J), and two input-only port (ports 4 and 9).

Table 9-1 summarizes the port functions. The pins of each port also have other functions.

Each I/O port includes a data direction register (DDR) that controls input/output, a data register (DR) that stores output data, and a port register (PORT) used to read the pin states. The input-only ports do not have a DR or DDR register.

Ports A to E have a built-in pull-up MOS function, and in addition to DR and DDR, have a MOS input pull-up control register (PCR) to control the on/off state of MOS input pull-up.

Ports 3, and A to C include an open-drain control register (ODR) that controls the on/off state of the output buffer PMOS.

When ports 10 to 13 and A to F are used as the output pins for expanded bus control signals, they can drive one TTL load plus a 90pF capacitance load. Those ports in other cases and ports 14 to 17 and 3 can drive one TTL load and a 30pF capacitance load. All I/O ports can drive Darlington transistors when set to output.

Port 1 pins (P16 and P14) and port 3 pins (P35 and P32) are Schmitt-trigger inputs.

See appendix C, I/O Port Block Diagrams, for a block diagram of each port.

Table 9-1 Port Functions

Port	Description	Pins	Mode 4	Mode 5	Mode 6	Mode 7
Port 1	• 8-bit I/O port • Schmitt-triggered input (P16, P14)	P17/PO15/TIOCB2/TCLKD P16/PO14/TIOCA2/$\overline{\text{IRQ1}}$ P15/PO13/TIOCB1/TCLKC P14/PO12/TIOCA1/$\overline{\text{IRQ0}}$ P13/PO11/TIOCD0/TCLKB/A23 P12/PO10/TIOCC0/TCLKA/A22 P11/PO9/TIOCB0/A21 P10/PO8/TIOCA0/A20	8 bit I/O port also functioning as TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, TIOCB2), PPG output pins (PO15 to PO8), interrupt input pins ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$), and address outputs (A20 to A23)			8-bit I/O port also functioning as TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, TIOCB2), PPG output pins (PO15 to PO8), interrupt input pins ($\overline{\text{IRQ0}}$, $\overline{\text{IRQ1}}$)
Port 3	• 6-bit I/O port • Open-drain output capability • Schmitt-triggered input (P35, P32)	P35/SCK1/SCL0*/$\overline{\text{IRQ5}}$ P34/RxD1/SDA0* P33/TxD1/SCL1* P32/SCK0/SDA1*/$\overline{\text{IRQ4}}$ P31/RxD0 P30/TxD0	6-bit I/O port also functioning as SCI (channel 0, 1) I/O pins (TxD0, RxD0, SCK0, TxD1, RxD1, SCK1), interrupt input pins ($\overline{\text{IRQ4}}$, $\overline{\text{IRQ5}}$), IIC (channel 0, 1) I/O pins (SCL0, SDA0, SCL1, SDA1)*			
Port 4	• 8-bit input port	P47 /AN7/DA1 P46 /AN6/DA0 P45 /AN5 P44 /AN4 P43 /AN3 P42 /AN2 P41 /AN1 P40/AN0	8-bit input port also functioning as A/D converter analog inputs (AN7 to AN0) and D/A converter analog outputs (DA1, DA0)			

Port	Description	Pins	Mode 4	Mode 5	Mode 6	Mode 7
Port 9	• 4-bit input port	P93/AN11 P92/AN10 P91/AN9 P90/AN8	4-bit input port also functioning as A/D converter analog inputs (AN11 to AN8)			
Port A	• 4-bit I/O port • Built-in MOS input pull-up • Open-drain output capability	PA3/A19/SCK2 PA2/A18/RxD2 PA1/A17/TxD2 PA0/A16	4-bit I/O port also functioning as SCI (channel 2) I/O pins (TxD2, RxD2, SCK2) and address outputs (A19 to A16)			4-bit I/O port also functioning as SCI (channel 2) I/O pins (TxD2, RxD2, SCK2)
Port B	• 8-bit I/O port • Built-in MOS input pull-up • Open-drain output capability	PB7/A15/TIOCB5 PB6/A14/TIOCA5 PB5/A13/TIOCB4 PB4/A12/TIOCA4 PB3/A11/TIOCD3 PB2/A10/TIOCC3 PB1/A9/TIOCB3 PB0/A8/TIOCA3	8-bit I/O port also functioning as TPU I/O pins (TIOCB5, TIOCA5, TIOCB4, TIOCA4, TIOCD3, TIOCC3, TIOCB3, TIIOCA3) and address outputs (A15 to A8)			8-bit I/O port also functioning as TPU I/O pins (TIOCB5, TIOCA5, TIOCB4, TIOCA4, TIOCD3, TIOCC3, TIOCB3, TIIOCA3)
Port C	• 8-bit I/O port • Built-in MOS input pull-up • Open-drain output capability	PC7/A7 PC6/A6 PC5/A5 PC4/A4 PC3/A3 PC2/A2 PC1/A1 PC0/A0	8-bit I/O port also functioning as address outputs (A7 to A0)			I/O port
Port D	• 8-bit I/O port • Built-in MOS input pull-up	PD7/D15 PD6/D14 PD5/D13 PD4/D12 PD3/D11 PD2/D10 PD1/D9 PD0/D8	Data bus input/output			I/O port

Port	Description	Pins	Mode 4	Mode 5	Mode 6	Mode 7
Port E	• 8-bit I/O port • Built-in MOS input pull-up	PE7/D7 PE6/D6 PE5/D5 PE4/D4 PE3/D3 PE2/D2 PE1/D1 PE0/D0	In 8-bit-bus mode: I/O port In 16-bit-bus mode: data bus input/output			I/O port
Port F	• 6-bit I/O port	PF7 /ø	When DDR = 0: input port When DDR = 1 (after reset): ø output			When DDR = 0 (after reset): input port When DDR = 1: ø output
		PF6 /$\overline{AS}$ PF5 /$\overline{RD}$ PF4 /$\overline{HWR}$ PF3/$\overline{LWR}$/$\overline{ADTRG}$/$\overline{IRQ3}$	$\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$ outputs $\overline{ADTRG}$, $\overline{IRQ3}$ input			I/O port $\overline{ADTRG}$, $\overline{IRQ3}$ input
		PF0/$\overline{IRQ2}$	$\overline{IRQ2}$ input, I/O port			
Port H	• 8-bit I/O port	PH7/PWM1H PH6/PWM1G PH5/PWM1F PH4/PWM1E PH3/PWM1D PH2/PWM1C PH1/PWM1B PH0/PWM1A	Function as both Motor Control PWM Timer output pins and 8-bit I/O port.			
Port J	• 8-bit I/O port	PJ7/PWM2H PJ6/PWM2G PJ5/PWM2F PJ4/PWM2E PJ3/PWM2D PJ2/PWM2C PJ1/PWM2B PJ0/PWM2A	Function as both Motor Control PWM Timer output pins and 8-bit I/O port.			

Note: * Pins for I^2C bus interface.

I^2C bus interface is available as an option in the H8S/2638 and H8S/2639.

9.2 Port 1

9.2.1 Overview

Port 1 is an 8-bit I/O port. Port 1 pins also function as PPG output pins (PO15 to PO8), TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, and TIOCB2), external interrupt pins ($\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$), and address bus output pins (A23 to A20). Port 1 pin functions change according to the operating mode.

Figure 9-1 shows the port 1 pin configuration.

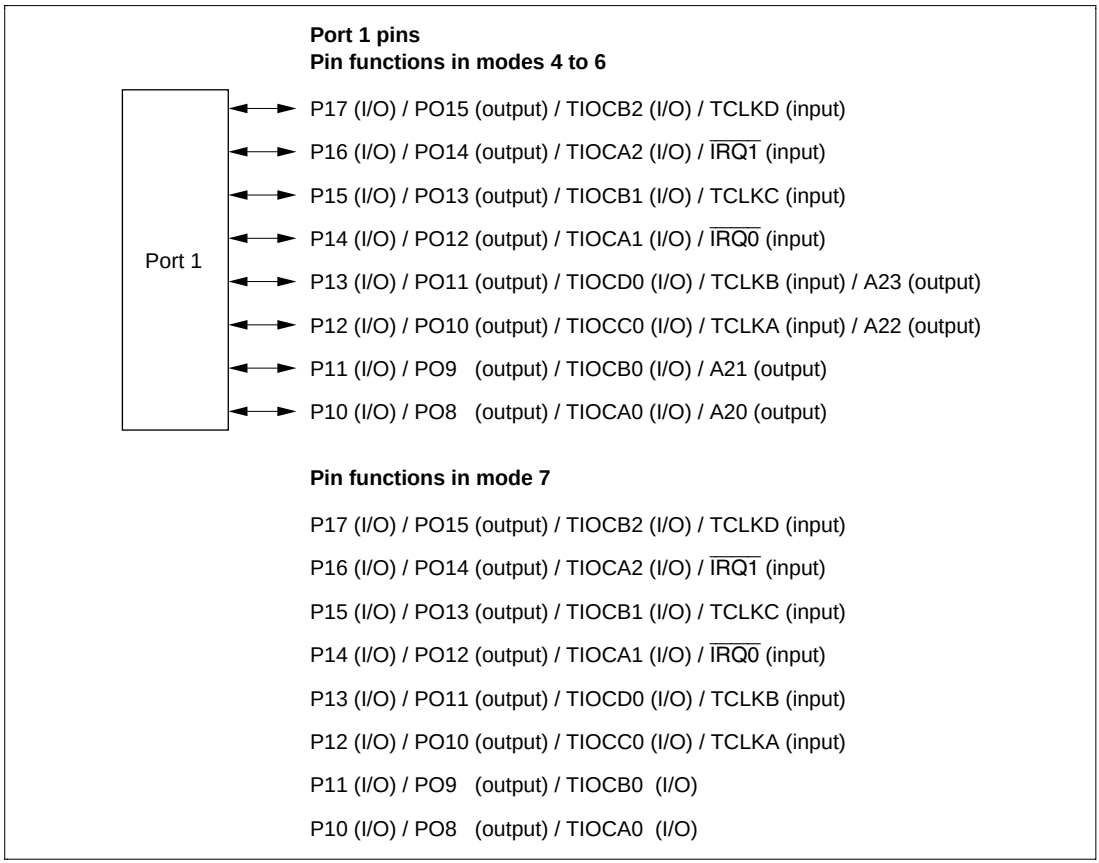


Figure 9-1 Port 1 Pin Functions

9.2.2 Register Configuration

Table 9-2 shows the port 1 register configuration.

Table 9-2 Port 1 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 1 data direction register	P1DDR	W	H'00	H'FE30
Port 1 data register	P1DR	R/W	H'00	H'FF00
Port 1 register	PORT1	R	Undefined	H'FFB0

Note: * Lower 16 bits of the address.

Port 1 Data Direction Register (P1DDR)

Bit	:	7	6	5	4	3	2	1	0
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 1. P1DDR cannot be read; if it is, an undefined value will be read.

Setting a P1DDR bit to 1 makes the corresponding port 1 pin an output pin, while clearing the bit to 0 makes the pin an input pin.

P1DDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port 1 Data Register (P1DR)

Bit	:	7	6	5	4	3	2	1	0
		P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P1DR is an 8-bit readable/writable register that stores output data for the port 1 pins (P17 to P10).

P1DR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port 1 Register (PORT1)

Bit	:	7	6	5	4	3	2	1	0
		P17	P16	P15	P14	P13	P12	P11	P10
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins P17 to P10.

PORT1 is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port 1 pins (P17 to P10) must always be performed on P1DR.

If a port 1 read is performed while P1DDR bits are set to 1, the P1DR values are read. If a port 1 read is performed while P1DDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORT1 contents are determined by the pin states, as P1DDR and P1DR are initialized. PORT1 retains its prior state in software standby mode.

9.2.3 Pin Functions

Port 1 pins also function as PPG output pins (PO15 to PO8), TPU I/O pins (TCLKA, TCLKB, TCLKC, TCLKD, TIOCA0, TIOCB0, TIOCC0, TIOCD0, TIOCA1, TIOCB1, TIOCA2, and TIOCB2), external interrupt input pins ($\overline{\text{IRQ0}}$ and $\overline{\text{IRQ1}}$), and address bus output pins (A23 to A20). Port 1 pin functions are shown in table 9-3.

Table 9-3 Port 1 Pin Functions

Pin | **Selection Method and Pin Functions**

P17/PO15/ TIOCB2/ TCLKD

The pin function is switched as shown below according to the combination of the TPU channel 2 setting (by bits MD3 to MD0 in TMDR2, bits IOB3 to IOB0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bits TPSC2 to TPSC0 in TCR0 and TCR5, bit NDER15 in NDERH, and bit P17DDR.

TPU Channel 2 Setting	Table Below (1)	Table Below (2)		
P17DDR	—	0	1	1
NDER15	—	—	0	1
Pin function	TIOCB2 output	P17 input	P17 output	PO15 output
		TIOCB2 input *1		
	TCLKD input *2			

Notes: *1 TIOCB2 input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 = 1.

*2 TCLKD input when the setting for either TCR0 or TCR5 is: TPSC2 to TPSC0 = B'111.
TCLKD input when channels 2 and 4 are set to phase counting mode.

TPU Channel 2 Setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

Pin	Selection Method and Pin Functions
P16/PO14/ TIOCA2/ $\overline{\text{IRQ1}}$	The pin function is switched as shown below according to the combination of the TPU channel 2 setting (by bits MD3 to MD0 in TMDR2, bits IOA3 to IOA0 in TIOR2, and bits CCLR1 and CCLR0 in TCR2), bit NDER14 in NDERH, and bit P16DDR.

TPU Channel 2 Setting	Table Below (1)	Table Below (2)		
P16DDR	—	0	1	1
NDER14	—	—	0	1
Pin function	TIOCA2 output	P16 input	P16 output	PO14 output
		TIOCA2 input *1		
	$\overline{\text{IRQ1}}$ input			

TPU Channel 2 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output *2	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCA2 input when MD3 to MD0 = B'0000 or B'01xx, and IOA3 = 1.

*2 TIOCB2 output is disabled.

Pin	Selection Method and Pin Functions
P15/PO13/ TIOCB1/TCLKC	The pin function is switched as shown below according to the combination of the TPU channel 1 setting (by bits MD3 to MD0 in TMDR1, bits IOB3 to IOB0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bits TPSC2 to TPSC0 in TCR0, TCR2, TCR4, and TCR5, bit NDER13 in NDERH, and bit P15DDR.

TPU Channel 1 Setting	Table Below (1)	Table Below (2)		
P15DDR	—	0	1	1
NDER13	—	—	0	1
Pin function	TIOCB1 output	P15 input	P15 output	PO13 output
		TIOCB1 input *1		
	TCLKC input *2			

Notes: *1 TIOCB1 input when MD3 to MD0 = B'0000 or B'01xx, and IOB3 to IOB0 = B'10xx.

*2 TCLKC input when the setting for either TCR0 or TCR2 is: TPSC2 to TPSC0 = B'110; or when the setting for either TCR4 or TCR5 is TPSC2 to TPSC0 = B'101.
TCLKC input when channels 2 and 4 are set to phase counting mode.

TPU Channel 1 Setting	(2)	(1)	(2)	(2)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'0010	B'0011		
IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
CCLR1, CCLR0	—	—	—	—	Other than B'10	B'10
Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don't care

Pin	Selection Method and Pin Functions
P14/PO12/ TIOCA1/$\overline{\text{IRQ0}}$	The pin function is switched as shown below according to the combination of the TPU channel 1 setting (by bits MD3 to MD0 in TMDR1, bits IOA3 to IOA0 in TIOR1, and bits CCLR1 and CCLR0 in TCR1), bit NDER12 in NDERH, and bit P14DDR.

TPU Channel 1 Setting	Table Below (1)	Table Below (2)		
P14DDR	—	0	1	1
NDER12	—	—	0	1
Pin function	TIOCA1 output	P14 input	P14 output	PO12 output
		TIOCA1 input *1		
	$\overline{\text{IRQ0}}$ input			

TPU Channel 1 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000, B'01xx		B'001x	B'0010	B'0011	
IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR1, CCLR0	—	—	—	—	Other than B'01	B'01
Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCA1 input when MD3 to MD0 = B'0000 or B'01xx, and IOA3 to IOA0 = B'10xx.

*2 TIOCB1 output is disabled.

Pin	Selection Method and Pin Functions
P13/PO11/ TIOCD0/TCLKB/ A23	The pin function is switched as shown below according to the combination of the operating mode, and the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, bits IOD3 to IOD0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR2, bits AE3 to AE0 in PFCR, bit NDER11 in NDERH, and bit P13DDR.

<table>
<tr><td>Operating mode</td><td colspan="5">Modes 4 to 6</td></tr>
<tr><td>AE3 to AE0</td><td colspan="4">B'0000 to B'1110</td><td>B'1111</td></tr>
<tr><td>TPU Channel 0 Setting</td><td>Table Below (1)</td><td colspan="3">Table Below (2)</td><td>—</td></tr>
<tr><td>P13DDR</td><td>—</td><td>0</td><td>1</td><td>1</td><td>—</td></tr>
<tr><td>NDER11</td><td>—</td><td>—</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="2">TIOCD0 output</td><td>P13 input</td><td>P13 output</td><td>PO11 output</td><td rowspan="3">A23 output</td></tr>
<tr><td colspan="3">TIOCD0 input *1</td></tr>
<tr><td colspan="4">TCLKB input *2</td></tr>
</table>

<table>
<tr><td>Operating mode</td><td colspan="4">Mode 7</td></tr>
<tr><td>AE3 to AE0</td><td colspan="4">—</td></tr>
<tr><td>TPU Channel 0 Setting</td><td>Table Below (1)</td><td colspan="3">Table Below (2)</td></tr>
<tr><td>P13DDR</td><td>—</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>NDER11</td><td>—</td><td>—</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="2">TIOCD0 output</td><td>P13 input</td><td>P13 output</td><td>PO11 output</td></tr>
<tr><td colspan="3">TIOCD0 input *1</td></tr>
<tr><td colspan="4">TCLKB input *2</td></tr>
</table>

Notes: *1 TIOCD0 input when MD3 to MD0 = B'0000, and IOD3 to IOD0 = B'10xx.

*2 TCLKB input when the setting for TCR0 to TCR2 is: TPSC2 to TPSC0 = B'101.

TCLKB input when channels 1 and 5 are set to phase counting mode.

Pin	Selection Method and Pin Functions						
P13/PO11/ TIOCD0/TCLKB/ A23	TPU Channel 0 Setting	(2)	(1)	(2)	(2)	(1)	(2)
	MD3 to MD0	B'0000		B'0010	B'0011		
	IOD3 to IOD0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
	CCLR2 to CCLR0	—	—	—	—	Other than B'110	B'110
	Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don’t care

Pin	Selection Method and Pin Functions
P12/PO10/ TIOCC0/TCLKA/ A22	The pin function is switched as shown below according to the combination of the operating mode, and the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, bits IOC3 to IOC0 in TIOR0L, and bits CCLR2 to CCLR0 in TCR0), bits TPSC2 to TPSC0 in TCR0 to TCR5, bits AE3 to AE0 in PFCR, bit NDER10 in NDERH, and bit P12DDR.

<table>
<tr><td>Operating mode</td><td colspan="5">Modes 4 to 6</td></tr>
<tr><td>AE3 to AE0</td><td colspan="4">B'0000 to B'1110</td><td>B'1111</td></tr>
<tr><td>TPU Channel 0 Setting</td><td>Table Below (1)</td><td colspan="3">Table Below (2)</td><td>—</td></tr>
<tr><td>P12DDR</td><td>—</td><td>0</td><td>1</td><td>1</td><td>—</td></tr>
<tr><td>NDER10</td><td>—</td><td>—</td><td>0</td><td>1</td><td>—</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="2">TIOCC0 output</td><td>P12 input</td><td>P12 output</td><td>PO10 output</td><td rowspan="3">A22 output</td></tr>
<tr><td colspan="3">TIOCC0 input *1</td></tr>
<tr><td colspan="4">TCLKA input *2</td></tr>
</table>

<table>
<tr><td>Operating mode</td><td colspan="4">Mode 7</td></tr>
<tr><td>AE3 to AE0</td><td colspan="4">—</td></tr>
<tr><td>TPU Channel 0 Setting</td><td>Table Below (1)</td><td colspan="3">Table Below (2)</td></tr>
<tr><td>P12DDR</td><td>—</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>NDER10</td><td>—</td><td>—</td><td>0</td><td>1</td></tr>
<tr><td rowspan="3">Pin function</td><td rowspan="2">TIOCC0 output</td><td>P12 input</td><td>P12 output</td><td>PO10 output</td></tr>
<tr><td colspan="3">TIOCC0 input *1</td></tr>
<tr><td colspan="4">TCLKA input *2</td></tr>
</table>

Pin	Selection Method and Pin Functions

P12/PO10/ TIOCC0/TCLKA/ A22

TPU Channel 0 Setting	(2)	(1)	(2)	(1)	(1)	(2)
MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
IOC3 to IOC0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
CCLR2 to CCLR0	—	—	—	—	Other than B'101	B'101
Output function	—	Output compare output	—	PWM mode 1 output*3	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCC0 input when MD3 to MD0 = B'0000, and IOC3 to IOC0 = B'10xx.

*2 TCLKA input when the setting for TCR0 to TCR5 is: TPSC2 to TPSC0 = B'100.

TCLKA input when channels 1 and 5 are set to phase counting mode.

*3 TIOCD0 output is disabled.

When BFA = 1 or BFB = 1 in TMDR0, output is disabled and setting (2) applies.

Pin	Selection Method and Pin Functions
P11/PO9/TIOCB0/ A21	The pin function is switched as shown below according to the combination of the operating mode, and the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, and bits IOB3 to IOB0 in TIOR0H), bits AE3 to AE0 in PFCR, bit NDER9 in NDERH, and bit P11DDR.

Operating mode	Modes 4 to 6				
AE3 to AE0	B'0000 to B'1101				B'1110 to B'1111
TPU Channel 0 Setting	Table Below (1)	Table Below (2)			—
P11DDR	—	0	1	1	—
NDER9	—	—	0	1	—
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output	A21 output
		TIOCB0 input *			

Operating mode	Mode 7			
AE3 to AE0	—			
TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P11DDR	—	0	1	1
NDER9	—	—	0	1
Pin function	TIOCB0 output	P11 input	P11 output	PO9 output
		TIOCB0 input *		

Note: * TIOCB0 input when MD3 to MD0 = B'0000, and IOB3 to IOB0 = B'10xx.

Pin	Selection Method and Pin Functions						
P11/PO9/TIOCB0/ A21	TPU Channel 0 Setting	(2)	(1)	(2)	(2)	(1)	(2)
	MD3 to MD0	B'0000		B'0010	B'0011		
	IOB3 to IOB0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	—	B'xx00	Other than B'xx00	
	CCLR2 to CCLR0	—	—	—	—	Other than B'010	B'010
	Output function	—	Output compare output	—	—	PWM mode 2 output	—

x: Don’t care

Pin	Selection Method and Pin Functions
P10/PO8/TIOCA0/ A20	The pin function is switched as shown below according to the combination of the operating mode, and the TPU channel 0 setting (by bits MD3 to MD0 in TMDR0, bits IOA3 to IOA0 in TIOR0H, and bits CCLR2 to CCLR0 in TCR0), bits AE3 to AE0 in PFCR, bit NDER8 in NDERH, SAE0 bit in DMABCRH, and bit P10DDR.

Operating mode	Modes 4 to 6				
AE3 to AE0	B'0000 to B'1110				B'1101 to B'1111
TPU Channel 0 Setting	Table Below (1)	Table Below (2)			—
P10DDR	—	0	1	1	—
NDER8	—	—	0	1	—
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output	A20 output
		TIOCA0 input *1			

Operating mode	Mode 7			
AE3 to AE0	—			
TPU Channel 0 Setting	Table Below (1)	Table Below (2)		
P10DDR	—	0	1	1
NDER8	—	—	0	1
Pin function	TIOCA0 output	P10 input	P10 output	PO8 output
		TIOCA0 input *1		

Pin	Selection Method and Pin Functions						
P10/PO8/TIOCA0/ A20	TPU Channel 0 Setting	(2)	(1)	(2)	(1)	(1)	(2)
	MD3 to MD0	B'0000		B'001x	B'0010	B'0011	
	IOA3 to IOA0	B'0000 B'0100 B'1xxx	B'0001 to B'0011 B'0101 to B'0111	B'xx00	Other than B'xx00		
	CCLR2 to CCLR0	—	—	—	—	Other than B'001	B'001
	Output function	—	Output compare output	—	PWM mode 1 output*2	PWM mode 2 output	—

x: Don't care

Notes: *1 TIOCA0 input when MD3 to MD0 = B'0000, and IOA3 to IOA0 = B'10xx.

*2 TIOCB0 output is disabled.

9.3 Port 3

9.3.1 Overview

Port 3 is an 6-bit I/O port. Port 3 is a multi-purpose port for SCI I/O pins (TxD0, RxD0, SCK0, TxD1, RxD1, SCK1), external interrupt input pins ($\overline{IRQ4}$, $\overline{IRQ5}$), and IIC I/O pins* (SCL0, SDA0, SCL1, SDA1). All of the port 3 pin functions have the same operating mode. The configuration for each of the port 3 pins is shown in figure. 9-2.

Note: * Available when using I^2C bus interface as an option in the H8S/2638 and H8S/2639.

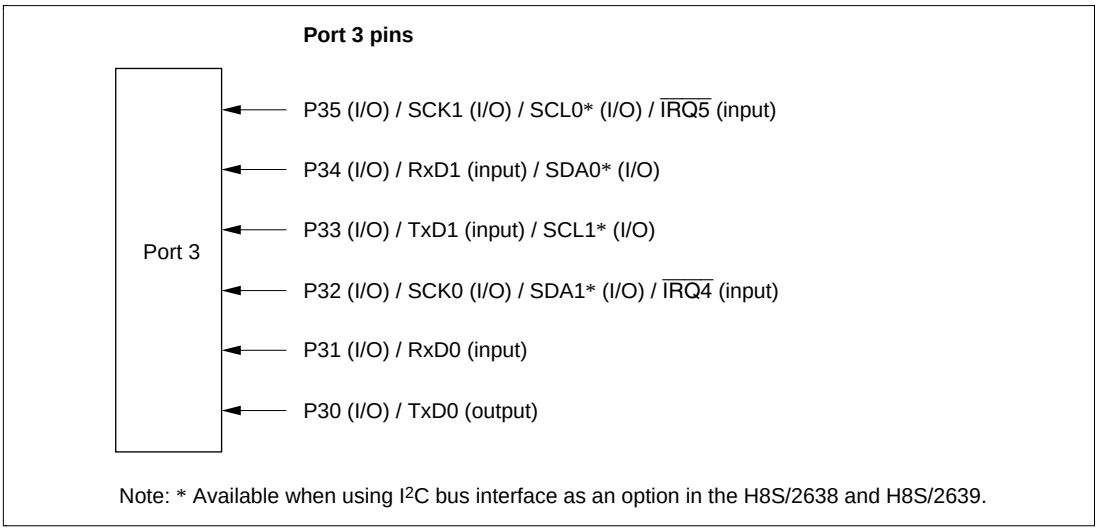


Figure 9-2 Port 3 Pin Functions

9.3.2 Register Configuration

Table 9-4 shows the configuration of port 3 registers.

Table 9-4 Port 3 Register Configuration

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port 3 data direction register	P3DDR	W	B'**000000	H'FE32
Port 3 data register	P3DR	R/W	B'**000000	H'FF02
Port 3 register	PORT3	R	Undefined	H'FFB2
Port 3 open drain control register	P3ODR	R/W	B'**000000	H'FE46

Notes: *1 Indicates the lower-place 16 bits.
*2 Indicates the values for bits 0 to 5.

Port 3 Data Direction Register (P3DDR)

Bit	7	6	5	4	3	2	1	0
	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	Undefined	Undefined	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

P3DDR is an 8-bit write-dedicated register, which specifies the I/O for each port 3 pin by bit. Read is disenabled. If a read is carried out, undefined values are read out.

By setting P3DDR to 1, the corresponding port 3 pins become output, and be clearing to 0 they become input.

P3DDR is initialized to B'**000000 by a reset and in hardware standby mode. The previous state is maintained in software standby mode. The pin state is determined by specifying SCI, IIC, P3DDR, and P3DR.

Port 3 Data Register (P3DR)

Bit	7	6	5	4	3	2	1	0
	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
Initial value	Undefined	Undefined	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

P3DR is an 8-bit readable/writable register, which stores the output data of port 3 pins (P35 to P30).

P3DR is initialized to B'**000000 by a reset and in hardware standby mode. The previous state is maintained in software standby mode.

Port 3 Register (PORT3)

Bit	7	6	5	4	3	2	1	0
	—	—	P35	P34	P33	P32	P31	P30
Initial value	Undefined	Undefined	—*	—*	—*	—*	—*	—*
Read/Write	—	—	R	R	R	R	R	R

Note: * Determined by the state of pins P35 to P30.

PORT3 is an 8-bit read-dedicated register, which reflects the state of pins. Write is disenabled. Always carry out writing off output data of port 3 pins (P35 to P30) to P3DR without fail.

When P3DDR is set to 1, if port 3 is read, the values of P3DR are read. When P3DDR is cleared to 0, if port 3 is read, the states of pins are read out.

P3DDR and P3DR are initialized by a reset and in hardware standby mode, so PORT3 is determined by the state of the pins. The previous state is maintained in software standby mode.

Port 3 Open Drain Control Register (P3ODR)

Bit	7	6	5	4	3	2	1	0
	—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR
Initial value	Undefined	Undefined	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

P3ODR is an 8-bit readable/writable register, which controls the on/off of port 3 pins (P35 to P30).

By setting P3ODR to 1, the port 3 pins become an open drain out, and when cleared to 0 they become CMOS output.

P3ODR is initialized to B'**000000 by a reset and in hardware standby mode. The previous state is maintained in software standby mode.

9.3.3 Pin Functions

The port 3 pins double as SCI I/O input pins (TxD0, RxD0, SCK0, TxD1, RxD1, and SCK1) external interrupt input pins ($\overline{IRQ4}$ and $\overline{IRQ5}$), and IIC I/O pins (SCL0, SDA0, SCL1, and SDA1). The functions of port 3 pins are shown in Table 9-5.

Table 9-5 Port 3 Pin Functions

Pin | **Selection Method and Pin Functions**

P35/SCK1/SCL0*[1]/$\overline{IRQ5}$

Switches as follows according to combinations of ICCR0 ICE bit*[1] of IIC0, bit C/$\overline{A}$ of SMR1, bits CKE0 and CKE1 of SCR1, and bit P35DDR.

When used as a SCL0 I/O pin, always be sure to clear the following bits to 0: bit C/$\overline{A}$ of SMR1, and bits CKE0 and CKE1 of SCR1. The SCL0 output format is NMOS open drain output, enabling direct bus driving.

ICE*[1]	0					1
CKE1	0				1	0
C/$\overline{A}$	0			1	—	0
CKE0	0		1	—	—	0
P35DDR	0	1	—	—	—	—
Pin function	P35 input	P35 output*	SCK1 output*	SCK1 output*	SCK1 input	SCL0 I/O
	$\overline{IRQ5}$ input					

Note: * When P35ODR = 1, it becomes NMOS open drain output*[2].

P34/RxD1/SDA0*[1]

Switches as follows according to combinations of ICCR0 ICE bit*[1] of IIC0, bit RE of SCR1 and bit P34DDR. The SDA0 output format is NMOS open drain output, enabling direct bus driving.

ICE*[1]	0			1
RE	0		1	—
P34DDR	0	1	—	—
Pin function	P34 input	P34 output*	RxD1 input	SDA0 I/O

Note: * When P34ODR = 1, it becomes NMOS open drain output*[2].

P33/TxD1/SCL1*[1]

Switches as follows according to combinations of ICCR1 ICE bit*[1] of IIC1, bit TE of SCR1 and bit P33DDR. The SCL1 output format is NMOS open drain output, enabling direct bus driving.

ICE*[1]	0			1
TE	0		1	—
P33DDR	0	1	—	—
Pin function	P33 input	P33 output*	TxD1 output*	SCL1 I/O

Note: * When P33ODR = 1, it becomes NMOS open drain output.

Pin	Selection Method and Pin Functions
P32/SCK0/ SDA1*[1]/$\overline{\text{IRQ4}}$	Switches as follows according to combinations of ICCR1 ICE bit*[1] of IIC1, bit C/$\overline{\text{A}}$ of SMR0, bits CKE0 and CKE1 of SCR0, and bit P32DDR. When used as a SDA1 I/O pin, always be sure to clear the following bits to 0: SMR0 C/$\overline{\text{A}}$ bit, SCR0 CKE0 and CKE1 bits. The SDA1 output format is NMOS open drain output, enabling direct bus driving.

ICE*[1]	0					1
CKE1	0				1	0
C/$\overline{\text{A}}$	0			1	—	0
CKE0	0		1	—	—	0
P32DDR	0	1	—	—	—	—
Pin function	P32 input	P32 output	SCK0 output*	SCK0 output*	SCK0 input	SDA1 I/O
	$\overline{\text{IRQ4}}$ input					

Note: * When P32ODR = 1, it becomes NMOS open drain output.

P31/RxD0/ IrRxD — Switches as follows according to combinations of bit RE of SCR0 and bit P31DDR.

RE	0		1
P31DDR	0	1	—
Pin function	P31 input	P31 output*	RxD0 input

Note: * When P31ODR = 1, it becomes NMOS open drain output.

P30/TxD0/ IrTxD — Switches as follows according to combinations of bit TE of SCR0 and bit P30DDR.

TE	0		1
P30DDR	0	1	—
Pin function	P30 input	P30 output*	TxD0 output*

Note: * When P30ODR = 1, it becomes NMOS open drain output.

Notes: *1 Available when using I^2C bus interface as an option (available in the H8S/2638 and H8S/2639 only).

*2 When using the I^2C bus interface in the H8S/2638 and H8S/2639, the output format for pins P35 and P34 is NMOS push-pull output. However, when P34 ODR (P35 ODR) is equal to 1, the output format is NMOS open drain output.

9.4 Port 4

9.4.1 Overview

Port 4 is an 8-bit input-only port. Port 4 pins also function as A/D converter analog input pins (AN0 to AN7) and D/A converter analog output pins (DA0, DA1). Port 4 pin functions are the same in all operating modes. Figure 9-3 shows the port 4 pin configuration.

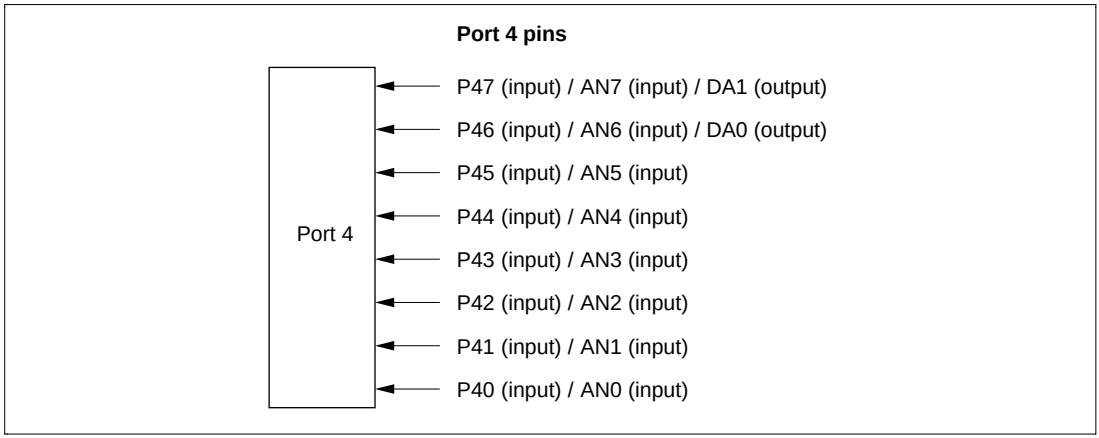


Figure 9-3 Port 4 Pin Functions

9.4.2 Register Configuration

Table 9-6 shows the port 4 register configuration. Port 4 is an input-only port, and does not have a data direction register or data register.

Table 9-6 Port 4 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 4 register	PORT4	R	Undefined	H'FFB3

Note: * Lower 16 bits of the address.

Port 4 Register (PORT4): The pin states are always read when a port 4 read is performed.

Bit	:	7	6	5	4	3	2	1	0
		P47	P46	P45	P44	P43	P42	P41	P40
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins P47 to P40.

9.4.3 Pin Functions

Port 4 pins also function as A/D converter analog input pins (AN0 to AN7) and D/A converter analog output pins (DA0 and DA1).

9.5 Port 9

9.5.1 Overview

Port 9 is a 4-bit input-only port. Port 9 pins also function as A/D converter analog input pins (AN8 to AN11). Port 9 pin functions are the same in all operating modes. Figure 9-4 shows the port 9 pin configuration.

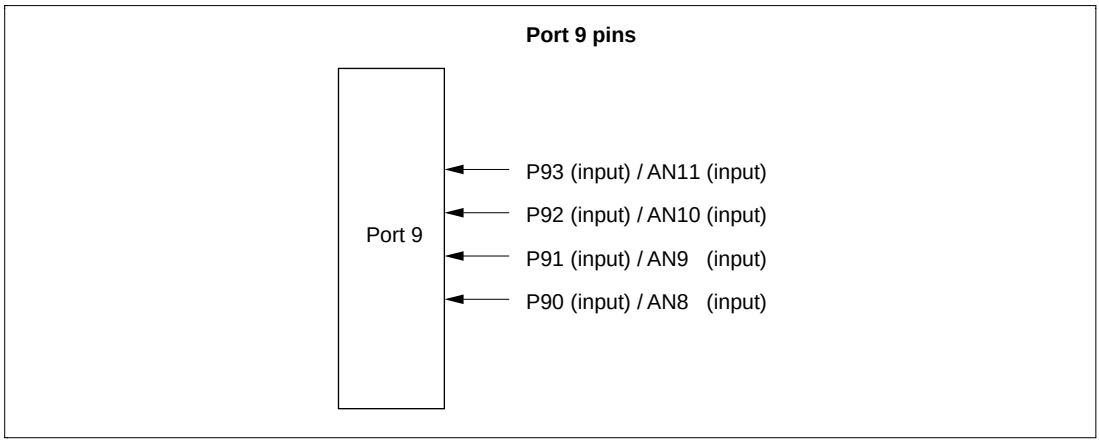


Figure 9-4 Port 9 Pin Functions

9.5.2 Register Configuration

Table 9-7 shows the port 9 register configuration. Port 9 is an input-only port, and does not have a data direction register or data register.

Table 9-7 Port 9 Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port 9 register	PORT9	R	Undefined	H'FFB8

Note: * Lower 16 bits of the address.

Port 9 Register (PORT9): The pin states are always read when a port 9 read is performed.

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	P93	P92	P91	P90
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	—	—	—	—	R	R	R	R

Note: * Determined by state of pins P93 to P90.

9.5.3 Pin Functions

Port 9 pins also function as A/D converter analog input pins (AN8 to AN11) are multipurpose pins which function as A/D converter analog input pins (AN8 to AN11).

9.6 Port A

9.6.1 Overview

Port A is a 4-bit I/O port. Port A pins also function as address bus outputs and SCI2 I/O pins (SCK2, RxD2, and TxD2). The pin functions change according to the operating mode.

Port A has a built-in MOS input pull-up function that can be controlled by software.

Figure 9-5 shows the port A pin configuration.

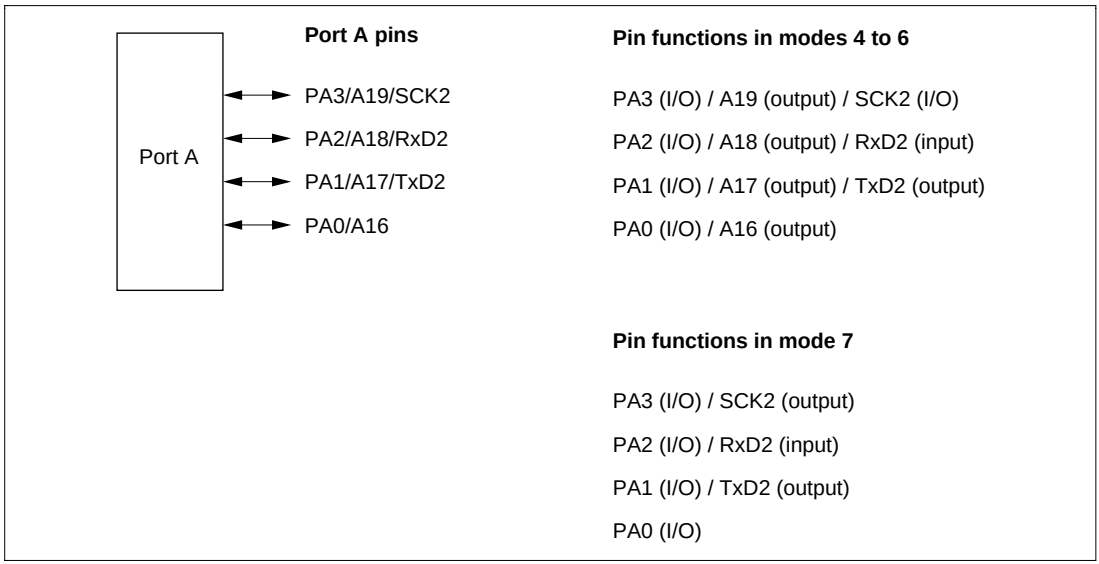


Figure 9-5 Port A Pin Functions

9.6.2 Register Configuration

Table 9-8 shows the port A register configuration.

Table 9-8 Port A Registers

Name	Abbreviation	R/W	Initial Value*2	Address*1
Port A data direction register	PADDR	W	H'0	H'FE39
Port A data register	PADR	R/W	H'0	H'FF09
Port A register	PORTA	R	Undefined	H'FFB9
Port A MOS pull-up control register	PAPCR	R/W	H'0	H'FF40
Port A open-drain control register	PAODR	R/W	H'0	H'FF47

Notes: *1 Lower 16 bits of the address.
*2 Value of bits 3 to 0.

Port A Data Direction Register (PADDR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	W	W	W	W

PADDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port A. PADDR cannot be read; if it is, an undefined value will be read.

Bits 7 to 4 are reserved; they return an undetermined value if read.

PADDR is initialized to H'0 (bits 3 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

- Modes 4 to 6

  The corresponding port A pins become address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, irrespective of the value of bits PA3DDR to PA0DDR. When pins are not used as address outputs, setting a PADDR bit to 1 makes the corresponding port A pin an output port, while clearing the bit to 0 makes the pin an input port.

- Mode 7

  Setting a PADDR bit to 1 makes the corresponding port A pin an output port, while clearing the bit to 0 makes the pin an input port.

Port A Data Register (PADR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

PADR is an 8-bit readable/writable register that stores output data for the port A pins (PA3 to PA0).

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modified.

PADR is initialized to H'0 (bits 3 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port A Register (PORTA)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3	PA2	PA1	PA0
Initial value	:	Undefined	Undefined	Undefined	Undefined	—*	—*	—*	—*
R/W	:	—	—	—	—	R	R	R	R

Note: * Determined by state of pins PA3 to PA0.

PORTA is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port A pins (PA3 to PA0) must always be performed on PADR.

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modified.

If a port A read is performed while PADDR bits are set to 1, the PADR values are read. If a port A read is performed while PADDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTA contents are determined by the pin states, as PADDR and PADR are initialized. PORTA retains its prior state in software standby mode.

Port A MOS Pull-Up Control Register (PAPCR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

PAPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port A on an individual bit basis.

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modified. In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR, in the SCI's SCMR, SMR, and SCR, and in DDR, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.

In mode 7, if a pin is in the input state in accordance with the settings in the SCI's SCMR, SMR, and SCR, and in DDR, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.

PAPCR is initialized by a reset or to H'0 (bits 3 to 0), and in hardware standby mode. It retains its prior state in software standby mode.

Port A Open Drain Control Register (PAODR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR
Initial value	:	Undefined	Undefined	Undefined	Undefined	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

PAODR is an 8-bit readable/writable register that controls whether PMOS is on or off for each port A pin (PA3 to PA0).

Bits 7 to 4 are reserved; they return an undetermined value if read, and cannot be modified.

When pins are not address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, setting a PAODR bit makes the corresponding port A pin an NMOS open-drain output, while clearing the bit to 0 makes the pin a CMOS output.

PAODR is initialized to H'0 (bits 3 to 0) by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

9.6.3 Pin Functions

Port A pins also function as SCI input/output pins (TxD2, RxD2, SCK2) and address bus output pins (A19 to A16). Port A pin functions are shown in table 9-9.

Table 9-9 Port A Pin Functions

Pin	Selection Method and Pin Functions
PA3/A19/SCK2	The pin function is switched as shown below according to the operating mode, bits AE3 to AE0 in PFCR, bit C/$\overline{A}$ in SMR and bits CKE0 and CKE1 in SCR of SCI2, and bit PA3DDR.

Operating mode	Modes 4 to 6					
AE3 to AE0	B'0000 to B'1011					B'1100 to B'1111
CKE1	0				1	—
C/$\overline{A}$	0			1	—	—
CKE0	0		1	—	—	—
PA3DDR	0	1	—	—	—	—
Pin function	PA3 input	PA3 output	SCK2 output	SCK2 output	SCK2 input	A19 output

Operating mode	Mode 7				
CKE1	0				1
C/$\overline{A}$	0			1	—
CKE0	0		1	—	—
PA3DDR	0	1	—	—	—
Pin function	PA3 input	PA3 output	SCK2 output	SCK2 output	SCK2 input

Pin	Selection Method and Pin Functions

PA2/A18/RxD2

The pin function is switched as shown below according to the operating mode, bits AE3 to AE0 in PFCR, bit RE in SCR of SCI2, and bit PA2DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000 to B'1011			B'1011 to B'1111
RE	0		1	—
PA2DDR	0	1	—	—
Pin function	PA2 input	PA2 output	RxD2 input	A18 output

Operating mode	Mode 7		
RE	0		1
PA2DDR	0	1	—
Pin function	PA2 input	PA2 output	RxD2 input

PA1/A17/TxD2

The pin function is switched as shown below according to the operating mode, bits AE3 to AE0 in PFCR, bit TE in SCR of SCI2, and bit PA1DDR.

Operating mode	Modes 4 to 6			
AE3 to AE0	B'0000 to B'1001			B'1010 to B'1111
TE	0		1	—
PA1DDR	0	1	—	—
Pin function	PA1 input	PA1 output	TxD2 output	A17 output

Operating mode	Mode 7		
TE	0		1
PA1DDR	0	1	—
Pin function	PA1 input	PA1 output	TxD2 output

Pin	Selection Method and Pin Functions
PA0/A16	The pin function is switched as shown below according to the operating mode, bits AE3 to AE0 in PFCR, and bit PA0DDR.

Operating mode	Modes 4 to 6		
AE3 to AE0	B'0000 to B'1000		B'1001 to B'1111
PA0DDR	0	1	—
Pin function	PA0 input	PA0 output	A16 output

Operating mode	Mode 7	
PA0DDR	0	1
Pin function	PA0 input	PA0 output

9.6.4 Pin Functions

Modes 4 to 6: In modes 4 to 6, port A pins function as address outputs according to the setting of AE3 to AE0 in PFCR; when they do not function as address outputs, the pins function as SCI I/O pins and I/O ports.

Port A pin functions in modes 4 to 6 are shown in figure 9-6.

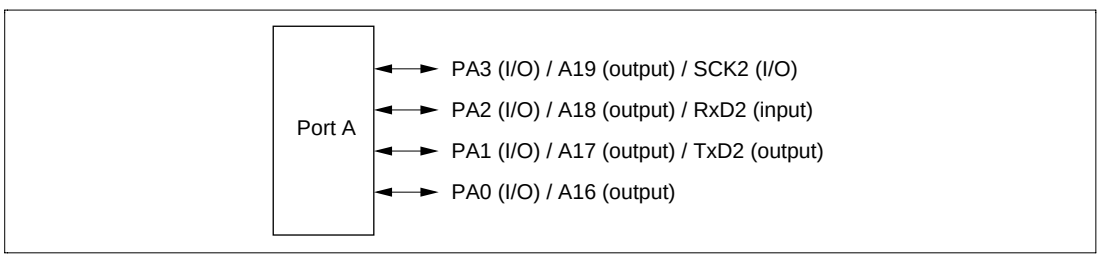


Figure 9-6 Port A Pin Functions (Modes 4 to 6)

Mode 7: In mode 7, port A pins function as I/O ports and SCI2 I/O pins (SCK2, TxD2, RxD2). Input or output can be specified for each pin on an individual bit basis. Setting a PADDR bit to 1 makes the corresponding port A pin an output port, while clearing the bit to 0 makes the pin an input port.

Port A pin functions are shown in figure 9-7.

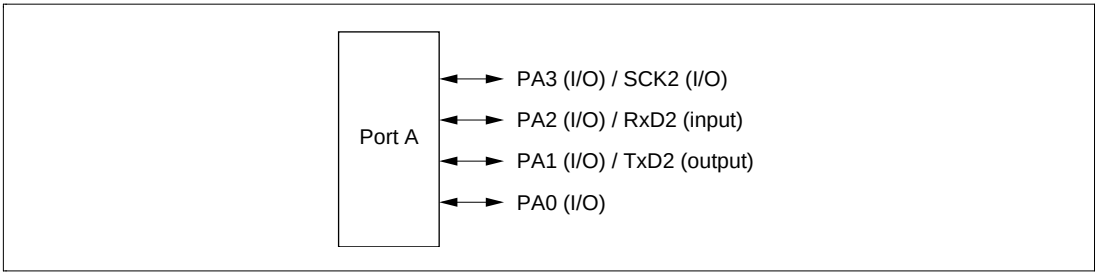


Figure 9-7 Port A Pin Functions (Mode 7)

9.6.5 MOS Input Pull-Up Function

Port A has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be specified as on or off on an individual bit basis.

In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR, in the SCI's SCMR, SMR, and SCR, and in DDR, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.

In mode 7, if a pin is in the input state in accordance with the settings in the SCI's SCMR, SMR, and SCR, and in DDR, setting the corresponding PAPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset, and in hardware standby mode. The prior state is retained in software standby mode.

Table 9-10summarizes the MOS input pull-up states.

Table 9-10 MOS Input Pull-Up States (Port A)

Pin States	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Address output or SCI output	OFF	OFF	OFF	OFF
Other than above			ON/OFF	ON/OFF

Legend:
OFF : MOS input pull-up is always off.
ON/OFF : On when PADDR = 0 and PAPCR = 1; otherwise off.

9.7 Port B

9.7.1 Overview

Port B is an 8-bit I/O port. Port B pins also function as TPU I/O pins (TIOCA3, TIOCB3, TIOCC3, TIOCD3, TIOCA4, TIOCB4, TIOCA5, TIOCB5) and as address outputs; the pin functions change according to the operating mode.

Port B has a built-in MOS input pull-up function that can be controlled by software.

Figure 9-8 shows the port B pin configuration.

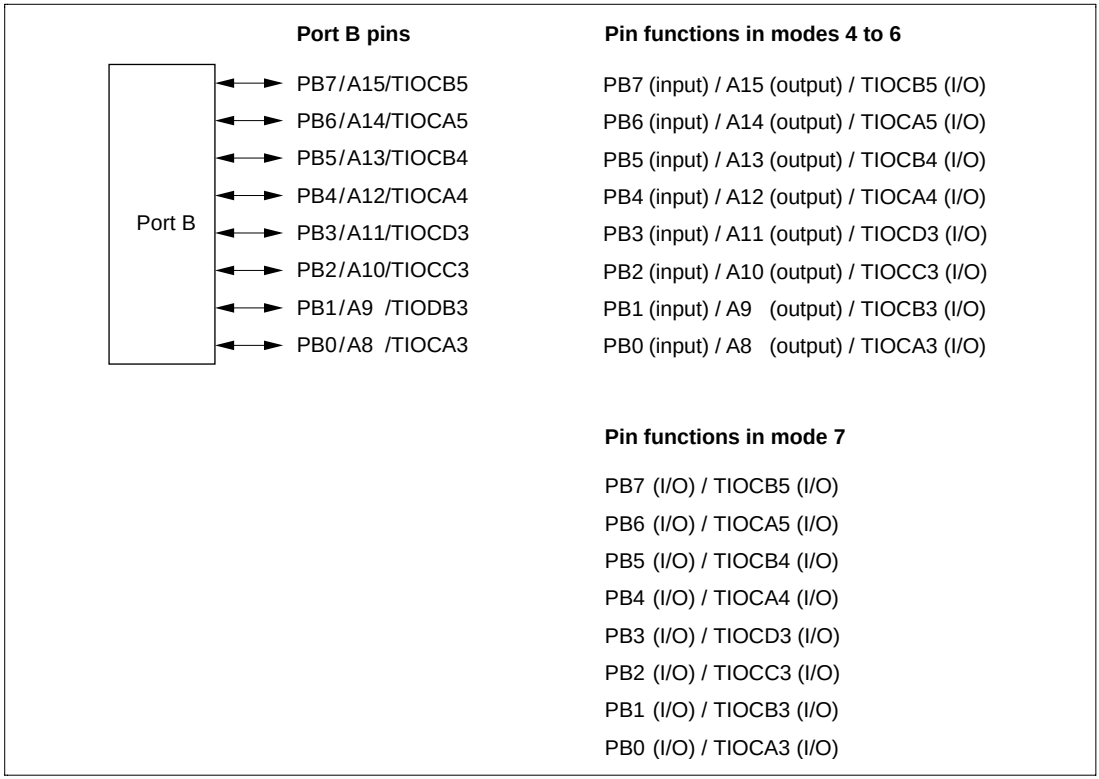


Figure 9-8 Port B Pin Functions

9.7.2 Register Configuration

Table 9-10 shows the port B register configuration.

Table 9-10 Port B Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port B data direction register	PBDDR	W	H'00	H'FE3A
Port B data register	PBDR	R/W	H'00	H'FF0A
Port B register	PORTB	R	Undefined	H'FFBA
Port B MOS pull-up control register	PBPCR	R/W	H'00	H'FF41
Port B open-drain control register	PBODR	R/W	H'00	H'FE48

Note: * Lower 16 bits of the address.

Port B Data Direction Register (PBDDR)

Bit	:	7	6	5	4	3	2	1	0
		PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PBDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port B. PBDDR cannot be read; if it is, an undefined value will be read.

PBDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when a transition is made to software standby mode.

- Modes 4 to 6

  The corresponding port B pins become address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, irrespective of the value of the PBDDR bits. When pins are not used as address outputs, setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

- Mode 7

  Setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

Port B Data Register (PBDR)

Bit	:	7	6	5	4	3	2	1	0
		PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PBDR is an 8-bit readable/writable register that stores output data for the port B pins (PB7 to PB0). PBDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port B Register (PORTB)

Bit	:	7	6	5	4	3	2	1	0
		PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PB7 to PB0.

PORTB is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port B pins (PB7 to PB0) must always be performed on PBDR.

If a port B read is performed while PBDDR bits are set to 1, the PBDR values are read. If a port B read is performed while PBDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTB contents are determined by the pin states, as PBDDR and PBDR are initialized. PORTB retains its prior state in software standby mode.

Port B MOS Pull-Up Control Register (PBPCR)

Bit	:	7	6	5	4	3	2	1	0
		PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PBPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port B on an individual bit basis.

In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR, in the TPU's TIOR, and in DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

In mode 7, if a pin is in the input state in accordance with the settings in the TPU's TIOR and in DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

PBPCR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port B Open Drain Control Register (PBODR)

Bit	:	7	6	5	4	3	2	1	0
		PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PBODR is an 8-bit readable/writable register that controls the PMOS on/off state for each port B pin (PB7 to PB0).

When pins are not address outputs in accordance with the setting of bits AE3 to AE0 in PFCR, setting a PBODR bit makes the corresponding port B pin an NMOS open-drain output, while clearing the bit to 0 makes the pin a CMOS output.

PBODR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

9.7.3 Pin Functions

Modes 4 to 6: In modes 4 to 6, the corresponding port B pins become address outputs in accordance with the setting of bits AE3 to AE0 in PFCR. When pins are not used as address outputs, they function as TPU I/O pins and I/O ports.

Port B pin functions in modes 4 to 6 are shown in figure 9-9.

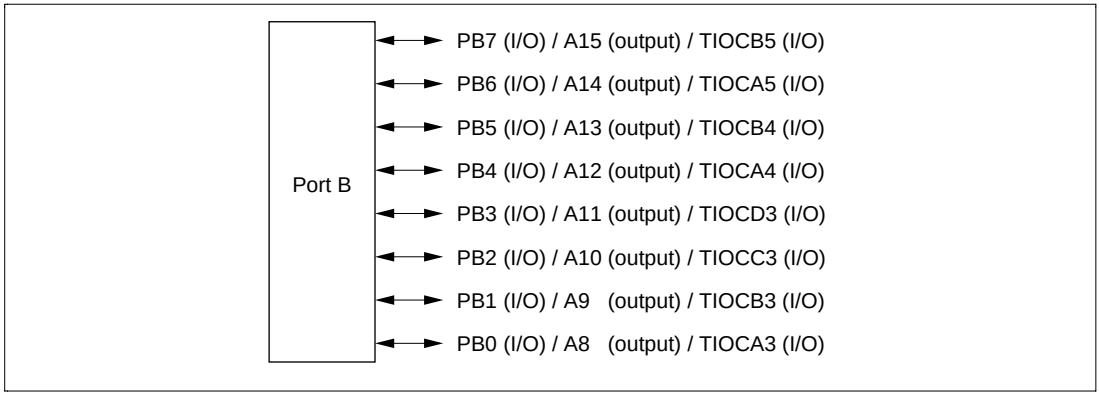


Figure 9-9 Port B Pin Functions (Modes 4 to 6)

Mode 7: In mode 7, port B pins function as I/O ports and TPU I/O pins (TIOCA3, TIOCB3, TIOCC3, TIOCD3, TIOCA4, TIOCB4, TIOCA5, and TIOCB5). Input or output can be specified for each pin on an individual bit basis. Setting a PBDDR bit to 1 makes the corresponding port B pin an output port, while clearing the bit to 0 makes the pin an input port.

Port B pin functions in mode 7 are shown in figure 9-10.

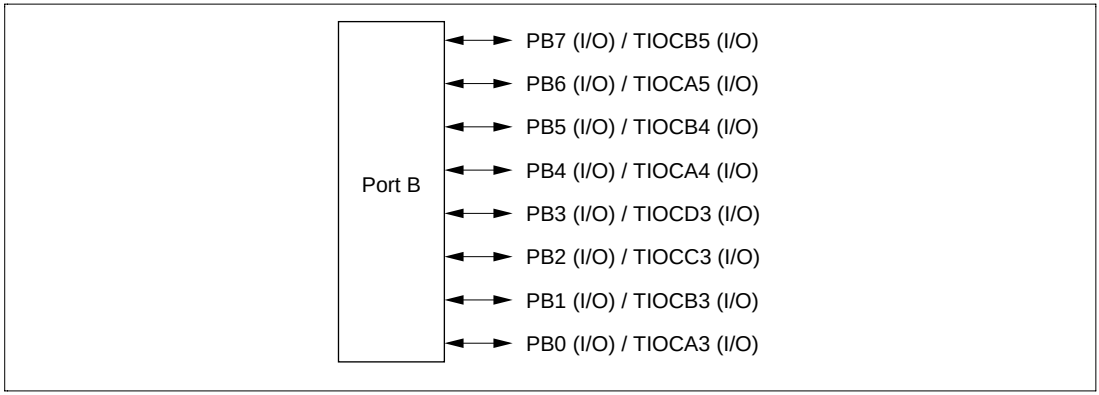


Figure 9-10 Port B Pin Functions (Mode 7)

9.7.4 MOS Input Pull-Up Function

Port B has a built-in MOS input pull-up function that can be controlled by software. MOS input pull-up can be specified as on or off on an individual bit basis.

In modes 4 to 6, if a pin is in the input state in accordance with the settings in PFCR, in the TPU's TIOR, and in DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

In mode 7, if a pin is in the input state in accordance with the settings in the TPU's TIOR and in DDR, setting the corresponding PBPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset, and in hardware standby mode. The prior state is retained by a manual reset or in software standby mode.

Table 9-11 summarizes the MOS input pull-up states.

Table 9-11 MOS Input Pull-Up States (Port B)

Pin States	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Address output or TPU output	OFF	OFF	OFF	OFF
Other than above			ON/OFF	ON/OFF

Legend:

OFF: MOS input pull-up is always off.

ON/OFF: On when PBDDR = 0 and PBPCR = 1; otherwise off.

9.8 Port C

9.8.1 Overview

Port C is an 8-bit I/O port. Port C has an address bus output function. The pin functions change according to the operating mode.

Port C has a built-in MOS input pull-up function that can be controlled by software.

Figure 9-11 shows the port C pin configuration.

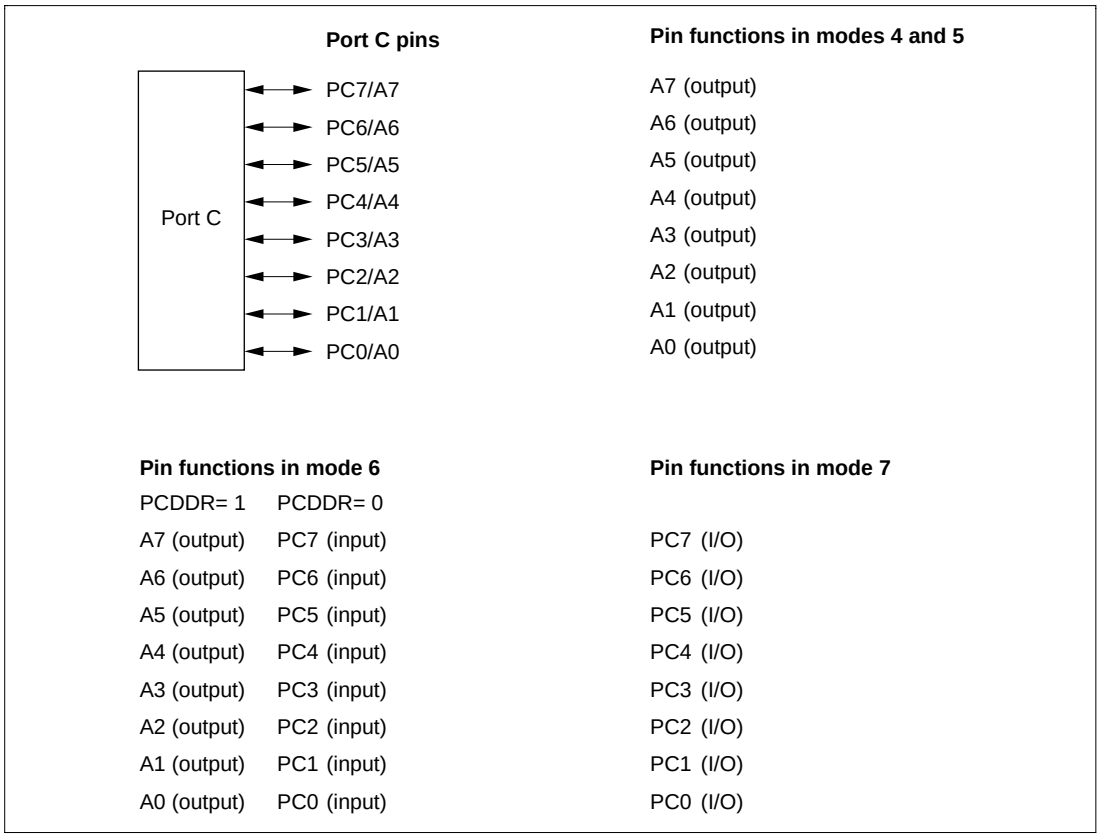


Figure 9-11 Port C Pin Functions

9.8.2 Register Configuration

Table 9-12 shows the port C register configuration.

Table 9-12 Port C Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port C data direction register	PCDDR	W	H'00	H'FE3B
Port C data register	PCDR	R/W	H'00	H'FF0B
Port C register	PORTC	R	Undefined	H'FFBB
Port C MOS pull-up control register	PCPCR	R/W	H'00	H'FF42
Port C open-drain control register	PCODR	R/W	H'00	H'FE49

Note: * Lower 16 bits of the address.

Port C Data Direction Register (PCDDR)

Bit	:	7	6	5	4	3	2	1	0
		PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PCDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port C. PCDDR cannot be read; if it is, an undefined value will be read.

PCDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode. The OPE bit in SBYCR is used to select whether the address output pins retain their output state or become high-impedance when the mode is changed to software standby mode.

- Modes 4 and 5
  The corresponding port C pins are address outputs irrespective of the value of the PCDDR bits.
- Mode 6
  Setting a PCDDR bit to 1 makes the corresponding port C pin an address output, while clearing the bit to 0 makes the pin an input port.
- Mode 7
  Setting a PCDDR bit to 1 makes the corresponding port C pin an output port, while clearing the bit to 0 makes the pin an input port.

Port C Data Register (PCDR)

Bit	:	7	6	5	4	3	2	1	0
		PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCDR is an 8-bit readable/writable register that stores output data for the port C pins (PC7 to PC0).

PCDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port C Register (PORTC)

Bit	:	7	6	5	4	3	2	1	0
		PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PC7 to PC0.

PORTC is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port C pins (PC7 to PC0) must always be performed on PCDR.

If a port C read is performed while PCDDR bits are set to 1, the PCDR values are read. If a port C read is performed while PCDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTC contents are determined by the pin states, as PCDDR and PCDR are initialized. PORTC retains its prior state in software standby mode.

Port C MOS Pull-Up Control Register (PCPCR)

Bit	:	7	6	5	4	3	2	1	0
		PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port C on an individual bit basis.

In modes 6 and 7, if PCPCR is set to 1 when the port is in the input state in accordance with the settings of PCDDR, the MOS input pull-up is set to ON.

PCPCR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

Port C Open Drain Control Register (PCODR)

Bit	7	6	5	4	3	2	1	0
	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCDDR is an 8-bit Read/Write register and controls PMOS On/Off of each pin (PC7 to PC0) of port C.

If PCODR is set to 1 by setting AE3 to AE0 in PFCR in mode other than address output mode, port C pins function as NMOS open drain outputs and when the setting is cleared to 0, the pins function as CMOS outputs.

PCODR is initialized to H'00 in reset mode or hardware standby mode. PCODR retains the last state in software standby mode.

9.8.3 Pin Functions for Each Mode

Modes 4 and 5: In mode 4 and 5, port C pins function as address outputs automatically.

Figure 9-12 shows the port C pin functions.

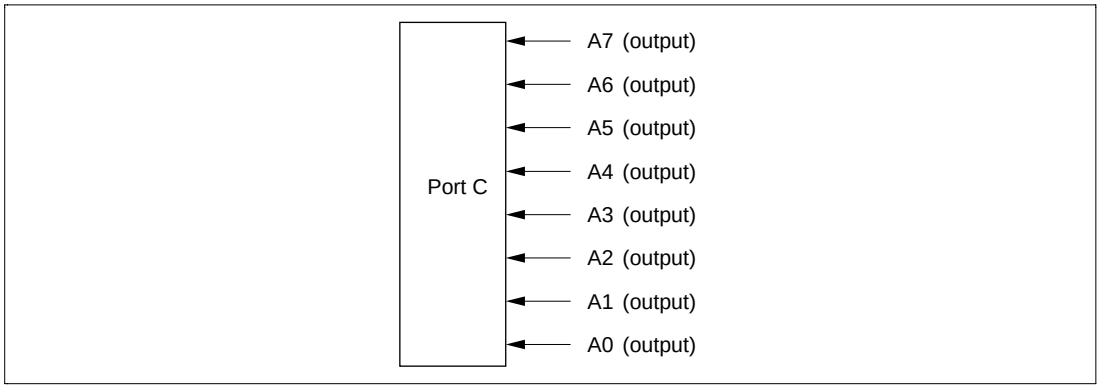


Figure 9-12 Port C Pin Functions (Modes 4 and 5)

Mode 6: In mode 6, port C pints function as address outputs or input ports and I/O can be specified in bit units. When each bit in PCDDR is set to 1, the corresponding pin functions as an address output and when the bit cleared to 0, the pin functions as an input port.

Figure 9-13 shows the port C pin functions.

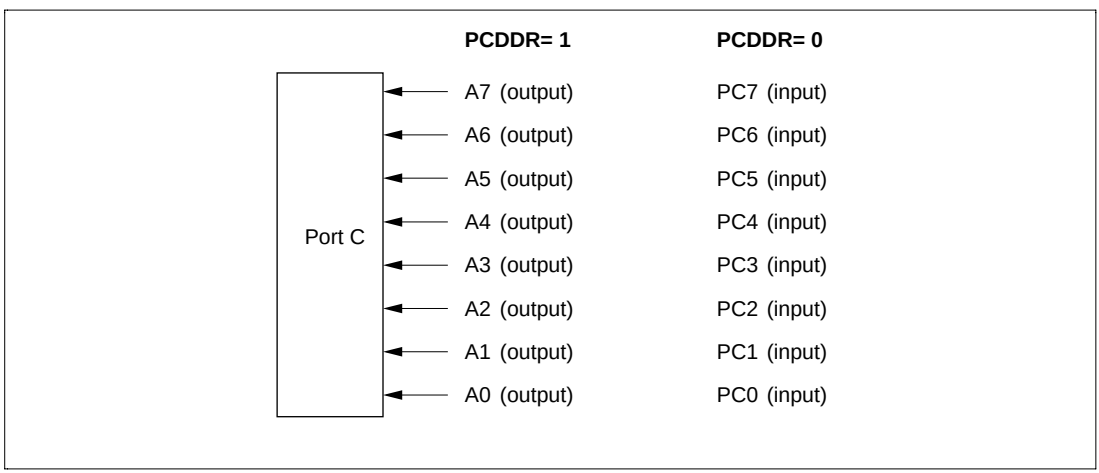


Figure 9-13 Port C Pin Functions (Mode 6)

Mode 7: In mode 7, port C pins function as I/O ports and I/O can be specified for each pin in bit units. When each bit in PCDDR is set to 1, the corresponding pin functions as an output port and when the bit is cleared to 0, the pin functions as an input port.

Figure 9-14 shows the port C pin functions.

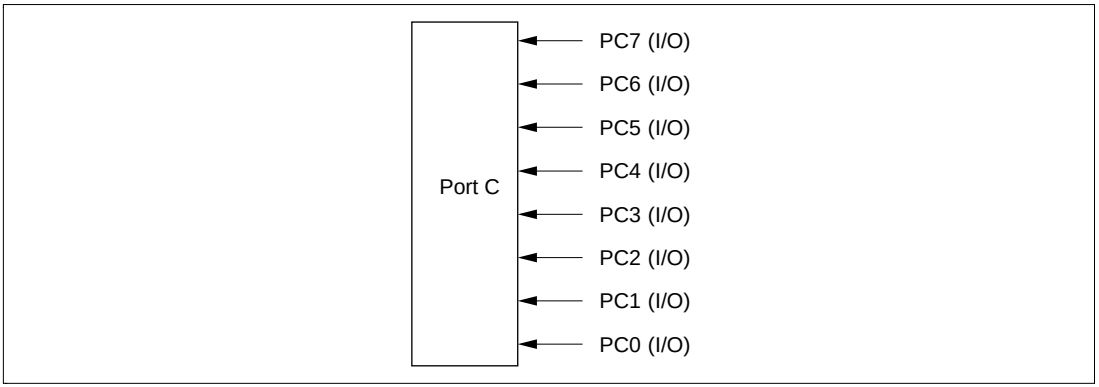


Figure 9-14 Port C Pin Functions (Mode 7)

9.8.4 MOS Input Pull-Up Function

Port C has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 6 and 7, and can be specified as on or off on an individual bit basis.

In modes 6 and 7, when PCPCR is set to 1 in the input state by setting of PCDDR, the MOS input pull-up is set to ON.

The MOS input pull-up function is in the off state after a reset, and in hardware standby mode. The prior state is retained by a manual reset or in software standby mode.

Table 9-13 summarizes the MOS input pull-up states.

Table 9-13 MOS Input Pull-Up States (Port C)

Pin States	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
Address output	OFF	OFF	OFF	OFF
Other than above			ON/OFF	ON/OFF

Legend:

OFF: MOS input pull-up is always off.

ON/OFF: On when PCDDR = 0 and PCPCR = 1; otherwise off.

9.9 Port D

9.9.1 Overview

Port D is an 8-bit I/O port. Port D has a data bus I/O function, and the pin functions change according to the operating mode.

Port D has a built-in MOS input pull-up function that can be controlled by software.

Figure 9-15 shows the port D pin configuration.

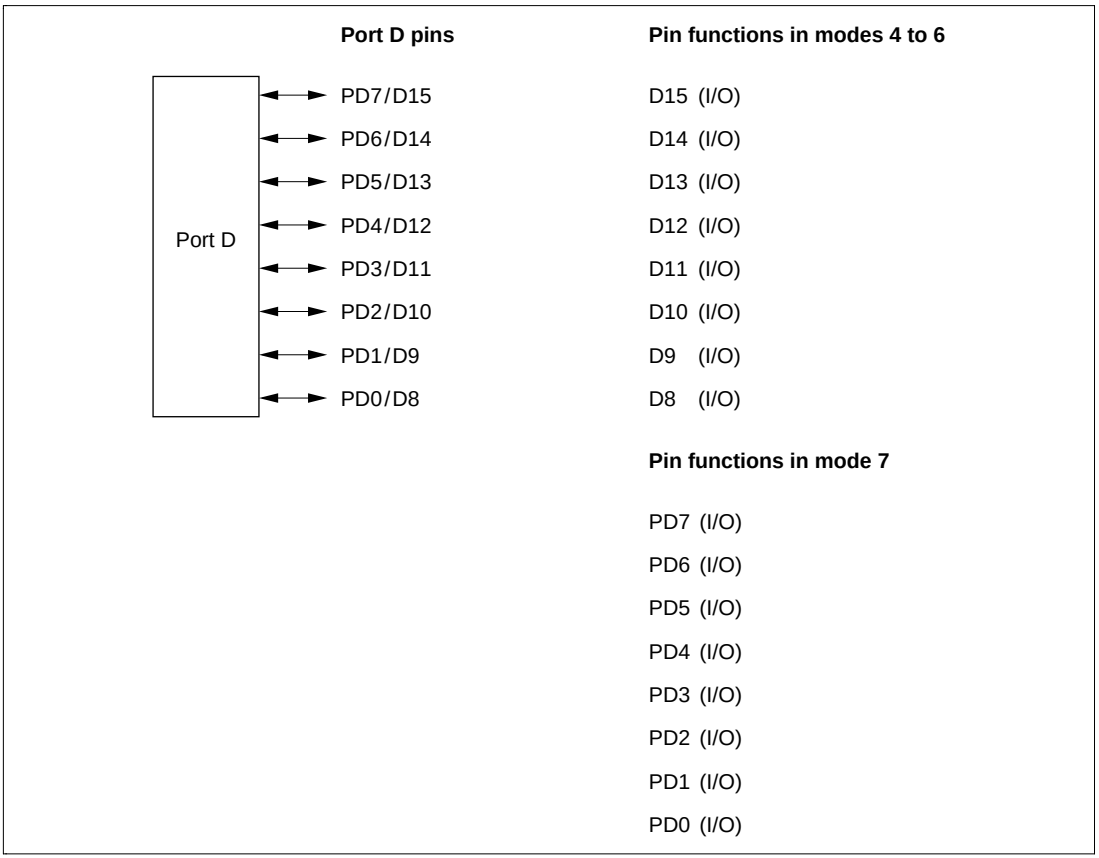


Figure 9-15 Port D Pin Functions

9.9.2 Register Configuration

Table 9-14 shows the port D register configuration.

Table 9-14 Port D Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port D data direction register	PDDDR	W	H'00	H'FE3C
Port D data register	PDDR	R/W	H'00	H'FF0C
Port D register	PORTD	R	Undefined	H'FFBC
Port D MOS pull-up control register	PDPCR	R/W	H'00	H'FE43

Note: * Lower 16 bits of the address.

Port D Data Direction Register (PDDDR)

Bit	:	7	6	5	4	3	2	1	0
		PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PDDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port D. PDDDR cannot be read; if it is, an undefined value will be read.

PDDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

- Modes 4 to 6

  The input/output direction specification by PDDDR is ignored, and port D is automatically designated for data I/O.

- Mode 7

  Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, while clearing the bit to 0 makes the pin an input port.

Port D Data Register (PDDR)

Bit	:	7	6	5	4	3	2	1	0
		PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDDR is an 8-bit readable/writable register that stores output data for the port D pins (PD7 to PD0).

PDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port D Register (PORTD)

Bit	:	7	6	5	4	3	2	1	0
		PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PD7 to PD0.

PORTD is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port D pins (PD7 to PD0) must always be performed on PDDR.

If a port D read is performed while PDDDR bits are set to 1, the PDDR values are read. If a port D read is performed while PDDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTD contents are determined by the pin states, as PDDDR and PDDR are initialized. PORTD retains its prior state in software standby mode.

Port D MOS Pull-Up Control Register (PDPCR)

Bit	:	7	6	5	4	3	2	1	0
		PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port D on an individual bit basis.

When a PDDDR bit is cleared to 0 (input port setting) in mode 7, setting the corresponding PDPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PDPCR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

9.9.3 Pin Functions

Modes 4 to 6: In modes 4 to 6, port D pins are automatically designated as data I/O pins.

Port D pin functions in modes 4 to 6 are shown in figure 9-16.

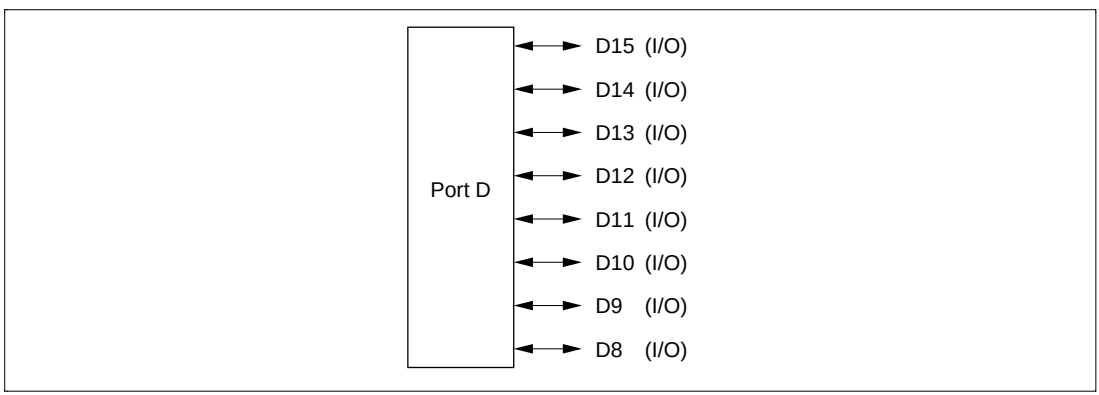


Figure 9-16 Port D Pin Functions (Modes 4 to 6)

Mode 7: In mode 7, port D pins function as I/O ports. Input or output can be specified for each pin on an individual bit basis. Setting a PDDDR bit to 1 makes the corresponding port D pin an output port, while clearing the bit to 0 makes the pin an input port.

Port D pin functions in mode 7 are shown in figure 9-17.

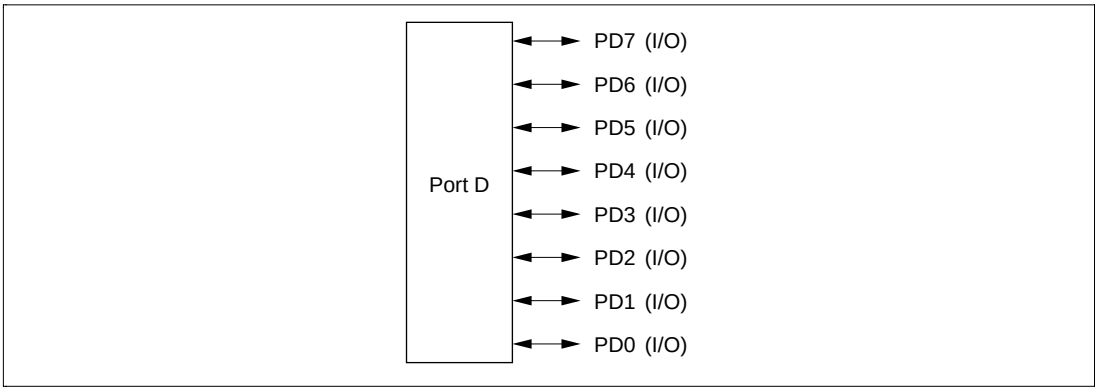


Figure 9-17 Port D Pin Functions (Mode 7)

9.9.4 MOS Input Pull-Up Function

Port D has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in mode 7, and can be specified as on or off on an individual bit basis.

When a PDDDR bit is cleared to 0 in mode 7, setting the corresponding PDPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset, and in hardware standby mode. The prior state is retained in software standby mode.

Table 9-15 summarizes the MOS input pull-up states.

Table 9-15 MOS Input Pull-Up States (Port D)

Modes	Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
4 to 6	OFF	OFF	OFF	OFF
7			ON/OFF	ON/OFF

Legend:
OFF: MOS input pull-up is always off.
ON/OFF: On when PDDDR = 0 and PDPCR = 1; otherwise off.

9.10 Port E

9.10.1 Overview

Port E is an 8-bit I/O port. Port E has a data bus I/O function, and the pin functions change according to the operating mode and whether 8-bit or 16-bit bus mode is selected.

Port E has a built-in MOS input pull-up function that can be controlled by software.

Figure 9-18 shows the port E pin configuration.

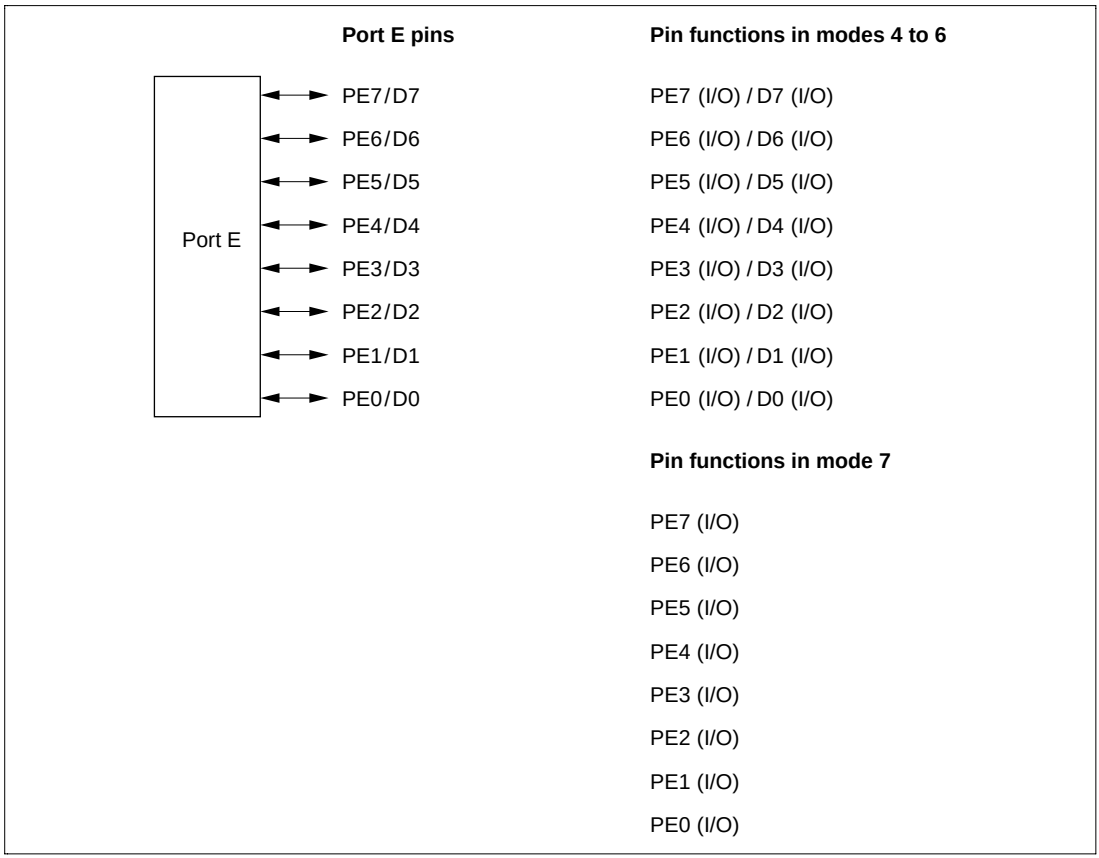


Figure 9-18 Port E Pin Functions

9.10.2 Register Configuration

Table 9-16 shows the port E register configuration.

Table 9-16 Port E Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port E data direction register	PEDDR	W	H'00	H'FE3D
Port E data register	PEDR	R/W	H'00	H'FF0D
Port E register	PORTE	R	Undefined	H'FFBD
Port E MOS pull-up control register	PEPCR	R/W	H'00	H'FE44

Note: * Lower 16 bits of the address.

Port E Data Direction Register (PEDDR)

Bit	:	7	6	5	4	3	2	1	0
		PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PEDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port E. PEDDR cannot be read; if it is, an undefined value will be read.

PEDDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state by a manual reset or in software standby mode.

- Modes 4 to 6

  When 8-bit bus mode has been selected, port E pins function as I/O ports. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

  When 16-bit bus mode has been selected, the input/output direction specification by PEDDR is ignored, and port E is designated for data I/O.

  For details of 8-bit and 16-bit bus modes, see section 7, Bus Controller.

- Mode 7

  Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

Port E Data Register (PEDR)

Bit	:	7	6	5	4	3	2	1	0
		PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEDR is an 8-bit readable/writable register that stores output data for the port E pins (PE7 to PE0).

PEDR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port E Register (PORTE)

Bit	:	7	6	5	4	3	2	1	0
		PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by state of pins PE7 to PE0.

PORTE is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port E pins (PE7 to PE0) must always be performed on PEDR.

If a port E read is performed while PEDDR bits are set to 1, the PEDR values are read. If a port E read is performed while PEDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTE contents are determined by the pin states, as PEDDR and PEDR are initialized. PORTE retains its prior state in software standby mode.

Port E MOS Pull-Up Control Register (PEPCR)

Bit	:	7	6	5	4	3	2	1	0
		PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEPCR is an 8-bit readable/writable register that controls the MOS input pull-up function incorporated into port E on an individual bit basis.

When a PEDDR bit is cleared to 0 (input port setting) with 8-bit bus mode selected in mode 4, 5, or 6, or in mode 7, setting the corresponding PEPCR bit to 1 turns on the MOS input pull-up for the corresponding pin.

PEPCR is initialized to H'00 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

9.10.3 Pin Functions

Modes 4 to 6: In modes 4 to 6, when 8-bit access is designated and 8-bit bus mode is selected, port E pins are automatically designated as I/O ports. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

When 16-bit bus mode is selected, the input/output direction specification by PEDDR is ignored, and port E is designated for data I/O.

Port E pin functions in modes 4 to 6 are shown in figure 9-19.

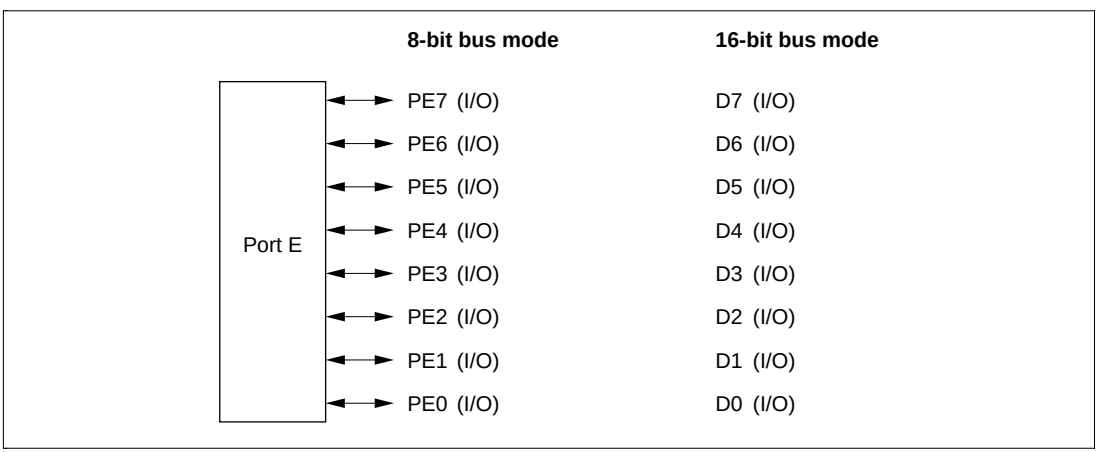


Figure 9-19 Port E Pin Functions (Modes 4 to 6)

Mode 7: In mode 7, port E pins function as I/O ports. Input or output can be specified for each pin on a bit-by-bit basis. Setting a PEDDR bit to 1 makes the corresponding port E pin an output port, while clearing the bit to 0 makes the pin an input port.

Port E pin functions in mode 7 are shown in figure 9-20.

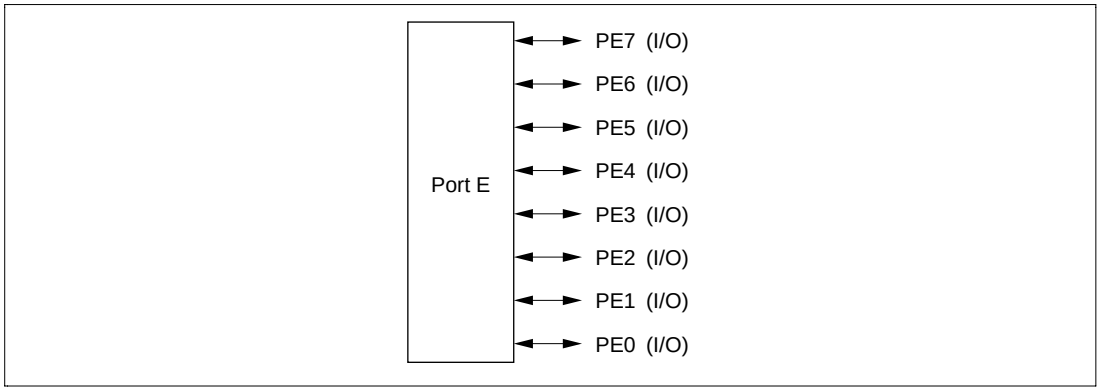


Figure 9-20 Port E Pin Functions (Mode 7)

9.10.4 MOS Input Pull-Up Function

Port E has a built-in MOS input pull-up function that can be controlled by software. This MOS input pull-up function can be used in modes 4 to 6 when 8-bit bus mode is selected, or in mode 7, and can be specified as on or off on an individual bit basis.

When a PEDDR bit is cleared to 0 in mode 4 to 6 when 8-bit bus mode is selected, or in mode 7, setting the corresponding PEPCR bit to 1 turns on the MOS input pull-up for that pin.

The MOS input pull-up function is in the off state after a reset, and in hardware standby mode. The prior state is retained in software standby mode.

Table 9-17 summarizes the MOS input pull-up states.

Table 9-17 MOS Input Pull-Up States (Port E)

Modes		Reset	Hardware Standby Mode	Software Standby Mode	In Other Operations
7		OFF	OFF	ON/OFF	ON/OFF
4 to 6	8-bit bus				
	16-bit bus			OFF	OFF

Legend:
OFF: MOS input pull-up is always off.
ON/OFF: On when PEDDR = 0 and PEPCR = 1; otherwise off.

9.11 Port F

9.11.1 Overview

Port F is a 6-bit I/O port. Port F pins also function as external interrupt input pins ($\overline{IRQ2}$ and $\overline{IRQ3}$), A/D trigger input pin ($\overline{ADTRG}$), bus control signal input/output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$), and the system clock (ø) output pin.

Figure 9-21 shows the port F pin configuration.

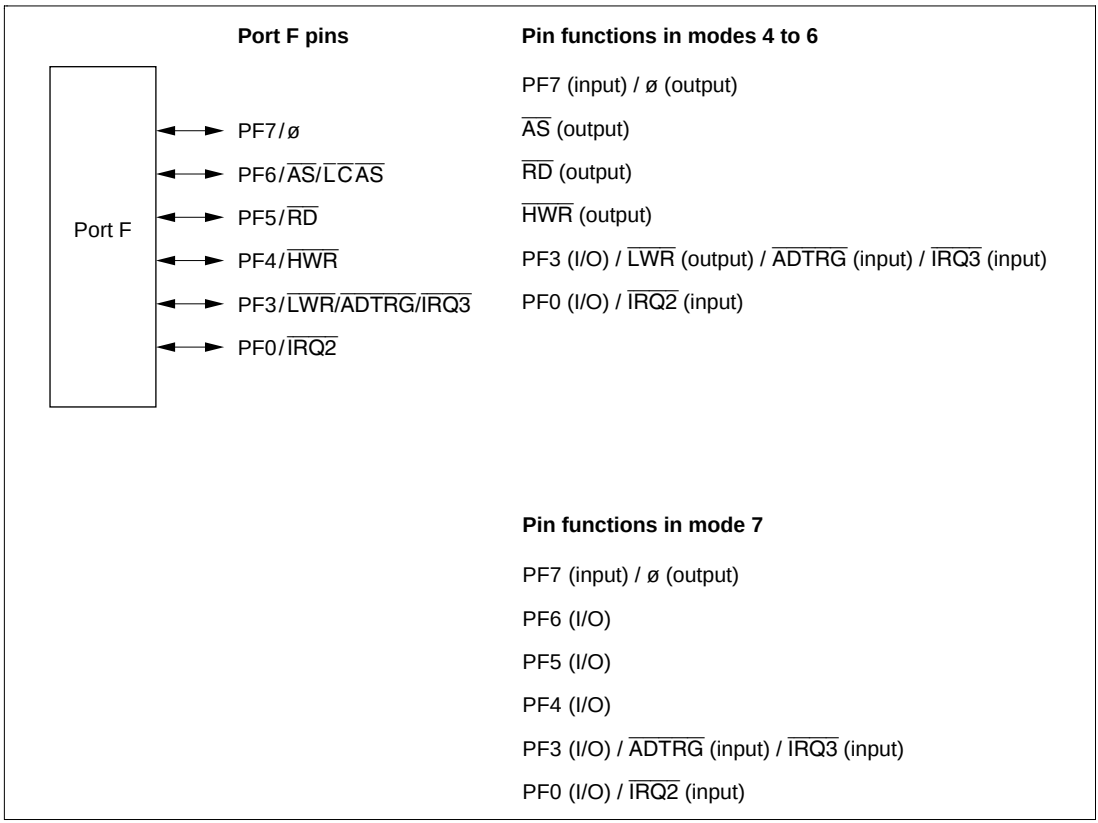


Figure 9-21 Port F Pin Functions

9.11.2 Register Configuration

Table 9-18 shows the port F register configuration.

Table 9-18 Port F Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
Port F data direction register	PFDDR	W	B'10000**0*[2]/ B'00000**0*[2]	H'FE3E
Port F data register	PFDR	R/W	B'00000**0	H'FF0E
Port F register	PORTF	R	Undefined	H'FFBE

Notes: *1 Lower 16 bits of the address.
*2 Initial value depends on the mode.

Port F Data Direction Register (PFDDR)

Bit :	7	6	5	4	3	2	1	0
	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	—	—	PF0DDR
Modes 4 to 6								
Initial value :	1	0	0	0	0	undefined	undefined	0
R/W :	W	W	W	W	W	—	—	W
Mode 7								
Initial value :	0	0	0	0	0	undefined	undefined	0
R/W :	W	W	W	W	W	—	—	W

PFDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port F. PFDDR cannot be read; if it is, an undefined value will be read.

PFDDR is initialized by a reset, and in hardware standby mode, to B'10000**0 in modes 4 to 6, and to B'00000**0 in mode 7. It retains its prior state in software standby mode. The OPE bit in SBYCR is used to select whether the bus control output pins retain their output state or become high-impedance when a transition is made to software standby mode.

- Modes 4 to 6

  Pin PF7 functions as the ø output pin when the corresponding PFDDR bit is set to 1, and as an input port when the bit is cleared to 0.

  The input/output direction specified by PFDDR is ignored for pins PF6 to PF3, which are automatically designated as bus control outputs ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$).

  Pin PF0 is setting a PFDDR bit to 1 makes the corresponding port F pin an output port, while clearing the bit to 0 makes the pin an input port.

- Mode 7

  Setting a PFDDR bit to 1 makes the corresponding port F pin PF6 to PF3, PF0 an output port, or in the case of pin PF7, the ø output pin. Clearing the bit to 0 makes the pin an input port.

Port F Data Register (PFDR)

Bit	:	7	6	5	4	3	2	1	0
		PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	—	—	PF0DR
Initial value	:	0	0	0	0	0	undefined	undefined	0
R/W	:	R/W	R/W	R/W	R/W	R/W	—	—	R/W

PFDR is an 8-bit readable/writable register that stores output data for the port F pins (PF7 to PF3, PF0).

PFDR is initialized to B'00000**0 by a reset, and in hardware standby mode. It retains its prior state in software standby mode.

Port F Register (PORTF)

Bit	:	7	6	5	4	3	2	1	0
		PF7	PF6	PF5	PF4	PF3	—	—	PF0
Initial value	:	—*	—*	—*	—*	—*	undefined	undefined	—*
R/W	:	R	R	R	R	R	—	—	R

Note: * Determined by state of pins PF7 to PF3, PF0.

PORTF is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port F pins (PF7 to PF3, PF0) must always be performed on PFDR.

If a port F read is performed while PFDDR bits are set to 1, the PFDR values are read. If a port F read is performed while PFDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTF contents are determined by the pin states, as PFDDR and PFDR are initialized. PORTF retains its prior state in software standby mode.

9.11.3 Pin Functions

Port F pins also function as external interrupt input pins (IRQ2 and IRQ3), A/D trigger input pin ($\overline{\text{ADTRG}}$), bus control signal input/output pins ($\overline{\text{AS}}$, $\overline{\text{RD}}$, $\overline{\text{HWR}}$, $\overline{\text{LWR}}$), and the system clock (ø) output pin. The pin functions differ between modes 4 to 6, and mode 7. Port F pin functions are shown in table 9-19.

Table 9-19 Port F Pin Functions

Pin	Selection Method and Pin Functions
PF7/ø	The pin function is switched as shown below according to bit PF7DDR.

PF7DDR	0	1
Pin function	PF7 input	ø output

PF6/$\overline{\text{AS}}$	The pin function is switched as shown below according to bit PF6DDR.

Operating Mode	Modes 4 to 6	Mode 7	
PF6DDR	—	0	1
Pin function	$\overline{\text{AS}}$ output	PF6 input	PF6 output

PF5/$\overline{\text{RD}}$	The pin function is switched as shown below according to the operating mode and bit PF5DDR.

Operating Mode	Modes 4 to 6	Mode 7	
PF5DDR	—	0	1
Pin function	$\overline{\text{RD}}$ output	PF5 input	PF5 output

PF4/$\overline{\text{HWR}}$	The pin function is switched as shown below according to the operating mode and bit PF4DDR.

Operating Mode	Modes 4 to 6	Mode 7	
PF4DDR	—	0	1
Pin function	$\overline{\text{HWR}}$ output	PF4 input	PF4 output

Pin	Selection Method and Pin Functions
PF3/$\overline{\text{LWR}}$/ $\overline{\text{ADTRG}}$/$\overline{\text{IRQ3}}$	The pin function is switched as shown below according to the operating mode, the bus mode, A/D converter bits TRGS1 and TRGS0, and bit PF3DDR.

Operating mode	Modes 4 to 6			Mode 7	
Bus mode	16-bit bus mode	8-bit bus mode		—	
PF3DDR	—	0	1	0	1
Pin function	$\overline{\text{LWR}}$ output pin	PF3 input pin	PF3 output pin	PF3 input pin	PF3 output pin
		$\overline{\text{ADTRG}}$ input pin*1			
		$\overline{\text{IRQ3}}$ input pin*2			

Notes: *1 $\overline{\text{ADTRG}}$ input when TRGS0=TRGS1=1.

*2 When used as an external interrupt input pin, do not use as an I/O pin for another function.

PF0/$\overline{\text{IRQ2}}$ — The pin function is switched as shown below according to the combination of the operating mode, and bits BRLE and PF0DDR.

PF0DDR	0	1
Pin function	PF0 input	PF0 output
	$\overline{\text{IRQ2}}$ input	

9.12 Port H

9.12.1 Overview

Port H is an 8-bit I/O port. Port H pins also function as motor control PWM timer output pins (PWM1A to PWM1H).

Figure 9-22 shows the port H pin configuration.

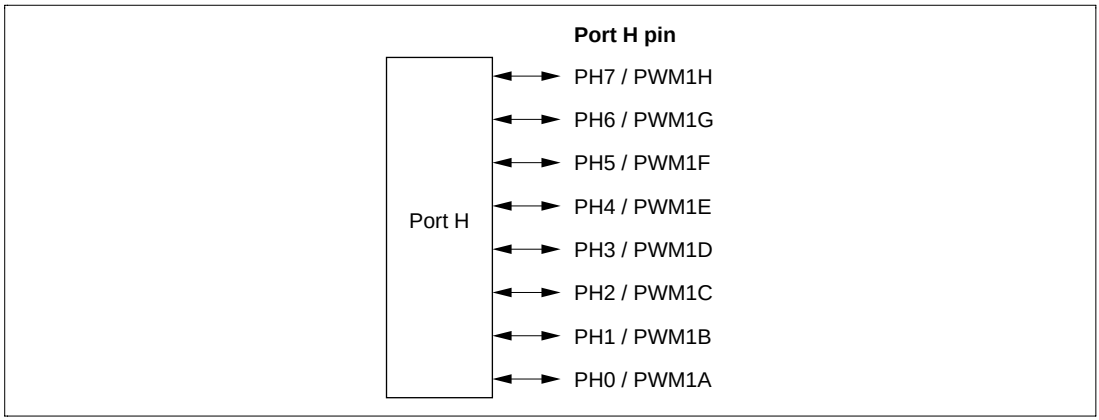


Figure 9-22 Port H Pin Functions

9.12.2 Register Configuration

Table 9-20 shows the port H register configuration.

Table 9-20 Port H Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port H data direction register	PHDDR	W	H'00	H'FC20
Port H data register	PHDR	RW	H'00	H'FC24
Port H register	PORTH	R	Undefined	H'FC28

Note: * Lower 16 bits of the address.

Port H Data Direction Register (PHDDR)

Bit	:	7	6	5	4	3	2	1	0
		PH7DDR	PH6DDR	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PHDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port H. PHDDR cannot be read. If it is, an undefined value will be read.

PHDDR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

Port H Data Register (PHDR)

Bit	:	7	6	5	4	3	2	1	0
		PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PHDR is an 8-bit readable/writeable register that stores output data for the port H pins (PH7 to PH0).

PHDR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

Port H Register (PORTH)

Bit	:	7	6	5	4	3	2	1	0
		PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

Note: * Determined by the state of PH7 to PH0

PORTH is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port H pins (PH7 to PH0) must always be performed on PHDR.

If a port H read is performed while PHDDR bits are set to 1, the PHDR values are read. If a port H read is performed while PHDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTH contents are determined by the pin states, as PHDDR and PHDR are initialized. PORTH retains its prior state in software standby mode.

9.12.3 Pin Functions

As shown in Table 9-21, the port H pin functions can be switched, bit by bit, by changing the values of OE1A to OE1H of motor control PWM timer PWOCR1 and PHDDR.

Table 9-21 Port H Pin Functions

0E1A to 0E1H	1	0	
PHDDR	—	0	1
Pin function	PWM output	PH7 to PH0 input	PH7 to PH0 output

9.13 Port J

9.13.1 Overview

Port J is an 8-bit I/O port. Port J pins also function as motor control PWM timer output pins (PWM2A to PWM2H).

Figure 9-23 shows the port J pin configuration.

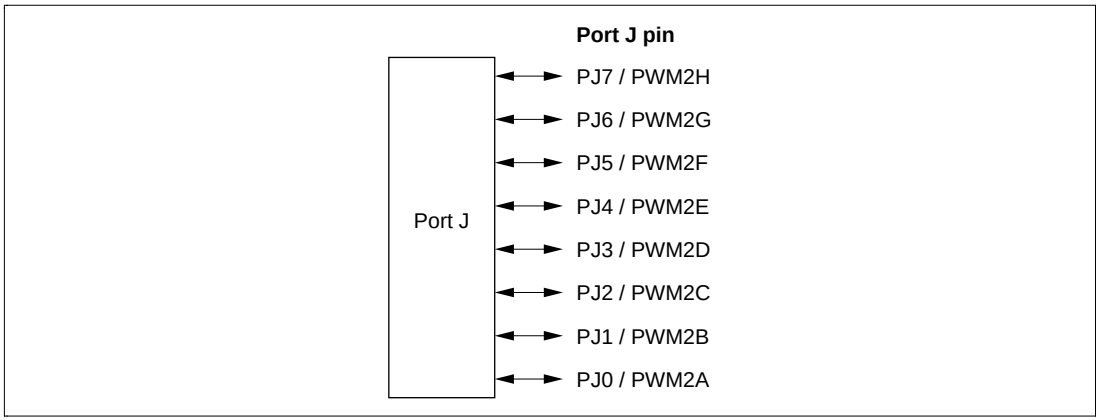


Figure 9-23 Port J Pin Functions

9.13.2 Register Configuration

Table 9-22 shows the port J register configuration.

Table 9-22 Port J Registers

Name	Abbreviation	R/W	Initial Value	Address*
Port J data direction register	PJDDR	W	H'00	H'FC21
Port J data register	PJDR	RW	H'00	H'FC25
Port J register	PORTJ	R	Undefined	H'FC29

Note: * Lower 16 bits of the address

Port J Data Direction Register (PJDDR)

Bit	:	7	6	5	4	3	2	1	0
		PJ7DDR	PJ6DDR	PJ5DDR	PJ4DDR	PJ3DDR	PJ2DDR	PJ1DDR	PJ0DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

PJDDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port J. PJDDR cannot be read. If it is, an undefined value will be read.

PJDDR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

Port J Data Register (PJDR)

Bit	:	7	6	5	4	3	2	1	0
		PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PJDR is an 8-bit readable/writeable register that stores output data for the port J pins (PJ7 to PJ0).

PJDR is initialized to H'00 by a reset and in hardware standby mode. It retains its prior state in software standby mode.

Port J Register (PORTJ)

Bit	:	7	6	5	4	3	2	1	0
		PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
Initial value	:	—*	—*	—*	—*	—*	—*	—*	—*
R/W	:	R	R	R	R	R	R	R	R

PORTJ is an 8-bit read-only register that shows the pin states. It cannot be written to. Writing of output data for the port J pins (PJ7 to PJ0) must always be performed on PJDR.

If a port J read is performed while PJDDR bits are set to 1, the PJDR values are read. If a port J read is performed while PJDDR bits are cleared to 0, the pin states are read.

After a reset and in hardware standby mode, PORTJ contents are determined by the pin states, as PJDDR and PJDR are initialized. PORTJ retains its prior state in software standby mode.

9.13.3 Pin Functions

As shown in Table 9-23, the port J pin functions can be switched, bit by bit, by changing the values of OE2A to OE2H of motor control PWM timer PWOCR2 and PJDDR.

Table 9-23 Port J Pin Functions

OE2A to OE2H	1	0	
PJDDR	—	0	1
Pin function	PWM output	PJ7 to PJ0 input	PJ7 to PJ0 output

Section 10 16-Bit Timer Pulse Unit (TPU)

10.1 Overview

The chip has an on-chip 16-bit timer pulse unit (TPU) that comprises six 16-bit timer channels.

10.1.1 Features

- Maximum 16-pulse input/output
  - — A total of 16 timer general registers (TGRs) are provided (four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5), each of which can be set independently as an output compare/input capture register
  - — TGRC and TGRD for channels 0 and 3 can also be used as buffer registers
- Selection of 8 counter input clocks for each channel
- The following operations can be set for each channel:
  - — Waveform output at compare match: Selection of 0, 1, or toggle output
  - — Input capture function: Selection of rising edge, falling edge, or both edge detection
  - — Counter clear operation: Counter clearing possible by compare match or input capture
  - — Synchronous operation: Multiple timer counters (TCNT) can be written to simultaneously
  - — Simultaneous clearing by compare match and input capture possible
  - — Register simultaneous input/output possible by counter synchronous operation
  - — PWM mode: Any PWM output duty can be set
  - — Maximum of 15-phase PWM output possible by combination with synchronous operation
- Buffer operation settable for channels 0 and 3
  - — Input capture register double-buffering possible
  - — Automatic rewriting of output compare register possible
- Phase counting mode settable independently for each of channels 1, 2, 4, and 5
  - — Two-phase encoder pulse up/down-count possible
- Cascaded operation
  - — Channel 2 (channel 5) input clock operates as 32-bit counter by setting channel 1 (channel 4) overflow/underflow
- Fast access via internal 16-bit bus
  - — Fast access is possible via a 16-bit bus interface
- 26 interrupt sources

— For channels 0 and 3, four compare match/input capture dual-function interrupts and one overflow interrupt can be requested independently
— For channels 1, 2, 4, and 5, two compare match/input capture dual-function interrupts, one overflow interrupt, and one underflow interrupt can be requested independently

- Automatic transfer of register data
  — Block transfer, 1-word data transfer, and 1-byte data transfer possible by data transfer controller (DTC)

- Programmable pulse generator (PPG) output trigger can be generated
  — Channel 0 to 3 compare match/input capture signals can be used as PPG output trigger

- A/D converter conversion start trigger can be generated
  — Channel 0 to 5 compare match A/input capture A signals can be used as A/D converter conversion start trigger

- Module stop mode can be set
  — As the initial setting, TPU operation is halted. Register access is enabled by exiting module stop mode.

Table 10-1 lists the functions of the TPU.

Table 10-1 TPU Functions

Item		Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
Count clock		ø/1 ø/4 ø/16 ø/64 TCLKA TCLKB TCLKC TCLKD	ø/1 ø/4 ø/16 ø/64 ø/256 TCLKA TCLKB	ø/1 ø/4 ø/16 ø/64 ø/1024 TCLKA TCLKB TCLKC	ø/1 ø/4 ø/16 ø/64 ø/256 ø/1024 ø/4096 TCLKA	ø/1 ø/4 ø/16 ø/64 ø/1024 TCLKA TCLKC	ø/1 ø/4 ø/16 ø/64 ø/256 TCLKA TCLKC TCLKD
General registers		TGR0A TGR0B	TGR1A TGR1B	TGR2A TGR2B	TGR3A TGR3B	TGR4A TGR4B	TGR5A TGR5B
General registers/ buffer registers		TGR0C TGR0D	—	—	TGR3C TGR3D	—	—
I/O pins		TIOCA0 TIOCB0 TIOCC0 TIOCD0	TIOCA1 TIOCB1	TIOCA2 TIOCB2	TIOCA3 TIOCB3 TIOCC3 TIOCD3	TIOCA4 TIOCB4	TIOCA5 TIOCB5
Counter clear function		TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
Compare match output	0 output	○	○	○	○	○	○
	1 output	○	○	○	○	○	○
	Toggle output	○	○	○	○	○	○
Input capture function		○	○	○	○	○	○
Synchronous operation		○	○	○	○	○	○
PWM mode		○	○	○	○	○	○
Phase counting mode		—	○	○	—	○	○
Buffer operation		○	—	—	○	—	—

Item	Channel 0	Channel 1	Channel 2	Channel 3	Channel 4	Channel 5
DTC activation	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture	TGR compare match or input capture
A/D converter trigger	TGR0A compare match or input capture	TGR1A compare match or input capture	TGR2A compare match or input capture	TGR3A compare match or input capture	TGR4A compare match or input capture	TGR5A compare match or input capture
PPG trigger	TGR0A/ TGR0B compare match or input capture	TGR1A/ TGR1B compare match or input capture	TGR2A/ TGR2B compare match or input capture	TGR3A/ TGR3B compare match or input capture	—	—
Interrupt sources	5 sources • Compare match or input capture 0A • Compare match or input capture 0B • Compare match or input capture 0C • Compare match or input capture 0D • Overflow	4 sources • Compare match or input capture 1A • Compare match or input capture 1B • Overflow • Underflow	4 sources • Compare match or input capture 2A • Compare match or input capture 2B • Overflow • Underflow	5 sources • Compare match or input capture 3A • Compare match or input capture 3B • Compare match or input capture 3C • Compare match or input capture 3D • Overflow	4 sources • Compare match or input capture 4A • Compare match or input capture 4B • Overflow • Underflow	4 sources • Compare match or input capture 5A • Compare match or input capture 5B • Overflow • Underflow

Legend

○ : Possible

— : Not possible

10.1.2 Block Diagram

Figure 10-1 shows a block diagram of the TPU.

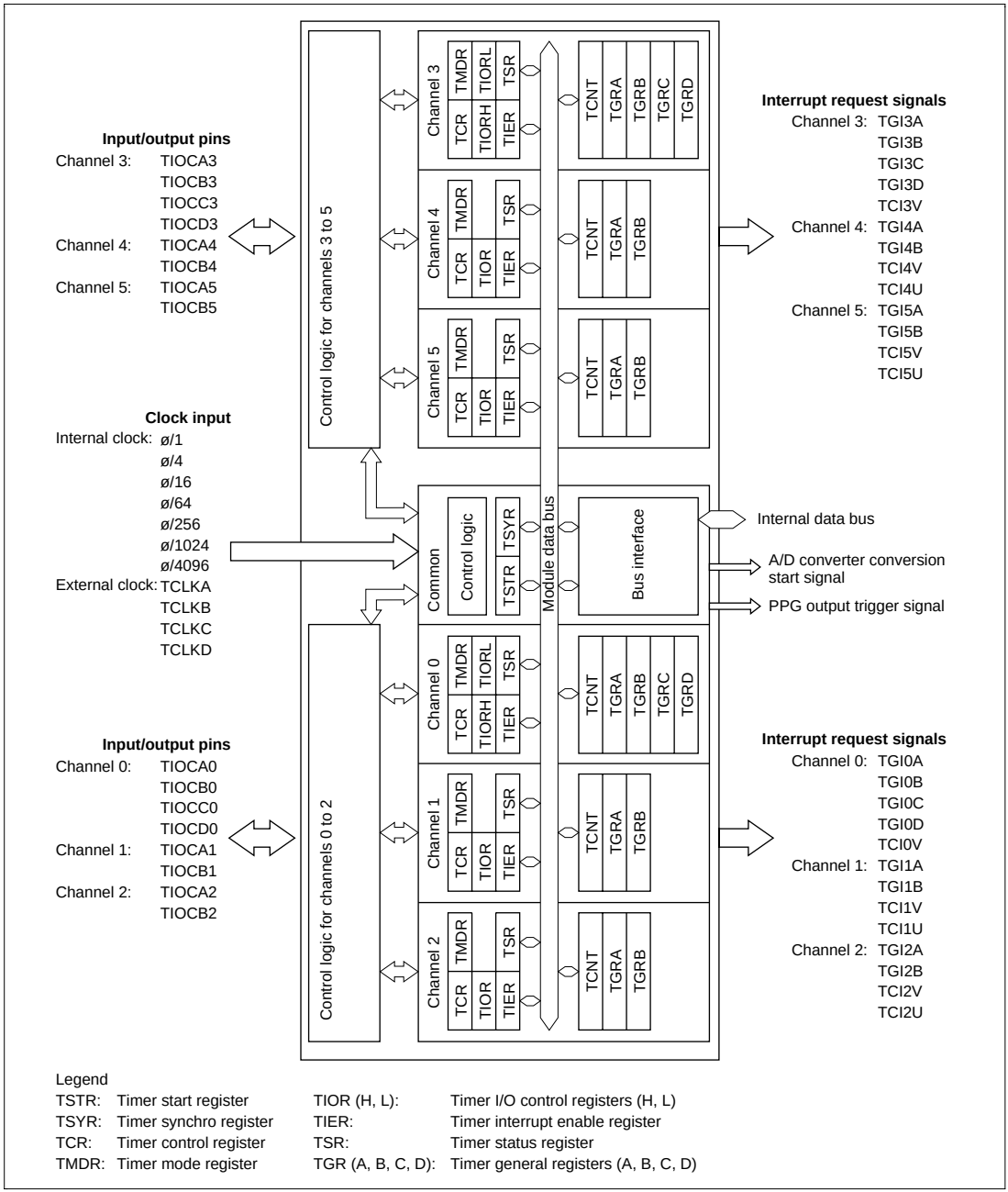


Figure 10-1 Block Diagram of TPU

10.1.3 Pin Configuration

Table 10-2 summarizes the TPU pins.

Table 10-2 TPU Pins

Channel	Name	Symbol	I/O	Function
All	Clock input A	TCLKA	Input	External clock A input pin (Channel 1 and 5 phase counting mode A phase input)
	Clock input B	TCLKB	Input	External clock B input pin (Channel 1 and 5 phase counting mode B phase input)
	Clock input C	TCLKC	Input	External clock C input pin (Channel 2 and 4 phase counting mode A phase input)
	Clock input D	TCLKD	Input	External clock D input pin (Channel 2 and 4 phase counting mode B phase input)
0	Input capture/out compare match A0	TIOCA0	I/O	TGR0A input capture input/output compare output/PWM output pin
	Input capture/out compare match B0	TIOCB0	I/O	TGR0B input capture input/output compare output/PWM output pin
	Input capture/out compare match C0	TIOCC0	I/O	TGR0C input capture input/output compare output/PWM output pin
	Input capture/out compare match D0	TIOCD0	I/O	TGR0D input capture input/output compare output/PWM output pin
1	Input capture/out compare match A1	TIOCA1	I/O	TGR1A input capture input/output compare output/PWM output pin
	Input capture/out compare match B1	TIOCB1	I/O	TGR1B input capture input/output compare output/PWM output pin
2	Input capture/out compare match A2	TIOCA2	I/O	TGR2A input capture input/output compare output/PWM output pin
	Input capture/out compare match B2	TIOCB2	I/O	TGR2B input capture input/output compare output/PWM output pin

Channel	Name	Symbol	I/O	Function
3	Input capture/out compare match A3	TIOCA3	I/O	TGR3A input capture input/output compare output/PWM output pin
	Input capture/out compare match B3	TIOCB3	I/O	TGR3B input capture input/output compare output/PWM output pin
	Input capture/out compare match C3	TIOCC3	I/O	TGR3C input capture input/output compare output/PWM output pin
	Input capture/out compare match D3	TIOCD3	I/O	TGR3D input capture input/output compare output/PWM output pin
4	Input capture/out compare match A4	TIOCA4	I/O	TGR4A input capture input/output compare output/PWM output pin
	Input capture/out compare match B4	TIOCB4	I/O	TGR4B input capture input/output compare output/PWM output pin
5	Input capture/out compare match A5	TIOCA5	I/O	TGR5A input capture input/output compare output/PWM output pin
	Input capture/out compare match B5	TIOCB5	I/O	TGR5B input capture input/output compare output/PWM output pin

10.1.4 Register Configuration

Table 10-3 summarizes the TPU registers.

Table 10-3 TPU Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address *1
0	Timer control register 0	TCR0	R/W	H'00	H'FF10
	Timer mode register 0	TMDR0	R/W	H'C0	H'FF11
	Timer I/O control register 0H	TIOR0H	R/W	H'00	H'FF12
	Timer I/O control register 0L	TIOR0L	R/W	H'00	H'FF13
	Timer interrupt enable register 0	TIER0	R/W	H'40	H'FF14
	Timer status register 0	TSR0	R/(W)*2	H'C0	H'FF15
	Timer counter 0	TCNT0	R/W	H'0000	H'FF16
	Timer general register 0A	TGR0A	R/W	H'FFFF	H'FF18
	Timer general register 0B	TGR0B	R/W	H'FFFF	H'FF1A
	Timer general register 0C	TGR0C	R/W	H'FFFF	H'FF1C
	Timer general register 0D	TGR0D	R/W	H'FFFF	H'FF1E
1	Timer control register 1	TCR1	R/W	H'00	H'FF20
	Timer mode register 1	TMDR1	R/W	H'C0	H'FF21
	Timer I/O control register 1	TIOR1	R/W	H'00	H'FF22
	Timer interrupt enable register 1	TIER1	R/W	H'40	H'FF24
	Timer status register 1	TSR1	R/(W) *2	H'C0	H'FF25
	Timer counter 1	TCNT1	R/W	H'0000	H'FF26
	Timer general register 1A	TGR1A	R/W	H'FFFF	H'FF28
	Timer general register 1B	TGR1B	R/W	H'FFFF	H'FF2A
2	Timer control register 2	TCR2	R/W	H'00	H'FF30
	Timer mode register 2	TMDR2	R/W	H'C0	H'FF31
	Timer I/O control register 2	TIOR2	R/W	H'00	H'FF32
	Timer interrupt enable register 2	TIER2	R/W	H'40	H'FF34
	Timer status register 2	TSR2	R/(W) *2	H'C0	H'FF35
	Timer counter 2	TCNT2	R/W	H'0000	H'FF36
	Timer general register 2A	TGR2A	R/W	H'FFFF	H'FF38
	Timer general register 2B	TGR2B	R/W	H'FFFF	H'FF3A

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
3	Timer control register 3	TCR3	R/W	H'00	H'FE80
	Timer mode register 3	TMDR3	R/W	H'C0	H'FE81
	Timer I/O control register 3H	TIOR3H	R/W	H'00	H'FE82
	Timer I/O control register 3L	TIOR3L	R/W	H'00	H'FE83
	Timer interrupt enable register 3	TIER3	R/W	H'40	H'FE84
	Timer status register 3	TSR3	R/(W)*[2]	H'C0	H'FE85
	Timer counter 3	TCNT3	R/W	H'0000	H'FE86
	Timer general register 3A	TGR3A	R/W	H'FFFF	H'FE88
	Timer general register 3B	TGR3B	R/W	H'FFFF	H'FE8A
	Timer general register 3C	TGR3C	R/W	H'FFFF	H'FE8C
	Timer general register 3D	TGR3D	R/W	H'FFFF	H'FE8E
4	Timer control register 4	TCR4	R/W	H'00	H'FE90
	Timer mode register 4	TMDR4	R/W	H'C0	H'FE91
	Timer I/O control register 4	TIOR4	R/W	H'00	H'FE92
	Timer interrupt enable register 4	TIER4	R/W	H'40	H'FE94
	Timer status register 4	TSR4	R/(W) *[2]	H'C0	H'FE95
	Timer counter 4	TCNT4	R/W	H'0000	H'FE96
	Timer general register 4A	TGR4A	R/W	H'FFFF	H'FE98
	Timer general register 4B	TGR4B	R/W	H'FFFF	H'FE9A
5	Timer control register 5	TCR5	R/W	H'00	H'FEA0
	Timer mode register 5	TMDR5	R/W	H'C0	H'FEA1
	Timer I/O control register 5	TIOR5	R/W	H'00	H'FEA2
	Timer interrupt enable register 5	TIER5	R/W	H'40	H'FEA4
	Timer status register 5	TSR5	R/(W) *[2]	H'C0	H'FEA5
	Timer counter 5	TCNT5	R/W	H'0000	H'FEA6
	Timer general register 5A	TGR5A	R/W	H'FFFF	H'FEA8
	Timer general register 5B	TGR5B	R/W	H'FFFF	H'FEAA
All	Timer start register	TSTR	R/W	H'00	H'FEB0
	Timer synchro register	TSYR	R/W	H'00	H'FEB1
	Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address.

*2 Can only be written with 0 for flag clearing.

10.2 Register Descriptions

10.2.1 Timer Control Register (TCR)

Channel 0: TCR0

Channel 3: TCR3

Bit	:	7	6	5	4	3	2	1	0
		CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 1: TCR1

Channel 2: TCR2

Channel 4: TCR4

Channel 5: TCR5

Bit	:	7	6	5	4	3	2	1	0
		—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The TCR registers are 8-bit registers that control the TCNT channels. The TPU has six TCR registers, one for each of channels 0 to 5. The TCR registers are initialized to H'00 by a reset, and in hardware standby mode.

TCR register settings should be made only when TCNT operation is stopped.

Bits 7 to 5—Counter Clear 2, 1, and 0 (CCLR2, CCLR1, CCLR0): These bits select the TCNT counter clearing source.

Channel	Bit 7 CCLR2	Bit 6 CCLR1	Bit 5 CCLR0	Description
0, 3	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation *1
	1	0	0	TCNT clearing disabled
			1	TCNT cleared by TGRC compare match/input capture *2
		1	0	TCNT cleared by TGRD compare match/input capture *2
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation *1

Channel	Bit 7 Reserved*3	Bit 6 CCLR1	Bit 5 CCLR0	Description
1, 2, 4, 5	0	0	0	TCNT clearing disabled (Initial value)
			1	TCNT cleared by TGRA compare match/input capture
		1	0	TCNT cleared by TGRB compare match/input capture
			1	TCNT cleared by counter clearing for another channel performing synchronous clearing/ synchronous operation *1

Notes: *1 Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

*2 When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

*3 Bit 7 is reserved in channels 1, 2, 4, and 5. It is always read as 0 and cannot be modified.

Bits 4 and 3—Clock Edge 1 and 0 (CKEG1, CKEG0): These bits select the input clock edge. When the input clock is counted using both edges, the input clock period is halved (e.g. ø/4 both edges = ø/2 rising edge). If phase counting mode is used on channels 1, 2, 4, and 5, this setting is ignored and the phase counting mode setting has priority.

Bit 4 CKEG1	Bit 3 CKEG0	Description	
0	0	Count at rising edge	(Initial value)
	1	Count at falling edge	
1	—	Count at both edges	

Note: Internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Bits 2 to 0—Time Prescaler 2, 1, and 0 (TPSC2 to TPSC0): These bits select the TCNT counter clock. The clock source can be selected independently for each channel. Table 10-4 shows the clock sources that can be set for each channel.

Table 10-4 TPU Clock Sources

	Internal Clock							External Clock				Overflow/ Underflow on Another
Channel	ø/1	ø/4	ø/16	ø/64	ø/256	ø/1024	ø/4096	TCLKA	TCLKB	TCLKC	TCLKD	Channel
0	○	○	○	○				○	○	○	○	
1	○	○	○	○	○			○	○			○
2	○	○	○	○		○		○	○	○		
3	○	○	○	○	○	○	○	○				
4	○	○	○	○		○		○		○		○
5	○	○	○	○	○			○		○	○	

Legend
○: Setting
Blank: No setting

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
0	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	External clock: counts on TCLKC pin input	
			1	External clock: counts on TCLKD pin input	

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
1	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	Internal clock: counts on ø/256	
			1	Counts on TCNT2 overflow/underflow	

Note: This setting is ignored when channel 1 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
2	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKB pin input	
		1	0	External clock: counts on TCLKC pin input	
			1	Internal clock: counts on ø/1024	

Note: This setting is ignored when channel 2 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
3	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	Internal clock: counts on ø/1024	
		1	0	Internal clock: counts on ø/256	
			1	Internal clock: counts on ø/4096	

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
4	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKC pin input	
		1	0	Internal clock: counts on ø/1024	
			1	Counts on TCNT5 overflow/underflow	

Note: This setting is ignored when channel 4 is in phase counting mode.

Channel	Bit 2 TPSC2	Bit 1 TPSC1	Bit 0 TPSC0	Description	
5	0	0	0	Internal clock: counts on ø/1	(Initial value)
			1	Internal clock: counts on ø/4	
		1	0	Internal clock: counts on ø/16	
			1	Internal clock: counts on ø/64	
	1	0	0	External clock: counts on TCLKA pin input	
			1	External clock: counts on TCLKC pin input	
		1	0	Internal clock: counts on ø/256	
			1	External clock: counts on TCLKD pin input	

Note: This setting is ignored when channel 5 is in phase counting mode.

10.2.2 Timer Mode Register (TMDR)

Channel 0: TMDR0

Channel 3: TMDR3

Bit	:	7	6	5	4	3	2	1	0
		—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Channel 1: TMDR1

Channel 2: TMDR2

Channel 4: TMDR4

Channel 5: TMDR5

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	MD3	MD2	MD1	MD0
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

The TMDR registers are 8-bit readable/writable registers that are used to set the operating mode for each channel. The TPU has six TMDR registers, one for each channel. The TMDR registers are initialized to H'C0 by a reset, and in hardware standby mode.

TMDR register settings should be made only when TCNT operation is stopped.

Bits 7 and 6—Reserved: These bits are always read as 1 and cannot be modified.

Bit 5—Buffer Operation B (BFB): Specifies whether TGRB is to operate in the normal way, or TGRB and TGRD are to be used together for buffer operation. When TGRD is used as a buffer register, TGRD input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRD, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5 BFB	Description	
0	TGRB operates normally	(Initial value)
1	TGRB and TGRD used together for buffer operation	

Bit 4—Buffer Operation A (BFA): Specifies whether TGRA is to operate in the normal way, or TGRA and TGRC are to be used together for buffer operation. When TGRC is used as a buffer register, TGRC input capture/output compare is not generated.

In channels 1, 2, 4, and 5, which have no TGRC, bit 4 is reserved. It is always read as 0 and cannot be modified.

Bit 4 BFA	Description	
0	TGRA operates normally	(Initial value)
1	TGRA and TGRC used together for buffer operation	

Bits 3 to 0—Modes 3 to 0 (MD3 to MD0): These bits are used to set the timer operating mode.

Bit 3 MD3*[1]	Bit 2 MD2*[2]	Bit 1 MD1	Bit 0 MD0	Description	
0	0	0	0	Normal operation	(Initial value)
			1	Reserved	
		1	0	PWM mode 1	
			1	PWM mode 2	
	1	0	0	Phase counting mode 1	
			1	Phase counting mode 2	
		1	0	Phase counting mode 3	
			1	Phase counting mode 4	
1	*	*	*	—	

*: Don't care

Notes: *1 MD3 is a reserved bit. In a write, it should always be written with 0.

*2 Phase counting mode cannot be set for channels 0 and 3. In this case, 0 should always be written to MD2.

10.2.3 Timer I/O Control Register (TIOR)

Channel 0: TIOR0H

Channel 1: TIOR1

Channel 2: TIOR2

Channel 3: TIOR3H

Channel 4: TIOR4

Channel 5: TIOR5

Bit	:	7	6	5	4	3	2	1	0
		IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel 0: TIOR0L

Channel 3: TIOR3L

Bit	:	7	6	5	4	3	2	1	0
		IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

The TIOR registers are 8-bit registers that control the TGR registers. The TPU has eight TIOR registers, two each for channels 0 and 3, and one each for channels 1, 2, 4, and 5. The TIOR registers are initialized to H'00 by a reset, and in hardware standby mode.

Care is required since TIOR is affected by the TMDR setting. The initial output specified by TIOR is valid when the counter is stopped (the CST bit in TSTR is cleared to 0). Note also that, in PWM mode 2, the output at the point at which the counter is cleared to 0 is specified.

Bits 7 to 4— I/O Control B3 to B0 (IOB3 to IOB0)
I/O Control D3 to D0 (IOD3 to IOD0):

Bits IOB3 to IOB0 specify the function of TGRB.
Bits IOD3 to IOD0 specify the function of TGRD.

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
0	0	0	0	0	TGR0B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count- up/count-down*[1]

*: Don't care

Note: *1 When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ø/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

Channel	Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	Description		
0	0	0	0	0	TGR0D is output compare register*2	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0D is input capture register*2	Capture input source is TIOCD0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/count-down*1

*: Don't care

Notes: *1 When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ø/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

*2 When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
1	0	0	0	0	TGR1B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at generation of TGR0C compare match/input capture

*: Don’t care

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
2	0	0	0	0	TGR2B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don’t care

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
3	0	0	0	0	TGR3B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3B is input capture register	Capture input source is TIOCB3 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down*[1]

*: Don't care

Note: *1 When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ø/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

Channel	Bit 7 IOD3	Bit 6 IOD2	Bit 5 IOD1	Bit 4 IOD0	Description		
3	0	0	0	0	TGR3D is output compare register*2	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3D is input capture register*2	Capture input source is TIOCD3 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down*1

*: Don't care

Notes: *1 When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ø/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

*2 When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
4	0	0	0	0	TGR4B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR4B is input capture register	Capture input source is TIOCB4 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR3C compare match/ input capture	Input capture at generation of TGR3C compare match/ input capture

*: Don’t care

Channel	Bit 7 IOB3	Bit 6 IOB2	Bit 5 IOB1	Bit 4 IOB0	Description		
5	0	0	0	0	TGR5B is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR5B is input capture register	Capture input source is TIOCB5 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don’t care

Bits 3 to 0— I/O Control A3 to A0 (IOA3 to IOA0)
I/O Control C3 to C0 (IOC3 to IOC0):

IOA3 to IOA0 specify the function of TGRA.
IOC3 to IOC0 specify the function of TGRC.

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
0	0	0	0	0	TGR0A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/ count clock	Input capture at TCNT1 count-up/count-down

*: Don’t care

Channel	Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	Description		
0	0	0	0	0	TGR0C is output compare register*1	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR0C is input capture register*1	Capture input source is TIOCC0 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/count-down

*: Don't care

Note: *1 When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
1	0	0	0	0	TGR1A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at generation of channel 0/TGR0A compare match/input capture

*: Don’t care

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
2	0	0	0	0	TGR2A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don’t care

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
3	0	0	0	0	TGR3A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3A is input capture register	Capture input source is TIOCA3 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down

*: Don't care

Channel	Bit 3 IOC3	Bit 2 IOC2	Bit 1 IOC1	Bit 0 IOC0	Description		
3	0	0	0	0	TGR3C is output compare register*1	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR3C is input capture register*1	Capture input source is TIOCC3 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/count-down

*: Don't care

Note: *1 When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
4	0	0	0	0	TGR4A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	0	0	0	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges
		1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at generation of TGR3A compare match/input capture

*: Don’t care

Channel	Bit 3 IOA3	Bit 2 IOA2	Bit 1 IOA1	Bit 0 IOA0	Description		
5	0	0	0	0	TGR5A is output compare register	Output disabled	(Initial value)
				1		Initial output is 0 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
		1	0	0		Output disabled	
				1		Initial output is 1 output	0 output at compare match
			1	0			1 output at compare match
				1			Toggle output at compare match
	1	*	0	0	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at rising edge
				1			Input capture at falling edge
			1	*			Input capture at both edges

*: Don’t care

10.2.4 Timer Interrupt Enable Register (TIER)

Channel 0: TIER0

Channel 3: TIER3

Bit	:	7	6	5	4	3	2	1	0
		TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
R/W	:	R/W	—	—	R/W	R/W	R/W	R/W	R/W

Channel 1: TIER1

Channel 2: TIER2

Channel 4: TIER4

Channel 5: TIER5

Bit	:	7	6	5	4	3	2	1	0
		TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	:	0	1	0	0	0	0	0	0
R/W	:	R/W	—	R/W	R/W	—	—	R/W	R/W

The TIER registers are 8-bit registers that control enabling or disabling of interrupt requests for each channel. The TPU has six TIER registers, one for each channel. The TIER registers are initialized to H'40 by a reset, and in hardware standby mode.

Bit 7—A/D Conversion Start Request Enable (TTGE): Enables or disables generation of A/D conversion start requests by TGRA input capture/compare match.

Bit 7 TTGE	Description	
0	A/D conversion start request generation disabled	(Initial value)
1	A/D conversion start request generation enabled	

Bit 6—Reserved: This bit is always read as 1 and cannot be modified.

Bit 5—Underflow Interrupt Enable (TCIEU): Enables or disables interrupt requests (TCIU) by the TCFU flag when the TCFU flag in TSR is set to 1 in channels 1, 2, 4, and 5.

In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5 TCIEU	Description	
0	Interrupt requests (TCIU) by TCFU disabled	(Initial value)
1	Interrupt requests (TCIU) by TCFU enabled	

Bit 4—Overflow Interrupt Enable (TCIEV): Enables or disables interrupt requests (TCIV) by the TCFV flag when the TCFV flag in TSR is set to 1.

Bit 4 TCIEV	Description	
0	Interrupt requests (TCIV) by TCFV disabled	(Initial value)
1	Interrupt requests (TCIV) by TCFV enabled	

Bit 3—TGR Interrupt Enable D (TGIED): Enables or disables interrupt requests (TGID) by the TGFD bit when the TGFD bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified.

Bit 3 TGIED	Description	
0	Interrupt requests (TGID) by TGFD bit disabled	(Initial value)
1	Interrupt requests (TGID) by TGFD bit enabled	

Bit 2—TGR Interrupt Enable C (TGIEC): Enables or disables interrupt requests (TGIC) by the TGFC bit when the TGFC bit in TSR is set to 1 in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.

Bit 2 TGIEC	Description	
0	Interrupt requests (TGIC) by TGFC bit disabled	(Initial value)
1	Interrupt requests (TGIC) by TGFC bit enabled	

Bit 1—TGR Interrupt Enable B (TGIEB): Enables or disables interrupt requests (TGIB) by the TGFB bit when the TGFB bit in TSR is set to 1.

Bit 1 TGIEB	Description	
0	Interrupt requests (TGIB) by TGFB bit disabled	(Initial value)
1	Interrupt requests (TGIB) by TGFB bit enabled	

Bit 0—TGR Interrupt Enable A (TGIEA): Enables or disables interrupt requests (TGIA) by the TGFA bit when the TGFA bit in TSR is set to 1.

Bit 0 TGIEA	Description	
0	Interrupt requests (TGIA) by TGFA bit disabled	(Initial value)
1	Interrupt requests (TGIA) by TGFA bit enabled	

10.2.5 Timer Status Register (TSR)

Channel 0: TSR0

Channel 3: TSR3

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Can only be written with 0 for flag clearing.

Channel 1: TSR1

Channel 2: TSR2

Channel 4: TSR4

Channel 5: TSR5

Bit	:	7	6	5	4	3	2	1	0
		TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	:	1	1	0	0	0	0	0	0
R/W	:	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Note: * Can only be written with 0 for flag clearing.

The TSR registers are 8-bit registers that indicate the status of each channel. The TPU has six TSR registers, one for each channel. The TSR registers are initialized to H'C0 by a reset, and in hardware standby mode.

Bit 7—Count Direction Flag (TCFD): Status flag that shows the direction in which TCNT counts in channels 1, 2, 4, and 5.

In channels 0 and 3, bit 7 is reserved. It is always read as 1 and cannot be modified.

Bit 7 TCFD	Description	
0	TCNT counts down	
1	TCNT counts up	(Initial value)

Bit 6—Reserved: This bit is always read as 1 and cannot be modified.

Bit 5—Underflow Flag (TCFU): Status flag that indicates that TCNT underflow has occurred when channels 1, 2, 4, and 5 are set to phase counting mode.

In channels 0 and 3, bit 5 is reserved. It is always read as 0 and cannot be modified.

Bit 5 TCFU	Description	
0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1	(Initial value)
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)	

Bit 4—Overflow Flag (TCFV): Status flag that indicates that TCNT overflow has occurred.

Bit 4 TCFV	Description	
0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1	(Initial value)
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)	

Bit 3—Input Capture/Output Compare Flag D (TGFD): Status flag that indicates the occurrence of TGRD input capture or compare match in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 3 is reserved. It is always read as 0 and cannot be modified.

Bit 3 TGFD	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register

Bit 2—Input Capture/Output Compare Flag C (TGFC): Status flag that indicates the occurrence of TGRC input capture or compare match in channels 0 and 3.

In channels 1, 2, 4, and 5, bit 2 is reserved. It is always read as 0 and cannot be modified.

Bit 2 TGFC	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIC interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register

Bit 1—Input Capture/Output Compare Flag B (TGFB): Status flag that indicates the occurrence of TGRB input capture or compare match.

Bit 1 TGFB	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Bit 0—Input Capture/Output Compare Flag A (TGFA): Status flag that indicates the occurrence of TGRA input capture or compare match.

Bit 0 TGFA	Description
0	[Clearing conditions] (Initial value) • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

10.2.6 Timer Counter (TCNT)

Channel 0: TCNT0 (up-counter)

Channel 1: TCNT1 (up/down-counter*)

Channel 2: TCNT2 (up/down-counter*)

Channel 3: TCNT3 (up-counter)

Channel 4: TCNT4 (up/down-counter*)

Channel 5: TCNT5 (up/down-counter*)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * These counters can be used as up/down-counters only in phase counting mode or when counting overflow/underflow on another channel. In other cases they function as up-counters.

The TCNT registers are 16-bit counters. The TPU has six TCNT counters, one for each channel.

The TCNT counters are initialized to H'0000 by a reset, and in hardware standby mode.

The TCNT counters cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

10.2.7 Timer General Register (TGR)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	:	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The TGR registers are 16-bit registers with a dual function as output compare and input capture registers. The TPU has 16 TGR registers, four each for channels 0 and 3 and two each for channels 1, 2, 4, and 5. TGRC and TGRD for channels 0 and 3 can also be designated for operation as buffer registers*. The TGR registers are initialized to H'FFFF by a reset, and in hardware standby mode.

The TGR registers cannot be accessed in 8-bit units; they must always be accessed as a 16-bit unit.

Note: * TGR buffer register combinations are TGRA—TGRC and TGRB—TGRD.

10.2.8 Timer Start Register (TSTR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	CST5	CST4	CST3	CST2	CST1	CST0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

TSTR is an 8-bit readable/writable register that selects operation/stoppage for channels 0 to 5. TSTR is initialized to H'00 by a reset, and in hardware standby mode. When setting the operating mode in TMDR or setting the count clock in TCR, first stop the TCNT counter.

Bits 7 and 6—Reserved: Should always be written with 0.

Bits 5 to 0—Counter Start 5 to 0 (CST5 to CST0): These bits select operation or stoppage for TCNT.

Bit n CSTn	Description	
0	TCNTn count operation is stopped	(Initial value)
1	TCNTn performs count operation	

n = 5 to 0

Note: If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value.

10.2.9 Timer Synchro Register (TSYR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	—	—	R/W	R/W	R/W	R/W	R/W	R/W

TSYR is an 8-bit readable/writable register that selects independent operation or synchronous operation for the channel 0 to 4 TCNT counters. A channel performs synchronous operation when the corresponding bit in TSYR is set to 1.

TSYR is initialized to H'00 by a reset, and in hardware standby mode.

Bits 7 and 6—Reserved: Should always be written with 0.

Bits 5 to 0—Timer Synchro 5 to 0 (SYNC5 to SYNC0): These bits select whether operation is independent of or synchronized with other channels.

When synchronous operation is selected, synchronous presetting of multiple channels*[1], and synchronous clearing through counter clearing on another channel*[2] are possible.

Notes: *1 To set synchronous operation, the SYNC bits for at least two channels must be set to 1.

*2 To set synchronous clearing, in addition to the SYNC bit , the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR.

Bit n SYNCn	Description
0	TCNTn operates independently (TCNT presetting/clearing is unrelated to other channels) (Initial value)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

n = 5 to 0

10.2.10 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRA is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPA5 bit in MSTPCRA is set to 1, TPU operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 23A.5, 23B.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 5—Module Stop (MSTPA5): Specifies the TPU module stop mode.

Bit 5 MSTPA5	Description	
0	TPU module stop mode cleared	
1	TPU module stop mode set	(Initial value)

10.3 Interface to Bus Master

10.3.1 16-Bit Registers

TCNT and TGR are 16-bit registers. As the data bus to the bus master is 16 bits wide, these registers can be read and written to in 16-bit units.

These registers cannot be read or written to in 8-bit units; 16-bit access must always be used.

An example of 16-bit register access operation is shown in figure 10-2.

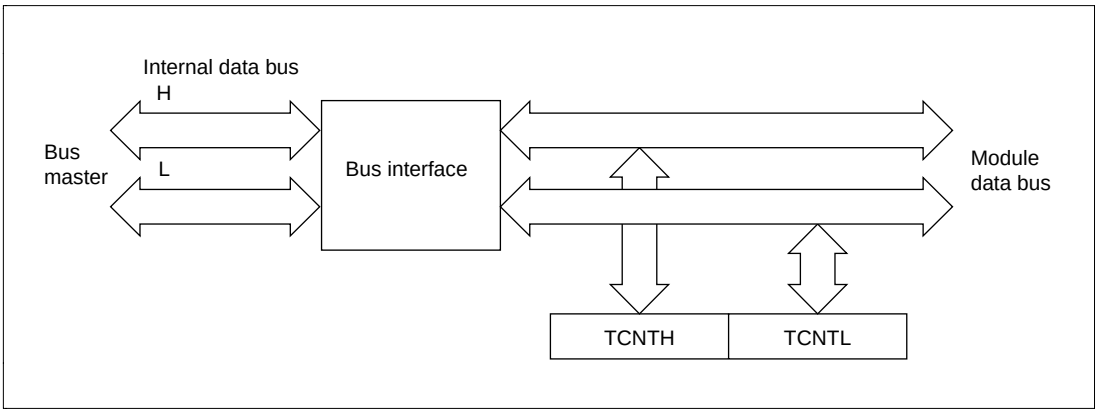


Figure 10-2 16-Bit Register Access Operation [Bus Master ↔ TCNT (16 Bits)]

10.3.2 8-Bit Registers

Registers other than TCNT and TGR are 8-bit. As the data bus to the CPU is 16 bits wide, these registers can be read and written to in 16-bit units. They can also be read and written to in 8-bit units.

Examples of 8-bit register access operation are shown in figures 10-3, 10-4, and 10-5.

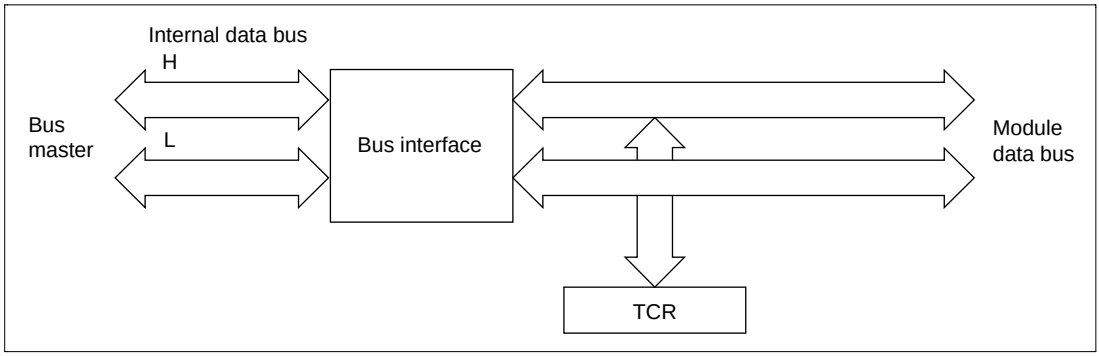


Figure 10-3 8-Bit Register Access Operation [Bus Master ↔ TCR (Upper 8 Bits)]

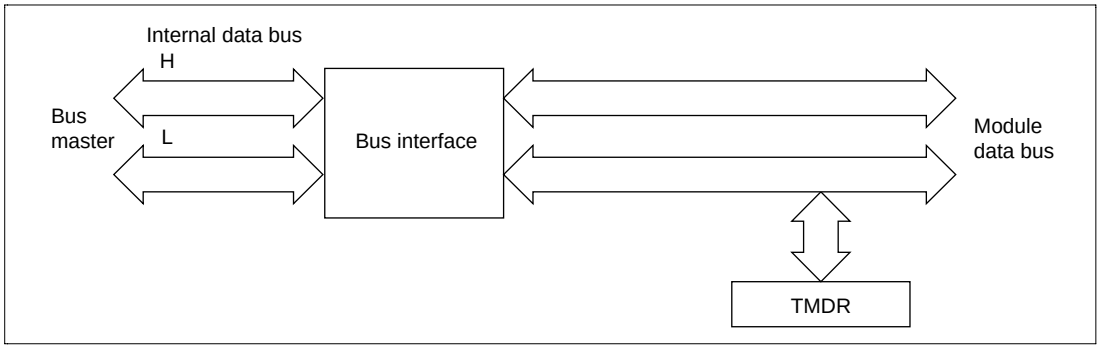


Figure 10-4 8-Bit Register Access Operation [Bus Master ↔ TMDR (Lower 8 Bits)]

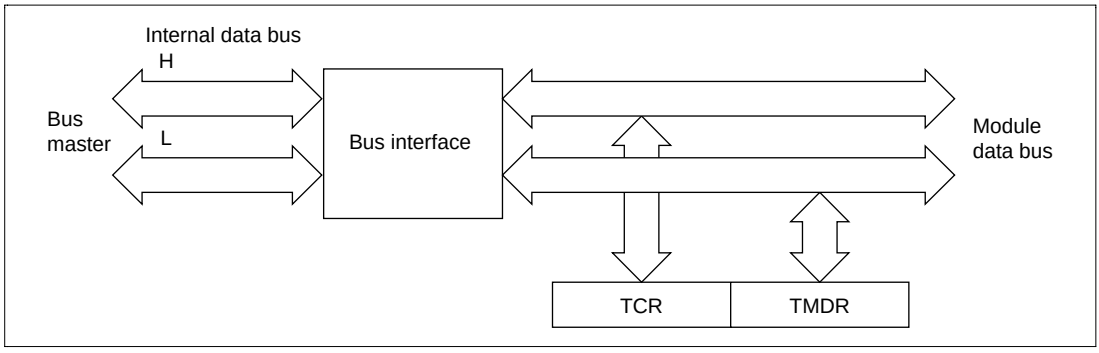


Figure 10-5 8-Bit Register Access Operation [Bus Master ↔ TCR and TMDR (16 Bits)]

10.4 Operation

10.4.1 Overview

Operation in each mode is outlined below.

Normal Operation: Each channel has a TCNT and TGR register. TCNT performs up-counting, and is also capable of free-running operation, synchronous counting, and external event counting.

Each TGR can be used as an input capture register or output compare register.

Synchronous Operation: When synchronous operation is designated for a channel, TCNT for that channel performs synchronous presetting. That is, when TCNT for a channel designated for synchronous operation is rewritten, the TCNT counters for the other channels are also rewritten at the same time. Synchronous clearing of the TCNT counters is also possible by setting the timer synchronization bits in TSYR for channels designated for synchronous operation.

Buffer Operation

- When TGR is an output compare register

  When a compare match occurs, the value in the buffer register for the relevant channel is transferred to TGR.

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transfer to TGR and the value previously held in TGR is transferred to the buffer register.

Cascaded Operation: The channel 1 counter (TCNT1), channel 2 counter (TCNT2), channel 4 counter (TCNT4), and channel 5 counter (TCNT5) can be connected together to operate as a 32-bit counter.

PWM Mode: In this mode, a PWM waveform is output. The output level can be set by means of TIOR. A PWM waveform with a duty of between 0% and 100% can be output, according to the setting of each TGR register.

Phase Counting Mode: In this mode, TCNT is incremented or decremented by detecting the phases of two clocks input from the external clock input pins in channels 1, 2, 4, and 5. When phase counting mode is set, the corresponding TCLK pin functions as the clock pin, and TCNT performs up- or down-counting.

This can be used for two-phase encoder pulse input.

10.4.2 Basic Functions

Counter Operation: When one of bits CST0 to CST5 is set to 1 in TSTR, the TCNT counter for the corresponding channel starts counting. TCNT can operate as a free-running counter, periodic counter, and so on.

- Example of count operation setting procedure

  Figure 10-6 shows an example of the count operation setting procedure.

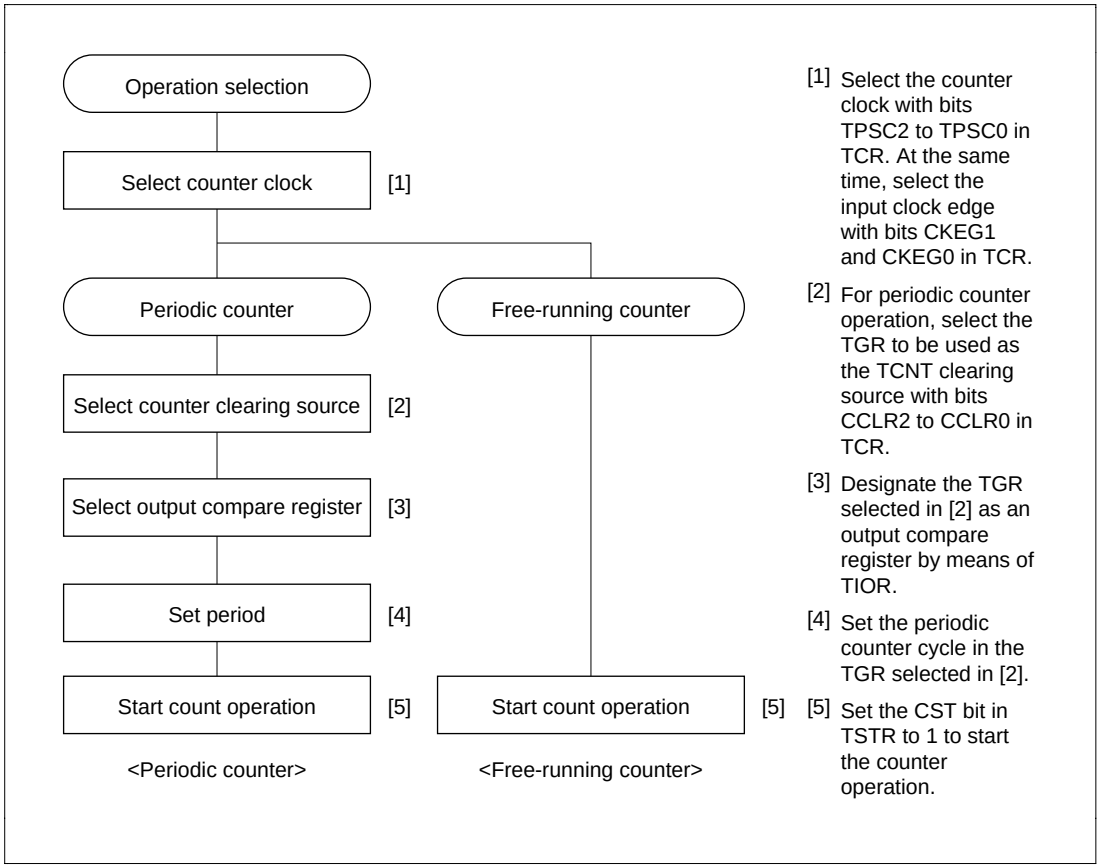


Figure 10-6 Example of Counter Operation Setting Procedure

- Free-running count operation and periodic count operation

  Immediately after a reset, the TPU's TCNT counters are all designated as free-running counters. When the relevant bit in TSTR is set to 1 the corresponding TCNT counter starts up-count operation as a free-running counter. When TCNT overflows (from H'FFFF to H'0000), the TCFV bit in TSR is set to 1. If the value of the corresponding TCIEV bit in TIER is 1 at this point, the TPU requests an interrupt. After overflow, TCNT starts counting up again from H'0000.

  Figure 10-7 illustrates free-running counter operation.

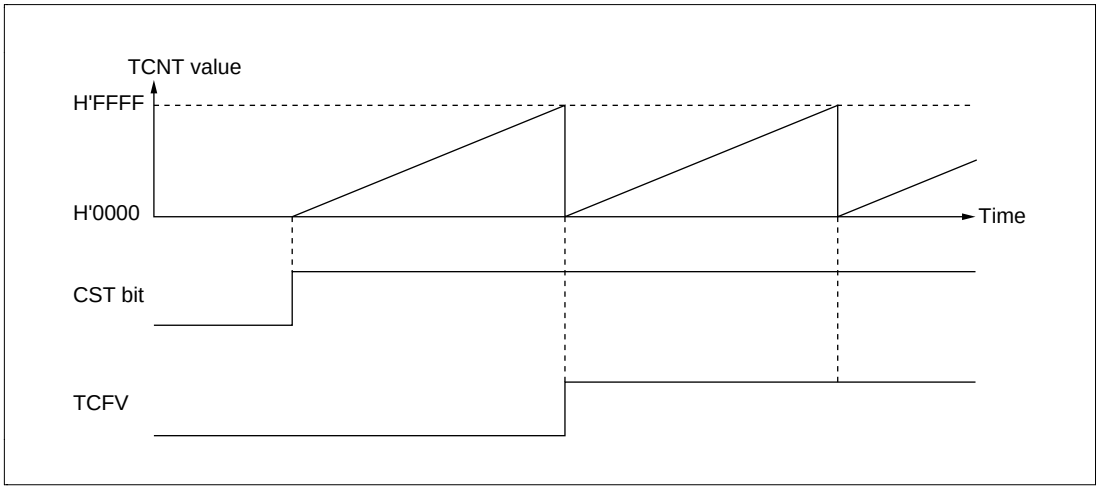


Figure 10-7 Free-Running Counter Operation

When compare match is selected as the TCNT clearing source, the TCNT counter for the relevant channel performs periodic count operation. The TGR register for setting the period is designated as an output compare register, and counter clearing by compare match is selected by means of bits CCLR2 to CCLR0 in TCR. After the settings have been made, TCNT starts up-count operation as periodic counter when the corresponding bit in TSTR is set to 1. When the count value matches the value in TGR, the TGF bit in TSR is set to 1 and TCNT is cleared to H'0000.

If the value of the corresponding TGIE bit in TIER is 1 at this point, the TPU requests an interrupt. After a compare match, TCNT starts counting up again from H'0000.

Figure 10-8 illustrates periodic counter operation.

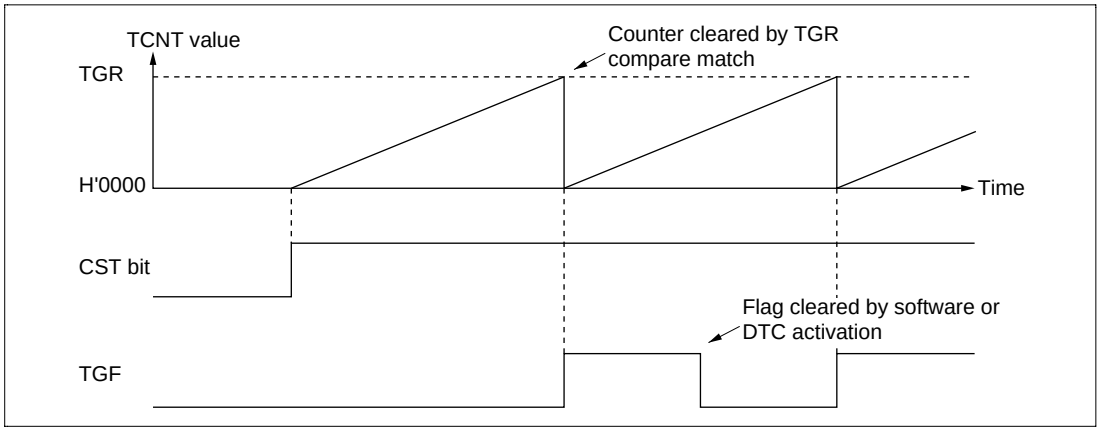


Figure 10-8 Periodic Counter Operation

Waveform Output by Compare Match: The TPU can perform 0, 1, or toggle output from the corresponding output pin using compare match.

- Example of setting procedure for waveform output by compare match
  Figure 10-9 shows an example of the setting procedure for waveform output by compare match

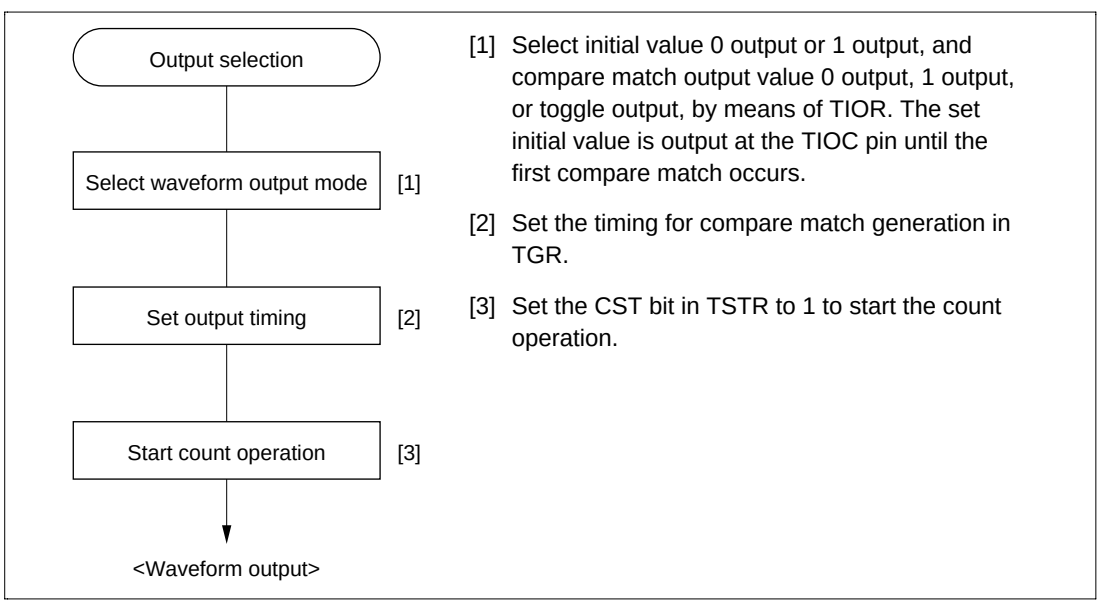


Figure 10-9 Example Of Setting Procedure For Waveform Output By Compare Match

- Examples of waveform output operation

  Figure 10-10 shows an example of 0 output/1 output.

  In this example TCNT has been designated as a free-running counter, and settings have been made so that 1 is output by compare match A, and 0 is output by compare match B. When the set level and the pin level coincide, the pin level does not change.

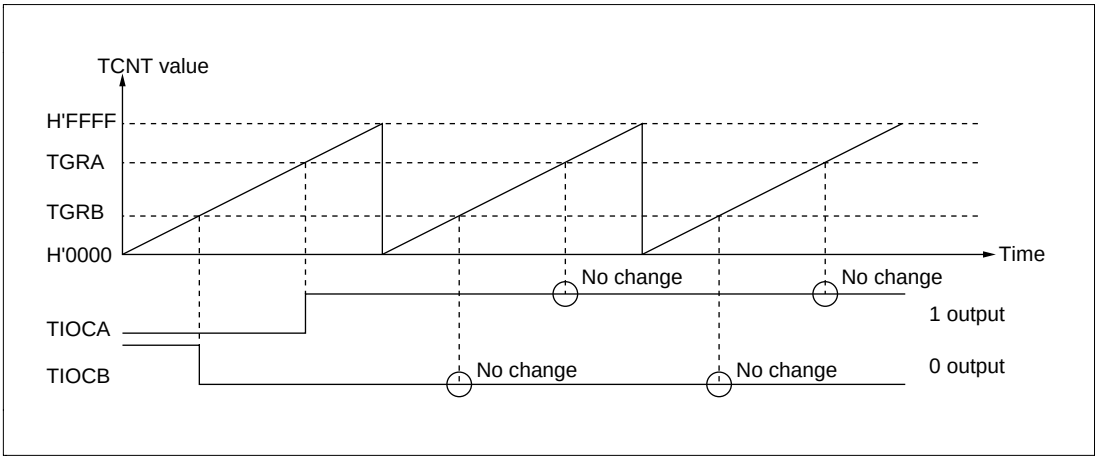


Figure 10-10 Example of 0 Output/1 Output Operation

Figure 10-11 shows an example of toggle output.

In this example TCNT has been designated as a periodic counter (with counter clearing performed by compare match B), and settings have been made so that output is toggled by both compare match A and compare match B.

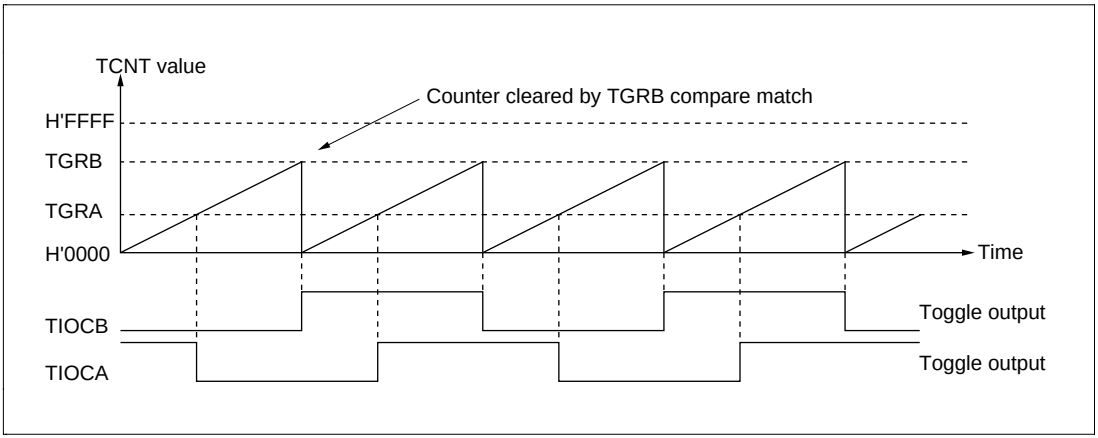


Figure 10-11 Example of Toggle Output Operation

Input Capture Function: The TCNT value can be transferred to TGR on detection of the TIOC pin input edge.

Rising edge, falling edge, or both edges can be selected as the detected edge. For channels 0, 1, 3, and 4, it is also possible to specify another channel's counter input clock or compare match signal as the input capture source.

Note: When another channel's counter input clock is used as the input capture input for channels 0 and 3, ø/1 should not be selected as the counter input clock used for input capture input. Input capture will not be generated if ø/1 is selected.

- Example of input capture operation setting procedure
  Figure 10-12 shows an example of the input capture operation setting procedure.

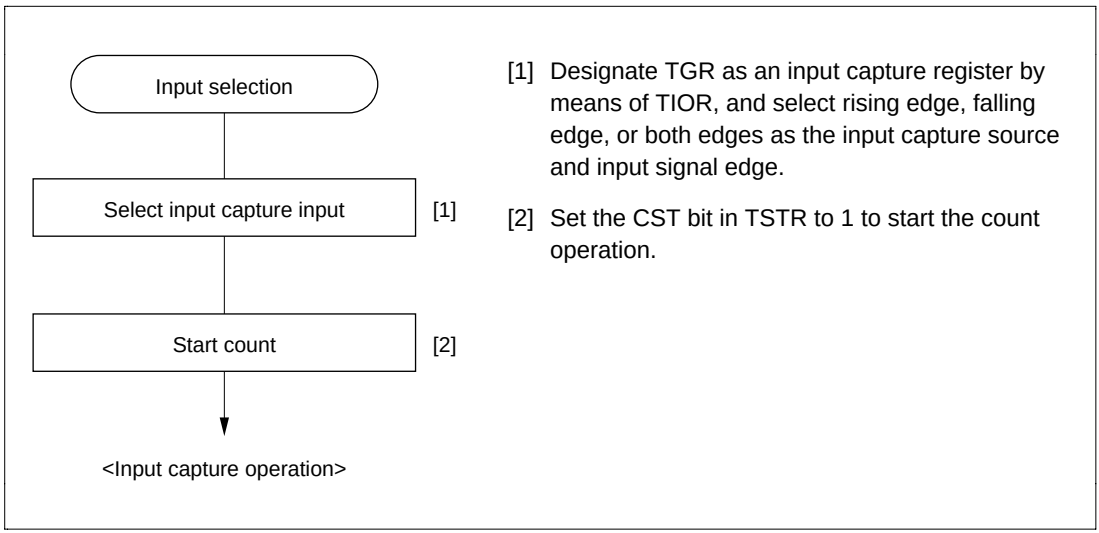


Figure 10-12 Example of Input Capture Operation Setting Procedure

- Example of input capture operation

  Figure 10-13 shows an example of input capture operation.

  In this example both rising and falling edges have been selected as the TIOCA pin input capture input edge, falling edge has been selected as the TIOCB pin input capture input edge, and counter clearing by TGRB input capture has been designated for TCNT.

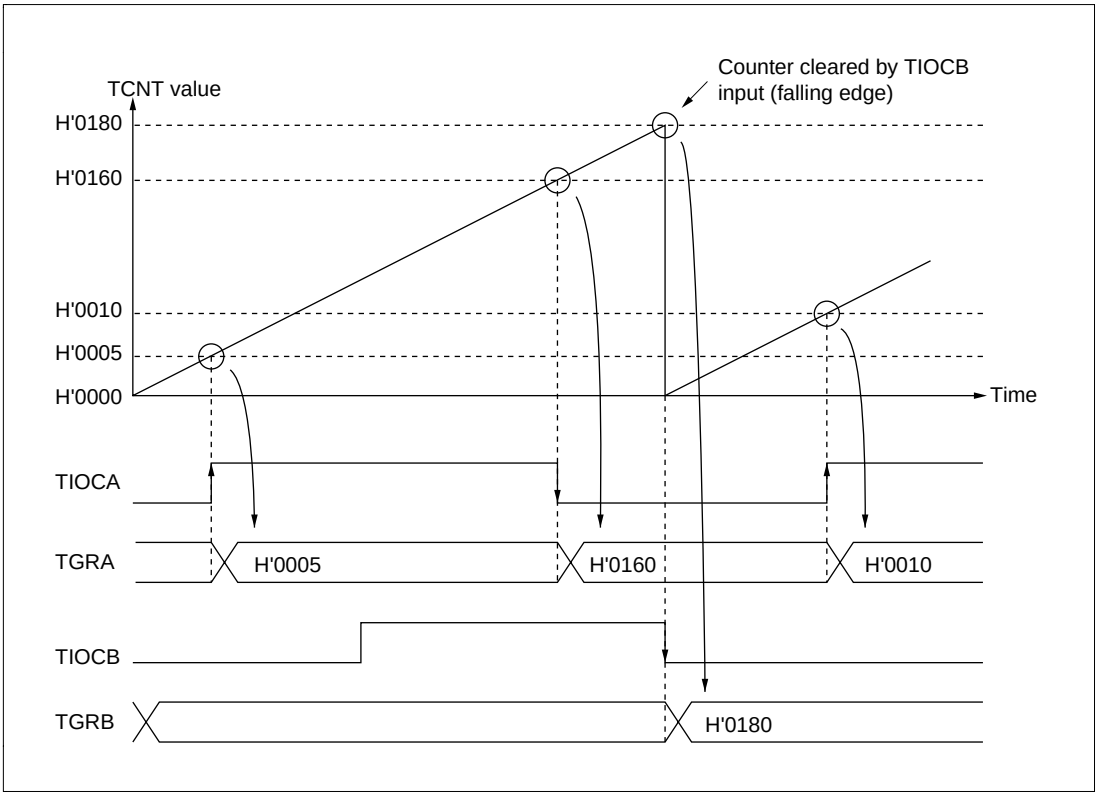


Figure 10-13 Example of Input Capture Operation

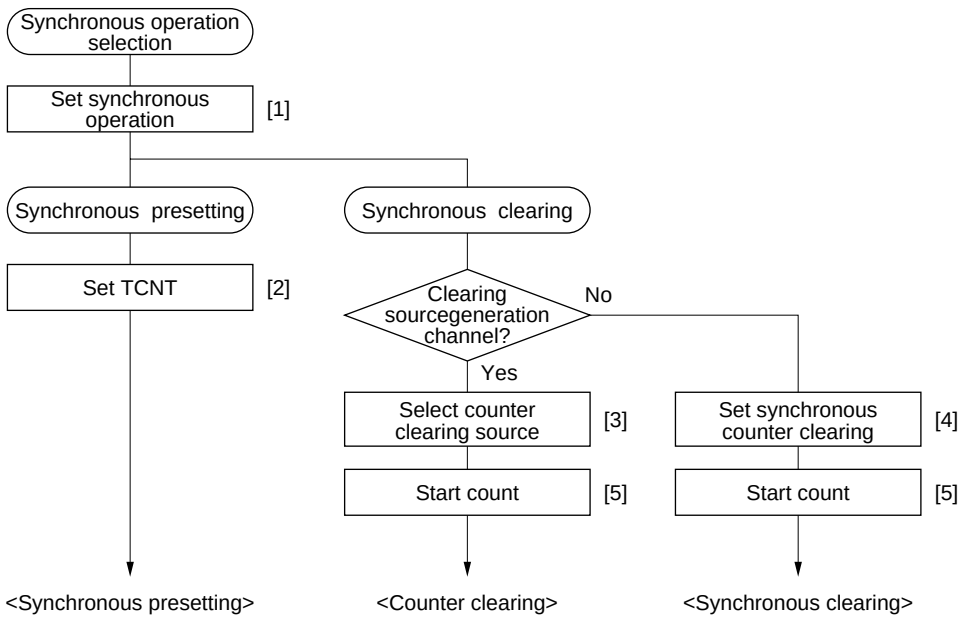
10.4.3 Synchronous Operation

In synchronous operation, the values in a number of TCNT counters can be rewritten simultaneously (synchronous presetting). Also, a number of TCNT counters can be cleared simultaneously by making the appropriate setting in TCR (synchronous clearing).

Synchronous operation enables TGR to be incremented with respect to a single time base.

Channels 0 to 5 can all be designated for synchronous operation.

Example of Synchronous Operation Setting Procedure: Figure 10-14 shows an example of the synchronous operation setting procedure.



[1] Set to 1 the SYNC bits in TSYR corresponding to the channels to be designated for synchronous operation.

[2] When the TCNT counter of any of the channels designated for synchronous operation is written to, the same value is simultaneously written to the other TCNT counters.

[3] Use bits CCLR2 to CCLR0 in TCR to specify TCNT clearing by input capture/output compare, etc.

[4] Use bits CCLR2 to CCLR0 in TCR to designate synchronous clearing for the counter clearing source.

[5] Set to 1 the CST bits in TSTR for the relevant channels, to start the count operation.

Figure 10-14 Example of Synchronous Operation Setting Procedure

Example of Synchronous Operation: Figure 10-15 shows an example of synchronous operation.

In this example, synchronous operation and PWM mode 1 have been designated for channels 0 to 2, TGR0B compare match has been set as the channel 0 counter clearing source, and synchronous clearing has been set for the channel 1 and 2 counter clearing source.

Three-phase PWM waveforms are output from pins TIOC0A, TIOC1A, and TIOC2A. At this time, synchronous presetting, and synchronous clearing by TGR0B compare match, is performed for channel 0 to 2 TCNT counters, and the data set in TGR0B is used as the PWM cycle.

For details of PWM modes, see section 10.4.6, PWM Modes.

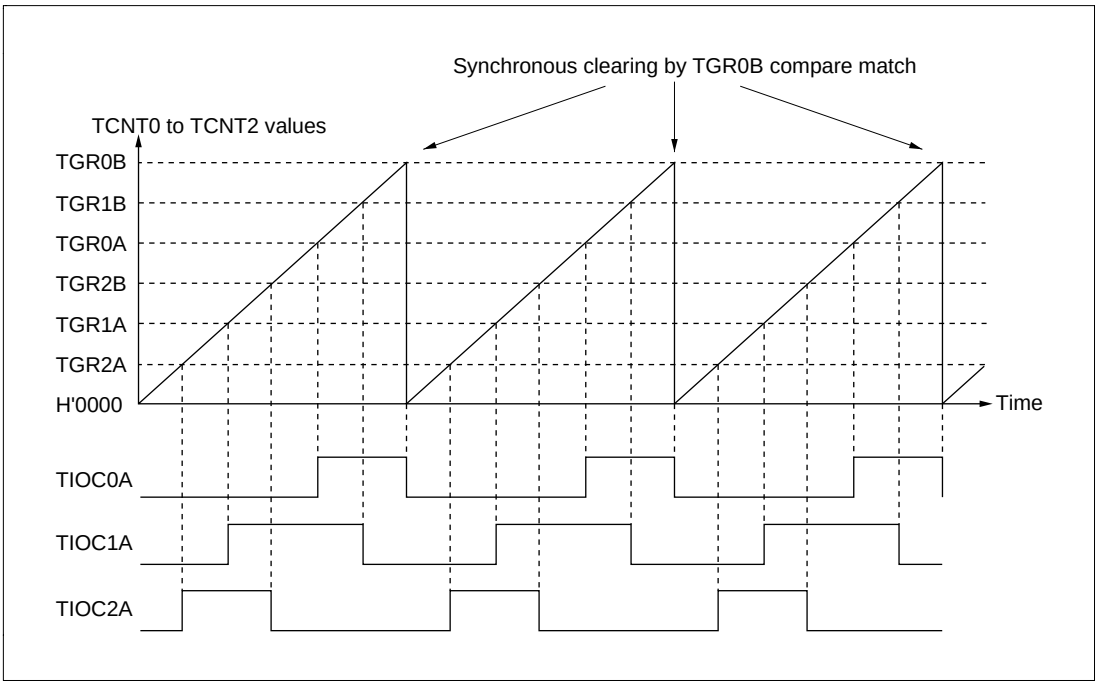


Figure 10-15 Example of Synchronous Operation

10.4.4 Buffer Operation

Buffer operation, provided for channels 0 and 3, enables TGRC and TGRD to be used as buffer registers.

Buffer operation differs depending on whether TGR has been designated as an input capture register or as a compare match register.

Table 10-5 shows the register combinations used in buffer operation.

Table 10-5 Register Combinations in Buffer Operation

Channel	Timer General Register	Buffer Register
0	TGR0A	TGR0C
	TGR0B	TGR0D
3	TGR3A	TGR3C
	TGR3B	TGR3D

- When TGR is an output compare register
  When a compare match occurs, the value in the buffer register for the corresponding channel is transferred to the timer general register.

  This operation is illustrated in figure 10-16.

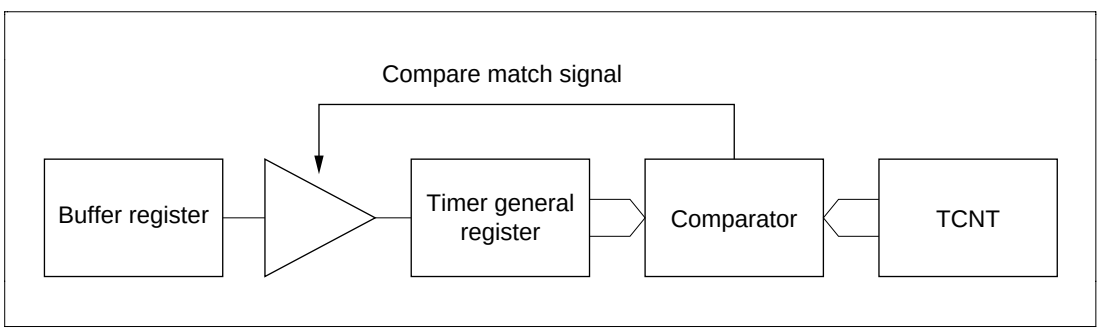


Figure 10-16 Compare Match Buffer Operation

- When TGR is an input capture register

  When input capture occurs, the value in TCNT is transferred to TGR and the value previously held in the timer general register is transferred to the buffer register.

  This operation is illustrated in figure 10-17.

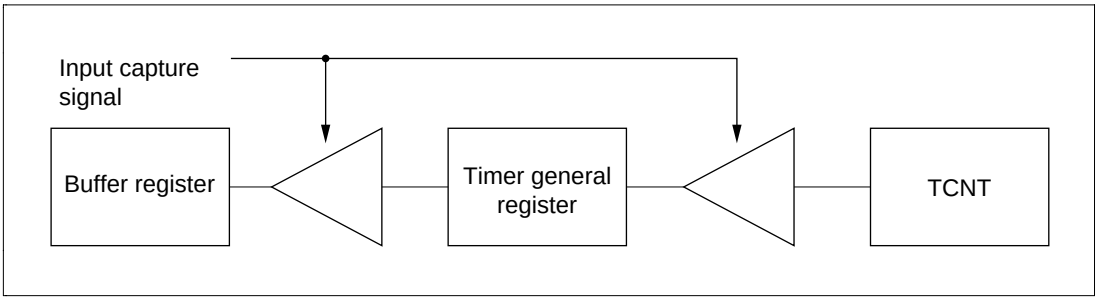


Figure 10-17 Input Capture Buffer Operation

Example of Buffer Operation Setting Procedure: Figure 10-18 shows an example of the buffer operation setting procedure.

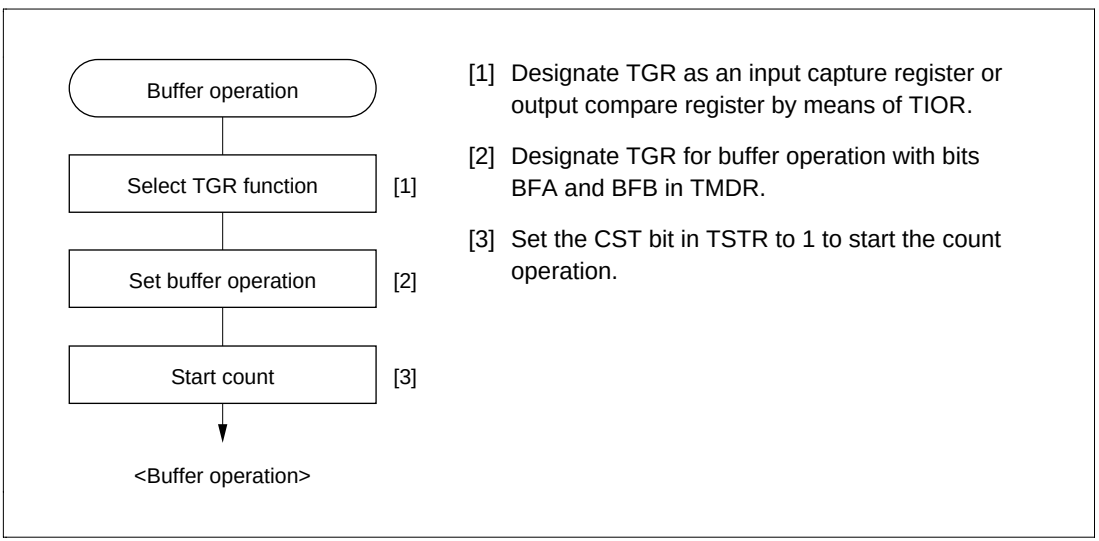


Figure 10-18 Example of Buffer Operation Setting Procedure

Examples of Buffer Operation

- When TGR is an output compare register

  Figure 10-19 shows an operation example in which PWM mode 1 has been designated for channel 0, and buffer operation has been designated for TGRA and TGRC. The settings used in this example are TCNT clearing by compare match B, 1 output at compare match A, and 0 output at compare match B.

  As buffer operation has been set, when compare match A occurs the output changes and the value in buffer register TGRC is simultaneously transferred to timer general register TGRA. This operation is repeated each time compare match A occurs.

  For details of PWM modes, see section 10.4.6, PWM Modes.

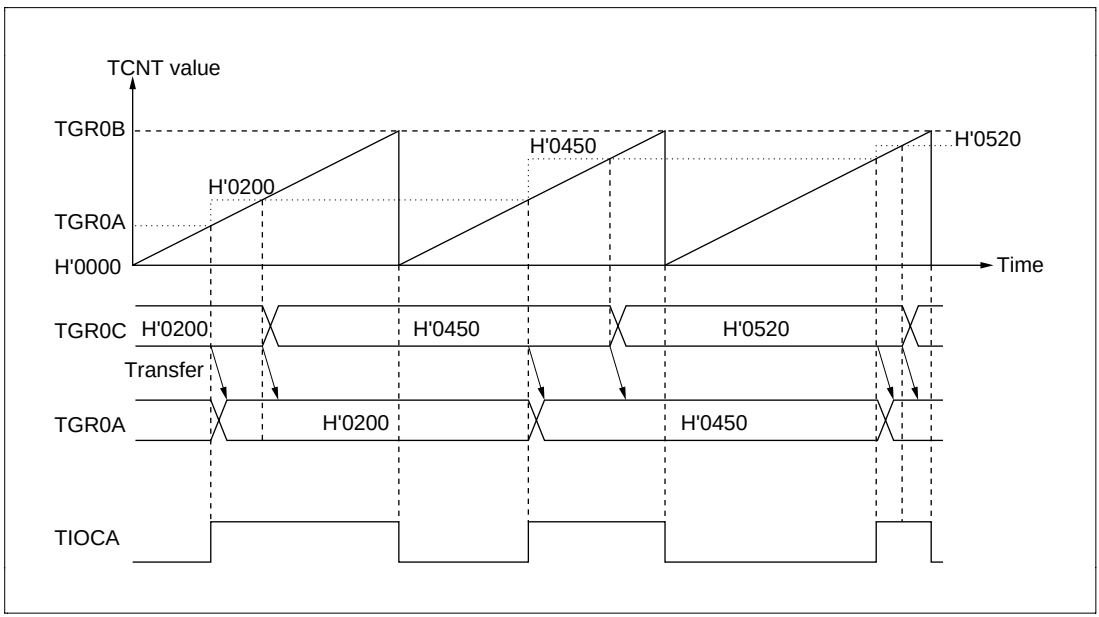


Figure 10-19 Example of Buffer Operation (1)

- When TGR is an input capture register

  Figure 10-20 shows an operation example in which TGRA has been designated as an input capture register, and buffer operation has been designated for TGRA and TGRC.

  Counter clearing by TGRA input capture has been set for TCNT, and both rising and falling edges have been selected as the TIOCA pin input capture input edge.

  As buffer operation has been set, when the TCNT value is stored in TGRA upon occurrence of input capture A, the value previously stored in TGRA is simultaneously transferred to TGRC.

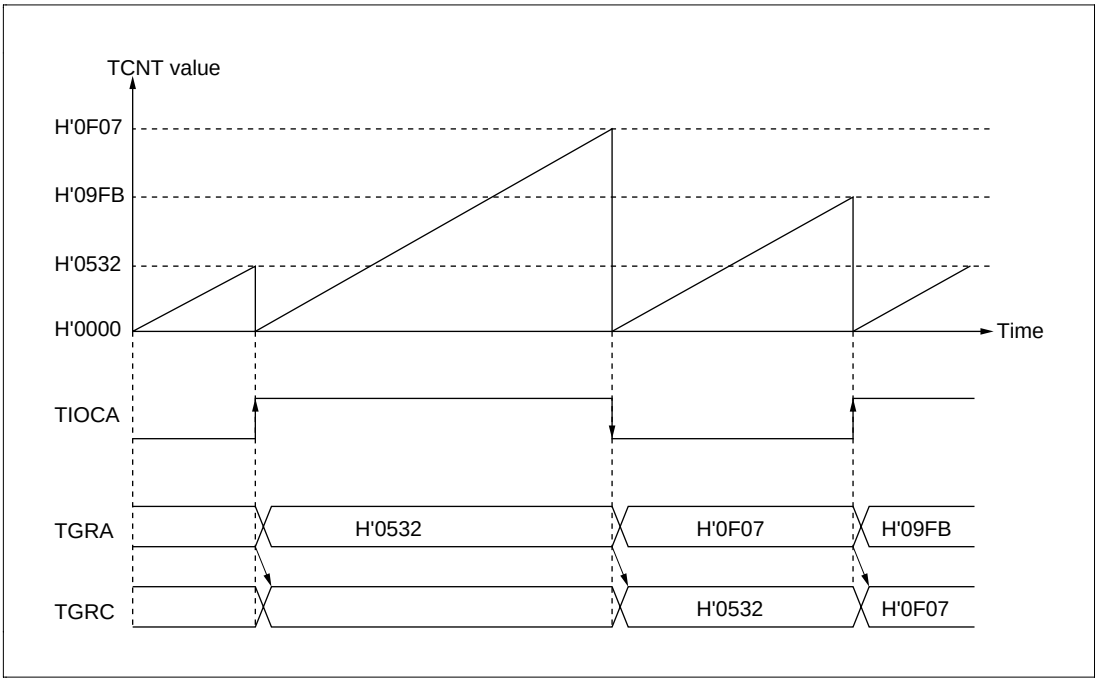


Figure 10-20 Example of Buffer Operation (2)

10.4.5 Cascaded Operation

In cascaded operation, two 16-bit counters for different channels are used together as a 32-bit counter.

This function works by counting the channel 1 (channel 4) counter clock upon overflow/underflow of TCNT2 (TCNT5) as set in bits TPSC2 to TPSC0 in TCR.

Underflow occurs only when the lower 16-bit TCNT is in phase-counting mode.

Table 10-6 shows the register combinations used in cascaded operation.

Note: When phase counting mode is set for channel 1 or 4, the counter clock setting is invalid and the counter operates independently in phase counting mode.

Table 10-6 Cascaded Combinations

Combination	Upper 16 Bits	Lower 16 Bits
Channels 1 and 2	TCNT1	TCNT2
Channels 4 and 5	TCNT4	TCNT5

Example of Cascaded Operation Setting Procedure: Figure 10-21 shows an example of the setting procedure for cascaded operation.

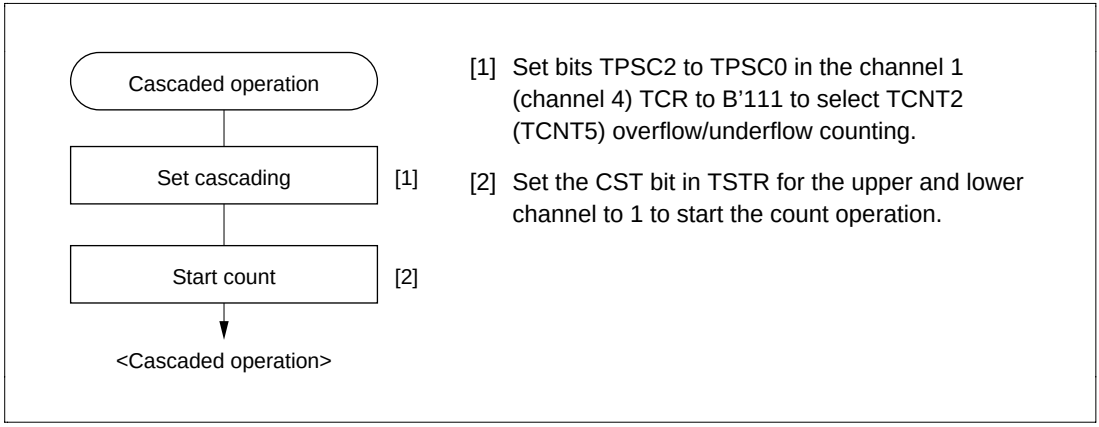


Figure 10-21 Cascaded Operation Setting Procedure

Examples of Cascaded Operation: Figure 10-22 illustrates the operation when counting upon TCNT2 overflow/underflow has been set for TCNT1, TGR1A and TGR2A have been designated as input capture registers, and TIOC pin rising edge has been selected.

When a rising edge is input to the TIOCA1 and TIOCA2 pins simultaneously, the upper 16 bits of the 32-bit data are transferred to TGR1A, and the lower 16 bits to TGR2A.

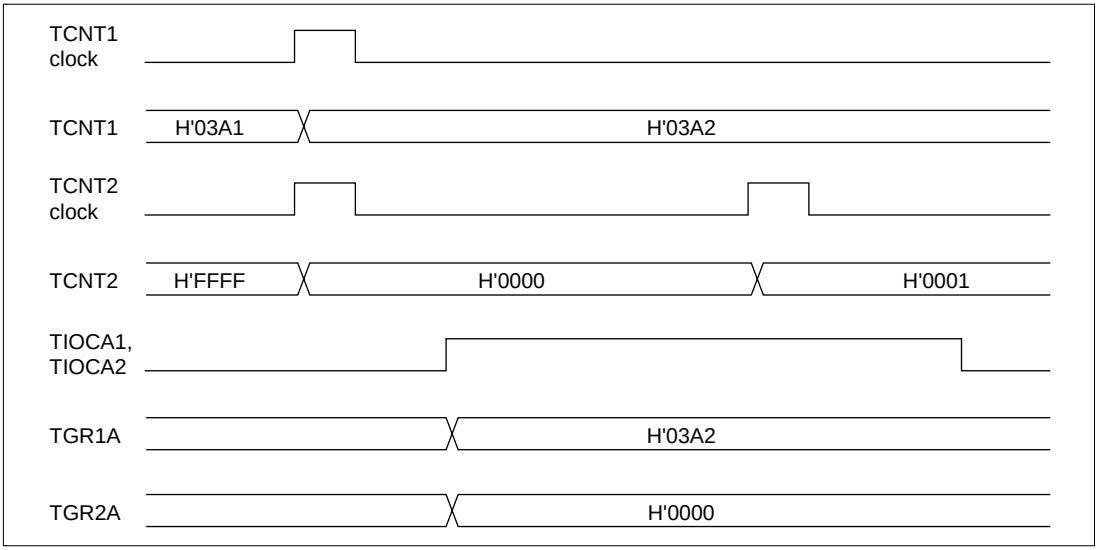


Figure 10-22 Example of Cascaded Operation (1)

Figure 10-23 illustrates the operation when counting upon TCNT2 overflow/underflow has been set for TCNT1, and phase counting mode has been designated for channel 2.

TCNT1 is incremented by TCNT2 overflow and decremented by TCNT2 underflow.

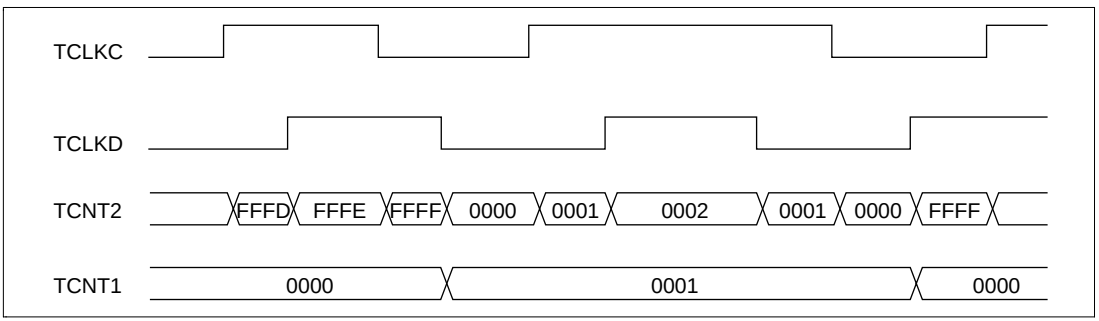


Figure 10-23 Example of Cascaded Operation (2)

10.4.6 PWM Modes

In PWM mode, PWM waveforms are output from the output pins. 0, 1, or toggle output can be selected as the output level in response to compare match of each TGR.

Designating TGR compare match as the counter clearing source enables the period to be set in that register. All channels can be designated for PWM mode independently. Synchronous operation is also possible.

There are two PWM modes, as described below.

- PWM mode 1

  PWM output is generated from the TIOCA and TIOCC pins by pairing TGRA with TGRB and TGRC with TGRD. The output specified by bits IOA3 to IOA0 and IOC3 to IOC0 in TIOR is output from the TIOCA and TIOCC pins at compare matches A and C, and the output specified by bits IOB3 to IOB0 and IOD3 to IOD0 in TIOR is output at compare matches B and D. The initial output value is the value set in TGRA or TGRC. If the set values of paired TGRs are identical, the output value does not change when a compare match occurs.

  In PWM mode 1, a maximum 8-phase PWM output is possible.

- PWM mode 2

  PWM output is generated using one TGR as the cycle register and the others as duty registers. The output specified in TIOR is performed by means of compare matches. Upon counter clearing by a synchronization register compare match, the output value of each pin is the initial value set in TIOR. If the set values of the cycle and duty registers are identical, the output value does not change when a compare match occurs.

  In PWM mode 2, a maximum 15-phase PWM output is possible by combined use with synchronous operation.

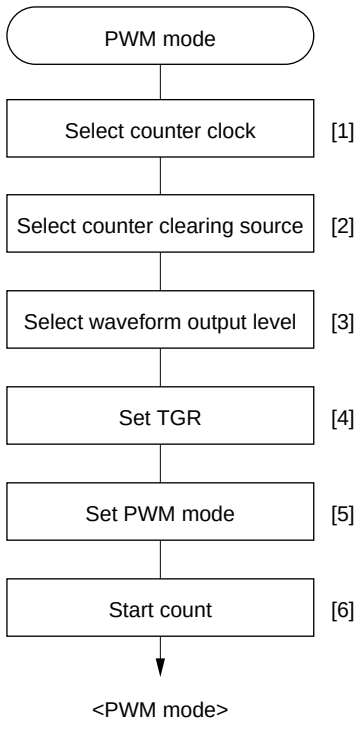
The correspondence between PWM output pins and registers is shown in table 10-7.

Table 10-7 PWM Output Registers and Output Pins

		Output Pins	
Channel	**Registers**	**PWM Mode 1**	**PWM Mode 2**
0	TGR0A	TIOCA0	TIOCA0
	TGR0B		TIOCB0
	TGR0C	TIOCC0	TIOCC0
	TGR0D		TIOCD0
1	TGR1A	TIOCA1	TIOCA1
	TGR1B		TIOCB1
2	TGR2A	TIOCA2	TIOCA2
	TGR2B		TIOCB2
3	TGR3A	TIOCA3	TIOCA3
	TGR3B		TIOCB3
	TGR3C	TIOCC3	TIOCC3
	TGR3D		TIOCD3
4	TGR4A	TIOCA4	TIOCA4
	TGR4B		TIOCB4
5	TGR5A	TIOCA5	TIOCA5
	TGR5B		TIOCB5

Note: In PWM mode 2, PWM output is not possible for the TGR register in which the period is set.

Example of PWM Mode Setting Procedure: Figure 10-24 shows an example of the PWM mode setting procedure.



[1] Select the counter clock with bits TPSC2 to TPSC0 in TCR. At the same time, select the input clock edge with bits CKEG1 and CKEG0 in TCR.

[2] Use bits CCLR2 to CCLR0 in TCR to select the TGR to be used as the TCNT clearing source.

[3] Use TIOR to designate the TGR as an output compare register, and select the initial value and output value.

[4] Set the cycle in the TGR selected in [2], and set the duty in the other the TGR.

[5] Select the PWM mode with bits MD3 to MD0 in TMDR.

[6] Set the CST bit in TSTR to 1 to start the count operation.

Figure 10-24 Example of PWM Mode Setting Procedure

Examples of PWM Mode Operation: Figure 10-25 shows an example of PWM mode 1 operation.

In this example, TGRA compare match is set as the TCNT clearing source, 0 is set for the TGRA initial output value and output value, and 1 is set as the TGRB output value.

In this case, the value set in TGRA is used as the period, and the values set in TGRB registers as the duty.

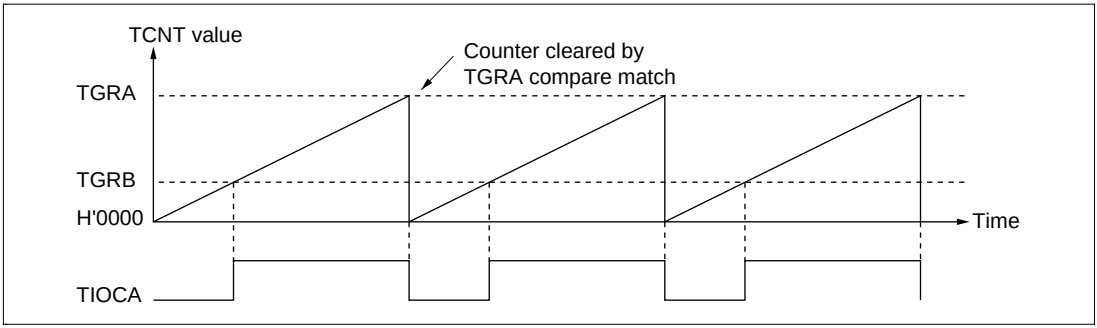


Figure 10-25 Example of PWM Mode Operation (1)

Figure 10-26 shows an example of PWM mode 2 operation.

In this example, synchronous operation is designated for channels 0 and 1, TGR1B compare match is set as the TCNT clearing source, and 0 is set for the initial output value and 1 for the output value of the other TGR registers (TGR0A to TGR0D, TGR1A), to output a 5-phase PWM waveform.

In this case, the value set in TGR1B is used as the cycle, and the values set in the other TGRs as the duty.

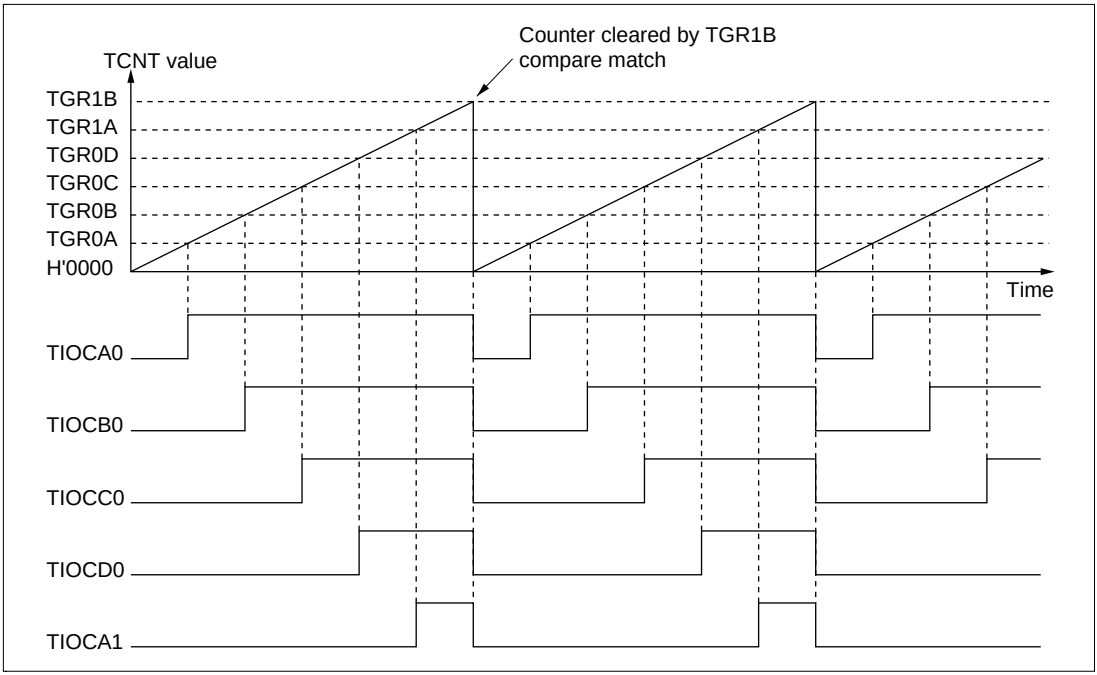


Figure 10-26 Example of PWM Mode Operation (2)

Figure 10-27 shows examples of PWM waveform output with 0% duty and 100% duty in PWM mode.

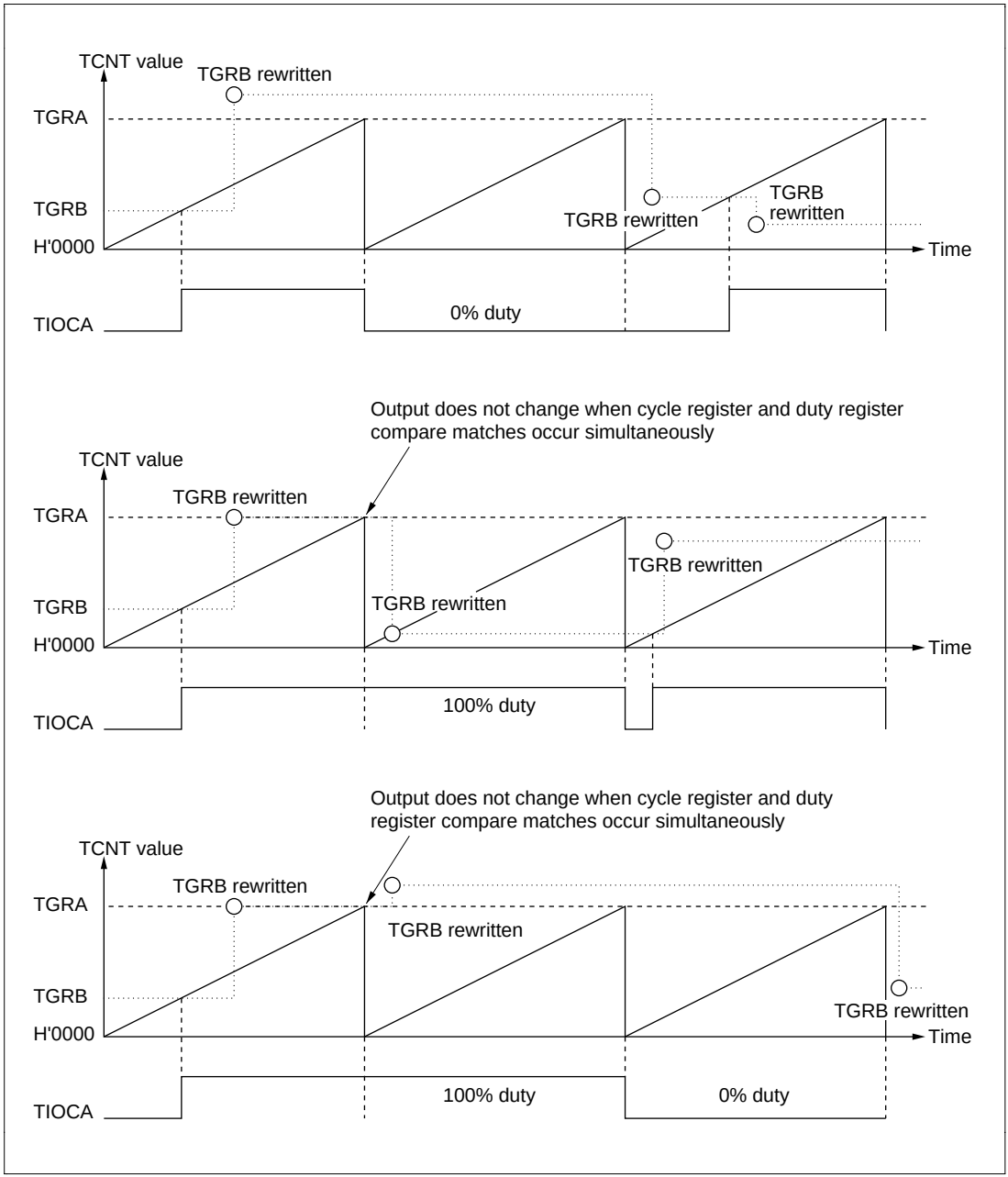


Figure 10-27 Example of PWM Mode Operation (3)

10.4.7 Phase Counting Mode

In phase counting mode, the phase difference between two external clock inputs is detected and TCNT is incremented/decremented accordingly. This mode can be set for channels 1, 2, 4, and 5.

When phase counting mode is set, an external clock is selected as the counter input clock and TCNT operates as an up/down-counter regardless of the setting of bits TPSC2 to TPSC0 and bits CKEG1 and CKEG0 in TCR. However, the functions of bits CCLR1 and CCLR0 in TCR, and of TIOR, TIER, and TGR are valid, and input capture/compare match and interrupt functions can be used.

When overflow occurs while TCNT is counting up, the TCFV flag in TSR is set; when underflow occurs while TCNT is counting down, the TCFU flag is set.

The TCFD bit in TSR is the count direction flag. Reading the TCFD flag provides an indication of whether TCNT is counting up or down.

Table 10-8 shows the correspondence between external clock pins and channels.

Table 10-8 Phase Counting Mode Clock Input Pins

	External Clock Pins	
Channels	**A-Phase**	**B-Phase**
When channel 1 or 5 is set to phase counting mode	TCLKA	TCLKB
When channel 2 or 4 is set to phase counting mode	TCLKC	TCLKD

Example of Phase Counting Mode Setting Procedure: Figure 10-28 shows an example of the phase counting mode setting procedure.

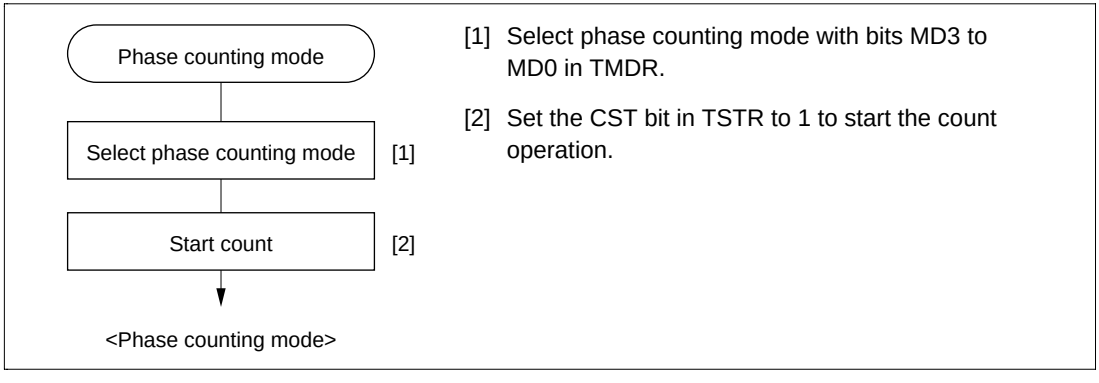


Figure 10-28 Example of Phase Counting Mode Setting Procedure

Examples of Phase Counting Mode Operation: In phase counting mode, TCNT counts up or down according to the phase difference between two external clocks. There are four modes, according to the count conditions.

- Phase counting mode 1

  Figure 10-29 shows an example of phase counting mode 1 operation, and table 10-9 summarizes the TCNT up/down-count conditions.

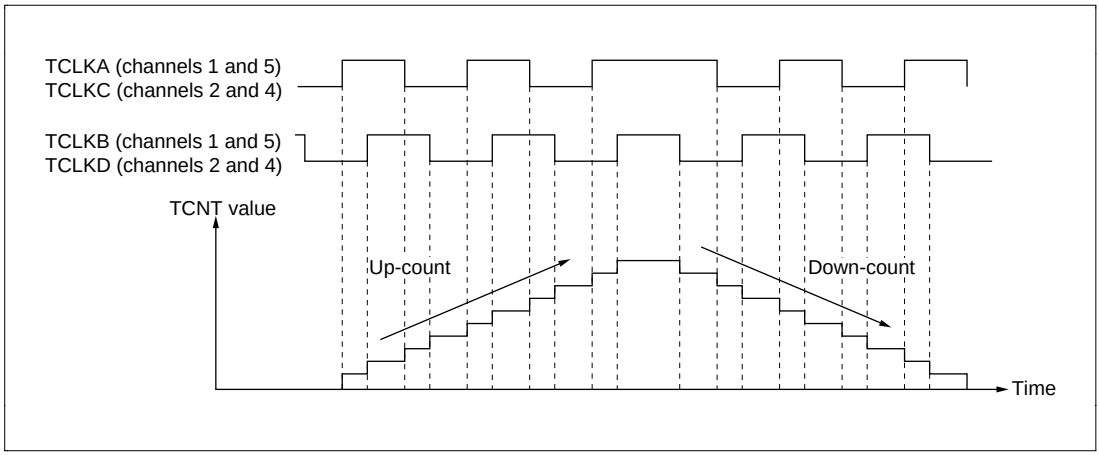


Figure 10-29 Example of Phase Counting Mode 1 Operation

Table 10-9 Up/Down-Count Conditions in Phase Counting Mode 1

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	
↓	Low level	

Legend

↑ : Rising edge

↓ : Falling edge

- Phase counting mode 2

  Figure 10-30 shows an example of phase counting mode 2 operation, and table 10-10 summarizes the TCNT up/down-count conditions.

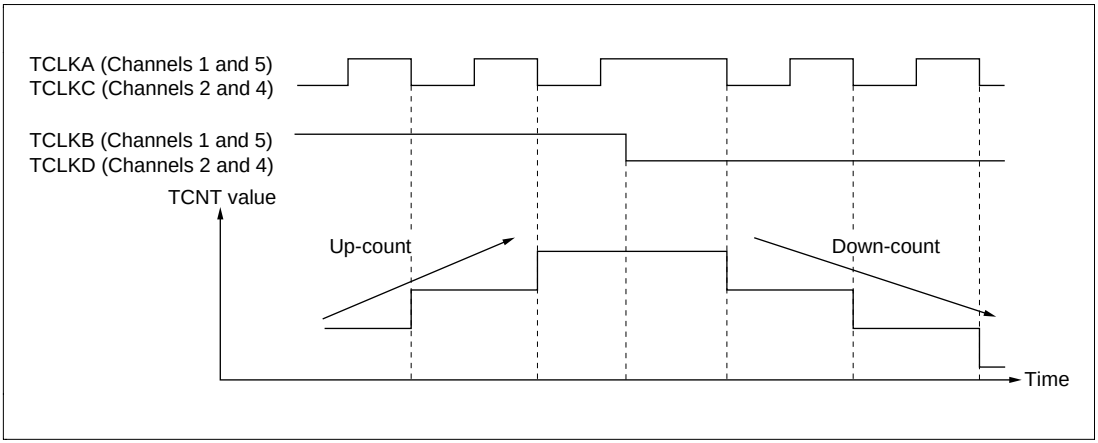


Figure 10-30 Example of Phase Counting Mode 2 Operation

Table 10-10 Up/Down-Count Conditions in Phase Counting Mode 2

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Don't care
Low level	↓	Don't care
↑	Low level	Don't care
↓	High level	Up-count
High level	↓	Don't care
Low level	↑	Don't care
↑	High level	Don't care
↓	Low level	Down-count

Legend

↑ : Rising edge

↓ : Falling edge

- Phase counting mode 3

  Figure 10-31 shows an example of phase counting mode 3 operation, and table 10-11 summarizes the TCNT up/down-count conditions.

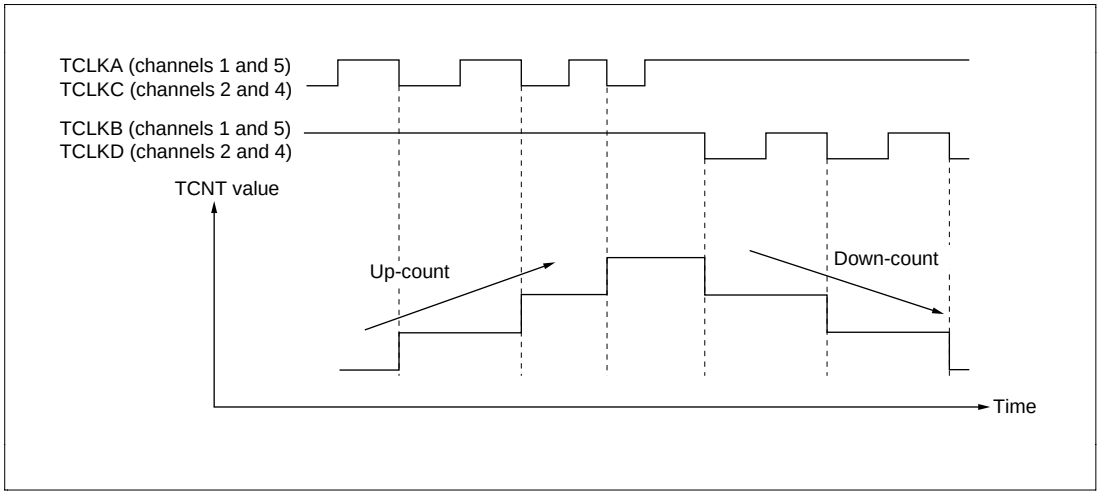


Figure 10-31 Example of Phase Counting Mode 3 Operation

Table 10-11 Up/Down-Count Conditions in Phase Counting Mode 3

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	⌐↑	Don't care
Low level	⌐↓	Don't care
⌐↑	Low level	Don't care
⌐↓	High level	Up-count
High level	⌐↓	Down-count
Low level	⌐↑	Don't care
⌐↑	High level	Don't care
⌐↓	Low level	Don't care

Legend

⌐↑ : Rising edge

⌐↓ : Falling edge

- Phase counting mode 4

  Figure 10-32 shows an example of phase counting mode 4 operation, and table 10-12 summarizes the TCNT up/down-count conditions.

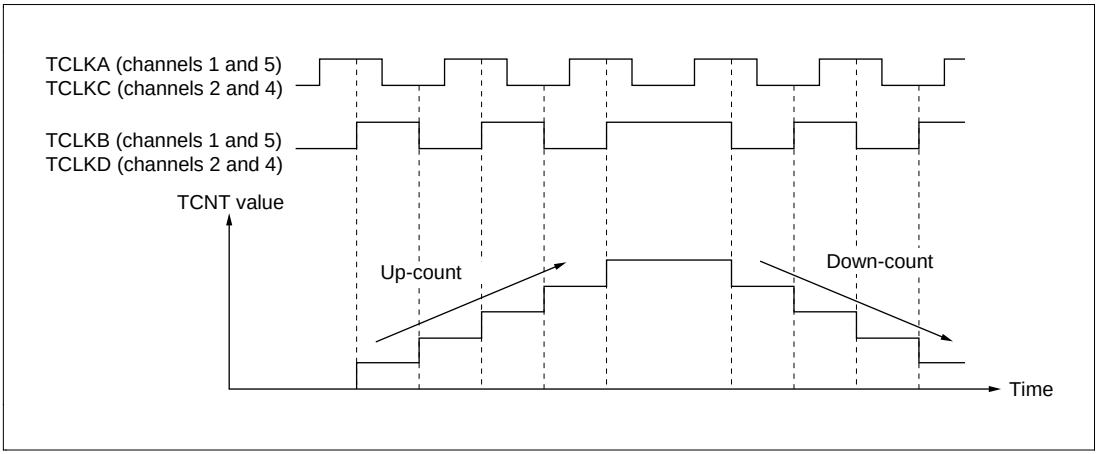


Figure 10-32 Example of Phase Counting Mode 4 Operation

Table 10-12 Up/Down-Count Conditions in Phase Counting Mode 4

TCLKA (Channels 1 and 5) TCLKC (Channels 2 and 4)	TCLKB (Channels 1 and 5) TCLKD (Channels 2 and 4)	Operation
High level	↑	Up-count
Low level	↓	
↑	Low level	Don't care
↓	High level	
High level	↓	Down-count
Low level	↑	
↑	High level	Don't care
↓	Low level	

Legend

↑ : Rising edge

↓ : Falling edge

Phase Counting Mode Application Example: Figure 10-33 shows an example in which phase counting mode is designated for channel 1, and channel 1 is coupled with channel 0 to input servo motor 2-phase encoder pulses in order to detect the position or speed.

Channel 1 is set to phase counting mode 1, and the encoder pulse A-phase and B-phase are input to TCLKA and TCLKB.

Channel 0 operates with TCNT counter clearing by TGR0C compare match; TGR0A and TGR0C are used for the compare match function, and are set with the speed control period and position control period. TGR0B is used for input capture, with TGR0B and TGR0D operating in buffer mode. The channel 1 counter input clock is designated as the TGR0B input capture source, and detection of the pulse width of 2-phase encoder 4-multiplication pulses is performed.

TGR1A and TGR1B for channel 1 are designated for input capture, channel 0 TGR0A and TGR0C compare matches are selected as the input capture source, and store the up/down-counter values for the control periods.

This procedure enables accurate position/speed detection to be achieved.

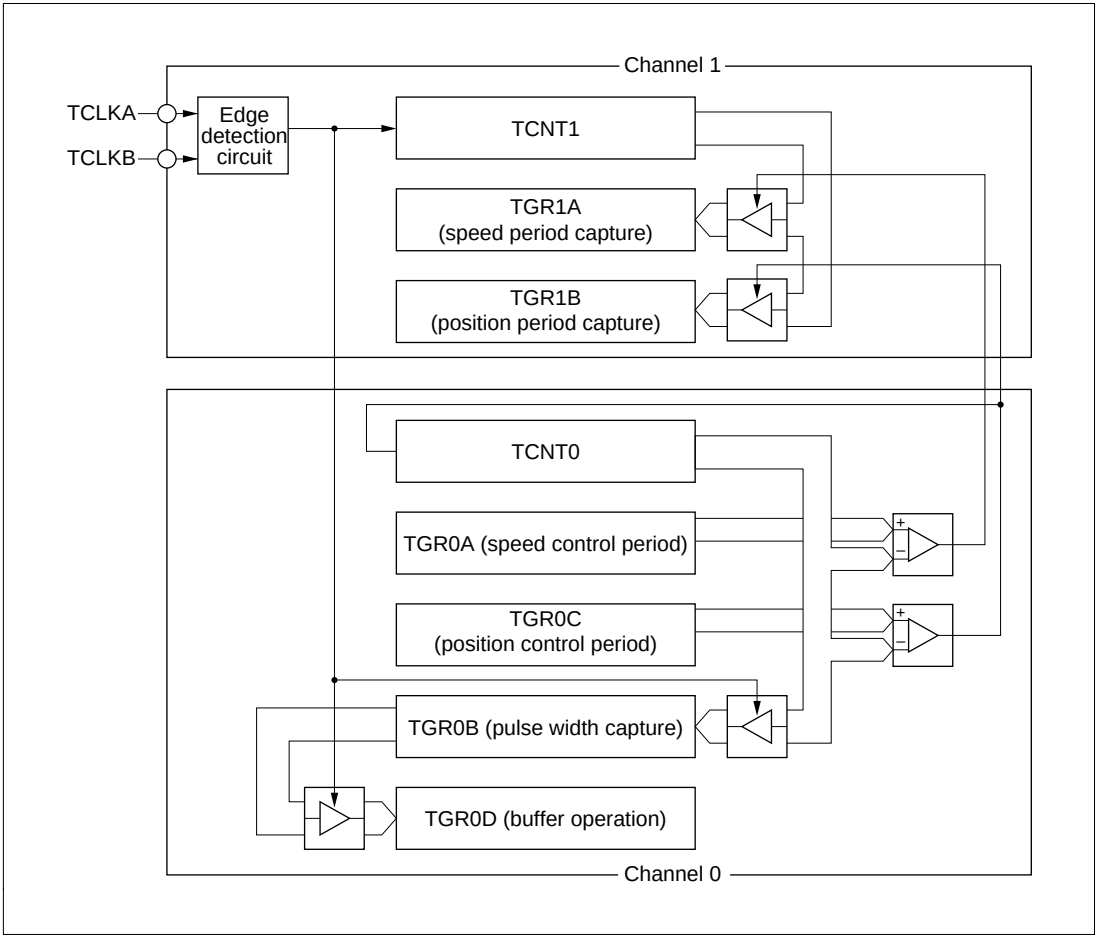


Figure 10-33 Phase Counting Mode Application Example

10.5 Interrupts

10.5.1 Interrupt Sources and Priorities

There are three kinds of TPU interrupt source: TGR input capture/compare match, TCNT overflow, and TCNT underflow. Each interrupt source has its own status flag and enable/disabled bit, allowing generation of interrupt request signals to be enabled or disabled individually.

When an interrupt request is generated, the corresponding status flag in TSR is set to 1. If the corresponding enable/disable bit in TIER is set to 1 at this time, an interrupt is requested. The interrupt request is cleared by clearing the status flag to 0.

Relative channel priorities can be changed by the interrupt controller, but the priority order within a channel is fixed. For details, see section 5, Interrupt Controller.

Table 10-13 lists the TPU interrupt sources.

Table 10-13 TPU Interrupts

Channel	Interrupt Source	Description	DTC Activation	Priority
0	TGI0A	TGR0A input capture/compare match	Possible	High
	TGI0B	TGR0B input capture/compare match	Possible	
	TGI0C	TGR0C input capture/compare match	Possible	
	TGI0D	TGR0D input capture/compare match	Possible	
	TCI0V	TCNT0 overflow	Not possible	
1	TGI1A	TGR1A input capture/compare match	Possible	
	TGI1B	TGR1B input capture/compare match	Possible	
	TCI1V	TCNT1 overflow	Not possible	
	TCI1U	TCNT1 underflow	Not possible	
2	TGI2A	TGR2A input capture/compare match	Possible	
	TGI2B	TGR2B input capture/compare match	Possible	
	TCI2V	TCNT2 overflow	Not possible	
	TCI2U	TCNT2 underflow	Not possible	
3	TGI3A	TGR3A input capture/compare match	Possible	
	TGI3B	TGR3B input capture/compare match	Possible	
	TGI3C	TGR3C input capture/compare match	Possible	
	TGI3D	TGR3D input capture/compare match	Possible	
	TCI3V	TCNT3 overflow	Not possible	
4	TGI4A	TGR4A input capture/compare match	Possible	
	TGI4B	TGR4B input capture/compare match	Possible	
	TCI4V	TCNT4 overflow	Not possible	
	TCI4U	TCNT4 underflow	Not possible	
5	TGI5A	TGR5A input capture/compare match	Possible	
	TGI5B	TGR5B input capture/compare match	Possible	
	TCI5V	TCNT5 overflow	Not possible	
	TCI5U	TCNT5 underflow	Not possible	Low

Note: This table shows the initial state immediately after a reset. The relative channel priorities can be changed by the interrupt controller.

Input Capture/Compare Match Interrupt: An interrupt is requested if the TGIE bit in TIER is set to 1 when the TGF flag in TSR is set to 1 by the occurrence of a TGR input capture/compare match on a particular channel. The interrupt request is cleared by clearing the TGF flag to 0. The TPU has 16 input capture/compare match interrupts, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

Overflow Interrupt: An interrupt is requested if the TCIEV bit in TIER is set to 1 when the TCFV flag in TSR is set to 1 by the occurrence of TCNT overflow on a channel. The interrupt request is cleared by clearing the TCFV flag to 0. The TPU has six overflow interrupts, one for each channel.

Underflow Interrupt: An interrupt is requested if the TCIEU bit in TIER is set to 1 when the TCFU flag in TSR is set to 1 by the occurrence of TCNT underflow on a channel. The interrupt request is cleared by clearing the TCFU flag to 0. The TPU has four underflow interrupts, one each for channels 1, 2, 4, and 5.

10.5.2 DTC Activation

DTC Activation: The DTC can be activated by the TGR input capture/compare match interrupt for a channel. For details, see section 9, Data Transfer Controller.

A total of 16 TPU input capture/compare match interrupts can be used as DTC activation sources, four each for channels 0 and 3, and two each for channels 1, 2, 4, and 5.

10.5.3 A/D Converter Activation

The A/D converter can be activated by the TGRA input capture/compare match for a channel.

If the TTGE bit in TIER is set to 1 when the TGFA flag in TSR is set to 1 by the occurrence of a TGRA input capture/compare match on a particular channel, a request to start A/D conversion is sent to the A/D converter. If the TPU conversion start trigger has been selected on the A/D converter side at this time, A/D conversion is started.

In the TPU, a total of six TGRA input capture/compare match interrupts can be used as A/D converter conversion start sources, one for each channel.

10.6 Operation Timing

10.6.1 Input/Output Timing

TCNT Count Timing: Figure 10-34 shows TCNT count timing in internal clock operation, and figure 10-35 shows TCNT count timing in external clock operation.

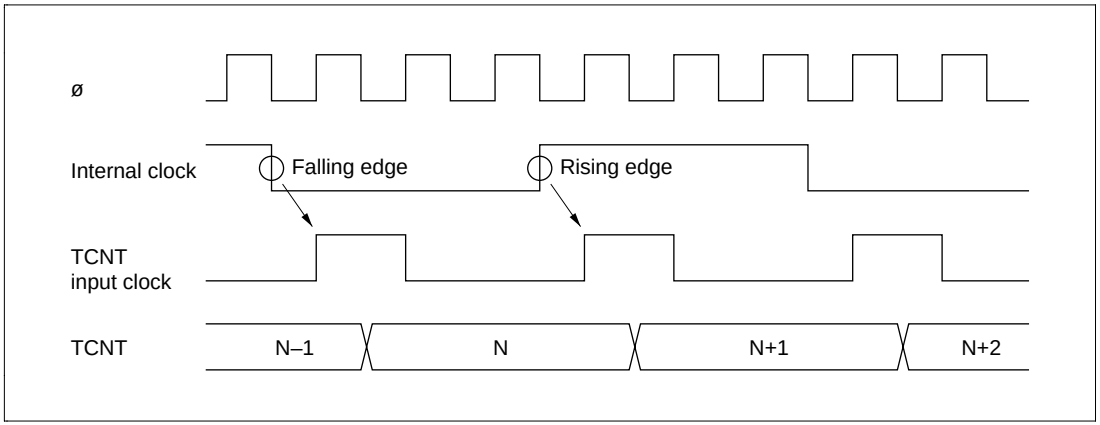


Figure 10-34 Count Timing in Internal Clock Operation

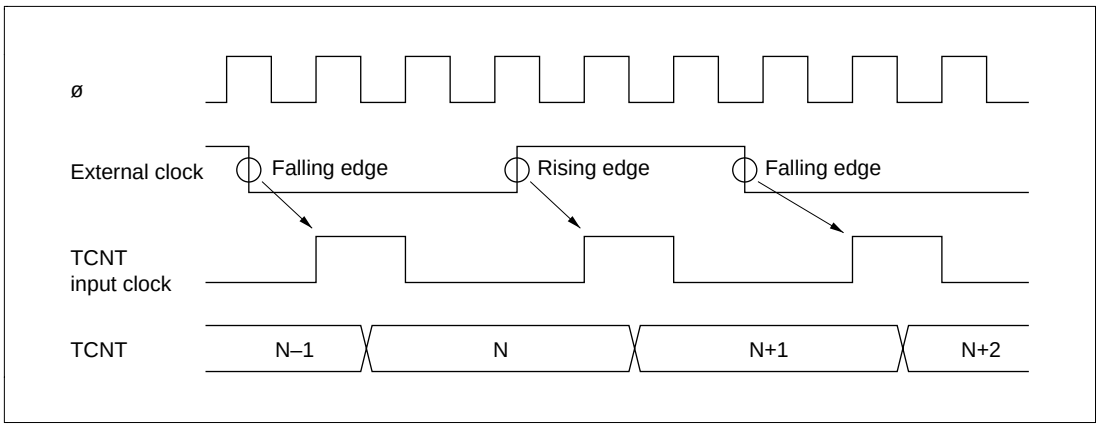


Figure 10-35 Count Timing in External Clock Operation

Output Compare Output Timing: A compare match signal is generated in the final state in which TCNT and TGR match (the point at which the count value matched by TCNT is updated). When a compare match signal is generated, the output value set in TIOR is output at the output compare output pin. After a match between TCNT and TGR, the compare match signal is not generated until the TCNT input clock is generated.

Figure 10-36 shows output compare output timing.

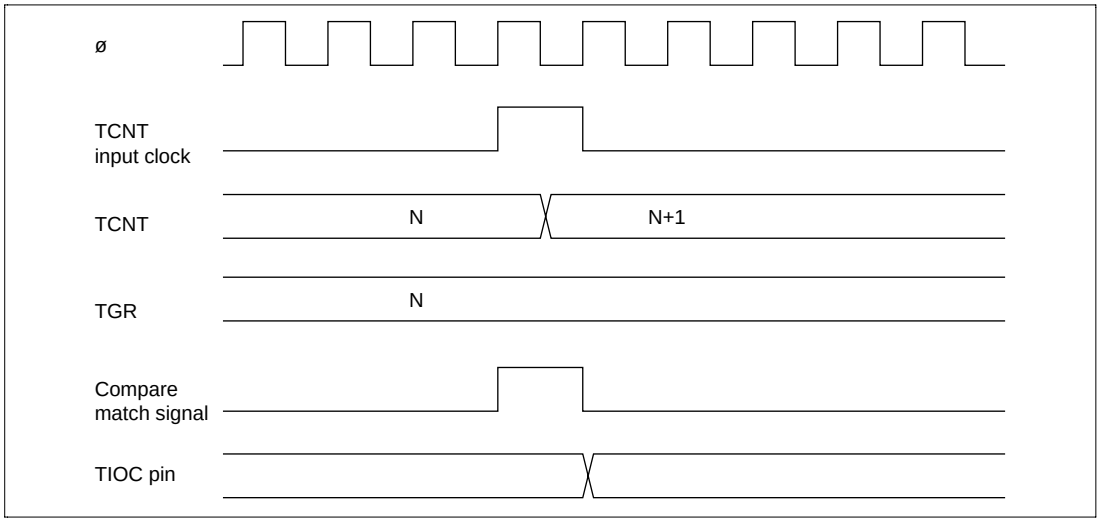


Figure 10-36 Output Compare Output Timing

Input Capture Signal Timing: Figure 10-37 shows input capture signal timing.

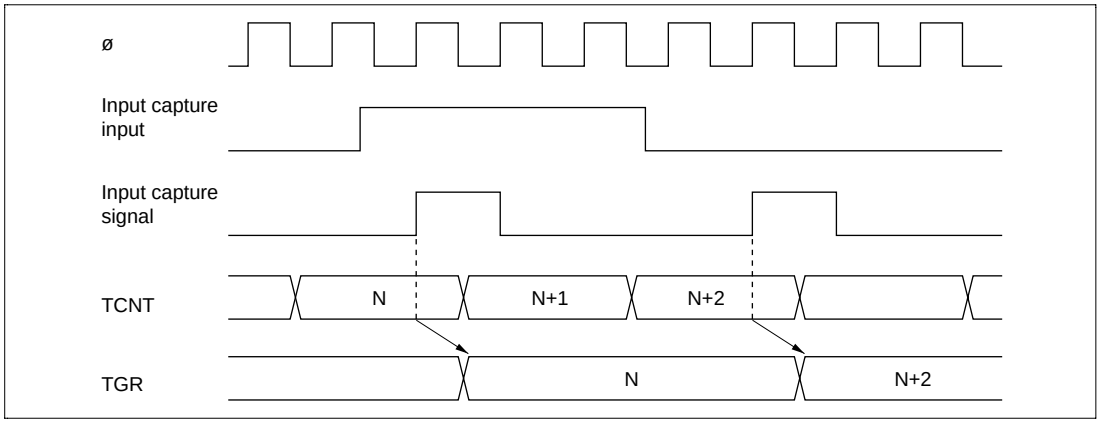


Figure 10-37 Input Capture Input Signal Timing

Timing for Counter Clearing by Compare Match/Input Capture: Figure 10-38 shows the timing when counter clearing by compare match occurrence is specified, and figure 10-39 shows the timing when counter clearing by input capture occurrence is specified.

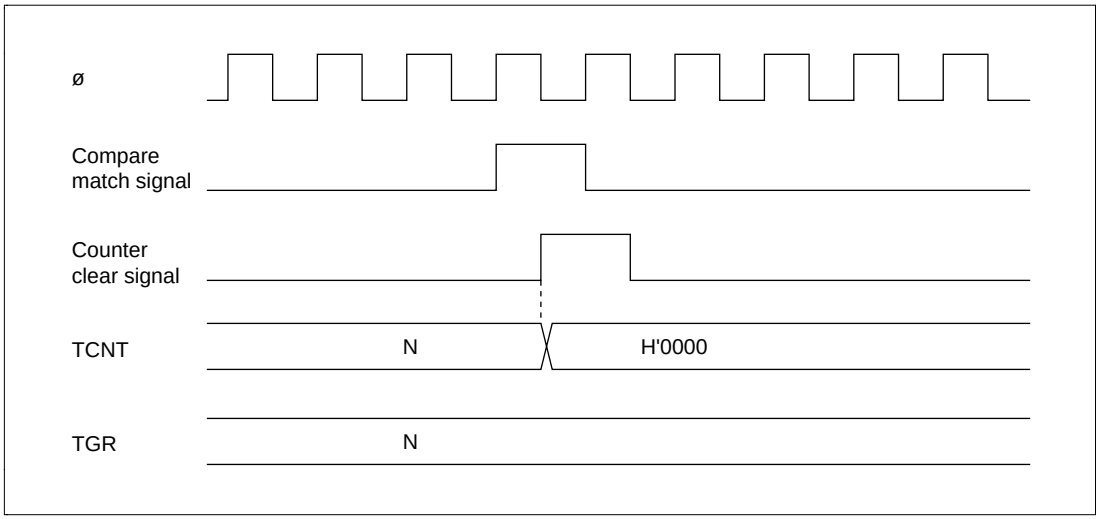


Figure 10-38 Counter Clear Timing (Compare Match)

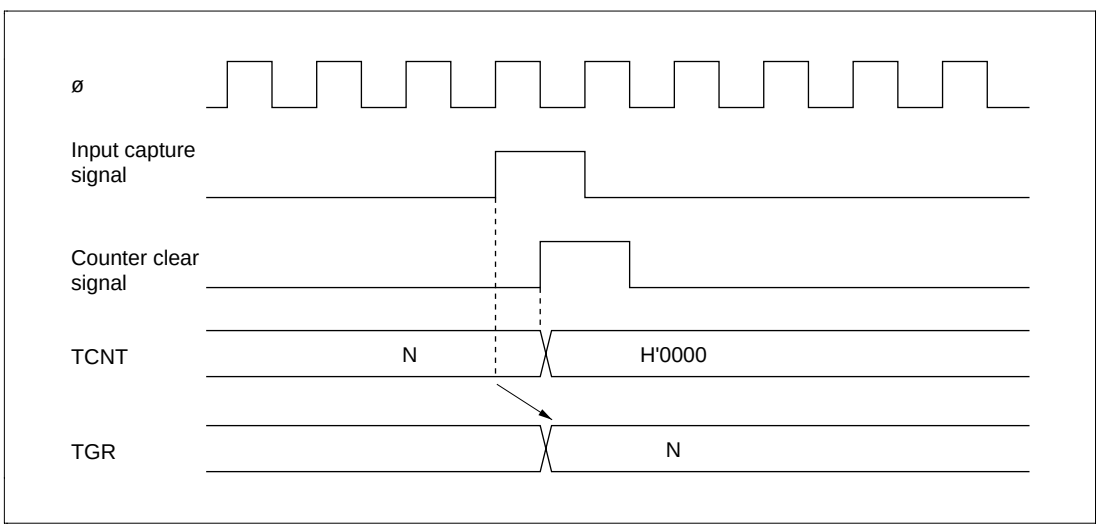


Figure 10-39 Counter Clear Timing (Input Capture)

Buffer Operation Timing: Figures 10-40 and 10-41 show the timing in buffer operation.

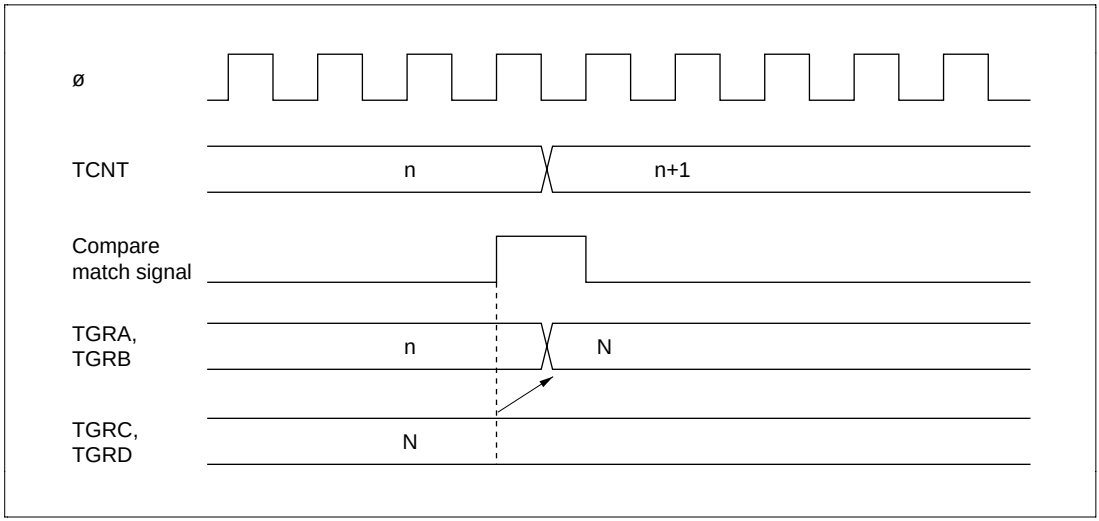


Figure 10-40 Buffer Operation Timing (Compare Match)

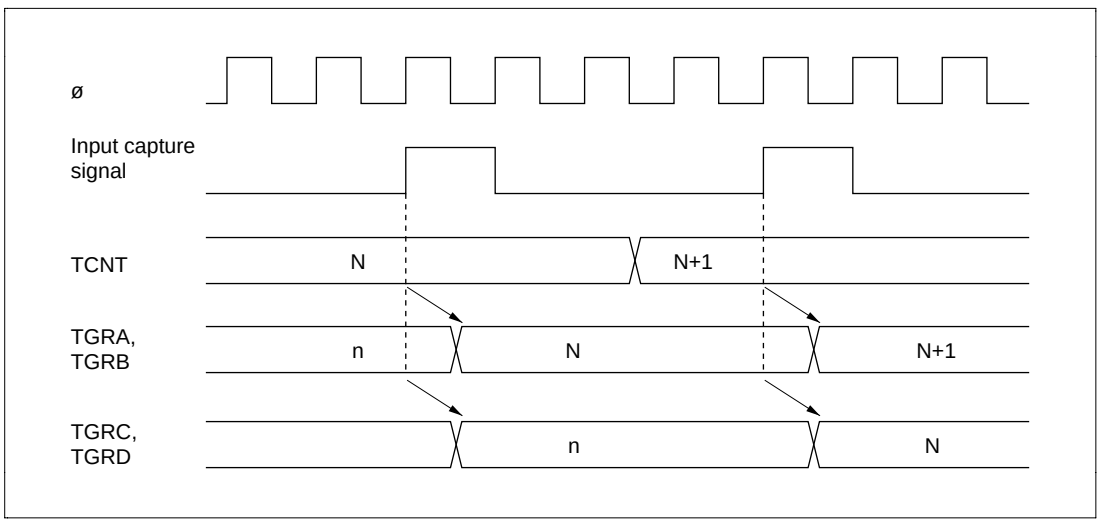


Figure 10-41 Buffer Operation Timing (Input Capture)

10.6.2 Interrupt Signal Timing

TGF Flag Setting Timing in Case of Compare Match: Figure 10-42 shows the timing for setting of the TGF flag in TSR by compare match occurrence, and TGI interrupt request signal timing.

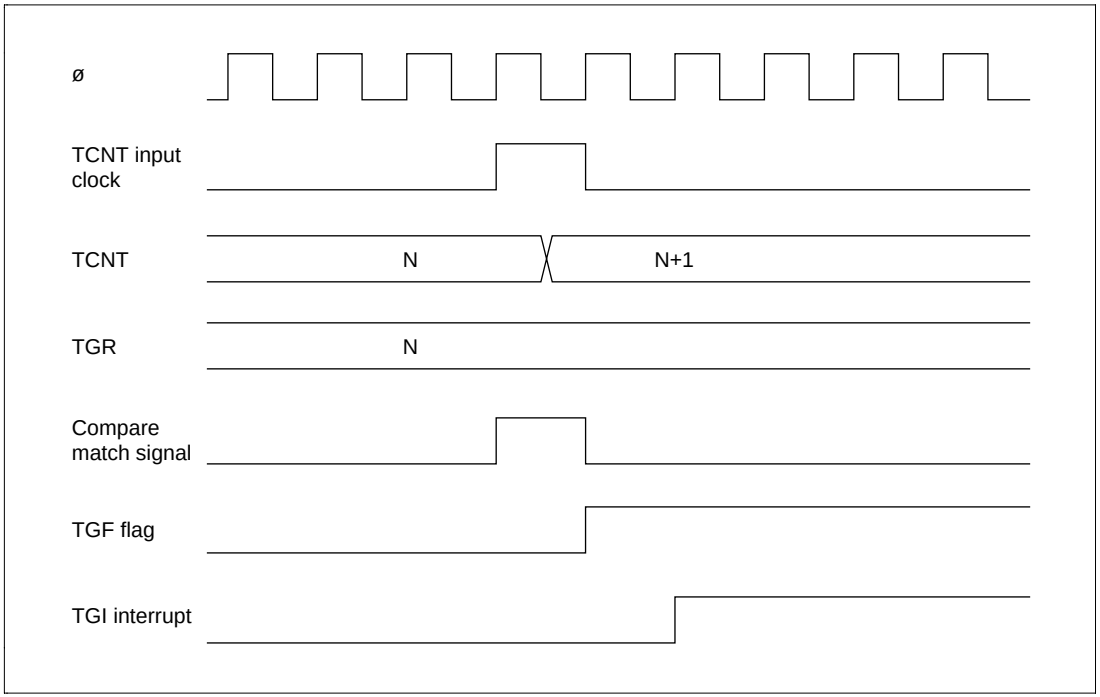


Figure 10-42 TGI Interrupt Timing (Compare Match)

TGF Flag Setting Timing in Case of Input Capture: Figure 10-43 shows the timing for setting of the TGF flag in TSR by input capture occurrence, and TGI interrupt request signal timing.

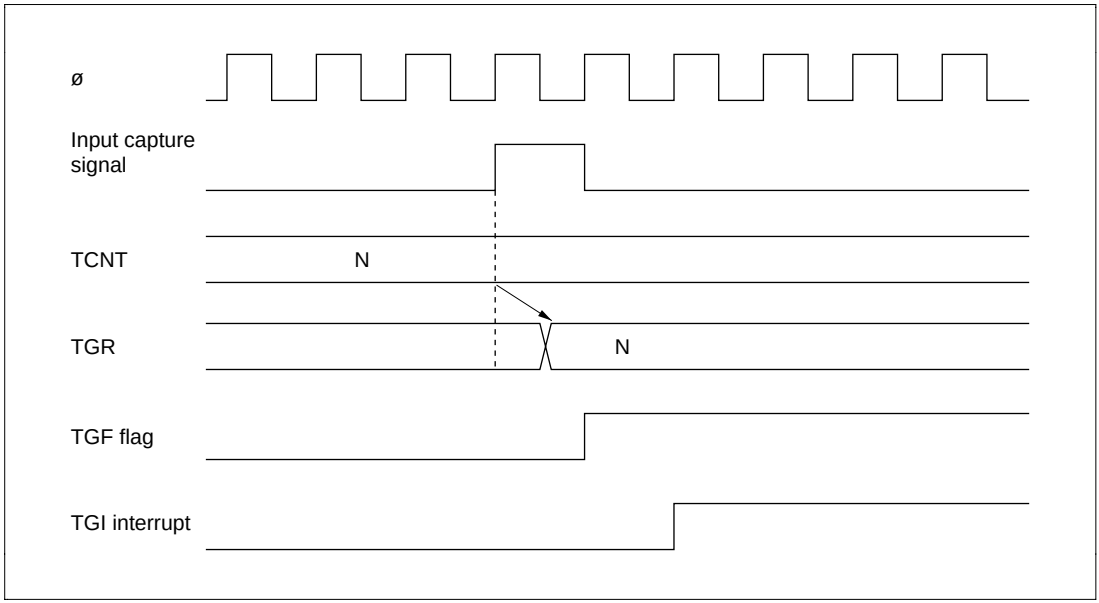


Figure 10-43 TGI Interrupt Timing (Input Capture)

TCFV Flag/TCFU Flag Setting Timing: Figure 10-44 shows the timing for setting of the TCFV flag in TSR by overflow occurrence, and TCIV interrupt request signal timing.

Figure 10-45 shows the timing for setting of the TCFU flag in TSR by underflow occurrence, and TCIU interrupt request signal timing.

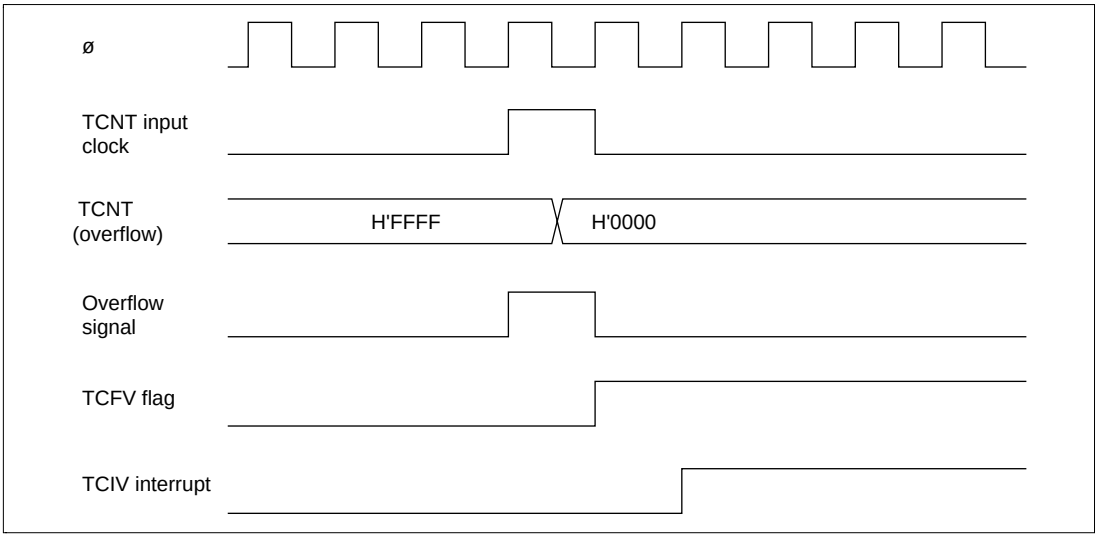


Figure 10-44 TCIV Interrupt Setting Timing

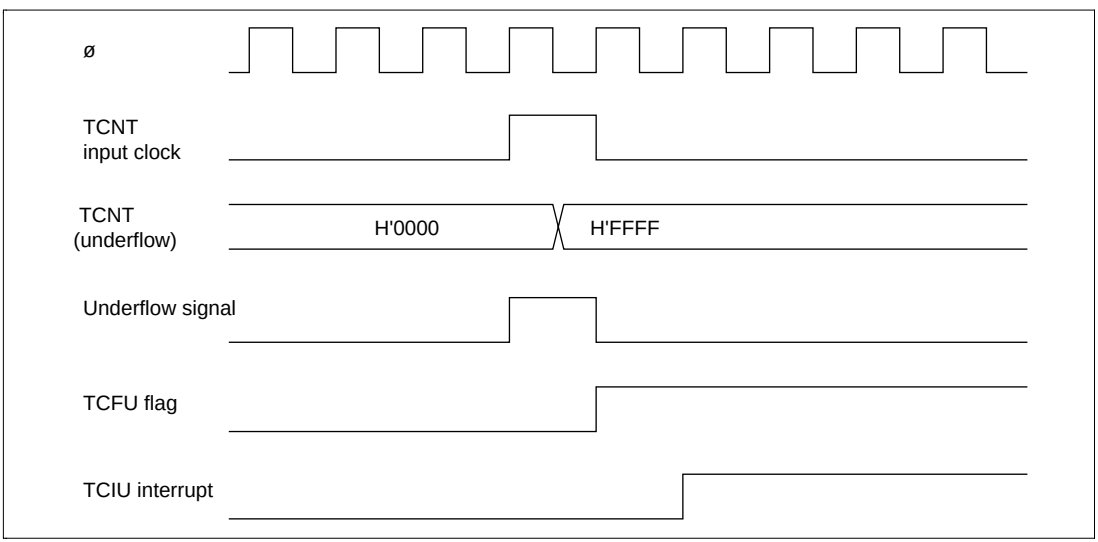


Figure 10-45 TCIU Interrupt Setting Timing

Status Flag Clearing Timing: After a status flag is read as 1 by the CPU, it is cleared by writing 0 to it. When the DTC is activated, the flag is cleared automatically. Figure 10-46 shows the timing for status flag clearing by the CPU, and figure 10-47 shows the timing for status flag clearing by the DTC.

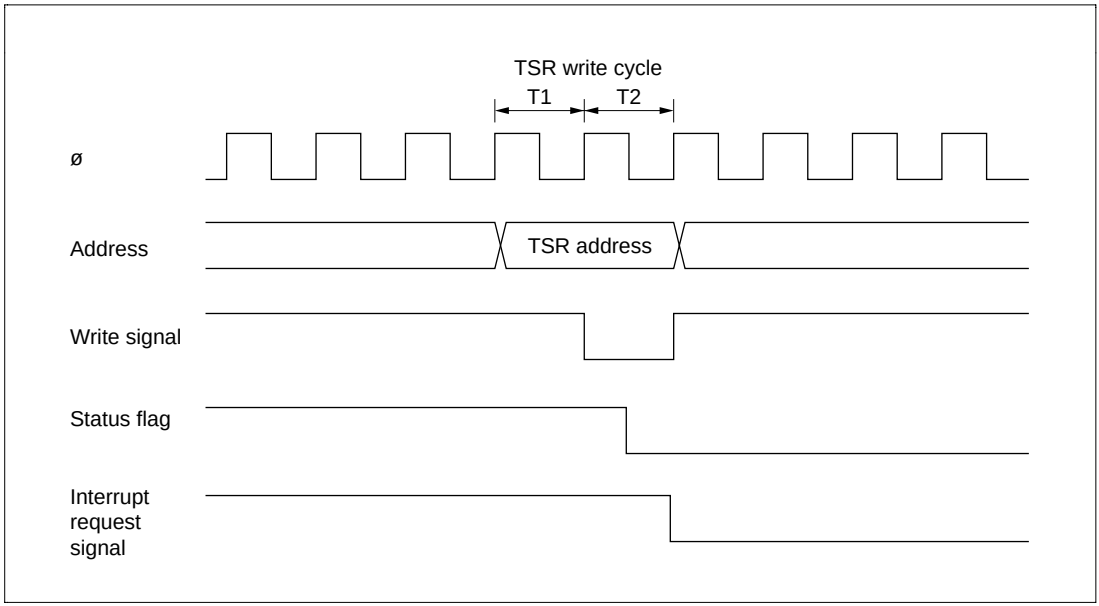


Figure 10-46 Timing for Status Flag Clearing by CPU

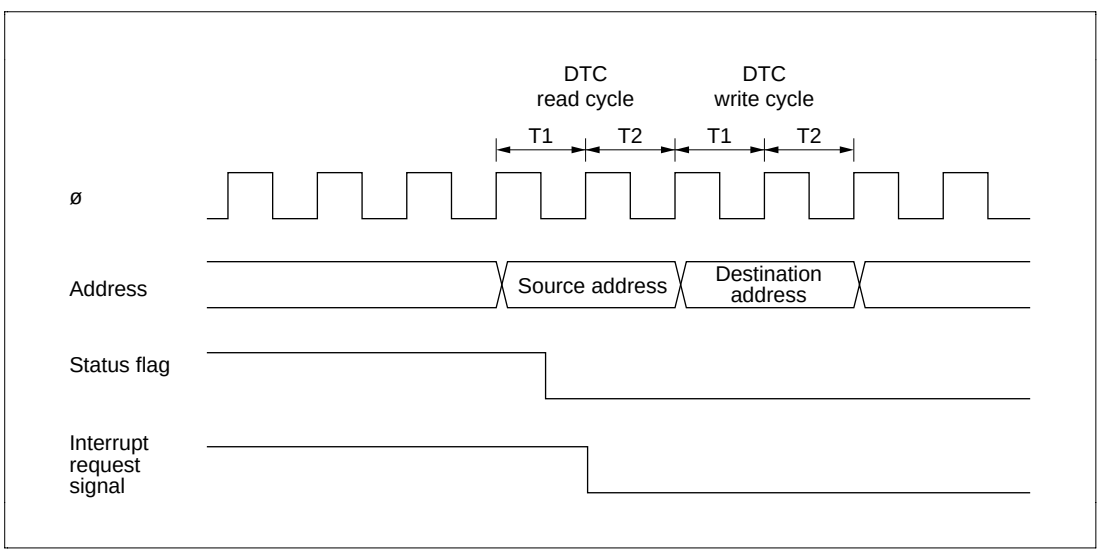


Figure 10-47 Timing for Status Flag Clearing by DTC Activation

10.7 Usage Notes

Note that the kinds of operation and contention described below occur during TPU operation.

Input Clock Restrictions: The input clock pulse width must be at least 1.5 states in the case of single-edge detection, and at least 2.5 states in the case of both-edge detection. The TPU will not operate properly with a narrower pulse width.

In phase counting mode, the phase difference and overlap between the two input clocks must be at least 1.5 states, and the pulse width must be at least 2.5 states. Figure 10-48 shows the input clock conditions in phase counting mode.

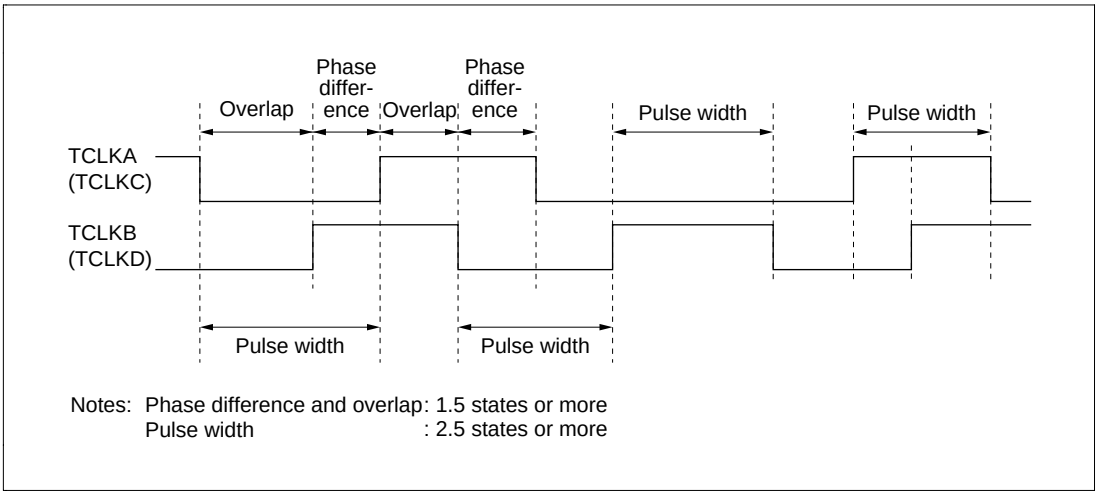


Figure 10-48 Phase Difference, Overlap, and Pulse Width in Phase Counting Mode

Caution on Period Setting: When counter clearing by compare match is set, TCNT is cleared in the final state in which it matches the TGR value (the point at which the count value matched by TCNT is updated). Consequently, the actual counter frequency is given by the following formula:

$$f = \frac{ø}{(N + 1)}$$

Where f : Counter frequency
ø : Operating frequency
N : TGR set value

Contention between TCNT Write and Clear Operations: If the counter clear signal is generated in the T2 state of a TCNT write cycle, TCNT clearing takes precedence and the TCNT write is not performed.

Figure 10-49 shows the timing in this case.

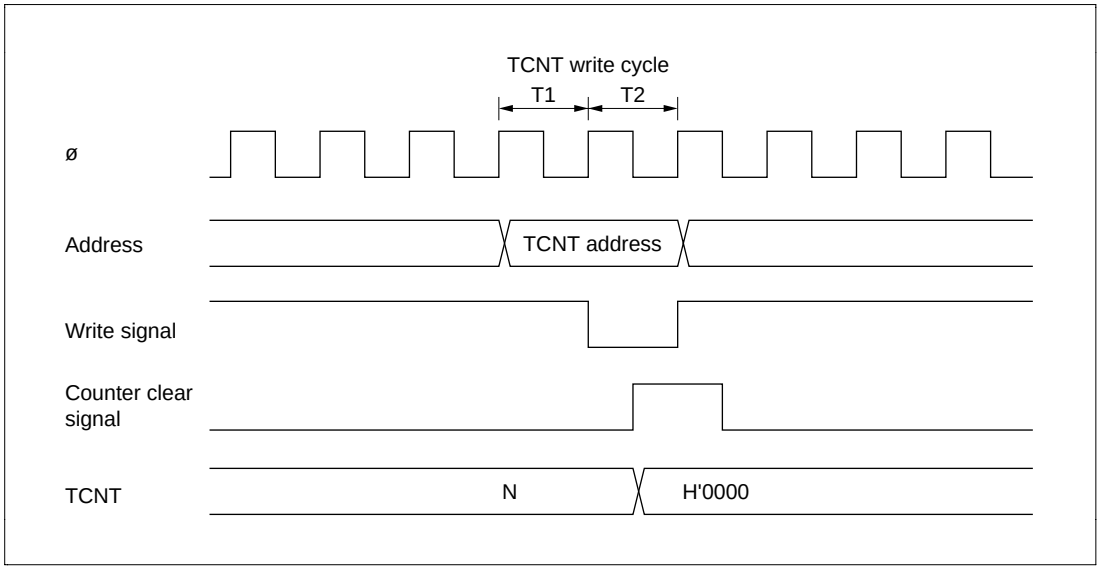


Figure 10-49 Contention between TCNT Write and Clear Operations

Contention between TCNT Write and Increment Operations: If incrementing occurs in the T2 state of a TCNT write cycle, the TCNT write takes precedence and TCNT is not incremented.

Figure 10-50 shows the timing in this case.

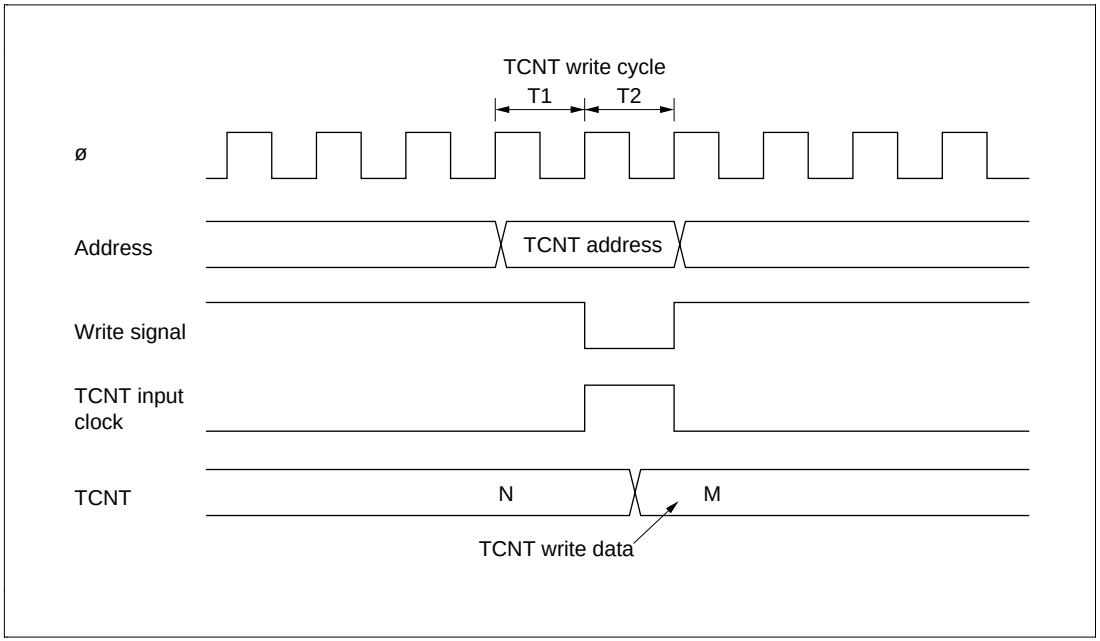


Figure 10-50 Contention between TCNT Write and Increment Operations

Contention between TGR Write and Compare Match: If a compare match occurs in the T2 state of a TGR write cycle, the TGR write takes precedence and the compare match signal is inhibited. A compare match does not occur even if the same value as before is written.

Figure 10-51 shows the timing in this case.

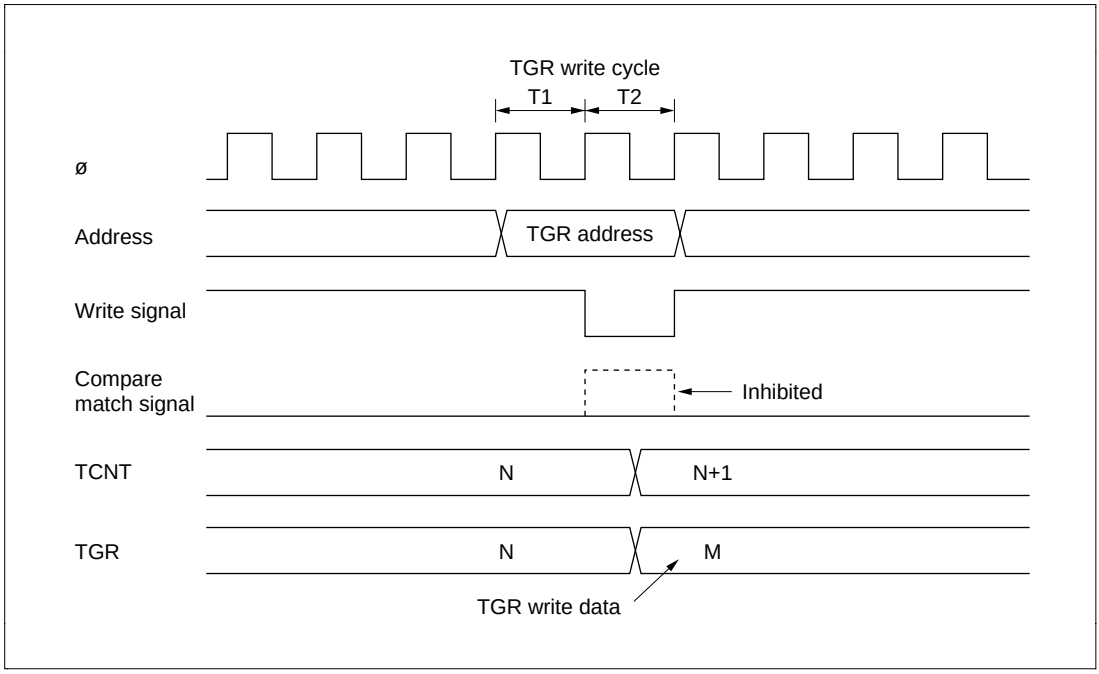


Figure 10-51 Contention between TGR Write and Compare Match

Contention between Buffer Register Write and Compare Match: If a compare match occurs in the T2 state of a TGR write cycle, the data transferred to TGR by the buffer operation will be the data prior to the write.

Figure 10-52 shows the timing in this case.

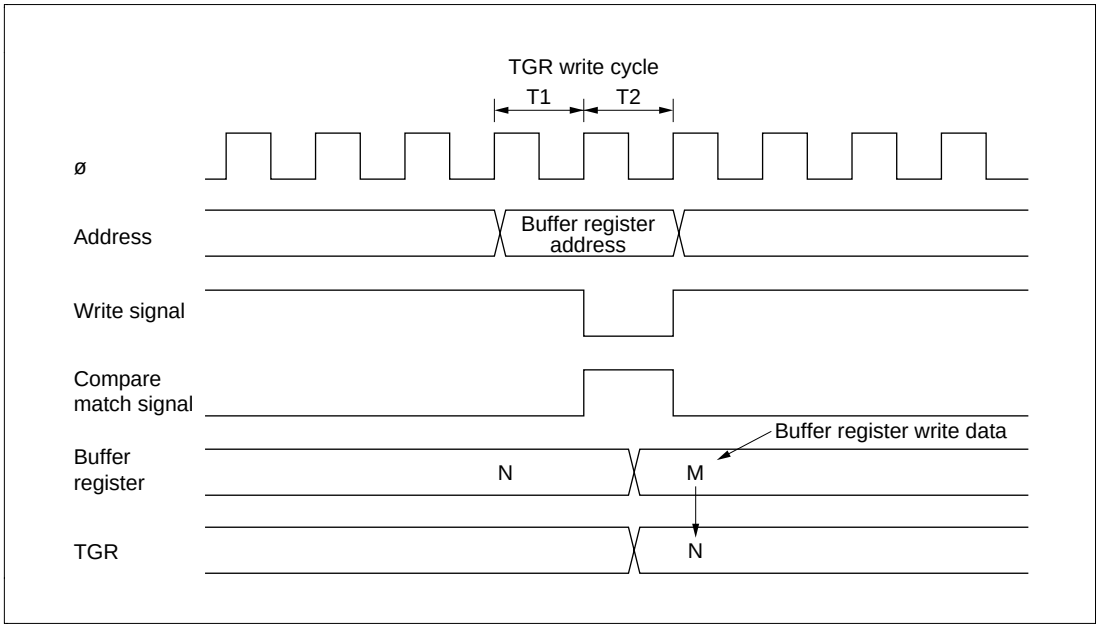


Figure 10-52 Contention between Buffer Register Write and Compare Match

Contention between TGR Read and Input Capture: If the input capture signal is generated in the T1 state of a TGR read cycle, the data that is read will be the data after input capture transfer.

Figure 10-53 shows the timing in this case.

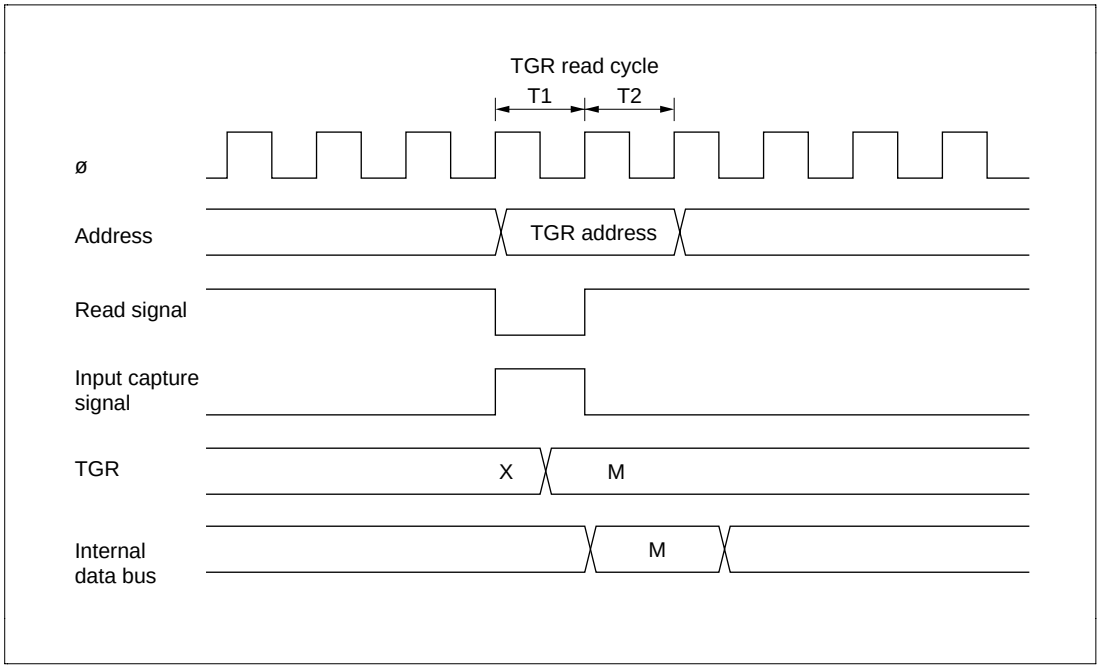


Figure 10-53 Contention between TGR Read and Input Capture

Contention between TGR Write and Input Capture: If the input capture signal is generated in the T2 state of a TGR write cycle, the input capture operation takes precedence and the write to TGR is not performed.

Figure 10-54 shows the timing in this case.

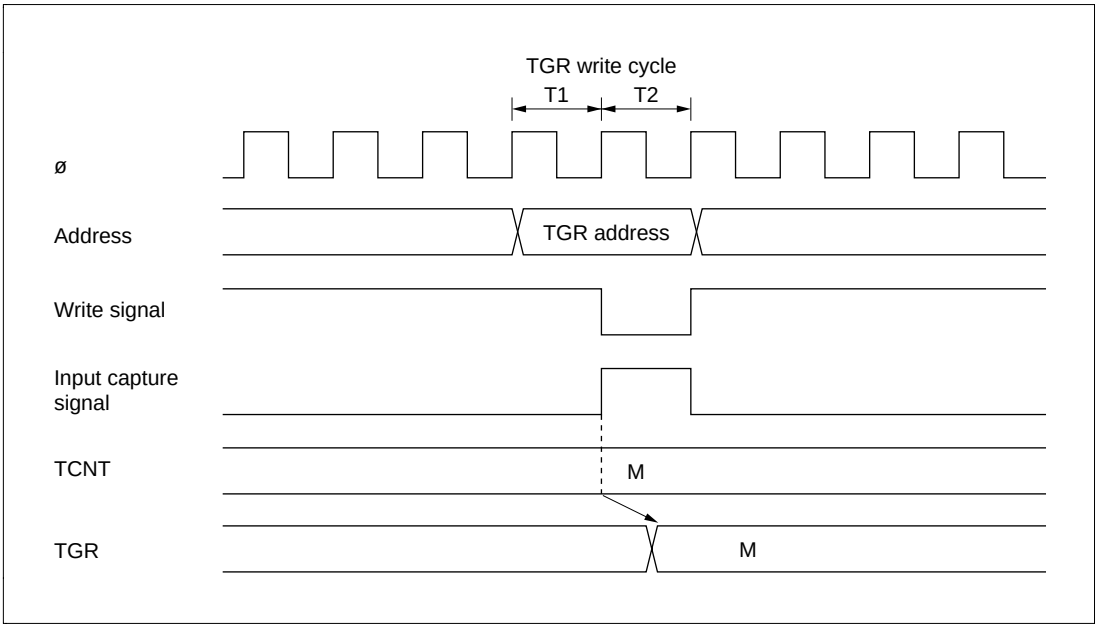


Figure 10-54 Contention between TGR Write and Input Capture

Contention between Buffer Register Write and Input Capture: If the input capture signal is generated in the T2 state of a buffer write cycle, the buffer operation takes precedence and the write to the buffer register is not performed.

Figure 10-55 shows the timing in this case.

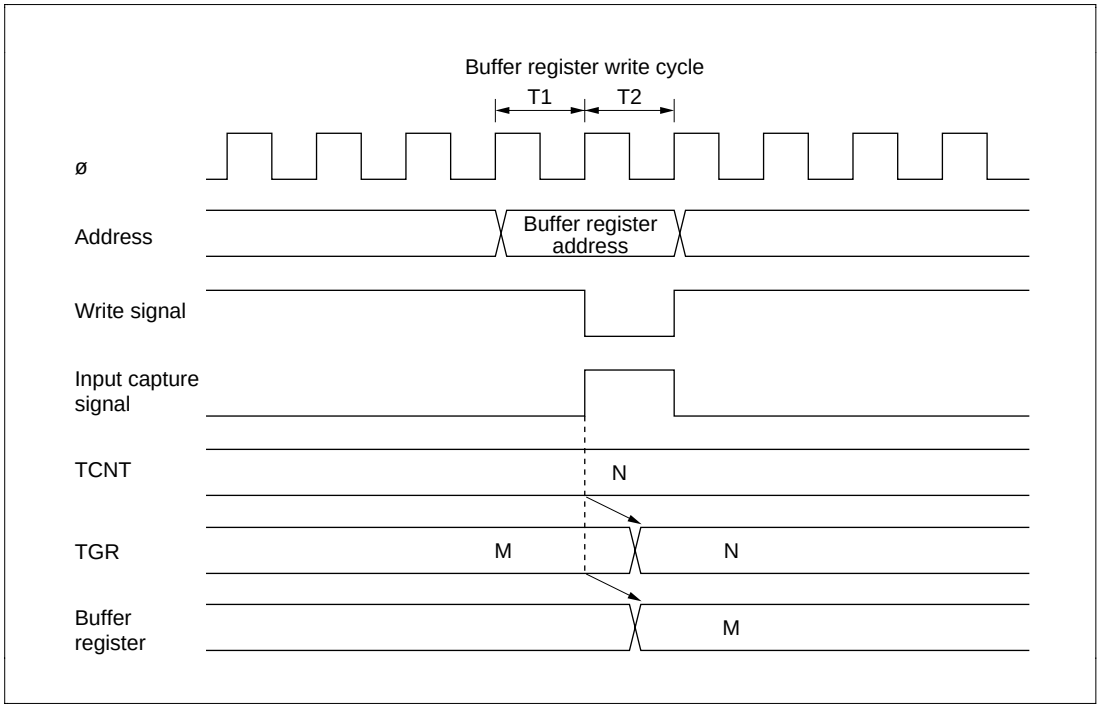


Figure 10-55 Contention between Buffer Register Write and Input Capture

Contention between Overflow/Underflow and Counter Clearing: If overflow/underflow and counter clearing occur simultaneously, the TCFV/TCFU flag in TSR is not set and TCNT clearing takes precedence.

Figure 10-56 shows the operation timing when a TGR compare match is specified as the clearing source, and H'FFFF is set in TGR.

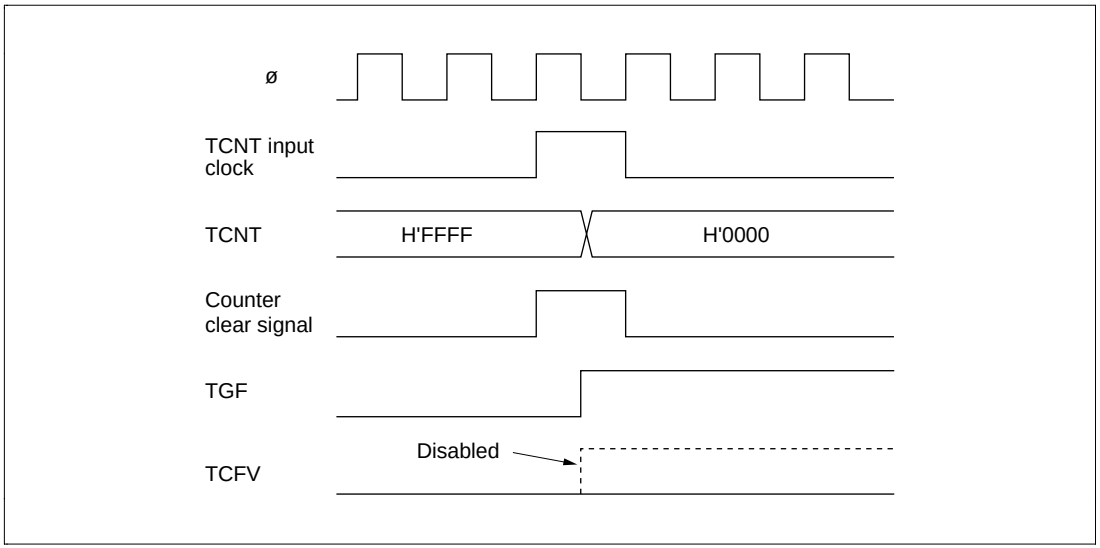


Figure 10-56 Contention between Overflow and Counter Clearing

Contention between TCNT Write and Overflow/Underflow: If there is an up-count or down-count in the T2 state of a TCNT write cycle, and overflow/underflow occurs, the TCNT write takes precedence and the TCFV/TCFU flag in TSR is not set.

Figure 10-57 shows the operation timing when there is contention between TCNT write and overflow.

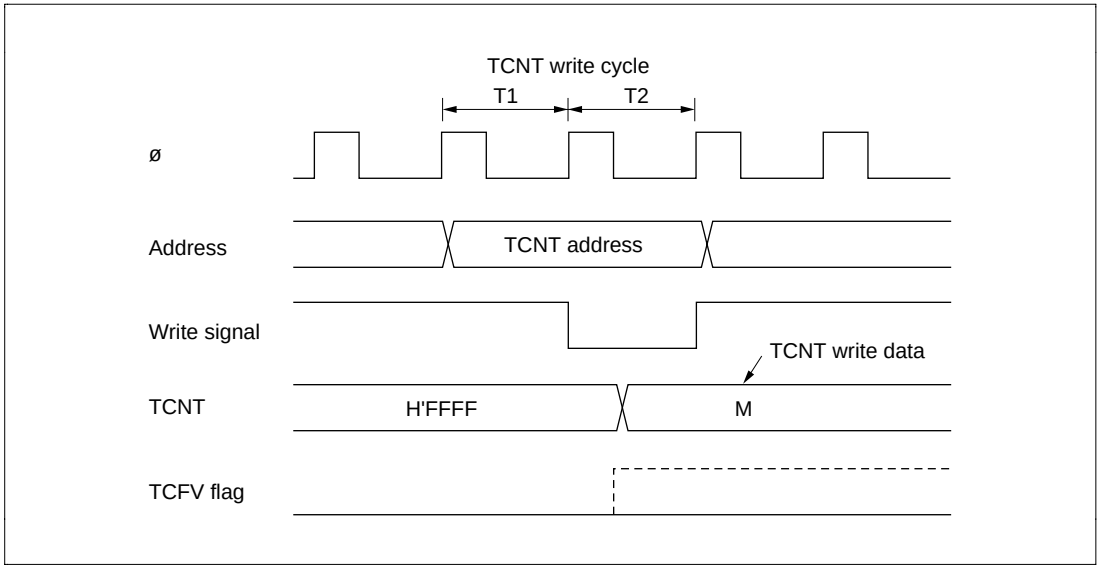


Figure 10-57 Contention between TCNT Write and Overflow

Multiplexing of I/O Pins: In the chip, the TCLKA input pin is multiplexed with the TIOCC0 I/O pin, the TCLKB input pin with the TIOCD0 I/O pin, the TCLKC input pin with the TIOCB1 I/O pin, and the TCLKD input pin with the TIOCB2 I/O pin. When an external clock is input, compare match output should not be performed from a multiplexed pin.

Interrupts and Module Stop Mode: If module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Section 11 Programmable Pulse Generator (PPG)

11.1 Overview

The chip has a built-in programmable pulse generator (PPG) that provides pulse outputs by using the 16-bit timer-pulse unit (TPU) as a time base. The PPG pulse outputs are divided into 4-bit groups (group 3 and group 2) that can operate both simultaneously and independently.

11.1.1 Features

PPG features are listed below.

- 8-bit output data
  - — Maximum 8-bit data can be output, and output can be enabled on a bit-by-bit basis
- Two output groups
  - — Output trigger signals can be selected in 4-bit groups to provide up to two different 4-bit outputs
- Selectable output trigger signals
  - — Output trigger signals can be selected for each group from the compare match signals of four TPU channels
- Non-overlap mode
  - — A non-overlap margin can be provided between pulse outputs
- Can operate together with the data transfer controller (DTC)
  - — The compare match signals selected as output trigger signals can activate the DTC for sequential output of data without CPU intervention
- Settable inverted output
  - — Inverted data can be output for each group
- Module stop mode can be set
  - — As the initial setting, PPG operation is halted. Register access is enabled by exiting module stop mode

11.1.2 Block Diagram

Figure 11-1 shows a block diagram of the PPG.

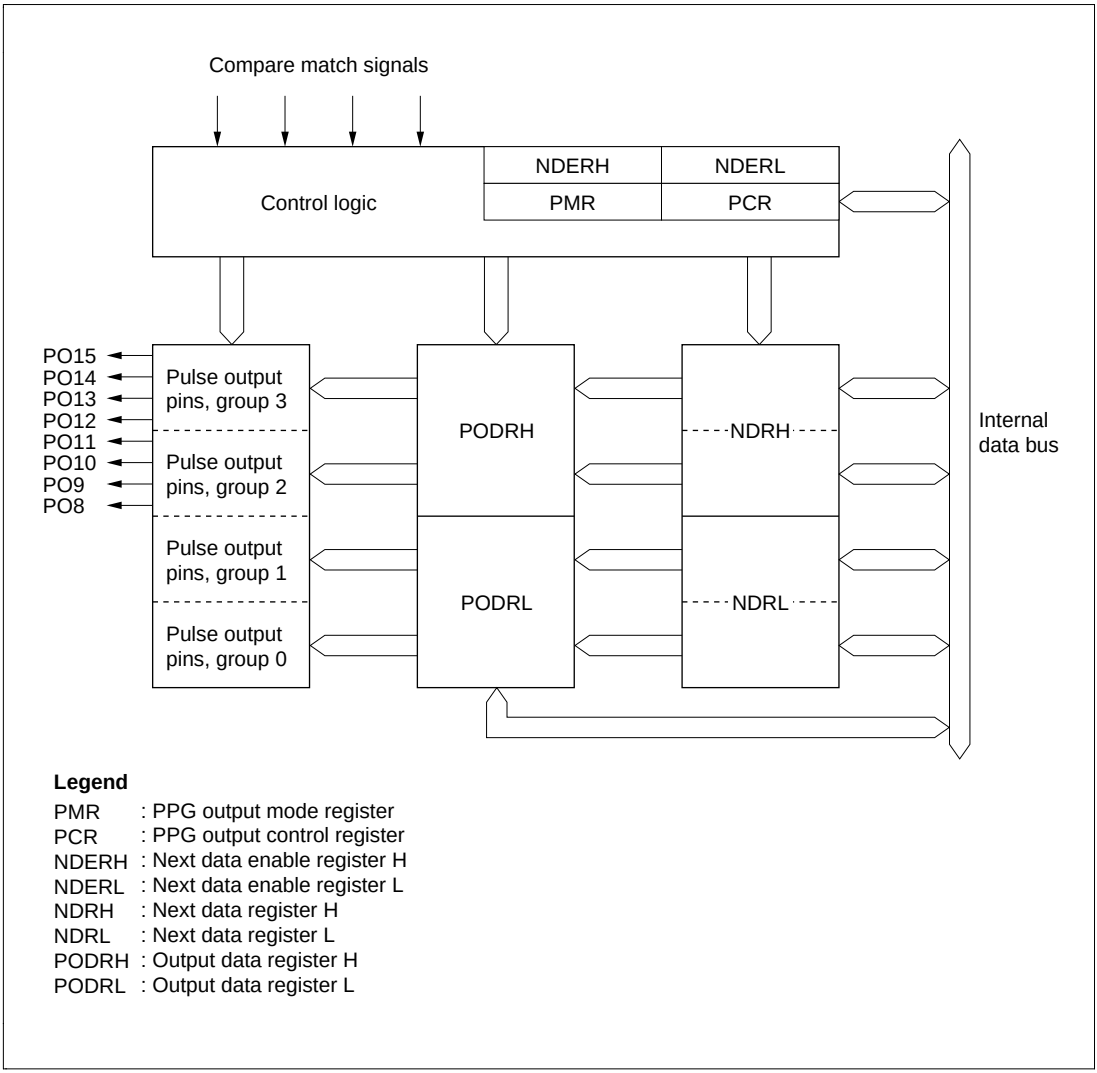


Figure 11-1 Block Diagram of PPG

11.1.3 Pin Configuration

Table 11-1 summarizes the PPG pins.

Table 11-1 PPG Pins

Name	Symbol	I/O	Function
Pulse output 8	PO8	Output	Group 2 pulse output
Pulse output 9	PO9	Output	
Pulse output 10	PO10	Output	
Pulse output 11	PO11	Output	
Pulse output 12	PO12	Output	Group 3 pulse output
Pulse output 13	PO13	Output	
Pulse output 14	PO14	Output	
Pulse output 15	PO15	Output	

11.1.4 Registers

Table 11-2 summarizes the PPG registers.

Table 11-2 PPG Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
PPG output control register	PCR	R/W	H'FF	H'FE26
PPG output mode register	PMR	R/W	H'F0	H'FE27
Next data enable register H	NDERH	R/W	H'00	H'FE28
Next data enable register L*[4]	NDERL	R/W	H'00	H'FE29
Output data register H	PODRH	R/(W)*[2]	H'00	H'FE2A
Output data register L*[4]	PODRL	R/(W)*[2]	H'00	H'FE2B
Next data register H	NDRH	R/W	H'00	H'FE2C*[3] H'FE2E
Next data register L*[4]	NDRL	R/W	H'00	H'FE2D*[3] H'FE2F
Port 1 data direction register	P1DDR	W	H'00	H'FE30
Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address.

*2 Bits used for pulse output cannot be written to.

*3 When the same output trigger is selected for pulse output groups 2 and 3 by the PCR setting, the NDRH address is H'FE2C. When the output triggers are different, the NDRH address is H'FE2E for group 2 and H'FE2C for group 3.

Similarly, when the same output trigger is selected for pulse output groups 0 and 1 by the PCR setting, the NDRL address is H'FE2D. When the output triggers are different, the NDRL address is H'FE2F for group 0 and H'FE2D for group 1.

*4 The chip has no pins corresponding to pulse output groups 0 and 1.

11.2 Register Descriptions

11.2.1 Next Data Enable Registers H and L (NDERH, NDERL)

NDERH

Bit	:	7	6	5	4	3	2	1	0
		NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NDERL

Bit	:	7	6	5	4	3	2	1	0
		NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

NDERH and NDERL are 8-bit readable/writable registers that enable or disable pulse output on a bit-by-bit basis.

If a bit is enabled for pulse output by NDERH or NDERL, the NDR value is automatically transferred to the corresponding PODR bit when the TPU compare match event specified by PCR occurs, updating the output value. If pulse output is disabled, the bit value is not transferred from NDR to PODR and the output value does not change.

NDERH and NDERL are each initialized to H'00 by a reset and in hardware standby mode. They are not initialized in software standby mode.

NDERH Bits 7 to 0—Next Data Enable 15 to 8 (NDER15 to NDER8): These bits enable or disable pulse output on a bit-by-bit basis.

Bits 7 to 0 NDER15 to NDER8	Description
0	Pulse outputs PO15 to PO8 are disabled (NDR15 to NDR8 are not transferred to POD15 to POD8) (Initial value)
1	Pulse outputs PO15 to PO8 are enabled (NDR15 to NDR8 are transferred to POD15 to POD8)

NDERL Bits 7 to 0—Next Data Enable 7 to 0 (NDER7 to NDER0): These bits enable or disable pulse output on a bit-by-bit basis.

Bits 7 to 0 NDER7 to NDER0	Description
0	Pulse outputs PO7 to PO0 are disabled (NDR7 to NDR0 are not transferred to POD7 to POD0) (Initial value)
1	Pulse outputs PO7 to PO0 are enabled (NDR7 to NDR0 are transferred to POD7 to POD0)

11.2.2 Output Data Registers H and L (PODRH, PODRL)

PODRH

Bit	:	7	6	5	4	3	2	1	0
		POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

PODRL

Bit	:	7	6	5	4	3	2	1	0
		POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * A bit that has been set for pulse output by NDER is read-only.

PODRH and PODRL are 8-bit readable/writable registers that store output data for use in pulse output. However, the chip has no pins corresponding to PODRL.

11.2.3 Next Data Registers H and L (NDRH, NDRL)

NDRH and NDRL are 8-bit readable/writable registers that store the next data for pulse output. During pulse output, the contents of NDRH and NDRL are transferred to the corresponding bits in PODRH and PODRL when the TPU compare match event specified by PCR occurs. The NDRH and NDRL addresses differ depending on whether pulse output groups have the same output trigger or different output triggers. For details see section 11.2.4, Notes on NDR Access.

NDRH and NDRL are each initialized to H'00 by a reset and in hardware standby mode. They are not initialized in software standby mode.

11.2.4 Notes on NDR Access

The NDRH and NDRL addresses differ depending on whether pulse output groups have the same output trigger or different output triggers.

Same Trigger for Pulse Output Groups: If pulse output groups 2 and 3 are triggered by the same compare match event, the NDRH address is H'FE2C. The upper 4 bits belong to group 3 and the lower 4 bits to group 2. Address H'FE2E consists entirely of reserved bits that cannot be modified and are always read as 1.

Address H'FE2C

Bit	:	7	6	5	4	3	2	1	0
		NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address H'FE2E

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	—	—	—	—
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	—	—	—	—	—	—	—	—

If pulse output groups 0 and 1 are triggered by the same compare match event, the NDRL address is H'FE2D. The upper 4 bits belong to group 1 and the lower 4 bits to group 0. Address H'FE2F consists entirely of reserved bits that cannot be modified and are always read as 1. However, the chip has no output pins corresponding to pulse output groups 0 and 1.

Address H'FE2D

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address H'FE2F

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
R/W	—	—	—	—	—	—	—	—

Different Triggers for Pulse Output Groups: If pulse output groups 2 and 3 are triggered by different compare match events, the address of the upper 4 bits in NDRH (group 3) is H'FE2C and the address of the lower 4 bits (group 2) is H'FE2E. Bits 3 to 0 of address H'FE2C and bits 7 to 4 of address H'FE2E are reserved bits that cannot be modified and are always read as 1.

Address H'FE2C

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
R/W	R/W	R/W	R/W	R/W	—	—	—	—

Address H'FE2E

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	1	1	1	1	0	0	0	0
R/W	—	—	—	—	R/W	R/W	R/W	R/W

If pulse output groups 0 and 1 are triggered by different compare match event, the address of the upper 4 bits in NDRL (group 1) is H'FE2D and the address of the lower 4 bits (group 0) is H'FE2F. Bits 3 to 0 of address H'FE2D and bits 7 to 4 of address H'FE2F are reserved bits that cannot be modified and are always read as 1. However, the chip has no output pins corresponding to pulse output groups 0 and 1.

Address H'FE2D

Bit	:	7	6	5	4	3	2	1	0
		NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	:	0	0	0	0	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	—	—	—	—

Address H'FE2F

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	:	1	1	1	1	0	0	0	0
R/W	:	—	—	—	—	R/W	R/W	R/W	R/W

11.2.5 PPG Output Control Register (PCR)

Bit	:	7	6	5	4	3	2	1	0
		G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCR is an 8-bit readable/writable register that selects output trigger signals for PPG outputs on a group-by-group basis.

PCR is initialized to H'FF by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 7 and 6—Group 3 Compare Match Select 1 and 0 (G3CMS1, G3CMS0): These bits select the compare match that triggers pulse output group 3 (pins PO15 to PO12).

		Description	
Bit 7 G3CMS1	**Bit 6 G3CMS0**	**Output Trigger for Pulse Output Group 3**	
0	0	Compare match in TPU channel 0	
	1	Compare match in TPU channel 1	
1	0	Compare match in TPU channel 2	
	1	Compare match in TPU channel 3	(Initial value)

Bits 5 and 4—Group 2 Compare Match Select 1 and 0 (G2CMS1, G2CMS0): These bits select the compare match that triggers pulse output group 2 (pins PO11 to PO8).

		Description	
Bit 5 G2CMS1	**Bit 4 G2CMS0**	**Output Trigger for Pulse Output Group 2**	
0	0	Compare match in TPU channel 0	
	1	Compare match in TPU channel 1	
1	0	Compare match in TPU channel 2	
	1	Compare match in TPU channel 3	(Initial value)

Bits 3 and 2—Group 1 Compare Match Select 1 and 0 (G1CMS1, G1CMS0): These bits select the compare match that triggers pulse output group 1 (pins PO7 to PO4). However, the chip has no output pins corresponding to pulse output group 1.

		Description	
Bit 3 G1CMS1	**Bit 2 G1CMS0**	**Output Trigger for Pulse Output Group 1**	
0	0	Compare match in TPU channel 0	
	1	Compare match in TPU channel 1	
1	0	Compare match in TPU channel 2	
	1	Compare match in TPU channel 3	(Initial value)

Bits 1 and 0—Group 0 Compare Match Select 1 and 0 (G0CMS1, G0CMS0): These bits select the compare match that triggers pulse output group 0 (pins PO3 to PO0). However, the chip has no output pins corresponding to pulse output group 0.

		Description	
Bit 1 G0CMS1	**Bit 0 G0CMS0**	**Output Trigger for Pulse Output Group 0**	
0	0	Compare match in TPU channel 0	
	1	Compare match in TPU channel 1	
1	0	Compare match in TPU channel 2	
	1	Compare match in TPU channel 3	(Initial value)

11.2.6 PPG Output Mode Register (PMR)

Bit	:	7	6	5	4	3	2	1	0
		G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	:	1	1	1	1	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PMR is an 8-bit readable/writable register that selects pulse output inversion and non-overlapping operation for each group.

The output trigger period of a non-overlapping operation PPG output waveform is set in TGRB and the non-overlap margin is set in TGRA. The output values change at compare match A and B.

For details, see section 11.3.4, Non-Overlapping Pulse Output.

PMR is initialized to H'F0 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Group 3 Inversion (G3INV): Selects direct output or inverted output for pulse output group 3 (pins PO15 to PO12).

Bit 7 G3INV	Description
0	Inverted output for pulse output group 3 (low-level output at pin for a 1 in PODRH)
1	Direct output for pulse output group 3 (high-level output at pin for a 1 in PODRH) (Initial value)

Bit 6—Group 2 Inversion (G2INV): Selects direct output or inverted output for pulse output group 2 (pins PO11 to PO8).

Bit 6 G2INV	Description
0	Inverted output for pulse output group 2 (low-level output at pin for a 1 in PODRH)
1	Direct output for pulse output group 2 (high-level output at pin for a 1 in PODRH) (Initial value)

Bit 5—Group 1 Inversion (G1INV): Selects direct output or inverted output for pulse output group 1 (pins PO7 to PO4). However, the chip has no pins corresponding to pulse output group 1.

Bit 5 G1INV	Description
0	Inverted output for pulse output group 1 (low-level output at pin for a 1 in PODRL)
1	Direct output for pulse output group 1 (high-level output at pin for a 1 in PODRL) (Initial value)

Bit 4—Group 0 Inversion (G0INV): Selects direct output or inverted output for pulse output group 0 (pins PO3 to PO0). However, the chip has no pins corresponding to pulse output group 0.

Bit 4 G0INV	Description
0	Inverted output for pulse output group 0 (low-level output at pin for a 1 in PODRL)
1	Direct output for pulse output group 0 (high-level output at pin for a 1 in PODRL) (Initial value)

Bit 3—Group 3 Non-Overlap (G3NOV): Selects normal or non-overlapping operation for pulse output group 3 (pins PO15 to PO12).

Bit 3 G3NOV	Description
0	Normal operation in pulse output group 3 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 3 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Bit 2—Group 2 Non-Overlap (G2NOV): Selects normal or non-overlapping operation for pulse output group 2 (pins PO11 to PO8).

Bit 2 G2NOV	Description
0	Normal operation in pulse output group 2 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 2 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Bit 1—Group 1 Non-Overlap (G1NOV): Selects normal or non-overlapping operation for pulse output group 1 (pins PO7 to PO4). However, the chip has no pins corresponding to pulse output group 1.

Bit 1 G1NOV	Description
0	Normal operation in pulse output group 1 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 1 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Bit 0—Group 0 Non-Overlap (G0NOV): Selects normal or non-overlapping operation for pulse output group 0 (pins PO3 to PO0). However, the chip has no pins corresponding to pulse output group 0.

Bit 0 G0NOV	Description
0	Normal operation in pulse output group 0 (output values updated at compare match A in the selected TPU channel) (Initial value)
1	Non-overlapping operation in pulse output group 0 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

11.2.7 Port 1 Data Direction Register (P1DDR)

Bit	:	7	6	5	4	3	2	1	0
		P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	W	W	W	W	W	W	W	W

P1DDR is an 8-bit write-only register, the individual bits of which specify input or output for the pins of port 1.

Port 1 is multiplexed with pins PO15 to PO8. Bits corresponding to pins used for PPG output must be set to 1. For further information about P1DDR, see section 9.2, Port 1.

11.2.8 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRA is a 16-bit readable/writable register that performs module stop mode control.

When the MSTPA3 bit in MSTPCRA is set to 1, PPG operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 23A.5, 23B.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Bit 3—Module Stop (MSTPA3): Specifies the PPG module stop mode.

Bit 3 MSTPA3	Description	
0	PPG module stop mode cleared	
1	PPG module stop mode set	(Initial value)

11.3 Operation

11.3.1 Overview

PPG pulse output is enabled when the corresponding bits in P1DDR and NDER are set to 1. In this state the corresponding PODR contents are output.

When the compare match event specified by PCR occurs, the corresponding NDR bit contents are transferred to PODR to update the output values.

Figure 11-2 illustrates the PPG output operation and table 11-3 summarizes the PPG operating conditions.

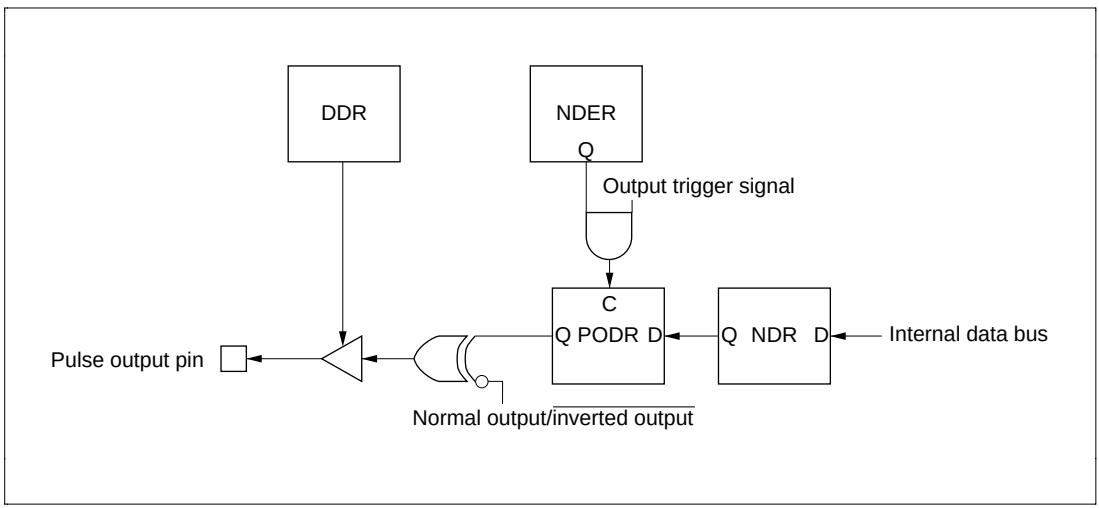


Figure 11-2 PPG Output Operation

Table 11-3 PPG Operating Conditions

NDER	DDR	Pin Function
0	0	Generic input port
	1	Generic output port
1	0	Generic input port (but the PODR bit is a read-only bit, and when compare match occurs, the NDR bit value is transferred to the PODR bit)
	1	PPG pulse output

Sequential output of data of up to 16 bits is possible by writing new output data to NDR before the next compare match. For details of non-overlapping operation, see section 11.3.4, Non-Overlapping Pulse Output.

11.3.2 Output Timing

If pulse output is enabled, NDR contents are transferred to PODR and output when the specified compare match event occurs. Figure 11-3 shows the timing of these operations for the case of normal output in groups 2 and 3, triggered by compare match A.

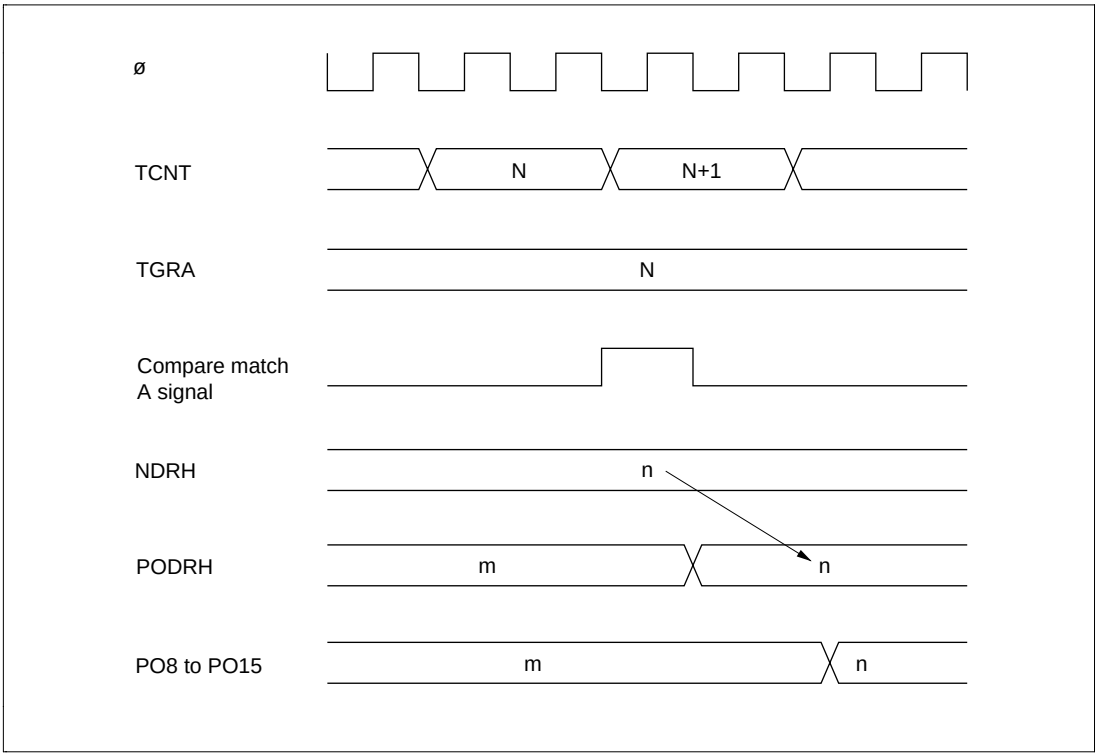
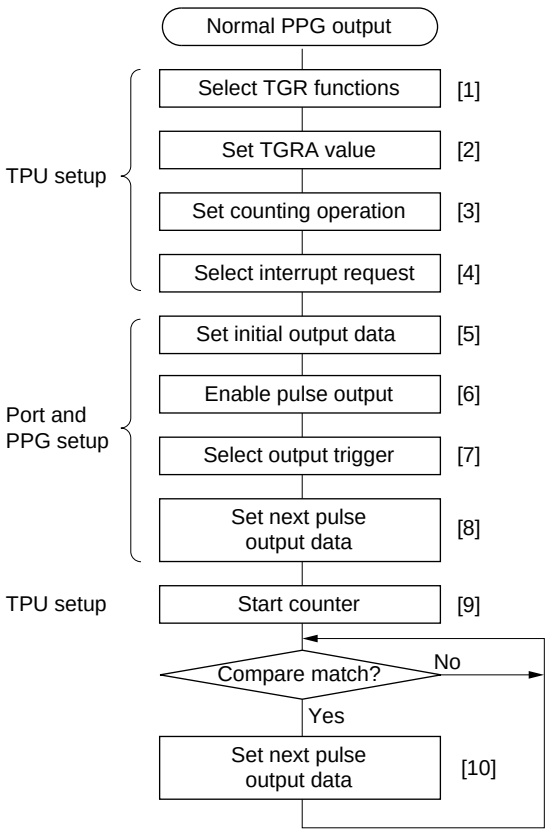


Figure 11-3 Timing of Transfer and Output of NDR Contents (Example)

11.3.3 Normal Pulse Output

Sample Setup Procedure for Normal Pulse Output: Figure 11-4 shows a sample procedure for setting up normal pulse output.



[1] Set TIOR to make TGRA an output compare register (with output disabled)

[2] Set the PPG output trigger period

[3] Select the counter clock source with bits TPSC2 to TPSC0 in TCR. Select the counter clear source with bits CCLR1 and CCLR0.

[4] Enable the TGIA interrupt in TIER. The DTC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the DDR and NDER bits for the pins to be used for pulse output to 1.

[7] Select the TPU compare match event to be used as the output trigger in PCR.

[8] Set the next pulse output values in NDR.

[9] Set the CST bit in TSTR to 1 to start the TCNT counter.

[10] At each TGIA interrupt, set the next output values in NDR.

Figure 11-4 Setup Procedure for Normal Pulse Output (Example)

Example of Normal Pulse Output (Example of Five-Phase Pulse Output): Figure 11-5 shows an example in which pulse output is used for cyclic five-phase pulse output.

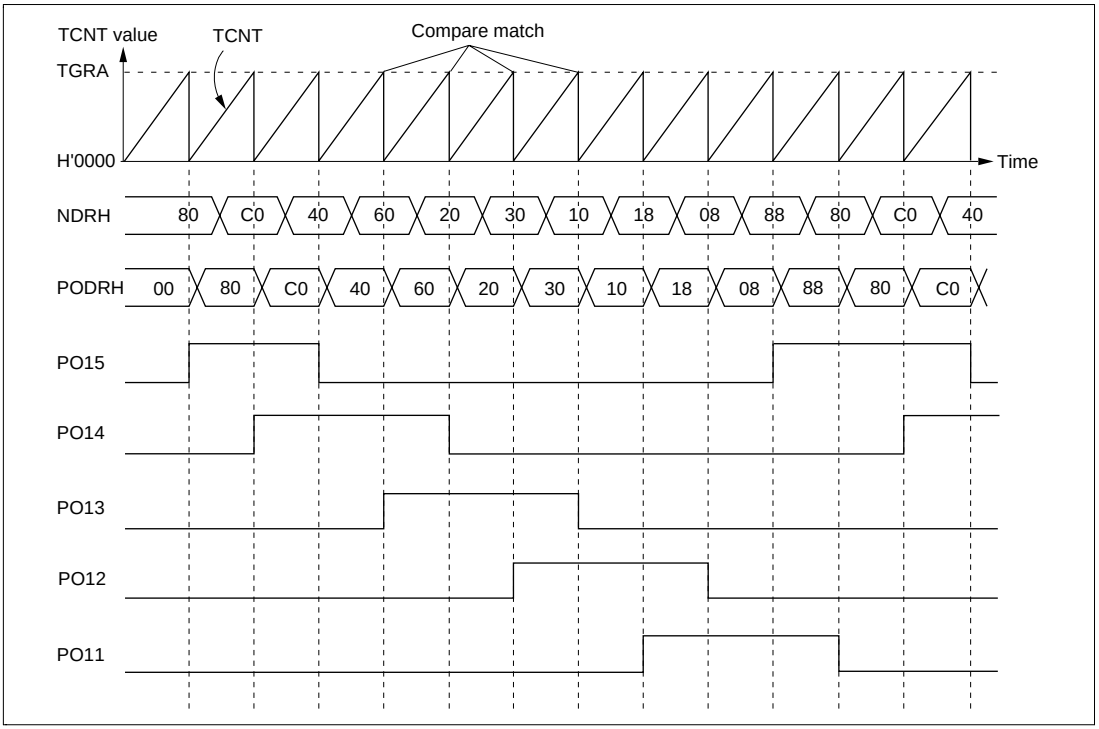


Figure 11-5 Normal Pulse Output Example (Five-Phase Pulse Output)

[1] Set up the TPU channel to be used as the output trigger channel so that TGRA is an output compare register and the counter will be cleared by compare match A. Set the trigger period in TGRA and set the TGIEA bit in TIER to 1 to enable the compare match A (TGIA) interrupt.

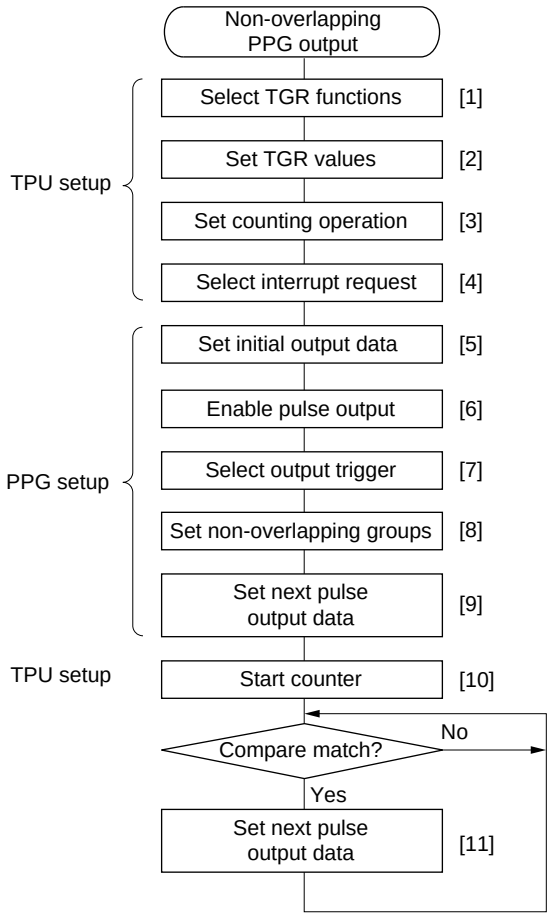
[2] Write H'F8 in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Write output data H'80 in NDRH.

[3] The timer counter in the TPU channel starts. When compare match A occurs, the NDRH contents are transferred to PODRH and output. The TGIA interrupt handling routine writes the next output data (H'C0) in NDRH.

[4] Five-phase overlapping pulse output (one or two phases active at a time) can be obtained subsequently by writing H'40, H'60, H'20, H'30. H'10, H'18, H'08, H'88... at successive TGIA interrupts. If the DTC is set for activation by this interrupt, pulse output can be obtained without imposing a load on the CPU.

11.3.4 Non-Overlapping Pulse Output

Sample Setup Procedure for Non-Overlapping Pulse Output: Figure 11-6 shows a sample procedure for setting up non-overlapping pulse output.



[1] Set TIOR to make TGRA and TGRB an output compare registers (with output disabled)

[2] Set the pulse output trigger period in TGRB and the non-overlap margin in TGRA.

[3] Select the counter clock source with bits TPSC2 to TPSC0 in TCR. Select the counter clear source with bits CCLR1 and CCLR0.

[4] Enable the TGIA interrupt in TIER. The DTC can also be set up to transfer data to NDR.

[5] Set the initial output values in PODR.

[6] Set the DDR and NDER bits for the pins to be used for pulse output to 1.

[7] Select the TPU compare match event to be used as the pulse output trigger in PCR.

[8] In PMR, select the groups that will operate in non-overlap mode.

[9] Set the next pulse output values in NDR.

[10] Set the CST bit in TSTR to 1 to start the TCNT counter.

[11] At each TGIA interrupt, set the next output values in NDR.

Figure 11-6 Setup Procedure for Non-Overlapping Pulse Output (Example)

Example of Non-Overlapping Pulse Output (Example of Four-Phase Complementary Non-Overlapping Output): Figure 11-7 shows an example in which pulse output is used for four-phase complementary non-overlapping pulse output.

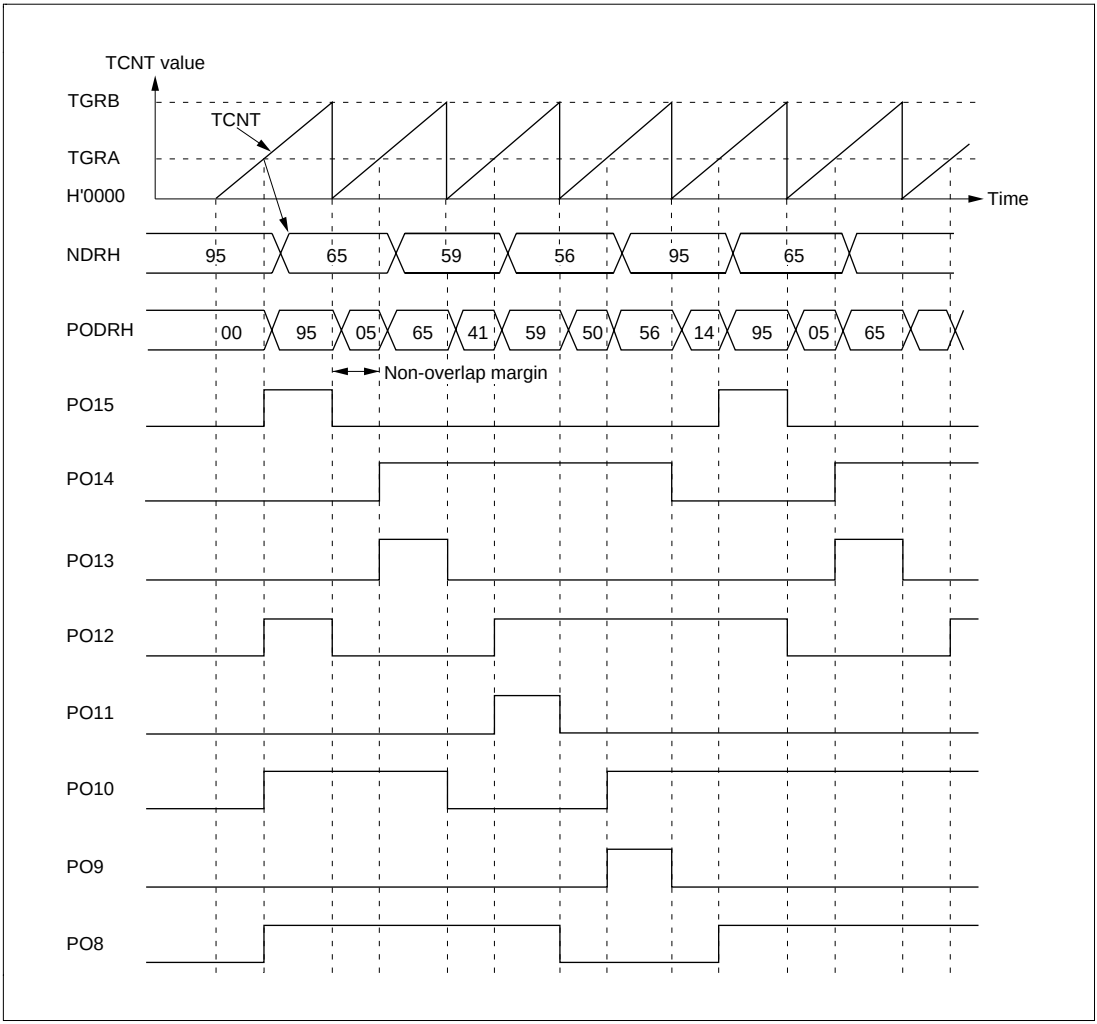


Figure 11-7 Non-Overlapping Pulse Output Example (Four-Phase Complementary)

[1] Set up the TPU channel to be used as the output trigger channel so that TGRA and TGRB are output compare registers. Set the trigger period in TGRB and the non-overlap margin in TGRA, and set the counter to be cleared by compare match B. Set the TGIEA bit in TIER to 1 to enable the TGIA interrupt.

[2] Write H'FF in P1DDR and NDERH, and set the G3CMS1, G3CMS0, G2CMS1, and G2CMS0 bits in PCR to select compare match in the TPU channel set up in the previous step to be the output trigger. Set the G3NOV and G2NOV bits in PMR to 1 to select non-overlapping output. Write output data H'95 in NDRH.

[3] The timer counter in the TPU channel starts. When a compare match with TGRB occurs, outputs change from 1 to 0. When a compare match with TGRA occurs, outputs change from 0 to 1 (the change from 0 to 1 is delayed by the value set in TGRA). The TGIA interrupt handling routine writes the next output data (H'65) in NDRH.

[4] Four-phase complementary non-overlapping pulse output can be obtained subsequently by writing H'59, H'56, H'95... at successive TGIA interrupts. If the DTC is set for activation by this interrupt, pulse output can be obtained without imposing a load on the CPU.

11.3.5 Inverted Pulse Output

If the G3INV, G2INV, G1INV, and G0INV bits in PMR are cleared to 0, values that are the inverse of the PODR contents can be output.

Figure 11-8 shows the outputs when G3INV and G2INV are cleared to 0, in addition to the settings of figure 11-7.

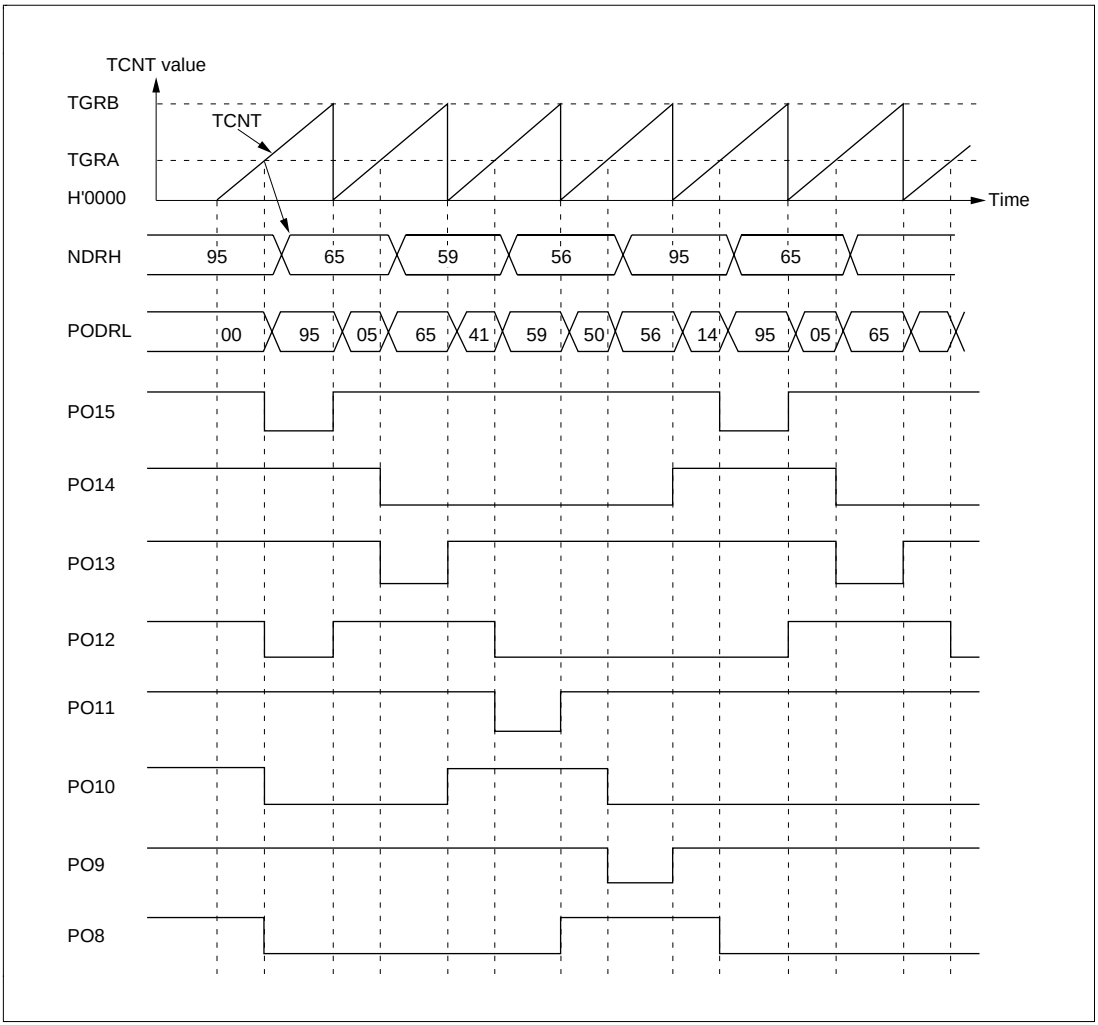


Figure 11-8 Inverted Pulse Output (Example)

11.3.6 Pulse Output Triggered by Input Capture

Pulse output can be triggered by TPU input capture as well as by compare match. If TGRA functions as an input capture register in the TPU channel selected by PCR, pulse output will be triggered by the input capture signal.

Figure 11-9 shows the timing of this output.

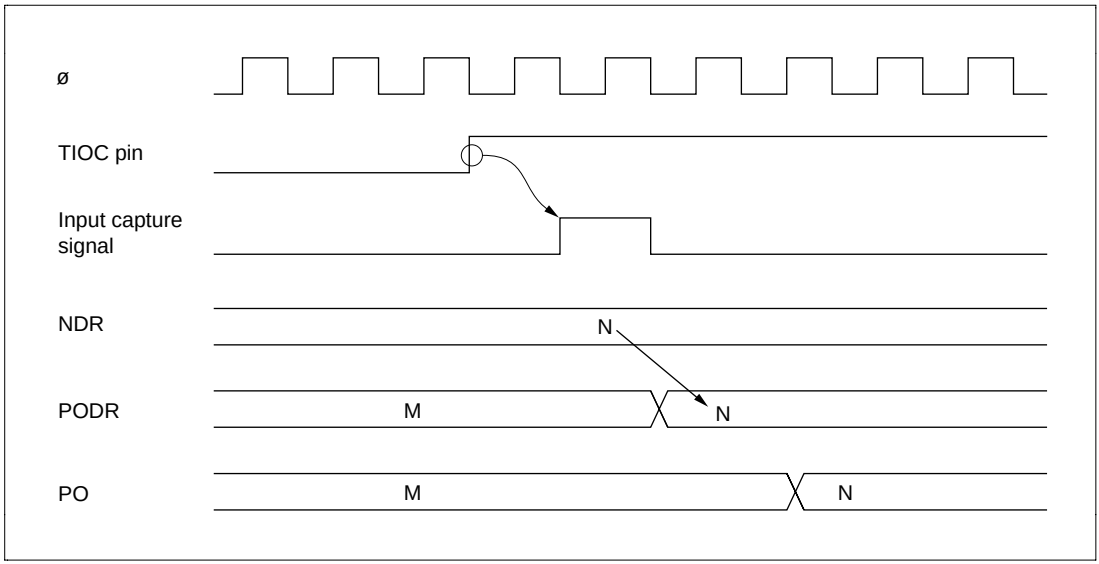


Figure 11-9 Pulse Output Triggered by Input Capture (Example)

11.4 Usage Notes

Operation of Pulse Output Pins: Pins PO8 to PO15 are also used for other peripheral functions such as the TPU. When output by another peripheral function is enabled, the corresponding pins cannot be used for pulse output. Note, however, that data transfer from NDR bits to PODR bits takes place, regardless of the usage of the pins.

Pin functions should be changed only under conditions in which the output trigger event will not occur.

Note on Non-Overlapping Output: During non-overlapping operation, the transfer of NDR bit values to PODR bits takes place as follows.

- NDR bits are always transferred to PODR bits at compare match A.
- At compare match B, NDR bits are transferred only if their value is 0. Bits are not transferred if their value is 1.

Figure 11-10 illustrates the non-overlapping pulse output operation.

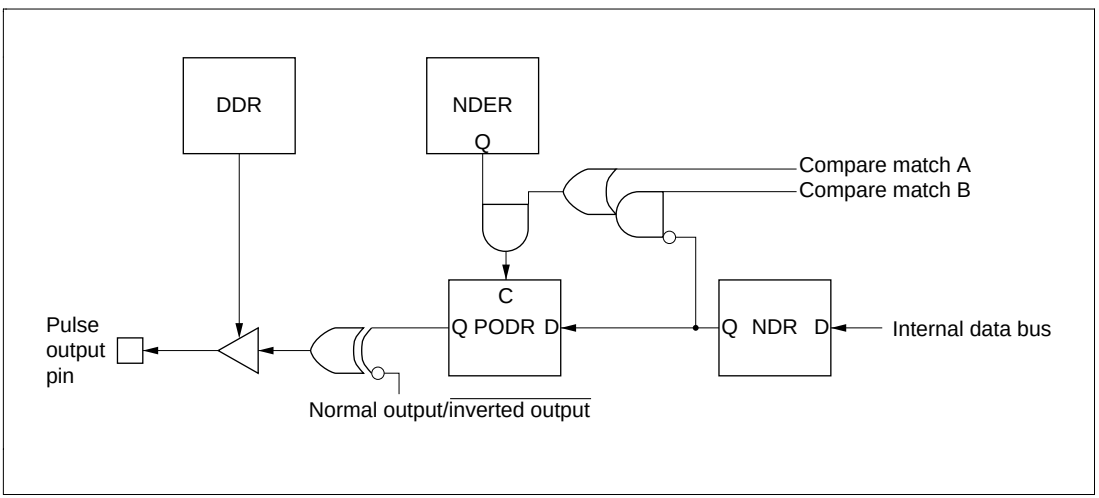


Figure 11-10 Non-Overlapping Pulse Output

Therefore, 0 data can be transferred ahead of 1 data by making compare match B occur before compare match A. The NDR contents should not be altered during the interval from compare match B to compare match A (the non-overlap margin).

This can be accomplished by having the TGIA interrupt handling routine write the next data in NDR, or by having the TGIA interrupt activate the DTC. Note, however, that the next data must be written before the next compare match B occurs.

Figure 11-11 shows the timing of this operation.

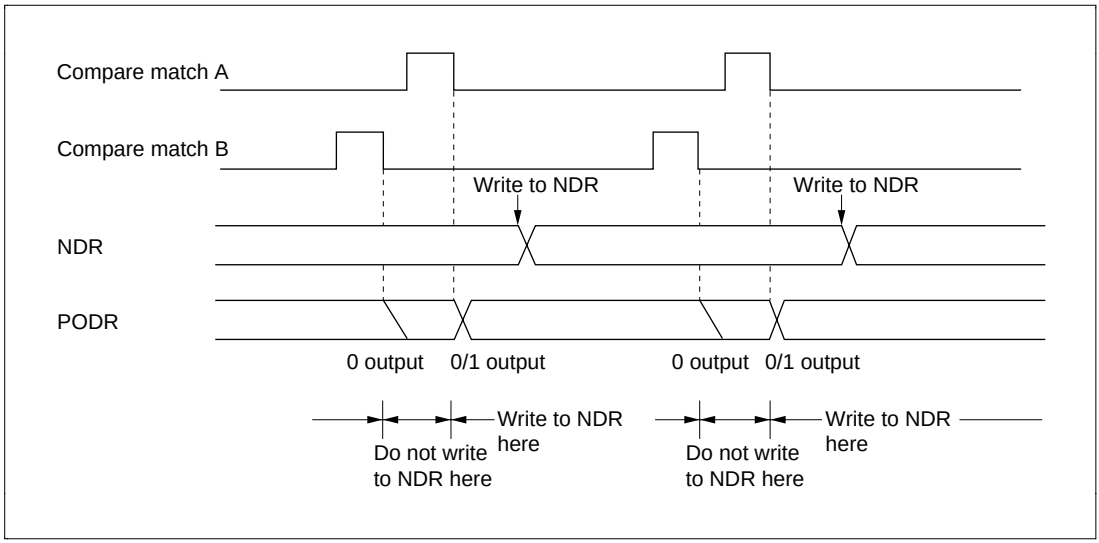


Figure 11-11 Non-Overlapping Operation and NDR Write Timing

Section 12 Watchdog Timer

12.1 Overview

The chip has two channel inbuilt watchdog timers (WDT0/WDT1*[1]). The WDT can also generate an internal reset signal for the chip if a system crash prevents the CPU from writing to the timer counter, allowing it to overflow.

When this watchdog function is not needed, the WDT can be used as an interval timer. In interval timer operation, an interval timer interrupt is generated each time the counter overflows.

12.1.1 Features

WDT features are listed below.

- Switchable between watchdog timer mode and interval timer mode
- An internal reset can be issued if the timer counter overflows.
  In the watchdog timer mode, the WDT can generate an internal reset.
- Interrupt generation when in interval timer mode
  If the counter overflows, the WDT generates an interval timer interrupt.
- WDT0 and WDT1 respectively allow eight and sixteen types of counter input clock to be selected
  The maximum interval of the WDT is given as a system clock cycle $\times$ 131072 $\times$ 256.
  A subclock*[2] may be selected for the input counter of WDT1.
  Where a subclock is selected, the maximum interval is given as a subclock cycle $\times$ 256 $\times$ 256.

Notes: *1 Versions of the H8S/2636 and H8S/2638 other than the U-mask and W-mask versions have an on-chip single-channel watchdog timer (WDT0). (The U-mask and W-mask versions of the H8S/2636, H8S/2638, and H8S/2639 support use of two watchdog timer channels (WDT0 and WDT1).)

*2 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only of the H8S/2636, H8S/2638, and H8S/2639. See section 22A.7, Subclock Oscillator, for the method of fixing pins when OSC1 and OSC2 are not used. The H8S/2639 has no OSC1 and OSC2 pins.

12.1.2 Block Diagram

Figures 12-1 (a) and 12-1 (b) show block diagrams of the WDT.

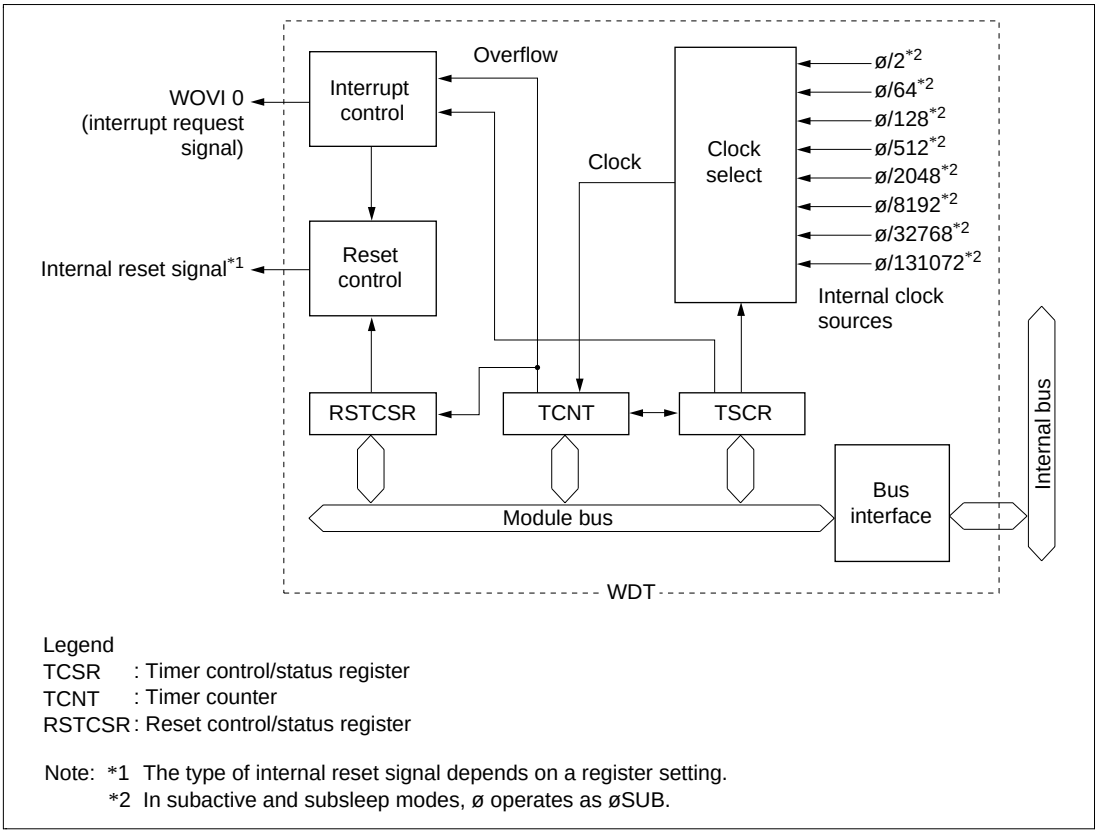
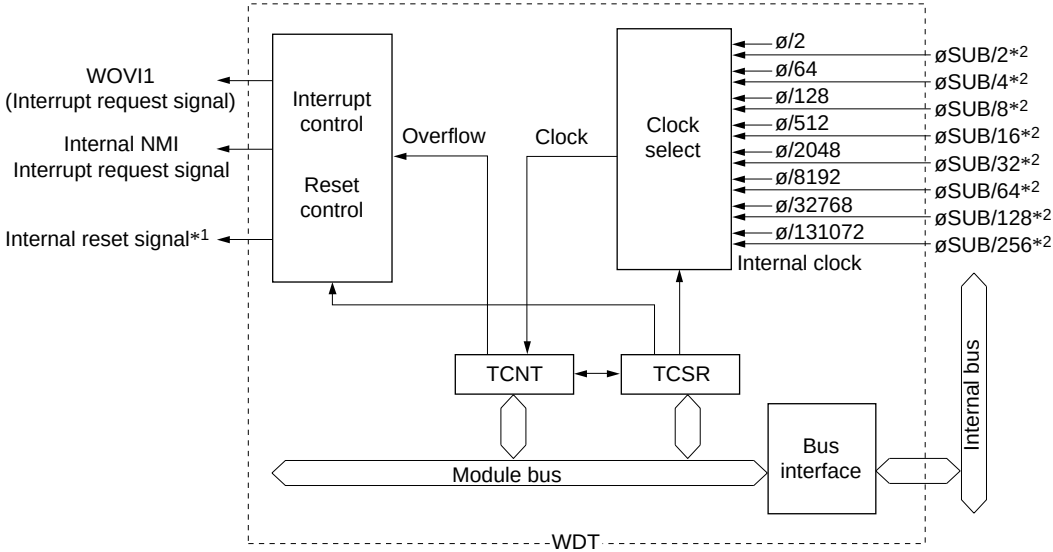


Figure 12-1 (a) Block Diagram of WDT0



Legend:
TCSR : Timer control/status register
TCNT : Timer counter

Notes: *1 An internal reset signal can be generated by setting the register
The reset thus generated is a reset
*2 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only of the H8S/2636, H8S/2638, and H8S/2639 only.

Figure 12-1 (b) Block Diagram of WDT1

12.1.3 Pin Configuration

There are no pins related to the WDT.

12.1.4 Register Configuration

The WDT has five registers, as summarized in table 12-1. These registers control clock selection, WDT mode switching, and the reset signal.

Table 12-1 WDT Registers

					Address*[1]	
Channel	**Name**	**Abbreviation**	**R/W**	**Initial Value**	**Write*[2]**	**Read**
0	Timer control/status register 0	TCSR0	R/(W)*[3]	H'18	H'FF74	H'FF74
	Timer counter 0	TCNT0	R/W	H'00	H'FF74	H'FF75
	Reset control/status register	RSTCSR0	R/(W)*[3]	H'1F	H'FF76	H'FF77
1	Timer control/status register 1	TCSR1	R/(W)*[3]	H'00	H'FFA2	H'FFA2
	Timer counter 1	TCNT1	R/W	H'00	H'FFA2	H'FFA3

Notes: *1 Lower 16 bits of the address.
*2 For details of write operations, see section 12.2.4, Notes on Register Access.
*3 Only a write of 0 is permitted to bit 7, to clear the flag.

12.2 Register Descriptions

12.2.1 Timer Counter (TCNT)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TCNT is an 8-bit readable/writable* up-counter.

When the TME bit is set to 1 in TCSR, TCNT starts counting pulses generated from the internal clock source selected by bits CKS2 to CKS0 in TCSR. When the count overflows (changes from H'FF to H'00), an interval timer interrupt (WOVI) is generated, depending on the mode selected by the WT/$\overline{IT}$ bit in TCSR.

TCNT is initialized to H'00 by a reset, in hardware standby mode, or when the TME bit is cleared to 0. It is not initialized in software standby mode.

Note: * TCNT is write-protected by a password to prevent accidental overwriting. For details see section 12.2.4, Notes on Register Access.

12.2.2 Timer Control/Status Register (TCSR)

TCSR0

Bit	:	7	6	5	4	3	2	1	0
		OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	:	0	0	0	1	1	0	0	0
R/W	:	R/(W)*	R/W	R/W	—	—	R/W	R/W	R/W

Note: * Only a 0 may be written to this bit to clear the flag.

TCSR1

Bit	:	7	6	5	4	3	2	1	0
		OVF	WT/$\overline{IT}$	TME	PSS*[2]	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*[1]	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: *1 Only a 0 may be written to this bit to clear the flag.

*2 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only of the H8S/2636, H8S/2638, and H8S/2639 only.

TCSR is an 8-bit readable/writable* register. Its functions include selecting the clock source to be input to TCNT, and the timer mode.

TCSR0 (TCSR1) is initialized to H'18 (H'00) by a reset and in hardware standby mode. It is not initialized in software standby mode.

Note: * TCSR is write-protected by a password to prevent accidental overwriting. For details see section 12.2.4, Notes on Register Access.

Bit 7—Overflow Flag (OVF): Indicates that TCNT has overflowed from H'FF to H'00.

Bit 7 OVF	Description	
0	[Clearing conditions]	(Initial value)
	• Cleared when 0 is written to the TME bit (Only applies to WDT1)	
	• Cleared by reading TCSR* when OVF = 1, then writing 0 to OVF	
1	[Setting condition]	
	When TCNT overflows (changes from H'FF to H'00) When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset.	

Note: * When interval timer interrupts are disabled and OVF is polled, read the OVF = 1 state at least twice.

Bit 6—Timer Mode Select (WT/$\overline{IT}$): Selects whether the WDT is used as a watchdog timer or interval timer. This selection determines whether WDT0 issues an internal reset when TCNT overflows while bit RSTE of the reset control/status register (RSTCSR) is set to 1. In the interval timer mode, WDT0 sends a WOVI interrupt request to the CPU. WDT1, on the other hand, requests a reset or an NMI interrupt from the CPU if the watchdog timer mode is chosen, whereas it requests a WOVI interrupt from the CPU if the interval timer mode is chosen.

WDT0 Mode Select

TCSR0 WT/$\overline{IT}$	Description	
0	Interval timer mode: WDT0 requests an interval timer interrupt (WOVI) from the CPU when the TCNT overflows.	(Initial value)
1	Watchdog timer mode: A reset is issued when the TCNT overflows if the RSTE bit of RSTCSR is set to 1.*	

Note: * For details see section 12.2.3, Reset Control/Status Register (RSTCSR).

WDT1 Mode Select

TCSR1 WT/$\overline{IT}$	Description	
0	Interval timer mode: WDT1 requests an interval timer interrupt (WOVI) from the CPU when the TCNT overflows.	(Initial value)
1	Watchdog timer mode: WDT1 requests a reset or an NMI interrupt from the CPU when the TCNT overflows.	

Bit 5—Timer Enable (TME): Selects whether TCNT runs or is halted.

Bit 5 TME	Description	
0	TCNT is initialized to H'00 and halted	(Initial value)
1	TCNT counts	

WDT0 TCSR Bit 4—Reserved Bit: A read operation on this bit always causes a 1 to be read out. Every write operation on this bit is invalidated.

WDT1 TCSR Bit 4—Prescaler Select (PSS): This bit is used to select an input clock source for the TCNT of WDT1.

See the descriptions of Clock Select 2 to 0 for details.

WDT1 TCSR Bit 4 PSS	Description	
0	The TCNT counts frequency-division clock pulses of the ø based prescaler (PSM).	(Initial value)
1	The TCNT counts frequency-division clock pulses of the ø SUB*-based prescaler (PSS).	

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are not available in the HD64F2636F and HD64F2638F, but are available in the HD64F2636UF and HD64F2638UF (U-mask version).

WDT0 TCSR Bit 3—Reserved Bit: A read operation on this bit always causes a 1 to be read out. Every write operation on this bit is invalidated.

WDT1 TCSR Bit 3—Reset or NMI (RST/$\overline{\text{NMI}}$): This bit is used to choose between an internal reset request and an NMI request when the TCNT overflows during the watchdog timer mode.

Bit 3 RST/$\overline{\text{NMI}}$	Description	
0	NMI request.	(Initial value)
1	Internal reset request.	

Bits 2 to 0: Clock Select 2 to 0 (CKS2 to CKS0): These bits select one of eight internal clock sources, obtained by dividing the system clock (ø) or subclock* (ø SUB), for input to TCNT.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions, and in them the PSS bit is reserved. Only 0 should be written to this bit.

WDT0 Input Clock Select

			Description	
Bit 2 CKS2	**Bit 1 CKS1**	**Bit 0 CKS0**	**Clock**	**Overflow Period*[1] (where ø = 20 MHz)**
0	0	0	ø/2*[2] (initial value)	25.6 µs
		1	ø/64*[2]	819.2 µs
	1	0	ø/128*[2]	1.6 ms
		1	ø/512*[2]	6.6 ms
1	0	0	ø/2048*[2]	26.2 ms
		1	ø/8192*[2]	104.9 ms
	1	0	ø/32768*[2]	419.4 ms
		1	ø/131072*[2]	1.68 s

Notes: *1 An overflow period is the time interval between the start of counting up from H'00 on the TCNT and the occurrence of a TCNT overflow.

*2 In subactive and subsleep modes, ø operates as øSUB.

WDT1 Input Clock Select

				Description	
Bit 4 PSS	**Bit 2 CKS2**	**Bit 1 CKS1**	**Bit 0 CKS0**	**Clock**	**Overflow Period*[1] (where ø = 20 MHz) (where ø SUB*[2] = 32.768 kHz)**
0	0	0	0	ø/2 (initial value)	25.6 µs
			1	ø/64	819.2 µs
		1	0	ø/128	1.6 ms
			1	ø/512	6.6 ms
	1	0	0	ø/2048	26.2 ms
			1	ø/8192	104.9 ms
		1	0	ø/32768	419.4 ms
0	1	1	1	ø/131072	1.68 s
1	0	0	0	øSUB/2*[2]	15.6 ms
			1	øSUB/4*[2]	31.3 ms
		1	0	øSUB/8*[2]	62.5 ms
			1	øSUB/16*[2]	125 ms
	1	0	0	øSUB/32*[2]	250 ms
			1	øSUB/64*[2]	500 ms
		1	0	øSUB/128*[2]	1 s
			1	øSUB/256*[2]	2 s

Notes: *1 An overflow period is the time interval between the start of counting up from H'00 on the TCNT and the occurrence of a TCNT overflow.

*2 Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

12.2.3 Reset Control/Status Register (RSTCSR)

Bit	:	7	6	5	4	3	2	1	0
		WOVF	RSTE	RSTS	—	—	—	—	—
Initial value	:	0	0	0	1	1	1	1	1
R/W	:	R/(W)*	R/W	R/W	—	—	—	—	—

Note: * Can only be written with 0 for flag clearing.

RSTCSR is an 8-bit readable/writable* register that controls the generation of the internal reset signal when TCNT overflows, and selects the type of internal reset signal.

RSTCSR is initialized to H'1F by a reset signal from the $\overline{\text{RES}}$ pin, but not by the WDT internal reset signal caused by overflows.

Note: * RSTCSR is write-protected by a password to prevent accidental overwriting. For details see section 12.2.4, Notes on Register Access.

Bit 7—Watchdog Overflow Flag (WOVF): Indicates that TCNT has overflowed (changed from H'FF to H'00) during watchdog timer operation. This bit is not set in interval timer mode.

Bit 7 WOVF	Description	
0	[Clearing condition]	(Initial value)
	Cleared by reading TCSR when WOVF = 1, then writing 0 to WOVF	
1	[Setting condition]	
	Set when TCNT overflows (changed from H'FF to H'00) during watchdog timer operation	

Bit 6—Reset Enable (RSTE): Specifies whether or not a reset signal is generated in the H8S/2636 if TCNT overflows during watchdog timer operation.

Bit 6 RSTE	Description	
0	Reset signal is not generated if TCNT overflows*	(Initial value)
1	Reset signal is generated if TCNT overflows	

Note: * The modules within the chip are not reset, but TCNT and TCSR within the WDT are reset.

Bit 5—Reset Select (RSTS): Selects the type of internal reset generated if TCNT overflows during watchdog timer operation.

For details of the types of reset, see section 4, Exception Handling.

Bit 5 RSTS	Description	
0	Reset	(Initial value)
1	Do not set	

Bits 4 to 0—Reserved: Always read as 1 and cannot be modified.

12.2.4 Notes on Register Access

The watchdog timer's TCNT, TCSR, and RSTCSR registers differ from other registers in being more difficult to write to. The procedures for writing to and reading these registers are given below.

Writing to TCNT and TCSR: These registers must be written to by a word transfer instruction. They cannot be written to with byte instructions.

Figure 12-2 shows the format of data written to TCNT and TCSR. TCNT and TCSR both have the same write address. For a write to TCNT, the upper byte of the written word must contain H'5A and the lower byte must contain the write data. For a write to TCSR, the upper byte of the written word must contain H'A5 and the lower byte must contain the write data. This transfers the write data from the lower byte to TCNT or TCSR.

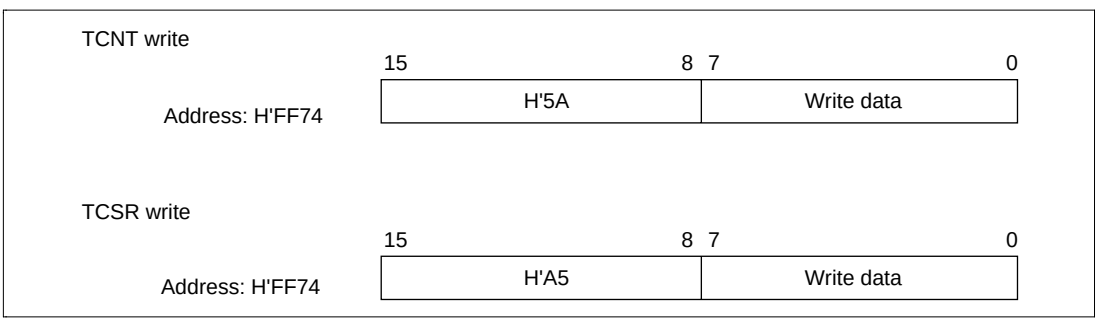


Figure 12-2 Format of Data Written to TCNT and TCSR (WDT0)

Writing to RSTCSR: RSTCSR must be written to by word transfer instruction to address H'FF76. It cannot be written to with byte instructions.

Figure 12-3 shows the format of data written to RSTCSR. The method of writing 0 to the WOVF bit differs from that for writing to the RSTE bit.

To write 0 to the WOVF bit, the write data must have H'A5 in the upper byte and H'00 in the lower byte. This clears the WOVF bit to 0, but has no effect on the RSTE bit. To write to the RSTE bit, the upper byte must contain H'5A and the lower byte must contain the write data. This writes the values in bits 6 and 5 of the lower byte into the RSTE bit, but has no effect on the WOVF bit.

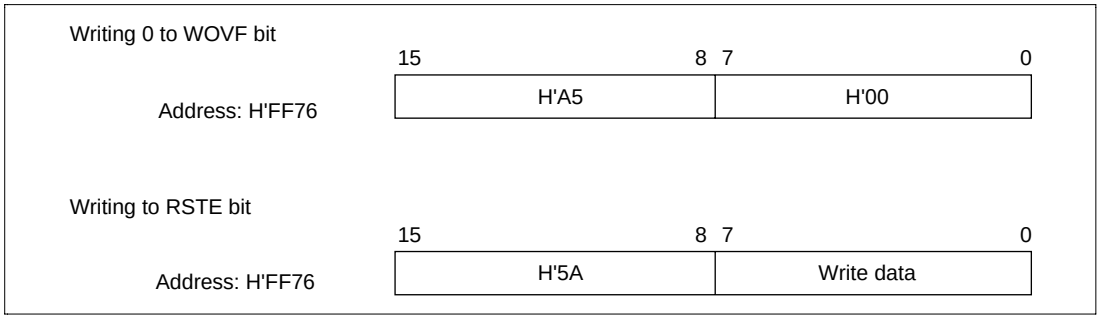


Figure 12-3 Format of Data Written to RSTCSR (WDT0)

Reading TCNT, TCSR, and RSTCSR (WDT0): These registers are read in the same way as other registers. The read addresses are H'FF74 for TCSR, H'FF75 for TCNT, and H'FF77 for RSTCSR.

12.3 Operation

12.3.1 Watchdog Timer Operation

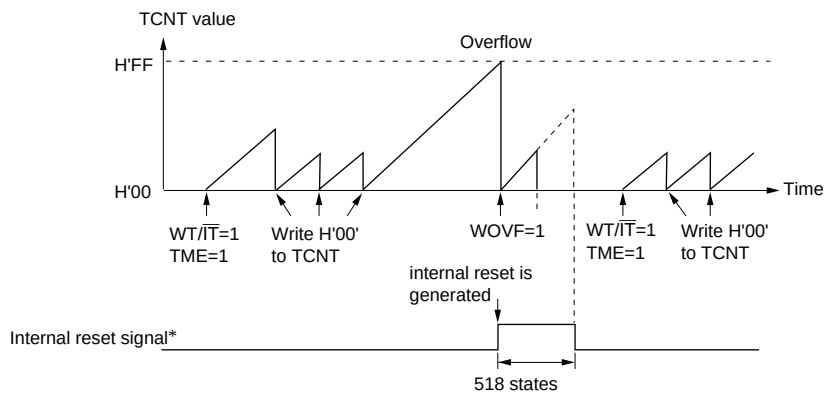
To use the WDT as a watchdog timer, set the WT/$\overline{IT}$ bit in TCSR and the TME bit to 1. Software must prevent TCNT overflows by rewriting the TCNT value (normally by writing H'00) before overflow occurs. This ensures that TCNT does not overflow while the system is operating normally. If TCNT overflows without being rewritten because of a system malfunction or other error, an internal reset is issued, in the case of WDT0, if the RSTE bit in RSTCSR is set to 1.

If a reset caused by a signal input to the $\overline{RES}$ pin occurs at the same time as a reset caused by a WDT overflow, the $\overline{RES}$ pin reset has priority and the WOVF bit in RSTCSR is cleared to 0.

In the case of WDT1, the chip is reset, or an NMI interrupt request is generated, for 516 system clock periods (516ø) (515 or 516 clock periods when the clock source is ø/SUB* (PSS = 1)). This is illustrated in figure 12-4 (b).

An NMI request from the watchdog timer and an interrupt request from the NMI pin are both treated as having the same vector. So, avoid handling an NMI request from the watchdog timer and an interrupt request from the NMI pin at the same time.

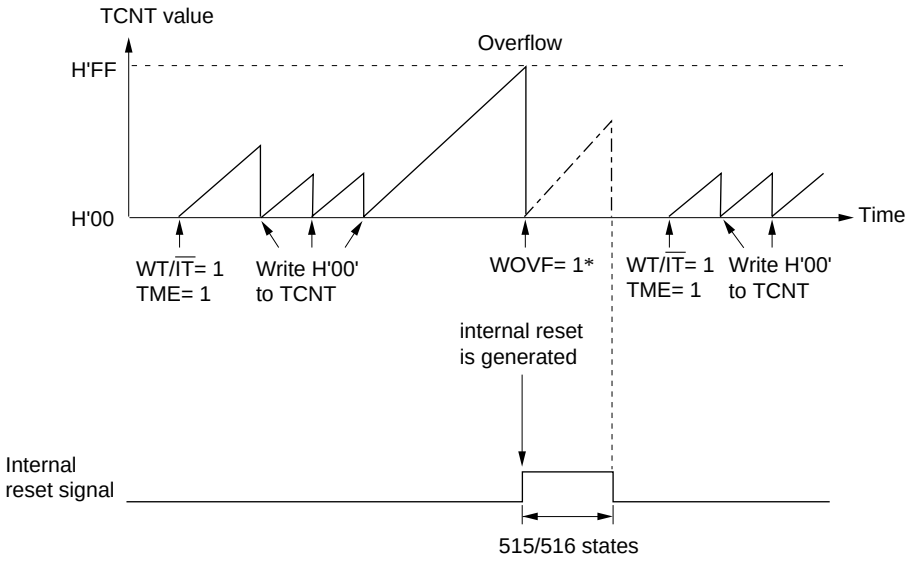
Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only. These functions cannot be used with the other versions.



Legend
WT/$\overline{IT}$: Timer mode select bit
TME : Timer enable bit

Note: * The internal reset signal is generated only if the RSTE bit is set to 1.

Figure 12-4 (a) WDT0 Watchdog Timer Operation



Legend

WT/$\overline{IT}$: Timer mode select bit
TME : Timer enable bit

Note: * The WOVF bit is set to 1 and then cleared to 0 by an internal reset.

Figure 12-4 (b) WDT1 Watchdog Timer Operation

12.3.2 Interval Timer Operation

To use the WDT as an interval timer, clear the WT/$\overline{IT}$ bit in TCSR to 0 and set the TME bit to 1. An interval timer interrupt (WOVI) is generated each time TCNT overflows, provided that the WDT is operating as an interval timer, as shown in figure 12-5. This function can be used to generate interrupt requests at regular intervals.

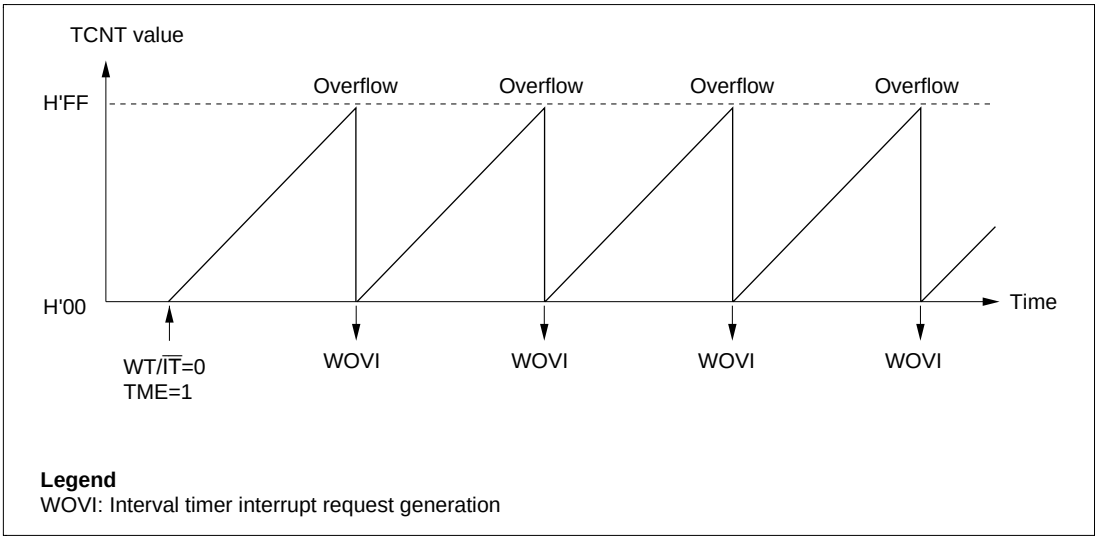


Figure 12-5 Interval Timer Operation

12.3.3 Timing of Setting Overflow Flag (OVF)

The OVF flag is set to 1 if TCNT overflows during interval timer operation. At the same time, an interval timer interrupt (WOVI) is requested. This timing is shown in figure 12-6.

With WDT1, the OVF bit of the TCSR is set to 1 and a simultaneous NMI interrupt is requested when the TCNT overflows if the NMI request has been chosen in the watchdog timer mode.

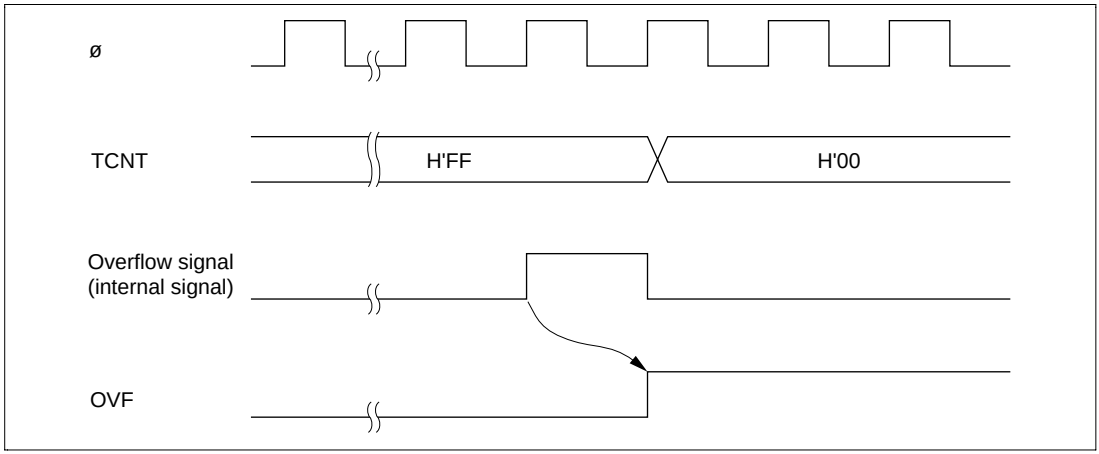


Figure 12-6 Timing of Setting of OVF

12.3.4 Timing of Setting of Watchdog Timer Overflow Flag (WOVF)

In the WDT0, the WOVF flag is set to 1 if TCNT overflows during watchdog timer operation. If TCNT overflows while the RSTE bit in RSTCSR is set to 1, an internal reset signal is generated for the entire chip. Figure 12-7 shows the timing in this case.

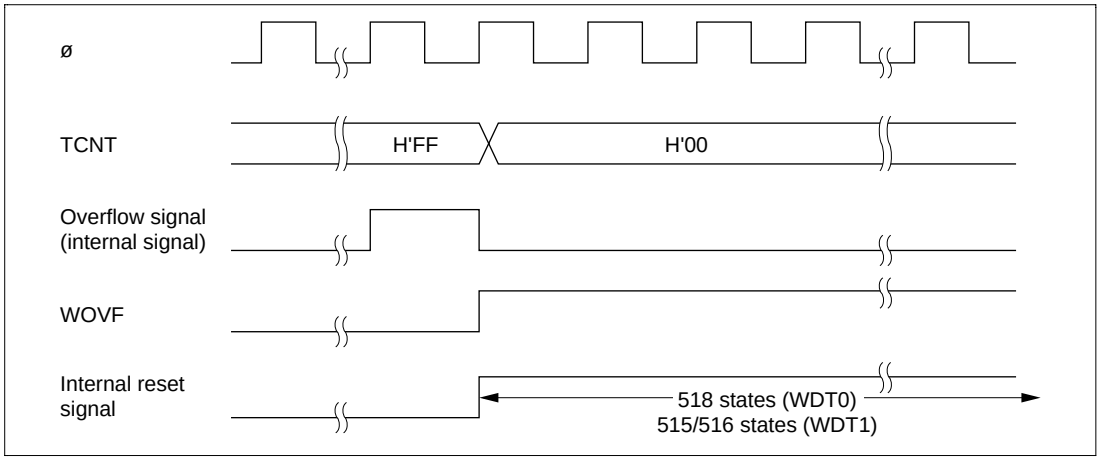


Figure 12-7 Timing of Setting of WOVF

12.4 Interrupts

During interval timer mode operation, an overflow generates an interval timer interrupt (WOVI). The interval timer interrupt is requested whenever the OVF flag is set to 1 in TCSR. OVF must be cleared to 0 in the interrupt handling routine.

If an NMI request has been chosen in the watchdog timer mode, an NMI request is generated when a TCNT overflow occurs.

12.5 Usage Notes

12.5.1 Contention between Timer Counter (TCNT) Write and Increment

If a timer counter clock pulse is generated during the T_2 state of a TCNT write cycle, the write takes priority and the timer counter is not incremented. Figure 12-8 shows this operation.

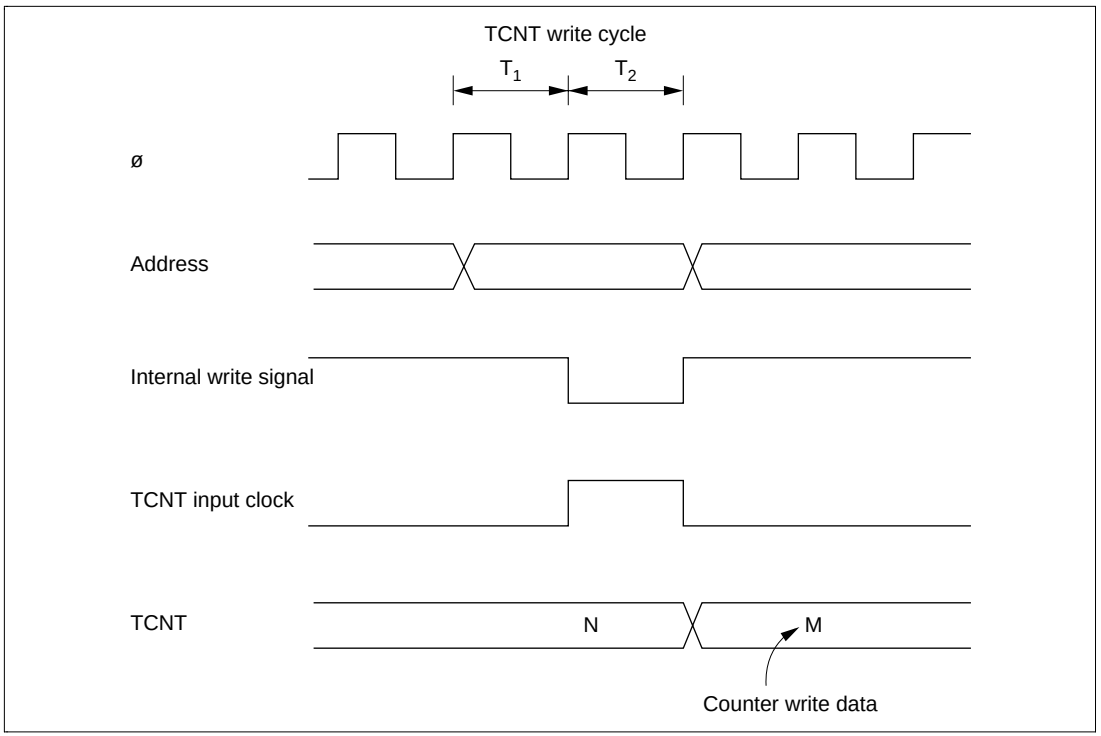


Figure 12-8 Contention between TCNT Write and Increment

12.5.2 Changing Value of PSS* and CKS2 to CKS0

If bits PSS and CKS2 to CKS0 in TCSR are written to while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before changing the value of bits PSS and CKS2 to CKS0.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

12.5.3 Switching between Watchdog Timer Mode and Interval Timer Mode

If the mode is switched from watchdog timer to interval timer, or vice versa, while the WDT is operating, errors could occur in the incrementation. Software must stop the watchdog timer (by clearing the TME bit to 0) before switching the mode.

12.5.4 Internal Reset in Watchdog Timer Mode

The chip is not reset internally if TCNT overflows while the RSTE bit is cleared to 0 during watchdog timer operation, but TCNT and TSCR of the WDT are reset.

12.5.5 OVF Flag Clearing in Interval Timer Mode

If conflict occurs between OVF flag clearing and OVF flag reading in interval timer mode, the flag may not be cleared by writing 0 to OVF even though the OVF = 1 state has been read. When interval timer interrupts are disabled and the OVF flag is polled, for instance, and there is a possibility of conflict between OVF flag setting and reading, the OVF = 1 state should be read at least twice before writing 0 to OVF in order to clear the flag.

Section 13 Serial Communication Interface (SCI)

13.1 Overview

The chip is equipped with 3 independent serial communication interface (SCI) channels. The SCI can handle both asynchronous and clocked synchronous serial communication. A function is also provided for serial communication between processors (multiprocessor communication function).

13.1.1 Features

SCI features are listed below.

- Choice of asynchronous or clocked synchronous serial communication mode

  Asynchronous mode

  — Serial data communication executed using asynchronous system in which synchronization is achieved character by character

  Serial data communication can be carried out with standard asynchronous communication chips such as a Universal Asynchronous Receiver/Transmitter (UART) or Asynchronous Communication Interface Adapter (ACIA)

  — A multiprocessor communication function is provided that enables serial data communication with a number of processors

  — Choice of 12 serial data transfer formats

Data length	:	7 or 8 bits
Stop bit length	:	1 or 2 bits
Parity	:	Even, odd, or none
Multiprocessor bit	:	1 or 0

  — Receive error detection : Parity, overrun, and framing errors

  — Break detection : Break can be detected by reading the RxD pin level directly in case of a framing error

  Clocked Synchronous mode

  — Serial data communication synchronized with a clock

  Serial data communication can be carried out with other chips that have a synchronous communication function

  — One serial data transfer format

Data length	:	8 bits

  — Receive error detection : Overrun errors detected

- Full-duplex communication capability
  - — The transmitter and receiver are mutually independent, enabling transmission and reception to be executed simultaneously
  - — Double-buffering is used in both the transmitter and the receiver, enabling continuous transmission and continuous reception of serial data
- Choice of LSB-first or MSB-first transfer
  - — Can be selected regardless of the communication mode* (except in the case of asynchronous mode 7-bit data)

    Note: * Descriptions in this section refer to LSB-first transfer.
- On-chip baud rate generator allows any bit rate to be selected
- Choice of serial clock source: internal clock from baud rate generator or external clock from SCK pin
- Four interrupt sources
  - — Four interrupt sources — transmit-data-empty, transmit-end, receive-data-full, and receive error — that can issue requests independently
  - — The transmit-data-empty interrupt and receive data full interrupts can activate the data transfer controller (DTC) to execute data transfer
- Module stop mode can be set
  - — As the initial setting, SCI operation is halted. Register access is enabled by exiting module stop mode.

13.1.2 Block Diagram

Figure 13-1 shows a block diagram of the SCI.

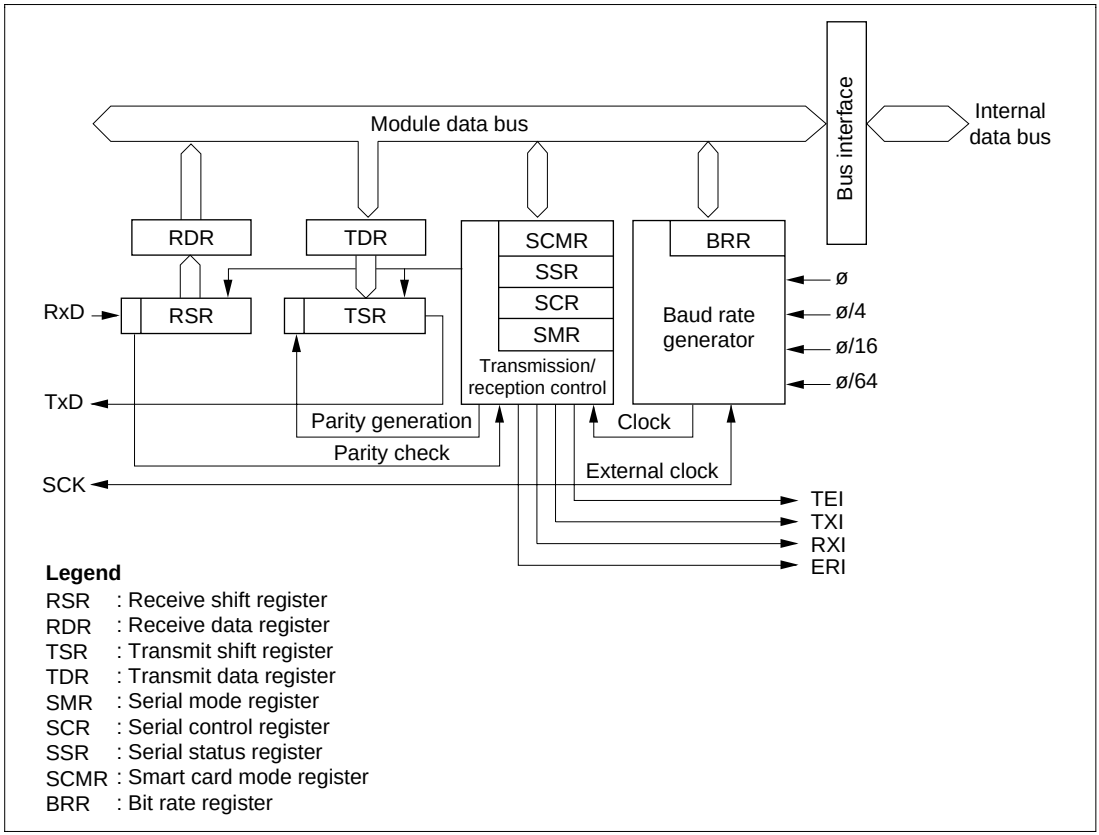


Figure 13-1 Block Diagram of SCI

13.1.3 Pin Configuration

Table 13-1 shows the serial pins for each SCI channel.

Table 13-1 SCI Pins

Channel	Pin Name	Symbol*	I/O	Function
0	Serial clock pin 0	SCK0	I/O	SCI0 clock input/output
	Receive data pin 0	RxD0	Input	SCI0 receive data input
	Transmit data pin 0	TxD0	Output	SCI0 transmit data output
1	Serial clock pin 1	SCK1	I/O	SCI1 clock input/output
	Receive data pin 1	RxD1	Input	SCI1 receive data input
	Transmit data pin 1	TxD1	Output	SCI1 transmit data output
2	Serial clock pin 2	SCK2	I/O	SCI2 clock input/output
	Receive data pin 2	RxD2	Input	SCI2 receive data input
	Transmit data pin 2	TxD2	Output	SCI2 transmit data output

Note: * Pin names SCK, RxD, and TxD are used in the text for all channels, omitting the channel designation.

13.1.4 Register Configuration

The SCI has the internal registers shown in table 13-2. These registers are used to specify asynchronous mode or clocked synchronous mode, the data format , and the bit rate, and to control transmitter/receiver.

Table 13-2 SCI Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
0	Serial mode register 0	SMR0	R/W	H'00	H'FF78
	Bit rate register 0	BRR0	R/W	H'FF	H'FF79
	Serial control register 0	SCR0	R/W	H'00	H'FF7A
	Transmit data register 0	TDR0	R/W	H'FF	H'FF7B
	Serial status register 0	SSR0	R/(W)*[2]	H'84	H'FF7C
	Receive data register 0	RDR0	R	H'00	H'FF7D
	Smart card mode register 0	SCMR0	R/W	H'F2	H'FF7E
1	Serial mode register 1	SMR1	R/W	H'00	H'FF80
	Bit rate register 1	BRR1	R/W	H'FF	H'FF81
	Serial control register 1	SCR1	R/W	H'00	H'FF82
	Transmit data register 1	TDR1	R/W	H'FF	H'FF83
	Serial status register 1	SSR1	R/(W)*[2]	H'84	H'FF84
	Receive data register 1	RDR1	R	H'00	H'FF85
	Smart card mode register 1	SCMR1	R/W	H'F2	H'FF86
2	Serial mode register 2	SMR2	R/W	H'00	H'FF88
	Bit rate register 2	BRR2	R/W	H'FF	H'FF89
	Serial control register 2	SCR2	R/W	H'00	H'FF8A
	Transmit data register 2	TDR2	R/W	H'FF	H'FF8B
	Serial status register 2	SSR2	R/(W)*[2]	H'84	H'FF8C
	Receive data register 2	RDR2	R	H'00	H'FF8D
	Smart card mode register 2	SCMR2	R/W	H'F2	H'FF8E
All	Module stop control register B	MSTPCRB	R/W	H'FF	H'FDE9

Notes: *1 Lower 16 bits of the address.
*2 Can only be written with 0 for flag clearing.

13.2 Register Descriptions

13.2.1 Receive Shift Register (RSR)

Bit	:	7	6	5	4	3	2	1	0
R/W	:	—	—	—	—	—	—	—	—

RSR is a register used to receive serial data.

The SCI sets serial data input from the RxD pin in RSR in the order received, starting with the LSB (bit 0), and converts it to parallel data. When one byte of data has been received, it is transferred to RDR automatically.

RSR cannot be directly read or written to by the CPU.

13.2.2 Receive Data Register (RDR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R

RDR is a register that stores received serial data.

When the SCI has received one byte of serial data, it transfers the received serial data from RSR to RDR where it is stored, and completes the receive operation. After this, RSR is receive-enabled.

Since RSR and RDR function as a double buffer in this way, enables continuous receive operations to be performed.

RDR is a read-only register, and cannot be written to by the CPU.

RDR is initialized to H'00 by a reset, in standby mode, watch mode*, subactive mode*, and subsleep mode* or module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only. These functions cannot be used with the other versions.

13.2.3 Transmit Shift Register (TSR)

Bit	:	7	6	5	4	3	2	1	0
R/W	:	—	—	—	—	—	—	—	—

TSR is a register used to transmit serial data.

To perform serial data transmission, the SCI first transfers transmit data from TDR to TSR, then sends the data to the TxD pin starting with the LSB (bit 0).

When transmission of one byte is completed, the next transmit data is transferred from TDR to TSR, and transmission started, automatically. However, data transfer from TDR to TSR is not performed if the TDRE bit in SSR is set to 1.

TSR cannot be directly read or written to by the CPU.

13.2.4 Transmit Data Register (TDR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TDR is an 8-bit register that stores data for serial transmission.

When the SCI detects that TSR is empty, it transfers the transmit data written in TDR to TSR and starts serial transmission. Continuous serial transmission can be carried out by writing the next transmit data to TDR during serial transmission of the data in TSR.

TDR can be read or written to by the CPU at all times.

TDR is initialized to H'FF by a reset, in standby mode, watch mode*, subactive mode*, and subsleep mode* or module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

13.2.5 Serial Mode Register (SMR)

Bit	:	7	6	5	4	3	2	1	0
		C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SMR is an 8-bit register used to set the SCI's serial transfer format and select the baud rate generator clock source.

SMR can be read or written to by the CPU at all times.

SMR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Communication Mode (C/$\overline{A}$): Selects asynchronous mode or clocked synchronous mode as the SCI operating mode.

Bit 7

C/$\overline{A}$	Description	
0	Asynchronous mode	(Initial value)
1	Clocked synchronous mode	

Bit 6—Character Length (CHR): Selects 7 or 8 bits as the data length in asynchronous mode. In clocked synchronous mode, a fixed data length of 8 bits is used regardless of the CHR setting.

Bit 6

CHR	Description	
0	8-bit data	(Initial value)
1	7-bit data*	

Note: * When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it is not possible to choose between LSB-first or MSB-first transfer.

Bit 5—Parity Enable (PE): In asynchronous mode, selects whether or not parity bit addition is performed in transmission, and parity bit checking in reception. In clocked synchronous mode with a multiprocessor format, parity bit addition and checking is not performed, regardless of the PE bit setting.

Bit 5

PE	Description	
0	Parity bit addition and checking disabled	(Initial value)
1	Parity bit addition and checking enabled*	

Note:* When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

Bit 4—Parity Mode (O/$\overline{E}$): Selects either even or odd parity for use in parity addition and checking.

The O/$\overline{E}$ bit setting is only valid when the PE bit is set to 1, enabling parity bit addition and checking, in asynchronous mode. The O/$\overline{E}$ bit setting is invalid in clocked synchronous mode, when parity addition and checking is disabled in asynchronous mode, and when a multiprocessor format is used.

Bit 4

O/$\overline{E}$	Description	
0	Even parity*[1]	(Initial value)
1	Odd parity*[2]	

Notes: *1 When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even.

In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is even.

*2 When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd.

In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.

Bit 3—Stop Bit Length (STOP): Selects 1 or 2 bits as the stop bit length in asynchronous mode. The STOP bits setting is only valid in asynchronous mode. If clocked synchronous mode is set the STOP bit setting is invalid since stop bits are not added.

Bit 3

STOP	Description	
0	1 stop bit: In transmission, a single 1 bit (stop bit) is added to the end of a transmit character before it is sent.	(Initial value)
1	2 stop bits: In transmission, two 1 bits (stop bits) are added to the end of a transmit character before it is sent.	

In reception, only the first stop bit is checked, regardless of the STOP bit setting. If the second stop bit is 1, it is treated as a stop bit; if it is 0, it is treated as the start bit of the next transmit character.

Bit 2—Multiprocessor Mode (MP): Selects multiprocessor format. When multiprocessor format is selected, the PE bit and O/$\overline{E}$ bit parity settings are invalid. The MP bit setting is only valid in asynchronous mode; it is invalid in clocked synchronous mode.

For details of the multiprocessor communication function, see section 13.3.3, Multiprocessor Communication Function.

Bit 2

MP	Description	
0	Multiprocessor function disabled	(Initial value)
1	Multiprocessor format selected	

Bits 1 and 0—Clock Select 1 and 0 (CKS1, CKS0): These bits select the clock source for the baud rate generator. The clock source can be selected from ø, ø/4, ø/16, and ø/64, according to the setting of bits CKS1 and CKS0.

For the relation between the clock source, the bit rate register setting, and the baud rate, see section 13.2.8, Bit Rate Register.

Bit 1	Bit 0		
CKS1	**CKS0**	**Description**	
0	0	ø clock	(Initial value)
	1	ø/4 clock	
1	0	ø/16 clock	
	1	ø/64 clock	

13.2.6 Serial Control Register (SCR)

Bit	:	7	6	5	4	3	2	1	0
		TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SCR is a register that performs enabling or disabling of SCI transfer operations, serial clock output in asynchronous mode, and interrupt requests, and selection of the serial clock source.

SCR can be read or written to by the CPU at all times.

SCR is initialized to H'00 by a reset and in standby mode.

Bit 7—Transmit Interrupt Enable (TIE): Enables or disables transmit data empty interrupt (TXI) request generation when serial transmit data is transferred from TDR to TSR and the TDRE flag in SSR is set to 1.

Bit 7		
TIE	**Description**	
0	Transmit data empty interrupt (TXI) requests disabled*	(Initial value)
1	Transmit data empty interrupt (TXI) requests enabled	

Note:* TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.

Bit 6—Receive Interrupt Enable (RIE): Enables or disables receive data full interrupt (RXI) request and receive error interrupt (ERI) request generation when serial receive data is transferred from RSR to RDR and the RDRF flag in SSR is set to 1.

Bit 6

RIE	Description
0	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request disabled* (Initial value)
1	Receive data full interrupt (RXI) request and receive error interrupt (ERI) request enabled

Note:* RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or clearing the RIE bit to 0.

Bit 5—Transmit Enable (TE): Enables or disables the start of serial transmission by the SCI.

Bit 5

TE	Description
0	Transmission disabled*[1] (Initial value)
1	Transmission enabled*[2]

Notes: *1 The TDRE flag in SSR is fixed at 1.

*2 In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0.

SMR setting must be performed to decide the transfer format before setting the TE bit to 1.

Bit 4—Receive Enable (RE): Enables or disables the start of serial reception by the SCI.

Bit 4

RE	Description
0	Reception disabled*[1] (Initial value)
1	Reception enabled*[2]

Notes: *1 Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.

*2 Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode.

SMR setting must be performed to decide the transfer format before setting the RE bit to 1.

Bit 3—Multiprocessor Interrupt Enable (MPIE): Enables or disables multiprocessor interrupts. The MPIE bit setting is only valid in asynchronous mode when the MP bit in SMR is set to 1.

The MPIE bit setting is invalid in clocked synchronous mode or when the MP bit is cleared to 0.

Bit 3

MPIE	Description	
0	Multiprocessor interrupts disabled (normal reception performed) [Clearing conditions] • When the MPIE bit is cleared to 0 • When MPB= 1 data is received	(Initial value)
1	Multiprocessor interrupts enabled* Receive interrupt (RXI) requests, receive error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received.	

Note: * When receive data including MPB = 0 is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and ORER flags in SSR , is not performed. When receive data including MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting is enabled.

Bit 2—Transmit End Interrupt Enable (TEIE): Enables or disables transmit end interrupt (TEI) request generation when there is no valid transmit data in TDR in MSB data transmission.

Bit 2

TEIE	Description	
0	Transmit end interrupt (TEI) request disabled*	(Initial value)
1	Transmit end interrupt (TEI) request enabled*	

Note: * TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0.

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the SCI clock source and enable or disable clock output from the SCK pin. The combination of the CKE1 and CKE0 bits determines whether the SCK pin functions as an I/O port, the serial clock output pin, or the serial clock input pin.

The setting of the CKE0 bit, however, is only valid for internal clock operation (CKE1 = 0) in asynchronous mode. The CKE0 bit setting is invalid in clocked synchronous mode, and in the case of external clock operation (CKE1 = 1). Note that the SCI's operating mode must be decided using SMR before setting the CKE1 and CKE0 bits.

For details of clock source selection, see table 13.9 in section 13.3, Operation.

Bit 1	Bit 0		
CKE1	**CKE0**	**Description**	
0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port*[1]
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output*[1]
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*[2]
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*[3]
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*[3]
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input

Notes: *1 Initial value
*2 Outputs a clock of the same frequency as the bit rate.
*3 Inputs a clock with a frequency 16 times the bit rate.

13.2.7 Serial Status Register (SSR)

Bit	:	7	6	5	4	3	2	1	0
		TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT
Initial value	:	1	0	0	0	0	1	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * Only 0 can be written, to clear the flag.

SSR is an 8-bit register containing status flags that indicate the operating status of the SCI, and multiprocessor bits.

SSR can be read or written to by the CPU at all times. However, 1 cannot be written to flags TDRE, RDRF, ORER, PER, and FER. Also note that in order to clear these flags they must be read as 1 beforehand. The TEND flag and MPB flag are read-only flags and cannot be modified.

SSR is initialized to H'84 by a reset, in standby mode, watch mode*, subactive mode*, and subsleep mode* or module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Bit 7—Transmit Data Register Empty (TDRE): Indicates that data has been transferred from TDR to TSR and the next serial data can be written to TDR.

Bit 7

TDRE	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written to TDR

Bit 6—Receive Data Register Full (RDRF): Indicates that the received data is stored in RDR.

Bit 6

RDRF	Description
0	[Clearing conditions] (Initial value) • When 0 is written to RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and reads data from RDR
1	[Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR

Note: RDR and the RDRF flag are not affected and retain their previous values when an error is detected during reception or when the RE bit in SCR is cleared to 0.

If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.

Bit 5—Overrun Error (ORER): Indicates that an overrun error occurred during reception, causing abnormal termination.

Bit 5

ORER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written to ORER after reading ORER = 1
1	[Setting condition] When the next serial reception is completed while RDRF = 1*2

Notes: *1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*2 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

Bit 4—Framing Error (FER): Indicates that a framing error occurred during reception in asynchronous mode, causing abnormal termination.

Bit 4

FER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written to FER after reading FER = 1
1	[Setting condition] When the SCI checks whether the stop bit at the end of the receive data when reception ends, and the stop bit is 0 *2

Notes: *1 The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*2 In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the FER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

Bit 3—Parity Error (PER): Indicates that a parity error occurred during reception using parity addition in asynchronous mode, causing abnormal termination.

Bit 3

PER	Description
0	[Clearing condition] (Initial value)*1 When 0 is written to PER after reading PER = 1
1	[Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/$\overline{E}$ bit in SMR*2

Notes: *1 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*2 If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

Bit 2—Transmit End (TEND): Indicates that there is no valid data in TDR when the last bit of the transmit character is sent, and transmission has been ended.

The TEND flag is read-only and cannot be modified.

Bit 2

TEND	Description
0	[Clearing conditions] • When 0 is written to TDRE after reading TDRE = 1 • When the DMAC or DTC is activated by a TXI interrupt and writes data to TDR
1	[Setting conditions] (Initial value) • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character

Bit 1—Multiprocessor Bit (MPB): When reception is performed using multiprocessor format in asynchronous mode, MPB stores the multiprocessor bit in the receive data.

MPB is a read-only bit, and cannot be modified.

Bit 1

MPB	Description
0	[Clearing condition] (Initial value)* When data with a 0 multiprocessor bit is received
1	[Setting condition] When data with a 1 multiprocessor bit is received

Note: * Retains its previous state when the RE bit in SCR is cleared to 0 with multiprocessor format.

Bit 0—Multiprocessor Bit Transfer (MPBT): When transmission is performed using multiprocessor format in asynchronous mode, MPBT stores the multiprocessor bit to be added to the transmit data.

The MPBT bit setting is invalid when multiprocessor format is not used, when not transmitting, and in clocked synchronous mode.

Bit 0

MPBT	Description
0	Data with a 0 multiprocessor bit is transmitted (Initial value)
1	Data with a 1 multiprocessor bit is transmitted

13.2.8 Bit Rate Register (BRR)

Bit	:	7	6	5	4	3	2	1	0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

BRR is an 8-bit register that sets the serial transfer bit rate in accordance with the baud rate generator operating clock selected by bits CKS1 and CKS0 in SMR.

BRR can be read or written to by the CPU at all times.

BRR is initialized to H'FF by a reset and in standby mode.

As baud rate generator control is performed independently for each channel, different values can be set for each channel.

Table 13-3 shows sample BRR settings in asynchronous mode, and table 13-4 shows sample BRR settings in clocked synchronous mode.

Table 13-3 BRR Settings for Various Bit Rates (Asynchronous Mode)

	ø = 4 MHz			ø = 4.9152 MHz			ø = 5 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	70	0.03	2	86	0.31	2	88	–0.25
150	1	207	0.16	1	255	0.00	2	64	0.16
300	1	103	0.16	1	127	0.00	1	129	0.16
600	0	207	0.16	0	255	0.00	1	64	0.16
1200	0	103	0.16	0	127	0.00	0	129	0.16
2400	0	51	0.16	0	63	0.00	0	64	0.16
4800	0	25	0.16	0	31	0.00	0	32	–1.36
9600	0	12	0.16	0	15	0.00	0	15	1.73
19200	—	—	—	0	7	0.00	0	7	1.73
31250	0	3	0.00	0	4	–1.70	0	4	0.00
38400	—	—	—	0	3	0.00	0	3	1.73

Bit Rate (bit/s)	ø = 6 MHz			ø = 6.144 MHz			ø = 7.3728 MHz			ø = 8 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	106	–0.44	2	108	0.08	2	130	–0.07	2	141	0.03
150	2	77	0.16	2	79	0.00	2	95	0.00	2	103	0.16
300	1	155	0.16	1	159	0.00	1	191	0.00	1	207	0.16
600	1	77	0.16	1	79	0.00	1	95	0.00	1	103	0.16
1200	0	155	0.16	0	159	0.00	0	191	0.00	0	207	0.16
2400	0	77	0.16	0	79	0.00	0	95	0.00	0	103	0.16
4800	0	38	0.16	0	39	0.00	0	47	0.00	0	51	0.16
9600	0	19	–2.34	0	19	0.00	0	23	0.00	0	25	0.16
19200	0	9	–2.34	0	9	0.00	0	11	0.00	0	12	0.16
31250	0	5	0.00	0	5	2.40	—	—	—	0	7	0.00
38400	0	4	–2.34	0	4	0.00	0	5	0.00	—	—	—

Bit Rate (bit/s)	ø = 9.8304 MHz			ø = 10 MHz			ø = 12 MHz			ø = 12.288 MHz		
	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)	n	N	Error (%)
110	2	174	–0.26	2	177	–0.25	2	212	0.03	2	217	0.08
150	2	127	0.00	2	129	0.16	2	155	0.16	2	159	0.00
300	1	255	0.00	2	64	0.16	2	77	0.16	2	79	0.00
600	1	127	0.00	1	129	0.16	1	155	0.16	1	159	0.00
1200	0	255	0.00	1	64	0.16	1	77	0.16	1	79	0.00
2400	0	127	0.00	0	129	0.16	0	155	0.16	0	159	0.00
4800	0	63	0.00	0	64	0.16	0	77	0.16	0	79	0.00
9600	0	31	0.00	0	32	–1.36	0	38	0.16	0	39	0.00
19200	0	15	0.00	0	15	1.73	0	19	–2.34	0	19	0.00
31250	0	9	–1.70	0	9	0.00	0	11	0.00	0	11	2.40
38400	0	7	0.00	0	7	1.73	0	9	–2.34	0	9	0.00

	ø = 14 MHz			ø = 14.7456 MHz			ø = 16 MHz			ø = 17.2032 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	2	248	–0.17	3	64	0.70	3	70	0.03	3	75	0.48
150	2	181	0.16	2	191	0.00	2	207	0.16	2	223	0.00
300	2	90	0.16	2	95	0.00	2	103	0.16	2	111	0.00
600	1	181	0.16	1	191	0.00	1	207	0.16	1	223	0.00
1200	1	90	0.16	1	95	0.00	1	103	0.16	1	111	0.00
2400	0	181	0.16	0	191	0.00	0	207	0.16	0	223	0.00
4800	0	90	0.16	0	95	0.00	0	103	0.16	0	111	0.00
9600	0	45	–0.93	0	47	0.00	0	51	0.16	0	55	0.00
19200	0	22	–0.93	0	23	0.00	0	25	0.16	0	27	0.00
31250	0	13	0.00	0	14	–1.70	0	15	0.00	0	16	1.20
38400	—	—	—	0	11	0.00	0	12	0.16	0	13	0.00

	ø = 18 MHz			ø = 19.6608 MHz			ø = 20 MHz		
Bit Rate (bit/s)	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**	**n**	**N**	**Error (%)**
110	3	79	–0.12	3	86	0.31	3	88	–0.25
150	2	233	0.16	2	255	0.00	3	64	0.16
300	2	116	0.16	2	127	0.00	2	129	0.16
600	1	233	0.16	1	255	0.00	2	64	0.16
1200	1	116	0.16	1	127	0.00	1	129	0.16
2400	0	233	0.16	0	255	0.00	1	64	0.16
4800	0	116	0.16	0	127	0.00	0	129	0.16
9600	0	58	–0.69	0	63	0.00	0	64	0.16
19200	0	28	1.02	0	31	0.00	0	32	–1.36
31250	0	17	0.00	0	19	–1.70	0	19	0.00
38400	0	14	–2.34	0	15	0.00	0	15	1.73

Table 13-4 BRR Settings for Various Bit Rates (Clocked Synchronous Mode)

Bit Rate	ø = 4 MHz		ø = 8 MHz		ø = 10 MHz		ø = 16 MHz		ø = 20 MHz	
(bit/s)	n	N	n	N	n	N	n	N	n	N
110	—	—								
250	2	249	3	124	—	—	3	249		
500	2	124	2	249	—	—	3	124	—	—
1 k	1	249	2	124	—	—	2	249	—	—
2.5 k	1	99	1	199	1	249	2	99	2	124
5 k	0	199	1	99	1	124	1	199	1	249
10 k	0	99	0	199	0	249	1	99	1	124
25 k	0	39	0	79	0	99	0	159	0	199
50 k	0	19	0	39	0	49	0	79	0	99
100 k	0	9	0	19	0	24	0	39	0	49
250 k	0	3	0	7	0	9	0	15	0	19
500 k	0	1	0	3	0	4	0	7	0	9
1 M	0	0*	0	1			0	3	0	4
2.5 M					0	0*			0	1
5 M									0	0*

Note: As far as possible, the setting should be made so that the error is no more than 1%.

Legend

Blank : Cannot be set.

— : Can be set, but there will be a degree of error.

* : Continuous transfer is not possible.

The BRR setting is found from the following formulas.

Asynchronous mode:

$$N = \frac{ø}{64 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Clocked synchronous mode:

$$N = \frac{ø}{8 \times 2^{2n-1} \times B} \times 10^6 - 1$$

Where B: Bit rate (bit/s)
N: BRR setting for baud rate generator ($0 \leq N \leq 255$)
ø: Operating frequency (MHz)
n: Baud rate generator input clock (n = 0 to 3)
(See the table below for the relation between n and the clock.)

		SMR Setting	
n	**Clock**	**CKS1**	**CKS0**
0	ø	0	0
1	ø/4	0	1
2	ø/16	1	0
3	ø/64	1	1

The bit rate error in asynchronous mode is found from the following formula:

$$\text{Error (\%)} = \{ \frac{ø \times 10^6}{(N + 1) \times B \times 64 \times 2^{2n-1}} - 1 \} \times 100$$

Table 13-5 shows the maximum bit rate for each frequency in asynchronous mode. Tables 13-6 and 13-7 show the maximum bit rates with external clock input.

Table 13-5 Maximum Bit Rate for Each Frequency (Asynchronous Mode)

ø (MHz)	Maximum Bit Rate (bit/s)	n	N
4	125000	0	0
4.9152	153600	0	0
5	156250	0	0
6	187500	0	0
6.144	192000	0	0
7.3728	230400	0	0
8	250000	0	0
9.8304	307200	0	0
10	312500	0	0
12	375000	0	0
12.288	384000	0	0
14	437500	0	0
14.7456	460800	0	0
16	500000	0	0
17.2032	537600	0	0
18	562500	0	0
19.6608	614400	0	0
20	625000	0	0

Table 13-6 Maximum Bit Rate with External Clock Input (Asynchronous Mode)

ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
4	1.0000	62500
4.9152	1.2288	76800
5	1.2500	78125
6	1.5000	93750
6.144	1.5360	96000
7.3728	1.8432	115200
8	2.0000	125000
9.8304	2.4576	153600
10	2.5000	156250
12	3.0000	187500
12.288	3.0720	192000
14	3.5000	218750
14.7456	3.6864	230400
16	4.0000	250000
17.2032	4.3008	268800
18	4.5000	281250
19.6608	4.9152	307200
20	5.0000	312500

Table 13-7 Maximum Bit Rate with External Clock Input (Clocked Synchronous Mode)

ø (MHz)	External Input Clock (MHz)	Maximum Bit Rate (bit/s)
4	0.6667	666666.7
6	1.0000	1000000.0
8	1.3333	1333333.3
10	1.6667	1666666.7
12	2.0000	2000000.0
14	2.3333	2333333.3
16	2.6667	2666666.7
18	3.0000	3000000.0
20	3.3333	3333333.3

13.2.9 Smart Card Mode Register (SCMR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

SCMR selects LSB-first or MSB-first by means of bit SDIR. Except in the case of asynchronous mode 7-bit data, LSB-first or MSB-first can be selected regardless of the serial communication mode. The descriptions in this chapter refer to LSB-first transfer.

For details of the other bits in SCMR, see 14.2.1, Smart Card Mode Register (SCMR).

SCMR is initialized to H'F2 by a reset and in standby mode.

Bits 7 to 4—Reserved: These bits are always read as 1 and cannot be modified.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

This bit is valid when 8-bit data is used as the transmit/receive format.

Bit 3

SDIR	Description	
0	TDR contents are transmitted LSB-first	(Initial value)
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

Bit 2—Smart Card Data Invert (SINV): Specifies inversion of the data logic level. The SINV bit does not affect the logic level of the parity bit(s): parity bit inversion requires inversion of the O/$\overline{E}$ bit in SMR.

Bit 2

SINV	Description	
0	TDR contents are transmitted without modification Receive data is stored in RDR without modification	(Initial value)
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form	

Bit 1—Reserved: This bit is always read as 1 and cannot be modified.

Bit 0—Smart Card Interface Mode Select (SMIF): When the smart card interface operates as a normal SCI, 0 should be written in this bit.

Bit 0

SMIF	Description	
0	Operates as normal SCI (smart card interface function disabled)	(Initial value)
1	Smart card interface function enabled	

13.2.10 Module Stop Control Register B (MSTPCRB)

MSTPCRB

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRB is 8-bit readable/writable registers that perform module stop mode control.

Setting any of bits MSTPB7 to MSTBP5 to 1 stops SCI0 to SCI2 operating and enter module stop mode on completion of the bus cycle. For details, see section 23A.5, 23B.5, Module Stop Mode.

MSTPCRB is initialized to H'FF by a reset and in hardware standby mode. They are not initialized by a manual reset and in software standby mode.

Bit 7—Module Stop (MSTPB7): Specifies the SCI0 module stop mode.

Bit 7		
MSTPB7	**Description**	
0	SCI0 module stop mode is cleared	
1	SCI0 module stop mode is set	(Initial value)

Bit 6—Module Stop (MSTPB6): Specifies the SCI1 module stop mode.

Bit 6		
MSTPB6	**Description**	
0	SCI1 module stop mode is cleared	
1	SCI1 module stop mode is set	(Initial value)

Bit 5—Module Stop (MSTPB5): Specifies the SCI2 module stop mode.

Bit 5		
MSTPB5	**Description**	
0	SCI2 module stop mode is cleared	
1	SCI2 module stop mode is set	(Initial value)

13.3 Operation

13.3.1 Overview

The SCI can carry out serial communication in two modes: asynchronous mode in which synchronization is achieved character by character, and clocked synchronous mode in which synchronization is achieved with clock pulses.

Selection of asynchronous or clocked synchronous mode and the transmission format is made using SMR as shown in table 13-8. The SCI clock is determined by a combination of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR, as shown in table 13-9.

Asynchronous Mode

- Data length: Choice of 7 or 8 bits
- Choice of parity addition, multiprocessor bit addition, and addition of 1 or 2 stop bits (the combination of these parameters determines the transfer format and character length)
- Detection of framing, parity, and overrun errors, and breaks, during reception
- Choice of internal or external clock as SCI clock source
  - — When internal clock is selected:

    The SCI operates on the baud rate generator clock and a clock with the same frequency as the bit rate can be output
  - — When external clock is selected:

    A clock with a frequency of 16 times the bit rate must be input (the on-chip baud rate generator is not used)

Clocked Synchronous Mode

- Transfer format: Fixed 8-bit data
- Detection of overrun errors during reception
- Choice of internal or external clock as SCI clock source
  - — When internal clock is selected:

    The SCI operates on the baud rate generator clock and a serial clock is output off-chip
  - — When external clock is selected:

    The on-chip baud rate generator is not used, and the SCI operates on the input serial clock

Table 13-8 SMR Settings and Serial Transfer Format Selection

SMR Settings						SCI Transfer Format			
Bit 7	Bit 6	Bit 2	Bit 5	Bit 3		Data	Multi Processor	Parity	Stop Bit
C/$\overline{A}$	CHR	MP	PE	STOP	Mode	Length	Bit	Bit	Length
0	0	0	0	0	Asynchronous mode	8-bit data	No	No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	1		0	0		7-bit data		No	1 bit
				1					2 bits
			1	0				Yes	1 bit
				1					2 bits
	0	1	—	0	Asynchronous mode (multi-processor format)	8-bit data	Yes	No	1 bit
			—	1					2 bits
	1		—	0		7-bit data			1 bit
			—	1					2 bits
1	—	—	—	—	Clocked synchronous mode	8-bit data	No		None

Table 13-9 SMR and SCR Settings and SCI Clock Source Selection

SMR	SCR Setting			SCI Transmit/Receive Clock	
Bit 7	Bit 1	Bit 0		Clock	
C/$\overline{A}$	CKE1	CKE0	Mode	Source	SCK Pin Function
0	0	0	Asynchronous mode	Internal	SCI does not use SCK pin
		1			Outputs clock with same frequency as bit rate
	1	0		External	Inputs clock with frequency of 16 times the bit rate
		1			
1	0	0	Clocked synchronous mode	Internal	Outputs serial clock
		1			
	1	0		External	Inputs serial clock
		1			

13.3.2 Operation in Asynchronous Mode

In asynchronous mode, characters are sent or received, each preceded by a start bit indicating the start of communication and stop bits indicating the end of communication. Serial communication is thus carried out with synchronization established on a character-by-character basis.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 13-2 shows the general format for asynchronous serial communication.

In asynchronous serial communication, the transmission line is usually held in the mark state (high level). The SCI monitors the transmission line, and when it goes to the space state (low level), recognizes a start bit and starts serial communication.

One serial communication character consists of a start bit (low level), followed by data (in LSB-first order), a parity bit (high or low level), and finally stop bits (high level).

In asynchronous mode, the SCI performs synchronization at the falling edge of the start bit in reception. The SCI samples the data on the 8th pulse of a clock with a frequency of 16 times the length of one bit, so that the transfer data is latched at the center of each bit.

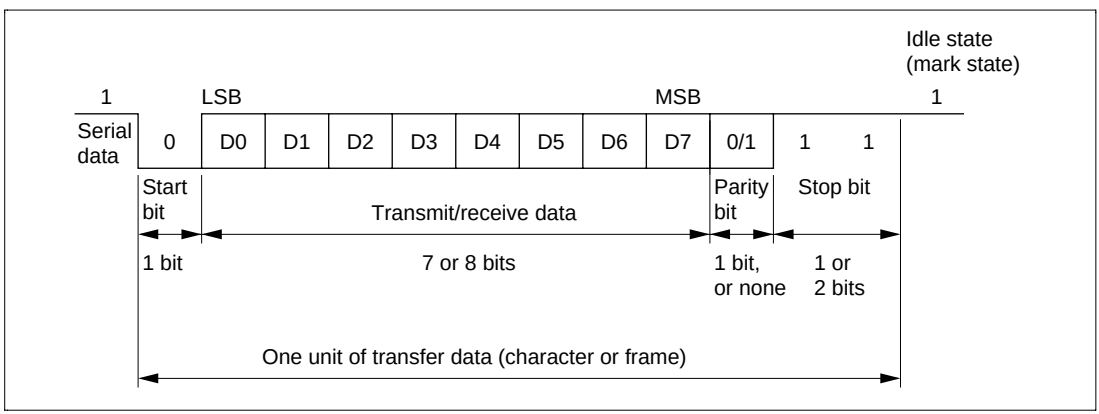


Figure 13-2 Data Format in Asynchronous Communication (Example with 8-Bit Data, Parity, Two Stop Bits)

Data Transfer Format: Table 13-10 shows the data transfer formats that can be used in asynchronous mode. Any of 12 transfer formats can be selected according to the SMR setting.

Table 13-10 Serial Transfer Formats (Asynchronous Mode)

SMR Settings				Serial Transfer Format and Frame Length
CHR	PE	MP	STOP	1 2 3 4 5 6 7 8 9 10 11 12
0	0	0	0	S 8-bit data STOP
0	0	0	1	S 8-bit data STOP STOP
0	1	0	0	S 8-bit data P STOP
0	1	0	1	S 8-bit data P STOP STOP
1	0	0	0	S 7-bit data STOP
1	0	0	1	S 7-bit data STOP STOP
1	1	0	0	S 7-bit data P STOP
1	1	0	1	S 7-bit data P STOP STOP
0	—	1	0	S 8-bit data MPB STOP
0	—	1	1	S 8-bit data MPB STOP STOP
1	—	1	0	S 7-bit data MPB STOP
1	—	1	1	S 7-bit data MPB STOP STOP

Legend

S : Start bit
STOP : Stop bit
P : Parity bit
MPB : Multiprocessor bit

Clock: Either an internal clock generated by the on-chip baud rate generator or an external clock input at the SCK pin can be selected as the SCI's serial clock, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 13-9.

When an external clock is input at the SCK pin, the clock frequency should be 16 times the bit rate used.

When the SCI is operated on an internal clock, the clock can be output from the SCK pin. The frequency of the clock output in this case is equal to the bit rate, and the phase is such that the rising edge of the clock is in the middle of the transmit data, as shown in figure 13-3.

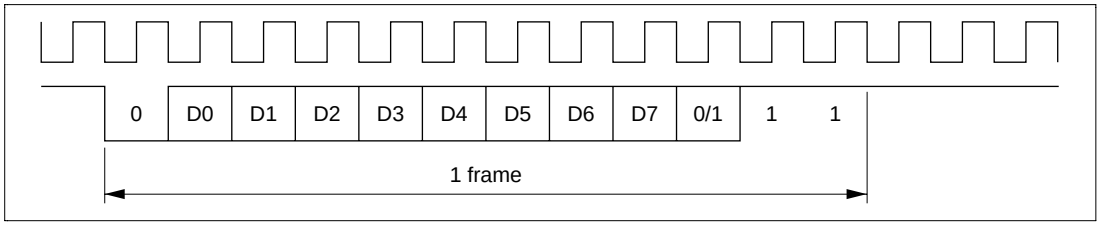


Figure 13-3 Relation between Output Clock and Transfer Data Phase (Asynchronous Mode)

Data Transfer Operations:

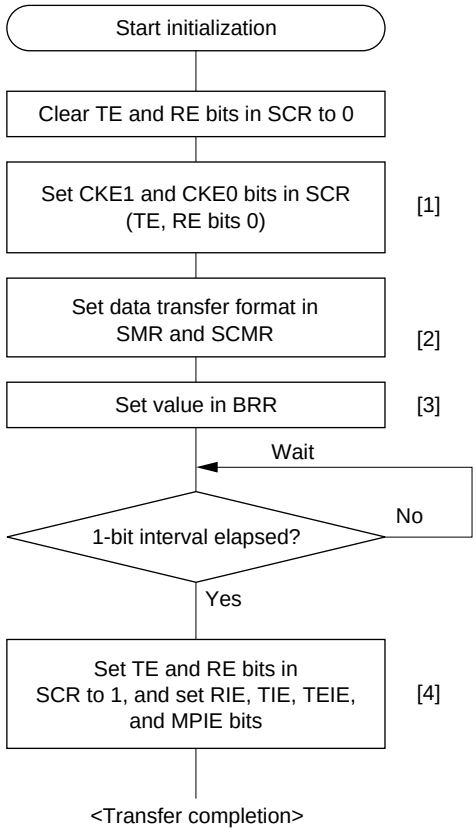
- SCI initialization (asynchronous mode)

  Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

  When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

  When an external clock is used the clock should not be stopped during operation, including initialization, since operation is uncertain.

Figure 13-4 shows a sample SCI initialization flowchart.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, and bits TE and RE, to 0.

When the clock is selected in asynchronous mode, it is output immediately after SCR settings are made.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.

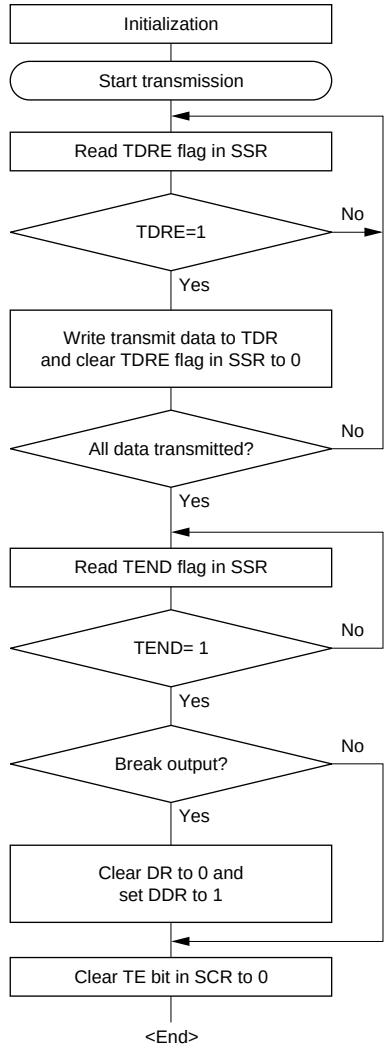
Setting the TE and RE bits enables the TxD and RxD pins to be used.

Figure 13-4 Sample SCI Initialization Flowchart

- Serial data transmission (asynchronous mode)

  Figure 13-5 shows a sample flowchart for serial transmission.

  The following procedure should be used for serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request, and date is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set DDR for the port corresponding to the TxD pin to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 13-5 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

[1] The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

[2] After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission.

If the TIE bit is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.

The serial transmit data is sent from the TxD pin in the following order.

[a] Start bit:

One 0-bit is output.

[b] Transmit data:

8-bit or 7-bit data is output in LSB-first order.

[c] Parity bit or multiprocessor bit:

One parity bit (even or odd parity), or one multiprocessor bit is output.

A format in which neither a parity bit nor a multiprocessor bit is output can also be selected.

[d] Stop bit(s):

One or two 1-bits (stop bits) are output.

[e] Mark state:

1 is output continuously until the start bit that starts the next transmission is sent.

[3] The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is cleared to 0, the data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the “mark state” is entered in which 1 is output continuously. If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

Figure 13-6 shows an example of the operation for transmission in asynchronous mode.

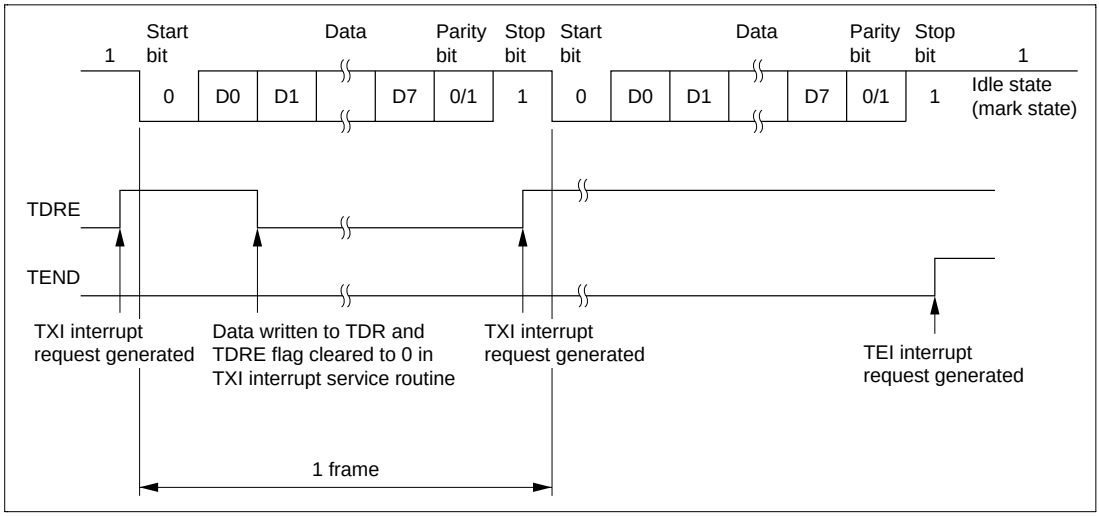
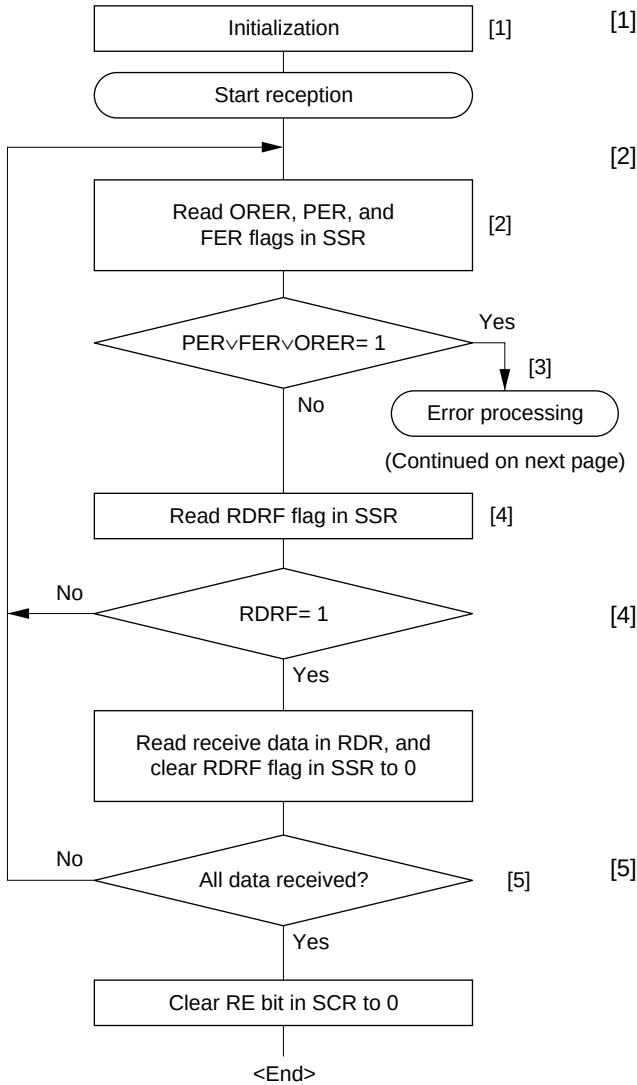


Figure 13-6 Example of Operation in Transmission in Asynchronous Mode (Example with 8-Bit Data, Parity, One Stop Bit)

- Serial data reception (asynchronous mode)

  Figure 13-7 shows a sample flowchart for serial reception.

  The following procedure should be used for serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing and break detection:
If a receive error occurs, read the ORER, PER, and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER, PER, and FER flags are all cleared to 0. Reception cannot be resumed if any of these flags are set to 1. In the case of a framing error, a break can be detected by reading the value of the input port corresponding to the RxD pin.

[4] SCI status check and receive data read :
Read SSR and check that RDRF = 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the stop bit for the current frame is received, read the RDRF flag, read RDR, and clear the RDRF flag to 0. The RDRF flag is cleared automatically when DTC is activated by an RXI interrupt and the RDR value is read.

Figure 13-7 Sample Serial Reception Data Flowchart

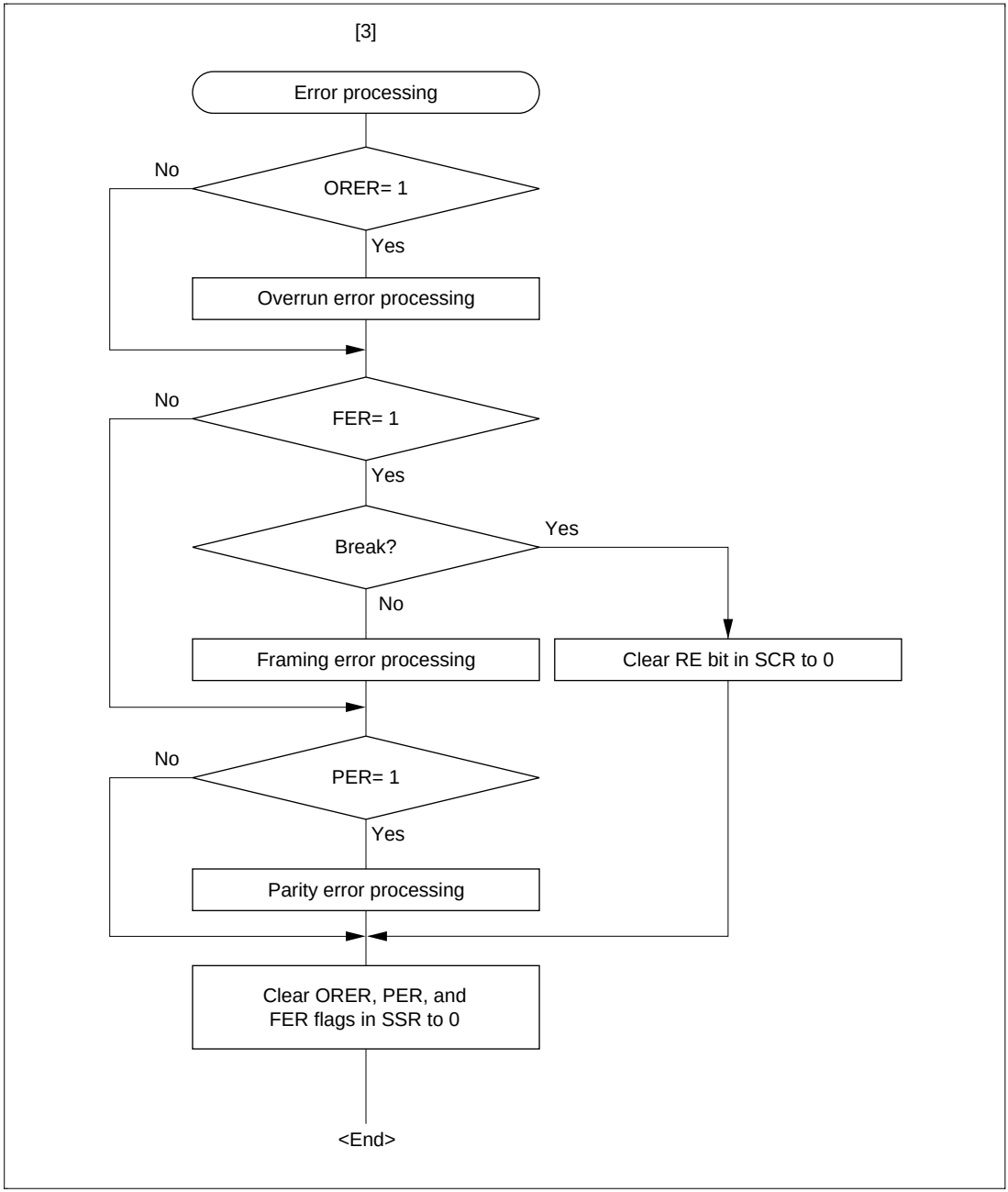


Figure 13-7 Sample Serial Reception Data Flowchart (cont)

In serial reception, the SCI operates as described below.

[1] The SCI monitors the transmission line, and if a 0 stop bit is detected, performs internal synchronization and starts reception.

[2] The received data is stored in RSR in LSB-to-MSB order.

[3] The parity bit and stop bit are received.

After receiving these bits, the SCI carries out the following checks.

[a] Parity check:

The SCI checks whether the number of 1 bits in the receive data agrees with the parity (even or odd) set in the O/$\overline{E}$ bit in SMR.

[b] Stop bit check:

The SCI checks whether the stop bit is 1.

If there are two stop bits, only the first is checked.

[c] Status check:

The SCI checks whether the RDRF flag is 0, indicating that the receive data can be transferred from RSR to RDR.

If all the above checks are passed, the RDRF flag is set to 1, and the receive data is stored in RDR.

If a receive error* is detected in the error check, the operation is as shown in table 13-11.

Note: * Subsequent receive operations cannot be performed when a receive error has occurred. Also note that the RDRF flag is not set to 1 in reception, and so the error flags must be cleared to 0.

[4] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data full interrupt (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER, PER, or FER flag changes to 1, a receive error interrupt (ERI) request is generated.

Table 13-11 Receive Errors and Conditions for Occurrence

Receive Error	Abbreviation	Occurrence Condition	Data Transfer
Overrun error	ORER	When the next data reception is completed while the RDRF flag in SSR is set to 1	Receive data is not transferred from RSR to RDR.
Framing error	FER	When the stop bit is 0	Receive data is transferred from RSR to RDR.
Parity error	PER	When the received data differs from the parity (even or odd) set in SMR	Receive data is transferred from RSR to RDR.

Figure 13-8 shows an example of the operation for reception in asynchronous mode.

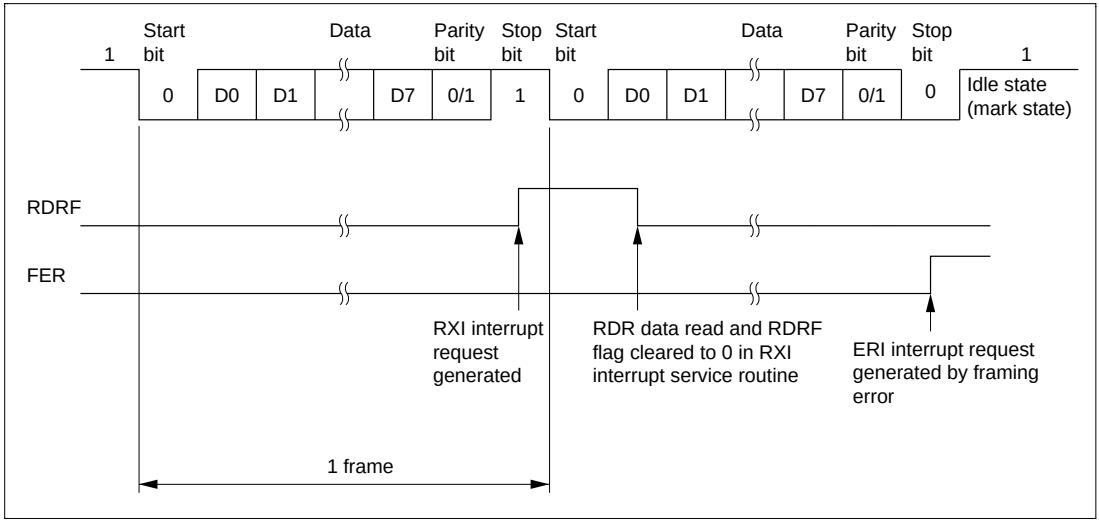


Figure 13-8 Example of SCI Operation in Reception (Example with 8-Bit Data, Parity, One Stop Bit)

13.3.3 Multiprocessor Communication Function

The multiprocessor communication function performs serial communication using the multiprocessor format, in which a multiprocessor bit is added to the transfer data, in asynchronous mode. Use of this function enables data transfer to be performed among a number of processors sharing transmission lines.

When multiprocessor communication is carried out, each receiving station is addressed by a unique ID code.

The serial communication cycle consists of two component cycles: an ID transmission cycle which specifies the receiving station , and a data transmission cycle. The multiprocessor bit is used to differentiate between the ID transmission cycle and the data transmission cycle.

The transmitting station first sends the ID of the receiving station with which it wants to perform serial communication as data with a 1 multiprocessor bit added. It then sends transmit data as data with a 0 multiprocessor bit added.

The receiving station skips the data until data with a 1 multiprocessor bit is sent.

When data with a 1 multiprocessor bit is received, the receiving station compares that data with its own ID. The station whose ID matches then receives the data sent next. Stations whose ID does not match continue to skip the data until data with a 1 multiprocessor bit is again received. In this way, data communication is carried out among a number of processors.

Figure 13-9 shows an example of inter-processor communication using the multiprocessor format.

Data Transfer Format: There are four data transfer formats.

When the multiprocessor format is specified, the parity bit specification is invalid.

For details, see table 13-10.

Clock: See the section on asynchronous mode.

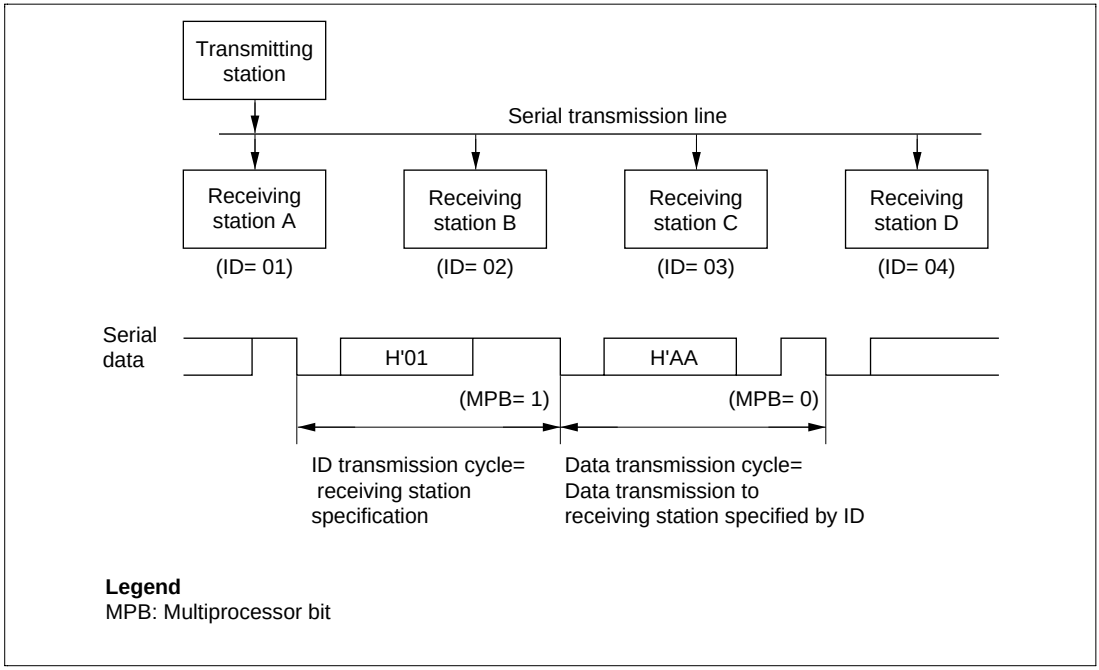


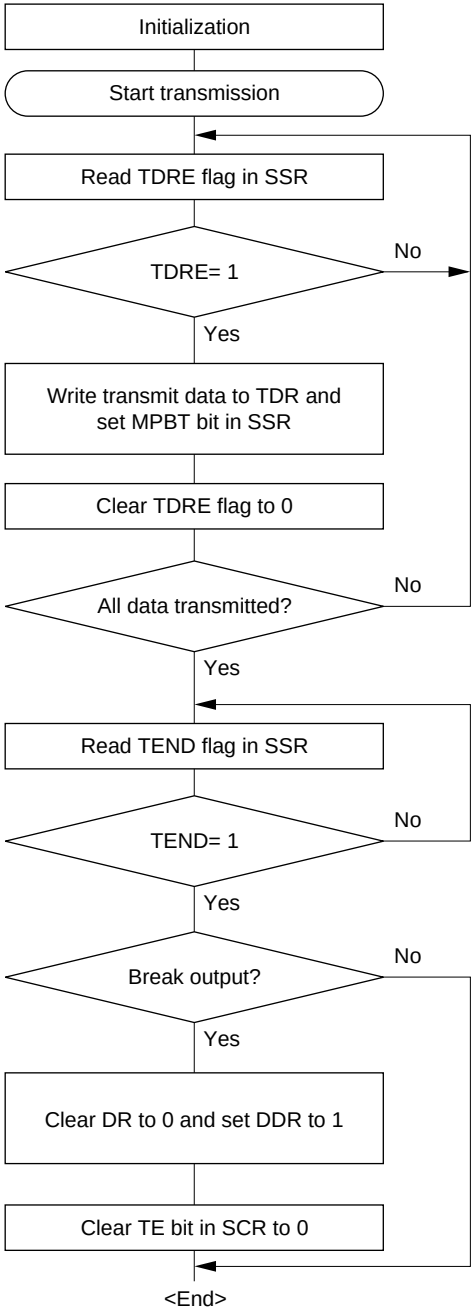
Figure 13-9 Example of Inter-Processor Communication Using Multiprocessor Format (Transmission of Data H'AA to Receiving Station A)

Data Transfer Operations:

- Multiprocessor serial data transmission

  Figure 13-10 shows a sample flowchart for multiprocessor serial data transmission.

  The following procedure should be used for multiprocessor serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.
After the TE bit is set to 1, a frame of 1s is output, and transmission is enabled.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR. Set the MPBT bit in SSR to 0 or 1. Finally, clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0. Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request, and data is written to TDR.

[4] Break output at the end of serial transmission:
To output a break in serial transmission, set the port DDR to 1, clear DR to 0, then clear the TE bit in SCR to 0.

Figure 13-10 Sample Multiprocessor Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

[1] The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

[2] After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission.

If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.

The serial transmit data is sent from the TxD pin in the following order.

[a] Start bit:

One 0-bit is output.

[b] Transmit data:

8-bit or 7-bit data is output in LSB-first order.

[c] Multiprocessor bit

One multiprocessor bit (MPBT value) is output.

[d] Stop bit(s):

One or two 1-bits (stop bits) are output.

[e] Mark state:

1 is output continuously until the start bit that starts the next transmission is sent.

[3] The SCI checks the TDRE flag at the timing for sending the stop bit.

If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, the stop bit is sent, and then serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the stop bit is sent, and then the mark state is entered in which 1 is output continuously. If the TEIE bit in SCR is set to 1 at this time, a transmission end interrupt (TEI) request is generated.

Figure 13-11 shows an example of SCI operation for transmission using the multiprocessor format.

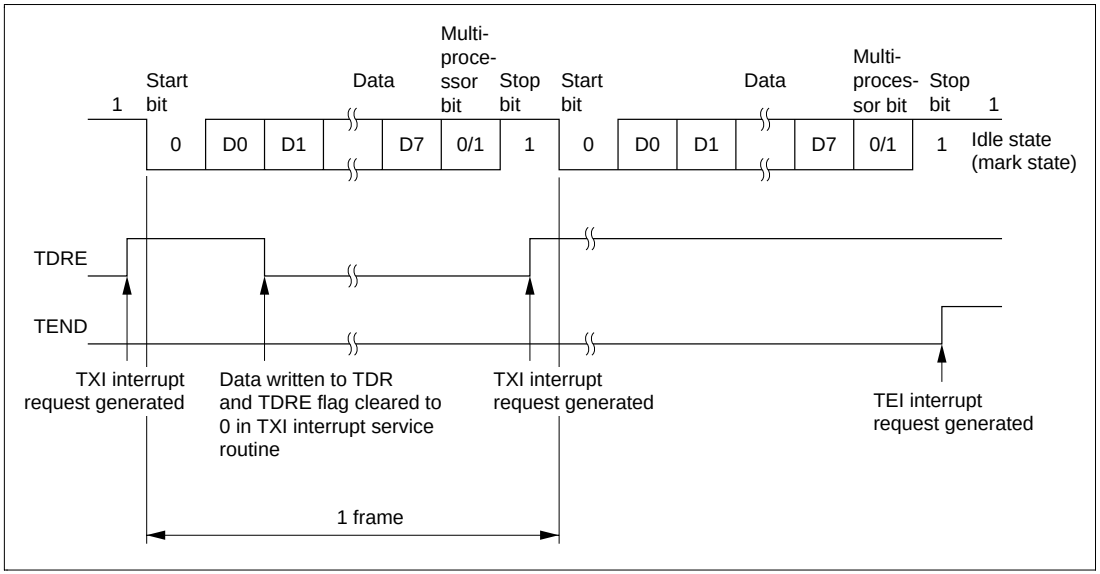
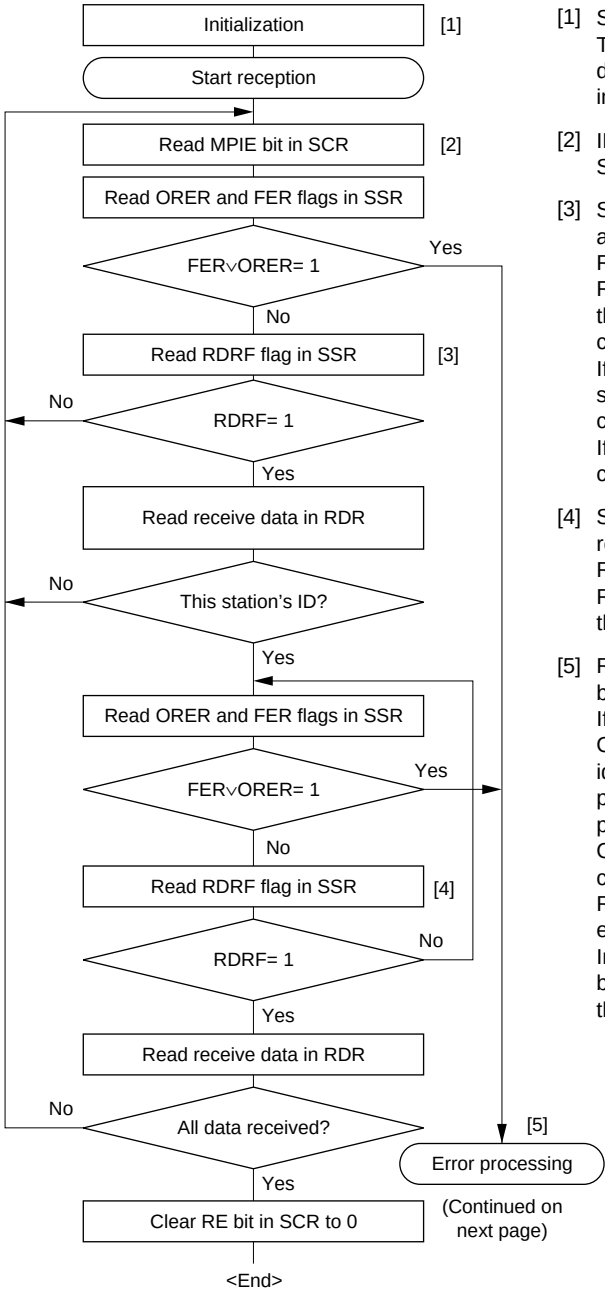


Figure 13-11 Example of SCI Operation in Transmission (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

- Multiprocessor serial data reception

  Figure 13-12 shows a sample flowchart for multiprocessor serial reception.

  The following procedure should be used for multiprocessor serial data reception.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] ID reception cycle:
Set the MPIE bit in SCR to 1.

[3] SCI status check, ID reception and comparison:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and compare it with this station's ID.
If the data is not this station's ID, set the MPIE bit to 1 again, and clear the RDRF flag to 0.
If the data is this station's ID, clear the RDRF flag to 0.

[4] SCI status check and data reception:
Read SSR and check that the RDRF flag is set to 1, then read the data in RDR.

[5] Receive error processing and break detection:
If a receive error occurs, read the ORER and FER flags in SSR to identify the error. After performing the appropriate error processing, ensure that the ORER and FER flags are all cleared to 0.
Reception cannot be resumed if either of these flags is set to 1.
In the case of a framing error, a break can be detected by reading the RxD pin value.

Figure 13-12 Sample Multiprocessor Serial Reception Flowchart

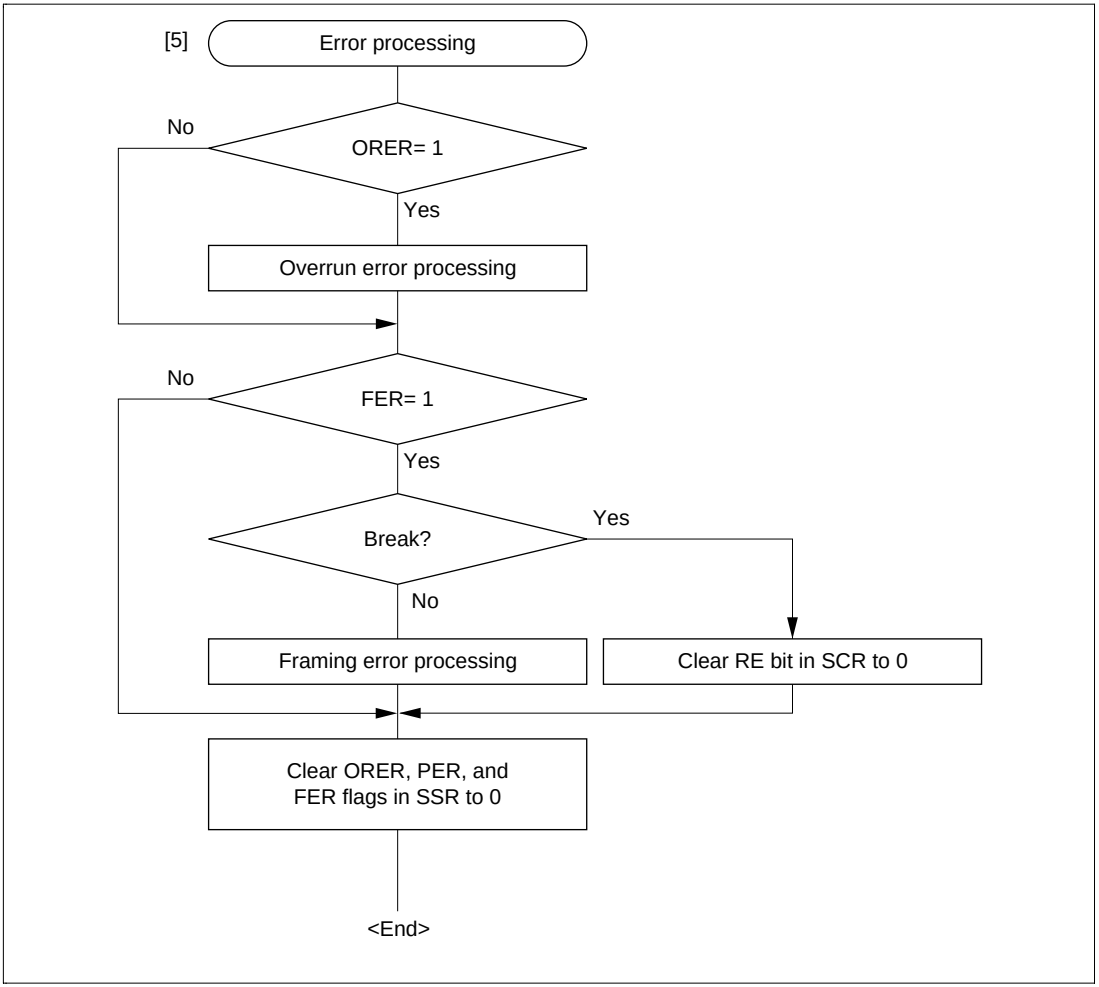


Figure 13-12 Sample Multiprocessor Serial Reception Flowchart (cont)

Figure 13-13 shows an example of SCI operation for multiprocessor format reception.

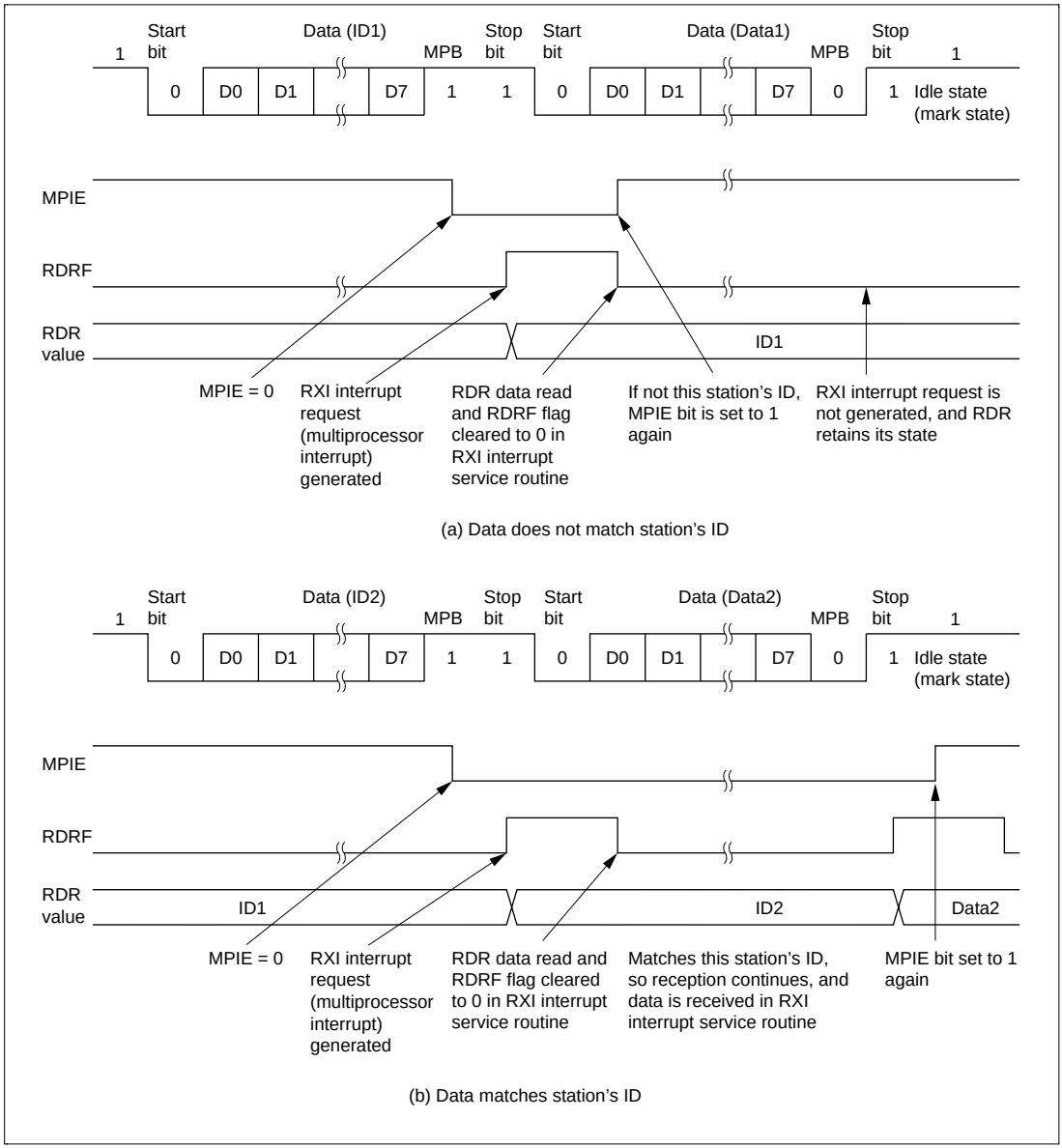


Figure 13-13 Example of SCI Operation in Reception (Example with 8-Bit Data, Multiprocessor Bit, One Stop Bit)

13.3.4 Operation in Clocked Synchronous Mode

In clocked synchronous mode, data is transmitted or received in synchronization with clock pulses, making it suitable for high-speed serial communication.

Inside the SCI, the transmitter and receiver are independent units, enabling full-duplex communication by use of a common clock. Both the transmitter and the receiver also have a double-buffered structure, so that data can be read or written during transmission or reception, enabling continuous data transfer.

Figure 13-14 shows the general format for clocked synchronous serial communication.

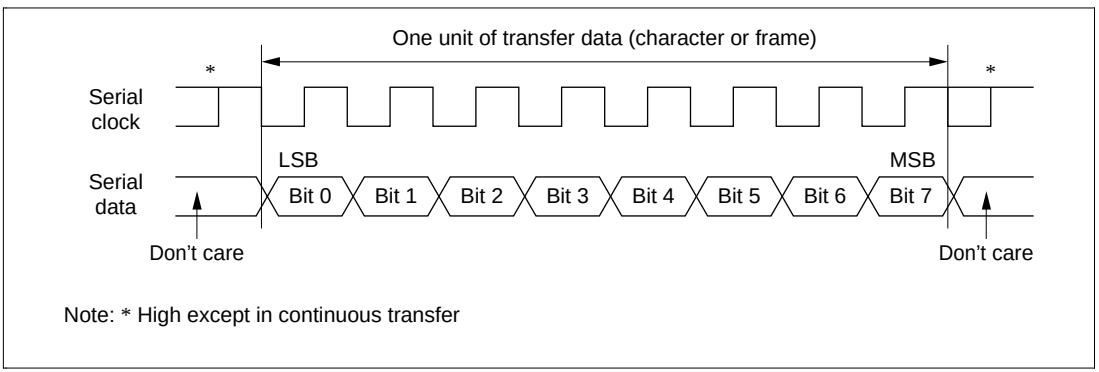


Figure 13-14 Data Format in Synchronous Communication

In clocked synchronous serial communication, data on the transmission line is output from one falling edge of the serial clock to the next. Data confirmation is guaranteed at the rising edge of the serial clock.

In clocked serial communication, one character consists of data output starting with the LSB and ending with the MSB. After the MSB is output, the transmission line holds the MSB state.

In clocked synchronous mode, the SCI receives data in synchronization with the rising edge of the serial clock.

Data Transfer Format: A fixed 8-bit data format is used.

No parity or multiprocessor bits are added.

Clock: Either an internal clock generated by the on-chip baud rate generator or an external serial clock input at the SCK pin can be selected, according to the setting of the C/$\overline{A}$ bit in SMR and the CKE1 and CKE0 bits in SCR. For details of SCI clock source selection, see table 13-9.

When the SCI is operated on an internal clock, the serial clock is output from the SCK pin.

Eight serial clock pulses are output in the transfer of one character, and when no transfer is performed the clock is fixed high. When only receive operations are performed, however, the serial clock is output until an overrun error occurs or the RE bit is cleared to 0. If you want to perform receive operations in units of one character, you should select an external clock as the clock source.

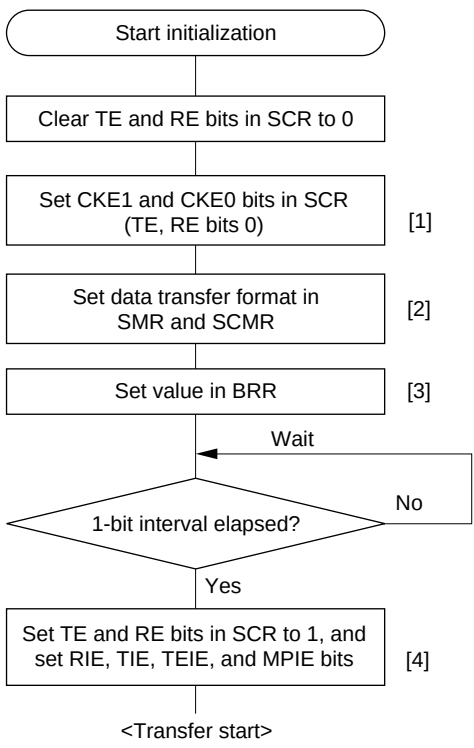
Data Transfer Operations:

- SCI initialization (clocked synchronous mode)

  Before transmitting and receiving data, you should first clear the TE and RE bits in SCR to 0, then initialize the SCI as described below.

  When the operating mode, transfer format, etc., is changed, the TE and RE bits must be cleared to 0 before making the change using the following procedure. When the TE bit is cleared to 0, the TDRE flag is set to 1 and TSR is initialized. Note that clearing the RE bit to 0 does not change the contents of the RDRF, PER, FER, and ORER flags, or the contents of RDR.

  Figure 13-15 shows a sample SCI initialization flowchart.



[1] Set the clock selection in SCR. Be sure to clear bits RIE, TIE, TEIE, and MPIE, TE and RE, to 0.

[2] Set the data transfer format in SMR and SCMR.

[3] Write a value corresponding to the bit rate to BRR. Not necessary if an external clock is used.

[4] Wait at least one bit interval, then set the TE bit or RE bit in SCR to 1. Also set the RIE, TIE, TEIE, and MPIE bits.
Setting the TE and RE bits enables the TxD and RxD pins to be used.

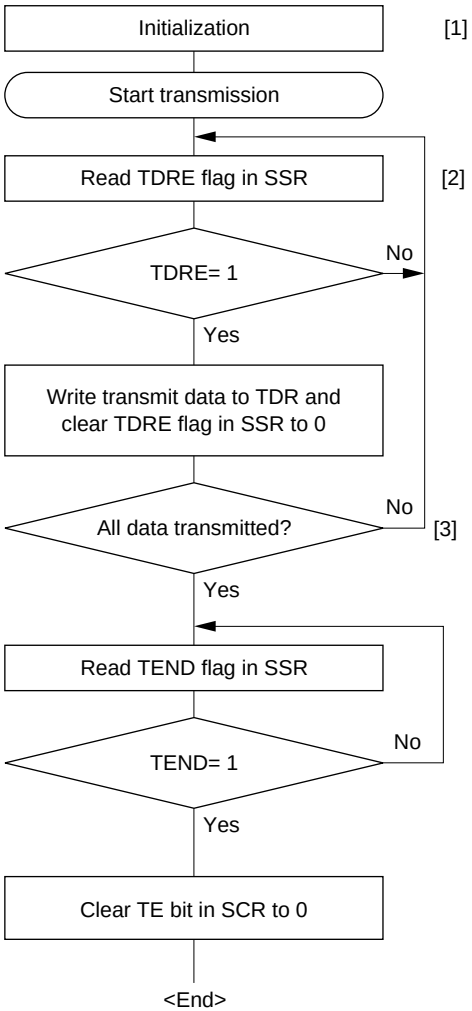
Note: In simultaneous transmit and receive operations, the TE and RE bits should both be cleared to 0 or set to 1 simultaneously.

Figure 13-15 Sample SCI Initialization Flowchart

- Serial data transmission (clocked synchronous mode)

  Figure 13-16 shows a sample flowchart for serial transmission.

  The following procedure should be used for serial data transmission.



[1] SCI initialization:
The TxD pin is automatically designated as the transmit data output pin.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.

[3] Serial transmission continuation procedure:
To continue serial transmission, be sure to read 1 from the TDRE flag to confirm that writing is possible, then write data to TDR, and then clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR.

Figure 13-16 Sample Serial Transmission Flowchart

In serial transmission, the SCI operates as described below.

[1] The SCI monitors the TDRE flag in SSR, and if is 0, recognizes that data has been written to TDR, and transfers the data from TDR to TSR.

[2] After transferring data from TDR to TSR, the SCI sets the TDRE flag to 1 and starts transmission. If the TIE bit in SCR is set to 1 at this time, a transmit data empty interrupt (TXI) is generated.

When clock output mode has been set, the SCI outputs 8 serial clock pulses. When use of an external clock has been specified, data is output synchronized with the input clock.

The serial transmit data is sent from the TxD pin starting with the LSB (bit 0) and ending with the MSB (bit 7).

[3] The SCI checks the TDRE flag at the timing for sending the MSB (bit 7).

If the TDRE flag is cleared to 0, data is transferred from TDR to TSR, and serial transmission of the next frame is started.

If the TDRE flag is set to 1, the TEND flag in SSR is set to 1, the MSB (bit 7) is sent, and the TxD pin maintains its state.

If the TEIE bit in SCR is set to 1 at this time, a TEI interrupt request is generated.

[4] After completion of serial transmission, the SCK pin is fixed high.

Figure 13-17 shows an example of SCI operation in transmission.

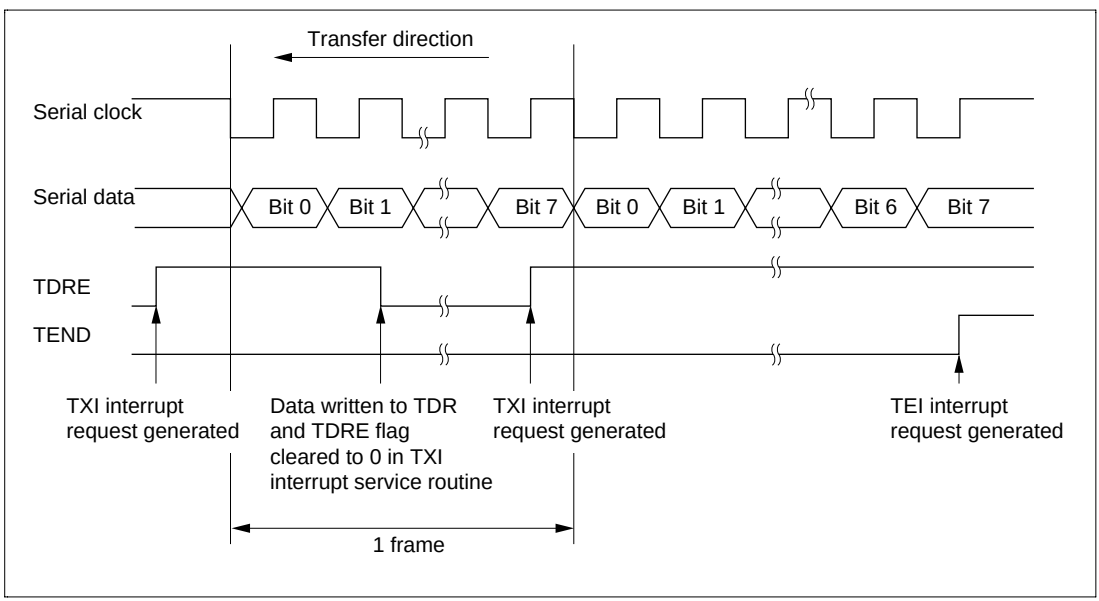


Figure 13-17 Example of SCI Operation in Transmission

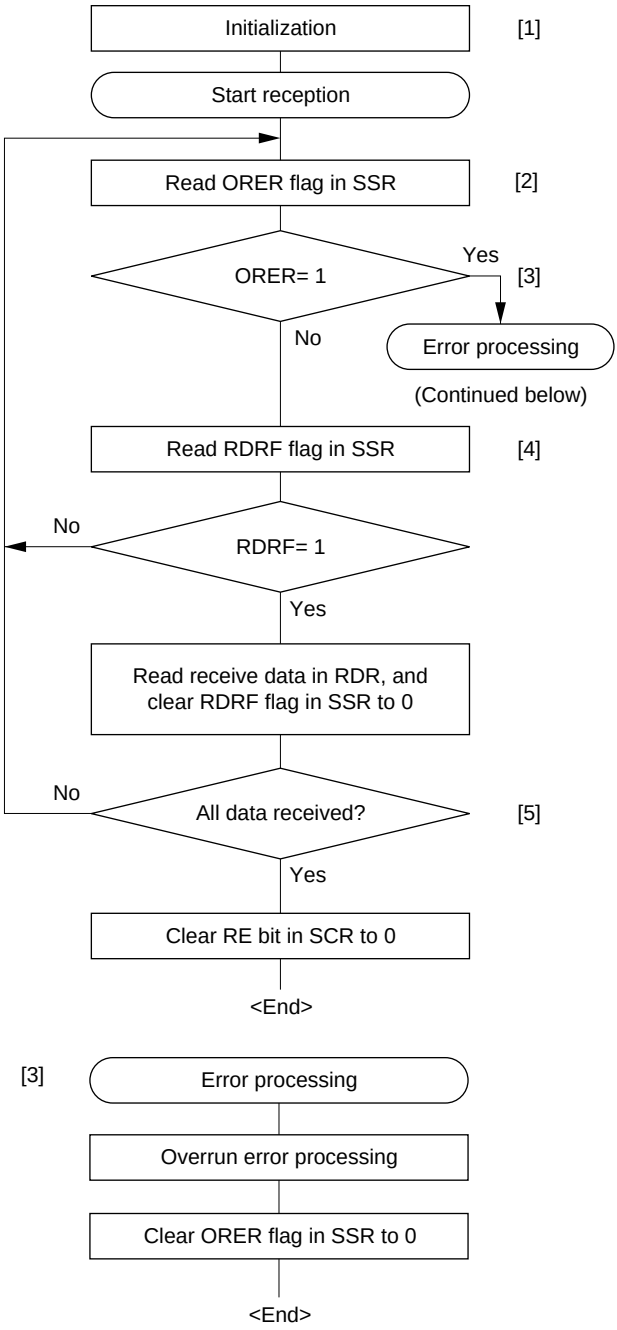
- Serial data reception (clocked synchronous mode)

  Figure 13-18 shows a sample flowchart for serial reception.

  The following procedure should be used for serial data reception.

  When changing the operating mode from asynchronous to clocked synchronous, be sure to check that the ORER, PER, and FER flags are all cleared to 0.

  The RDRF flag will not be set if the FER or PER flag is set to 1, and neither transmit nor receive operations will be possible.



[1] SCI initialization:
The RxD pin is automatically designated as the receive data input pin.

[2] [3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR , and after performing the appropriate error processing, clear the ORER flag to 0. Transfer cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial reception continuation procedure:
To continue serial reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. The RDRF flag is cleared automatically when the DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Figure 13-18 Sample Serial Reception Flowchart

In serial reception, the SCI operates as described below.

[1] The SCI performs internal initialization in synchronization with serial clock input or output.

[2] The received data is stored in RSR in LSB-to-MSB order.

After reception, the SCI checks whether the RDRF flag is 0 and the receive data can be transferred from RSR to RDR.

If this check is passed, the RDRF flag is set to 1, and the receive data is stored in RDR. If a receive error is detected in the error check, the operation is as shown in table 13-11.

Neither transmit nor receive operations can be performed subsequently when a receive error has been found in the error check.

[3] If the RIE bit in SCR is set to 1 when the RDRF flag changes to 1, a receive data full interrupt (RXI) request is generated.

Also, if the RIE bit in SCR is set to 1 when the ORER flag changes to 1, a receive error interrupt (ERI) request is generated.

Figure 13-19 shows an example of SCI operation in reception.

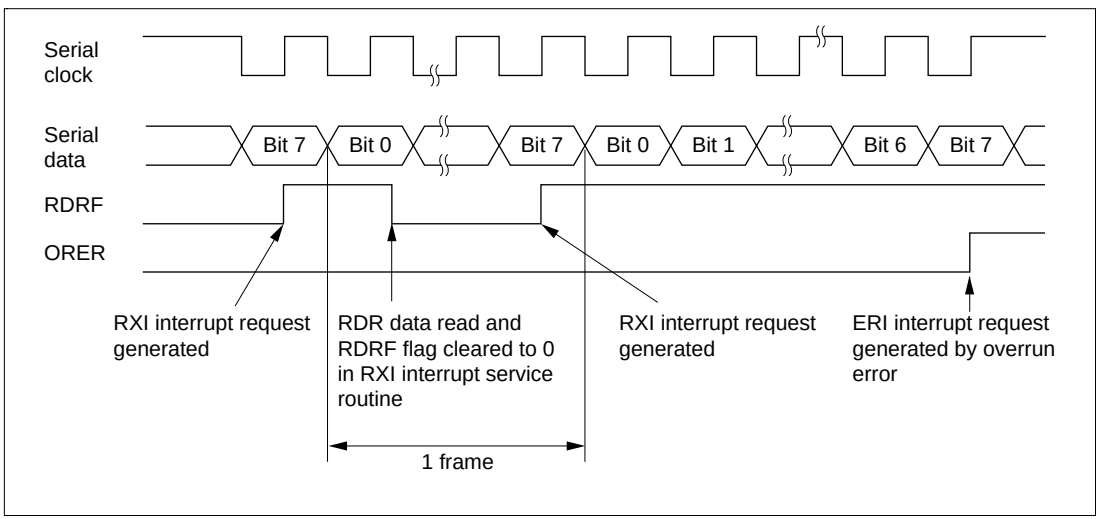
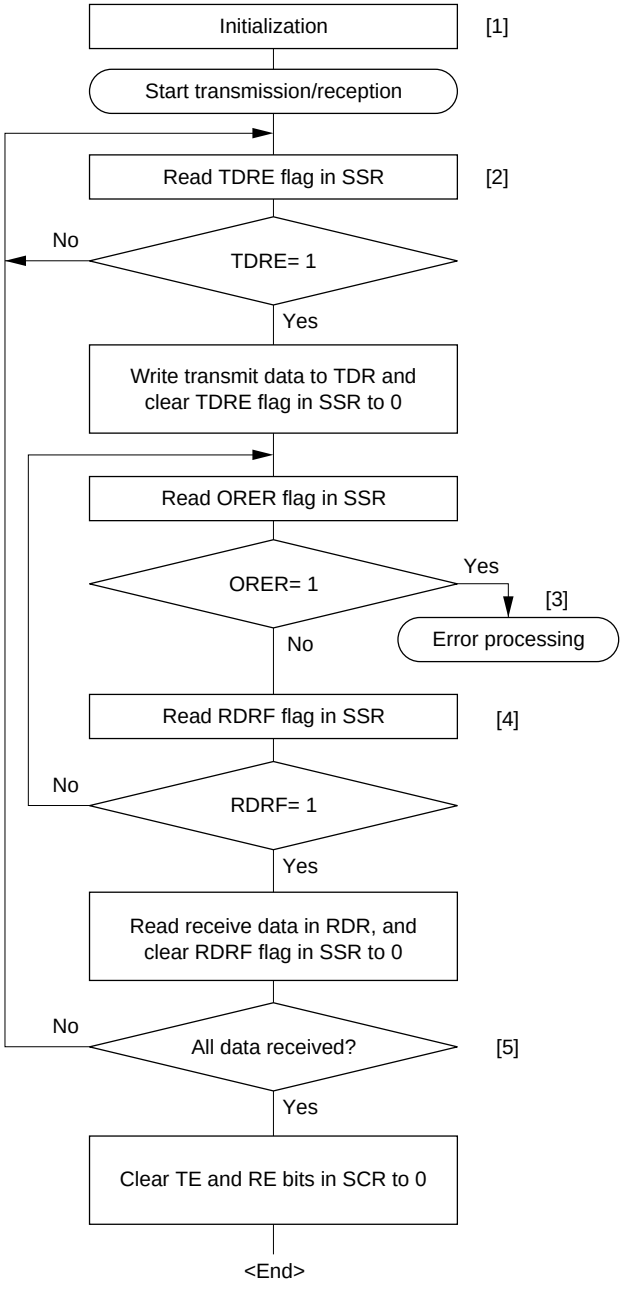


Figure 13-19 Example of SCI Operation in Reception

- Simultaneous serial data transmission and reception (clocked synchronous mode)

  Figure 13-20 shows a sample flowchart for simultaneous serial transmit and receive operations.

  The following procedure should be used for simultaneous serial data transmit and receive operations.



[1] SCI initialization:
The TxD pin is designated as the transmit data output pin, and the RxD pin is designated as the receive data input pin, enabling simultaneous transmit and receive operations.

[2] SCI status check and transmit data write:
Read SSR and check that the TDRE flag is set to 1, then write transmit data to TDR and clear the TDRE flag to 0.
Transition of the TDRE flag from 0 to 1 can also be identified by a TXI interrupt.

[3] Receive error processing:
If a receive error occurs, read the ORER flag in SSR , and after performing the appropriate error processing, clear the ORER flag to 0. Transmission/reception cannot be resumed if the ORER flag is set to 1.

[4] SCI status check and receive data read:
Read SSR and check that the RDRF flag is set to 1, then read the receive data in RDR and clear the RDRF flag to 0. Transition of the RDRF flag from 0 to 1 can also be identified by an RXI interrupt.

[5] Serial transmission/reception continuation procedure:
To continue serial transmission/ reception, before the MSB (bit 7) of the current frame is received, finish reading the RDRF flag, reading RDR, and clearing the RDRF flag to 0. Also, before the MSB (bit 7) of the current frame is transmitted, read 1 from the TDRE flag to confirm that writing is possible. Then write data to TDR and clear the TDRE flag to 0.
Checking and clearing of the TDRE flag is automatic when the DTC is activated by a transmit data empty interrupt (TXI) request and data is written to TDR. Also, the RDRF flag is cleared automatically when the DTC is activated by a receive data full interrupt (RXI) request and the RDR value is read.

Note: When switching from transmit or receive operation to simultaneous transmit and receive operations, first clear the TE bit and RE bit to 0, then set both these bits to 1 simultaneously.

Figure 13-20 Sample Flowchart of Simultaneous Serial Transmit and Receive Operations

13.4 SCI Interrupts

The SCI has four interrupt sources: the transmit-end interrupt (TEI) request, receive-error interrupt (ERI) request, receive-data-full interrupt (RXI) request, and transmit-data-empty interrupt (TXI) request. Table 13-13 shows the interrupt sources and their relative priorities. Individual interrupt sources can be enabled or disabled with the TIE, RIE, and TEIE bits in the SCR. Each kind of interrupt request is sent to the interrupt controller independently.

When the TDRE flag in SSR is set to 1, a TXI interrupt request is generated. When the TEND flag in SSR is set to 1, a TEI interrupt request is generated. A TXI interrupt can activate the DTC to perform data transfer. The TDRE flag is cleared to 0 automatically when data transfer is performed by the DTC. The DTC cannot be activated by a TEI interrupt request.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated. When the ORER, PER, or FER flag in SSR is set to 1, an ERI interrupt request is generated. An RXI interrupt can activate the DTC to perform data transfer. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC. The DTC cannot be activated by an ERI interrupt request.

Table 13-12 SCI Interrupt Sources

Channel	Interrupt Source	Description	DTC Activation	Priority*
0	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	High
	RXI	Interrupt due to receive data full state (RDRF)	Possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	
1	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	
	RXI	Interrupt due to receive data full state (RDRF)	Possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	
2	ERI	Interrupt due to receive error (ORER, FER, or PER)	Not possible	
	RXI	Interrupt due to receive data full state (RDRF)	Possible	
	TXI	Interrupt due to transmit data empty state (TDRE)	Possible	
	TEI	Interrupt due to transmission end (TEND)	Not possible	Low

Note: * This table shows the initial state immediately after a reset. Relative priorities among channels can be changed by means of the interrupt controller.

A TEI interrupt is requested when the TEND flag is set to 1 while the TEIE bit is set to 1. The TEND flag is cleared at the same time as the TDRE flag. Consequently, if a TEI interrupt and a TXI interrupt are requested simultaneously, the TXI interrupt may have priority for acceptance,

with the result that the TDRE and TEND flags are cleared. Note that the TEI interrupt will not be accepted in this case.

13.5 Usage Notes

The following points should be noted when using the SCI.

Relation between Writes to TDR and the TDRE Flag
The TDRE flag in SSR is a status flag that indicates that transmit data has been transferred from TDR to TSR. When the SCI transfers data from TDR to TSR, the TDRE flag is set to 1.

Data can be written to TDR regardless of the state of the TDRE flag. However, if new data is written to TDR when the TDRE flag is cleared to 0, the data stored in TDR will be lost since it has not yet been transferred to TSR. It is therefore essential to check that the TDRE flag is set to 1 before writing transmit data to TDR.

Operation when Multiple Receive Errors Occur Simultaneously
If a number of receive errors occur at the same time, the state of the status flags in SSR is as shown in table 13-14. If there is an overrun error, data is not transferred from RSR to RDR, and the receive data is lost.

Table 13-13 State of SSR Status Flags and Transfer of Receive Data

SSR Status Flags				Receive Data Transfer	
RDRF	ORER	FER	PER	RSR to RDR	Receive Error Status
1	1	0	0	X	Overrun error
0	0	1	0	◯	Framing error
0	0	0	1	◯	Parity error
1	1	1	0	X	Overrun error + framing error
1	1	0	1	X	Overrun error + parity error
0	0	1	1	◯	Framing error + parity error
1	1	1	1	X	Overrun error + framing error + parity error

Notes: ◯: Receive data is transferred from RSR to RDR.
X: Receive data is not transferred from RSR to RDR.

Break Detection and Processing (Asynchronous Mode Only): When framing error (FER) detection is performed, a break can be detected by reading the RxD pin value directly. In a break, the input from the RxD pin becomes all 0s, and so the FER flag is set, and the parity error flag (PER) may also be set.

Note that, since the SCI continues the receive operation after receiving a break, even if the FER flag is cleared to 0, it will be set to 1 again.

Sending a Break (Asynchronous Mode Only): The TxD pin has a dual function as an I/O port whose direction (input or output) is determined by DR and DDR. This can be used to send a break.

Between serial transmission initialization and setting of the TE bit to 1, the mark state is replaced by the value of DR (the pin does not function as the TxD pin until the TE bit is set to 1). Consequently, DDR and DR for the port corresponding to the TxD pin are first set to 1.

To send a break during serial transmission, first clear DR to 0, then clear the TE bit to 0.

When the TE bit is cleared to 0, the transmitter is initialized regardless of the current transmission state, the TxD pin becomes an I/O port, and 0 is output from the TxD pin.

Receive Error Flags and Transmit Operations (Clocked Synchronous Mode Only): Transmission cannot be started when a receive error flag (ORER, PER, or FER) is set to 1, even if the TDRE flag is cleared to 0. Be sure to clear the receive error flags to 0 before starting transmission.

Note also that receive error flags cannot be cleared to 0 even if the RE bit is cleared to 0.

Receive Data Sampling Timing and Reception Margin in Asynchronous Mode: In asynchronous mode, the SCI operates on a basic clock with a frequency of 16 times the transfer rate.

In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 8th pulse of the basic clock. This is illustrated in figure 13-21.

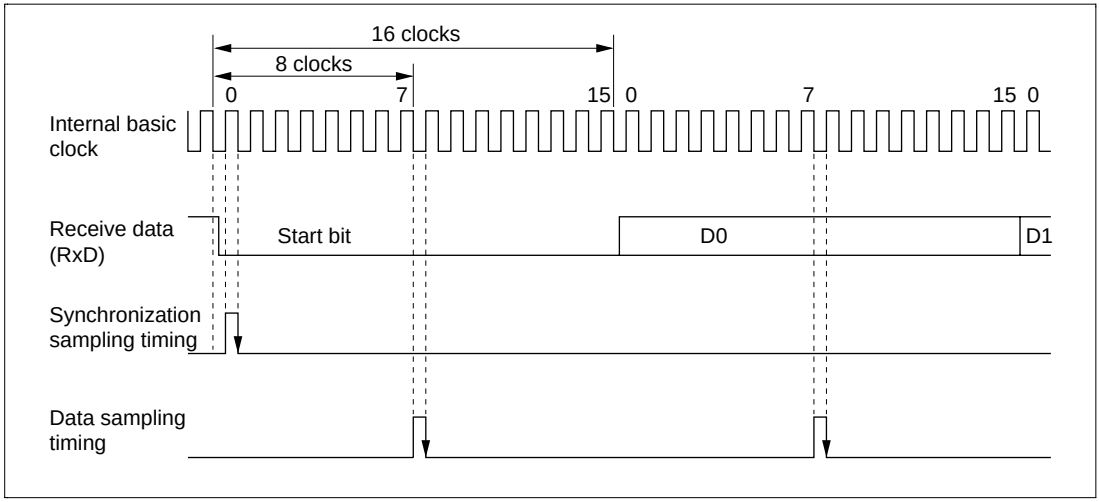


Figure 13-21 Receive Data Sampling Timing in Asynchronous Mode

Thus the reception margin in asynchronous mode is given by formula (1) below.

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

... Formula (1)

Where M : Reception margin (%)
N : Ratio of bit rate to clock (N = 16)
D : Clock duty (D = 0 to 1.0)
L : Frame length (L = 9 to 12)
F : Absolute value of clock rate deviation

Assuming values of F = 0 and D = 0.5 in formula (1), a reception margin of 46.875% is given by formula (2) below.

When D = 0.5 and F = 0,

$$M = (0.5 - \frac{1}{2 \times 16}) \times 100\%$$

$$= 46.875\%$$

... Formula (2)

However, this is only the computed value, and a margin of 20% to 30% should be allowed in system design.

Restrictions on Use of DTC

- When an external clock source is used as the serial clock, the transmit clock should not be input until at least 5 ø clock cycles after TDR is updated by the DTC. Misoperation may occur if the transmit clock is input within 4 ø clocks after TDR is updated. (Figure 13-22)
- When RDR is read by the DTC, be sure to set the activation source to the relevant SCI reception end interrupt (RXI).

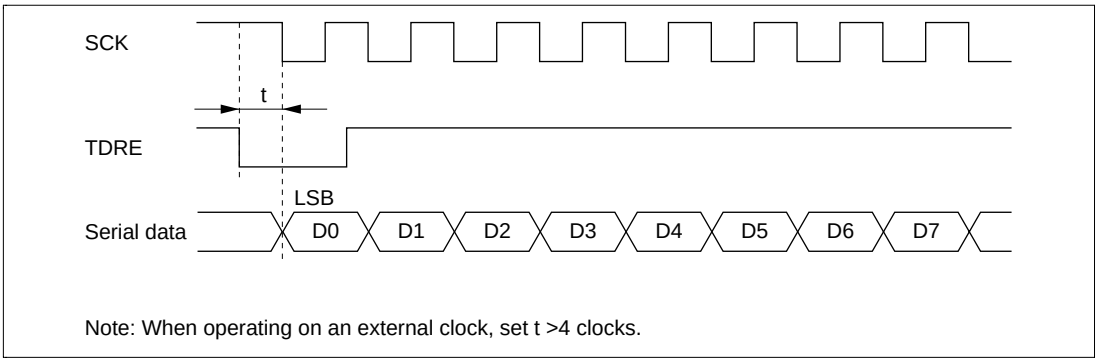


Figure 13-22 Example of Clocked Synchronous Transmission by DTC

Operation in Case of Mode Transition

- Transmission

  Operation should be stopped (by clearing TE, TIE, and TEIE to 0) before making a module stop mode, software standby mode, watch mode, subactive mode, or subsleep mode transition. TSR, TDR, and SSR are reset. The output pin states in module stop mode, software standby mode, watch mode, subactive mode, or subsleep mode depend on the port settings, and becomes high-level output after the relevant mode is cleared. If a transition is made during transmission, the data being transmitted will be undefined. When transmitting without changing the transmit mode after the relevant mode is cleared, transmission can be started by setting TE to 1 again, and performing the following sequence: SSR read -> TDR write -> TDRE clearance. To transmit with a different transmit mode after clearing the relevant mode, the procedure must be started again from initialization. Figure 13-23 shows a sample flowchart for mode transition during transmission. Port pin states are shown in figures 13-24 and 13-25.

  Operation should also be stopped (by clearing TE, TIE, and TEIE to 0) before making a transition from transmission by DTC transfer to module stop mode, software standby mode, watch mode, subactive mode, or subsleep mode transition. To perform transmission with the DTC after the relevant mode is cleared, setting TE and TIE to 1 will set the TXI flag and start DTC transmission.

- Reception

  Receive operation should be stopped (by clearing RE to 0) before making a module stop mode, software standby mode, watch mode, subactive mode, or subsleep mode transition. RSR, RDR, and SSR are reset. If a transition is made without stopping operation, the data being received will be invalid.

  To continue receiving without changing the reception mode after the relevant mode is cleared, set RE to 1 before starting reception. To receive with a different receive mode, the procedure must be started again from initialization.

  Figure 13-26 shows a sample flowchart for mode transition during reception.

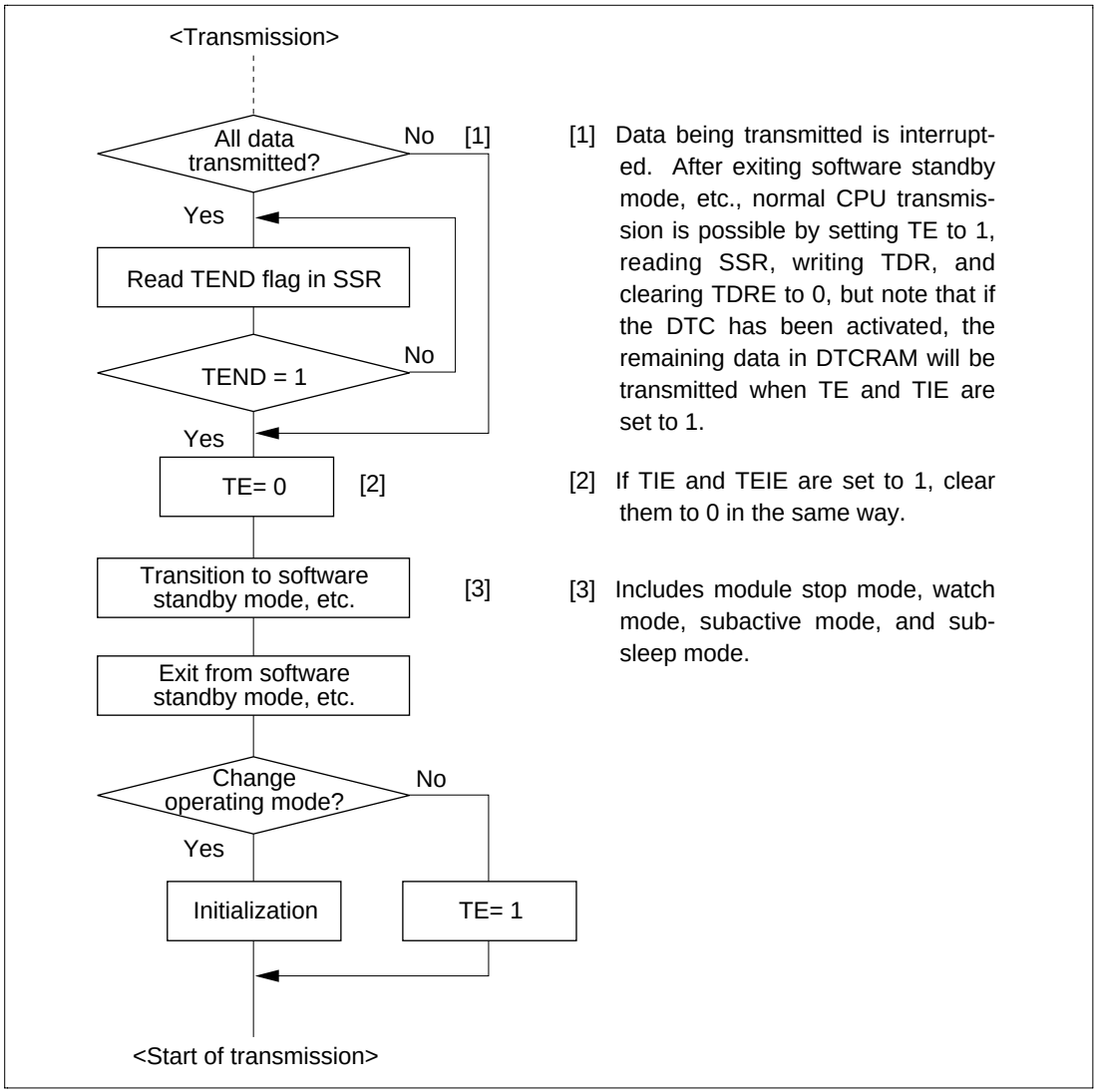


Figure 13-23 Sample Flowchart for Mode Transition during Transmission

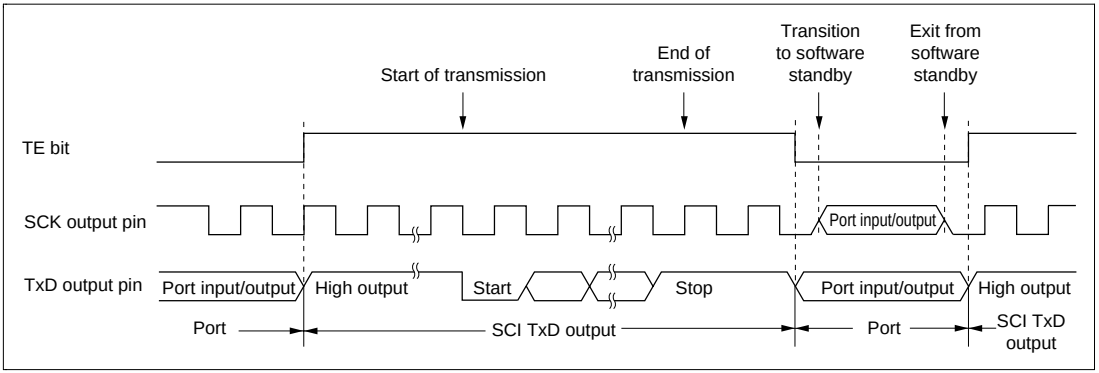
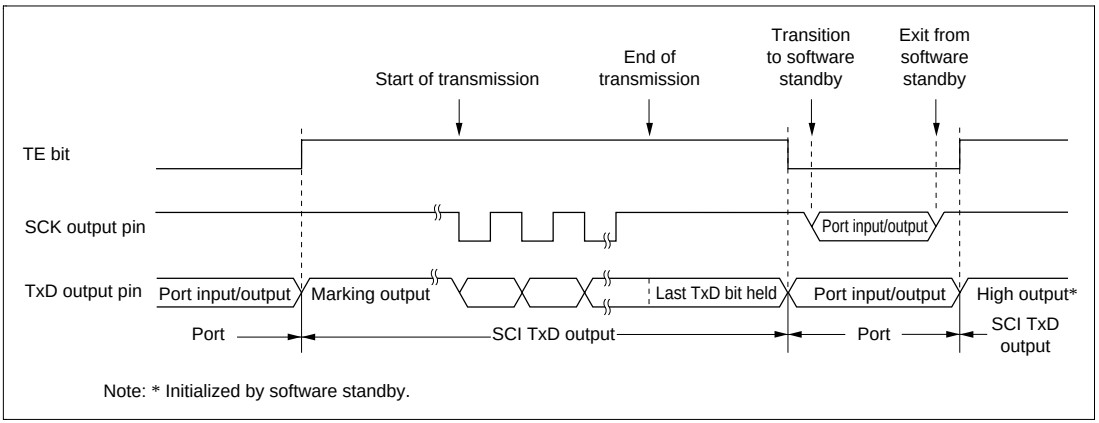


Figure 13-24 Asynchronous Transmission Using Internal Clock



Note: * Initialized by software standby.

Figure 13-25 Synchronous Transmission Using Internal Clock

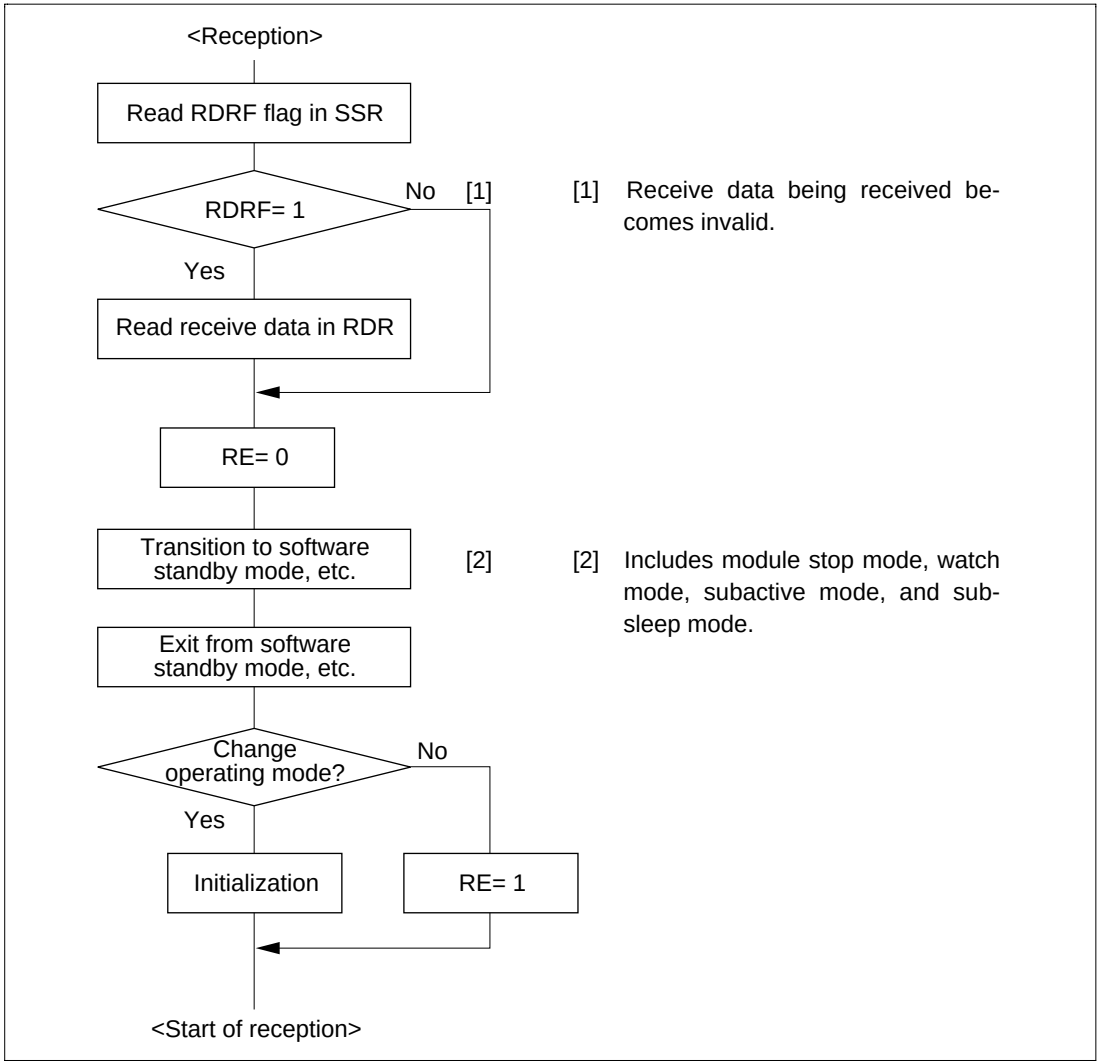


Figure 13-26 Sample Flowchart for Mode Transition during Reception

Switching from SCK Pin Function to Port Pin Function:

- Problem in Operation: When switching the SCK pin function to the output port function (high-level output) by making the following settings while DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1 (synchronous mode), low-level output occurs for one half-cycle.

1. End of serial data transmission
2. TE bit = 0
3. C/$\overline{A}$ bit = 0 ... switchover to port output
4. Occurrence of low-level output (see figure 13-27)

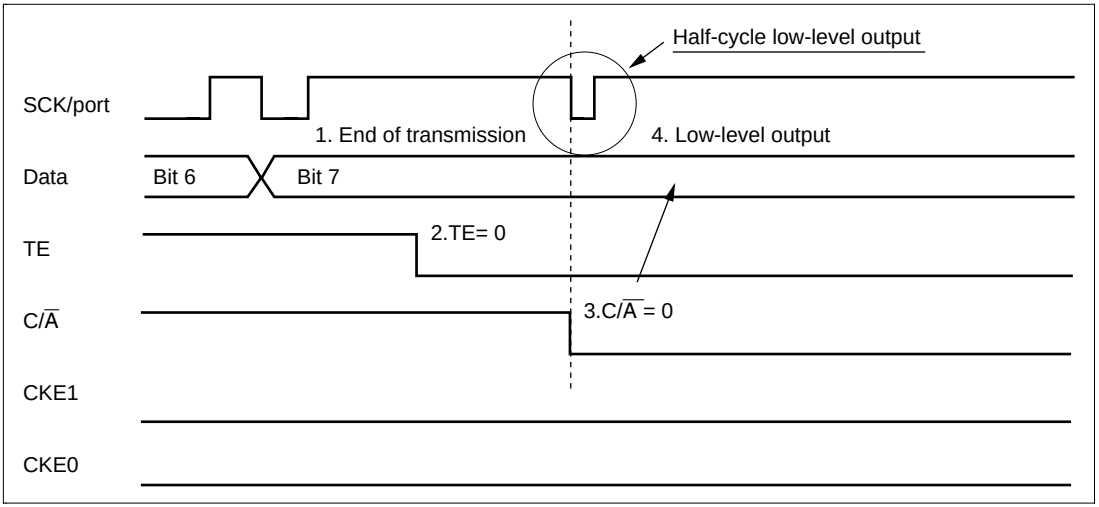


Figure 13-27 Operation when Switching from SCK Pin Function to Port Pin Function

- Sample Procedure for Avoiding Low-Level Output: As this sample procedure temporarily places the SCK pin in the input state, the SCK/port pin should be pulled up beforehand with an external circuit.

  With DDR = 1, DR = 1, C/$\overline{A}$ = 1, CKE1 = 0, CKE0 = 0, and TE = 1, make the following settings in the order shown.

1. End of serial data transmission
2. TE bit = 0
3. CKE1 bit = 1
4. C/$\overline{A}$ bit = 0 ... switchover to port output
5. CKE1 bit = 0

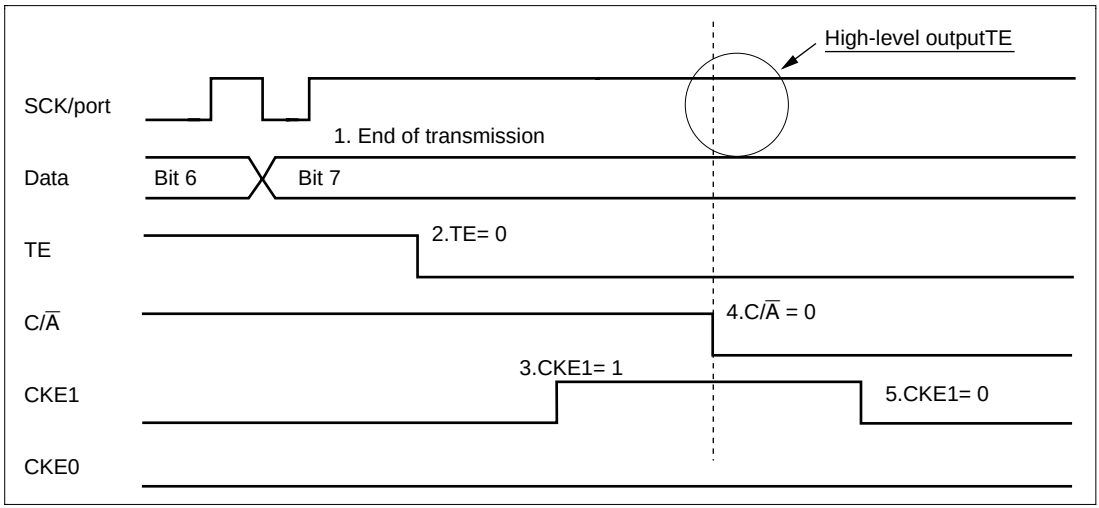


Figure 13-28 Operation when Switching from SCK Pin Function to Port Pin Function (Example of Preventing Low-Level Output)

Section 14 Smart Card Interface

14.1 Overview

SCI supports an IC card (Smart Card) interface conforming to ISO/IEC 7816-3 (Identification Card) as a serial communication interface extension function.

Switching between the normal serial communication interface and the Smart Card interface is carried out by means of a register setting.

14.1.1 Features

Features of the Smart Card interface supported by the chip are as follows.

- Asynchronous mode
  - — Data length: 8 bits
  - — Parity bit generation and checking
  - — Transmission of error signal (parity error) in receive mode
  - — Error signal detection and automatic data retransmission in transmit mode
  - — Direct convention and inverse convention both supported
- On-chip baud rate generator allows any bit rate to be selected
- Three interrupt sources
  - — Three interrupt sources (transmit data empty, receive data full, and transmit/receive error) that can issue requests independently
  - — The transmit data empty interrupt and receive data full interrupt can activate the data transfer controller (DTC) to execute data transfer

14.1.2 Block Diagram

Figure 14-1 shows a block diagram of the Smart Card interface.

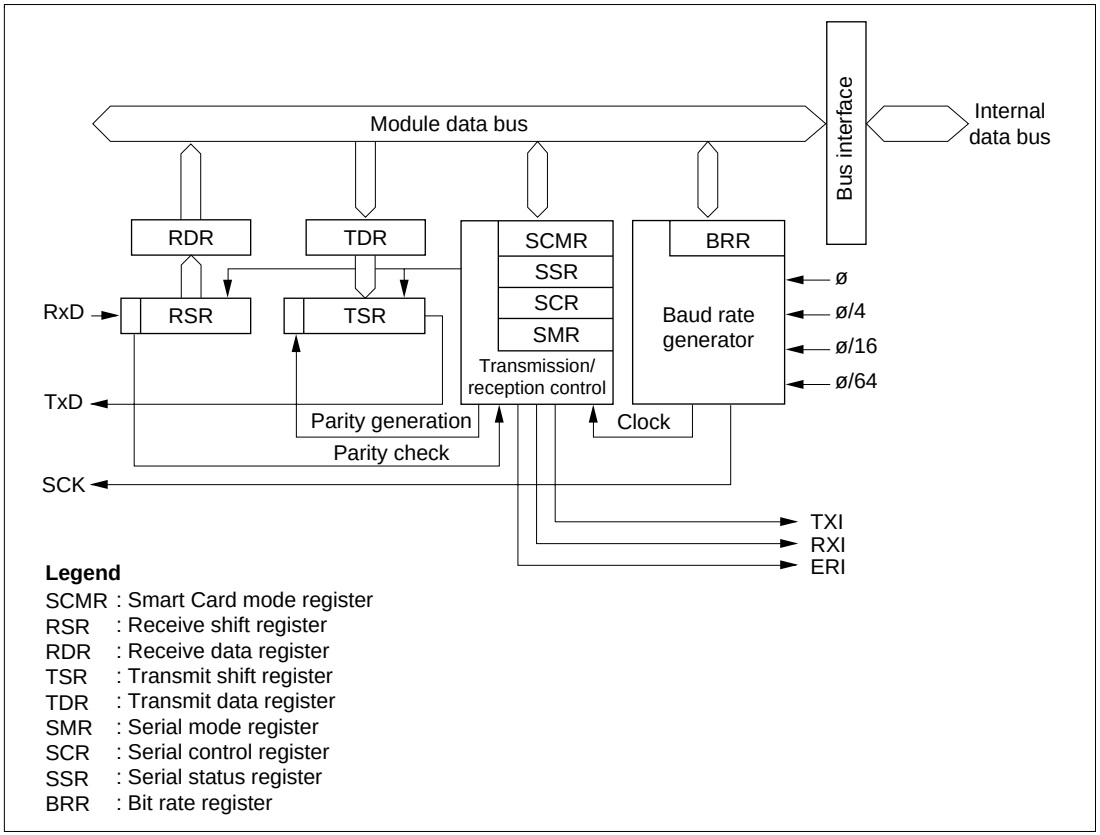


Figure 14-1 Block Diagram of Smart Card Interface

14.1.3 Pin Configuration

Table 14-1 shows the Smart Card interface pin configuration.

Table 14-1 Smart Card Interface Pins

Channel	Pin Name	Symbol	I/O	Function
0	Serial clock pin 0	SCK0	I/O	SCI0 clock input/output
	Receive data pin 0	RxD0	Input	SCI0 receive data input
	Transmit data pin 0	TxD0	Output	SCI0 transmit data output
1	Serial clock pin 1	SCK1	I/O	SCI1 clock input/output
	Receive data pin 1	RxD1	Input	SCI1 receive data input
	Transmit data pin 1	TxD1	Output	SCI1 transmit data output
2	Serial clock pin 2	SCK2	I/O	SCI2 clock input/output
	Receive data pin 2	RxD2	Input	SCI2 receive data input
	Transmit data pin 2	TxD2	Output	SCI2 transmit data output

14.1.4 Register Configuration

Table 14-2 shows the registers used by the Smart Card interface. Details of SMR, BRR, SCR, TDR, RDR, and MSTPCR are the same as for the normal SCI function: see the register descriptions in section 13, Serial Communication Interface.

Table 14-2 Smart Card Interface Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
0	Serial mode register 0	SMR0	R/W	H'00	H'FF78
	Bit rate register 0	BRR0	R/W	H'FF	H'FF79
	Serial control register 0	SCR0	R/W	H'00	H'FF7A
	Transmit data register 0	TDR0	R/W	H'FF	H'FF7B
	Serial status register 0	SSR0	R/(W)*[2]	H'84	H'FF7C
	Receive data register 0	RDR0	R	H'00	H'FF7D
	Smart card mode register 0	SCMR0	R/W	H'F2	H'FF7E
1	Serial mode register 1	SMR1	R/W	H'00	H'FF80
	Bit rate register 1	BRR1	R/W	H'FF	H'FF81
	Serial control register 1	SCR1	R/W	H'00	H'FF82
	Transmit data register 1	TDR1	R/W	H'FF	H'FF83
	Serial status register 1	SSR1	R/(W)*[2]	H'84	H'FF84
	Receive data register 1	RDR1	R	H'00	H'FF85
	Smart card mode register 1	SCMR1	R/W	H'F2	H'FF86
2	Serial mode register 2	SMR2	R/W	H'00	H'FF88
	Bit rate register 2	BRR2	R/W	H'FF	H'FF89
	Serial control register 2	SCR2	R/W	H'00	H'FF8A
	Transmit data register 2	TDR2	R/W	H'FF	H'FF8B
	Serial status register 2	SSR2	R/(W)*[2]	H'84	H'FF8C
	Receive data register 2	RDR2	R	H'00	H'FF8D
	Smart card mode register 2	SCMR2	R/W	H'F2	H'FF8E
All	Module stop control register B	MSTPCRB	R/W	H'FF	H'FDE9

Notes: *1 Lower 16 bits of the address.
*2 Can only be written with 0 for flag clearing.

14.2 Register Descriptions

Registers added with the Smart Card interface and bits for which the function changes are described here.

14.2.1 Smart Card Mode Register (SCMR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	SDIR	SINV	—	SMIF
Initial value	:	1	1	1	1	0	0	1	0
R/W	:	—	—	—	—	R/W	R/W	—	R/W

SCMR is an 8-bit readable/writable register that selects the Smart Card interface function.

SCMR is initialized to H'F2 by a reset and in standby mode.

Bits 7 to 4—Reserved: These bits are always read as 1 and cannot be modified.

Bit 3—Smart Card Data Transfer Direction (SDIR): Selects the serial/parallel conversion format.

Bit 3 SDIR	Description	
0	TDR contents are transmitted LSB-first	(Initial value)
	Receive data is stored in RDR LSB-first	
1	TDR contents are transmitted MSB-first	
	Receive data is stored in RDR MSB-first	

Bit 2—Smart Card Data Invert (SINV): Specifies inversion of the data logic level. This function is used together with the SDIR bit for communication with an inverse convention card. The SINV bit does not affect the logic level of the parity bit. For parity-related setting procedures, see section 14.3.4, Register Settings.

Bit 2 SINV	Description	
0	TDR contents are transmitted as they are	(Initial value)
	Receive data is stored as it is in RDR	
1	TDR contents are inverted before being transmitted	
	Receive data is stored in inverted form in RDR	

Bit 1—Reserved: This bit is always read as 1 and cannot be modified.

Bit 0—Smart Card Interface Mode Select (SMIF): Enables or disables the Smart Card interface function.

Bit 0 SMIF	Description	
0	Smart Card interface function is disabled	(Initial value)
1	Smart Card interface function is enabled	

14.2.2 Serial Status Register (SSR)

Bit	:	7	6	5	4	3	2	1	0
		TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT
Initial value	:	1	0	0	0	0	1	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/W

Note: * Only 0 can be written, to clear these flags.

Bit 4 of SSR has a different function in Smart Card interface mode. Coupled with this, the setting conditions for bit 2, TEND, are also different.

Bits 7 to 5—Operate in the same way as for the normal SCI. For details, see section 13.2.7, Serial Status Register (SSR).

Bit 4—Error Signal Status (ERS): In Smart Card interface mode, bit 4 indicates the status of the error signal sent back from the receiving end in transmission. Framing errors are not detected in Smart Card interface mode.

Bit 4 ERS	Description
0	Normal reception, with no error signal [Clearing condition] (Initial value) • Upon reset, and in standby mode or module stop mode • When 0 is written to ERS after reading ERS = 1
1	Error signal sent from receiver indicating detection of parity error [Setting condition] When the Low level of the error signal is sampled

Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its previous state.

Bits 3 to 0—Operate in the same way as for the normal SCI. For details, see section 13.2.7, Serial Status Register (SSR).

However, the setting conditions for the TEND bit, are as shown below.

Bit 2 TEND	Description
0	Transmission is in progress [Clearing conditions] (Initial value) • When 0 is written to TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and write data to TDR
1	Transmission has ended [Setting conditions] • Upon reset, and in standby mode or module stop mode • When the TE bit in SCR is 0 and the ERS bit is also 0 • When TDRE = 1 and ERS = 0 (normal transmission) 2.5 etu after transmission of a 1-byte serial character when GM = 0 and BLK = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.5 etu after transmission of a 1-byte serial character when GM = 0 and BLK = 1 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after transmission of a 1-byte serial character when GM = 1 and BLK = 0 • When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after transmission of a 1-byte serial character when GM = 1 and BLK = 1

Note: etu: Elementary Time Unit (time for transfer of 1 bit)

14.2.3 Serial Mode Register (SMR)

Bit	:	7	6	5	4	3	2	1	0
		GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: When the smart card interface is used, be sure to make the 1 setting shown for bit 5.

The function of bits 7, 6, 3, and 2 of SMR changes in Smart Card interface mode.

Bit 7—GSM Mode (GM): Sets the smart card interface function to GSM mode.

This bit is cleared to 0 when the normal smart card interface is used. In GSM mode, this bit is set to 1, the timing of setting of the TEND flag that indicates transmission completion is advanced and clock output control mode addition is performed. The contents of the clock output control mode addition are specified by bits 1 and 0 of the serial control register (SCR).

Bit 7 GM	**Description**
0	Normal smart card interface mode operation (Initial value) • TEND flag generation 12.5 etu (11.5 etu in block transfer mode) after beginning of start bit • Clock output ON/OFF control only
1	GSM mode smart card interface mode operation • TEND flag generation 11.0 etu after beginning of start bit • High/Low fixing control possible in addition to clock output ON/OFF control (set by SCR)

Note: etu: Elementary time unit (time for transfer of 1 bit)

Bit 6—Block Transfer Mode (BLK): Selects block transfer mode.

Bit 6	
BLK	**Description**
0	Normal Smart Card interface mode operation • Error signal transmission/detection and automatic data retransmission performed • TXI interrupt generated by TEND flag • TEND flag set 12.5 etu after start of transmission (11.0 etu in GSM mode)
1	Block transfer mode operation • Error signal transmission/detection and automatic data retransmission not performed • TXI interrupt generated by TDRE flag • TEND flag set 11.5 etu after start of transmission (11.0 etu in GSM mode)

Bits 3 and 2—Basic Clock Pulse 1 and 2 (BCP1, BCP0): These bits specify the number of basic clock periods in a 1-bit transfer interval on the Smart Card interface.

Bit 3	Bit 2		
BCP1	**BCP0**	**Description**	
0	0	32 clock periods	(Initial value)
	1	64 clock periods	
1	0	372 clock periods	
	1	256 clock periods	

Bits 5, 4, 1, and 0: Operate in the same way as for the normal SCI. For details, see section 13.2.5, serial mode register (SMR).

14.2.4 Serial Control Register (SCR)

Bit	:	7	6	5	4	3	2	1	0
		TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

In smart card interface mode, the function of bits 1 and 0 of SCR changes when bit 7 of the serial mode register (SMR) is set to 1.

Bits 7 to 2—Operate in the same way as for the normal SCI.

For details, see section 13.2.6, Serial Control Register (SCR).

Bits 1 and 0—Clock Enable 1 and 0 (CKE1, CKE0): These bits are used to select the SCI clock source and enable or disable clock output from the SCK pin.

In smart card interface mode, in addition to the normal switching between clock output enabling and disabling, the clock output can be specified as to be fixed high or low.

SCMR	SMR	SCR Setting		
SMIF	**C/$\overline{A}$, GM**	**CKE1**	**CKE0**	**SCK Pin Function**
0	See the SCI			
1	0	0	0	Operates as port I/O pin
1	0	0	1	Outputs clock as SCK output pin
1	1	0	0	Operates as SCK output pin, with output fixed low
1	1	0	1	Outputs clock as SCK output pin
1	1	1	0	Operates as SCK output pin, with output fixed high
1	1	1	1	Outputs clock as SCK output pin

14.3 Operation

14.3.1 Overview

The main functions of the Smart Card interface are as follows.

- One frame consists of 8-bit data plus a parity bit.
- In transmission, a guard time of at least 2 etu (Elementary Time Unit: the time for transfer of one bit) is left between the end of the parity bit and the start of the next frame.
- If a parity error is detected during reception, a low error signal level is output for one etu period, 10.5 etu after the start bit.
- If the error signal is sampled during transmission, the same data is transmitted automatically after the elapse of 2 etu or longer. (except in block transfer mode)
- Only asynchronous communication is supported; there is no clocked synchronous communication function.

14.3.2 Pin Connections

Figure 14-2 shows a schematic diagram of Smart Card interface related pin connections.

In communication with an IC card, since both transmission and reception are carried out on a single data transmission line, the TxD pin and RxD pin should be connected with the LSI pin. The data transmission line should be pulled up to the V_{CC} power supply with a resistor.

When the clock generated on the Smart Card interface is used by an IC card, the SCK pin output is input to the CLK pin of the IC card. No connection is needed if the IC card uses an internal clock.

LSI port output is used as the reset signal.

Other pins must normally be connected to the power supply or ground.

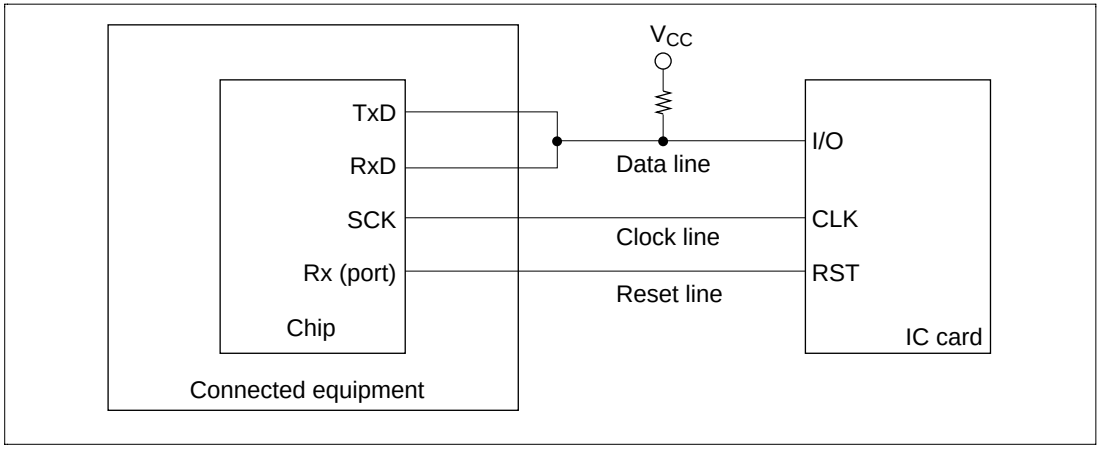


Figure 14-2 Schematic Diagram of Smart Card Interface Pin Connections

Note: If an IC card is not connected, and the TE and RE bits are both set to 1, closed transmission/reception is possible, enabling self-diagnosis to be carried out.

14.3.3 Data Format

Normal Transfer Mode: Figure 14-3 shows the normal Smart Card interface data format. In reception in this mode, a parity check is carried out on each frame, and if an error is detected an error signal is sent back to the transmitting end, and retransmission of the data is requested. If an error signal is sampled during transmission, the same data is retransmitted.

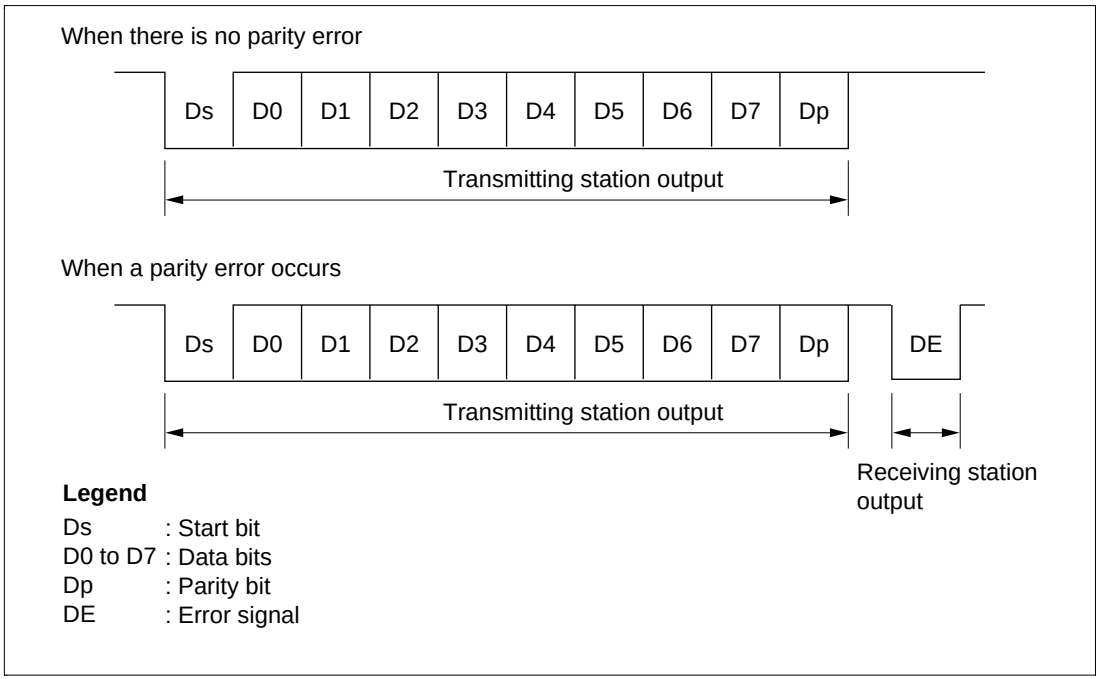


Figure 14-3 Normal Smart Card Interface Data Format

The operation sequence is as follows.

[1] When the data line is not in use it is in the high-impedance state, and is fixed high with a pull-up resistor.

[2] The transmitting station starts transfer of one frame of data. The data frame starts with a start bit (Ds, low-level), followed by 8 data bits (D0 to D7) and a parity bit (Dp).

[3] With the Smart Card interface, the data line then returns to the high-impedance state. The data line is pulled high with a pull-up resistor.

[4] The receiving station carries out a parity check.

If there is no parity error and the data is received normally, the receiving station waits for reception of the next data.

If a parity error occurs, however, the receiving station outputs an error signal (DE, low-level) to request retransmission of the data. After outputting the error signal for the prescribed length of time, the receiving station places the signal line in the high-impedance state again. The signal line is pulled high again by a pull-up resistor.

[5] If the transmitting station does not receive an error signal, it proceeds to transmit the next data frame.

If it does receive an error signal, however, it returns to step [2] and retransmits the erroneous data.

Block Transfer Mode: The operation sequence in block transfer mode is as follows.

[1] When the data line in not in use it is in the high-impedance state, and is fixed high with a pull-up resistor.

[2] The transmitting station starts transfer of one frame of data. The data frame starts with a start bit (Ds, low-level), followed by 8 data bits (D0 to D7) and a parity bit (Dp).

[3] With the Smart Card interface, the data line then returns to the high-impedance state. The data line is pulled high with a pull-up resistor.

[4] After reception, a parity error check is carried out, but an error signal is not output even if an error has occurred. When an error occurs reception cannot be continued, so the error flag should be cleared to 0 before the parity bit of the next frame is received.

[5] The transmitting station proceeds to transmit the next data frame.

14.3.4 Register Settings

Table 14-3 shows a bit map of the registers used by the smart card interface.

Bits indicated as 0 or 1 must be set to the value shown. The setting of other bits is described below.

Table 14-3 Smart Card Interface Register Settings

	Bit							
Register	**Bit 7**	**Bit 6**	**Bit 5**	**Bit 4**	**Bit 3**	**Bit 2**	**Bit 1**	**Bit 0**
SMR	GM	BLK	1	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
BRR	BRR7	BRR6	BRR5	BRR4	BRR3	BRR2	BRR1	BRR0
SCR	TIE	RIE	TE	RE	0	0	CKE1*	CKE0
TDR	TDR7	TDR6	TDR5	TDR4	TDR3	TDR2	TDR1	TDR0
SSR	TDRE	RDRF	ORER	ERS	PER	TEND	0	0
RDR	RDR7	RDR6	RDR5	RDR4	RDR3	RDR2	RDR1	RDR0
SCMR	—	—	—	—	SDIR	SINV	—	SMIF

Notes: — : Unused bit.
*: The CKE1 bit must be cleared to 0 when the GM bit in SMR is cleared to 0.

SMR Setting: The GM bit is cleared to 0 in normal smart card interface mode, and set to 1 in GSM mode. The O/$\overline{E}$ bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

Bits CKS1 and CKS0 select the clock source of the on-chip baud rate generator. Bits BCP1 and BCP0 select the number of basic clock periods in a 1-bit transfer interval. For details, see section 14.3.5, Clock.

The BLK bit is cleared to 0 in normal smart card interface mode, and set to 1 in block transfer mode.

BRR Setting: BRR is used to set the bit rate. See section 14.3.5, Clock, for the method of calculating the value to be set.

SCR Setting: The function of the TIE, RIE, TE, and RE bits is the same as for the normal SCI. For details, see section 13, Serial Communication Interface.

Bits CKE1 and CKE0 specify the clock output. When the GM bit in SMR is cleared to 0, set these bits to B'00 if a clock is not to be output, or to B'01 if a clock is to be output. When the GM bit in SMR is set to 1, clock output is performed. The clock output can also be fixed high or low.

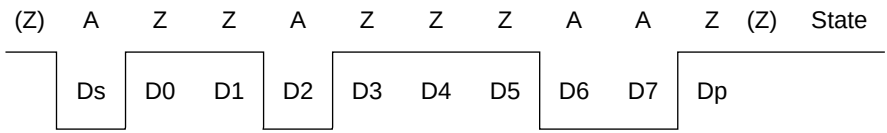
Smart Card Mode Register (SCMR) Setting: The SDIR bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

The SINV bit is cleared to 0 if the IC card is of the direct convention type, and set to 1 if of the inverse convention type.

The SMIF bit is set to 1 in the case of the Smart Card interface.

Examples of register settings and the waveform of the start character are shown below for the two types of IC card (direct convention and inverse convention).

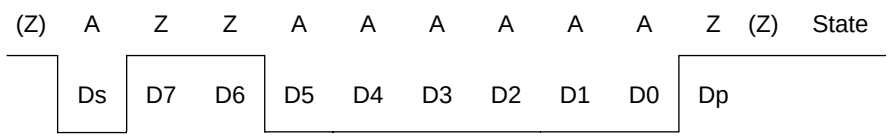
- Direct convention (SDIR = SINV = O/$\overline{E}$ = 0)



With the direct convention type, the logic 1 level corresponds to state Z and the logic 0 level to state A, and transfer is performed in LSB-first order. The start character data above is H'3B.

The parity bit is 1 since even parity is stipulated for the Smart Card.

- Inverse convention (SDIR = SINV = O/$\overline{E}$ = 1)



With the inverse convention type, the logic 1 level corresponds to state A and the logic 0 level to state Z, and transfer is performed in MSB-first order. The start character data above is H'3F.

The parity bit is 0, corresponding to state Z, since even parity is stipulated for the Smart Card.

With the H8S/2636, inversion specified by the SINV bit applies only to the data bits, D7 to D0. For parity bit inversion, the O/$\overline{E}$ bit in SMR is set to odd parity mode (the same applies to both transmission and reception).

14.3.5 Clock

Only an internal clock generated by the on-chip baud rate generator can be used as the transmit/receive clock for the smart card interface. The bit rate is set with BRR and the CKS1, CKS0, BCP1 and BCP0 bits in SMR. The formula for calculating the bit rate is as shown below. Table 14-5 shows some sample bit rates.

If clock output is selected by setting CKE0 to 1, a clock is output from the SCK pin. The clock frequency is determined by the bit rate and the setting of bits BCP1 and BCP0.

$$B = \frac{ø}{S \times 2^{2n+1} \times (N + 1)} \times 10^6$$

Where: N = Value set in BRR ($0 \leq N \leq 255$)
B = Bit rate (bit/s)
ø = Operating frequency (MHz)
n = See table 14-4
S = Number of internal clocks in 1-bit period, set by BCP1 and BCP0

Table 14-4 Correspondence between n and CKS1, CKS0

n	CKS1	CKS0
0	0	0
1		1
2	1	0
3		1

Table 14-5 Examples of Bit Rate B (bit/s) for Various BRR Settings (When n = 0 and S = 372)

	ø (MHz)						
N	**10.00**	**10.714**	**13.00**	**14.285**	**16.00**	**18.00**	**20.00**
0	13441	14400	17473	19200	21505	24194	26882
1	6720	7200	8737	9600	10753	12097	13441
2	4480	4800	5824	6400	7168	8065	8961

Note: Bit rates are rounded to the nearest whole number.

The method of calculating the value to be set in the bit rate register (BRR) from the operating frequency and bit rate, on the other hand, is shown below. N is an integer, $0 \leq N \leq 255$, and the smaller error is specified.

$$N = \frac{ø}{S \times 2^{2n+1} \times B} \times 10^6 - 1$$

Table 14-6 Examples of BRR Settings for Bit Rate B (bit/s) (When n = 0 and S = 372)

	ø (MHz)															
	7.1424		10.00		10.7136		13.00		14.2848		16.00		18.00		20.00	
bit/s	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error	N	Error
9600	0	0.00	1	30	1	25	1	8.99	1	0.00	1	12.01	2	15.99	2	6.60

Note: A blank means no setting is available.

Table 14-7 Maximum Bit Rate at Various Frequencies (Smart Card Interface Mode) (when S = 372)

ø (MHz)	Maximum Bit Rate (bit/s)	N	n
7.1424	9600	0	0
10.00	13441	0	0
10.7136	14400	0	0
13.00	17473	0	0
14.2848	19200	0	0
16.00	21505	0	0
18.00	24194	0	0
20.00	26882	0	0

The bit rate error is given by the following formula:

$$\text{Error (\%)} = \left(\frac{ø}{S \times 2^{2n+1} \times B \times (N + 1)} \times 10^6 - 1 \right) \times 100$$

14.3.6 Data Transfer Operations

Initialization: Before transmitting and receiving data, initialize the SCI as described below. Initialization is also necessary when switching from transmit mode to receive mode, or vice versa.

[1] Clear the TE and RE bits in SCR to 0.

[2] Clear the error flags ERS, PER, and ORER in SSR to 0.

[3] Set the GM, BLK, $\overline{O}$/E, BCP1, BCP0, CKS1, CKS0 bits in SMR. Set the PE bit to 1.

[4] Set the SMIF, SDIR, and SINV bits in SCMR.
When the SMIF bit is set to 1, the TxD and RxD pins are both switched from ports to SCI pins, and are placed in the high-impedance state.

[5] Set the value corresponding to the bit rate in BRR.

[6] Set the CKE0 and CKE1 bits in SCR. Clear the TIE, RIE, TE, RE, MPIE, and TEIE bits to 0.
If the CKE0 bit is set to 1, the clock is output from the SCK pin.

[7] Wait at least one bit interval, then set the TIE, RIE, TE, and RE bits in SCR. Do not set the TE bit and RE bit at the same time, except for self-diagnosis.

Serial Data Transmission: As data transmission in smart card mode involves error signal sampling and retransmission processing, the processing procedure is different from that for the normal SCI. Figure 14-4 shows a flowchart for transmitting, and figure 14-5 shows the relation between a transmit operation and the internal registers.

[1] Perform Smart Card interface mode initialization as described above in Initialization.

[2] Check that the ERS error flag in SSR is cleared to 0.

[3] Repeat steps [2] and [3] until it can be confirmed that the TEND flag in SSR is set to 1.

[4] Write the transmit data to TDR, clear the TDRE flag to 0, and perform the transmit operation. The TEND flag is cleared to 0.

[5] When transmitting data continuously, go back to step [2].

[6] To end transmission, clear the TE bit to 0.

With the above processing, interrupt servicing or data transfer by the DTC is possible.

If transmission ends and the TEND flag is set to 1 while the TIE bit is set to 1 and interrupt requests are enabled, a transmit data empty interrupt (TXI) request will be generated. If an error occurs in transmission and the ERS flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a transfer error interrupt (ERI) request will be generated.

The timing for setting the TEND flag depends on the value of the GM bit in SMR. The TEND flag set timing is shown in figure 14-6.

If the DTC is activated by a TXI request, the number of bytes set in the DTC can be transmitted automatically, including automatic retransmission.

For details, see Interrupt Operation and Data Transfer Operation by DTC below.

Note: For block transfer mode, see section 13.3.2, Operation in Asynchronous Mode.

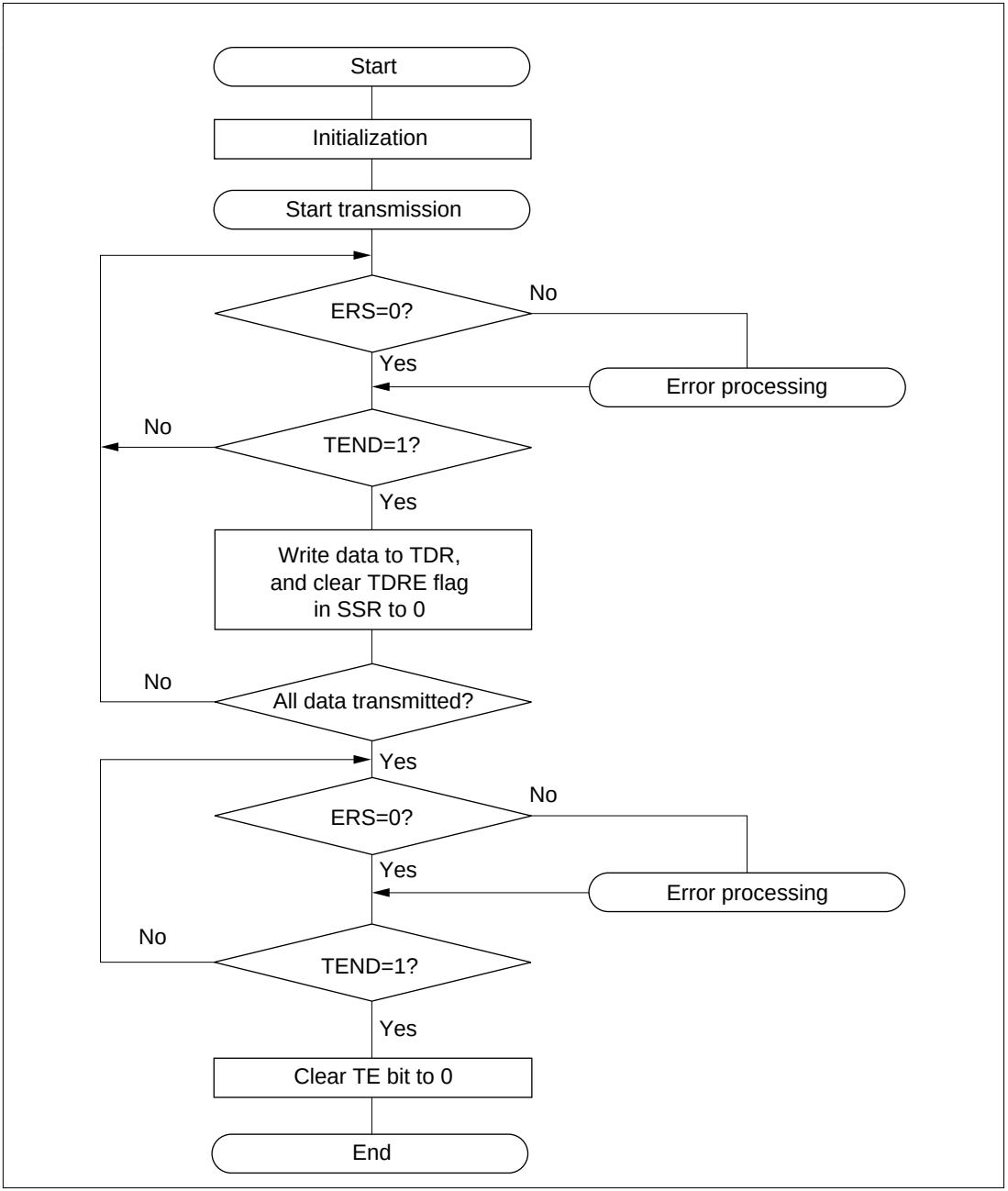
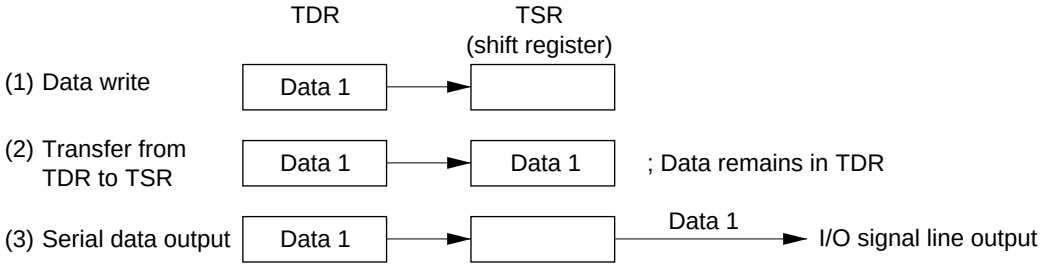


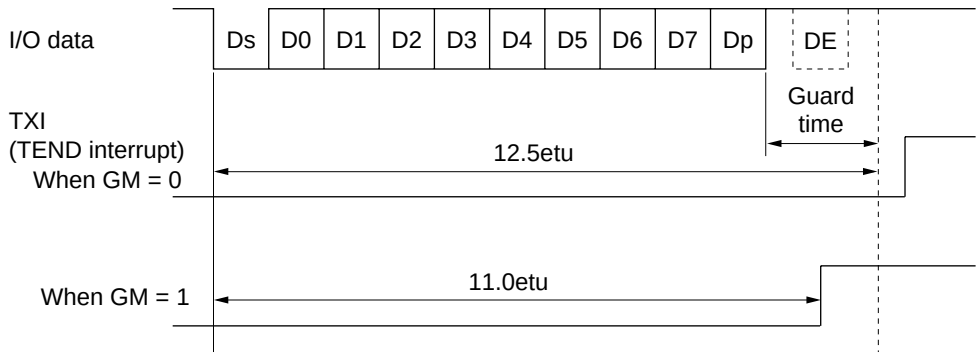
Figure 14-4 Example of Transmission Processing Flow



In case of normal transmission: TEND flag is set
In case of transmit error: ERS flag is set
Steps (2) and (3) above are repeated until the TEND flag is set

Note: When the ERS flag is set, it should be cleared until transfer of the last bit (D7 in LSB-first transmission, D0 in MSB-first transmission) of the next transfer data to be transmitted has been completed.

Figure 14-5 Relation Between Transmit Operation and Internal Registers



Legend
Ds : Start bit
D0 to D7 : Data bits
Dp : Parity bit
DE : Error signal

Figure 14-6 TEND Flag Generation Timing in Transmission Operation

Serial Data Reception (Except Block Transfer Mode): Data reception in Smart Card mode uses the same processing procedure as for the normal SCI. Figure 14-7 shows an example of the transmission processing flow.

[1] Perform Smart Card interface mode initialization as described above in Initialization.

[2] Check that the ORER flag and PER flag in SSR are cleared to 0. If either is set, perform the appropriate receive error processing, then clear both the ORER and the PER flag to 0.

[3] Repeat steps [2] and [3] until it can be confirmed that the RDRF flag is set to 1.

[4] Read the receive data from RDR.

[5] When receiving data continuously, clear the RDRF flag to 0 and go back to step [2].

[6] To end reception, clear the RE bit to 0.

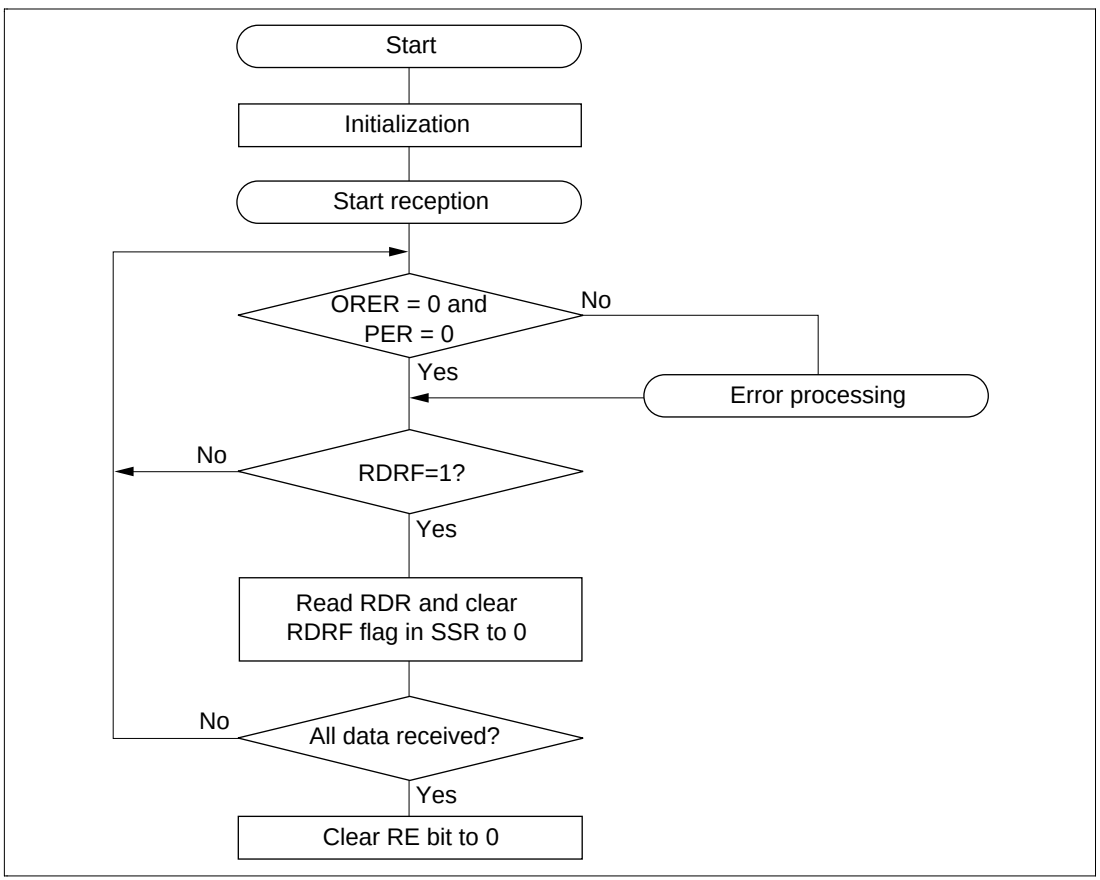


Figure 14-7 Example of Reception Processing Flow

With the above processing, interrupt servicing or data transfer by the DTC is possible.

If reception ends and the RDRF flag is set to 1 while the RIE bit is set to 1 and interrupt requests are enabled, a receive data full interrupt (RXI) request will be generated. If an error occurs in reception and either the ORER flag or the PER flag is set to 1, a transfer error interrupt (ERI) request will be generated.

If the DTC is activated by an RXI request, the receive data in which the error occurred is skipped, and only the number of bytes of receive data set in the DTC are transferred.

For details, see Interrupt Operation and Data Transfer Operation by DTC followings.

If a parity error occurs during reception and the PER is set to 1, the received data is still transferred to RDR, and therefore this data can be read.

Note: For block transfer mode, see section 13.3.2, Operation in Asynchronous Mode.

Mode Switching Operation: When switching from receive mode to transmit mode, first confirm that the receive operation has been completed, then start from initialization, clearing RE bit to 0 and setting TE bit to 1. The RDRF flag or the PER and ORER flags can be used to check that the receive operation has been completed.

When switching from transmit mode to receive mode, first confirm that the transmit operation has been completed, then start from initialization, clearing TE bit to 0 and setting RE bit to 1. The TEND flag can be used to check that the transmit operation has been completed.

Fixing Clock Output Level: When the GM bit in SMR is set to 1, the clock output level can be fixed with bits CKE1 and CKE0 in SCR. At this time, the minimum clock pulse width can be made the specified width.

Figure 14-8 shows the timing for fixing the clock output level. In this example, GSM is set to 1, CKE1 is cleared to 0, and the CKE0 bit is controlled.

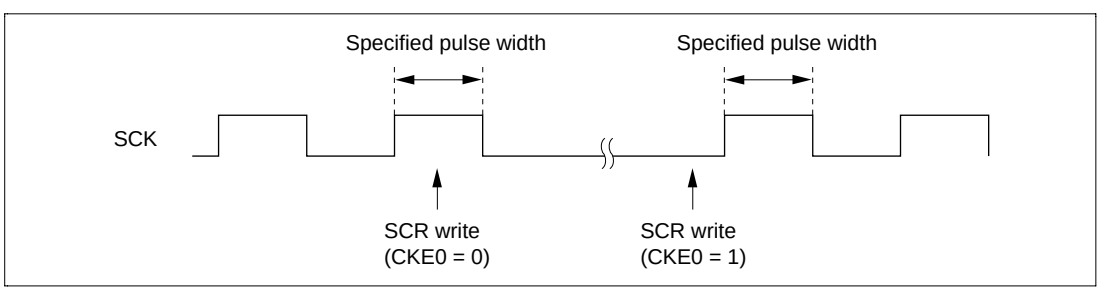


Figure 14-8 Timing for Fixing Clock Output Level

Interrupt Operation (Except Block Transfer Mode): There are three interrupt sources in smart card interface mode: transmit data empty interrupt (TXI) requests, transfer error interrupt (ERI)

requests, and receive data full interrupt (RXI) requests. The transmit end interrupt (TEI) request is not used in this mode.

When the TEND flag in SSR is set to 1, a TXI interrupt request is generated.

When the RDRF flag in SSR is set to 1, an RXI interrupt request is generated.

When any of flags ORER, PER, and ERS in SSR is set to 1, an ERI interrupt request is generated. The relationship between the operating states and interrupt sources is shown in table 14-8.

Note: For block transfer mode, see section 13.4, SCI Interrupts.

Table 14-8 Smart Card Mode Operating States and Interrupt Sources

Operating State		Flag	Enable Bit	Interrupt Source	DTC Activation
Transmit Mode	Normal operation	TEND	TIE	TXI	Possible
	Error	ERS	RIE	ERI	Not possible
Receive Mode	Normal operation	RDRF	RIE	RXI	Possible
	Error	PER, ORER	RIE	ERI	Not possible

Data Transfer Operation by DTC: In smart card mode, as with the normal SCI, transfer can be carried out using the DTC. In a transmit operation, the TDRE flag is also set to 1 at the same time as the TEND flag in SSR, and a TXI interrupt is generated. If the TXI request is designated beforehand as a DTC activation source, the DTC will be activated by the TXI request, and transfer of the transmit data will be carried out. The TDRE and TEND flags are automatically cleared to 0 when data transfer is performed by the DTC. In the event of an error, the SCI retransmits the same data automatically. During this period, TEND remains cleared to 0 and the DTC is not activated. Therefore, the SCI and DTC will automatically transmit the specified number of bytes, including retransmission in the event of an error. However, the ERS flag is not cleared automatically when an error occurs, and so the RIE bit should be set to 1 beforehand so that an ERI request will be generated in the event of an error, and the ERS flag will be cleared.

When performing transfer using the DTC, it is essential to set and enable the DTC before carrying out SCI setting. For details of the DTC setting procedures, see section 8, Data Transfer Controller (DTC).

In a receive operation, an RXI interrupt request is generated when the RDRF flag in SSR is set to 1. If the RXI request is designated beforehand as a DTC activation source, the DTC will be activated by the RXI request, and transfer of the receive data will be carried out. The RDRF flag is cleared to 0 automatically when data transfer is performed by the DTC. If an error occurs, an error

flag is set but the RDRF flag is not. Consequently, the DTC is not activated, but instead, an ERI interrupt request is sent to the CPU. Therefore, the error flag should be cleared.

Note: For block transfer mode, see section 13.4, SCI Interrupts.

14.3.7 Operation in GSM Mode

Switching the Mode: When switching between smart card interface mode and software standby mode, the following switching procedure should be followed in order to maintain the clock duty.

- When changing from smart card interface mode to software standby mode

[1] Set the data register (DR) and data direction register (DDR) corresponding to the SCK pin to the value for the fixed output state in software standby mode.

[2] Write 0 to the TE bit and RE bit in the serial control register (SCR) to halt transmit/receive operation. At the same time, set the CKE1 bit to the value for the fixed output state in software standby mode.

[3] Write 0 to the CKE0 bit in SCR to halt the clock.

[4] Wait for one serial clock period.
During this interval, clock output is fixed at the specified level, with the duty preserved.

[5] Make the transition to the software standby state.

- When returning to smart card interface mode from software standby mode

[6] Exit the software standby state.

[7] Write 1 to the CKE0 bit in SCR and output the clock. Signal generation is started with the normal duty.

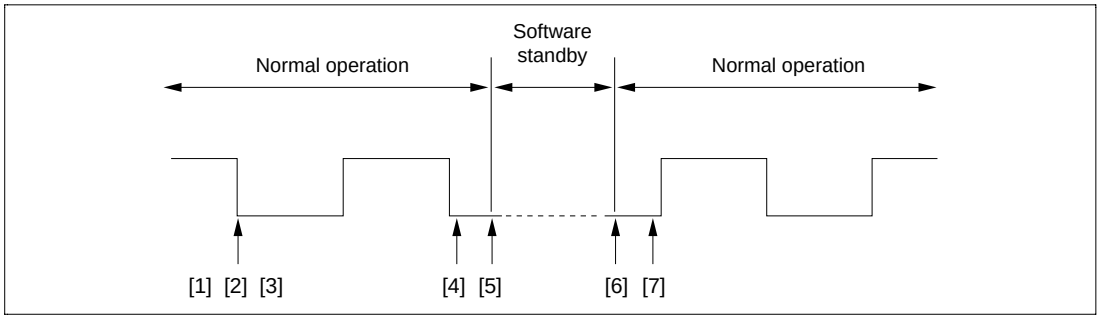


Figure 14-9 Clock Halt and Restart Procedure

Powering On: To secure the clock duty from power-on, the following switching procedure should be followed.

[1] The initial state is port input and high impedance. Use a pull-up resistor or pull-down resistor to fix the potential.

[2] Fix the SCK pin to the specified output level with the CKE1 bit in SCR.

[3] Set SMR and SCMR, and switch to smart card mode operation.

[4] Set the CKE0 bit in SCR to 1 to start clock output.

14.3.8 Operation in Block Transfer Mode

Operation in block transfer mode is the same as in SCI asynchronous mode, except for the following points. For details, see section 13.3.2, Operation in Asynchronous Mode.

Data Format: The data format is 8 bits with parity. There is no stop bit, but there is a 2-bit (1-bit or more in reception) error guard time.

Also, except during transmission (with start bit, data bits, and parity bit), the transmission pins go to the high-impedance state, so the signal lines must be fixed high with a pull-up resistor.

Transmit/Receive Clock: Only an internal clock generated by the on-chip baud rate generator can be used as the transmit/receive clock. The number of basic clock periods in a 1-bit transfer interval can be set to 32, 64, 372, or 256 with bits BCP1 and BCP0. For details, see section 14.3.5, Clock.

ERS (FER) Flag: As with the normal Smart Card interface, the ERS flag indicates the error signal status, but since error signal transmission and reception is not performed, this flag is always cleared to 0.

14.4 Usage Notes

The following points should be noted when using the SCI as a Smart Card interface.

Receive Data Sampling Timing and Reception Margin in Smart Card Interface Mode: In Smart Card interface mode, the SCI operates on a basic clock with a frequency of 32, 64, 372, or 256 times the transfer rate (as determined by bits BCP1 and BCP0).

In reception, the SCI samples the falling edge of the start bit using the basic clock, and performs internal synchronization. Receive data is latched internally at the rising edge of the 16th, 32nd, 186th, or 128th pulse of the basic clock. Figure 14-10 shows the receive data sampling timing when using a clock of 372 times the transfer rate.

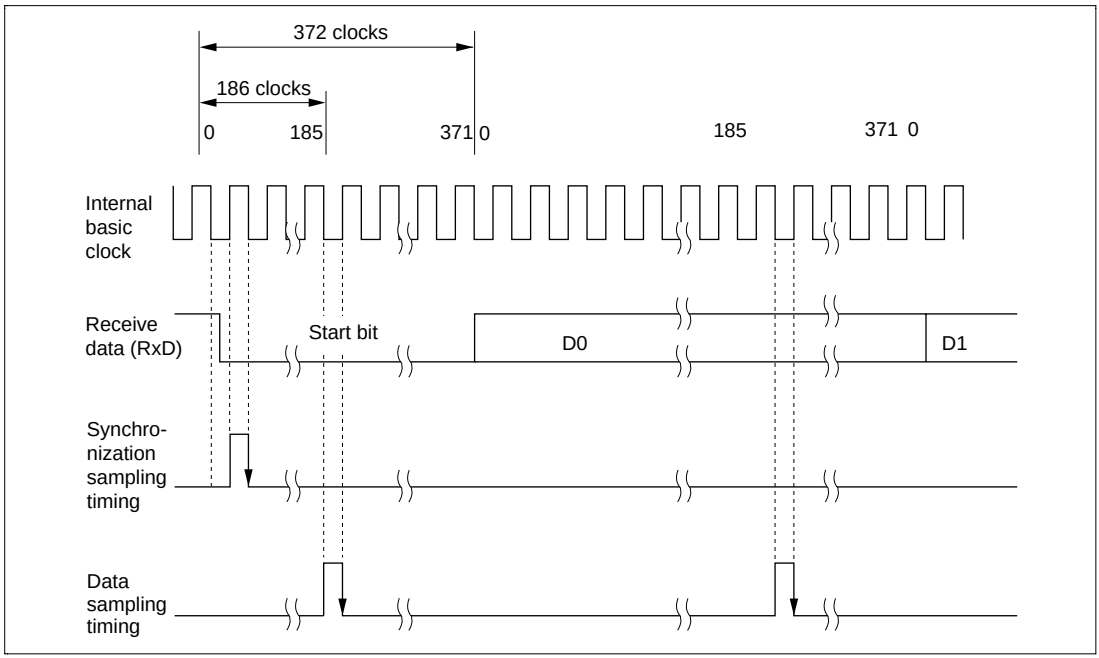


Figure 14-10 Receive Data Sampling Timing in Smart Card Mode (Using Clock of 372 Times the Transfer Rate)

Thus the reception margin in asynchronous mode is given by the following formula.

Formula for reception margin in smart card interface mode

$$M = \left| (0.5 - \frac{1}{2N}) - (L - 0.5)\,F - \frac{|D - 0.5|}{N}(1 + F) \right| \times 100\%$$

Where M: Reception margin (%)
N: Ratio of bit rate to clock (N = 32, 64, 372, and 256)
D: Clock duty (D = 0 to 1.0)
L: Frame length (L = 10)
F: Absolute value of clock frequency deviation

Assuming values of F = 0, D = 0.5 and N = 372 in the above formula, the reception margin formula is as follows.

When D = 0.5 and F = 0,
$M = (0.5 - 1/2 \times 372) \times 100\%$
$= 49.866\%$

Retransfer Operations (Except Block Transfer Mode): Retransfer operations are performed by the SCI in receive mode and transmit mode as described below.

- Retransfer operation when SCI is in receive mode
  Figure 14-11 illustrates the retransfer operation when the SCI is in receive mode.

[1] If an error is found when the received parity bit is checked, the PER bit in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The PER bit in SSR should be kept cleared to 0 until the next parity bit is sampled.

[2] The RDRF bit in SSR is not set for a frame in which an error has occurred.

[3] If no error is found when the received parity bit is checked, the PER bit in SSR is not set to 1.

[4] If no error is found when the received parity bit is checked, the receive operation is judged to have been completed normally, and the RDRF flag in SSR is automatically set to 1. If the RIE bit in SCR is enabled at this time, an RXI interrupt request is generated.
If DTC data transfer by an RXI source is enabled, the contents of RDR can be read automatically. When the RDR data is read by the DTC, the RDRF flag is automatically cleared to 0.

[5] When a normal frame is received, the pin retains the high-impedance state at the timing for error signal transmission.

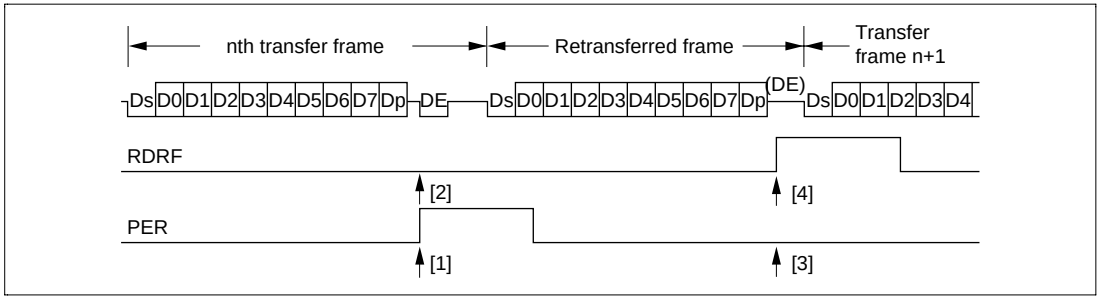


Figure 14-11 Retransfer Operation in SCI Receive Mode

- Retransfer operation when SCI is in transmit mode

  Figure 14-12 illustrates the retransfer operation when the SCI is in transmit mode.

[6] If an error signal is sent back from the receiving end after transmission of one frame is completed, the ERS bit in SSR is set to 1. If the RIE bit in SCR is enabled at this time, an ERI interrupt request is generated. The ERS bit in SSR should be kept cleared to 0 until the next parity bit is sampled.

[7] The TEND bit in SSR is not set for a frame for which an error signal indicating an abnormality is received.

[8] If an error signal is not sent back from the receiving end, the ERS bit in SSR is not set.

[9] If an error signal is not sent back from the receiving end, transmission of one frame, including a retransfer, is judged to have been completed, and the TEND bit in SSR is set to 1. If the TIE bit in SCR is enabled at this time, a TXI interrupt request is generated.

If data transfer by the DTC by means of the TXI source is enabled, the next data can be written to TDR automatically. When data is written to TDR by the DTC, the TDRE bit is automatically cleared to 0.

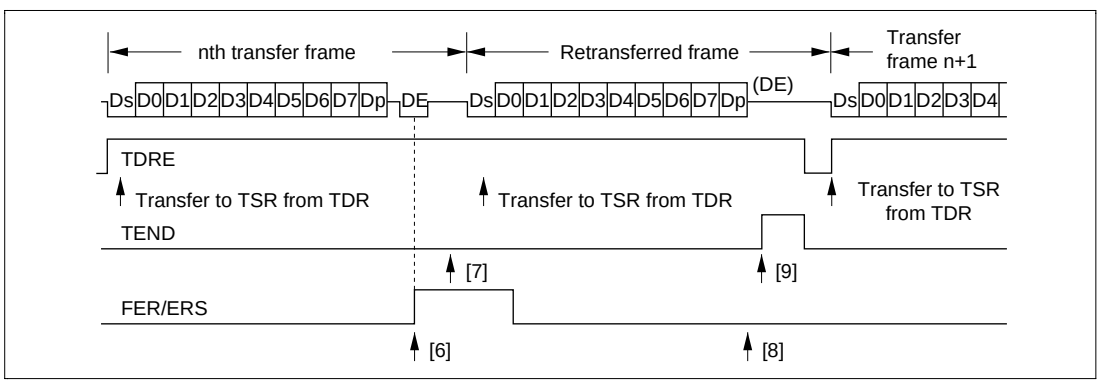


Figure 14-12 Retransfer Operation in SCI Transmit Mode

Section 15 I^2C Bus Interface [Option] (Only for the H8S/2638 and H8S/2639)

A two-channel I^2C bus interface is available as an option in the H8S/2638. Observe the following notes when using this option.

A W is added to the part number in products in which this optional function is used.

Examples: HD64F2638WF*

Note: * When the optional function is used in a U-mask version, “U” is replaced with “W”.
Example: HD64F2638UF → HD64F2638WF

15.1 Overview

A two-channel I^2C bus interface is available for the H8S/2638 and H8S/2639 as an option. The I^2C bus interface conforms to and provides a subset of the Philips I^2C bus (inter-IC bus) interface functions. The register configuration that controls the I^2C bus differs partly from the Philips configuration, however.

Each I^2C bus interface channel uses only one data line (SDA) and one clock line (SCL) to transfer data, saving board and connector space.

15.1.1 Features

- Selection of addressing format or non-addressing format
  - — I^2C bus format: addressing format with acknowledge bit, for master/slave operation
  - — Serial format: non-addressing format without acknowledge bit, for master operation only
- Conforms to Philips I^2C bus interface (I^2C bus format)
- Two ways of setting slave address (I^2C bus format)
- Start and stop conditions generated automatically in master mode (I^2C bus format)
- Selection of acknowledge output levels when receiving (I^2C bus format)
- Automatic loading of acknowledge bit when transmitting (I^2C bus format)
- Wait function in master mode (I^2C bus format)
  A wait can be inserted by driving the SCL pin low after data transfer, excluding acknowledgement. The wait can be cleared by clearing the interrupt flag.

- Wait function in slave mode (I^2C bus format)

  A wait request can be generated by driving the SCL pin low after data transfer, excluding acknowledgement. The wait request is cleared when the next transfer becomes possible.

- Three interrupt sources
  - — Data transfer end (including transmission mode transition with I^2C bus format and address reception after loss of master arbitration)
  - — Address match: when any slave address matches or the general call address is received in slave receive mode (I^2C bus format)
  - — Stop condition detection

- Selection of 16 internal clocks (in master mode)

- Direct bus drive (with SCL and SDA pins)
  - — Two pins—P35/SCL0 and P34/SDA0—(normally NMOS push-pull outputs) function as NMOS open-drain outputs when the bus drive function is selected.
  - — Two pins—P33/SCL1 and P32/SDA1—(normally CMOS pins) function as NMOS-only outputs when the bus drive function is selected.

15.1.2 Block Diagram

Figure 15-1 shows a block diagram of the I^2C bus interface.

Figure 15-2 shows an example of I/O pin connections to external circuits. Channel 0 I/O pins are NMOS open drains, and it is possible to apply voltages in excess of the power supply (V_{CC}) voltage for this LSI. Set the upper limit of voltage applied to the power supply (V_{CC}) power supply range + 0.3 V, i.e. 5.8 V. Channel 1 I/O pins are driven solely by NMOS, so in terms of appearance they carry out the same operations as an NMOS open drain. However, the voltage which can be applied to the I/O pins depends on the voltage of the power supply (V_{CC}) of this LSI.

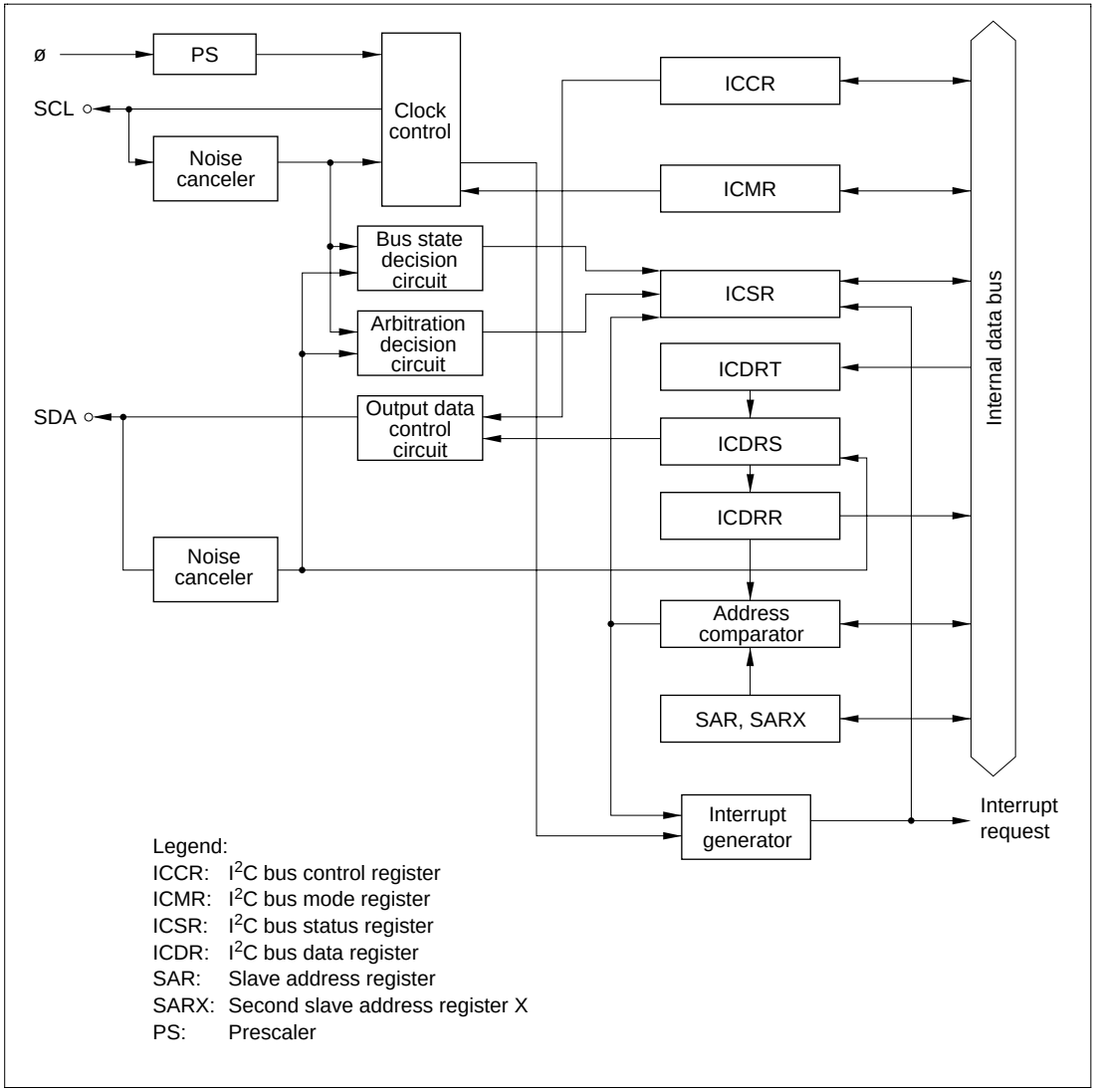


Figure 15-1 Block Diagram of I²C Bus Interface

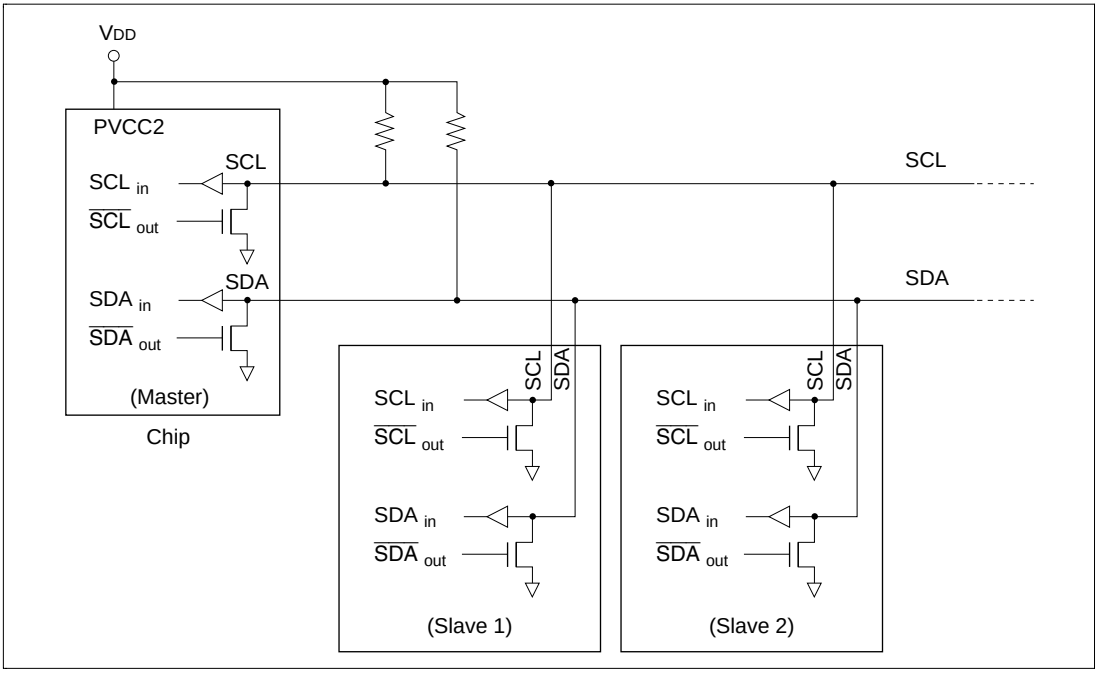


Figure 15-2 I²C Bus Interface Connections (Example: H8S/2638 and H8S/2639 Chip as Master)

15.1.3 Input/Output Pins

Table 15-1 summarizes the input/output pins used by the I²C bus interface.

Table 15-1 I²C Bus Interface Pins

Channel	Name	Abbreviation	I/O	Function
0	Serial clock	SCL0	I/O	IIC0 serial clock input/output
	Serial data	SDA0	I/O	IIC0 serial data input/output
1	Serial clock	SCL1	I/O	IIC1 serial clock input/output
	Serial data	SDA1	I/O	IIC1 serial data input/output

Note: In the text, the channel subscript is omitted, and only SCL and SDA are used.

15.1.4 Register Configuration

Table 15-2 summarizes the registers of the I²C bus interface.

Table 15-2 Register Configuration

Channel	Name	Abbreviation	R/W	Initial Value	Address*[1]
0	I²C bus control register	ICCR0	R/W	H'01	H'FF78*[3]
	I²C bus status register	ICSR0	R/W	H'00	H'FF79*[3]
	I²C bus data register	ICDR0	R/W	—	H'FF7E*[2, 3]
	I²C bus mode register	ICMR0	R/W	H'00	H'FF7F*[2, 3]
	Slave address register	SAR0	R/W	H'00	H'FF7F*[2, 3]
	Second slave address register	SARX0	R/W	H'01	H'FF7E*[2, 3]
1	I²C bus control register	ICCR1	R/W	H'01	H'FF80*[3]
	I²C bus status register	ICSR1	R/W	H'00	H'FF81*[3]
	I²C bus data register	ICDR1	R/W	—	H'FF86*[2, 3]
	I²C bus mode register	ICMR1	R/W	H'00	H'FF87*[2, 3]
	Slave address register	SAR1	R/W	H'00	H'FF87*[2, 3]
	Second slave address register	SARX1	R/W	H'01	H'FF86*[2, 3]
Common	Serial control register X	SCRX	R/W	H'00	H'FDB4
	DDC switch register	DDCSWR	R/W	H'0F	H'FDB5
	Module stop control register B	MSTPCRB	R/W	H'FF	H'FDE9

Notes: *1 Lower 16 bits of the address.

*2 The register that can be written or read depends on the ICE bit in the I²C bus control register. The slave address register can be accessed when ICE = 0, and the I²C bus mode register can be accessed when ICE = 1.

*3 The I²C bus interface registers are assigned to the same addresses as other registers. Register selection is performed by means of the IICE bit in the serial control register X (SCRX).

15.2 Register Descriptions

15.2.1 I²C Bus Data Register (ICDR)

Bit	:	7	6	5	4	3	2	1	0
		ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

- ICDRR

Bit	:	7	6	5	4	3	2	1	0
		ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	R	R	R	R	R	R	R	R

- ICDRS

Bit	:	7	6	5	4	3	2	1	0
		ICDRS7	ICDRS6	ICDRR5	ICDRS4	ICDRS3	ICDRS2	ICDRS1	ICDRS0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	—	—	—	—	—	—	—	—

- ICDRT

Bit	:	7	6	5	4	3	2	1	0
		ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	W	W	W	W	W	W	W	W

- TDRE, RDRF (internal flags)

Bit	:							—	—
								TDRE	RDRF
Initial value	:							0	0
R/W	:							—	—

ICDR is an 8-bit readable/writable register that is used as a transmit data register when transmitting and a receive data register when receiving. ICDR is divided internally into a shift register (ICDRS), receive buffer (ICDRR), and transmit buffer (ICDRT). ICDRS cannot be read or written by the CPU, ICDRR is read-only, and ICDRT is write-only. Data transfers among the three registers are performed automatically in coordination with changes in the bus state, and affect the status of internal flags such as TDRE and RDRF.

If IIC is in transmit mode and the next data is in ICDRT (the TDRE flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRT to ICDRS. If IIC is in receive mode and no previous data remains in ICDRR (the RDRF flag is 0) following transmission/reception of one frame of data using ICDRS, data is transferred automatically from ICDRS to ICDRR.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

ICDR is assigned to the same address as SARX, and can be written and read only when the ICE bit is set to 1 in ICCR.

The value of ICDR is undefined after a reset.

The TDRE and RDRF flags are set and cleared under the conditions shown below. Setting the TDRE and RDRF flags affects the status of the interrupt flags.

TDRE	Description
0	The next transmit data is in ICDR (ICDRT), or transmission cannot be started (Initial value) [Clearing conditions] • When transmit data is written in ICDR (ICDRT) in transmit mode (TRS = 1) • When a stop condition is detected in the bus line state after a stop condition is issued with the I^2C bus format or serial format selected • When a stop condition is detected with the I^2C bus format selected • In receive mode (TRS = 0) (A 0 write to TRS during transfer is valid after reception of a frame containing an acknowledge bit)
1	The next transmit data can be written in ICDR (ICDRT) [Setting conditions] • In transmit mode (TRS = 1), when a start condition is detected in the bus line state after a start condition is issued in master mode with the I^2C bus format or serial format selected • When using formatless mode in transmit mode (TRS = 1) • When data is transferred from ICDRT to ICDRS (Data transfer from ICDRT to ICDRS when TRS = 1 and TDRE = 0, and ICDRS is empty) • When a switch is made from receive mode (TRS = 0) to transmit mode (TRS = 1) after detection of a start condition

RDRF	Description
0	The data in ICDR (ICDRR) is invalid (Initial value) [Clearing condition] When ICDR (ICDRR) receive data is read in receive mode
1	The ICDR (ICDRR) receive data can be read [Setting condition] When data is transferred from ICDRS to ICDRR (Data transfer from ICDRS to ICDRR in case of normal termination with TRS = 0 and RDRF = 0)

15.2.2 Slave Address Register (SAR)

Bit	:	7	6	5	4	3	2	1	0
		SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SAR is an 8-bit readable/writable register that stores the slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SAR match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SAR is assigned to the same address as ICMR, and can be written and read only when the ICE bit is cleared to 0 in ICCR.

SAR is initialized to H'00 by a reset and in hardware standby mode.

Bits 7 to 1—Slave Address (SVA6 to SVA0): Set a unique address in bits SVA6 to SVA0, differing from the addresses of other slave devices connected to the I^2C bus.

Bit 0—Format Select (FS): Used together with the FSX bit in SARX and the SW bit in DDCSWR to select the communication format.

- I^2C bus format: addressing format with acknowledge bit
- Synchronous serial format: non-addressing format without acknowledge bit, for master mode only

The FS bit also specifies whether or not SAR slave address recognition is performed in slave mode.

DDCSWR Bit 6	SAR Bit 0	SARX Bit 0	
SW	**FS**	**FSX**	**Operating Mode**
0	0	0	I^2C bus format • SAR and SARX slave addresses recognized
		1	I^2C bus format (Initial value) • SAR slave address recognized • SARX slave address ignored
	1	0	I^2C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	—	—	Must not be set.

15.2.3 Second Slave Address Register (SARX)

Bit	:	7	6	5	4	3	2	1	0
		SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

SARX is an 8-bit readable/writable register that stores the second slave address and selects the communication format. When the chip is in slave mode (and the addressing format is selected), if the upper 7 bits of SARX match the upper 7 bits of the first frame received after a start condition, the chip operates as the slave device specified by the master device. SARX is assigned to the same address as ICDR, and can be written and read only when the ICE bit is cleared to 0 in ICCR.

SARX is initialized to H'01 by a reset and in hardware standby mode.

Bits 7 to 1—Second Slave Address (SVAX6 to SVAX0): Set a unique address in bits SVAX6 to SVAX0, differing from the addresses of other slave devices connected to the I^2C bus.

Bit 0—Format Select X (FSX): Used together with the FS bit in SAR and the SW bit in DDCSWR to select the communication format.

- I^2C bus format: addressing format with acknowledge bit
- Synchronous serial format: non-addressing format without acknowledge bit, for master mode only

The FSX bit also specifies whether or not SARX slave address recognition is performed in slave mode. For details, see the description of the FS bit in SAR.

15.2.4 I^2C Bus Mode Register (ICMR)

Bit	:	7	6	5	4	3	2	1	0
		MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ICMR is an 8-bit readable/writable register that selects whether the MSB or LSB is transferred first, performs master mode wait control, and selects the master mode transfer clock frequency and the transfer bit count. ICMR is assigned to the same address as SAR. ICMR can be written and read only when the ICE bit is set to 1 in ICCR.

ICMR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—MSB-First/LSB-First Select (MLS): Selects whether data is transferred MSB-first or LSB-first.

If the number of bits in a frame, excluding the acknowledge bit, is less than 8, transmit data and receive data are stored differently. Transmit data should be written justified toward the MSB side when MLS = 0, and toward the LSB side when MLS = 1. Receive data bits read from the LSB side should be treated as valid when MLS = 0, and bits read from the MSB side when MLS = 1.

Do not set this bit to 1 when the I^2C bus format is used.

Bit 7

MLS	Description	
0	MSB-first	(Initial value)
1	LSB-first	

Bit 6—Wait Insertion Bit (WAIT): Selects whether to insert a wait between the transfer of data and the acknowledge bit, in master mode with the I^2C bus format. When WAIT is set to 1, after the fall of the clock for the final data bit, the IRIC flag is set to 1 in ICCR, and a wait state begins (with SCL at the low level). When the IRIC flag is cleared to 0 in ICCR, the wait ends and the acknowledge bit is transferred. If WAIT is cleared to 0, data and acknowledge bits are transferred consecutively with no wait inserted.

The IRIC flag in ICCR is set to 1 on completion of the acknowledge bit transfer, regardless of the WAIT setting.

The setting of this bit is invalid in slave mode.

Bit 6

WAIT	Description	
0	Data and acknowledge bits transferred consecutively	(Initial value)
1	Wait inserted between data and acknowledge bits	

Bits 5 to 3—Serial Clock Select (CKS2 to CKS0): These bits, together with the IICX1 (channel 1) or IICX0 (channel 0) bit in the SCRX register, select the serial clock frequency in master mode. They should be set according to the required transfer rate.

SCRX Bit 5 or 6	Bit 5	Bit 4	Bit 3		Transfer Rate				
IICX	CKS2	CKS1	CKS0	Clock	ø = 5 MHz	ø = 8 MHz	ø = 10 MHz	ø = 16 MHz	ø = 20 MHz
0	0	0	0	ø/28	179 kHz	286 kHz	357 kHz	571 kHz*	714 kHz*
			1	ø/40	125 kHz	200 kHz	250 kHz	400 kHz	500 kHz*
		1	0	ø/48	104 kHz	167 kHz	208 kHz	333 kHz	417 kHz*
			1	ø/64	78.1 kHz	125 kHz	156 kHz	250 kHz	313 kHz
	1	0	0	ø/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
			1	ø/100	50.0 kHz	80.0 kHz	100 kHz	160 kHz	200 kHz
		1	0	ø/112	44.6 kHz	71.4 kHz	89.3 kHz	143 kHz	179 kHz
			1	ø/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	0	0	0	ø/56	89.3 kHz	143 kHz	179 kHz	286 kHz	357 kHz
			1	ø/80	62.5 kHz	100 kHz	125 kHz	200 kHz	250 kHz
		1	0	ø/96	52.1 kHz	83.3 kHz	104 kHz	167 kHz	208 kHz
			1	ø/128	39.1 kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
	1	0	0	ø/160	31.3 kHz	50.0 kHz	62.5 kHz	100 kHz	125 kHz
			1	ø/200	25.0 kHz	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz
		1	0	ø/224	22.3 kHz	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz
			1	ø/256	19.5 kHz	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz

Note: * These rates are outside the ranges stipulated in the I^2C bus interface specifications (normal mode: max. 100 kHz, high-speed mode: max. 400 kHz).

Bits 2 to 0—Bit Counter (BC2 to BC0): Bits BC2 to BC0 specify the number of bits to be transferred next. With the I²C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the data is transferred with one addition acknowledge bit. Bit BC2 to BC0 settings should be made during an interval between transfer frames. If bits BC2 to BC0 are set to a value other than 000, the setting should be made while the SCL line is low..

The bit counter is initialized to 000 by a reset and when a start condition is detected. The value returns to 000 at the end of a data transfer, including the acknowledge bit.

Bit 2	Bit 1	Bit 0	Bits/Frame		
BC2	**BC1**	**BC0**	**Synchronous Serial Format**	**I²C Bus Format**	
0	0	0	8	9	(Initial value)
		1	1	2	
	1	0	2	3	
		1	3	4	
1	0	0	4	5	
		1	5	6	
	1	0	6	7	
		1	7	8	

15.2.5 I²C Bus Control Register (ICCR)

Bit	:	7	6	5	4	3	2	1	0
		ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	W

Note: * Only 0 can be written, for flag clearing.

ICCR is an 8-bit readable/writable register that enables or disables the I²C bus interface, enables or disables interrupts, selects master or slave mode and transmission or reception, enables or disables acknowledgement, confirms the I²C bus interface bus status, issues start/stop conditions, and performs interrupt flag confirmation.

ICCR is initialized to H'01 by a reset and in hardware standby mode.

Bit 7—I²C Bus Interface Enable (ICE): Selects whether or not the I²C bus interface is to be used. When ICE is set to 1, port pins function as SCL and SDA input/output pins and transfer operations are enabled. When ICE is cleared to 0, the I²C bus interface module is halted and its internal states are cleared.

The SAR and SARX registers can be accessed when ICE is 0. The ICMR and ICDR registers can be accessed when ICE is 1.

Bit 7

ICE	Description	
0	I²C bus interface module disabled, with SCL and SDA signal pins set to port function	(Initial value)
	I²C bus interface module internal states initialized	
	SAR and SARX can be accessed	
1	I²C bus interface module enabled for transfer operations (pins SCL and SCA are driving the bus)	
	ICMR and ICDR can be accessed	

Bit 6—I²C Bus Interface Interrupt Enable (IEIC): Enables or disables interrupts from the I²C bus interface to the CPU.

Bit 6

IEIC	Description	
0	Interrupts disabled	(Initial value)
1	Interrupts enabled	

Bit 5—Master/Slave Select (MST)

Bit 4—Transmit/Receive Select (TRS)

MST selects whether the I²C bus interface operates in master mode or slave mode.

TRS selects whether the I²C bus interface operates in transmit mode or receive mode.

In master mode with the I²C bus format, when arbitration is lost, MST and TRS are both reset by hardware, causing a transition to slave receive mode. In slave receive mode with the addressing format (FS = 0 or FSX = 0), hardware automatically selects transmit or receive mode according to the R/W bit in the first frame after a start condition.

Modification of the TRS bit during transfer is deferred until transfer of the frame containing the acknowledge bit is completed, and the changeover is made after completion of the transfer.

MST and TRS select the operating mode as follows.

Bit 5	Bit 4		
MST	**TRS**	**Operating Mode**	
0	0	Slave receive mode	(Initial value)
	1	Slave transmit mode	
1	0	Master receive mode	
	1	Master transmit mode	

Bit 5

MST	Description	
0	Slave mode	(Initial value)
	[Clearing conditions] 1. When 0 is written by software 2. When bus arbitration is lost after transmission is started in I^2C bus format master mode	
1	Master mode [Setting conditions] 1. When 1 is written by software (in cases other than clearing condition 2) 2. When 1 is written in MST after reading MST = 0 (in case of clearing condition 2)	

Bit 4

TRS	Description	
0	Receive mode	(Initial value)
	[Clearing conditions] 1. When 0 is written by software (in cases other than setting condition 3) 2. When 0 is written in TRS after reading TRS = 1 (in case of clearing condition 3) 3. When bus arbitration is lost after transmission is started in I^2C bus format master mode 4. When the SW bit in DDCSWR changes from 1 to 0	
1	Transmit mode [Setting conditions] 1. When 1 is written by software (in cases other than clearing conditions 3 and 4) 2. When 1 is written in TRS after reading TRS = 0 (in case of clearing conditions 3 and 4) 3. When a 1 is received as the R/W bit of the first frame in I^2C bus format slave mode	

Bit 3—Acknowledge Bit Judgement Selection (ACKE): Specifies whether the value of the acknowledge bit returned from the receiving device when using the I^2C bus format is to be ignored and continuous transfer is performed, or transfer is to be aborted and error handling, etc., performed if the acknowledge bit is 1. When the ACKE bit is 0, the value of the received acknowledge bit is not indicated by the ACKB bit, which is always 0.

In the H8S/2633 Series, the DTC can be used to perform continuous transfer. The DTC is activated when the IRTR interrupt flag is set to 1 (IRTR is one of two interrupt flags, the other being IRIC). When the ACKE bit is 0, the TDRE, IRIC, and IRTR flags are set on completion of data transmission, regardless of the value of the acknowledge bit. When the ACKE bit is 1, the TDRE, IRIC, and IRTR flags are set on completion of data transmission when the acknowledge bit is 0, and the IRIC flag alone is set on completion of data transmission when the acknowledge bit is 1.

When the DTC is activated, the TDRE, IRIC, and IRTR flags are cleared to 0 after the specified number of data transfers have been executed. Consequently, interrupts are not generated during continuous data transfer, but if data transmission is completed with a 1 acknowledge bit when the ACKE bit is set to 1, the DTC is not activated and an interrupt is generated, if enabled.

Depending on the receiving device, the acknowledge bit may be significant, in indicating completion of processing of the received data, for instance, or may be fixed at 1 and have no significance.

Bit 3

ACKE	Description	
0	The value of the acknowledge bit is ignored, and continuous transfer is performed	(Initial value)
1	If the acknowledge bit is 1, continuous transfer is interrupted	

Bit 2—Bus Busy (BBSY): The BBSY flag can be read to check whether the I^2C bus (SCL, SDA) is busy or free. In master mode, this bit is also used to issue start and stop conditions.

A high-to-low transition of SDA while SCL is high is recognized as a start condition, setting BBSY to 1. A low-to-high transition of SDA while SCL is high is recognized as a stop condition, clearing BBSY to 0.

To issue a start condition, use a MOV instruction to write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, use a MOV instruction to write 0 in BBSY and 0 in SCP. It is not possible to write to BBSY in slave mode; the I^2C bus interface must be set to master transmit mode before issuing a start condition. MST and TRS should both be set to 1 before writing 1 in BBSY and 0 in SCP.

Bit 2		
BBSY	**Description**	
0	Bus is free	(Initial value)
	[Clearing condition]	
	When a stop condition is detected	
1	Bus is busy	
	[Setting condition]	
	When a start condition is detected	

Bit 1—I²C Bus Interface Interrupt Request Flag (IRIC): Indicates that the I²C bus interface has issued an interrupt request to the CPU. IRIC is set to 1 at the end of a data transfer, when a slave address or general call address is detected in slave receive mode, when bus arbitration is lost in master transmit mode, and when a stop condition is detected. IRIC is set at different times depending on the FS bit in SAR and the WAIT bit in ICMR. See section 15.3.6, IRIC Setting Timing and SCL Control. The conditions under which IRIC is set also differ depending on the setting of the ACKE bit in ICCR.

IRIC is cleared by reading IRIC after it has been set to 1, then writing 0 in IRIC.

When the DTC is used, IRIC is cleared automatically and transfer can be performed continuously without CPU intervention.

Bit 1

IRIC	Description
0	Waiting for transfer, or transfer in progress (Initial value) [Clearing conditions] 1. When 0 is written in IRIC after reading IRIC = 1 2. When ICDR is written or read by the DTC (When the TDRE or RDRF flag is cleared to 0) (This is not always a clearing condition; see the description of DTC operation for details)
1	Interrupt requested [Setting conditions] • I^2C bus format master mode 1. When a start condition is detected in the bus line state after a start condition is issued (when the TDRE flag is set to 1 because of first frame transmission) 2. When a wait is inserted between the data and acknowledge bit when WAIT = 1 3. At the end of data transfer (at the rise of the 9th transmit/receive clock pulse, or at the fall of the 8th transmit/receive clock pulse when using wait insertion) 4. When a slave address is received after bus arbitration is lost (when the AL flag is set to 1) 5. When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) • I^2C bus format slave mode 1. When the slave address (SVA, SVAX) matches (when the AAS and AASX flags are set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) 2. When the general call address is detected (when FS = 0 and the ADZ flag is set to 1) and at the end of data transfer up to the subsequent retransmission start condition or stop condition detection (when the TDRE or RDRF flag is set to 1) 3. When 1 is received as the acknowledge bit when the ACKE bit is 1 (when the ACKB bit is set to 1) 4. When a stop condition is detected (when the STOP or ESTP flag is set to 1) • Synchronous serial format 1. At the end of data transfer (when the TDRE or RDRF flag is set to 1) 2. When a start condition is detected with serial format selected When any other condition arises in which the TDRE or RDRF flag is set to 1

When, with the I^2C bus format selected, IRIC is set to 1 and an interrupt is generated, other flags must be checked in order to identify the source that set IRIC to 1. Although each source has a corresponding flag, caution is needed at the end of a transfer.

When the TDRE or RDRF internal flag is set, the readable IRTR flag may or may not be set. The IRTR flag (the DTC start request flag) is not set at the end of a data transfer up to detection of a retransmission start condition or stop condition after a slave address (SVA) or general call address match in I^2C bus format slave mode.

Even when the IRIC flag and IRTR flag are set, the TDRE or RDRF internal flag may not be set. The IRIC and IRTR flags are not cleared at the end of the specified number of transfers in continuous transfer using the DTC. The TDRE or RDRF flag is cleared, however, since the specified number of ICDR reads or writes have been completed.

Table 15-3 shows the relationship between the flags and the transfer states.

Table 15-3 Flags and Transfer States

MST	TRS	BBSY	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB	State
1/0	1/0	0	0	0	0	0	0	0	0	0	Idle state (flag clearing required)
1	1	0	0	0	0	0	0	0	0	0	Start condition issuance
1	1	1	0	0	1	0	0	0	0	0	Start condition established
1	1/0	1	0	0	0	0	0	0	0	0/1	Master mode wait
1	1/0	1	0	0	1	0	0	0	0	0/1	Master mode transmit/receive end
0	0	1	0	0	0	1/0	1	1/0	1/0	0	Arbitration lost
0	0	1	0	0	0	0	0	1	0	0	SAR match by first frame in slave mode
0	0	1	0	0	0	0	0	1	1	0	General call address match
0	0	1	0	0	0	1	0	0	0	0	SARX match
0	1/0	1	0	0	0	0	0	0	0	0/1	Slave mode transmit/receive end (except after SARX match)
0	1/0	1	0	0	1	1	0	0	0	0	Slave mode transmit/receive end (after SARX match)
0	1	1	0	0	0	1	0	0	0	1	
0	1/0	0	1/0	1/0	0	0	0	0	0	0/1	Stop condition detected

Bit 0—Start Condition/Stop Condition Prohibit (SCP): Controls the issuing of start and stop conditions in master mode. To issue a start condition, write 1 in BBSY and 0 in SCP. A retransmit start condition is issued in the same way. To issue a stop condition, write 0 in BBSY and 0 in SCP. This bit is always read as 1. If 1 is written, the data is not stored.

Bit 0		
SCP	**Description**	
0	Writing 0 issues a start or stop condition, in combination with the BBSY flag	
1	Reading always returns a value of 1	(Initial value)
	Writing is ignored	

15.2.6 I²C Bus Status Register (ICSR)

Bit	:	7	6	5	4	3	2	1	0
		ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

Note: * Only 0 can be written, for flag clearing.

ICSR is an 8-bit readable/writable register that performs flag confirmation and acknowledge confirmation and control.

ICSR is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Error Stop Condition Detection Flag (ESTP): Indicates that a stop condition has been detected during frame transfer in I²C bus format slave mode.

Bit 7

ESTP	Description	
0	No error stop condition	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in ESTP after reading ESTP = 1	
	2. When the IRIC flag is cleared to 0	
1	• In I^2C bus format slave mode Error stop condition detected [Setting condition] When a stop condition is detected during frame transfer	
	• In other modes No meaning	

Bit 6—Normal Stop Condition Detection Flag (STOP): Indicates that a stop condition has been detected after completion of frame transfer in I^2C bus format slave mode.

Bit 6

STOP	Description	
0	No normal stop condition	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in STOP after reading STOP = 1	
	2. When the IRIC flag is cleared to 0	
1	• In I^2C bus format slave mode Normal stop condition detected [Setting condition] When a stop condition is detected after completion of frame transfer	
	• In other modes No meaning	

Bit 5—I^2C Bus Interface Continuous Transmission/Reception Interrupt Request Flag (IRTR): Indicates that the I^2C bus interface has issued an interrupt request to the CPU, and the source is completion of reception/transmission of one frame in continuous transmission/reception for which DTC activation is possible. When the IRTR flag is set to 1, the IRIC flag is also set to 1 at the same time.

IRTR flag setting is performed when the TDRE or RDRF flag is set to 1. IRTR is cleared by reading IRTR after it has been set to 1, then writing 0 in IRTR. IRTR is also cleared automatically when the IRIC flag is cleared to 0.

Bit 5

IRTR	Description	
0	Waiting for transfer, or transfer in progress	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in IRTR after reading IRTR = 1	
	2. When the IRIC flag is cleared to 0	
1	Continuous transfer state	
	[Setting condition]	
	• In I^2C bus interface slave mode When the TDRE or RDRF flag is set to 1 when AASX = 1	
	• In other modes When the TDRE or RDRF flag is set to 1	

Bit 4—Second Slave Address Recognition Flag (AASX): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVAX6 to SVAX0 in SARX.

AASX is cleared by reading AASX after it has been set to 1, then writing 0 in AASX. AASX is also cleared automatically when a start condition is detected.

Bit 4

AASX	Description	
0	Second slave address not recognized	(Initial value)
	[Clearing conditions]	
	1. When 0 is written in AASX after reading AASX = 1	
	2. When a start condition is detected	
	3. In master mode	
1	Second slave address recognized	
	[Setting condition]	
	When the second slave address is detected in slave receive mode and FSX = 0	

Bit 3—Arbitration Lost (AL): This flag indicates that arbitration was lost in master mode. The I^2C bus interface monitors the bus. When two or more master devices attempt to seize the bus at nearly the same time, if the I^2C bus interface detects data differing from the data it sent, it sets AL to 1 to indicate that the bus has been taken by another master.

AL is cleared by reading AL after it has been set to 1, then writing 0 in AL. In addition, AL is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 3

AL	Description
0	Bus arbitration won (Initial value) [Clearing conditions] 1. When ICDR data is written (transmit mode) or read (receive mode) 2. When 0 is written in AL after reading AL = 1
1	Arbitration lost [Setting conditions] 1. If the internal SDA and SDA pin disagree at the rise of SCL in master transmit mode 2. If the internal SCL line is high at the fall of SCL in master transmit mode

Bit 2—Slave Address Recognition Flag (AAS): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition matches bits SVA6 to SVA0 in SAR, or if the general call address (H'00) is detected.

AAS is cleared by reading AAS after it has been set to 1, then writing 0 in AAS. In addition, AAS is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 2

AAS	Description
0	Slave address or general call address not recognized (Initial value) [Clearing conditions] 1. When ICDR data is written (transmit mode) or read (receive mode) 2. When 0 is written in AAS after reading AAS = 1 3. In master mode
1	Slave address or general call address recognized [Setting condition] When the slave address or general call address is detected in slave receive mode and FS = 0

Bit 1—General Call Address Recognition Flag (ADZ): In I^2C bus format slave receive mode, this flag is set to 1 if the first frame following a start condition is the general call address (H'00).

ADZ is cleared by reading ADZ after it has been set to 1, then writing 0 in ADZ. In addition, ADZ is reset automatically by write access to ICDR in transmit mode, or read access to ICDR in receive mode.

Bit 1

ADZ	Description	
0	General call address not recognized	(Initial value)
	[Clearing conditions]	
	1. When ICDR data is written (transmit mode) or read (receive mode)	
	2. When 0 is written in ADZ after reading ADZ = 1	
	3. In master mode	
1	General call address recognized	
	[Setting condition]	
	When the general call address is detected in slave receive mode and (FSX = 0 or FS = 0)	

Bit 0—Acknowledge Bit (ACKB): Stores acknowledge data. In transmit mode, after the receiving device receives data, it returns acknowledge data, and this data is loaded into ACKB. In receive mode, after data has been received, the acknowledge data set in this bit is sent to the transmitting device.

When this bit is read, in transmission (when TRS = 1), the value loaded from the bus line (returned by the receiving device) is read. In reception (when TRS = 0), the value set by internal software is read.

Bit 0

ACKB	Description	
0	Receive mode: 0 is output at acknowledge output timing	(Initial value)
	Transmit mode: Indicates that the receiving device has acknowledged the data (signal is 0)	
1	Receive mode: 1 is output at acknowledge output timing	
	Transmit mode: Indicates that the receiving device has not acknowledged the data (signal is 1)	

15.2.7 Serial Control Register X (SCRX)

Bit	:	7	6	5	4	3	2	1	0
		—	IICX1	IICX0	IICE	—	—	—	—
Initial value	:	0	0	0	0	1	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R	R/W	R/W	R/W

SCRX is an 8-bit readable/writable register that controls register access, the I^{2}C interface operating mode (when the on-chip IIC option is included), and on-chip flash memory control (F-ZTAT versions). If a module controlled by SCRX is not used, do not write 1 to the corresponding bit.

SCRX is initialized to H'00 by a reset and in hardware standby mode.

Bit 7—Reserved: Do not set 1.

Bit 6—I^{2}C Transfer Select 1 (IICX1): This bit, together with bits CKS2 to CKS0 in ICMR of IIC1, selects the transfer rate in master mode. For details, see section 15.2.4, I^{2}C Bus Mode Register (ICMR).

Bit 5—I^{2}C Transfer Select 0 (IICX0): This bit, together with bits CKS2 to CKS0 in ICMR of IIC0, selects the transfer rate in master mode. For details, see section 15.2.4, I^{2}C Bus Mode Register (ICMR).

Bit 4—I^{2}C Master Enable (IICE): Controls CPU access to the I^{2}C bus interface data and control registers (ICCR, ICSR, ICDR/SARX, ICMR/SAR).

Bit 4

IICE	Description	
0	CPU access to I^{2}C bus interface data and control registers is disabled	(Initial value)
1	CPU access to I^{2}C bus interface data and control registers is enabled	

Bit 3— Reserved: Always returns a value of 1 if it is read.

Bits 2 to 0—Reserved: Do not set 1.

15.2.8 DDC Switch Register (DDCSWR)

Bit	:	7	6	5	4	3	2	1	0
		—	—	—	—	CLR3	CLR2	CLR1	CLR0
Initial value	:	0	0	0	0	1	1	1	1
R/W	:	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	W*2	W*2	W*2	W*2

Notes: *1 Should always be written with 0.
*2 Always read as 1.

DDCSWR is an 8-bit readable/writable register that is used to initialize the IIC module.

DDCSWR is initialized to H'0F by a reset and in hardware standby mode.

Bits 7 to 4—Reserved: Should always be written with 0.

Bits 3 to 0—IIC Clear 3 to 0 (CLR3 to CLR0): These bits control initialization of the internal state of IIC0 and IIC1.

These bits can only be written to; if read they will always return a value of 1.

When a write operation is performed on these bits, a clear signal is generated for the internal latch circuit of the corresponding module(s), and the internal state of the IIC module(s) is initialized.

The write data for these bits is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR.

When clearing is required again, all the bits must be written to in accordance with the setting.

Bit 3	Bit 2	Bit 1	Bit 0	
CLR3	**CLR2**	**CLR1**	**CLR0**	**Description**
0	0	—	—	Setting prohibited
	1	0	0	Setting prohibited
			1	IIC0 internal latch cleared
		1	0	IIC1 internal latch cleared
			1	IIC0 and IIC1 internal latches cleared
1	—	—	—	Invalid setting

15.2.9 Module Stop Control Register B (MSTPCRB)

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRB is an 8-bit readable/writable register that perform module stop mode control.

When the MSTPB4 or MSTPB3 bit is set to 1, operation of the corresponding IIC channel is halted at the end of the bus cycle, and a transition is made to module stop mode. For details, see section 23A.5, 23B.5, Module Stop Mode.

MSTPCRB is initialized to H'FF by a power-on reset and in hardware standby mode. It is not initialized by a manual reset and in software standby mode.

Bit 4—Module Stop (MSTPB4): Specifies IIC channel 0 module stop mode.

Bit 4

MSTPB4	Description	
0	IIC channel 0 module stop mode is cleared	
1	IIC channel 0 module stop mode is set	(Initial value)

Bit 3—Module Stop (MSTPB3): Specifies IIC channel 1 module stop mode.

Bit 3

MSTPB3	Description	
0	IIC channel 1 module stop mode is cleared	
1	IIC channel 1 module stop mode is set	(Initial value)

15.3 Operation

15.3.1 I²C Bus Data Format

The I²C bus interface has serial and I²C bus formats.

The I²C bus formats are addressing formats with an acknowledge bit. These are shown in figures 15-3 (a) and (b). The first frame following a start condition always consists of 8 bits.

The serial format is a non-addressing format with no acknowledge bit. Although start and stop conditions must be issued, this format can be used as a synchronous serial format. This is shown in figure 15-4.

Figure 15-5 shows the I²C bus timing.

The symbols used in figures 15-3 to 15-5 are explained in table 15-4.

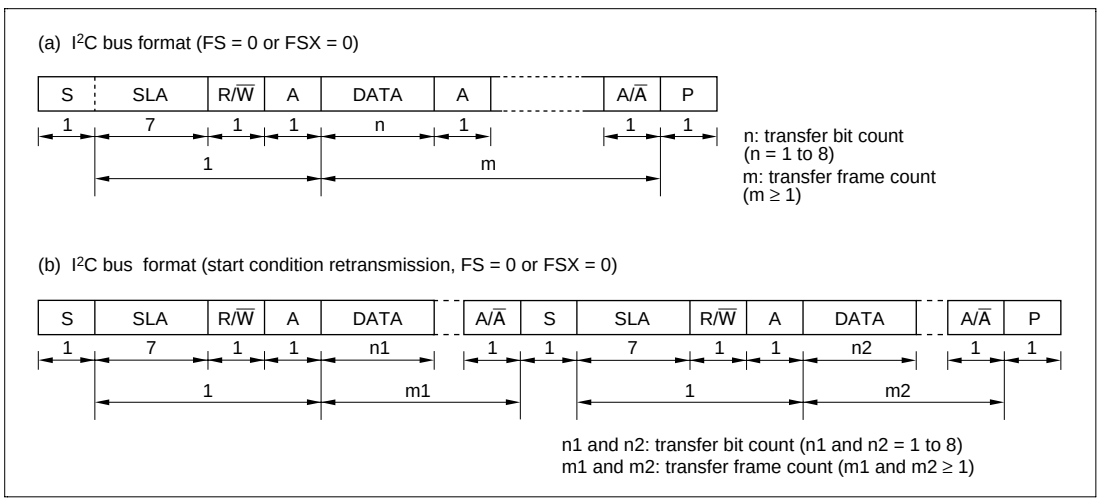


Figure 15-3 I²C Bus Data Formats (I²C Bus Formats)

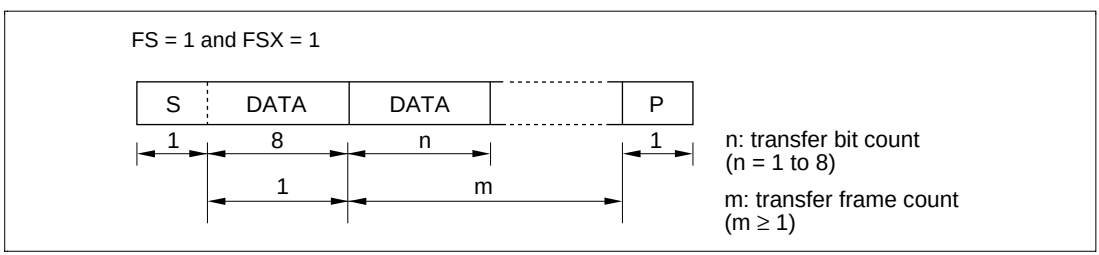


Figure 15-4 I²C Bus Data Format (Serial Format)

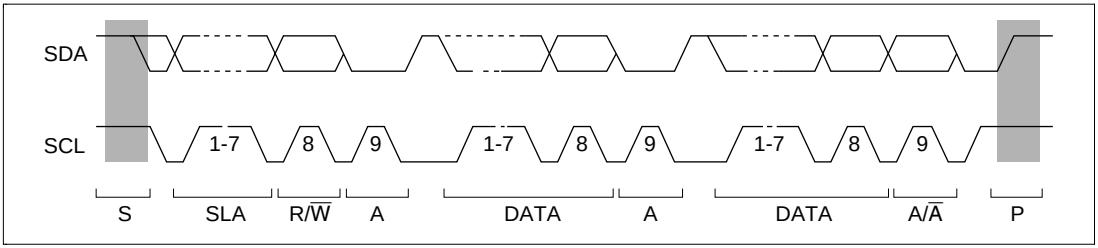


Figure 15-5 I²C Bus Timing

Table 15-4 I²C Bus Data Format Symbols

Legend	
S	Start condition. The master device drives SDA from high to low while SCL is high
SLA	Slave address, by which the master device selects a slave device
R/$\overline{W}$	Indicates the direction of data transfer: from the slave device to the master device when R/$\overline{W}$ is 1, or from the master device to the slave device when R/$\overline{W}$ is 0
A	Acknowledge. The receiving device (the slave in master transmit mode, or the master in master receive mode) drives SDA low to acknowledge a transfer
DATA	Transferred data. The bit length is set by bits BC2 to BC0 in ICMR. The MSB-first or LSB-first format is selected by bit MLS in ICMR
P	Stop condition. The master device drives SDA from low to high while SCL is high

15.3.2 Master Transmit Operation

In I²C bus format master transmit mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal.

The procedure and operations for transmitting data sequentially, synchronized with ICDR write operations, are described below.

(1) Set the ICE bit in ICCR to 1. Set bits MLS, WAIT, and CKS2 to CKS0 in ICMR, and bit IICX in STCR, according to the operating mode.
(2) Read the BBSY flag in ICCR to confirm that the bus is free.
(3) Set bits MST and TRS to 1 in ICCR to select master transmit mode.
(4) Write 1 to BBSY and 0 to SCP in ICCR. This changes SDA from high to low when SCL is high, and generates the start condition.
(5) The IRIC and IRTR flags are set to 1 when the start condition is generated. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.
(6) Write the data (slave address + R/W) to ICDR.
With the I²C bus format (when the FS bit in SAR or the FSX bit in SARX is 0), the first frame

data following the start condition indicates the 7-bit slave address and transmit/receive direction.

Next, clear the IRIC flag to 0 to indicate the end of the transfer.

Writing to ICDR and clearing the IRIC flag to 0 should be performed with no gap in between to prevent the processing of other interrupts from intervening.

If the transmission time required for 1 byte elapses before the IRIC flag is cleared, it will not be not possible to determine the end of the transfer.

The master device sequentially sends the transmit clock and the data written to ICDR using the timing shown in figure 15-6. The selected slave device (i.e., the slave device with the matching slave address) drives SDA low at the 9th transmit clock pulse and returns an acknowledge signal.

(7) When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse.

After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.

(8) Read the ACKB bit in ICSR to confirm that its value is 0.

If the value of ACKB is 1, indicating that the slave device has not sent an acknowledge signal, perform the end of transfer processing described in step (12) and then perform the transmission operation once again.

(9) Write the transmit data to ICDR.

Next, clear the IRIC flag to 0 to indicate the end of the transfer.

As in step (6), writing to ICDR and clearing the IRIC flag to 0 should be performed with no gap in between.

Transmission of the next frame is performed in synchronization with the internal clock.

(10) When one frame of data has been transmitted, the IRIC flag is set to 1 at the rise of the 9th transmit clock pulse.

After one frame has been transmitted, SCL is automatically fixed low in synchronization with the internal clock until the next transmit data is written.

(11) Read the ACKB bit in ICSR.

Confirm that value of ACKB is 0, indicating that the slave device has sent an acknowledge signal. To continue to send data, return to step (9) and continue from there. If the value of ACKB is 1, indicating that the slave device has not sent an acknowledge signal, perform the end of transfer processing described in step (12).

(12) Clear the IRIC flag to 0.

Write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

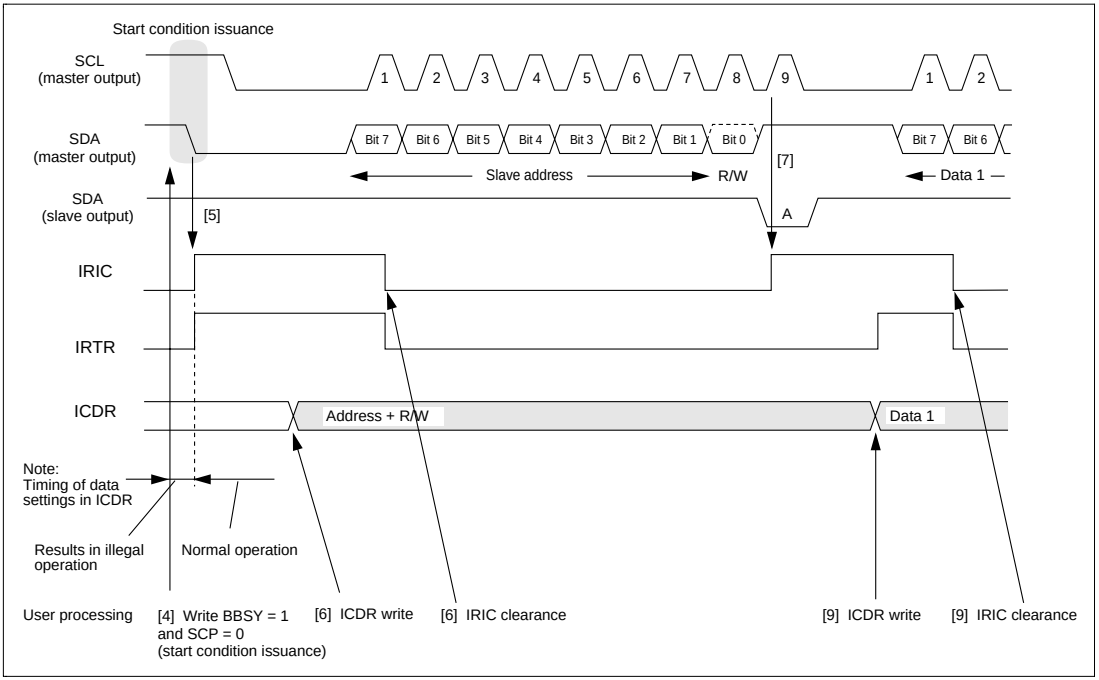


Figure 15-6 Example of Master Transmit Mode Operation Timing (MLS = WAIT = 0)

15.3.3 Master Receive Operation

In I^2C bus format master receive mode, the master device outputs the receive clock, receives data, and returns an acknowledge signal. The slave device transmits data.

The procedure and operations for receiving data sequentially, synchronized with ICDR read operations, are described below.

(1) Clear the TRS bit in ICCR to 0 to switch from transmit mode to receive mode.
Set the WAIT bit in ICMR to 1.
Clear the ACKB bit in ICSR to 0 (acknowledge data setting).

(2) When ICDR is read (dummy data read), reception is started; the receive clock is output, and data received, in synchronization with the internal clock.
Clear the IRIC flag to 0 to determine the next wait.
Reading from ICDR and clearing the IRIC flag to 0 should be performed with no gap in between to prevent the processing of other interrupts from intervening.
If the transmission time required for 1 byte elapses before the IRIC flag is cleared, it will not be possible to determine the end of the transfer.

(3) The IRIC flag is set to 1 at the rise of the 8th receive clock pulse of the frame.
If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU.
SCL is automatically fixed low in synchronization with the internal clock until the IRIC flag is cleared.
If the single frame is the last of the data to be transmitted, perform the end of transfer processing described in step (10).

(4) Clear the IRIC flag to cancel the wait.
The master device drives SDA low at the 9th receive clock pulse to return an acknowledge signal.

(5) When one frame of data has been received, the IRIC flag and IRTR flag are set to 1 at the rise of the 9th receive clock pulse.
The master device then outputs the next receive data and receive clock.

(6) Read the receive data from ICDR.

(7) Clear the IRIC flag to 0 to determine the next wait. Since the wait is cancelled in step (4) (or step (9)), the duration between the clearing of the IRIC flag and the completion of the data receive operation described in steps (5) through (7) must not exceed the transmission time required for 1 byte.

(8) The IRIC flag is set to 1 at the rise of the 8th receive clock pulse of the frame.
SCL is automatically fixed low in synchronization with the internal clock before the IRIC flag is cleared.
If the frame is the last of the data to be transmitted, perform the end of transfer processing described in step (10).

(9) Clear the IRIC flag to cancel the wait.
The master device drives SDA low at the 9th receive clock pulse to return an acknowledge signal.

Data can be received continuously by repeating steps (5) through (9).

(10) Set the ACKB bit in ICSR to 1 to specify the acknowledge signal for the final reception. Set the TRS bit in ICCR to 1 to switch from the receive to the transmit mode.

(11) Clear the IRIC flag to 0 to cancel the wait.

(12) When one frame of data has been received, the IRIC flag is set to 1 at the rise of the 9th receive clock pulse.

(13) Clear WAIT bit in ICMR to 0 to cancel the wait mode is canceled.
Next, read the receive data from ICDR and Clear the IRIC flag to 0.
Clear the IRIC flag when WAIT is equal to 0.
(If the WAIT bit is cleared to 0 after the IRIC flag is cleared, the SDA line will be fixed low and it will not be possible to issue a stop condition when a stop condition generate instruction is executed.)

(14) Write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition.

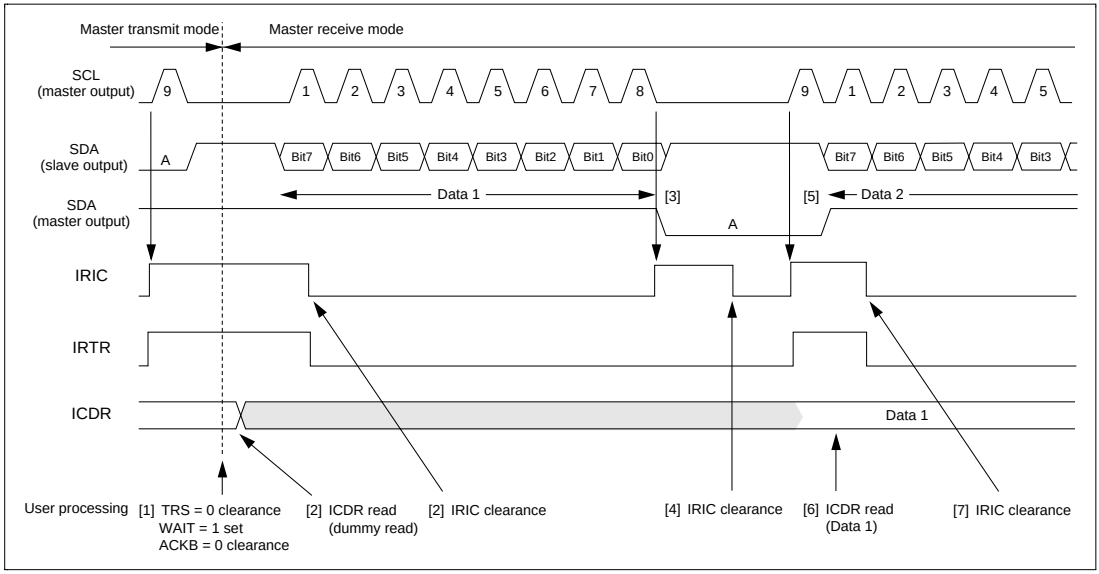


Figure 15-7 (a) Example of Master Receive Mode Operation Timing (MLS = ACKB = 0, WAIT = 1)

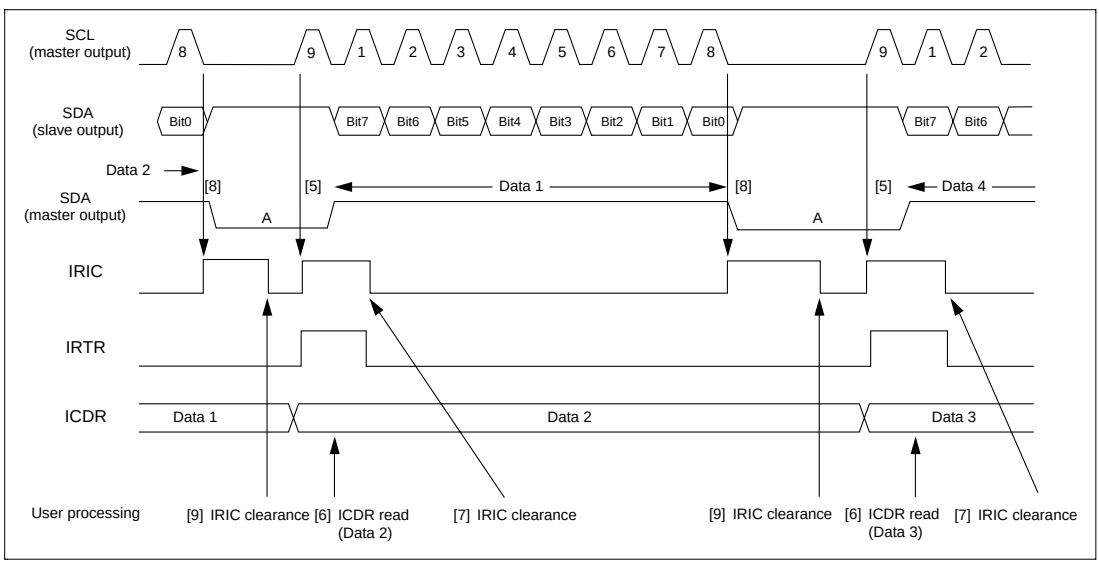


Figure 15-7 (b) Example of Master Receive Mode Operation Timing (MLS = ACKB = 0, WAIT = 1)

15.3.4 Slave Receive Operation

In slave receive mode, the master device outputs the transmit clock and transmit data, and the slave device returns an acknowledge signal. The reception procedure and operations in slave receive mode are described below.

(1) Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

(2) When the start condition output by the master device is detected, the BBSY flag in ICCR is set to 1.

(3) When the slave address matches in the first frame following the start condition, the device operates as the slave device specified by the master device. If the 8th data bit (R/$\overline{W}$) is 0, the TRS bit in ICCR remains cleared to 0, and slave receive operation is performed.

(4) At the 9th clock pulse of the receive frame, the slave device drives SDA low and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. If the RDRF internal flag has been cleared to 0, it is set to 1, and the receive operation continues. If the RDRF internal flag has been set to 1, the slave device drives SCL low from the fall of the receive clock until data is read into ICDR.

(5) Read ICDR and clear the IRIC flag in ICCR to 0. The RDRF flag is cleared to 0.

Receive operations can be performed continuously by repeating steps (4) and (5). When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

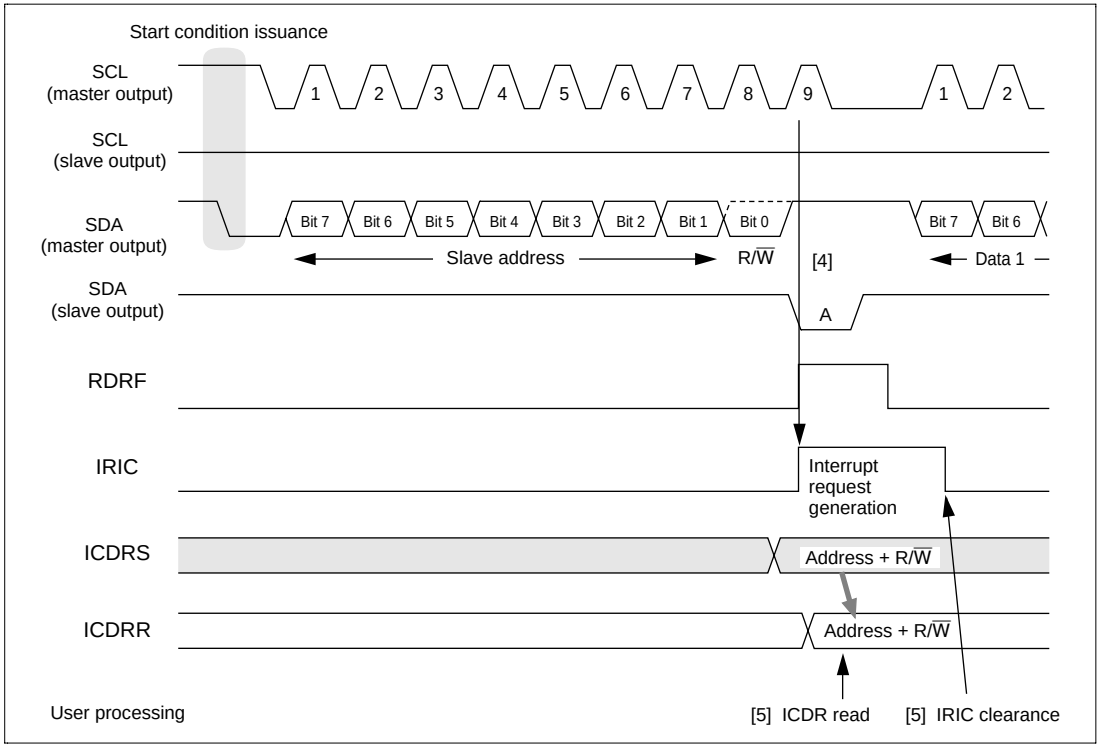


Figure 15-8 Example of Slave Receive Mode Operation Timing (1)
(MLS = ACKB = 0)

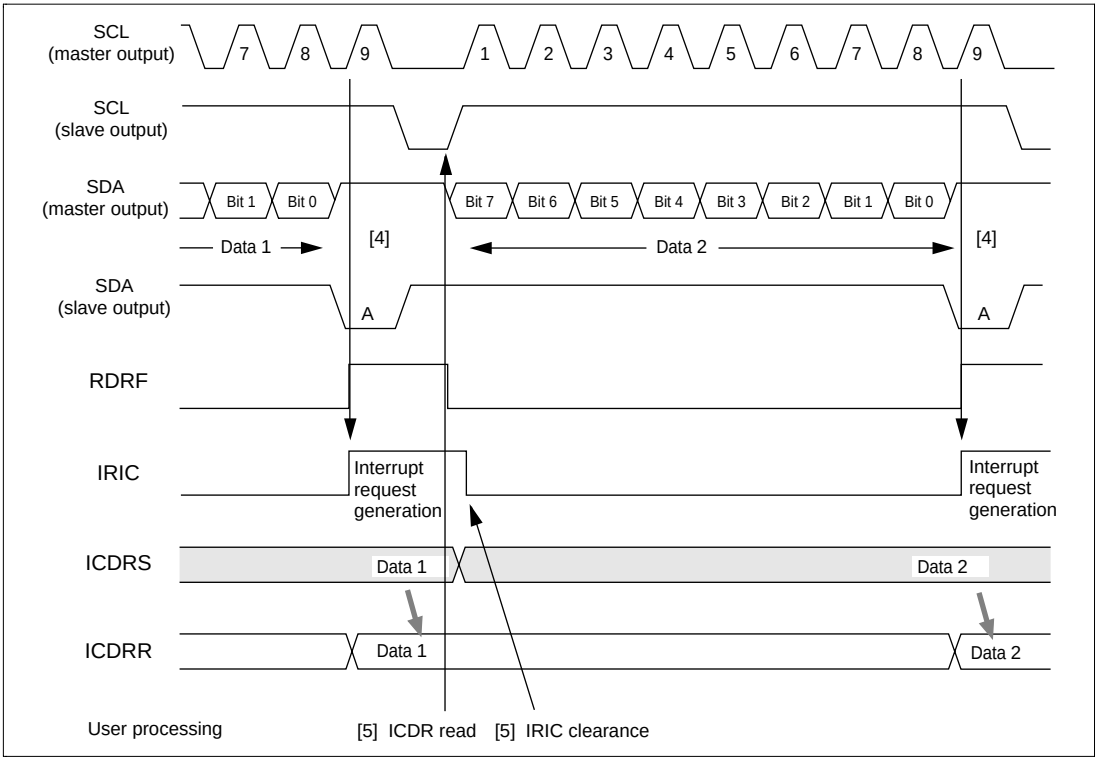


Figure 15-9 Example of Slave Receive Mode Operation Timing (2)
(MLS = ACKB = 0)

15.3.5 Slave Transmit Operation

In slave transmit mode, the slave device outputs the transmit data, while the master device outputs the receive clock and returns an acknowledge signal. The transmission procedure and operations in slave transmit mode are described below.

(1) Set the ICE bit in ICCR to 1. Set the MLS bit in ICMR and the MST and TRS bits in ICCR according to the operating mode.

(2) When the slave address matches in the first frame following detection of the start condition, the slave device drives SDA low at the 9th clock pulse and returns an acknowledge signal. At the same time, the IRIC flag in ICCR is set to 1. If the IEIC bit in ICCR has been set to 1, an interrupt request is sent to the CPU. .If the 8th data bit (R/$\overline{W}$) is 1, the TRS bit in ICCR is set to 1, and the mode changes to slave transmit mode automatically. The TDRF flag is set to 1. The slave device drives SCL low from the fall of the transmit clock until ICDR data is written.

(3) After clearing the IRIC flag to 0, write data to ICDR. The TDRE internal flag is cleared to 0. The written data is transferred to ICDRS, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again. After clearing the IRIC flag to 0, write the next data to ICDR. The

slave device sequentially sends the data written into ICDR in accordance with the clock output by the master device at the timing shown in figure 15-10.

(4) When one frame of data has been transmitted, the IRIC flag in ICCR is set to 1 at the rise of the 9th transmit clock pulse. If the TDRE internal flag has been set to 1, this slave device drives SCL low from the fall of the transmit clock until data is written to ICDR. The master device drives SDA low at the 9th clock pulse, and returns an acknowledge signal. As this acknowledge signal is stored in the ACKB bit in ICSR, this bit can be used to determine whether the transfer operation was performed normally. When the TDRE internal flag is 0, the data written into ICDR is transferred to ICDRS, transmission is started, and the TDRE internal flag and the IRIC and IRTR flags are set to 1 again.

(5) To continue transmission, clear the IRIC flag to 0, then write the next data to be transmitted into ICDR. The TDRE internal flag is cleared to 0.

Transmit operations can be performed continuously by repeating steps (4) and (5). To end transmission, write H'FF to ICDR to release SDA on the slave side. When SDA is changed from low to high when SCL is high, and the stop condition is detected, the BBSY flag in ICCR is cleared to 0.

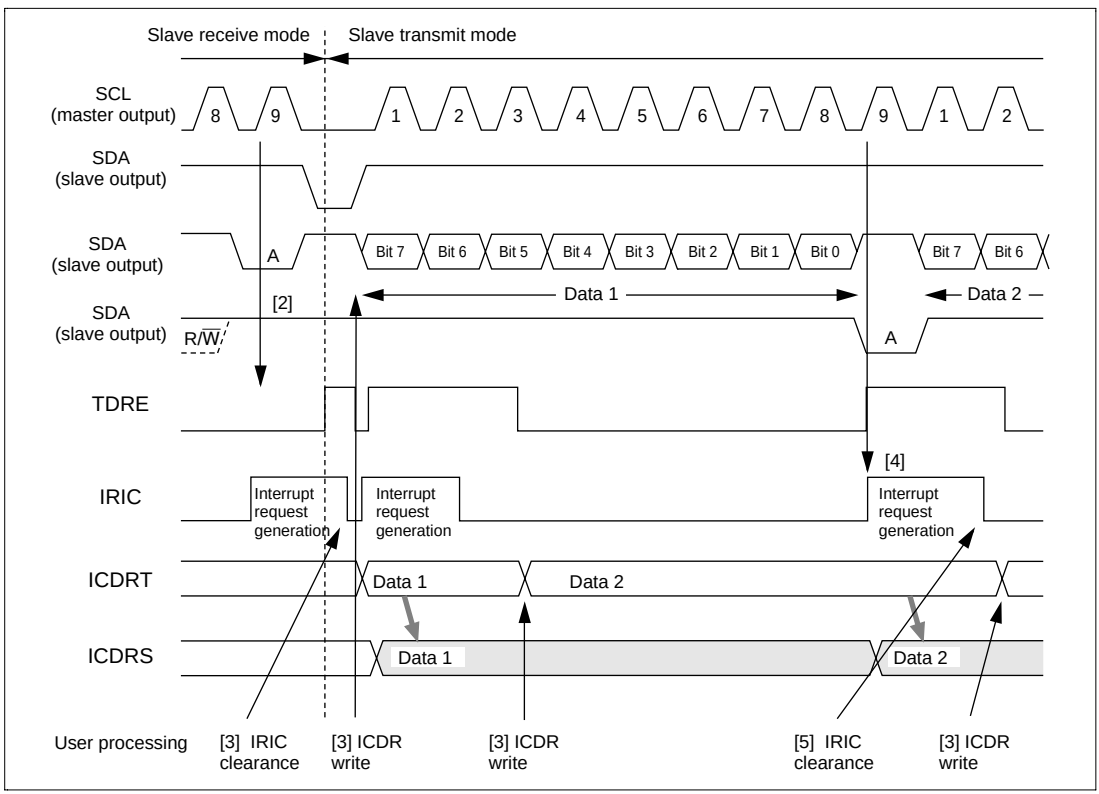


Figure 15-10 Example of Slave Transmit Mode Operation Timing (MLS = 0)

15.3.6 IRIC Setting Timing and SCL Control

The interrupt request flag (IRIC) is set at different times depending on the WAIT bit in ICMR, the FS bit in SAR, and the FSX bit in SARX. If the TDRE or RDRF internal flag is set to 1, SCL is automatically held low after one frame has been transferred; this timing is synchronized with the internal clock. Figure 15-11 shows the IRIC set timing and SCL control.

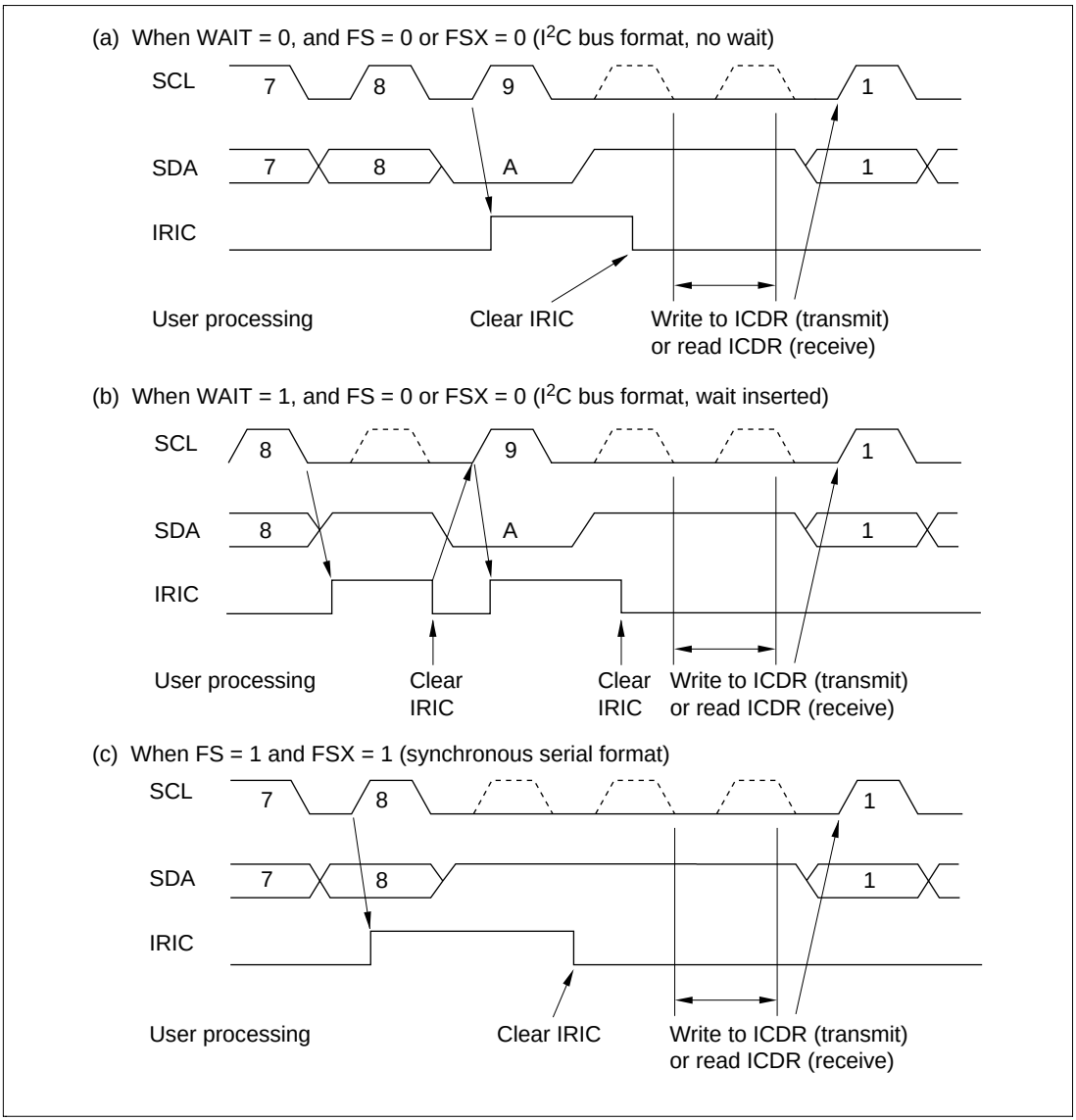


Figure 15-11 IRIC Setting Timing and SCL Control

15.3.7 Operation Using the DTC

The I^2C bus format provides for selection of the slave device and transfer direction by means of the slave address and the R/$\overline{W}$ bit, confirmation of reception with the acknowledge bit, indication of the last frame, and so on. Therefore, continuous data transfer using the DTC must be carried out in conjunction with CPU processing by means of interrupts.

Table 15-5 shows some examples of processing using the DTC. These examples assume that the number of transfer data bytes is known in slave mode.

Table 15-5 Examples of Operation Using the DTC

Item	Master Transmit Mode	Master Receive Mode	Slave Transmit Mode	Slave Receive Mode
Slave address + R/$\overline{W}$ bit transmission/ reception	Transmission by DTC (ICDR write)	Transmission by CPU (ICDR write)	Reception by CPU (ICDR read)	Reception by CPU (ICDR read)
Dummy data read	—	Processing by CPU (ICDR read)	—	—
Actual data transmission/ reception	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)	Transmission by DTC (ICDR write)	Reception by DTC (ICDR read)
Dummy data (H'FF) write	—	—	Processing by DTC (ICDR write)	—
Last frame processing	Not necessary	Reception by CPU (ICDR read)	Not necessary	Reception by CPU (ICDR read)
Transfer request processing after last frame processing	1st time: Clearing by CPU 2nd time: End condition issuance by CPU	Not necessary	Automatic clearing on detection of end condition during transmission of dummy data (H'FF)	Not necessary
Setting of number of DTC transfer data frames	Transmission: Actual data count + 1 (+1 equivalent to slave address + R/$\overline{W}$ bits)	Reception: Actual data count	Transmission: Actual data count + 1 (+1 equivalent to dummy data (H'FF))	Reception: Actual data count

15.3.8 Noise Canceler

The logic levels at the SCL and SDA pins are routed through noise cancelers before being latched internally. Figure 15-12 shows a block diagram of the noise canceler circuit.

The noise canceler consists of two cascaded latches and a match detector. The SCL (or SDA) input signal is sampled on the system clock, but is not passed forward to the next circuit unless the outputs of both latches agree. If they do not agree, the previous value is held.

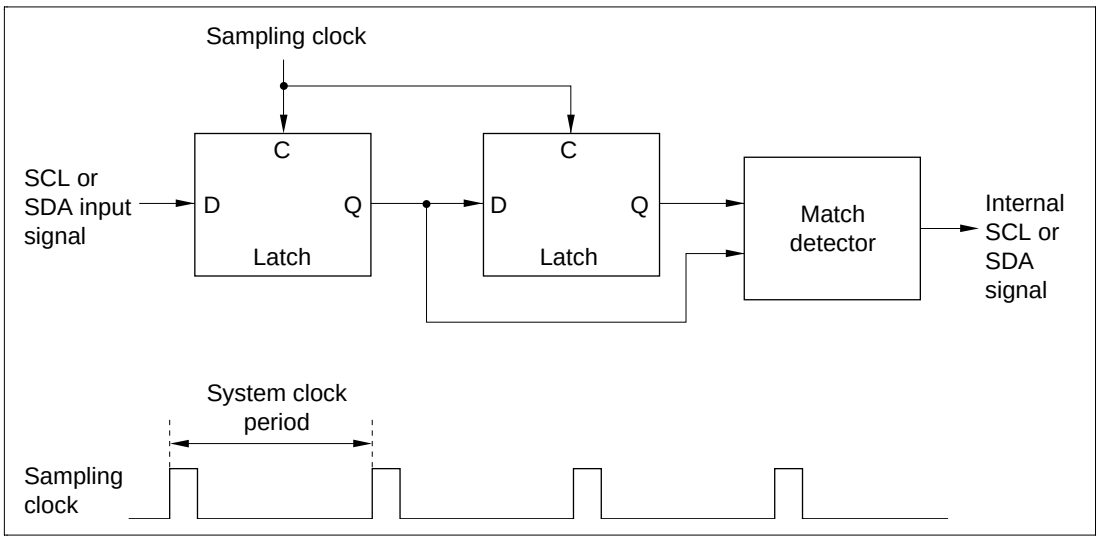


Figure 15-12 Block Diagram of Noise Canceler

15.3.9 Sample Flowcharts

Figures 15-13 to 15-16 show sample flowcharts for using the I^2C bus interface in each mode.

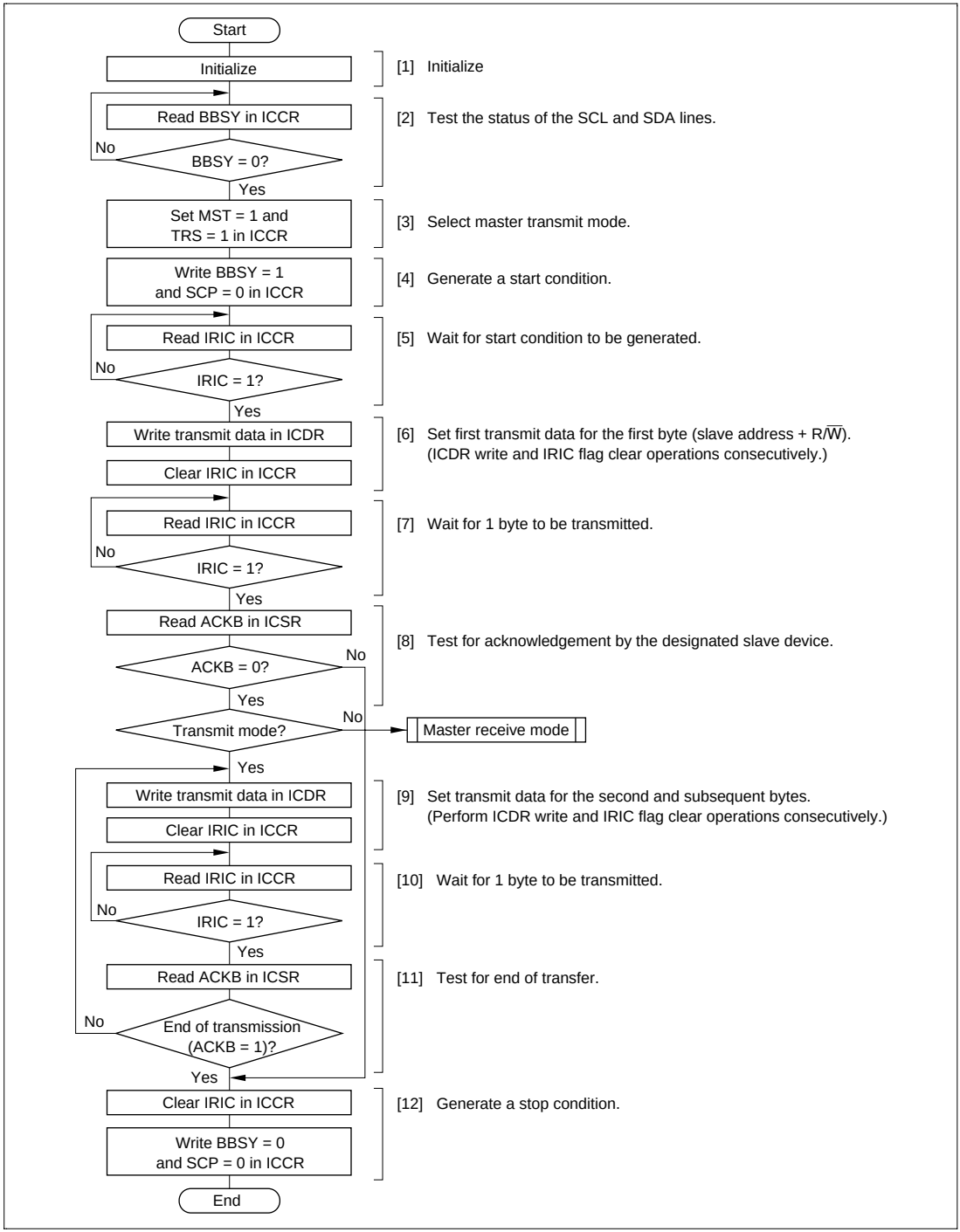


Figure 15-13 Flowchart for Master Transmit Mode (Example)

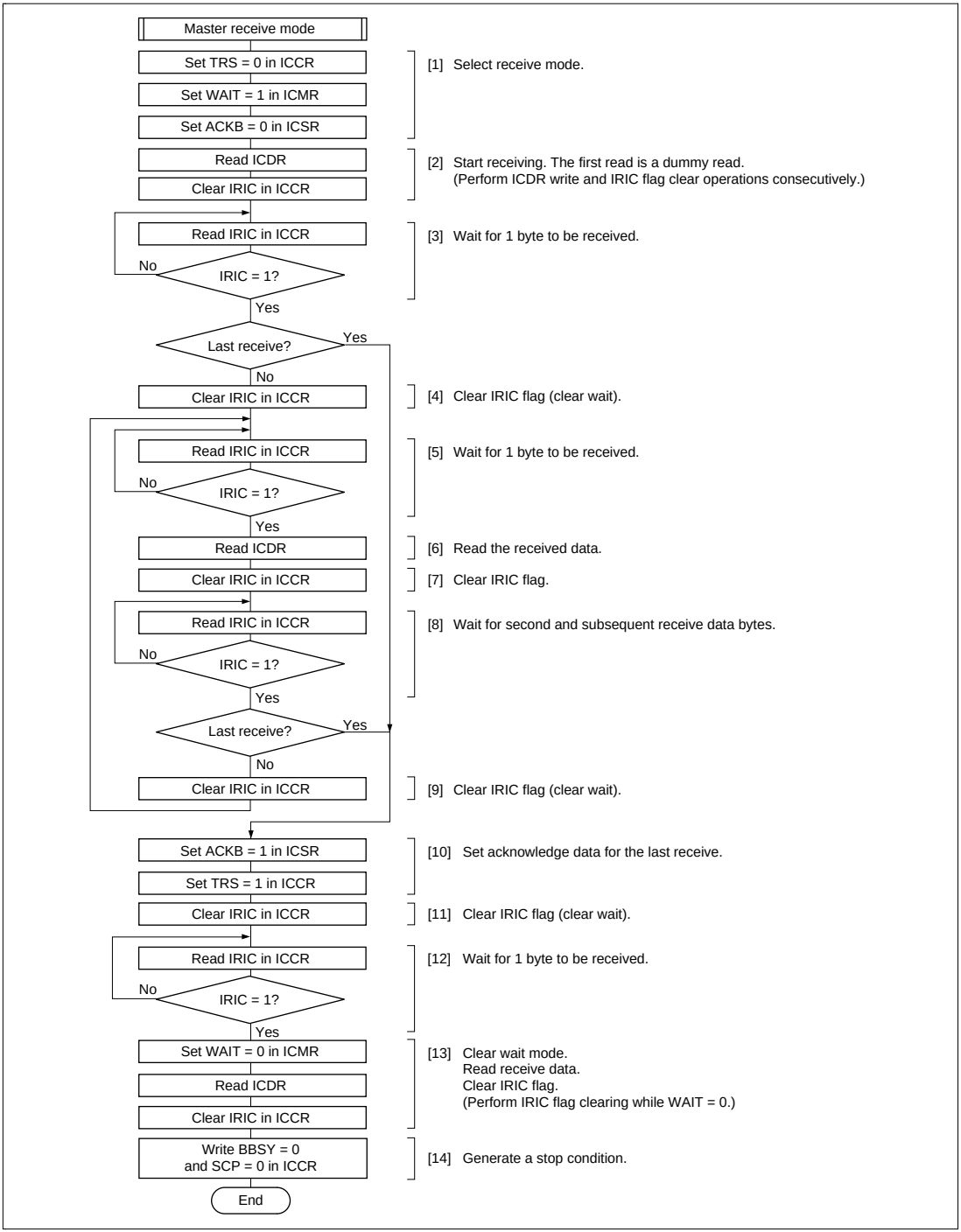


Figure 15-14 Flowchart for Master Receive Mode (Example)

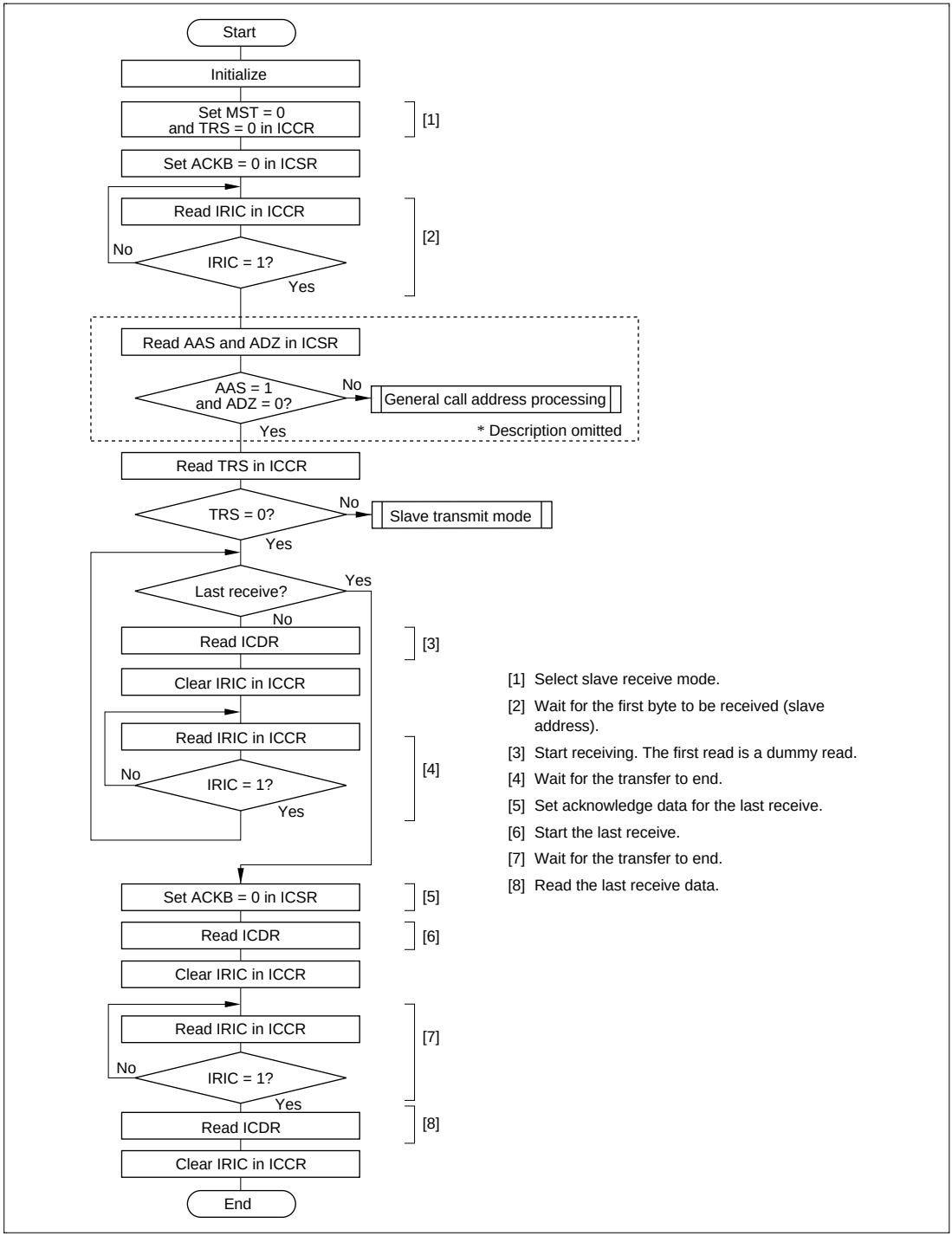


Figure 15-15 Flowchart for Slave Transmit Mode (Example)

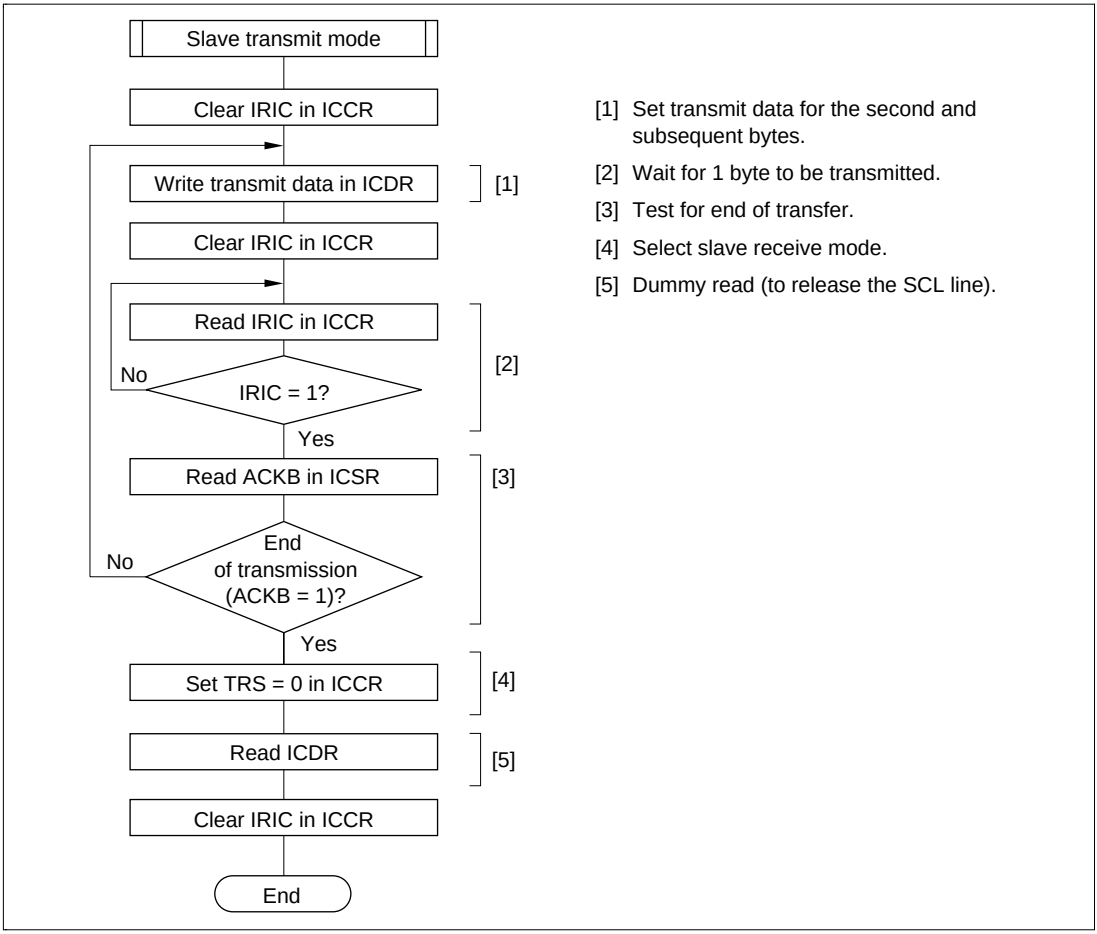


Figure 15-16 Flowchart for Slave Receive Mode (Example)

15.3.10 Initialization of Internal State

The IIC has a function for forcible initialization of its internal state if a deadlock occurs during communication.

Initialization is executed by (1) setting bits CLR3 to CLR0 in the DDCSWR register or (2) clearing the ICE bit. For details of settings for bits CLR3 to CLR0, see section 15.2.8, DDC Switch Register (DDCSWR).

Scope of Initialization: The initialization executed by this function covers the following items:

- TDRE and RDRF internal flags
- Transmit/receive sequencer and internal operating clock counter
- Internal latches for retaining the output state of the SCL and SDA pins (wait, clock, data output, etc.)

The following items are not initialized:

- Actual register values (ICDR, SAR, SARX, ICMR, ICCR, ICSR, DDCSWR, STCR)
- Internal latches used to retain register read information for setting/clearing flags in the ICMR, ICCR, ICSR, and DDCSWR registers
- The value of the ICMR register bit counter (BC2 to BC0)
- Generated interrupt sources (interrupt sources transferred to the interrupt controller)

Notes on Initialization:

- Interrupt flags and interrupt sources are not cleared, and so flag clearing measures must be taken as necessary.
- Basically, other register flags are not cleared either, and so flag clearing measures must be taken as necessary.
- When initialization is performed by means of the DDCSWR register, the write data for bits CLR3 to CLR0 is not retained. To perform IIC clearance, bits CLR3 to CLR0 must be written to simultaneously using an MOV instruction. Do not use a bit manipulation instruction such as BCLR. Similarly, when clearing is required again, all the bits must be written to simultaneously in accordance with the setting.
- If a flag clearing setting is made during transmission/reception, the IIC module will stop transmitting/receiving at that point and the SCL and SDA pins will be released. When transmission/reception is started again, register initialization, etc., must be carried out as necessary to enable correct communication as a system.

The value of the BBSY bit cannot be modified directly by this module clear function, but since the stop condition pin waveform is generated according to the state and release timing of the SCL and SDA pins, the BBSY bit may be cleared as a result. Similarly, state switching of other bits and flags may also have an effect.

To prevent problems caused by these factors, the following procedure should be used when initializing the IIC state.

1. Execute initialization of the internal state according to the setting of bits CLR3 to CLR0, or according to the ICE bit.
2. Execute a stop condition issuance instruction (write 0 to BBSY and SCP) to clear the BBST bit to 0, and wait for two transfer rate clock cycles.

3. Re-execute initialization of the internal state according to the setting of bits CLR3 to CLR0, or according to the ICE bit.
4. Initialize (re-set) the IIC registers.

15.4 Usage Notes

- In master mode, if an instruction to generate a start condition is immediately followed by an instruction to generate a stop condition, neither condition will be output correctly. To output consecutive start and stop conditions, after issuing the instruction that generates the start condition, read the relevant ports, check that SCL and SDA are both low, then issue the instruction that generates the stop condition. Note that SCL may not yet have gone low when BBSY is cleared to 0.

- Either of the following two conditions will start the next transfer. Pay attention to these conditions when reading or writing to ICDR.
  - — Write access to ICDR when ICE = 1 and TRS = 1 (including automatic transfer from ICDRT to ICDRS)
  - — Read access to ICDR when ICE = 1 and TRS = 0 (including automatic transfer from ICDRS to ICDRR)

- Table 15-6 shows the timing of SCL and SDA output in synchronization with the internal clock. Timings on the bus are determined by the rise and fall times of signals affected by the bus load capacitance, series resistance, and parallel resistance.

Table 15-6 I^2C Bus Timing (SCL and SDA Output)

Item	Symbol	Output Timing	Unit	Notes
SCL output cycle time	t_{SCLO}	$28t_{cyc}$ to $256t_{cyc}$	ns	Figure 24-22 (reference)
SCL output high pulse width	t_{SCLHO}	$0.5t_{SCLO}$	ns	
SCL output low pulse width	t_{SCLLO}	$0.5t_{SCLO}$	ns	
SDA output bus free time	t_{BUFO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Start condition output hold time	t_{STAHO}	$0.5t_{SCLO} – 1t_{cyc}$	ns	
Retransmission start condition output setup time	t_{STASO}	$1t_{SCLO}$	ns	
Stop condition output setup time	t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$	ns	
Data output setup time (master)	t_{SDASO}	$1t_{SCLLO} – 3t_{cyc}$	ns	
Data output setup time (slave)		$1t_{SCLL} – 3t_{cyc}$		
Data output hold time	t_{SDAHO}	$3t_{cyc}$	ns	

- SCL and SDA input is sampled in synchronization with the internal clock. The AC timing therefore depends on the system clock cycle t_{cyc}, as shown in table 24-8 in section 24, Electrical Characteristics. Note that the I^2C bus interface AC timing specifications will not be met with a system clock frequency of less than 5 MHz.

- The I^2C bus interface specification for the SCL rise time t_{sr} is under 1000 ns (300 ns for high-speed mode). In master mode, the I^2C bus interface monitors the SCL line and synchronizes one bit at a time during communication. If t_{sr} (the time for SCL to go from low to V_{IH}) exceeds the time determined by the input clock of the I^2C bus interface, the high period of SCL is extended. The SCL rise time is determined by the pull-up resistance and load capacitance of the SCL line. To insure proper operation at the set transfer rate, adjust the pull-up resistance and load capacitance so that the SCL rise time does not exceed the values given in the table below.

Table 15-7 Permissible SCL Rise Time (t_{Sr}) Values

			Time Indication					
IICX	t_{cyc} Indication		I^2C Bus Specification (Max.)	ø = 5 MHz	ø = 8 MHz	ø = 10 MHz	ø = 16 MHz	ø = 20 MHz
0	$7.5t_{cyc}$	Standard mode	1000 ns	1000 ns	937 ns	750 ns	468 ns	375 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns
1	$17.5t_{cyc}$	Standard mode	1000 ns	1000 ns	1000 ns	1000 ns	1000 ns	875 ns
		High-speed mode	300 ns	300 ns	300 ns	300 ns	300 ns	300 ns

- The I^2C bus interface specifications for the SCL and SDA rise and fall times are under 1000 ns and 300 ns. The I^2C bus interface SCL and SDA output timing is prescribed by t_{Scyc} and t_{cyc}, as shown in table 15-6. However, because of the rise and fall times, the I^2C bus interface specifications may not be satisfied at the maximum transfer rate. Table 15-8 shows output timing calculations for different operating frequencies, including the worst-case influence of rise and fall times.

  t_{BUFO} fails to meet the I^2C bus interface specifications at any frequency. The solution is either (a) to provide coding to secure the necessary interval (approximately 1 µs) between issuance of a stop condition and issuance of a start condition, or (b) to select devices whose input timing permits this output timing for use as slave devices connected to the I^2C bus.

  t_{SCLLO} in high-speed mode and t_{STASO} in standard mode fail to satisfy the I^2C bus interface specifications for worst-case calculations of t_{Sr}/t_{Sf}. Possible solutions that should be

investigated include (a) adjusting the rise and fall times by means of a pull-up resistor and capacitive load, (b) reducing the transfer rate to meet the specifications, or (c) selecting devices whose input timing permits this output timing for use as slave devices connected to the I^2C bus.

Table 15-8 I^2C Bus Timing (with Maximum Influence of t_{Sr}/t_{Sf})

			Time Indication (at Maximum Transfer Rate) [ns]						
Item	**t_{cyc} Indication**		**t_{Sr}/t_{Sf} Influence (Max.)**	**I^2C Bus Specification (Min.)**	**ø = 5 MHz**	**ø = 8 MHz**	**ø = 10 MHz**	**ø = 16 MHz**	**ø = 20 MHz**
t_{SCLHO}	$0.5t_{SCLO}$ ($-t_{Sr}$)	Standard mode	–1000	4000	4000	4000	4000	4000	4000
		High-speed mode	–300	600	950	950	950	950	950
t_{SCLLO}	$0.5t_{SCLO}$ ($-t_{Sf}$)	Standard mode	–250	4700	4750	4750	4750	4750	4750
		High-speed mode	–250	1300	1000*1	1000*1	1000*1	1000*1	1000*1
t_{BUFO}	$0.5t_{SCLO} - 1t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	4700	3800*1	3875*1	3900*1	3938*1	3950*1
		High-speed mode	–300	1300	750*1	825*1	850*1	888*1	900*1
t_{STAHO}	$0.5t_{SCLO} - 1t_{cyc}$ ($-t_{Sf}$)	Standard mode	–250	4000	4550	4625	4650	4688	4700
		High-speed mode	–250	600	800	875	900	938	950
t_{STASO}	$1t_{SCLO}$ ($-t_{Sr}$)	Standard mode	–1000	4700	9000	9000	9000	9000	9000
		High-speed mode	–300	600	2200	2200	2200	2200	2200
t_{STOSO}	$0.5t_{SCLO} + 2t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	4000	4400	4250	4200	4125	4100
		High-speed mode	–300	600	1350	1200	1150	1075	1050
t_{SDASO} (master)	$1t_{SCLLO}$*2 – $3t_{cyc}$ ($-t_{Sr}$)	Standard mode	–1000	250	3100	3325	3400	3513	3550
		High-speed mode	–200	100	400	625	700	813	850
t_{SDASO} (slave)	$1t_{SCLL}$*2 – $3t_{cyc}$*2 ($-t_{Sr}$)	Standard mode	–1000	250	3100	3325	3400	3513	3550
		High-speed mode	–300	100	400	625	700	813	850

Item	t_{cyc} Indication		Time Indication (at Maximum Transfer Rate) [ns] t_{Sr}/t_{Sf} Influence (Max.)	I^2C Bus Specification (Min.)	ø = 5 MHz	ø = 8 MHz	ø = 10 MHz	ø = 16 MHz	ø = 20 MHz
t_{SDAHO}	$3t_{cyc}$	Standard mode	0	0	600	375	300	188	150
		High-speed mode	0	0	600	375	300	188	150

Notes: *1 Does not meet the I^2C bus interface specification. Remedial action such as the following is necessary: (a) secure a start/stop condition issuance interval; (b) adjust the rise and fall times by means of a pull-up resistor and capacitive load; (c) reduce the transfer rate; (d) select slave devices whose input timing permits this output timing.

The values in the above table will vary depending on the settings of the IICX bit and bits CKS0 to CKS2. Depending on the frequency it may not be possible to achieve the maximum transfer rate; therefore, whether or not the I^2C bus interface specifications are met must be determined in accordance with the actual setting conditions.

*2 Calculated using the I^2C bus specification values (standard mode: 4700 ns min.; high-speed mode: 1300 ns min.).

- Note on ICDR Read at End of Master Reception

  To halt reception at the end of a receive operation in master receive mode, set the TRS bit to 1 and write 0 to BBSY and SCP in ICCR. This changes SDA from low to high when SCL is high, and generates the stop condition. After this, receive data can be read by means of an ICDR read, but if data remains in the buffer the ICDRS receive data will not be transferred to ICDR, and so it will not be possible to read the second byte of data.

  If it is necessary to read the second byte of data, issue the stop condition in master receive mode (i.e. with the TRS bit cleared to 0). When reading the receive data, first confirm that the BBSY bit in the ICCR register is cleared to 0, the stop condition has been generated, and the bus has been released, then read the ICDR register with TRS cleared to 0.

  Note that if the receive data (ICDR data) is read in the interval between execution of the instruction for issuance of the stop condition (writing of 0 to BBSY and SCP in ICCR) and the actual generation of the stop condition, the clock may not be output correctly in subsequent master transmission.

  Clearing of the MST bit after completion of master transmission/reception, or other modifications of IIC control bits to change the transmit/receive operating mode or settings, must be carried out during interval (a) in figure 15-17 (after confirming that the BBSY bit has been cleared to 0 in the ICCR register).

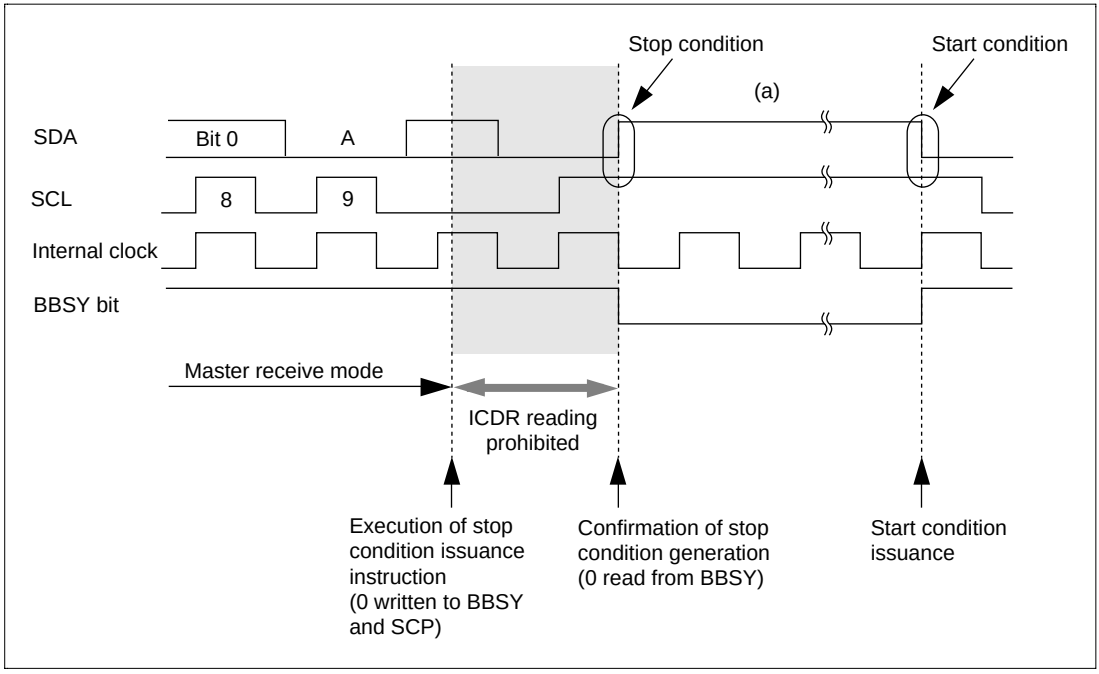
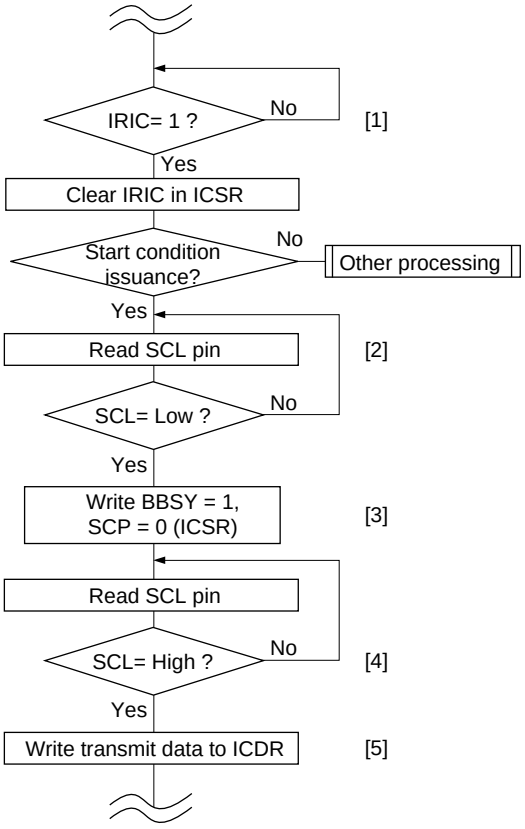


Figure 15-17 Points for Attention Concerning Reading of Master Receive Data

- Notes on Start Condition Issuance for Retransmission

  Figure 15-18 shows the timing of start condition issuance for retransmission, and the timing for subsequently writing data to ICDR, together with the corresponding flowchart.



[1] Wait for end of 1-byte transfer

[2] Determine whether SCL is low

[3] Issue restart condition instruction for retransmission

[4] Determine whether SCL is high

[5] Set transmit data (slave address + R/$\overline{W}$)

Note: Program so that processing from [3] to [5] is executed continuously.

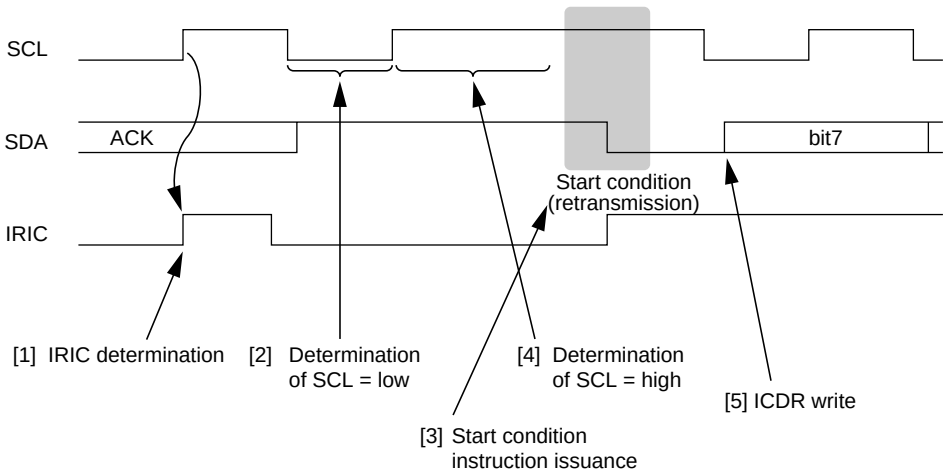


Figure 15-18 Flowchart and Timing of Start Condition Instruction Issuance for Retransmission

- Notes on I^2C Bus Interface Stop Condition Instruction Issuance

  If the rise time of the 9th SCL acknowledge exceeds the specification because the bus load capacitance is large, or if there is a slave device of the type that drives SCL low to effect a wait, issue the stop condition instruction after reading SCL and determining it to be low, as shown below.

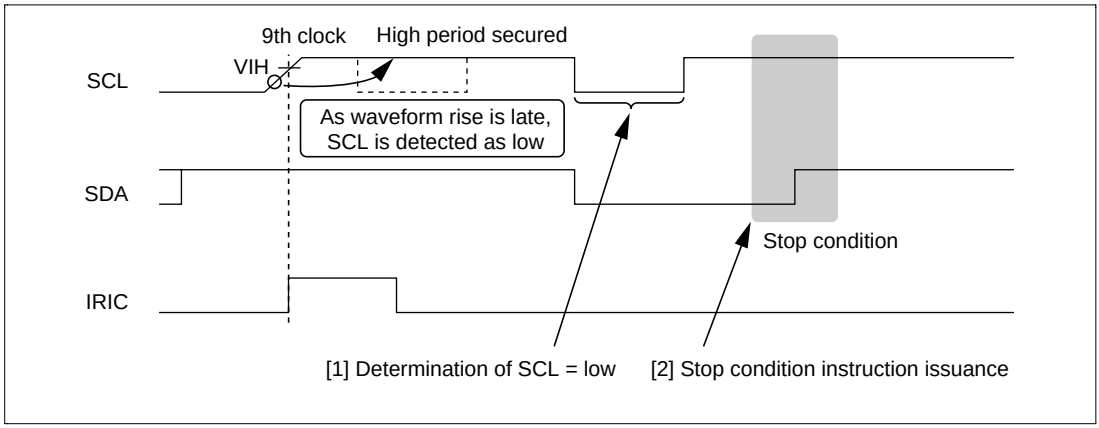


Figure 15-19 Timing of Stop Condition Issuance

Section 16 Hitachi Controller Area Network (HCAN)

16.1 Overview

The HCAN is a module for controlling a controller area network (CAN) for realtime communication in vehicular and industrial equipment systems, etc. The chip has a 2-channel on-chip HCAN module.

Reference: Bosch CAN Specification Version 2.0 1991, Robert Bosch GmbH

16.1.1 Features

- CAN version: Bosch 2.0B active compatible
  - — Communication systems:
    NRZ (Non-Return to Zero) system (with bit-stuffing function)
    Broadcast communication system
  - — Transmission path: Bidirectional 2-wire serial communication
  - — Communication speed: Max. 1 Mbps
  - — Data length: 0 to 8 bytes
- Number of channel: 2 (HCAN0, HCAN1)
- Data buffers: 16 per channel (one receive-only buffer and 15 buffers settable for transmission/reception)
- Data transmission: Choice of two methods:
  - — Mailbox (buffer) number order (low-to-high)
  - — Message priority (identifier) high-to-low order
- Data reception: Two methods:
  - — Message identifier match (transmit/receive-setting buffers)
  - — Reception with message identifier masked (receive-only)
- CPU interrupts: Two interrupt vectors for 12 interrupt causes per channel:
  - — Error interrupt
  - — Reset processing interrupt
  - — Message reception interrupt (mailbox 1 to 15)
  - — Message reception interrupt (mailbox 0)
  - — Message transmission interrupt
- HCAN operating modes: Support for various modes:
  - — Hardware reset
  - — Software reset
  - — Normal status (error-active, error-passive)

— Bus off status
— HCAN configuration mode
— HCAN sleep mode
— HCAN halt mode

- Other features: DTC can be activated by message reception mailbox (HCAN mailbox 0 only)

16.1.2 Block Diagram

Figure 16-1 shows a block diagram of the HCAN.

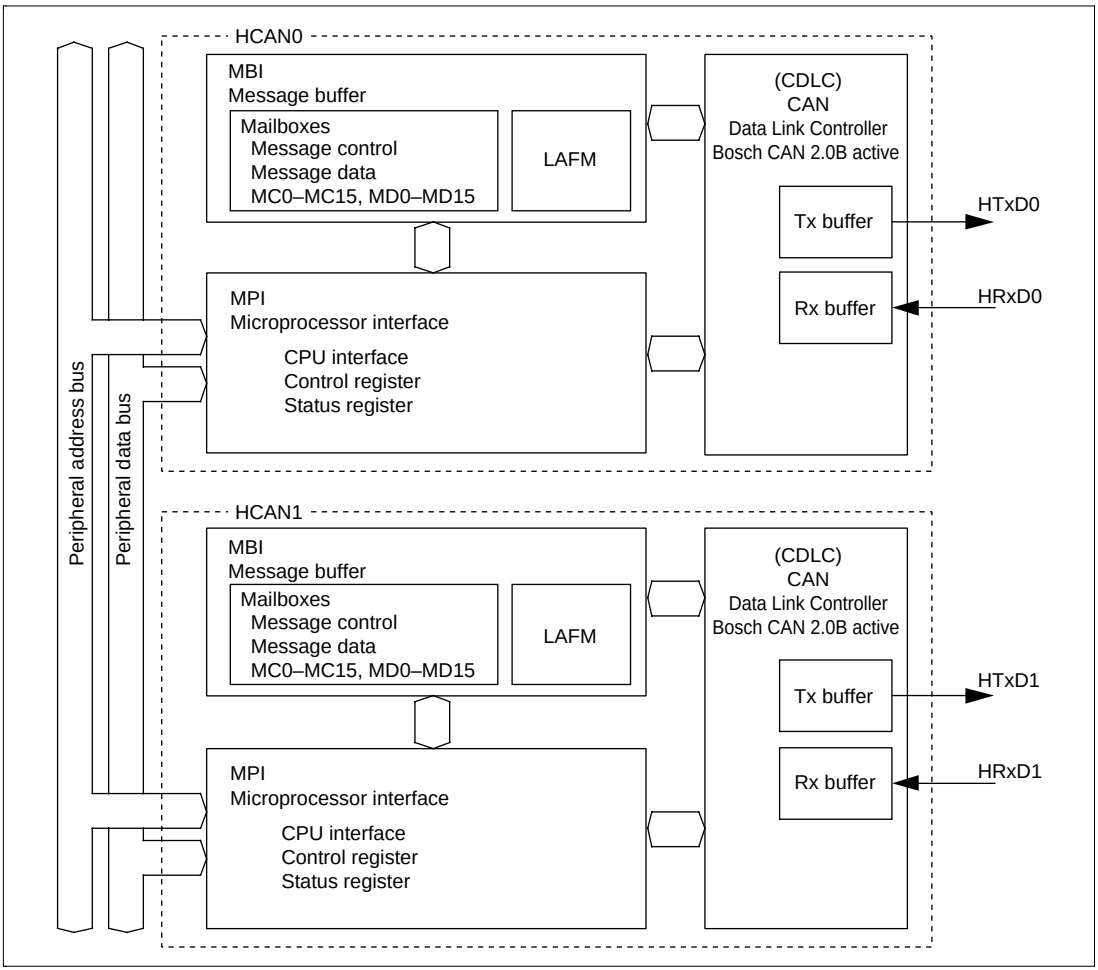


Figure 16-1 HCAN Block Diagram

Message Buffer Interface (MBI): The MBI, consisting of mailboxes and a local acceptance filter mask (LAFM), stores CAN transmit/receive messages (identifiers, data, etc.) Transmit messages are written by the CPU. For receive messages, the data received by the CDLC is stored automatically.

Microprocessor Interface (MPI): The MPI, consisting of a bus interface, control register, status register, etc., controls HCAN internal data, statuses, and so forth.

CAN Data Link Controller (CDLC): The CDLC performs transmission and reception of messages conforming to the Bosch CAN Ver. 2.0B active standard (data frames, remote frames, error frames, overload frames, inter-frame spacing), as well as CRC checking, bus arbitration, and other functions.

16.1.3 Pin Configuration

Table 16-1 shows the HCAN's pins.

When using HCAN pins, settings must be made in the HCAN configuration mode (during initialization: MCR0 = 1 and GSR3 = 1).

Table 16-1 HCAN Pins

Channel	Name	Abbreviation	Input/Output	Function
0	HCAN transmit data pin 0	HTxD0	Output	Channel 0 CAN bus transmission pin
	HCAN receive data pin 0	HRxD0	Input	Channel 0 CAN bus reception pin
1	HCAN transmit data pin 1	HTxD1	Output	Channel 1 CAN bus transmission pin
	HCAN receive data pin 1	HRxD1	Input	Channel 1 CAN bus reception pin

A bus driver is necessary between the pins and the CAN bus. A Philips PCA82C250 compatible model is recommended.

16.1.4 Register Configuration

Table 16-2 lists the HCAN's registers.

Table 16-2 HCAN Registers

Channel	Name	Abbre-viation	R/W	Initial Value	Address*	Access Size
0	Master control register	MCR	R/W	H'01	H'F800	8 bits 16 bits
	General status register	GSR	R/W	H'0C	H'F801	8 bits
	Bit configuration register	BCR	R/W	H'0000	H'F802	8/16 bits
	Mailbox configuration register	MBCR	R/W	H'0100	H'F804	8/16 bits
	Transmit wait register	TXPR	R/W	H'0000	H'F806	8/16 bits
	Transmit wait cancel register	TXCR	R/W	H'0000	H'F808	8/16 bits
	Transmit acknowledge register	TXACK	R/W	H'0000	H'F80A	8/16 bits
	Abort acknowledge register	ABACK	R/W	H'0000	H'F80C	8/16 bits
	Receive complete register	RXPR	R/W	H'0000	H'F80E	8/16 bits
	Remote request register	RFPR	R/W	H'0000	H'F810	8/16 bits
	Interrupt register	IRR	R/W	H'0100	H'F812	8/16 bits
	Mailbox interrupt mask register	MBIMR	R/W	H'FFFF	H'F814	8/16 bits
	Interrupt mask register	IMR	R/W	H'FEFF	H'F816	8/16 bits
	Receive error counter	REC	R	H'00	H'F818	8 bits 16 bits
	Transmit error counter	TEC	R	H'00	H'F819	8 bits
	Unread message status register	UMSR	R/W	H'0000	H'F81A	8/16 bits
	Local acceptance filter mask L	LAFML	R/W	H'0000	H'F81C	8/16 bits
	Local acceptance filter mask H	LAFMH	R/W	H'0000	H'F81E	8/16 bits

Channel	Name	Abbreviation	R/W	Initial Value	Address*	Access Size
0	Message control 0 [1:8]	MC0 [1:8]	R/W	Undefined	H'F820	8/16 bits
	Message control 1 [1:8]	MC1 [1:8]	R/W	Undefined	H'F828	8/16 bits
	Message control 2 [1:8]	MC2 [1:8]	R/W	Undefined	H'F830	8/16 bits
	Message control 3 [1:8]	MC3 [1:8]	R/W	Undefined	H'F838	8/16 bits
	Message control 4 [1:8]	MC4 [1:8]	R/W	Undefined	H'F840	8/16 bits
	Message control 5 [1:8]	MC5 [1:8]	R/W	Undefined	H'F848	8/16 bits
	Message control 6 [1:8]	MC6 [1:8]	R/W	Undefined	H'F850	8/16 bits
	Message control 7 [1:8]	MC7 [1:8]	R/W	Undefined	H'F858	8/16 bits
	Message control 8 [1:8]	MC8 [1:8]	R/W	Undefined	H'F860	8/16 bits
	Message control 9 [1:8]	MC9 [1:8]	R/W	Undefined	H'F868	8/16 bits
	Message control 10 [1:8]	MC10 [1:8]	R/W	Undefined	H'F870	8/16 bits
	Message control 11 [1:8]	MC11 [1:8]	R/W	Undefined	H'F878	8/16 bits
	Message control 12 [1:8]	MC12 [1:8]	R/W	Undefined	H'F880	8/16 bits
	Message control 13 [1:8]	MC13 [1:8]	R/W	Undefined	H'F888	8/16 bits
	Message control 14 [1:8]	MC14 [1:8]	R/W	Undefined	H'F890	8/16 bits
	Message control 15 [1:8]	MC15 [1:8]	R/W	Undefined	H'F898	8/16 bits
	Message data 0 [1:8]	MD0 [1:8]	R/W	Undefined	H'F8B0	8/16 bits
	Message data 1 [1:8]	MD1 [1:8]	R/W	Undefined	H'F8B8	8/16 bits
	Message data 2 [1:8]	MD2 [1:8]	R/W	Undefined	H'F8C0	8/16 bits
	Message data 3 [1:8]	MD3 [1:8]	R/W	Undefined	H'F8C8	8/16 bits
	Message data 4 [1:8]	MD4 [1:8]	R/W	Undefined	H'F8D0	8/16 bits
	Message data 5 [1:8]	MD5 [1:8]	R/W	Undefined	H'F8D8	8/16 bits
	Message data 6 [1:8]	MD6 [1:8]	R/W	Undefined	H'F8E0	8/16 bits
	Message data 7 [1:8]	MD7 [1:8]	R/W	Undefined	H'F8E8	8/16 bits
	Message data 8 [1:8]	MD8 [1:8]	R/W	Undefined	H'F8F0	8/16 bits
	Message data 9 [1:8]	MD9 [1:8]	R/W	Undefined	H'F8F8	8/16 bits
	Message data 10 [1:8]	MD10 [1:8]	R/W	Undefined	H'F900	8/16 bits
	Message data 11 [1:8]	MD11 [1:8]	R/W	Undefined	H'F908	8/16 bits
	Message data 12 [1:8]	MD12 [1:8]	R/W	Undefined	H'F910	8/16 bits
	Message data 13 [1:8]	MD13 [1:8]	R/W	Undefined	H'F918	8/16 bits
	Message data 14 [1:8]	MD14 [1:8]	R/W	Undefined	H'F920	8/16 bits
	Message data 15 [1:8]	MD15 [1:8]	R/W	Undefined	H'F928	8/16 bits

Channel	Name	Abbre-viation	R/W	Initial Value	Address*	Access Size
1	Master control register	MCR	R/W	H'01	H'FA00	8 bits 16 bits
	General status register	GSR	R/W	H'0C	H'FA01	8 bits
	Bit configuration register	BCR	R/W	H'0000	H'FA02	8/16 bits
	Mailbox configuration register	MBCR	R/W	H'0100	H'FA04	8/16 bits
	Transmit wait register	TXPR	R/W	H'0000	H'FA06	8/16 bits
	Transmit wait cancel register	TXCR	R/W	H'0000	H'FA08	8/16 bits
	Transmit acknowledge register	TXACK	R/W	H'0000	H'FA0A	8/16 bits
	Abort acknowledge register	ABACK	R/W	H'0000	H'FA0C	8/16 bits
	Receive complete register	RXPR	R/W	H'0000	H'FA0E	8/16 bits
	Remote request register	RFPR	R/W	H'0000	H'FA10	8/16 bits
	Interrupt register	IRR	R/W	H'0100	H'FA12	8/16 bits
	Mailbox interrupt mask register	MBIMR	R/W	H'FFFF	H'FA14	8/16 bits
	Interrupt mask register	IMR	R/W	H'FEFF	H'FA16	8/16 bits
	Receive error counter	REC	R	H'00	H'FA18	8 bits 16 bits
	Transmit error counter	TEC	R	H'00	H'FA19	8 bits
	Unread message status register	UMSR	R/W	H'0000	H'FA1A	8/16 bits
	Local acceptance filter mask L	LAFML	R/W	H'0000	H'FA1C	8/16 bits
	Local acceptance filter mask H	LAFMH	R/W	H'0000	H'FA1E	8/16 bits

Channel	Name	Abbreviation	R/W	Initial Value	Address*	Access Size
1	Message control 0 [1:8]	MC0 [1:8]	R/W	Undefined	H'FA20	8/16 bits
	Message control 1 [1:8]	MC1 [1:8]	R/W	Undefined	H'FA28	8/16 bits
	Message control 2 [1:8]	MC2 [1:8]	R/W	Undefined	H'FA30	8/16 bits
	Message control 3 [1:8]	MC3 [1:8]	R/W	Undefined	H'FA38	8/16 bits
	Message control 4 [1:8]	MC4 [1:8]	R/W	Undefined	H'FA40	8/16 bits
	Message control 5 [1:8]	MC5 [1:8]	R/W	Undefined	H'FA48	8/16 bits
	Message control 6 [1:8]	MC6 [1:8]	R/W	Undefined	H'FA50	8/16 bits
	Message control 7 [1:8]	MC7 [1:8]	R/W	Undefined	H'FA58	8/16 bits
	Message control 8 [1:8]	MC8 [1:8]	R/W	Undefined	H'FA60	8/16 bits
	Message control 9 [1:8]	MC9 [1:8]	R/W	Undefined	H'FA68	8/16 bits
	Message control 10 [1:8]	MC10 [1:8]	R/W	Undefined	H'FA70	8/16 bits
	Message control 11 [1:8]	MC11 [1:8]	R/W	Undefined	H'FA78	8/16 bits
	Message control 12 [1:8]	MC12 [1:8]	R/W	Undefined	H'FA80	8/16 bits
	Message control 13 [1:8]	MC13 [1:8]	R/W	Undefined	H'FA88	8/16 bits
	Message control 14 [1:8]	MC14 [1:8]	R/W	Undefined	H'FA90	8/16 bits
	Message control 15 [1:8]	MC15 [1:8]	R/W	Undefined	H'FA98	8/16 bits
	Message data 0 [1:8]	MD0 [1:8]	R/W	Undefined	H'FAB0	8/16 bits
	Message data 1 [1:8]	MD1 [1:8]	R/W	Undefined	H'FAB8	8/16 bits
	Message data 2 [1:8]	MD2 [1:8]	R/W	Undefined	H'FAC0	8/16 bits
	Message data 3 [1:8]	MD3 [1:8]	R/W	Undefined	H'FAC8	8/16 bits
	Message data 4 [1:8]	MD4 [1:8]	R/W	Undefined	H'FAD0	8/16 bits
	Message data 5 [1:8]	MD5 [1:8]	R/W	Undefined	H'FAD8	8/16 bits
	Message data 6 [1:8]	MD6 [1:8]	R/W	Undefined	H'FAE0	8/16 bits
	Message data 7 [1:8]	MD7 [1:8]	R/W	Undefined	H'FAE8	8/16 bits
	Message data 8 [1:8]	MD8 [1:8]	R/W	Undefined	H'FAF0	8/16 bits
	Message data 9 [1:8]	MD9 [1:8]	R/W	Undefined	H'FAF8	8/16 bits
	Message data 10 [1:8]	MD10 [1:8]	R/W	Undefined	H'FB00	8/16 bits
	Message data 11 [1:8]	MD11 [1:8]	R/W	Undefined	H'FB08	8/16 bits
	Message data 12 [1:8]	MD12 [1:8]	R/W	Undefined	H'FB10	8/16 bits
	Message data 13 [1:8]	MD13 [1:8]	R/W	Undefined	H'FB18	8/16 bits
	Message data 14 [1:8]	MD14 [1:8]	R/W	Undefined	H'FB20	8/16 bits
	Message data 15 [1:8]	MD15 [1:8]	R/W	Undefined	H'FB28	8/16 bits
All	Module stop control register C	MSTPCRC	R/W	H'FF	H'FDEA	8/16 bits

Note: * Lower 16 bits of the address

16.2 Register Descriptions

16.2.1 Master Control Register (MCR)

The master control register (MCR) is an 8-bit readable/writable register that controls the CAN interface.

MCR

Bit:	7	6	5	4	3	2	1	0
	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R	R/W	R	R	R/W	R/W	R/W

Bit 7—HCAN Sleep Mode Release (MCR7): Enables or disables HCAN sleep mode release by bus operation.

Bit 7: MCR7	Description	
0	HCAN sleep mode release by CAN bus operation disabled	(Initial value)
1	HCAN sleep mode release by CAN bus operation enabled	

Bit 6—Reserved: This bit always reads 0. The write value should always be 0.

Bit 5—HCAN Sleep Mode (MCR5): Enables or disables HCAN sleep mode transition.

Bit 5: MCR5	Description	
0	HCAN sleep mode released	(Initial value)
1	Transition to HCAN sleep mode enabled	

Bits 4 and 3—Reserved: These bits always read 0. The write value should always be 0.

Bit 2—Message Transmission Method (MCR2): Selects the transmission method for transmit messages.

Bit 2: MCR2	Description	
0	Transmission order determined by message identifier priority	(Initial value)
1	Transmission order determined by mailbox (buffer) number priority (TXPR1 > TXPR15)	

Bit 1—Halt Request (MCR1): Controls halting of the HCAN module.

Bit 1: MCR1	Description	
0	HCAN normal operating mode	(Initial value)
1	HCAN halt mode transition request	

Bit 0—Reset Request (MCR0): Controls resetting of the HCAN module.

Bit 0: MCR0	Description	
0	Normal operating mode (MCR0 = 0 and GSR3 = 0) [Setting condition] When 0 is written after an HCAN reset	
1	HCAN reset mode transition request	(Initial value)

In order for GSR3 to change from 1 to 0 after 0 is written to MCR0, time is required before the HCAN is internally reset. There is consequently a delay before GSR3 is cleared to 0 after MCR0 is cleared to 0.

16.2.2 General Status Register (GSR)

The general status register (GSR) is an 8-bit readable register that indicates the status of the CAN bus.

GSR

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	GSR3	GSR2	GSR1	GSR0
Initial value:	0	0	0	0	1	1	0	0
R/W:	R	R	R	R	R	R	R	R

Bits 7 to 4—Reserved: These bits always read 0.

Bit 3—Reset Status Bit (GSR3): Indicates whether the HCAN module is in the normal operating state or the reset state. Writes are invalid.

Bit 3: MCR3	Description
0	Normal operating state [Setting condition] After an HCAN internal reset
1	Configuration mode [Reset condition] MCR0 reset mode and sleep mode (Initial value)

Bit 2—Message Transmission Status Flag (GSR2): Flag that indicates whether the module is currently in the message transmission period. The "message transmission period" is the period from the start of message transmission (SOF) until the end of a 3-bit intermission interval after EOF (End of Frame). Writes are invalid.

Bit 2: GSR2	Description
0	Message transmission period
1	[Reset condition] (Initial value) Idle period

Bit 1—Transmit/Receive Warning Flag (GSR1): Flag that indicates an error warning. Writes are invalid.

Bit 1: GSR1	Description
0	[Reset condition] When TEC < 96 and REC < 96 or TEC ≥ 256 (Initial value)
1	When TEC ≥ 96 or REC ≥ 96

Bit 0—Bus Off Flag (GSR0): Flag that indicates the bus off state. Writes are invalid.

Bit 0: GSR0	Description
0	[Reset condition] Recovery from bus off state (Initial value)
1	When TEC ≥ 256 (bus off state)

16.2.3 Bit Configuration Register (BCR)

The bit configuration register (BCR) is a 16-bit readable/writable register that is used to set CAN bit timing parameters and the baud rate prescaler.

BCR

Bit:	15	14	13	12	11	10	9	8
	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

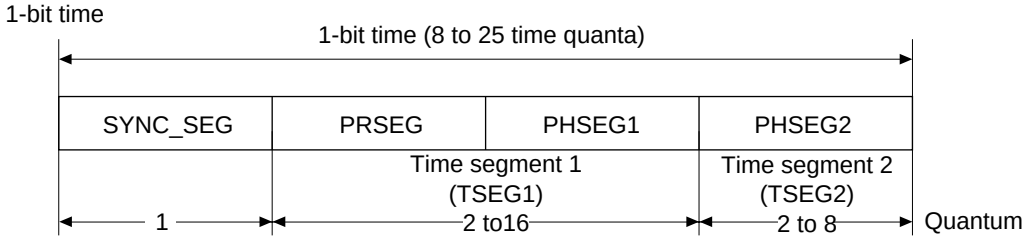
Bit:	7	6	5	4	3	2	1	0
	BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 and 14—Resynchronization Jump Width (SJW): These bits set the bit synchronization range.

Bit 15: BCR7	Bit 14: BCR6	Description	
0	0	Bit synchronization width = 1 time quantum	(Initial value)
	1	Bit synchronization width = 2 time quanta	
1	0	Bit synchronization width = 3 time quanta	
	1	Bit synchronization width = 4 time quanta	

Bits 13 to 8—Baud Rate Prescaler (BRP): These bits are used to set the CAN bus baud rate.

Bit 13: BCR5	Bit 12: BCR4	Bit 11: BCR3	Bit 10: BCR2	Bit 9: BCR1	Bit 8: BCR0	Description	
0	0	0	0	0	0	2 × system clock	(Initial value)
0	0	0	0	0	1	4 × system clock	
0	0	0	0	1	0	6 × system clock	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
.	.	.	.	.	.	.	
1	1	1	1	1	1	128 × system clock	



SYNC_SEG: Segment for establishing synchronization of nodes on the CAN bus. (Normal bit edge transitions occur in this segment.)
PRSEG: Segment for compensating for physical delay between networks.
PHSEG1: Buffer segment for correcting phase drift (positive). (This segment is extended when synchronization (resynchronization) is established.)
PHSEG2: Buffer segment for correcting phase drift (negative). (This segment is shortened when synchronization (resynchronization) is established.)

HCAN bit rate calculation:

$$\text{Bit rate} = \frac{f_{CLK}}{2 \times (BRP + 1) \times (3 + TSEG1 + TSEG2)}$$

f_{CLK}: peripheral clock (ø)

Note: The BCR values are used for BRP, TSEG1, and TSEG2.

BCR Setting Constraints

$TSEG1 > TSEG2 \geq (1 + SJW)$ (SJW = 1 to 4)

3 + TSEG1 + TSEG2 = 8 to 25 time quanta

These constraints allow the setting range shown in table 16-3 for TSEG1 and TSEG2 in BCR.

Table 16-3 Setting Range for TSEG1 and TSEG2 in BCR

		TSEG2 (BCR [14:12])						
		001	**010**	**011**	**100**	**101**	**110**	**111**
TSEG1 (BCR [11:8])	**0011**	No	Yes	No	No	No	No	No
	0100	Yes*	Yes	Yes	No	No	No	No
	0101	Yes*	Yes	Yes	Yes	No	No	No
	0110	Yes*	Yes	Yes	Yes	Yes	No	No
	0111	Yes*	Yes	Yes	Yes	Yes	Yes	No
	1000	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1001	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1010	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1011	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1100	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1101	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1110	Yes*	Yes	Yes	Yes	Yes	Yes	Yes
	1111	Yes*	Yes	Yes	Yes	Yes	Yes	Yes

Note: The time quanta value for TSEG1 and TSEG2 is the TSEG value + 1.

* Setting is enabled except when BRP [13:8] = B'000000.

Bit 7—Bit Sample Point (BSP): Sets the point at which data is sampled.

Bit 7: BCR15	**Description**
0	Bit sampling at one point (end of time segment 1 (TSEG1)) (Initial value)
1	Bit sampling at three points (end of time segment 1 (TSEG1) and preceding and following time quanta)

Bits 6 to 4—Time Segment 2 (TSEG2): These bits are used to set the segment for correcting 1-bit time error. A value from 2 to 8 can be set.

Bit 6: BCR14	Bit 5: BCR13	Bit 4: BCR12	Description	
0	0	0	Setting prohibited	(Initial value)
		1	TSEG2 = 2 time quanta	
	1	0	TSEG2 = 3 time quanta	
		1	TSEG2 = 4 time quanta	
1	0	0	TSEG2 = 5 time quanta	
		1	TSEG2 = 6 time quanta	
	1	0	TSEG2 = 7 time quanta	
		1	TSEG2 = 8 time quanta	

Bits 3 to 0—Time Segment 1 (TSEG1): These bits are used to set the segment for absorbing output buffer, CAN bus, and input buffer delay. A value from 1 to 16 can be set.

Bit 3: BCR11	Bit 2: BCR10	Bit 1: BCR9	Bit 0: BCR8	Description	
0	0	0	0	Setting prohibited	(Initial value)
0	0	0	1	Setting prohibited	
0	0	1	0	Setting prohibited	
0	0	1	1	TSEG1 = 4 time quanta	
0	1	0	0	TSEG1 = 5 time quanta	
.	.	.	.	.	
.	.	.	.	.	
.	.	.	.	.	
1	1	1	1	TSEG1 = 16 time quanta	

16.2.4 Mailbox Configuration Register (MBCR)

The mailbox configuration register (MBCR) is a 16-bit readable/writable register that is used to set mailbox (buffer) transmission/reception.

MBCR

Bit:	15	14	13	12	11	10	9	8
	MBCR7	MBCR6	MBCR5	MBCR4	MBCR3	MBCR2	MBCR1	—
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	7	6	5	4	3	2	1	0
	MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9	MBCR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 9 and 7 to 0—Mailbox Setting Register (MBCR7 to MBCR1, MBCR15 to MBCR8): These bits set the polarity of the corresponding mailboxes.

Bit x: MBCRx	Description	
0	Corresponding mailbox is set for transmission	(Initial value)
1	Corresponding mailbox is set for reception	

(x = 15 to 9, 7 to 0)

Bit 8—Reserved: This bit always reads 1. The write value should always be 1.

16.2.5 Transmit Wait Register (TXPR)

The transmit wait register (TXPR) is a 16-bit readable/writable register that is used to set a transmit wait after a transmit message is stored in a mailbox (buffer) (CAN bus arbitration wait).

TXPR

Bit:	15	14	13	12	11	10	9	8
	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	7	6	5	4	3	2	1	0
	TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 9 and 7 to 0—Transmit Wait Register (TXPR7 to TXPR1, TXPR15 to TXPR8): These bits set a transmit wait for the corresponding mailboxes.

Bit x: TXPRx	Description
0	Transmit message idle state in corresponding mailbox (Initial value) [Clearing condition] Message transmission completion and cancellation completion
1	Transmit message transmit wait in corresponding mailbox (CAN bus arbitration)

(x = 15 to 9, 7 to 0)

Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

16.2.6 Transmit Wait Cancel Register (TXCR)

The transmit wait cancel register (TXCR) is a 16-bit readable/writable register that controls cancellation of transmit wait messages in mailboxes (buffers).

TXCR

Bit:	15	14	13	12	11	10	9	8
	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	7	6	5	4	3	2	1	0
	TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

(x = 15 to 9, 7 to 0)

Bits 15 to 9 and 7 to 0—Transmit Wait Cancel Register (TXCR7 to TXCR1, TXCR15 to TXCR8): These bits control cancellation of transmit wait messages in the corresponding HCAN mailboxes.

Bit x: TXCRx	Description
0	Transmit message cancellation idle state in corresponding mailbox (Initial value) [Clearing condition] Completion of TXPR clearing (when transmit message is canceled normally)
1	TXPR cleared for corresponding mailbox (transmit message cancellation)

(x = 15 to 9, 7 to 0)

Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

16.2.7 Transmit Acknowledge Register (TXACK)

The transmit acknowledge register (TXACK) is a 16-bit readable/writable register containing status flags that indicate normal transmission of mailbox (buffer) transmit messages.

TXACK

Bit:	15	14	13	12	11	10	9	8
	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R

Bit:	7	6	5	4	3	2	1	0
	TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only a write of 1 is permitted, to clear the flag.

Bits 15 to 9 and 7 to 0—Transmit Acknowledge Register (TXACK7 to TXACK1, TXACK15 to TXACK8): These bits indicate that a transmit message in the corresponding HCAN mailbox has been transmitted normally.

Bit x: TXACKx	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Completion of message transmission for corresponding mailbox

(x = 15 to 9, 7 to 0)

Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

16.2.8 Abort Acknowledge Register (ABACK)

The abort acknowledge register (ABACK) is a 16-bit readable/writable register containing status flags that indicate normal cancellation (aborting) of a mailbox (buffer) transmit messages.

ABACK

Bit:	15	14	13	12	11	10	9	8
	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R

Bit:	7	6	5	4	3	2	1	0
	ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only a write of 1 is permitted, to clear the flag.

Bits 15 to 9 and 7 to 0—Abort Acknowledge Register (ABACK7 to ABACK1, ABACK15 to ABACK8): These bits indicate that a transmit message in the corresponding mailbox has been canceled (aborted) normally.

Bit x: ABACKx	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Completion of transmit message cancellation for corresponding mailbox

(x = 15 to 9, 7 to 0)

Bit 8—Reserved: This bit always reads 0. The write value should always be 0.

16.2.9 Receive Complete Register (RXPR)

The receive complete register (RXPR) is a 16-bit readable/writable register containing status flags that indicate normal reception of messages (data frame or remote frame) in mailboxes (buffers).

In the case of remote frame reception, the corresponding remote request register (RFPR) is also set simultaneously.

RXPR

Bit:	15	14	13	12	11	10	9	8
	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	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
	RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only a write of 1 is permitted, to clear the flag.

Bits 15 to 0—Receive Complete Register (RXPR7 to RXPR0, RXPR15 to RXPR8): These bits indicate that a receive message has been received normally in the corresponding mailbox.

Bit x: RXPRx	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Completion of message (data frame or remote frame) reception in corresponding mailbox

(x = 15 to 0)

16.2.10 Remote Request Register (RFPR)

The remote request register (RFPR) is a 16-bit readable/writable register containing status flags that indicate normal reception of remote frames in mailboxes (buffers). When a bit in this register is set, the corresponding reception complete bit is set simultaneously.

RFPR

Bit:	15	14	13	12	11	10	9	8
	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	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
	RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * Only a write of 1 is permitted, to clear the flag.

Bits 15 to 0—Remote Request Register (RFPR7 to RFPR0, RFPR15 to RFPR8): These bits indicate that a remote frame has been received normally in the corresponding mailbox.

Bit x: RFPRx	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Completion of remote frame reception in corresponding mailbox

(x = 15 to 0)

16.2.11 Interrupt Register (IRR)

The interrupt register (IRR) is a 16-bit readable/writable register containing status flags for the various interrupt sources.

IRR

Bit:	15	14	13	12	11	10	9	8
	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0
Initial value:	0	0	0	0	0	0	0	1
R/W:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R	R	R/(W)*

Bit:	7	6	5	4	3	2	1	0
	—	—	—	IRR12	—	—	IRR9	IRR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	—	—	—	R/(W)*	—	—	R	R/(W)*

Note: * Only a write of 1 is permitted, to clear the flag.

Bit 15—Overload Frame/Bus Off Recovery Interrupt Flag (IRR7): Status flag indicating that the HCAN has transmitted an overload frame or recovered from the bus off state.

Bit 15: IRR7	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Overload frame transmission [Setting conditions] • Error active/passive state When overload frame is transmitted

Bit 14—Bus Off Interrupt Flag (IRR6): Status flag indicating the bus off state caused by the transmit error counter.

Bit 14: IRR6	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Bus off state caused by transmit error [Setting condition] When TEC ≥ 256

Bit 13—Error Passive Interrupt Flag (IRR5): Status flag indicating the error passive state caused by the transmit/receive error counter.

Bit 13: IRR5	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error passive state caused by transmit/receive error [Setting condition] When TEC ≥ 128 or REC ≥ 128

Bit 12—Receive Overload Warning Interrupt Flag (IRR4): Status flag indicating the error warning state caused by the receive error counter.

Bit 12: IRR4	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by receive error [Setting condition] When REC ≥ 96

Bit 11—Transmit Overload Warning Interrupt Flag (IRR3): Status flag indicating the error warning state caused by the transmit error counter.

Bit 11: IRR3	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Error warning state caused by transmit error [Setting condition] When TEC ≥ 96

Bit 10—Remote Frame Request Interrupt Flag (IRR2): Status flag indicating that a remote frame has been received in a mailbox (buffer).

Bit 10: IRR2	Description
0	[Clearing condition] Clearing of all bits in RFPR (remote request register) of mailbox for which receive interrupt requests are enabled by MBIMR (Initial value)
1	Remote frame received and stored in mailbox [Setting conditions] When remote frame reception is completed, when corresponding MBIMR = 0

Bit 9—Receive Message Interrupt Flag (IRR1): Status flag indicating that a mailbox (buffer) receive message has been received normally.

Bit 9: IRR1	Description
0	[Clearing condition] Clearing of all bits in RXPR (receive complete register) of mailbox for which receive interrupt requests are enabled by MBIMR (Initial value)
1	Data frame or remote frame received and stored in mailbox [Setting conditions] When data frame or remote frame reception is completed, when corresponding MBIMR = 0

Bit 8—Reset Interrupt Flag (IRR0): Status flag indicating that the HCAN module has been reset. This bit cannot be masked in the interrupt mask register (IMR). If this bit is not cleared after reset input or recovery from software standby mode, interrupt handling will be performed as soon as interrupts are enabled by the interrupt controller.

Bit 8: IRR0	Description
0	[Clearing condition] Writing 1
1	Hardware reset (HCAN module stop*, software standby) (Initial value) [Setting condition] When reset processing is completed after a hardware reset (HCAN module stop*, software standby)

Note: * After reset or hardware standby release, the module stop bit is initialized to 1, and so the HCAN enters the module stop state.

Bits 7 to 5, 3, and 2—Reserved: These bits always read 0. The write value should always be 0.

Bit 4—Bus Operation Interrupt Flag (IRR12): Status flag indicating detection of a dominant bit due to bus operation when the HCAN module is in HCAN sleep mode.

Bit 4: IRR12	Description
0	CAN bus idle state (Initial value) [Clearing condition] Writing 1
1	CAN bus operation in HCAN sleep mode [Setting condition] Bus operation (dominant bit detection) in HCAN sleep mode

Bit 1—Unread Interrupt Flag (IRR9): Status flag indicating that a receive message has been overwritten while still unread.

Bit 1: IRR9	Description
0	[Clearing condition] Clearing of all bits in UMSR (unread message status register) (Initial value)
1	Unread message overwrite [Setting condition] When UMSR (unread message status register) is set

Bit 0—Mailbox Empty Interrupt Flag (IRR8): Status flag indicating that the next transmit message can be stored in the mailbox.

Bit 0: IRR8	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Transmit message has been transmitted or aborted, and new message can be stored [Setting condition] When TXPR (transmit wait register) is cleared by completion of transmission or completion of transmission abort

16.2.12 Mailbox Interrupt Mask Register (MBIMR)

The mailbox interrupt mask register (MBIMR) is a 16-bit readable/writable register containing flags that enable or disable individual mailbox (buffer) interrupt requests.

MBIMR

Bit:	15	14	13	12	11	10	9	8
	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0
Initial value:	1	1	1	1	1	1	1	1
R/W:	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
	MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0—Mailbox Interrupt Mask (MBIMRx): Flags that enable or disable individual mailbox interrupt requests.

Bit x: MBIMRx	Description	
0	[Transmitting] Interrupt request to CPU due to TXPR clearing [Receiving] Interrupt request to CPU due to RXPR setting	
1	Interrupt requests to CPU disabled	(Initial value)

(x = 15 to 0)

16.2.13 Interrupt Mask Register (IMR)

The interrupt mask register (IMR) is a 16-bit readable/writable register containing flags that enable or disable requests by individual interrupt sources.

IMR

Bit:	15	14	13	12	11	10	9	8
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	—
Initial value:	1	1	1	1	1	1	1	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit:	7	6	5	4	3	2	1	0
	—	—	—	IMR12	—	—	IMR9	IMR8
Initial value:	1	1	1	1	1	1	1	1
R/W:	R	R	R	R/W	R	R	R/W	R/W

Bit 15—Overload Frame/Bus Off Recovery Interrupt Mask (IMR7): Enables or disables overload frame/bus off recovery interrupt requests.

Bit 15: IMR7	Description
0	Overload frame/bus off recovery interrupt request (OVR0) to CPU by IRR7 enabled
1	Overload frame/bus off recovery interrupt request (OVR0) to CPU by IRR7 disabled (Initial value)

Bit 14—Bus Off Interrupt Mask (IMR6): Enables or disables bus off interrupt requests caused by the transmit error counter.

Bit 14: IMR6	Description
0	Bus off interrupt request (ERS0) to CPU by IRR6 enabled
1	Bus off interrupt request (ERS0) to CPU by IRR6 disabled (Initial value)

Bit 13—Error Passive Interrupt Mask (IMR5): Enables or disables error passive interrupt requests caused by the transmit/receive error counter.

Bit 13: IMR5	Description
0	Error passive interrupt request (ERS0) to CPU by IRR5 enabled
1	Error passive interrupt request (ERS0) to CPU by IRR5 disabled (Initial value)

Bit 12—Receive Overload Warning Interrupt Mask (IMR4): Enables or disables error warning interrupt requests caused by the receive error counter.

Bit 12: IMR4	Description
0	REC error warning interrupt request (OVR0) to CPU by IRR4 enabled
1	REC error warning interrupt request (OVR0) to CPU by IRR4 disabled (Initial value)

Bit 11—Transmit Overload Warning Interrupt Mask (IMR3): Enables or disables error warning interrupt requests caused by the transmit error counter.

Bit 11: IMR3	Description
0	TEC error warning interrupt request (OVR0) to CPU by IRR3 enabled
1	TEC error warning interrupt request (OVR0) to CPU by IRR3 disabled (Initial value)

Bit 10—Remote Frame Request Interrupt Mask (IMR2): Enables or disables remote frame reception interrupt requests.

Bit 10: IMR2	Description
0	Remote frame reception interrupt request (OVR0) to CPU by IRR2 enabled
1	Remote frame reception interrupt request (OVR0) to CPU by IRR2 disabled (Initial value)

Bit 9—Receive Message Interrupt Mask (IMR1): Enables or disables message reception interrupt requests.

Bit 9: IMR1	Description
0	Message reception interrupt request (RM1) to CPU by IRR1 enabled
1	Message reception interrupt request (RM1) to CPU by IRR1 disabled (Initial value)

Bit 8—Reserved: The reset flag cannot be masked. This bit always reads 0. The write value should always be 0.

Bits 7 to 5, 3, and 2—Reserved: These bits always read 1. The write value should always be 1.

Bit 4—Bus Operation Interrupt Mask (IMR12): Enables or disables interrupt requests due to bus operation in sleep mode.

Bit 4: IMR12	Description
0	Bus operation interrupt request (OVR0) to CPU by IRR12 enabled
1	Bus operation interrupt request (OVR0) to CPU by IRR12 disabled (Initial value)

Bit 1—Unread Interrupt Mask (IMR9): Enables or disables unread receive message overwrite interrupt requests.

Bit 1: IMR9	Description
0	Unread message overwrite interrupt request (OVR0) to CPU by IRR9 enabled
1	Unread message overwrite interrupt request (OVR0) to CPU by IRR9 disabled (Initial value)

Bit 0—Mailbox Empty Interrupt Mask (IMR8): Enables or disables mailbox empty interrupt requests.

Bit 0: IMR8	Description
0	Mailbox empty interrupt request (SLE0) to CPU by IRR8 enabled
1	Mailbox empty interrupt request (SLE0) to CPU by IRR8 disabled (Initial value)

16.2.14 Receive Error Counter (REC)

The receive error counter (REC) is an 8-bit read-only register that functions as a counter indicating the number of receive message errors on the CAN bus. The count value is stipulated in the CAN protocol.

REC

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

16.2.15 Transmit Error Counter (TEC)

The transmit error counter (TEC) is an 8-bit read-only register that functions as a counter indicating the number of transmit message errors on the CAN bus. The count value is stipulated in the CAN protocol.

TEC

Bit:	7	6	5	4	3	2	1	0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R	R	R	R	R	R

16.2.16 Unread Message Status Register (UMSR)

The unread message status register (UMSR) is a 16-bit readable/writable register containing status flags that indicate, for individual mailboxes (buffers), that a received message has been overwritten by a new receive message before being read. When a message is overwritten by a new receive message, the old data is lost.

UMSR

Bit:	15	14	13	12	11	10	9	8
	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0
Initial value:	0	0	0	0	0	0	0	0
R/W:	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
	UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bits 15 to 0—Unread Message Status Flags (UMSRx): Status flags indicating that an unread receive message has been overwritten.

Bit x: UMSRx	Description
0	[Clearing condition] Writing 1 (Initial value)
1	Unread receive message is overwritten by a new message [Setting condition] When a new message is received before RXPR is cleared

16.2.17 Local Acceptance Filter Masks (LAFML, LAFMH)

The local acceptance filter masks (LAFML, LAFMH) are 16-bit readable/writable registers that filter receive messages to be stored in the receive-only mailbox (MC0, MD0) according to the identifier. In these registers, consist of LAFMH15: MSB to LAFMH5: LSB are 11 standard/extended identifier bits, and LAFMH1: MSB to LAFML0: LSB are 18 extended identifier bits.

LAFML

Bit:	15	14	13	12	11	10	9	8
	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0
Initial value:	0	0	0	0	0	0	0	0
R/W:	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
	LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LAFMH

Bit:	15	14	13	12	11	10	9	8
	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1	LAFMH0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R	R	R	R/W	R/W

Bit:	7	6	5	4	3	2	1	0
	LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9	LAFMH8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LAFMH Bits 7 to 0 and 15 to 13–11-Bit Identifier Filter (LAFMH7 to LAFMH5, LAFMH15 to LAFMH8): Filter mask bits for the first 11 bits of the receive message identifier (for both standard and extended identifiers).

Bit x: LAFMHx	Description
0	Stored in MC0 and MD0 (receive-only mailbox) depending on bit match between MC0 message identifier and receive message identifier (Initial value)
1	Stored in MC0 and MD0 (receive-only mailbox) regardless of bit match between MC0 message identifier and receive message identifier

(x = 15 to 0)

LAFMH Bits 12 to 10—Reserved: These bits always read 0. The write value should always be 0.

LAFMH Bits 9 and 8, LAFML bits 15 to 0–18-Bit Identifier Filter (LAFMH1, LAFMH0, LAFML7 to LAFML0, LAFML15 to LAFML8): Filter mask bits for the 18 bits of the receive message identifier (extended).

Bit x: LAFMHx LAFMLx	Description
0	Stored in MC0 (receive-only mailbox) depending on bit match between MC0 message identifier and receive message identifier (Initial value)
1	Stored in MC0 (receive-only mailbox) regardless of bit match between MC0 message identifier and receive message identifier

(x = 15 to 0)

16.2.18 Message Control (MC0 to MC15)

The message control register sets (MC0 to MC15) consist of eight 8-bit readable/writable registers (MCx[1] to MCx[8]). The HCAN has 16 sets of these registers (MC0 to MC15).

The initial value of these registers is undefined, so they must be initialized (by writing 0 or 1).

MCx [1]

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [2]

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [3]

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [4]

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [5]

Bit:	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*:Undefined

MCx [6]

Bit:	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [7]

Bit:	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MCx [8]

Bit:	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*:Undefined
(x = 15 to 0)

MCx[1] Bits 7 to 4—Reserved: The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).

MCx[1] Bits 3 to 0—Data Length Code (DLC): These bits indicate the required length of data frames and remote frames.

Bit 3: DLC3	Bit 2: DLC2	Bit 1: DLC1	Bit 0: DLC0	Description
0	0	0	0	Data length = 0 bytes
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MCx[2] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).

MCx[3] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).

MCx[4] Bits 7 to 0—Reserved: The initial value of these bits is undefined; they must be initialized (by writing 0 or 1).

MCx[6] Bits 7 to 0: Standard Identifier (STD_ID10 to STD_ID3)

MCx[5] Bits 7 to 5: Standard Identifier (STD_ID2 to STD_ID0)

These bits set the identifier (standard identifier) of data frames and remote frames.

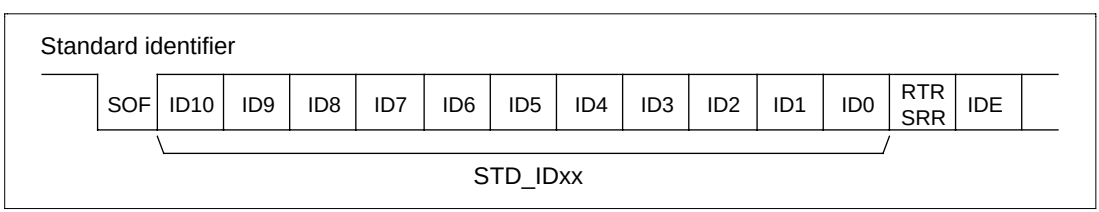


Figure 16-2 Standard identifier

MCx[5] Bit 4—Remote Transmission Request (RTR): Used to distinguish between data frames and remote frames.

Bit 4: RTR	Description
0	Data frame
1	Remote frame

MCx[5] Bit 3—Identifier Extension (IDE): Used to distinguish between the standard format and extended format of data frames and remote frames.

Bit 3: IDE	Description
0	Standard format
1	Extended format

MCx[5] Bit 2—Reserved: The initial value of this bit is undefined; it must be initialized (by writing 0 or 1).

MCx[5] Bits 1 and 0: Extended Identifier (EXD_ID17, EXD_ID16)

MCx[8] Bits 7 to 0: Extended Identifier (EXD_ID15 to EXD_ID8)

MCx[7] Bits 7 to 0: Extended Identifier (EXD_ID7 to EXD_ID0)

These bits set the identifier (extended identifier) of data frames and remote frames.

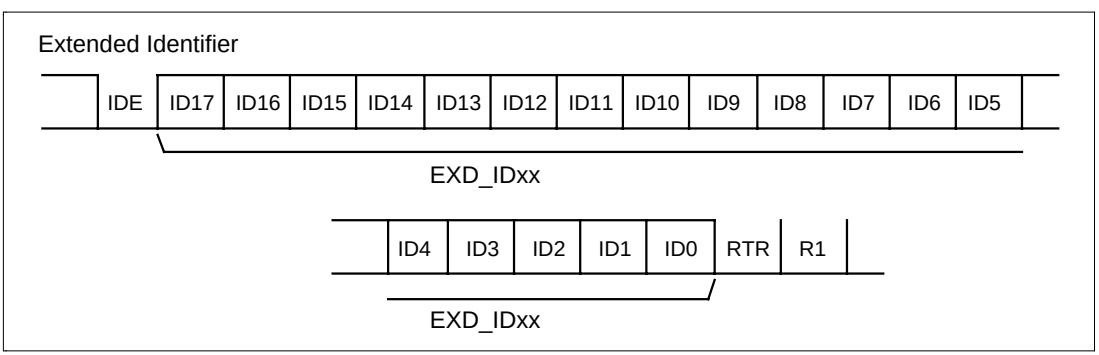


Figure 16-3 Extended identifier

16.2.19 Message Data (MD0 to MD15)

The message data register sets (MD0 to MD15) consist of eight 8-bit readable/writable registers (MDx[1] to MDx[8]). The HCAN has 16 sets of these registers (MD0 to MD15).

The initial value of these registers is undefined, so they must be initialized (by writing 0 or 1).

MDx [1]

Bit:	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [2]

Bit:	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [3]

Bit:	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [4]

Bit:	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [5]

Bit:	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [6]

Bit:	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [7]

Bit:	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx [8]

Bit:	7	6	5	4	3	2	1	0
Initial value:	*	*	*	*	*	*	*	*
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*:Undefined
(x = 0 to 15)

16.2.20 Module Stop Control Register C (MSTPCRC)

Bit:	7	6	5	4	3	2	1	0
	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value:	1	1	1	1	1	1	1	1
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPC3 and MSTPC2 bits are set to 1, HCAN0 and 1 operation is stopped at the end of the bus cycle, and module stop mode is entered. Register read/write accesses are not possible in module stop mode. For details, see section 23A.5, 23B.5, Module Stop Mode.

MSTPCRC is initialized to H'FF by a reset, and in hardware standby mode. It is not initialized in software standby mode.

Bit 3—Module Stop (MSTPC3): Specifies the HCAN module stop mode.

Bit 3: MSTPC3	Description	
0	HCAN0 module stop mode is cleared	
1	HCAN0 module stop mode is set	(Initial value)

Bit 2—Module Stop (MSTPC2): Specifies the HCAN module stop mode.

Bit 2: MSTPC2	Description	
0	HCAN1 module stop mode is cleared	
1	HCAN1 module stop mode is set	(Initial value)

16.3 Operation

This LSI device is equipped with 2-channel HCAN modules, which are controlled independently. Both modules have identical specifications, and they are controlled in the same manner.

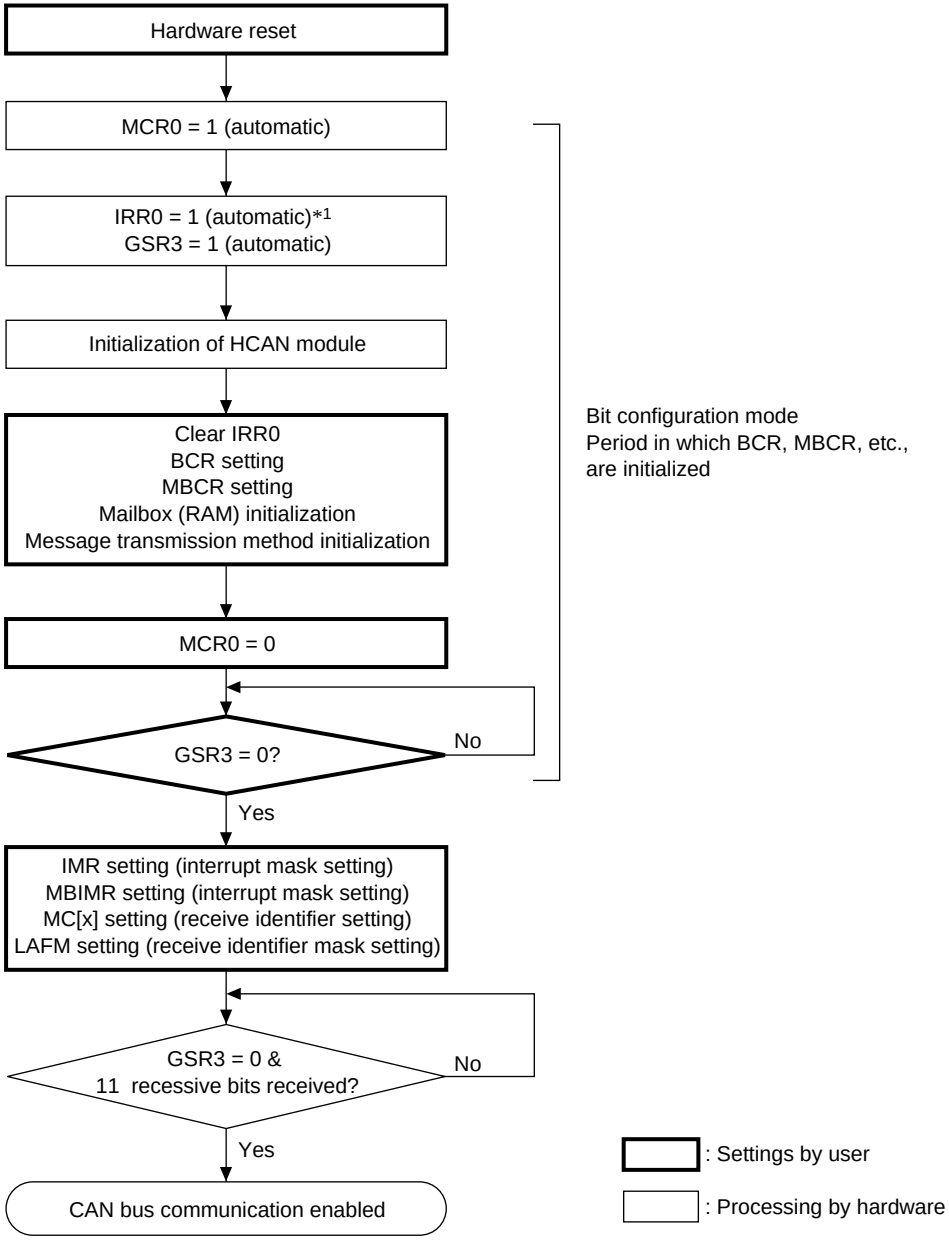
16.3.1 Hardware and Software Resets

The HCAN can be reset by a hardware reset or software reset.

Hardware Reset (HCAN Module Stop, Reset*, Hardware*/Software Standby): Initialization is performed by automatic setting of the MCR reset request bit (MCR0) in MCR and the reset state bit (GSR3) in GSR within the HCAN (hardware reset). At the same time, all internal registers are initialized. However mailbox contents are retained. A flowchart of this reset is shown in figure 16-4.

Note: * In a reset and in hardware standby mode, the module stop bit is initialized to 1 and the HCAN enters the module stop state.

Software Reset (Write to MCR0): In normal operation initialization is performed by setting the MCR reset request bit (MCR0) in MCR (Software reset). With this kind of reset, if the CAN controller is performing a communication operation (transmission or reception), the initialization state is not entered until the message has been completed. During initialization, the reset state bit (GSR3) in GSR is set. In this kind of initialization, the error counters (TEC and REC) are initialized but other registers and RAM (mailboxes) are not. A flowchart of this reset is shown in figure 16-5.



Notes: *1 When IRR0 is set to 1 (automatically) due to a hardware reset*2, a "hardware reset initiated reset processing" interrupt is generated.

*2 In a reset and in hardware standby mode, the module stop bit is initialized to 1 and the HCAN enters the module stop state.

Figure 16-4 Hardware Reset Flowchart

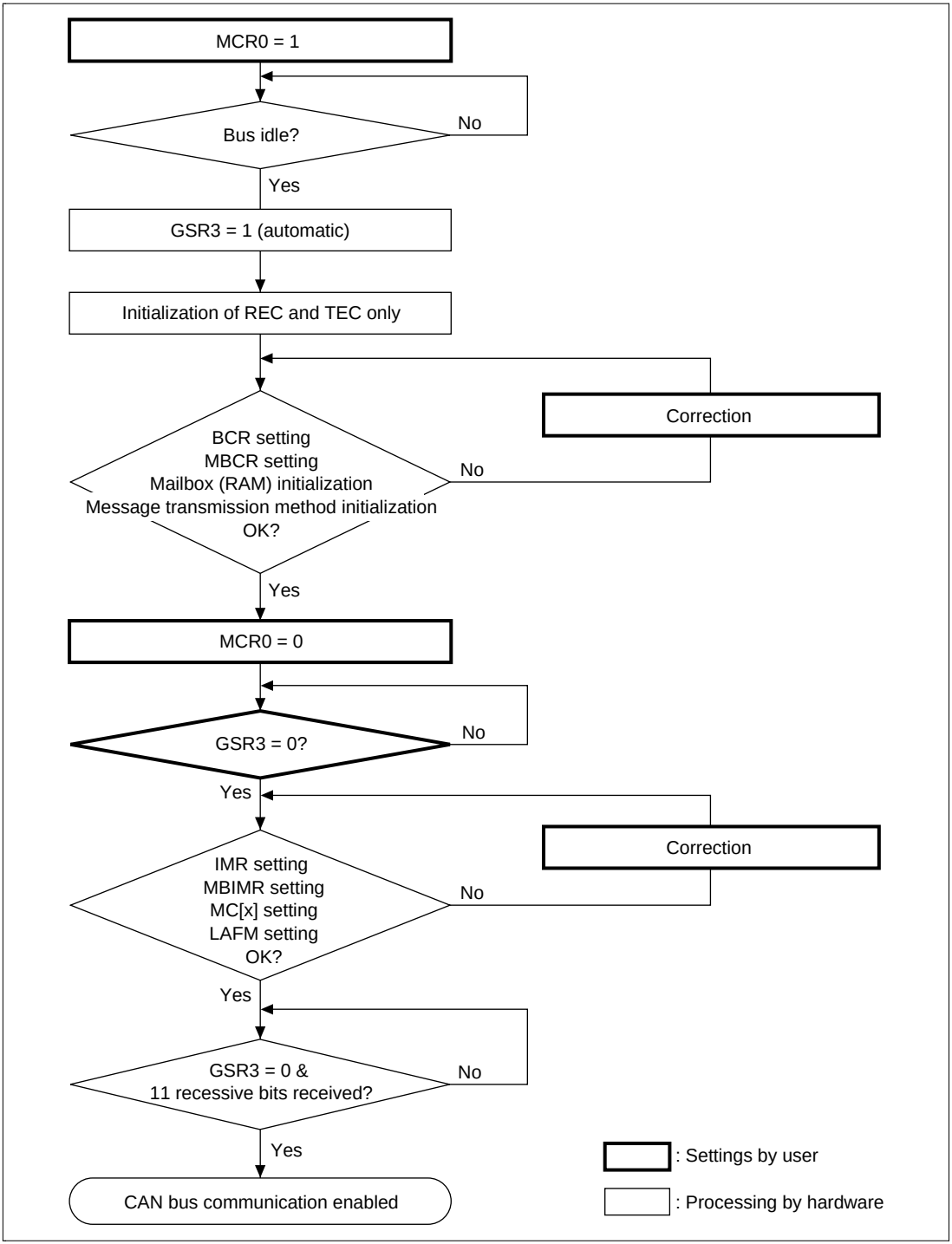


Figure 16-5 Software Reset Flowchart

16.3.2 Initialization after Hardware Reset

After a hardware reset, the following initialization processing should be carried out:

- Clearing of IRR0 bit in interrupt register (IRR)
- Bit rate setting
- Mailbox transmit/receive settings
- Mailbox (RAM) initialization
- Message transmission method setting

These initial settings must be made while the HCAN is in bit configuration mode. Configuration mode is a state in which the reset request bit (MCR0) in the master control register (MCR) is 1 and the reset status bit in the general status register (GSR) is also 1 (GSR3 = 1). Configuration mode is exited by clearing the reset request bit in MCR to 0; when MCR0 is cleared to 0, the HCAN automatically clears the reset state bit (GSR3) in the general status register (GSR). The power-up sequence then begins, and communication with the CAN bus is possible as soon as the sequence ends. The power-up sequence consists of the detection of 11 consecutive recessive bits.

IRR0 Clearing: The reset interrupt flag (IRR0) is always set after a reset or recovery from software standby mode. As an HCAN interrupt is initiated immediately when interrupts are enabled, IRR0 should be cleared.

Bit Rate Settings: As bit rate settings, a baud rate setting and bit timing setting must be made each time a CAN node begins communication. The baud rate and bit timing settings are made in the bit configuration register (BCR).

a. Note

   BCR can be written to at all times, but should only be modified in configuration mode.

   Settings should be made so that all CAN controllers connected to the CAN bus have the same baud rate and bit width.

   Limits for the settable variables (TSEG1, TSEG2, BRP, sample point, and SJW) are shown in table 16-4.

Table 16-4 BCR Setting Limits

Name	Abbreviation	Bits	Initial Value	Min. Value	Max. Value
Time segment 1	TSEG1	4	0	3	15
Time segment 2	TSEG2	3	0	1	7
Baud rate prescaler	BRP	6	0	0	63
Sample point	SAM	1	0	0	1
Synchronization jump width	SJW	2	0	1	3

b. Settable Variable Limits

- The bit width consists of the total of the settable time quanta (TQ). TQ (number of system clocks) is determined by the baud rate prescaler (BRP).

$$TQ = \frac{2 \times (BRP + 1)}{f_{CLK}}$$

- The minimum value of SJW is stipulated in the CAN specifications.

$$4 \geq SJW \geq 1$$

- The minimum value of TSEG1 is stipulated in the CAN specifications.

$$TSEG1 > 2 + TSEG2$$

- The minimum value of TSEG2 is stipulated in the CAN specifications.

$$TSEG2 \geq (1 + SJW)$$

The following formula is used to calculate the baud rate.

$$\text{Bit rate [b/s]} = \frac{f_{CLK}}{2 \times (BRP + 1) \times (3 + TSEG1 + TSEG2)}$$

Note: $f_{CLK} = \phi$ (system clock)
The BCR value are used for BRP, TSEG1, and TSEG2.

Example: With a 1 Mb/s baud rate and a 20 MHz input clock:

$$1 \text{ Mb/s} = \frac{20 \text{ MHz}}{2 \times (0 + 1) \times (3 + 4 + 3)}$$

Item	Set Values	Actual Values
f_{CLK}	20 MHz	—
BRP	0 (B'000000)	System clock × 2
TSEG1	4 (B'0100)	5TQ
TSEG2	3 (B'011)	4TQ

Mailbox Transmit/Receive Settings: HCAN0, 1 each have 16 mailboxes. Mailbox 0 is receive-only, while mailboxes 1 to 15 can be set for transmission or reception. Mailboxes that can be set for transmission or reception must be designated either for transmission use or for reception use before communication begins. The Initial status of mailboxes 1 to 15 is for transmission (while mailbox 0 is for reception only). Mailbox transmit/receive settings are not initialized by a software reset.

- Setting for transmission

  Transmit mailbox setting (mailboxes 1 to 15)

  Clearing a corresponding mailbox in the mailbox configuration register (MBCR) to 0 designates the specified mailbox for transmission use. After a reset, mailboxes are initialized for transmission use, so this setting is not necessary.

- Setting for reception

  Transmit/receive mailbox setting (mailboxes 1 to 15)

  Setting a bit to 1 in the mailbox configuration register (MBCR) designates the corresponding mailbox for reception use. When setting mailboxes for reception, to improve message transmission efficiency, high-priority messages should be set in low-to-high mailbox order (priority order: mailbox 1 > mailbox 15).

- Receive-only mailbox (mailbox 0)

  No setting is necessary, as this mailbox is always used for reception.

Mailbox (Message Control/Data (MCx[x], MDx[x]) Initial Settings: After power is supplied, all registers and RAM (message control/data, control registers, status registers, etc.) are initialized. Message control/data (MCx[x], MDx[x]) only are in RAM, and so their values are undefined. Initial values must therefore be set in all the mailboxes (by writing 0s or 1s).

Setting the Message Transmission Method: Either of the following message transmission methods can be selected with the message transmission method bit (MCR2) in the master control register (MCR):

a. Transmission order determined by message identifier priority
b. Transmission order determined by mailbox number priority

When a is selected, if a number of messages are designated as waiting for transmission (TXPR = 1), the message with the highest priority set in the message identifier (MCx[5]–MCx[8]) is stored in the transmit buffer. CAN bus arbitration is then carried out for the message in the transmit buffer, and message transmission is performed when the transmission right is acquired. When the TXPR bit is set, internal arbitration is performed again, and the highest-priority message is found and stored in the transmit buffer.

When b is selected, if a number of messages are designated as waiting for transmission (TXPR = 1), messages are stored in the transmit buffer in low-to-high mailbox order (priority order: mailbox 1 > mailbox 15). CAN bus arbitration is then carried out for the messages in the transmit buffer, and message transmission is performed when the bus is acquired.

16.3.3 Transmit Mode

Message transmission is performed using mailboxes 1 to 15. The transmission procedure is described below, and a transmission flowchart is shown in figure 16-6.

Initialization (after hardware reset only)

a. Clearing of IRR0 bit in interrupt register (IRR)
b. Bit rate settings
c. Mailbox transmit/receive settings
d. Mailbox (RAM) initialization
e. Message transmission method setting

Interrupt and transmit data settings

a. CPU interrupt source setting
b. Arbitration field setting
c. Control field setting
d. Data field setting

Message transmission and interrupts

a. Message transmission wait
b. Message transmission completion and interrupt
c. Message transmission cancellation
d. Message retransmission

Initialization (After Hardware Reset Only): These settings should be made while the HCAN is in bit configuration mode.

- IRR0 clearing

  The reset interrupt flag (IRR0) is always set after a reset or recovery from software standby mode. As an HCAN interrupt is initiated immediately when interrupts are enabled, IRR0 should be cleared.

- Bit rate settings

  Set values relating to the CAN bus communication speed and resynchronization. Refer to Bit Rate Settings in 16.3.2 Initialization after Hardware Reset for details.

- Mailbox transmit/receive settings

  Mailbox transmit/receive settings should be made in advance. A total of 30 mailbox can be set for transmission or reception (mailboxes 1 to 15 in HCAN0 and HCAN1). To set a mailbox for transmission, clear the corresponding bit to 0 in the mailbox configuration register (MBCR). Refer to Mailbox Transmit/Receive Settings in 16.3.2 Initialization after Hardware Reset for details.

- Mailbox (RAM) initialization

  As message control/data registers (MCx[x], MDx[x]) are configured in RAM, their initial values after powering on are undefined, and so bit initialization is necessary. Write 0s or 1s to the mailboxes. See Mailbox (Message Control/Data (MCx[x], MDx[x]) Initial Setting in 16.3.2, Initialization after a Hardware Reset, for details.

- Message transmission method setting

  Set the transmission method for mailboxes designated for transmission. The following two transmission methods can be used. Refer to Message Transmission Method Setting in 16.3.2 Initialization after Hardware Reset for details.

  a. Transmission order determined by message identifier priority
  b. Transmission order determined by mailbox number priority

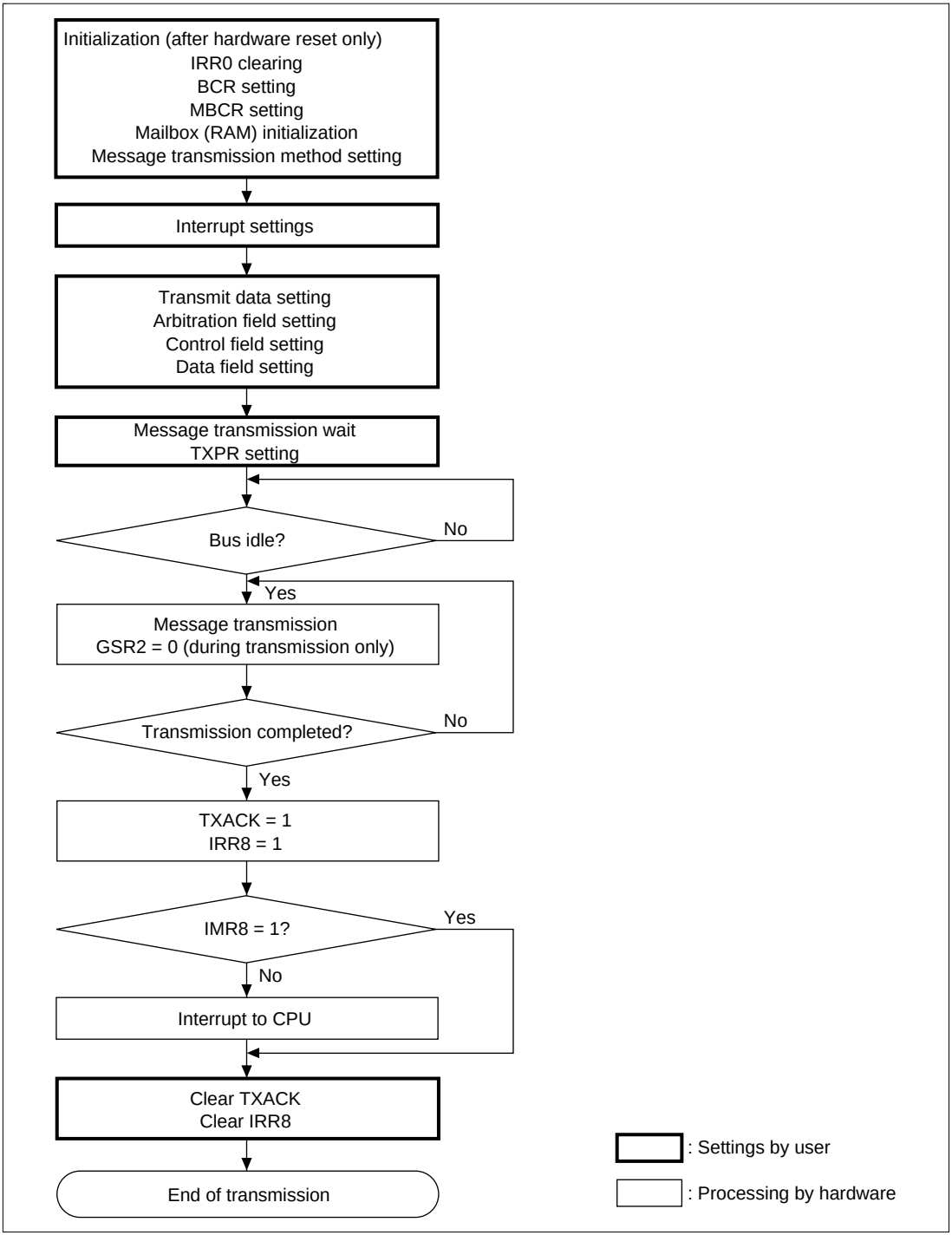


Figure 16-6 Transmission Flowchart

Interrupt and Transmit Data Settings: When mailbox initialization is finished, CPU interrupt source settings and data settings must be made. Interrupt source settings are made in the mailbox interrupt register (MBIMR) and interrupt mask register (IMR), while transmit data settings are made by writing the necessary data from the arbitration field, control field, and data field, described below, in the corresponding message control (MCx[1]—[8]) and message data (MDx[1]—[8]).

- CPU interrupt source setting

  Transmission acknowledge and transmission abort acknowledge interrupts can be masked for individual mailboxes in the mailbox interrupt mask register (MBIMR). Interrupt register (IRR) interrupts can be masked in the interrupt mask register (IMR).

- Arbitration field setting

  In the arbitration field, the 11-bit identifier (STD_ID0–STD_ID10) and RTR bit (standard format) or 29-bit identifier (STD_ID0–STD_ID10, EXT_ID0–EXT_ID17) and IDE.RTR bit (extended format) are set. The registers to be set are MCx[5]–MCx[8].

- Control field setting

  In the control field, the byte length of the data to be transmitted is set in DLC0–DLC3. The register to be set is MCx[1].

- Data field setting

  In the data field, the data to be transmitted is set in byte units in the range of 0 to 8 bytes. The registers to be set are MDx[1]–MDx[8].

The number of bytes in the data actually transmitted depends on the data length code (DLC) in the control field. If a value exceeding the value set in DLC is set in the data field, only the number of bytes set in DLC will actually be transmitted.

Message Transmission and Interrupts:

- Message transmission wait

  If message transmission is to be performed after completion of the message control (MCx[1]–MCx[8]) and message data (MDx[1]–MDx[8]).settings, transmission is started by setting the corresponding mailbox transmit wait bit (TXPR1–TXPR15) to 1 in the transmit wait register (TXPR). The following two transmission methods can be used:

  a. Transmission order determined by message identifier priority
  b. Transmission order determined by mailbox number priority

  When a is selected, if a number of messages are designated as waiting for transmission (TXPR = 1), messages are stored in the transmit buffer in low-to-high mailbox order (priority order: mailbox 1 > mailbox 15). CAN bus arbitration is then carried out for the messages in the transmit buffer, and message transmission is performed when the bus is acquired.

When b is selected, if a number of messages are designated as waiting for transmission (TXPR = 1), the message with the highest priority set in the message identifier (MCx[5]–MCx[8]) is stored in the transmit buffer. CAN bus arbitration is then carried out for the message in the transmit buffer, and message transmission is performed when the transmission right is acquired. When the TXPR bit is set, internal arbitration is performed again, the highest-priority message is found and stored in the transmit buffer, CAN bus arbitration is carried out in the same way, and message transmission is performed when the transmission right is acquired.

- Message transmission completion and interrupt

  When a message is transmitted error-free using the above procedure, the corresponding acknowledge bit (TXACK1–TXACK15) in the transmit acknowledge register (TXACK) and transmit wait bit (TXPR1–TXPR15) in the transmit wait register (TXPR) are automatically initialized. Also, if the corresponding bit (MBIMR1-MBIMR15) in the mailbox interrupt mask register (MBIMR) and the mailbox empty interrupt bit (IRR8) in the interrupt mask register (IMR) are set to the interrupt enable state at the same time, an interrupt can be sent to the CPU.

- Message transmission cancellation

  Transmission cancellation can be specified for a message stored in a mailbox as a transmit wait message. A transmit wait message is canceled by setting the bit for the corresponding mailbox (TXCR1–TXCR15) to 1 in the transmit cancel register (TXCR). When cancellation is executed, the transmit wait register (TXPR) is automatically reset, and the corresponding bit is set to 1 in the abort acknowledge register (ABACK). An interrupt can be requested. Also, if the mailbox empty interrupt (IRR8) is enabled for the bits (MBIMR1-MBIMR15) corresponding to the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR), interrupts may be sent to the CPU.

  However, a transmit wait message cannot be canceled at the following times:

  a. During internal arbitration or CAN bus arbitration
  b. During data frame or remote frame transmission

  Also, transmission cannot be canceled by clearing the transmit wait register (TXPR). Figure 16-5 shows a flowchart of transmit message cancellation.

- Message retransmission

  If transmission of a transmit message is aborted in the following cases, the message is retransmitted automatically:

  a. CAN bus arbitration failure (failure to acquire the bus)
  b. Error during transmission (bit error, stuff error, CRC error, frame error, ACK error)

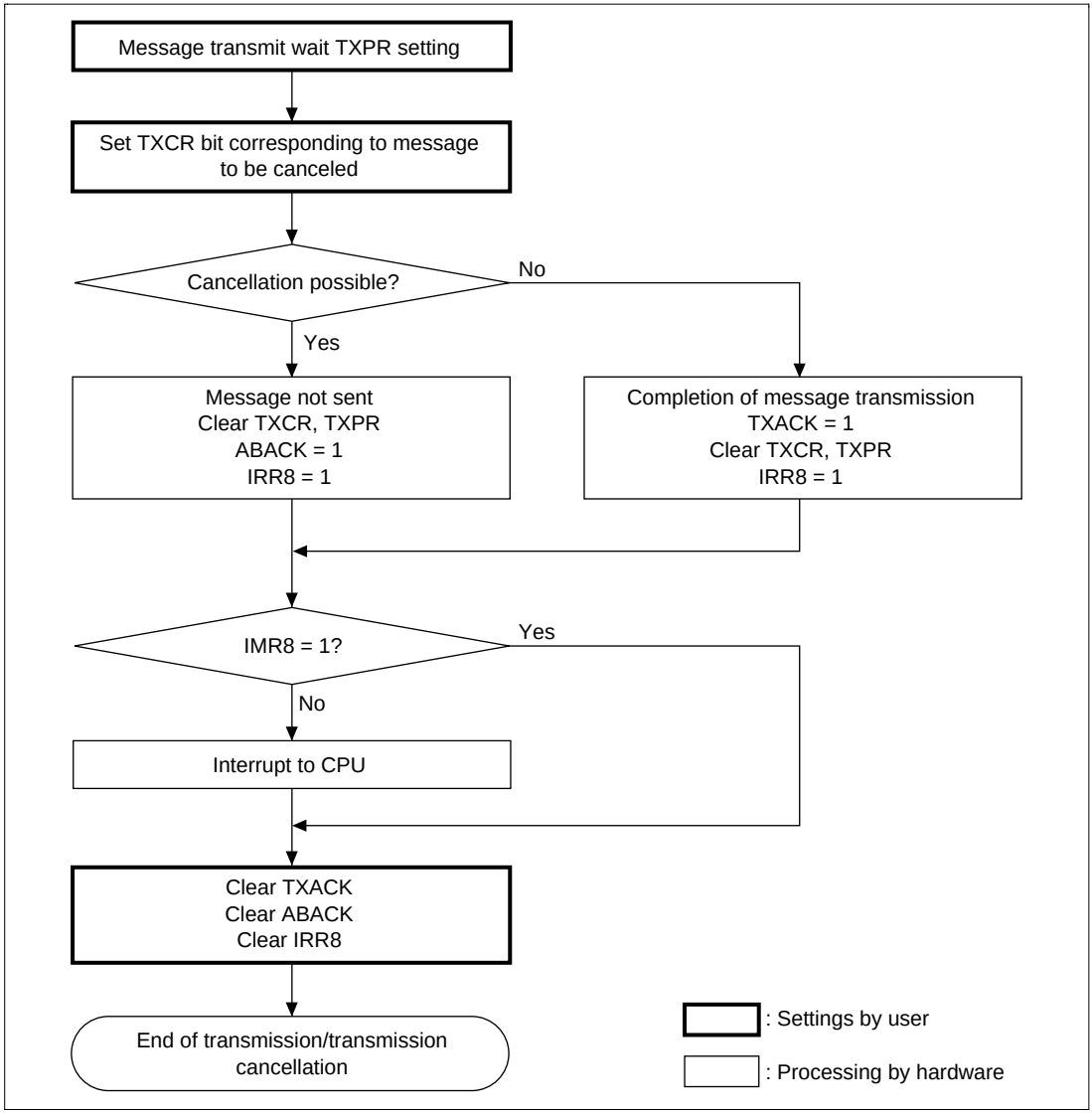


Figure 16-7 Transmit Message Cancellation Flowchart

16.3.4 Receive Mode

Message reception is performed using mailboxes 0 and 1 to 15. The reception procedure is described below, and a reception flowchart is shown in figure 16-8.

Initialization (after hardware reset only)

a. Clearing of IRR0 bit in interrupt register (IRR)
b. Bit rate settings
c. Mailbox transmit/receive settings
d. Mailbox (RAM) initialization

Interrupt and receive message settings

a. CPU interrupt source setting
b. Arbitration field setting
c. Local acceptance filter mask (LAFM) settings

Message reception and interrupts

a. Message reception CRC check
b. Data frame reception
c. Remote frame reception
d. Unread message reception

Initialization (After Hardware Reset Only): These settings should be made while the HCAN is in bit configuration mode.

- IRR0 clearing

  The reset interrupt flag (IRR0) is always set after a reset or recovery from software standby mode. As an HCAN interrupt is initiated immediately when interrupts are enabled, IRR0 should be cleared.

- Bit rate settings

  Set values relating to the CAN bus communication speed and resynchronization. Refer to Bit Rate Settings in 16.3.2 Initialization after Hardware Reset for details.

- Mailbox transmit/receive settings

  Each channel has one receive-only mailbox (mailbox 0) plus 15 mailboxes that can be set for reception. Thus a total of 32 mailboxes can be used for reception. To set a mailbox for reception, set the corresponding bit to 1 in the mailbox configuration register (MBCR). The initial setting for mailboxes is 0, designating transmission use. Refer to Mailbox Transmit/Receive Settings in 16.3.2 Initialization after Hardware Reset for details.

- Mailbox (RAM) initialization

  As message control/data registers (MCx[x], MDx[x]) are configured in RAM, their initial values after powering on are undefined, and so bit initialization is necessary. Write 0s or 1s to the mailboxes. See Mailbox (Message Control/Data (MCx[x], MDx[x]) Initial Setting in 16.3.2, Initialization after a Hardware Reset, for details.

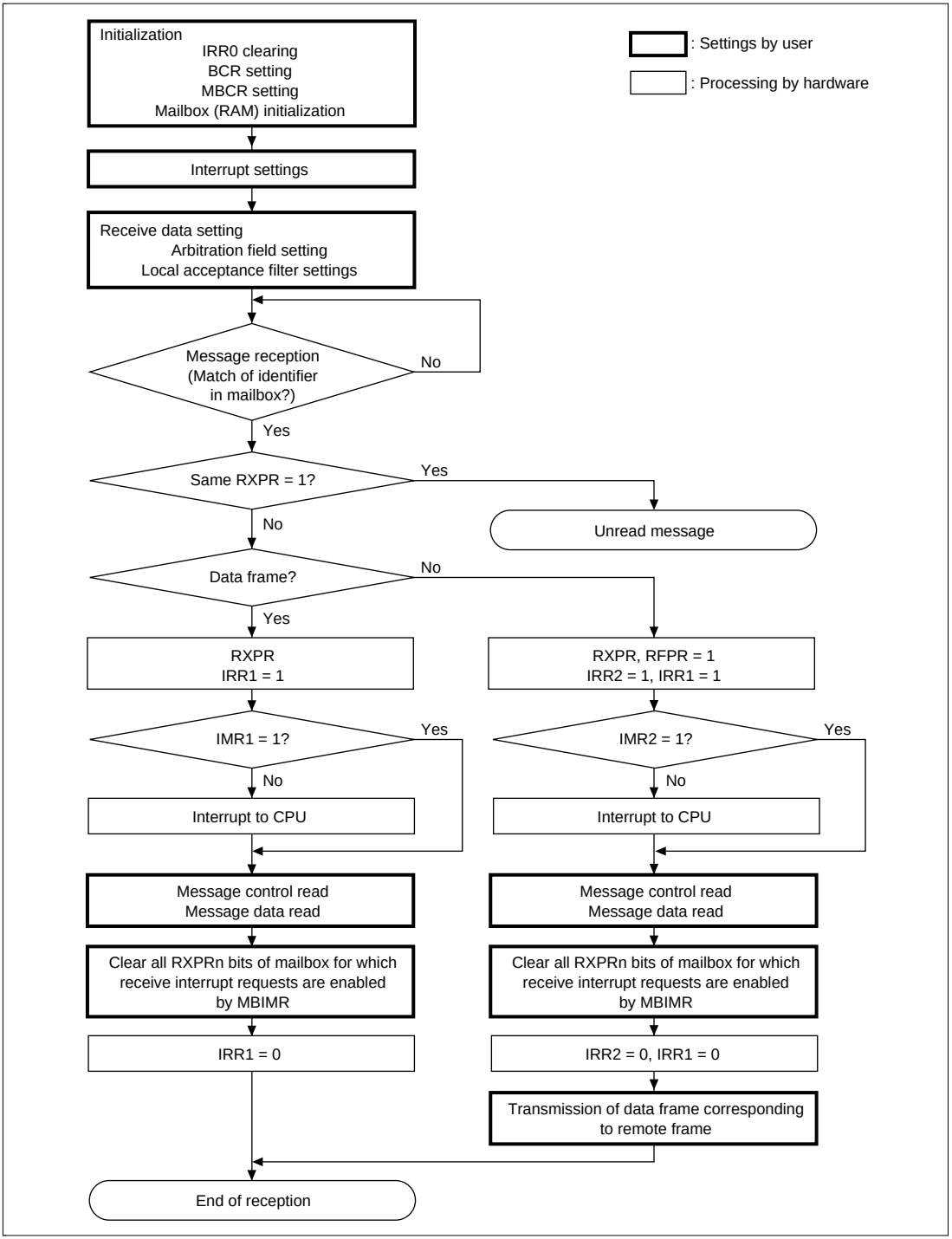


Figure 16-8 Reception Flowchart

Interrupt and Receive Message Settings: When mailbox initialization is finished, CPU interrupt source settings and receive message specifications must be made. Interrupt source settings are made in the mailbox interrupt register (MBIMR) and interrupt mask register (IMR). To receive a message, the identifier must be set in advance in the message control (MCx[1]–MCx[8]) for the receiving mailbox. When a message is received, all the bits in the receive message identifier are compared, and if a 100% match is found, the message is stored in the matching mailbox. Mailbox 0 (MB0) has a local acceptance filter mask (LAFM) that allows Don't Care settings to be made.

- CPU interrupt source settings

  When transmitting, transmission acknowledge and transmission abort acknowledge interrupts can be masked for individual mailboxes in the mailbox interrupt mask register (MBIMR). When receiving, data frame and remote frame receive wait interrupts can be masked. Interrupt register (IRR) interrupts can be masked in the interrupt mask register (IMR).

- Arbitration field setting

  In the arbitration field, the identifier (STD_ID0–STD_ID10, EXT_ID0–EXT_ID17) of the message to be received is set. If all the bits in the set identifier do not match, the message is not stored in a mailbox.

  Example: Mailbox 1 010_1010_1010 (standard identifier)

  Only one kind of message identifier can be received by MB1
  Identifier 1: 010_1010_1010

- Local acceptance filter mask (LAFM) setting

  The local acceptance filter mask is provided for mailbox 0 (MB0) only, enabling a Don't Care specification to be made for all bits in the received identifier. This allows various kinds of messages to be received.

  Example: Mailbox 0 010_1010_1010 (standard identifier)
  LAFM 000_0000_0011 (0: Care, 1: Don't Care)

  A total of four kinds of message identifiers can be received by MB0
  Identifier 1: 010_1010_1000
  Identifier 2: 010_1010_1001
  Identifier 3: 010_1010_1010
  Identifier 4: 010_1010_1011

Message Reception and Interrupts:

- Message reception CRC check

  When a message is received, a CRC check is performed automatically (by hardware). If the result of the CRC check is normal, ACK is transmitted in the ACK field irrespective of whether or not the message can be received.

- Data frame reception

  If the received message is confirmed to be error-free by the CRC check, etc., the identifier in the mailbox (and also LAFM in the case of mailbox 0 only) and the identifier of the receive message are compared, and if a complete match is found, the message is stored in the mailbox. The message identifier comparison is carried out on each mailbox in turn, starting with mailbox 0 and ending with mailbox 15. If a complete match is found, the comparison ends at that point, the message is stored in the matching mailbox, and the corresponding receive complete bit (RXPR0–RXPR15) is set in the receive complete register (RXPR). However, when a mailbox 0 LAFM comparison is carried out, even if the identifier matches, the mailbox comparison sequence does not end at that point, but continues with mailbox 1 and then the remaining mailboxes. It is therefore possible for a message matching mailbox 0 to be received by another mailbox (however, the same message cannot be stored in more than one of mailboxes 1 to 15). If the corresponding bit (MBIMR0–MBIMR15) in the mailbox interrupt mask register (MBIMR) and the receive message interrupt mask (IMR1) in the interrupt mask register (IMR) are set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

- Remote frame reception

  Two kinds of messages—data frames and remote frames—can be stored in mailboxes. A remote frame differs from a data frame in that the remote reception request bit (RTR) in the message control register (MC[x]5) and the data field are 0 bytes. The data length to be returned in a data frame must be stored in the data length code (DLC) in the control field.

  When a remote frame (RTR = recessive) is received, the corresponding bit is set in the remote request wait register (RFPR). If the corresponding bit (MBIMR0–MBIMR15) in the mailbox interrupt mask register (MBIMR) and the remote frame request interrupt mask (IRR2) in the interrupt mask register (IMR) are set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

- Unread message reception

  When a received message matches the identifier in a mailbox, the message is stored in the mailbox. If a message overwrite occurs before the CPU reads the message, the corresponding bit (UMSR0–UMSR15) is set in the unread message register (UMSR). In overwriting of an unread message, when a new message is received before the corresponding bit in the receive complete register (RXPR) has been cleared, the unread message register (UMSR) is set. If the unread interrupt flag (IRR9) in the interrupt mask register (IMR) is set to the interrupt enable

value at this time, an interrupt can be sent to the CPU. Figure 16-9 shows a flowchart of unread message overwriting.

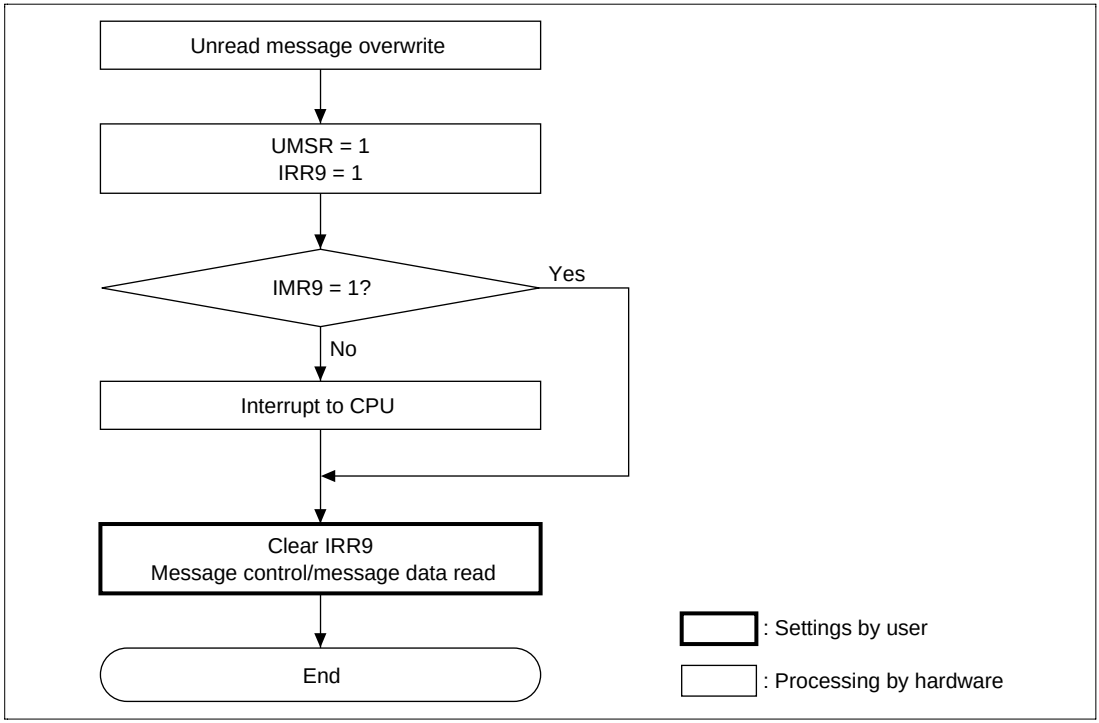


Figure 16-9 Unread Message Overwrite Flowchart

16.3.5 HCAN Sleep Mode

The HCAN is provided with an HCAN sleep mode that places the HCAN module in the sleep state to reduce current dissipation. Figure 16-10 shows a flowchart of the HCAN sleep mode.

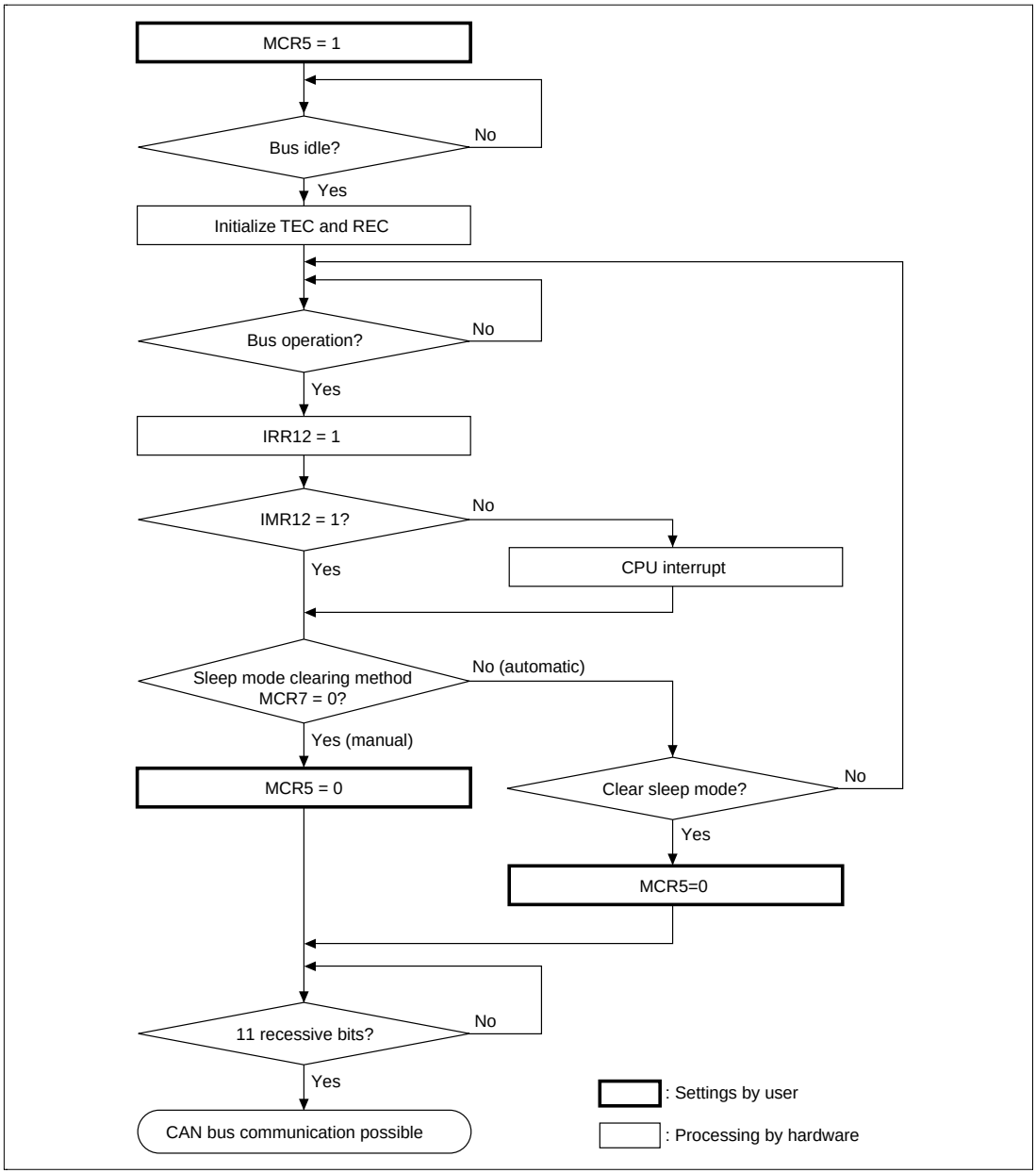


Figure 16-10 HCAN Sleep Mode Flowchart

HCAN sleep mode is entered by setting the HCAN sleep mode bit (MCR5) to 1 in the master control register (MCR). If the CAN bus is operating, the transition to HCAN sleep mode is delayed until the bus becomes idle.

Either of the following methods of clearing HCAN sleep mode can be selected by making a setting in the MCR7 bit.

1. Clearing by software
2. Clearing by CAN bus operation

Eleven recessive bits must be received after HCAN sleep mode is cleared before CAN bus communication is enabled again.

Clearing by software: HCAN sleep mode is cleared by writing a 0 to MCR5 from the CPU.

Clearing by CAN bus operation: Clearing by CAN bus operation occurs automatically when the CAN bus performs an operation and this change is detected. In this case, the first message is not received in the mailbox, and normal reception starts from the next message. When a change is detected on the CAN bus in HCAN sleep mode, the bus operation interrupt flag (IRR12) is set in the interrupt register (IRR). If the bus interrupt mask (IMR12) in the interrupt mask register (IMR) is set to the interrupt enable value at this time, an interrupt can be sent to the CPU.

16.3.6 HCAN Halt Mode

The HCAN halt mode is provided to enable mailbox settings to be changed without performing an HCAN hardware or software reset. Figure 16-11 shows a flowchart of the HCAN halt mode.

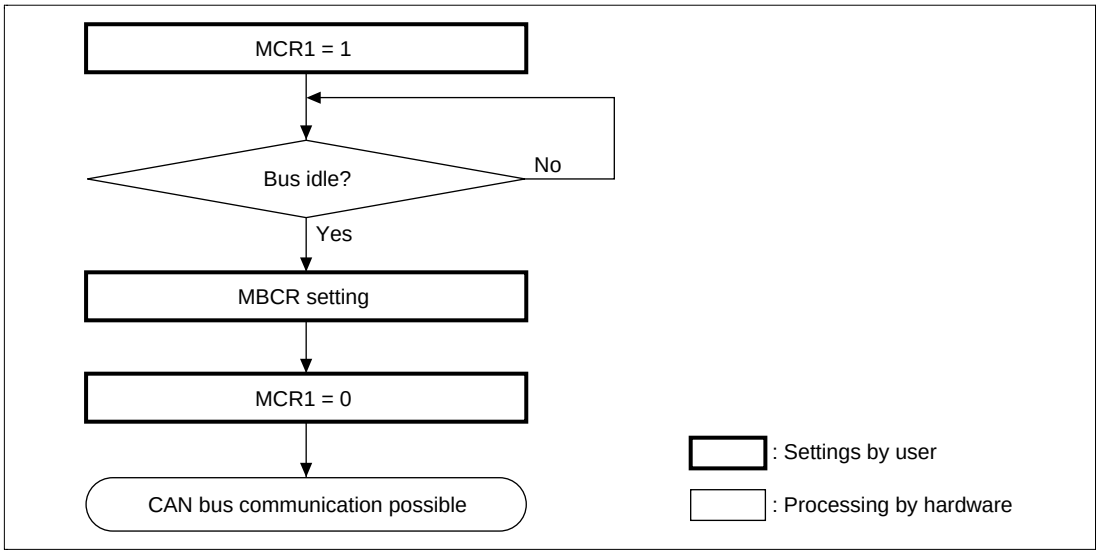


Figure 16-11 HCAN Halt Mode Flowchart

HCAN halt mode is entered by setting the halt request bit (MCR1) to 1 in the master control register (MCR). However, if the CAN bus is operating at the time of a transition, the transition to HCAN ALT mode is delayed until the bus becomes idle.

HCAN halt mode is cleared by clearing MCR1 to 0.

16.3.7 Interrupt Interface

There are 12 HCAN interrupt sources, to which five independent interrupt vectors are assigned. Table 16-5 lists the HCAN interrupt sources.

With the exception of the reset processing vector (IRR0), these sources can be masked. Masking is implemented using the mailbox interrupt mask register (MBIMR) and interrupt mask register (IMR).

Table 16-5 HCAN Interrupt Sources

Channel	IPR Bits	Vector	Vector Number	IRR Bit	Description
HCAN0	IPRM (2–0)	ERS0	108	IRR5	Error passive interrupt (TEC ≥ 128 or REC ≥ 128)
				IRR6	Bus off interrupt (TEC ≥ 256)
		OVR0	108	IRR0	Hardware reset processing interrupt
				IRR2	Remote frame reception interrupt
				IRR3	Error warning interrupt (TEC ≥ 96)
				IRR4	Error warning interrupt (REC ≥ 96)
				IRR7	Overload frame transmission interrupt/bus off recovery interrupt (11 recessive bits × 128 times)
				IRR9	Unread message overwrite interrupt
				IRR12	HCAN sleep mode CAN bus operation interrupt
		RM0	109	IRR1	Mailbox 0 message reception interrupt
		RM1	108	IRR1	Mailbox 1-15 message reception interrupt
		SLE0	108	IRR8	Message transmission/cancellation interrupt
HCAN1	IPRM (6–4)	ERS0	106	IRR5	Error passive interrupt (TEC ≥ 128 or REC ≥ 128)
				IRR6	Bus off interrupt (TEC ≥ 256)
		OVR0	106	IRR0	Hardware reset processing interrupt
				IRR2	Remote frame reception interrupt
				IRR3	Error warning interrupt (TEC ≥ 96)
				IRR4	Error warning interrupt (REC ≥ 96)
				IRR7	Overload frame transmission interrupt/bus off recovery interrupt (11 recessive bits × 128 times)
				IRR9	Unread message overwrite interrupt
				IRR12	HCAN sleep mode CAN bus operation interrupt
		RM0	107	IRR1	Mailbox 0 message reception interrupt
		RM1	106	IRR1	Mailbox 1-15 message reception interrupt
		SLE0	106	IRR8	Message transmission/cancellation interrupt

16.3.8 DTC Interface

The DTC can be activated by reception of a message in the HCAN's mailbox 0. When DTC transfer ends after DTC activation has been set, the RXPR0 and RFPR0 flags are acknowledge signal automatically. An interrupt request due to a receive interrupt from the HCAN cannot be sent to the CPU in this case. Figure 16-12 shows a DTC transfer flowchart.

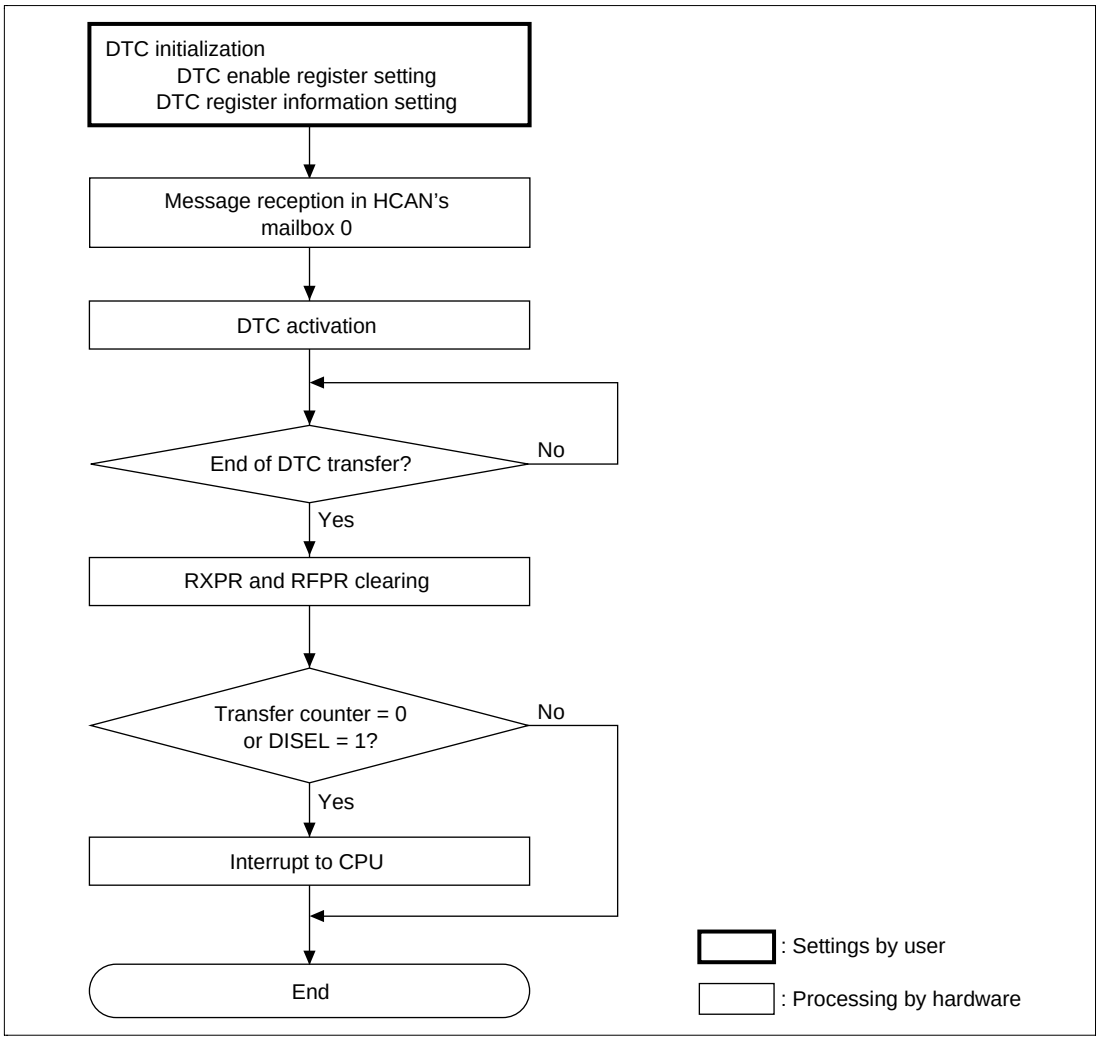


Figure 16-12 DTC Transfer Flowchart

16.4 CAN Bus Interface

A bus transceiver IC is necessary to connect the chip to a CAN bus. A Philips PCA82C250 transceiver IC, or compatible device, is recommended. Figure 16-13 shows a sample connection diagram.

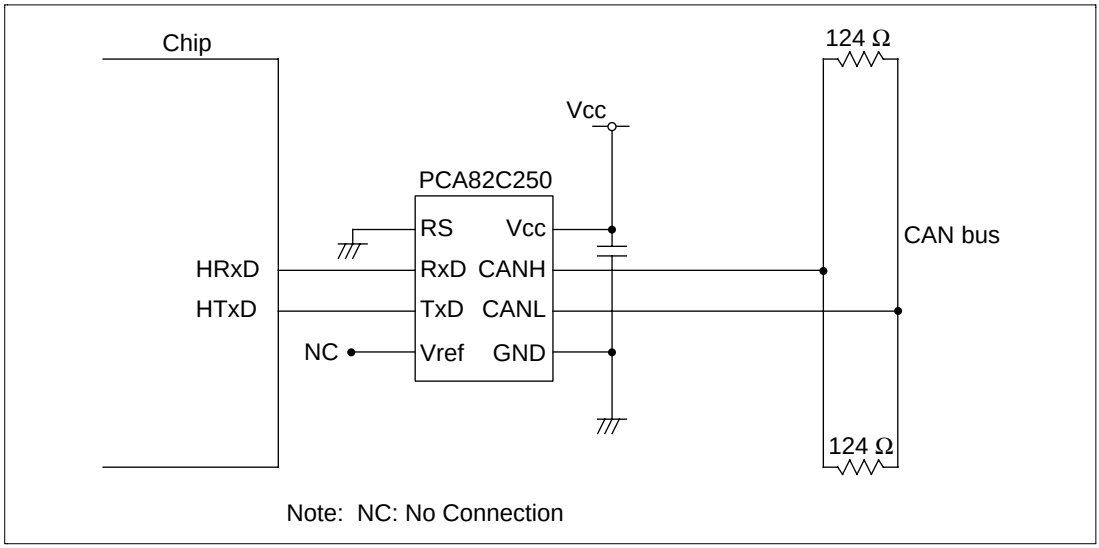


Figure 16-13 High-Speed Interface Using PCA82C250

16.5 Usage Notes

1. Reset

   The HCAN is reset by a reset, and in hardware standby mode and software standby mode. All the registers are initialized in a reset, but mailboxes (message control (MCx[x])/message data (MDx[x]) are not. However, after powering on, mailboxes (message control (MCx[x])/message data (MDx[x]) are initialized, and their values are undefined. Therefore, mailbox initialization must always be carried out after a reset or a transition to hardware standby mode or software standby mode. Also, the reset interrupt flag (IRR0) is always set after reset input or recovery from software standby mode. As this bit cannot be masked in the interrupt mask register (IMR), if HCAN interrupts are set as enabled by the interrupt controller without this flag having been cleared, an HCAN interrupt will be initiated immediately. IRR0 must therefore be cleared during initialization.

2. HCAN sleep mode

   The bus operation interrupt flag (IRR12) in the interrupt register (IRR) is set by bus operation in HCAN sleep mode. Therefore, this flag is not used by the HCAN to indicate sleep mode release. Also note that the reset status bit (GSR3) in the general status register (GSR) is set in sleep mode.

3. Interrupts

   When the mailbox interrupt mask register (MBIMR) is set, the interrupt register (IRR8.2.1) is not set by reception completion, transmission completion, or transmission cancellation for the set mailboxes.

4. Error counters

   In the case of error active and error passive, REC and TEC normally count up and down. In the bus off state, 11-bit recessive sequences are counted (REC + 1) using REC. If REC reaches 96 during the count, IRR4 and GSR1 are set.

5. Register access

   Byte or word access can be used on all HCAN registers. Longword access cannot be used.

6. HCAN medium-speed mode

   HCAN registers cannot be read or written to in medium-speed mode.

7. Register retention during standby

   All HCAN registers are initialized in hardware standby mode and software standby mode.

8. Usage of bit manipulation instructions

   The HCAN status flags are cleared by writing 1, so do not use a bit manipulation instruction to clear a flag.

   When clearing a flag, use the MOV instruction to write 1 to only the bit that is to be cleared.

Section 17 A/D Converter

17.1 Overview

The chip incorporates a successive approximation type 10-bit A/D converter that allows up to twelve analog input channels to be selected.

17.1.1 Features

A/D converter features are listed below.

- 10-bit resolution
- Twelve input channels
- Settable analog conversion voltage range
  - — Conversion of analog voltages with the reference voltage pin (Vref) as the analog reference voltage
- High-speed conversion
  - — Minimum conversion time: 13.3 µs per channel (at 20 MHz operation)
- Choice of single mode or scan mode
  - — Single mode: Single-channel A/D conversion
  - — Scan mode: Continuous A/D conversion on 1 to 4 channels
- Four data registers
  - — Conversion results are held in a 16-bit data register for each channel
- Sample and hold function
- Three kinds of conversion start
  - — Choice of software or timer conversion start trigger (TPU), or $\overline{\text{ADTRG}}$ pin
- A/D conversion end interrupt generation
  - — A/D conversion end interrupt (ADI) request can be generated at the end of A/D conversion
- Module stop mode can be set
  - — As the initial setting, A/D converter operation is halted. Register access is enabled by exiting module stop mode.

17.1.2 Block Diagram

Figure 17-1 shows a block diagram of the A/D converter.

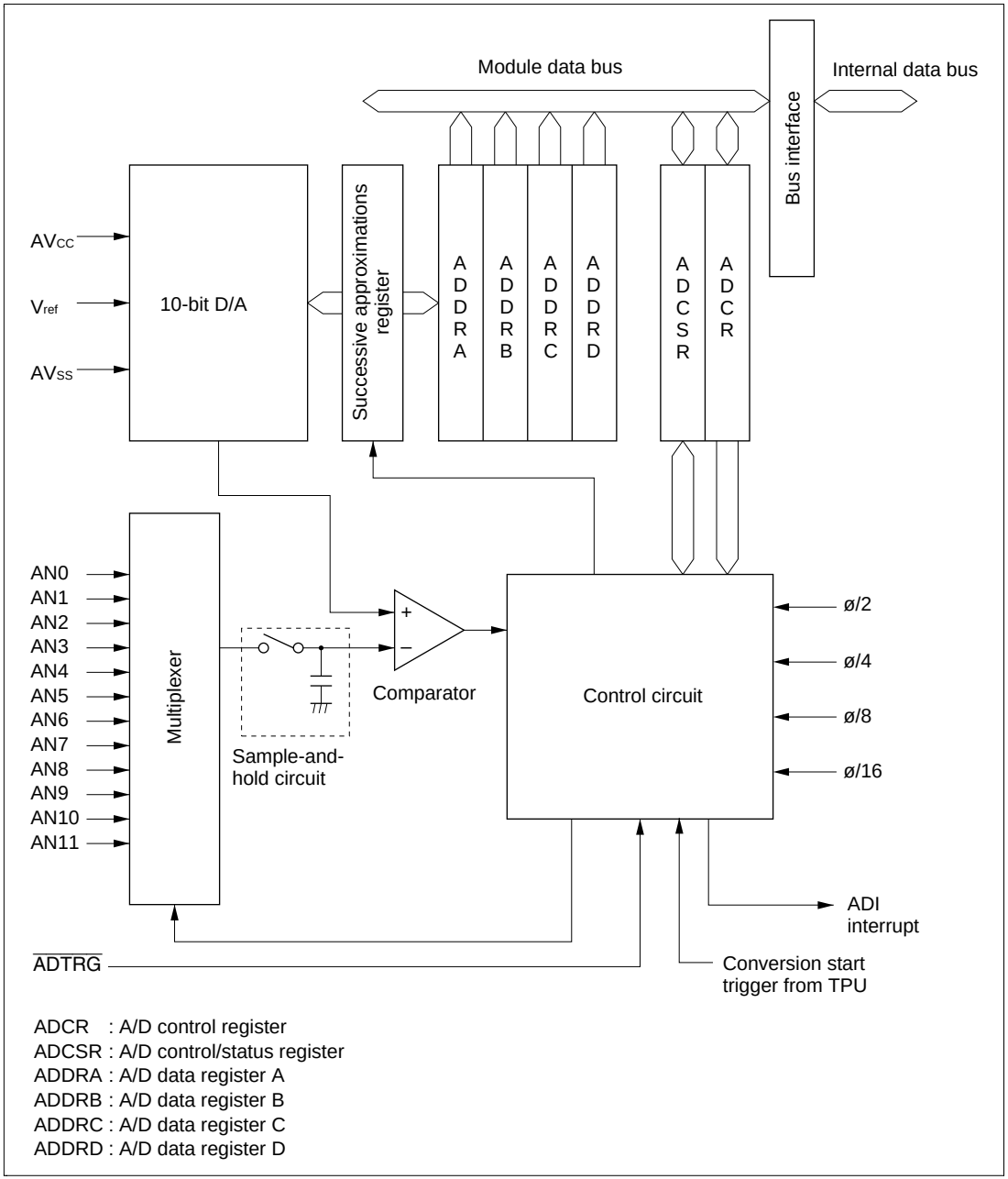


Figure 17-1 Block Diagram of A/D Converter

17.1.3 Pin Configuration

Table 17-1 summarizes the input pins used by the A/D converter.

The AVCC and AVSS pins are the power supply pins for the analog block in the A/D converter. The Vref pin is the A/D conversion reference voltage pin.

The 12 analog input pins are divided into two channel sets and two groups, with analog input pins 0 to 7 (AN0 to AN7) comprising channel set 0, analog input pins 8 to 11 (AN8 to AN11) comprising channel set 1, analog input pins 0 to 3 and 8 to 11 (AN0 to AN3, AN8 to AN11) comprising group 0, and analog input pins 4 to 7 (AN4 to AN7) comprising group 1.

Table 17-1 A/D Converter Pins

Pin Name	Symbol	I/O	Function
Analog power supply pin	AVCC	Input	Analog block power supply
Analog ground pin	AVSS	Input	Analog block ground and reference voltage
Reference voltage pin	Vref	Input	A/D conversion reference voltage
Analog input pin 0	AN0	Input	Channel set 0 (CH3 = 0) group 0 analog inputs
Analog input pin 1	AN1	Input	
Analog input pin 2	AN2	Input	
Analog input pin 3	AN3	Input	
Analog input pin 4	AN4	Input	Channel set 0 (CH3 = 0) group 1 analog inputs
Analog input pin 5	AN5	Input	
Analog input pin 6	AN6	Input	
Analog input pin 7	AN7	Input	
Analog input pin 8	AN8	Input	Channel set 1 (CH3 = 1) group 0 analog inputs
Analog input pin 9	AN9	Input	
Analog input pin 10	AN10	Input	
Analog input pin 11	AN11	Input	
A/D external trigger input pin	$\overline{\text{ADTRG}}$	Input	External trigger input for starting A/D conversion

17.1.4 Register Configuration

Table 17-2 summarizes the registers of the A/D converter.

Table 17-2 A/D Converter Registers

Name	Abbreviation	R/W	Initial Value	Address*[1]
A/D data register AH	ADDRAH	R	H'00	H'FF90
A/D data register AL	ADDRAL	R	H'00	H'FF91
A/D data register BH	ADDRBH	R	H'00	H'FF92
A/D data register BL	ADDRBL	R	H'00	H'FF93
A/D data register CH	ADDRCH	R	H'00	H'FF94
A/D data register CL	ADDRCL	R	H'00	H'FF95
A/D data register DH	ADDRDH	R	H'00	H'FF96
A/D data register DL	ADDRDL	R	H'00	H'FF97
A/D control/status register	ADCSR	R/(W)*[2]	H'00	H'FF98
A/D control register	ADCR	R/W	H'33	H'FF99
Module stop control register A	MSTPCRA	R/W	H'3F	H'FDE8

Notes: *1 Lower 16 bits of the address.
*2 Bit 7 can only be written with 0 for flag clearing.

17.2 Register Descriptions

17.2.1 A/D Data Registers A to D (ADDRA to ADDRD)

Bit	:	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
		AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
R/W	:	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

There are four 16-bit read-only ADDR registers, ADDRA to ADDRD, used to store the results of A/D conversion.

The 10-bit data resulting from A/D conversion is transferred to the ADDR register for the selected channel and stored there. The upper 8 bits of the converted data are transferred to the upper byte (bits 15 to 8) of ADDR, and the lower 2 bits are transferred to the lower byte (bits 7 and 6) and stored. Bits 5 to 0 are always read as 0.

The correspondence between the analog input channels and ADDR registers is shown in table 17-3.

ADDR can always be read by the CPU. The upper byte can be read directly, but for the lower byte, data transfer is performed via a temporary register (TEMP). For details, see section 17.3, Interface to Bus Master.

The ADDR registers are initialized to H'0000 by a reset, and in standby mode or module stop mode.

Table 17-3 Analog Input Channels and Corresponding ADDR Registers

Analog Input Channel				
Channel Set 0 (CH3 = 0)		Channel Set 1 (CH3 = 1)		
Group 0	Group 1	Group 0	Group 1	A/D Data Register
AN0	AN4	AN8	Setting prohibited	ADDRA
AN1	AN5	AN9	Setting prohibited	ADDRB
AN2	AN6	AN10	Setting prohibited	ADDRC
AN3	AN7	AN11	Setting prohibited	ADDRD

17.2.2 A/D Control/Status Register (ADCSR)

Bit	:	7	6	5	4	3	2	1	0
		ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only 0 can be written to bit 7, to clear this flag.

ADCSR is an 8-bit readable/writable register that controls A/D conversion operations.

ADCSR is initialized to H'00 by a reset, and in hardware standby mode or module stop mode.

Bit 7—A/D End Flag (ADF): Status flag that indicates the end of A/D conversion.

Bit 7

ADF	Description	
0	[Clearing conditions]	(Initial value)
	• When 0 is written to the ADF flag after reading ADF = 1	
	• When the DTC is activated by an ADI interrupt and ADDR is read	
1	[Setting conditions]	
	• Single mode: When A/D conversion ends	
	• Scan mode: When A/D conversion ends on all specified channels	

Bit 6—A/D Interrupt Enable (ADIE): Selects enabling or disabling of interrupt (ADI) requests at the end of A/D conversion.

Bit 6

ADIE	Description	
0	A/D conversion end interrupt (ADI) request disabled	(Initial value)
1	A/D conversion end interrupt (ADI) request enabled	

Bit 5—A/D Start (ADST): Selects starting or stopping on A/D conversion. Holds a value of 1 during A/D conversion.

The ADST bit can be set to 1 by software, a timer conversion start trigger, or the A/D external trigger input pin ($\overline{\text{ADTRG}}$).

Bit 5

ADST	Description
0	• A/D conversion stopped (Initial value)
1	• Single mode: A/D conversion is started. Cleared to 0 automatically when conversion on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues sequentially on the selected channels until ADST is cleared to 0 by software, a reset, or a transition to standby mode or module stop mode.

Bit 4—Scan Mode (SCAN): Selects single mode or scan mode as the A/D conversion operating mode. See section 17.4, Operation, for single mode and scan mode operation. Only set the SCAN bit while conversion is stopped (ADST = 0).

Bit 4

SCAN	Description
0	Single mode (Initial value)
1	Scan mode

Bit 3—Channel Select 3 (CH3): Switches the analog input pins assigned to group 0 or group 1. Setting CH3 to 1 enables AN8 to AN11 to be used instead of AN0 to AN7.

Bit 3

CH3	Description
1	AN8 to AN11 are group 0 analog input pins
0	AN0 to AN3 are group 0 analog input pins, AN4 to AN7 are group 1 analog input pins (Initial value)

Bits 2 to 0—Channel Select 2 to 0 (CH2 to CH0): Together with the SCAN bit, these bits select the analog input channels.

Only set the input channel while conversion is stopped (ADST = 0).

Channel Selection				Description	
CH3	**CH2**	**CH1**	**CH0**	**Single Mode (SCAN = 0)**	**Scan Mode (SCAN = 1)**
0	0	0	0	AN0 (Initial value)	AN0
			1	AN1	AN0, AN1
		1	0	AN2	AN0 to AN2
			1	AN3	AN0 to AN3
	1	0	0	AN4	AN4
			1	AN5	AN4, AN5
		1	0	AN6	AN4 to AN6
			1	AN7	AN4 to AN7
1	0	0	0	AN8	AN8
			1	AN9	AN8, AN9
		1	0	AN10	AN8 to AN10
			1	AN11	AN8 to AN11
	1	0	0	Setting prohibited	Setting prohibited
			1	Setting prohibited	Setting prohibited
		1	0	Setting prohibited	Setting prohibited
			1	Setting prohibited	Setting prohibited

17.2.3 A/D Control Register (ADCR)

Bit	:	7	6	5	4	3	2	1	0
		TRGS1	TRGS0	—	—	CKS1	CKS0	—	—
Initial value	:	0	0	1	1	0	0	1	1
R/W	:	R/W	R/W	—	—	R/W	R/W	—	—

ADCR is an 8-bit readable/writable register that enables or disables external triggering of A/D conversion operations and sets the A/D conversion time.

ADCR is initialized to H'33 by a reset, and in standby mode or module stop mode.

Bits 7 and 6—Timer Trigger Select 1 and 0 (TRGS1, TRGS0): Select enabling or disabling of the start of A/D conversion by a trigger signal. Only set bits TRGS1 and TRGS0 while conversion is stopped (ADST = 0).

Bit 7	Bit 6		
TRGS1	**TRGS0**	**Description**	
0	0	A/D conversion start by software is enabled	(Initial value)
	1	A/D conversion start by TPU conversion start trigger is enabled	
1	0	Setting prohibited	
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled	

Bits 5, 4, 1, and 0—Reserved: These bits are reserved; they are always read as 1 and cannot be modified.

Bits 3 and 2—Clock Select 1 and 0 (CKS1, CKS0): These bits select the A/D conversion time. The conversion time should be changed only when ADST = 0.

Set bits CKS1 and CKS0 to give a conversion time of at least 10 µs.

Bit 3	Bit 2		
CKS1	**CKS0**	**Description**	
0	0	Conversion time = 530 states (max.)	(Initial value)
	1	Conversion time = 266 states (max.)	
1	0	Conversion time = 134 states (max.)	
	1	Conversion time = 68 states (max.)	

17.2.4 Module Stop Control Register A (MSTPCRA)

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCR is an 8-bit readable/writable register that performs module stop mode control.

When the MSTPA1 bit in MSTPCR is set to 1, A/D converter operation stops at the end of the bus cycle and a transition is made to module stop mode. Registers cannot be read or written to in module stop mode. For details, see section 23A.5, 23B.5, Module Stop Mode.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized by a reset and in software standby mode.

Bit 1—Module Stop (MSTPA1): Specifies the A/D converter module stop mode.

Bit 1

MSTPA1	Description	
0	A/D converter module stop mode cleared	
1	A/D converter module stop mode set	(Initial value)

17.3 Interface to Bus Master

ADDRA to ADDRD are 16-bit registers, and the data bus to the bus master is 8 bits wide. Therefore, in accesses by the bus master, the upper byte is accessed directly, but the lower byte is accessed via a temporary register (TEMP).

A data read from ADDR is performed as follows. When the upper byte is read, the upper byte value is transferred to the CPU and the lower byte value is transferred to TEMP. Next, when the lower byte is read, the TEMP contents are transferred to the CPU.

When reading ADDR. always read the upper byte before the lower byte. It is possible to read only the upper byte, but if only the lower byte is read, incorrect data may be obtained.

Figure 17-2 shows the data flow for ADDR access.

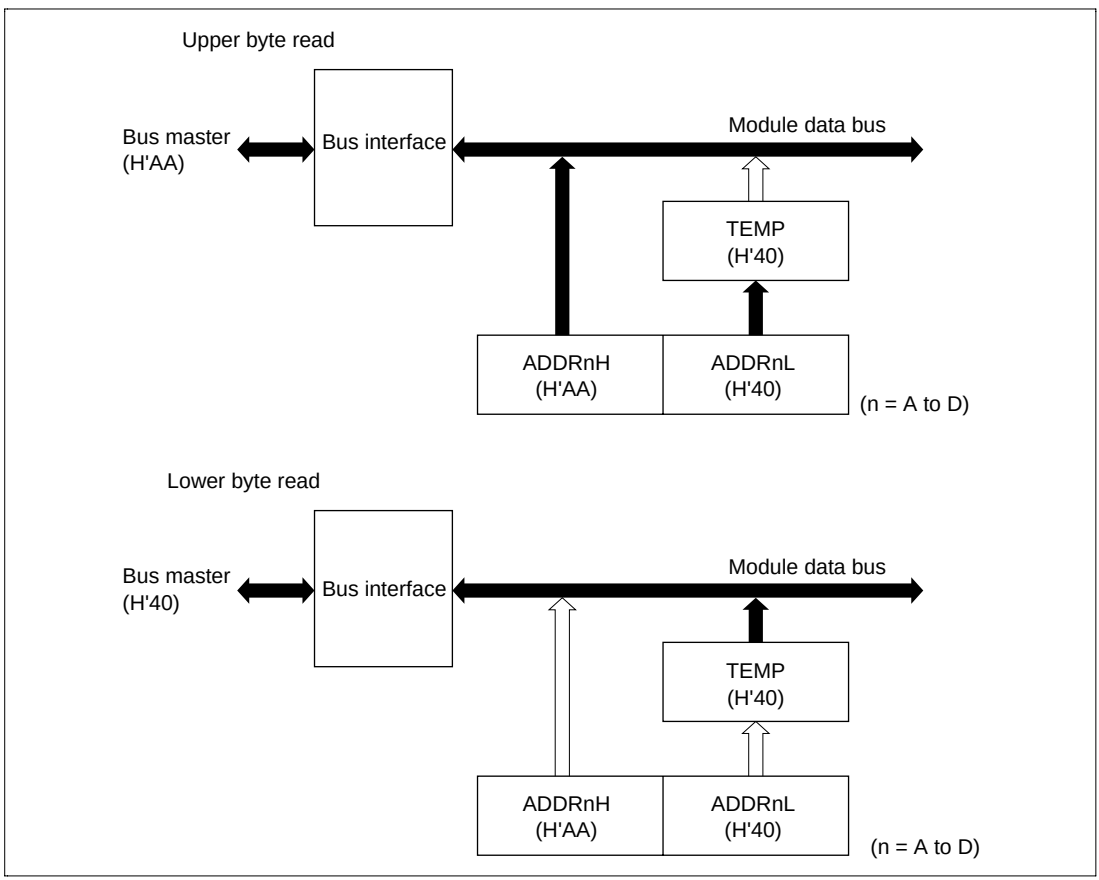


Figure 17-2 ADDR Access Operation (Reading H'AA40)

17.4 Operation

The A/D converter operates by successive approximation with 10-bit resolution. It has two operating modes: single mode and scan mode.

17.4.1 Single Mode (SCAN = 0)

Single mode is selected when A/D conversion is to be performed on a single channel only. A/D conversion is started when the ADST bit is set to 1, according to the software or external trigger input. The ADST bit remains set to 1 during A/D conversion, and is automatically cleared to 0 when conversion ends.

On completion of conversion, the ADF flag is set to 1. If the ADIE bit is set to 1 at this time, an ADI interrupt request is generated. The ADF flag is cleared by writing 0 after reading ADCSR.

When the operating mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again. The ADST bit can be set at the same time as the operating mode or input channel is changed.

Typical operations when channel 1 (AN1) is selected in single mode are described next. Figure 17-3 shows a timing diagram for this example.

[1] Single mode is selected (SCAN = 0), input channel AN1 is selected (CH3 = 0, CH2 = 0, CH1 = 0, CH0 = 1), the A/D interrupt is enabled (ADIE = 1), and A/D conversion is started (ADST = 1).

[2] When A/D conversion is completed, the result is transferred to ADDRB. At the same time the ADF flag is set to 1, the ADST bit is cleared to 0, and the A/D converter becomes idle.

[3] Since ADF = 1 and ADIE = 1, an ADI interrupt is requested.

[4] The A/D interrupt handling routine starts.

[5] The routine reads ADCSR, then writes 0 to the ADF flag.

[6] The routine reads and processes the connection result (ADDRB).

[7] Execution of the A/D interrupt handling routine ends. After that, if the ADST bit is set to 1, A/D conversion starts again and steps [2] to [7] are repeated.

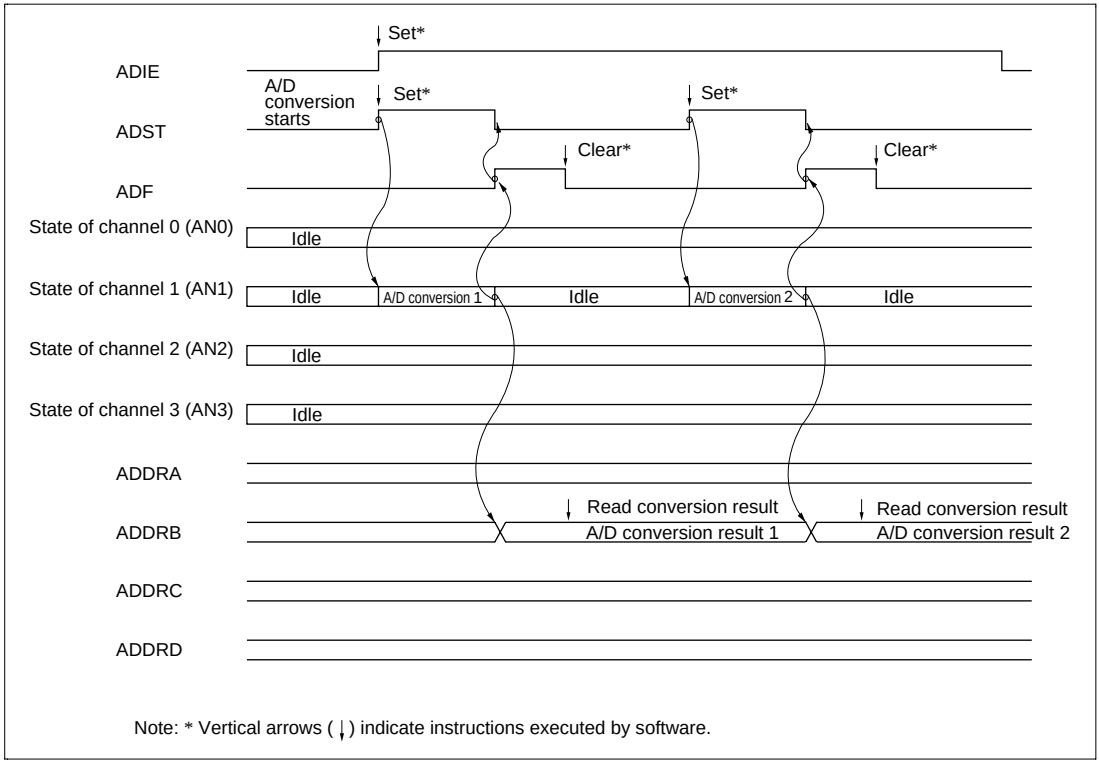


Figure 17-3 Example of A/D Converter Operation (Single Mode, Channel 1 Selected)

17.4.2 Scan Mode (SCAN = 1)

Scan mode is useful for monitoring analog inputs in a group of one or more channels. When the ADST bit is set to 1 by a software, timer or external trigger input, A/D conversion starts on the first channel in the group (AN0). When two or more channels are selected, after conversion of the first channel ends, conversion of the second channel (AN1) starts immediately. A/D conversion continues cyclically on the selected channels until the ADST bit is cleared to 0. The conversion results are transferred for storage into the ADDR registers corresponding to the channels.

When the operating mode or analog input channel must be changed during analog conversion, to prevent incorrect operation, first clear the ADST bit to 0 in ADCSR to halt A/D conversion. After making the necessary changes, set the ADST bit to 1 to start A/D conversion again from the first channel (AN0). The ADST bit can be set at the same time as the operating mode or input channel is changed.

Typical operations when three channels (AN0 to AN2) are selected in scan mode are described next. Figure 17-4 shows a timing diagram for this example.

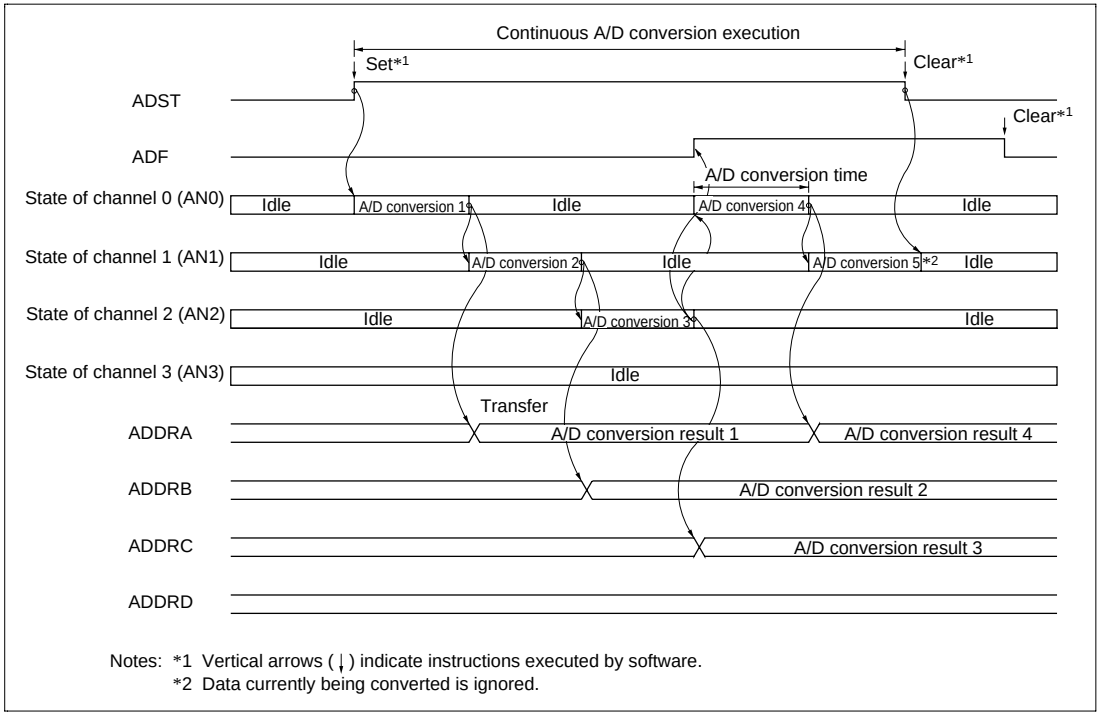
[1] Scan mode is selected (SCAN = 1), channel set 0 is selected (CH3 = 0), scan group 0 is selected (CH2 = 0), analog input channels AN0 to AN2 are selected (CH1 = 1, CH0 = 0), and A/D conversion is started (ADST = 1)

[2] When A/D conversion of the first channel (AN0) is completed, the result is transferred to ADDRA. Next, conversion of the second channel (AN1) starts automatically.

[3] Conversion proceeds in the same way through the third channel (AN2).

[4] When conversion of all the selected channels (AN0 to AN2) is completed, the ADF flag is set to 1 and conversion of the first channel (AN0) starts again. If the ADIE bit is set to 1 at this time, an ADI interrupt is requested after A/D conversion ends.

[5] Steps [2] to [4] are repeated as long as the ADST bit remains set to 1. When the ADST bit is cleared to 0, A/D conversion stops. After that, if the ADST bit is set to 1, A/D conversion starts again from the first channel (AN0).



Notes: *1 Vertical arrows (↓) indicate instructions executed by software.
*2 Data currently being converted is ignored.

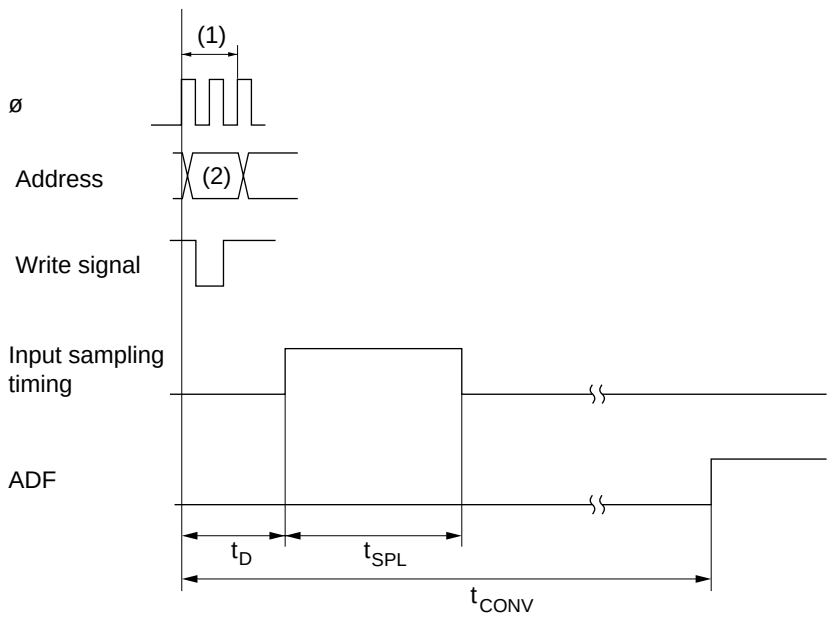
Figure 17-4 Example of A/D Converter Operation (Scan Mode, 3 Channels AN0 to AN2 Selected)

17.4.3 Input Sampling and A/D Conversion Time

The A/D converter has a built-in sample-and-hold circuit. The A/D converter samples the analog input at a time t_D after the ADST bit is set to 1, then starts conversion. Figure 17-5 shows the A/D conversion timing. Table 17-4 indicates the A/D conversion time.

As indicated in figure 17-5, the A/D conversion time includes t_D and the input sampling time. The length of t_D varies depending on the timing of the write access to ADCSR. The total conversion time therefore varies within the ranges indicated in table 17-4.

In scan mode, the values given in table 17-4 apply to the first conversion time. The values given in table 17-5 apply to the second and subsequent conversions. In both cases, set bits CKS1 and CKS0 in ADCR to give a conversion time of at least 10 µs.



Legend
(1) : ADCSR write cycle
(2) : ADCSR address
t_D : A/D conversion start delay
t_{SPL} : Input sampling time
t_{CONV} : A/D conversion time

Figure 17-5 A/D Conversion Timing

Table 17-4 A/D Conversion Time (Single Mode)

		CKS1 = 0						CKS1 = 0					
		CKS0 = 0			CKS0 = 1			CKS0 = 0			CKS0 = 1		
Item	**Symbol**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**
A/D conversion start delay	t_D	18	—	33	10	—	17	6	—	9	4	—	5
Input sampling time	t_{SPL}	—	127	—	—	63	—	—	31	—	—	15	—
A/D conversion time	t_{CONV}	515	—	530	259	—	266	131	—	134	67	—	68

Note: Values in the table are the number of states.

Table 17-5 A/D Conversion Time (Scan Mode)

CKS1	CKS0	Conversion Time (State)
0	0	512 (Fixed)
	1	256 (Fixed)
1	0	128 (Fixed)
	1	64 (Fixed)

17.4.4 External Trigger Input Timing

A/D conversion can be externally triggered. When the TRGS1 and TRGS0 bits are set to 11 in ADCR, external trigger input is enabled at the $\overline{\text{ADTRG}}$ pin. A falling edge at the $\overline{\text{ADTRG}}$ pin sets the ADST bit to 1 in ADCSR, starting A/D conversion. Other operations, in both single and scan modes, are the same as if the ADST bit has been set to 1 by software. Figure 17-6 shows the timing.

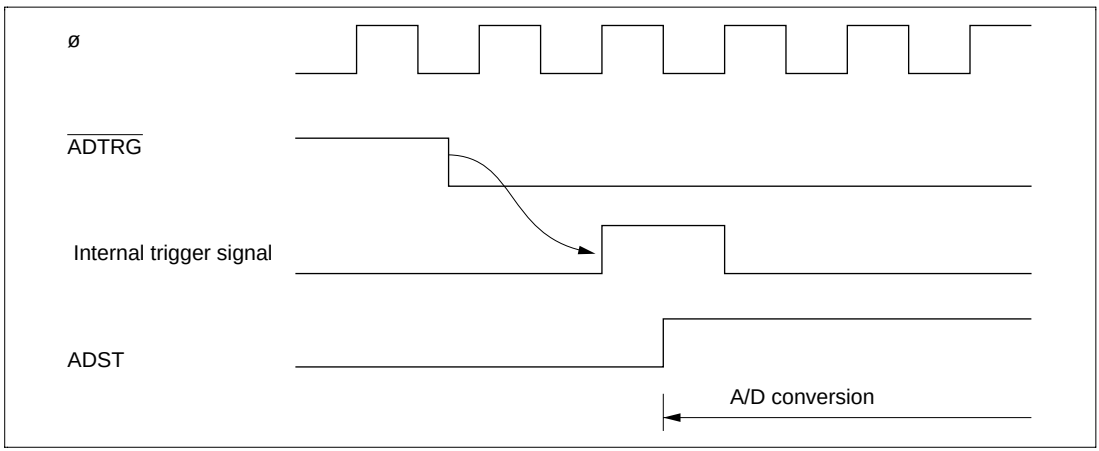


Figure 17-6 External Trigger Input Timing

17.5 Interrupts

The A/D converter generates an A/D conversion end interrupt (ADI) at the end of A/D conversion. ADI interrupt requests can be enabled or disabled by means of the ADIE bit in ADCSR.

The DTC can be activated by an ADI interrupt. Having the converted data read by the DTC in response to an ADI interrupt enables continuous conversion to be achieved without imposing a load on software.

The A/D converter interrupt source is shown in table 17-6.

Table 17-6 A/D Converter Interrupt Source

Interrupt Source	Description	DTC Activation
ADI	Interrupt due to end of conversion	Possible

17.6 Usage Notes

The following points should be noted when using the A/D converter.

Setting Range of Analog Power Supply and Other Pins:

(1) Analog input voltage range

The voltage applied to analog input pin ANn during A/D conversion should be in the range AVSS ≤ ANn ≤ Vref.

(2) Relation between AVCC, AVSS and VCC, VSS

As the relationship between AVSS and VSS, set AVSS = VSS. If the A/D converter is not used, set AVCC = VCC, and do not leave the AVCC and AVSS pins open or no account.

(3) Vref input range

The analog reference voltage input at the Vref pin set in the range Vref ≤ AVCC.

If conditions (1), (2), and (3) above are not met, the reliability of the device may be adversely affected.

Notes on Board Design: In board design, digital circuitry and analog circuitry should be as mutually isolated as possible, and layout in which digital circuit signal lines and analog circuit signal lines cross or are in close proximity should be avoided as far as possible. Failure to do so may result in incorrect operation of the analog circuitry due to inductance, adversely affecting A/D conversion values.

Also, digital circuitry must be isolated from the analog input signals (AN0 to AN11), analog reference power supply (Vref), and analog power supply (AVCC) by the analog ground (AVSS). Also, the analog ground (AVSS) should be connected at one point to a stable digital ground (VSS) on the board.

Notes on Noise Countermeasures: A protection circuit connected to prevent damage due to an abnormal voltage such as an excessive surge at the analog input pins (AN0 to AN11) and analog reference power supply (Vref) should be connected between AVCC and AVSS as shown in figure 17-7.

Also, the bypass capacitors connected to AVCC and Vref and the filter capacitor connected to AN0 to AN11 must be connected to AVSS.

If a filter capacitor is connected as shown in figure 17-7, the input currents at the analog input pins (AN0 to AN11) are averaged, and so an error may arise. Also, when A/D conversion is performed frequently, as in scan mode, if the current charged and discharged by the capacitance of the sample-and-hold circuit in the A/D converter exceeds the current input via the input impedance (R_{in}), an error will arise in the analog input pin voltage. Careful consideration is therefore required when deciding the circuit constants.

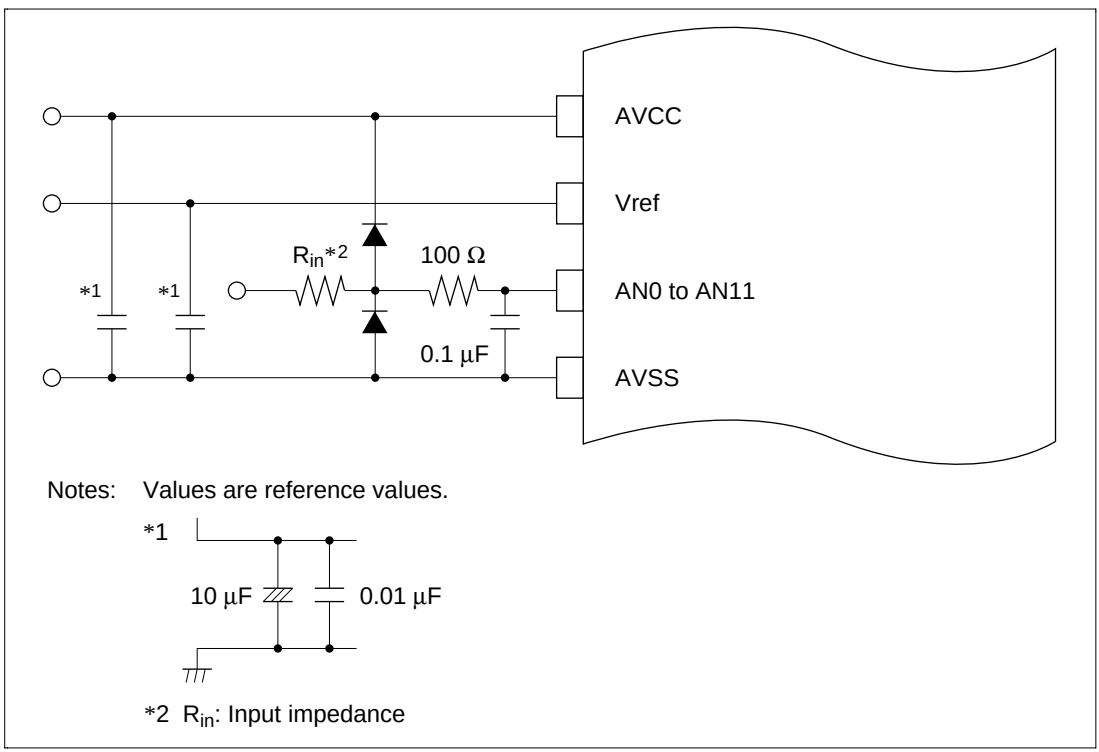
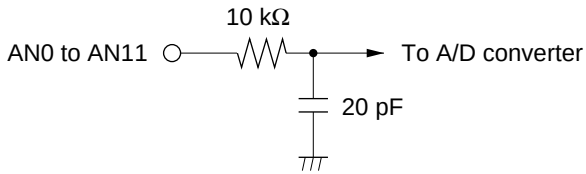


Figure 17-7 Example of Analog Input Protection Circuit

Table 17-7 Analog Pin Specifications

Item	Min	Max	Unit
Analog input capacitance	—	20	pF
Permissible signal source impedance	—	5	kΩ



Note: Values are reference values.

Figure 17-8 Analog Input Pin Equivalent Circuit

A/D Conversion Precision Definitions: The chip's A/D conversion precision definitions are given below.

- Resolution

  The number of A/D converter digital output codes

- Offset error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from the minimum voltage value B'0000000000 (H'00) to B'0000000001 (H'01) (see figure 17-10).

- Full-scale error

  The deviation of the analog input voltage value from the ideal A/D conversion characteristic when the digital output changes from B'1111111110 (H'3E) to B'1111111111 (H'3F) (see figure 17-10).

- Quantization error

  The deviation inherent in the A/D converter, given by 1/2 LSB (see figure 17-9).

- Nonlinearity error

  The error with respect to the ideal A/D conversion characteristic between the zero voltage and the full-scale voltage. Does not include the offset error, full-scale error, or quantization error.

- Absolute precision

  The deviation between the digital value and the analog input value. Includes the offset error, full-scale error, quantization error, and nonlinearity error.

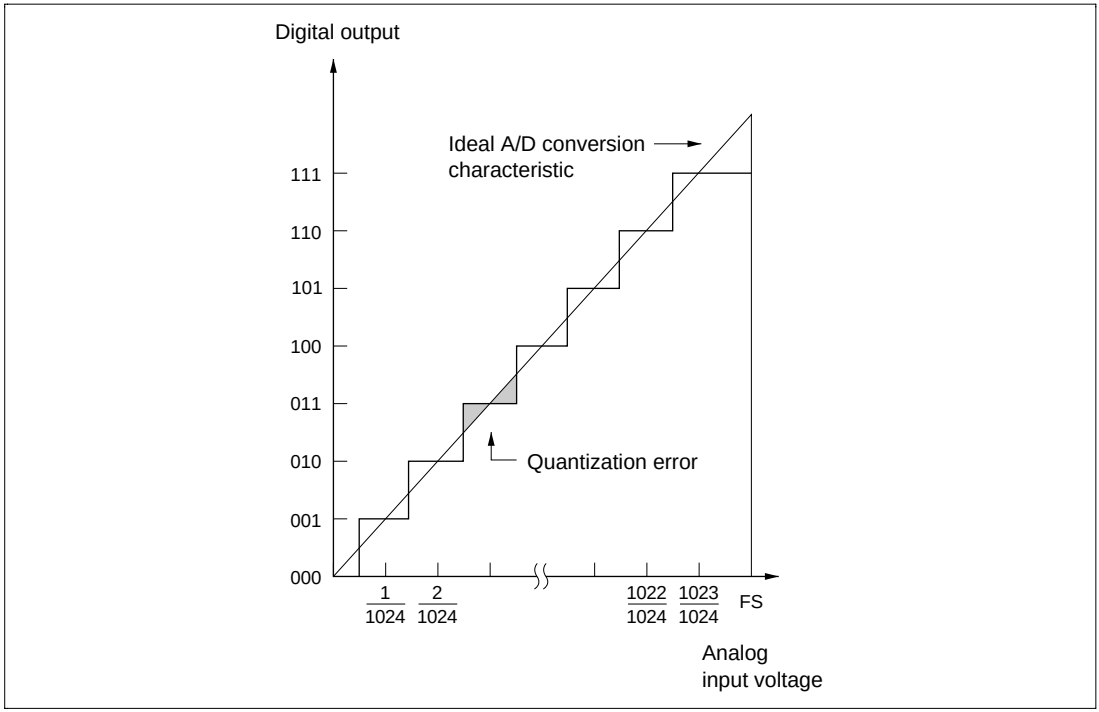


Figure 17-9 A/D Conversion Precision Definitions (1)

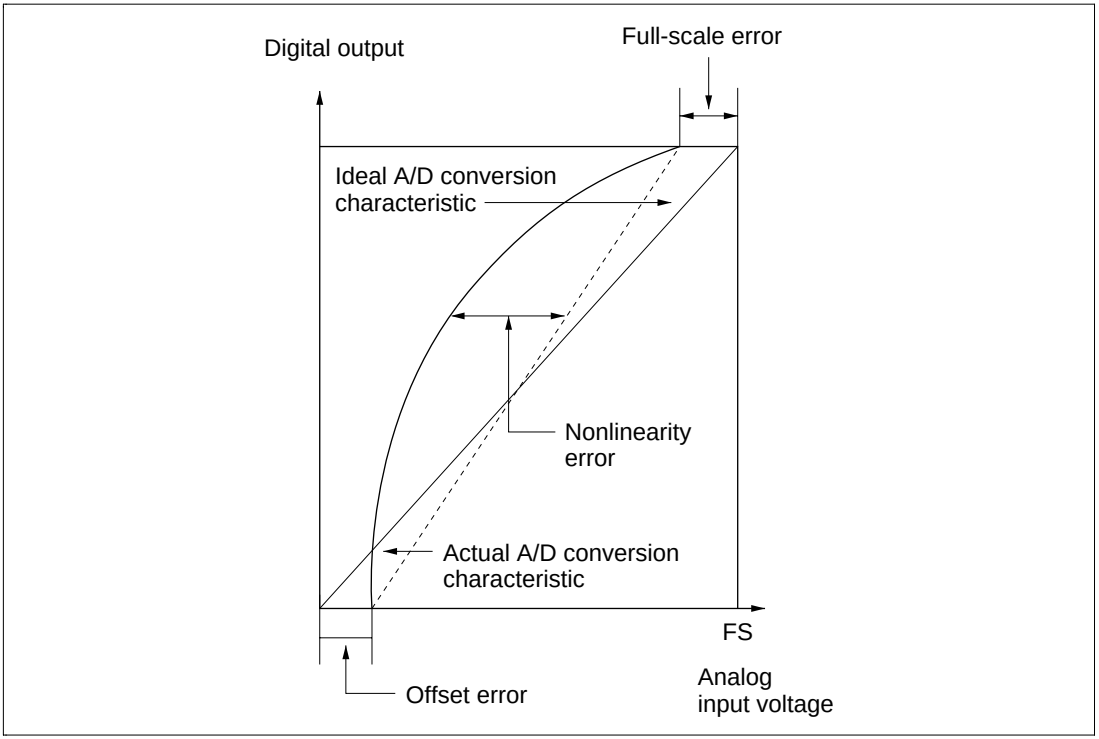


Figure 17-10 A/D Conversion Precision Definitions (2)

Permissible Signal Source Impedance: The chip's analog input is designed so that conversion precision is guaranteed for an input signal for which the signal source impedance is 10 kΩ or less. This specification is provided to enable the A/D converter's sample-and-hold circuit input capacitance to be charged within the sampling time; if the sensor output impedance exceeds 10 kΩ, charging may be insufficient and it may not be possible to guarantee the A/D conversion precision.

However, if a large capacitance is provided externally, the input load will essentially comprise only the internal input resistance of 10 kΩ, and the signal source impedance is ignored.

However, since a low-pass filter effect is obtained in this case, it may not be possible to follow an analog signal with a large differential coefficient (e.g., 5 mV/µs or greater).

When converting a high-speed analog signal, a low-impedance buffer should be inserted.

Influences on Absolute Precision: Adding capacitance results in coupling with GND, and therefore noise in GND may adversely affect absolute precision. Be sure to make the connection to an electrically stable GND such as AVSS.

Care is also required to insure that filter circuits do not communicate with digital signals on the mounting board, so acting as antennas.

Figure 17-11 shows an example of analog input circuit.

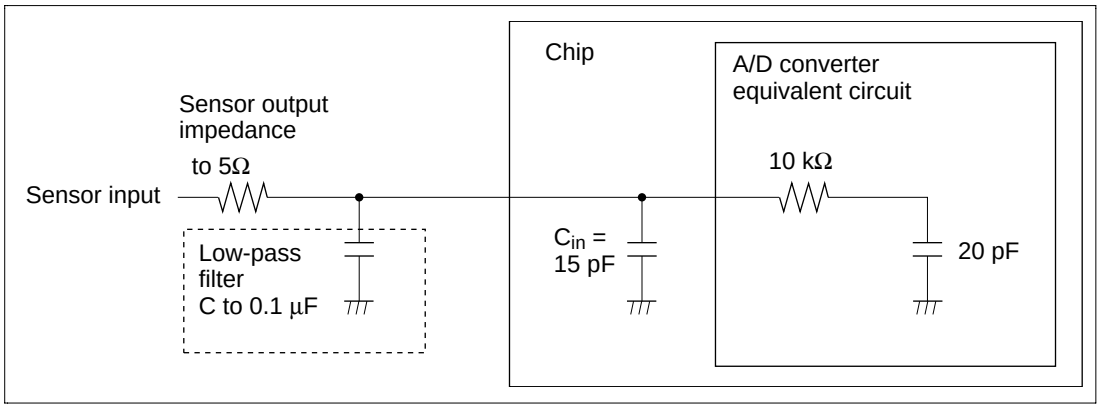


Figure 17-11 Example of Analog Input Circuit

Section 18 D/A Converter

18.1 Overview

The chip has an on-chip D/A converter module with two channels.

18.1.1 Features

Features of the D/A converter module are listed below.

- Eight-bit resolution
- Two-channel output
- Maximum conversion time: 10 µs (with 20-pF load capacitance)
- Output voltage: 0 V to Vref
- D/A output retention in software standby mode
- Possible to set module stop mode

  Operation of D/A converter is disenabled by initial values. It is possible to access the register by canceling module stop mode.

18.1.2 Block Diagram

Figure 18-1 shows a block diagram of the D/A converter.

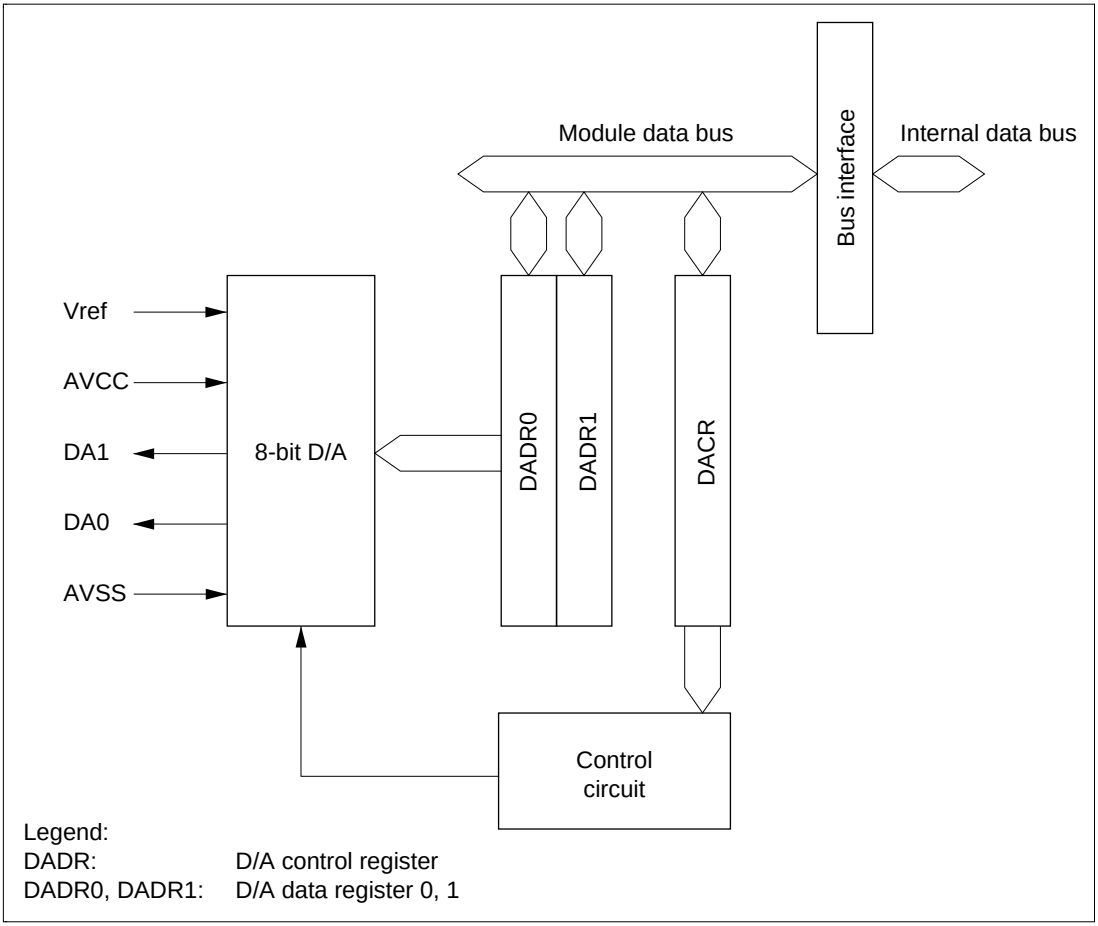


Figure 18-1 Block Diagram of D/A Converter

18.1.3 Input and Output Pins

Table 18-1 lists the input and output pins used by the D/A converter module.

Table 18-1 Input and Output Pins of D/A Converter Module

Name	Abbreviation	I/O	Function
Analog supply voltage	AVCC	Input	Power supply for analog circuits
Analog ground	AVSS	Input	Ground and reference voltage for analog circuits
Analog output 0	DA0	Output	Analog output channel 0
Analog output 1	DA1	Output	Analog output channel 1
Reference voltage	Vref	Input	Reference voltage of analog section

18.1.4 Register Configuration

Table 18-2 lists the registers of the D/A converter module.

Table 18-2 D/A Converter Registers

Channel	Name	Abbreviation	R/W	Initial Value	Address*
0, 1	D/A data register 0	DADR0	R/W	H'00	H'FFA4
	D/A data register 1	DADR1	R/W	H'00	H'FFA5
	D/A control register 01	DACR01	R/W	H'1F	H'FFA6
All	Module stop control register A	MSTPCRA	R/W	H'3F	H'FDF8

Note: * Lower 16 bits of the address.

18.2 Register Descriptions

18.2.1 D/A Data Registers 0, 1 (DADR0, DADR1)

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

D/A data registers 0, 1 (DADR0, DADR1) are 8-bit readable/writable registers that store data to be converted. When analog output is enabled, the value in the D/A data register is converted and output continuously at the analog output pin.

The D/A data registers are initialized to H'00 by a reset and in hardware standby mode.

18.2.2 D/A Control Register 01 (DACR01)

Bit	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value	0	0	0	1	1	1	1	1
Read/Write	R/W	R/W	R/W	—	—	—	—	—

DACR01 is an 8-bit readable/writable register that controls the operation of the D/A converter module.

DACR01 is initialized to H'1F by a reset and in hardware standby mode.

Bit 7—D/A Output Enable 1 (DAOE1): Controls D/A conversion and analog output.

Bit 7 DAOE1	Description	
0	Analog output DA1 is disabled	(Initial value)
1	D/A conversion is enabled on channel 1. Analog output DA1 is enabled	

Bit 6—D/A Output Enable 0 (DAOE0): Controls D/A conversion and analog output.

Bit 6 DAOE0	Description	
0	Analog output DA0 is disabled	(Initial value)
1	D/A conversion is enabled on channel 0. Analog output DA0 is enabled	

Bit 5—D/A Enable (DAE): Controls D/A conversion, in combination with bits DAOE0 and DAOE1. D/A conversion is controlled independently on channels 0 and 1 when DAE = 0. Channels 0 and 1 are controlled together when DAE = 1.

Output of the converted results is always controlled independently by DAOE0 and DAOE1.

Bit 7 DAOE1	Bit 6 DAOE0	Bit 5 DAE	D/A conversion
0	0	*	Disabled on channels 0 and 1
	1	0	Enabled on channel 0 Disabled on channel 1
		1	Enabled on channels 0 and 1
1	0	0	Disabled on channel 0 Enabled on channel 1
		1	Enabled on channels 0 and 1
	1	*	Enabled on channels 0 and 1

*: Don't care

If the chip enters software standby mode while D/A conversion is enabled, the D/A output is retained and the analog power supply current is the same as during D/A conversion. If it is necessary to reduce the analog power supply current in software standby mode, disable D/A output by clearing both the DAOE0 and DAOE1 bits to 0.

Bits 4 to 0—Reserved: These bits cannot be modified and are always read as 1.

18.2.3 Module Stop Control Register A (MSTPCRA)

MSTPCRA

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRA is an 8-bit readable/writable registers that performs module stop mode control. When the MSTPA2 is set to 1, the D/A converter halts and enters module stop mode at the end of the bus cycle. Register read/write is disenabled in module stop mode. See section 23A.5, 23B.5, Module Stop Mode, for details.

MSTPCRA is initialized to H'3F by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 2—Module Stop (MSTPA2): Specifies D/A converter (channels 0 and 1) module stop mode.

Bit 2

MSTPA2	Description	
0	D/A converter (channels 0 and 1) module stop mode is cleared	
1	D/A converter (channels 0 and 1) module stop mode is set	(Initial value)

18.3 Operation

The D/A converter module has one built-in D/A converter circuits that can operate independently.

D/A conversion is performed continuously whenever enabled by the D/A control register (DACR). When a new value is written in DADR0 or DADR1, conversion of the new value begins immediately. The converted result is output by setting the DAOE0 or DAOE1 bit to 1.

An example of conversion on channel 0 is given next. Figure 18-2 shows the timing.

- Software writes the data to be converted in DADR0.
- D/A conversion begins when the DAOE0 bit in DACR is set to 1. After the elapse of the conversion time, analog output appears at the DA0 pin. Contents of DADR / 256 × Vref
  This output continues until a new value is written in DADR0 or the DAOE0 bit is cleared to 0.
- If a new value is written in DADR0, conversion begins immediately. Output of the converted result begins after the conversion time.
- When the DAOE0 bit is cleared to 0, DA0 becomes an input pin.

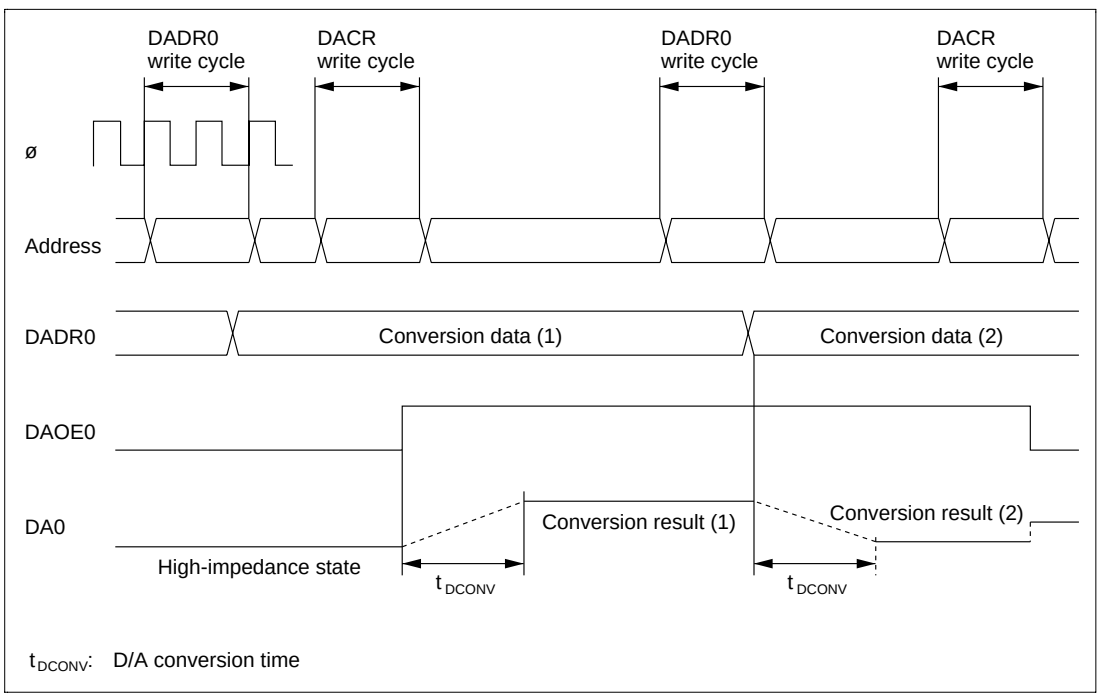


Figure 18-2 D/A Conversion (Example)

Section 19 Motor Control PWM Timer

19.1 Overview

The chip has an on-chip motor control PWM (pulse width modulator) with a maximum capability of 16 pulse outputs.

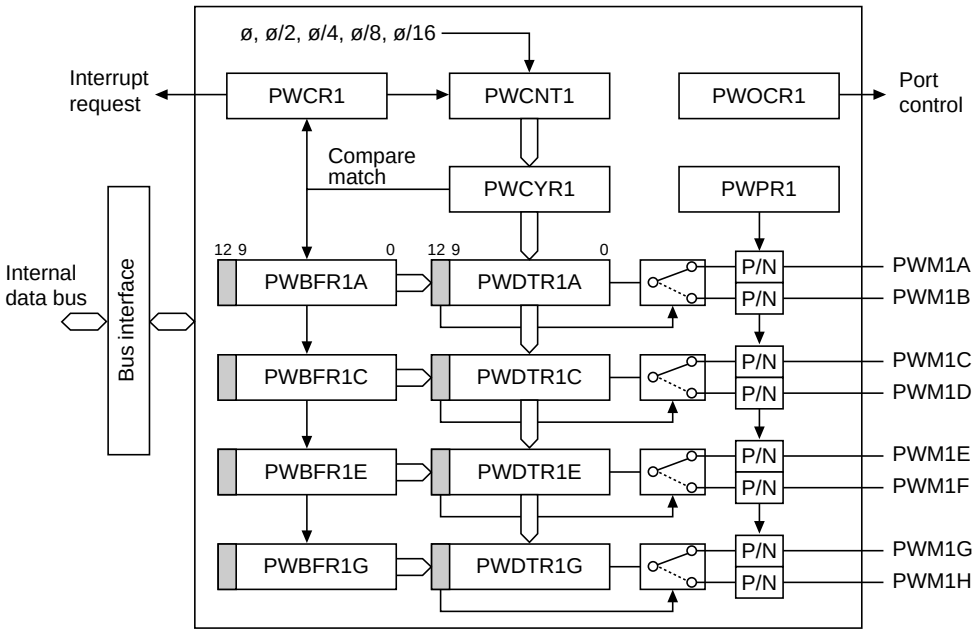
19.1.1 Features

Features of the motor control PWM are given below.

- Maximum of 16 pulse outputs
  - — Two 10-bit PWM channels, each with eight outputs.
  - — Each channel is provided with a 10-bit counter (PWCNT) and cycle register (PWCYR).
  - — Duty and output polarity can be set for each output.
- Buffered duty registers
  - — Duty registers (PWDTR) are provided with buffer registers (PWBFR), with data transferred automatically every cycle.
  - — Channel 1 has four duty registers and four buffer registers.
  - — Channel 2 has eight duty registers and four buffer registers.
- 0% to 100% duty
  - — A duty cycle of 0% to 100% can be set by means of a duty register setting.
- Five operating clocks
  - — There is a choice of five operating clocks (ø, ø/2, ø/4, ø/8, ø/16).
- High-speed access via internal 16-bit-bus
  - — High-speed access is possible via a 16-bit bus interface.
- Two interrupt sources
  - — An interrupt can be requested independently for each channel by a cycle register compare match.
- Automatic transfer of register data
  - — Block transfer and one-word data transfer are possible by activating the data transfer controller (DTC).
- Module stop mode
  - — As the initial setting, PWM operation is halted. Register access is enabled by clearing module stop mode.

19.1.2 Block Diagram

Figure 19-1 shows a block diagram of PWM channel 1.

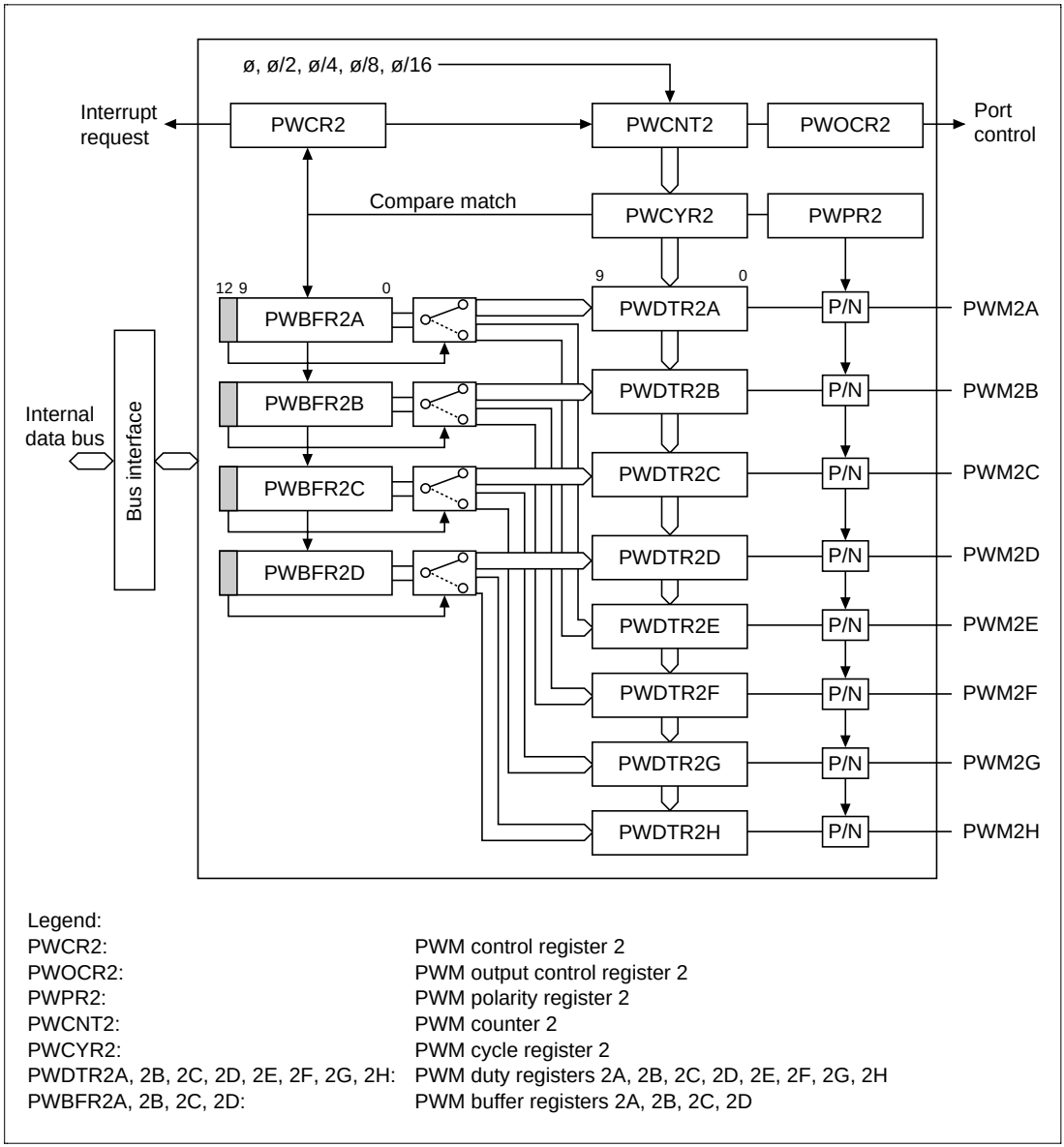


Legend:

PWCR1:	PWM control register 1
PWOCR1:	PWM output control register 1
PWPR1:	PWM polarity register 1
PWCNT1:	PWM counter 1
PWCYR1:	PWM cycle register 1
PWDTR1A, 1C, 1E, 1G:	PWM duty registers 1A, 1C, 1E, 1G
PWBFR1A, 1C, 1E, 1G:	PWM buffer registers 1A, 1C, 1E, 1G

Figure 19-1 Block Diagram of PWM Channel 1

Figure 19-2 shows a block diagram of PWM channel 2.



Legend:

PWCR2:	PWM control register 2
PWOCR2:	PWM output control register 2
PWPR2:	PWM polarity register 2
PWCNT2:	PWM counter 2
PWCYR2:	PWM cycle register 2
PWDTR2A, 2B, 2C, 2D, 2E, 2F, 2G, 2H:	PWM duty registers 2A, 2B, 2C, 2D, 2E, 2F, 2G, 2H
PWBFR2A, 2B, 2C, 2D:	PWM buffer registers 2A, 2B, 2C, 2D

Figure 19-2 Block Diagram of PWM Channel 2

19.1.3 Pin Configuration

Table 19-1 shows the PWM pin configuration.

Table 19-1 PWM Pin Configuration

Name	Abbrev.	I/O	Function
PWM output pin 1A	PWM1A	Output	Channel 1A PWM output
PWM output pin 1B	PWM1B	Output	Channel 1B PWM output
PWM output pin 1C	PWM1C	Output	Channel 1C PWM output
PWM output pin 1D	PWM1D	Output	Channel 1D PWM output
PWM output pin 1E	PWM1E	Output	Channel 1E PWM output
PWM output pin 1F	PWM1F	Output	Channel 1F PWM output
PWM output pin 1G	PWM1G	Output	Channel 1G PWM output
PWM output pin 1H	PWM1H	Output	Channel 1H PWM output
PWM output pin 2A	PWM2A	Output	Channel 2A PWM output
PWM output pin 2B	PWM2B	Output	Channel 2B PWM output
PWM output pin 2C	PWM2C	Output	Channel 2C PWM output
PWM output pin 2D	PWM2D	Output	Channel 2D PWM output
PWM output pin 2E	PWM2E	Output	Channel 2E PWM output
PWM output pin 2F	PWM2F	Output	Channel 2F PWM output
PWM output pin 2G	PWM2G	Output	Channel 2G PWM output
PWM output pin 2H	PWM2H	Output	Channel 2H PWM output

19.1.4 Register Configuration

Table 19-2 shows the register configuration of the PWM.

Table 19-2 PWM Registers

Channel	Name	Abbrev.	R/W	Initial Value	Address*
1	PWM control register 1	PWCR1	R/W	H'C0	H'FC00
	PWM output control register 1	PWOCR1	R/W	H'00	H'FC02
	PWM polarity register 1	PWPR1	R/W	H'00	H'FC04
	PWM cycle register 1	PWCYR1	R/W	H'FFFF	H'FC06
	PWM buffer register 1A	PWBFR1A	R/W	H'EC00	H'FC08
	PWM buffer register 1C	PWBFR1C	R/W	H'EC00	H'FC0A
	PWM buffer register 1E	PWBFR1E	R/W	H'EC00	H'FC0C
	PWM buffer register 1G	PWBFR1G	R/W	H'EC00	H'FC0E
2	PWM control register 2	PWCR2	R/W	H'C0	H'FC10
	PWM output control register 2	PWOCR2	R/W	H'00	H'FC12
	PWM polarity register 2	PWPR2	R/W	H'00	H'FC14
	PWM cycle register 2	PWCYR2	R/W	H'FFFF	H'FC16
	PWM buffer register 2A	PWBFR2A	R/W	H'EC00	H'FC18
	PWM buffer register 2B	PWBFR2B	R/W	H'EC00	H'FC1A
	PWM buffer register 2C	PWBFR2C	R/W	H'EC00	H'FC1C
	PWM buffer register 2D	PWBFR2D	R/W	H'EC00	H'FC1E
All	Module stop control register D	MSTPCRD	R/W	B'11******	H'FC60

Note: * Lower 16 bits of the address.

19.2 Register Descriptions

19.2.1 PWM Control Registers 1 and 2 (PWCR1, PWCR2)

Bit	7	6	5	4	3	2	1	0
	—	—	IE	CMF	CST	CKS2	CKS1	CKS0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W*	R/W	R/W	R/W	R/W

Note: * Only 0 can be written, to clear the flag.

PWCR is an 8-bit read/write register that performs interrupt enabling, starting/stopping, and counter (PWCNT) clock selection. It also contains a flag that indicates a compare match with the cycle register (PWCYR). PWCR1 is the channel 1 register, and PWCR2 is the channel 2 register.

PWCR is initialized to H'C0 upon reset, and in standby mode, watch mode*, subactive mode*, subsleep mode*, and module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Bits 7 and 6—Reserved: They are always read as 1 and cannot be modified.

Bit 5—Interrupt Enable (IE): Bit 5 selects enabling or disabling of an interrupt in the event of a compare match with the PWCYR register for the corresponding channel.

Bit 5: IE	Description	
0	Interrupt disabled	(Initial value)
1	Interrupt enabled	

Bit 4—Compare Match Flag (CMF): Bit 4 indicates the occurrence of a compare match with the PWCYR register for the corresponding channel.

Bit 4: CMF	Description	
0	[Clearing conditions] • When 0 is written to CMF after reading CMF = 1 • When the DTC is activated by a compare match interrupt, and the DISEL bit in the DTC's MRB register is 0	(Initial value)
1	[Setting condition] When PWCNT = PWCYR	

Bit 3—Counter Start (CST): Bit 3 selects starting or stopping of the PWCNT counter for the corresponding channel.

Bit 3: CST	Description	
0	PWCNT is stopped	(Initial value)
1	PWCNT is started	

Bits 2 to 0—Clock Select (CKS): Bits 2 to 0 select the clock for the PWCNT counter in the corresponding channel.

Bit 2: CKS2	Bit 1: CKS1	Bit 0: CKS0	Description	
0	0	0	Internal clock: counts on ø/1	(Initial value)
		1	Internal clock: counts on ø/2	
	1	0	Internal clock: counts on ø/4	
		1	Internal clock: counts on ø/8	
1	*	*	Internal clock: counts on ø/16	

*: Don't care

19.2.2 PWM Output Control Registers 1 and 2 (PWOCR1, PWOCR2)

PWOCR1

Bit	7	6	5	4	3	2	1	0
	OE1H	OE1G	OE1F	OE1E	OE1D	OE1C	OE1B	OE1A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWOCR2

Bit	7	6	5	4	3	2	1	0
	OE2H	OE2G	OE2F	OE2E	OE2D	OE2C	OE2B	OE2A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWOCR is an 8-bit read/write register that enables or disables PWM output. PWOCR1 controls outputs PWM1H to PWM1A, and PWOCR2 controls outputs PWM2H to PWM2A.

PWOCR is initialized to H'00 upon reset, and in standby mode, watch mode*, subactive mode*, subsleep mode*, and module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Bits 7 to 0—Output Enable (OE): Each of these bits enables or disables the corresponding PWM output.

Bits 7 to 0:

OE	Description	
0	PWM output is disabled	(Initial value)
1	PWM output is enabled	

19.2.3 PWM Polarity Registers 1 and 2 (PWPR1, PWPR2)

PWPR1

Bit	7	6	5	4	3	2	1	0
	OPS1H	OPS1G	OPS1F	OPS1E	OPS1D	OPS1C	OPS1B	OPS1A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWPR2

Bit	7	6	5	4	3	2	1	0
	OPS2H	OPS2G	OPS2F	OPS2E	OPS2D	OPS2C	OPS2B	OPS2A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWPR is an 8-bit read/write register that selects the PWM output polarity. PWPR1 controls outputs PWM1H to PWM1A, and PWPR2 controls outputs PWM2H to PWM2A.

PWPR is initialized to H'00 upon reset, and in standby mode, watch mode*, subactive mode*, subsleep mode*, and module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Bits 7 to 0—Output Polarity Select (OPS): Each of these bits selects the polarity of the corresponding PWM output.

Bits 7 to 0: OPS	Description	
0	PWM direct output	(Initial value)
1	PWM inverse output	

19.2.4 PWM Counters 1 and 2 (PWCNT1, PWCNT2)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—										
Initial value	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

PWCNT is a 10-bit up-counter incremented by the input clock. The input clock is selected by clock select bits 2 to 0 (CKS2 to CKS0) in PWCR.

PWCNT1 is used as the channel 1 time base, and PWCNT2 as the channel 2 time base.

PWCNT is initialized to H'FC00 when the counter start bit (CST) in PWCR is cleared to 0, and also upon reset and in standby mode, watch mode*, subactive mode*, subsleep mode*, and module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

19.2.5 PWM Cycle Registers 1 and 2 (PWCYR1, PWCYR2)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—										
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PWCYR is a 16-bit read/write register that sets the PWM conversion cycle. When a PWCYR compare match occurs, PWCNT is cleared and data is transferred from the buffer register (PWBFR) to the duty register (PWDTR). PWCYR1 is used for the channel 1 conversion cycle setting, and PWCYR2 for the channel 2 conversion cycle setting.

PWCYR should be written to only while PWCNT is stopped. A value of H'FC00 must not be set.

PWCYR is initialized to H'FFFF upon reset, and in standby mode, watch mode*, subactive mode*, subsleep mode*, and module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

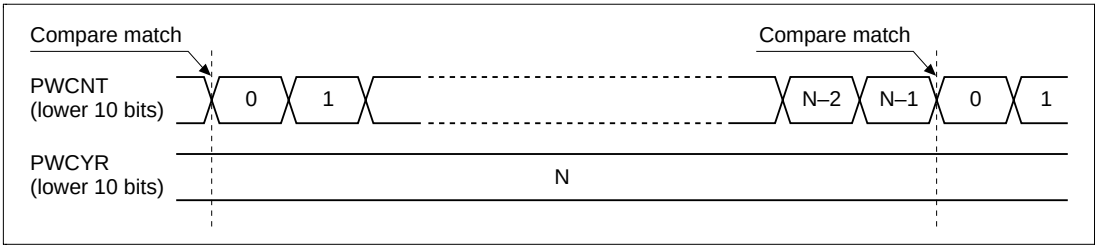


Figure 19-3 Cycle Register Compare Match

19.2.6 PWM Duty Registers 1A, 1C, 1E, 1G (PWDTR1A, 1C, 1E, 1G)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	OTS	—	—	DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
Initial value	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

There are four PWDTR1x registers (PWDTR1A, 1C, 1E, 1G). PWDTR1A is used for outputs PWM1A and PWM1B, PWDTR1C for outputs PWM1C and PWM1D, PWDTR1E for outputs PWM1E and PWM1F, and PWDTR1G for outputs PWM1G and PWM1H.

PWDTR1 cannot be read or written to directly. When a PWCYR1 compare match occurs, data is transferred from buffer register 1 (PWBFR1) to PWDTR1.

PWDTR1x is initialized to H'EC00 when the counter start bit (CST) in PWCR1 is cleared to 0, and also upon reset and in standby mode, watch mode*, subactive mode*, subsleep mode*, and module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Bits 15 to 13—Reserved: These bits cannot be read from or written to.

Bit 12—Output Terminal Select (OTS): Bit 12 selects the pin used for PWM output according to the value in bit 12 in the buffer register that is transferred by a PWCYR1 compare match. Unselected pins output a low level (or a high level when the corresponding bit in PWPR1 is set to 1).

Register	Bit 12: OTS	Description	
PWDTR1A	0	PWM1A output selected	(Initial value)
	1	PWM1B output selected	
PWDTR1C	0	PWM1C output selected	(Initial value)
	1	PWM1D output selected	
PWDTR1E	0	PWM1E output selected	(Initial value)
	1	PWM1F output selected	
PWDTR1G	0	PWM1G output selected	(Initial value)
	1	PWM1H output selected	

Bits 11 and 10—Reserved: These bits cannot be read from or written to.

Bits 9 to 0—Duty (DT): Bits 9 to 0 set the PWM output duty according to the values in bits 9 to 0 in the buffer register that is transferred by a PWCYR1 compare match. A high level (or a low level when the corresponding bit in PWPR1 is set to 1) is output from the time PWCNT1 is cleared by a PWCYR1 compare match until a PWDTR1 compare match occurs. When all the bits are 0, there is no high-level output period (no low-level output period when the corresponding bit in PWPR1 is set to 1).

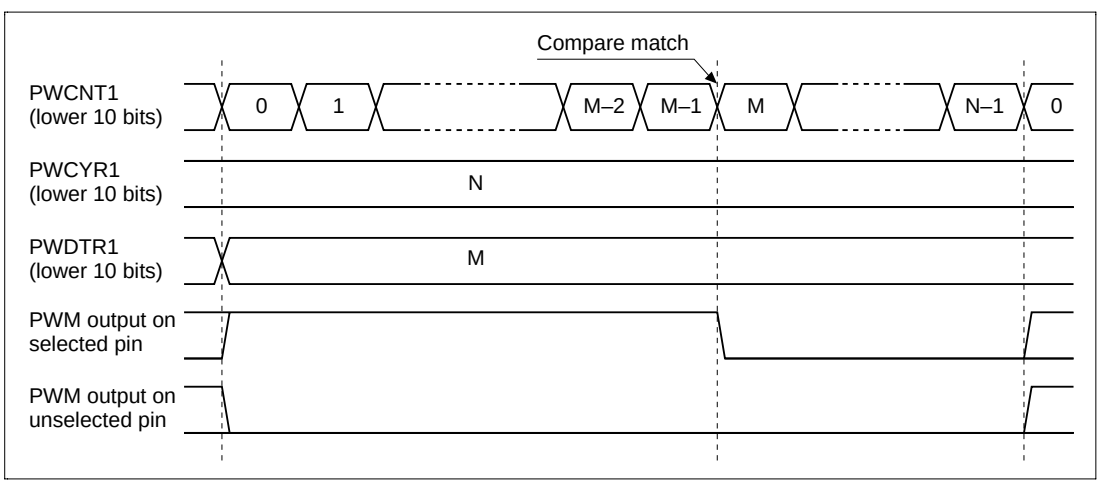


Figure 19-4 Duty Register Compare Match (OPS = 0 in PWPR1)

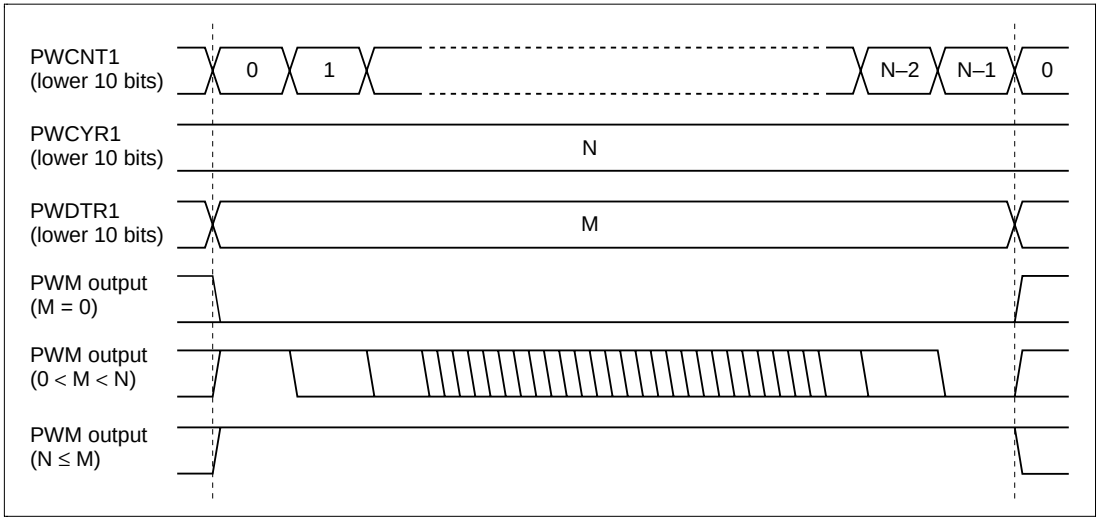


Figure 19-5 Differences in PWM Output According to Duty Register Set Value (OPS = 0 in PWPR1)

19.2.7 PWM Buffer Registers 1A, 1C, 1E, 1G (PWBFR1A, 1C, 1E, 1G)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	OTS	—	—	DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
Initial value	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
Read/Write	—	—	—	R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

There are four 16-bit read/write PWBFR1 registers (PWBFR1A, 1C, 1E, 1G). When a PWCYR1 compare match occurs, data is transferred from PWBFR1A to PWDTR1A, from PWBFR1C to PWDTR1C, from PWBFR1E to PWDTR1E, and from PWBFR1G to PWDTR1G.

PWBFR1 is initialized to H'EC00 upon reset, and in standby mode, watch mode*, subactive mode*, subsleep mode*, and module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Bits 15 to 13—Reserved: They are always read as 1 and cannot be modified.

Bit 12—Output Terminal Select (OTS): Bit 12 is the data transferred to bit 12 of PWDTR1.

Bits 11 and 10—Reserved: They are always read as 1 and cannot be modified.

Bits 9 to 0—Duty (DT): Bits 9 to 0 comprise the data transferred to bits 9 to 0 in PWDTR1.

19.2.8 PWM Duty Registers 2A to 2H (PWDTR2A to PWDTR2H)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
Initial value	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
Read/Write	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

There are eight PWDTR2 registers (PWDTR2A to PWDTR2H). PWDTR2A is used for output PWM2A, PWDTR2B for output PWM2B, PWDTR2C for output PWM2C, PWDTR2D for output PWM2D, PWDTR2E for output PWM2E, PWDTR2F for output PWM2F, PWDTR2G for output PWM2G, and PWDTR2H for output PWM2H.

PWDTR2 cannot be read or written to directly. When a PWCYR2 compare match occurs, data is transferred from buffer register 2 (PWBFR2) to PWDTR2.

PWDTR2 is initialized to H'EC00 when the counter start bit (CST) in PWCR2 is cleared to 0, and also upon reset and in standby mode, watch mode*, subactive mode*, subsleep mode*, and module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Bits 15 to 10—Reserved: These bits cannot be read from or written to.

Bits 9 to 0—Duty (DT): Bits 9 to 0 set the PWM output duty according to the values in bits 9 to 0 in the buffer register that is transferred by a PWCYR2 compare match. A high level (or a low level when the corresponding bit in PWPR2 is set to 1) is output from the time PWCNT2 is cleared by a PWCYR2 compare match until a PWDTR2 compare match occurs. When all the bits are 0, there is no high-level output period (no low-level output period when the corresponding bit in PWPR2 is set to 1).

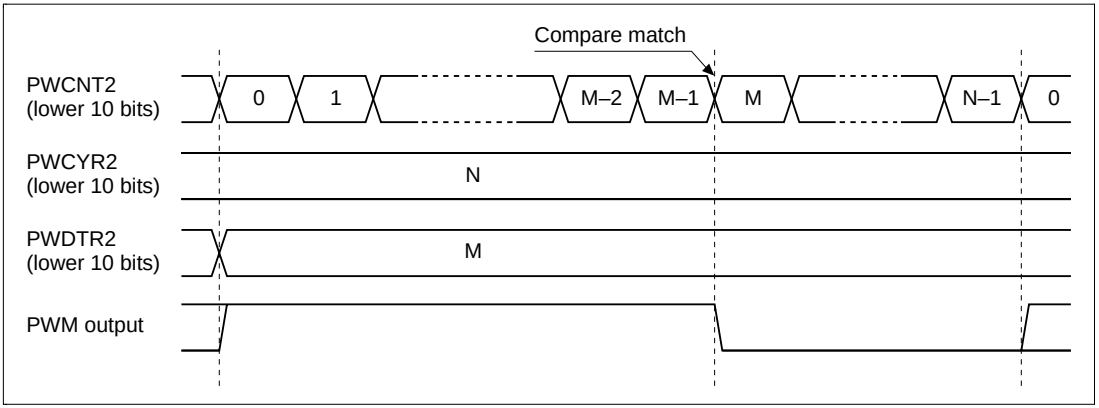


Figure 19-6 Duty Register Compare Match (OPS = 0 in PWPR2)

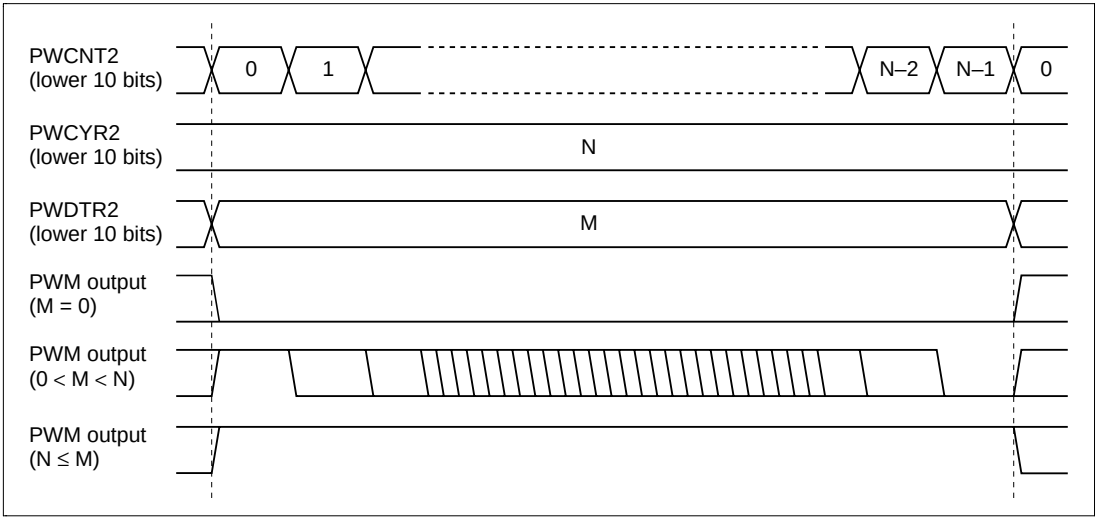


Figure 19-7 Differences in PWM Output According to Duty Register Set Value (OPS = 0 in PWPR2)

19.2.9 PWM Buffer Registers 2A to 2D (PWBFR2A to PWBFR2D)

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TDS	—	—	DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
Initial value	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
Read/Write	—	—	—	R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

There are four 16-bit read/write PWBFR2 registers (PWBFR2A to PWBFR2D). When a PWCYR2 compare match occurs, data is transferred from PWBFR2A to PWDTR2A or PWDTR2E, from PWBFR2B to PWDTR2B or PWDTR2F, from PWBFR2C to PWDTR2C or

PWDTR2G, and from PWBFR2D to PWDTR2D or PWDTR2H. The transfer destination is determined by the value of the TDS bit.

PWBFR2 is initialized to H'EC00 upon reset, and in standby mode, watch mode*, subactive mode*, subsleep mode*, and module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Bits 15 to 13—Reserved: They are always read as 1 and cannot be modified.

Bit 12—Transfer Destination Select (TDS): Bit 12 selects the PWDTR2 register to which data is to be transferred.

Register	Bit 12: TDS	Description	
PWBFR2A	0	PWDTR2A selected	(Initial value)
	1	PWDTR2E selected	
PWBFR2B	0	PWDTR2B selected	(Initial value)
	1	PWDTR2F selected	
PWBFR2C	0	PWDTR2C selected	(Initial value)
	1	PWDTR2G selected	
PWBFR2D	0	PWDTR2D selected	(Initial value)
	1	PWDTR2H selected	

Bits 11 and 10—Reserved: They are always read as 1 and cannot be modified.

Bits 9 to 0—Duty (DT): Bits 9 to 0 comprise the data transferred to bits 9 to 0 in PWDTR2.

19.2.10 Module Stop Control Register D (MSTPCRD)

Bit	7	6	5	4	3	2	1	0
	MSTPD7	MSTPD6	MSTPD5	MSTPD4	MSTPD3	MSTPD2	MSTPD1	MSTPD0
Initial value	1	1	undefined	undefined	undefined	undefined	undefined	undefined
Read/Write	R/W	R/W	—	—	—	—	—	—

MSTPCRD is an 8-bit read/write register that performs module stop mode control.

When the MSTPD7 bit is set to 1, PWM timer operation is stopped at the end of the bus cycle, and module stop mode is entered. For details, see section 23A.5, 23B.5, Module Stop Mode.

MSTPCRD is initialized by a reset and in hardware standby mode. It is not initialized by a manual reset or in software standby mode.

Bit 7—Module Stop (MSTPD7): Bit 7 specifies the PWM module stop mode.

Bit 7: MSTPD7	Description	
0	PWM module stop mode is cleared	
1	PWM module stop mode is set	(Initial value)

19.3 Bus Master Interface

19.3.1 16-Bit Data Registers

PWCYR1/2, PWBFR1A/C/E/G, and PWBFR2A/B/C/D are 16-bit registers. These registers are linked to the bus master by a 16-bit data bus, and can be read or written in 16-bit units. They cannot be read by 8-bit access; 16-bit access must always be used.

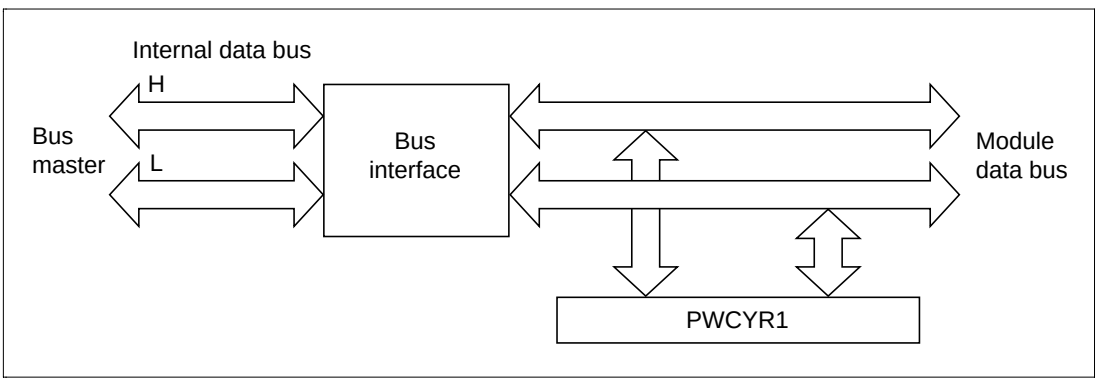


Figure 19-8 16-Bit Register Access Operation (Bus Master ↔ PWCYR1 (16 Bits))

19.3.2 8-Bit Data Registers

PWCR1/2, PWOCR1/2, and PWPR1/2 are 8-bit registers that can be read and written to in 8-bit units. These registers are linked to the bus master by a 16-bit data bus, and can be read or written by 16-bit access; in this case, the lower 8 bits will always be read as H'FF.

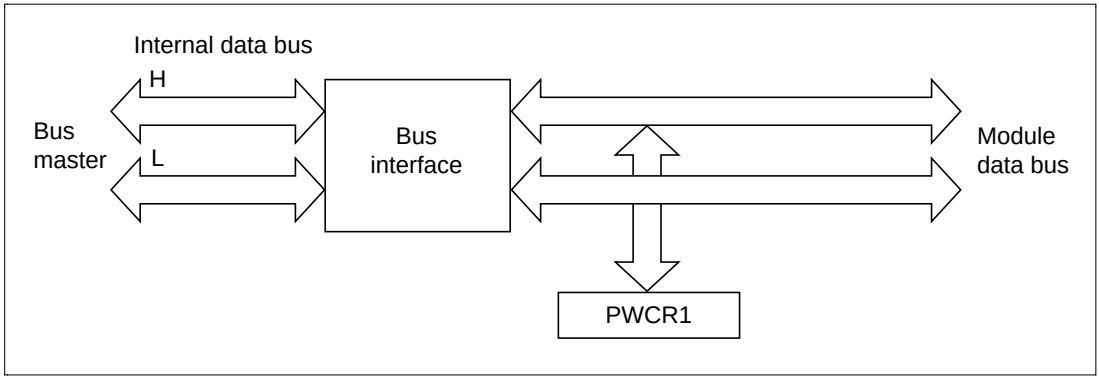


Figure 19-9 8-Bit Register Access Operation (Bus Master ↔ PWCR1 (Upper 8 Bits))

19.4 Operation

19.4.1 PWM Channel 1 Operation

PWM waveforms are output from pins PWM1A to PWM1H as shown in figure 19-10.

Initial Settings: Set the PWM output polarity in PWPR1; enable the pins for PWM output with PWOCR1; select the clock to be input to PWCNT1 with bits CKS2 to CKS0 in PWCR1; set the PWM conversion cycle in PWCYR1; and set the first frame of data in PWBFR1A, PWBFR1C, PWBFR1E, and PWBFR1G.

Activation: When the CST bit in PWCR1 is set to 1, a compare match between PWCNT1 and PWCYR1 is generated. Data is transferred from PWBFR1A to PWDTR1A, from PWBFR1C to PWDTR1C, from PWBFR1E to PWDTR1E, and from PWBFR1G to PWDTR1G. PWCNT1 starts counting up. At the same time the CMF bit in PWCR1 is set, so that, if the IE bit in PWCR1 has been set, an interrupt can be requested or the DTC can be activated.

Waveform Output: The PWM outputs selected by the OTS bits in PWDTR1A/C/E/G go high when a compare match occurs between PWCNT1 and PWCYR1. The PWM outputs not selected by the OTS bits are low. When a compare match occurs between PWCNT1 and PWDTR1A/C/E/G, the corresponding PWM output goes low. If the corresponding bit in PWPR1 is set to 1, the output is inverted.

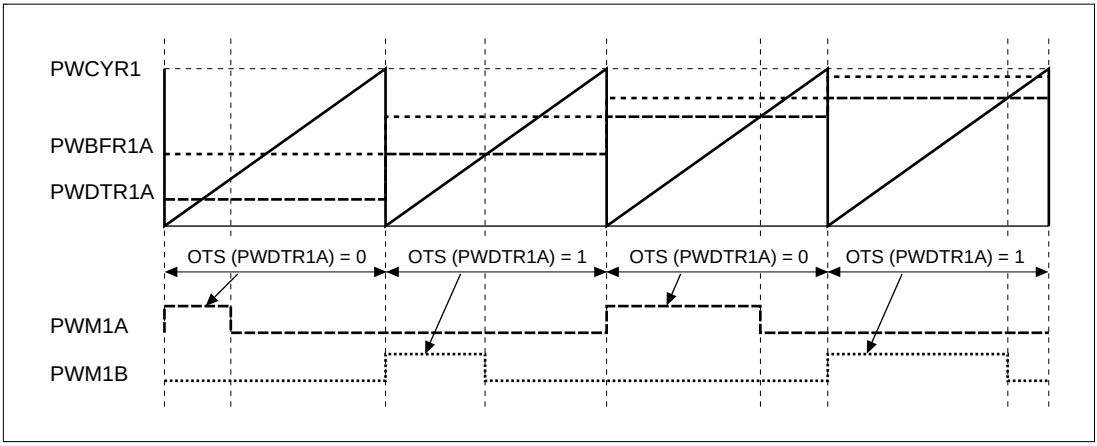


Figure 19-10 PWM Channel 1 Operation

Next Frame: When a compare match occurs between PWCNT1 and PWCYR1, data is transferred from PWBFR1A to PWDTR1A, from PWBFR1C to PWDTR1C, from PWBFR1E to PWDTR1E, and from PWBFR1G to PWDTR1G. PWCNT1 is reset and starts counting up from H'000. The CMF bit in PWCR1 is set, and if the IE bit in PWCR1 has been set, an interrupt can be requested or the DTC can be activated.

Stopping: When the CST bit in PWCR1 is cleared to 0, PWCNT1 is reset and stops. All PWM outputs go low (or high if the corresponding bit in PWPR1 is set to 1).

19.4.2 PWM Channel 2 Operation

PWM waveforms are output from pins PWM2A to PWM2H as shown in figure 19-11.

Initial Settings: Set the PWM output polarity in PWPR2; enable the pins for PWM output with PWOCR2; select the clock to be input to PWCNT2 with bits CKS2 to CKS0 in PWCR2; set the PWM conversion cycle in PWCYR2; and set the first frame of data in PWBFR2A, PWBFR2B, PWBFR2C, and PWBFR2D.

Activation: When the CST bit in PWCR2 is set to 1, a compare match between PWCNT2 and PWCYR2 is generated. Data is transferred from PWBFR2A to PWDTR2A or PWDTR2E, from PWBFR2B to PWDTR2B or PWDTR2F, from PWBFR2C to PWDTR2C or PWDTR2G, and from PWBFR2D to PWDTR2D or PWDTR2H, according to the value of the TDS bit. PWCNT2 starts counting up. At the same time the CMF bit in PWCR2 is set, so that, if the IE bit in PWCR2 has been set, an interrupt can be requested or the DTC can be activated.

Waveform Output: The PWM outputs go high when a compare match occurs between PWCNT2 and PWCYR2. When a compare match occurs between PWCNT2 and PWDTR2A-H, the corresponding PWM output goes low. If the corresponding bit in PWPR2 is set to 1, the output is inverted.

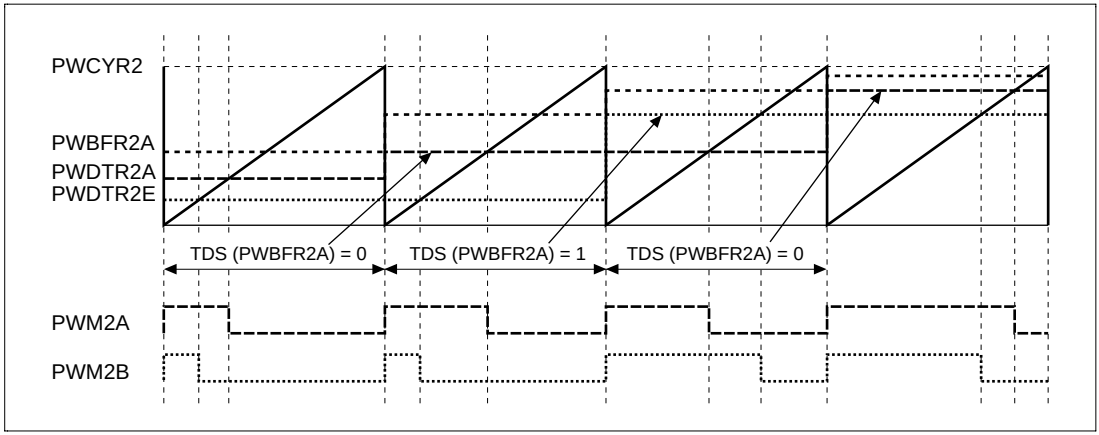


Figure 19-11 PWM Channel 2 Operation

Next Frame: When a compare match occurs between PWCNT2 and PWCYR2, data is transferred from PWBFR2A to PWDTR2A or PWDTR2E, from PWBFR2B to PWDTR2B or PWDTR2F, from PWBFR2C to PWDTR2C or PWDTR2G, and from PWBFR2D to PWDTR2D or PWDTR2H, according to the value of the TDS bit. PWCNT2 is reset and starts counting up from H'000. The CMF bit in PWCR2 is set, and if the IE bit in PWCR2 has been set, an interrupt can be requested or the DTC can be activated.

Stopping: When the CST bit in PWCR2 is cleared to 0, PWCNT2 is reset and stops. PWDTR2A to PWDTR2H are reset. All PWM outputs go low (or high if the corresponding bit in PWPR2 is set to 1).

19.5 Usage Note

Contention between Buffer Register Write and Compare Match

If a PWBFR write is performed in the state immediately after a cycle register compare match, the PWM output does not change, but as the duty register is also rewritten at the same time as the buffer register, normal PWM output will not be achieved.

If a PWBFR write is performed in the state immediately after a cycle register compare match, the buffer register and duty register are overwritten. PWM output changed by the cycle register compare match is not changed in the overwrite of the duty register due to contention. This may result in unanticipated duty output. In the case of channel 2, the duty register used as the transfer destination is selected by the TDS bit of the buffer register when an overwrite of the duty register occurs due to contention. This can also result in an unintended overwrite of the duty register.

Buffer register rewriting must be completed before automatic transfer by the DTC (data transfer controller), exception handling due to a compare match interrupt, or the occurrence of a cycle register compare match on detection of the rise of CMF (compare match flag) in PWCR.

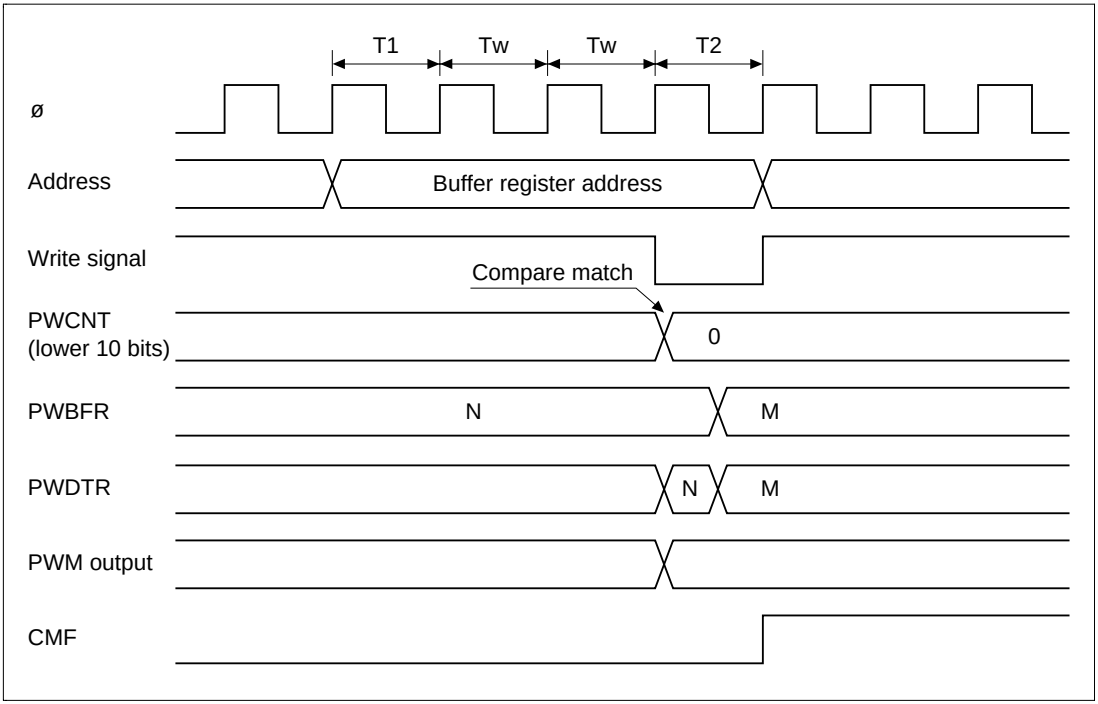


Figure 19-12 PWM Channel 1 Operation

20.1 Overview

The H8S/2636 has 4 kbytes, and H8S/2638 and H8S/2639 have 16 kbytes of on-chip high-speed static RAM. The RAM is connected to the CPU by a 16-bit data bus, enabling one-state access by the CPU to both byte data and word data. This makes it possible to perform fast word data transfer.

The on-chip RAM can be enabled or disabled by means of the RAM enable bit (RAME) in the system control register (SYSCR).

20.1.1 Block Diagram

Figure 20-1 shows a block diagram of the on-chip RAM.

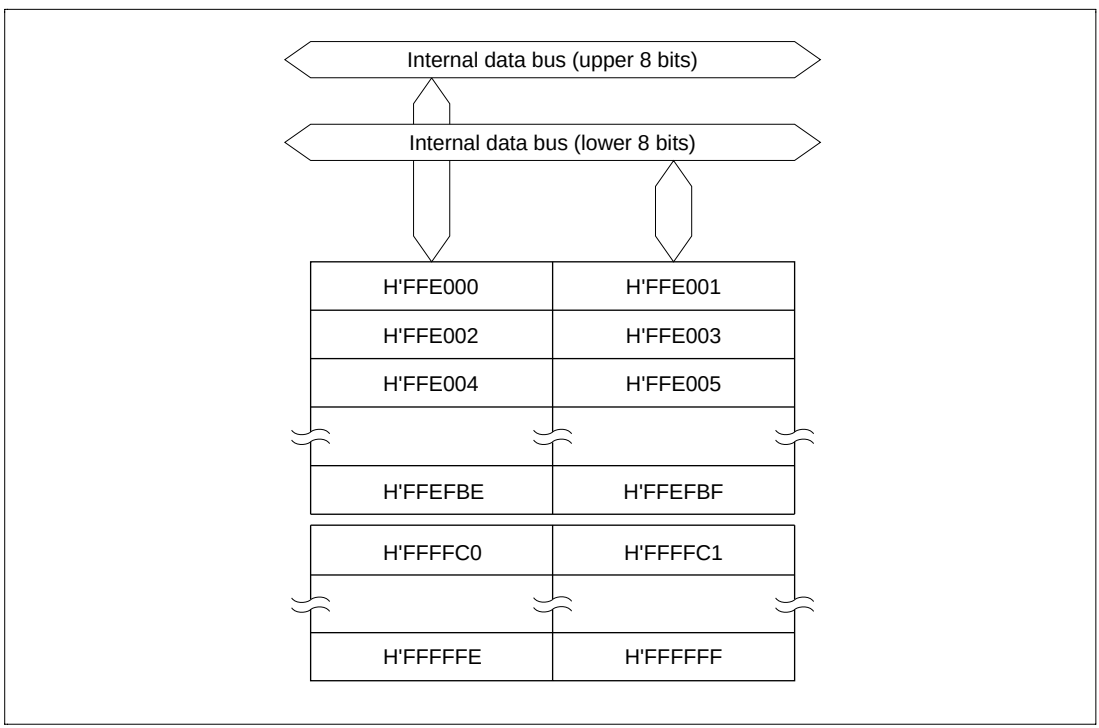


Figure 20-1 (a) Block Diagram of RAM (H8S/2636)

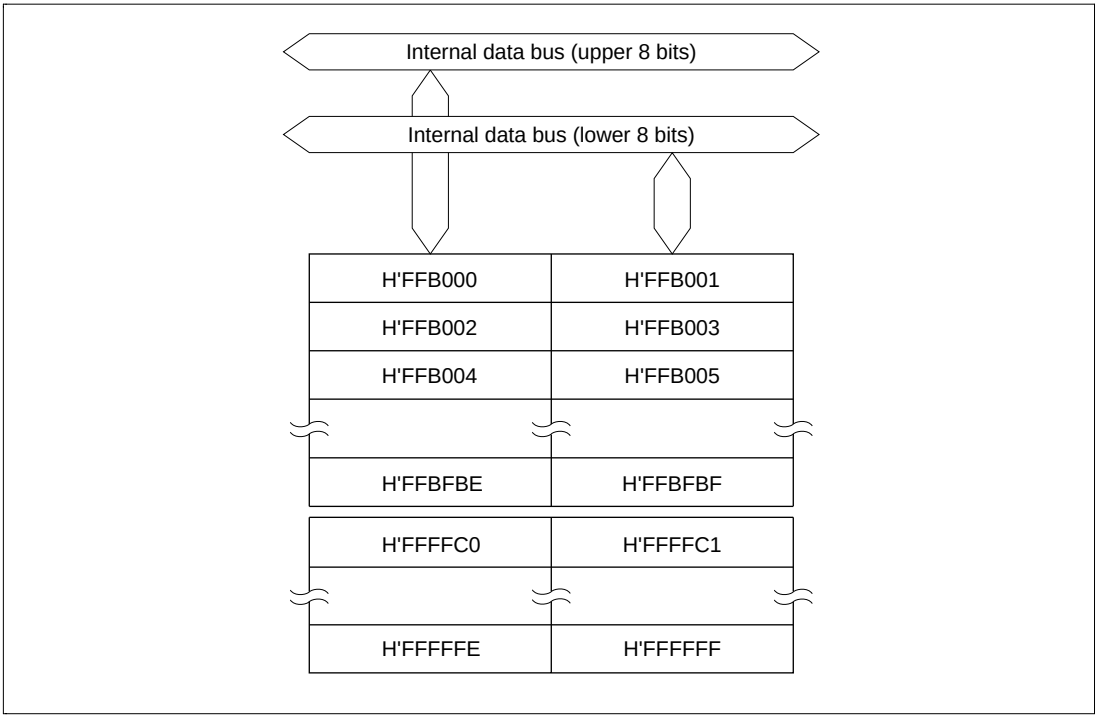


Figure 20-1 (b) Block Diagram of RAM (H8S/2638 and H8S/2639)

20.1.2 Register Configuration

The on-chip RAM is controlled by SYSCR. Table 20-1 shows the address and initial value of SYSCR.

Table 20-1 RAM Register

Name	Abbreviation	R/W	Initial Value	Address*
System control register	SYSCR	R/W	H'01	H'FDE5

Note: * Lower 16 bits of the address.

20.2 Register Descriptions

20.2.1 System Control Register (SYSCR)

Bit	:	7	6	5	4	3	2	1	0
		MACS	—	INTM1	INTM0	NMIEG	—	—	RAME
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	—	R/W	R/W	R/W	R/W	—	R/W

The on-chip RAM is enabled or disabled by the RAME bit in SYSCR. For details of other bits in SYSCR, see section 3.2.2, System Control Register (SYSCR).

Bit 0—RAM Enable (RAME): Enables or disables the on-chip RAM. The RAME bit is initialized when the reset state is released. It is not initialized in software standby mode.

Bit 0

RAME	Description	
0	On-chip RAM is disabled	
1	On-chip RAM is enabled	(Initial value)

20.3 Operation

When the RAME bit is set to 1, accesses to addresses H'FFE000 to H'FFEFBF (for the H8S/2636), and H'FFB000 to H'FFEFBF (for the H8S/2638 and H8S/2639) and H'FFFFC0 to H'FFFFFF in the chip are directed to the on-chip RAM. When the RAME bit is cleared to 0, the off-chip address space is accessed.

Since the on-chip RAM is connected to the CPU by an internal 16-bit data bus, it can be written to and read in byte or word units. Each type of access can be performed in one state.

Even addresses use the upper 8 bits, and odd addresses use the lower 8 bits. Word data must start at an even address.

20.4 Usage Notes

When Using the DTC: DTC register information can be located in addresses H'FFEBC0 to H'FFEFBF. When the DTC is used, the RAME bit must not be cleared to 0.

Reserved Areas: Addresses H'FFB000 to H'FFDFFF in the H8S/2636 are reserved areas that cannot be read or written to. When the RAME bit is cleared to 0, the off-chip address space is accessed.

21A.1 Overview

The H8S/2636 has 128 kbytes of on-chip flash memory, or 128 kbytes of on-chip mask ROM. The ROM is connected to the bus master via a 16-bit data bus, enabling both byte and word data to be accessed in one state. Instruction fetching is thus speeded up, and processing speed increased.

The on-chip ROM is enabled and disabled by setting the mode pins (MD2 to MD0).

The flash memory version can be erased and programmed on-board, as well as with a special-purpose PROM programmer.

21A.1.1 Block Diagram

Figure 21A-1 shows a block diagram of 128-kbyte ROM.

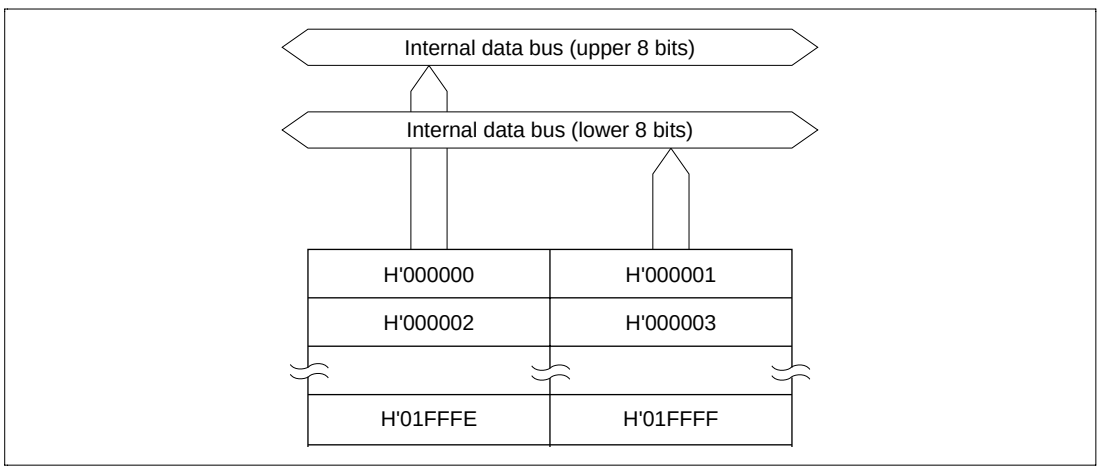


Figure 21A-1 Block Diagram of ROM (128 kbytes)

21A.1.2 Register Configuration

The H8S/2636 operating mode is controlled by the mode pins and the BCRL register. The register configuration is shown in table 21A-1.

Table 21A-1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R/W	Undefined	H'FDE7

Note: * Lower 16 bits of the address.

21A.2 Register Descriptions

21A.2.1 Mode Control Register (MDCR)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	MDS2	MDS1	MDS0
Initial value:	1	0	0	0	0	—*	—*	—*
R/W:	R/W	—	—	—	—	R	R	R

Note: * Determined by pins MD2 to MD0.

MDCR is an 8-bit read-only register used to monitor the current H8/2633 Series operating mode.

Bit 7—Reserved: Only 1 should be written to this bit.

Bits 6 to 3—Reserved: Read-only bits, always read as 0.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to pins MD2 to MD0. MDS2 to MDS0 are read-only bits, and cannot be modified. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are canceled by a power-on reset, but are retained in a manual reset.

21A.3 Operation

The on-chip ROM is connected to the CPU by a 16-bit data bus, and both byte and word data can be accessed in one state. Even addresses are connected to the upper 8 bits, and odd addresses to the lower 8 bits. Word data must start at an even address.

The on-chip ROM is enabled and disabled by setting the mode pins (MD2, MD1, and MD0). These settings are shown in table 21A-2.

Table 21A-2 Operating Modes and ROM (F-ZTAT Version)

		Mode Pins				
	Operating Mode	**FWE**	**MD2**	**MD1**	**MD0**	**On-Chip ROM**
Mode 0	—	0	0	0	0	—
Mode 1					1	
Mode 2				1	0	
Mode 3					1	
Mode 4	Advanced expanded mode with on-chip ROM disabled		1	0	0	Disabled
Mode 5	Advanced expanded mode with on-chip ROM disabled				1	
Mode 6	Advanced expanded mode with on-chip ROM enabled			1	0	Enabled (128 kbytes)
Mode 7	Advanced single-chip mode				1	Enabled (128 kbytes)
Mode 8	—	1	0	0	0	—
Mode 9					1	
Mode 10	Boot mode (advanced expanded mode with on-chip ROM enabled)*1			1	0	Enabled (128 kbytes)
Mode 11	Boot mode (advanced single-chip mode)*2				1	Enabled (128 kbytes)
Mode 12	—		1	0	0	—
Mode 13					1	
Mode 14	User program mode (advanced expanded mode with on-chip ROM enabled)*1			1	0	Enabled (128 kbytes)
Mode 15	User program mode (advanced single-chip mode)*2				1	Enabled (128 kbytes)

Notes: *1 Apart from the fact that flash memory can be erased and programmed, operation is the same as in advanced expanded mode with on-chip ROM enabled.

*2 Apart from the fact that flash memory can be erased and programmed, operation is the same as in advanced single-chip mode.

Table 21A-3 Operating Modes and ROM (Mask ROM Version)

		Mode Pins			
	Operating Mode	**MD2**	**MD1**	**MD0**	**On-Chip ROM**
Mode 0	—	0	0	0	—
Mode 1				1	
Mode 2			1	0	
Mode 3				1	
Mode 4	Advanced expanded mode with on-chip ROM disabled	1	0	0	Disabled
Mode 5	Advanced expanded mode with on-chip ROM disabled			1	
Mode 6	Advanced expanded mode with on-chip ROM enabled		1	0	Enabled (128 kbytes)
Mode 7	Advanced single-chip mode			1	Enabled (128 kbytes)

21A.4 Flash Memory Overview

The H8S/2636 has 128 kbytes of on-chip flash memory, or 128 kbytes of on-chip mask ROM. The features of the flash memory are summarized below.

- Four flash memory operating modes
  - — Program mode
  - — Erase mode
  - — Program-verify mode
  - — Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Block erase (in single-block units) can be performed. To erase the entire flash memory, each block must be erased in turn. Blocks of 1 kbyte, 8 kbytes, 16 kbytes, 28 kbytes, and 32 kbytes can be erased as required on the H8S/2636.
- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 128-byte programming, equivalent to 80 µs (typ.) per byte, and the erase time is 100 ms (typ.).
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - — Boot mode
  - — User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation in RAM

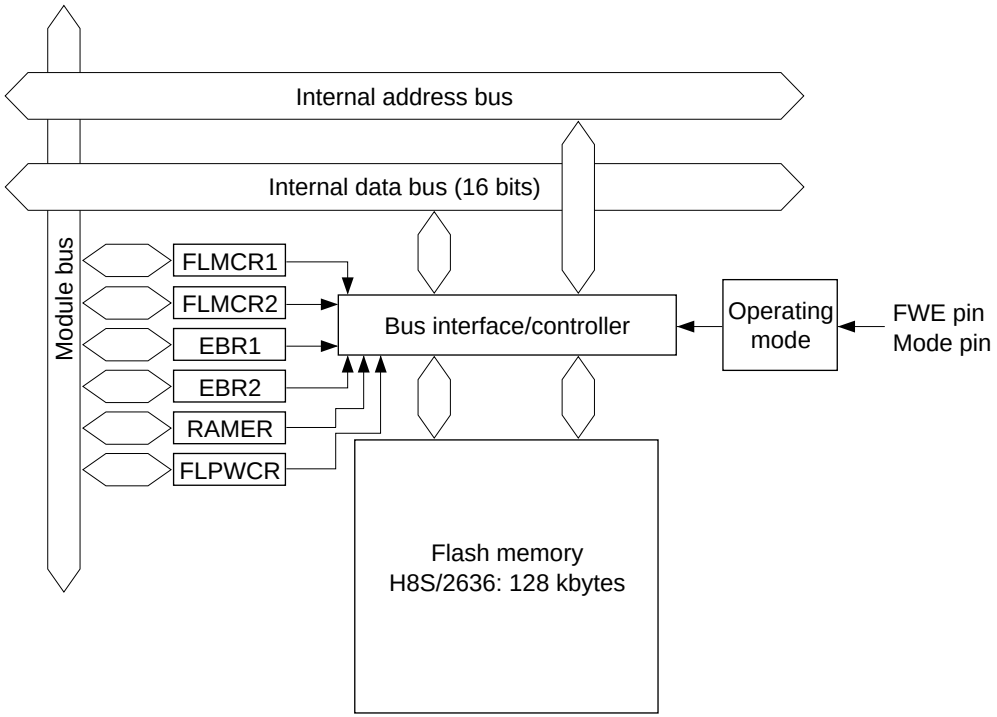
  Flash memory programming can be emulated in real time by overlapping a part of RAM onto flash memory.
- Protect modes

  There are two protect modes, hardware and software, which allow protected status to be designated for flash memory program/erase/verify operations.
- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

21A.5 Overview

21A.5.1 Block Diagram



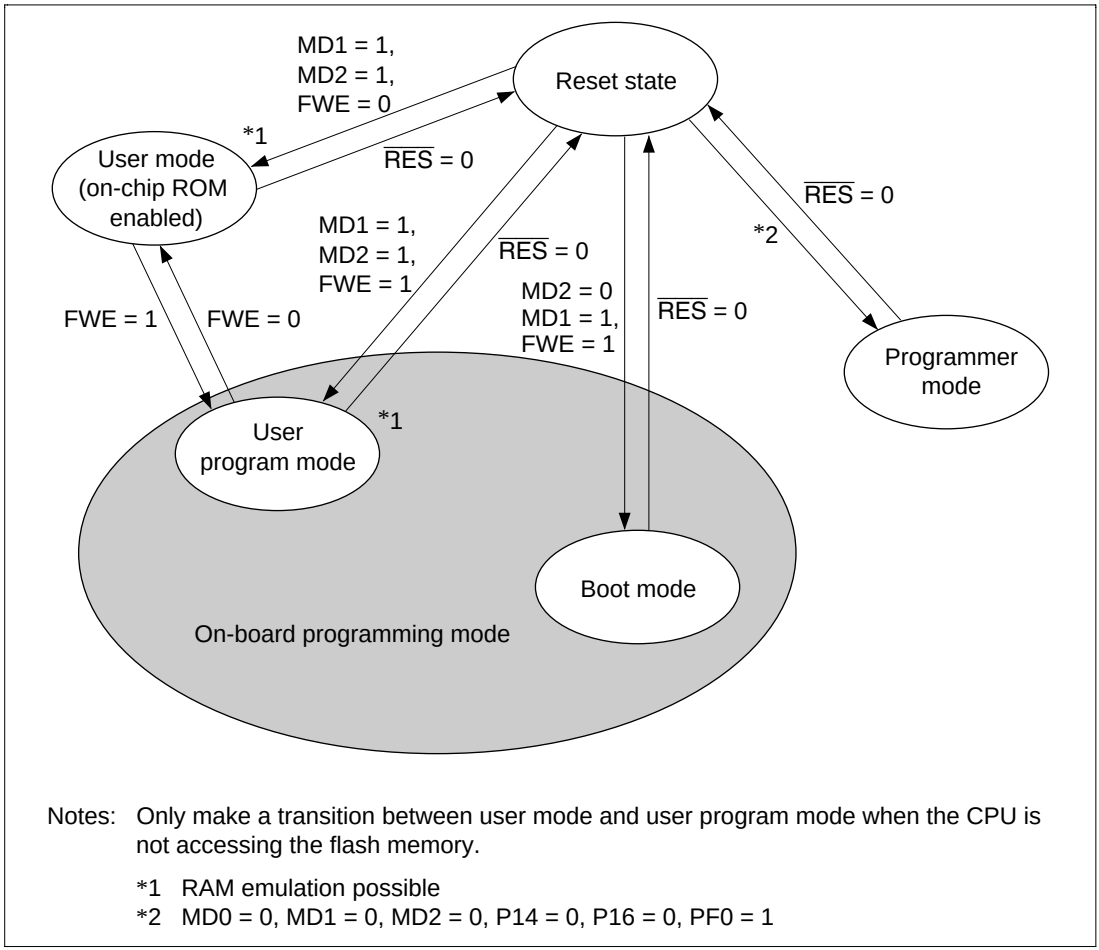
Legend
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2
RAMER: RAM emulation register
FLPWCR: Flash memory power control register

Figure 21A-2 Block Diagram of Flash Memory

21A.5.2 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, the microcomputer enters an operating mode as shown in figure 21A-3. In user mode, flash memory can be read but not programmed or erased.

The boot, user program and programmer modes are provided as modes to write and erase the flash memory.



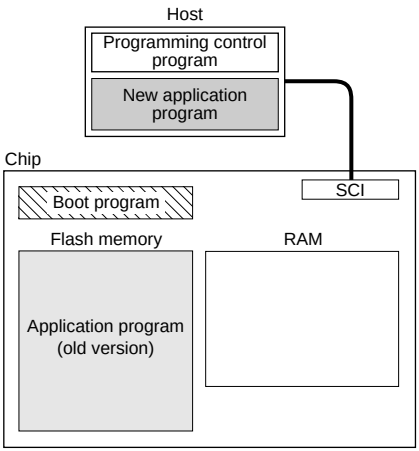
Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

*1 RAM emulation possible
*2 MD0 = 0, MD1 = 0, MD2 = 0, P14 = 0, P16 = 0, PF0 = 1

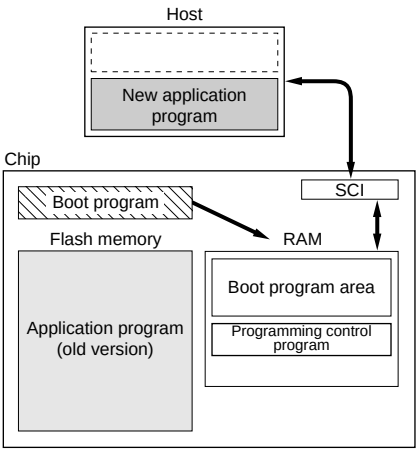
Figure 21A-3 Flash Memory State Transitions

Boot Mode

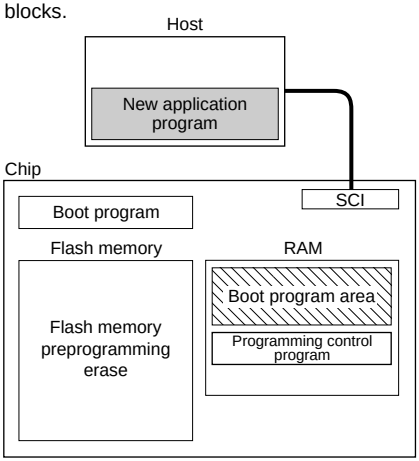
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



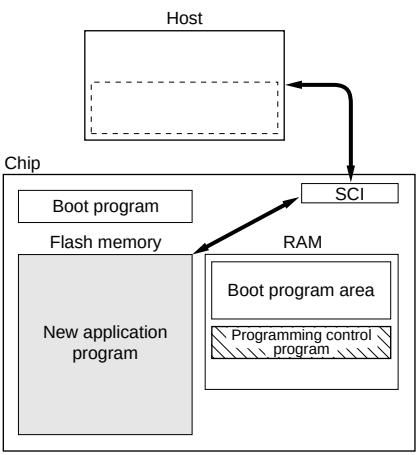
2. Programming control program transfer
   When boot mode is entered, the boot program in the chip (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.

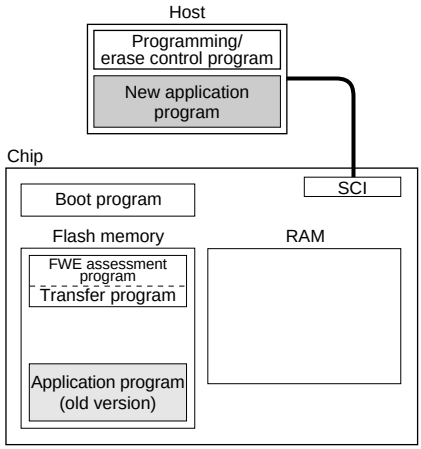


4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

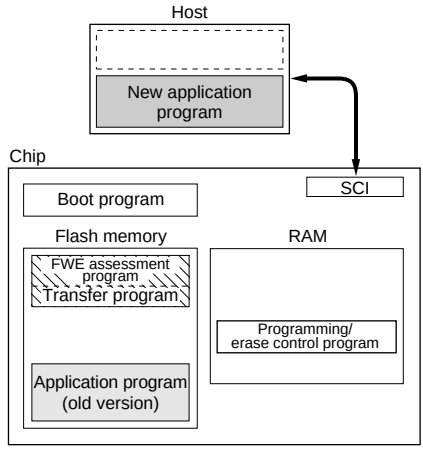


Program execution state

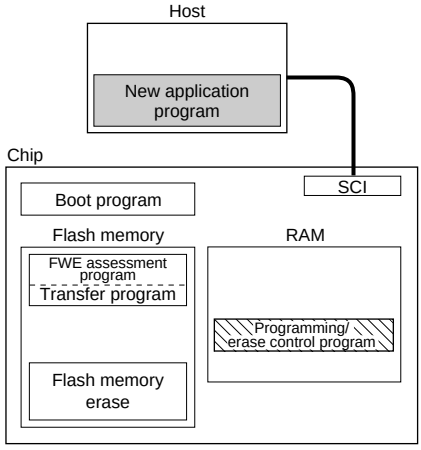
1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



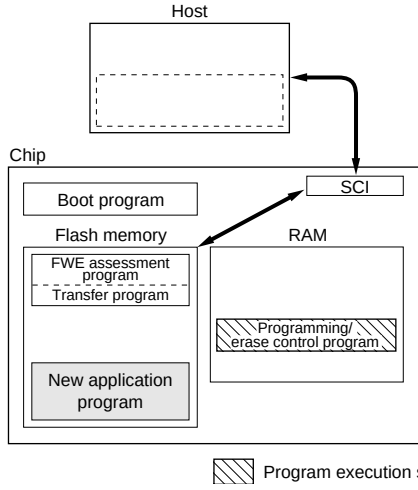
2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.



3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



21A.5.4 Flash Memory Emulation in RAM

Emulation should be performed in user mode or user program mode. When the emulation block set in RAMER is accessed while the emulation function is being executed, data written in the overlap RAM is read.

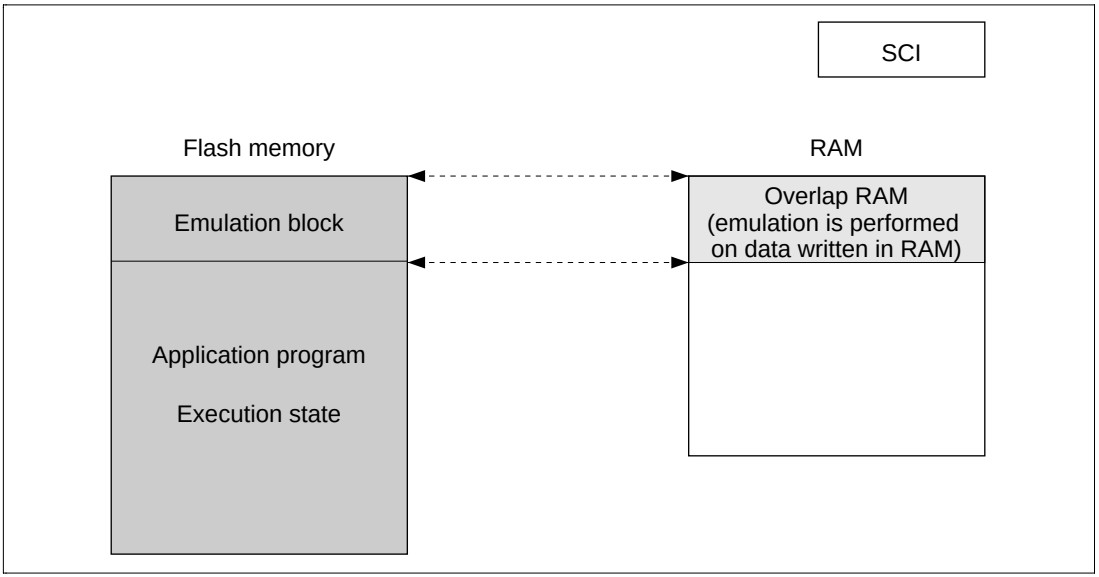


Figure 21A-4 Reading Overlap RAM Data in User Mode or User Program Mode

When overlap RAM data is confirmed, the RAMS bit is cleared, RAM overlap is released, and writes should actually be performed to the flash memory.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

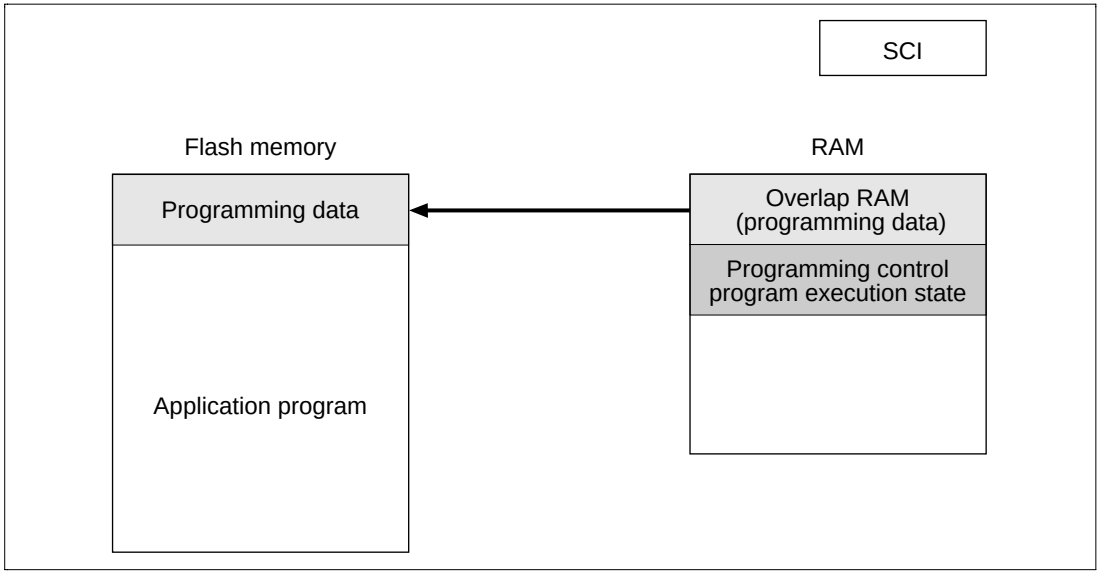


Figure 21A-5 Writing Overlap RAM Data in User Program Mode

21A.5.5 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	(2)	(1) (2) (3)

(1) Erase/erase-verify

(2) Program/program-verify

(3) Emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

21A.5.6 Block Configuration

The flash memory is divided into two 32 kbytes blocks, one 28 kbytes block, one 16 kbytes block, two 8 kbytes blocks, and four 1 kbyte blocks on the H8S/2636.

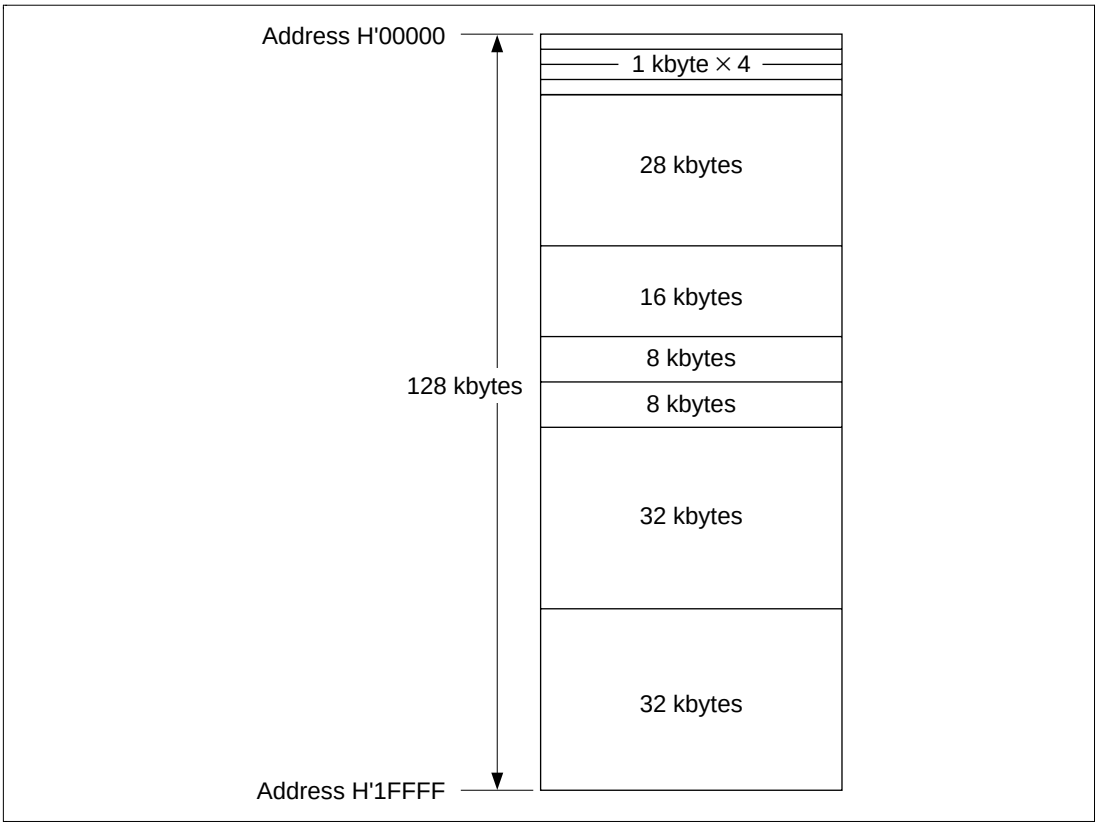


Figure 21A-6 Block Configuration (H8S/2636)

21A.6 Pin Configuration

The flash memory is controlled by means of the pins shown in table 21A-4.

Table 21A-4 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash write enable	FWE	Input	Flash program/erase protection by hardware
Mode 2	MD2	Input	Sets LSI operating mode
Mode 1	MD1	Input	Sets LSI operating mode
Mode 0	MD0	Input	Sets LSI operating mode
Port F0	PF0	Input	Sets LSI operating mode when MD2 = MD1 = MD0 =0
Port 16	P16	Input	Sets LSI operating mode when MD2 = MD1 = MD0 =0
Port 14	P14	Input	Sets LSI operating mode when MD2 = MD1 = MD0 =0
Transmit data	TxD1	Output	Serial transmit data output
Receive data	RxD1	Input	Serial receive data input

21A.7 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 21A-5.

Table 21A-5 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address*1
Flash memory control register 1	FLMCR1*4	R/W	H'00*2	H'FFA8
Flash memory control register 2	FLMCR2*4	R	H'00	H'FFA9
Erase block register 1	EBR1*4	R/W	H'00*3	H'FFAA
Erase block register 2	EBR2*4	R/W	H'00*3	H'FFAB
RAM emulation register	RAMER*4	R/W	H'00	H'FEDB
Flash memory power control register	FLPWCR*4	R/W	H'00*3	H'FFAC

Notes: *1 Lower 16 bits of the address.
*2 When a high level is input to the FWE pin, the initial value is H'80.
*3 When a low level is input to the FWE pin, or if a high level is input and the SWE1 bit in FLMCR1 is not set, these registers are initialized to H'00.
*4 FLMCR1, FLMCR2, EBR1, and EBR2, RAMER, and FLPWCR are 8-bit registers. Use byte access on these registers.

21A.8 Register Descriptions

21A.8.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode for addresses H'00000 to H'1FFFF is entered by setting SWE bit to 1 when FWE = 1, then setting the PV or EV bit. Program mode for addresses H'00000 to H'1FFFF is entered by setting SWE bit to 1 when FWE = 1, then setting the PSU bit, and finally setting the P bit. Erase mode for addresses H'00000 to H'1FFFF is entered by setting SWE bit to 1 when FWE = 1, then setting the ESU bit, and finally setting the E bit. FLMCR1 is initialized by a reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writes are enabled only in the following cases: Writes to bit SWE of FLMCR1 enabled when FWE = 1, to bits ESU, PSU, EV, and PV when FWE = 1 and SWE = 1, to bit E when FWE = 1, SWE = 1 and ESU = 1, and to bit P when FWE = 1, SWE = 1, and PSU = 1.

Bit:	7	6	5	4	3	2	1	0
	FWE	SWE	ESU	PSU	EV	PV	E	P
Initial value:	—*	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the state of the FWE pin.

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memory programming/erasing.

Bit 7: FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Bit 6—Software Write Enable Bit (SWE): Enables or disables flash memory programming and erasing. Set this bit when setting bits 5 to 0, bits 7 to 0 of EBR1, and bits 1 and 0 of EBR2.

Bit 6: SWE	Description	
0	Writes disabled	(Initial value)
1	Writes enabled [Setting condition] When FWE = 1	

Bit 5—Erase Setup Bit (ESU): Prepares for a transition to erase mode. Set this bit to 1 before setting the E bit in FLMCR1 to 1. Do not set the SWE, PSU, EV, PV, E, or P bit at the same time.

Bit 5: ESU	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 4—Program Setup Bit (PSU): Prepares for a transition to program mode. Set this bit to 1 before setting the P bit in FLMCR1 to 1. Do not set the SWE, ESU, EV, PV, E, or P bit at the same time.

Bit 4: PSU	Description	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 3—Erase-Verify (EV): Selects erase-verify mode transition or clearing. Do not set the SWE, ESU, PSU, PV, E, or P bit at the same time.

Bit 3: EV	Description	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 2—Program-Verify (PV): Selects program-verify mode transition or clearing. Do not set the SWE, ESU, PSU, EV, E, or P bit at the same time.

Bit 2: PV	Description	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 1—Erase (E): Selects erase mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or P bit at the same time.

Bit 1: E	Description	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU = 1	

Bit 0—Program (P): Selects program mode transition or clearing. Do not set the SWE, PSU, ESU, EV, PV, or E bit at the same time.

Bit 0: P	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU = 1	

21A.8.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is an 8-bit register used for flash memory operating mode control. FLMCR2 is initialized to H'00 by a reset, and in hardware standby mode and software standby mode. When on-chip flash memory is disabled, a read will return H'00.

Bit:	7	6	5	4	3	2	1	0
	FLER	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	—	—	—	—	—	—	—

Note: FLMCR2 is a read-only register, and should not be written to.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7: FLER	Description	
0	Flash memory is operating normally Flash memory program/erase protection (error protection) is disabled [Clearing condition] Reset or hardware standby mode	(Initial value)
1	An error has occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enabled [Setting condition] See 21A.8.3 Error Protection	

Bits 6 to 0—Reserved: These bits always read 0.

21A.8.3 Erase Block Register 1 (EBR1)

EBR1 is an 8-bit register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set. When a bit in EBR1 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit, as this will cause all the bits in both EBR1 and EBR2 to be automatically cleared to 0. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 21A-6.

Bit:	7	6	5	4	3	2	1	0
	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

21A.8.4 Erase Block Register 2 (EBR2)

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EBR2 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin. Bit 0 will be initialized to 0 if bit SWE of FLMCR1 is not set, even though a high level is input to pin FWE. When a bit in EBR2 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit, as this will cause all the bits in both EBR1 and EBR2 to be automatically cleared to 0. Bits 7 to 2 are reserved and must only be written with 0. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory block configuration is shown in table 21A-6.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	EB9	EB8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21A-6 Flash Memory Erase Blocks (H8S/2636)

Block (Size)	Addresses
EB0 (1 kbyte)	H'000000–H'0003FF
EB1 (1 kbyte)	H'000400–H'0007FF
EB2 (1 kbyte)	H'000800–H'000BFF
EB3 (1 kbyte)	H'000C00–H'000FFF
EB4 (28 kbytes)	H'001000–H'007FFF
EB5 (16 kbytes)	H'008000–H'00BFFF
EB6 (8 kbytes)	H'00C000–H'00DFFF
EB7 (8 kbytes)	H'00E000–H'00FFFF
EB8 (32 kbytes)	H'010000–H'017FFF
EB9 (32 kbytes)	H'018000–H'01FFFF

21A.8.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER initialized to H'00 by a reset and in hardware standby mode. It is not initialized by software standby mode. RAMER settings should be made in user mode or user program mode.

Flash memory area divisions are shown in table 21A-7. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Reserved: These bits always read 0.

Bits 5 and 4—Reserved: Only 0 may be written to these bits.

Bit 3—RAM Select (RAMS): Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, all flash memory block are program/erase-protected.

Bit 3: RAMS	Description	
0	Emulation not selected	(Initial value)
	Program/erase-protection of all flash memory blocks is disabled	
1	Emulation selected	
	Program/erase-protection of all flash memory blocks is enabled	

Bits 2, 1 and 0—Flash Memory Area Selection: These bits are used together with bit 3 to select the flash memory area to be overlapped with RAM. (See table 21A-7.)

Table 21A-7 Flash Memory Area Divisions (H8S/2636)

Addresses	Block Name	RAMS	RAM2	RAM1	RAM0
H'FFE000–H'FFE3FF	RAM area 1 kbyte	0	*	*	*
H'000000–H'0003FF	EB0 (1 kbyte)	1	0	0	*
H'000400–H'0007FF	EB1 (1 kbyte)	1	0	1	*
H'000800–H'000BFF	EB2 (1 kbyte)	1	1	0	*
H'000C00–H'000FFF	EB3 (1 kbyte)	1	1	1	*

*: Don't care

21A.8.6 Flash Memory Power Control Register (FLPWCR)

Bit:	7	6	5	4	3	2	1	0
	PDWND	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

FLPWCR enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode*.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Bit 7—Power-Down Disable (PDWND): Enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode. For details, see section 21A.15, Flash Memory and Power-Down States.

The sub-active mode cannot be used in the HD64F2636F.

When writing to this bit in the HD64F2636F, be sure to write 0.

Bit 7: PDWND	Description	
0	Transition to flash memory power-down mode enabled	(Initial value)
1	Transition to flash memory power-down mode disabled	

Bits 6 to 0—Reserved: These bits always read 0.

21A.9 On-Board Programming Modes

When pins are set to on-board programming mode and a reset-start is executed, a transition is made to the on-board programming state in which program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 21A-8. For a diagram of the transitions to the various flash memory modes, see figure 21A-3.

Table 21A-8 Setting On-Board Programming Modes

Mode		FWE	MD2	MD1	MD0
Boot mode	Expanded mode	1	0	1	0
	Single-chip mode		0	1	1
User program mode	Expanded mode	1	1	1	0
	Single-chip mode		1	1	1

21A.9.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The SCI channel to be used is set to asynchronous mode.

When a reset-start is executed after the LSI's pins have been set to boot mode, the boot program built into the LSI is started and the programming control program prepared in the host is serially transmitted to the LSI via the SCI. In the LSI, the programming control program received via the SCI is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address of the programming control program area and the programming control program execution state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 21A-7, and the boot mode execution procedure in figure 21A-8.

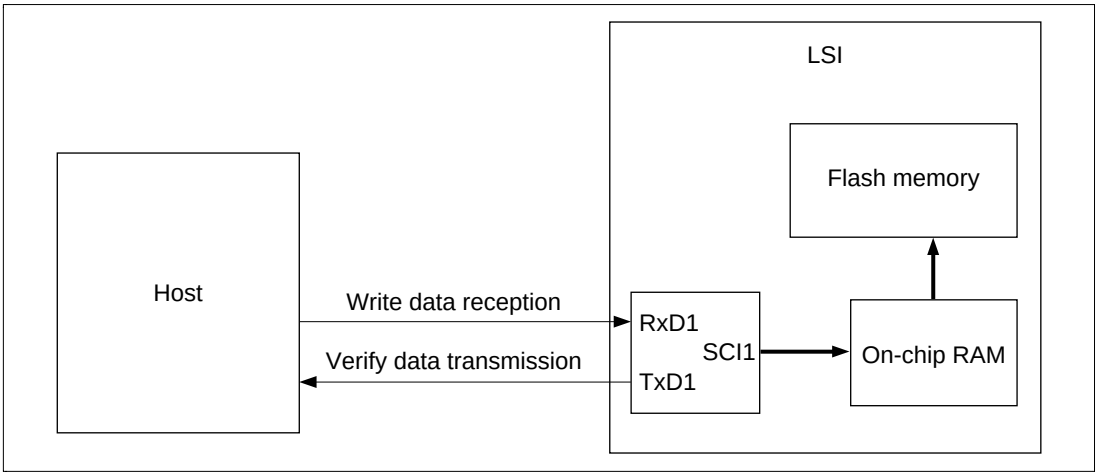
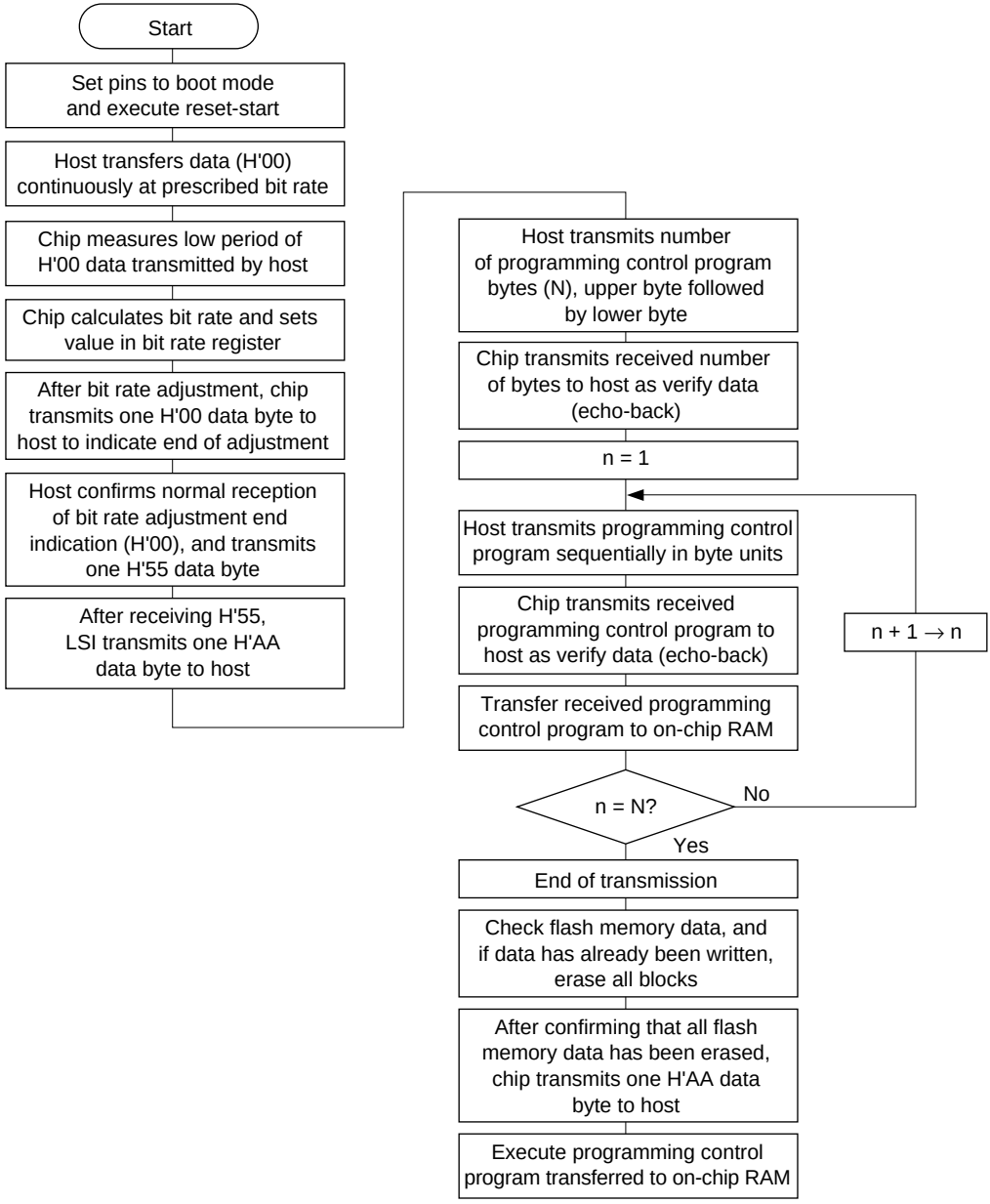


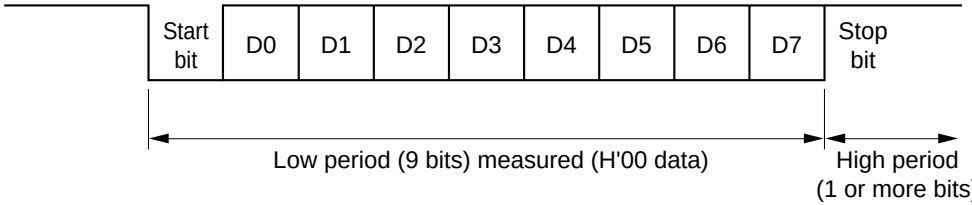
Figure 21A-7 System Configuration in Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operations are halted.

Figure 21A-8 Boot Mode Execution Procedure

Automatic SCI Bit Rate Adjustment



When boot mode is initiated, the LSI measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The LSI calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the LSI. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the LSI's system clock frequency, there will be a discrepancy between the bit rates of the host and the LSI. Set the host transfer bit rate at 19,200, 9,600 or 4,800 bps to operate the SCI properly.

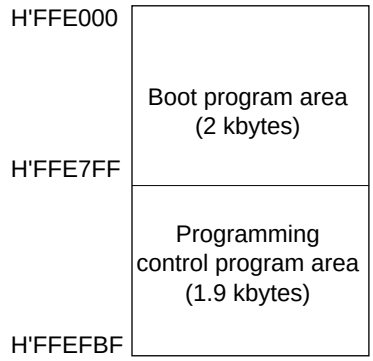
Table 21A-9 shows host transfer bit rates and system clock frequencies for which automatic adjustment of the LSI bit rate is possible. The boot program should be executed within this system clock range.

Table 21A-9 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	System Clock Frequency for Which Automatic Adjustment of LSI Bit Rate is Possible
19,200 bps	16–20 MHz
9,600 bps	8–20 HHz
4,800 bps	4–20 MHz

Note: The system clock frequency used in boot mode is generated by an external crystal oscillator element. PLL frequency multiplication is not used.

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the RAM area is divided into an area used by the boot program and an area to which the programming control program is transferred via the SCI, as shown in figure 21A-9. The boot program area cannot be used until the execution state in boot mode switches to the programming control program transferred from the host.



Note: The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note also that the boot program remains in this area of the on-chip RAM even after control branches to the programming control program.

Figure 21A-9 RAM Areas in Boot Mode (H8S/2636)

Notes on Use of Boot Mode:

- When the chip comes out of reset in boot mode, it measures the low-level period of the input at the SCI's RxD1 pin. The reset should end with RxD1 high. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period of the RxD1 pin.
- In boot mode, if any data has been programmed into the flash memory (if all data is not 1), all flash memory blocks are erased. Boot mode is for use when user program mode is unavailable, such as the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.
- Interrupts cannot be used while the flash memory is being programmed or erased.
- The RxD1 and TxD1 pins should be pulled up on the board.
- Before branching to the programming control program (RAM area H'FFE7FF), the chip terminates transmit and receive operations by the on-chip SCI (channel 1) (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. The transmit data output pin, TxD1, goes to the high-level output state (P33DDR = 1, P33DR = 1).

The contents of the CPU's internal general registers are undefined at this time, so these registers must be initialized immediately after branching to the programming control program. In particular, since the stack pointer (SP) is used implicitly in subroutine calls, etc., a stack area must be specified for use by the programming control program.

The initial values of other on-chip registers are not changed.

- Boot mode can be entered by making the pin settings shown in table 21A-8 and executing a reset-start.

  Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the FWE pin and mode pins, and executing reset release*[1]. Boot mode can also be cleared by a WDT overflow reset.

  Do not change the mode pin input levels in boot mode, and do not drive the FWE pin low*[3] while the boot program is being executed or while flash memory is being programmed or erased.

- If the mode pin input levels are changed (for example, from low to high) during a reset, the state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$) will change according to the change in the microcomputer's operating mode*[2].

  Therefore, care must be taken to make pin settings to prevent these pins from becoming output signal pins during a reset, or to prevent collision with signals outside the microcomputer.

Notes: *1 Mode pin and FWE pin input must satisfy the mode programming setup time (t_{MDS} = 4 states) with respect to the reset release timing.

*2 See appendix D, Pin States.

*3 For precautions on applying and disconnecting FWE, see section 21A.16, Flash Memory Programming and Erasing Precautions.

21A.9.2 User Program Mode

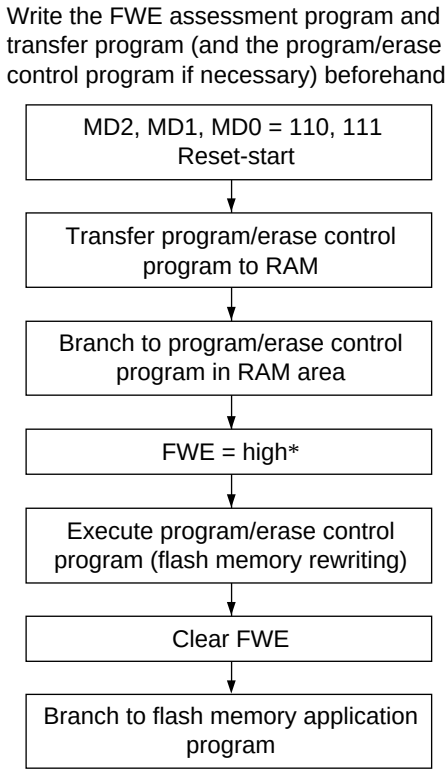
When set to user program mode, the chip can program and erase its flash memory by executing a user program/erase control program. Therefore, on-board reprogramming of the on-chip flash memory can be carried out by providing on-board means of FWE control and supply of programming data, and storing a program/erase control program in part of the program area as necessary.

To select user program mode, select a mode that enables the on-chip flash memory (mode 6 or 7), and apply a high level to the FWE pin. In this mode, on-chip supporting modules other than flash memory operate as they normally would in modes 6 and 7.

The flash memory itself cannot be read while the SWE bit is set to 1 to perform programming or erasing, so the control program that performs programming and erasing should be run in on-chip

RAM or external memory. If the program is to be located in external memory, the instruction for writing to flash memory, and the following instruction, should be placed in on-chip RAM.

Figure 21A-10 shows the procedure for executing the program/erase control program when transferred to on-chip RAM.



Note: Do not apply a constant high level to the FWE pin. Apply a high level to the FWE pin only when the flash memory is programmed or erased. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

* For precautions on applying and disconnecting FWE, see section 21A.16, Flash Memory Programming and Erasing Precautions.

Figure 21A-10 User Program Mode Execution Procedure

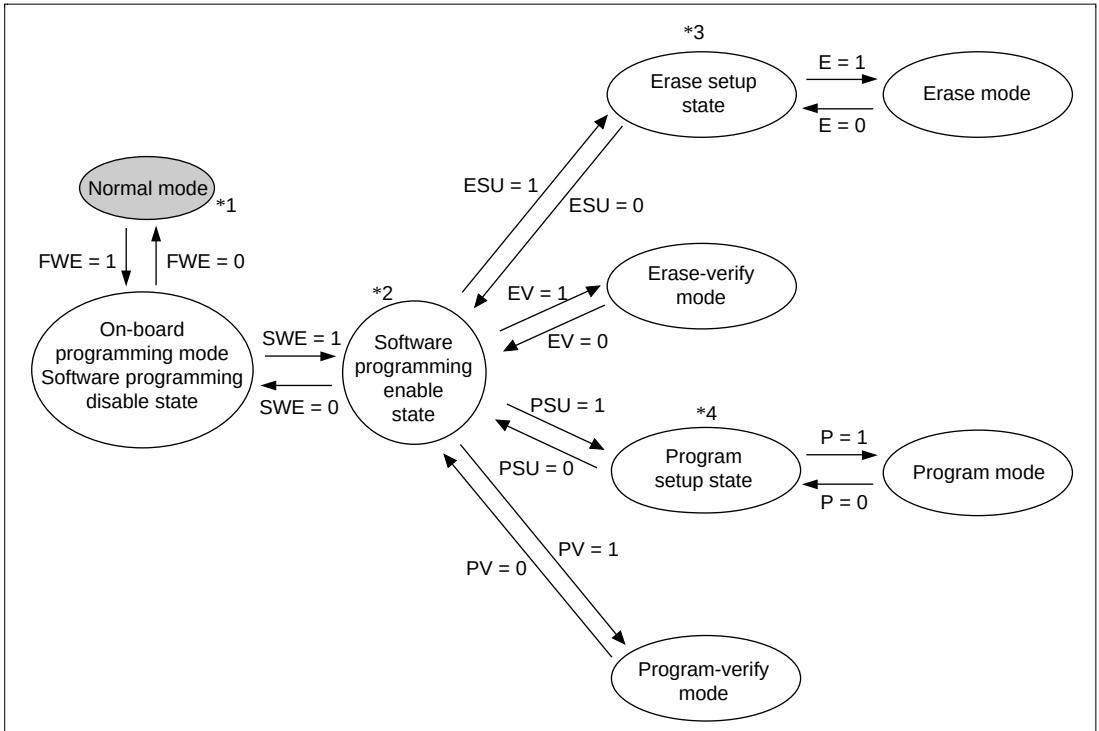
21A.10 Flash Memory Programming/Erasing

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes for addresses H'000000 to H'01FFFF are made by setting the PSU, ESU, P, E, PV, and EV bits in FLMCR1.

The flash memory cannot be read while being programmed or erased. Therefore, the program (user program) that controls flash memory programming/erasing should be located and executed in on-chip RAM or external memory. If the program is to be located in external memory, the instruction for writing to flash memory, and the following instruction, should be placed in on-chip RAM. Also ensure that the DTC is not activated before or after execution of the flash memory write instruction.

In the following operation descriptions, wait times after setting or clearing individual bits in FLMCR1 are given as parameters; for details of the wait times, see section 24.7, Flash Memory Characteristics.

Notes:
1. Operation is not guaranteed if setting/resetting of the SWE, ESU, PSU, EV, PV, E, and P bits in FLMCR1 is executed by a program in flash memory.
2. When programming or erasing, set FWE to 1 (programming/erasing will not be executed if FWE = 0).
3. Programming must be executed in the erased state. Do not perform additional programming on addresses that have already been programmed.



Notes: In order to perform a normal read of flash memory, SWE must be cleared to 0. Also note that verify-reads can be performed during the programming/erasing process.

*1 : Normal mode : On-board programming mode
*2 Do not make a state transition by setting or clearing multiple bits simultaneously.
*3 After a transition from erase mode to the erase setup state, do not enter erase mode without passing through the software programming enable state.
*4 After a transition from program mode to the program setup state, do not enter program mode without passing through the software programming enable state.

Figure 21A-11 FLMCR1 Bit Settings and State Transitions

21A.10.1 Program Mode

When writing data or programs to flash memory, the program/program-verify flowchart shown in figure 21A-12 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 128 bytes at a time.

The wait times after bits are set or cleared in the flash memory control register 1 (FLMCR1) and the maximum number of programming operations (N) are shown in table 24.10 in section 24.1.7, Flash Memory Characteristics.

Following the elapse of (t_{sswe}) µs or more after the SWE bit is set to 1 in FLMCR1, 128-byte data is written consecutively to the write addresses. The lower 8 bits of the first address written to must be H'00 and H'80, 128 consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. Set a value greater than ($t_{spsu} + t_{sp} + t_{cp} + t_{cpsu}$) µs as the WDT overflow period. Preparation for entering program mode (program setup) is performed next by setting the PSU bit in FLMCR1. The operating mode is then switched to program mode by setting the P bit in FLMCR1 after the elapse of at least (t_{spsu}) µs. The time during which the P bit is set is the flash memory programming time. Make a program setting so that the time for one programming operation is within the range of (t_{sp}) µs.

The wait time after P bit setting must be changed according to the degree of progress through the programming operation. For details see “Notes on Program/Program-Verify Procedure.”

21A.10.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of the given programming time, clear the P bit in FLMCR1, then wait for at least (t_{cp}) µs before clearing the PSU bit to exit program mode. After exiting program mode, the watchdog timer setting is also cleared. The operating mode is then switched to program-verify mode by setting the PV bit in FLMCR1. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (t_{spv}) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (t_{spvr}) µs after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 21A-12) and transferred to RAM. After verification of 128 bytes of data has been completed, exit program-verify mode, wait for at least (t_{cpv}) µs, then clear the SWE bit in FLMCR1. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. The maximum number of repetitions of the program/program-verify sequence is indicated by the maximum programming count (N). Leave a wait time of at least (t_{cswe}) µs after clearing SWE.

Notes on Program/Program-Verify Procedure

1. In order to perform 128-byte-unit programming, the lower 8 bits of the write start address must be H'00 or H'80.

2. When performing continuous writing of 128-byte data to flash memory, byte-unit transfer should be used.

   128-byte data transfer is necessary even when writing fewer than 128 bytes of data. Write H'FF data to the extra addresses.

3. Verify data is read in word units.

4. The write pulse is applied and a flash memory write executed while the P bit in FLMCR1 is set. In the chip, write pulses should be applied as follows in the program/program-verify procedure to prevent voltage stress on the device and loss of write data reliability.
   a. After write pulse application, perform a verify-read in program-verify mode and apply a write pulse again for any bits read as 1 (reprogramming processing). When all the 0-write bits in the 128-byte write data are read as 0 in the verify-read operation, the program/program-verify procedure is completed. In the chip, the number of loops in reprogramming processing is guaranteed not to exceed the maximum value of the maximum programming count (N).
   b. After write pulse application, a verify-read is performed in program-verify mode, and programming is judged to have been completed for bits read as 0. The following processing is necessary for programmed bits.

When programming is completed at an early stage in the program/program-verify procedure:

If programming is completed in the 1st to 6th reprogramming processing loop, additional programming should be performed on the relevant bits. Additional programming should only be performed on bits which first return 0 in a verify-read in certain reprogramming processing.

When programming is completed at a late stage in the program/program-verify procedure:

If programming is completed in the 7th or later reprogramming processing loop, additional programming is not necessary for the relevant bits.

c. If programming of other bits is incomplete in the 128 bytes, reprogramming processing should be executed. If a bit for which programming has been judged to be completed is read as 1 in a subsequent verify-read, a write pulse should again be applied to that bit.

5. The period for which the P bit in FLMCR1 is set (the write pulse width) should be changed according to the degree of progress through the program/program-verify procedure. For detailed wait time specifications, see section 24.1.7, Flash Memory Characteristics.

Item	Symbol	Item	Symbol
Wait time after P bit setting	t_{sp}	When reprogramming loop count (n) is 1 to 6	t_{sp30}
		When reprogramming loop count (n) is 7 or more	t_{sp200}
		In case of additional programming processing*	t_{sp10}

Note: * Additional programming processing is necessary only when the reprogramming loop count (n) is 1 to 6.

6. The program/program-verify flowchart for the H8S/2636 is shown in figure 21A-12.

To cover the points noted above, bits on which reprogramming processing is to be executed, and bits on which additional programming is to be executed, must be determined as shown below.

Since reprogram data and additional-programming data vary according to the progress of the programming procedure, it is recommended that the following data storage areas (128 bytes each) be provided in RAM.

Reprogram Data Computation Table

(D)	Result of Verify-Read after Write Pulse Application (V)	(X) Result of Operation	Comments
0	0	1	Programming completed: reprogramming processing not to be executed
0	1	0	Programming incomplete: reprogramming processing to be executed
1	0	1	—
1	1	1	Still in erased state: no action

Legend

(D): Source data of bits on which programming is executed

(X): Source data of bits on which reprogramming is executed

Additional-Programming Data Computation Table

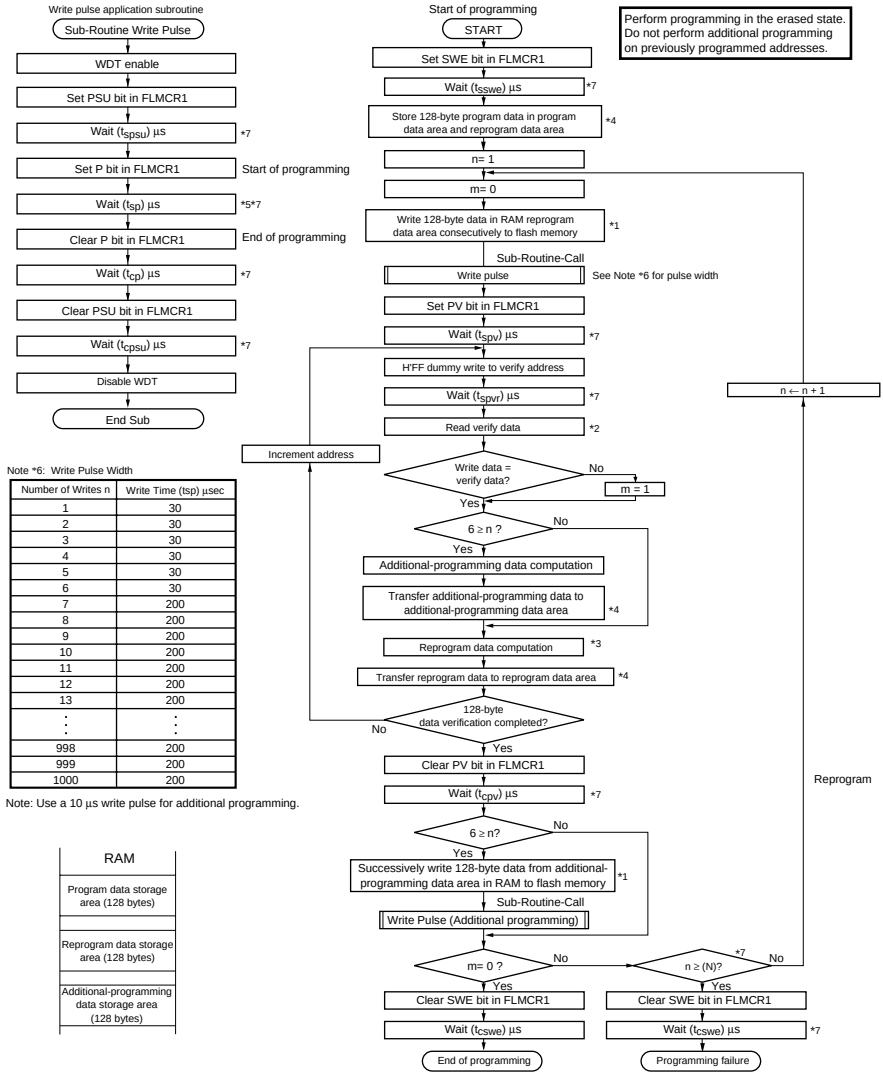
(X')	Result of Verify-Read after Write Pulse Application (V)	(Y) Result of Operation	Comments
0	0	0	Programming by write pulse application judged to be completed: additional programming processing to be executed
0	1	1	Programming by write pulse application incomplete: additional programming processing not to be executed
1	0	1	Programming already completed: additional programming processing not to be executed
1	1	1	Still in erased state: no action

Legend

(Y): Data of bits on which additional programming is executed

(X'): Data of bits on which reprogramming is executed in a certain reprogramming loop

7. It is necessary to execute additional programming processing during the course of the chip program/program-verify procedure. However, once 128-byte-unit programming is finished, additional programming should not be carried out on the same address area. When executing reprogramming, an erase must be executed first. Note that normal operation of reads, etc., is not guaranteed if additional programming is performed on addresses for which a program/program-verify operation has finished.



Note *6: Write Pulse Width

Number of Writes n	Write Time (tsp) µsec
1	30
2	30
3	30
4	30
5	30
6	30
7	200
8	200
9	200
10	200
11	200
12	200
13	200
⋮	⋮
998	200
999	200
1000	200

Note: Use a 10 µs write pulse for additional programming.

Notes: *1 Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80.
A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
*2 Verify data is read in 16-bit (word) units.
*3 Reprogram data is determined by the operation shown in the table below (comparison between the data stored in the program data area and the verify data). Bits for which the reprogram data is 0 are programmed in the next reprogramming loop. Therefore, even bits for which programming has been completed will be subjected to programming once again if the result of the subsequent verify operation is NG.
*4 A 128-byte area for storing program data, a 128-byte area for storing reprogram data, and a 128-byte area for storing additional data must be provided in RAM. The contents of the reprogram data area and additional data area are modified as programming proceeds.
*5 A write pulse of 30 µs or 200 µs is applied according to the progress of the programming operation. See Note *6 for details of the pulse widths. When writing of additional-programming data is executed, a 10 µs write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is applied.
*7 The wait times and value of N are shown in section 24.1.7, Flash Memory characteristics.

Reprogram Data Computation Table

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
0	1	0	Programming incomplete; reprogram
1	0	1	
1	1	1	Still in erased state; no action

Additional-Programming Data Computation Table

Reprogram Data (X')	Verify Data (V)	Additional-Programming Data (Y)	Comments
0	0	0	Additional programming to be executed
0	1	1	Additional programming not to be executed
1	0	1	Additional programming not to be executed
1	1	1	Additional programming not to be executed

Figure 21A-12 Program/Program-Verify Flowchart

21A.10.3 Erase Mode

When erasing flash memory, the single-block erase flowchart shown in figure 21A-13 should be followed.

The wait times after bits are set or cleared in the flash memory control register 1 (FLMCR1) and the maximum number of erase operations (N) are shown in table 24.10 in section 24.1.7, Flash Memory Characteristics.

To erase flash memory contents, make a 1-bit setting for the flash memory area to be erased in erase block register 1 and 2 (EBR1, EBR2) at least (t_{sswe}) µs after setting the SWE bit to 1 in FLMCR1. Next, the watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. Set a value greater than (t_{se}) ms + (t_{sesu} + t_{ce} + t_{cesu}) µs as the WDT overflow period. Preparation for entering erase mode (erase setup) is performed next by setting the ESU bit in FLMCR1. The operating mode is then switched to erase mode by setting the E bit in FLMCR1 after the elapse of at least (t_{sesu}) µs. The time during which the E bit is set is the flash memory erase time. Ensure that the erase time does not exceed (t_{se}) ms.

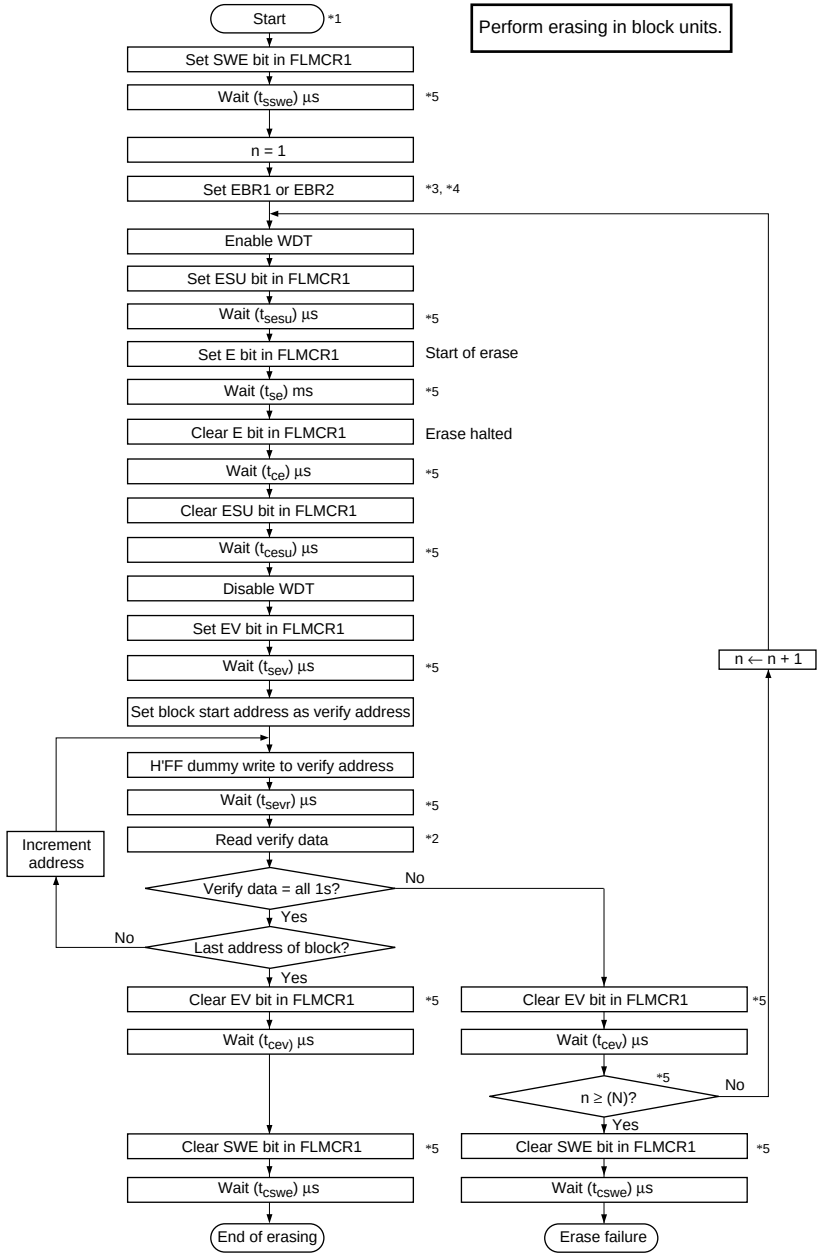
Note: With flash memory erasing, preprogramming (setting all memory data in the memory to be erased to all 0) is not necessary before starting the erase procedure.

21A.10.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the fixed erase time, clear the E bit in FLMCR1, then wait for at least (t_{ce}) µs before clearing the ESU bit to exit erase mode. After exiting erase mode, the watchdog timer setting is also cleared. The operating mode is then switched to erase-verify mode by setting the EV bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (t_{sev}) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (t_{sevr}) µs after the dummy write before performing this read operation. If the read data has been erased (all 1), a dummy write is performed to the next address, and erase-verify is performed. If the read data is unerased, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence is indicated by the maximum erase count (N). When verification is completed, exit erase-verify mode, and wait for at least (t_{cev}) µs. If erasure has been completed on all the erase blocks, clear the SWE bit in FLMCR1, and leave a wait time of at least (t_{cswe}) µs.

If erasing multiple blocks, set a single bit in EBR1/EBR2 for the next block to be erased, and repeat the erase/erase-verify sequence as before.



Notes: *1 Prewriting (setting erase block data to all 0s) is not necessary.
*2 Verify data is read in 16-bit (word) units.
*3 Make only a single-bit specification in the erase block registers (EBR1 and EBR2). Two or more bits must not be set simultaneously.
*4 Erasing is performed in block units. To erase multiple blocks, each block must be erased in turn.
*5 The wait times and the value of N are shown in section 24.1.7, Flash Memory Characteristics.

Figure 21A-13 Erase/Erase-Verify Flowchart

21A.11 Protection

There are three kinds of flash memory program/erase protection: hardware protection, software protection, and error protection.

21A.11.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2). The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained in the error-protected state. (See table 21A-10.)

Table 21A-10 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1, FLMCR2, (except bit FLER) EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Yes	Yes
Reset/standby protection	• In a reset (including a WDT reset) and in standby mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

21A.11.2 Software Protection

Software protection can be implemented by setting the SWE bit in FLMCR1, erase block register 1 (EBR1), erase block register 2 (EBR2), and the RAMS bit in the RAM emulation register (RAMER). When software protection is in effect, setting the P or E bit in flash memory control register 1 (FLMCR1), does not cause a transition to program mode or erase mode. (See table 21A-11.)

Table 21A-11 Software Protection

Item	Description	Functions	
		Program	Erase
SWE bit protection	• Setting bit SWE1 in FLMCR1 to 0 will place area H'000000 to H'01FFFFF in the program/erase-protected state. (Execute the program in the on-chip RAM, external memory)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) and erase block register 2 (EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

21A.11.3 Error Protection

In error protection, an error is detected when chip runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the chip malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

1. When the flash memory of the relevant address area is read during programming/erasing (including vector read and instruction fetch)
2. Immediately after exception handling (excluding a reset) during programming/erasing
3. When a SLEEP instruction (including software standby) is executed during programming/erasing
4. When the CPU releases the bus to the DTC

Error protection is released only by a reset and in hardware standby mode.

Figure 21A-14 shows the flash memory state transition diagram.

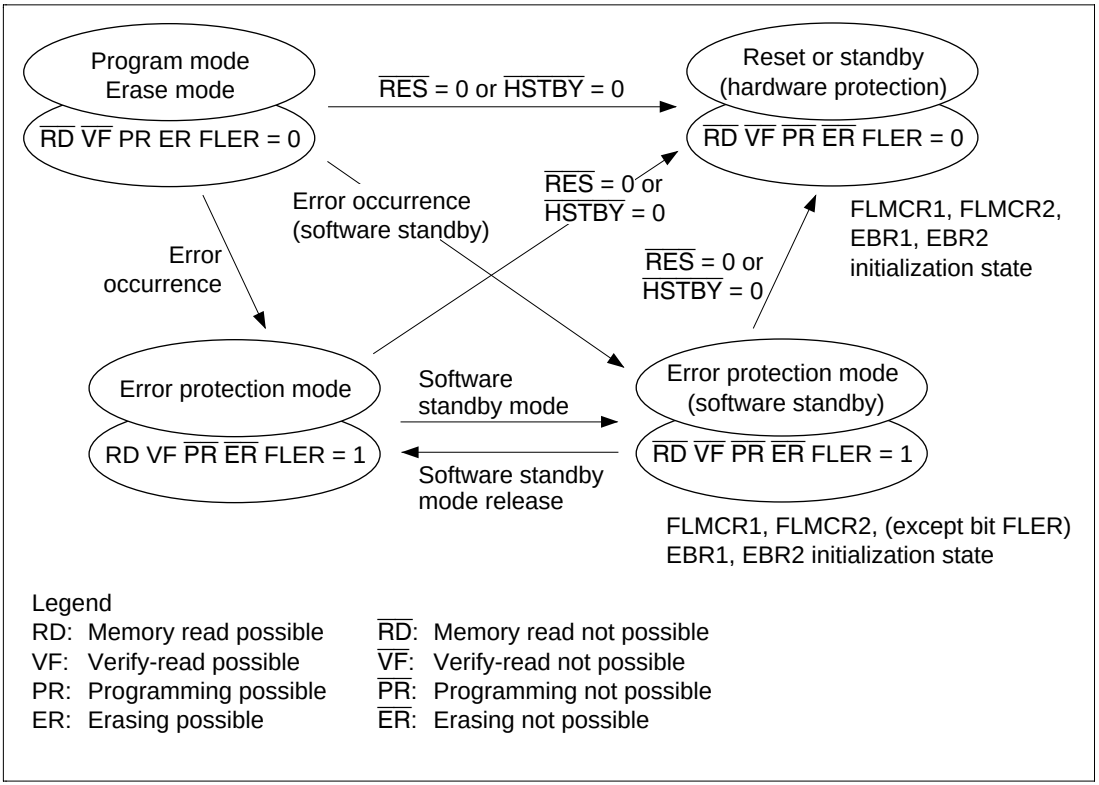


Figure 21A-14 Flash Memory State Transitions

21A.12 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses cannot be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 21A-15 shows an example of emulation of real-time flash memory programming.

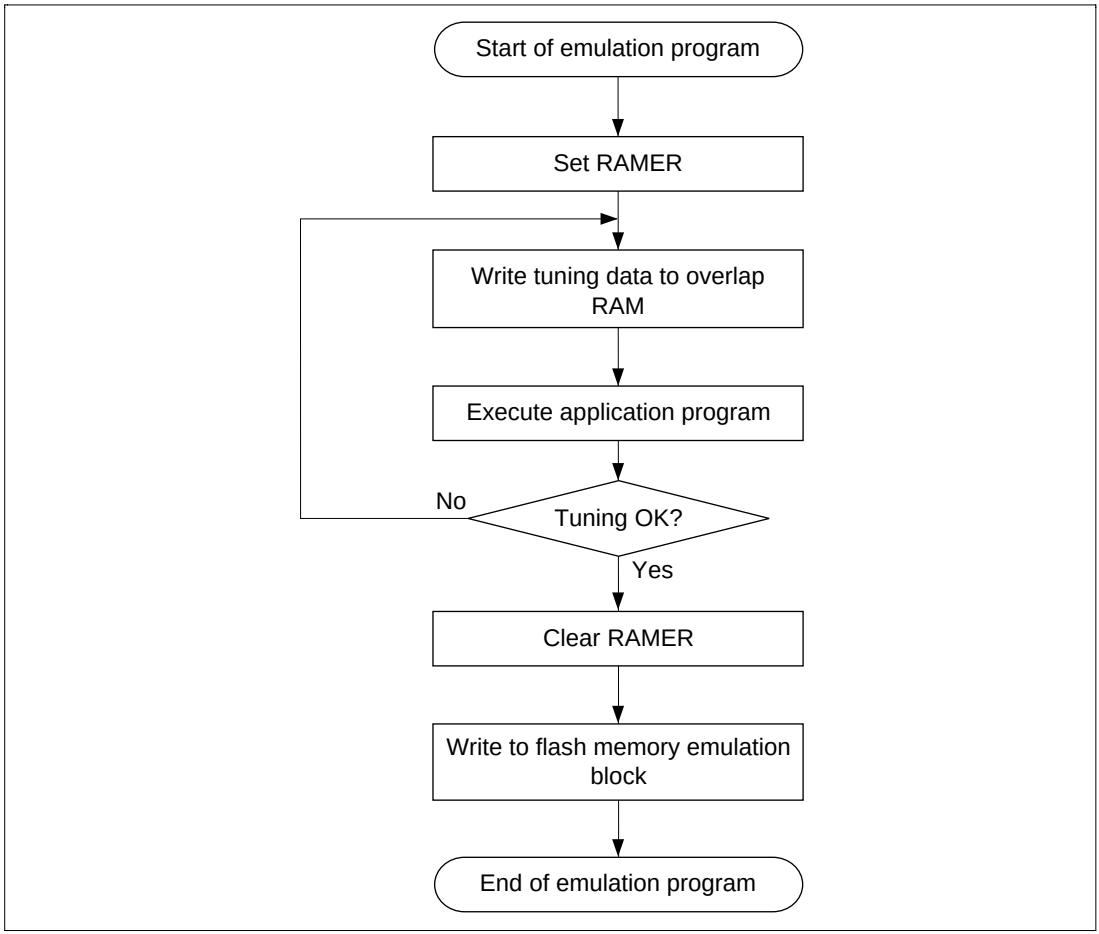


Figure 21A-15 Flowchart for Flash Memory Emulation in RAM

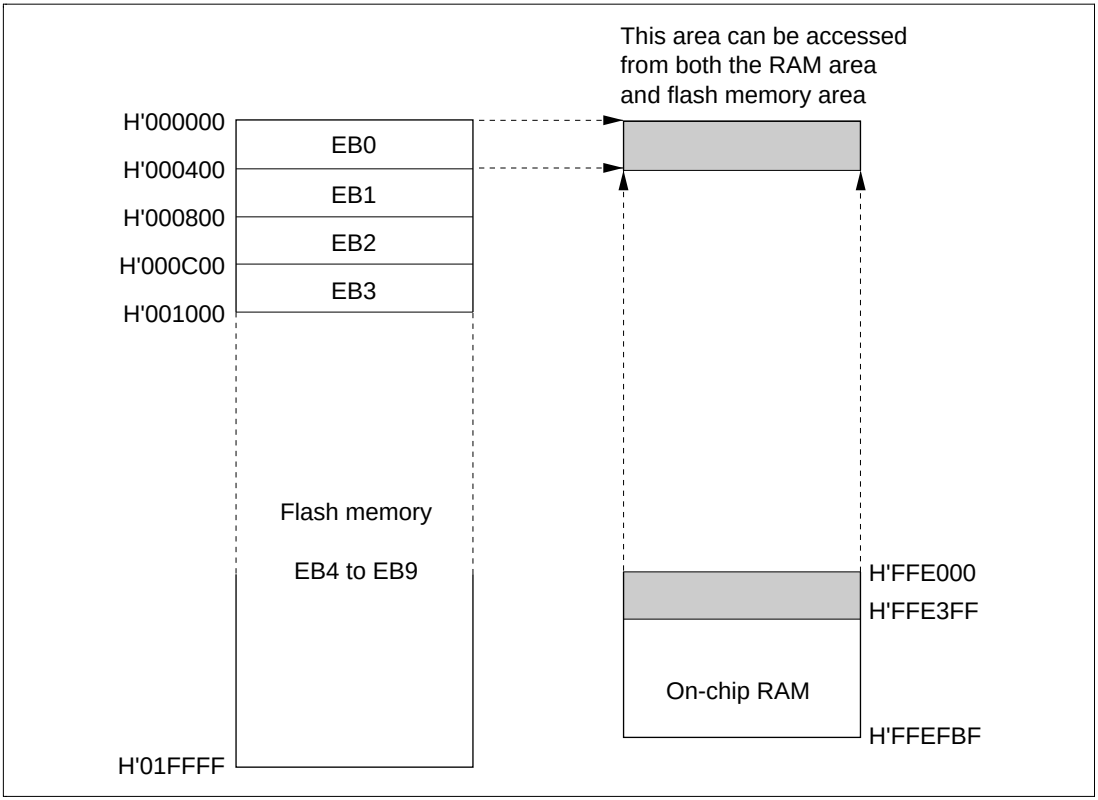


Figure 21A-16 Example of RAM Overlap Operation (H8S/2636)

Example in which Flash Memory Block Area EB0 is Overlapped

1. Set bits RAMS, RAM2 to RAM0 in RAMER to 1, 0, 0, 0, to overlap part of RAM onto the area (EB0) for which real-time programming is required.
2. Real-time programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB0).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM2 to RAM0 (emulation protection). In this state, setting the P or E bit in flash memory control register 1 (FLMCR1), will not cause a transition to program mode or erase mode. When actually programming or erasing a flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

21A.13 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI interrupt is disabled when flash memory is being programmed or erased (when the P or E bit is set in FLMCR1), and while the boot program is executing in boot mode*[1], to give priority to the program or erase operation. There are three reasons for this:

1. Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. In the interrupt exception handling sequence during programming or erasing, the vector would not be read correctly*[2], possibly resulting in MCU runaway.
3. If interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabling interrupt, as an exception to the general rule. However, this provision does not guarantee normal erasing and programming or MCU operation. All requests, including NMI interrupt, must therefore be restricted inside and outside the MCU when programming or erasing flash memory. NMI interrupt is also disabled in the error-protection state while the P or E bit remains set in FLMCR1.

Notes: *1 Interrupt requests must be disabled inside and outside the MCU until the programming control program has completed programming.

*2 The vector may not be read correctly in this case for the following two reasons:

- If flash memory is read while being programmed or erased (while the P or E bit is set in FLMCR1), correct read data will not be obtained (undetermined values will be returned).
- If the interrupt entry in the vector table has not been programmed yet, interrupt exception handling will not be executed correctly.

21A.14 Flash Memory Programmer Mode

Programs and data can be written and erased in programmer mode as well as in the on-board programming modes. In programmer mode, flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

In programmer mode, set the mode pins to programmer mode (see table 21A-12) and input a 12 MHz input clock.

Table 21A-12 shows the pin settings for programmer mode.

Table 21A-12 Programmer Mode Pin Settings

Pin Names	Settings
Mode pins: MD2, MD1, MD0	Low level input to MD2, MD1, and MD0.
Mode setting pins: PF0, P16, P14	High level input to PF0, low level input to P16 and P14
FWE pin	High level input (in auto-program and auto-erase modes)
$\overline{\text{RES}}$ pin	Reset circuit
XTAL, EXTAL, PLLCAP, PLLVSS pins	Oscillator circuit
VCL	Internal voltage step-down circuit

21A.14.1 Socket Adapter and Memory Map

In programmer mode in which the PROM writer is used, reading from memory (verification), writing, and initializing the flash memory (erasing all of its contents) are enabled. At this time, a dedicated conversion socket adapter must be attached to a general-purpose PROM writer. Table 21A-13 shows the types of the socket adapters. For programmer mode on this LSI, one of the socket adapters listed in table 21A-13 should be used.

Table 21A-13 Type of Socket Adapter

Product Name	Package Type	Socket Adapter Type	Manufacturer
HD64F2636UF	128 pin QFP	ME2636ESHF1H	Minato Electronics Inc.
HD64F2636F	(FP-128B)	HF2636Q128D4001	Data I/O Japan Corporation

The memory map of on-chip ROM is shown in figure 21A-17

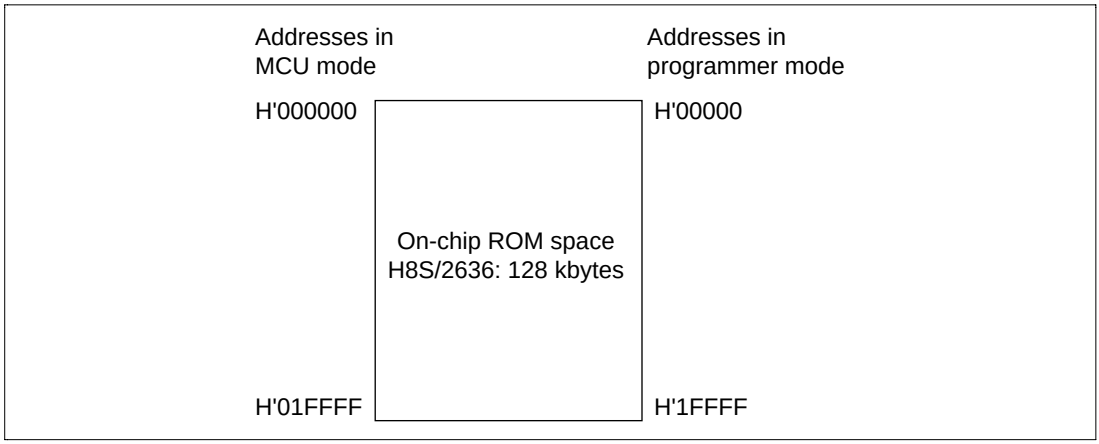


Figure 21A-17 On-Chip ROM Memory Map

21A.14.2 Programmer Mode Operation

Table 21A-14 shows how the different operating modes are set when using programmer mode, and table 21A-15 lists the commands used in programmer mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.
- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-programming.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the I/O6 signal. In status read mode, error information is output if an error occurs.

Table 21A-14 Settings for Various Operating Modes In Programmer Mode

	Pin Names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**I/O7– I/O0**	**A18–A0**
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-z	X
Command write	H or L	L	H	L	Data input	*Ain
Chip disable	H or L	H	X	X	Hi-z	X

Notes: 1. Chip disable is not a standby state; internally, it is an operation state.
2. *Ain indicates that there is also address input in auto-program mode.
3. For command writes in auto-program and auto-erase modes, input a high level to the FWE pin.

Table 21A-15 Programmer Mode Commands

	Number	1st Cycle			2nd Cycle		
Command Name	**of Cycles**	**Mode**	**Address**	**Data**	**Mode**	**Address**	**Data**
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	WA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

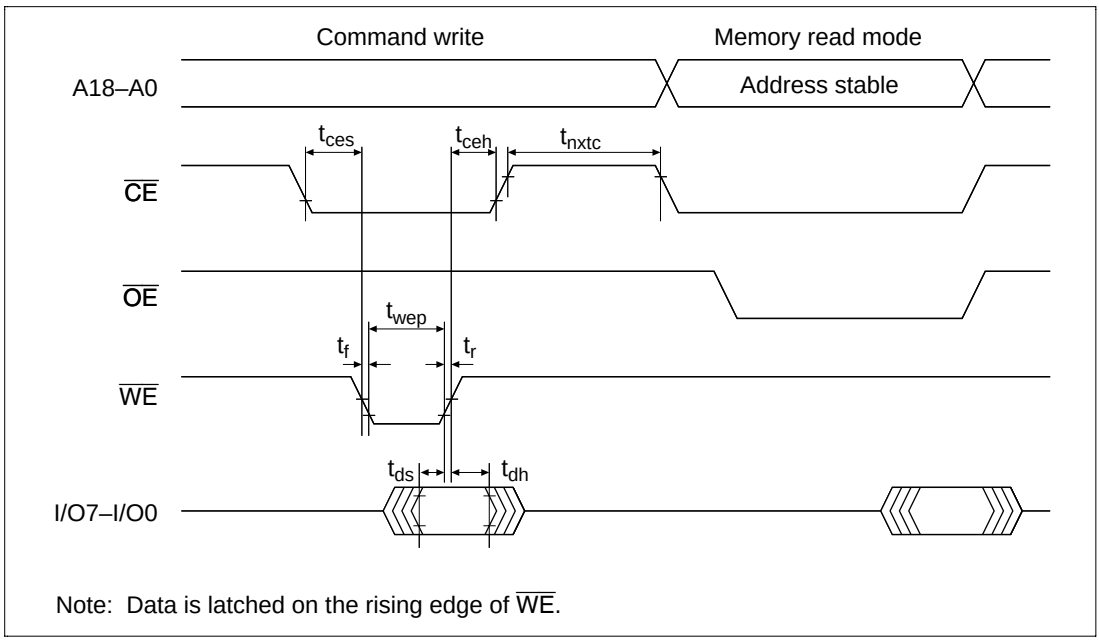
Notes: 1. In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

21A.14.3 Memory Read Mode

1. After completion of auto-program/auto-erase/status read operations, a transition is made to the command wait state. When reading memory contents, a transition to memory read mode must first be made with a command write, after which the memory contents are read.
2. In memory read mode, command writes can be performed in the same way as in the command wait state.
3. Once memory read mode has been entered, consecutive reads can be performed.
4. After powering on, memory read mode is entered.

Table 21A-16 AC Characteristics in Transition to Memory Read Mode (Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

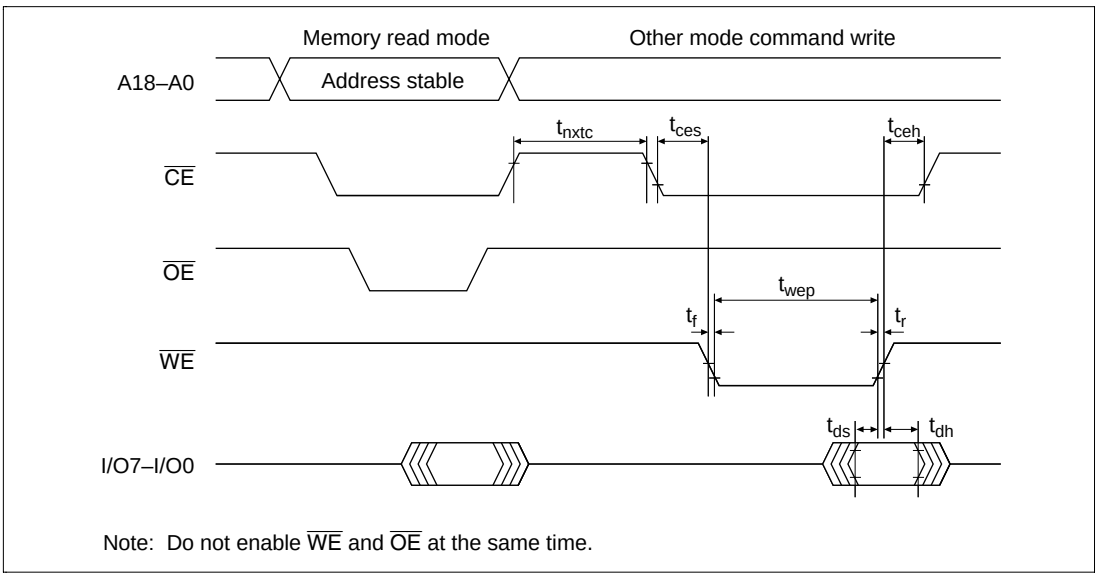


Note: Data is latched on the rising edge of $\overline{WE}$.

Figure 21A-18 Timing Waveforms for Memory Read after Memory Write

Table 21A-17 AC Characteristics in Transition from Memory Read Mode to Another Mode (Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns



Note: Do not enable $\overline{WE}$ and $\overline{OE}$ at the same time.

Figure 21A-19 Timing Waveforms in Transition from Memory Read Mode to Another Mode

Table 21A-18 AC Characteristics in Memory Read Mode (Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Access time	t_{acc}	—	20	µs
$\overline{CE}$ output delay time	t_{ce}	—	150	ns
$\overline{OE}$ output delay time	t_{oe}	—	150	ns
Output disable delay time	t_{df}	—	100	ns
Data output hold time	t_{oh}	5	—	ns

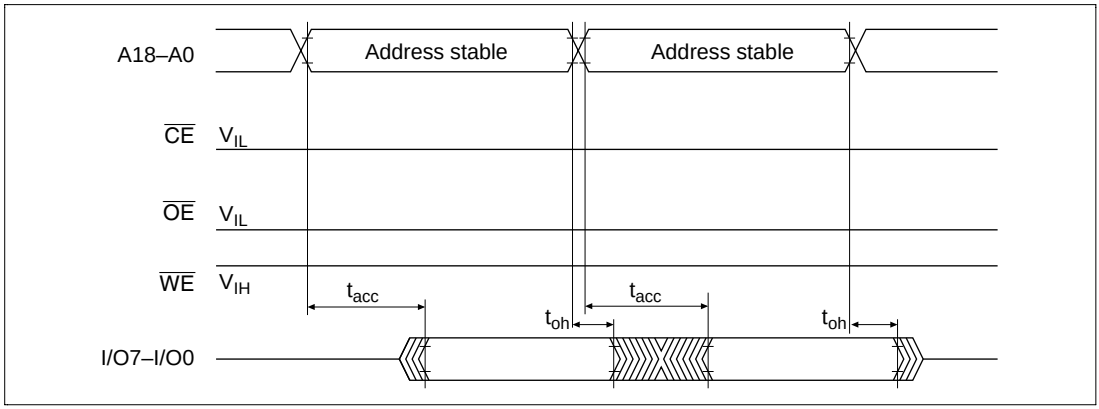


Figure 21A-20 $\overline{CE}$ and $\overline{OE}$ Enable State Read Timing Waveforms

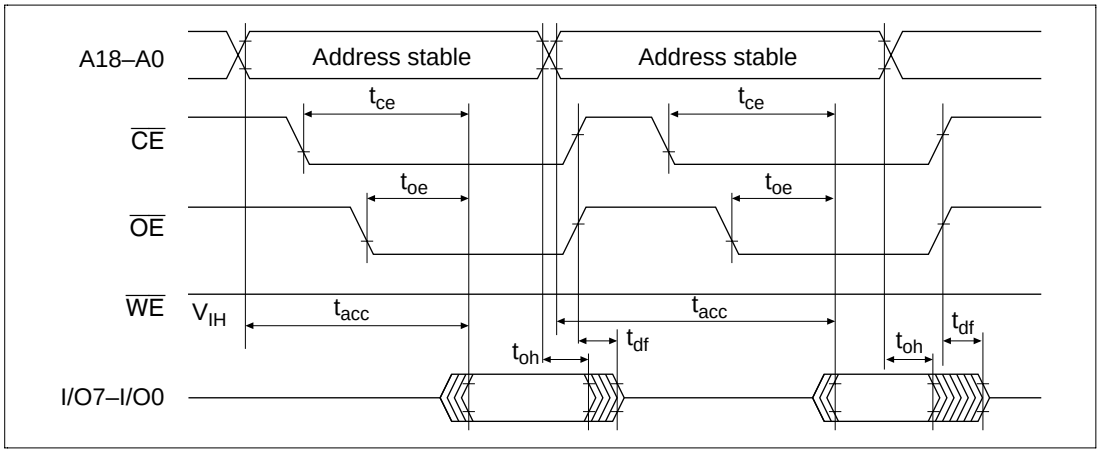


Figure 21A-21 $\overline{CE}$ and $\overline{OE}$ Clock System Read Timing Waveforms

21A.14.4 Auto-Program Mode

1. In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
2. A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. The lower 7 bits of the transfer address must be low. If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
4. Memory address transfer is performed in the second cycle (figure 21A-22). Do not perform transfer after the third cycle.
5. Do not perform a command write during a programming operation.
6. Perform one auto-program operation for a 128-byte block for each address. Two or more additional programming operations cannot be performed on a previously programmed address block.
7. Confirm normal end of auto-programming by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-program operation end decision pin).
8. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 21A-19 AC Characteristics in Auto-Program Mode (Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
Status polling start time	t_{wsts}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Address setup time	t_{as}	0	—	ns
Address hold time	t_{ah}	60	—	ns
Memory write time	t_{write}	1	3000	ms
Write setup time	t_{pns}	100	—	ns
Write end setup time	t_{pnh}	100	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

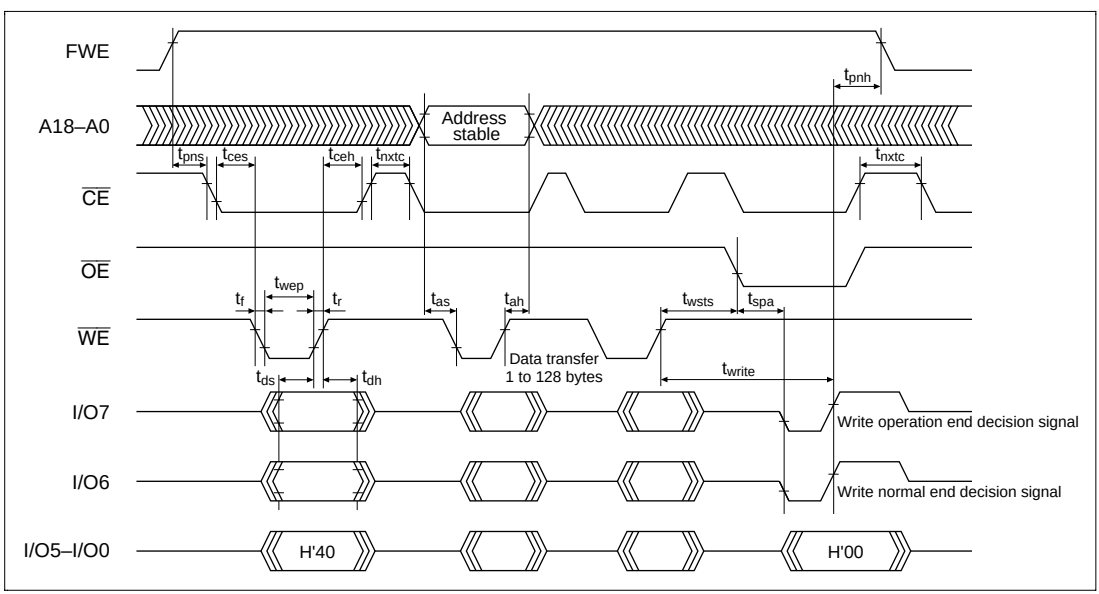


Figure 21A-22 Auto-Program Mode Timing Waveforms

21A.14.5 Auto-Erase Mode

1. Auto-erase mode supports only entire memory erasing.
2. Do not perform a command write during auto-erasing.
3. Confirm normal end of auto-erasing by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-erase operation end decision pin).
4. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 21A-20 AC Characteristics in Auto-Erase Mode (Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
Status polling start time	t_{ests}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Memory erase time	t_{erase}	100	40000	ms
Erase setup time	t_{ens}	100	—	ns
Erase end setup time	t_{enh}	100	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

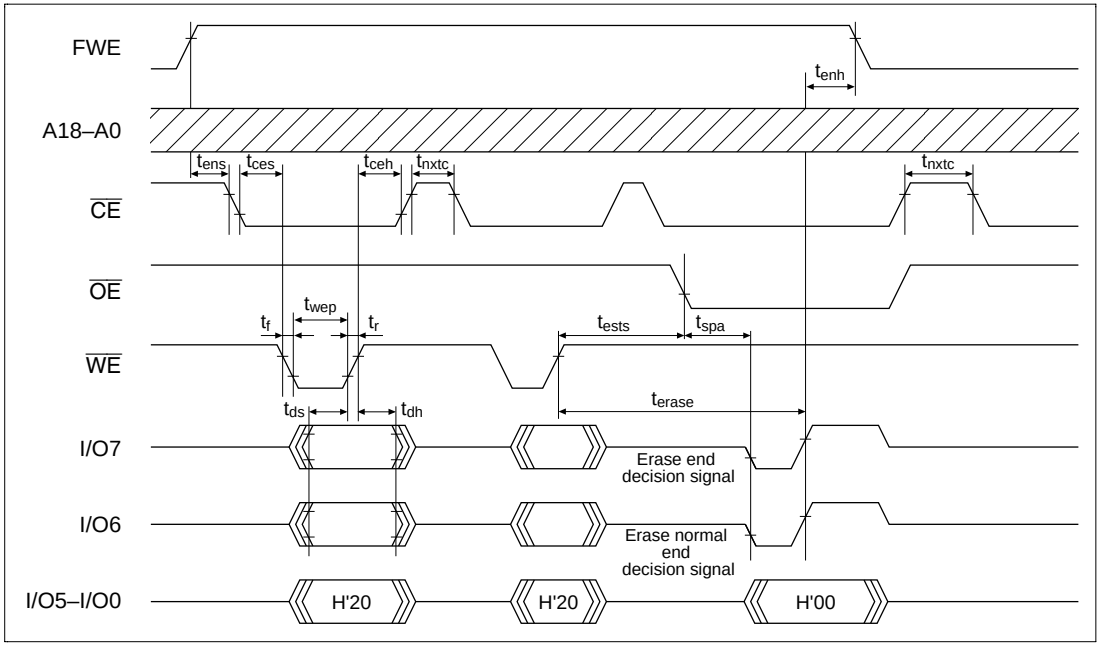


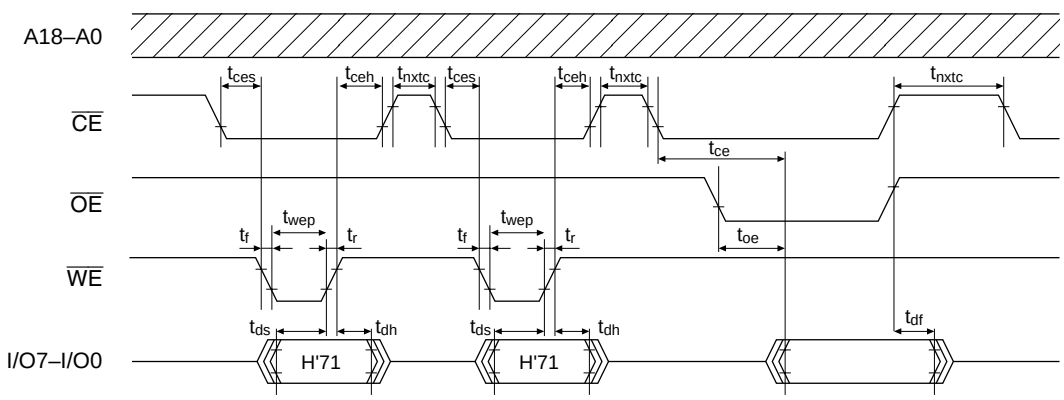
Figure 21A-23 Auto-Erase Mode Timing Waveforms

21A.14.6 Status Read Mode

1. Status read mode is provided to identify the kind of abnormal end. Use this mode when an abnormal end occurs in auto-program mode or auto-erase mode.
2. The return code is retained until a command write other than a status read mode command write is executed.

Table 21A-21 AC Characteristics in Status Read Mode (Conditions: V_{CC} = 5.0 V ± 0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Read time after command write	t_{std}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{OE}$ output delay time	t_{oe}	—	150	ns
Disable delay time	t_{df}	—	100	ns
$\overline{CE}$ output delay time	t_{ce}	—	150	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns



Note: I/O2 and I/O3 are undefined.

Figure 21A-24 Status Read Mode Timing Waveforms

Table 21A-22 Status Read Mode Return Commands

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end decision	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Program-ming error: 1 Otherwise: 0	Erasing error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address error: 1 Otherwise: 0

Note: I/O2 and I/O3 are undefined.

21A.14.7 Status Polling

1. The I/O7 status polling flag indicates the operating status in auto-program/auto-erase mode.
2. The I/O6 status polling flag indicates a normal or abnormal end in auto-program/auto-erase mode.

Table 21A-23 Status Polling Output Truth Table

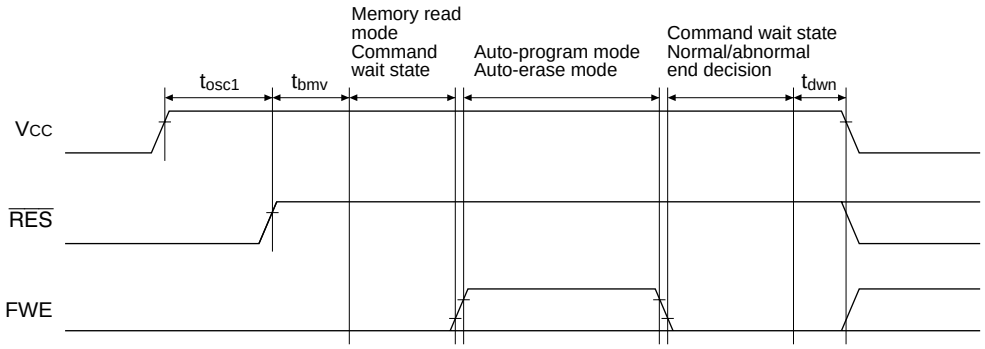
Pin Name	During Internal Operation	Abnormal End	—	Normal End
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O0–I/O5	0	0	0	0

21A.14.8 Programmer Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the programmer mode setup period. After the programmer mode setup time, a transition is made to memory read mode.

Table 21A-24 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit
Standby release (oscillation stabilization time)	t_{osc1}	30	—	ms
Programmer mode setup time	t_{bmv}	10	—	ms
VCC hold time	t_{dwn}	0	—	ms



Note: When using other than the automatic write mode and automatic erase mode, drive the FWE input pin low.

Figure 21A-25 Oscillation Stabilization Time, Boot Program Transfer Time, and Power-Down Sequence

21A.14.9 Notes on Memory Programming

1. When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
2. When performing programming using programmer mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Hitachi. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.

2. Auto-programming should be performed once only on the same address block. Additional programming cannot be performed on previously programmed address blocks.

21A.15 Flash Memory and Power-Down States

In addition to its normal operating state, the flash memory has power-down states in which power consumption is reduced by halting part or all of the internal power supply circuitry.

There are three flash memory operating states:

(1) Normal operating mode: The flash memory can be read and written to.
(2) Power-down mode: Part of the power supply circuitry is halted, and the flash memory can be read when the LSI is operating on the subclock*.
(3) Standby mode: All flash memory circuits are halted, and the flash memory cannot be read or written to.

States (2) and (3) are flash memory power-down states. Table 21A-25 shows the correspondence between the operating states of the LSI and the flash memory.

Table 21A-25 Flash Memory Operating States

LSI Operating State	Flash Memory Operating State
High-speed mode	Normal mode (read/write)
Medium-speed mode	
Sleep mode	
Subactive mode*	When PDWND = 0: Power-down mode (read-only)
Subsleep mode*	When PDWND = 1: Normal mode (read-only)
Watch mode*	Standby mode
Software standby mode	
Hardware standby mode	

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

21A.15.1 Notes on Power-Down States

1. When the flash memory is in a power-down state, part or all of the internal power supply circuitry is halted. Therefore, a power supply circuit stabilization period must be provided when returning to normal operation. When the flash memory returns to its normal operating state from a power-down state, bits STS2 to STS0 in SBYCR must be set to provide a wait time of at least 20 µs (power supply stabilization time), even if an oscillation stabilization period is not necessary.
2. In a power-down state, FLMCR1, FLMCR2, EBR1, EBR2, RAMER, and FLPWCR cannot be read from or written to.

21A.16 Flash Memory Programming and Erasing Precautions

Precautions concerning the use of on-board programming mode, the RAM emulation function, and programmer mode are summarized below.

1. **Use the specified voltages and timing for programming and erasing.**

   Applied voltages in excess of the rating can permanently damage the device. Use a PROM programmer that supports the Hitachi microcomputer device type with 128-kbyte on-chip flash memory.

   Only use the specified socket adapter. Failure to observe these points may result in damage to the device.

2. **Powering on and off (see figures 21A-26 to 21A-28)**

   Do not apply a high level to the FWE pin until V_{CC} has stabilized. Also, drive the FWE pin low before turning off V_{CC}.

   When applying or disconnecting V_{CC} power, fix the FWE pin low and place the flash memory in the hardware protection state.

   The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery.

3. **FWE application/disconnection (see figures 21A-26 to 21A-28)**

   FWE application should be carried out when MCU operation is in a stable condition. If MCU operation is not stable, fix the FWE pin low and set the protection state.

   The following points must be observed concerning FWE application and disconnection to prevent unintentional programming or erasing of flash memory:

   - Apply FWE when the V_{CC} voltage has stabilized within its rated voltage range.

     Apply FWE when oscillation has stabilized (after the elapse of the oscillation settling time).
   - In boot mode, apply and disconnect FWE during a reset.
   - In user program mode, FWE can be switched between high and low level regardless of a reset state.

     FWE input can also be switched during execution of a program in flash memory.
   - Do not apply FWE if program runaway has occurred.
   - Disconnect FWE only when the SWE, ESU, PSU, EV, PV, P, and E bits in FLMCR1 are cleared.

     Make sure that the SWE, ESU, PSU, EV, PV, P, and E bits are not set by mistake when applying or disconnecting FWE.

4. **Do not apply a constant high level to the FWE pin.**

   Apply a high level to the FWE pin only when programming or erasing flash memory. A system configuration in which a high level is constantly applied to the FWE pin should be avoided. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

5. **Use the recommended algorithm when programming and erasing flash memory.**

   The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P or E bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

6. **Do not set or clear the SWE bit during execution of a program in flash memory.**

   Do not set or clear the SWE bit during execution of a program in flash memory. Wait for at least 100 µs after clearing the SWE bit before executing a program or reading data in flash memory. When the SWE bit is set, data in flash memory can be rewritten, but when SWE = 1, flash memory can only be read in program-verify or erase-verify mode. Access flash memory only for verify operations (verification during programming/erasing). Do not clear the SWE bit during programming, erasing, or verifying.

   Similarly, when using the RAM emulation function while a high level is being input to the FWE pin, the SWE bit must be cleared before executing a program or reading data in flash memory. However, the RAM area overlapping flash memory space can be read and written to regardless of whether the SWE bit is set or cleared.

7. **Do not use interrupts while flash memory is being programmed or erased.**

   All interrupt requests, including NMI, should be disabled during FWE application to give priority to program/erase operations.

8. **Do not perform additional programming. Erase the memory before reprogramming.**

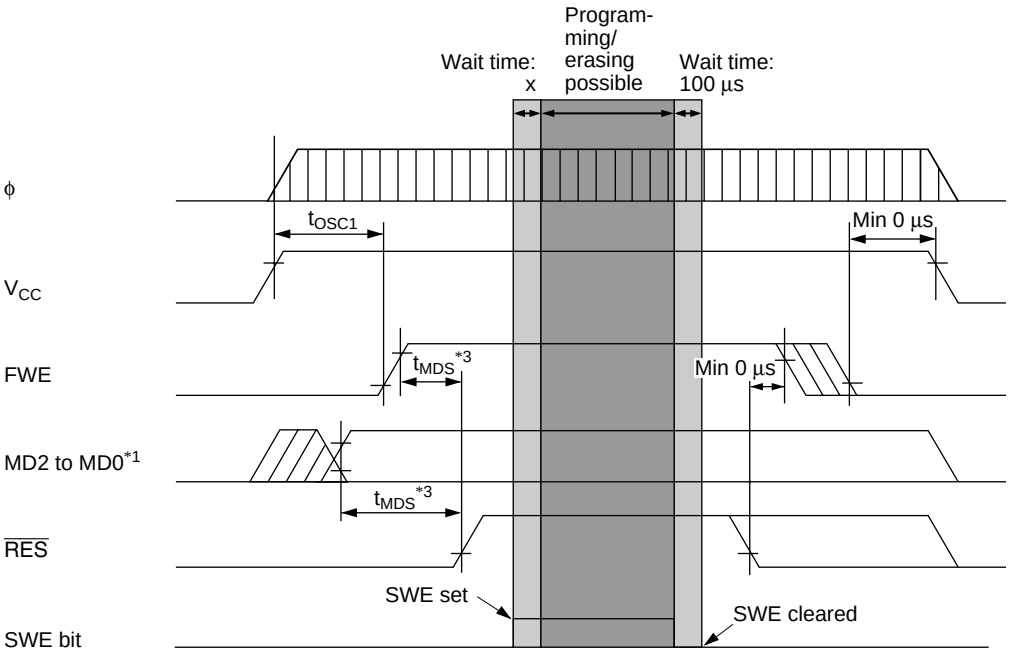
   In on-board programming, perform only one programming operation on a 128-byte programming unit block. In programmer mode, also, perform only one programming operation on a 128-byte programming unit block. Further programming must only be executed after this programming unit block has been erased.

9. **Before programming, check that the chip is correctly mounted in the PROM programmer.**

   Overcurrent damage to the device can result if the index marks on the PROM programmer socket, socket adapter, and chip are not correctly aligned.

10. **Do not touch the socket adapter or chip during programming.**

    Touching either of these can cause contact faults and write errors.

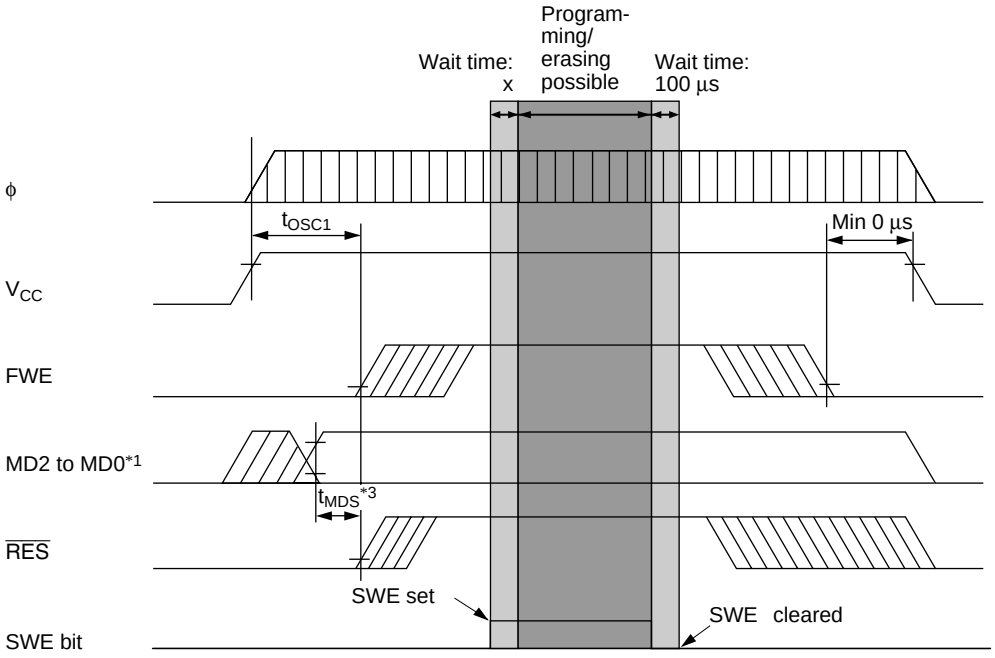


Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: *1 Except when switching modes, the level of the mode pins (MD2–MD0) must be fixed until power-off by pulling the pins up or down.
*2 See section 24.1.7, Flash Memory Characteristics.
*3 Mode programming setup time t_{MDS} (min) = 200 ns

Figure 21A-26 Power-On/Off Timing (Boot Mode)

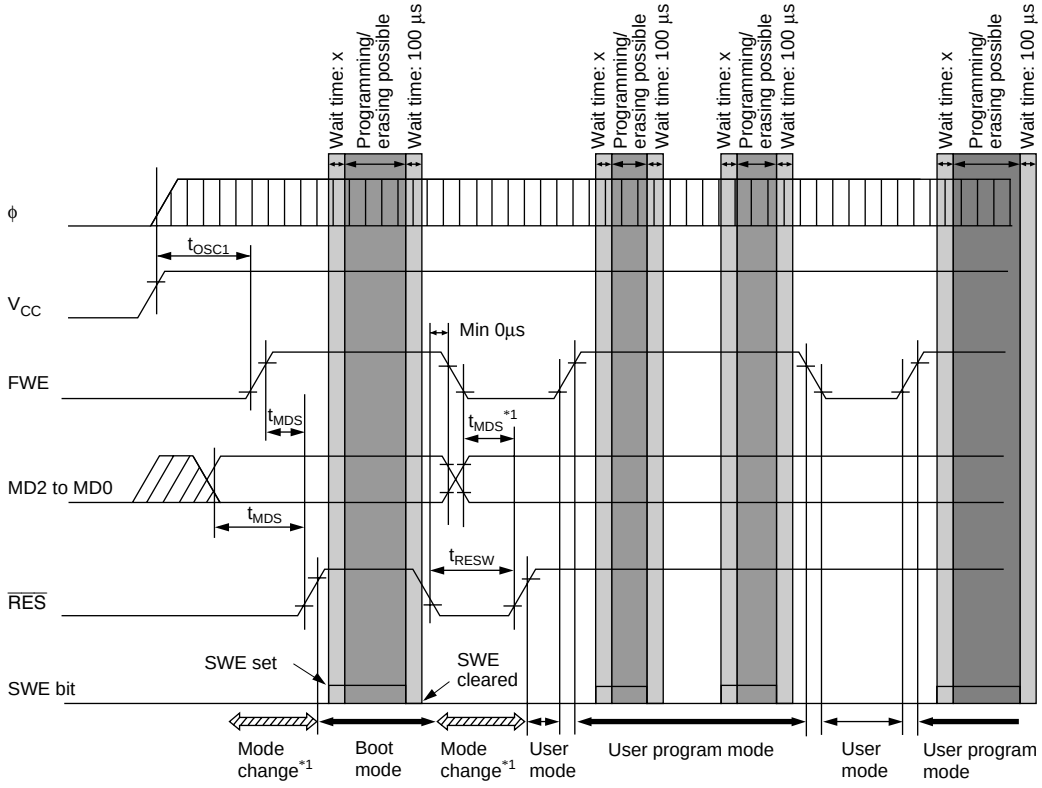


Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: *1 Except when switching modes, the level of the mode pins (MD2–MD0) must be fixed until power-off by pulling the pins up or down.
*2 See section 24.1.7, Flash Memory Characteristics.
*3 Mode programming setup time t_{MDS} (min) = 200 ns

Figure 21A-27 Power-On/Off Timing (User Program Mode)



Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*3

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: *1 When entering boot mode or making a transition from boot mode to another mode, mode switching must be carried out by means of $\overline{RES}$ input. The state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{WR}$) will change during this switchover interval (the interval during which the $\overline{RES}$ pin input is low), and therefore these pins should not be used as output signals during this time.

*2 When making a transition from boot mode to another mode, a mode programming setup time, t_{MDS} (min), of 200 ns is necessary with respect to the $\overline{RES}$ clearance timing.

*3 See section 24.1.7, Flash Memory Characteristics.

Figure 21A-28 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

21A.17 Note on Switching from F-ZTAT Version to Mask ROM Version

The mask ROM version does not have the internal registers for flash memory control that are provided in the F-ZTAT version. Table 21A-26 lists the registers that are present in the F-ZTAT version but not in the mask ROM version. If a register listed in table 21A-26 is read in the mask ROM version, an undefined value will be returned. Therefore, if application software developed on the F-ZTAT version is switched to a mask ROM version product, it must be modified to ensure that the registers in table 21A-26 have no effect.

Table 21A-26 Registers Present in F-ZTAT Version but Absent in Mask ROM Version

Register	Abbreviation	Address
Flash memory control register 1	FLMCR1	H'FFC8
Flash memory control register 2	FLMCR2	H'FFC9
Erase block register 1	EBR1	H'FFCA
Erase block register 2	EBR2	H'FFCB
RAM emulation register	RAMER	H'FEDB

21B.1 Overview

The H8S/2638 and H8S/2639 have 256 kbytes of on-chip flash memory, or 256 kbytes of on-chip mask ROM. The ROM is connected to the bus master via a 16-bit data bus, enabling both byte and word data to be accessed in one state. Instruction fetching is thus speeded up, and processing speed increased.

The on-chip ROM is enabled and disabled by setting the mode pins (MD2 to MD0).

The flash memory version can be erased and programmed on-board, as well as with a special-purpose PROM programmer.

21B.1.1 Block Diagram

Figure 21B-1 shows a block diagram of 256-kbyte ROM.

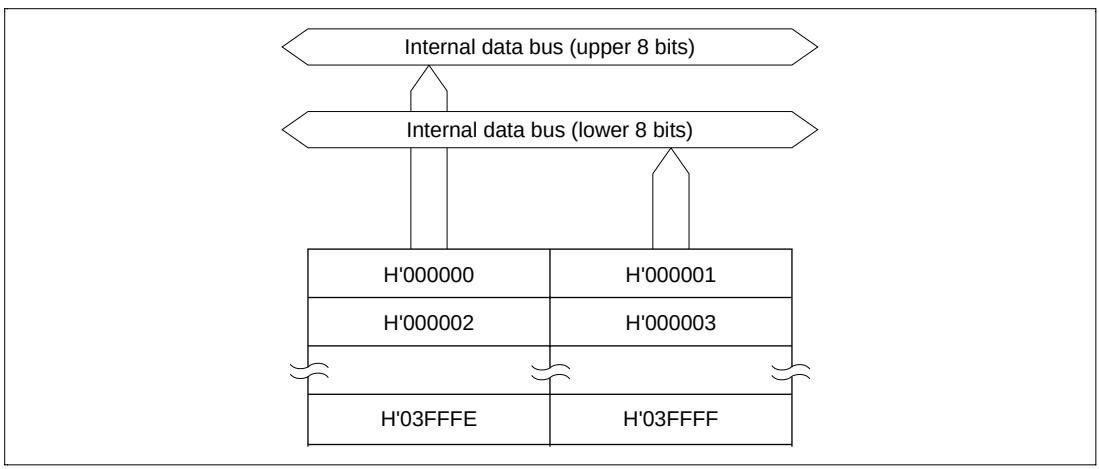


Figure 21B-1 Block Diagram of ROM (256 kbytes)

21B.1.2 Register Configuration

The H8S/2638 and H8S/2639 operating mode is controlled by the mode pins and the BCRL register. The register configuration is shown in table 21B-1.

Table 21B-1 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address*
Mode control register	MDCR	R/W	Undefined	H'FDE7

Note: * Lower 16 bits of the address.

21B.2 Register Descriptions

21B.2.1 Mode Control Register (MDCR)

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	—	MDS2	MDS1	MDS0
Initial value:	1	0	0	0	0	—*	—*	—*
R/W:	R/W	—	—	—	—	R	R	R

Note: * Determined by pins MD2 to MD0.

MDCR is an 8-bit read-only register used to monitor the current H8/2633 Series operating mode.

Bit 7—Reserved: Only 1 should be written to this bit.

Bits 6 to 3—Reserved: Read-only bits, always read as 0.

Bits 2 to 0—Mode Select 2 to 0 (MDS2 to MDS0): These bits indicate the input levels at pins MD2 to MD0 (the current operating mode). Bits MDS2 to MDS0 correspond to pins MD2 to MD0. MDS2 to MDS0 are read-only bits, and cannot be modified. The mode pin (MD2 to MD0) input levels are latched into these bits when MDCR is read. These latches are canceled by a power-on reset, but are retained in a manual reset.

21B.3 Operation

The on-chip ROM is connected to the CPU by a 16-bit data bus, and both byte and word data can be accessed in one state. Even addresses are connected to the upper 8 bits, and odd addresses to the lower 8 bits. Word data must start at an even address.

The on-chip ROM is enabled and disabled by setting the mode pins (MD2, MD1, and MD0). These settings are shown in table 21B-2.

Table 21B-2 Operating Modes and ROM (F-ZTAT Version)

		Mode Pins				
	Operating Mode	**FWE**	**MD2**	**MD1**	**MD0**	**On-Chip ROM**
Mode 0	—	0	0	0	0	—
Mode 1					1	
Mode 2				1	0	
Mode 3					1	
Mode 4	Advanced expanded mode with on-chip ROM disabled		1	0	0	Disabled
Mode 5	Advanced expanded mode with on-chip ROM disabled				1	
Mode 6	Advanced expanded mode with on-chip ROM enabled			1	0	Enabled (256 kbytes)
Mode 7	Advanced single-chip mode				1	Enabled (256 kbytes)
Mode 8	—	1	0	0	0	—
Mode 9					1	
Mode 10	Boot mode (advanced expanded mode with on-chip ROM enabled)*1			1	0	Enabled (256 kbytes)
Mode 11	Boot mode (advanced single-chip mode)*2				1	Enabled (256 kbytes)
Mode 12	—		1	0	0	—
Mode 13					1	
Mode 14	User program mode (advanced expanded mode with on-chip ROM enabled)*1			1	0	Enabled (256 kbytes)
Mode 15	User program mode (advanced single-chip mode)*2				1	Enabled (256 kbytes)

Notes: *1 Apart from the fact that flash memory can be erased and programmed, operation is the same as in advanced expanded mode with on-chip ROM enabled.

*2 Apart from the fact that flash memory can be erased and programmed, operation is the same as in advanced single-chip mode.

Table 21B-3 Operating Modes and ROM (Mask ROM Version)

		Mode Pins			
	Operating Mode	**MD2**	**MD1**	**MD0**	**On-Chip ROM**
Mode 0	—	0	0	0	—
Mode 1				1	
Mode 2			1	0	
Mode 3				1	
Mode 4	Advanced expanded mode with on-chip ROM disabled	1	0	0	Disabled
Mode 5	Advanced expanded mode with on-chip ROM disabled			1	
Mode 6	Advanced expanded mode with on-chip ROM enabled		1	0	Enabled (256 kbytes)
Mode 7	Advanced single-chip mode			1	Enabled (256 kbytes)

21B.4 Features

The H8S/2638 and H8S/2639 have 256 kbytes of on-chip flash memory. The features of the flash memory are summarized below.

- Four flash memory operating modes
  - — Program mode
  - — Erase mode
  - — Program-verify mode
  - — Erase-verify mode
- Programming/erase methods

  The flash memory is programmed 128 bytes at a time. Block erase (in single-block units) can be performed. To erase the entire flash memory, each block must be erased in turn. Block erasing can be performed as required on 4 kbytes, 32 kbytes, and 64 kbytes blocks.
- Programming/erase times

  The flash memory programming time is 10 ms (typ.) for simultaneous 128-byte programming, equivalent to 80 µs (typ.) per byte, and the erase time is 100 ms (typ.).
- Reprogramming capability

  The flash memory can be reprogrammed up to 100 times.
- On-board programming modes

  There are two modes in which flash memory can be programmed/erased/verified on-board:
  - — Boot mode
  - — User program mode
- Automatic bit rate adjustment

  With data transfer in boot mode, the LSI's bit rate can be automatically adjusted to match the transfer bit rate of the host.
- Flash memory emulation in RAM

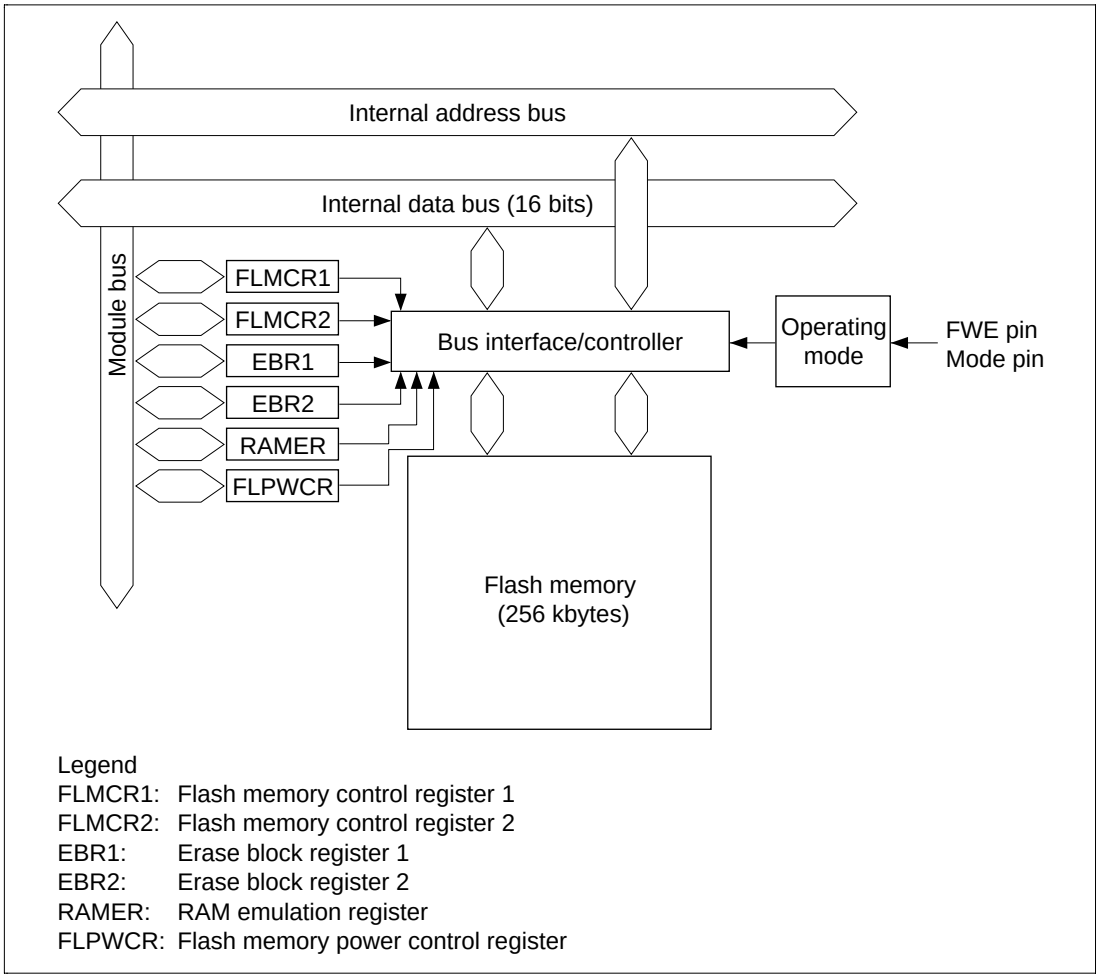
  Flash memory programming can be emulated in real time by overlapping a part of RAM onto flash memory.
- Protect modes

  There are three protect modes, hardware, software, and error protection, which allow protected status to be designated for flash memory program/erase/verify operations.
- Programmer mode

  Flash memory can be programmed/erased in programmer mode, using a PROM programmer, as well as in on-board programming mode.

21B.5 Overview

21B.5.1 Block Diagram



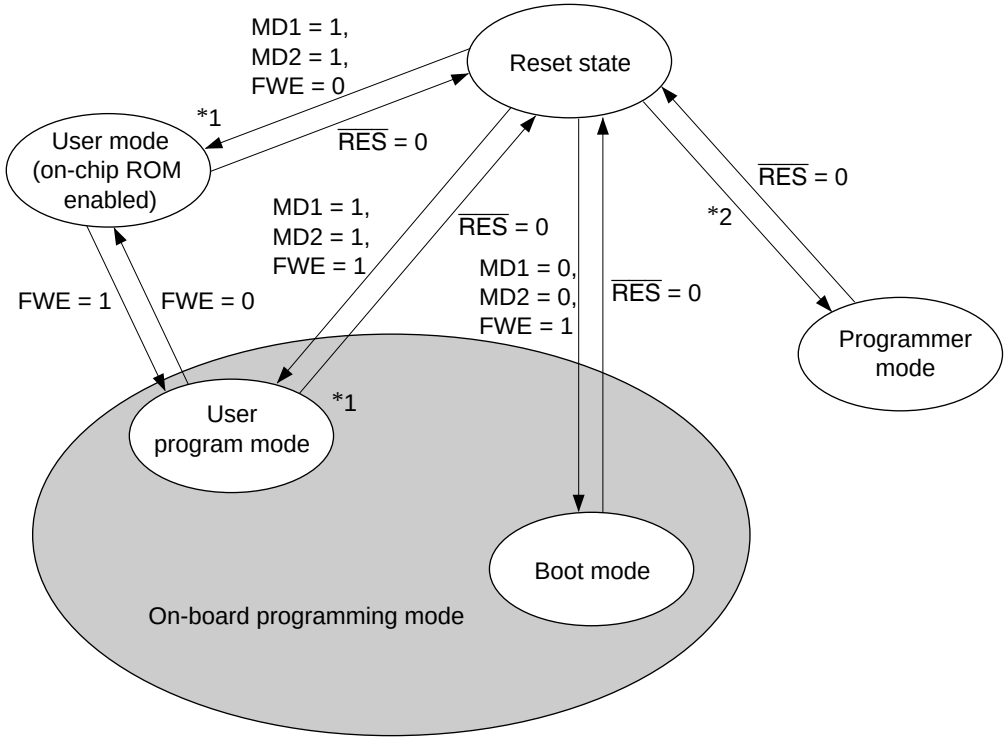
Legend
FLMCR1: Flash memory control register 1
FLMCR2: Flash memory control register 2
EBR1: Erase block register 1
EBR2: Erase block register 2
RAMER: RAM emulation register
FLPWCR: Flash memory power control register

Figure 21B-2 Block Diagram of Flash Memory

21B.5.2 Mode Transitions

When the mode pins and the FWE pin are set in the reset state and a reset-start is executed, the microcomputer enters an operating mode as shown in figure 21B-3. In user mode, flash memory can be read but not programmed or erased.

The boot, user program and programmer modes are provided as modes to write and erase the flash memory.



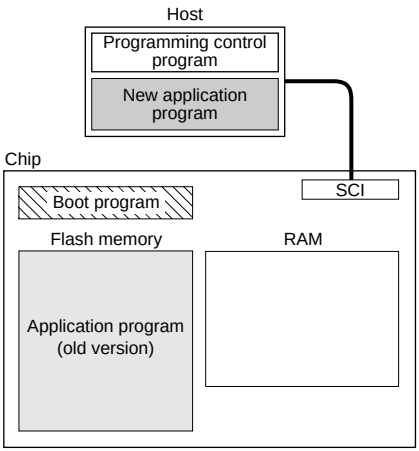
Notes: Only make a transition between user mode and user program mode when the CPU is not accessing the flash memory.

*1 RAM emulation possible
*2 MD0 = 0, MD1 = 0, MD2 = 0, P14 = 0, P16 = 0, PF0 = 1

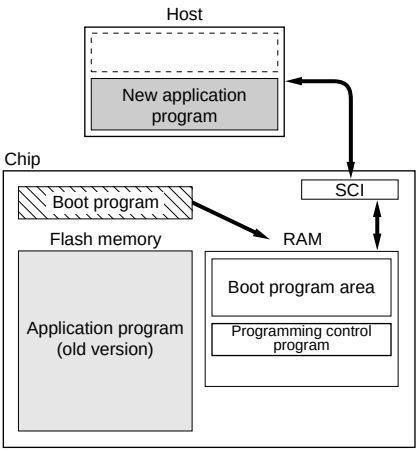
Figure 21B-3 Flash Memory State Transitions

Boot Mode

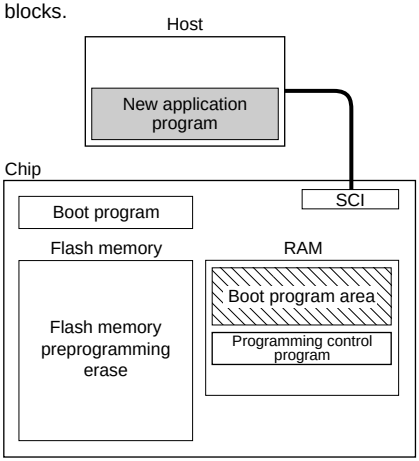
1. Initial state
   The old program version or data remains written in the flash memory. The user should prepare the programming control program and new application program beforehand in the host.



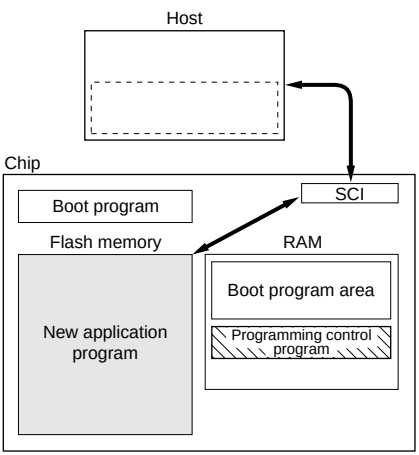
2. Programming control program transfer
   When boot mode is entered, the boot program in the chip (originally incorporated in the chip) is started and the programming control program in the host is transferred to RAM via SCI communication. The boot program required for flash memory erasing is automatically transferred to the RAM boot program area.



3. Flash memory initialization
   The erase program in the boot program area (in RAM) is executed, and the flash memory is initialized (to H'FF). In boot mode, total flash memory erasure is performed, without regard to blocks.

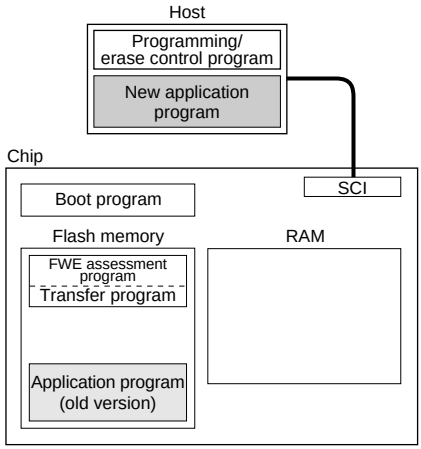


4. Writing new application program
   The programming control program transferred from the host to RAM is executed, and the new application program in the host is written into the flash memory.

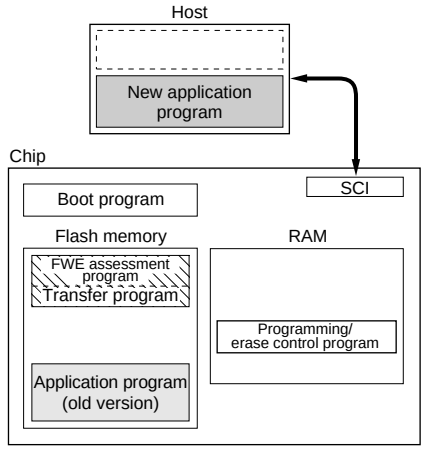


Program execution state

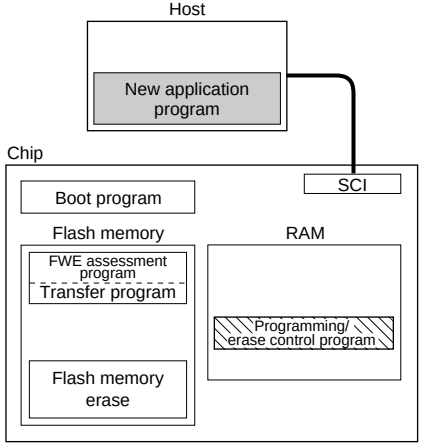
1. Initial state
   The FWE assessment program that confirms that user program mode has been entered, and the program that will transfer the programming/erase control program from flash memory to on-chip RAM should be written into the flash memory by the user beforehand. The programming/erase control program should be prepared in the host or in the flash memory.



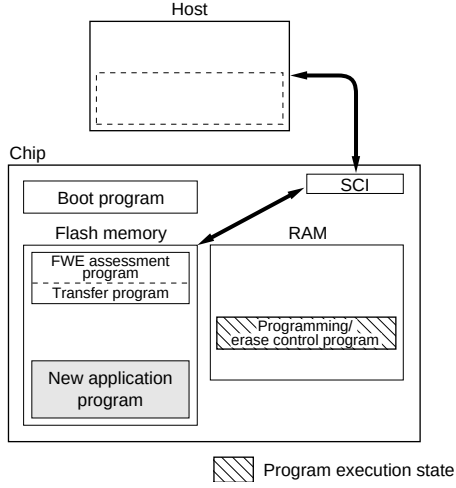
2. Programming/erase control program transfer
   When user program mode is entered, user software confirms this fact, executes transfer program in the flash memory, and transfers the programming/erase control program to RAM.



3. Flash memory initialization
   The programming/erase program in RAM is executed, and the flash memory is initialized (to H'FF). Erasing can be performed in block units, but not in byte units.



4. Writing new application program
   Next, the new application program in the host is written into the erased flash memory blocks. Do not write to unerased blocks.



21B.5.4 Flash Memory Emulation in RAM

Emulation should be performed in user mode or user program mode. When the emulation block set in RAMER is accessed while the emulation function is being executed, data written in the overlap RAM is read.

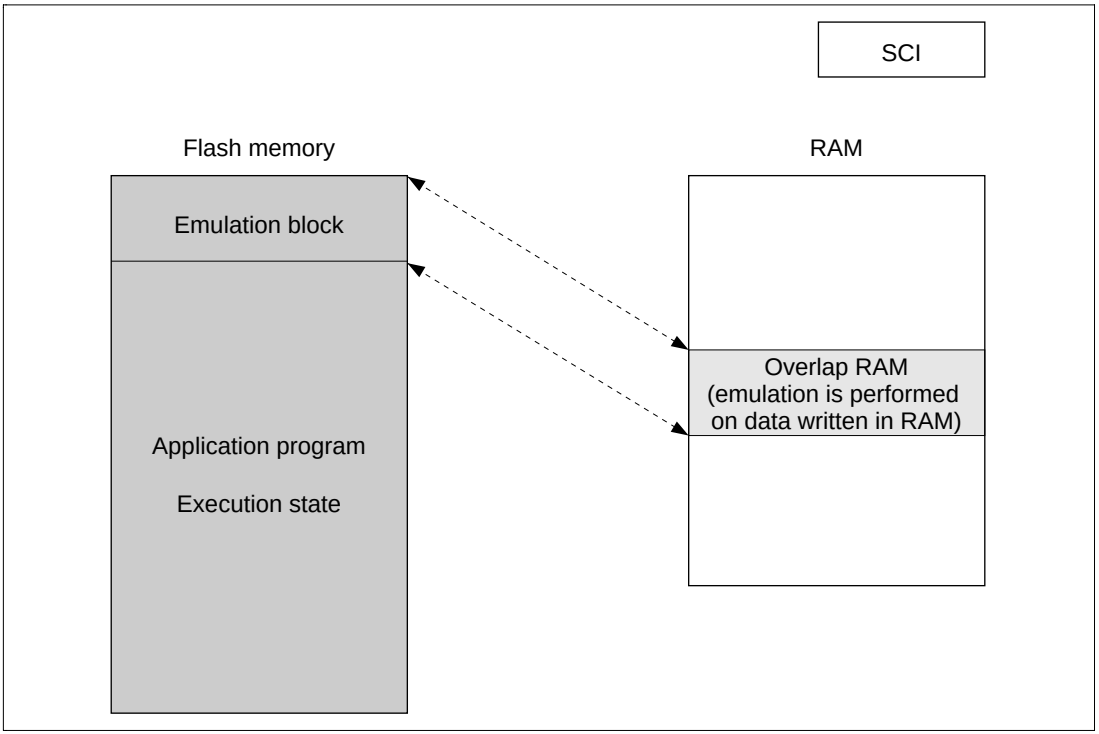


Figure 21B-4 Reading Overlap RAM Data in User Mode or User Program Mode

When overlap RAM data is confirmed, the RAMS bit is cleared, RAM overlap is released, and writes should actually be performed to the flash memory.

When the programming control program is transferred to RAM, ensure that the transfer destination and the overlap RAM do not overlap, as this will cause data in the overlap RAM to be rewritten.

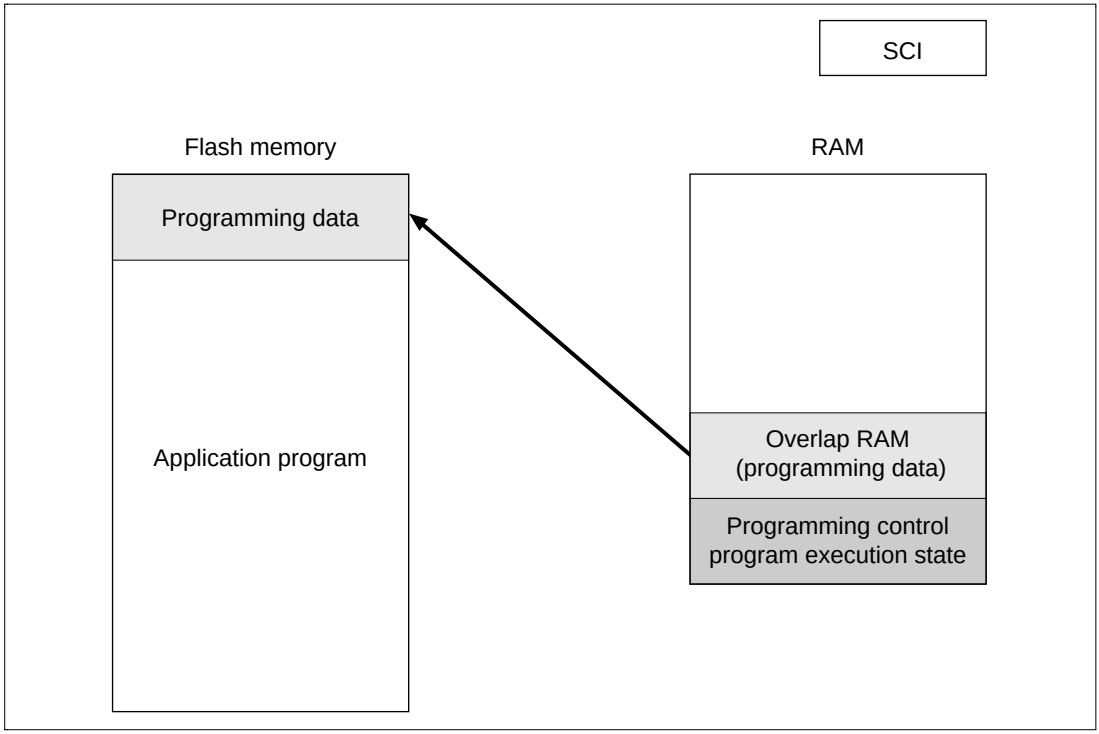


Figure 21B-5 Writing Overlap RAM Data in User Program Mode

21B.5.5 Differences between Boot Mode and User Program Mode

Table 21B-4 Differences between Boot Mode and User Program Mode

	Boot Mode	User Program Mode
Total erase	Yes	Yes
Block erase	No	Yes
Programming control program*	Program/program-verify	Erase/erase-verify Program/program-verify Emulation

Note: * To be provided by the user, in accordance with the recommended algorithm.

21B.5.6 Block Configuration

The flash memory is divided into three 64 kbytes blocks, one 32 kbytes block, and eight 4 kbytes blocks.

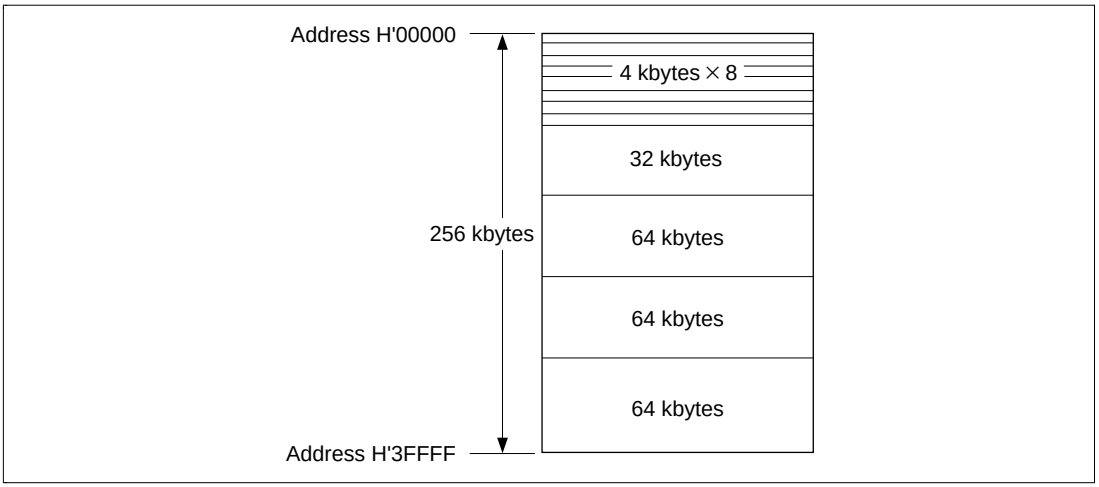


Figure 21B-6 Flash Memory Block Configuration

21B.6 Pin Configuration

The flash memory is controlled by means of the pins shown in table 21B-5.

Table 21B-5 Pin Configuration

Pin Name	Abbreviation	I/O	Function
Reset	$\overline{\text{RES}}$	Input	Reset
Flash write enable	FWE	Input	Flash memory program/erase protection by hardware
Mode 2	MD2	Input	Sets MCU operating mode
Mode 1	MD1	Input	Sets MCU operating mode
Mode 0	MD0	Input	Sets MCU operating mode
Port F0	PF0	Input	Sets MCU operating mode in programmer mode
Port 16	P16	Input	Sets MCU operating mode in programmer mode
Port 14	P14	Input	Sets MCU operating mode in programmer mode
Transmit data	TxD1	Output	Serial transmit data output
Receive data	RxD1	Input	Serial receive data input

21B.7 Register Configuration

The registers used to control the on-chip flash memory when enabled are shown in table 21B-6.

Table 21B-6 Register Configuration

Register Name	Abbreviation	R/W	Initial Value	Address*1
Flash memory control register 1	FLMCR1*4	R/W	H'00*2	H'FFA8
Flash memory control register 2	FLMCR2*4	R	H'00	H'FFA9
Erase block register 1	EBR1*4	R/W	H'00*3	H'FFAA
Erase block register 2	EBR2*4	R/W	H'00*3	H'FFAB
RAM emulation register	RAMER*4	R/W	H'00	H'FEDB
Flash memory power control register	FLPWCR*4	R/W	H'00*3	H'FFAC

Notes: *1 Lower 16 bits of the address.
*2 When a high level is input to the FWE pin, the initial value is H'80.
*3 When a low level is input to the FWE pin, or if a high level is input and the SWE1 bit in FLMCR1 is not set, these registers are initialized to H'00.
*4 FLMCR1, FLMCR2, EBR1, and EBR2, RAMER, and FLPWCR are 8-bit registers. Use byte access on these registers.

21B.8 Register Descriptions

21B.8.1 Flash Memory Control Register 1 (FLMCR1)

FLMCR1 is an 8-bit register used for flash memory operating mode control. Program-verify mode or erase-verify mode for addresses H'00000 to H'3FFFF is entered by setting SWE bit to 1 when FWE = 1, then setting the PV or EV bit. Program mode for addresses H'00000 to H'3FFFF is entered by setting SWE bit to 1 when FWE = 1, then setting the PSU bit, and finally setting the P bit. Erase mode for addresses H'00000 to H'3FFFF is entered by setting SWE bit to 1 when FWE = 1, then setting the ESU bit, and finally setting the E bit. FLMCR1 is initialized by a reset, and in hardware standby mode and software standby mode. Its initial value is H'80 when a high level is input to the FWE pin, and H'00 when a low level is input. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

Writes are enabled only in the following cases: Writes to bit SWE of FLMCR1 enabled when FWE = 1, to bits ESU, PSU, EV, and PV when FWE = 1 and SWE = 1, to bit E when FWE = 1, SWE = 1 and ESU = 1, and to bit P when FWE = 1, SWE = 1, and PSU = 1.

Bit:	7	6	5	4	3	2	1	0
	FWE	SWE	ESU	PSU	EV	PV	E	P
Initial value:	—*	0	0	0	0	0	0	0
R/W:	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Determined by the state of the FWE pin.

Bit 7—Flash Write Enable Bit (FWE): Sets hardware protection against flash memory programming/erasing.

Bit 7

FWE	Description
0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Bit 6—Software Write Enable Bit (SWE): This bit selects write and erase valid/invalid of the flash memory. Set it when setting bits 5 to 0, bits 7 to 0 of EBR1, and bits 3 to 0 of EBR2.

Bit 6

SWE	Description	
0	Writes disabled	(Initial value)
1	Writes enabled [Setting condition] When FWE = 1	

Bit 5—Erase Setup Bit (ESU): Prepares for a transition to erase mode. Set this bit to 1 before setting the E bit in FLMCR1 to 1. Do not set the SWE, PSU, EV, PV, E, or P bit at the same time.

Bit 5

ESU	Description	
0	Erase setup cleared	(Initial value)
1	Erase setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 4—Program Setup Bit (PSU): Prepares for a transition to program mode. Set this bit to 1 before setting the P bit in FLMCR1 to 1. Do not set the SWE, ESU, EV, PV, E, or P bit at the same time.

Bit 4		
PSU	**Description**	
0	Program setup cleared	(Initial value)
1	Program setup [Setting condition] When FWE = 1 and SWE = 1	

Bit 3—Erase-Verify (EV): Selects erase-verify mode transition or clearing. Do not set the SWE, ESU, PSU, PV, E, or P bit at the same time.

Bit 3		
EV	**Description**	
0	Erase-verify mode cleared	(Initial value)
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 2—Program-Verify (PV): Selects program-verify mode transition or clearing. Do not set the SWE, ESU, PSU, EV, E, or P bit at the same time.

Bit 2		
PV	**Description**	
0	Program-verify mode cleared	(Initial value)
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1	

Bit 1—Erase (E): Selects erase mode transition or clearing. Do not set the SWE, ESU, PSU, EV, PV, or P bit at the same time.

Bit 1		
E	**Description**	
0	Erase mode cleared	(Initial value)
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU = 1	

Bit 0—Program (P): Selects program mode transition or clearing. Do not set the SWE, PSU, ESU, EV, PV, or E bit at the same time.

Bit 0

P	Description	
0	Program mode cleared	(Initial value)
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU = 1	

21B.8.2 Flash Memory Control Register 2 (FLMCR2)

FLMCR2 is an 8-bit register used for flash memory operating mode control. FLMCR2 is initialized to H'00 by a reset, and in hardware standby mode and software standby mode. When on-chip flash memory is disabled, a read will return H'00.

Bit:	7	6	5	4	3	2	1	0
	FLER	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	—	—	—	—	—	—	—

Note: FLMCR2 is a read-only register, and should not be written to.

Bit 7—Flash Memory Error (FLER): Indicates that an error has occurred during an operation on flash memory (programming or erasing). When FLER is set to 1, flash memory goes to the error-protection state.

Bit 7

FLER	Description	
0	Flash memory is operating normally Flash memory program/erase protection (error protection) is disabled [Clearing condition] Reset or hardware standby mode	(Initial value)
1	An error has occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enabled [Setting condition] See 21B.8.3 Error Protection	

Bits 6 to 0—Reserved: These bits always read 0.

21B.8.3 Erase Block Register 1 (EBR1)

EBR1 is an 8-bit register that specifies the flash memory erase area block by block. EBR1 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin, and when a high level is input to the FWE pin and the SWE bit in FLMCR1 is not set. When a bit in EBR1 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit, as this will cause all the bits in both EBR1 and EBR2 to be automatically cleared to 0. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory erase block configuration is shown in table 21B-4.

Bit:	7	6	5	4	3	2	1	0
	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

21B.8.4 Erase Block Register 2 (EBR2)

EBR2 is an 8-bit register that specifies the flash memory erase area block by block. EBR2 is initialized to H'00 by a reset, in hardware standby mode and software standby mode, when a low level is input to the FWE pin. Bit 0 will be initialized to 0 if bit SWE of FLMCR1 is not set, even though a high level is input to pin FWE. When a bit in EBR2 is set to 1, the corresponding block can be erased. Other blocks are erase-protected. Only one of the bits of EBR1 and EBR2 combined can be set. Do not set more than one bit, as this will cause all the bits in both EBR1 and EBR2 to be automatically cleared to 0. Bits 7 to 4 are reserved and must only be written with 0. When on-chip flash memory is disabled, a read will return H'00, and writes are invalid.

The flash memory erase block configuration is shown in table 21B-7.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	EB11	EB10	EB9	EB8
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Table 21B-7 Flash Memory Erase Blocks

Block (Size)	Addresses
EB0 (4 kbytes)	H'000000–H'000FFF
EB1 (4 kbytes)	H'001000–H'001FFF
EB2 (4 kbytes)	H'002000–H'002FFF
EB3 (4 kbytes)	H'003000–H'003FFF
EB4 (4 kbytes)	H'004000–H'004FFF
EB5 (4 kbytes)	H'005000–H'005FFF
EB6 (4 kbytes)	H'006000–H'006FFF
EB7 (4 kbytes)	H'007000–H'007FFF
EB8 (32 kbytes)	H'008000–H'00FFFF
EB9 (64 kbytes)	H'010000–H'01FFFF
EB10 (64 kbytes)	H'020000–H'02FFFF
EB11 (64 kbytes)	H'030000–H'03FFFF

21B.8.5 RAM Emulation Register (RAMER)

RAMER specifies the area of flash memory to be overlapped with part of RAM when emulating real-time flash memory programming. RAMER initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode. RAMER settings should be made in user mode or user program mode.

Flash memory area divisions are shown in table 21B-5. To ensure correct operation of the emulation function, the ROM for which RAM emulation is performed should not be accessed immediately after this register has been modified. Normal execution of an access immediately after register modification is not guaranteed.

Bit:	7	6	5	4	3	2	1	0
	—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value:	0	0	0	0	0	0	0	0
R/W:	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Bits 7 and 6—Reserved: These bits always read 0.

Bits 5 and 4—Reserved: Only 0 may be written to these bits.

Bit 3—RAM Select (RAMS): Specifies selection or non-selection of flash memory emulation in RAM. When RAMS = 1, all flash memory block are program/erase-protected.

Bit 3

RAMS	Description	
0	Emulation not selected	(Initial value)
	Program/erase-protection of all flash memory blocks is disabled	
1	Emulation selected	
	Program/erase-protection of all flash memory blocks is enabled	

Bits 2 to 0—Flash Memory Area Selection: These bits are used together with bit 3 to select the flash memory area to be overlapped with RAM. (See table 21B-8.)

Table 21B-8 Flash Memory Area Divisions

Addresses	Block Name	RAMS	RAM1	RAM1	RAM0
H'FFD000–H'FFDFFF	RAM area 4 kbytes	0	*	*	*
H'000000–H'000FFF	EB0 (4 kbytes)	1	0	0	0
H'001000–H'001FFF	EB1 (4 kbytes)	1	0	0	1
H'002000–H'002FFF	EB2 (4 kbytes)	1	0	1	0
H'003000–H'003FFF	EB3 (4 kbytes)	1	0	1	1
H'004000–H'004FFF	EB4 (4 kbytes)	1	1	0	0
H'005000–H'005FFF	EB5 (4 kbytes)	1	1	0	1
H'006000–H'006FFF	EB6 (4 kbytes)	1	1	1	0
H'007000–H'007FFF	EB7 (4 kbytes)	1	1	1	1

*: Don't care

21B.8.6 Flash Memory Power Control Register (FLPWCR)

Bit:	7	6	5	4	3	2	1	0
	PDWND	—	—	—	—	—	—	—
Initial value:	0	0	0	0	0	0	0	0
R/W:	R/W	R	R	R	R	R	R	R

FLPWCR enables or disables a transition to the flash memory power-down mode when the LSI switches to subactive mode*.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are not available in versions other than the U-mask and W-mask versions.

Bit 7—Power-Down Disable (PDWND): The subactive mode is not available in versions other than the U-mask and W-mask versions.

Only 0 should be written to this bit in the case of versions other than the U-mask and W-mask versions. Only 0 should be written to this bit.

See section 21.B.14, Flash Memory and Power-Down States, for more information.

Bit 7

PDWND	Description	
0	Transition to flash memory power-down mode enabled	(Initial value)
1	Transition to flash memory power-down mode disabled	

Bits 6 to 0—Reserved: These bits always read 0.

21B.9 On-Board Programming Modes

When pins are set to on-board programming mode and a reset-start is executed, a transition is made to the on-board programming state in which program/erase/verify operations can be performed on the on-chip flash memory. There are two on-board programming modes: boot mode and user program mode. The pin settings for transition to each of these modes are shown in table 21B-9. For a diagram of the transitions to the various flash memory modes, see figure 21B-6.

Table 21B-9 Setting On-Board Programming Modes

Mode		FWE	MD2	MD1	MD0
Boot mode	Expanded mode	1	0	1	0
	Single-chip mode		0	1	1
User program mode	Expanded mode	1	1	1	0
	Single-chip mode		1	1	1

21B.9.1 Boot Mode

When boot mode is used, the flash memory programming control program must be prepared in the host beforehand. The SCI channel to be used is set to asynchronous mode.

When a reset-start is executed after the H8S/2638' pins have been set to boot mode, the boot program built into the H8S/2638 is started and the programming control program prepared in the host is serially transmitted to the H8S/2638 via the SCI. In the H8S/2638, the programming control program received via the SCI is written into the programming control program area in on-chip RAM. After the transfer is completed, control branches to the start address of the

programming control program area and the programming control program execution state is entered (flash memory programming is performed).

The transferred programming control program must therefore include coding that follows the programming algorithm given later.

The system configuration in boot mode is shown in figure 21B-7, and the boot mode execution procedure in figure 21B-8.

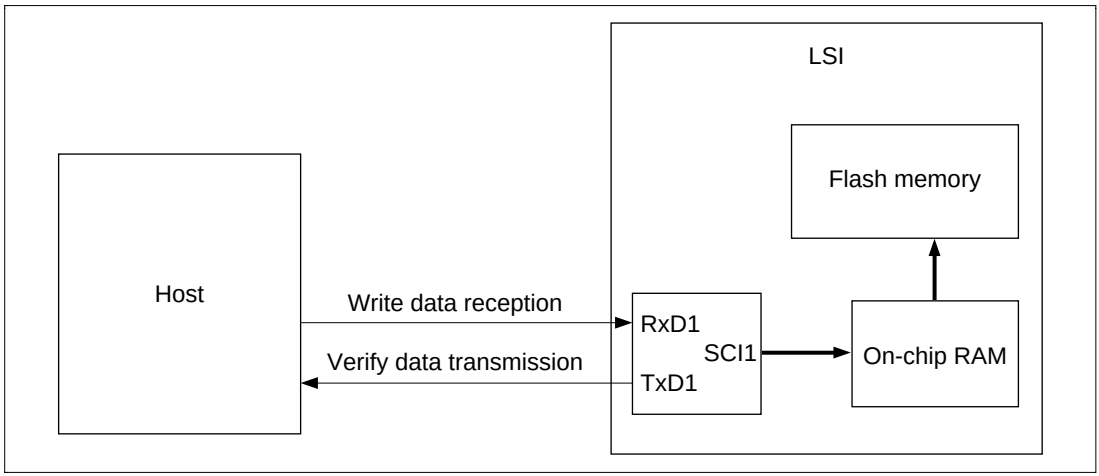
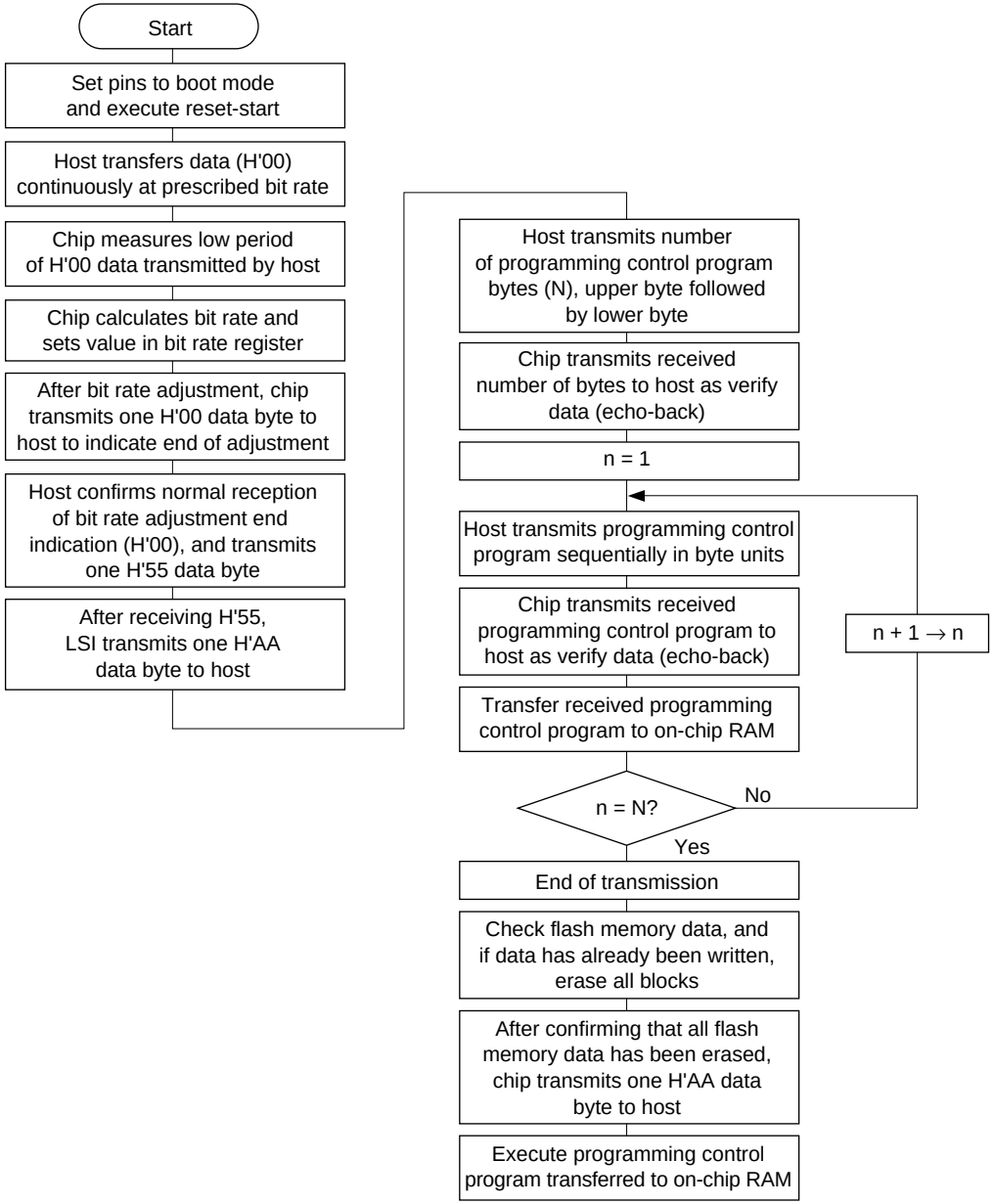


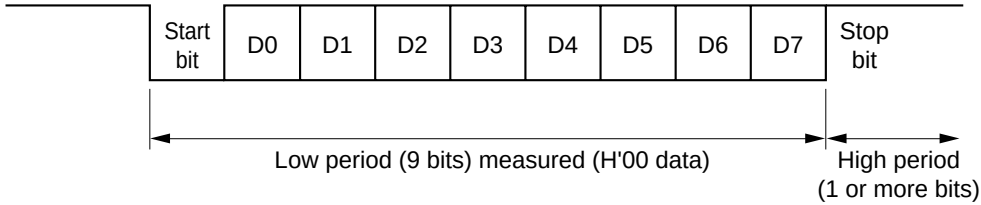
Figure 21B-7 System Configuration in Boot Mode



Note: If a memory cell does not operate normally and cannot be erased, one H'FF byte is transmitted as an erase error, and the erase operation and subsequent operations are halted.

Figure 21B-8 Boot Mode Execution Procedure

Automatic SCI Bit Rate Adjustment



When boot mode is initiated, the LSI measures the low period of the asynchronous SCI communication data (H'00) transmitted continuously from the host. The SCI transmit/receive format should be set as follows: 8-bit data, 1 stop bit, no parity. The LSI calculates the bit rate of the transmission from the host from the measured low period, and transmits one H'00 byte to the host to indicate the end of bit rate adjustment. The host should confirm that this adjustment end indication (H'00) has been received normally, and transmit one H'55 byte to the LSI. If reception cannot be performed normally, initiate boot mode again (reset), and repeat the above operations. Depending on the host's transmission bit rate and the LSI's system clock frequency, there will be a discrepancy between the bit rates of the host and the LSI. Set the host transfer bit rate at 4,800, 9,600 or 19,200 bps to operate the SCI properly.

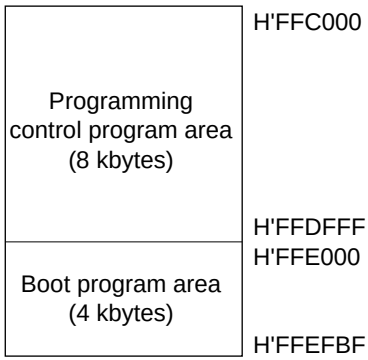
Table 21B-10 shows host transfer bit rates and system clock frequencies for which automatic adjustment of the LSI bit rate is possible. The boot program should be executed within this system clock range.

Table 21B-10 System Clock Frequencies for which Automatic Adjustment of LSI Bit Rate is Possible

Host Bit Rate	System Clock Frequency for Which Automatic Adjustment of LSI Bit Rate is Possible
4,800 bps	4 to 20 MHz
9,600 bps	8 to 20 MHz
19,200 bps	16 to 20 MHz

Note: The system clock frequency used in boot mode is generated by an external crystal oscillator element. PLL frequency multiplication is not used.

On-Chip RAM Area Divisions in Boot Mode: In boot mode, the RAM area is divided into an area used by the boot program and an area to which the programming control program is transferred via the SCI, as shown in figure 21B-9. The boot program area cannot be used until the execution state in boot mode switches to the programming control program transferred from the host.



Note: The boot program area cannot be used until a transition is made to the execution state for the programming control program transferred to RAM. Note also that the boot program remains in this area of the on-chip RAM even after control branches to the programming control program.

Figure 21B-9 RAM Areas in Boot Mode

Notes on Use of Boot Mode:

- When the chip comes out of reset in boot mode, it measures the low-level period of the input at the SCI's RxD1 pin. The reset should end with RxD1 high. After the reset ends, it takes approximately 100 states before the chip is ready to measure the low-level period of the RxD1 pin.
- In boot mode, if any data has been programmed into the flash memory (if all data is not 1), all flash memory blocks are erased. Boot mode is for use when user program mode is unavailable, such as the first time on-board programming is performed, or if the program activated in user program mode is accidentally erased.
- Interrupts cannot be used while the flash memory is being programmed or erased.
- The RxD1 and TxD1 pins should be pulled up on the board.
- Before branching to the programming control program (RAM area H'FFC000), the chip terminates transmit and receive operations by the on-chip SCI (channel 1) (by clearing the RE and TE bits in SCR to 0), but the adjusted bit rate value remains set in BRR. The transmit data output pin, TxD1, goes to the high-level output state (P33DDR = 1, P33DR = 1).

The contents of the CPU's internal general registers are undefined at this time, so these registers must be initialized immediately after branching to the programming control program. In particular, since the stack pointer (SP) is used implicitly in subroutine calls, etc., a stack area must be specified for use by the programming control program.

The initial values of other on-chip registers are not changed.

- Boot mode can be entered by making the pin settings shown in table 21B-9 and executing a reset-start.

  Boot mode can be cleared by driving the reset pin low, waiting at least 20 states, then setting the FWE pin and mode pins, and executing reset release*[1]. Boot mode can also be cleared by a WDT overflow reset.

  Do not change the mode pin input levels in boot mode, and do not drive the FWE pin low while the boot program is being executed or while flash memory is being programmed or erased*[2].

- If the mode pin input levels are changed (for example, from low to high) during a reset, the state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$) will change according to the change in the microcomputer's operating mode*[3].

  Therefore, care must be taken to make pin settings to prevent these pins from becoming output signal pins during a reset, or to prevent collision with signals outside the microcomputer.

Notes: *1 Mode pin and FWE pin input must satisfy the mode programming setup time (t_{MDS} = 4 states) with respect to the reset release timing.

*2 For precautions on applying and disconnecting FWE, see section 21B.16, Flash Memory Programming and Erasing Precautions.

*3 See appendix D, Pin States.

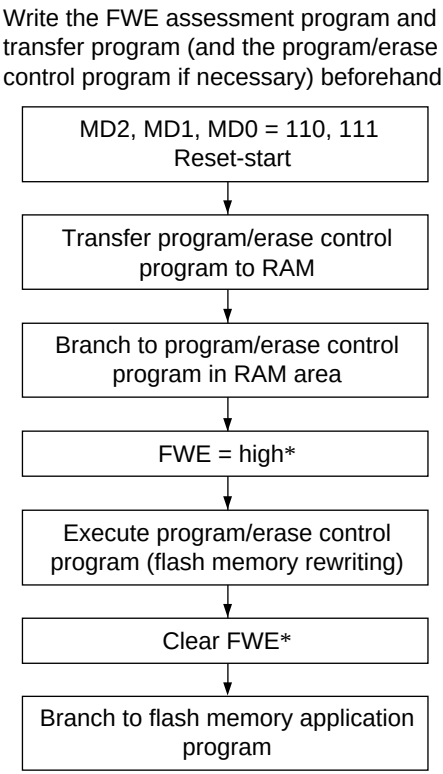
21B.9.2 User Program Mode

When set to user program mode, the chip can program and erase its flash memory by executing a user program/erase control program. Therefore, on-board reprogramming of the on-chip flash memory can be carried out by providing on-board means of FWE control and supply of programming data, and storing a program/erase control program in part of the program area as necessary.

To select user program mode, select a mode that enables the on-chip flash memory (mode 6 or 7), and apply a high level to the FWE pin. In this mode, on-chip supporting modules other than flash memory operate as they normally would in modes 6 and 7.

The flash memory itself cannot be read while the SWE bit is set to 1 to perform programming or erasing, so the control program that performs programming and erasing should be run in on-chip RAM or external memory. If the program is to be located in external memory, the instruction for writing to flash memory, and the following instruction, should be placed in on-chip RAM.

Figure 21B-10 shows the procedure for executing the program/erase control program when transferred to on-chip RAM.



Note: Do not apply a constant high level to the FWE pin. Apply a high level to the FWE pin only when the flash memory is programmed or erased. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

* For further information on FWE application and disconnection, see section 21B.16, Flash Memory Programming and Erasing Precautions.

Figure 21B-10 User Program Mode Execution Procedure

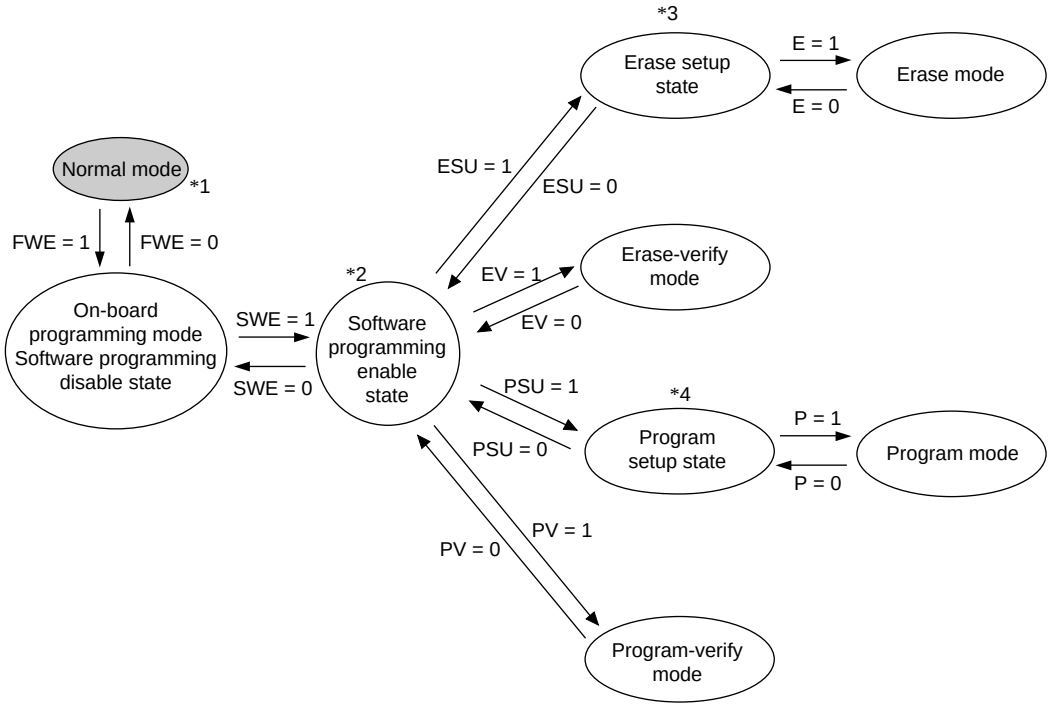
21B.10 Programming/Erasing Flash Memory

A software method, using the CPU, is employed to program and erase flash memory in the on-board programming modes. There are four flash memory operating modes: program mode, erase mode, program-verify mode, and erase-verify mode. Transitions to these modes are made by setting the PSU, ESU, P, E, PV, and EV bits in FLMCR1 for addresses H'000000 to H'03FFFF.

The flash memory cannot be read while it is being written or erased. The flash memory cannot be read while being programmed or erased. Therefore, the program (user program) that controls flash memory programming/erasing should be located and executed in on-chip RAM or external memory. If the program is to be located in external memory, the instruction for writing to flash memory, and the following instruction, should be placed in on-chip RAM. Also ensure that the DTC is not activated before or after execution of the flash memory write instruction.

In the following operation descriptions, wait times after setting or clearing individual bits in FLMCR1 are given as parameters; for details of the wait times, see section 24.2.7, Flash Memory Characteristics.

Notes:
1. Operation is not guaranteed if bits SWE, ESU, PSU, EV, PV, E, and P of FLMCR1 are set/reset by a program in flash memory in the corresponding address areas.
2. When programming or erasing, set FWE to 1 (programming/erasing will not be executed if FWE = 0).
3. Programming should be performed in the erased state. Do not perform additional programming on previously programmed addresses.



Notes: In order to perform a normal read of flash memory, SWE must be cleared to 0. Also note that verify-reads can be performed during the programming/erasing process.

*1 : Normal mode : On-board programming mode
*2 Do not make a state transition by setting or clearing multiple bits simultaneously.
*3 After a transition from erase mode to the erase setup state, do not enter erase mode without passing through the software programming enable state.
*4 After a transition from program mode to the program setup state, do not enter program mode without passing through the software programming enable state.

Figure 21B-11 FLMCR1 Bit Settings and State Transitions

21B.10.1 Program Mode

When writing data or programs to flash memory, the program/program-verify flowchart shown in figure 21B-12 should be followed. Performing programming operations according to this flowchart will enable data or programs to be written to flash memory without subjecting the device to voltage stress or sacrificing program data reliability. Programming should be carried out 128 bytes at a time.

The wait times after bits are set or cleared in the flash memory control register 1 (FLMCR1) and the maximum number of programming operations (N) are shown in table 24.10 in section 24.1.7, Flash Memory Characteristics.

Following the elapse of (t_{sswe}) µs or more after the SWE bit is set to 1 in FLMCR1, 128-byte data is written consecutively to the write addresses. The lower 8 bits of the first address written to must be H'00 and H'80, 128 consecutive byte data transfers are performed. The program address and program data are latched in the flash memory. A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.

Next, the watchdog timer (WDT) is set to prevent overprogramming due to program runaway, etc. Set a value greater than ($t_{spsu} + t_{sp} + t_{cp} + t_{cpsu}$) µs as the WDT overflow period. Preparation for entering program mode (program setup) is performed next by setting the PSU bit in FLMCR1. The operating mode is then switched to program mode by setting the P bit in FLMCR1 after the elapse of at least (t_{spsu}) µs. The time during which the P bit is set is the flash memory programming time. Make a program setting so that the time for one programming operation is within the range of (t_{sp}) µs.

The wait time after P bit setting must be changed according to the degree of progress through the programming operation. For details see “Notes on Program/Program-Verify Procedure.”

21B.10.2 Program-Verify Mode

In program-verify mode, the data written in program mode is read to check whether it has been correctly written in the flash memory.

After the elapse of the given programming time, clear the P bit in FLMCR1, then wait for at least (t_{cp}) µs before clearing the PSU bit to exit program mode. After exiting program mode, the watchdog timer setting is also cleared. The operating mode is then switched to program-verify mode by setting the PV bit in FLMCR1. Before reading in program-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (t_{spv}) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (t_{spvr}) µs after the dummy write before performing this read operation. Next, the originally written data is compared with the verify data, and reprogram data is computed (see figure 21B-12) and transferred to RAM. After verification of 128 bytes of data has been completed, exit program-verify mode, wait for at least (t_{cpv}) µs, then clear the SWE bit in FLMCR1. If reprogramming is necessary, set program mode again, and repeat the program/program-verify sequence as before. The maximum number of repetitions of the program/program-verify sequence is indicated by the maximum programming count (N). Leave a wait time of at least (t_{cswe}) µs after clearing SWE.

Notes on Program/Program-Verify Procedure

1. In order to perform 128-byte-unit programming, the lower 8 bits of the write start address must be H'00 or H'80.

2. When performing continuous writing of 128-byte data to flash memory, byte-unit transfer should be used.

   128-byte data transfer is necessary even when writing fewer than 128 bytes of data. Write H'FF data to the extra addresses.

3. Verify data is read in word units.

4. The write pulse is applied and a flash memory write executed while the P bit in FLMCR1 is set. In the chip, write pulses should be applied as follows in the program/program-verify procedure to prevent voltage stress on the device and loss of write data reliability.
   a. After write pulse application, perform a verify-read in program-verify mode and apply a write pulse again for any bits read as 1 (reprogramming processing). When all the 0-write bits in the 128-byte write data are read as 0 in the verify-read operation, the program/program-verify procedure is completed. In the chip, the number of loops in reprogramming processing is guaranteed not to exceed the maximum value of the maximum programming count (N).
   b. After write pulse application, a verify-read is performed in program-verify mode, and programming is judged to have been completed for bits read as 0. The following processing is necessary for programmed bits.

When programming is completed at an early stage in the program/program-verify procedure:

If programming is completed in the 1st to 6th reprogramming processing loop, additional programming should be performed on the relevant bits. Additional programming should only be performed on bits which first return 0 in a verify-read in certain reprogramming processing.

When programming is completed at a late stage in the program/program-verify procedure:

If programming is completed in the 7th or later reprogramming processing loop, additional programming is not necessary for the relevant bits.

c. If programming of other bits is incomplete in the 128 bytes, reprogramming processing should be executed. If a bit for which programming has been judged to be completed is read as 1 in a subsequent verify-read, a write pulse should again be applied to that bit.

5. The period for which the P bit in FLMCR1 is set (the write pulse width) should be changed according to the degree of progress through the program/program-verify procedure. For detailed wait time specifications, see section 24.1.7, Flash Memory Characteristics.

Item	Symbol	Item	Symbol
Wait time after P bit setting	t_{sp}	When reprogramming loop count (n) is 1 to 6	t_{sp30}
		When reprogramming loop count (n) is 7 or more	t_{sp200}
		In case of additional programming processing*	t_{sp10}

Note: * Additional programming processing is necessary only when the reprogramming loop count (n) is 1 to 6.

6. The program/program-verify flowchart for the H8S/2636 is shown in figure 21B-12.

To cover the points noted above, bits on which reprogramming processing is to be executed, and bits on which additional programming is to be executed, must be determined as shown below.

Since reprogram data and additional-programming data vary according to the progress of the programming procedure, it is recommended that the following data storage areas (128 bytes each) be provided in RAM.

Reprogram Data Computation Table

(D)	Result of Verify-Read after Write Pulse Application (V)	(X) Result of Operation	Comments
0	0	1	Programming completed: reprogramming processing not to be executed
0	1	0	Programming incomplete: reprogramming processing to be executed
1	0	1	—
1	1	1	Still in erased state: no action

Legend

(D): Source data of bits on which programming is executed

(X): Source data of bits on which reprogramming is executed

Additional-Programming Data Computation Table

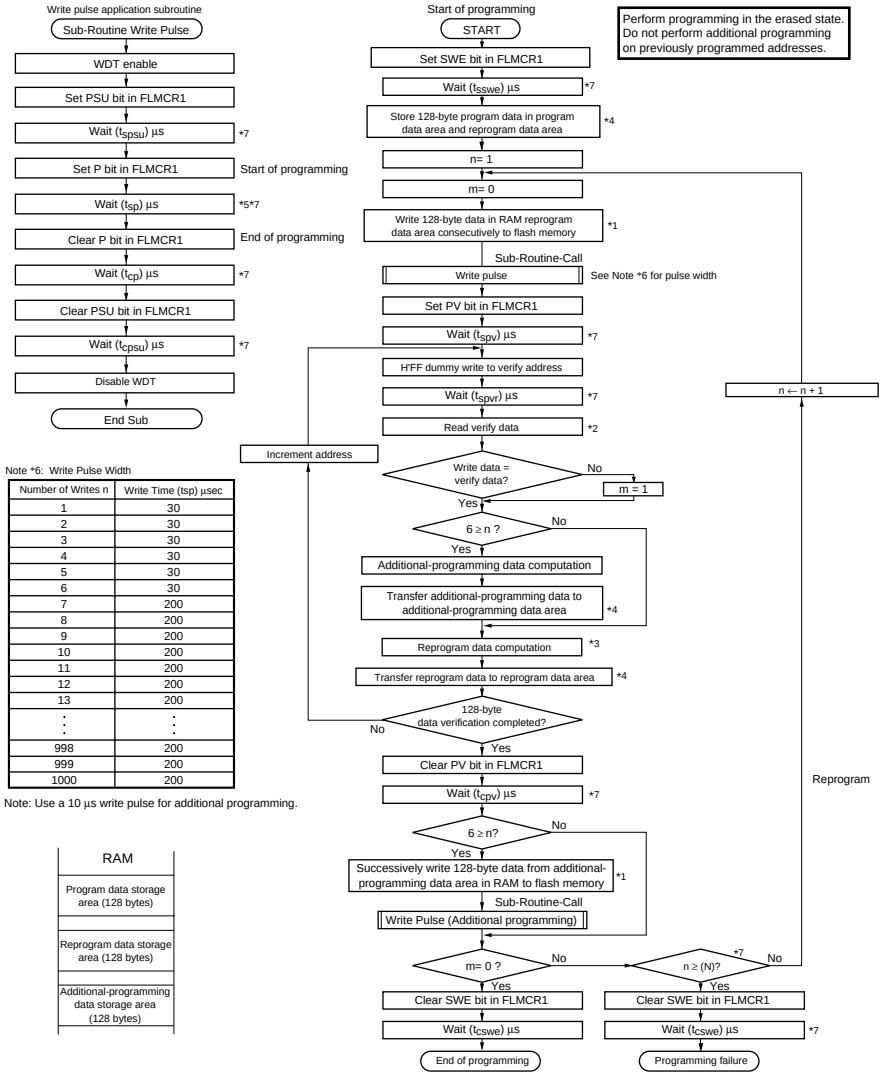
(X')	Result of Verify-Read after Write Pulse Application (V)	(Y) Result of Operation	Comments
0	0	0	Programming by write pulse application judged to be completed: additional programming processing to be executed
0	1	1	Programming by write pulse application incomplete: additional programming processing not to be executed
1	0	1	Programming already completed: additional programming processing not to be executed
1	1	1	Still in erased state: no action

Legend

(Y): Data of bits on which additional programming is executed

(X'): Data of bits on which reprogramming is executed in a certain reprogramming loop

7. It is necessary to execute additional programming processing during the course of the chip program/program-verify procedure. However, once 128-byte-unit programming is finished, additional programming should not be carried out on the same address area. When executing reprogramming, an erase must be executed first. Note that normal operation of reads, etc., is not guaranteed if additional programming is performed on addresses for which a program/program-verify operation has finished.



Note *6: Write Pulse Width

Number of Writes n	Write Time (tsp) µsec
1	30
2	30
3	30
4	30
5	30
6	30
7	200
8	200
9	200
10	200
11	200
12	200
13	200
⋮	⋮
998	200
999	200
1000	200

Note: Use a 10 µs write pulse for additional programming.

Notes: *1 Data transfer is performed by byte transfer. The lower 8 bits of the first address written to must be H'00 or H'80.
A 128-byte data transfer must be performed even if writing fewer than 128 bytes; in this case, H'FF data must be written to the extra addresses.
*2 Verify data is read in 16-bit (word) units.
*3 Reprogram data is determined by the operation shown in the table below (comparison between the data stored in the program data area and the verify data). Bits for which the reprogram data is 0 are programmed in the next reprogramming loop. Therefore, even bits for which programming has been completed will be subjected to programming once again if the result of the subsequent verify operation is NG.
*4 A 128-byte area for storing program data, a 128-byte area for storing reprogram data, and a 128-byte area for storing additional data must be provided in RAM. The contents of the reprogram data area and additional data area are modified as programming proceeds.
*5 A write pulse of 30 µs or 200 µs is applied according to the progress of the programming operation. See Note *6 for details of the pulse widths. When writing of additional-programming data is executed, a 10 µs write pulse should be applied. Reprogram data X' means reprogram data when the write pulse is applied.
*7 The wait times and value of N are shown in section 24.1.7, Flash Memory characteristics.

Reprogram Data Computation Table

Original Data (D)	Verify Data (V)	Reprogram Data (X)	Comments
0	0	1	Programming completed
0	1	0	Programming incomplete; reprogram
1	0	1	
1	1	1	Still in erased state; no action

Additional-Programming Data Computation Table

Reprogram Data (X')	Verify Data (V)	Additional-Programming Data (Y)	Comments
0	0	0	Additional programming to be executed
0	1	1	Additional programming not to be executed
1	0	1	Additional programming not to be executed
1	1	1	Additional programming not to be executed

Figure 21B-12 Program/Program-Verify Flowchart

21B.10.3 Erase Mode

When erasing flash memory, the single-block erase flowchart shown in figure 21B-13 should be followed.

The wait times after bits are set or cleared in the flash memory control register 1 (FLMCR1) and the maximum number of erase operations (N) are shown in table 24.10 in section 24.1.7, Flash Memory Characteristics.

To erase flash memory contents, make a 1-bit setting for the flash memory area to be erased in erase block register 1 and 2 (EBR1, EBR2) at least (t_{sswe}) µs after setting the SWE bit to 1 in FLMCR1. Next, the watchdog timer (WDT) is set to prevent overerasing due to program runaway, etc. Set a value greater than (t_{se}) ms + (t_{sesu} + t_{ce} + t_{cesu}) µs as the WDT overflow period. Preparation for entering erase mode (erase setup) is performed next by setting the ESU bit in FLMCR1. The operating mode is then switched to erase mode by setting the E bit in FLMCR1 after the elapse of at least (t_{sesu}) µs. The time during which the E bit is set is the flash memory erase time. Ensure that the erase time does not exceed (t_{se}) ms.

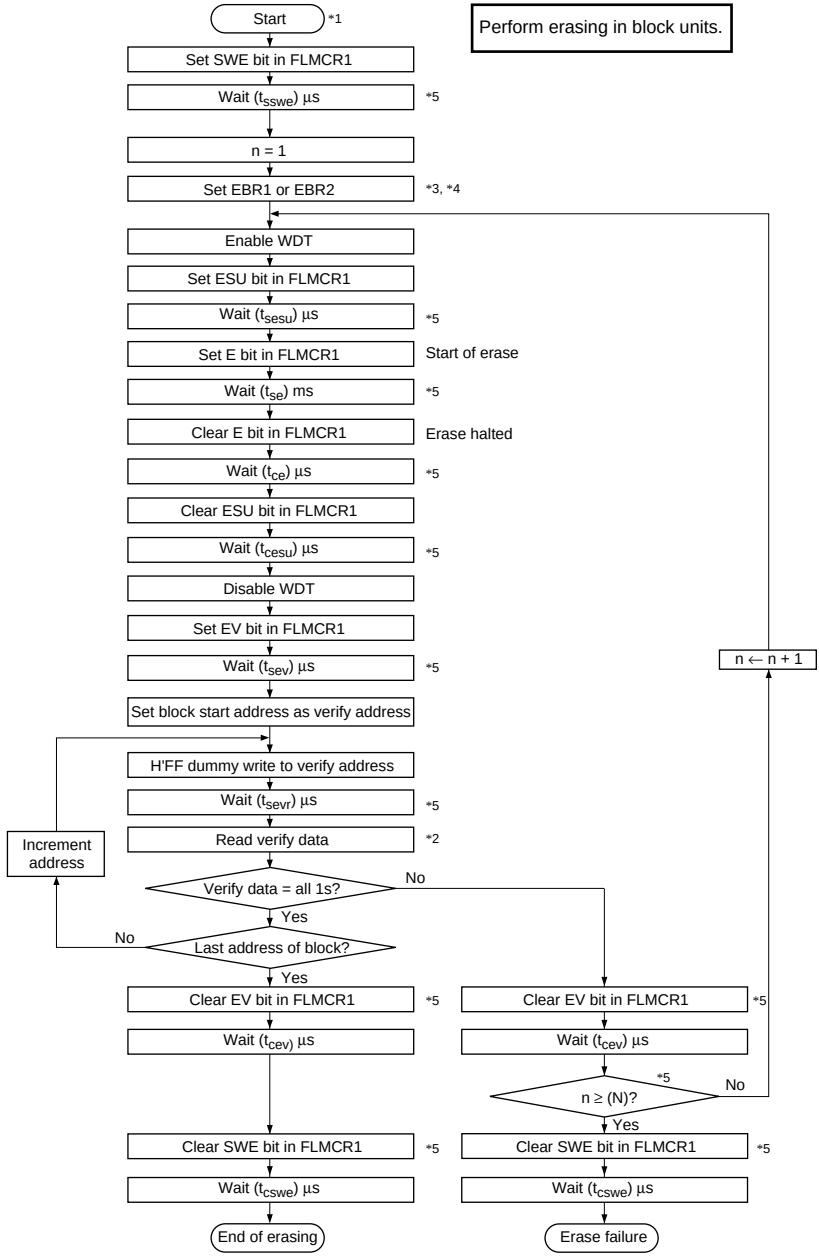
Note: With flash memory erasing, preprogramming (setting all memory data in the memory to be erased to all 0) is not necessary before starting the erase procedure.

21B.10.4 Erase-Verify Mode

In erase-verify mode, data is read after memory has been erased to check whether it has been correctly erased.

After the elapse of the fixed erase time, clear the E bit in FLMCR1, then wait for at least (t_{ce}) µs before clearing the ESU bit to exit erase mode. After exiting erase mode, the watchdog timer setting is also cleared. The operating mode is then switched to erase-verify mode by setting the EV bit in FLMCR1. Before reading in erase-verify mode, a dummy write of H'FF data should be made to the addresses to be read. The dummy write should be executed after the elapse of (t_{sev}) µs or more. When the flash memory is read in this state (verify data is read in 16-bit units), the data at the latched address is read. Wait at least (t_{sevr}) µs after the dummy write before performing this read operation. If the read data has been erased (all 1), a dummy write is performed to the next address, and erase-verify is performed. If the read data is unerased, set erase mode again, and repeat the erase/erase-verify sequence as before. The maximum number of repetitions of the erase/erase-verify sequence is indicated by the maximum erase count (N). When verification is completed, exit erase-verify mode, and wait for at least (t_{cev}) µs. If erasure has been completed on all the erase blocks, clear the SWE bit in FLMCR1, and leave a wait time of at least (t_{cswe}) µs.

If erasing multiple blocks, set a single bit in EBR1/EBR2 for the next block to be erased, and repeat the erase/erase-verify sequence as before.



Notes: *1 Prewriting (setting erase block data to all 0s) is not necessary.
*2 Verify data is read in 16-bit (word) units.
*3 Make only a single-bit specification in the erase block registers (EBR1 and EBR2). Two or more bits must not be set simultaneously.
*4 Erasing is performed in block units. To erase multiple blocks, each block must be erased in turn.
*5 The wait times and the value of N are shown in section 24.1.7, Flash Memory Characteristics.

Figure 21B-13 Erase/Erase-Verify Flowchart

21B.11 Protection

There are three kinds of flash memory program/erase protection: hardware protection, software protection, and error protection.

21B.11.1 Hardware Protection

Hardware protection refers to a state in which programming/erasing of flash memory is forcibly disabled or aborted. Hardware protection is reset by settings in flash memory control register 1 (FLMCR1), flash memory control register 2 (FLMCR2), erase block register 1 (EBR1), and erase block register 2 (EBR2). The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained in the error-protected state. (See table 21B-11.)

Table 21B-11 Hardware Protection

		Functions	
Item	**Description**	**Program**	**Erase**
FWE pin protection	• When a low level is input to the FWE pin, FLMCR1, FLMCR2, (except bit FLER) EBR1, and EBR2 are initialized, and the program/erase-protected state is entered.	Yes	Yes
Reset/standby protection	• In a reset (including a WDT reset) and in standby mode, FLMCR1, FLMCR2, EBR1, and EBR2 are initialized, and the program/erase-protected state is entered. • In a reset via the $\overline{\text{RES}}$ pin, the reset state is not entered unless the $\overline{\text{RES}}$ pin is held low until oscillation stabilizes after powering on. In the case of a reset during operation, hold the $\overline{\text{RES}}$ pin low for the $\overline{\text{RES}}$ pulse width specified in the AC Characteristics section.	Yes	Yes

21B.11.2 Software Protection

Software protection can be implemented by setting the SWE bit in FLMCR1, erase block register 1 (EBR1), erase block register 2 (EBR2), and the RAMS bit in the RAM emulation register (RAMER). When software protection is in effect, setting the P or E bit in flash memory control register 1 (FLMCR1), does not cause a transition to program mode or erase mode. (See table 21B-12.)

Table 21B-12 Software Protection

Item	Description	Functions	
		Program	Erase
SWE bit protection	• Setting bit SWE in FLMCR1 to 0 will place area H'000000 to H'03FFFF in the program/erase-protected state. (Execute the program in the on-chip RAM, external memory)	Yes	Yes
Block specification protection	• Erase protection can be set for individual blocks by settings in erase block register 1 (EBR1) and erase block register 2 (EBR2). • Setting EBR1 and EBR2 to H'00 places all blocks in the erase-protected state.	—	Yes
Emulation protection	• Setting the RAMS bit to 1 in the RAM emulation register (RAMER) places all blocks in the program/erase-protected state.	Yes	Yes

21B.11.3 Error Protection

In error protection, an error is detected when chip runaway occurs during flash memory programming/erasing, or operation is not performed in accordance with the program/erase algorithm, and the program/erase operation is aborted. Aborting the program/erase operation prevents damage to the flash memory due to overprogramming or overerasing.

If the chip malfunctions during flash memory programming/erasing, the FLER bit is set to 1 in FLMCR2 and the error protection state is entered. The FLMCR1, FLMCR2, EBR1, and EBR2 settings are retained, but program mode or erase mode is aborted at the point at which the error occurred. Program mode or erase mode cannot be re-entered by re-setting the P or E bit. However, PV and EV bit setting is enabled, and a transition can be made to verify mode.

FLER bit setting conditions are as follows:

1. When the flash memory of the relevant address area is read during programming/erasing (including vector read and instruction fetch)
2. Immediately after exception handling (excluding a reset) during programming/erasing
3. When a SLEEP instruction (including software standby) is executed during programming/erasing
4. When the CPU releases the bus to the DTC

Error protection is released only by a reset and in hardware standby mode.

Figure 21B-14 shows the flash memory state transition diagram.

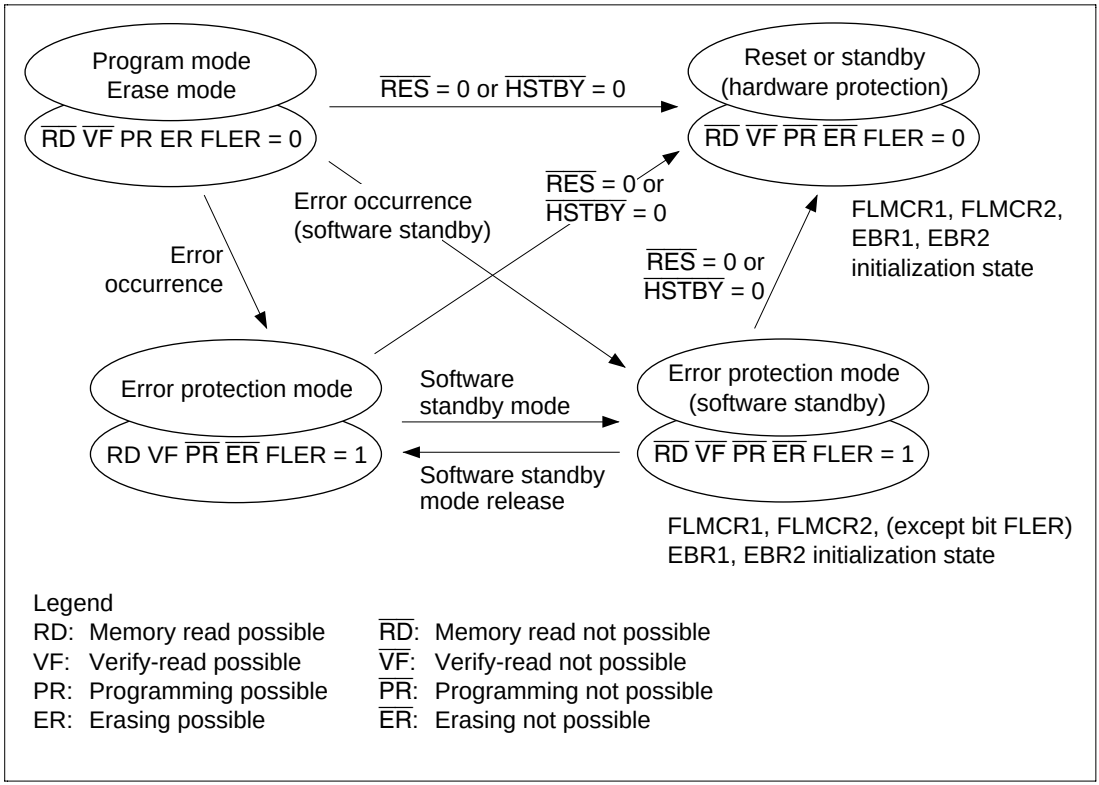


Figure 21B-14 Flash Memory State Transitions

21B.12 Flash Memory Emulation in RAM

Making a setting in the RAM emulation register (RAMER) enables part of RAM to be overlapped onto the flash memory area so that data to be written to flash memory can be emulated in RAM in real time. After the RAMER setting has been made, accesses cannot be made from the flash memory area or the RAM area overlapping flash memory. Emulation can be performed in user mode and user program mode. Figure 21B-15 shows an example of emulation of real-time flash memory programming.

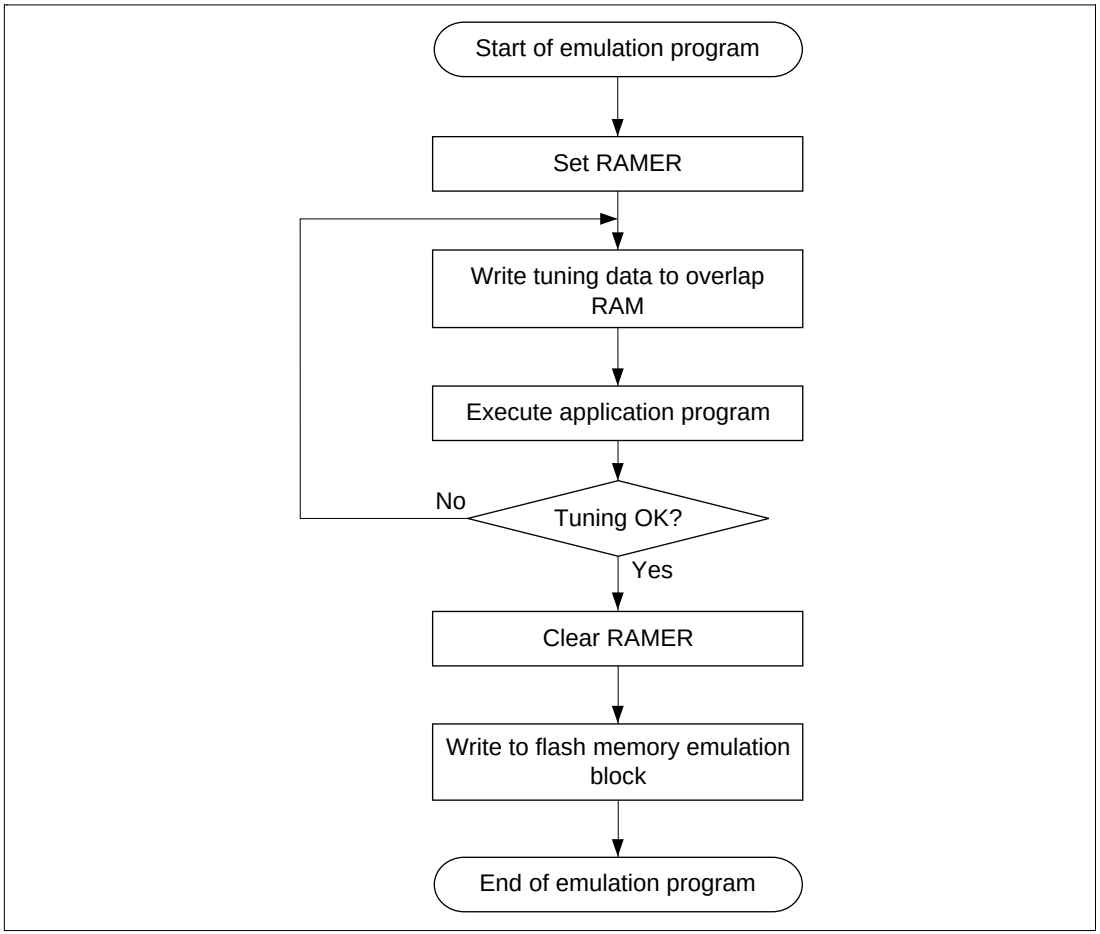


Figure 21B-15 Flowchart for Flash Memory Emulation in RAM

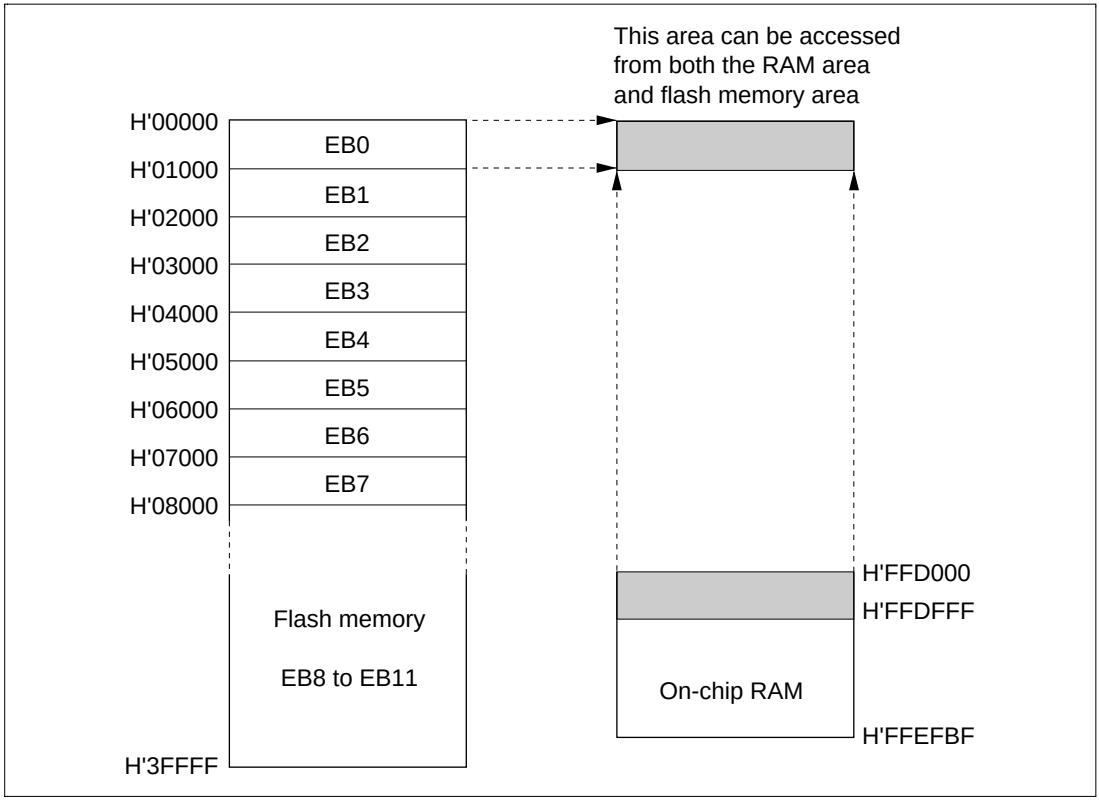


Figure 21B-16 Example of RAM Overlap Operation

Example in which Flash Memory Block Area EB0 is Overlapped

1. Set bits RAMS, RAM2 to RAM0 in RAMER to 1, 0, 0, 0, to overlap part of RAM onto the area (EB0) for which real-time programming is required.
2. Real-time programming is performed using the overlapping RAM.
3. After the program data has been confirmed, the RAMS bit is cleared, releasing RAM overlap.
4. The data written in the overlapping RAM is written into the flash memory space (EB0).

Notes: 1. When the RAMS bit is set to 1, program/erase protection is enabled for all blocks regardless of the value of RAM2 to RAM0 (emulation protection). In this state, setting the P or E bit in flash memory control register 1 (FLMCR1), will not cause a transition to program mode or erase mode. When actually programming or erasing a flash memory area, the RAMS bit should be cleared to 0.

2. A RAM area cannot be erased by execution of software in accordance with the erase algorithm while flash memory emulation in RAM is being used.
3. Block area EB0 contains the vector table. When performing RAM emulation, the vector table is needed in the overlap RAM.

21B.13 Interrupt Handling when Programming/Erasing Flash Memory

All interrupts, including NMI interrupt is disabled when flash memory is being programmed or erased (when the P or E bit is set in FLMCR1), and while the boot program is executing in boot mode*[1], to give priority to the program or erase operation. There are three reasons for this:

1. Interrupt during programming or erasing might cause a violation of the programming or erasing algorithm, with the result that normal operation could not be assured.
2. In the interrupt exception handling sequence during programming or erasing, the vector would not be read correctly*[2], possibly resulting in MCU runaway.
3. If interrupt occurred during boot program execution, it would not be possible to execute the normal boot mode sequence.

For these reasons, in on-board programming mode alone there are conditions for disabling interrupt, as an exception to the general rule. However, this provision does not guarantee normal erasing and programming or MCU operation. All requests, including NMI interrupt, must therefore be restricted inside and outside the MCU when programming or erasing flash memory. NMI interrupt is also disabled in the error-protection state while the P or E bit remains set in FLMCR1.

Notes: *1 Interrupt requests must be disabled inside and outside the MCU until the programming control program has completed programming.

*2 The vector may not be read correctly in this case for the following two reasons:

- If flash memory is read while being programmed or erased (while the P or E bit is set in FLMCR1), correct read data will not be obtained (undetermined values will be returned).
- If the interrupt entry in the vector table has not been programmed yet, interrupt exception handling will not be executed correctly.

21B.14 Flash Memory Programmer Mode

Programs and data can be written and erased in programmer mode as well as in the on-board programming modes. In programmer mode, flash memory read mode, auto-program mode, auto-erase mode, and status read mode are supported. In auto-program mode, auto-erase mode, and status read mode, a status polling procedure is used, and in status read mode, detailed internal signals are output after execution of an auto-program or auto-erase operation.

In programmer mode, set the mode pins to programmer mode (see table 21B-13) and input a 12 MHz input clock.

Table 21B-13 shows the pin settings for programmer mode.

Table 21B-13 Programmer Mode Pin Settings

Pin Names	Settings
Mode pins: MD2, MD1, MD0	Low level input to MD2, MD1, and MD0.
Mode setting pins: PF0, P16, P14	High level input to PF0, low level input to P16 and P14
FWE pin	High level input (in auto-program and auto-erase modes)
$\overline{\text{RES}}$ pin	Reset circuit
XTAL, EXTAL, PLLCAP, PLLVSS pins	Oscillator circuit
VCL	Internal voltage step-down circuit

21B.14.1 Socket Adapter and Memory Map

In programmer mode in which the PROM writer is used, reading from memory (verification), writing, and initializing the flash memory (erasing all of its contents) are enabled. At this time, a dedicated conversion socket adapter must be attached to a general-purpose PROM writer. Table 21B-14 shows the types of the socket adapters. For programmer mode on this LSI, one of the socket adapters listed in table 21B-14 should be used.

Table 21B-14 Type of Socket Adapter

Product Name	Package Type	Socket Adapter Type	Manufacturer
HD64F2638F HD64F2638UF HD64F2638WF HD64F2639UF HD64F2639WF	128 pin QFP (FP-128B)	ME2638ESHF1H	Minato Electronics Inc.
		HF2638Q128D4001	Data I/O Japan Corporation

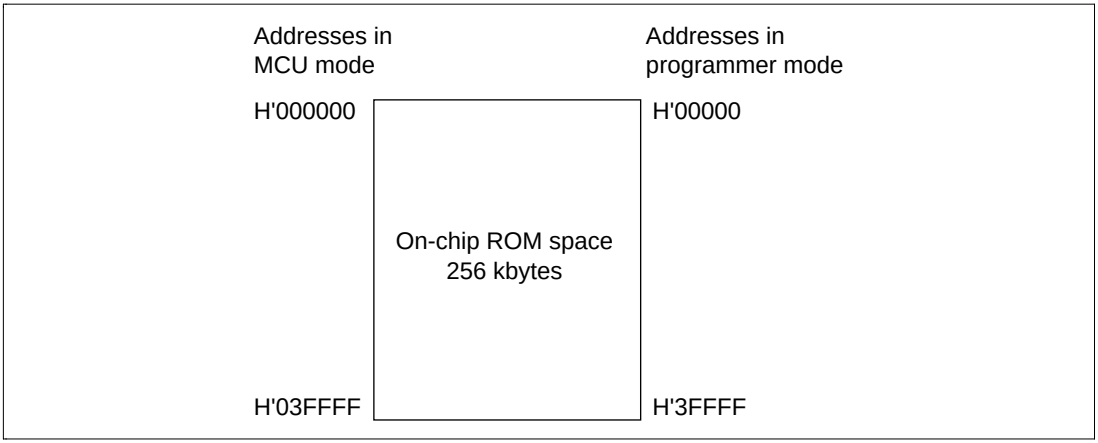


Figure 21B-17 On-Chip ROM Memory Map

21B.14.2 Programmer Mode Operation

Table 21B-15 shows how the different operating modes are set when using programmer mode, and table 21B-16 lists the commands used in programmer mode. Details of each mode are given below.

- Memory Read Mode

  Memory read mode supports byte reads.
- Auto-Program Mode

  Auto-program mode supports programming of 128 bytes at a time. Status polling is used to confirm the end of auto-programming.
- Auto-Erase Mode

  Auto-erase mode supports automatic erasing of the entire flash memory. Status polling is used to confirm the end of auto-programming.
- Status Read Mode

  Status polling is used for auto-programming and auto-erasing, and normal termination can be confirmed by reading the I/O6 signal. In status read mode, error information is output if an error occurs.

Table 21B-15 Settings for Various Operating Modes In Programmer Mode

	Pin Names					
Mode	**FWE**	**$\overline{CE}$**	**$\overline{OE}$**	**$\overline{WE}$**	**I/O7– I/O0**	**A18–A0**
Read	H or L	L	L	H	Data output	Ain
Output disable	H or L	L	H	H	Hi-z	X
Command write	H or L*3	L	H	L	Data input	Ain*2
Chip disable*1	H or L	H	X	X	Hi-z	X

Notes: *1 Chip disable is not a standby state; internally, it is an operation state.
*2 Ain indicates that there is also address input in auto-program mode.
*3 For command writes in auto-program and auto-erase modes, input a high level to the FWE pin.

Table 21B-16 Programmer Mode Commands

	Number	1st Cycle			2nd Cycle		
Command Name	**of Cycles**	**Mode**	**Address**	**Data**	**Mode**	**Address**	**Data**
Memory read mode	1 + n	Write	X	H'00	Read	RA	Dout
Auto-program mode	129	Write	X	H'40	Write	WA	Din
Auto-erase mode	2	Write	X	H'20	Write	X	H'20
Status read mode	2	Write	X	H'71	Write	X	H'71

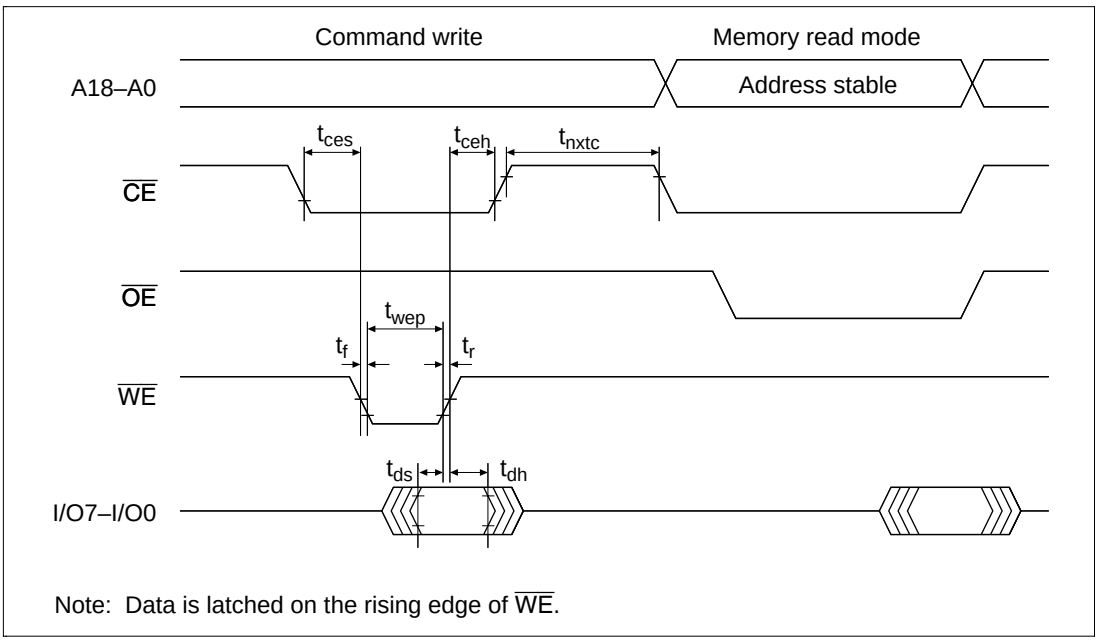
Notes: 1. In auto-program mode, 129 cycles are required for command writing by a simultaneous 128-byte write.
2. In memory read mode, the number of cycles depends on the number of address write cycles (n).

21B.14.3 Memory Read Mode

1. After completion of auto-program/auto-erase/status read operations, a transition is made to the command wait state. When reading memory contents, a transition to memory read mode must first be made with a command write, after which the memory contents are read.
2. In memory read mode, command writes can be performed in the same way as in the command wait state.
3. Once memory read mode has been entered, consecutive reads can be performed.
4. After powering on, memory read mode is entered.

Table 21B-17 AC Characteristics in Transition to Memory Read Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns



Note: Data is latched on the rising edge of $\overline{WE}$.

Figure 21B-18 Timing Waveforms for Memory Read after Memory Write

Table 21B-18 AC Characteristics in Transition from Memory Read Mode to Another Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

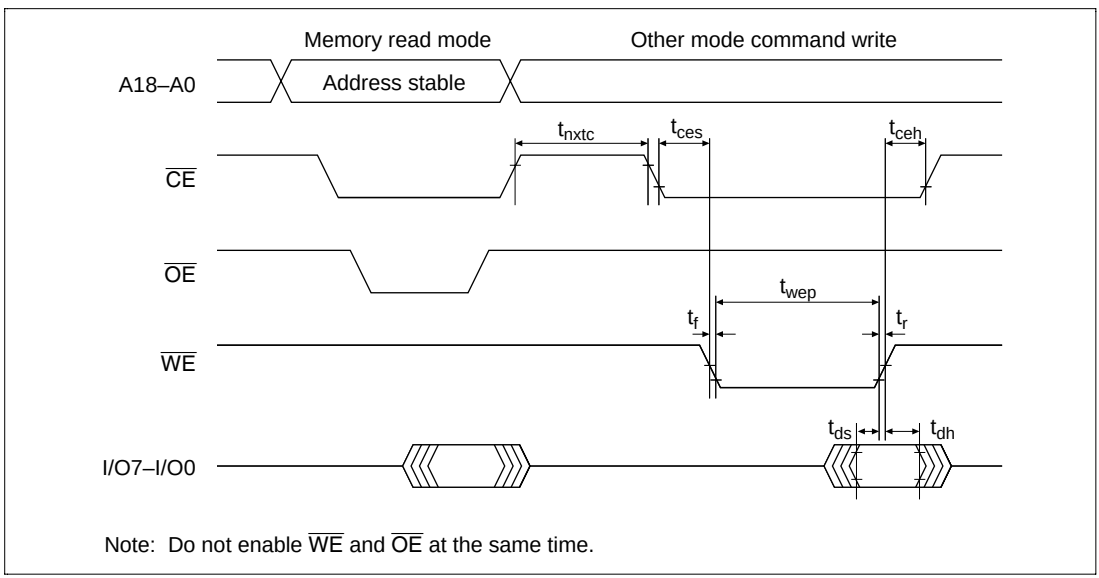


Figure 21B-19 Timing Waveforms in Transition from Memory Read Mode to Another Mode

Table 21B-19 AC Characteristics in Memory Read Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Access time	t_{acc}	—	20	µs
$\overline{CE}$ output delay time	t_{ce}	—	150	ns
$\overline{OE}$ output delay time	t_{oe}	—	150	ns
Output disable delay time	t_{df}	—	100	ns
Data output hold time	t_{oh}	5	—	ns

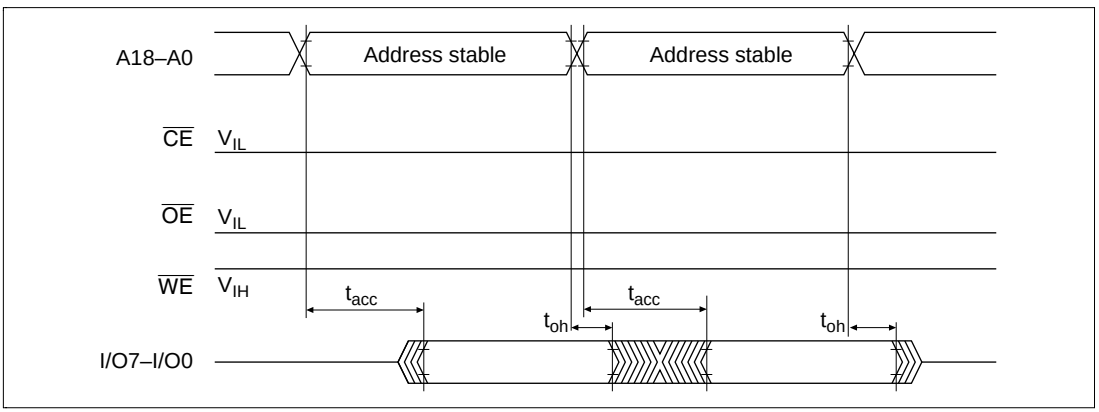


Figure 21B-20 $\overline{CE}$ and $\overline{OE}$ Enable State Read Timing Waveforms

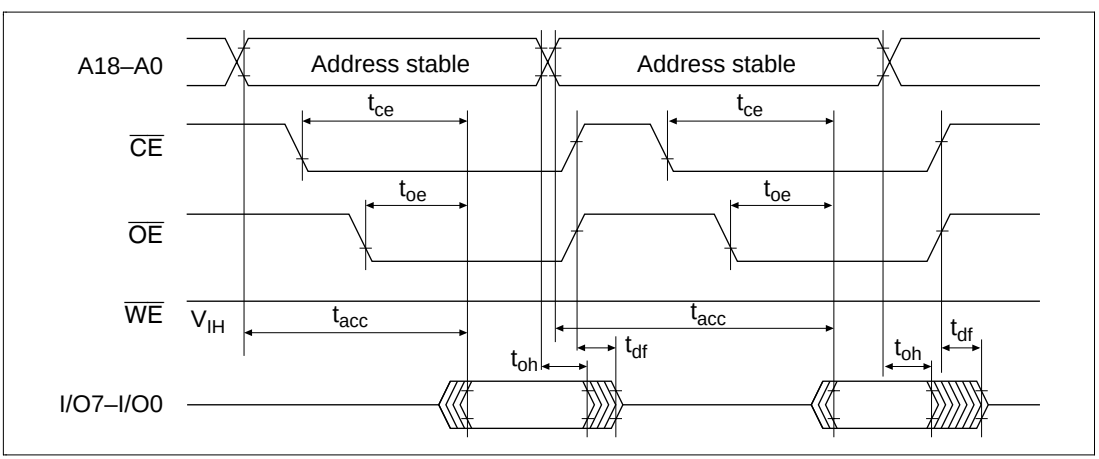


Figure 21B-21 $\overline{CE}$ and $\overline{OE}$ Clock System Read Timing Waveforms

21B.14.4 Auto-Program Mode

1. In auto-program mode, 128 bytes are programmed simultaneously. This should be carried out by executing 128 consecutive byte transfers.
2. A 128-byte data transfer is necessary even when programming fewer than 128 bytes. In this case, H'FF data must be written to the extra addresses.
3. The lower 7 bits of the transfer address must be low. If a value other than an effective address is input, processing will switch to a memory write operation but a write error will be flagged.
4. Memory address transfer is performed in the second cycle (figure 21B-22). Do not perform transfer after the third cycle.
5. Do not perform a command write during a programming operation.
6. Perform one auto-program operation for a 128-byte block for each address. Two or more additional programming operations cannot be performed on a previously programmed address block.
7. Confirm normal end of auto-programming by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-program operation end decision pin).
8. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 21B-20 AC Characteristics in Auto-Program Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
Status polling start time	t_{wsts}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Address setup time	t_{as}	0	—	ns
Address hold time	t_{ah}	60	—	ns
Memory write time	t_{write}	1	3000	ms
Write setup time	t_{pns}	100	—	ns
Write end setup time	t_{pnh}	100	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

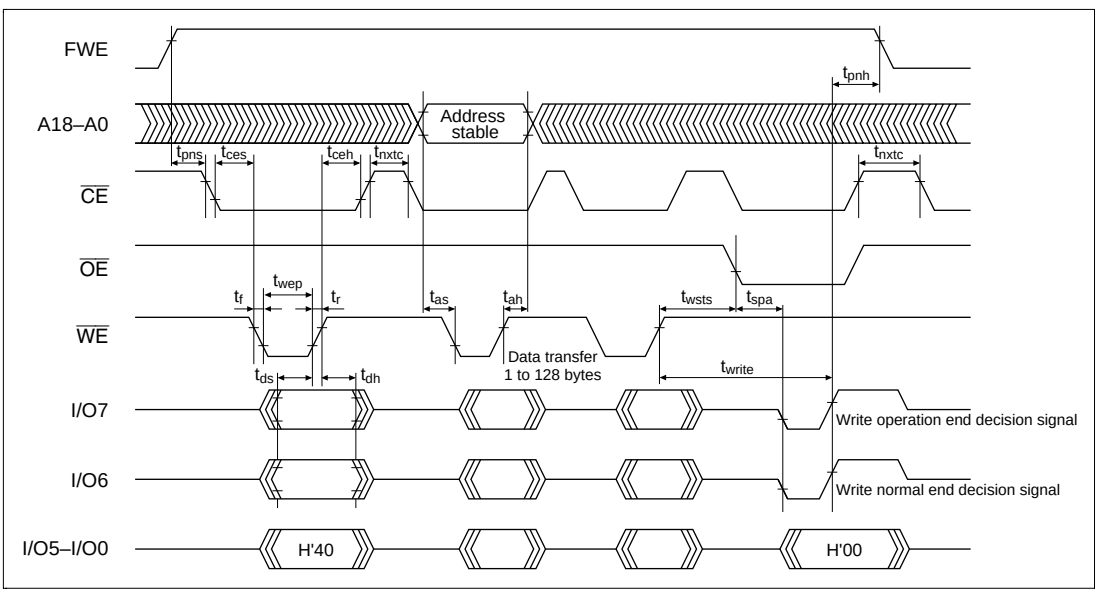


Figure 21B-22 Auto-Program Mode Timing Waveforms

21B.14.5 Auto-Erase Mode

1. Auto-erase mode supports only entire memory erasing.
2. Do not perform a command write during auto-erasing.
3. Confirm normal end of auto-erasing by checking I/O6. Alternatively, status read mode can also be used for this purpose (I/O7 status polling uses the auto-erase operation end decision pin).
4. Status polling I/O6 and I/O7 pin information is retained until the next command write. As long as the next command write has not been performed, reading is possible by enabling $\overline{CE}$ and $\overline{OE}$.

Table 21B-21 AC Characteristics in Auto-Erase Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Command write cycle	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
Status polling start time	t_{ests}	1	—	ms
Status polling access time	t_{spa}	—	150	ns
Memory erase time	t_{erase}	100	40000	ms
Erase setup time	t_{ens}	100	—	ns
Erase end setup time	t_{enh}	100	—	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns

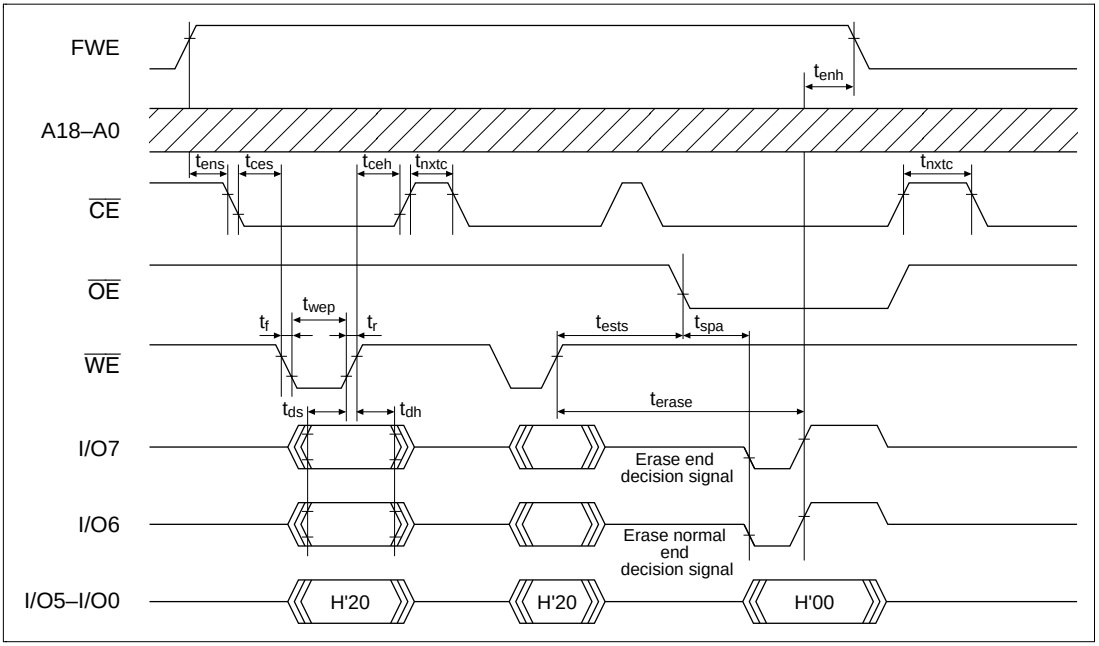


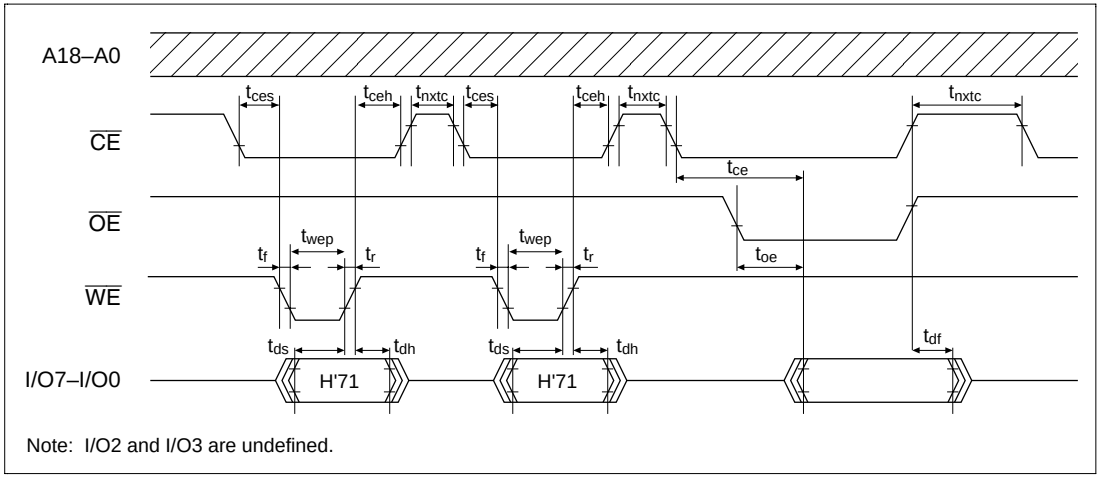
Figure 21B-23 Auto-Erase Mode Timing Waveforms

21B.14.6 Status Read Mode

1. Status read mode is provided to identify the kind of abnormal end. Use this mode when an abnormal end occurs in auto-program mode or auto-erase mode.
2. The return code is retained until a command write other than a status read mode command write is executed.

Table 21B-22 AC Characteristics in Status Read Mode (Conditions: V_{CC} = 5.0 V ±0.5 V, V_{SS} = 0 V, T_a = 25°C ±5°C)

Item	Symbol	Min	Max	Unit
Read time after command write	t_{nxtc}	20	—	µs
$\overline{CE}$ hold time	t_{ceh}	0	—	ns
$\overline{CE}$ setup time	t_{ces}	0	—	ns
Data hold time	t_{dh}	50	—	ns
Data setup time	t_{ds}	50	—	ns
Write pulse width	t_{wep}	70	—	ns
$\overline{OE}$ output delay time	t_{oe}	—	150	ns
Disable delay time	t_{df}	—	100	ns
$\overline{CE}$ output delay time	t_{ce}	—	150	ns
$\overline{WE}$ rise time	t_r	—	30	ns
$\overline{WE}$ fall time	t_f	—	30	ns



Note: I/O2 and I/O3 are undefined.

Figure 21B-24 Status Read Mode Timing Waveforms

Table 21B-23 Status Read Mode Return Commands

Pin Name	I/O7	I/O6	I/O5	I/O4	I/O3	I/O2	I/O1	I/O0
Attribute	Normal end decision	Command error	Program-ming error	Erase error	—	—	Program-ming or erase count exceeded	Effective address error
Initial value	0	0	0	0	0	0	0	0
Indications	Normal end: 0 Abnormal end: 1	Command error: 1 Otherwise: 0	Program-ming error: 1 Otherwise: 0	Erasing error: 1 Otherwise: 0	—	—	Count exceeded: 1 Otherwise: 0	Effective address error: 1 Otherwise: 0

Note: I/O2 and I/O3 are undefined.

21B.14.7 Status Polling

1. The I/O7 status polling flag indicates the operating status in auto-program/auto-erase mode.
2. The I/O6 status polling flag indicates a normal or abnormal end in auto-program/auto-erase mode.

Table 21B-24 Status Polling Output Truth Table

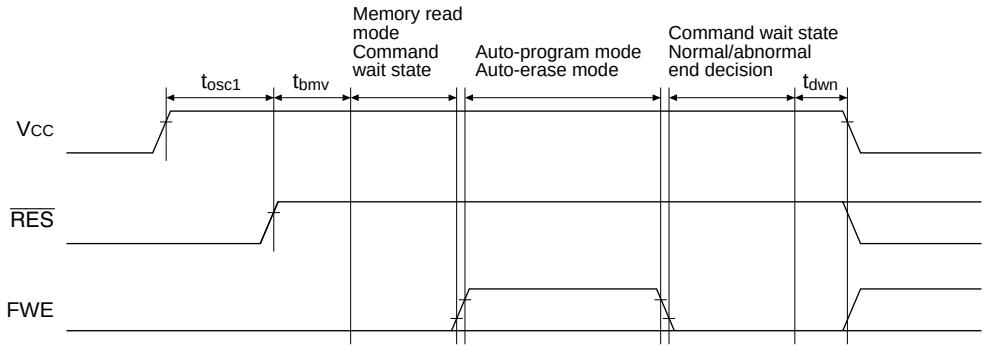
Pin Name	During Internal Operation	Abnormal End	—	Normal End
I/O7	0	1	0	1
I/O6	0	0	1	1
I/O0–I/O5	0	0	0	0

21B.14.8 Programmer Mode Transition Time

Commands cannot be accepted during the oscillation stabilization period or the programmer mode setup period. After the programmer mode setup time, a transition is made to memory read mode.

Table 21B-25 Stipulated Transition Times to Command Wait State

Item	Symbol	Min	Max	Unit
Standby release (oscillation stabilization time)	t_{osc1}	30	—	ms
Programmer mode setup time	t_{bmv}	10	—	ms
V_{CC} hold time	t_{dwn}	0	—	ms



Note: When using other than the automatic write mode and automatic erase mode, drive the FWE input pin low.

Figure 21B-25 Oscillation Stabilization Time, Boot Program Transfer Time, and Power-Down Sequence

21B.14.9 Notes on Memory Programming

1. When programming addresses which have previously been programmed, carry out auto-erasing before auto-programming.
2. When performing programming using programmer mode on a chip that has been programmed/erased in an on-board programming mode, auto-erasing is recommended before carrying out auto-programming.

Notes: 1. The flash memory is initially in the erased state when the device is shipped by Hitachi. For other chips for which the erasure history is unknown, it is recommended that auto-erasing be executed to check and supplement the initialization (erase) level.

2. Auto-programming should be performed once only on the same address block. Additional programming cannot be performed on previously programmed address blocks.

21B.15 Flash Memory and Power-Down States

In addition to its normal operating state, the flash memory has power-down states in which power consumption is reduced by halting part or all of the internal power supply circuitry.

There are three flash memory operating states:

(1) Normal operating mode: The flash memory can be read and written to.
(2) Power-down mode: Part of the power supply circuitry is halted, and the flash memory can be read when the LSI is operating on the subclock.
(3) Standby mode: All flash memory circuits are halted, and the flash memory cannot be read or written to.

States (2) and (3) are flash memory power-down states. Table 21B-26 shows the correspondence between the operating states of the LSI and the flash memory.

Table 21B-26 Flash Memory Operating States

LSI Operating State	Flash Memory Operating State
High-speed mode	Normal mode (read/write)
Medium-speed mode	
Sleep mode	
Subactive mode*	When PDWND = 0: Power-down mode (read-only)
Subsleep mode*	When PDWND = 1: Normal mode (read-only)
Watch mode*	Standby mode
Software standby mode	
Hardware standby mode	

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are not available in versions other than the U-mask and W-mask versions.

21B.15.1 Notes on Power-Down States

1. When the flash memory is in a power-down state, part or all of the internal power supply circuitry is halted. Therefore, a power supply circuit stabilization period must be provided when returning to normal operation. When the flash memory returns to its normal operating state from a power-down state, bits STS2 to STS0 in SBYCR must be set to provide a wait time of at least 20 µs (power supply stabilization time), even if an oscillation stabilization period is not necessary.
2. In a power-down state, FLMCR1, FLMCR2, EBR1, EBR2, RAMER, and FLPWCR cannot be read from or written to.

21B.16 Flash Memory Programming and Erasing Precautions

Precautions concerning the use of on-board programming mode, the RAM emulation function, and programmer mode are summarized below.

Use the specified voltages and timing for programming and erasing: Applied voltages in excess of the rating can permanently damage the device. Use a PROM programmer that supports the Hitachi microcomputer device type with 256-kbyte on-chip flash memory.

Only use the specified socket adapter. Failure to observe these points may result in damage to the device.

Powering on and off (see figures 21B-25 to 21B-27): Do not apply a high level to the FWE pin until V_{CC} has stabilized. Also, drive the FWE pin low before turning off V_{CC}.

When applying or disconnecting V_{CC} power, fix the FWE pin low and place the flash memory in the hardware protection state.

The power-on and power-off timing requirements should also be satisfied in the event of a power failure and subsequent recovery.

FWE application/disconnection (see figures 21B-25 to 21B-27): FWE application should be carried out when MCU operation is in a stable condition. If MCU operation is not stable, fix the FWE pin low and set the protection state.

The following points must be observed concerning FWE application and disconnection to prevent unintentional programming or erasing of flash memory:

- Apply FWE when the V_{CC} voltage has stabilized within its rated voltage range.
  Apply FWE when oscillation has stabilized (after the elapse of the oscillation stabilization time).
- In boot mode, apply and disconnect FWE during a reset.
- In user program mode, FWE can be switched between high and low level regardless of the reset state. FWE input can also be switched during execution of a program in flash memory.
- Do not apply FWE if program runaway has occurred.
- Disconnect FWE only when the SWE, ESU, PSU, EV, PV, P, and E bits in FLMCR1 are cleared.
  Make sure that the SWE, ESU, PSU, EV, PV, P, and E bits are not set by mistake when applying or disconnecting FWE.

Do not apply a constant high level to the FWE pin: Apply a high level to the FWE pin only when programming or erasing flash memory. A system configuration in which a high level is constantly applied to the FWE pin should be avoided. Also, while a high level is applied to the FWE pin, the watchdog timer should be activated to prevent overprogramming or overerasing due to program runaway, etc.

Use the recommended algorithm when programming and erasing flash memory: The recommended algorithm enables programming and erasing to be carried out without subjecting the device to voltage stress or sacrificing program data reliability. When setting the P or E bit in FLMCR1, the watchdog timer should be set beforehand as a precaution against program runaway, etc.

Do not set or clear the SWE bit during execution of a program in flash memory: Wait for at least 100 µs after clearing the SWE bit before executing a program or reading data in flash memory. When the SWE bit is set, data in flash memory can be rewritten, but when SWE = 1, flash memory can only be read in program-verify or erase-verify mode. Access flash memory only for verify operations (verification during programming/erasing). Also, do not clear the SWE bit during programming, erasing, or verifying.

Similarly, when using the RAM emulation function while a high level is being input to the FWE pin, the SWE bit must be cleared before executing a program or reading data in flash memory.

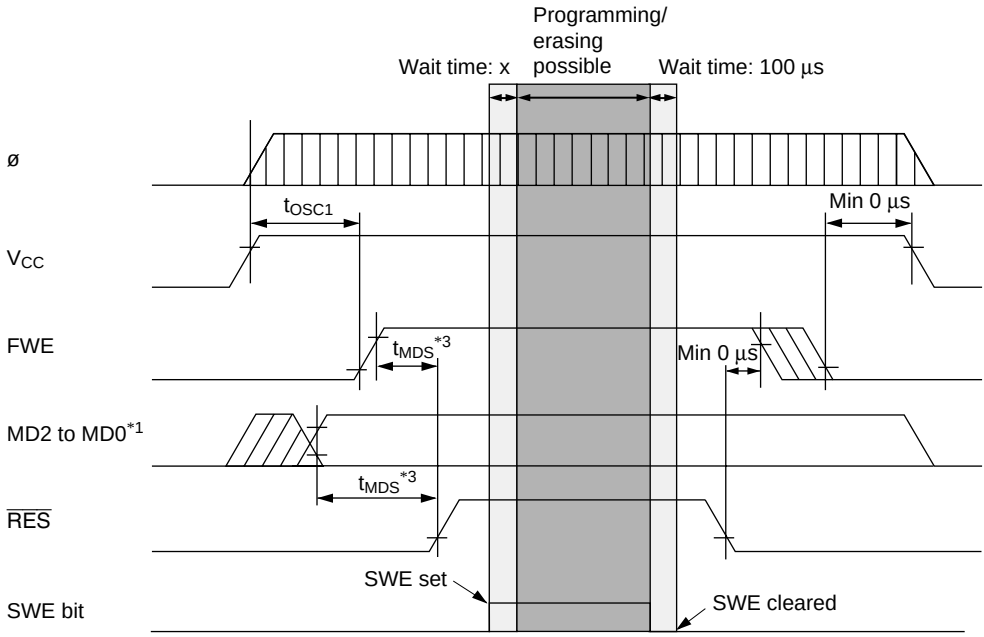
However, the RAM area overlapping flash memory space can be read and written to regardless of whether the SWE bit is set or cleared.

Do not use interrupts while flash memory is being programmed or erased: All interrupt requests, including NMI, should be disabled during FWE application to give priority to program/erase operations.

Do not perform additional programming. Erase the memory before reprogramming: In on-board programming, perform only one programming operation on a 128-byte programming unit block. In programmer mode, too, perform only one programming operation on a 128-byte programming unit block. Programming should be carried out with the entire programming unit block erased.

Before programming, check that the chip is correctly mounted in the PROM programmer: Overcurrent damage to the device can result if the index marks on the PROM programmer socket, socket adapter, and chip are not correctly aligned.

Do not touch the socket adapter or chip during programming: Touching either of these can cause contact faults and write errors.

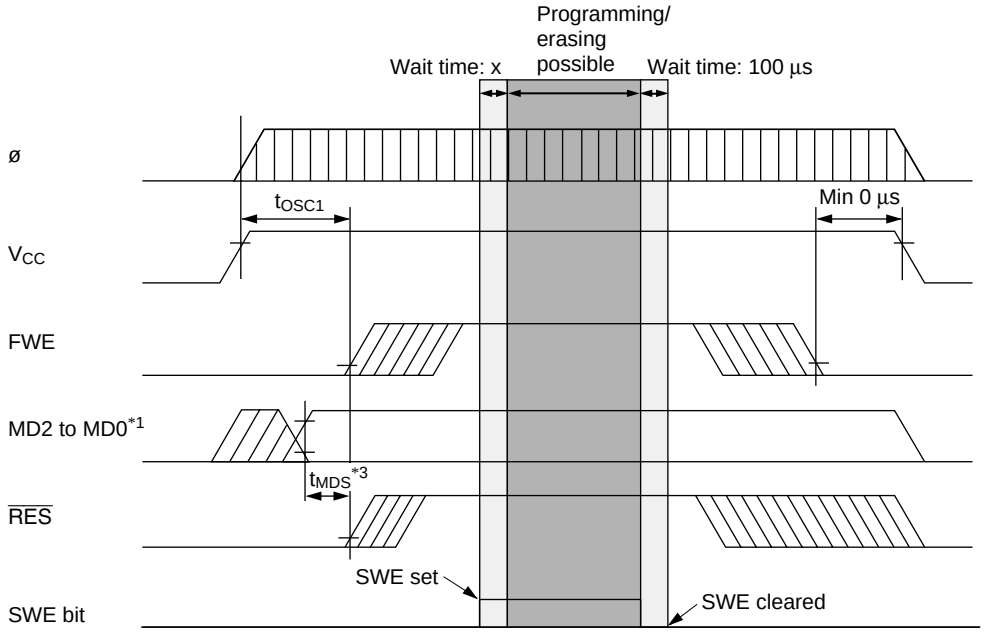


Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: *1 Except when switching modes, the level of the mode pins (MD2 to MD0) must be fixed until power-off by pulling the pins up or down.
*2 See section 24.2.7, Flash Memory Characteristics.
*3 Mode programming setup time t_{MDS} (min) = 200 ns

Figure 21B-26 Power-On/Off Timing (Boot Mode)

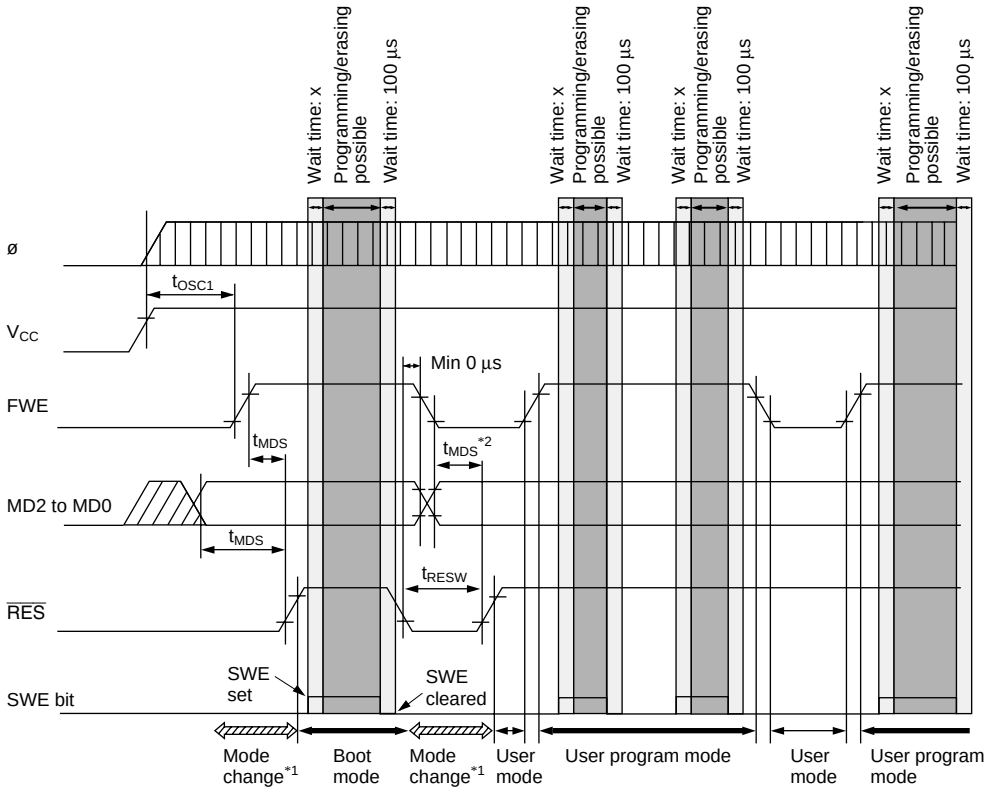


Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*2

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: *1 Except when switching modes, the level of the mode pins (MD2 to MD0) must be fixed until power-off by pulling the pins up or down.
*2 See section 24.2.7, Flash Memory Characteristics.
*3 Mode programming setup time t_{MDS} (min) = 200 ns

Figure 21B-27 Power-On/Off Timing (User Program Mode)



Period during which flash memory access is prohibited
(x: Wait time after setting SWE bit)*3

Period during which flash memory can be programmed
(Execution of program in flash memory prohibited, and data reads other than verify operations prohibited)

Notes: *1 When entering boot mode or making a transition from boot mode to another mode, mode switching must be carried out by means of $\overline{RES}$ input. The state of ports with multiplexed address functions and bus control output pins ($\overline{AS}$, $\overline{RD}$, $\overline{WR}$) will change during this switchover interval (the interval during which the $\overline{RES}$ pin input is low), and therefore these pins should not be used as output signals during this time.

*2 When making a transition from boot mode to another mode, a mode programming setup time t_{MDS} (min) of 200 ns is necessary with respect to $\overline{RES}$ clearance timing.

*3 See section 24.2.7, Flash Memory Characteristics.

Figure 21B-28 Mode Transition Timing
(Example: Boot Mode → User Mode ↔ User Program Mode)

21B.17 Note on Switching from F-ZTAT Version to Mask ROM Version

The mask ROM version does not have the internal registers for flash memory control that are provided in the F-ZTAT version. Table 21B-24 lists the registers that are present in the F-ZTAT version but not in the mask ROM version. If a register listed in table 21B-24 is read in the mask ROM version, an undefined value will be returned. Therefore, if application software developed on the F-ZTAT version is switched to a mask ROM version product, it must be modified to ensure that the registers in table 21B-27 have no effect.

Table 21B-27 Registers Present in F-ZTAT Version but Absent in Mask ROM Version

Register	Abbreviation	Address
Flash memory control register 1	FLMCR1	H'FFC8
Flash memory control register 2	FLMCR2	H'FFC9
Erase block register 1	EBR1	H'FFCA
Erase block register 2	EBR2	H'FFCB
RAM emulation register	RAMER	H'FEDB

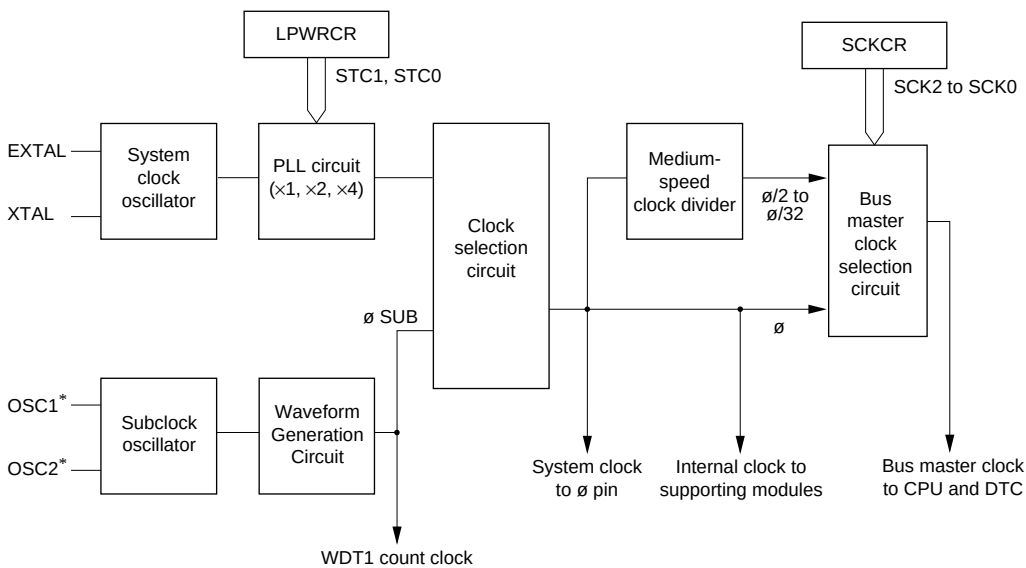
22A.1 Overview

The chip has a built-in clock pulse generator (CPG) that generates the system clock (ø), the bus master clock, and internal clocks.

The clock pulse generator consists of an oscillator, PLL (phase-locked loop) circuit, clock selection circuit, medium-speed clock divider, bus master clock selection circuit, subclock oscillator, and waveform shaping circuit. The frequency can be changed by means of the PLL circuit in the CPG. Frequency changes are performed by software by means of settings in the system clock control register (SCKCR) and low-power control register (LPWRCR).

22A.1.1 Block Diagram

Figure 22A-1 shows a block diagram of the clock pulse generator.



Legend:

LPWRCR: Low-power control register

SCKCR: System clock control register

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.

These functions cannot be used with the other versions.

See section 22A.7 Subclock Oscillator, for the method of fixing pins OSC1 and OSC2.

Figure 22A-1 Block Diagram of Clock Pulse Generator

22A.1.2 Register Configuration

The clock pulse generator is controlled by SCKCR and LPWRCR. Table 22A-1 shows the register configuration.

Table 22A-1 Clock Pulse Generator Register

Name	Abbreviation	R/W	Initial Value	Address*
System clock control register	SCKCR	R/W	H'00	H'FDE6
Low-power control register	LPWRCR	R/W	H'00	H'FDEC

Note:* Lower 16 bits of the address.

22A.2 Register Descriptions

22A.2.1 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	—	—	—	R/W	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ø clock output control and medium-speed mode control, selection of operation when the PLL circuit frequency multiplication factor is changed, and medium-speed mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—ø Clock Output Disable (PSTOP): In combination with the DDR of the applicable port, this bit controls ø output. See section 23A.8, 23B.12 ø Clock Output Disable Function for details.

	Description			
Bit 7 PSTOP	**Normal Operating State**	**Sleep Mode**	**Software Standby Mode**	**Hardware Standby Mode**
0	ø output (initial value)	ø output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bits 6 to 4—Reserved: These bits are always read as 0 and cannot be modified.

Bit 3—Frequency Multiplication Factor Switching Mode Select (STCS): Selects the operation when the PLL circuit frequency multiplication factor is changed.

Bit 3	
STCS	**Description**
0	Specified multiplication factor is valid after transition to software standby mode, watch mode*, and subactive mode* (Initial value)
1	Specified multiplication factor is valid immediately after STC bits are rewritten

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Bits 2 to 0—System Clock Select 2 to 0 (SCK2 to SCK0): These bits select the bus master clock.

Bit 2	Bit 1	Bit 0		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master is in high-speed mode	(Initial value)
		1	Medium-speed clock is ø/2	
	1	0	Medium-speed clock is ø/4	
		1	Medium-speed clock is ø/8	
1	0	0	Medium-speed clock is ø/16	
		1	Medium-speed clock is ø/32	
	1	—	—	

22A.2.2 Low-Power Control Register (LPWRCR)

Bit	7	6	5	4	3	2	1	0
	DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	STC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LPWRCR is an 8-bit readable/writable register that performs power-down mode control. The following pertains to bits 1 and 0. For details of the other bits, see Section 23A.2.3, 23B.2.3, Low Power Control Register (LPWRCR). LPWRCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 1 and 0—Frequency Multiplication Factor (STC1, STC0): The STC bits specify the frequency multiplication factor of the PLL circuit.

Bit 1	Bit 0		
STC1	**STC0**	**Description**	
0	0	×1	(Initial value)
	1	×2	
1	0	×4	
	1	Setting prohibited	

Note: A system clock frequency multiplied by the multiplication factor (STC1 and STC0) should not exceed the maximum operating frequency defined in section 24 Electrical Characteristics.

22A.3 Oscillator

A crystal oscillator is used to supply clock pulses.

In either case, the input clock should not exceed 20 MHz.

22A.3.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 22A-2. Select the damping resistance R_d according to table 22A-2. An AT-cut parallel-resonance crystal should be used.

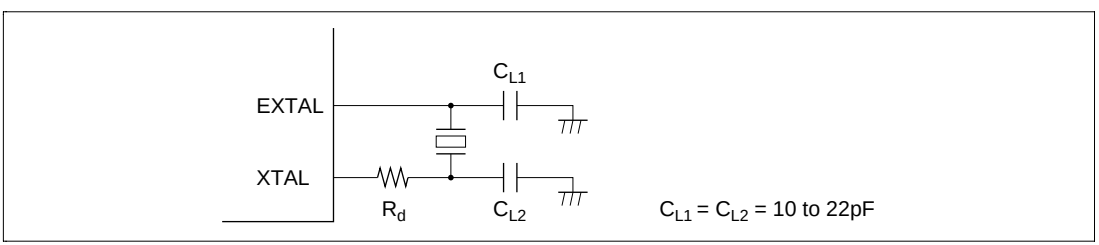


Figure 22A-2 Connection of Crystal Resonator (Example)

Table 22A-2 Damping Resistance Value

Frequency (MHz)	4	8	12	16	20
R_d (Ω)	500	200	0	0	0

Crystal Resonator: Figure 22A-3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 22A-3. The crystal resonator frequency should not exceed 20 MHz.

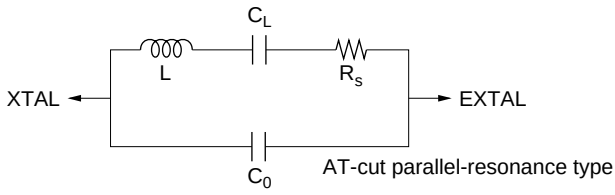


Figure 22A-3 Crystal Resonator Equivalent Circuit

Table 22A-3 Crystal Resonator Parameters

Frequency (MHz)	4	8	12	16	20
R_S max (Ω)	120	80	60	50	40
C_0 max (pF)	7	7	7	7	7

Note on Board Design: When a crystal resonator is connected, the following points should be noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 22A-4.

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins.

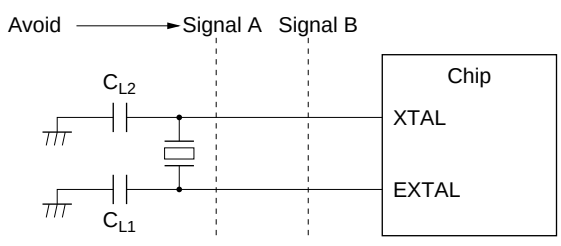
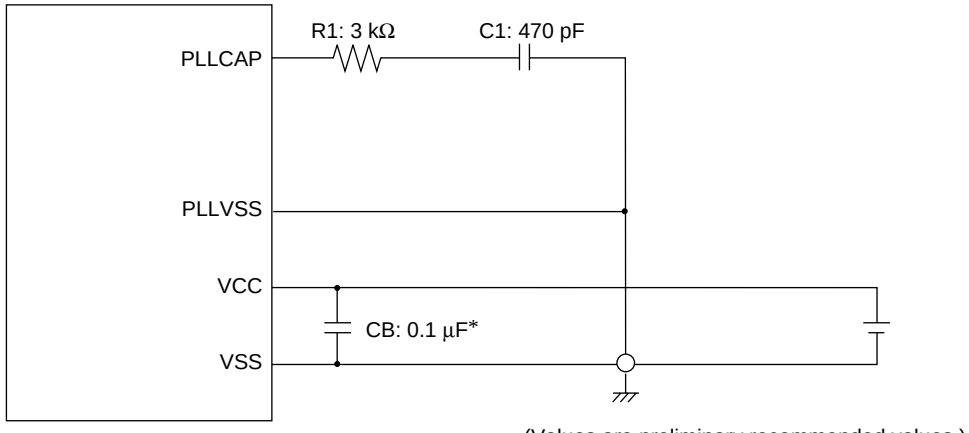


Figure 22A-4 Example of Incorrect Board Design

External circuitry such as that shown below is recommended around the PLL.



Note: * CB is laminated ceramic capacitors.

Figure 22A-5 Points for Attention when Using PLL Oscillation Circuit

Place oscillation stabilization capacitor C1 and resistor R1 close to the PLLCAP pin, and ensure that no other signal lines cross this line. Supply the C1 ground from PLLVSS.

Separate PLLVSS from the other VCC and VSS lines at the board power supply source, and be sure to insert bypass capacitors CB close to the pins.

22A.3.2 External Clock Input

Circuit Configuration

An external clock signal can be input as shown in the examples in figure 22A-6. If the XTAL pin is left open, make sure that stray capacitance is no more than 10 pF.

In example (b), make sure that the external clock is held high in standby mode.

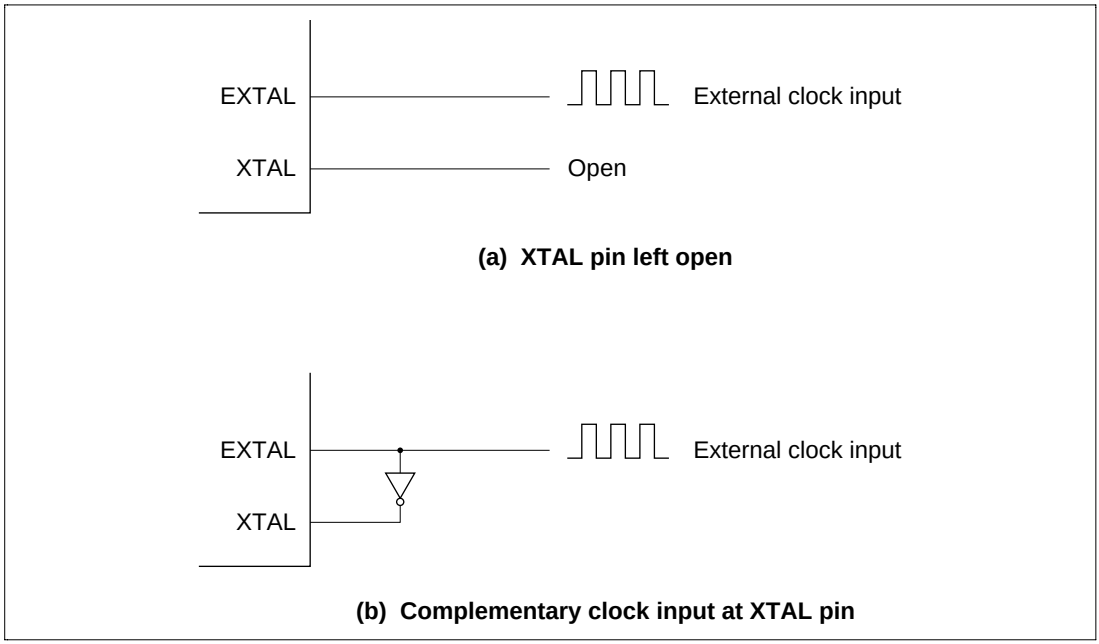


Figure 22A-6 External Clock Input (Examples)

External Clock

Table 22A-4 and figure 22A-7 show the input conditions for the external clock.

Table 22A-4 External Clock Input Conditions

		V_{cc} = 5.0 V ±10%				
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**	
External clock input low pulse width	t_{EXL}	15	—	ns	Figure 22A-7	
External clock input high pulse width	t_{EXH}	15	—	ns		
External clock rise time	t_{EXr}	—	5	ns		
External clock fall time	t_{EXf}	—	5	ns		
Clock low pulse	t_{CL}	0.4	0.6	t_{cyc}	ø ≥ 5 MHz	Figure 24-5
width level		80	—	ns	ø < 5 MHz	
Clock high pulse width	t_{CH}	0.4	0.6	t_{cyc}	ø ≥ 5 MHz	
level		80	—	ns	ø < 5 MHz	

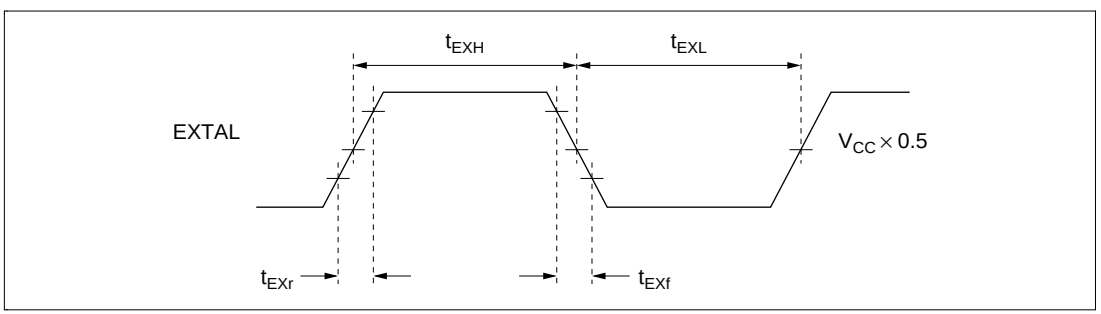


Figure 22A-7 External Clock Input Timing

22A.4 PLL Circuit

The PLL circuit has the function of multiplying the frequency of the clock from the oscillator by a factor of 1, 2, or 4. The multiplication factor is set with the STC bits in SCKCR. The phase of the rising edge of the internal clock is controlled so as to match that at the EXTAL pin.

When the multiplication factor of the PLL circuit is changed, the operation varies according to the setting of the STCS bit in SCKCR.

When STCS = 0 (initial value), the setting becomes valid after a transition to software standby mode, watch mode*, or subactive mode*. The transition time count is performed in accordance with the setting of bits STS2 to STS0 in SBYCR.

[1] The initial PLL circuit multiplication factor is 1.

[2] A value is set in bits STS2 to STS0 to give the specified transition time.

[3] The target value is set in STC1 and STC0, and a transition is made to software standby mode, watch mode*, or subactive mode*.

[4] The clock pulse generator stops and the value set in STC1 and STC0 becomes valid.

[5] Software standby mode, watch mode*, or subactive mode* is cleared, and a transition time is secured in accordance with the setting in STS2 to STS0.

[6] After the set transition time has elapsed, the LSI resumes operation using the target multiplication factor.

If a PC break is set for the SLEEP instruction that causes a transition to software standby mode in [1], software standby mode is entered and break exception handling is executed after the oscillation stabilization time. In this case, the instruction following the SLEEP instruction is executed after execution of the RTE instruction.

When STCS = 1, the LSI operates on the changed multiplication factor immediately after bits STC1 and STC0 are rewritten.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

22A.5 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate ø/2, ø/4, ø/8, ø/16, and ø/32.

22A.6 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the system clock (ø) or one of the medium-speed clocks (ø/2, ø/4, or ø/8, ø/16, and ø/32) to be supplied to the bus master, according to the settings of the SCK2 to SCK0 bits in SCKCR.

22A.7 Subclock Oscillator

Connecting 32.768kHz Quartz Oscillator (U Mask, W Mask): To supply a clock to the subclock divider, connect a 32.768kHz quartz oscillator, as shown in Figure 22A-8. See Section 22A.3.1, “Notes on Board Design” for notes on connecting quartz oscillators.

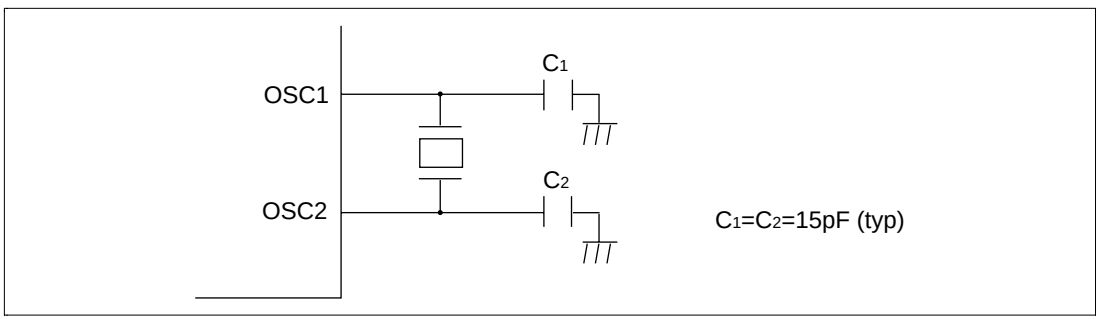


Figure 22A-8 Example Connection of 32.768kHz Quartz Oscillator

Figure 22A-9 shows the equivalence circuit for a 32.768kHz oscillator.

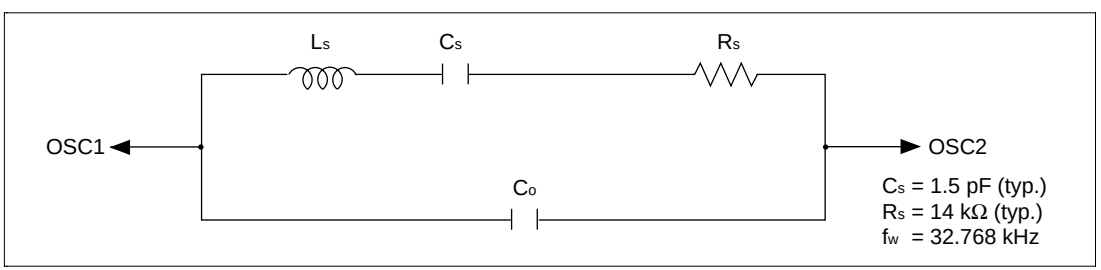


Figure 22A-9 Equivalence Circuit for 32.768kHz Oscillator

Handling pins when subclock not required: If no subclock is required, connect the OSC1 pin to VSS and leave OSC2 open, as shown in Figure 22A-10.

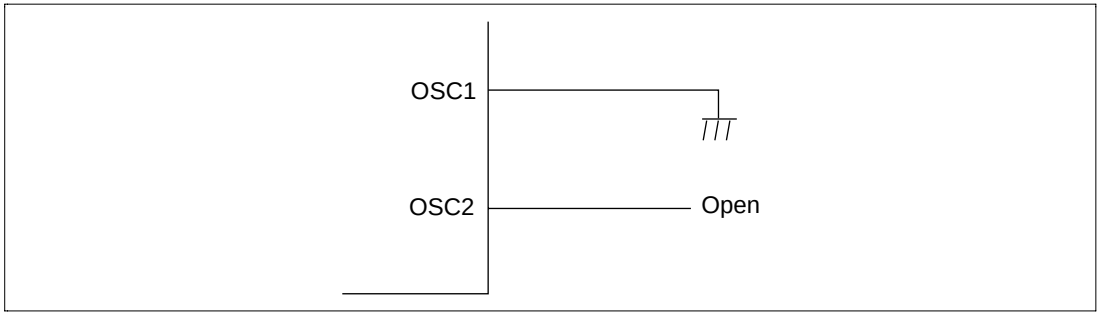


Figure 22A-10 Pin Handling When Subclock Not Required

22A.8 Subclock Waveform Generation Circuit

To eliminate noise from the subclock input to OSCI, the subclock is sampled using the dividing clock ø. The sampling frequency is set using the NESEL bit of LPWRCR. For details, see Section 23A.2.3, 23B.2.3, Low Power Control Register (LPWRCR).

No sampling is performed in sub-active mode*, sub-sleep mode*, or watch mode*.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

22A.9 Note on Crystal Resonator

Since various characteristics related to the crystal resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, for both the mask versions and F-ZTAT versions, using the resonator connection examples shown in this section as a guide. As the resonator circuit ratings will depend on the floating capacitance of the resonator and the mounting circuit, the ratings should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

22B.1 Overview

The chip has a built-in clock pulse generator (CPG) that generates the system clock (ø), the bus master clock, and internal clocks.

The clock pulse generator consists of an oscillator, PLL (phase-locked loop) circuit, system clock selection circuit, medium-speed clock divider, bus master clock selection circuit, and subclock divider. The frequency can be changed by means of the PLL circuit in the CPG. Frequency changes are performed by software by means of settings in the system clock control register (SCKCR) and low-power control register (LPWRCR).

22B.1.1 Block Diagram

Figure 22B-1 shows a block diagram of the clock pulse generator.

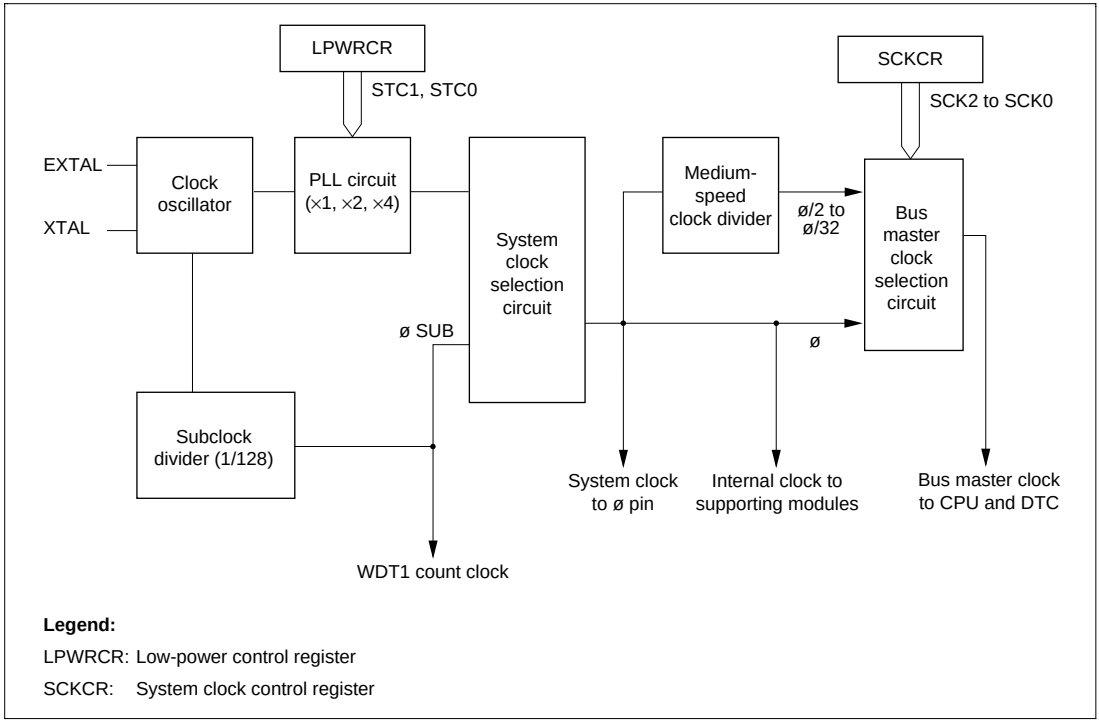


Figure 22B-1 Block Diagram of Clock Pulse Generator

22B.1.2 Register Configuration

The clock pulse generator is controlled by SCKCR and LPWRCR. Table 22B-1 shows the register configuration.

Table 22B-1 Clock Pulse Generator Register

Name	Abbreviation	R/W	Initial Value	Address*
System clock control register	SCKCR	R/W	H'00	H'FDE6
Low-power control register	LPWRCR	R/W	H'00	H'FDEC

Note:* Lower 16 bits of the address.

22B.2 Register Descriptions

22B.2.1 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0
Initial value:		0	0	0	0	0	0	0	0
R/W	:	R/W	—	—	—	R/W	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ø clock output control and medium-speed mode control, selection of operation when the PLL circuit frequency multiplication factor is changed, and medium-speed mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—ø Clock Output Disable (PSTOP): In combination with the DDR of the applicable port, this bit controls ø output. See section 23B.12 ø Clock Output Disable Function for details.

	Description			
Bit 7 PSTOP	**Normal Operating State**	**Sleep Mode**	**Software Standby Mode**	**Hardware Standby Mode**
0	ø output (initial value)	ø output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bits 6 to 4—Reserved: These bits are always read as 0 and cannot be modified.

Bit 3—Frequency Multiplication Factor Switching Mode Select (STCS): Selects the operation when the PLL circuit frequency multiplication factor is changed.

Bit 3	
STCS	**Description**
0	Specified multiplication factor is valid after transition to software standby mode, watch mode, and subactive mode (Initial value)
1	Specified multiplication factor is valid immediately after STC bits are rewritten

Bits 2 to 0—System Clock Select 2 to 0 (SCK2 to SCK0): These bits select the bus master clock.

Bit 2	Bit 1	Bit 0	
SCK2	**SCK1**	**SCK0**	**Description**
0	0	0	Bus master is in high-speed mode (Initial value)
		1	Medium-speed clock is ø/2
	1	0	Medium-speed clock is ø/4
		1	Medium-speed clock is ø/8
1	0	0	Medium-speed clock is ø/16
		1	Medium-speed clock is ø/32
	1	—	—

22B.2.2 Low-Power Control Register (LPWRCR)

Bit	:	7	6	5	4	3	2	1	0
		DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	STC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LPWRCR is an 8-bit readable/writable register that performs power-down mode control. The following pertains to bits 1 and 0. For details of the other bits, see Section 23B.2.3, Low Power Control Register (LPWRCR). LPWRCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bits 1 and 0—Frequency Multiplication Factor (STC1, STC0): The STC bits specify the frequency multiplication factor of the PLL circuit.

Bit 1	Bit 0		
STC1	**STC0**	**Description**	
0	0	×1	(Initial value)
	1	×2	
1	0	×4	
	1	Setting prohibited	

Note: A system clock frequency multiplied by the multiplication factor (STC1 and STC0) should not exceed the maximum operating frequency defined in section 24 Electrical Characteristics.

22B.3 Oscillator

A crystal oscillator is used to supply clock pulses.

In either case, the input clock should not exceed 20 MHz.

22B.3.1 Connecting a Crystal Resonator

Circuit Configuration: A crystal resonator can be connected as shown in the example in figure 22B-2. Select the damping resistance R_d according to table 22B-2. An AT-cut parallel-resonance crystal should be used.

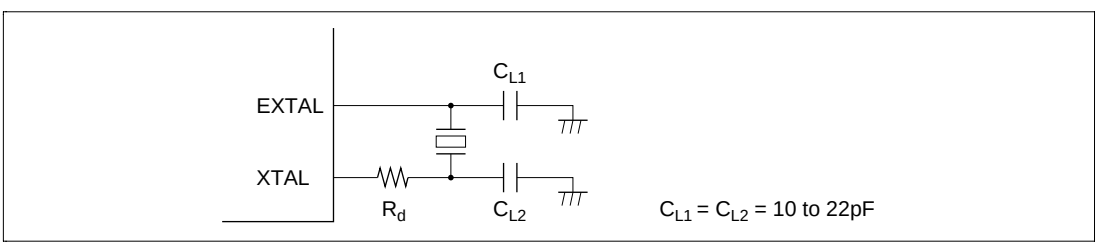


Figure 22B-2 Connection of Crystal Resonator (Example)

Table 22B-2 Damping Resistance Value

Frequency (MHz)	**4**	**5**
R_d (Ω)	500	200

Crystal Resonator: Figure 22B-3 shows the equivalent circuit of the crystal resonator. Use a crystal resonator that has the characteristics shown in table 18-3. The crystal resonator frequency should not exceed 20 MHz.

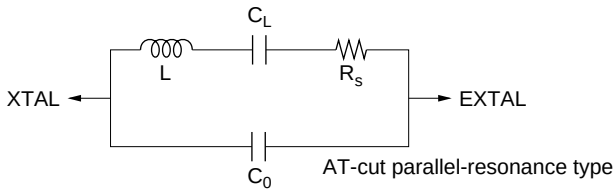


Figure 22B-3 Crystal Resonator Equivalent Circuit

Table 22B-3 Crystal Resonator Parameters

Frequency (MHz)	4	5
R_S max (Ω)	120	80
C_0 max (pF)	7	7

Note on Board Design: When a crystal resonator is connected, the following points should be noted:

Other signal lines should be routed away from the oscillator circuit to prevent induction from interfering with correct oscillation. See figure 22B-4.

When designing the board, place the crystal resonator and its load capacitors as close as possible to the XTAL and EXTAL pins.

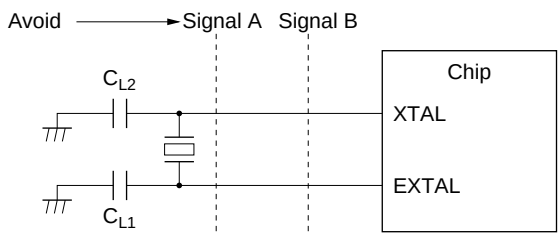
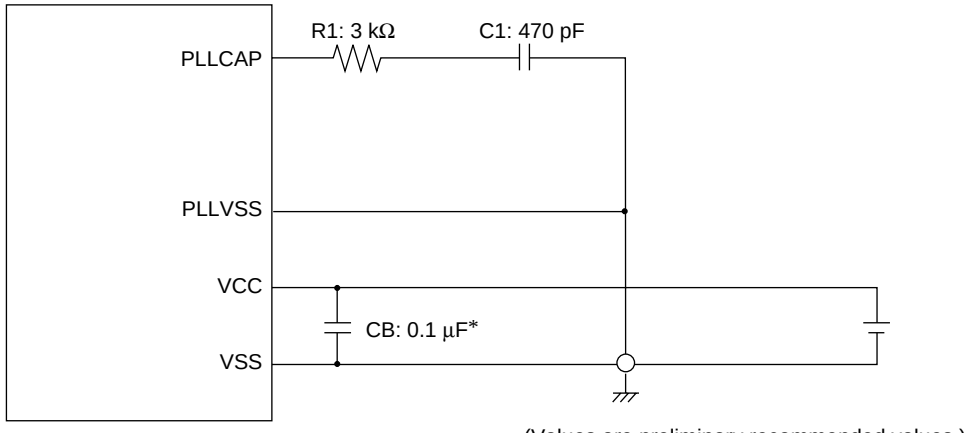


Figure 22B-4 Example of Incorrect Board Design

External circuitry such as that shown below is recommended around the PLL.



Note: * CB is laminated ceramic capacitors.

Figure 22B-5 Points for Attention when Using PLL Oscillation Circuit

Place oscillation stabilization capacitor C1 and resistor R1 close to the PLLCAP pin, and ensure that no other signal lines cross this line. Supply the C1 ground from PLLVSS.

Separate PLLVSS from the other VCC and VSS lines at the board power supply source, and be sure to insert bypass capacitors CB close to the pins.

22B.3.2 External Clock Input

Circuit Configuration

An external clock signal can be input as shown in the examples in figure 22B-6. If the XTAL pin is left open, make sure that stray capacitance is no more than 10 pF.

In example (b), make sure that the external clock is held high in standby mode.

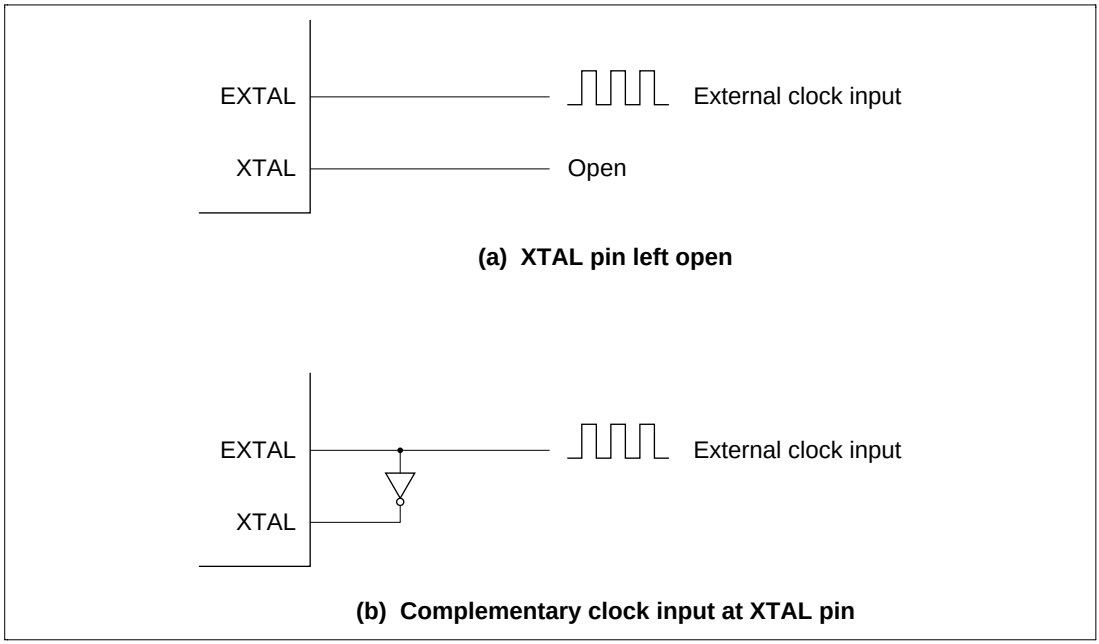


Figure 22B-6 External Clock Input (Examples)

External Clock

Table 22B-4 and figure 22B-7 show the input conditions for the external clock.

Table 22B-4 External Clock Input Conditions

		V_{cc} = 5.0 V ±10%				
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**	
External clock input low pulse width	t_{EXL}	15	—	ns	Figure 22B-7	
External clock input high pulse width	t_{EXH}	15	—	ns		
External clock rise time	t_{EXr}	—	5	ns		
External clock fall time	t_{EXf}	—	5	ns		
Clock low pulse	t_{CL}	0.4	0.6	t_{cyc}	ø ≥ 5 MHz	Figure 24-5
width level		80	—	ns	ø < 5 MHz	
Clock high pulse width	t_{CH}	0.4	0.6	t_{cyc}	ø ≥ 5 MHz	
level		80	—	ns	ø < 5 MHz	

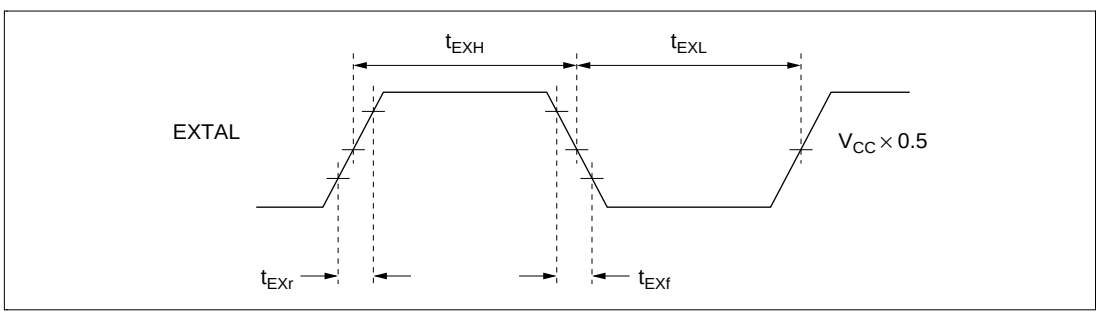


Figure 22B-7 External Clock Input Timing

22B.4 PLL Circuit

The PLL circuit has the function of multiplying the frequency of the clock from the oscillator by a factor of 1, 2, or 4. The multiplication factor is set with the STC bits in SCKCR. The phase of the rising edge of the internal clock is controlled so as to match that at the EXTAL pin.

When the multiplication factor of the PLL circuit is changed, the operation varies according to the setting of the STCS bit in SCKCR.

When STCS = 0 (initial value), the setting becomes valid after a transition to software standby mode, watch mode, or subactive mode. The transition time count is performed in accordance with the setting of bits STS2 to STS0 in SBYCR.

[1] The initial PLL circuit multiplication factor is 1.

[2] A value is set in bits STS2 to STS0 to give the specified transition time.

[3] The target value is set in STC1 and STC0, and a transition is made to software standby mode, watch mode, or subactive mode.

[4] The clock pulse generator stops and the value set in STC1 and STC0 becomes valid.

[5] Software standby mode, watch mode, or subactive mode is cleared, and a transition time is secured in accordance with the setting in STS2 to STS0.

[6] After the set transition time has elapsed, the LSI resumes operation using the target multiplication factor.

If a PC break is set for the SLEEP instruction that causes a transition to software standby mode in [1], software standby mode is entered and break exception handling is executed after the oscillation stabilization time. In this case, the instruction following the SLEEP instruction is executed after execution of the RTE instruction.

When STCS = 1, the LSI operates on the changed multiplication factor immediately after bits STC1 and STC0 are rewritten.

22B.5 Medium-Speed Clock Divider

The medium-speed clock divider divides the system clock to generate ø/2, ø/4, ø/8, ø/16, and ø/32.

22B.6 Bus Master Clock Selection Circuit

The bus master clock selection circuit selects the system clock (ø) or one of the medium-speed clocks (ø/2, ø/4, or ø/8, ø/16, and ø/32) to be supplied to the bus master, according to the settings of the SCK2 to SCK0 bits in SCKCR.

22B.7 Subclock Divider

The subclock divider divides the input clock into 1/128 to generate ø SUB.

22B.8 Note on Crystal Resonator

Since various characteristics related to the crystal resonator are closely linked to the user's board design, thorough evaluation is necessary on the user's part, for both the mask versions and F-ZTAT versions, using the resonator connection examples shown in this section as a guide. As the resonator circuit ratings will depend on the floating capacitance of the resonator and the mounting circuit, the ratings should be determined in consultation with the resonator manufacturer. The design must ensure that a voltage exceeding the maximum rating is not applied to the oscillator pin.

Subclock functions are not available in the HD64F2636F, HD64F2638F, HD6432636F and HD6432638F.

23A.1 Overview

In addition to the normal program execution state, the chip has nine power-down modes in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip supporting modules, and so on.

The chip operating modes are as follows:

(1) High-speed mode
(2) Medium-speed mode
(3) Sleep mode
(4) Module stop mode
(5) Software standby mode
(6) Hardware standby mode

(2) to (6) are low power dissipation states. Sleep mode is CPU states, medium-speed mode is a CPU and bus master state, and module stop mode is an internal peripheral function (including bus masters other than the CPU) state. Some of these states can be combined.

After a reset, the LSI is in high-speed mode with modules other than the DTC in module stop mode.

Notes: 1. Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.
2. See section 22A.7, Subclock Oscillator, for the method of fixing pins OSC1 and OSC2 when not used.

Table 23A-1 shows the internal state of the LSI in the respective modes. Table 23A-2 shows the conditions for shifting between the low power dissipation modes.

Figure 23A-1 is a mode transition diagram.

Table 23A-1 LSI Internal States in Each Mode

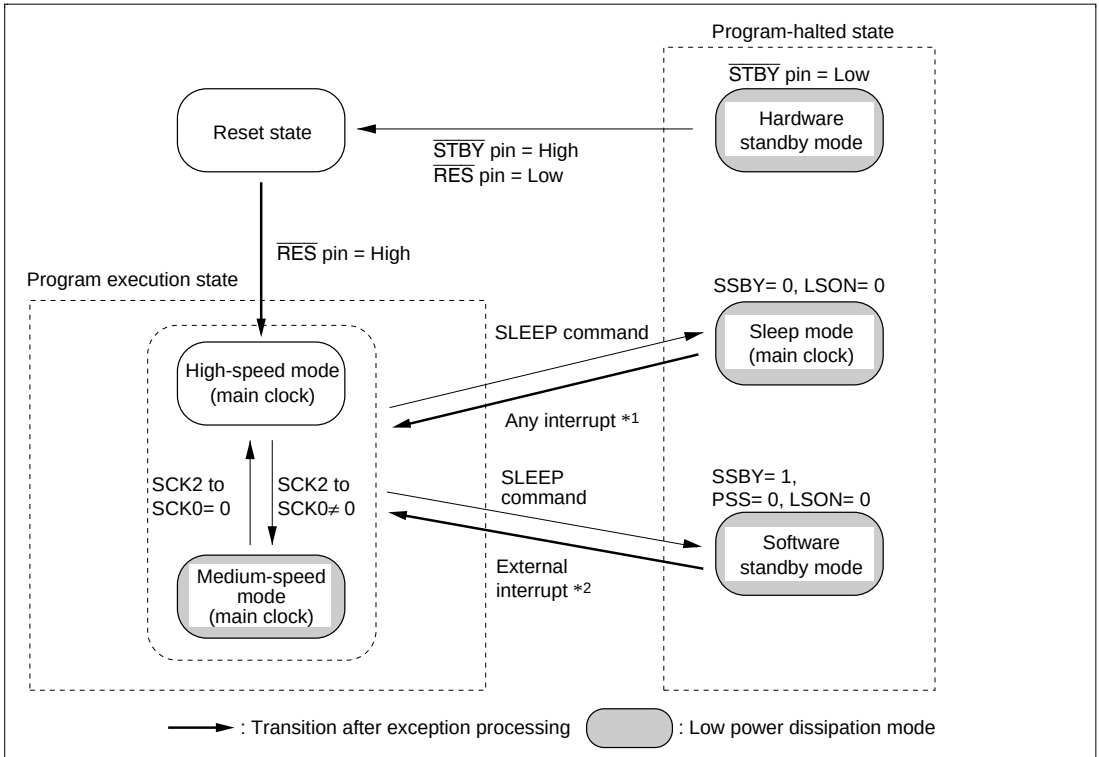
Function		High-Speed	Medium-Speed	Sleep	Module Stop	Software Standby	Hardware Standby
System clock pulse generator		Functioning	Functioning	Functioning	Functioning	Halted	Halted
CPU	Instructions	Functioning	Medium-speed operation	Halted (retained)	High/medium-speed operation	Halted (retained)	Halted (undefined)
	Registers						
External interrupts	NMI	Functioning	Functioning	Functioning	Functioning	Functioning	Halted
	IRQ0–IRQ5						
Peripheral functions	WDT1	Functioning	Functioning	Functioning	—	Halted (retained)	Halted (reset)
	WDT0	Functioning	Functioning	Functioning	—	Halted (retained)	Halted (reset)
	DTC	Functioning	Medium-speed operation	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	PBC	Functioning	Medium-speed operation	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	TPU	Functioning	Functioning	Functioning	Halted (retained)	Halted (retained)	Halted (reset)
	IIC0*						
	IIC1*						
	PPG						
	D/A0, 1						
	SCI0	Functioning	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)
	SCI1						
	SCI2						
	PWM						
	A/D						
	RAM	Functioning	Functioning	Functioning (DTC)	Functioning	Retained	Retained
	I/O	Functioning	Functioning	Functioning	Functioning	Retained	High impedance
	HCAN	Functioning	Functioning	Functioning	Halted (reset)	Halted (reset)	Halted (reset)

Notes: “Halted (retained)” means that internal register values are retained. The internal state is “operation suspended.”

“Halted (reset)” means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

* The I^2C bus interface is available as an option in the H8S/2638 and H8S/2639.



Notes: *1 All interrupts
*2 NMI and IRQ0 to IRQ5

- When a transition is made between modes by means of an interrupt, the transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
- From any state except hardware standby mode, a transition to the reset state occurs when $\overline{RES}$ is driven low.
- From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ is driven low.

Figure 23A-1 Mode Transition Diagram

Table 23A.2 Low Power Dissipation Mode Transition Conditions

Pre-Transition State	Status of Control Bit at Transition				State After Transition Invoked by SLEEP Command	State After Transition Back from Low Power Mode Invoked by Interrupt
	SSBY	PSS	LSON	DTON		
High-speed/ Medium-speed	0	*	0	*	Sleep	High-speed/Medium-speed
	0	*	1	*	—	—
	1	0	0	*	Software standby	High-speed/Medium-speed
	1	0	1	*	—	—
	1	1	0	0	—	—
	1	1	1	0	—	—
	1	1	0	1	—	—
	1	1	1	1	—	—
Sub-active	0	0	*	*	—	—
	0	1	0	*	—	—
	0	1	1	*	—	—
	1	0	*	*	—	—
	1	1	0	0	—	High-speed
	1	1	1	0	—	—
	1	1	0	1	High-speed	—
	1	1	1	1	—	—

* : Don’t care

—: Do not set.

23A.1.1 Register Configuration

Power-down modes are controlled by the SBYCR, SCKCR, LPWRCR, TCSR (WDT1), and MSTPCR registers. Table 23A-3 summarizes these registers.

Table 23A-3 Power-Down Mode Registers

Name	Abbreviation	R/W	Initial Value	Address*
Standby control register	SBYCR	R/W	H'58	H'FDE4
System clock control register	SCKCR	R/W	H'00	H'FDE6
Low-power control register	LPWRCR	R/W	H'00	H'FDEC
Timer control/status register	TCSR	R/W	H'00	H'FFA2
Module stop control register A, B, C, D	MSTPCRA	R/W	H'3F	H'FDE8
	MSTPCRB	R/W	H'FF	H'FDE9
	MSTPCRC	R/W	H'FF	H'FDEA
	MSTPCRD	R/W	B'11******	H'FC60

Note: * Lower 16 bits of the address.

23A.2 Register Descriptions

23A.2.1 Standby Control Register (SBYCR)

Bit	:	7	6	5	4	3	2	1	0
		SSBY	STS2	STS1	STS0	OPE	—	—	—
Initial value	:	0	1	0	1	1	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	—	—	—

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

SBYCR is initialized to H'58 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Software Standby (SSBY): When making a low power dissipation mode transition by executing the SLEEP instruction, the operating mode is determined in combination with other control bits.

Note that the value of the SSBY bit does not change even when shifting between modes using interrupts.

Bit 7

SSBY	Description
0	Shifts to sleep mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. (Initial value)
1	Shifts to software standby mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode.

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the MCU wait time for clock stabilization when shifting to high-speed mode or medium-speed mode by using a specific interrupt or command to cancel software standby mode. With a quartz oscillator (Table 23A-5), select a wait time of 8ms (oscillation stabilization time) or more, depending on the operating frequency. With an external clock, there are no specific wait requirements.

Bit 6	Bit 5	Bit 4		
STS2	**STS1**	**STS0**	**Description**	
0	0	0	Standby time = 8192 states	
		1	Standby time = 16384 states	
	1	0	Standby time = 32768 states	
		1	Standby time = 65536 states	
1	0	0	Standby time = 131072 states	
		1	Standby time = 262144 states	(Initial value)
	1	0	Reserved	
		1	Standby time = 16 states	

Bit 3—Output Port Enable (OPE): This bit specifies whether the output of the address bus and bus control signals ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$) is retained or set to high-impedance state in the software standby mode.

Bit 3		
OPE	**Description**	
0	In software standby mode, address bus and bus control signals are high-impedance.	
1	In software standby mode, the output state of the address bus and bus control signals is retained.	(Initial value)

Bits 2 to 0—Reserved: These bits always return 0 when read, and cannot be written to.

23A.2.2 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	—	—	—	R/W	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ø clock output control and medium-speed mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—ø Clock Output Disable (PSTOP): In combination with the DDR of the applicable port, this bit controls ø output. See Section 23A.8, ø Clock Output Disable Function for details.

	Description			
Bit 7				
PSTOP	**High-Speed Mode, Medium-Speed Mode**	**Sleep Mode**	**Software Standby Mode**	**Hardware Standby Mode**
0	ø output (initial value)	ø output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bits 6 to 4—Reserved: These bits are always read as 0 and cannot be modified.

Bit 3—Frequency Multiplication Factor Switching Mode Select (STCS): Selects the operation when the PLL circuit frequency multiplication factor is changed.

Bit 3	
STCS	**Description**
0	Specified multiplication factor is valid after transition to software standby mode (Initial value)
1	Specified multiplication factor is valid immediately after STC bits are rewritten

Bits 2 to 0—System clock select (SCK2 to SCK0): These bits select the bus master clock in high-speed mode, and medium-speed mode.

Bit 2	**Bit 1**	**Bit 0**		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master in high-speed mode	(Initial value)
		1	Medium-speed clock is ø/2	
	1	0	Medium-speed clock is ø/4	
		1	Medium-speed clock is ø/8	
1	0	0	Medium-speed clock is ø/16	
		1	Medium-speed clock is ø/32	
	1	—	—	

23A.2.3 Low-Power Control Register (LPWRCR)

Bit	:	7	6	5	4	3	2	1	0
		DTON	LSON	NESEL	SUBSTP	RFCUT	—	STC1	STC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

The LPWRCR is an 8-bit read/write register that controls the low power dissipation modes.

The LPWRCR is initialized to H'00 at a reset and when in hardware standby mode. It is not initialized in software standby mode. The following describes bits 7 to 2. For details of other bits, see Section 22A.2.2, Low-Power Control Register.

Bits 7 to 3—Reserved: Bits DTON, LSON, NESEL, SUBSTP and RFCUT must always be written with 0 in the HD64F2636F and HD64F2638F, as this version does not support subclock operation.

Bit 2—Reserved: Only write 0 to this bit.

23A.2.4 Timer Control/Status Register (TCSR)

Bit	:	7	6	5	4	3	2	1	0
		OVF	WT/$\overline{\text{IT}}$	TME	PSS	RST/$\overline{\text{NMI}}$	CKS2	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Only write 0 to clear the flag.

TCSR is an 8-bit read/write register that selects the clock input to WDT1 TCNT and the mode.

Here, we describe bit 4. For details of the other bits in this register, see Section 12.2.2, Timer Control/Status Register (TCSR).

The TCSR is initialized to H'00 at a reset and when in hardware standby mode. It is not initialized in software standby mode.

Bit 4—Reserved: The PSS bit must always be written with 0 since no subclock functions are available in versions other than the U-mask and W-mask versions.

23A.2.5 Module Stop Control Register (MSTPCR)

MSTPCRA

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRB

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC

Bit	:	7	6	5	4	3	2	1	0
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRD

Bit	:	7	6	5	4	3	2	1	0
		MSTPD7	MSTPD6	MSTPD5	MSTPD4	MSTPD3	MSTPD2	MSTPD1	MSTPD0
Initial value	:	1	1	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	:	R/W	R/W	—	—	—	—	—	—

MSTPCR, comprising four 8-bit readable/writable registers, performs module stop mode control.

MSTPCRA to MSTPCRC are initialized to H'3FFFFF by a reset and in hardware standby mode. MSTPCRD is initialized to B'11****** by a reset and in hardware standby mode. They are not initialized in software standby mode.

MSTPCRA/MSTPCRB/MSTPCRC Bits 7 to 0, MSTPCRD Bits 7 and 6—Module Stop (MSTPA7 to MSTPA0, MSTPB7 to MSTPB0, MSTPC7 to MSTPC0, MSTPD7 and MSTPD6): These bits specify module stop mode. See table 23A-4 for the method of selecting the on-chip peripheral functions.

MSTPCRA/MSTPCRB/ MSTPCRC Bits 7 to 0, MSTPCRD Bits 7 and 6	
MSTPA7 to MSTPA0, MSTPB7 to MSTPB0, MSTPC7 to MSTPC0, MSTPD7 and MSTPD6	**Description**
0	Module stop mode is cleared (initial value of MSTPA7 and MSTPA6)
1	Module stop mode is set (initial value of MSTPA5–0, MSTPB7–0, MSTPC7–0, and MSTPC7, 6)

23A.3 Medium-Speed Mode

In high-speed mode, when the SCK2 to SCK0 bits in SCKCR are set to 1, the operating mode changes to medium-speed mode as soon as the current bus cycle ends. In medium-speed mode, the CPU operates on the operating clock (ø/2, ø/4, ø/8, ø/16, or ø/32) specified by the SCK2 to SCK0 bits. The bus masters other than the CPU (DTC) also operate in medium-speed mode. On-chip supporting modules other than the bus masters always operate on the high-speed clock (ø).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if ø/4 is selected as the operating clock, on-chip memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, and LSON bit in LPWRCR is cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored.

When the SLEEP instruction is executed with the SSBY bit = 1, LPWRCR LSON bit = 0, and TCSR (WDT1) PSS bit = 0, operation shifts to the software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is set low and medium-speed mode is cancelled, operation shifts to the reset state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 23A-2 shows the timing for transition to and clearance of medium-speed mode.

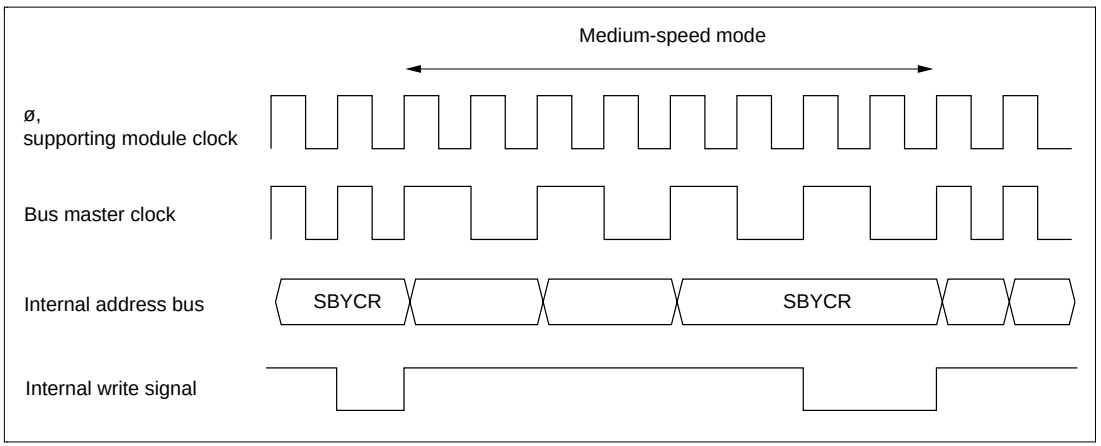


Figure 23A-2 Medium-Speed Mode Transition and Clearance Timing

23A.4 Sleep Mode

23A.4.1 Sleep Mode

When the SLEEP instruction is executed when the SBYCR SSBY bit = 0 and the LPWRCR LSON bit = 0, the CPU enters the sleep mode. In sleep mode, CPU operation stops but the contents of the CPUís internal registers are retained. Other supporting modules do not stop.

23A.4.2 Exiting Sleep Mode

Sleep mode is exited by any interrupt, or signals at the $\overline{\text{RES}}$, or $\overline{\text{STBY}}$ pins.

Exiting Sleep Mode by Interrupts: When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by the CPU.

Exiting Sleep Mode by $\overline{\text{RES}}$ pin: Setting the $\overline{\text{RES}}$ pin level Low selects the reset state. After the stipulated reset input duration, driving the $\overline{\text{RES}}$ pin High starts the CPU performing reset exception processing.

Exiting Sleep Mode by $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin level is driven Low, a transition is made to hardware standby mode.

23A.5 Module Stop Mode

23A.5.1 Module Stop Mode

Module stop mode can be set for individual on-chip supporting modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

Table 23A-4 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI, Motor control PWM, A/D converter and HCAN are retained.

After reset clearance, all modules other than DTC are in module stop mode.

When an on-chip supporting module is in module stop mode, read/write access to its registers is disabled.

Table 23A-4 MSTP Bits and Corresponding On-Chip Supporting Modules

Register	Bit	Module
MSTPCRA	MSTPA6	Data transfer controller (DTC)
	MSTPA5	16-bit timer pulse unit (TPU)
	MSTPA3	Programmable pulse generator (PPG)
	MSTPA2	D/A converter (channel 0, 1)
	MSTPA1	A/D converter
	MSTPA0*[1]	
MSTPCRB	MSTPB7	Serial communication interface 0 (SCI0)
	MSTPB6	Serial communication interface 1 (SCI1)
	MSTPB5	Serial communication interface 2 (SCI2)
	MSTPB4	I^2C bus interface 0 (IIC0)*[2]
	MSTPB3	I^2C bus interface 1 (IIC1)*[2]
	MSTPB0*[1]	
MSTPCRC	MSTPC4	PC break controller (PBC)
	MSTPC3	HCAN0
	MSTPC2	HCAN1
	MSTPC1*[1]	
	MSTPC0*[1]	
MSTPCRD	MSTPD7	Motor control PWM (PWM)
	MSTPD6*[1]	

Notes: *1 MSTPA0, MSTPB0 and MSTPC1 to MSTPC0 and MSTPD6 are readable/writable bits with an initial value of 1.

*2 The I^2C bus interface is available as an option in the H8S/2638 and H8S/2639. In other products, MSTB4 and MSTB3 are readable and writable bits that have 1 as their initial value.

23A.5.2 Usage Notes

DTC Module Stop: Depending on the operating status of the DTC, the MSTPA7 and MSTPA6 bits may not be set to 1. Setting of the DTC module stop mode should be carried out only when the respective module is not activated.

For details, refer to section 8, Data Transfer Controller (DTC).

On-Chip Supporting Module Interrupt: Relevant interrupt operations cannot be performed in module stop mode. Consequently, if module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Writing to MSTPCR: MSTPCR should only be written to by the CPU.

23A.6 Software Standby Mode

23A.6.1 Software Standby Mode

A transition is made to software standby mode when the SLEEP instruction is executed when the SBYCR SSBY bit = 1 and the LPWRCR LSON bit = 0, and the TCSR (WDT1) PSS bit = 0. In this mode, the CPU, on-chip supporting modules, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip supporting modules other than the SCI, A/D converter, Motor control, PWM, HCAN and I/O ports, are retained. Whether the address bus and bus control signals are placed in the high-impedance state.

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

23A.6.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$), or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an NMI or IRQ0 to IRQ5 interrupt request signal is input, clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SYSCR, stable clocks are supplied to the entire chip, software standby mode is cleared, and interrupt exception handling is started.

  When clearing software standby mode with an IRQ0 to IRQ5 interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ5 is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.

- Clearing with the $\overline{\text{RES}}$ pin

  When the $\overline{\text{RES}}$ pin is driven low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire chip. Note that the $\overline{\text{RES}}$ pin must be held low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin goes high, the CPU begins reset exception handling.

- Clearing with the $\overline{\text{STBY}}$ pin

  When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

23A.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode

Bits STS2 to STS0 in SBYCR should be set as described below.

Using a Crystal Oscillator: Set bits STS2 to STS0 so that the standby time is at least 8 ms (the oscillation stabilization time).

Table 23A-5 shows the standby times for different operating frequencies and settings of bits STS2 to STS0.

Table 23A-5 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz	Unit
0	0	0	8192 states	0.41	0.51	0.68	0.8	1.0	1.3	2.0	ms
		1	16384 states	0.82	1.0	1.3	1.6	2.0	2.7	4.1	
	1	0	32768 states	1.6	2.0	2.7	3.3	4.1	5.5	8.2	
		1	65536 states	3.3	4.1	5.5	6.6	8.2	10.9	16.4	
1	0	0	131072 states	6.6	8.2	10.9	13.1	16.4	21.8	32.8	
		1	262144 states	13.1	16.4	21.8	26.2	32.8	43.6	65.6	
	1	0	Reserved	—	—	—	—	—	—	—	µs
		1	16 states*	0.8	1.0	1.3	1.6	2.0	2.6	4.0	

: Recommended time setting

Note: * Do not use this setting in the version with built-in flash memory.

23A.6.4 Software Standby Mode Application Example

Figure 23A-3 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software standby mode is cleared at the rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.

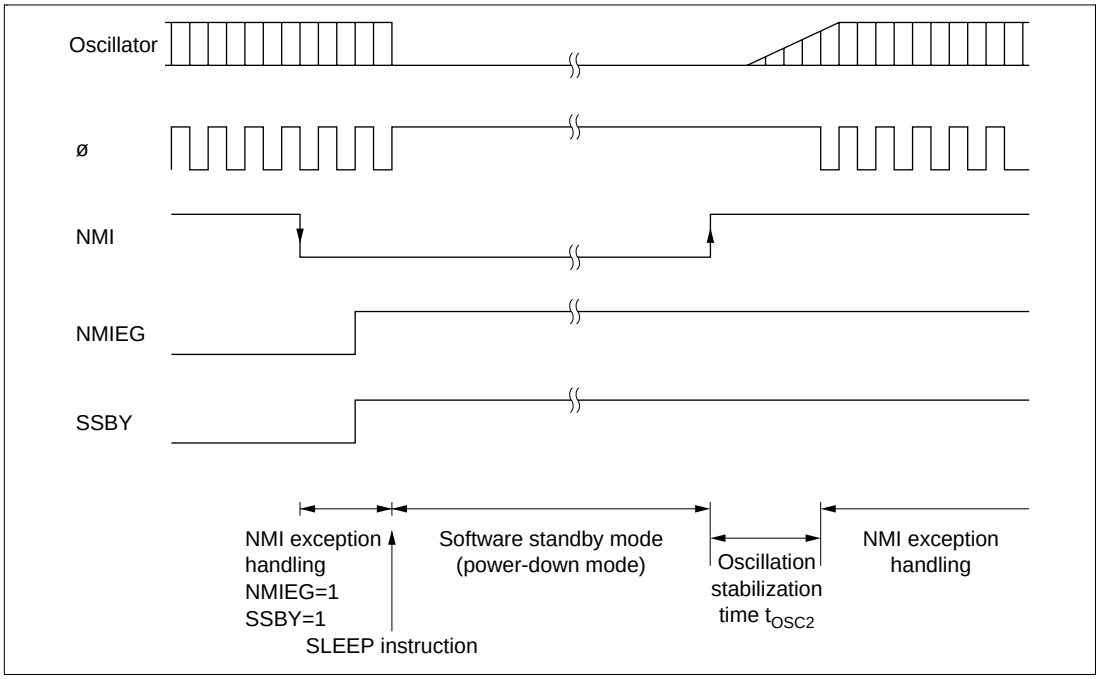


Figure 23A-3 Software Standby Mode Application Example

23A.6.5 Usage Notes

I/O Port Status: In software standby mode, I/O port states are retained. If the OPE bit is set to 1, the address bus and bus control signal output is also retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

Current Dissipation during Oscillation Stabilization Wait Period: Current dissipation increases during the oscillation stabilization wait period.

Write Data Buffer Function: The write data buffer function and software standby mode cannot be used at the same time. When the write data buffer function is used, the WDBE bit in BCRL should be cleared to 0 to cancel the write data buffer function before entering software standby mode. Also check that external writes have finished, by reading external addresses, etc., before executing a SLEEP instruction to enter software standby mode. See section 7.9, Write Data Buffer Function, for details of the write data buffer function.

23A.7 Hardware Standby Mode

23A.7.1 Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power dissipation. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD2 to MD0) while the chip is in hardware standby mode.

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (at least 8 ms—the oscillation stabilization time—when using a crystal oscillator). When the $\overline{\text{RES}}$ pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

23A.7.2 Hardware Standby Mode Timing

Figure 23A-4 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high, waiting for the oscillation stabilization time, then changing the $\overline{\text{RES}}$ pin from low to high.

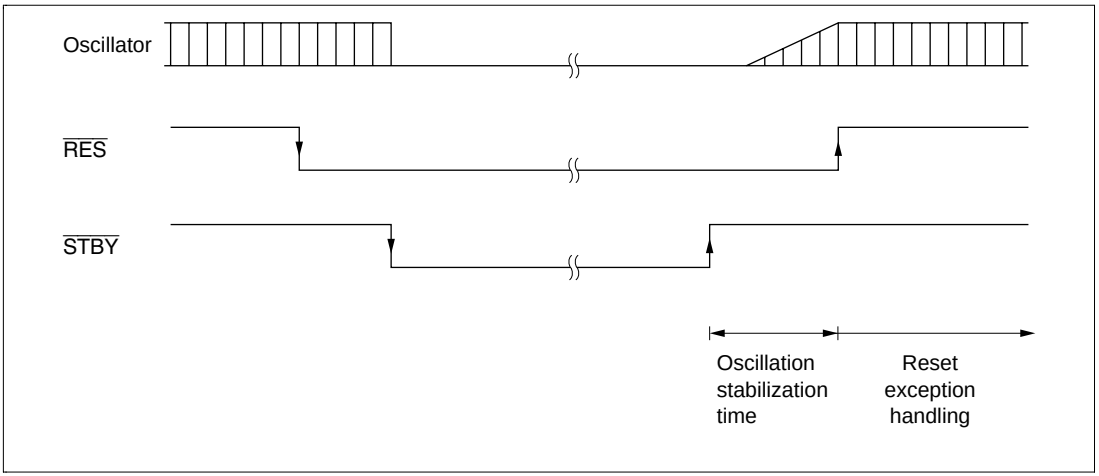


Figure 23A-4 Hardware Standby Mode Timing

23A.8 ø Clock Output Disabling Function

Output of the ø clock can be controlled by means of the PSTOP bit in SCKCR, and DDR for the corresponding port. When the PSTOP bit is set to 1, the ø clock stops at the end of the bus cycle, and ø output goes high. ø clock output is enabled when the PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0, ø clock output is disabled and input port mode is set. Table 23A-6 shows the state of the ø pin in each processing state.

Table 23A-6 ø Pin State in Each Processing State

DDR	0	1	1
PSTOP	—	0	1
Hardware standby mode	High impedance	High impedance	High impedance
Software standby	High impedance	Fixed high	Fixed high
Sleep mode	High impedance	ø output	Fixed high
High-speed mode, medium-speed mode	High impedance	ø output	Fixed high

Section 23B Power-Down Modes [HD64F2636UF, HD6432636UF, HD64F2638UF, HD6432638UF, HD64F2638WF, HD6432638WF, HD64F2639UF, HD6432639UF, HD64F2639WF, HD6432639WF]

23B.1 Overview

In addition to the normal program execution state, the chip has nine power-down modes in which operation of the CPU and oscillator is halted and power dissipation is reduced. Low-power operation can be achieved by individually controlling the CPU, on-chip supporting modules, and so on.

The chip operating modes are as follows:

(1) High-speed mode
(2) Medium-speed mode
(3) Subactive mode* (U-mask, W-mask version only)
(4) Sleep mode
(5) Subsleep mode* (U-mask, W-mask version only)
(6) Watch mode* (U-mask, W-mask version only)
(7) Module stop mode
(8) Software standby mode
(9) Hardware standby mode

(2) to (9) are low power dissipation states. Sleep mode and sub-sleep mode are CPU states, medium-speed mode is a CPU and bus master state, sub-active mode is a CPU and bus master and internal peripheral function state, and module stop mode is an internal peripheral function (including bus masters other than the CPU) state. Some of these states can be combined.

After a reset, the LSI is in high-speed mode with modules other than the DTC in module stop mode.

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

Table 23B-1 shows the internal state of the LSI in the respective modes. Table 23B-2 shows the conditions for shifting between the low power dissipation modes.

Figure 23B-1 is a mode transition diagram.

Table 23B-1 LSI Internal States in Each Mode (H8S/2636, H8S/2638)

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Sub-active	Subsleep	Software Standby	Hardware Standby
System clock pulse generator		Function-ing	Function-ing	Function-ing	Function-ing	Halted	Halted	Halted	Halted	Halted
Subclock pulse generator		Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Halted
CPU	Instructions Registers	Function-ing	Medium-speed operation	Halted (retained)	High/ medium-speed operation	Halted (retained)	Subclock operation	Halted (retained)	Halted (retained)	Halted (undefined)
External interrupts	NMI	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Halted
	IRQ0–IRQ5									
Peripheral functions	WDT1	Function-ing	Function-ing	Function-ing	—	Subclock operation	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	WDT0	Function-ing	Function-ing	Function ing	—	Halted (retained)	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	DTC	Function-ing	Medium-speed operation	Function-ing	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	PBC	Function-ing	Medium-speed operation	Function-ing	Halted (retained)	Halted (retained)	Subclock operation	Halted (retained)	Halted (retained)	Halted (reset)
	TPU	Function-ing	Function-ing	Function-ing	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	IIC0*									
	IIC1*									
	PPG									
	D/A0, 1									
	SCI0	Function-ing	Function-ing	Function-ing	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)
	SCI1									
	SCI2									
	PWM									
	A/D									
	RAM	Function-ing	Function-ing	Function-ing (DTC)	Function-ing	Retained	Function-ing	Retained	Retained	Retained
	I/O	Function-ing	Function-ing	Function-ing	Function-ing	Retained	Function-ing	Retained	Retained	High impedance
	HCAN	Function-ing	Function-ing	Function-ing	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)

Notes: “Halted (retained)” means that internal register values are retained. The internal state is “operation suspended.”

“Halted (reset)” means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).

* The I^2C bus interface is available as an option in the H8S/2638 and H8S/2639.

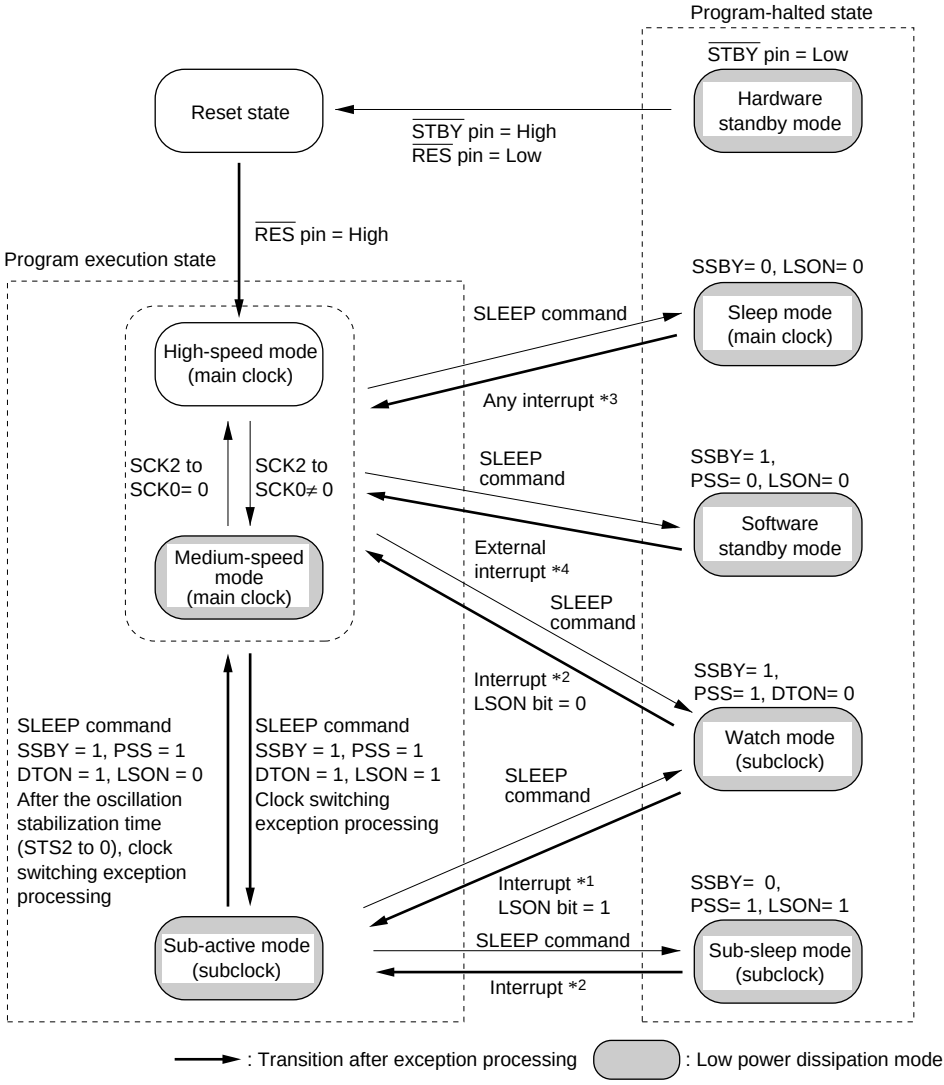
Table 23B-2 LSI Internal States in Each Mode (H8S/2639 Series)

Function		High-Speed	Medium-Speed	Sleep	Module Stop	Watch	Sub-active	Subsleep	Software Standby	Hardware Standby
System clock (φ)		Function-ing	Function-ing	Function-ing	Function-ing	Halted	Halted	Halted	Halted	Halted
Clock pulse generator		Function-ing	Function-ing	Function-ing	Function-ing	Function-ing*1	Function-ing*1	Function-ing*1	Function-ing*1	Halted
Subclock (φ Sub)*1		Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Halted
CPU	Instructions Registers	Function-ing	Medium-speed operation	Halted (retained)	High/ medium-speed operation	Halted (retained)	Subclock operation	Halted (retained)	Halted (retained)	Halted (undefined)
External interrupts	NMI	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Function-ing	Halted
	IRQ0–IRQ5									
Peripheral functions	WDT1	Function-ing	Function-ing	Function-ing	—	Subclock operation	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	WDT0	Function-ing	Function-ing	Function ing	—	Halted (retained)	Subclock operation	Subclock operation	Halted (retained)	Halted (reset)
	DTC	Function-ing	Medium-speed operation	Function-ing	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	TPU	Function-ing	Function-ing	Function-ing	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (retained)	Halted (reset)
	IIC0*2									
	IIC1*2									
	RBC									
	PPG									
	D/A0, 1									
	SCI0	Function-ing	Function-ing	Function-ing	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)
	SCI1									
	SCI2									
	PWM									
	A/D									
	RAM	Function-ing	Function-ing	Function-ing (DTC)	Function-ing	Retained	Function-ing	Retained	Retained	Retained
	I/O	Function-ing	Function-ing	Function-ing	Function-ing	Retained	Function-ing	Retained	Retained	High impedance
	HCAN	Function-ing	Function-ing	Function-ing	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)	Halted (reset)

Notes: "Halted (retained)" means that internal register values are retained. The internal state is "operation suspended."

"Halted (reset)" means that internal register values and internal states are initialized.

In module stop mode, only modules for which a stop setting has been made are halted (reset or retained).



Notes: *1 NMI, IRQ0 to IRQ5, and WDT1 interrupts
*2 NMI, IRQ0 to IRQ5, IWDT0 interrupts, and WDT1 interrupt.
*3 All interrupts
*4 NMI and IRQ0 to IRQ5

- When a transition is made between modes by means of an interrupt, the transition cannot be made on interrupt source generation alone. Ensure that interrupt handling is performed after accepting the interrupt request.
- From any state except hardware standby mode, a transition to the reset state occurs when $\overline{RES}$ is driven Low.
- From any state, a transition to hardware standby mode occurs when $\overline{STBY}$ is driven low.
- Always select high-speed mode before making a transition to watch mode or sub-active mode.

Figure 23B-1 Mode Transition Diagram

Table 23B.3 Low Power Dissipation Mode Transition Conditions

Pre-Transition State	Status of Control Bit at Transition				State After Transition Invoked by SLEEP Command	State After Transition Back from Low Power Mode Invoked by Interrupt
	SSBY	PSS	LSON	DTON		
High-speed/ Medium-speed	0	*	0	*	Sleep	High-speed/Medium-speed
	0	*	1	*	—	—
	1	0	0	*	Software standby	High-speed/Medium-speed
	1	0	1	*	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Sub-active
	1	1	0	1	—	—
	1	1	1	1	Sub-active	—
Sub-active	0	0	*	*	—	—
	0	1	0	*	—	—
	0	1	1	*	Sub-sleep	Sub-active
	1	0	*	*	—	—
	1	1	0	0	Watch	High-speed
	1	1	1	0	Watch	Sub-active
	1	1	0	1	High-speed	—
	1	1	1	1	—	—

* : Don’t care

—: Do not set.

23B.1.1 Register Configuration

Power-down modes are controlled by the SBYCR, SCKCR, LPWRCR, TCSR (WDT1), and MSTPCR registers. Table 23B-4 summarizes these registers.

Table 23B-4 Power-Down Mode Registers

Name	Abbreviation	R/W	Initial Value	Address*
Standby control register	SBYCR	R/W	H'58	H'FDE4
System clock control register	SCKCR	R/W	H'00	H'FDE6
Low-power control register	LPWRCR	R/W	H'00	H'FDEC
Timer control/status register	TCSR	R/W	H'00	H'FFA2
Module stop control register A, B, C, D	MSTPCRA	R/W	H'3F	H'FDE8
	MSTPCRB	R/W	H'FF	H'FDE9
	MSTPCRC	R/W	H'FF	H'FDEA
	MSTPCRD	R/W	B'11******	H'FC60

Note: * Lower 16 bits of the address.

23B.2 Register Descriptions

23B.2.1 Standby Control Register (SBYCR)

Bit	:	7	6	5	4	3	2	1	0
		SSBY	STS2	STS1	STS0	OPE	—	—	—
Initial value	:	0	1	0	1	1	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	—	—	—

SBYCR is an 8-bit readable/writable register that performs power-down mode control.

SBYCR is initialized to H'58 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—Software Standby (SSBY): When making a low power dissipation mode transition by executing the SLEEP instruction, the operating mode is determined in combination with other control bits.

Note that the value of the SSBY bit does not change even when shifting between modes using interrupts.

Bit 7

SSBY	Description
0	Shifts to sleep mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. Shifts to sub-sleep mode when the SLEEP instruction is executed in sub-active mode. (Initial value)
1	Shifts to software standby mode, sub-active mode, and watch mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode. Shifts to watch mode or high-speed mode when the SLEEP instruction is executed in sub-active mode.

Bits 6 to 4—Standby Timer Select 2 to 0 (STS2 to STS0): These bits select the MCU wait time for clock stabilization when shifting to high-speed mode or medium-speed mode by using a specific interrupt or command to cancel software standby mode, watch mode, or sub-active mode. With a quartz oscillator (Table 23B-6), select a wait time of 8ms (oscillation stabilization time) or more, depending on the operating frequency. With an external clock, there are no specific wait requirements.

Bit 6	Bit 5	Bit 4		
STS2	**STS1**	**STS0**	**Description**	
0	0	0	Standby time = 8192 states	
		1	Standby time = 16384 states	
	1	0	Standby time = 32768 states	
		1	Standby time = 65536 states	
1	0	0	Standby time = 131072 states	
		1	Standby time = 262144 states	(Initial value)
	1	0	Reserved	
		1	Standby time = 16 states	

Bit 3—Output Port Enable (OPE): This bit specifies whether the output of the address bus and bus control signals ($\overline{AS}$, $\overline{RD}$, $\overline{HWR}$, $\overline{LWR}$) is retained or set to high-impedance state in the software standby mode, watch mode, and when making a direct transition.

Bit 3		
OPE	**Description**	
0	In software standby mode, watch mode, and when making a direct transition, address bus and bus control signals are high-impedance.	
1	In software standby mode, watch mode, and when making a direct transition, the output state of the address bus and bus control signals is retained.	(Initial value)

Bits 2 to 0—Reserved: These bits always return 0 when read, and cannot be written to.

23B.2.2 System Clock Control Register (SCKCR)

Bit	:	7	6	5	4	3	2	1	0
		PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	—	—	—	R/W	R/W	R/W	R/W

SCKCR is an 8-bit readable/writable register that performs ø clock output control and medium-speed mode control.

SCKCR is initialized to H'00 by a reset and in hardware standby mode. It is not initialized in software standby mode.

Bit 7—ø Clock Output Disable (PSTOP): In combination with the DDR of the applicable port, this bit controls ø output. See Section 23B.12, ø Clock Output Disable Function for details.

	Description			
Bit 7	**High-Speed Mode, Medium-Speed Mode, Sub-Active Mode**	**Sleep Mode, Sub-Sleep Mode**	**Software Standby Mode, Watch Mode, Direct Transition**	**Hardware Standby Mode**
PSTOP				
0	ø output (initial value)	ø output	Fixed high	High impedance
1	Fixed high	Fixed high	Fixed high	High impedance

Bits 6 to 4—Reserved: These bits are always read as 0 and cannot be modified.

Bit 3—Frequency Multiplication Factor Switching Mode Select (STCS): Selects the operation when the PLL circuit frequency multiplication factor is changed.

Bit 3	
STCS	**Description**
0	Specified multiplication factor is valid after transition to software standby mode, watch mode, or subactive mode (Initial value)
1	Specified multiplication factor is valid immediately after STC bits are rewritten

Bits 2 to 0—System clock select (SCK2 to SCK0): These bits select the bus master clock in high-speed mode, medium-speed mode, and sub-active mode.

Set SCK2 to SCK0 all to 0 when shifting to operation in watch mode or sub-active mode.

Bit 2	Bit 1	Bit 0		
SCK2	**SCK1**	**SCK0**	**Description**	
0	0	0	Bus master in high-speed mode	(Initial value)
		1	Medium-speed clock is ø/2	
	1	0	Medium-speed clock is ø/4	
		1	Medium-speed clock is ø/8	
1	0	0	Medium-speed clock is ø/16	
		1	Medium-speed clock is ø/32	
	1	—	—	

23B.2.3 Low-Power Control Register (LPWRCR)

Bit	:	7	6	5	4	3	2	1	0
		DTON*	LSON*	NESEL*	SUBSTP*	RFCUT*	—	STC1	STC0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Note: * Bits 7 to 3 in LPWRCR are valid in the U-mask and W-mask versions; they are reserved bits in all other versions.
See section 23B.2.3, Low-Power Control Register (LPWRCR), for more information.

The LPWRCR is an 8-bit read/write register that controls the low power dissipation modes.

The LPWRCR is initialized to H'00 at a reset and when in hardware standby mode. It is not initialized in software standby mode. The following describes bits 7 to 2. For details of other bits, see Section 22B.2.2, Low-Power Control Register.

Bit 7—Direct Transition ON Flag (DTON): When shifting to low power dissipation mode by executing the SLEEP instruction, this bit specifies whether or not to make a direct transition between high-speed mode or medium-speed mode and the sub-active modes. The selected operating mode after executing the SLEEP instruction is determined by the combination of other control bits.

Bit 7	
DTON	**Description**
0	• When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, software standby mode, or watch mode*. • When the SLEEP instruction is executed in sub-active mode, operation shifts to sub-sleep mode or watch mode. (Initial value)
1	• When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts directly to sub-active mode*, or shifts to sleep mode or software standby mode. • When the SLEEP instruction is executed in sub-active mode, operation shifts directly to high-speed mode, or shifts to sub-sleep mode.

Note: * Always set high-speed mode when shifting to watch mode or sub-active mode.

Bit 6—Low-Speed ON Flag (LSON): When shifting to low power dissipation mode by executing the SLEEP instruction, this bit specifies the operating mode, in combination with other control bits. This bit also controls whether to shift to high-speed mode or sub-active mode when watch mode is cancelled.

Bit 6	
LSON	**Description**
0	• When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, software standby mode, or watch mode*. • When the SLEEP instruction is executed in sub-active mode, operation shifts to watch mode or shifts directly to high-speed mode. • Operation shifts to high-speed mode when watch mode is cancelled. (Initial value)
1	• When the SLEEP instruction is executed in high-speed mode, operation shifts to watch mode or sub-active mode. • When the SLEEP instruction is executed in sub-active mode, operation shifts to sub-sleep mode or watch mode. • Operation shifts to sub-active mode when watch mode is cancelled.

Note: * Always set high-speed mode when shifting to watch mode or sub-active mode.

Bit 5—Noise Elimination Sampling Frequency Select (NESEL): This bit selects the sampling frequency of the subclock (øSUB) generated by the subclock oscillator is sampled by the clock (ø) generated by the system clock oscillator. Set this bit to 0 when ø=5MHz or more. This setting is disabled in sub-active mode, sub-sleep mode, and watch mode.

Bit 5

NESEL	Description	
0	Sampling using 1/32 xø	(Initial value)
1	Sampling using 1/4 xø	

Bit 4—Subclock enable (SUBSTP): This bit enables/disables subclock generation.

Bit 4

SUBSTP	Description	
0	Enables subclock generation	(Initial value)
1	Disables subclock generation	

Bit 3—Oscillation Circuit Feedback Resistance Control Bit (RFCUT): This bit turns the internal feedback resistance of the main clock oscillation circuit ON/OFF.

Bit 3

RFCUT	Description	
0	When the main clock is oscillating, sets the feedback resistance ON. When the main clock is stopped, sets the feedback resistance OFF.	(Initial value)
1	Sets the feedback resistance OFF.	

Bit 2—Reserved: Only write 0 to this bit.

23B.2.4 Timer Control/Status Register (TCSR)

Bit	:	7	6	5	4	3	2	1	0
		OVF	WT/$\overline{IT}$	TME	PSS*2	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*1	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Notes: *1 Only write 0 to clear the flag.

*2 Bit 4 (PSS) in TCSR of WDT1 is valid in the U-mask and W-mask versions. In versions other than the U-mask and W-mask versions, however, the PSS bit must always be written with 0 since no subclock functions are available.
See section 23B.2.4, Timer Control/Status Register (TCSR), for details.

TCSR is an 8-bit read/write register that selects the clock input to WDT1 TCNT and the mode.

Here, we describe bit 4. For details of the other bits in this register, see Section 12.2.2, Timer Control/Status Register (TCSR).

The TCSR is initialized to H'00 at a reset and when in hardware standby mode. It is not initialized in software standby mode.

Bit 4—Prescaler select (PSS): This bit selects the clock source input to WDT1 TCNT.

It also controls operation when shifting low power dissipation modes. The operating mode selected after the SLEEP instruction is executed is determined in combination with other control bits.

For details, see the description for clock selection in Section 12.2.2, Timer Control/Status Register (TCSR), and this section.

Bit 4 PSS	Description
0	• TCNT counts the divided clock from the ø -based prescaler (PSM). • When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode or software standby mode. (Initial value)
1	• TCNT counts the divided clock from the øsubclock-based prescaler (PSS). • When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, watch mode*1,*2, or sub-active mode*1,*2. • When the SLEEP instruction is executed in sub-active mode*2, operation shifts to sub-sleep mode*2, watch mode*2, or high-speed mode.

Notes: *1 Always set high-speed mode when shifting to watch mode or sub-active mode.

*2 Bit 4 (PSS) in TCSR of WDT1 is valid in the U-mask and W-mask versions. In versions other than the U-mask and W-mask versions, however, the PSS bit must always be written with 0 since no subclock functions are available.

23B.2.5 Module Stop Control Register (MSTPCR)

MSTPCRA

Bit	:	7	6	5	4	3	2	1	0
		MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	:	0	0	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRB

Bit	:	7	6	5	4	3	2	1	0
		MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRC

Bit	:	7	6	5	4	3	2	1	0
		MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	:	1	1	1	1	1	1	1	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MSTPCRD

Bit	:	7	6	5	4	3	2	1	0
		MSTPD7	MSTPD6	MSTPD5	MSTPD4	MSTPD3	MSTPD2	MSTPD1	MSTPD0
Initial value	:	1	1	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
R/W	:	R/W	R/W	—	—	—	—	—	—

MSTPCR, comprising four 8-bit readable/writable registers, performs module stop mode control.

MSTPCRA to MSTPCRC are initialized to H'3FFFFF by a reset and in hardware standby mode. MSTPCRD is initialized to B'11****** by a reset and in hardware standby mode. They are not initialized in software standby mode.

MSTPCRA/MSTPCRB/MSTPCRC Bits 7 to 0, MSTPCRD Bits 7 and 6—Module Stop (MSTPA7 to MSTPA0, MSTPB7 to MSTPB0, MSTPC7 to MSTPC0, MSTPD7 and MSTPD6): These bits specify module stop mode. See table 23B-5 for the method of selecting the on-chip peripheral functions.

MSTPCRA/MSTPCRB/ MSTPCRC Bits 7 to 0, MSTPCRD Bits 7 and 6	
MSTPA7 to MSTPA0, MSTPB7 to MSTPB0, MSTPC7 to MSTPC0, MSTPD7 and MSTPD6	**Description**
0	Module stop mode is cleared (initial value of MSTPA7 and MSTPA6)
1	Module stop mode is set (initial value of MSTPA5–0, MSTPB7–0, MSTPC7–0, and MSTPC7, 6)

23B.3 Medium-Speed Mode

In high-speed mode, when the SCK2 to SCK0 bits in SCKCR are set to 1, the operating mode changes to medium-speed mode as soon as the current bus cycle ends. In medium-speed mode, the CPU operates on the operating clock (ø/2, ø/4, ø/8, ø/16, or ø/32) specified by the SCK2 to SCK0 bits. The bus masters other than the CPU (DTC) also operate in medium-speed mode. On-chip supporting modules other than the bus masters always operate on the high-speed clock (ø).

In medium-speed mode, a bus access is executed in the specified number of states with respect to the bus master operating clock. For example, if ø/4 is selected as the operating clock, on-chip memory is accessed in 4 states, and internal I/O registers in 8 states.

Medium-speed mode is cleared by clearing all of bits SCK2 to SCK0 to 0. A transition is made to high-speed mode and medium-speed mode is cleared at the end of the current bus cycle.

If a SLEEP instruction is executed when the SSBY bit in SBYCR is cleared to 0, and LSON bit in LPWRCR is cleared to 0, a transition is made to sleep mode. When sleep mode is cleared by an interrupt, medium-speed mode is restored.

When the SLEEP instruction is executed with the SSBY bit = 1, LPWRCR LSON bit = 0, and TCSR (WDT1) PSS bit = 0, operation shifts to the software standby mode. When software standby mode is cleared by an external interrupt, medium-speed mode is restored.

When the $\overline{\text{RES}}$ pin is set low and medium-speed mode is cancelled, operation shifts to the reset state. The same applies in the case of a reset caused by overflow of the watchdog timer.

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode.

Figure 23B-2 shows the timing for transition to and clearance of medium-speed mode.

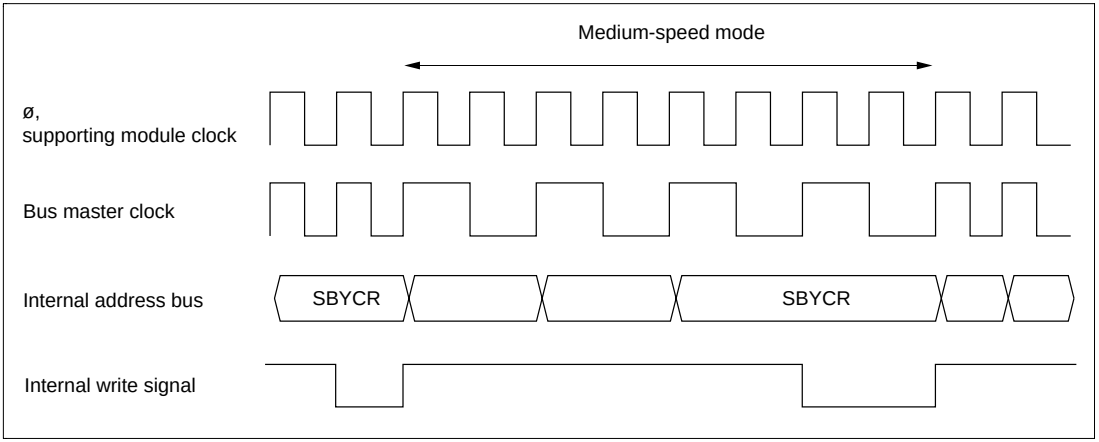


Figure 23B-2 Medium-Speed Mode Transition and Clearance Timing

23B.4 Sleep Mode

23B.4.1 Sleep Mode

When the SLEEP instruction is executed when the SBYCR SSBY bit = 0 and the LPWRCR LSON bit = 0, the CPU enters the sleep mode. In sleep mode, CPU operation stops but the contents of the CPUís internal registers are retained. Other supporting modules do not stop.

23B.4.2 Exiting Sleep Mode

Sleep mode is exited by any interrupt, or signals at the $\overline{\text{RES}}$, or $\overline{\text{STBY}}$ pins.

Exiting Sleep Mode by Interrupts: When an interrupt occurs, sleep mode is exited and interrupt exception processing starts. Sleep mode is not exited if the interrupt is disabled, or interrupts other than NMI are masked by the CPU.

Exiting Sleep Mode by $\overline{\text{RES}}$ pin: Setting the $\overline{\text{RES}}$ pin level Low selects the reset state. After the stipulated reset input duration, driving the $\overline{\text{RES}}$ pin High starts the CPU performing reset exception processing.

Exiting Sleep Mode by $\overline{\text{STBY}}$ Pin: When the $\overline{\text{STBY}}$ pin level is driven Low, a transition is made to hardware standby mode.

23B.5 Module Stop Mode

23B.5.1 Module Stop Mode

Module stop mode can be set for individual on-chip supporting modules.

When the corresponding MSTP bit in MSTPCR is set to 1, module operation stops at the end of the bus cycle and a transition is made to module stop mode. The CPU continues operating independently.

Table 23B-5 shows MSTP bits and the corresponding on-chip supporting modules.

When the corresponding MSTP bit is cleared to 0, module stop mode is cleared and the module starts operating at the end of the bus cycle. In module stop mode, the internal states of modules other than the SCI, Motor control PWM, A/D converter and HCAN are retained.

After reset clearance, all modules other than DTC are in module stop mode.

When an on-chip supporting module is in module stop mode, read/write access to its registers is disabled.

Table 23B-5 MSTP Bits and Corresponding On-Chip Supporting Modules

Register	Bit	Module
MSTPCRA	MSTPA6	Data transfer controller (DTC)
	MSTPA5	16-bit timer pulse unit (TPU)
	MSTPA3	Programmable pulse generator (PPG)
	MSTPA2	D/A converter (channel 0, 1)
	MSTPA1	A/D converter
	MSTPA0*1	
MSTPCRB	MSTPB7	Serial communication interface 0 (SCI0)
	MSTPB6	Serial communication interface 1 (SCI1)
	MSTPB5	Serial communication interface 2 (SCI2)
	MSTPB4	I^2C bus interface 0 (IIC0)*2
	MSTPB3	I^2C bus interface 1 (IIC1)*2
	MSTPB0*1	
MSTPCRC	MSTPC4	PC break controller (PBC)
	MSTPC3	HCAN0
	MSTPC2	HCAN1
	MSTPC1*1	
	MSTPC0*1	
MSTPCRD	MSTPD7	Motor control PWM (PWM)
	MSTPD6*1	

Notes: *1 MSTPA0, MSTPB0 and MSTPC1 to MSTPC0 and MSTPD6 are readable/writable bits with an initial value of 1.

*2 The I^2C bus interface is available as an option in the H8S/2638 and H8S/2639. When this optional feature is not used or in H8S/2636, MSTB4 and MSTB3 are readable and writable bits that have 1 as their initial value.

23B.5.2 Usage Notes

DTC Module Stop: Depending on the operating status of the DTC, the MSTPA7 and MSTPA6 bits may not be set to 1. Setting of the DTC module stop mode should be carried out only when the respective module is not activated.

For details, refer to section 8, Data Transfer Controller (DTC).

On-Chip Supporting Module Interrupt: Relevant interrupt operations cannot be performed in module stop mode. Consequently, if module stop mode is entered when an interrupt has been requested, it will not be possible to clear the CPU interrupt source or the DTC activation source. Interrupts should therefore be disabled before entering module stop mode.

Writing to MSTPCR: MSTPCR should only be written to by the CPU.

23B.6 Software Standby Mode

23B.6.1 Software Standby Mode

A transition is made to software standby mode when the SLEEP instruction is executed when the SBYCR SSBY bit = 1 and the LPWRCR LSON bit = 0, and the TCSR (WDT1) PSS bit = 0. In this mode, the CPU, on-chip supporting modules, and oscillator all stop. However, the contents of the CPU's internal registers, RAM data, and the states of on-chip supporting modules other than the SCI, A/D converter, Motor control PWM, HCAN and I/O ports, are retained. Whether the address bus and bus control signals are placed in the high-impedance state.

In this mode the oscillator stops, and therefore power dissipation is significantly reduced.

23B.6.2 Clearing Software Standby Mode

Software standby mode is cleared by an external interrupt (NMI pin, or pins $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$), or by means of the $\overline{\text{RES}}$ pin or $\overline{\text{STBY}}$ pin.

- Clearing with an interrupt

  When an NMI or IRQ0 to IRQ5 interrupt request signal is input, clock oscillation starts, and after the elapse of the time set in bits STS2 to STS0 in SYSCR, stable clocks are supplied to the entire chip, software standby mode is cleared, and interrupt exception handling is started.

  When clearing software standby mode with an IRQ0 to IRQ5 interrupt, set the corresponding enable bit to 1 and ensure that no interrupt with a higher priority than interrupts IRQ0 to IRQ5 is generated. Software standby mode cannot be cleared if the interrupt has been masked on the CPU side or has been designated as a DTC activation source.

- Clearing with the $\overline{\text{RES}}$ pin

  When the $\overline{\text{RES}}$ pin is driven Low, clock oscillation is started. At the same time as clock oscillation starts, clocks are supplied to the entire chip. Note that the $\overline{\text{RES}}$ pin must be held Low until clock oscillation stabilizes. When the $\overline{\text{RES}}$ pin goes high, the CPU begins reset exception handling.

- Clearing with the $\overline{\text{STBY}}$ pin

  When the $\overline{\text{STBY}}$ pin is driven Low, a transition is made to hardware standby mode.

23B.6.3 Setting Oscillation Stabilization Time after Clearing Software Standby Mode

Bits STS2 to STS0 in SBYCR should be set as described below.

Using a Crystal Oscillator: Set bits STS2 to STS0 so that the standby time is at least 8 ms (the oscillation stabilization time).

Table 23B-6 shows the standby times for different operating frequencies and settings of bits STS2 to STS0.

Table 23B-6 Oscillation Stabilization Time Settings

STS2	STS1	STS0	Standby Time	20 MHz	16 MHz	12 MHz	10 MHz	8 MHz	6 MHz	4 MHz	Unit
0	0	0	8192 states	0.41	0.51	0.68	0.8	1.0	1.3	2.0	ms
		1	16384 states	0.82	1.0	1.3	1.6	2.0	2.7	4.1	
	1	0	32768 states	1.6	2.0	2.7	3.3	4.1	5.5	8.2	
		1	65536 states	3.3	4.1	5.5	6.6	8.2	10.9	16.4	
1	0	0	131072 states	6.6	8.2	10.9	13.1	16.4	21.8	32.8	
		1	262144 states	13.1	16.4	21.8	26.2	32.8	43.6	65.6	
	1	0	Reserved	—	—	—	—	—	—	—	µs
		1	16 states*	0.8	1.0	1.3	1.6	2.0	2.6	4.0	

: Recommended time setting

Note: * Do not use this setting in the version with built-in flash memory.

23B.6.4 Software Standby Mode Application Example

Figure 23B-3 shows an example in which a transition is made to software standby mode at the falling edge on the NMI pin, and software standby mode is cleared at the rising edge on the NMI pin.

In this example, an NMI interrupt is accepted with the NMIEG bit in SYSCR cleared to 0 (falling edge specification), then the NMIEG bit is set to 1 (rising edge specification), the SSBY bit is set to 1, and a SLEEP instruction is executed, causing a transition to software standby mode.

Software standby mode is then cleared at the rising edge on the NMI pin.

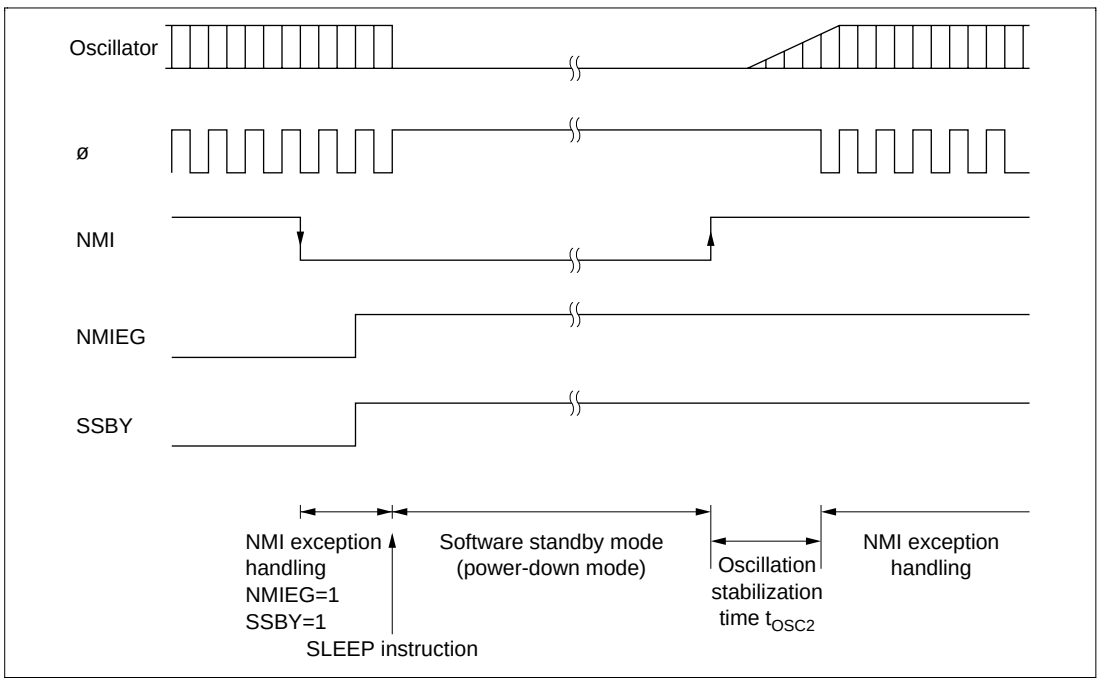


Figure 23B-3 Software Standby Mode Application Example

23B.6.5 Usage Notes

I/O Port Status: In software standby mode, I/O port states are retained. If the OPE bit is set to 1, the address bus and bus control signal output is also retained. Therefore, there is no reduction in current dissipation for the output current when a high-level signal is output.

Current Dissipation during Oscillation Stabilization Wait Period: Current dissipation increases during the oscillation stabilization wait period.

Write Data Buffer Function: The write data buffer function and software standby mode cannot be used at the same time. When the write data buffer function is used, the WDBE bit in BCRL should be cleared to 0 to cancel the write data buffer function before entering software standby mode. Also check that external writes have finished, by reading external addresses, etc., before executing a SLEEP instruction to enter software standby mode. See section 7.9, Write Data Buffer Function, for details of the write data buffer function.

23B.7 Hardware Standby Mode

23B.7.1 Hardware Standby Mode

When the $\overline{\text{STBY}}$ pin is driven low, a transition is made to hardware standby mode from any mode.

In hardware standby mode, all functions enter the reset state and stop operation, resulting in a significant reduction in power dissipation. As long as the prescribed voltage is supplied, on-chip RAM data is retained. I/O ports are set to the high-impedance state.

In order to retain on-chip RAM data, the RAME bit in SYSCR should be cleared to 0 before driving the $\overline{\text{STBY}}$ pin low.

Do not change the state of the mode pins (MD2 to MD0) while the chip is in hardware standby mode.

Hardware standby mode is cleared by means of the $\overline{\text{STBY}}$ pin and the $\overline{\text{RES}}$ pin. When the $\overline{\text{STBY}}$ pin is driven high while the $\overline{\text{RES}}$ pin is low, the reset state is set and clock oscillation is started. Ensure that the $\overline{\text{RES}}$ pin is held low until the clock oscillator stabilizes (at least 8 ms—the oscillation stabilization time—when using a crystal oscillator). When the $\overline{\text{RES}}$ pin is subsequently driven high, a transition is made to the program execution state via the reset exception handling state.

23B.7.2 Hardware Standby Mode Timing

Figure 23B-4 shows an example of hardware standby mode timing.

When the $\overline{\text{STBY}}$ pin is driven low after the $\overline{\text{RES}}$ pin has been driven low, a transition is made to hardware standby mode. Hardware standby mode is cleared by driving the $\overline{\text{STBY}}$ pin high, waiting for the oscillation stabilization time, then changing the $\overline{\text{RES}}$ pin from low to high.

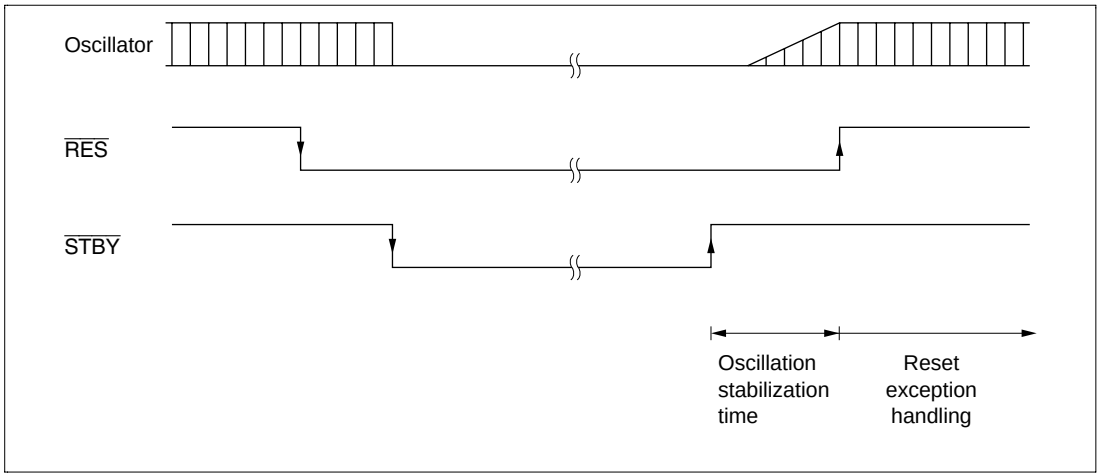


Figure 23B-4 Hardware Standby Mode Timing

23B.8 Watch Mode (U-Mask, W-Mask Version Only)

23B.8.1 Watch Mode

CPU operation makes a transition to watch mode when the SLEEP instruction is executed in high-speed mode or sub-active mode with SBYCR SSBY=1, LPWRCR DTON = 0, and TCSR (WDT1) PSS = 1.

In watch mode, the CPU is stopped and supporting modules other than WDT1 are also stopped. The contents of the CPUís internal registers, the data in internal RAM, and the statuses of the internal supporting modules (excluding the SCI, ADC, HCAN, and Motor control PWM) and I/O ports are retained.

23B.8.2 Exiting Watch Mode

Watch mode is exited by any interrupt (WOVI interrupt, NMI pin, or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$), or signals at the $\overline{\text{RES}}$, or $\overline{\text{STBY}}$ pins.

(1) Exiting Watch Mode by Interrupts

When an interrupt occurs, watch mode is exited and a transition is made to high-speed mode or medium-speed mode when the LPWRCR LSON bit = 0 or to sub-active mode when the LSON bit = 1. When a transition is made to high-speed mode, a stable clock is supplied to all LSI circuits and interrupt exception processing starts after the time set in SBYCR STS2 to STS0 has elapsed. In the case of IRQ0 to IRQ5 interrupts, no transition is made from watch mode if the corresponding enable bit has been cleared to 0, and, in the case of interrupts from the internal supporting modules, the interrupt enable register has been set to disable the reception of that interrupt, or is masked by the CPU.

See Section 23B.6.3, “Setting Oscillation Stabilization Time after clearing Software Standby Mode” for how to set the oscillation stabilization time when making a transition from watch mode to high-speed mode.

(2) Exiting Watch Mode by $\overline{\text{RES}}$ pins

For exiting watch mode by the $\overline{\text{RES}}$ pins, see, Clearing with the $\overline{\text{RES}}$ pins in Section 23B.6.2, Clearing Software Standby Mode.

(3) Exiting Watch Mode by $\overline{\text{STBY}}$ pin

When the $\overline{\text{STBY}}$ pin level is driven Low, a transition is made to hardware standby mode.

23B.8.3 Notes

(1) I/O Port Status

The status of the I/O ports is retained in watch mode. Also, when the OPE bit is set to 1, the address bus and bus control signals continue to be output. Therefore, when a High level is output, the current consumption is not diminished by the amount of current to support the High level output.

(2) Current Consumption when Waiting for Oscillation Stabilization

The current consumption increases during stabilization of oscillation.

23B.9 Sub-Sleep Mode (U-Mask, W-Mask Version Only)

23B.9.1 Sub-Sleep Mode

When the SLEEP instruction is executed with the SBYCR SSBY bit = 0, LPWRCR LSON bit = 1, and TCSR (WDT1) PSS bit = 1, CPU operation shifts to sub-sleep mode.

In sub-sleep mode, the CPU is stopped. Supporting modules other than WDT0, and WDT1 are also stopped. The contents of the CPUís internal registers, the data in internal RAM, and the statuses of the internal supporting modules (excluding the SCI, ADC, HCAN, and Motor control PWM) and I/O ports are retained.

23B.9.2 Exiting Sub-Sleep Mode

Sub-sleep mode is exited by an interrupt (interrupts from internal supporting modules, NMI pin, or $\overline{\text{IRQ0}}$ to $\overline{\text{IRQ5}}$), or signals at the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pins.

(1) Exiting Sub-Sleep Mode by Interrupts

When an interrupt occurs, sub-sleep mode is exited and interrupt exception processing starts.

In the case of IRQ0 to IRQ5 interrupts, sub-sleep mode is not cancelled if the corresponding enable bit has been cleared to 0, and, in the case of interrupts from the internal supporting modules, the interrupt enable register has been set to disable the reception of that interrupt, or is masked by the CPU.

(2) Exiting Sub-Sleep Mode by $\overline{\text{RES}}$

For exiting sub-sleep mode by the $\overline{\text{RES}}$ pins, see, Clearing with the $\overline{\text{RES}}$ pins in Section 23B.6.2, Clearing Software Standby Mode.

(3) Exiting Sub-Sleep Mode by $\overline{\text{STBY}}$ Pin

When the $\overline{\text{STBY}}$ pin level is driven Low, a transition is made to hardware standby mode.

23B.10 Sub-Active Mode (U-Mask, W-Mask Version Only)

23B.10.1 Sub-Active Mode

When the SLEEP instruction is executed in high-speed mode with the SBYCR SSBY bit = 1, LPWRCR DTON bit = 1, LSON bit = 1, and TCSR (WDT1) PSS bit = 1, CPU operation shifts to sub-active mode. When an interrupt occurs in watch mode, and if the LSON bit of LPWRCR is 1, a transition is made to sub-active mode. And if an interrupt occurs in sub-sleep mode, a transition is made to sub-active mode.

In sub-active mode, the CPU operates at low speed on the subclock, and the program is executed step by step. Supporting modules other than WDT0, and WDT1 are also stopped.

When operating the CPU in sub-active mode, the SCKCR SCK2 to SCK0 bits must be set to 0.

23B.10.2 Exiting Sub-Active Mode

Sub-active mode is exited by the SLEEP instruction or the $\overline{\text{RES}}$ or $\overline{\text{STBY}}$ pins.

(1) Exiting Sub-Active Mode by SLEEP Instruction

When the SLEEP instruction is executed with the SBYCR SSBY bit = 1, LPWRCR DTON bit = 0, and TCSR (WDT1) PSS bit = 1, the CPU exits sub-active mode and a transition is made to watch mode. When the SLEEP instruction is executed with the SBYCR SSBY bit = 0, LPWRCR LSON bit = 1, and TCSR (WDT1) PSS bit = 1, a transition is made to sub-sleep mode. Finally, when the SLEEP instruction is executed with the SBYCR SSBY bit = 1, LPWRCR DTON bit = 1, LSON bit = 0, and TCSR (WDT1) PSS bit = 1, a direct transition is made to high-speed mode (SCK0 to SCK2 all 0).

See section 23B.11, Direct Transitions for details of direct transitions.

(2) Exiting Sub-Active Mode by $\overline{\text{RES}}$ Pins

For exiting sub-active mode by the $\overline{\text{RES}}$ pins, see, Claering with the $\overline{\text{RES}}$ pins in section 23B.6.2, Clearing Software Standby Mode.

(3) Exiting Sub-Active Mode by $\overline{\text{STBY}}$ Pin

When the $\overline{\text{STBY}}$ pin level is driven Low, a transition is made to hardware standby mode.

23B.11 Direct Transitions (U-Mask, W-Mask Version Only)

23B.11.1 Overview of Direct Transitions

There are three modes, high-speed, medium-speed, and sub-active, in which the CPU executes programs. When a direct transition is made, there is no interruption of program execution when shifting between high-speed and sub-active modes. Direct transitions are enabled by setting the LPWRCR DTON bit to 1, then executing the SLEEP instruction. After a transition, direct transition interrupt exception processing starts.

(1) Direct Transitions from High-Speed Mode to Sub-Active Mode

Execute the SLEEP instruction in high-speed mode when the SBYCR SSBY bit = 1, LPWRCR LSON bit = 1, and DTON bit = 1, and TSCR (WDT1) PSS bit = 1 to make a transition to sub-active mode.

(2) Direct Transitions from Sub-Active Mode to High-Speed Mode

Execute the SLEEP instruction in sub-active mode when the SBYCR SSBY bit = 1, LPWRCR LSON bit = 0, and DTON bit = 1, and TSCR (WDT1) PSS bit = 1 to make a direct transition to high-speed mode after the time set in SBYCR STS2 to STS0 has elapsed.

23B.12 ø Clock Output Disabling Function

Output of the ø clock can be controlled by means of the PSTOP bit in SCKCR, and DDR for the corresponding port. When the PSTOP bit is set to 1, the ø clock stops at the end of the bus cycle, and ø output goes high. ø clock output is enabled when the PSTOP bit is cleared to 0. When DDR for the corresponding port is cleared to 0, ø clock output is disabled and input port mode is set. Table 23B-7 shows the state of the ø pin in each processing state.

Table 23B-7 ø Pin State in Each Processing State

DDR	0	1	1
PSTOP	—	0	1
Hardware standby mode	High impedance	High impedance	High impedance
Software standby mode, watch mode*, and direct transition	High impedance	Fixed high	Fixed high
Sleep mode and subsleep mode*	High impedance	ø output	Fixed high
High-speed mode, medium-speed mode, and subactive mode*	High impedance	ø output	Fixed high
Subactive mode	High impedance	$ø_{SUB}$ output	Fixed high

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

23B.13 Usage Notes

1. When making a transition to sub-active mode or watch mode, set the DTC to enter module stop mode (write 1 to the relevant bits in MSTPCR), and then read the relevant bits to confirm that they are set to 1 before mode transition. Do not clear module stop mode (write 0 to the relevant bits in MSTPCR) until a transition from sub-active mode to high-speed mode or medium-speed mode has been performed.

   If a DTC activation source occurs in sub-active mode, the DTC will be activated only after module stop mode has been cleared and high-speed mode or medium-speed mode has been entered.
2. The on-chip peripheral modules (DTC and TPU) which halt operation in sub-active mode cannot clear an interrupt in sub-active mode. Therefore, if a transition is made to sub-active mode while an interrupt is requested, the CPU interrupt source cannot be cleared. Disable the interrupts of each on-chip peripheral module before executing a SLEEP instruction to enter sub-active mode or watch mode.

24.1 H8S/2636 Series Electrical Characteristics

24.1.1 Absolute Maximum Ratings

Table 24-1 lists the absolute maximum ratings.

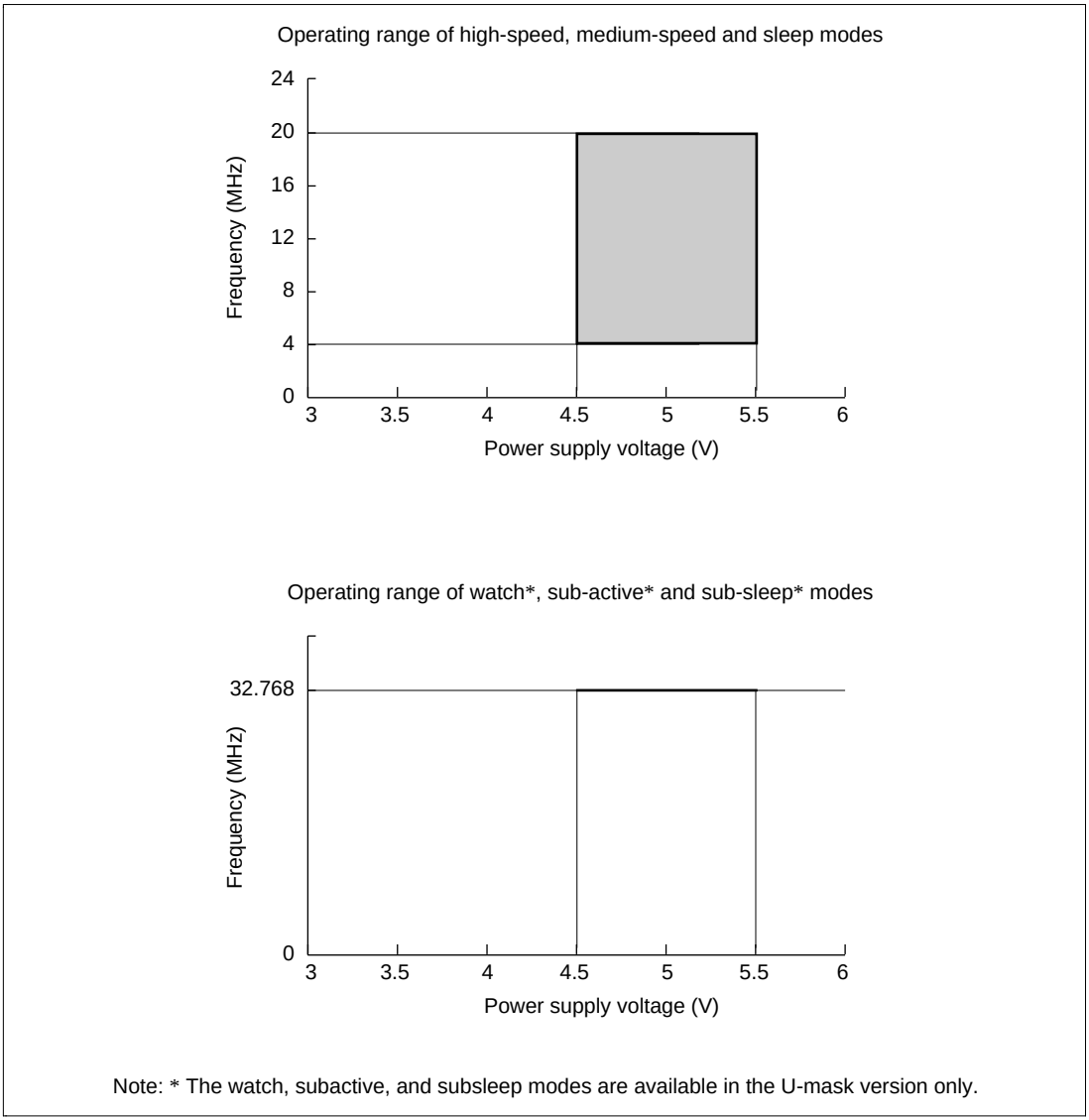
Table 24-1 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Input voltage (OSC1, OSC2)	V_{in}	–0.3 +4.3	V
Input voltage (XTAL, EXTAL)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Input voltage (port H and J)	V_{in}	–0.3 to $PWMV_{CC}$ +0.3	V
Input voltage (except XTAL, EXTAL, OSC1, OSC2, port 4, 9, H and J)	V_{in}	–0.3 to V_{CC} +0.3	V
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75	°C
		Wide-range specifications: –40 to +85	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum rating are exceeded.

24.1.2 Power Supply Voltage and Operating Frequency Range

Power supply voltage and operating frequency ranges (shaded areas) are shown in figure 24-1.



Note: * The watch, subactive, and subsleep modes are available in the U-mask version only.

Figure 24-1 Power Supply Voltage and Operating Ranges

24.1.3 DC Characteristics

Table 24-2 lists the DC characteristics. Table 24-3 lists the permissible output currents.

Table 24-2 DC Characteristics

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*1 *6

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	IRQ0 to IRQ5	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$		
		$V_T^+ - V_T^-$	0.4	—	—		
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, MD2 to MD0	V_{IH}	$V_{CC} - 0.7$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	Port 1, 3, F		2.2	—	$V_{CC} + 0.3$		
	Port A to E		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$		
	Port H, J		$PWMV_{CC} \times 0.8$	—	$PWMV_{CC} + 0.3$		
	HRxD0, HRxD1		2.2	—	$V_{CC} + 0.3$		
	Port 4 and 9		$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$		
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, MD2 to MD0	V_{IL}	–0.3	—	0.5	V	
	EXTAL		–0.3	—	0.8		
	Port 1, 3, F		–0.3	—	0.8		
	Port A to E		–0.3	—	$V_{CC} \times 0.2$		
	Port H, J		–0.3	—	$PWMV_{CC} \times 0.2$		
	HRxD0, HRxD1		–0.3	—	$V_{CC} \times 0.2$		
	Port 4, 9		–0.3	—	$AV_{CC} \times 0.2$		

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Output high voltage	Port1, 3, A to F, H,J HTxD0, HTxD1	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
			3.5	—	—		I_{OH} = –1 mA
	PWM1A to PWM1H, PWM2A to PWM2H		$PWMV_{CC}$ – 0.5		—		I_{OH} = –15 mA
Output low voltage	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	PWM1A to PWM1H, PWM2A to PWM2H		—	—	0.5	V	I_{OL} = 15 mA
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5
	$\overline{STBY}$, $\overline{NMI}$, MD2 to MD0		—	—	1.0		
	HRxD0, HRxD1, FWE		—	—	1.0		
	Port 4, 9		—	—	1.0		V_{in} = 0.5 to AV_{CC} – 0.5
Three-state leakage current (off state)	Port 1, 3, A to F, H, J HTxD0, HTxD1	$\lvert I_{TSI} \rvert$	—		1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5
MOS input pull-up current	Port A to E	$-I_P$	50		300	µA	V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
	NMI		—	—	30		
	All input pins except $\overline{RES}$ and NMI		—	—	15		

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Current dissipation*2	Normal operation	I_{CC}*4	—	75	90	mA	f = 20 MHz
	Sleep mode		—	65	80	mA	f = 20 MHz
	All modules stopped		—	57	—	mA	f = 20 MHz, (reference values)
	Medium-speed mode (ø/32)		—	49	—	mA	f = 20 MHz, (reference values)
	Subactive mode*5		—	130	220	µA	Using 32.768 kHz crystal resonator
	Subsleep mode*5		—	80	160	µA	Using 32.768 kHz crystal resonator
	Watch mode*5		—	30	60	µA	Using 32.768 kHz crystal resonator
	Standby mode*3		—	2.0	5.0	µA	$T_a \leq 50°C$
			—	—	20		$50°C < T_a$
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	1.0	2.0	mA	AV_{CC} = 5.0 V
	Idle		—	0.1	5.0	µA	
Reference current	During A/D and D/A conversion	AI_{CC}	—	4.0	5.0	mA	AV_{ref} = 5.0 V
	Idle		—	0.1	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: *1 If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref} , and AV_{SS} pins open. Apply a voltage between 4.5 V and 5.5 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$.

*2 Current dissipation values are for V_{IH} min = V_{CC} – 0.5 V, V_{IL} max = 0.5 V with all output pins unloaded and the on-chip pull-up resistors in the off state.

*3 The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.

*4 I_{CC} depends on V_{CC} and f as follows:

I_{CC} max = 30 (mA) + 0.6 (mA/(MHz × V)) × V_{CC} × f (normal operation)

I_{CC} max = 30 (mA) + 0.45 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

*5 The watch, subactive, and subsleep modes are available in the U-mask version only.

See section 22A.7 Subclock Oscillator, for the method of fixing pins OSC1 and OSC2.

*6 If the motor-control PWM timer is not used, do not leave the $PMWV_{CC}$, or $PMWV_{SS}$ pins open. If the motor-control PWM timer is not used, apply a voltage of between 4.5 and 5.5 V to the $PWMV_{CC}$ pin, for instance, by connecting it to V_{CC}.

Table 24-3 Permissible Output Currents

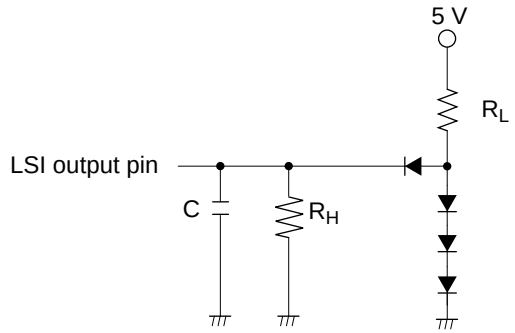
Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test condition
Permissible output low current (per pin)	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	I_{OL}	—	—	10	mA	
	PWM1A to PWM1H, PWM2A to PWM2H	I_{OL}	—	—	25	mA	T_a = 85°C
			—	—	30	mA	T_a = 25°C
			—	—	40	mA	T_a = –40°C
Permissible output low current (total)	Total of all output pins excepting PWM1A to PWM1H, and PWM2A to PWM2H	$\sum I_{OL}$	—	—	80	mA	
	Total of PWM1A to PWM1H, and PWM2A to PWM2H	$\sum I_{OL}$	—	—	150	mA	T_a = 85°C
			—	—	180	mA	T_a = 25°C
			—	—	220	mA	T_a = –40°C
Permissible output high current (per pin)	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	$-I_{OH}$	—	—	2.0	mA	
	PWM1A to PWM1H, PWM2A to PWM2H	$-I_{OH}$	—	—	25	mA	T_a = 85°C
			—	—	30	mA	T_a = 25°C
			—	—	40	mA	T_a = –40°C
Permissible output high current (total)	Total of all output pins excepting PWM1A to PWM1H, and PWM2A to PWM2H	$\sum -I_{OH}$	—	—	40	mA	
	Total of PWM1A to PWM1H, and PWM2A to PWM2H	$\sum -I_{OH}$	—	—	150	mA	T_a = 85°C
			—	—	180	mA	T_a = 25°C
			—	—	220	mA	T_a = –40°C

Note: To protect chip reliability, do not exceed the output current values in table 24-3.

24.1.4 AC Characteristics

Figure 24-2 show, the test conditions for the AC characteristics.



C = 50 pF: Ports 10 to 13, A to F
(In case of expansion bus control signal output pin setting)
C = 30 pF: All ports
R_L = 2.4 kΩ
R_H = 12 kΩ
Input/output timing measurement levels
· Low level : 0.8 V
· High level : 2.0 V

Figure 24-2 Output Load Circuit

(1) Clock Timing

Table 24-4 lists the clock timing

Table 24-4 Clock Timing

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition			
		20MHz			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
Clock cycle time	t_{cyc}	50	250	ns	Figure 24-7
Clock high pulse width	t_{CH}	15	—	ns	
Clock low pulse width	t_{CL}	15	—	ns	
Clock rise time	t_{Cr}	—	10	ns	
Clock fall time	t_{Cf}	—	10	ns	
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 24-8
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	ms	Figure 23A-3 Figure 23B-3
External clock output stabilization delay time	t_{DEXT}	2	—	ms	Figure 24-8
32 kHz clock oscillation settling time	t_{OSC3}	—	2	s	
Sub clock oscillator frequency	f_{SUB}	32.768		kHz	
Sub clock ($ø_{SUB}$) cycle time	f_{SUB}	30.5		µs	

(2) Control Signal Timing

Table 24-5 lists the control signal timing.

Table 24-5 Control Signal Timing

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 24-9
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 24-10
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

(3) Bus Timing

Table 24-6 lists the bus timing.

Table 24-6 Bus Timing

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
Address delay time	t_{AD}	—	35	ns	Figure 24-11 to
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 20$	—	ns	Figure 24-15
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 15$	—	ns	
$\overline{AS}$ delay time	t_{ASD}	—	20	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	20	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	20	ns	
Read data setup time	t_{RDS}	20	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time1	t_{ACC1}	—	$1.0 \times t_{cyc} - 48$	ns	
Read data access time2	t_{ACC2}	—	$1.5 \times t_{cyc} - 45$	ns	
Read data access time3	t_{ACC3}	—	$2.0 \times t_{cyc} - 45$	ns	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 45$	ns	
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 50$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	20	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	20	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 20$	—	ns	
Write data delay time	t_{WDD}	—	30	ns	
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 20$	—	ns	
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 10$	—	ns	

(4) Timing of On-Chip Supporting Modules

Table 24-7 lists the timing of on-chip supporting modules.

Table 24-7 Timing of On-Chip Supporting Modules

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

				Condition			
Item			**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
I/O port	Output data delay time		t_{PWD}	—	50	ns	Figure 24-16 Figure 24-17
	Output data delay time 2		t_{PWD2}	—	50		
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
PPG	Pulse output delay time		t_{POD}	—	50	ns	Figure 24-18
TPU	Timer output delay time		t_{TOCD}	—	50	ns	Figure 24-19
	Timer input setup time		t_{TICS}	30	—		
	Timer clock input setup time		t_{TCKS}	30	—	ns	Figure 24-20
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
PWM	Pulse output delay time		t_{MPWMOD}	—	50	ns	Figure 24-21

Item			Symbol	Condition Min	Max	Unit	Test Conditions
SCI	Input clock cycle	Asynchro-nous	t_{Scyc}	4	—	t_{cyc}	Figure 24-22
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time		t_{TXD}	—	50	ns	Figure 24-23
	Receive data setup time (synchronous)		t_{RXS}	50	—		
	Receive data hold time (synchronous)		t_{RXH}	50	—		
A/D converter	Trigger input setup time		t_{TRGS}	50	—	ns	Figure 24-24
HCAN*	Transmit data delay time		t_{HTXD}	—	100	ns	Figure 24-25
	Receive data setup time		t_{HRXS}	100	—		
	Receive data hold time		t_{HRXH}	100	—		

Note: The HCAN input signal is asynchronous. However, its state is judged to have changed at the leading edge (two clock cycles) of the CK clock signal shown in figure 24-25. The HCAN output signal is also asynchronous. Its state changes based on the leading edge (two clock cycles) of the CK clock signal shown in figure 24-25.

24.1.5 A/D Conversion Characteristics

Table 24-8 lists the A/D conversion characteristics.

Table 24-8 A/D Conversion Characteristics

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition			
Item	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	bits
Conversion time	10	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.5	LSB
Offset error	—	—	±3.5	LSB
Full-scale error	—	—	±3.5	LSB
Quantization	—	±0.5	—	LSB
Absolute accuracy	—	—	±4.0	LSB

24.1.6 D/A Conversion Characteristics

Table 24-9 shows the D/A conversion characteristics.

Table 24-9 D/A Conversion Characteristics

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition				
Item	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
Resolution	8	8	8	bits	
Conversion time	—	—	10	µs	20-pF capacitive load
Absolute accuracy	—	± 1.5	± 2.0	LSB	2-MΩ resistive load
	—	—	± 1.5	LSB	4-MΩ resistive load

24.1.7 Flash Memory Characteristics

Table 24-10 shows the flash memory characteristics.

Table 24-10 Flash Memory Characteristics

Conditions: V_{CC} =4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$, AV_{SS} = 0 V
T_a = 0 to +75°C (Programming/erasing operating temperature range: regular specification)

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Programming time*1,*2,*4		t_P	—	10	200	ms/ 128 bytes	
Erase time*1,*3,*5		t_E	—	100	1200	ms/block	
Reprogramming count		N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after PSU bit setting*1	t_{spsu}	50	50	—	µs	
	Wait time after P bit setting*1, *4	t_{sp30}	28	30	32	µs	Programming time wait
		t_{sp200}	198	200	202	µs	Programming time wait
		t_{sp10}	8	10	12	µs	Additional-programming time wait
	Wait time after P bit clear*1	t_{cp}	5	5	—	µs	
	Wait time after PSU bit clear*1	t_{cpsu}	5	5	—	µs	
	Wait time after PV bit setting*1	t_{spv}	4	4	—	µs	
	Wait time after H'FF dummy write*1	t_{spvr}	2	2	—	µs	
	Wait time after PV bit clear*1	t_{cpv}	2	2	—	µs	
	Wait time after SWE bit clear*1	t_{cswe}	100	100	—	µs	
	Maximum programming count*1,*4	N	—	—	1000	Times	
Erase	Wait time after SWE bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after ESU bit setting*1	t_{sesu}	100	100	—	µs	
	Wait time after E bit setting*1,*5	t_{se}	10	10	100	ms	Erase time wait
	Wait time after E bit clear*1	t_{ce}	10	10	—	µs	
	Wait time after ESU bit clear*1	t_{cesu}	10	10	—	µs	
	Wait time after EV bit setting*1	t_{sev}	20	20	—	µs	

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Erase	Wait time after H'FF dummy write*[1]	t_{sevr}	2	2	—	µs	
	Wait time after EV bit clear*[1]	t_{cev}	4	4	—	µs	
	Wait time after SWE bit clear*[1]	t_{cswe}	100	100	—	µs	
	Maximum erase count*[1,*5]	N	12	—	120	Times	

Notes: *1 Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.

*2 Programming time per 128 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR1) is set. It does not include the programming verification time.)

*3 Block erase time (Shows the total period for which the E-bit in FLMCR1 is set. It does not include the erase verification time.)

*4 To specify the maximum programming time value (t_P(max)) in the 128-byte programming algorithm, set the max. value (1000) for the maximum programming count (N).

The wait time after P bit setting should be changed as follows according to the value of the programming counter (n).

Programming counter (n) = 1 to 6: t_{sp30} = 30 µs

Programming counter (n) = 7 to 1000: t_{sp200} = 200 µs

[In additional programming]

Programming counter (n)= 1 to 6: t_{sp10} = 10 µs

*5 For the maximum erase time (t_E(max)), the following relationship applies between the wait time after E bit setting (t_{se}) and the maximum erase count (N):

t_E(max) = Wait time after E bit setting (t_{se}) x maximum erase count (N)

To set the maximum erase time, the values of (t_{se}) and (N) should be set so as to satisfy the above formula.

Examples: When t_{se} = 100 [ms], N = 12 times

When t_{se} = 10 [ms], N = 120 times

24.2 H8S/2638 Series Electrical Characteristics

24.2.1 Absolute Maximum Ratings

Table 24-11 lists the absolute maximum ratings.

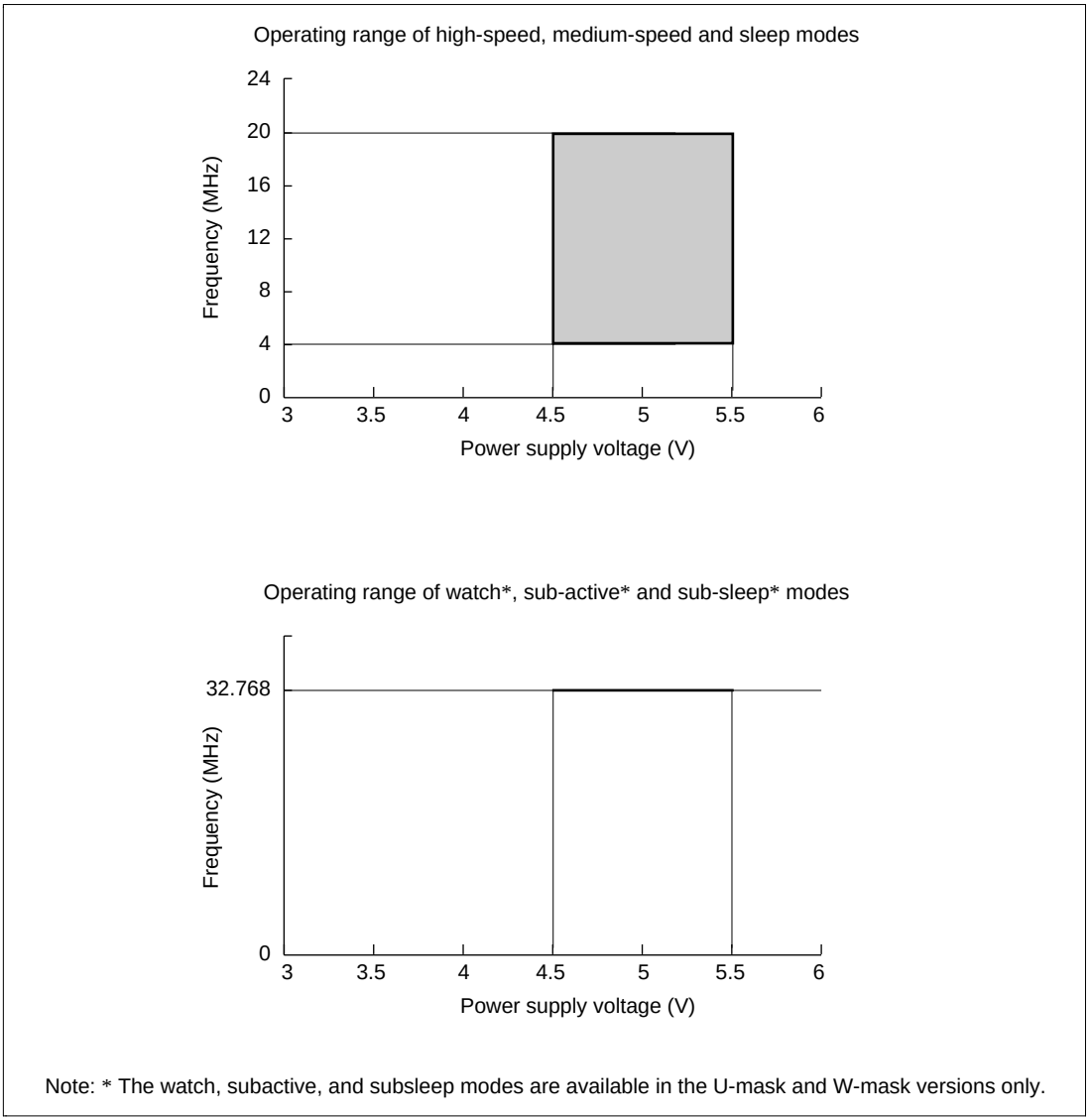
Table 24-11 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Input voltage (OSC1, OSC2)	V_{in}	–0.3 +4.3	V
Input voltage (XTAL, EXTAL)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Input voltage (port H and J)	V_{in}	–0.3 to $PWMV_{CC}$ +0.3	V
Input voltage (except XTAL, EXTAL, OSC1, OSC2, port 4, 9, H and J)	V_{in}	–0.3 to V_{CC} +0.3	V
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75	°C
		Wide-range specifications: –40 to +85	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum rating are exceeded.

24.2.2 Power Supply Voltage and Operating Frequency Range

Power supply voltage and operating frequency ranges (shaded areas) are shown in figure 24-3.



Note: * The watch, subactive, and subsleep modes are available in the U-mask and W-mask versions only.

Figure 24-3 Power Supply Voltage and Operating Ranges+

24.2.3 DC Characteristics

Table 24-12 lists the DC characteristics. Table 24-13 lists the permissible output currents.

Table 24-12 DC Characteristics

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*1 *6

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	IRQ0 to IRQ5	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$		
		$V_T^+ - V_T^-$	0.4	—	—		
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, MD2 to MD0	V_{IH}	$V_{CC} - 0.7$	—	$V_{CC} + 0.3$	V	
	EXTAL		$V_{CC} \times 0.7$	—	$V_{CC} + 0.3$		
	Port 1, 3, F		2.2	—	$V_{CC} + 0.3$		
	Port A to E		$V_{CC} \times 0.8$	—	$V_{CC} + 0.3$		
	Port H, J		$PWMV_{CC} \times 0.8$	—	$PWMV_{CC} + 0.3$		
	HRxD0, HRxD1		2.2	—	$V_{CC} + 0.3$		
	Port 4 and 9		$AV_{CC} \times 0.7$	—	$AV_{CC} + 0.3$		
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, MD2 to MD0	V_{IL}	–0.3	—	0.5	V	
	EXTAL		–0.3	—	0.8		
	Port 1, 3, F		–0.3	—	0.8		
	Port A to E		–0.3	—	$V_{CC} \times 0.2$		
	Port H, J		–0.3	—	$PWMV_{CC} \times 0.2$		
	HRxD0, HRxD1		–0.3	—	$V_{CC} \times 0.2$		
	Port 4, 9		–0.3	—	$AV_{CC} \times 0.2$		

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Output high voltage	Port1, 3, A to F, H,J HTxD0, HTxD1 (excluding P34 and P35)	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
	P34, P35		V_{CC} – 2.5	—	—		I_{OH} = –100 µA
	Port 1, 3, A to F, H, J HTxD0, HTxD1 (excluding P34 and P35)		3.5	—	—		I_{OH} = –1 mA
	PWM1A to PWM1H, PWM2A to PWM2H		PWMV_{CC} – 0.5		—		I_{OH} = –15 mA
Output low voltage	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	PWM1A to PWM1H, PWM2A to PWM2H		—	—	0.5	V	I_{OL} = 15 mA
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5
	$\overline{STBY}$, $\overline{NMI}$, MD2 to MD0		—	—	1.0		
	HRxD0, HRxD1, FWE		—	—	1.0		
	Port 4, 9		—	—	1.0		V_{in} = 0.5 to AV_{CC} – 0.5
Three-state leakage current (off state)	Port 1, 3, A to F, H, J HTxD0, HTxD1	$\lvert I_{TSI} \rvert$	—		1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5
MOS input pull-up current	Port A to E	$-I_P$	50		300	µA	V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
	NMI		—	—	30		
	All input pins except $\overline{RES}$ and NMI		—	—	15		

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Current dissipation*2	Normal operation	I_{CC}*4	—	75	90	mA	f = 20 MHz
	Sleep mode		—	65	80	mA	f = 20 MHz
	All modules stopped		—	57	—	mA	f = 20 MHz, (reference values)
	Medium-speed mode (ø/32)		—	49	—	mA	f = 20 MHz, (reference values)
	Subactive mode*5		—	130	220	µA	Using 32.768 kHz crystal resonator
	Subsleep mode*5		—	80	160	µA	Using 32.768 kHz crystal resonator
	Watch mode*5		—	30	60	µA	Using 32.768 kHz crystal resonator
	Standby mode*3		—	2.0	5.0	µA	$T_a \leq 50°C$
			—	—	20		$50°C < T_a$
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	1.0	2.0	mA	AV_{CC} = 5.0 V
	Idle		—	0.1	5.0	µA	
Reference current	During A/D and D/A conversion	AI_{CC}	—	4.0	5.0	mA	AV_{ref} = 5.0 V
	Idle		—	0.1	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: *1 If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref}, and AV_{SS} pins open. Apply a voltage between 4.5 V and 5.5 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$.

*2 Current dissipation values are for V_{IH} min = V_{CC} – 0.5 V, V_{IL} max = 0.5 V with all output pins unloaded and the on-chip pull-up resistors in the off state.

*3 The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.

*4 I_{CC} depends on V_{CC} and f as follows:

I_{CC} max = 30 (mA) + 0.6 (mA/(MHz × V)) × V_{CC} × f (normal operation)

I_{CC} max = 30 (mA) + 0.45 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

*5 The watch, subactive, and subsleep modes are available in the U-mask and W-mask versions only.

See section 22A.7 Subclock Oscillator, for the method of fixing pins OSC1 and OSC2.

*6 If the motor-control PWM timer is not used, do not leave the $PMWV_{CC}$, or $PMWV_{SS}$ pins open. If the motor-control PWM timer is not used, apply a voltage of between 4.5 and 5.5 V to the $PWMV_{CC}$ pin, for instance, by connecting it to V_{CC}.

Table 24-13 Permissible Output Currents

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test condition
Permissible output low current (per pin)	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	I_{OL}	—	—	10	mA	
	PWM1A to PWM1H, PWM2A to PWM2H	I_{OL}	—	—	25	mA	T_a = 85°C
			—	—	30	mA	T_a = 25°C
			—	—	40	mA	T_a = –40°C
Permissible output low current (total)	Total of all output pins excepting PWM1A to PWM1H, and PWM2A to PWM2H	$\sum I_{OL}$	—	—	80	mA	
	Total of PWM1A to PWM1H, and PWM2A to PWM2H	$\sum I_{OL}$	—	—	150	mA	T_a = 85°C
			—	—	180	mA	T_a = 25°C
			—	—	220	mA	T_a = –40°C
Permissible output high current (per pin)	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	$-I_{OH}$	—	—	2.0	mA	
	PWM1A to PWM1H, PWM2A to PWM2H	$-I_{OH}$	—	—	25	mA	T_a = 85°C
			—	—	30	mA	T_a = 25°C
			—	—	40	mA	T_a = –40°C
Permissible output high current (total)	Total of all output pins excepting PWM1A to PWM1H, and PWM2A to PWM2H	$-\sum I_{OH}$	—	—	40	mA	
	Total of PWM1A to PWM1H, and PWM2A to PWM2H	$-\sum I_{OH}$	—	—	150	mA	T_a = 85°C
			—	—	180	mA	T_a = 25°C
			—	—	220	mA	T_a = –40°C

Note: To protect chip reliability, do not exceed the output current values in table 24-13.

Table 24-14 Bus Drive Characteristics

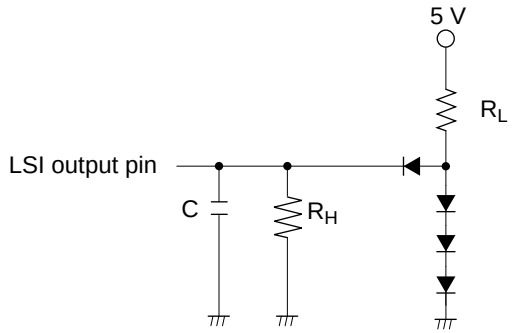
Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Applicable Pins: SCL1-0, SDA1-0

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	V_T^-	1.0	—	—	V	
	V_T^+	—	—	$V_{CC} \times 0.7$		
	$V_T^+ - V_T^-$	0.4	—	—		V_{CC} = 4.5 V to 5.5 V
Input high voltage	V_{IH}	$– V_{CC} \times 0.7$	—	V_{CC} + 0.5	V	
Input low voltage	V_{IL}	– 0.5	—	$V_{CC} \times 0.3$	V	
Output low voltage	V_{OL}	—	—	0.7	V	I_{OL} = 8 mA, V_{CC} = 4.5 V to 5.5 V
		—	—	0.4		I_{OL} = 3 mA, V_{CC} = 4.5 V to 5.5 V
		—	—	0.4		I_{OL} = 1.6 mA, V_{CC} = 3.3 V to 5.5 V
Input capacitance	C_{in}	—	—	20	pF	V_{in} = 0 V, f = 1MHz, T_a = 25 °C
Three-state leakage current (off state)	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 5.5 V
SCL, SDA, output fall time	t_{of}	20 + 0.1Cb	—	250	ns	

24.2.4 AC Characteristics

Figure 24-4 show, the test conditions for the AC characteristics.



C = 50 pF: Ports 10 to 13, A to F
(In case of expansion bus control signal output pin setting)
C = 30 pF: All ports
R_L = 2.4 kΩ
R_H = 12 kΩ
Input/output timing measurement levels
· Low level : 0.8 V
· High level : 2.0 V

Figure 24-4 Output Load Circuit

(1) Clock Timing

Table 24-15 lists the clock timing

Table 24-15 Clock Timing

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition			
		20MHz			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
Clock cycle time	t_{cyc}	50	250	ns	Figure 24-7
Clock high pulse width	t_{CH}	15	—	ns	
Clock low pulse width	t_{CL}	15	—	ns	
Clock rise time	t_{Cr}	—	10	ns	
Clock fall time	t_{Cf}	—	10	ns	
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 24-8
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	ms	Figure 23A-3 Figure 23B-3
External clock output stabilization delay time	t_{DEXT}	2	—	ms	Figure 24-8
32 kHz clock oscillation settling time	t_{OSC3}	—	2	s	
Sub clock oscillator frequency	f_{SUB}	32.768		kHz	
Sub clock ($ø_{SUB}$) cycle time	f_{SUB}	30.5		µs	

(2) Control Signal Timing

Table 24-16 lists the control signal timing.

Table 24-16 Control Signal Timing

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 24-9
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 24-10
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

(3) Bus Timing

Table 24-17 lists the bus timing.

Table 24-17 Bus Timing

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
Address delay time	t_{AD}	—	35	ns	Figure 24-11 to
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 20$	—	ns	Figure 24-15
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 15$	—	ns	
$\overline{AS}$ delay time	t_{ASD}	—	20	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	20	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	20	ns	
Read data setup time	t_{RDS}	20	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time1	t_{ACC1}	—	$1.0 \times t_{cyc} - 48$	ns	
Read data access time2	t_{ACC2}	—	$1.5 \times t_{cyc} - 45$	ns	
Read data access time3	t_{ACC3}	—	$2.0 \times t_{cyc} - 45$	ns	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 45$	ns	
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 50$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	20	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	20	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 20$	—	ns	
Write data delay time	t_{WDD}	—	30	ns	
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 20$	—	ns	
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 10$	—	ns	

(4) Timing of On-Chip Supporting Modules

Table 24-18 lists the timing of on-chip supporting modules.

Table 24-18 Timing of On-Chip Supporting Modules

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

				Condition			
Item			**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
I/O port	Output data delay time		t_{PWD}	—	50	ns	Figure 24-16 Figure 24-17
	Output data delay time 2		t_{PWD2}	—	50		
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
PPG	Pulse output delay time		t_{POD}	—	50	ns	Figure 24-18
TPU	Timer output delay time		t_{TOCD}	—	50	ns	Figure 24-19
	Timer input setup time		t_{TICS}	30	—		
	Timer clock input setup time		t_{TCKS}	30	—	ns	Figure 24-20
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
PWM	Pulse output delay time		t_{MPWMOD}	—	50	ns	Figure 24-21

Item			Symbol	Condition Min	Max	Unit	Test Conditions
SCI	Input clock cycle	Asynchro-nous	t_{Scyc}	4	—	t_{cyc}	Figure 24-22
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time		t_{TXD}	—	50	ns	Figure 24-23
	Receive data setup time (synchronous)		t_{RXS}	50	—		
	Receive data hold time (synchronous)		t_{RXH}	50	—		
A/D converter	Trigger input setup time		t_{TRGS}	50	—	ns	Figure 24-24
HCAN*	Transmit data delay time		t_{HTXD}	—	100	ns	Figure 24-25
	Receive data setup time		t_{HRXS}	100	—		
	Receive data hold time		t_{HRXH}	100	—		

Note: The HCAN input signal is asynchronous. However, its state is judged to have changed at the leading edge (two clock cycles) of the CK clock signal shown in figure 24-25. The HCAN output signal is also asynchronous. Its state changes based on the leading edge (two clock cycles) of the CK clock signal shown in figure 24-25.

Table 24-19 I^2C Bus Timing [Option]

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, ø = 5 MHz to maximum operating frequency, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition				
Item	**Symbol**	**Min**	**Typ**	**Max**	**Unit**	**Notes**
SCL input cycle time	t_{SCL}	$12t_{cyc}$	—	—	ns	Figure 24-26
SCL input high pulse width	t_{SCLH}	$3t_{cyc}$	—	—	ns	
SCL input low pulse width	t_{SCLL}	$5t_{cyc}$	—	—	ns	
SCL, SDA input rise time	t_{Sr}	—	—	$7.5t_{cyc}$*	ns	
SCL, SDA input fall time	t_{Sf}	—	—	300	ns	
SCL, SDA input spike pulse elimination time	t_{SP}	—	—	$1t_{cyc}$	ns	
SDA input bus free time	t_{BUF}	$5t_{cyc}$	—	—	ns	
Start condition input hold time	t_{STAH}	$3t_{cyc}$	—	—	ns	
Retransmission start condition input setup time	t_{STAS}	$3t_{cyc}$	—	—	ns	
Stop condition input setup time	t_{STOS}	$3t_{cyc}$	—	—	ns	
Data input setup time	t_{SDAS}	$0.5t_{cyc}$	—	—	ns	
Data input hold time	t_{SDAH}	0	—	—	ns	
SCL, SDA capacitive load	C_b	—	—	400	pF	

Note: Available when using I^2C bus interface as an option.

* $17.5t_{cyc}$ can be set according to the clock selected for use by the I^2C module. For details, see section 15.4, Usage Notes.

24.2.5 A/D Conversion Characteristics

Table 24-20 lists the A/D conversion characteristics.

Table 24-20 A/D Conversion Characteristics

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition			
Item	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	bits
Conversion time	10	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.5	LSB
Offset error	—	—	±3.5	LSB
Full-scale error	—	—	±3.5	LSB
Quantization	—	±0.5	—	LSB
Absolute accuracy	—	—	±4.0	LSB

24.2.6 D/A Conversion Characteristics

Table 24-21 shows the D/A conversion characteristics.

Table 24-21 D/A Conversion Characteristics

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition				
Item	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
Resolution	8	8	8	bits	
Conversion time	—	—	10	µs	20-pF capacitive load
Absolute accuracy	—	± 1.5	± 2.0	LSB	2-MΩ resistive load
	—	—	± 1.5	LSB	4-MΩ resistive load

24.2.7 Flash Memory Characteristics

Table 24-22 shows the flash memory characteristics.

Table 24-22 Flash Memory Characteristics

Conditions: V_{CC} =4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$, AV_{SS} = 0 V
T_a = 0 to +75°C (Programming/erasing operating temperature range: regular specification)

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Programming time*1,*2,*4		t_P	—	10	200	ms/ 128 bytes	
Erase time*1,*3,*5		t_E	—	100	1200	ms/block	
Reprogramming count		N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after PSU bit setting*1	t_{spsu}	50	50	—	µs	
	Wait time after P bit setting*1, *4	t_{sp30}	28	30	32	µs	Programming time wait
		t_{sp200}	198	200	202	µs	Programming time wait
		t_{sp10}	8	10	12	µs	Additional-programming time wait
	Wait time after P bit clear*1	t_{cp}	5	5	—	µs	
	Wait time after PSU bit clear*1	t_{cpsu}	5	5	—	µs	
	Wait time after PV bit setting*1	t_{spv}	4	4	—	µs	
	Wait time after H'FF dummy write*1	t_{spvr}	2	2	—	µs	
	Wait time after PV bit clear*1	t_{cpv}	2	2	—	µs	
	Wait time after SWE bit clear*1	t_{cswe}	100	100	—	µs	
	Maximum programming count*1,*4	N	—	—	1000	Times	
Erase	Wait time after SWE bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after ESU bit setting*1	t_{sesu}	100	100	—	µs	
	Wait time after E bit setting*1,*5	t_{se}	10	10	100	ms	Erase time wait
	Wait time after E bit clear*1	t_{ce}	10	10	—	µs	
	Wait time after ESU bit clear*1	t_{cesu}	10	10	—	µs	
	Wait time after EV bit setting*1	t_{sev}	20	20	—	µs	

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Erase	Wait time after H'FF dummy write*1	t_{sevr}	2	2	—	µs	
	Wait time after EV bit clear*1	t_{cev}	4	4	—	µs	
	Wait time after SWE bit clear*1	t_{cswe}	100	100	—	µs	
	Maximum erase count*1,*5	N	12	—	120	Times	

Notes: *1 Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.

*2 Programming time per 128 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR1) is set. It does not include the programming verification time.)

*3 Block erase time (Shows the total period for which the E-bit in FLMCR1 is set. It does not include the erase verification time.)

*4 To specify the maximum programming time value (t_P(max)) in the 128-byte programming algorithm, set the max. value (1000) for the maximum programming count (N).

The wait time after P bit setting should be changed as follows according to the value of the programming counter (n).

Programming counter (n) = 1 to 6: t_{sp30} = 30 µs

Programming counter (n) = 7 to 1000: t_{sp200} = 200 µs

[In additional programming]

Programming counter (n)= 1 to 6: t_{sp10} = 10 µs

*5 For the maximum erase time (t_E(max)), the following relationship applies between the wait time after E bit setting (t_{se}) and the maximum erase count (N):

t_E(max) = Wait time after E bit setting (t_{se}) x maximum erase count (N)

To set the maximum erase time, the values of (t_{se}) and (N) should be set so as to satisfy the above formula.

Examples: When t_{se} = 100 [ms], N = 12 times

When t_{se} = 10 [ms], N = 120 times

24.3 H8S/2639 Series Electrical Characteristics

24.3.1 Absolute Maximum Ratings

Table 24-23 lists the absolute maximum ratings.

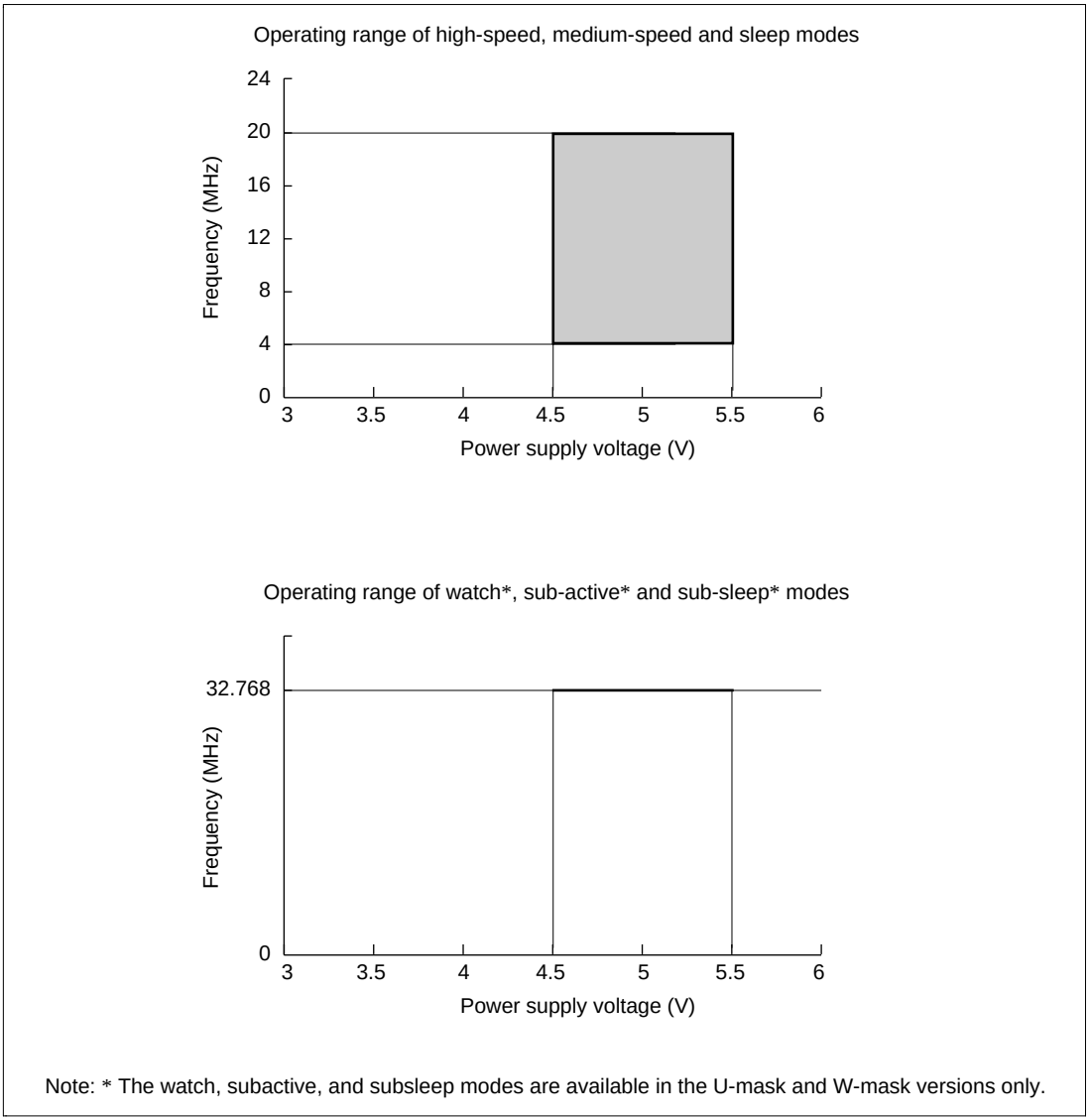
Table 24-23 Absolute Maximum Ratings

Item	Symbol	Value	Unit
Power supply voltage	V_{CC}	–0.3 to +7.0	V
Input voltage (OSC1, OSC2)	V_{in}	–0.3 +4.3	V
Input voltage (XTAL, EXTAL)	V_{in}	–0.3 to V_{CC} +0.3	V
Input voltage (port 4 and 9)	V_{in}	–0.3 to AV_{CC} +0.3	V
Input voltage (port H and J)	V_{in}	–0.3 to $PWMV_{CC}$ +0.3	V
Input voltage (except XTAL, EXTAL, OSC1, OSC2, port 4, 9, H and J)	V_{in}	–0.3 to V_{CC} +0.3	V
Reference voltage	V_{ref}	–0.3 to AV_{CC} +0.3	V
Analog power supply voltage	AV_{CC}	–0.3 to +7.0	V
Analog input voltage	V_{AN}	–0.3 to AV_{CC} +0.3	V
Operating temperature	T_{opr}	Regular specifications: –20 to +75	°C
		Wide-range specifications: –40 to +85	°C
Storage temperature	T_{stg}	–55 to +125	°C

Caution: Permanent damage to the chip may result if absolute maximum rating are exceeded.

24.3.2 Power Supply Voltage and Operating Frequency Range

Power supply voltage and operating frequency ranges (shaded areas) are shown in figure 24-5.



Note: * The watch, subactive, and subsleep modes are available in the U-mask and W-mask versions only.

Figure 24-5 Power Supply Voltage and Operating Ranges+

24.3.3 DC Characteristics

Table 24-24 lists the DC characteristics. Table 24-25 lists the permissible output currents.

Table 24-24 DC Characteristics

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)*1 *5

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	IRQ0 to IRQ5	V_T^-	1.0	—	—	V	
		V_T^+	—	—	$V_{CC} \times 0.7$		
		$V_T^+ - V_T^-$	0.4	—	—		
Input high voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, MD2 to MD0	V_{IH}	V_{CC} – 0.7	—	V_{CC} + 0.3	V	
	EXTAL		$V_{CC} \times 0.7$	—	V_{CC} + 0.3		
	Port 1, 3, F		2.2	—	V_{CC} + 0.3		
	Port A to E		$V_{CC} \times 0.8$	—	V_{CC} + 0.3		
	Port H, J		$PWMV_{CC} \times 0.8$	—	$PWMV_{CC}$ + 0.3		
	HRxD0, HRxD1		2.2	—	V_{CC} + 0.3		
	Port 4 and 9		$AV_{CC} \times 0.7$	—	AV_{CC} + 0.3		
Input low voltage	$\overline{RES}$, $\overline{STBY}$, NMI, FWE, MD2 to MD0	V_{IL}	–0.3	—	0.5	V	
	EXTAL		–0.3	—	0.8		
	Port 1, 3, F		–0.3	—	0.8		
	Port A to E		–0.3	—	$V_{CC} \times 0.2$		
	Port H, J		–0.3	—	$PWMV_{CC} \times 0.2$		
	HRxD0, HRxD1		–0.3	—	$V_{CC} \times 0.2$		
	Port 4, 9		–0.3	—	$AV_{CC} \times 0.2$		

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Output high voltage	Port1, 3, A to F, H,J HTxD0, HTxD1 (excluding P34 and P35)	V_{OH}	V_{CC} – 0.5	—	—	V	I_{OH} = –200 µA
	P34, P35		V_{CC} – 2.5	—	—		I_{OH} = –100 µA
	Port 1, 3, A to F, H, J HTxD0, HTxD1 (excluding P34 and P35)		3.5	—	—		I_{OH} = –1 mA
	PWM1A to PWM1H, PWM2A to PWM2H		PWMV_{CC} – 0.5		—		I_{OH} = –15 mA
Output low voltage	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	V_{OL}	—	—	0.4	V	I_{OL} = 1.6 mA
	PWM1A to PWM1H, PWM2A to PWM2H		—	—	0.5	V	I_{OL} = 15 mA
Input leakage current	$\overline{RES}$	$\lvert I_{in} \rvert$	—	—	1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5
	$\overline{STBY}$, $\overline{NMI}$, MD2 to MD0		—	—	1.0		
	HRxD0, HRxD1, FWE		—	—	1.0		
	Port 4, 9		—	—	1.0		V_{in} = 0.5 to AV_{CC} – 0.5
Three-state leakage current (off state)	Port 1, 3, A to F, H, J HTxD0, HTxD1	$\lvert I_{TSI} \rvert$	—		1.0	µA	V_{in} = 0.5 to V_{CC} – 0.5
MOS input pull-up current	Port A to E	$-I_P$	50		300	µA	V_{in} = 0 V
Input capacitance	$\overline{RES}$	C_{in}	—	—	30	pF	V_{in} = 0 V f = 1 MHz T_a = 25°C
	NMI		—	—	30		
	All input pins except $\overline{RES}$ and NMI		—	—	15		

Item		Symbol	Min	Typ	Max	Unit	Test Conditions
Current dissipation*2	Normal operation	I_{CC}*4	—	75	90	mA	f = 20 MHz
	Sleep mode		—	65	80	mA	f = 20 MHz
	All modules stopped		—	57	—	mA	f = 20 MHz, (reference values)
	Medium-speed mode (ø/32)		—	49	—	mA	f = 20 MHz, (reference values)
	Subactive mode		—	410	450	µA	Using 32.768 kHz crystal resonator
	Subsleep mode		—	360	400	µA	Using 32.768 kHz crystal resonator
	Watch mode		—	270	300	µA	Using 32.768 kHz crystal resonator
	Standby mode*3		—	2.0	5.0	µA	$T_a \leq 50°C$
			—	—	20		$50°C < T_a$
Analog power supply current	During A/D and D/A conversion	AI_{CC}	—	1.0	2.0	mA	AV_{CC} = 5.0 V
	Idle		—	0.1	5.0	µA	
Reference current	During A/D and D/A conversion	AI_{CC}	—	4.0	5.0	mA	AV_{ref} = 5.0 V
	Idle		—	0.1	5.0	µA	
RAM standby voltage		V_{RAM}	2.0	—	—	V	

Notes: *1 If the A/D and D/A converters are not used, do not leave the AV_{CC}, V_{ref} , and AV_{SS} pins open. Apply a voltage between 4.5 V and 5.5 V to the AV_{CC} and V_{ref} pins by connecting them to V_{CC}, for instance. Set $V_{ref} \leq AV_{CC}$.

*2 Current dissipation values are for V_{IH} min = V_{CC} – 0.5 V, V_{IL} max = 0.5 V with all output pins unloaded and the on-chip pull-up resistors in the off state.

*3 The values are for $V_{RAM} \leq V_{CC} < 3.0$ V, V_{IH} min = $V_{CC} \times 0.9$, and V_{IL} max = 0.3 V.

*4 I_{CC} depends on V_{CC} and f as follows:

I_{CC} max = 30 (mA) + 0.6 (mA/(MHz × V)) × V_{CC} × f (normal operation)

I_{CC} max = 30 (mA) + 0.45 (mA/(MHz × V)) × V_{CC} × f (sleep mode)

*5 If the motor-control PWM timer is not used, do not leave the $PMWV_{CC}$, or $PMWV_{SS}$ pins open. If the motor-control PWM timer is not used, apply a voltage of between 4.5 and 5.5 V to the $PWMV_{CC}$ pin, for instance, by connecting it to V_{CC}.

Table 24-25 Permissible Output Currents

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Item		Symbol	Min	Typ	Max	Unit	Test condition
Permissible output low current (per pin)	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	I_{OL}	—	—	10	mA	
	PWM1A to PWM1H, PWM2A to PWM2H	I_{OL}	—	—	25	mA	T_a = 85°C
			—	—	30	mA	T_a = 25°C
			—	—	40	mA	T_a = –40°C
Permissible output low current (total)	Total of all output pins excepting PWM1A to PWM1H, and PWM2A to PWM2H	$\sum I_{OL}$	—	—	80	mA	
	Total of PWM1A to PWM1H, and PWM2A to PWM2H	$\sum I_{OL}$	—	—	150	mA	T_a = 85°C
			—	—	180	mA	T_a = 25°C
			—	—	220	mA	T_a = –40°C
Permissible output high current (per pin)	All output pins except PWM1A to PWM1H, PWM2A to PWM2H	$–I_{OH}$	—	—	2.0	mA	
	PWM1A to PWM1H, PWM2A to PWM2H	$–I_{OH}$	—	—	25	mA	T_a = 85°C
			—	—	30	mA	T_a = 25°C
			—	—	40	mA	T_a = –40°C
Permissible output high current (total)	Total of all output pins excepting PWM1A to PWM1H, and PWM2A to PWM2H	$–\sum I_{OH}$	—	—	40	mA	
	Total of PWM1A to PWM1H, and PWM2A to PWM2H	$–\sum I_{OH}$	—	—	150	mA	T_a = 85°C
			—	—	180	mA	T_a = 25°C
			—	—	220	mA	T_a = –40°C

Note: To protect chip reliability, do not exceed the output current values in table 24-25.

Table 24-26 Bus Drive Characteristics

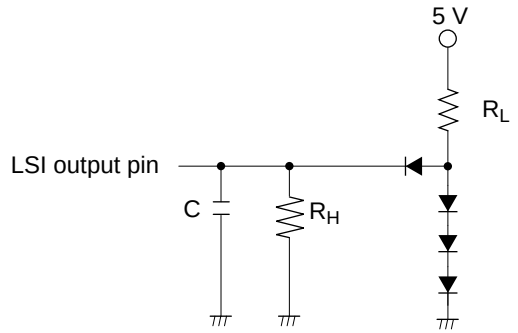
Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

Applicable Pins: SCL1-0, SDA1-0

Item	Symbol	Min	Typ	Max	Unit	Test Conditions
Schmitt trigger input voltage	V_T^-	1.0	—	—	V	
	V_T^+	—	—	$V_{CC} \times 0.7$		
	$V_T^+ - V_T^-$	0.4	—	—		V_{CC} = 4.5 V to 5.5 V
Input high voltage	V_{IH}	$– V_{CC} \times 0.7$	—	V_{CC} + 0.5	V	
Input low voltage	V_{IL}	– 0.5	—	$V_{CC} \times 0.3$	V	
Output low voltage	V_{OL}	—	—	0.7	V	I_{OL} = 8 mA, V_{CC} = 4.5 V to 5.5 V
		—	—	0.4		I_{OL} = 3 mA, V_{CC} = 4.5 V to 5.5 V
		—	—	0.4		I_{OL} = 1.6 mA, V_{CC} = 3.3 V to 5.5 V
Input capacitance	C_{in}	—	—	20	pF	V_{in} = 0 V, f = 1MHz, T_a = 25 °C
Three-state leakage current (off state)	$\lvert I_{TSI} \rvert$	—	—	1.0	µA	V_{in} = 0.5 V to V_{CC} – 5.5 V
SCL, SDA, output fall time	t_{of}	20 + 0.1Cb	—	250	ns	

24.3.4 AC Characteristics

Figure 24-6 show, the test conditions for the AC characteristics.



C = 50 pF: Ports 10 to 13, A to F
(In case of expansion bus control signal output pin setting)
C = 30 pF: All ports
R_L = 2.4 kΩ
R_H = 12 kΩ
Input/output timing measurement levels
· Low level : 0.8 V
· High level : 2.0 V

Figure 24-6 Output Load Circuit

(1) Clock Timing

Table 24-27 lists the clock timing

Table 24-27 Clock Timing

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition			
		20MHz			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
Clock cycle time	t_{cyc}	50	250	ns	Figure 24-7
Clock high pulse width	t_{CH}	15	—	ns	
Clock low pulse width	t_{CL}	15	—	ns	
Clock rise time	t_{Cr}	—	10	ns	
Clock fall time	t_{Cf}	—	10	ns	
Clock oscillator settling time at reset (crystal)	t_{OSC1}	20	—	ms	Figure 24-8
Clock oscillator settling time in software standby (crystal)	t_{OSC2}	8	—	ms	Figure 23A-3 Figure 23B-3
External clock output stabilization delay time	t_{DEXT}	2	—	ms	Figure 24-8
32 kHz clock oscillation settling time	t_{OSC3}	—	2	s	
Sub clock oscillator frequency	f_{SUB}	31.25	39.6	kHz	
Sub clock ($ø_{SUB}$) cycle time	f_{SUB}	25.6	32.0	µs	

(2) Control Signal Timing

Table 24-28 lists the control signal timing.

Table 24-28 Control Signal Timing

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
$\overline{RES}$ setup time	t_{RESS}	200	—	ns	Figure 24-9
$\overline{RES}$ pulse width	t_{RESW}	20	—	t_{cyc}	
NMI setup time	t_{NMIS}	150	—	ns	Figure 24-10
NMI hold time	t_{NMIH}	10	—		
NMI pulse width (exiting software standby mode)	t_{NMIW}	200	—	ns	
$\overline{IRQ}$ setup time	t_{IRQS}	150	—	ns	
$\overline{IRQ}$ hold time	t_{IRQH}	10	—	ns	
$\overline{IRQ}$ pulse width (exiting software standby mode)	t_{IRQW}	200	—	ns	

(3) Bus Timing

Table 24-29 lists the bus timing.

Table 24-29 Bus Timing

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition			
Item	**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
Address delay time	t_{AD}	—	35	ns	Figure 24-11 to
Address setup time	t_{AS}	$0.5 \times t_{cyc} - 20$	—	ns	Figure 24-15
Address hold time	t_{AH}	$0.5 \times t_{cyc} - 15$	—	ns	
$\overline{AS}$ delay time	t_{ASD}	—	20	ns	
$\overline{RD}$ delay time 1	t_{RSD1}	—	20	ns	
$\overline{RD}$ delay time 2	t_{RSD2}	—	20	ns	
Read data setup time	t_{RDS}	20	—	ns	
Read data hold time	t_{RDH}	0	—	ns	
Read data access time1	t_{ACC1}	—	$1.0 \times t_{cyc} - 48$	ns	
Read data access time2	t_{ACC2}	—	$1.5 \times t_{cyc} - 45$	ns	
Read data access time3	t_{ACC3}	—	$2.0 \times t_{cyc} - 45$	ns	
Read data access time 4	t_{ACC4}	—	$2.5 \times t_{cyc} - 45$	ns	
Read data access time 5	t_{ACC5}	—	$3.0 \times t_{cyc} - 50$	ns	
$\overline{WR}$ delay time 1	t_{WRD1}	—	20	ns	
$\overline{WR}$ delay time 2	t_{WRD2}	—	20	ns	
$\overline{WR}$ pulse width 1	t_{WSW1}	$1.0 \times t_{cyc} - 20$	—	ns	
$\overline{WR}$ pulse width 2	t_{WSW2}	$1.5 \times t_{cyc} - 20$	—	ns	
Write data delay time	t_{WDD}	—	30	ns	
Write data setup time	t_{WDS}	$0.5 \times t_{cyc} - 20$	—	ns	
Write data hold time	t_{WDH}	$0.5 \times t_{cyc} - 10$	—	ns	

(4) Timing of On-Chip Supporting Modules

Table 24-30 lists the timing of on-chip supporting modules.

Table 24-30 Timing of On-Chip Supporting Modules

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

				Condition			
Item			**Symbol**	**Min**	**Max**	**Unit**	**Test Conditions**
I/O port	Output data delay time		t_{PWD}	—	50	ns	Figure 24-16 Figure 24-17
	Output data delay time 2		t_{PWD2}	—	50		
	Input data setup time		t_{PRS}	30	—		
	Input data hold time		t_{PRH}	30	—		
PPG	Pulse output delay time		t_{POD}	—	50	ns	Figure 24-18
TPU	Timer output delay time		t_{TOCD}	—	50	ns	Figure 24-19
	Timer input setup time		t_{TICS}	30	—		
	Timer clock input setup time		t_{TCKS}	30	—	ns	Figure 24-20
	Timer clock pulse width	Single edge	t_{TCKWH}	1.5	—	t_{cyc}	
		Both edges	t_{TCKWL}	2.5	—		
PWM	Pulse output delay time		t_{MPWMOD}	—	50	ns	Figure 24-21

Item			Symbol	Condition Min	Max	Unit	Test Conditions
SCI	Input clock cycle	Asynchro-nous	t_{Scyc}	4	—	t_{cyc}	Figure 24-22
		Synchronous		6	—		
	Input clock pulse width		t_{SCKW}	0.4	0.6	t_{Scyc}	
	Input clock rise time		t_{SCKr}	—	1.5	t_{cyc}	
	Input clock fall time		t_{SCKf}	—	1.5		
	Transmit data delay time		t_{TXD}	—	50	ns	Figure 24-23
	Receive data setup time (synchronous)		t_{RXS}	50	—		
	Receive data hold time (synchronous)		t_{RXH}	50	—		
A/D converter	Trigger input setup time		t_{TRGS}	50	—	ns	Figure 24-24
HCAN*	Transmit data delay time		t_{HTXD}	—	100	ns	Figure 24-25
	Receive data setup time		t_{HRXS}	100	—		
	Receive data hold time		t_{HRXH}	100	—		

Note: The HCAN input signal is asynchronous. However, its state is judged to have changed at the leading edge (two clock cycles) of the CK clock signal shown in figure 24-25. The HCAN output signal is also asynchronous. Its state changes based on the leading edge (two clock cycles) of the CK clock signal shown in figure 24-25.

Table 24-31 I^2C Bus Timing [Option]

Conditions: V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, ø = 5 MHz to maximum operating frequency, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

		Condition				
Item	**Symbol**	**Min**	**Typ**	**Max**	**Unit**	**Notes**
SCL input cycle time	t_{SCL}	$12t_{cyc}$	—	—	ns	Figure 24-26
SCL input high pulse width	t_{SCLH}	$3t_{cyc}$	—	—	ns	
SCL input low pulse width	t_{SCLL}	$5t_{cyc}$	—	—	ns	
SCL, SDA input rise time	t_{Sr}	—	—	$7.5t_{cyc}$*	ns	
SCL, SDA input fall time	t_{Sf}	—	—	300	ns	
SCL, SDA input spike pulse elimination time	t_{SP}	—	—	$1t_{cyc}$	ns	
SDA input bus free time	t_{BUF}	$5t_{cyc}$	—	—	ns	
Start condition input hold time	t_{STAH}	$3t_{cyc}$	—	—	ns	
Retransmission start condition input setup time	t_{STAS}	$3t_{cyc}$	—	—	ns	
Stop condition input setup time	t_{STOS}	$3t_{cyc}$	—	—	ns	
Data input setup time	t_{SDAS}	$0.5t_{cyc}$	—	—	ns	
Data input hold time	t_{SDAH}	0	—	—	ns	
SCL, SDA capacitive load	C_b	—	—	400	pF	

Note: Available when using I^2C bus interface as an option.

* $17.5t_{cyc}$ can be set according to the clock selected for use by the I^2C module. For details, see section 15.4, Usage Notes.

24.3.5 A/D Conversion Characteristics

Table 24-32 lists the A/D conversion characteristics.

Table 24-32 A/D Conversion Characteristics

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition			
Item	**Min**	**Typ**	**Max**	**Unit**
Resolution	10	10	10	bits
Conversion time	10	—	—	µs
Analog input capacitance	—	—	20	pF
Permissible signal-source impedance	—	—	5	kΩ
Nonlinearity error	—	—	±3.5	LSB
Offset error	—	—	±3.5	LSB
Full-scale error	—	—	±3.5	LSB
Quantization	—	±0.5	—	LSB
Absolute accuracy	—	—	±4.0	LSB

24.3.6 D/A Conversion Characteristics

Table 24-33 shows the D/A conversion characteristics.

Table 24-33 D/A Conversion Characteristics

Condition : V_{CC} = 4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$ = AV_{SS} = 0 V, T_a = –20°C to +75°C (regular specifications), T_a = –40°C to +85°C (wide-range specifications)

	Condition				
Item	**Min**	**Typ**	**Max**	**Unit**	**Test Conditions**
Resolution	8	8	8	bits	
Conversion time	—	—	10	µs	20-pF capacitive load
Absolute accuracy	—	± 1.5	± 2.0	LSB	2-MΩ resistive load
	—	—	± 1.5	LSB	4-MΩ resistive load

24.3.7 Flash Memory Characteristics

Table 24-34 shows the flash memory characteristics.

Table 24-34 Flash Memory Characteristics

Conditions: V_{CC} =4.5 V to 5.5 V, $PWMV_{CC}$ = 4.5 V to 5.5 V, AV_{CC} = 4.5 V to 5.5 V, V_{ref} = 4.5 V to AV_{CC}, V_{SS} = $PWMV_{SS}$ = $PLLV_{SS}$, AV_{SS} = 0 V
T_a = 0 to +75°C (Programming/erasing operating temperature range: regular specification)

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Programming time*1,*2,*4		t_P	—	10	200	ms/ 128 bytes	
Erase time*1,*3,*5		t_E	—	100	1200	ms/block	
Reprogramming count		N_{WEC}	—	—	100	Times	
Programming	Wait time after SWE bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after PSU bit setting*1	t_{spsu}	50	50	—	µs	
	Wait time after P bit setting*1, *4	t_{sp30}	28	30	32	µs	Programming time wait
		t_{sp200}	198	200	202	µs	Programming time wait
		t_{sp10}	8	10	12	µs	Additional-programming time wait
	Wait time after P bit clear*1	t_{cp}	5	5	—	µs	
	Wait time after PSU bit clear*1	t_{cpsu}	5	5	—	µs	
	Wait time after PV bit setting*1	t_{spv}	4	4	—	µs	
	Wait time after H'FF dummy write*1	t_{spvr}	2	2	—	µs	
	Wait time after PV bit clear*1	t_{cpv}	2	2	—	µs	
	Wait time after SWE bit clear*1	t_{cswe}	100	100	—	µs	
	Maximum programming count*1,*4	N	—	—	1000	Times	
Erase	Wait time after SWE bit setting*1	t_{sswe}	1	1	—	µs	
	Wait time after ESU bit setting*1	t_{sesu}	100	100	—	µs	
	Wait time after E bit setting*1,*5	t_{se}	10	10	100	ms	Erase time wait
	Wait time after E bit clear*1	t_{ce}	10	10	—	µs	
	Wait time after ESU bit clear*1	t_{cesu}	10	10	—	µs	
	Wait time after EV bit setting*1	t_{sev}	20	20	—	µs	

Item		Symbol	Min	Typ	Max	Unit	Test Condition
Erase	Wait time after H'FF dummy write*[1]	t_{sevr}	2	2	—	µs	
	Wait time after EV bit clear*[1]	t_{cev}	4	4	—	µs	
	Wait time after SWE bit clear*[1]	t_{cswe}	100	100	—	µs	
	Maximum erase count*[1,*5]	N	12	—	120	Times	

Notes: *1 Make each time setting in accordance with the program/program-verify flowchart or erase/erase-verify flowchart.

*2 Programming time per 128 bytes (Shows the total period for which the P-bit in the flash memory control register (FLMCR1) is set. It does not include the programming verification time.)

*3 Block erase time (Shows the total period for which the E-bit in FLMCR1 is set. It does not include the erase verification time.)

*4 To specify the maximum programming time value (t_P(max)) in the 128-byte programming algorithm, set the max. value (1000) for the maximum programming count (N).

The wait time after P bit setting should be changed as follows according to the value of the programming counter (n).

Programming counter (n) = 1 to 6: t_{sp30} = 30 µs

Programming counter (n) = 7 to 1000: t_{sp200} = 200 µs

[In additional programming]

Programming counter (n)= 1 to 6: t_{sp10} = 10 µs

*5 For the maximum erase time (t_E(max)), the following relationship applies between the wait time after E bit setting (t_{se}) and the maximum erase count (N):

t_E(max) = Wait time after E bit setting (t_{se}) x maximum erase count (N)

To set the maximum erase time, the values of (t_{se}) and (N) should be set so as to satisfy the above formula.

Examples: When t_{se} = 100 [ms], N = 12 times

When t_{se} = 10 [ms], N = 120 times

24.4 Operation Timing

The operation timing is shown below.

24.4.1 Clock Timing

The clock timing is shown below.

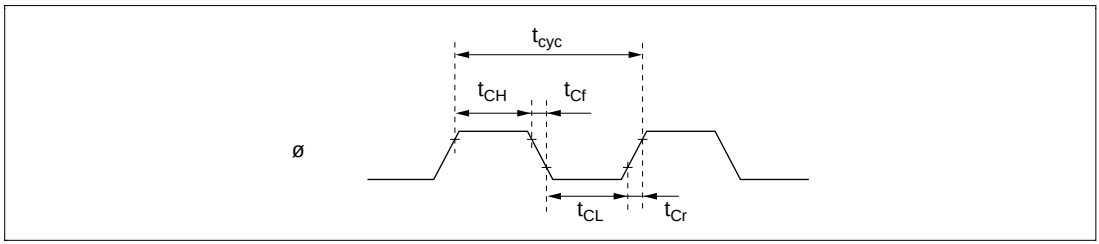


Figure 24-7 System Clock Timing

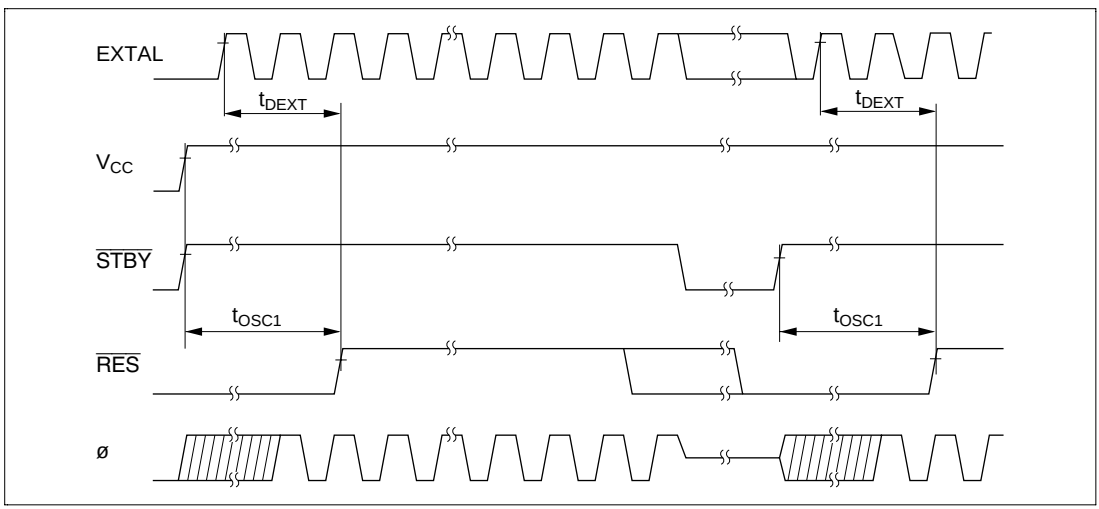


Figure 24-8 Oscillator Settling Timing

24.4.2 Control Signal Timing

The control signal timing is shown below.

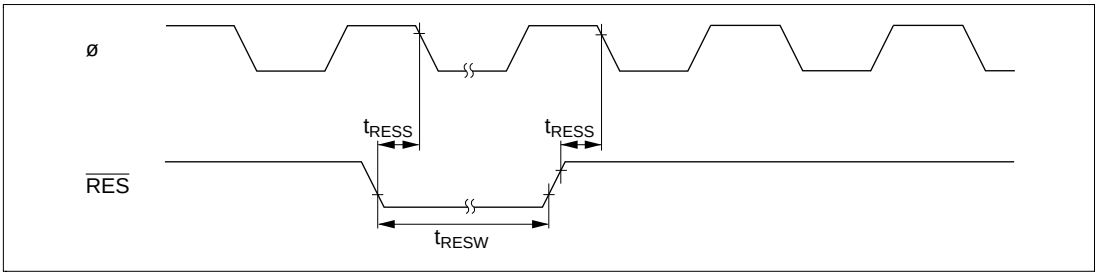


Figure 24-9 Reset Input Timing

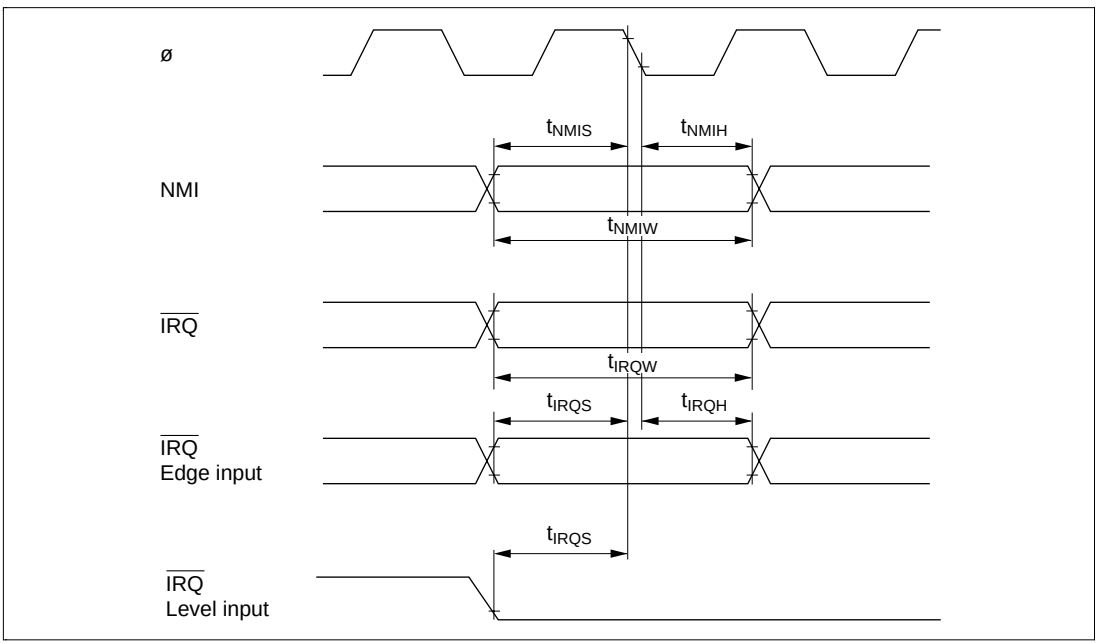


Figure 24-10 Interrupt Input Timing

24.4.3 Bus Timing

The bus timing is shown below.

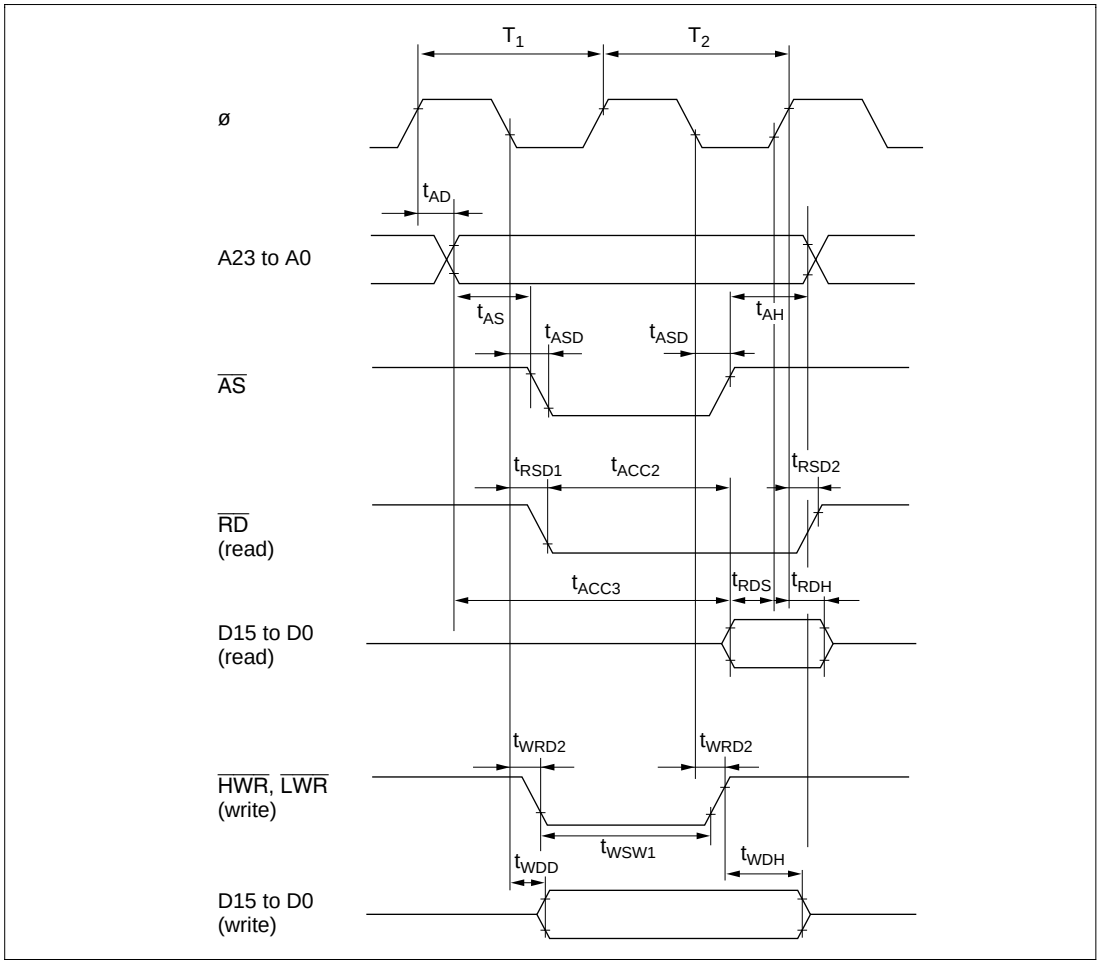


Figure 24-11 Basic Bus Timing (Two-State Access)

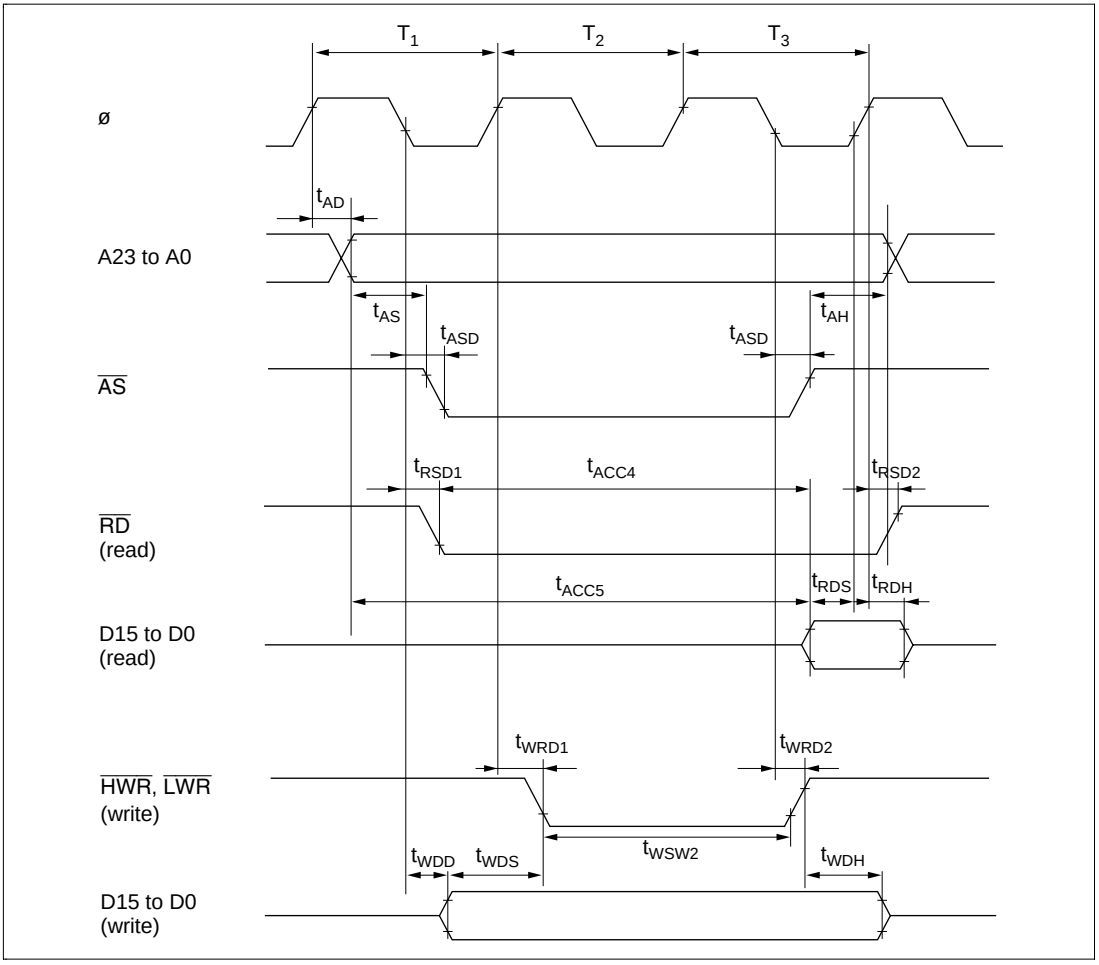


Figure 24-12 Basic Bus Timing (Three-State Access)

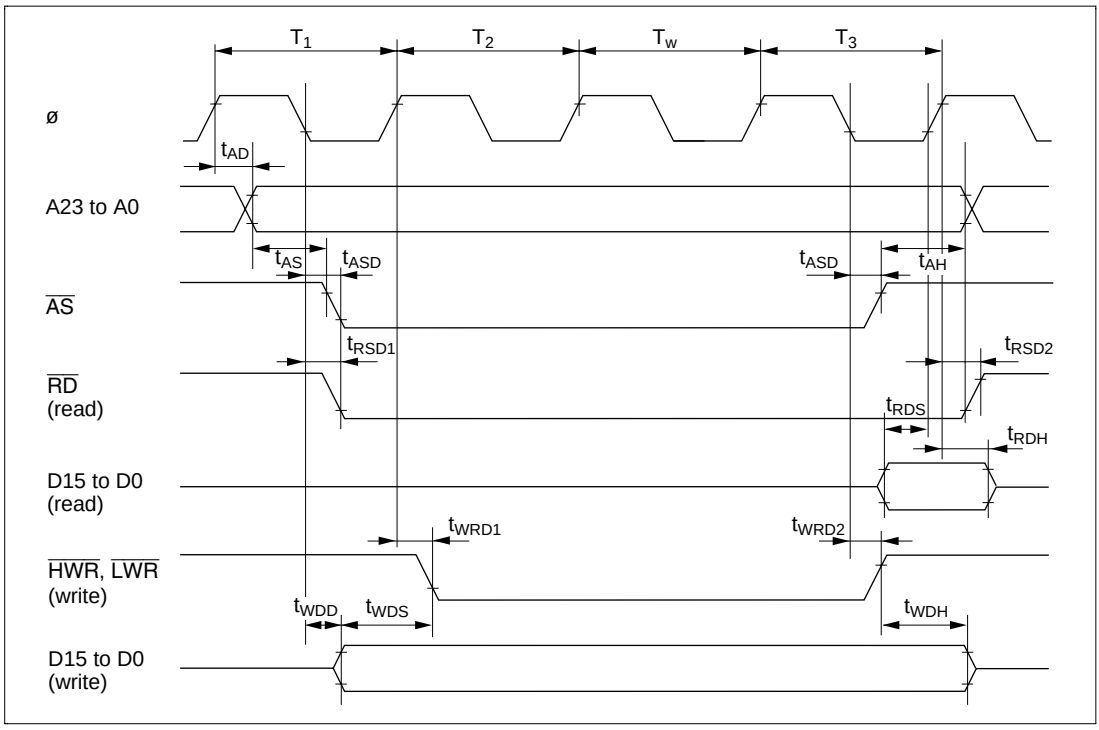


Figure 24-13 Basic Bus Timing (Three-State Access with One Wait State)

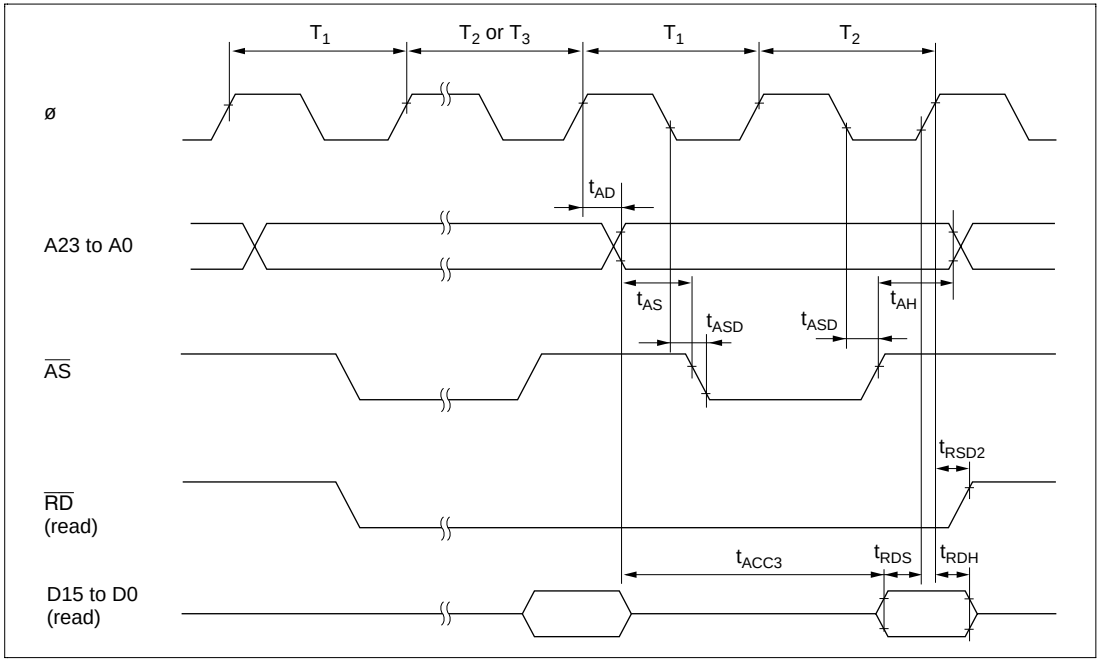


Figure 24-14 Burst ROM Access Timing (Two-State Access)

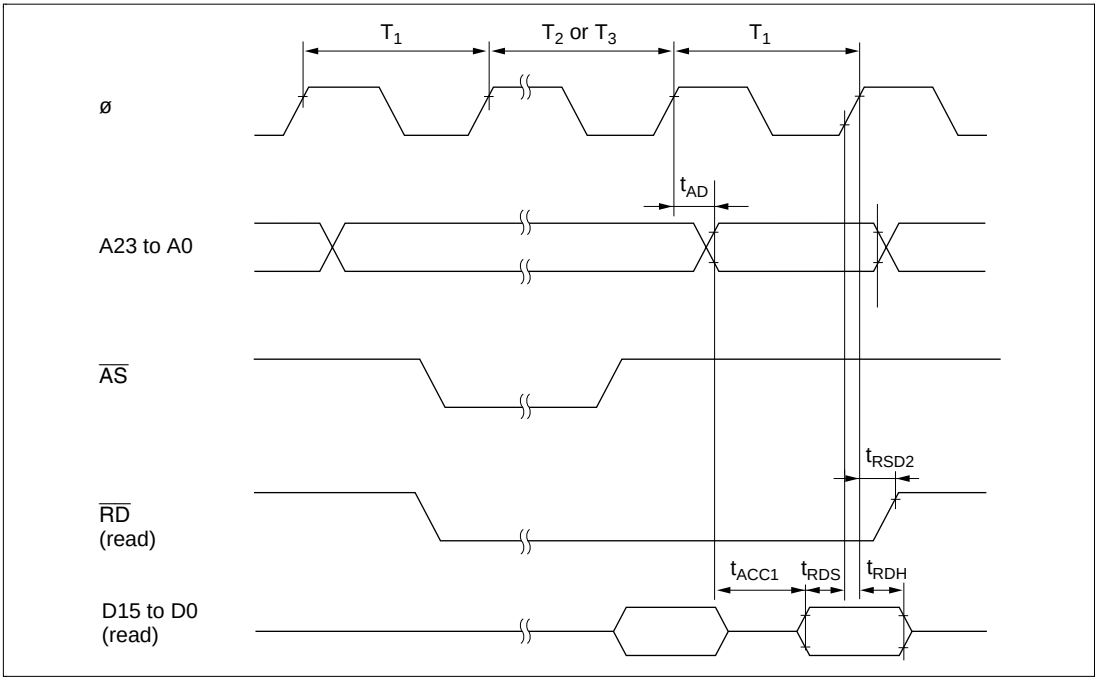


Figure 24-15 Burst ROM Access Timing (One-State Access)

24.4.4 On-Chip Supporting Module Timing

The on-chip supporting module timing is shown below.

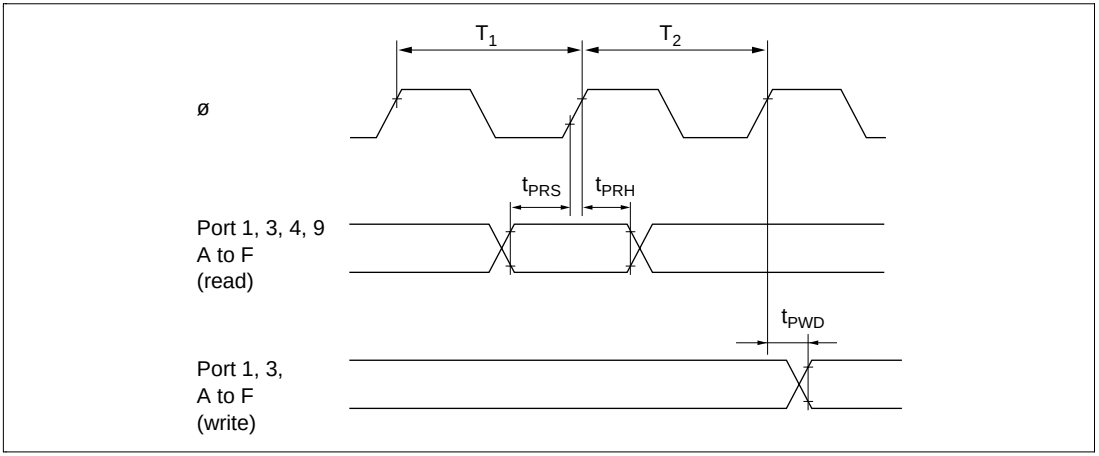


Figure 24-16 I/O Port Input/Output Timing (Port 1, 3, 4, 9, A to F)

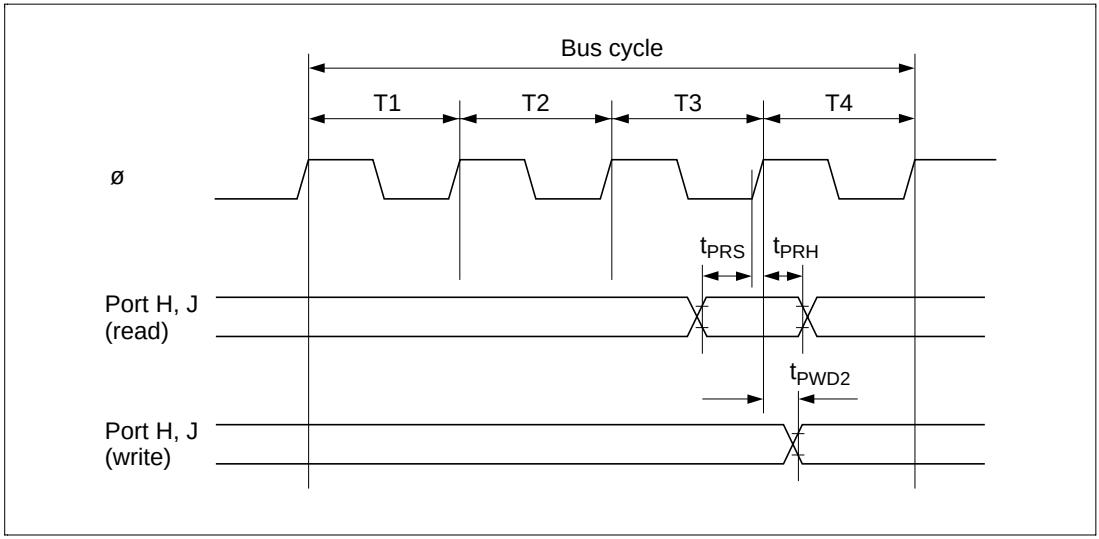


Figure 24-17 I/O Port (Port H and J) Input/Output Timing

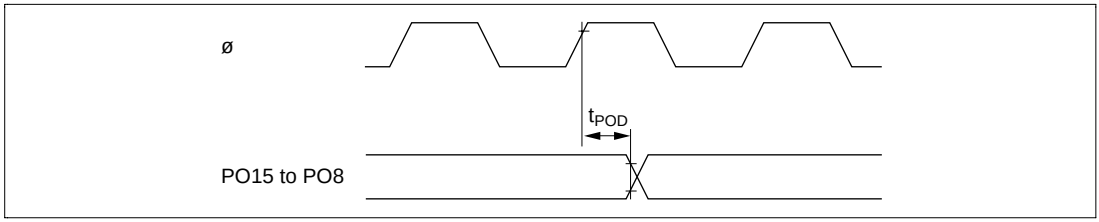


Figure 24-18 PPG Output Timing

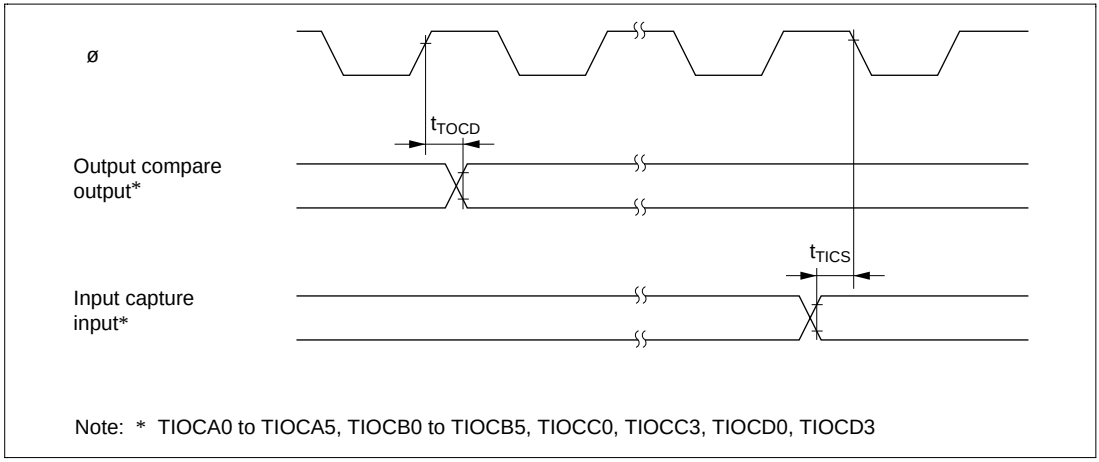


Figure 24-19 TPU Input/Output Timing

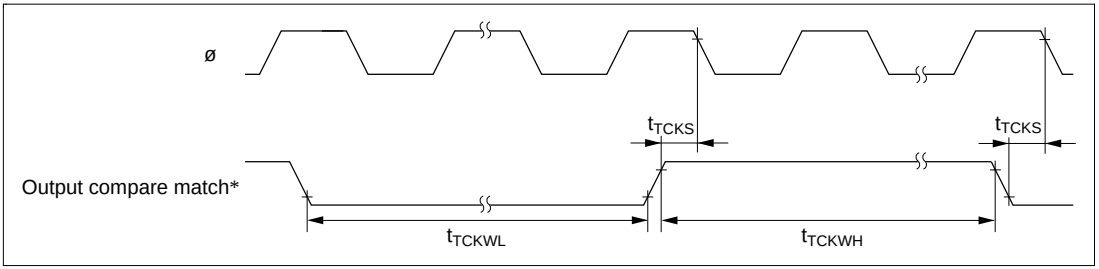


Figure 24-20 TPU Clock Input Timing

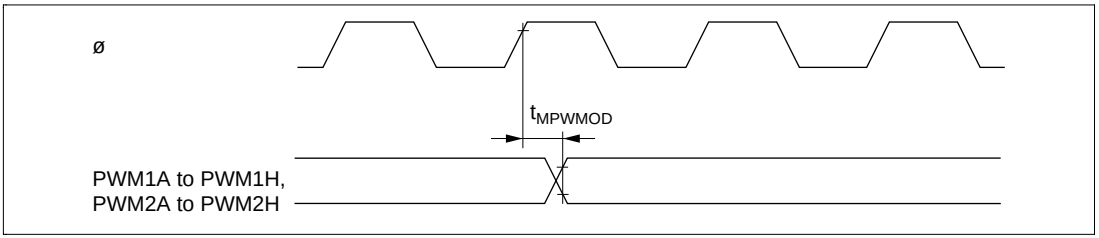


Figure 24-21 Motor Control PWM Output Timing

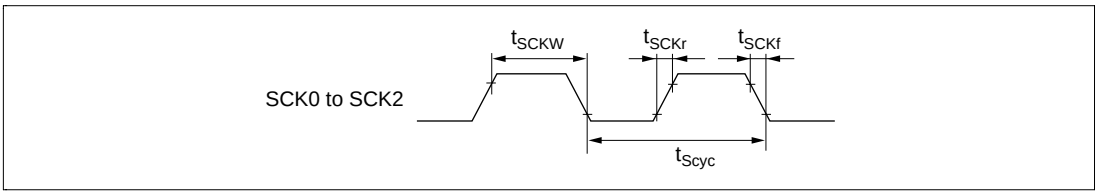


Figure 24-22 SCK Clock Input Timing

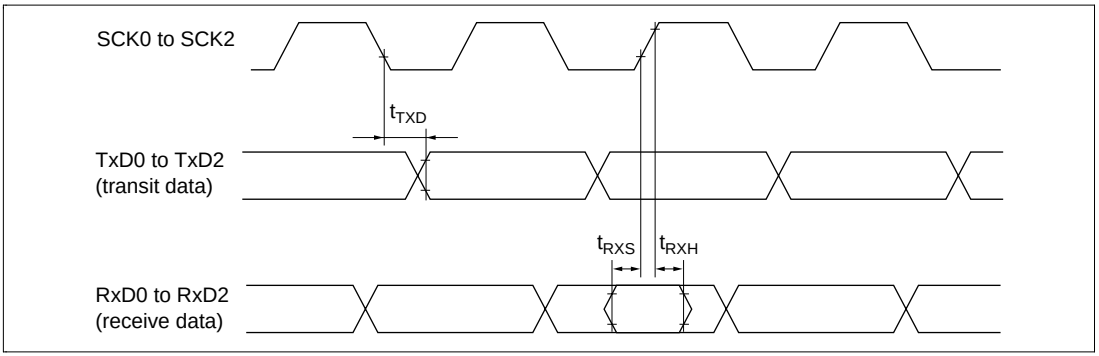


Figure 24-23 SCI Input/Output Timing (Clock Synchronous Mode)

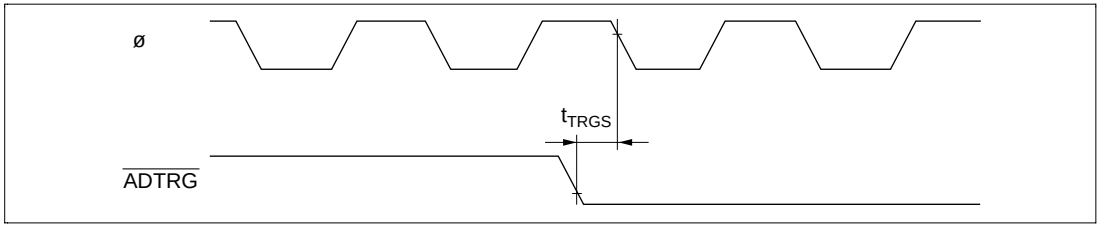


Figure 24-24 A/D Converter External Trigger Input Timing

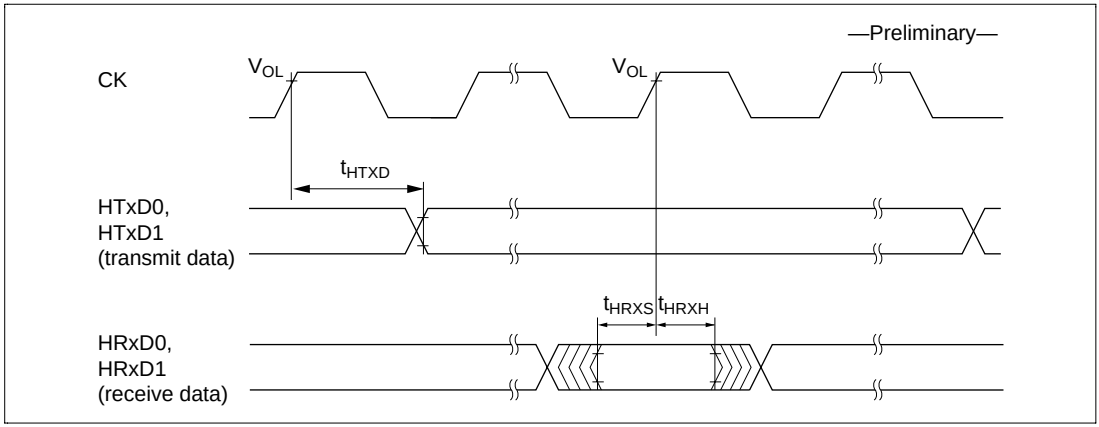
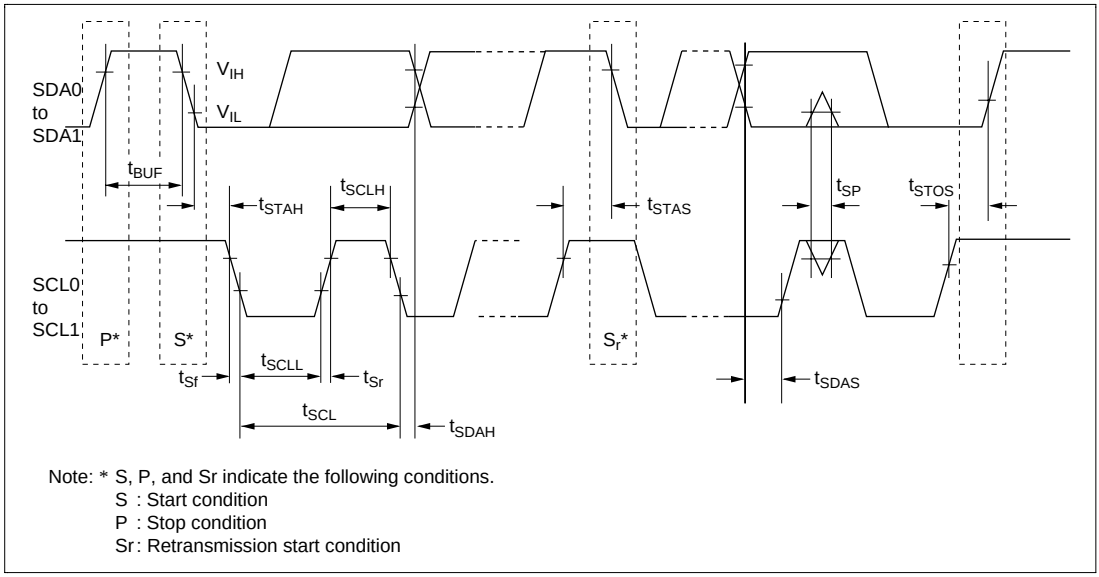


Figure 24-25 HCAN Input/Output Timing



Note: * S, P, and Sr indicate the following conditions.
S : Start condition
P : Stop condition
Sr : Retransmission start condition

Figure 24-26 I^2C Bus Interface Input/Output Timing (Option)

24.5 Usage Note

Although both the F-ZTAT and mask ROM versions fully meet the electrical specifications listed in this manual, there may be differences in the actual values of the electrical characteristics, operating margins, noise margins, and so forth, due to differences in the fabrication process, the on-chip ROM, and the layout patterns.

Therefore, if a system is evaluated using the F-ZTAT version, a similar evaluation should also be performed using the mask ROM version.

Appendix A Instruction Set

A.1 Instruction List

Operand Notation

Rd	General register (destination)*
Rs	General register (source)*
Rn	General register*
ERn	General register (32-bit register)
MAC	Multiply-and-accumulate register (32-bit register)
(EAd)	Destination operand
(EAs)	Source operand
EXR	Extended control register
CCR	Condition-code register
N	N (negative) flag in CCR
Z	Z (zero) flag in CCR
V	V (overflow) flag in CCR
C	C (carry) flag in CCR
PC	Program counter
SP	Stack pointer
#IMM	Immediate data
disp	Displacement
+	Add
–	Subtract
×	Multiply
÷	Divide
∧	Logical AND
∨	Logical OR
⊕	Logical exclusive OR
→	Transfer from the operand on the left to the operand on the right, or transition from the state on the left to the state on the right
¬	Logical NOT (logical complement)
( ) < >	Contents of operand
:8/:16/:24/:32	8-, 16-, 24-, or 32-bit length

Note: * General registers include 8-bit registers (R0H to R7H, R0L to R7L), 16-bit registers (R0 to R7, E0 to E7), and 32-bit registers (ER0 to ER7).

Condition Code Notation

Symbol	
↕	Changes according to the result of instruction
*	Undetermined (no guaranteed value)
0	Always cleared to 0
1	Always set to 1
—	Not affected by execution of the instruction

Table A-1 Instruction Set

(1) Data Transfer Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
MOV	MOV.B #xx:8,Rd	B	2									#xx:8→Rd8	—	—	↕	↕	0	—	1
	MOV.B Rs,Rd	B		2								Rs8→Rd8	—	—	↕	↕	0	—	1
	MOV.B @ERs,Rd	B			2							@ERs→Rd8	—	—	↕	↕	0	—	2
	MOV.B @(d:16,ERs),Rd	B				4						@(d:16,ERs)→Rd8	—	—	↕	↕	0	—	3
	MOV.B @(d:32,ERs),Rd	B				8						@(d:32,ERs)→Rd8	—	—	↕	↕	0	—	5
	MOV.B @ERs+,Rd	B					2					@ERs→Rd8,ERs32+1→ERs32	—	—	↕	↕	0	—	3
	MOV.B @aa:8,Rd	B						2				@aa:8→Rd8	—	—	↕	↕	0	—	2
	MOV.B @aa:16,Rd	B						4				@aa:16→Rd8	—	—	↕	↕	0	—	3
	MOV.B @aa:32,Rd	B						6				@aa:32→Rd8	—	—	↕	↕	0	—	4
	MOV.B Rs,@ERd	B			2							Rs8→@ERd	—	—	↕	↕	0	—	2
	MOV.B Rs,@(d:16,ERd)	B				4						Rs8→@(d:16,ERd)	—	—	↕	↕	0	—	3
	MOV.B Rs,@(d:32,ERd)	B				8						Rs8→@(d:32,ERd)	—	—	↕	↕	0	—	5
	MOV.B Rs,@-ERd	B					2					ERd32-1→ERd32,Rs8→@ERd	—	—	↕	↕	0	—	3
	MOV.B Rs,@aa:8	B						2				Rs8→@aa:8	—	—	↕	↕	0	—	2
	MOV.B Rs,@aa:16	B						4				Rs8→@aa:16	—	—	↕	↕	0	—	3
	MOV.B Rs,@aa:32	B						6				Rs8→@aa:32	—	—	↕	↕	0	—	4
	MOV.W #xx:16,Rd	W	4									#xx:16→Rd16	—	—	↕	↕	0	—	2
	MOV.W Rs,Rd	W		2								Rs16→Rd16	—	—	↕	↕	0	—	1
	MOV.W @ERs,Rd	W			2							@ERs→Rd16	—	—	↕	↕	0	—	2

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes) #xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|	Operation	Condition Code I	H	N	Z	V	C	No. of States*1 Advanced
MOV	MOV.W @(d:16,ERs),Rd	W				4						@(d:16,ERs)→Rd16	—	—	↕	↕	0	—	3
	MOV.W @(d:32,ERs),Rd	W				8						@(d:32,ERs)→Rd16	—	—	↕	↕	0	—	5
	MOV.W @ERs+,Rd	W					2					@ERs→Rd16,ERs32+2→ERs32	—	—	↕	↕	0	—	3
	MOV.W @aa:16,Rd	W						4				@aa:16→Rd16	—	—	↕	↕	0	—	3
	MOV.W @aa:32,Rd	W						6				@aa:32→Rd16	—	—	↕	↕	0	—	4
	MOV.W Rs,@ERd	W			2							Rs16→@ERd	—	—	↕	↕	0	—	2
	MOV.W Rs,@(d:16,ERd)	W				4						Rs16→@(d:16,ERd)	—	—	↕	↕	0	—	3
	MOV.W Rs,@(d:32,ERd)	W				8						Rs16→@(d:32,ERd)	—	—	↕	↕	0	—	5
	MOV.W Rs,@-ERd	W					2					ERd32-2→ERd32,Rs16→@ERd	—	—	↕	↕	0	—	3
	MOV.W Rs,@aa:16	W						4				Rs16→@aa:16	—	—	↕	↕	0	—	3
	MOV.W Rs,@aa:32	W						6				Rs16→@aa:32	—	—	↕	↕	0	—	4
	MOV.L #xx:32,ERd	L	6									#xx:32→ERd32	—	—	↕	↕	0	—	3
	MOV.L ERs,ERd	L		2								ERs32→ERd32	—	—	↕	↕	0	—	1
	MOV.L @ERs,ERd	L			4							@ERs→ERd32	—	—	↕	↕	0	—	4
	MOV.L @(d:16,ERs),ERd	L				6						@(d:16,ERs)→ERd32	—	—	↕	↕	0	—	5
	MOV.L @(d:32,ERs),ERd	L				10						@(d:32,ERs)→ERd32	—	—	↕	↕	0	—	7
	MOV.L @ERs+,ERd	L					4					@ERs→ERd32,ERs32+4→@ERs32	—	—	↕	↕	0	—	5
	MOV.L @aa:16,ERd	L						6				@aa:16→ERd32	—	—	↕	↕	0	—	5
	MOV.L @aa:32,ERd	L						8				@aa:32→ERd32	—	—	↕	↕	0	—	6

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	—		I	H	N	Z	V	C	Advanced
MOV	MOV.L ERs,@ERd	L			4							ERs32→@ERd	—	—	↕	↕	0	—	4
	MOV.L ERs,@(d:16,ERd)	L				6						ERs32→@(d:16,ERd)	—	—	↕	↕	0	—	5
	MOV.L ERs,@(d:32,ERd)	L				10						ERs32→@(d:32,ERd)	—	—	↕	↕	0	—	7
	MOV.L ERs,@-ERd	L					4					ERd32-4→ERd32,ERs32→@ERd	—	—	↕	↕	0	—	5
	MOV.L ERs,@aa:16	L						6				ERs32→@aa:16	—	—	↕	↕	0	—	5
	MOV.L ERs,@aa:32	L						8				ERs32→@aa:32	—	—	↕	↕	0	—	6
POP	POP.W Rn	W									2	@SP→Rn16,SP+2→SP	—	—	↕	↕	0	—	3
	POP.L ERn	L									4	@SP→ERn32,SP+4→SP	—	—	↕	↕	0	—	5
PUSH	PUSH.W Rn	W									2	SP-2→SP,Rn16→@SP	—	—	↕	↕	0	—	3
	PUSH.L ERn	L									4	SP-4→SP,ERn32→@SP	—	—	↕	↕	0	—	5
LDM*4	LDM @SP+,(ERm-ERn)	L									4	(@SP→ERn32,SP+4→SP) Repeated for each register restored	—	—	—	—	—	—	7/9/11 [1]
STM*4	STM (ERm-ERn),@-SP	L									4	(SP-4→SP,ERn32→@SP) Repeated for each register saved	—	—	—	—	—	—	7/9/11 [1]
MOVFPE	MOVFPE @aa:16,Rd	Cannot be used in this LSI																	[2]
MOVTPE	MOVTPE Rs,@aa:16	Cannot be used in this LSI																	[2]

(2) Arithmetic Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes) #xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|	Operation	Condition Code I	H	N	Z	V	C	No. of States*1 Advanced
ADD	ADD.B #xx:8,Rd	B	2									Rd8+#xx:8→Rd8	—	↕	↕	↕	↕	↕	1
	ADD.B Rs,Rd	B		2								Rd8+Rs8→Rd8	—	↕	↕	↕	↕	↕	1
	ADD.W #xx:16,Rd	W	4									Rd16+#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2
	ADD.W Rs,Rd	W		2								Rd16+Rs16→Rd16	—	[3]	↕	↕	↕	↕	1
	ADD.L #xx:32,ERd	L	6									ERd32+#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3
	ADD.L ERs,ERd	L		2								ERd32+ERs32→ERd32	—	[4]	↕	↕	↕	↕	1
ADDX	ADDX #xx:8,Rd	B	2									Rd8+#xx:8+C→Rd8	—	↕	↕	[5]	↕	↕	1
	ADDX Rs,Rd	B		2								Rd8+Rs8+C→Rd8	—	↕	↕	[5]	↕	↕	1
ADDS	ADDS #1,ERd	L		2								ERd32+1→ERd32	—	—	—	—	—	—	1
	ADDS #2,ERd	L		2								ERd32+2→ERd32	—	—	—	—	—	—	1
	ADDS #4,ERd	L		2								ERd32+4→ERd32	—	—	—	—	—	—	1
INC	INC.B Rd	B		2								Rd8+1→Rd8	—	—	↕	↕	↕	—	1
	INC.W #1,Rd	W		2								Rd16+1→Rd16	—	—	↕	↕	↕	—	1
	INC.W #2,Rd	W		2								Rd16+2→Rd16	—	—	↕	↕	↕	—	1
	INC.L #1,ERd	L		2								ERd32+1→ERd32	—	—	↕	↕	↕	—	1
	INC.L #2,ERd	L		2								ERd32+2→ERd32	—	—	↕	↕	↕	—	1
DAA	DAA Rd	B		2								Rd8 decimal adjust→Rd8	—	*	↕	↕	*	↕	1
SUB	SUB.B Rs,Rd	B		2								Rd8-Rs8→Rd8	—	↕	↕	↕	↕	↕	1
	SUB.W #xx:16,Rd	W	4									Rd16-#xx:16→Rd16	—	[3]	↕	↕	↕	↕	2

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
SUB	SUB.W Rs,Rd	W		2								Rd16-Rs16→Rd16	—	[3]	↕	↕	↕	↕	1
	SUB.L #xx:32,ERd	L	6									ERd32-#xx:32→ERd32	—	[4]	↕	↕	↕	↕	3
	SUB.L ERs,ERd	L		2								ERd32-ERs32→ERd32	—	[4]	↕	↕	↕	↕	1
SUBX	SUBX #xx:8,Rd	B	2									Rd8-#xx:8-C→Rd8	—	↕	↕	[5]	↕	↕	1
	SUBX Rs,Rd	B		2								Rd8-Rs8-C→Rd8	—	↕	↕	[5]	↕	↕	1
SUBS	SUBS #1,ERd	L		2								ERd32-1→ERd32	—	—	—	—	—	—	1
	SUBS #2,ERd	L		2								ERd32-2→ERd32	—	—	—	—	—	—	1
	SUBS #4,ERd	L		2								ERd32-4→ERd32	—	—	—	—	—	—	1
DEC	DEC.B Rd	B		2								Rd8-1→Rd8	—	—	↕	↕	↕	—	1
	DEC.W #1,Rd	W		2								Rd16-1→Rd16	—	—	↕	↕	↕	—	1
	DEC.W #2,Rd	W		2								Rd16-2→Rd16	—	—	↕	↕	↕	—	1
	DEC.L #1,ERd	L		2								ERd32-1→ERd32	—	—	↕	↕	↕	—	1
	DEC.L #2,ERd	L		2								ERd32-2→ERd32	—	—	↕	↕	↕	—	1
DAS	DAS Rd	B		2								Rd8 decimal adjust→Rd8	—	*	↕	↕	*	—	1
MULXU	MULXU.B Rs,Rd	B		2								Rd8×Rs8→Rd16 (unsigned multiplication)	—	—	—	—	—	—	12
	MULXU.W Rs,ERd	W		2								Rd16×Rs16→ERd32 (unsigned multiplication)	—	—	—	—	—	—	20
MULXS	MULXS.B Rs,Rd	B		4								Rd8×Rs8→Rd16 (signed multiplication)	—	—	↕	↕	—	—	13
	MULXS.W Rs,ERd	W		4								Rd16×Rs16→ERd32 (signed multiplication)	—	—	↕	↕	—	—	21

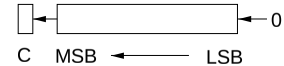
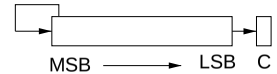
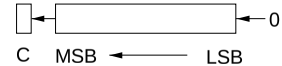
Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
DIVXU	DIVXU.B Rs,Rd	B		2								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (unsigned division)	—	—	[6]	[7]	—	—	12
	DIVXU.W Rs,ERd	W		2								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (unsigned division)	—	—	[6]	[7]	—	—	20
DIVXS	divxs.B Rs,Rd	B		4								Rd16÷Rs8→Rd16 (RdH: remainder, RdL: quotient) (signed division)	—	—	[8]	[7]	—	—	13
	DIVXS.W Rs,ERd	W		4								ERd32÷Rs16→ERd32 (Ed: remainder, Rd: quotient) (signed division)	—	—	[8]	[7]	—	—	21
CMP	CMP.B #xx:8,Rd	B	2									Rd8-#xx:8	—	↕	↕	↕	↕	↕	1
	CMP.B Rs,Rd	B		2								Rd8-Rs8	—	↕	↕	↕	↕	↕	1
	CMP.W #xx:16,Rd	W	4									Rd16-#xx:16	—	[3]	↕	↕	↕	↕	2
	CMP.W Rs,Rd	W		2								Rd16-Rs16	—	[3]	↕	↕	↕	↕	1
	CMP.L #xx:32,ERd	L	6									ERd32-#xx:32	—	[4]	↕	↕	↕	↕	3
	CMP.L ERs,ERd	L		2								ERd32-ERs32	—	[4]	↕	↕	↕	↕	1
NEG	NEG.B Rd	B		2								0-Rd8→Rd8	—	↕	↕	↕	↕	↕	1
	NEG.W Rd	W		2								0-Rd16→Rd16	—	↕	↕	↕	↕	↕	1
	NEG.L ERd	L		2								0-ERd32→ERd32	—	↕	↕	↕	↕	↕	1
EXTU	EXTU.W Rd	W		2								0→(<bit 15 to 8> of Rd16)	—	—	0	↕	0	—	1
	EXTU.L ERd	L		2								0→(<bit 31 to 16> of ERd32)	—	—	0	↕	0	—	1

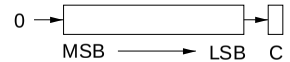
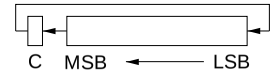
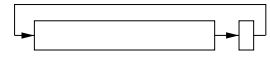
Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
EXTS	EXTS.W Rd	W		2								(<bit 7> of Rd16)→ (<bit 15 to 8> of Rd16)	—	—	↕	↕	0	—	1
	EXTS.L ERd	L		2								(<bit 15> of ERd32)→ (<bit 31 to 16> of ERd32)	—	—	↕	↕	0	—	1
TAS	TAS @ERd*3	B			4							@ERd-0→CCR set, (1)→ (<bit 7> of @ERd)	—	—	↕	↕	0	—	4
MAC	MAC @ERn+, @ERm+	—					4					@ERnx@ERm+MAC→MAC (signal multiplication) @ERn+2→ERn, ERm+2→ERm	—	—	— [11]	— [11]	— [11]	—	4
CLRMAC	CLRMAC	—									2	0→MACH, MACL	—	—	—	—	—	—	2 [12]
LDMAC	LDMAC ERs,MACH	L		2								ERs→MACH	—	—	—	—	—	—	2 [12]
	LDMAC ERs,MACL	L		2								ERs→MACL	—	—	—	—	—	—	2 [12]
STMAC	STMAC MACH,ERd	L		2								MACH→ERd	—	—	↕	↕	↕	—	1 [12]
	STMAC MACL,ERd	L		2								MACL→ERd	—	—	↕	↕	↕	—	1 [12]

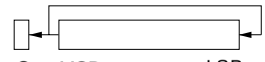
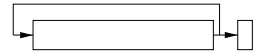
(3) Logical Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
AND	AND.B #xx:8,Rd	B	2									Rd8∧#xx:8→Rd8	—	—	↕	↕	0	—	1
	AND.B Rs,Rd	B		2								Rd8∧Rs8→Rd8	—	—	↕	↕	0	—	1
	AND.W #xx:16,Rd	W	4									Rd16∧#xx:16→Rd16	—	—	↕	↕	0	—	2
	AND.W Rs,Rd	W		2								Rd16∧Rs16→Rd16	—	—	↕	↕	0	—	1
	AND.L #xx:32,ERd	L	6									ERd32∧#xx:32→ERd32	—	—	↕	↕	0	—	3
	AND.L ERs,ERd	L		4								ERd32∧ERs32→ERd32	—	—	↕	↕	0	—	2
OR	OR.B #xx:8,Rd	B	2									Rd8∨#xx:8→Rd8	—	—	↕	↕	0	—	1
	OR.B Rs,Rd	B		2								Rd8∨Rs8→Rd8	—	—	↕	↕	0	—	1
	OR.W #xx:16,Rd	W	4									Rd16∨#xx:16→Rd16	—	—	↕	↕	0	—	2
	OR.W Rs,Rd	W		2								Rd16∨Rs16→Rd16	—	—	↕	↕	0	—	1
	OR.L #xx:32,ERd	L	6									ERd32∨#xx:32→ERd32	—	—	↕	↕	0	—	3
	OR.L ERs,ERd	L		4								ERd32∨ERs32→ERd32	—	—	↕	↕	0	—	2
XOR	XOR.B #xx:8,Rd	B	2									Rd8⊕#xx:8→Rd8	—	—	↕	↕	0	—	1
	XOR.B Rs,Rd	B		2								Rd8⊕Rs8→Rd8	—	—	↕	↕	0	—	1
	XOR.W #xx:16,Rd	W	4									Rd16⊕#xx:16→Rd16	—	—	↕	↕	0	—	2
	XOR.W Rs,Rd	W		2								Rd16⊕Rs16→Rd16	—	—	↕	↕	0	—	1
	XOR.L #xx:32,ERd	L	6									ERd32⊕#xx:32→ERd32	—	—	↕	↕	0	—	3
	XOR.L ERs,ERd	L		4								ERd32⊕ERs32→ERd32	—	—	↕	↕	0	—	2
NOT	NOT.B Rd	B		2								¬ Rd8→Rd8	—	—	↕	↕	0	—	1
	NOT.W Rd	W		2								¬ Rd16→Rd16	—	—	↕	↕	0	—	1
	NOT.L ERd	L		2								¬ ERd32→ERd32	—	—	↕	↕	0	—	1

(4) Shift Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)										Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
SHAL	SHAL.B Rd	B		2									—	—	↕	↕	↕	↕	1
	SHAL.B #2,Rd	B		2									—	—	↕	↕	↕	↕	1
	SHAL.W Rd	W		2									—	—	↕	↕	↕	↕	1
	SHAL.W #2,Rd	W		2									—	—	↕	↕	↕	↕	1
	SHAL.L ERd	L		2									—	—	↕	↕	↕	↕	1
	SHAL.L #2,ERd	L		2									—	—	↕	↕	↕	↕	1
SHAR	SHAR.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHAR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	SHAR.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHAR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	SHAR.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHAR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
SHLL	SHLL.B Rd	B		2									—	—	↕	↕	0	↕	1
	SHLL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	SHLL.W Rd	W		2									—	—	↕	↕	0	↕	1
	SHLL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	SHLL.L ERd	L		2									—	—	↕	↕	0	↕	1
	SHLL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
SHLR	SHLR.B Rd	B		2									—	—	0	↕	0	↕	1
	SHLR.B #2,Rd	B		2									—	—	0	↕	0	↕	1
	SHLR.W Rd	W		2									—	—	0	↕	0	↕	1
	SHLR.W #2,Rd	W		2									—	—	0	↕	0	↕	1
	SHLR.L ERd	L		2									—	—	0	↕	0	↕	1
	SHLR.L #2,ERd	L		2									—	—	0	↕	0	↕	1
ROTXL	ROTXL.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXL.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXL.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTXL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
ROTXR	ROTXR.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTXR.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTXR.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTXR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
ROTL	ROTL.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTL.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTL.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTL.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTL.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTL.L #2,ERd	L		2									—	—	↕	↕	0	↕	1
ROTR	ROTR.B Rd	B		2									—	—	↕	↕	0	↕	1
	ROTR.B #2,Rd	B		2									—	—	↕	↕	0	↕	1
	ROTR.W Rd	W		2									—	—	↕	↕	0	↕	1
	ROTR.W #2,Rd	W		2									—	—	↕	↕	0	↕	1
	ROTR.L ERd	L		2									—	—	↕	↕	0	↕	1
	ROTR.L #2,ERd	L		2									—	—	↕	↕	0	↕	1

(5) Bit-Manipulation Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BSET	BSET #xx:3,Rd	B		2								(#xx:3 of Rd8)←1	—	—	—	—	—	—	1
	BSET #xx:3,@ERd	B			4							(#xx:3 of @ERd)←1	—	—	—	—	—	—	4
	BSET #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←1	—	—	—	—	—	—	4
	BSET #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←1	—	—	—	—	—	—	5
	BSET #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←1	—	—	—	—	—	—	6
	BSET Rn,Rd	B		2								(Rn8 of Rd8)←1	—	—	—	—	—	—	1
	BSET Rn,@ERd	B			4							(Rn8 of @ERd)←1	—	—	—	—	—	—	4
	BSET Rn,@aa:8	B						4				(Rn8 of @aa:8)←1	—	—	—	—	—	—	4
	BSET Rn,@aa:16	B						6				(Rn8 of @aa:16)←1	—	—	—	—	—	—	5
	BSET Rn,@aa:32	B						8				(Rn8 of @aa:32)←1	—	—	—	—	—	—	6
BCLR	BCLR #xx:3,Rd	B		2								(#xx:3 of Rd8)←0	—	—	—	—	—	—	1
	BCLR #xx:3,@ERd	B			4							(#xx:3 of @ERd)←0	—	—	—	—	—	—	4
	BCLR #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)←0	—	—	—	—	—	—	4
	BCLR #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)←0	—	—	—	—	—	—	5
	BCLR #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)←0	—	—	—	—	—	—	6
	BCLR Rn,Rd	B		2								(Rn8 of Rd8)←0	—	—	—	—	—	—	1
	BCLR Rn,@ERd	B			4							(Rn8 of @ERd)←0	—	—	—	—	—	—	4
	BCLR Rn,@aa:8	B						4				(Rn8 of @aa:8)←0	—	—	—	—	—	—	4
	BCLR Rn,@aa:16	B						6				(Rn8 of @aa:16)←0	—	—	—	—	—	—	5

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BCLR	BCLR Rn,@aa:32	B						8				(Rn8 of @aa:32)←0	—	—	—	—	—	—	6
BNOT	BNOT #xx:3,Rd	B		2								(#xx:3 of Rd8)←[¬ (#xx:3 of Rd8)]	—	—	—	—	—	—	1
	BNOT #xx:3,@ERd	B			4							(#xx:3 of @ERd)← [¬ (#xx:3 of @ERd)]	—	—	—	—	—	—	4
	BNOT #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)← [¬ (#xx:3 of @aa:8)]	—	—	—	—	—	—	4
	BNOT #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)← [¬ (#xx:3 of @aa:16)]	—	—	—	—	—	—	5
	BNOT #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)← [¬ (#xx:3 of @aa:32)]	—	—	—	—	—	—	6
	BNOT Rn,Rd	B		2								(Rn8 of Rd8)←[¬ (Rn8 of Rd8)]	—	—	—	—	—	—	1
	BNOT Rn,@ERd	B			4							(Rn8 of @ERd)←[¬ (Rn8 of @ERd)]	—	—	—	—	—	—	4
	BNOT Rn,@aa:8	B						4				(Rn8 of @aa:8)←[¬ (Rn8 of @aa:8)]	—	—	—	—	—	—	4
	BNOT Rn,@aa:16	B						6				(Rn8 of @aa:16)← [¬ (Rn8 of @aa:16)]	—	—	—	—	—	—	5
	BNOT Rn,@aa:32	B						8				(Rn8 of @aa:32)← [¬ (Rn8 of @aa:32)]	—	—	—	—	—	—	6
BTST	BTST #xx:3,Rd	B		2								¬ (#xx:3 of Rd8)→Z	—	—	—	↕	—	—	1
	BTST #xx:3,@ERd	B			4							¬ (#xx:3 of @ERd)→Z	—	—	—	↕	—	—	3
	BTST #xx:3,@aa:8	B						4				¬ (#xx:3 of @aa:8)→Z	—	—	—	↕	—	—	3
	BTST #xx:3,@aa:16	B						6				¬ (#xx:3 of @aa:16)→Z	—	—	—	↕	—	—	4

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BTST	BTST #xx:3,@aa:32	B						8				¬ (#xx:3 of @aa:32)→Z	—	—	—	↕	—	—	5
	BTST Rn,Rd	B		2								¬ (Rn8 of Rd8)→Z	—	—	—	↕	—	—	1
	BTST Rn,@ERd	B			4							¬ (Rn8 of @ERd)→Z	—	—	—	↕	—	—	3
	BTST Rn,@aa:8	B						4				¬ (Rn8 of @aa:8)→Z	—	—	—	↕	—	—	3
	BTST Rn,@aa:16	B						6				¬ (Rn8 of @aa:16)→Z	—	—	—	↕	—	—	4
	BTST Rn,@aa:32	B						8				¬ (Rn8 of @aa:32)→Z	—	—	—	↕	—	—	5
BLD	BLD #xx:3,Rd	B		2								(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BLD #xx:3,@ERd	B			4							(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BLD #xx:3,@aa:8	B						4				(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BLD #xx:3,@aa:16	B						6				(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BLD #xx:3,@aa:32	B						8				(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BILD	BILD #xx:3,Rd	B		2								¬ (#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BILD #xx:3,@ERd	B			4							¬ (#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BILD #xx:3,@aa:8	B						4				¬ (#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BILD #xx:3,@aa:16	B						6				¬ (#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BILD #xx:3,@aa:32	B						8				¬ (#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BST	BST #xx:3,Rd	B		2								C→(#xx:3 of Rd8)	—	—	—	—	—	—	1
	BST #xx:3,@ERd	B			4							C→(#xx:3 of @ERd)	—	—	—	—	—	—	4
	BST #xx:3,@aa:8	B						4				C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BST	BST #xx:3,@aa:16	B						6				C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5
	BST #xx:3,@aa:32	B						8				C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6
BIST	BIST #xx:3,Rd	B		2								¬ C→(#xx:3 of Rd8)	—	—	—	—	—	—	1
	BIST #xx:3,@ERd	B			4							¬ C→(#xx:3 of @ERd)	—	—	—	—	—	—	4
	BIST #xx:3,@aa:8	B						4				¬ C→(#xx:3 of @aa:8)	—	—	—	—	—	—	4
	BIST #xx:3,@aa:16	B						6				¬ C→(#xx:3 of @aa:16)	—	—	—	—	—	—	5
	BIST #xx:3,@aa:32	B						8				¬ C→(#xx:3 of @aa:32)	—	—	—	—	—	—	6
BAND	BAND #xx:3,Rd	B		2								C∧(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BAND #xx:3,@ERd	B			4							C∧(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BAND #xx:3,@aa:8	B						4				C∧(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BAND #xx:3,@aa:16	B						6				C∧(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BAND #xx:3,@aa:32	B						8				C∧(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIAND	BIAND #xx:3,Rd	B		2								C∧[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIAND #xx:3,@ERd	B			4							C∧[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIAND #xx:3,@aa:8	B						4				C∧[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIAND #xx:3,@aa:16	B						6				C∧[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIAND #xx:3,@aa:32	B						8				C∧[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5
BOR	BOR #xx:3,Rd	B		2								C∨(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BOR #xx:3,@ERd	B			4							C∨(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
BOR	BOR #xx:3,@aa:8	B						4				C∨(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BOR #xx:3,@aa:16	B						6				C∨(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BOR #xx:3,@aa:32	B						8				C∨(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIOR	BIOR #xx:3,Rd	B		2								C∨[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIOR #xx:3,@ERd	B			4							C∨[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIOR #xx:3,@aa:8	B						4				C∨[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIOR #xx:3,@aa:16	B						6				C∨[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIOR #xx:3,@aa:32	B						8				C∨[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5
BXOR	BXOR #xx:3,Rd	B		2								C⊕(#xx:3 of Rd8)→C	—	—	—	—	—	↕	1
	BXOR #xx:3,@ERd	B			4							C⊕(#xx:3 of @ERd)→C	—	—	—	—	—	↕	3
	BXOR #xx:3,@aa:8	B						4				C⊕(#xx:3 of @aa:8)→C	—	—	—	—	—	↕	3
	BXOR #xx:3,@aa:16	B						6				C⊕(#xx:3 of @aa:16)→C	—	—	—	—	—	↕	4
	BXOR #xx:3,@aa:32	B						8				C⊕(#xx:3 of @aa:32)→C	—	—	—	—	—	↕	5
BIXOR	BIXOR #xx:3,Rd	B		2								C⊕[¬ (#xx:3 of Rd8)]→C	—	—	—	—	—	↕	1
	BIXOR #xx:3,@ERd	B			4							C⊕[¬ (#xx:3 of @ERd)]→C	—	—	—	—	—	↕	3
	BIXOR #xx:3,@aa:8	B						4				C⊕[¬ (#xx:3 of @aa:8)]→C	—	—	—	—	—	↕	3
	BIXOR #xx:3,@aa:16	B						6				C⊕[¬ (#xx:3 of @aa:16)]→C	—	—	—	—	—	↕	4
	BIXOR #xx:3,@aa:32	B						8				C⊕[¬ (#xx:3 of @aa:32)]→C	—	—	—	—	—	↕	5

(6) Branch Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		Branching Condition	I	H	N	Z	V	C	Advanced
Bcc	BRA d:8(BT d:8)	—							2			if condition is true then	Always	—	—	—	—	—	—	2
	BRA d:16(BT d:16)	—							4			PC←PC+d		—	—	—	—	—	—	3
	BRN d:8(BF d:8)	—							2			else next;	Never	—	—	—	—	—	—	2
	BRN d:16(BF d:16)	—							4					—	—	—	—	—	—	3
	BHI d:8	—							2				C∨Z=0	—	—	—	—	—	—	2
	BHI d:16	—							4					—	—	—	—	—	—	3
	BLS d:8	—							2				C∨Z=1	—	—	—	—	—	—	2
	BLS d:16	—							4					—	—	—	—	—	—	3
	BCC d:B(BHS d:8)	—							2				C=0	—	—	—	—	—	—	2
	BCC d:16(BHS d:16)	—							4					—	—	—	—	—	—	3
	BCS d:8(BLO d:8)	—							2				C=1	—	—	—	—	—	—	2
	BCS d:16(BLO d:16)	—							4					—	—	—	—	—	—	3
	BNE d:8	—							2				Z=0	—	—	—	—	—	—	2
	BNE d:16	—							4					—	—	—	—	—	—	3
	BEQ d:8	—							2				Z=1	—	—	—	—	—	—	2
	BEQ d:16	—							4					—	—	—	—	—	—	3
	BVC d:8	—							2				V=0	—	—	—	—	—	—	2
	BVC d:16	—							4					—	—	—	—	—	—	3

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes) #xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|	Operation	Branching Condition	Condition Code I	H	N	Z	V	C	No. of States*1 Advanced
Bcc	BVS d:8	—							2				V=1	—	—	—	—	—	—	2
	BVS d:16	—							4					—	—	—	—	—	—	3
	BPL d:8	—							2				N=0	—	—	—	—	—	—	2
	BPL d:16	—							4					—	—	—	—	—	—	3
	BMI d:8	—							2				N=1	—	—	—	—	—	—	2
	BMI d:16	—							4					—	—	—	—	—	—	3
	BGE d:8	—							2				N⊕V=0	—	—	—	—	—	—	2
	BGE d:16	—							4					—	—	—	—	—	—	3
	BLT d:8	—							2				N⊕V=1	—	—	—	—	—	—	2
	BLT d:16	—							4					—	—	—	—	—	—	3
	BGT d:8	—							2				Z∨(N⊕V)=0	—	—	—	—	—	—	2
	BGT d:16	—							4					—	—	—	—	—	—	3
	BLE d:8	—							2				Z∨(N⊕V)=1	—	—	—	—	—	—	2
	BLE d:16	—							4					—	—	—	—	—	—	3

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
JMP	JMP @ERn	—			2							PC←ERn	—	—	—	—	—	—	2
	JMP @aa:24	—						4				PC←aa:24	—	—	—	—	—	—	3
	JMP @@aa:8	—								2		PC←@aa:8	—	—	—	—	—	—	5
BSR	BSR d:8	—							2			PC→@-SP,PC←PC+d:8	—	—	—	—	—	—	4
	BSR d:16	—							4			PC→@-SP,PC←PC+d:16	—	—	—	—	—	—	5
JSR	JSR @ERn	—			2							PC→@-SP,PC←ERn	—	—	—	—	—	—	4
	JSR @aa:24	—						4				PC→@-SP,PC←aa:24	—	—	—	—	—	—	5
	JSR @@aa:8	—								2		PC→@-SP,PC←@aa:8	—	—	—	—	—	—	6
RTS	RTS	—									2	PC←@SP+	—	—	—	—	—	—	5

(7) System Control Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes) #xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|	Operation	Condition Code I	H	N	Z	V	C	No. of States*1 Advanced
TRAPA	TRAPA #xx:2	—										PC→@-SP,CCR→@-SP, EXR→@-SP,<vector>→PC	1	—	—	—	—	—	8 [9]
RTE	RTE	—										EXR←@SP+,CCR←@SP+, PC←@SP+	↕	↕	↕	↕	↕	↕	5 [9]
SLEEP	SLEEP	—										Transition to power-down state	—	—	—	—	—	—	2
LDC	LDC #xx:8,CCR	B	2									#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	LDC #xx:8,EXR	B	4									#xx:8→EXR	—	—	—	—	—	—	2
	LDC Rs,CCR	B		2								Rs8→CCR	↕	↕	↕	↕	↕	↕	1
	LDC Rs,EXR	B		2								Rs8→EXR	—	—	—	—	—	—	1
	LDC @ERs,CCR	W			4							@ERs→CCR	↕	↕	↕	↕	↕	↕	3
	LDC @ERs,EXR	W			4							@ERs→EXR	—	—	—	—	—	—	3
	LDC @(d:16,ERs),CCR	W				6						@(d:16,ERs)→CCR	↕	↕	↕	↕	↕	↕	4
	LDC @(d:16,ERs),EXR	W				6						@(d:16,ERs)→EXR	—	—	—	—	—	—	4
	LDC @(d:32,ERs),CCR	W				10						@(d:32,ERs)→CCR	↕	↕	↕	↕	↕	↕	6
	LDC @(d:32,ERs),EXR	W				10						@(d:32,ERs)→EXR	—	—	—	—	—	—	6
	LDC @ERs+,CCR	W					4					@ERs→CCR,ERs32+2→ERs32	↕	↕	↕	↕	↕	↕	4
	LDC @ERs+,EXR	W					4					@ERs→EXR,ERs32+2→ERs32	—	—	—	—	—	—	4
	LDC @aa:16,CCR	W						6				@aa:16→CCR	↕	↕	↕	↕	↕	↕	4
	LDC @aa:16,EXR	W						6				@aa:16→EXR	—	—	—	—	—	—	4
	LDC @aa:32,CCR	W						8				@aa:32→CCR	↕	↕	↕	↕	↕	↕	5
	LDC @aa:32,EXR	W						8				@aa:32→EXR	—	—	—	—	—	—	5

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes)									Operation	Condition Code						No. of States*1
			#xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|		I	H	N	Z	V	C	Advanced
STC	STC CCR,Rd	B		2								CCR→Rd8	—	—	—	—	—	—	1
	STC EXR,Rd	B		2								EXR→Rd8	—	—	—	—	—	—	1
	STC CCR,@ERd	W			4							CCR→@ERd	—	—	—	—	—	—	3
	STC EXR,@ERd	W			4							EXR→@ERd	—	—	—	—	—	—	3
	STC CCR,@(d:16,ERd)	W				6						CCR→@(d:16,ERd)	—	—	—	—	—	—	4
	STC EXR,@(d:16,ERd)	W				6						EXR→@(d:16,ERd)	—	—	—	—	—	—	4
	STC CCR,@(d:32,ERd)	W				10						CCR→@(d:32,ERd)	—	—	—	—	—	—	6
	STC EXR,@(d:32,ERd)	W				10						EXR→@(d:32,ERd)	—	—	—	—	—	—	6
	STC CCR,@-ERd	W					4					ERd32-2→ERd32,CCR→@ERd	—	—	—	—	—	—	4
	STC EXR,@-ERd	W					4					ERd32-2→ERd32,EXR→@ERd	—	—	—	—	—	—	4
	STC CCR,@aa:16	W						6				CCR→@aa:16	—	—	—	—	—	—	4
	STC EXR,@aa:16	W						6				EXR→@aa:16	—	—	—	—	—	—	4
	STC CCR,@aa:32	W						8				CCR→@aa:32	—	—	—	—	—	—	5
	STC EXR,@aa:32	W						8				EXR→@aa:32	—	—	—	—	—	—	5
ANDC	ANDC #xx:8,CCR	B	2									CCR∧#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	ANDC #xx:8,EXR	B	4									EXR∧#xx:8→EXR	—	—	—	—	—	—	2
ORC	ORC #xx:8,CCR	B	2									CCR∨#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	ORC #xx:8,EXR	B	4									EXR∨#xx:8→EXR	—	—	—	—	—	—	2
XORC	XORC #xx:8,CCR	B	2									CCR⊕#xx:8→CCR	↕	↕	↕	↕	↕	↕	1
	XORC #xx:8,EXR	B	4									EXR⊕#xx:8→EXR	—	—	—	—	—	—	2
NOP	NOP	—									2	PC←PC+2	—	—	—	—	—	—	1

(8) Block Transfer Instructions

Mnemonic		Operand Size	Addressing Mode/ Instruction Length (Bytes) #xx	Rn	@ERn	@(d,ERn)	@–ERn/@ERn+	@aa	@(d,PC)	@@aa	|	Operation	Condition Code I	H	N	Z	V	C	No. of States*1 Advanced
EEPMOV	EEPMOV.B	—									4	if R4L≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4L-1→R4L Until R4L=0 else next;	—	—	—	—	—	—	4+2n *2
	EEPMOV.W	—									4	if R4≠0 Repeat @ER5→@ER6 ER5+1→ER5 ER6+1→ER6 R4-1→R4 Until R4=0 else next;	—	—	—	—	—	—	4+2n *2

Notes: *1 The number of states is the number of states required for execution when the instruction and its operands are located in on-chip memory.
*2 n is the initial value of R4L or R4.
*3 Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.
*4 Only registers ER0 to ER6 should be used when using the STM/LDM instruction.
[1] Seven states for saving or restoring two registers, nine states for three registers, or eleven states for four registers.
[2] Cannot be used in this LSI.
[3] Set to 1 when a carry or borrow occurs at bit 11; otherwise cleared to 0.
[4] Set to 1 when a carry or borrow occurs at bit 27; otherwise cleared to 0.
[5] Retains its previous value when the result is zero; otherwise cleared to 0.
[6] Set to 1 when the divisor is negative; otherwise cleared to 0.
[7] Set to 1 when the divisor is zero; otherwise cleared to 0.
[8] Set to 1 when the quotient is negative; otherwise cleared to 0.
[9] One additional state is required for execution when EXR is valid.

A.2 Instruction Codes

Table A-2 shows the instruction codes.

Table A-2 Instruction Codes

Instruction	Mnemonic	Size	Instruction Format									
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
ADD	ADD.B #xx:8,Rd	B	8 rd	IMM								
	ADD.B Rs,Rd	B	0 8	rs rd								
	ADD.W #xx:16,Rd	W	7 9	1 rd	IMM							
	ADD.W Rs,Rd	W	0 9	rs rd								
	ADD.L #xx:32,ERd	L	7 A	1 0 erd	IMM							
	ADD.L ERs,ERd	L	0 A	1 ers 0 erd								
ADDS	ADDS #1,ERd	L	0 B	0 0 erd								
	ADDS #2,ERd	L	0 B	8 0 erd								
	ADDS #4,ERd	L	0 B	9 0 erd								
ADDX	ADDX #xx:8,Rd	B	9 rd	IMM								
	ADDX Rs,Rd	B	0 E	rs rd								
AND	AND.B #xx:8,Rd	B	E rd	IMM								
	AND.B Rs,Rd	B	1 6	rs rd								
	AND.W #xx:16,Rd	W	7 9	6 rd	IMM							
	AND.W Rs,Rd	W	6 6	rs rd								
	AND.L #xx:32,ERd	L	7 A	6 0 erd	IMM							
	AND.L ERs,ERd	L	0 1	F 0	6 6	0 ers 0 erd						
ANDC	ANDC #xx:8,CCR	B	0 6	IMM								
	ANDC #xx:8,EXR	B	0 1	4 1	0 6	IMM						
BAND	BAND #xx:3,Rd	B	7 6	0 IMM rd								
	BAND #xx:3,@ERd	B	7 C	0 erd 0	7 6	0 IMM 0						
	BAND #xx:3,@aa:8	B	7 E	abs	7 6	0 IMM 0						
	BAND #xx:3,@aa:16	B	6 A	1 0	abs		7 6	0 IMM 0				
	BAND #xx:3,@aa:32	B	6 A	3 0	abs				7 6	0 IMM 0		
Bcc	BRA d:8 (BT d:8)	—	4 0	disp								
	BRA d:16 (BT d:16)	—	5 8	0 0	disp							
	BRN d:8 (BF d:8)	—	4 1	disp								
	BRN d:16 (BF d:16)	—	5 8	1 0	disp							

Instruction	Mnemonic	Size	Instruction Format									
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
Bcc	BHI d:8	—	4 2	disp								
	BHI d:16	—	5 8	2 0	disp							
	BLS d:8	—	4 3	disp								
	BLS d:16	—	5 8	3 0	disp							
	BCC d:8 (BHS d:8)	—	4 4	disp								
	BCC d:16 (BHS d:16)	—	5 8	4 0	disp							
	BCS d:8 (BLO d:8)	—	4 5	disp								
	BCS d:16 (BLO d:16)	—	5 8	5 0	disp							
	BNE d:8	—	4 6	disp								
	BNE d:16	—	5 8	6 0	disp							
	BEQ d:8	—	4 7	disp								
	BEQ d:16	—	5 8	7 0	disp							
	BVC d:8	—	4 8	disp								
	BVC d:16	—	5 8	8 0	disp							
	BVS d:8	—	4 9	disp								
	BVS d:16	—	5 8	9 0	disp							
	BPL d:8	—	4 A	disp								
	BPL d:16	—	5 8	A 0	disp							
	BMI d:8	—	4 B	disp								
	BMI d:16	—	5 8	B 0	disp							
	BGE d:8	—	4 C	disp								
	BGE d:16	—	5 8	C 0	disp							
	BLT d:8	—	4 D	disp								
	BLT d:16	—	5 8	D 0	disp							
	BGT d:8	—	4 E	disp								
	BGT d:16	—	5 8	E 0	disp							
	BLE d:8	—	4 F	disp								
	BLE d:16	—	5 8	F 0	disp							

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
BCLR	BCLR #xx:3,Rd	B	7	2	0 IMM	rd														
	BCLR #xx:3,@ERd	B	7	D	0 erd	0	7	2	0 IMM	0										
	BCLR #xx:3,@aa:8	B	7	F	abs		7	2	0 IMM	0										
	BCLR #xx:3,@aa:16	B	6	A	1	8	abs				7	2	0 IMM	0						
	BCLR #xx:3,@aa:32	B	6	A	3	8	abs								7	2	0 IMM	0		
	BCLR Rn,Rd	B	6	2	rn	rd														
	BCLR Rn,@ERd	B	7	D	0 erd	0	6	2	rn	0										
	BCLR Rn,@aa:8	B	7	F	abs		6	2	rn	0										
	BCLR Rn,@aa:16	B	6	A	1	8	abs				6	2	rn	0						
	BCLR Rn,@aa:32	B	6	A	3	8	abs								6	2	rn	0		
BIAND	BIAND #xx:3,Rd	B	7	6	1 IMM	rd														
	BIAND #xx:3,@ERd	B	7	C	0 erd	0	7	6	1 IMM	0										
	BIAND #xx:3,@aa:8	B	7	E	abs		7	6	1 IMM	0										
	BIAND #xx:3,@aa:16	B	6	A	1	0	abs				7	6	1 IMM	0						
	BIAND #xx:3,@aa:32	B	6	A	3	0	abs								7	6	1 IMM	0		
BILD	BILD #xx:3,Rd	B	7	7	1 IMM	rd														
	BILD #xx:3,@ERd	B	7	C	0 erd	0	7	7	1 IMM	0										
	BILD #xx:3,@aa:8	B	7	E	abs		7	7	1 IMM	0										
	BILD #xx:3,@aa:16	B	6	A	1	0	abs				7	7	1 IMM	0						
	BILD #xx:3,@aa:32	B	6	A	3	0	abs								7	7	1 IMM	0		
BIOR	BIOR #xx:3,Rd	B	7	4	1 IMM	rd														
	BIOR #xx:3,@ERd	B	7	C	0 erd	0	7	4	1 IMM	0										
	BIOR #xx:3,@aa:8	B	7	E	abs		7	4	1 IMM	0										
	BIOR #xx:3,@aa:16	B	6	A	1	0	abs				7	4	1 IMM	0						
	BIOR #xx:3,@aa:32	B	6	A	3	0	abs								7	4	1 IMM	0		

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
BIST	BIST #xx:3,Rd	B	6	7	1 IMM	rd														
	BIST #xx:3,@ERd	B	7	D	0 erd	0	6	7	1 IMM	0										
	BIST #xx:3,@aa:8	B	7	F	abs		6	7	1 IMM	0										
	BIST #xx:3,@aa:16	B	6	A	1	8	abs				6	7	1 IMM	0						
	BIST #xx:3,@aa:32	B	6	A	3	8	abs								6	7	1 IMM	0		
BIXOR	BIXOR #xx:3,Rd	B	7	5	1 IMM	rd														
	BIXOR #xx:3,@ERd	B	7	C	0 erd	0	7	5	1 IMM	0										
	BIXOR #xx:3,@aa:8	B	7	E	abs		7	5	1 IMM	0										
	BIXOR #xx:3,@aa:16	B	6	A	1	0	abs				7	5	1 IMM	0						
	BIXOR #xx:3,@aa:32	B	6	A	3	0	abs								7	5	1 IMM	0		
BLD	BLD #xx:3,Rd	B	7	7	0 IMM	rd														
	BLD #xx:3,@ERd	B	7	C	0 erd	0	7	7	0 IMM	0										
	BLD #xx:3,@aa:8	B	7	E	abs		7	7	0 IMM	0										
	BLD #xx:3,@aa:16	B	6	A	1	0	abs				7	7	0 IMM	0						
	BLD #xx:3,@aa:32	B	6	A	3	0	abs								7	7	0 IMM	0		
BNOT	BNOT #xx:3,Rd	B	7	1	0 IMM	rd														
	BNOT #xx:3,@ERd	B	7	D	0 erd	0	7	1	0 IMM	0										
	BNOT #xx:3,@aa:8	B	7	F	abs		7	1	0 IMM	0										
	BNOT #xx:3,@aa:16	B	6	A	1	8	abs				7	1	0 IMM	0						
	BNOT #xx:3,@aa:32	B	6	A	3	8	abs								7	1	0 IMM	0		
	BNOT Rn,Rd	B	6	1	rn	rd														
	BNOT Rn,@ERd	B	7	D	0 erd	0	6	1	rn	0										
	BNOT Rn,@aa:8	B	7	F	abs		6	1	rn	0										
	BNOT Rn,@aa:16	B	6	A	1	8	abs				6	1	rn	0						
	BNOT Rn,@aa:32	B	6	A	3	8	abs								6	1	rn	0		

Instruction	Mnemonic	Size	Instruction Format																			
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte		10th byte	
BOR	BOR #xx:3,Rd	B	7	4	0 IMM	rd																
	BOR #xx:3,@ERd	B	7	C	0 erd	0	7	4	0 IMM	0												
	BOR #xx:3,@aa:8	B	7	E	abs		7	4	0 IMM	0												
	BOR #xx:3,@aa:16	B	6	A	1	0	abs				7	4	0 IMM	0								
	BOR #xx:3,@aa:32	B	6	A	3	0	abs								7	4	0 IMM	0				
BSET	BSET #xx:3,Rd	B	7	0	0 IMM	rd																
	BSET #xx:3,@ERd	B	7	D	0 erd	0	7	0	0 IMM	0												
	BSET #xx:3,@aa:8	B	7	F	abs		7	0	0 IMM	0												
	BSET #xx:3,@aa:16	B	6	A	1	8	abs				7	0	0 IMM	0								
	BSET #xx:3,@aa:32	B	6	A	3	8	abs								7	0	0 IMM	0				
	BSET Rn,Rd	B	6	0	rn	rd																
	BSET Rn,@ERd	B	7	D	0 erd	0	6	0	rn	0												
	BSET Rn,@aa:8	B	7	F	abs		6	0	rn	0												
	BSET Rn,@aa:16	B	6	A	1	8	abs				6	0	rn	0								
	BSET Rn,@aa:32	B	6	A	3	8	abs								6	0	rn	0				
BSR	BSR d:8	—	5	5	disp																	
	BSR d:16	—	5	C	0	0	disp															
BST	BST #xx:3,Rd	B	6	7	0 IMM	rd																
	BST #xx:3,@ERd	B	7	D	0 erd	0	6	7	0 IMM	0												
	BST #xx:3,@aa:8	B	7	F	abs		6	7	0 IMM	0												
	BST #xx:3,@aa:16	B	6	A	1	8	abs				6	7	0 IMM	0								
	BST #xx:3,@aa:32	B	6	A	3	8	abs								6	7	0 IMM	0				
BTST	BTST #xx:3,Rd	B	7	3	0 IMM	rd																
	BTST #xx:3,@ERd	B	7	C	0 erd	0	7	3	0 IMM	0												
	BTST #xx:3,@aa:8	B	7	E	abs		7	3	0 IMM	0												
	BTST #xx:3,@aa:16	B	6	A	1	0	abs				7	3	0 IMM	0								
	BTST #xx:3,@aa:32	B	6	A	3	0	abs								7	3	0 IMM	0				
	BTST Rn,Rd	B	6	3	rn	rd																
	BTST Rn,@ERd	B	7	C	0 erd	0	6	3	rn	0												

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
BTST	BTST Rn,@aa:8	B	7	E	abs		6	3	rn	0										
	BTST Rn,@aa:16	B	6	A	1	0	abs				6	3	rn	0						
	BTST Rn,@aa:32	B	6	A	3	0	abs								6	3	rn	0		
BXOR	BXOR #xx:3,Rd	B	7	5	0	IMM	rd													
	BXOR #xx:3,@ERd	B	7	C	0	erd	0	7	5	0	IMM	0								
	BXOR #xx:3,@aa:8	B	7	E	abs		7	5	0	IMM	0									
	BXOR #xx:3,@aa:16	B	6	A	1	0	abs				7	5	0	IMM	0					
	BXOR #xx:3,@aa:32	B	6	A	3	0	abs								7	5	0	IMM	0	
CLRMAC	CLRMAC	—	0	1	A	0														
CMP	CMP.B #xx:8,Rd	B	A	rd	IMM															
	CMP.B Rs,Rd	B	1	C	rs	rd														
	CMP.W #xx:16,Rd	W	7	9	2	rd	IMM													
	CMP.W Rs,Rd	W	1	D	rs	rd														
	CMP.L #xx:32,ERd	L	7	A	2	0	erd	IMM												
	CMP.L ERs,ERd	L	1	F	1	ers	0	erd												
DAA	DAA Rd	B	0	F	0	rd														
DAS	DAS Rd	B	1	F	0	rd														
DEC	DEC.B Rd	B	1	A	0	rd														
	DEC.W #1,Rd	W	1	B	5	rd														
	DEC.W #2,Rd	W	1	B	D	rd														
	DEC.L #1,ERd	L	1	B	7	0	erd													
	DEC.L #2,ERd	L	1	B	F	0	erd													
DIVXS	DIVXS.B Rs,Rd	B	0	1	D	0	5	1	rs	rd										
	DIVXS.W Rs,ERd	W	0	1	D	0	5	3	rs	0	erd									
DIVXU	DIVXU.B Rs,Rd	B	5	1	rs	rd														
	DIVXU.W Rs,ERd	W	5	3	rs	0	erd													
EEPMOV	EEPMOV.B	—	7	B	5	C	5	9	8	F										
	EEPMOV.W	—	7	B	D	4	5	9	8	F										

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
EXTS	EXTS.W Rd	W	1	7	D	rd														
	EXTS.L ERd	L	1	7	F	0 erd														
EXTU	EXTU.W Rd	W	1	7	5	rd														
	EXTU.L ERd	L	1	7	7	0 erd														
INC	INC.B Rd	B	0	A	0	rd														
	INC.W #1,Rd	W	0	B	5	rd														
	INC.W #2,Rd	W	0	B	D	rd														
	INC.L #1,ERd	L	0	B	7	0 erd														
	INC.L #2,ERd	L	0	B	F	0 erd														
JMP	JMP @ERn	—	5	9	0 ern	0														
	JMP @aa:24	—	5	A	abs															
	JMP @@aa:8	—	5	B	abs															
JSR	JSR @ERn	—	5	D	0 ern	0														
	JSR @aa:24	—	5	E	abs															
	JSR @@aa:8	—	5	F	abs															
LDC	LDC #xx:8,CCR	B	0	7	IMM															
	LDC #xx:8,EXR	B	0	1	4	1	0	7	IMM											
	LDC Rs,CCR	B	0	3	0	rs														
	LDC Rs,EXR	B	0	3	1	rs														
	LDC @ERs,CCR	W	0	1	4	0	6	9	0 ers	0										
	LDC @ERs,EXR	W	0	1	4	1	6	9	0 ers	0										
	LDC @(d:16,ERs),CCR	W	0	1	4	0	6	F	0 ers	0	disp									
	LDC @(d:16,ERs),EXR	W	0	1	4	1	6	F	0 ers	0	disp									
	LDC @(d:32,ERs),CCR	W	0	1	4	0	7	8	0 ers	0	6	B	2	0	disp					
	LDC @(d:32,ERs),EXR	W	0	1	4	1	7	8	0 ers	0	6	B	2	0	disp					
	LDC @ERs+,CCR	W	0	1	4	0	6	D	0 ers	0										
	LDC @ERs+,EXR	W	0	1	4	1	6	D	0 ers	0										
	LDC @aa:16,CCR	W	0	1	4	0	6	B	0	0	abs									
	LDC @aa:16,EXR	W	0	1	4	1	6	B	0	0	abs									

Instruction	Mnemonic	Size	Instruction Format															
			1st byte		2nd byte		3rd byte		4th byte		5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
LDC	LDC @aa:32,CCR	W	0	1	4	0	6	B	2	0	abs					
	LDC @aa:32,EXR	W	0	1	4	1	6	B	2	0	abs					
LDM*3	LDM.L @SP+, (ERn-ERn+1)	L	0	1	1	0	6	D	7	0 ern+1						
	LDM.L @SP+, (ERn-ERn+2)	L	0	1	2	0	6	D	7	0 ern+2						
	LDM.L @SP+, (ERn-ERn+3)	L	0	1	3	0	6	D	7	0 ern+3						
LDMAC	LDMAC ERs,MACH	L	0	3	2	0 ers										
	LDMAC ERs,MACL	L	0	3	3	0 ers										
MAC	MAC @ERn+,@ERm+	—	0	1	6	0	6	D	0 ern	0 erm						
MOV	MOV.B #xx:8,Rd	B	F	rd	IMM											
	MOV.B Rs,Rd	B	0	C	rs	rd										
	MOV.B @ERs,Rd	B	6	8	0 ers	rd										
	MOV.B @(d:16,ERs),Rd	B	6	E	0 ers	rd	disp									
	MOV.B @(d:32,ERs),Rd	B	7	8	0 ers	0	6	A	2	rd	disp					
	MOV.B @ERs+,Rd	B	6	C	0 ers	rd										
	MOV.B @aa:8,Rd	B	2	rd	abs											
	MOV.B @aa:16,Rd	B	6	A	0	rd	abs									
	MOV.B @aa:32,Rd	B	6	A	2	rd	abs									
	MOV.B Rs,@ERd	B	6	8	1 erd	rs										
	MOV.B Rs,@(d:16,ERd)	B	6	E	1 erd	rs	disp									
	MOV.B Rs,@(d:32,ERd)	B	7	8	0 erd	0	6	A	A	rs	disp					
	MOV.B Rs,@-ERd	B	6	C	1 erd	rs										
	MOV.B Rs,@aa:8	B	3	rs	abs											
	MOV.B Rs,@aa :16	B	6	A	8	rs	abs									
	MOV.B Rs,@aa:32	B	6	A	A	rs	abs									
	MOV.W #xx:16,Rd	W	7	9	0	rd	IMM									
	MOV.W Rs,Rd	W	0	D	rs	rd										
	MOV.W @ERs,Rd	W	6	9	0 ers	rd										
	MOV.W @(d:16,ERs),Rd	W	6	F	0 ers	rd	disp									
	MOV.W @(d:32,ERs),Rd	W	7	8	0 ers	0	6	B	2	rd	disp					

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte	10th byte
MOV	MOV.W @ERs+,Rd	W	6	D	0	ers	rd													
	MOV.W @aa:16,Rd	W	6	B	0	rd	abs													
	MOV.W @aa:32,Rd	W	6	B	2	rd	abs													
	MOV.W Rs,@ERd	W	6	9	1	erd	rs													
	MOV.W Rs,@(d:16,ERd)	W	6	F	1	erd	rs	disp												
	MOV.W Rs,@(d:32,ERd)	W	7	8	0	erd	0	6	B	A	rs	disp								
	MOV.W Rs,@-ERd	W	6	D	1	erd	rs													
	MOV.W Rs,@aa:16	W	6	B	8	rs	abs													
	MOV.W Rs,@aa:32	W	6	B	A	rs	abs													
	MOV.L #xx:32,Rd	L	7	A	0	0	erd	IMM												
	MOV.L ERs,ERd	L	0	F	1	ers	0	erd												
	MOV.L @ERs,ERd	L	0	1	0	0	6	9	0	ers	0	erd								
	MOV.L @(d:16,ERs),ERd	L	0	1	0	0	6	F	0	ers	0	erd	disp							
	MOV.L @(d:32,ERs),ERd	L	0	1	0	0	7	8	0	ers	0	6	B	2	0	erd	disp			
	MOV.L @ERs+,ERd	L	0	1	0	0	6	D	0	ers	0	erd								
	MOV.L @aa:16 ,ERd	L	0	1	0	0	6	B	0	0	erd	abs								
	MOV.L @aa:32 ,ERd	L	0	1	0	0	6	B	2	0	erd	abs								
	MOV.L ERs,@ERd	L	0	1	0	0	6	9	1	erd	0	ers								
	MOV.L ERs,@(d:16,ERd)	L	0	1	0	0	6	F	1	erd	0	ers	disp							
	MOV.L ERs,@(d:32,ERd)*[1]	L	0	1	0	0	7	8	0	erd	0	6	B	A	0	ers	disp			
	MOV.L ERs,@-ERd	L	0	1	0	0	6	D	1	erd	0	ers								
	MOV.L ERs,@aa:16	L	0	1	0	0	6	B	8	0	ers	abs								
	MOV.L ERs,@aa:32	L	0	1	0	0	6	B	A	0	ers	abs								
MOVFPE	MOVFPE @aa:16,Rd	B	Cannot be used in this LSI																	
MOVTPE	MOVTPE Rs,@aa:16	B																		
MULXS	MULXS.B Rs,Rd	B	0	1	C	0	5	0	rs	rd										
	MULXS.W Rs,ERd	W	0	1	C	0	5	2	rs	0	erd									
MULXU	MULXU.B Rs,Rd	B	5	0	rs	rd														
	MULXU.W Rs,ERd	W	5	2	rs	0	erd													

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte		9th byte		10th byte	
NEG	NEG.B Rd	B	1	7	8	rd														
	NEG.W Rd	W	1	7	9	rd														
	NEG.L ERd	L	1	7	B	0 erd														
NOP	NOP	—	0	0	0	0														
NOT	NOT.B Rd	B	1	7	0	rd														
	NOT.W Rd	W	1	7	1	rd														
	NOT.L ERd	L	1	7	3	0 erd														
OR	OR.B #xx:8,Rd	B	C	rd	IMM															
	OR.B Rs,Rd	B	1	4	rs	rd														
	OR.W #xx:16,Rd	W	7	9	4	rd	IMM													
	OR.W Rs,Rd	W	6	4	rs	rd														
	OR.L #xx:32,ERd	L	7	A	4	0 erd	IMM													
	OR.L ERs,ERd	L	0	1	F	0	6	4	0 ers		0 erd									
ORC	ORC #xx:8,CCR	B	0	4	IMM															
	ORC #xx:8,EXR	B	0	1	4	1	0	4	IMM											
POP	POP.W Rn	W	6	D	7	rn														
	POP.L ERn	L	0	1	0	0	6	D	7	0 ern										
PUSH	PUSH.W Rn	W	6	D	F	rn														
	PUSH.L ERn	L	0	1	0	0	6	D	F	0 ern										
ROTL	ROTL.B Rd	B	1	2	8	rd														
	ROTL.B #2, Rd	B	1	2	C	rd														
	ROTL.W Rd	W	1	2	9	rd														
	ROTL.W #2, Rd	W	1	2	D	rd														
	ROTL.L ERd	L	1	2	B	0 erd														
	ROTL.L #2, ERd	L	1	2	F	0 erd														

Instruction	Mnemonic	Size	Instruction Format											
			1st byte		2nd byte		3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
ROTR	ROTR.B Rd	B	1	3	8	rd								
	ROTR.B #2, Rd	B	1	3	C	rd								
	ROTR.W Rd	W	1	3	9	rd								
	ROTR.W #2, Rd	W	1	3	D	rd								
	ROTR.L ERd	L	1	3	B	0 erd								
	ROTR.L #2, ERd	L	1	3	F	0 erd								
ROTXL	ROTXL.B Rd	B	1	2	0	rd								
	ROTXL.B #2, Rd	B	1	2	4	rd								
	ROTXL.W Rd	W	1	2	1	rd								
	ROTXL.W #2, Rd	W	1	2	5	rd								
	ROTXL.L ERd	L	1	2	3	0 erd								
	ROTXL.L #2, ERd	L	1	2	7	0 erd								
ROTXR	ROTXR.B Rd	B	1	3	0	rd								
	ROTXR.B #2, Rd	B	1	3	4	rd								
	ROTXR.W Rd	W	1	3	1	rd								
	ROTXR.W #2, Rd	W	1	3	5	rd								
	ROTXR.L ERd	L	1	3	3	0 erd								
	ROTXR.L #2, ERd	L	1	3	7	0 erd								
RTE	RTE	—	5	6	7	0								
RTS	RTS	—	5	4	7	0								
SHAL	SHAL.B Rd	B	1	0	8	rd								
	SHAL.B #2, Rd	B	1	0	C	rd								
	SHAL.W Rd	W	1	0	9	rd								
	SHAL.W #2, Rd	W	1	0	D	rd								
	SHAL.L ERd	L	1	0	B	0 erd								
	SHAL.L #2, ERd	L	1	0	F	0 erd								

Instruction	Mnemonic	Size	Instruction Format																	
			1st byte		2nd byte		3rd byte		4th byte			5th byte				6th byte			7th byte	8th byte	9th byte	10th byte
SHAR	SHAR.B Rd	B	1	1	8	rd																
	SHAR.B #2, Rd	B	1	1	C	rd																
	SHAR.W Rd	W	1	1	9	rd																
	SHAR.W #2, Rd	W	1	1	D	rd																
	SHAR.L ERd	L	1	1	B	0 erd																
	SHAR.L #2, ERd	L	1	1	F	0 erd																
SHLL	SHLL.B Rd	B	1	0	0	rd																
	SHLL.B #2, Rd	B	1	0	4	rd																
	SHLL.W Rd	W	1	0	1	rd																
	SHLL.W #2, Rd	W	1	0	5	rd																
	SHLL.L ERd	L	1	0	3	0 erd																
	SHLL.L #2, ERd	L	1	0	7	0 erd																
SHLR	SHLR.B Rd	B	1	1	0	rd																
	SHLR.B #2, Rd	B	1	1	4	rd																
	SHLR.W Rd	W	1	1	1	rd																
	SHLR.W #2, Rd	W	1	1	5	rd																
	SHLR.L ERd	L	1	1	3	0 erd																
	SHLR.L #2, ERd	L	1	1	7	0 erd																
SLEEP	SLEEP	—	0	1	8	0																
STC	STC.B CCR,Rd	B	0	2	0	rd																
	STC.B EXR,Rd	B	0	2	1	rd																
	STC.W CCR,@ERd	W	0	1	4	0	6	9	1 erd	0												
	STC.W EXR,@ERd	W	0	1	4	1	6	9	1 erd	0												
	STC.W CCR,@(d:16,ERd)	W	0	1	4	0	6	F	1 erd	0	disp											
	STC.W EXR,@(d:16,ERd)	W	0	1	4	1	6	F	1 erd	0	disp											
	STC.W CCR,@(d:32,ERd)	W	0	1	4	0	7	8	0 erd	0	6	B	A	0	disp							
	STC.W EXR,@(d:32,ERd)	W	0	1	4	1	7	8	0 erd	0	6	B	A	0	disp							
	STC.W CCR,@-ERd	W	0	1	4	0	6	D	1 erd	0												
	STC.W EXR,@-ERd	W	0	1	4	1	6	D	1 erd	0												

Instruction	Mnemonic	Size	Instruction Format															
			1st byte		2nd byte		3rd byte		4th byte		5th byte	6th byte	7th byte	8th byte	9th byte	10th byte		
STC	STC.W CCR,@aa:16	W	0	1	4	0	6	B	8	0	abs							
	STC.W EXR,@aa:16	W	0	1	4	1	6	B	8	0	abs							
	STC.W CCR,@aa:32	W	0	1	4	0	6	B	A	0	abs							
	STC.W EXR,@aa:32	W	0	1	4	1	6	B	A	0	abs							
STM*3	STM.L(ERn-ERn+1), @-SP	L	0	1	1	0	6	D	F	0 ern								
	STM.L (ERn-ERn+2), @-SP	L	0	1	2	0	6	D	F	0 ern								
	STM.L (ERn-ERn+3), @-SP	L	0	1	3	0	6	D	F	0 ern								
STMAC	STMAC MACH,ERd	L	0	2	2	0 ers												
	STMAC MACL,ERd	L	0	2	3	0 ers												
SUB	SUB.B Rs,Rd	B	1	8	rs	rd												
	SUB.W #xx:16,Rd	W	7	9	3	rd	IMM											
	SUB.W Rs,Rd	W	1	9	rs	rd												
	SUB.L #xx:32,ERd	L	7	A	3	0 erd	IMM											
	SUB.L ERs,ERd	L	1	A	1 ers	0 erd												
SUBS	SUBS #1,ERd	L	1	B	0	0 erd												
	SUBS #2,ERd	L	1	B	8	0 erd												
	SUBS #4,ERd	L	1	B	9	0 erd												
SUBX	SUBX #xx:8,Rd	B	B	rd	IMM													
	SUBX Rs,Rd	B	1	E	rs	rd												
TAS	TAS @ERd*2	B	0	1	E	0	7	B	0 erd	C								
TRAPA	TRAPA #x:2	—	5	7	00 IMM	0												
XOR	XOR.B #xx:8,Rd	B	D	rd	IMM													
	XOR.B Rs,Rd	B	1	5	rs	rd												
	XOR.W #xx:16,Rd	W	7	9	5	rd	IMM											
	XOR.W Rs,Rd	W	6	5	rs	rd												
	XOR.L #xx:32,ERd	L	7	A	5	0 erd	IMM											
	XOR.L ERs,ERd	L	0	1	F	0	6	5	0 ers	0 erd								

Instruction	Mnemonic	Size	Instruction Format									
			1st byte	2nd byte	3rd byte	4th byte	5th byte	6th byte	7th byte	8th byte	9th byte	10th byte
XORC	XORC #xx:8,CCR	B	0 5	IMM								
	XORC #xx:8,EXR	B	0 1	4 1	0 5	IMM						

Notes: *1 Bit 7 of the 4th byte of the MOV.L ERs, @(d:32,ERd) instruction can be either 1 or 0.
*2 Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.
*3 Only registers ER0 to ER6 should be used when using the STM/LDM instruction.

Legend

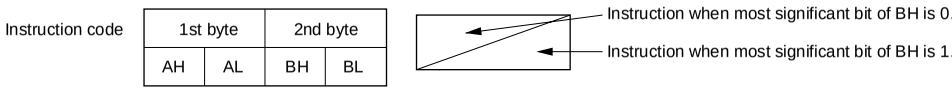
IMM:	Immediate data (2, 3, 8, 16, or 32 bits)
abs:	Absolute address (8, 16, 24, or 32 bits)
disp:	Displacement (8, 16, or 32 bits)
rs, rd, rn:	Register field (4 bits specifying an 8-bit or 16-bit register. The symbols rs, rd, and rn correspond to operand symbols Rs, Rd,and Rn.)
ers, erd, ern, erm:	Register field (3 bits specifying an address register or 32-bit register. The symbols ers, erd, ern, and erm correspond to operand symbols ERs, ERd, ERn, and ERm.)

The register fields specify general registers as follows.

Address Register 32-Bit Register		16-Bit Register		8-Bit Register	
Register Field	General Register	Register Field	General Register	Register Field	General Register
000	ER0	0000	R0	0000	R0H
001	ER1	0001	R1	0001	R1H
•	•	•	•	•	•
•	•	•	•	•	•
•	•	•	•	•	•
111	ER7	0111	R7	0111	R7H
		1000	E0	1000	R0L
		1001	E1	1001	R1L
		•	•	•	•
		•	•	•	•
		•	•	•	•
		1111	E7	1111	R7L

Table A-3 shows the operation code map.

Table A-3 Operation Code Map (1)



AH \ AL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	NOP	Table A.3(2)	STC / STMAC	LDC / LDMAC	ORC	XORC	ANDC	LDC	ADD		Table A.3(2)	Table A.3(2)	MOV		ADDX	Table A.3(2)
1	Table A.3(2)	Table A.3(2)	Table A.3(2)	Table A.3(2)	OR	XOR	AND	Table A.3(2)	SUB		Table A.3(2)	Table A.3(2)	CMP		SUBX	Table A.3(2)
2	MOV.B															
3																
4	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
5	MULXU	DIVXU	MULXU	DIVXU	RTS	BSR	RTE	TRAPA	Table A.3(2)	JMP			BSR	JSR		
6	BSET	BNOT	BCLR	BTST	OR	XOR	AND	BST / BIST	MOV		Table A.3(2)	MOV				
7					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD	MOV	Table A.3(2)	Table A.3(2)	EEPMOV	Table A.3(3)			
8	ADD															
9	ADDX															
A	CMP															
B	SUBX															
C	OR															
D	XOR															
E	AND															
F	MOV															

Table A-3 Operation Code Map (2)

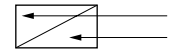
Instruction code	1st byte		2nd byte	
	AH	AL	BH	BL

AH AL \ BH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01	MOV	LDM / STM			LDC / STC		MAC		SLEEP		CLRMAC		Table A.3(3)	Table A.3(3)	TAS	Table A.3(3)
0A	INC								ADD							
0B	ADDS					INC		INC	ADDS					INC		INC
0F	DAA								MOV							
10	SHLL			SHLL				SHLL	SHAL			SHAL				SHAL
11	SHLR			SHLR				SHLR	SHAR			SHAR				SHAR
12	ROTXL			ROTXL				ROTXL	ROTL			ROTL				ROTL
13	ROTXR			ROTXR				ROTXR	ROTR			ROTR				ROTR
17	NOT			NOT		EXTU		EXTU	NEG			NEG		EXTS		EXTS
1A	DEC								SUB							
1B	SUBS					DEC		DEC	SUBS					DEC		DEC
1F	DAS								CMP							
58	BRA	BRN	BHI	BLS	BCC	BCS	BNE	BEQ	BVC	BVS	BPL	BMI	BGE	BLT	BGT	BLE
6A	MOV	Table A.3(4)	MOV	Table A.3(4)	MOVFPE*				MOV		MOV		MOVTPE*			
79	MOV	ADD	CMP	SUB	OR	XOR	AND									
7A	MOV	ADD	CMP	SUB	OR	XOR	AND									

Note: * Cannot be used in this LSI.

Table A-3 Operation Code Map (3)

Instruction code	1st byte		2nd byte		3rd byte		4th byte	
	AH	AL	BH	BL	CH	CL	DH	DL



AH AL BH BL CH \ CL	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
01C05	MULXS		MULXS													
01D05		DIVXS		DIVXS												
01F06					OR	XOR	AND									
7Cr06 *1				BTST												
7Cr07 *1				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
7Dr06 *1	BSET	BNOT	BCLR					BST / BIST								
7Dr07 *1	BSET	BNOT	BCLR													
7Eaa6 *2				BTST												
7Eaa7 *2				BTST	BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
7Faa6 *2	BSET	BNOT	BCLR					BST / BIST								
7Faa7 *2	BSET	BNOT	BCLR													

Notes: *1 r is the register specification field.
*2 aa is the absolute address specification.

Table A-3 Operation Code Map (4)

Instruction code	1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte	
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL

Instruction when most significant bit of FH is 0.
Instruction when most significant bit of FH is 1.

EL / AHALBHBLCHCLDHDLEH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6A10aaaa6*				BTST												
6A10aaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
6A18aaaa6*	BSET	BNOT	BCLR					BST / BIST								
6A18aaaa7*																

Instruction code	1st byte		2nd byte		3rd byte		4th byte		5th byte		6th byte		7th byte		8th byte	
	AH	AL	BH	BL	CH	CL	DH	DL	EH	EL	FH	FL	GH	GL	HH	HL

Instruction when most significant bit of HH is 0.
Instruction when most significant bit of HH is 1.

GL / AHALBHBL ... FHFLGH	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
6A30aaaaaaaa6*				BTST												
6A30aaaaaaaa7*					BOR / BIOR	BXOR / BIXOR	BAND / BIAND	BLD / BILD								
6A38aaaaaaaa6*	BSET	BNOT	BCLR					BST / BIST								
6A38aaaaaaaa7*																

Note: * aa is the absolute address specification.

A.4 Number of States Required for Instruction Execution

The tables in this section can be used to calculate the number of states required for instruction execution by the CPU. Table A-5 indicates the number of instruction fetch, data read/write, and other cycles occurring in each instruction. Table A-4 indicates the number of states required for each cycle. The number of states required for execution of an instruction can be calculated from these two tables as follows:

$$\text{Execution states} = I \times S_I + J \times S_J + K \times S_K + L \times S_L + M \times S_M + N \times S_N$$

Examples: Advanced mode, program code and stack located in external memory, on-chip supporting modules accessed in two states with 8-bit bus width, external devices accessed in three states with one wait state and 16-bit bus width.

1. BSET #0, @FFFFC7:8
   From table A-5:

   $I = L = 2, \quad J = K = M = N = 0$

   From table A-4:

   $S_I = 4, \quad S_L = 2$

   Number of states required for execution = $2 \times 4 + 2 \times 2 = 12$

2. JSR @@30
   From table A-5:

   $I = J = K = 2, \quad L = M = N = 0$

   From table A-4:

   $S_I = S_J = S_K = 4$

   Number of states required for execution = $2 \times 4 + 2 \times 4 + 2 \times 4 = 24$

Table A-4 Number of States per Cycle

		Access Conditions					
		On-Chip Supporting Module		External Device			
				8-Bit Bus		16-Bit Bus	
Cycle		On-Chip Memory	8-Bit Bus	16-Bit Bus	2-State Access	3-State Access	2-State Access	3-State Access
Instruction fetch	S_I	1	4	2	4	6 + 2m	2	3 + m
Branch address read	S_J							
Stack operation	S_K							
Byte data access	S_L		2		2	3 + m		
Word data access	S_M		4		4	6 + 2m		
Internal operation	S_N	1	1	1	1	1	1	1

Legend

m: Number of wait states inserted into external device access

Table A-5 Number of Cycles in Instruction Execution

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ADD	ADD.B #xx:8,Rd	1					
	ADD.B Rs,Rd	1					
	ADD.W #xx:16,Rd	2					
	ADD.W Rs,Rd	1					
	ADD.L #xx:32,ERd	3					
	ADD.L ERs,ERd	1					
ADDS	ADDS #1/2/4,ERd	1					
ADDX	ADDX #xx:8,Rd	1					
	ADDX Rs,Rd	1					
AND	AND.B #xx:8,Rd	1					
	AND.B Rs,Rd	1					
	AND.W #xx:16,Rd	2					
	AND.W Rs,Rd	1					
	AND.L #xx:32,ERd	3					
	AND.L ERs,ERd	2					
ANDC	ANDC #xx:8,CCR	1					
	ANDC #xx:8,EXR	2					
BAND	BAND #xx:3,Rd	1					
	BAND #xx:3,@ERd	2			1		
	BAND #xx:3,@aa:8	2			1		
	BAND #xx:3,@aa:16	3			1		
	BAND #xx:3,@aa:32	4			1		
Bcc	BRA d:8 (BT d:8)	2					
	BRN d:8 (BF d:8)	2					
	BHI d:8	2					
	BLS d:8	2					
	BCC d:8 (BHS d:8)	2					
	BCS d:8 (BLO d:8)	2					
	BNE d:8	2					
	BEQ d:8	2					
	BVC d:8	2					
	BVS d:8	2					
	BPL d:8	2					

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
Bcc	BMI d:8	2					
	BGE d:8	2					
	BLT d:8	2					
	BGT d:8	2					
	BLE d:8	2					
	BRA d:16 (BT d:16)	2					1
	BRN d:16 (BF d:16)	2					1
	BHI d:16	2					1
	BLS d:16	2					1
	BCC d:16 (BHS d:16)	2					1
	BCS d:16 (BLO d:16)	2					1
	BNE d:16	2					1
	BEQ d:16	2					1
	BVC d:16	2					1
	BVS d:16	2					1
	BPL d:16	2					1
	BMI d:16	2					1
	BGE d:16	2					1
	BLT d:16	2					1
	BGT d:16	2					1
	BLE d:16	2					1
BCLR	BCLR #xx:3,Rd	1					
	BCLR #xx:3,@ERd	2			2		
	BCLR #xx:3,@aa:8	2			2		
	BCLR #xx:3,@aa:16	3			2		
	BCLR #xx:3,@aa:32	4			2		
	BCLR Rn,Rd	1					
	BCLR Rn,@ERd	2			2		
	BCLR Rn,@aa:8	2			2		
	BCLR Rn,@aa:16	3			2		
	BCLR Rn,@aa:32	4			2		

Instruction	Mnemonic	Instruction Fetch	Branch Address Read	Stack Operation	Byte Data Access	Word Data Access	Internal Operation
		I	J	K	L	M	N
BIAND	BIAND #xx:3,Rd	1					
	BIAND #xx:3,@ERd	2			1		
	BIAND #xx:3,@aa:8	2			1		
	BIAND #xx:3,@aa:16	3			1		
	BIAND #xx:3,@aa:32	4			1		
BILD	BILD #xx:3,Rd	1					
	BILD #xx:3,@ERd	2			1		
	BILD #xx:3,@aa:8	2			1		
	BILD #xx:3,@aa:16	3			1		
	BILD #xx:3,@aa:32	4			1		
BIOR	BIOR #xx:8,Rd	1					
	BIOR #xx:8,@ERd	2			1		
	BIOR #xx:8,@aa:8	2			1		
	BIOR #xx:8,@aa:16	3			1		
	BIOR #xx:8,@aa:32	4			1		
BIST	BIST #xx:3,Rd	1					
	BIST #xx:3,@ERd	2			2		
	BIST #xx:3,@aa:8	2			2		
	BIST #xx:3,@aa:16	3			2		
	BIST #xx:3,@aa:32	4			2		
BIXOR	BIXOR #xx:3,Rd	1					
	BIXOR #xx:3,@ERd	2			1		
	BIXOR #xx:3,@aa:8	2			1		
	BIXOR #xx:3,@aa:16	3			1		
	BIXOR #xx:3,@aa:32	4			1		
BLD	BLD #xx:3,Rd	1					
	BLD #xx:3,@ERd	2			1		
	BLD #xx:3,@aa:8	2			1		
	BLD #xx:3,@aa:16	3			1		
	BLD #xx:3,@aa:32	4			1		

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BNOT	BNOT #xx:3,Rd	1					
	BNOT #xx:3,@ERd	2			2		
	BNOT #xx:3,@aa:8	2			2		
	BNOT #xx:3,@aa:16	3			2		
	BNOT #xx:3,@aa:32	4			2		
	BNOT Rn,Rd	1					
	BNOT Rn,@ERd	2			2		
	BNOT Rn,@aa:8	2			2		
	BNOT Rn,@aa:16	3			2		
	BNOT Rn,@aa:32	4			2		
BOR	BOR #xx:3,Rd	1					
	BOR #xx:3,@ERd	2			1		
	BOR #xx:3,@aa:8	2			1		
	BOR #xx:3,@aa:16	3			1		
	BOR #xx:3,@aa:32	4			1		
BSET	BSET #xx:3,Rd	1					
	BSET #xx:3,@ERd	2			2		
	BSET #xx:3,@aa:8	2			2		
	BSET #xx:3,@aa:16	3			2		
	BSET #xx:3,@aa:32	4			2		
	BSET Rn,Rd	1					
	BSET Rn,@ERd	2			2		
	BSET Rn,@aa:8	2			2		
	BSET Rn,@aa:16	3			2		
	BSET Rn,@aa:32	4			2		
BSR	BSR d:8	2		2			
	BSR d:16	2		2			1
BST	BST #xx:3,Rd	1					
	BST #xx:3,@ERd	2			2		
	BST #xx:3,@aa:8	2			2		
	BST #xx:3,@aa:16	3			2		
	BST #xx:3,@aa:32	4			2		

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
BTST	BTST #xx:3,Rd	1					
	BTST #xx:3,@ERd	2			1		
	BTST #xx:3,@aa:8	2			1		
	BTST #xx:3,@aa:16	3			1		
	BTST #xx:3,@aa:32	4			1		
	BTST Rn,Rd	1					
	BTST Rn,@ERd	2			1		
	BTST Rn,@aa:8	2			1		
	BTST Rn,@aa:16	3			1		
	BTST Rn,@aa:32	4			1		
BXOR	BXOR #xx:3,Rd	1					
	BXOR #xx:3,@ERd	2			1		
	BXOR #xx:3,@aa:8	2			1		
	BXOR #xx:3,@aa:16	3			1		
	BXOR #xx:3,@aa:32	4			1		
CLRMAC	CLRMAC	1					1*[3]
CMP	CMP.B #xx:8,Rd	1					
	CMP.B Rs,Rd	1					
	CMP.W #xx:16,Rd	2					
	CMP.W Rs,Rd	1					
	CMP.L #xx:32,ERd	3					
	CMP.L ERs,ERd	1					
DAA	DAA Rd	1					
DAS	DAS Rd	1					
DEC	DEC.B Rd	1					
	DEC.W #1/2,Rd	1					
	DEC.L #1/2,ERd	1					
DIVXS	DIVXS.B Rs,Rd	2					11
	DIVXS.W Rs,ERd	2					19
DIVXU	DIVXU.B Rs,Rd	1					11
	DIVXU.W Rs,ERd	1					19

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
EEPMOV	EEPMOV.B	2			2n+2*2		
	EEPMOV.W	2			2n+2*2		
EXTS	EXTS.W Rd	1					
	EXTS.L ERd	1					
EXTU	EXTU.W Rd	1					
	EXTU.L ERd	1					
INC	INC.B Rd	1					
	INC.W #1/2,Rd	1					
	INC.L #1/2,ERd	1					
JMP	JMP @ERn	2					
	JMP @aa:24	2					1
	JMP @@aa:8	2	2				1
JSR	JSR @ERn	2		2			
	JSR @aa:24	2		2			1
	JSR @@aa:8	2	2	2			
LDC	LDC #xx:8,CCR	1					
	LDC #xx:8,EXR	2					
	LDC Rs,CCR	1					
	LDC Rs,EXR	1					
	LDC @ERs,CCR	2				1	
	LDC @ERs,EXR	2				1	
	LDC @(d:16,ERs),CCR	3				1	
	LDC @(d:16,ERs),EXR	3				1	
	LDC @(d:32,ERs),CCR	5				1	
	LDC @(d:32,ERs),EXR	5				1	
	LDC @ERs+,CCR	2				1	1
	LDC @ERs+,EXR	2				1	1
	LDC @aa:16,CCR	3				1	
	LDC @aa:16,EXR	3				1	
	LDC @aa:32,CCR	4				1	
	LDC @aa:32,EXR	4				1	

Instruction	Mnemonic	Instruction Fetch	Branch Address Read	Stack Operation	Byte Data Access	Word Data Access	Internal Operation
		I	J	K	L	M	N
LDM*[5]	LDM.L @SP+, (ERn-ERn+1)	2		4			1
	LDM.L @SP+, (ERn-ERn+2)	2		6			1
	LDM.L @SP+, (ERn-ERn+3)	2		8			1
LDMAC	LDMAC ERs,MACH	1					1*[3]
	LDMAC ERs,MACL	1					1*[3]
MAC	MAC @ERn+,@ERm+	2				2	
MOV	MOV.B #xx:8,Rd	1					
	MOV.B Rs,Rd	1					
	MOV.B @ERs,Rd	1			1		
	MOV.B @(d:16,ERs),Rd	2			1		
	MOV.B @(d:32,ERs),Rd	4			1		
	MOV.B @ERs+,Rd	1			1		1
	MOV.B @aa:8,Rd	1			1		
	MOV.B @aa:16,Rd	2			1		
	MOV.B @aa:32,Rd	3			1		
	MOV.B Rs,@ERd	1			1		
	MOV.B Rs,@(d:16,ERd)	2			1		
	MOV.B Rs,@(d:32,ERd)	4			1		
	MOV.B Rs,@-ERd	1			1		1
	MOV.B Rs,@aa:8	1			1		
	MOV.B Rs,@aa:16	2			1		
	MOV.B Rs,@aa:32	3			1		
	MOV.W #xx:16,Rd	2					
	MOV.W Rs,Rd	1					
	MOV.W @ERs,Rd	1				1	
	MOV.W @(d:16,ERs),Rd	2				1	
	MOV.W @(d:32,ERs),Rd	4				1	
	MOV.W @ERs+,Rd	1				1	1
	MOV.W @aa:16,Rd	2				1	
	MOV.W @aa:32,Rd	3				1	
	MOV.W Rs,@ERd	1				1	

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
MOV	MOV.W Rs,@(d:16,ERd)	2				1	
	MOV.W Rs,@(d:32,ERd)	4				1	
	MOV.W Rs,@-ERd	1				1	1
	MOV.W Rs,@aa:16	2				1	
	MOV.W Rs,@aa:32	3				1	
	MOV.L #xx:32,ERd	3					
	MOV.L ERs,ERd	1					
	MOV.L @ERs,ERd	2				2	
	MOV.L @(d:16,ERs),ERd	3				2	
	MOV.L @(d:32,ERs),ERd	5				2	
	MOV.L @ERs+,ERd	2				2	1
	MOV.L @aa:16,ERd	3				2	
	MOV.L @aa:32,ERd	4				2	
	MOV.L ERs,@ERd	2				2	
	MOV.L ERs,@(d:16,ERd)	3				2	
	MOV.L ERs,@(d:32,ERd)	5				2	
	MOV.L ERs,@-ERd	2				2	1
	MOV.L ERs,@aa:16	3				2	
	MOV.L ERs,@aa:32	4				2	
MOVFPE	MOVFPE @:aa:16,Rd	Can not be used in this LSI					
MOVTPE	MOVTPE Rs,@:aa:16						
MULXS	MULXS.B Rs,Rd	2					2
	MULXS.W Rs,ERd	2					3
MULXU	MULXU.B Rs,Rd	1					2
	MULXU.W Rs,ERd	1					3
NEG	NEG.B Rd	1					
	NEG.W Rd	1					
	NEG.L ERd	1					
NOP	NOP	1					
NOT	NOT.B Rd	1					
	NOT.W Rd	1					
	NOT.L ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
OR	OR.B #xx:8,Rd	1					
	OR.B Rs,Rd	1					
	OR.W #xx:16,Rd	2					
	OR.W Rs,Rd	1					
	OR.L #xx:32,ERd	3					
	OR.L ERs,ERd	2					
ORC	ORC #xx:8,CCR	1					
	ORC #xx:8,EXR	2					
POP	POP.W Rn	1				1	1
	POP.L ERn	2				2	1
PUSH	PUSH.W Rn	1				1	1
	PUSH.L ERn	2				2	1
ROTL	ROTL.B Rd	1					
	ROTL.B #2,Rd	1					
	ROTL.W Rd	1					
	ROTL.W #2,Rd	1					
	ROTL.L ERd	1					
	ROTL.L #2,ERd	1					
ROTR	ROTR.B Rd	1					
	ROTR.B #2,Rd	1					
	ROTR.W Rd	1					
	ROTR.W #2,Rd	1					
	ROTR.L ERd	1					
	ROTR.L #2,ERd	1					
ROTXL	ROTXL.B Rd	1					
	ROTXL.B #2,Rd	1					
	ROTXL.W Rd	1					
	ROTXL.W #2,Rd	1					
	ROTXL.L ERd	1					
	ROTXL.L #2,ERd	1					

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
ROTXR	ROTXR.B Rd	1					
	ROTXR.B #2,Rd	1					
	ROTXR.W Rd	1					
	ROTXR.W #2,Rd	1					
	ROTXR.L ERd	1					
	ROTXR.L #2,ERd	1					
RTE	RTE	2		2/3*[1]			1
RTS	RTS	2		2			1
SHAL	SHAL.B Rd	1					
	SHAL.B #2,Rd	1					
	SHAL.W Rd	1					
	SHAL.W #2,Rd	1					
	SHAL.L ERd	1					
	SHAL.L #2,ERd	1					
SHAR	SHAR.B Rd	1					
	SHAR.B #2,Rd	1					
	SHAR.W Rd	1					
	SHAR.W #2,Rd	1					
	SHAR.L ERd	1					
	SHAR.L #2,ERd	1					
SHLL	SHLL.B Rd	1					
	SHLL.B #2,Rd	1					
	SHLL.W Rd	1					
	SHLL.W #2,Rd	1					
	SHLL.L ERd	1					
	SHLL.L #2,ERd	1					
SHLR	SHLR.B Rd	1					
	SHLR.B #2,Rd	1					
	SHLR.W Rd	1					
	SHLR.W #2,Rd	1					
	SHLR.L ERd	1					
	SHLR.L #2,ERd	1					
SLEEP	SLEEP	1					1

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
STC	STC.B CCR,Rd	1					
	STC.B EXR,Rd	1					
	STC.W CCR,@ERd	2				1	
	STC.W EXR,@ERd	2				1	
	STC.W CCR,@(d:16,ERd)	3				1	
	STC.W EXR,@(d:16,ERd)	3				1	
	STC.W CCR,@(d:32,ERd)	5				1	
	STC.W EXR,@(d:32,ERd)	5				1	
	STC.W CCR,@-ERd	2				1	1
	STC.W EXR,@-ERd	2				1	1
	STC.W CCR,@aa:16	3				1	
	STC.W EXR,@aa:16	3				1	
	STC.W CCR,@aa:32	4				1	
	STC.W EXR,@aa:32	4				1	
STM*[5]	STM.L (ERn-ERn+1), @-SP	2		4			1
	STM.L (ERn-ERn+2), @-SP	2		6			1
	STM.L (ERn-ERn+3), @-SP	2		8			1
STMAC	STMAC MACH,ERd	1					0*[3]
	STMAC MACL,ERd	1					0*[3]
SUB	SUB.B Rs,Rd	1					
	SUB.W #xx:16,Rd	2					
	SUB.W Rs,Rd	1					
	SUB.L #xx:32,ERd	3					
	SUB.L ERs,ERd	1					
SUBS	SUBS #1/2/4,ERd	1					
SUBX	SUBX #xx:8,Rd	1					
	SUBX Rs,Rd	1					
TAS*[4]	TAS @ERd	2			2		
TRAPA	TRAPA #x:2	2	2	2/3*[1]			2

Instruction	Mnemonic	Instruction Fetch I	Branch Address Read J	Stack Operation K	Byte Data Access L	Word Data Access M	Internal Operation N
XOR	XOR.B #xx:8,Rd	1					
	XOR.B Rs,Rd	1					
	XOR.W #xx:16,Rd	2					
	XOR.W Rs,Rd	1					
	XOR.L #xx:32,ERd	3					
	XOR.L ERs,ERd	2					
XORC	XORC #xx:8,CCR	1					
	XORC #xx:8,EXR	2					

Notes: *1 2 when EXR is invalid, 3 when EXR is valid.

*2 When n bytes of data are transferred.

*3 An internal operation may require between 0 and 3 additional states, depending on the preceding instruction.

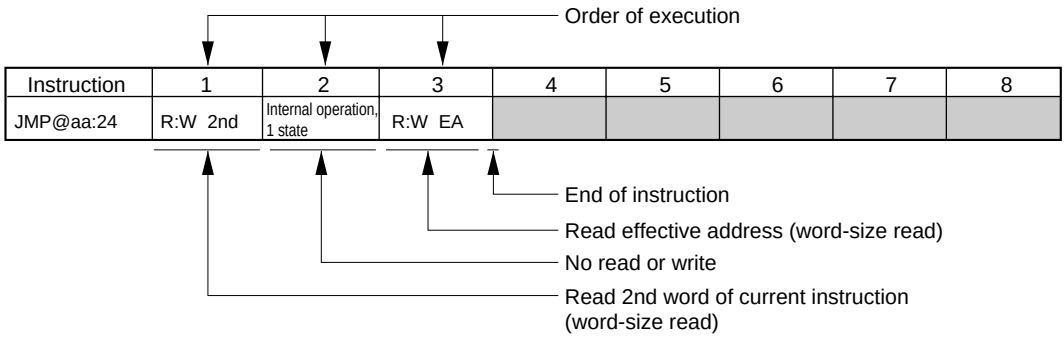
*4 Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.

*5 Only registers ER0 to ER6 should be used when using the STM/LDM instruction.

A.5 Bus States During Instruction Execution

Table A-6 indicates the types of cycles that occur during instruction execution by the CPU. See table A-4 for the number of states per cycle.

How to Read the Table:



Legend

R:B	Byte-size read
R:W	Word-size read
W:B	Byte-size write
W:W	Word-size write
:M	Transfer of the bus is not performed immediately after this cycle
2nd	Address of 2nd word (3rd and 4th bytes)
3rd	Address of 3rd word (5th and 6th bytes)
4th	Address of 4th word (7th and 8th bytes)
5th	Address of 5th word (9th and 10th bytes)
NEXT	Address of next instruction
EA	Effective address
VEC	Vector address

Figure A-1 shows timing waveforms for the address bus and the $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ signals during execution of the above instruction with an 8-bit bus, using three-state access with no wait states.

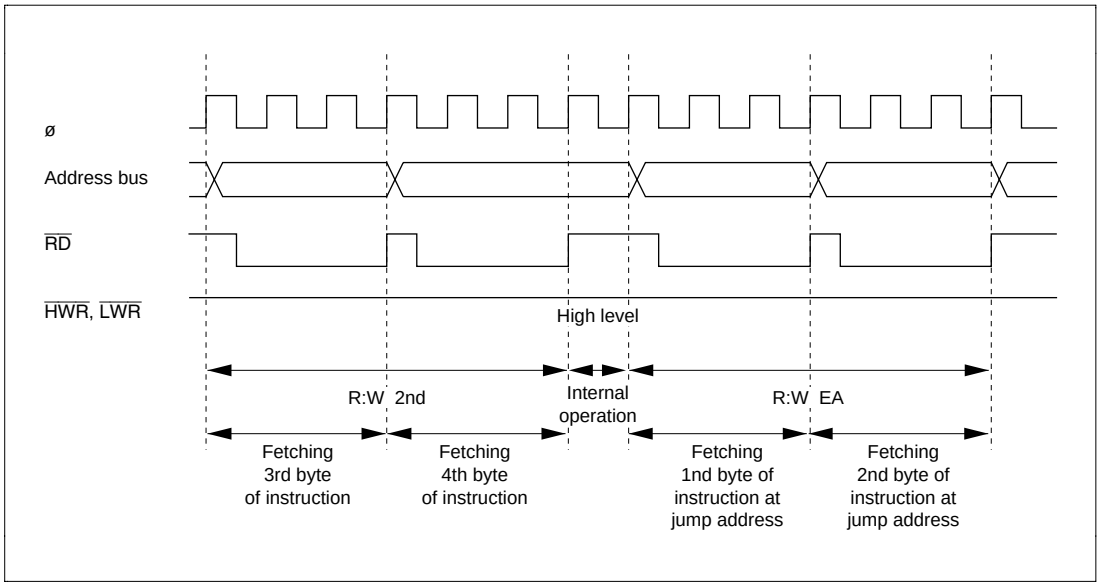


Figure A-1 Address Bus, $\overline{RD}$, $\overline{HWR}$, and $\overline{LWR}$ Timing (8-Bit Bus, Three-State Access, No Wait States)

Table A-6 Instruction Execution Cycles

Instruction	1	2	3	4	5	6	7	8	9
ADD.B #xx:8,Rd	R:W NEXT								
ADD.B Rs,Rd	R:W NEXT								
ADD.W #xx:16,Rd	R:W 2nd	R:W NEXT							
ADD.W Rs,Rd	R:W NEXT								
ADD.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
ADD.L ERs,ERd	R:W NEXT								
ADDS #1/2/4,ERd	R:W NEXT								
ADDX #xx:8,Rd	R:W NEXT								
ADDX Rs,Rd	R:W NEXT								
AND.B #xx:8,Rd	R:W NEXT								
AND.B Rs,Rd	R:W NEXT								
AND.W #xx:16,Rd	R:W 2nd	R:W NEXT							
AND.W Rs,Rd	R:W NEXT								
AND.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
AND.L ERs,ERd	R:W 2nd	R:W NEXT							
ANDC #xx:8,CCR	R:W NEXT								
ANDC #xx:8,EXR	R:W 2nd	R:W NEXT							
BAND #xx:3,Rd	R:W NEXT								
BAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BRA d:8 (BT d:8)	R:W NEXT	R:W EA							
BRN d:8 (BF d:8)	R:W NEXT	R:W EA							
BHI d:8	R:W NEXT	R:W EA							
BLS d:8	R:W NEXT	R:W EA							
BCC d:8 (BHS d:8)	R:W NEXT	R:W EA							
BCS d:8 (BLO d:8)	R:W NEXT	R:W EA							
BNE d:8	R:W NEXT	R:W EA							
BEQ d:8	R:W NEXT	R:W EA							
BVC d:8	R:W NEXT	R:W EA							
BVS d:8	R:W NEXT	R:W EA							
BPL d:8	R:W NEXT	R:W EA							
BMI d:8	R:W NEXT	R:W EA							
BGE d:8	R:W NEXT	R:W EA							
BLT d:8	R:W NEXT	R:W EA							
BGT d:8	R:W NEXT	R:W EA							

Instruction	1	2	3	4	5	6	7	8	9
BLE d:8	R:W NEXT	R:W EA							
BRA d:16 (BT d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BRN d:16 (BF d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BHI d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BLS d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BCC d:16 (BHS d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BCS d:16 (BLO d:16)	R:W 2nd	Internal operation, 1 state	R:W EA						
BNE d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BEQ d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BVC d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BVS d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BPL d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BMI d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BGE d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BLT d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BGT d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BLE d:16	R:W 2nd	Internal operation, 1 state	R:W EA						
BCLR #xx:3,Rd	R:W NEXT								
BCLR #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				

Instruction	1	2	3	4	5	6	7	8	9
BCLR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BCLR Rn,Rd	R:W NEXT								
BCLR Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BCLR Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BCLR Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BIAND #xx:3,Rd	R:W NEXT								
BIAND #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BIAND #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIAND #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIAND #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BILD #xx:3,Rd	R:W NEXT								
BILD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BILD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BILD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BILD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BIOR #xx:3,Rd	R:W NEXT								
BIOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BIOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BIST #xx:3,Rd	R:W NEXT								
BIST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BIST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BIST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BIST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BIXOR #xx:3,Rd	R:W NEXT								
BIXOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BIXOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BIXOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BIXOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BLD #xx:3,Rd	R:W NEXT								
BLD #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BLD #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BLD #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BLD #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BNOT #xx:3,Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
BNOT #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BNOT #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BNOT Rn,Rd	R:W NEXT								
BNOT Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BNOT Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BNOT Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BOR #xx:3,Rd	R:W NEXT								
BOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BSET #xx:3,Rd	R:W NEXT								
BSET #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BSET #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BSET Rn,Rd	R:W NEXT								
BSET Rn,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET Rn,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BSET Rn,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BSET Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BSR d:8	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)					
BSR d:16	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)				
BST #xx:3,Rd	R:W NEXT								
BST #xx:3,@ERd	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BST #xx:3,@aa:8	R:W 2nd	R:B:M EA	R:W:M NEXT	W:B EA					
BST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B:M EA	R:W:M NEXT	W:B EA				
BST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B:M EA	R:W:M NEXT	W:B EA			
BTST #xx:3,Rd	R:W NEXT								
BTST #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						

Instruction	1	2	3	4	5	6	7	8	9
BTST #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BTST #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BTST #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BTST Rn,Rd	R:W NEXT								
BTST Rn,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BTST Rn,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BTST Rn,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BTST Rn,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
BXOR #xx:3,Rd	R:W NEXT								
BXOR #xx:3,@ERd	R:W 2nd	R:B EA	R:W:M NEXT						
BXOR #xx:3,@aa:8	R:W 2nd	R:B EA	R:W:M NEXT						
BXOR #xx:3,@aa:16	R:W 2nd	R:W 3rd	R:B EA	R:W:M NEXT					
BXOR #xx:3,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:B EA	R:W:M NEXT				
CLRMAC	R:W NEXT	Internal operation, 1 state							
CMP.B #xx:8,Rd	R:W NEXT								
CMP.B Rs,Rd	R:W NEXT								
CMP.W #xx:16,Rd	R:W 2nd	R:W NEXT							
CMP.W Rs,Rd	R:W NEXT								
CMP.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
CMP.L ERs,ERd	R:W NEXT								
DAA Rd	R:W NEXT								
DAS Rd	R:W NEXT								
DEC.B Rd	R:W NEXT								
DEC.W #1/2,Rd	R:W NEXT								
DEC.L #1/2,ERd	R:W NEXT								
DIVXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 11 states						
DIVXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 19 states						
DIVXU.B Rs,Rd	R:W NEXT	Internal operation, 11 states							
DIVXU.W Rs,ERd	R:W NEXT	Internal operation, 19 states							
EEPMOV.B	R:W 2nd	R:B EAs*1	R:B EAd*1	R:B EAs*2	W:B EAd*2	R:W NEXT			
EEPMOV.W	R:W 2nd	R:B EAs*1	R:B EAd*1	R:B EAs*2	W:B EAd*2	R:W NEXT			
EXTS.W Rd	R:W NEXT			← Repeated n times*2 →					
EXTS.L ERd	R:W NEXT								
EXTU.W Rd	R:W NEXT								
EXTU.L ERd	R:W NEXT								
INC.B Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
INC.W #1/2,Rd	R:W NEXT								
INC.L #1/2,ERd	R:W NEXT								
JMP @ERn	R:W NEXT	R:W EA							
JMP @aa:24	R:W 2nd	Internal operation, 1 state	R:W EA						
JMP @@aa:8	R:W NEXT	R:W:M aa:8	R:W aa:8	Internal operation, 1 state	R:W EA				
JSR @ERn	R:W NEXT	R:W EA	W:W:M stack (H)	W:W stack (L)					
JSR @aa:24	R:W 2nd	Internal operation, 1 state	R:W EA	W:W:M stack (H)	W:W stack (L)				
JSR @@aa:8	R:W NEXT	R:W:M aa:8	R:W aa:8	W:W:M stack (H)	W:W stack (L)	R:W EA			
LDC #xx:8,CCR	R:W NEXT								
LDC #xx:8,EXR	R:W 2nd	R:W NEXT							
LDC Rs,CCR	R:W NEXT								
LDC Rs,EXR	R:W NEXT								
LDC @ERs,CCR	R:W 2nd	R:W NEXT	R:W EA						
LDC @ERs,EXR	R:W 2nd	R:W NEXT	R:W EA						
LDC @(d:16,ERs),CCR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC @(d:16,ERs),EXR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC @(d:32,ERs),CCR	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA			
LDC @(d:32,ERs),EXR	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	R:W EA			
LDC @ERs+,CCR	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W EA					
LDC @ERs+,EXR	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W EA					
LDC @aa:16,CCR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC @aa:16,EXR	R:W 2nd	R:W 3rd	R:W NEXT	R:W EA					
LDC @aa:32,CCR	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA				
LDC @aa:32,EXR	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA				
LDM.L @SP+, (ERn–ERn+1)*9	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3				
LDM.L @SP+,(ERn–ERn+2)*9	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3				
LDM.L @SP+,(ERn–ERn+3)*9	R:W 2nd	R:W NEXT	Internal operation, 1 state	R:W:M stack (H)*3	R:W stack (L)*3				
LDMAC ERs,MACH	R:W NEXT	Internal operation, 1 state		←Repeated n times *3→					

Instruction	1	2	3	4	5	6	7	8	9
LDMAC ERs,MACL	R:W NEXT	Internal operation, 1 state							
MAC @ERn+,@ERm+	R:W 2nd	R:W NEXT	R:W EAh	R:W EAm					
MOV.B #xx:8,Rd	R:W NEXT								
MOV.B Rs,Rd	R:W NEXT								
MOV.B @ERs,Rd	R:W NEXT	R:B EA							
MOV.B @(d:16,ERs),Rd	R:W 2nd	R:W NEXT	R:B EA						
MOV.B @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:B EA				
MOV.B @ERs+,Rd	R:W NEXT	Internal operation, 1 state	R:B EA						
MOV.B @aa:8,Rd	R:W NEXT	R:B EA							
MOV.B @aa:16,Rd	R:W 2nd	R:W NEXT	R:B EA						
MOV.B @aa:32,Rd	R:W 2nd	R:W 3rd	R:W NEXT	R:B EA					
MOV.B Rs,@ERd	R:W NEXT	W:B EA							
MOV.B Rs,@(d:16,ERd)	R:W 2nd	R:W NEXT	W:B EA						
MOV.B Rs,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:B EA				
MOV.B Rs,@–ERd	R:W NEXT	Internal operation, 1 state	W:B EA						
MOV.B Rs,@aa:8	R:W NEXT	W:B EA							
MOV.B Rs,@aa:16	R:W 2nd	R:W NEXT	W:B EA						
MOV.B Rs,@aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:B EA					
MOV.W #xx:16,Rd	R:W 2nd	R:W NEXT							
MOV.W Rs,Rd	R:W NEXT								
MOV.W @ERs,Rd	R:W NEXT	R:W EA							
MOV.W @(d:16,ERs),Rd	R:W 2nd	R:W NEXT	R:W EA						
MOV.W @(d:32,ERs),Rd	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	R:W EA				
MOV.W @ERs+, Rd	R:W NEXT	Internal operation, 1 state	R:W EA						
MOV.W @aa:16,Rd	R:W 2nd	R:W NEXT	R:W EA						
MOV.W @aa:32,Rd	R:W 2nd	R:W 3rd	R:W NEXT	R:B EA					
MOV.W Rs,@ERd	R:W NEXT	W:W EA							
MOV.W Rs,@(d:16,ERd)	R:W 2nd	R:W NEXT	W:W EA						
MOV.W Rs,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:E 4th	R:W NEXT	W:W EA				
MOV.W Rs,@–ERd	R:W NEXT	Internal operation, 1 state	W:W EA						
MOV.W Rs,@aa:16	R:W 2nd	R:W NEXT	W:W EA						
MOV.W Rs,@aa:32	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					

Instruction	1	2	3	4	5	6	7	8	9
MOV.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
MOV.L ERs,ERd	R:W NEXT								
MOV.L @ERs,ERd	R:W 2nd	R:W:M NEXT	R:W:M EA	R:W EA+2					
MOV.L @(d:16,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2				
MOV.L @(d:32,ERs),ERd	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	R:W:M EA	R:W EA+2		
MOV.L @ERs+,ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2				
MOV.L @aa:16,ERd	R:W 2nd	R:W:M 3rd	R:W NEXT	R:W:M EA	R:W EA+2				
MOV.L @aa:32,ERd	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	R:W:M EA	R:W EA+2			
MOV.L ERs,@ERd	R:W 2nd	R:W:M NEXT	W:W:M EA	W:W EA+2					
MOV.L ERs,@(d:16,ERd)	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2				
MOV.L ERs,@(d:32,ERd)	R:W 2nd	R:W:M 3rd	R:W:M 4th	R:W 5th	R:W NEXT	W:W:M EA	W:W EA+2		
MOV.L ERs,@–ERd	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M EA	W:W EA+2				
MOV.L ERs,@aa:16	R:W 2nd	R:W:M 3rd	R:W NEXT	W:W:M EA	W:W EA+2				
MOV.L ERs,@aa:32	R:W 2nd	R:W:M 3rd	R:W 4th	R:W NEXT	W:W:M EA	W:W EA+2			
MOVFPE @aa:16,Rd	Cannot be used in this LSI								
MOVTPE Rs,@aa:16									
MULXS.B Rs,Rd	R:W 2nd	R:W NEXT	Internal operation, 2 states						
MULXS.W Rs,ERd	R:W 2nd	R:W NEXT	Internal operation, 3 states						
MULXU.B Rs,Rd	R:W NEXT	Internal operation, 2 states							
MULXU.W Rs,ERd	R:W NEXT	Internal operation, 3 states							
NEG.B Rd	R:W NEXT								
NEG.W Rd	R:W NEXT								
NEG.L ERd	R:W NEXT								
NOP	R:W NEXT								
NOT.B Rd	R:W NEXT								
NOT.W Rd	R:W NEXT								
NOT.L ERd	R:W NEXT								
OR.B #xx:8,Rd	R:W NEXT								
OR.B Rs,Rd	R:W NEXT								
OR.W #xx:16,Rd	R:W 2nd	R:W NEXT							
OR.W Rs,Rd	R:W NEXT								
OR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
OR.L ERs,ERd	R:W 2nd	R:W NEXT							
ORC #xx:8,CCR	R:W NEXT								
ORC #xx:8,EXR	R:W 2nd	R:W NEXT							

Instruction	1	2	3	4	5	6	7	8	9
POP.W Rn	R:W NEXT	Internal operation, 1 state	R:W EA						
POP.L ERn	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	R:W:M EA	R:W EA+2				
PUSH.W Rn	R:W NEXT	Internal operation, 1 state	W:W EA						
PUSH.L ERn	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M EA	W:W EA+2				
ROTL.B Rd	R:W NEXT								
ROTL.B #2,Rd	R:W NEXT								
ROTL.W Rd	R:W NEXT								
ROTL.W #2,Rd	R:W NEXT								
ROTL.L ERd	R:W NEXT								
ROTL.L #2,ERd	R:W NEXT								
ROTR.B Rd	R:W NEXT								
ROTR.B #2,Rd	R:W NEXT								
ROTR.W Rd	R:W NEXT								
ROTR.W #2,Rd	R:W NEXT								
ROTR.L ERd	R:W NEXT								
ROTR.L #2,ERd	R:W NEXT								
ROTXL.B Rd	R:W NEXT								
ROTXL.B #2,Rd	R:W NEXT								
ROTXL.W Rd	R:W NEXT								
ROTXL.W #2,Rd	R:W NEXT								
ROTXL.L ERd	R:W NEXT								
ROTXL.L #2,ERd	R:W NEXT								
ROTXR.B Rd	R:W NEXT								
ROTXR.B #2,Rd	R:W NEXT								
ROTXR.W Rd	R:W NEXT								
ROTXR.W #2,Rd	R:W NEXT								
ROTXR.L ERd	R:W NEXT								
ROTXR.L #2,ERd	R:W NEXT								
RTE	R:W NEXT	R:W stack (EXR)	R:W stack (H)	R:W stack (L)	Internal operation, 1 state	R:W*4			
RTS	R:W NEXT	R:W:M stack (H)	R:W stack (L) 1 state	Internal operation,	R:W*4				
SHAL.B Rd	R:W NEXT								

Instruction	1	2	3	4	5	6	7	8	9
SHAL.B #2,Rd	R:W NEXT								
SHAL.W Rd	R:W NEXT								
SHAL.W #2,Rd	R:W NEXT								
SHAL.L ERd	R:W NEXT								
SHAL.L #2,ERd	R:W NEXT								
SHAR.B Rd	R:W NEXT								
SHAR.B #2,Rd	R:W NEXT								
SHAR.W Rd	R:W NEXT								
SHAR.W #2,Rd	R:W NEXT								
SHAR.L ERd	R:W NEXT								
SHAR.L #2,ERd	R:W NEXT								
SHLL.B Rd	R:W NEXT								
SHLL.B #2,Rd	R:W NEXT								
SHLL.W Rd	R:W NEXT								
SHLL.W #2,Rd	R:W NEXT								
SHLL.L ERd	R:W NEXT								
SHLL.L #2,ERd	R:W NEXT								
SHLR.B Rd	R:W NEXT								
SHLR.B #2,Rd	R:W NEXT								
SHLR.W Rd	R:W NEXT								
SHLR.W #2,Rd	R:W NEXT								
SHLR.L ERd	R:W NEXT								
SHLR.L #2,ERd	R:W NEXT								
SLEEP	R:W NEXT	Internal operation:M							
STC CCR,Rd	R:W NEXT								
STC EXR,Rd	R:W NEXT								
STC CCR,@ERd	R:W 2nd	R:W NEXT	W:W EA						
STC EXR,@ERd	R:W 2nd	R:W NEXT	W:W EA						
STC CCR,@(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC EXR,@(d:16,ERd)	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC CCR,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA			
STC EXR,@(d:32,ERd)	R:W 2nd	R:W 3rd	R:W 4th	R:W 5th	R:W NEXT	W:W EA			
STC CCR,@–ERd	R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA					
STC EXR,@–ERd	R:W 2nd	R:W NEXT	Internal operation, 1 state	W:W EA					
STC CCR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					
STC EXR,@aa:16	R:W 2nd	R:W 3rd	R:W NEXT	W:W EA					

Instruction	1	2	3	4	5	6	7	8	9
STC CCR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA				
STC EXR,@aa:32	R:W 2nd	R:W 3rd	R:W 4th	R:W NEXT	W:W EA				
STM.L(ERn–ERn+1),@–SP*9	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STM.L(ERn–ERn+2),@–SP*9	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STM.L(ERn–ERn+3),@–SP*9	R:W 2nd	R:W:M NEXT	Internal operation, 1 state	W:W:M stack (H)*3	W:W stack (L)*3				
STMAC MACH,ERd	R:W NEXT								
STMAC MACL,ERd	R:W NEXT								
SUB.B Rs,Rd	R:W NEXT								
SUB.W #xx:16,Rd	R:W 2nd	R:W NEXT							
SUB.W Rs,Rd	R:W NEXT								
SUB.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
SUB.L ERs,ERd	R:W NEXT								
SUBS #1/2/4,ERd	R:W NEXT								
SUBX #xx:8,Rd	R:W NEXT								
SUBX Rs,Rd	R:W NEXT								
TAS @ERd*8	R:W 2nd	R:W NEXT	R:B:M EA	W:B EA					
TRAPA #x:2	R:W NEXT	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W*7
XOR.B #xx8,Rd	R:W NEXT								
XOR.B Rs,Rd	R:W NEXT								
XOR.W #xx:16,Rd	R:W 2nd	R:W NEXT							
XOR.W Rs,Rd	R:W NEXT								
XOR.L #xx:32,ERd	R:W 2nd	R:W 3rd	R:W NEXT						
XOR.L ERs,ERd	R:W 2nd	R:W NEXT							
XORC #xx:8,CCR	R:W NEXT								
XORC #xx:8,EXR	R:W 2nd	R:W NEXT							

Instruction	1	2	3	4	5	6	7	8	9
Reset exception handling	R:W VEC	R:W VEC+2	Internal operation, 1 state	R:W*5					
Interrupt exception handling	R:W*6	Internal operation, 1 state	W:W stack (L)	W:W stack (H)	W:W stack (EXR)	R:W:M VEC	R:W VEC+2	Internal operation, 1 state	R:W*7

Notes: *1 EAs is the contents of ER5. EAd is the contents of ER6.
*2 EAs is the contents of ER5. EAd is the contents of ER6. Both registers are incremented by 1 after execution of the instruction. n is the initial value of R4L or R4. If n = 0, these bus cycles are not executed.
*3 Repeated two times to save or restore two registers, three times for three registers, or four times for four registers.
*4 Start address after return.
*5 Start address of the program.
*6 Prefetch address, equal to two plus the PC value pushed onto the stack. In recovery from sleep mode or software standby mode the read operation is replaced by an internal operation.
*7 Start address of the interrupt-handling routine.
*8 Only register ER0, ER1, ER4, or ER5 should be used when using the TAS instruction.
*9 Only registers ER0 to ER6 should be used when using the STM/LDM instruction.

A.6 Condition Code Modification

This section indicates the effect of each CPU instruction on the condition code. The notation used in the table is defined below.

m = { 31 for longword operands; 15 for word operands; 7 for byte operands

Si	The i-th bit of the source operand
Di	The i-th bit of the destination operand
Ri	The i-th bit of the result
Dn	The specified bit in the destination operand
—	Not affected
↕	Modified according to the result of the instruction (see definition)
0	Always cleared to 0
1	Always set to 1
*	Undetermined (no guaranteed value)
Z'	Z flag before instruction execution
C'	C flag before instruction execution

Table A-7 Condition Code Modification

Instruction	H	N	Z	V	C	Definition
ADD	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{Rm–4}$ + Sm–4 · $\overline{Rm–4}$ N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$ V = Sm · Dm · $\overline{Rm}$ + $\overline{Sm}$ · $\overline{Dm}$ · Rm C = Sm · Dm + Dm · $\overline{Rm}$ + Sm · $\overline{Rm}$
ADDS	—	—	—	—	—	
ADDX	↕	↕	↕	↕	↕	H = Sm–4 · Dm–4 + Dm–4 · $\overline{Rm–4}$ + Sm–4 · $\overline{Rm–4}$ N = Rm Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$ V = Sm · Dm · $\overline{Rm}$ + $\overline{Sm}$ · $\overline{Dm}$ · Rm C = Sm · Dm + Dm · $\overline{Rm}$ + Sm · $\overline{Rm}$
AND	—	↕	↕	0	—	N = Rm Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
ANDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result. No flags change when the operand is EXR.
BAND	—	—	—	—	↕	C = C' · Dn
Bcc	—	—	—	—	—	
BCLR	—	—	—	—	—	
BIAND	—	—	—	—	↕	C = C' · $\overline{Dn}$
BILD	—	—	—	—	↕	C = $\overline{Dn}$
BIOR	—	—	—	—	↕	C = C' + $\overline{Dn}$
BIST	—	—	—	—	—	
BIXOR	—	—	—	—	↕	C = C' · Dn + $\overline{C'}$ · $\overline{Dn}$
BLD	—	—	—	—	↕	C = Dn
BNOT	—	—	—	—	—	
BOR	—	—	—	—	↕	C = C' + Dn
BSET	—	—	—	—	—	
BSR	—	—	—	—	—	
BST	—	—	—	—	—	
BTST	—	—	↕	—	—	Z = $\overline{Dn}$
BXOR	—	—	—	—	↕	C = C' · $\overline{Dn}$ + $\overline{C'}$ · Dn
CLRMAC	—	—	—	—	—	

Instruction	H	N	Z	V	C	Definition
CMP	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4
						N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm
						C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
DAA	*	↕	↕	*	↕	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						C: decimal arithmetic carry
DAS	*	↕	↕	*	↕	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						C: decimal arithmetic borrow
DEC	—	↕	↕	↕	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = Dm · $\overline{Rm}$
DIVXS	—	↕	↕	—	—	N = Sm · $\overline{Dm}$ + $\overline{Sm}$ · Dm
						Z = $\overline{Sm}$ · $\overline{Sm–1}$ · ····· · $\overline{S0}$
DIVXU	—	↕	↕	—	—	N = Sm
						Z = $\overline{Sm}$ · $\overline{Sm–1}$ · ····· · $\overline{S0}$
EEPMOV	—	—	—	—	—	
EXTS	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
EXTU	—	0	↕	0	—	Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
INC	—	↕	↕	↕	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = $\overline{Dm}$ · Rm
JMP	—	—	—	—	—	
JSR	—	—	—	—	—	
LDC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.
LDM	—	—	—	—	—	
LDMAC	—	—	—	—	—	
MAC	—	—	—	—	—	

Instruction	H	N	Z	V	C	Definition
MOV	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
MOVFPE						Can not be used in this LSI
MOVTPE						
MULXS	—	↕	↕	—	—	N = R2m
						Z = $\overline{R2m} \cdot \overline{R2m-1} \cdot \cdots\cdots \cdot \overline{R0}$
MULXU	—	—	—	—	—	
NEG	↕	↕	↕	↕	↕	H = Dm–4 + Rm–4
						N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
						V = Dm · Rm
						C = Dm + Rm
NOP	—	—	—	—	—	
NOT	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
OR	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
ORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.
POP	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
PUSH	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
ROTL	—	↕	↕	0	↕	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
						C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
ROTR	—	↕	↕	0	↕	N = Rm
						Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$
						C = D0 (1-bit shift) or C = D1 (2-bit shift)

Instruction	H	N	Z	V	C	Definition
ROTXL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
ROTXR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
RTE	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
RTS	—	—	—	—	—	
SHAL	—	↕	↕	↕	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ V = $\overline{Dm \cdot Dm-1 + \overline{Dm} \cdot \overline{Dm-1}}$ (1-bit shift) V = $\overline{Dm \cdot Dm-1 \cdot Dm-2 \cdot \overline{Dm} \cdot \overline{Dm-1} \cdot \overline{Dm-2}}$ (2-bit shift) C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
SHAR	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
SHLL	—	↕	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = Dm (1-bit shift) or C = Dm–1 (2-bit shift)
SHLR	—	0	↕	0	↕	N = Rm Z = $\overline{Rm} \cdot \overline{Rm-1} \cdot \cdots\cdots \cdot \overline{R0}$ C = D0 (1-bit shift) or C = D1 (2-bit shift)
SLEEP	—	—	—	—	—	
STC	—	—	—	—	—	
STM	—	—	—	—	—	
STMAC	—	↕	↕	↕	—	N = 1 if MAC instruction resulted in negative value in MAC register Z = 1 if MAC instruction resulted in zero value in MAC register V = 1 if MAC instruction resulted in overflow

Instruction	H	N	Z	V	C	Definition
SUB	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4
						N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
						V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm
						C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
SUBS	—	—	—	—	—	
SUBX	↕	↕	↕	↕	↕	H = Sm–4 · $\overline{Dm–4}$ + $\overline{Dm–4}$ · Rm–4 + Sm–4 · Rm–4
						N = Rm
						Z = Z' · $\overline{Rm}$ · ····· · $\overline{R0}$
						V = $\overline{Sm}$ · Dm · $\overline{Rm}$ + Sm · $\overline{Dm}$ · Rm
						C = Sm · $\overline{Dm}$ + $\overline{Dm}$ · Rm + Sm · Rm
TAS	—	↕	↕	0	—	N = Dm
						Z = $\overline{Dm}$ · $\overline{Dm–1}$ · ····· · $\overline{D0}$
TRAPA	—	—	—	—	—	
XOR	—	↕	↕	0	—	N = Rm
						Z = $\overline{Rm}$ · $\overline{Rm–1}$ · ····· · $\overline{R0}$
XORC	↕	↕	↕	↕	↕	Stores the corresponding bits of the result.
						No flags change when the operand is EXR.

B.1 Address

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'EBC0 to H'EFBF	MRA	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz	DTC	8/16/32
	MRB	CHNE	DISEL	—	—	—	—	—	—		
	SAR										
	DAR										
	CRA										
	CRB										
H'F800	MCR0	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	HCAN0	8/16
H'F801	GSR0	—	—	—	—	GSR3	GSR2	GSR1	GSR0		
H'F802	BCR0	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0		
H'F803		BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8		
H'F804	MBCR	MBCR7	MBCR6	MBCR5	MBCR4	MBCR3	MBCR2	MBCR1	—		
H'F805		MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9	MBCR8		
H'F806	TXPR	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—		
H'F807		TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8		
H'F808	TXCR	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—		
H'F809		TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8		
H'F80A	TXACK	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—		
H'F80B		TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8		
H'F80C	ABACK	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—		
H'F80D		ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8		
H'F80E	RXPR	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0		
H'F80F		RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8		
H'F810	RFPR	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0	HCAN0	8/16
H'F811		RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8		
H'F812	IRR	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0		
H'F813		—	—	—	IRR12	—	—	IRR9	IRR8		
H'F814	MBIMR	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0		
H'F815		MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8		
H'F816	IMR	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	—		
H'F817		—	—	—	IMR12	—	—	IMR9	IMR8		
H'F818	REC										
H'F819	TEC										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'F81A	UMSR	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0	HCAN0	8/16
H'F81B		UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8		
H'F81C	LAFML	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0		
H'F81D		LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8		
H'F81E	LAFMH	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1	LAFMH0		
H'F81F		LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9	LAFMH8		
H'F820	MC0[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F821	MC0[2]	—	—	—	—	—	—	—	—		
H'F822	MC0[3]	—	—	—	—	—	—	—	—		
H'F823	MC0[4]	—	—	—	—	—	—	—	—		
H'F824	MC0[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F825	MC0[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F826	MC0[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F827	MC0[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F828	MC1[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'F829	MC1[2]	—	—	—	—	—	—	—	—		
H'F82A	MC1[3]	—	—	—	—	—	—	—	—		
H'F82B	MC1[4]	—	—	—	—	—	—	—	—		
H'F82C	MC1[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F82D	MC1[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F82E	MC1[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F82F	MC1[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F830	MC2[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F831	MC2[2]	—	—	—	—	—	—	—	—		
H'F832	MC2[3]	—	—	—	—	—	—	—	—		
H'F833	MC2[4]	—	—	—	—	—	—	—	—		
H'F834	MC2[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F835	MC2[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F836	MC2[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F837	MC2[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F838	MC3[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F839	MC3[2]	—	—	—	—	—	—	—	—		
H'F83A	MC3[3]	—	—	—	—	—	—	—	—		
H'F83B	MC3[4]	—	—	—	—	—	—	—	—		
H'F83C	MC3[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F83D	MC3[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F83E	MC3[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F83F	MC3[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'F840	MC4[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F841	MC4[2]	—	—	—	—	—	—	—	—		
H'F842	MC4[3]	—	—	—	—	—	—	—	—		
H'F843	MC4[4]	—	—	—	—	—	—	—	—		
H'F844	MC4[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F845	MC4[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F846	MC4[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F847	MC4[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F848	MC5[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F849	MC5[2]	—	—	—	—	—	—	—	—		
H'F84A	MC5[3]	—	—	—	—	—	—	—	—		
H'F84B	MC5[4]	—	—	—	—	—	—	—	—		
H'F84C	MC5[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F84D	MC5[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F84E	MC5[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F84F	MC5[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F850	MC6[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F851	MC6[2]	—	—	—	—	—	—	—	—		
H'F852	MC6[3]	—	—	—	—	—	—	—	—		
H'F853	MC6[4]	—	—	—	—	—	—	—	—		
H'F854	MC6[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F855	MC6[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F856	MC6[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F857	MC6[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F858	MC7[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'F859	MC7[2]	—	—	—	—	—	—	—	—		
H'F85A	MC7[3]	—	—	—	—	—	—	—	—		
H'F85B	MC7[4]	—	—	—	—	—	—	—	—		
H'F85C	MC7[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F85D	MC7[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F85E	MC7[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F85F	MC7[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F860	MC8[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F861	MC8[2]	—	—	—	—	—	—	—	—		
H'F862	MC8[3]	—	—	—	—	—	—	—	—		
H'F863	MC8[4]	—	—	—	—	—	—	—	—		
H'F864	MC8[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F865	MC8[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F866	MC8[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F867	MC8[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'F868	MC9[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F869	MC9[2]	—	—	—	—	—	—	—	—		
H'F86A	MC9[3]	—	—	—	—	—	—	—	—		
H'F86B	MC9[4]	—	—	—	—	—	—	—	—		
H'F86C	MC9[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F86D	MC9[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F86E	MC9[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F86F	MC9[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F870	MC10[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F871	MC10[2]	—	—	—	—	—	—	—	—		
H'F872	MC10[3]	—	—	—	—	—	—	—	—		
H'F873	MC10[4]	—	—	—	—	—	—	—	—		
H'F874	MC10[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F875	MC10[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F876	MC10[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F877	MC10[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F878	MC11[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'F879	MC11[2]	—	—	—	—	—	—	—	—		
H'F87A	MC11[3]	—	—	—	—	—	—	—	—		
H'F87B	MC11[4]	—	—	—	—	—	—	—	—		
H'F87C	MC11[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F87D	MC11[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F87E	MC11[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F87F	MC11[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F880	MC12[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F881	MC12[2]	—	—	—	—	—	—	—	—		
H'F882	MC12[3]	—	—	—	—	—	—	—	—		
H'F883	MC12[4]	—	—	—	—	—	—	—	—		
H'F884	MC12[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F885	MC12[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F886	MC12[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F887	MC12[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F888	MC13[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F889	MC13[2]	—	—	—	—	—	—	—	—		
H'F88A	MC13[3]	—	—	—	—	—	—	—	—		
H'F88B	MC13[4]	—	—	—	—	—	—	—	—		
H'F88C	MC13[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F88D	MC13[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F88E	MC13[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F88F	MC13[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'F890	MC14[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN0	8/16
H'F891	MC14[2]	—	—	—	—	—	—	—	—		
H'F892	MC14[3]	—	—	—	—	—	—	—	—		
H'F893	MC14[4]	—	—	—	—	—	—	—	—		
H'F894	MC14[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F895	MC14[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F896	MC14[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F897	MC14[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F898	MC15[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'F899	MC15[2]	—	—	—	—	—	—	—	—		
H'F89A	MC15[3]	—	—	—	—	—	—	—	—		
H'F89B	MC15[4]	—	—	—	—	—	—	—	—		
H'F89C	MC15[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'F89D	MC15[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'F89E	MC15[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'F89F	MC15[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'F8B0	MD01									HCAN0	8/16
H'F8B1	MD02										
H'F8B2	MD03										
H'F8B3	MD04										
H'F8B4	MD05										
H'F8B5	MD06										
H'F8B6	MD07										
H'F8B7	MD08										
H'F8B8	MD11									HCAN0	8/16
H'F8B9	MD12										
H'F8BA	MD13										
H'F8BB	MD14										
H'F8BC	MD15										
H'F8BD	MD16										
H'F8BE	MD17										
H'F8BF	MD18										
H'F8C0	MD21									HCAN0	8/16
H'F8C1	MD22										
H'F8C2	MD23										
H'F8C3	MD24										
H'F8C4	MD25										
H'F8C5	MD26										
H'F8C6	MD27										
H'F8C7	MD28										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'F8C8	MD31									HCAN0	8/16
H'F8C9	MD32										
H'F8CA	MD33										
H'F8CB	MD34										
H'F8CC	MD35										
H'F8CD	MD36										
H'F8CE	MD37										
H'F8CF	MD38										
H'F8D0	MD41									HCAN0	8/16
H'F8D1	MD42										
H'F8D2	MD43										
H'F8D3	MD44										
H'F8D4	MD45										
H'F8D5	MD46										
H'F8D6	MD47										
H'F8D7	MD48										
H'F8D8	MD51									HCAN0	8/16
H'F8D9	MD52										
H'F8DA	MD53										
H'F8DB	MD54										
H'F8DC	MD55										
H'F8DD	MD56										
H'F8DE	MD57										
H'F8DF	MD58										
H'F8E0	MD61									HCAN0	8/16
H'F8E1	MD62										
H'F8E2	MD63										
H'F8E3	MD64										
H'F8E4	MD65										
H'F8E5	MD66										
H'F8E6	MD67										
H'F8E7	MD68										
H'F8E8	MD71									HCAN0	8/16
H'F8E9	MD72										
H'F8EA	MD73										
H'F8EB	MD74										
H'F8EC	MD75										
H'F8ED	MD76										
H'F8EE	MD77										
H'F8EF	MD78										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'F8F0	MD81									HCAN0	8/16
H'F8F1	MD82										
H'F8F2	MD83										
H'F8F3	MD84										
H'F8F4	MD85										
H'F8F5	MD86										
H'F8F6	MD87										
H'F8F7	MD88										
H'F8F8	MD91									HCAN0	8/16
H'F8F9	MD92										
H'F8FA	MD93										
H'F8FB	MD94										
H'F8FC	MD95										
H'F8FD	MD96										
H'F8FE	MD97										
H'F8FF	MD98										
H'F900	MD101									HCAN0	8/16
H'F901	MD102										
H'F902	MD103										
H'F903	MD104										
H'F904	MD105										
H'F905	MD106										
H'F906	MD107										
H'F907	MD108										
H'F908	MD111									HCAN0	8/16
H'F909	MD112										
H'F90A	MD113										
H'F90B	MD114										
H'F90C	MD115										
H'F90D	MD116										
H'F90E	MD117										
H'F90F	MD118										
H'F910	MD121									HCAN0	8/16
H'F911	MD122										
H'F912	MD123										
H'F913	MD124										
H'F914	MD125										
H'F915	MD126										
H'F916	MD127										
H'F917	MD128										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'F918	MD131									HCAN0	8/16
H'F919	MD132										
H'F91A	MD133										
H'F91B	MD134										
H'F91C	MD135										
H'F91D	MD136										
H'F91E	MD137										
H'F91F	MD138										
H'F920	MD141									HCAN0	8/16
H'F921	MD142										
H'F922	MD143										
H'F923	MD144										
H'F924	MD145										
H'F925	MD146										
H'F926	MD147										
H'F927	MD148										
H'F928	MD151										
H'F929	MD152										
H'F92A	MD153										
H'F92B	MD154										
H'F92C	MD155										
H'F92D	MD156										
H'F92E	MD157										
H'F92F	MD158										
H'FA00	MCR	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0	HCAN1	8/16
H'FA01	GSR	—	—	—	—	GSR3	GSR2	GSR1	GSR0		
H'FA02	BCR	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0		
H'FA03		BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8		
H'FA04	MBCR	MBCR7	MBCR6	MBCR5	MBCR4	MBCR3	MBCR2	MBCR1	—		
H'FA05		MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9	MBCR8		
H'FA06	TXPR	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—		
H'FA07		TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8		
H'FA08	TXCR	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—		
H'FA09		TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8		
H'FA0A	TXACK	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—		
H'FA0B		TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8		
H'FA0C	ABACK	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—		
H'FA0D		ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8		
H'FA0E	RXPR	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0		
H'FA0F		RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FA10	RFPR	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0	HCAN1	8/16
H'FA11		RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8		
H'FA12	IRR	IRR7	IRR6	IRR5	IRR4	IRR3	IRR2	IRR1	IRR0		
H'FA13		—	—	—	IRR12	—	—	IRR9	IRR8		
H'FA14	MBIMR	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0		
H'FA15		MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8		
H'FA16	IMR	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	IMR0		
H'FA17		—	—	—	IMR12	—	—	IMR9	IMR8		
H'FA18	REC										
H'FA19	TEC										
H'FA1A	UMSR	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0		
H'FA1B		UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8		
H'FA1C	LAFML	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0		
H'FA1D		LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8		
H'FA1E	LAFMH	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1	LAFMH0		
H'FA1F		LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9	LAFMH8		
H'FA20	MC0[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN1	8/16
H'FA21	MC0[2]	—	—	—	—	—	—	—	—		
H'FA22	MC0[3]	—	—	—	—	—	—	—	—		
H'FA23	MC0[4]	—	—	—	—	—	—	—	—		
H'FA24	MC0[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA25	MC0[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA26	MC0[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA27	MC0[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA28	MC1[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'FA29	MC1[2]	—	—	—	—	—	—	—	—		
H'FA2A	MC1[3]	—	—	—	—	—	—	—	—		
H'FA2B	MC1[4]	—	—	—	—	—	—	—	—		
H'FA2C	MC1[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA2D	MC1[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA2E	MC1[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA2F	MC1[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA30	MC2[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN1	8/16
H'FA31	MC2[2]	—	—	—	—	—	—	—	—		
H'FA32	MC2[3]	—	—	—	—	—	—	—	—		
H'FA33	MC2[4]	—	—	—	—	—	—	—	—		
H'FA34	MC2[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA35	MC2[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA36	MC2[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA37	MC2[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA38	MC3[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'FA39	MC3[2]	—	—	—	—	—	—	—	—		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FA3A	MC3[3]	—	—	—	—	—	—	—	—	HCAN1	8/16
H'FA3B	MC3[4]	—	—	—	—	—	—	—	—		
H'FA3C	MC3[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA3D	MC3[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA3E	MC3[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA3F	MC3[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA40	MC4[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN1	8/16
H'FA41	MC4[2]	—	—	—	—	—	—	—	—		
H'FA42	MC4[3]	—	—	—	—	—	—	—	—		
H'FA43	MC4[4]	—	—	—	—	—	—	—	—		
H'FA44	MC4[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA45	MC4[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA46	MC4[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA47	MC4[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA48	MC5[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'FA49	MC5[2]	—	—	—	—	—	—	—	—		
H'FA4A	MC5[3]	—	—	—	—	—	—	—	—		
H'FA4B	MC5[4]	—	—	—	—	—	—	—	—		
H'FA4C	MC5[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA4D	MC5[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA4E	MC5[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA4F	MC5[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA50	MC6[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN1	8/16
H'FA51	MC6[2]	—	—	—	—	—	—	—	—		
H'FA52	MC6[3]	—	—	—	—	—	—	—	—		
H'FA53	MC6[4]	—	—	—	—	—	—	—	—		
H'FA54	MC6[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA55	MC6[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA56	MC6[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA57	MC6[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA58	MC7[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'FA59	MC7[2]	—	—	—	—	—	—	—	—		
H'FA5A	MC7[3]	—	—	—	—	—	—	—	—		
H'FA5B	MC7[4]	—	—	—	—	—	—	—	—		
H'FA5C	MC7[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA5D	MC7[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA5E	MC7[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA5F	MC7[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FA60	MC8[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN1	8/16
H'FA61	MC8[2]	—	—	—	—	—	—	—	—		
H'FA62	MC8[3]	—	—	—	—	—	—	—	—		
H'FA63	MC8[4]	—	—	—	—	—	—	—	—		
H'FA64	MC8[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA65	MC8[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA66	MC8[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA67	MC8[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA68	MC9[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'FA69	MC9[2]	—	—	—	—	—	—	—	—		
H'FA6A	MC9[3]	—	—	—	—	—	—	—	—		
H'FA6B	MC9[4]	—	—	—	—	—	—	—	—		
H'FA6C	MC9[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA6D	MC9[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA6E	MC9[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA6F	MC9[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA70	MC10[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN1	8/16
H'FA71	MC10[2]	—	—	—	—	—	—	—	—		
H'FA72	MC10[3]	—	—	—	—	—	—	—	—		
H'FA73	MC10[4]	—	—	—	—	—	—	—	—		
H'FA74	MC10[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA75	MC10[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA76	MC10[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA77	MC10[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA78	MC11[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'FA79	MC11[2]	—	—	—	—	—	—	—	—		
H'FA7A	MC11[3]	—	—	—	—	—	—	—	—		
H'FA7B	MC11[4]	—	—	—	—	—	—	—	—		
H'FA7C	MC11[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA7D	MC11[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA7E	MC11[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA7F	MC11[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA80	MC12[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN1	8/16
H'FA81	MC12[2]	—	—	—	—	—	—	—	—		
H'FA82	MC12[3]	—	—	—	—	—	—	—	—		
H'FA83	MC12[4]	—	—	—	—	—	—	—	—		
H'FA84	MC12[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA85	MC12[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA86	MC12[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA87	MC12[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA88	MC13[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'FA89	MC13[2]	—	—	—	—	—	—	—	—		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FA8A	MC13[3]	—	—	—	—	—	—	—	—	HCAN1	8/16
H'FA8B	MC13[4]	—	—	—	—	—	—	—	—		
H'FA8C	MC13[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA8D	MC13[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA8E	MC13[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA8F	MC13[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA90	MC14[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0	HCAN1	8/16
H'FA91	MC14[2]	—	—	—	—	—	—	—	—		
H'FA92	MC14[3]	—	—	—	—	—	—	—	—		
H'FA93	MC14[4]	—	—	—	—	—	—	—	—		
H'FA94	MC14[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA95	MC14[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA96	MC14[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA97	MC14[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FA98	MC15[1]	—	—	—	—	DLC3	DLC2	DLC1	DLC0		
H'FA99	MC15[2]	—	—	—	—	—	—	—	—		
H'FA9A	MC15[3]	—	—	—	—	—	—	—	—		
H'FA9B	MC15[4]	—	—	—	—	—	—	—	—		
H'FA9C	MC15[5]	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16		
H'FA9D	MC15[6]	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3		
H'FA9E	MC15[7]	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0		
H'FA9F	MC15[8]	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8		
H'FAB0	MD01									HCAN1	8/16
H'FAB1	MD02										
H'FAB2	MD03										
H'FAB3	MD04										
H'FAB4	MD05										
H'FAB5	MD06										
H'FAB6	MD07										
H'FAB7	MD08										
H'FAB8	MD11										
H'FAB9	MD12										
H'FABA	MD13										
H'FABB	MD14										
H'FABC	MD15										
H'FABD	MD16										
H'FABE	MD17										
H'FABF	MD18										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FAC0	MD21									HCAN1	8/16
H'FAC1	MD22										
H'FAC2	MD23										
H'FAC3	MD24										
H'FAC4	MD25										
H'FAC5	MD26										
H'FAC6	MD27										
H'FAC7	MD28										
H'FAC8	MD31										
H'FAC9	MD32										
H'FACA	MD33										
H'FACB	MD34										
H'FACC	MD35										
H'FACD	MD36										
H'FACE	MD37										
H'FACF	MD38										
H'FAD0	MD41									HCAN1	8/16
H'FAD1	MD42										
H'FAD2	MD43										
H'FAD3	MD44										
H'FAD4	MD45										
H'FAD5	MD46										
H'FAD6	MD47										
H'FAD7	MD48										
H'FAD8	MD51										
H'FAD9	MD52										
H'FADA	MD53										
H'FADB	MD54										
H'FADC	MD55										
H'FADD	MD56										
H'FADE	MD57										
H'FADF	MD58										
H'FAE0	MD61									HCAN1	8/16
H'FAE1	MD62										
H'FAE2	MD63										
H'FAE3	MD64										
H'FAE4	MD65										
H'FAE5	MD66										
H'FAE6	MD67										
H'FAE7	MD68										
H'FAE8	MD71										
H'FAE9	MD72										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FAEA	MD73									HCAN1	8/16
H'FAEB	MD74										
H'FAEC	MD75										
H'FAED	MD76										
H'FAEE	MD77										
H'FAEF	MD78										
H'FAF0	MD81									HCAN1	8/16
H'FAF1	MD82										
H'FAF2	MD83										
H'FAF3	MD84										
H'FAF4	MD85										
H'FAF5	MD86										
H'FAF6	MD87										
H'FAF7	MD88										
H'FAF8	MD91										
H'FAF9	MD92										
H'FAFA	MD93										
H'FAFB	MD94										
H'FAFC	MD95										
H'FAFD	MD96										
H'FAFE	MD97										
H'FAFF	MD98										
H'FB00	MD101									HCAN1	8/16
H'FB01	MD102										
H'FB02	MD103										
H'FB03	MD104										
H'FB04	MD105										
H'FB05	MD106										
H'FB06	MD107										
H'FB07	MD108										
H'FB08	MD111										
H'FB09	MD112										
H'FB0A	MD113										
H'FB0B	MD114										
H'FB0C	MD115										
H'FB0D	MD116										
H'FB0E	MD117										
H'FB0F	MD118										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FB10	MD121									HCAN1	8/16
H'FB11	MD122										
H'FB12	MD123										
H'FB13	MD124										
H'FB14	MD125										
H'FB15	MD126										
H'FB16	MD127										
H'FB17	MD128										
H'FB18	MD131										
H'FB19	MD132										
H'FB1A	MD133										
H'FB1B	MD134										
H'FB1C	MD135										
H'FB1D	MD136										
H'FB1E	MD137										
H'FB1F	MD138										
H'FB20	MD141									HCAN1	8/16
H'FB21	MD142										
H'FB22	MD143										
H'FB23	MD144										
H'FB24	MD145										
H'FB25	MD146										
H'FB26	MD147										
H'FB27	MD148										
H'FB28	MD151										
H'FB29	MD152										
H'FB2A	MD153										
H'FB2B	MD154										
H'FB2C	MD155										
H'FB2D	MD156										
H'FB2E	MD157										
H'FB2F	MD158										

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FC00	PWCR1	—	—	IE	CMF	CST	CKS2	CKS1	CKS0	Motor control	16
H'FC02	PWOCR1	OE1H	OE1G	OE1F	OE1E	OE1D	OE1C	OE1B	OE1A	PWM timer 1	
H'FC04	PWPR1	OPS1H	OPS1G	OPS1F	OPS1E	OPS1D	OPS1C	OPS1B	OPS1A		
H'FC06	PWCYR1	—	—	—	—	—	—				
H'FC08	PWBFR1A	—	—	—	OTS	—	—	DT9	DT8		
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0		
H'FC0A	PWBFR1C	—	—	—	OTS	—	—	DT9	DT8		
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0		
H'FC0C	PWBFR1E	—	—	—	OTS	—	—	DT9	DT8		
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0		
H'FC0E	PWBFR1G	—	—	—	OTS	—	—	DT9	DT8		
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0		
H'FC10	PWCR2	—	—	IE	CMF	CST	CKS2	CKS1	CKS0	Motor control	16
H'FC12	PWOCR2	OE2H	OE2G	OE2F	OE2E	OE2D	OE2C	OE2B	OE2A	PWM timer 2	
H'FC14	PWPR2	OPS2H	OPS2G	OPS2F	OPS2E	OPS2D	OPS2C	OPS2B	OPS2A		
H'FC16	PWCYR2	—	—	—	—	—	—				
H'FC18	PWBFR2A	—	—	—	TDS	—	—	DT9	DT8		
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0		
H'FC1A	PWBFR2B	—	—	—	TDS	—	—	DT9	DT8		
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0		
H'FC1C	PWBFR2C	—	—	—	TDS	—	—	DT9	DT8		
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0		
H'FC1E	PWBFR2D	—	—	—	TDS	—	—	DT9	DT8		
		DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0		
H'FC20	PHDDR	PH7DDR	PH6DDR	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR	PORT	16
H'FC21	PJDDR	PJ7DDR	PJ6DDR	PJ5DDR	PJ4DDR	PJ3DDR	PJ2DDR	PJ1DDR	PJ0DDR		
H'FC24	PHDR	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR		
H'FC25	PJDR	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR		
H'FC28	PORTH	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0		
H'FC29	PORTJ	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FC60	MSTPCRD	MSTPD7	—	—	—	—	—	—	—	SYSTEM	8
H'FDB4	SCKX*[4]	—	IICX1	IICX0	IICE	—	—	—	—	IIC*[4]	8
H'FDB5	DDCSWR*[4]	SWE	SW	IE	IF	CLR3	CLR2	CLR1	CLR0		
H'FDE4	SBYCR	SSBY	STS2	STS1	STS0	OPE	—	—	—	SYSTEM	8
H'FDE5	SYSCR	MACS	—	INTM1	INTM0	NMIEG	—	—	RAME		
H'FDE6	SCKCR	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0		
H'FDE7	MDCR	—	—	—	—	—	MDS2	MDS1	MDS1		
H'FDE8	MSTPCRA	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0		
H'FDE9	MSTPCRB	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0		
H'FDEA	MSTPCRC	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0		
H'FDEB	PFCR	—	—	—	—	AE3	AE2	AE1	AE0		
H'FDEC	LPWRCR	DTON*[3]	LSON*[3]	NESEL*[3]	SUBSTP*[3]	RFCUT*[3]	—	STC1	STC0		
H'FE00	BARA	—	—	—	—	—	—	—	—	PBC	8
H'FE01		BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16		
H'FE02		BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8		
H'FE03		BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0		
H'FE04	BARB	—	—	—	—	—	—	—	—		
H'FE05		BAA23	BAA22	BAA21	BAA20	BAA19	BAA18	BAA17	BAA16		
H'FE06		BAA15	BAA14	BAA13	BAA12	BAA11	BAA10	BAA9	BAA8		
H'FE07		BAA7	BAA6	BAA5	BAA4	BAA3	BAA2	BAA1	BAA0		
H'FE08	BCRA	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA		
H'FE09	BCRB	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA		
H'FE12	ISCRH	—	—	—	—	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA	INT	8
H'FE13	ISCRL	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA		
H'FE14	IER	—	—	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E		
H'FE15	ISR	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F		
H'FE16	DTCERA	DTCEA7	DTCEA6	DTCEA5	DTCEA4	DTCEA3	DTCEA2	DTCEA1	DTCEA0	DTC	8
H'FE17	DTCERB	DTCEB7	DTCEB6	DTCEB5	DTCEB4	DTCEB3	DTCEB2	DTCEB1	DTCEB0		
H'FE18	DTCERC	DTCEC7	DTCEC6	DTCEC5	DTCEC4	DTCEC3	DTCEC2	DTCEC1	DTCEC0		
H'FE19	DTCERD	DTCED7	DTCED6	DTCED5	DTCED4	DTCED3	DTCED2	DTCED1	DTCED0		
H'FE1A	DTCERE	DTCEE7	DTCEE6	DTCEE5	DTCEE4	DTCEE3	DTCEE2	DTCEE1	DTCEE0		
H'FE1B	DTCERF	DTCEF7	DTCEF6	DTCEF5	DTCEF4	DTCEF3	DTCEF2	DTCEF1	DTCEF0		
H'FE1C	DTCERG	DTCEG7	DTCEG6	DTCEG5	DTCEG4	DTCEG3	DTCEG2	DTCEG1	DTCEG0		
H'FE1F	DTVECR	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0		
H'FE26	PCR	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0	PPG	8
H'FE27	PMR	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV		
H'FE28	NDERH	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8		
H'FE29	NDERL	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0		
H'FE2A	PODRH	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8		
H'FE2B	PODRL	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0		
H'FE2C	NDRH	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8		
H'FE2D	NDRL	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FE2E	NDRH	—	—	—	—	NDR11	NDR10	NDR9	NDR8	PPG	8
H'FE2F	NDRL	—	—	—	—	NDR3	NDR2	NDR1	NDR0		
H'FE30	P1DDR	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR	PORT	8
H'FE32	P3DDR	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR		
H'FE39	PADDR	—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR		
H'FE3A	PBDDR	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR		
H'FE3B	PCDDR	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR		
H'FE3C	PDDDR	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR		
H'FE3D	PEDDR	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR		
H'FE3E	PFDDR	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	—	—	PF0DDR		
H'FE40	PAPCR	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR	PORT	8
H'FE41	PBPCR	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR		
H'FE42	PCPCR	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR		
H'FE43	PDPCR	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR		
H'FE44	PEPCR	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR		
H'FE46	P3ODR	—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR		
H'FE47	PAODR	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR		
H'FE48	PBODR	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR		
H'FE49	PCODR	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR		
H'FE80	TCR3	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU3	8/16
H'FE81	TMDR3	—	—	BFB	BFA	MD3	MD2	MD1	MD0		
H'FE82	TIOR3H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FE83	TIOR3L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0		
H'FE84	TIER3	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
H'FE85	TSR3	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA		
H'FE86	TCNT3										
H'FE87											
H'FE88	TGR3A										
H'FE89											
H'FE8A	TGR3B										
H'FE8B											
H'FE8C	TGR3C										
H'FE8D											
H'FE8E	TGR3D										
H'FE8F											
H'FE90	TCR4	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU4	8/16
H'FE91	TMDR4	—	—	—	—	MD3	MD2	MD1	MD0		
H'FE92	TIOR4	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FE94	TIER4	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FE95	TSR4	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FE96	TCNT4									TPU4	8/16
H'FE97											
H'FE98	TGR4A										
H'FE99											
H'FE9A	TGR4B										
H'FE9B											
H'FEA0	TCR5	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU5	8/16
H'FEA1	TMDR5	—	—	—	—	MD3	MD2	MD1	MD0		
H'FEA2	TIOR5	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FEA4	TIER5	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FEA5	TSR5	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FEA6	TCNT5										
H'FEA7											
H'FEA8	TGR5A										
H'FEA9											
H'FEAA	TGR5B										
H'FEAB											
H'FEB0	TSTR	—	—	CST5	CST4	CST3	CST2	CST1	CST0	TPU All	8
H'FEB1	TSYR	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0		
H'FEC0	IPRA	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0	INT	8
H'FEC1	IPRB	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC2	IPRC	—	—	—	—	—	IPR2	IPR1	IPR0		
H'FEC3	IPRD	—	IPR6	IPR5	IPR4	—	—	—	—		
H'FEC4	IPRE	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC5	IPRF	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC6	IPRG	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC7	IPRH	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FEC9	IPRJ	—	—	—	—	—	IPR2	IPR1	IPR0		
H'FECA	IPRK	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECC	IPRM	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0		
H'FECE	Reserved										
H'FED0	ABWCR	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0	Bus controller	8
H'FED1	ASTCR	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0		
H'FED2	WCRH	W71	W70	W61	W60	W51	W50	W41	W40		
H'FED3	WCRL	W31	W30	W21	W20	W11	W10	W01	W00		
H'FED4	BCRH	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—		
H'FED5	BCRL	—	—	—	—	—	—	WDBE	—		
H'FEDB	RAMER	—	—	—	—	RAMS	RAM2	RAM1	RAM0	ROM	8
H'FF00	P1DR	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR	PORT	8
H'FF02	P3DR	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR		
H'FF09	PADR	—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FF0A	PBDR	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR	PORT	8
H'FF0B	PCDR	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR		
H'FF0C	PDDR	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR		
H'FF0D	PEDR	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR		
H'FF0E	PFDR	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	—	—	PF0DR		
H'FF10	TCR0	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU0	8/16
H'FF11	TMDR0	—	—	BFB	BFA	MD3	MD2	MD1	MD0		
H'FF12	TIOR0H	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FF13	TIOR0L	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0		
H'FF14	TIER0	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA		
H'FF15	TSR0	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA		
H'FF16	TNCT0										
H'FF17											
H'FF18	TGR0A										
H'FF19											
H'FF1A	TGR0B										
H'FF1B											
H'FF1C	TGR0C										
H'FF1D											
H'FF1E	TGR0D										
H'FF1F											
H'FF20	TCR1	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU1	8/16
H'FF21	TMDR1	—	—	—	—	MD3	MD2	MD1	MD0		
H'FF22	TIOR1	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FF24	TIER1	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FF25	TSR1	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FF26	TNCT1										
H'FF27											
H'FF28	TGR1A										
H'FF29											
H'FF2A	TGR1B										
H'FF2B											
H'FF30	TCR2	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0	TPU2	8/16
H'FF31	TMDR2	—	—	—	—	MD3	MD2	MD1	MD0		
H'FF32	TIOR2	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0		
H'FF34	TIER2	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA		
H'FF35	TSR2	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA		
H'FF36	TNCT2										
H'FF37											
H'FF38	TGR2A										
H'FF39											
H'FF3A	TGR2B										
H'FF3B											

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FF74 (read/write)	TCSR0	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0	WDT0	8
H'FF75 (read)	TCNT0										
H'FF76	—	—	—	—	—	—	—	—	—		
H'FF77 (read)	RSTCSR0	WOVF	RSTE	—	—	—	—	—	—		
H'FF78	SMR0	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI0, IIC0, Smart card interface 0	8
	SMR0	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0		
	ICCR0*[4]	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP		
H'FF79	BRR0										
	ICSR0*[4]	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB		
H'FF7A	SCR0	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF7B	TDR0										
H'FF7C	SSR0	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
	SSR0	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT		
H'FF7D	RDR0										
H'FF7E	SCMR0	—	—	—	—	SDIR	SINV	—	SMIF		
	ICDR0/ SARX0*[4]	ICDR7/ SVAX6	ICDR6/ SVAX5	ICDR5/ SVAX4	ICDR4/ SVAX3	ICDR3/ SVAX2	ICDR2/ SVAX1	ICDR1/ SVAX0	ICDR0/FSX		
H'FF7F	ICMR0/ SAR0	MLS/ SVA6	WAIT/ SVA5	CKS2/ SVA4	CKS1/ SVA3	CKS0/ SVA2	BC2/ SVA1	BC1/ SVA0	BC0/FS	IIC0*[4]	
H'FF80	SMR1	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI1, IIC1, Smart card interface 1	8
	SMR1	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0		
	ICCR1*[4]	ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP		
H'FF81	BRR1										
	ICSR1*[4]	ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB		
H'FF82	SCR1	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF83	TDR1										
H'FF84	SSR1	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
	SSR1	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT		
H'FF85	RDR1										
H'FF86	SCMR1	—	—	—	—	SDIR	SINV	—	SMIF		
	ICDR1/ SARX1*[4]	ICDR7/ SVARX6	ICDR6/ SVARX5	ICDR5/ SVARX4	ICDR4/ SVARX3	ICDR3/ SVARX2	ICDR2/ SVARX1	ICDR1/ SVARX0	ICDR0/FSX		
H'FF87	ICMR1/ SAR1*[4]	MLS/ SVA6	WAIT/ SVA5	CKS2/ SVA4	CKS1/ SVA3	CKS0/ SVA2	BC2/ SVA1	BC1/ SVA0	BC0/FS	IIC1*[4]	
H'FF88	SMR2	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0	SCI2, Smart card interface 2	8
	SMR2	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0		
H'FF89	BRR2										
H'FF8A	SCR2	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0		
H'FF8B	TDR2										
H'FF8C	SSR2	TDRE	RDRF	ORER	FER	PER	TEND	MPB	MPBT		
	SSR2	TDRE	RDRF	ORER	ERS	PER	TEND	MPB	MPBT		
H'FF8D	RDR2										
H'FF8E	SCMR2	—	—	—	—	SDIR	SINV	—	SMIF		

Address	Register Name	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Module Name	Data Bus Width
H'FF90	ADDRAH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	A/D	8
H'FF91	ADDRAL	AD1	AD0	—	—	—	—	—	—		
H'FF92	ADDRBH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FF93	ADDRBL	AD1	AD0	—	—	—	—	—	—		
H'FF94	ADDRCH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FF95	ADDRCL	AD1	AD0	—	—	—	—	—	—		
H'FF96	ADDRDH	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2		
H'FF97	ADDRDL	AD1	AD0	—	—	—	—	—	—		
H'FF98	ADCSR	ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0		
H'FF99	ADCR	TRGS1	TRGS0	—	—	CKS1	CKS0	—	—		
H'FFA2 (read/write)	TCSR1	OVF	WT/$\overline{IT}$	TME	PSS*[1]	RST/$\overline{NMI}$	CKS2	CKS1	CKS0	WDT1	16
H'FFA3 (read)	TCNT1										
H'FFA4	DADR0									D/A0, 1	8
H'FFA5	DADR1										
H'FFA6	DACR01	DAOE1	DAOE0	DAE	—	—	—	—	—		
H'FFA8	FLMCR1	FWE	SWE	ESU	PSU	EV	PV	E	P	FLASH	8
H'FFA9	FLMCR2	FLER	—	—	—	—	—	—	—		
H'FFAA	EBR1	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0		
H'FFAB	EBR2	—	—	—	—	EB11*[5]	EB10*[5]	EB9	EB8		
H'FFAC	FLPWCR	PDWND*[2]	—	—	—	—	—	—	—		
H'FFB0	PORT1	P17	P16	P15	P14	P13	P12	P11	P10	PORT	8
H'FFB2	PORT3	—	—	P35	P34	P33	P32	P31	P30		
H'FFB3	PORT4	P47	P46	P45	P44	P43	P42	P41	P40		
H'FFB8	PORT9	—	—	—	—	P93	P92	P91	P90		
H'FFB9	PORTA	—	—	—	—	PA3	PA2	PA1	PA0		
H'FFBA	PORTB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0		
H'FFBB	PORTC	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0		
H'FFBC	PORTD	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0		
H'FFBD	PORTE	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0		
H'FFBE	PORTF	PF7	PF6	PF5	PF4	PF3	—	—	PF0		

Notes: *1 Bit 4 (PSS) in TCSR of WDT1 is valid in the U-mask and W-mask versions. In versions other than the U-mask and W-mask versions, however, the PSS bit must always be written with 0 since no subclock functions are available.

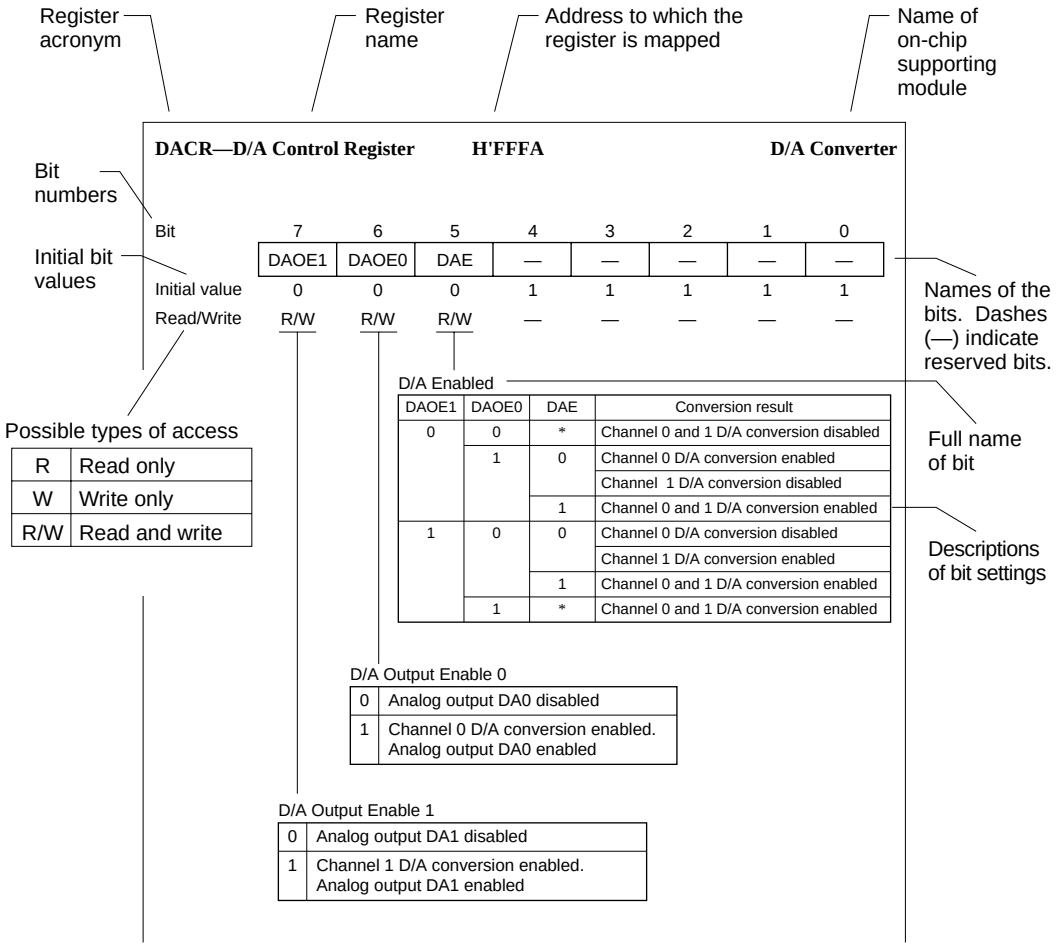
*2 Subclock functions (subactive mode, subsleep mode, and watch mode) are not available in versions other than the U-mask and W-mask versions. Subclock functions may be used with the U-mask and W-mask versions.

*3 Bits DTON, LSON, NESEL, and SUBSTP in LPWRCR are valid in the U-mask and W-mask versions. In versions other than the U-mask and W-mask versions, however, these bits must always be written with 0 since no subclock functions are available.

*4 An I^2C bus interface can only be added to the H8S/2638 and H8S/2639. Therefore, IIC related registers are valid only in the H8S/2638 and H8S/2639.

*5 These bits are valid in the H8S/2638 and H8S/2639 only. They are reserved bits in the H8S/2636.

B.2 Functions



DAOE1	DAOE0	DAE	Conversion result
0	0	*	Channel 0 and 1 D/A conversion disabled
	1	0	Channel 0 D/A conversion enabled
			Channel 1 D/A conversion disabled
		1	Channel 0 and 1 D/A conversion enabled
1	0	0	Channel 0 D/A conversion disabled
			Channel 1 D/A conversion enabled
		1	Channel 0 and 1 D/A conversion enabled
	1	*	Channel 0 and 1 D/A conversion enabled

Bit	7	6	5	4	3	2	1	0
	SM1	SM0	DM1	DM0	MD1	MD0	DTS	Sz
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	—	—	—

DTC Data Transfer Size

0	Byte-size transfer
1	Word-size transfer

DTC Transfer Mode Select

0	Destination side is repeat area or block area
1	Source side is repeat area or block area

DTC Mode

0	0	Normal mode
	1	Repeat mode
1	0	Block transfer mode
	1	—

Destination Address Mode

0	—	DAR is fixed
1	0	DAR is incremented after a transfer (by 1 when Sz = 0; by 2 when Sz = 1)
	1	DAR is decremented after a transfer (by 1 when Sz = 0; by 2 when Sz = 1)

Source Address Mode

0	—	SAR is fixed
1	0	SAR is incremented after a transfer (by 1 when Sz = 0; by 2 when Sz = 1)
	1	SAR is decremented after a transfer (by 1 when Sz = 0; by 2 when Sz = 1)

Bit	7	6	5	4	3	2	1	0
	CHNE	DISEL	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	—	—	—

DTC Interrupt Select

0	After a data transfer ends, the CPU interrupt is disabled unless the transfer counter is 0
1	After a data transfer ends, the CPU interrupt is enabled

DTC Chain Transfer Enable

0	End of DTC data transfer
1	DTC chain transfer

SAR—DTC Source Address Register **H'EBC0–H'EFBF** **DTC**

Bit	23	22	21	20	19	- - -	4	3	2	1	0
						- - -					
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	- - -	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	- - -	—	—	—	—	—

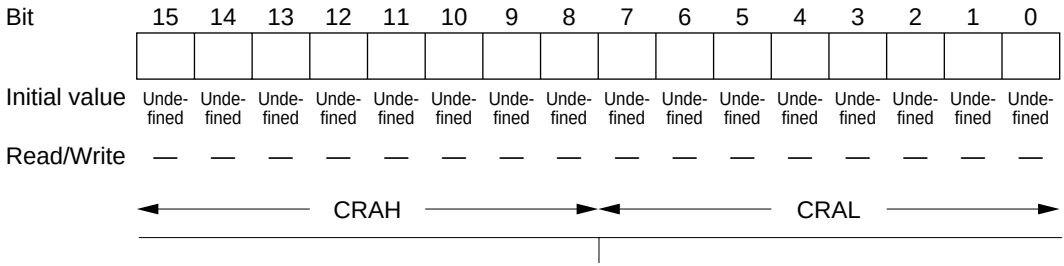
Specify DTC transfer data source address

DAR—DTC Destination Address Register **H'EBC0–H'EFBF** **DTC**

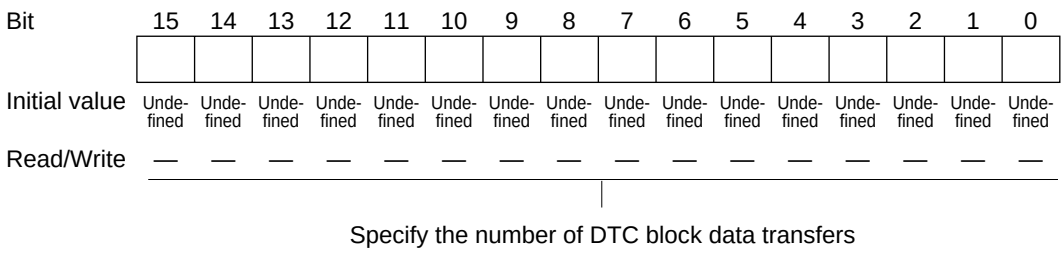
Bit	23	22	21	20	19	- - -	4	3	2	1	0
						- - -					
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	- - -	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	—	—	—	—	—	- - -	—	—	—	—	—

Specify DTC transfer data destination address

CRA—DTC Transfer Count Register A **H'EBC0–H'EFBF** **DTC**



CRB—DTC Transfer Count Register B **H'EBC0–H'EFBF** **DTC**



Bit	7	6	5	4	3	2	1	0
	MCR7	—	MCR5	—	—	MCR2	MCR1	MCR0
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R	R/W	R	R	R/W	R/W	R/W

Reset Request

0	Normal operating mode (MCR0 = 0 and GSR3 = 0) [Setting condition] When 0 is written after an HCAN reset
1	HCAN reset mode transition request

Halt Request

0	HCAN normal operating mode
1	HCAN halt mode transition request

Message Transmission Method

0	Transmission order determined by message identifier priority
1	Transmission order determined by mailbox (buffer) number priority (TXPR1 > TXPR15)

HCAN Sleep Mode

0	HCAN sleep mode released
1	Transition to HCAN sleep mode enabled

HCAN Sleep Mode Release

0	HCAN sleep mode release by CAN bus operation disabled
1	HCAN sleep mode release by CAN bus operation enabled

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	GSR3	GSR2	GSR1	GSR0
Initial value	0	0	0	0	1	1	0	0
Read/Write	R	R	R	R	R	R	R	R

Bus Off Flag

0	[Reset condition] Recovery from bus off state
1	When TEC ≥ 256 (bus off state)

Transmit/Receive Warning Flag

0	[Reset condition] When TEC < 96 and REC < 96 or TEC ≥ 256
1	When TEC ≥ 96 or REC ≥ 96

Message Transmission Status Flag

0	Message transmission period
1	[Reset condition] Idle state

Reset Status Bit

0	Normal operating state [Setting condition] After an HCAN internal reset
1	Configuration mode [Reset condition] MCR0 reset mode and sleep mode

Bit	15	14	13	12	11	10	9	8
	BCR7	BCR6	BCR5	BCR4	BCR3	BCR2	BCR1	BCR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Resynchronization Jump Width

0	0	Bit synchronization width = 1 time quantum
	1	Bit synchronization width = 2 time quanta
1	0	Bit synchronization width = 3 time quanta
	1	Bit synchronization width = 4 time quanta

Baud Rate Prescaler

0	0	0	0	0	0	2 × system clock
0	0	0	0	0	1	4 × system clock
0	0	0	0	1	0	6 × system clock
:	:	:	:	:	:	:
1	1	1	1	1	1	128 × system clock

Bit	7	6	5	4	3	2	1	0
	BCR15	BCR14	BCR13	BCR12	BCR11	BCR10	BCR9	BCR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time Segment 2

0	0	0	Setting prohibited
		1	TSEG2 = 2 time quanta
	1	0	TSEG2 = 3 time quanta
		1	TSEG2 = 4 time quanta
1	0	0	TSEG2 = 5 time quanta
		1	TSEG2 = 6 time quanta
	1	0	TSEG2 = 7 time quanta
		1	TSEG2 = 8 time quanta

Time Segment 1

0	0	0	0	Setting prohibited
0	0	0	1	Setting prohibited
0	0	1	0	Setting prohibited
0	0	1	1	TSEG1 = 4 time quanta
0	1	0	0	TSEG1 = 5 time quanta
:	:	:	:	:
1	1	1	1	TSEG1 = 16 time quanta

Bit Sample Point

0	Bit sampling at one point (end of time segment 1 (TSEG1))
1	Bit sampling at three points (end of time segment 1 (TSEG1) and preceding and following time quanta)

MBCR0—Mailbox Configuration Register **H'F804** **HCAN0**
MBCR1—Mailbox Configuration Register **H'FA04** **HCAN1**

Bit	15	14	13	12	11	10	9	8
	MBCR7	MBCR6	MBCR5	MBCR4	MBCR3	MBCR2	MBCR1	—
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	7	6	5	4	3	2	1	0
	MBCR15	MBCR14	MBCR13	MBCR12	MBCR11	MBCR10	MBCR9	MBCR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Mailbox Setting Register

0	Corresponding mailbox is set for transmission
1	Corresponding mailbox is set for reception

TXPR0—Transmit Wait Register **H'F806** **HCAN0**
TXPR1—Transmit Wait Register **H'FA06** **HCAN1**

Bit	15	14	13	12	11	10	9	8
	TXPR7	TXPR6	TXPR5	TXPR4	TXPR3	TXPR2	TXPR1	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	7	6	5	4	3	2	1	0
	TXPR15	TXPR14	TXPR13	TXPR12	TXPR11	TXPR10	TXPR9	TXPR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transmit Wait Register

0	Transmit message idle state in corresponding mailbox [Clearing condition] Message transmission completion and cancellation completion
1	Transmit message transmit wait in corresponding mailbox (CAN bus arbitration)

TXCR0—Transmit Wait Cancel Register **H'F808** **HCAN0**
TXCR1—Transmit Wait Cancel Register **H'FA08** **HCAN1**

Bit	15	14	13	12	11	10	9	8
	TXCR7	TXCR6	TXCR5	TXCR4	TXCR3	TXCR2	TXCR1	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	7	6	5	4	3	2	1	0
	TXCR15	TXCR14	TXCR13	TXCR12	TXCR11	TXCR10	TXCR9	TXCR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transmit Wait Cancel Register

0	Transmit message cancellation idle state in corresponding mailbox [Clearing condition] Completion of TXPR clearing (when transmit message is canceled normally)
1	TXPR cleared for corresponding mailbox (transmit message cancellation)

TXACK0—Transmit Acknowledge Register **H'F80A** **HCAN0**
TXACK1—Transmit Acknowledge Register **H'FA04** **HCAN1**

Bit	15	14	13	12	11	10	9	8
	TXACK7	TXACK6	TXACK5	TXACK4	TXACK3	TXACK2	TXACK1	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	7	6	5	4	3	2	1	0
	TXACK15	TXACK14	TXACK13	TXACK12	TXACK11	TXACK10	TXACK9	TXACK8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Transmit Acknowledge Register

0	[Clearing condition] Writing 1
1	Completion of message transmission for corresponding mailbox

ABACK0—Abort Acknowledge Register **H'F80C** **HCAN0**
ABACK1—Abort Acknowledge Register **H'FA0C** **HCAN1**

Bit	15	14	13	12	11	10	9	8
	ABACK7	ABACK6	ABACK5	ABACK4	ABACK3	ABACK2	ABACK1	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Bit	7	6	5	4	3	2	1	0
	ABACK15	ABACK14	ABACK13	ABACK12	ABACK11	ABACK10	ABACK9	ABACK8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Abort Acknowledge Register

0	[Clearing condition] Writing 1
1	Completion of transmit message cancellation for corresponding mailbox

RXPR0—Receive Complete Register **H'F80E** **HCAN0**
RXPR1—Receive Complete Register **H'FA0E** **HCAN1**

Bit	15	14	13	12	11	10	9	8
	RXPR7	RXPR6	RXPR5	RXPR4	RXPR3	RXPR2	RXPR1	RXPR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	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
	RXPR15	RXPR14	RXPR13	RXPR12	RXPR11	RXPR10	RXPR9	RXPR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Receive Complete Register

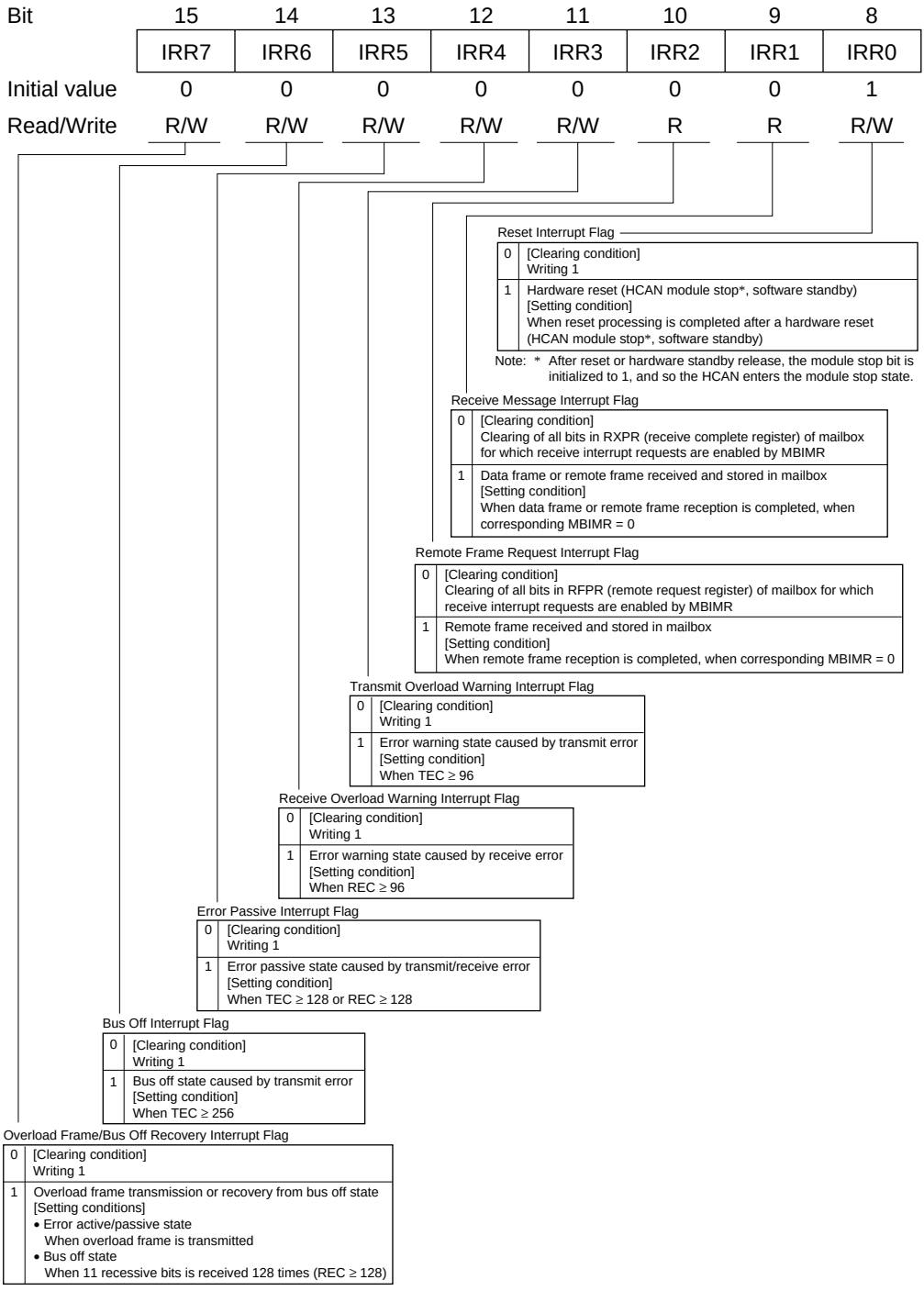
0	[Clearing condition] Writing 1
1	Completion of message (data frame or remote frame) reception in corresponding mailbox

Bit	15	14	13	12	11	10	9	8
	RFPR7	RFPR6	RFPR5	RFPR4	RFPR3	RFPR2	RFPR1	RFPR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	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
	RFPR15	RFPR14	RFPR13	RFPR12	RFPR11	RFPR10	RFPR9	RFPR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Remote Request Register

0	[Clearing condition] Writing 1
1	Completion of remote frame reception in corresponding mailbox


Bit
15
14
13
12
11
10
9
8
IRR7
IRR6
IRR5
IRR4
IRR3
IRR2
IRR1
IRR0
Initial value
0
0
0
0
0
0
0
1
Read/Write
R/W
R/W
R/W
R/W
R/W
R
R
R/W
Reset Interrupt Flag
0 [Clearing condition] Writing 1
1 Hardware reset (HCAN module stop*, software standby) [Setting condition] When reset processing is completed after a hardware reset (HCAN module stop*, software standby)
Note: * After reset or hardware standby release, the module stop bit is initialized to 1, and so the HCAN enters the module stop state.
Receive Message Interrupt Flag
0 [Clearing condition] Clearing of all bits in RXPR (receive complete register) of mailbox for which receive interrupt requests are enabled by MBIMR
1 Data frame or remote frame received and stored in mailbox [Setting condition] When data frame or remote frame reception is completed, when corresponding MBIMR = 0
Remote Frame Request Interrupt Flag
0 [Clearing condition] Clearing of all bits in RFPR (remote request register) of mailbox for which receive interrupt requests are enabled by MBIMR
1 Remote frame received and stored in mailbox [Setting condition] When remote frame reception is completed, when corresponding MBIMR = 0
Transmit Overload Warning Interrupt Flag
0 [Clearing condition] Writing 1
1 Error warning state caused by transmit error [Setting condition] When TEC ≥ 96
Receive Overload Warning Interrupt Flag
0 [Clearing condition] Writing 1
1 Error warning state caused by receive error [Setting condition] When REC ≥ 96
Error Passive Interrupt Flag
0 [Clearing condition] Writing 1
1 Error passive state caused by transmit/receive error [Setting condition] When TEC ≥ 128 or REC ≥ 128
Bus Off Interrupt Flag
0 [Clearing condition] Writing 1
1 Bus off state caused by transmit error [Setting condition] When TEC ≥ 256
Overload Frame/Bus Off Recovery Interrupt Flag
0 [Clearing condition] Writing 1
1 Overload frame transmission or recovery from bus off state [Setting conditions]
• Error active/passive state When overload frame is transmitted
• Bus off state When 11 recessive bits is received 128 times (REC ≥ 128)

Bit	7	6	5	4	3	2	1	0
	—	—	—	IRR12	—	—	IRR9	IRR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	—	R/W	—	—	R	R/W

Mailbox Empty Interrupt Flag

0	[Clearing condition] Writing 1
1	Transmit message has been transmitted or aborted, and new message can be stored [Setting condition] When TXPR (transmit wait register) is cleared by completion of transmission or completion of transmission abort

Unread Interrupt Flag

0	[Clearing condition] Clearing of all bits in UMSR (unread message status register)
1	Unread message overwrite [Setting condition] When UMSR (unread message status register) is set

Bus Operation Interrupt Flag

0	CAN bus idle state [Clearing condition] Writing 1
1	CAN bus operation in HCAN sleep mode [Setting condition] Bus operation (dominant bit detection) in HCAN sleep mode

Bit	15	14	13	12	11	10	9	8
	MBIMR7	MBIMR6	MBIMR5	MBIMR4	MBIMR3	MBIMR2	MBIMR1	MBIMR0
Initial value	1	1	1	1	1	1	1	1
Read/Write	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
	MBIMR15	MBIMR14	MBIMR13	MBIMR12	MBIMR11	MBIMR10	MBIMR9	MBIMR8
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Mailbox Interrupt Mask

0	[Transmitting] Interrupt request to CPU due to TXPR clearing [Receiving] Interrupt request to CPU due to RXPR setting
1	Interrupt requests to CPU disabled

Bit	15	14	13	12	11	10	9	8
	IMR7	IMR6	IMR5	IMR4	IMR3	IMR2	IMR1	—
Initial value	1	1	1	1	1	1	1	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R

Receive Message Interrupt Mask

0	Message reception interrupt request (RM1) to CPU by IRR1 enabled
1	Message reception interrupt request (RM1) to CPU by IRR1 disabled

Remote Frame Request Interrupt Mask

0	Remote frame reception interrupt request (OVR0) to CPU by IRR2 enabled
1	Remote frame reception interrupt request (OVR0) to CPU by IRR2 disabled

Transmit Overload Warning Interrupt Mask

0	TEC error warning interrupt request (OVR0) to CPU by IRR3 enabled
1	TEC error warning interrupt request (OVR0) to CPU by IRR3 disabled

Receive Overload Warning Interrupt Mask

0	REC error warning interrupt request (OVR0) to CPU by IRR4 enabled
1	REC error warning interrupt request (OVR0) to CPU by IRR4 disabled

Error Passive Interrupt Mask

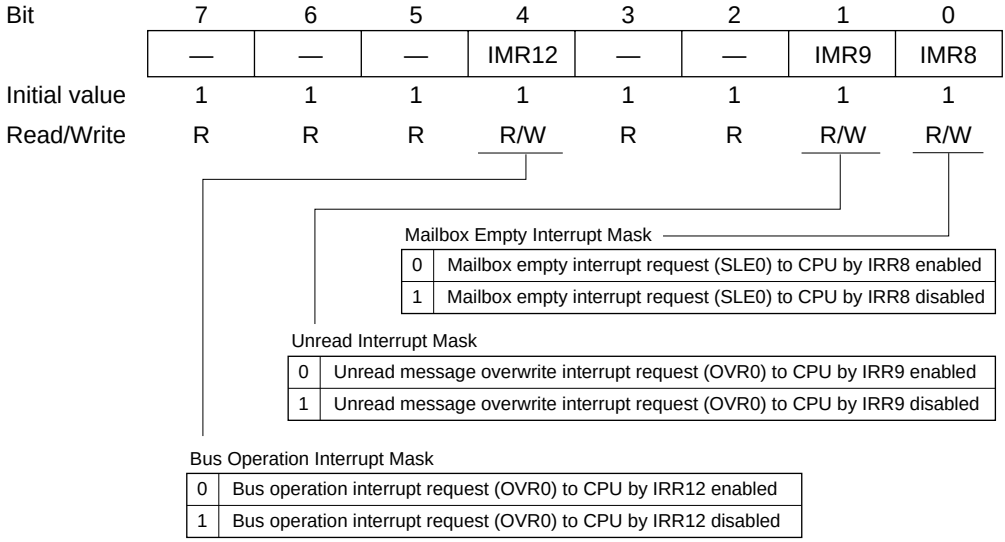
0	Error passive interrupt request to CPU by IRR5 enabled
1	Error passive interrupt request to CPU by IRR5 disabled

Bus Off Interrupt Mask

0	Bus off interrupt request to CPU by IRR6 enabled
1	Bus off interrupt request to CPU by IRR6 disabled

Overload Frame/Bus Off Recovery Interrupt Mask

0	Overload frame/bus off recovery interrupt request to CPU by IRR7 enabled
1	Overload frame/bus off recovery interrupt request to CPU by IRR7 disabled


Bit
7
6
5
4
3
2
1
0
—
—
—
IMR12
—
—
IMR9
IMR8
Initial value
1
1
1
1
1
1
1
1
Read/Write
R
R
R
R/W
R
R
R/W
R/W
Mailbox Empty Interrupt Mask
0 Mailbox empty interrupt request (SLE0) to CPU by IRR8 enabled
1 Mailbox empty interrupt request (SLE0) to CPU by IRR8 disabled
Unread Interrupt Mask
0 Unread message overwrite interrupt request (OVR0) to CPU by IRR9 enabled
1 Unread message overwrite interrupt request (OVR0) to CPU by IRR9 disabled
Bus Operation Interrupt Mask
0 Bus operation interrupt request (OVR0) to CPU by IRR12 enabled
1 Bus operation interrupt request (OVR0) to CPU by IRR12 disabled

REC0—Receive Error Counter **H'F818** **HCAN0**
REC1—Receive Error Counter **H'FA18** **HCAN1**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

TEC0—Transmit Error Counter **H'F819** **HCAN0**
TEC1—Transmit Error Counter **H'FA19** **HCAN1**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

UMSR0—Unread Message Status Register **H'F81A** **HCAN0**
UMSR1—Unread Message Status Register **H'FA1A** **HCAN1**

Bit	15	14	13	12	11	10	9	8
	UMSR7	UMSR6	UMSR5	UMSR4	UMSR3	UMSR2	UMSR1	UMSR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	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
	UMSR15	UMSR14	UMSR13	UMSR12	UMSR11	UMSR10	UMSR9	UMSR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Unread Message Status Flags

0	[Clearing condition] Writing 1
1	Unread receive message is overwritten by a new message [Setting condition] When a new message is received before RXPR is cleared

LAFML0—Local Acceptance Filter Masks L **H'F81C** **HCAN0**
LAFMH0—Local Acceptance Filter Masks H **H'F81E** **HCAN0**
LAFML1—Local Acceptance Filter Masks L **H'FA1C** **HCAN1**
LAFMH1—Local Acceptance Filter Masks H **H'FA1E** **HCAN1**

Bit	15	14	13	12	11	10	9	8
	LAFML7	LAFML6	LAFML5	LAFML4	LAFML3	LAFML2	LAFML1	LAFML0
Initial value	0	0	0	0	0	0	0	0
Read/Write	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
	LAFML15	LAFML14	LAFML13	LAFML12	LAFML11	LAFML10	LAFML9	LAFML8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LAFMH

Bit	15	14	13	12	11	10	9	8
	LAFMH7	LAFMH6	LAFMH5	—	—	—	LAFMH1	LAFMH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R	R	R	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	LAFMH15	LAFMH14	LAFMH13	LAFMH12	LAFMH11	LAFMH10	LAFMH9	LAFMH8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

LAFMH Bits 7 to 0 and 15 to 13–11-Bit Identifier Filter

0	Stored in MC0 and MD0 (receive-only mailbox) depending on bit match between MC0 message identifier and receive message identifier (Care)
1	Stored in MC0 and MD0 (receive-only mailbox) regardless of bit match between MC0 message identifier and receive message identifier (Don't Care)

LAFMH Bits 9 and 8, LAFML bits 15 to 0–18-Bit Identifier Filter

0	Stored in MC0 (receive-only mailbox) depending on bit match between MC0 message identifier and receive message identifier (Care)
1	Stored in MC0 (receive-only mailbox) regardless of bit match between MC0 message identifier and receive message identifier (Don't Care)

MC0[1]—Message Control 0[1]	**H'F820**	**HCAN0**
MC0[2]—Message Control 0[2]	**H'F821**	**HCAN0**
MC0[3]—Message Control 0[3]	**H'F822**	**HCAN0**
MC0[4]—Message Control 0[4]	**H'F823**	**HCAN0**
MC0[5]—Message Control 0[5]	**H'F824**	**HCAN0**
MC0[6]—Message Control 0[6]	**H'F825**	**HCAN0**
MC0[7]—Message Control 0[7]	**H'F826**	**HCAN0**
MC0[8]—Message Control 0[8]	**H'F827**	**HCAN0**

MC0[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 bytes
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC0[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC0[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC0[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC0[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC0[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC0[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC0[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC1[1]—Message Control 1[1]	**H'F828**	**HCAN0**
MC1[2]—Message Control 1[2]	**H'F829**	**HCAN0**
MC1[3]—Message Control 1[3]	**H'F82A**	**HCAN0**
MC1[4]—Message Control 1[4]	**H'F82B**	**HCAN0**
MC1[5]—Message Control 1[5]	**H'F82C**	**HCAN0**
MC1[6]—Message Control 1[6]	**H'F82D**	**HCAN0**
MC1[7]—Message Control 1[7]	**H'F82E**	**HCAN0**
MC1[8]—Message Control 1[8]	**H'F82F**	**HCAN0**

MC1[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC1[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC1[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC1[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC1[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC1[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC1[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC1[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC2[1]—Message Control 2[1]	**H'F830**	**HCAN0**
MC2[2]—Message Control 2[2]	**H'F831**	**HCAN0**
MC2[3]—Message Control 2[3]	**H'F832**	**HCAN0**
MC2[4]—Message Control 2[4]	**H'F833**	**HCAN0**
MC2[5]—Message Control 2[5]	**H'F834**	**HCAN0**
MC2[6]—Message Control 2[6]	**H'F835**	**HCAN0**
MC2[7]—Message Control 2[7]	**H'F836**	**HCAN0**
MC2[8]—Message Control 2[8]	**H'F837**	**HCAN0**

MC2[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC2[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC2[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC2[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC2[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC2[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC2[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC2[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC3[1]—Message Control 3[1]	**H'F838**	**HCAN0**
MC3[2]—Message Control 3[2]	**H'F839**	**HCAN0**
MC3[3]—Message Control 3[3]	**H'F83A**	**HCAN0**
MC3[4]—Message Control 3[4]	**H'F83B**	**HCAN0**
MC3[5]—Message Control 3[5]	**H'F83C**	**HCAN0**
MC3[6]—Message Control 3[6]	**H'F83D**	**HCAN0**
MC3[7]—Message Control 3[7]	**H'F83E**	**HCAN0**
MC3[8]—Message Control 3[8]	**H'F83F**	**HCAN0**

MC3[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC3[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC3[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC3[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC3[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC3[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC3[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC3[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC4[1]—Message Control 4[1]	**H'F840**	**HCAN0**
MC4[2]—Message Control 4[2]	**H'F841**	**HCAN0**
MC4[3]—Message Control 4[3]	**H'F842**	**HCAN0**
MC4[4]—Message Control 4[4]	**H'F843**	**HCAN0**
MC4[5]—Message Control 4[5]	**H'F844**	**HCAN0**
MC4[6]—Message Control 4[6]	**H'F845**	**HCAN0**
MC4[7]—Message Control 4[7]	**H'F846**	**HCAN0**
MC4[8]—Message Control 4[8]	**H'F847**	**HCAN0**

MC4[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC4[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC4[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC4[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC4[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC4[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC4[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC4[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC5[1]—Message Control 5[1]	**H'F848**	**HCAN0**
MC5[2]—Message Control 5[2]	**H'F849**	**HCAN0**
MC5[3]—Message Control 5[3]	**H'F84A**	**HCAN0**
MC5[4]—Message Control 5[4]	**H'F84B**	**HCAN0**
MC5[5]—Message Control 5[5]	**H'F84C**	**HCAN0**
MC5[6]—Message Control 5[6]	**H'F84D**	**HCAN0**
MC5[7]—Message Control 5[7]	**H'F84E**	**HCAN0**
MC5[8]—Message Control 5[8]	**H'F84F**	**HCAN0**

MC5[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC5[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC5[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC5[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC5[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC5[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC5[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC5[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC6[1]—Message Control 6[1]	**H'F850**	**HCAN0**
MC6[2]—Message Control 6[2]	**H'F851**	**HCAN0**
MC6[3]—Message Control 6[3]	**H'F852**	**HCAN0**
MC6[4]—Message Control 6[4]	**H'F853**	**HCAN0**
MC6[5]—Message Control 6[5]	**H'F854**	**HCAN0**
MC6[6]—Message Control 6[6]	**H'F855**	**HCAN0**
MC6[7]—Message Control 6[7]	**H'F856**	**HCAN0**
MC6[8]—Message Control 6[8]	**H'F857**	**HCAN0**

MC6[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC6[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC6[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC6[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC6[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC6[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC6[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC6[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC7[1]—Message Control 7[1]	**H'F858**	**HCAN0**
MC7[2]—Message Control 7[2]	**H'F859**	**HCAN0**
MC7[3]—Message Control 7[3]	**H'F85A**	**HCAN0**
MC7[4]—Message Control 7[4]	**H'F85B**	**HCAN0**
MC7[5]—Message Control 7[5]	**H'F85C**	**HCAN0**
MC7[6]—Message Control 7[6]	**H'F85D**	**HCAN0**
MC7[7]—Message Control 7[7]	**H'F85E**	**HCAN0**
MC7[8]—Message Control 7[8]	**H'F85F**	**HCAN0**

MC7[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC7[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC7[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC7[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC7[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC7[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC7[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC7[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC8[1]—Message Control 8[1]	**H'F860**	**HCAN0**
MC8[2]—Message Control 8[2]	**H'F861**	**HCAN0**
MC8[3]—Message Control 8[3]	**H'F862**	**HCAN0**
MC8[4]—Message Control 8[4]	**H'F863**	**HCAN0**
MC8[5]—Message Control 8[5]	**H'F864**	**HCAN0**
MC8[6]—Message Control 8[6]	**H'F865**	**HCAN0**
MC8[7]—Message Control 8[7]	**H'F866**	**HCAN0**
MC8[8]—Message Control 8[8]	**H'F867**	**HCAN0**

MC8[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC8[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC8[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC8[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC8[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC8[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC8[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC8[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC9[1]—Message Control 9[1]	**H'F868**	**HCAN0**
MC9[2]—Message Control 9[2]	**H'F869**	**HCAN0**
MC9[3]—Message Control 9[3]	**H'F86A**	**HCAN0**
MC9[4]—Message Control 9[4]	**H'F86B**	**HCAN0**
MC9[5]—Message Control 9[5]	**H'F86C**	**HCAN0**
MC9[6]—Message Control 9[6]	**H'F86D**	**HCAN0**
MC9[7]—Message Control 9[7]	**H'F86E**	**HCAN0**
MC9[8]—Message Control 9[8]	**H'F86F**	**HCAN0**

MC9[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC9[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC9[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC9[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC9[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC9[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC9[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC9[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC10[1]—Message Control 10[1]	**H'F870**	**HCAN0**
MC10[2]—Message Control 10[2]	**H'F871**	**HCAN0**
MC10[3]—Message Control 10[3]	**H'F872**	**HCAN0**
MC10[4]—Message Control 10[4]	**H'F873**	**HCAN0**
MC10[5]—Message Control 10[5]	**H'F874**	**HCAN0**
MC10[6]—Message Control 10[6]	**H'F875**	**HCAN0**
MC10[7]—Message Control 10[7]	**H'F876**	**HCAN0**
MC10[8]—Message Control 10[8]	**H'F877**	**HCAN0**

MC10[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC10[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC10[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC10[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC10[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC10[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC10[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC10[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC11[1]—Message Control 11[1]	**H'F878**	**HCAN0**
MC11[2]—Message Control 11[2]	**H'F879**	**HCAN0**
MC11[3]—Message Control 11[3]	**H'F87A**	**HCAN0**
MC11[4]—Message Control 11[4]	**H'F87B**	**HCAN0**
MC11[5]—Message Control 11[5]	**H'F87C**	**HCAN0**
MC11[6]—Message Control 11[6]	**H'F87D**	**HCAN0**
MC11[7]—Message Control 11[7]	**H'F87E**	**HCAN0**
MC11[8]—Message Control 11[8]	**H'F87F**	**HCAN0**

MC11[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC11[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC11[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC11[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC11[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC11[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC11[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC11[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC12[1]—Message Control 12[1]	**H'F880**	**HCAN0**
MC12[2]—Message Control 12[2]	**H'F881**	**HCAN0**
MC12[3]—Message Control 12[3]	**H'F882**	**HCAN0**
MC12[4]—Message Control 12[4]	**H'F883**	**HCAN0**
MC12[5]—Message Control 12[5]	**H'F884**	**HCAN0**
MC12[6]—Message Control 12[6]	**H'F885**	**HCAN0**
MC12[7]—Message Control 12[7]	**H'F886**	**HCAN0**
MC12[8]—Message Control 12[8]	**H'F887**	**HCAN0**

MC12[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC12[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC12[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC12[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC12[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC12[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC12[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC12[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC13[1]—Message Control 13[1]	**H'F888**	**HCAN0**
MC13[2]—Message Control 13[2]	**H'F889**	**HCAN0**
MC13[3]—Message Control 13[3]	**H'F88A**	**HCAN0**
MC13[4]—Message Control 13[4]	**H'F88B**	**HCAN0**
MC13[5]—Message Control 13[5]	**H'F88C**	**HCAN0**
MC13[6]—Message Control 13[6]	**H'F88D**	**HCAN0**
MC13[7]—Message Control 13[7]	**H'F88E**	**HCAN0**
MC13[8]—Message Control 13[8]	**H'F88F**	**HCAN0**

MC13[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC13[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC13[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC13[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC13[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC13[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC13[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC13[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC14[1]—Message Control 14[1]	**H'F890**	**HCAN0**
MC14[2]—Message Control 14[2]	**H'F891**	**HCAN0**
MC14[3]—Message Control 14[3]	**H'F892**	**HCAN0**
MC14[4]—Message Control 14[4]	**H'F893**	**HCAN0**
MC14[5]—Message Control 14[5]	**H'F894**	**HCAN0**
MC14[6]—Message Control 14[6]	**H'F895**	**HCAN0**
MC14[7]—Message Control 14[7]	**H'F896**	**HCAN0**
MC14[8]—Message Control 14[8]	**H'F897**	**HCAN0**

MC14[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC14[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC14[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC14[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC14[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC14[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC14[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC14[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC15[1]—Message Control 15[1]	**H'F898**	**HCAN0**
MC15[2]—Message Control 15[2]	**H'F899**	**HCAN0**
MC15[3]—Message Control 15[3]	**H'F89A**	**HCAN0**
MC15[4]—Message Control 15[4]	**H'F89B**	**HCAN0**
MC15[5]—Message Control 15[5]	**H'F89C**	**HCAN0**
MC15[6]—Message Control 15[6]	**H'F89D**	**HCAN0**
MC15[7]—Message Control 15[7]	**H'F89E**	**HCAN0**
MC15[8]—Message Control 15[8]	**H'F89F**	**HCAN0**

MC15[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC15[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC15[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC15[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC15[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC15[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC15[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC15[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MD0[1]—Message Data 0[1]	**H'F8B0**	**HCAN0**
MD0[2]—Message Data 0[2]	**H'F8B1**	**HCAN0**
MD0[3]—Message Data 0[3]	**H'F8B2**	**HCAN0**
MD0[4]—Message Data 0[4]	**H'F8B3**	**HCAN0**
MD0[5]—Message Data 0[5]	**H'F8B4**	**HCAN0**
MD0[6]—Message Data 0[6]	**H'F8B5**	**HCAN0**
MD0[7]—Message Data 0[7]	**H'F8B6**	**HCAN0**
MD0[8]—Message Data 0[8]	**H'F8B7**	**HCAN0**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD1[1]—Message Data 1[1]	**H'F8B8**	**HCAN0**
MD1[2]—Message Data 1[2]	**H'F8B9**	**HCAN0**
MD1[3]—Message Data 1[3]	**H'F8BA**	**HCAN0**
MD1[4]—Message Data 1[4]	**H'F8BB**	**HCAN0**
MD1[5]—Message Data 1[5]	**H'F8BC**	**HCAN0**
MD1[6]—Message Data 1[6]	**H'F8BD**	**HCAN0**
MD1[7]—Message Data 1[7]	**H'F8BE**	**HCAN0**
MD1[8]—Message Data 1[8]	**H'F8BF**	**HCAN0**

	Bit	7	6	5	4	3	2	1	0
MDx[1]									
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD2[1]—Message Data 2[1]	**H'F8C0**	**HCAN0**
MD2[2]—Message Data 2[2]	**H'F8C1**	**HCAN0**
MD2[3]—Message Data 2[3]	**H'F8C2**	**HCAN0**
MD2[4]—Message Data 2[4]	**H'F8C3**	**HCAN0**
MD2[5]—Message Data 2[5]	**H'F8C4**	**HCAN0**
MD2[6]—Message Data 2[6]	**H'F8C5**	**HCAN0**
MD2[7]—Message Data 2[7]	**H'F8C6**	**HCAN0**
MD2[8]—Message Data 2[8]	**H'F8C7**	**HCAN0**

	Bit	7	6	5	4	3	2	1	0
MDx[1]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD3[1]—Message Data 3[1]	**H'F8C8**	**HCAN0**
MD3[2]—Message Data 3[2]	**H'F8C9**	**HCAN0**
MD3[3]—Message Data 3[3]	**H'F8CA**	**HCAN0**
MD3[4]—Message Data 3[4]	**H'F8CB**	**HCAN0**
MD3[5]—Message Data 3[5]	**H'F8CC**	**HCAN0**
MD3[6]—Message Data 3[6]	**H'F8CD**	**HCAN0**
MD3[7]—Message Data 3[7]	**H'F8CE**	**HCAN0**
MD3[8]—Message Data 3[8]	**H'F8CF**	**HCAN0**

		7	6	5	4	3	2	1	0
MDx[1]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD4[1]—Message Data 4[1]	**H'F8D0**	**HCAN0**
MD4[2]—Message Data 4[2]	**H'F8D1**	**HCAN0**
MD4[3]—Message Data 4[3]	**H'F8D2**	**HCAN0**
MD4[4]—Message Data 4[4]	**H'F8D3**	**HCAN0**
MD4[5]—Message Data 4[5]	**H'F8D4**	**HCAN0**
MD4[6]—Message Data 4[6]	**H'F8D5**	**HCAN0**
MD4[7]—Message Data 4[7]	**H'F8D6**	**HCAN0**
MD4[8]—Message Data 4[8]	**H'F8D7**	**HCAN0**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD5[1]—Message Data 5[1]	**H'F8D8**	**HCAN0**
MD5[2]—Message Data 5[2]	**H'F8D9**	**HCAN0**
MD5[3]—Message Data 5[3]	**H'F8DA**	**HCAN0**
MD5[4]—Message Data 5[4]	**H'F8DB**	**HCAN0**
MD5[5]—Message Data 5[5]	**H'F8DC**	**HCAN0**
MD5[6]—Message Data 5[6]	**H'F8DD**	**HCAN0**
MD5[7]—Message Data 5[7]	**H'F8DE**	**HCAN0**
MD5[8]—Message Data 5[8]	**H'F8DF**	**HCAN0**

		7	6	5	4	3	2	1	0
MDx[1]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD6[1]—Message Data 6[1]	**H'F8E0**	**HCAN0**
MD6[2]—Message Data 6[2]	**H'F8E1**	**HCAN0**
MD6[3]—Message Data 6[3]	**H'F8E2**	**HCAN0**
MD6[4]—Message Data 6[4]	**H'F8E3**	**HCAN0**
MD6[5]—Message Data 6[5]	**H'F8E4**	**HCAN0**
MD6[6]—Message Data 6[6]	**H'F8E5**	**HCAN0**
MD6[7]—Message Data 6[7]	**H'F8E6**	**HCAN0**
MD6[8]—Message Data 6[8]	**H'F8E7**	**HCAN0**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD7[1]—Message Data 7[1]	**H'F8E8**	**HCAN0**
MD7[2]—Message Data 7[2]	**H'F8E9**	**HCAN0**
MD7[3]—Message Data 7[3]	**H'F8EA**	**HCAN0**
MD7[4]—Message Data 7[4]	**H'F8EB**	**HCAN0**
MD7[5]—Message Data 7[5]	**H'F8EC**	**HCAN0**
MD7[6]—Message Data 7[6]	**H'F8ED**	**HCAN0**
MD7[7]—Message Data 7[7]	**H'F8EE**	**HCAN0**
MD7[8]—Message Data 7[8]	**H'F8EF**	**HCAN0**

	Bit	7	6	5	4	3	2	1	0
MDx[1]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD8[1]—Message Data 8[1]	**H'F8F0**	**HCAN0**
MD8[2]—Message Data 8[2]	**H'F8F1**	**HCAN0**
MD8[3]—Message Data 8[3]	**H'F8F2**	**HCAN0**
MD8[4]—Message Data 8[4]	**H'F8F3**	**HCAN0**
MD8[5]—Message Data 8[5]	**H'F8F4**	**HCAN0**
MD8[6]—Message Data 8[6]	**H'F8F5**	**HCAN0**
MD8[7]—Message Data 8[7]	**H'F8F6**	**HCAN0**
MD8[8]—Message Data 8[8]	**H'F8F7**	**HCAN0**

		7	6	5	4	3	2	1	0
MDx[1]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

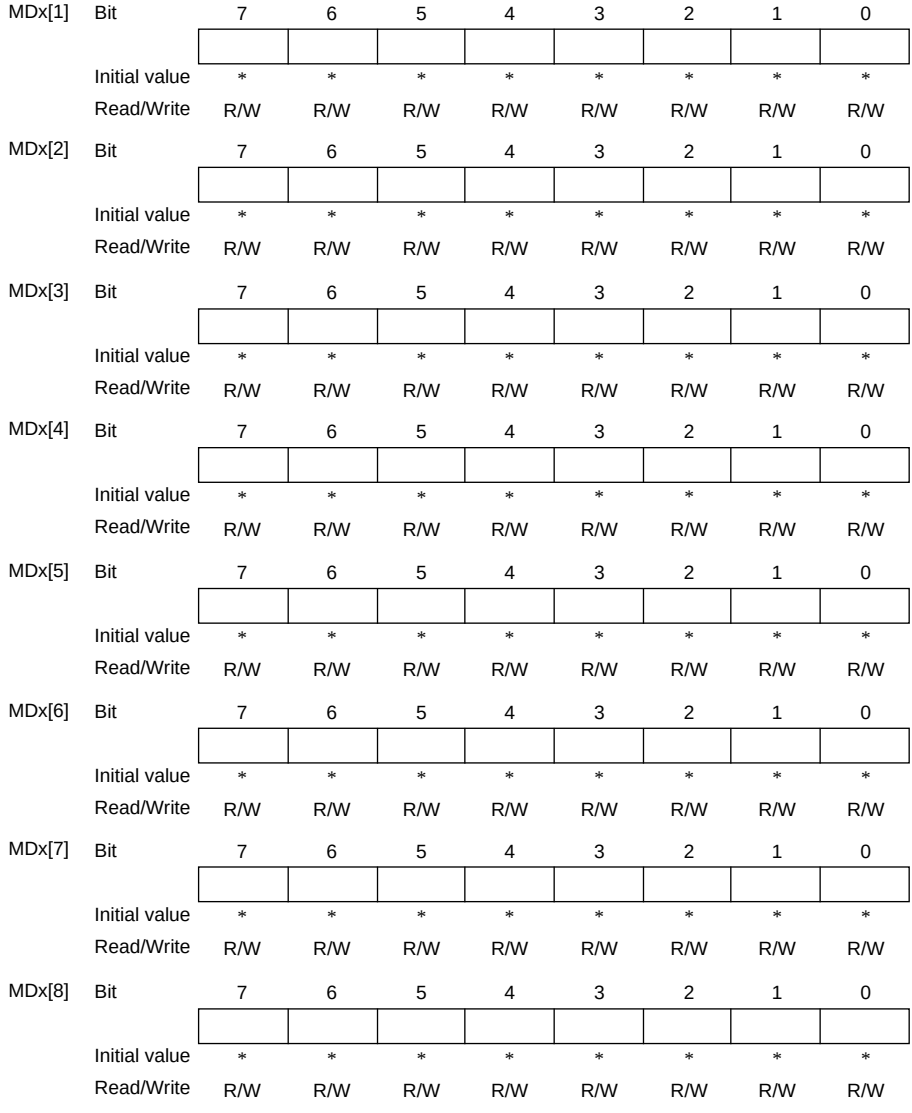
*: Undefined
x = 0 to 15

MD9[1]—Message Data 9[1]	**H'F8F8**	**HCAN0**
MD9[2]—Message Data 9[2]	**H'F8F9**	**HCAN0**
MD9[3]—Message Data 9[3]	**H'F8FA**	**HCAN0**
MD9[4]—Message Data 9[4]	**H'F8FB**	**HCAN0**
MD9[5]—Message Data 9[5]	**H'F8FC**	**HCAN0**
MD9[6]—Message Data 9[6]	**H'F8FD**	**HCAN0**
MD9[7]—Message Data 9[7]	**H'F8FE**	**HCAN0**
MD9[8]—Message Data 9[8]	**H'F8FF**	**HCAN0**

	Bit	7	6	5	4	3	2	1	0
MDx[1]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD10[1]—Message Data 10[1]	**H'F900**	**HCAN0**
MD10[2]—Message Data 10[2]	**H'F901**	**HCAN0**
MD10[3]—Message Data 10[3]	**H'F902**	**HCAN0**
MD10[4]—Message Data 10[4]	**H'F903**	**HCAN0**
MD10[5]—Message Data 10[5]	**H'F904**	**HCAN0**
MD10[6]—Message Data 10[6]	**H'F905**	**HCAN0**
MD10[7]—Message Data 10[7]	**H'F906**	**HCAN0**
MD10[8]—Message Data 10[8]	**H'F907**	**HCAN0**



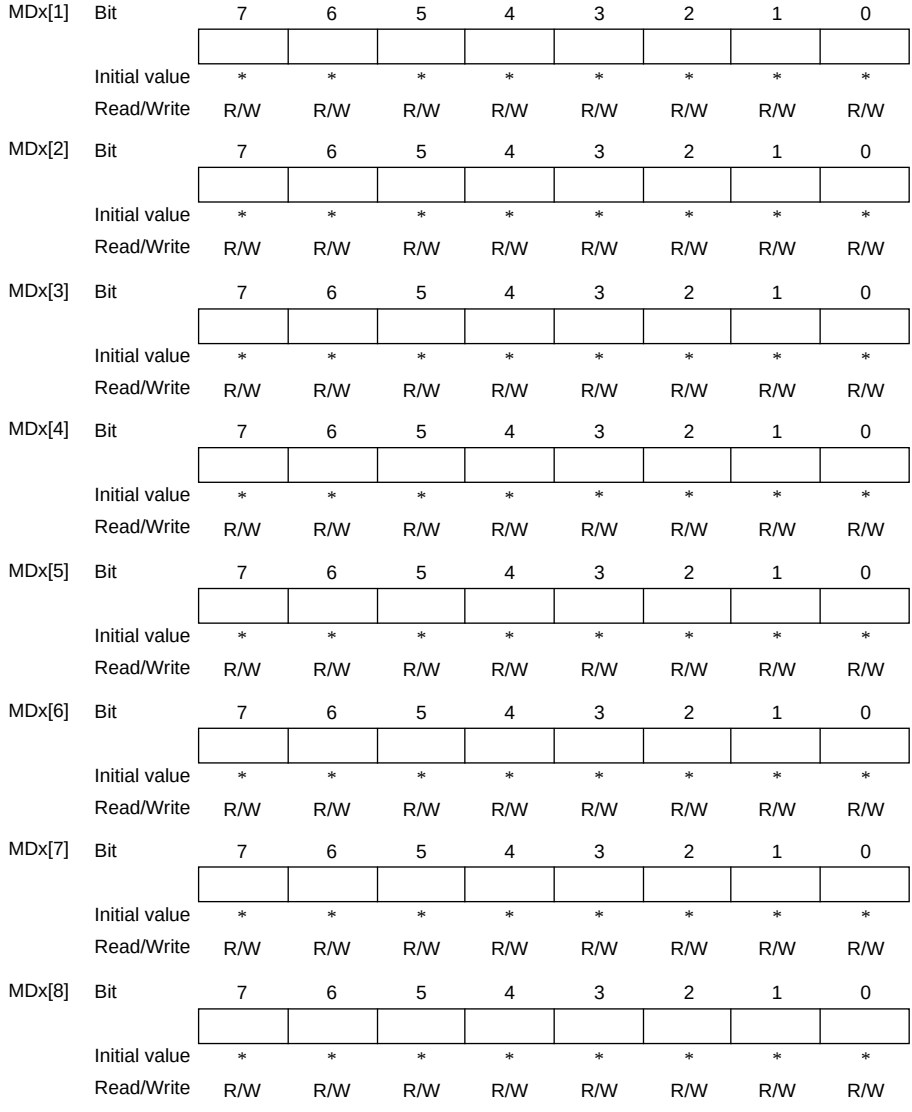
*: Undefined
x = 0 to 15

MD11[1]—Message Data 11[1]	**H'F908**	**HCAN0**
MD11[2]—Message Data 11[2]	**H'F909**	**HCAN0**
MD11[3]—Message Data 11[3]	**H'F90A**	**HCAN0**
MD11[4]—Message Data 11[4]	**H'F90B**	**HCAN0**
MD11[5]—Message Data 11[5]	**H'F90C**	**HCAN0**
MD11[6]—Message Data 11[6]	**H'F90D**	**HCAN0**
MD11[7]—Message Data 11[7]	**H'F90E**	**HCAN0**
MD11[8]—Message Data 11[8]	**H'F90F**	**HCAN0**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

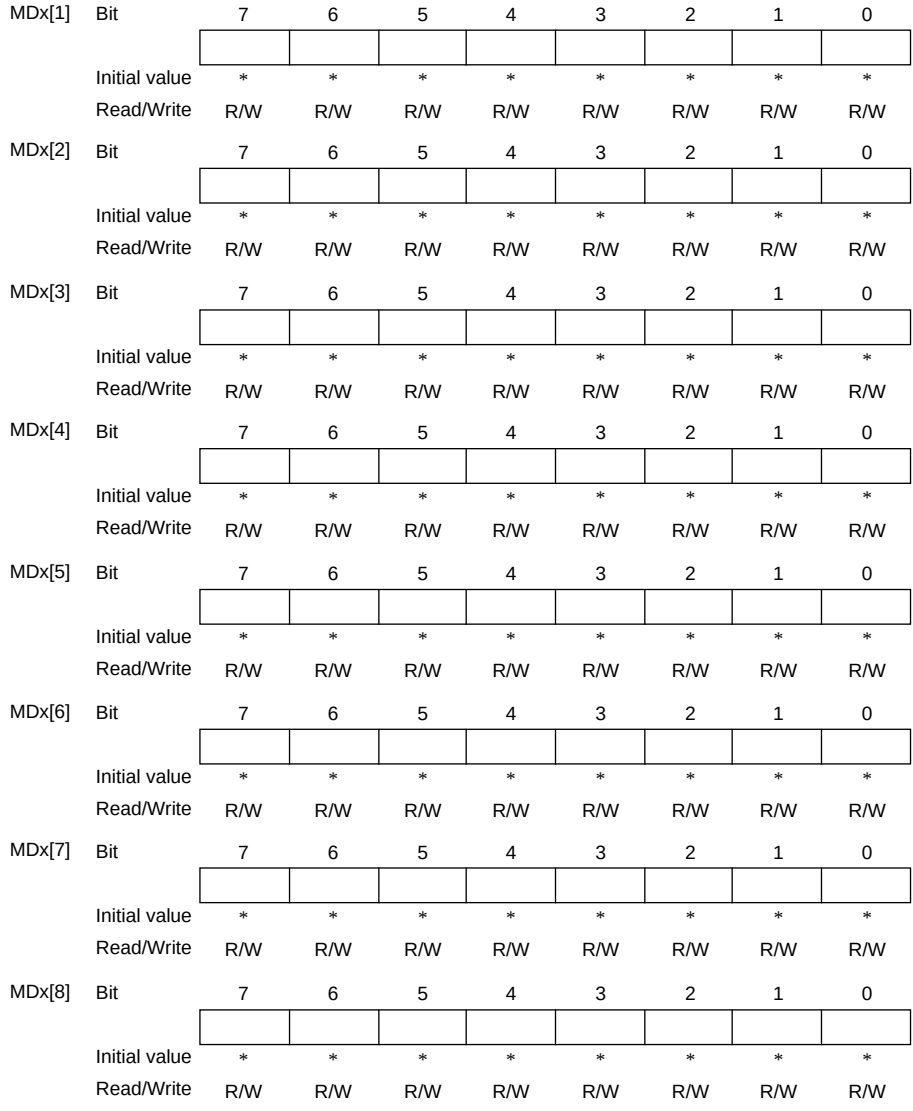
*: Undefined
x = 0 to 15

MD12[1]—Message Data 12[1]	**H'F910**	**HCAN0**
MD12[2]—Message Data 12[2]	**H'F911**	**HCAN0**
MD12[3]—Message Data 12[3]	**H'F912**	**HCAN0**
MD12[4]—Message Data 12[4]	**H'F913**	**HCAN0**
MD12[5]—Message Data 12[5]	**H'F914**	**HCAN0**
MD12[6]—Message Data 12[6]	**H'F915**	**HCAN0**
MD12[7]—Message Data 12[7]	**H'F916**	**HCAN0**
MD12[8]—Message Data 12[8]	**H'F917**	**HCAN0**



*: Undefined
x = 0 to 15

MD13[1]—Message Data 13[1]	**H'F918**	**HCAN0**
MD13[2]—Message Data 13[2]	**H'F919**	**HCAN0**
MD13[3]—Message Data 13[3]	**H'F91A**	**HCAN0**
MD13[4]—Message Data 13[4]	**H'F91B**	**HCAN0**
MD13[5]—Message Data 13[5]	**H'F91C**	**HCAN0**
MD13[6]—Message Data 13[6]	**H'F91D**	**HCAN0**
MD13[7]—Message Data 13[7]	**H'F91E**	**HCAN0**
MD13[8]—Message Data 13[8]	**H'F91F**	**HCAN0**



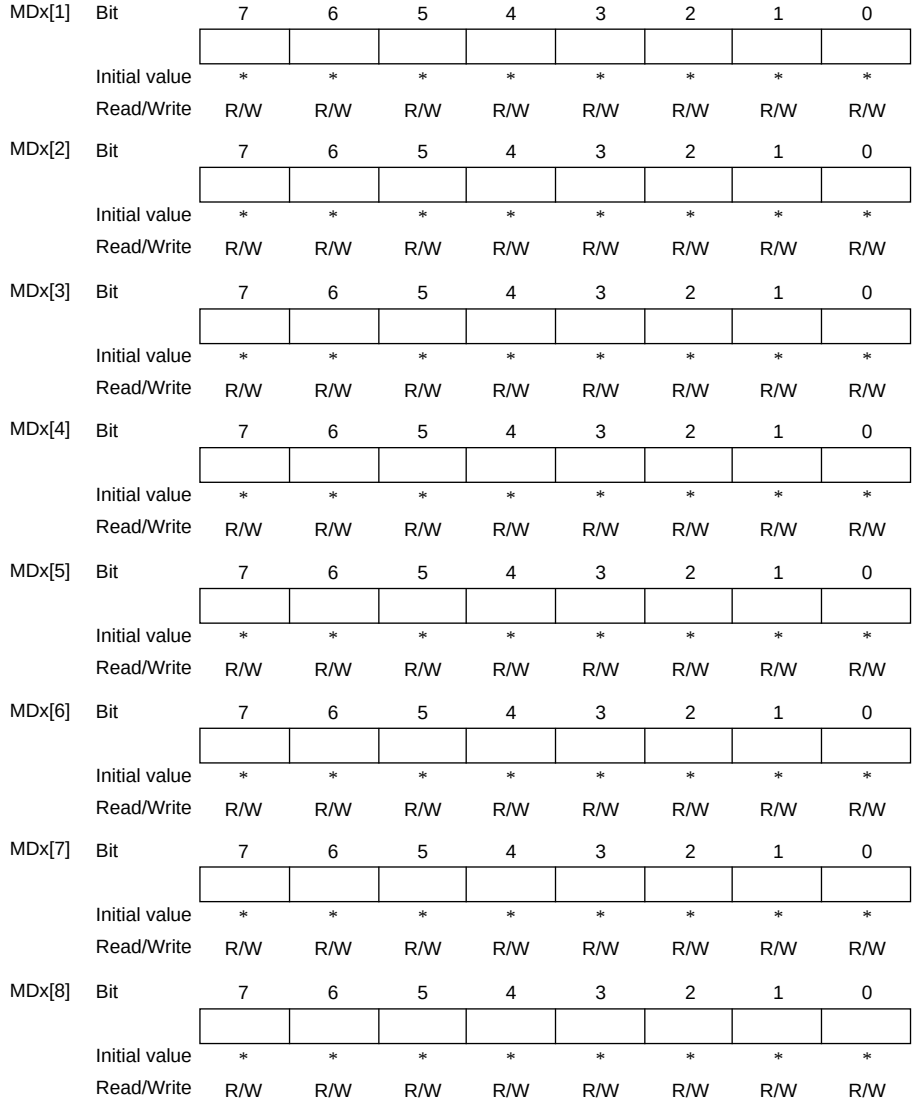
*: Undefined
x = 0 to 15

MD14[1]—Message Data 14[1]	**H'F920**	**HCAN0**
MD14[2]—Message Data 14[2]	**H'F921**	**HCAN0**
MD14[3]—Message Data 14[3]	**H'F922**	**HCAN0**
MD14[4]—Message Data 14[4]	**H'F923**	**HCAN0**
MD14[5]—Message Data 14[5]	**H'F924**	**HCAN0**
MD14[6]—Message Data 14[6]	**H'F925**	**HCAN0**
MD14[7]—Message Data 14[7]	**H'F926**	**HCAN0**
MD14[8]—Message Data 14[8]	**H'F927**	**HCAN0**

	Bit	7	6	5	4	3	2	1	0
MDx[1]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD15[1]—Message Data 15[1]	**H'F928**	**HCAN0**
MD15[2]—Message Data 15[2]	**H'F929**	**HCAN0**
MD15[3]—Message Data 15[3]	**H'F92A**	**HCAN0**
MD15[4]—Message Data 15[4]	**H'F92B**	**HCAN0**
MD15[5]—Message Data 15[5]	**H'F92C**	**HCAN0**
MD15[6]—Message Data 15[6]	**H'F92D**	**HCAN0**
MD15[7]—Message Data 15[7]	**H'F92E**	**HCAN0**
MD15[8]—Message Data 15[8]	**H'F92F**	**HCAN0**



*: Undefined
x = 0 to 15

MC0[1]—Message Control 0[1]	**H'FA20**	**HCAN1**
MC0[2]—Message Control 0[2]	**H'FA21**	**HCAN1**
MC0[3]—Message Control 0[3]	**H'FA22**	**HCAN1**
MC0[4]—Message Control 0[4]	**H'FA23**	**HCAN1**
MC0[5]—Message Control 0[5]	**H'FA24**	**HCAN1**
MC0[6]—Message Control 0[6]	**H'FA25**	**HCAN1**
MC0[7]—Message Control 0[7]	**H'FA26**	**HCAN1**
MC0[8]—Message Control 0[8]	**H'FA27**	**HCAN1**

MC0[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC0[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC0[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC0[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC0[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC0[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC0[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC0[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC1[1]—Message Control 1[1]	**H'FA28**	**HCAN1**
MC1[2]—Message Control 1[2]	**H'FA29**	**HCAN1**
MC1[3]—Message Control 1[3]	**H'FA2A**	**HCAN1**
MC1[4]—Message Control 1[4]	**H'FA2B**	**HCAN1**
MC1[5]—Message Control 1[5]	**H'FA2C**	**HCAN1**
MC1[6]—Message Control 1[6]	**H'FA2D**	**HCAN1**
MC1[7]—Message Control 1[7]	**H'FA2E**	**HCAN1**
MC1[8]—Message Control 1[8]	**H'FA2F**	**HCAN1**

MC1[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC1[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC1[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC1[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC1[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC1[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC1[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC1[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC2[1]—Message Control 2[1]	**H'FA30**	**HCAN1**
MC2[2]—Message Control 2[2]	**H'FA31**	**HCAN1**
MC2[3]—Message Control 2[3]	**H'FA32**	**HCAN1**
MC2[4]—Message Control 2[4]	**H'FA33**	**HCAN1**
MC2[5]—Message Control 2[5]	**H'FA34**	**HCAN1**
MC2[6]—Message Control 2[6]	**H'FA35**	**HCAN1**
MC2[7]—Message Control 2[7]	**H'FA36**	**HCAN1**
MC2[8]—Message Control 2[8]	**H'FA37**	**HCAN1**

MC2[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC2[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC2[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC2[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC2[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC2[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC2[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC2[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC3[1]—Message Control 3[1]	**H'FA38**	**HCAN1**
MC3[2]—Message Control 3[2]	**H'FA39**	**HCAN1**
MC3[3]—Message Control 3[3]	**H'FA3A**	**HCAN1**
MC3[4]—Message Control 3[4]	**H'FA3B**	**HCAN1**
MC3[5]—Message Control 3[5]	**H'FA3C**	**HCAN1**
MC3[6]—Message Control 3[6]	**H'FA3D**	**HCAN1**
MC3[7]—Message Control 3[7]	**H'FA3E**	**HCAN1**
MC3[8]—Message Control 3[8]	**H'FA3F**	**HCAN1**

MC3[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC3[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC3[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC3[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC3[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC3[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC3[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC3[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC4[1]—Message Control 4[1]	**H'FA40**	**HCAN1**
MC4[2]—Message Control 4[2]	**H'FA41**	**HCAN1**
MC4[3]—Message Control 4[3]	**H'FA42**	**HCAN1**
MC4[4]—Message Control 4[4]	**H'FA43**	**HCAN1**
MC4[5]—Message Control 4[5]	**H'FA44**	**HCAN1**
MC4[6]—Message Control 4[6]	**H'FA45**	**HCAN1**
MC4[7]—Message Control 4[7]	**H'FA46**	**HCAN1**
MC4[8]—Message Control 4[8]	**H'FA47**	**HCAN1**

MC4[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC4[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC4[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC4[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC4[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC4[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC4[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC4[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC5[1]—Message Control 5[1]	**H'FA48**	**HCAN1**
MC5[2]—Message Control 5[2]	**H'FA49**	**HCAN1**
MC5[3]—Message Control 5[3]	**H'FA4A**	**HCAN1**
MC5[4]—Message Control 5[4]	**H'FA4B**	**HCAN1**
MC5[5]—Message Control 5[5]	**H'FA4C**	**HCAN1**
MC5[6]—Message Control 5[6]	**H'FA4D**	**HCAN1**
MC5[7]—Message Control 5[7]	**H'FA4E**	**HCAN1**
MC5[8]—Message Control 5[8]	**H'FA4F**	**HCAN1**

MC5[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC5[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC5[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC5[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC5[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC5[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC5[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC5[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC6[1]—Message Control 6[1]	**H'FA50**	**HCAN1**
MC6[2]—Message Control 6[2]	**H'FA51**	**HCAN1**
MC6[3]—Message Control 6[3]	**H'FA52**	**HCAN1**
MC6[4]—Message Control 6[4]	**H'FA53**	**HCAN1**
MC6[5]—Message Control 6[5]	**H'FA54**	**HCAN1**
MC6[6]—Message Control 6[6]	**H'FA55**	**HCAN1**
MC6[7]—Message Control 6[7]	**H'FA56**	**HCAN1**
MC6[8]—Message Control 6[8]	**H'FA57**	**HCAN1**

MC6[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC6[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC6[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC6[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC6[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC6[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC6[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC6[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC7[1]—Message Control 7[1]	**H'FA58**	**HCAN1**
MC7[2]—Message Control 7[2]	**H'FA59**	**HCAN1**
MC7[3]—Message Control 7[3]	**H'FA5A**	**HCAN1**
MC7[4]—Message Control 7[4]	**H'FA5B**	**HCAN1**
MC7[5]—Message Control 7[5]	**H'FA5C**	**HCAN1**
MC7[6]—Message Control 7[6]	**H'FA5D**	**HCAN1**
MC7[7]—Message Control 7[7]	**H'FA5E**	**HCAN1**
MC7[8]—Message Control 7[8]	**H'FA5F**	**HCAN1**

MC7[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC7[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC7[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC7[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC7[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC7[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC7[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC7[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC8[1]—Message Control 8[1]	**H'FA60**	**HCAN1**
MC8[2]—Message Control 8[2]	**H'FA61**	**HCAN1**
MC8[3]—Message Control 8[3]	**H'FA62**	**HCAN1**
MC8[4]—Message Control 8[4]	**H'FA63**	**HCAN1**
MC8[5]—Message Control 8[5]	**H'FA64**	**HCAN1**
MC8[6]—Message Control 8[6]	**H'FA65**	**HCAN1**
MC8[7]—Message Control 8[7]	**H'FA66**	**HCAN1**
MC8[8]—Message Control 8[8]	**H'FA67**	**HCAN1**

MC8[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC8[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC8[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC8[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC8[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC8[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC8[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC8[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC9[1]—Message Control 9[1]	**H'FA68**	**HCAN1**
MC9[2]—Message Control 9[2]	**H'FA69**	**HCAN1**
MC9[3]—Message Control 9[3]	**H'FA6A**	**HCAN1**
MC9[4]—Message Control 9[4]	**H'FA6B**	**HCAN1**
MC9[5]—Message Control 9[5]	**H'FA6C**	**HCAN1**
MC9[6]—Message Control 9[6]	**H'FA6D**	**HCAN1**
MC9[7]—Message Control 9[7]	**H'FA6E**	**HCAN1**
MC9[8]—Message Control 9[8]	**H'FA6F**	**HCAN1**

MC9[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC9[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC9[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC9[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC9[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC9[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC9[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC9[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC10[1]—Message Control 10[1]	**H'FA70**	**HCAN1**
MC10[2]—Message Control 10[2]	**H'FA71**	**HCAN1**
MC10[3]—Message Control 10[3]	**H'FA72**	**HCAN1**
MC10[4]—Message Control 10[4]	**H'FA73**	**HCAN1**
MC10[5]—Message Control 10[5]	**H'FA74**	**HCAN1**
MC10[6]—Message Control 10[6]	**H'FA75**	**HCAN1**
MC10[7]—Message Control 10[7]	**H'FA76**	**HCAN1**
MC10[8]—Message Control 10[8]	**H'FA77**	**HCAN1**

MC10[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC10[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC10[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC10[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC10[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC10[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC10[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC10[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC11[1]—Message Control 11[1]	**H'FA78**	**HCAN1**
MC11[2]—Message Control 11[2]	**H'FA79**	**HCAN1**
MC11[3]—Message Control 11[3]	**H'FA7A**	**HCAN1**
MC11[4]—Message Control 11[4]	**H'FA7B**	**HCAN1**
MC11[5]—Message Control 11[5]	**H'FA7C**	**HCAN1**
MC11[6]—Message Control 11[6]	**H'FA7D**	**HCAN1**
MC11[7]—Message Control 11[7]	**H'FA7E**	**HCAN1**
MC11[8]—Message Control 11[8]	**H'FA7F**	**HCAN1**

MC11[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC11[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC11[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC11[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC11[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC11[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC11[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC11[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC12[1]—Message Control 12[1]	**H'FA80**	**HCAN1**
MC12[2]—Message Control 12[2]	**H'FA81**	**HCAN1**
MC12[3]—Message Control 12[3]	**H'FA82**	**HCAN1**
MC12[4]—Message Control 12[4]	**H'FA83**	**HCAN1**
MC12[5]—Message Control 12[5]	**H'FA84**	**HCAN1**
MC12[6]—Message Control 12[6]	**H'FA85**	**HCAN1**
MC12[7]—Message Control 12[7]	**H'FA86**	**HCAN1**
MC12[8]—Message Control 12[8]	**H'FA87**	**HCAN1**

MC12[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC12[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC12[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC12[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC12[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC12[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC12[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC12[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC13[1]—Message Control 13[1]	**H'FA88**	**HCAN1**
MC13[2]—Message Control 13[2]	**H'FA89**	**HCAN1**
MC13[3]—Message Control 13[3]	**H'FA8A**	**HCAN1**
MC13[4]—Message Control 13[4]	**H'FA8B**	**HCAN1**
MC13[5]—Message Control 13[5]	**H'FA8C**	**HCAN1**
MC13[6]—Message Control 13[6]	**H'FA8D**	**HCAN1**
MC13[7]—Message Control 13[7]	**H'FA8E**	**HCAN1**
MC13[8]—Message Control 13[8]	**H'FA8F**	**HCAN1**

MC13[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC13[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC13[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC13[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC13[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC13[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC13[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC13[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC14[1]—Message Control 14[1]	**H'FA90**	**HCAN1**
MC14[2]—Message Control 14[2]	**H'FA91**	**HCAN1**
MC14[3]—Message Control 14[3]	**H'FA92**	**HCAN1**
MC14[4]—Message Control 14[4]	**H'FA93**	**HCAN1**
MC14[5]—Message Control 14[5]	**H'FA94**	**HCAN1**
MC14[6]—Message Control 14[6]	**H'FA95**	**HCAN1**
MC14[7]—Message Control 14[7]	**H'FA96**	**HCAN1**
MC14[8]—Message Control 14[8]	**H'FA97**	**HCAN1**

MC14[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC14[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC14[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC14[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC14[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC14[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC14[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC14[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC15[1]—Message Control 15[1]	**H'FA98**	**HCAN1**
MC15[2]—Message Control 15[2]	**H'FA99**	**HCAN1**
MC15[3]—Message Control 15[3]	**H'FA9A**	**HCAN1**
MC15[4]—Message Control 15[4]	**H'FA9B**	**HCAN1**
MC15[5]—Message Control 15[5]	**H'FA9C**	**HCAN1**
MC15[6]—Message Control 15[6]	**H'FA9D**	**HCAN1**
MC15[7]—Message Control 15[7]	**H'FA9E**	**HCAN1**
MC15[8]—Message Control 15[8]	**H'FA9F**	**HCAN1**

MC15[1]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	DLC3	DLC2	DLC1	DLC0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Data Length Code

0	0	0	0	Data length = 0 byte
			1	Data length = 1 byte
		1	0	Data length = 2 bytes
			1	Data length = 3 bytes
	1	0	0	Data length = 4 bytes
			1	Data length = 5 bytes
		1	0	Data length = 6 bytes
			1	Data length = 7 bytes
1	0/1	0/1	0/1	Data length = 8 bytes

MC15[2]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC15[3]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC15[4]

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MC15[5]

Bit	7	6	5	4	3	2	1	0
	STD_ID2	STD_ID1	STD_ID0	RTR	IDE	—	EXD_ID17	EXD_ID16
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier
Set the identifier (extended identifier) of data frames and remote frames

Identifier Extension

0	Standard format
1	Extended format

Remote Transmission Request

0	Data frame
1	Remote frame

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC15[6]

Bit	7	6	5	4	3	2	1	0
	STD_ID10	STD_ID9	STD_ID8	STD_ID7	STD_ID6	STD_ID5	STD_ID4	STD_ID3
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Standard Identifier
Set the identifier (standard identifier) of data frames and remote frames

MC15[7]

Bit	7	6	5	4	3	2	1	0
	EXD_ID7	EXD_ID6	EXD_ID5	EXD_ID4	EXD_ID3	EXD_ID2	EXD_ID1	EXD_ID0
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MC15[8]

Bit	7	6	5	4	3	2	1	0
	EXD_ID15	EXD_ID14	EXD_ID13	EXD_ID12	EXD_ID11	EXD_ID10	EXD_ID9	EXD_ID8
Initial value	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Extended Identifier

Set the identifier (extended identifier) of data frames and remote frames

MD0[1]—Message Data 0[1]	**H'FAB0**	**HCAN1**
MD0[2]—Message Data 0[2]	**H'FAB1**	**HCAN1**
MD0[3]—Message Data 0[3]	**H'FAB2**	**HCAN1**
MD0[4]—Message Data 0[4]	**H'FAB3**	**HCAN1**
MD0[5]—Message Data 0[5]	**H'FAB4**	**HCAN1**
MD0[6]—Message Data 0[6]	**H'FAB5**	**HCAN1**
MD0[7]—Message Data 0[7]	**H'FAB6**	**HCAN1**
MD0[8]—Message Data 0[8]	**H'FAB7**	**HCAN1**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD1[1]—Message Data 1[1]	**H'FAB8**	**HCAN1**
MD1[2]—Message Data 1[2]	**H'FAB9**	**HCAN1**
MD1[3]—Message Data 1[3]	**H'FABA**	**HCAN1**
MD1[4]—Message Data 1[4]	**H'FABB**	**HCAN1**
MD1[5]—Message Data 1[5]	**H'FABC**	**HCAN1**
MD1[6]—Message Data 1[6]	**H'FABD**	**HCAN1**
MD1[7]—Message Data 1[7]	**H'FABE**	**HCAN1**
MD1[8]—Message Data 1[8]	**H'FABF**	**HCAN1**

	Bit	7	6	5	4	3	2	1	0
MDx[1]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD2[1]—Message Data 2[1]	**H'FAC0**	**HCAN1**
MD2[2]—Message Data 2[2]	**H'FAC1**	**HCAN1**
MD2[3]—Message Data 2[3]	**H'FAC2**	**HCAN1**
MD2[4]—Message Data 2[4]	**H'FAC3**	**HCAN1**
MD2[5]—Message Data 2[5]	**H'FAC4**	**HCAN1**
MD2[6]—Message Data 2[6]	**H'FAC5**	**HCAN1**
MD2[7]—Message Data 2[7]	**H'FAC6**	**HCAN1**
MD2[8]—Message Data 2[8]	**H'FAC7**	**HCAN1**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD3[1]—Message Data 3[1]	**H'FAC8**	**HCAN1**
MD3[2]—Message Data 3[2]	**H'FAC9**	**HCAN1**
MD3[3]—Message Data 3[3]	**H'FACA**	**HCAN1**
MD3[4]—Message Data 3[4]	**H'FACB**	**HCAN1**
MD3[5]—Message Data 3[5]	**H'FACC**	**HCAN1**
MD3[6]—Message Data 3[6]	**H'FACD**	**HCAN1**
MD3[7]—Message Data 3[7]	**H'FACE**	**HCAN1**
MD3[8]—Message Data 3[8]	**H'FACF**	**HCAN1**

		7	6	5	4	3	2	1	0
MDx[1]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD4[1]—Message Data 4[1]	**H'FAD0**	**HCAN1**
MD4[2]—Message Data 4[2]	**H'FAD1**	**HCAN1**
MD4[3]—Message Data 4[3]	**H'FAD2**	**HCAN1**
MD4[4]—Message Data 4[4]	**H'FAD3**	**HCAN1**
MD4[5]—Message Data 4[5]	**H'FAD4**	**HCAN1**
MD4[6]—Message Data 4[6]	**H'FAD5**	**HCAN1**
MD4[7]—Message Data 4[7]	**H'FAD6**	**HCAN1**
MD4[8]—Message Data 4[8]	**H'FAD7**	**HCAN1**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD5[1]—Message Data 5[1]	**H'FAD8**	**HCAN1**
MD5[2]—Message Data 5[2]	**H'FAD9**	**HCAN1**
MD5[3]—Message Data 5[3]	**H'FADA**	**HCAN1**
MD5[4]—Message Data 5[4]	**H'FADB**	**HCAN1**
MD5[5]—Message Data 5[5]	**H'FADC**	**HCAN1**
MD5[6]—Message Data 5[6]	**H'FADD**	**HCAN1**
MD5[7]—Message Data 5[7]	**H'FADE**	**HCAN1**
MD5[8]—Message Data 5[8]	**H'FADF**	**HCAN1**

		7	6	5	4	3	2	1	0
MDx[1]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD6[1]—Message Data 6[1]	**H'FAE0**	**HCAN1**
MD6[2]—Message Data 6[2]	**H'FAE1**	**HCAN1**
MD6[3]—Message Data 6[3]	**H'FAE2**	**HCAN1**
MD6[4]—Message Data 6[4]	**H'FAE3**	**HCAN1**
MD6[5]—Message Data 6[5]	**H'FAE4**	**HCAN1**
MD6[6]—Message Data 6[6]	**H'FAE5**	**HCAN1**
MD6[7]—Message Data 6[7]	**H'FAE6**	**HCAN1**
MD6[8]—Message Data 6[8]	**H'FAE7**	**HCAN1**

	Bit	7	6	5	4	3	2	1	0
MDx[1]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD7[1]—Message Data 7[1]	**H'FAE8**	**HCAN1**
MD7[2]—Message Data 7[2]	**H'FAE9**	**HCAN1**
MD7[3]—Message Data 7[3]	**H'FAEA**	**HCAN1**
MD7[4]—Message Data 7[4]	**H'FAEB**	**HCAN1**
MD7[5]—Message Data 7[5]	**H'FAEC**	**HCAN1**
MD7[6]—Message Data 7[6]	**H'FAED**	**HCAN1**
MD7[7]—Message Data 7[7]	**H'FAEE**	**HCAN1**
MD7[8]—Message Data 7[8]	**H'FAEF**	**HCAN1**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD8[1]—Message Data 8[1]	**H'FAF0**	**HCAN1**
MD8[2]—Message Data 8[2]	**H'FAF1**	**HCAN1**
MD8[3]—Message Data 8[3]	**H'FAF2**	**HCAN1**
MD8[4]—Message Data 8[4]	**H'FAF3**	**HCAN1**
MD8[5]—Message Data 8[5]	**H'FAF4**	**HCAN1**
MD8[6]—Message Data 8[6]	**H'FAF5**	**HCAN1**
MD8[7]—Message Data 8[7]	**H'FAF6**	**HCAN1**
MD8[8]—Message Data 8[8]	**H'FAF7**	**HCAN1**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD9[1]—Message Data 9[1]	**H'FAF8**	**HCAN1**
MD9[2]—Message Data 9[2]	**H'FAF9**	**HCAN1**
MD9[3]—Message Data 9[3]	**H'FAFA**	**HCAN1**
MD9[4]—Message Data 9[4]	**H'FAFB**	**HCAN1**
MD9[5]—Message Data 9[5]	**H'FAFC**	**HCAN1**
MD9[6]—Message Data 9[6]	**H'FAFD**	**HCAN1**
MD9[7]—Message Data 9[7]	**H'FAFE**	**HCAN1**
MD9[8]—Message Data 9[8]	**H'FAFF**	**HCAN1**

		7	6	5	4	3	2	1	0
MDx[1]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD10[1]—Message Data 10[1]	**H'FB00**	**HCAN1**
MD10[2]—Message Data 10[2]	**H'FB01**	**HCAN1**
MD10[3]—Message Data 10[3]	**H'FB02**	**HCAN1**
MD10[4]—Message Data 10[4]	**H'FB03**	**HCAN1**
MD10[5]—Message Data 10[5]	**H'FB04**	**HCAN1**
MD10[6]—Message Data 10[6]	**H'FB05**	**HCAN1**
MD10[7]—Message Data 10[7]	**H'FB06**	**HCAN1**
MD10[8]—Message Data 10[8]	**H'FB07**	**HCAN1**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD11[1]—Message Data 11[1]	**H'FB08**	**HCAN1**
MD11[2]—Message Data 11[2]	**H'FB09**	**HCAN1**
MD11[3]—Message Data 11[3]	**H'FB0A**	**HCAN1**
MD11[4]—Message Data 11[4]	**H'FB0B**	**HCAN1**
MD11[5]—Message Data 11[5]	**H'FB0C**	**HCAN1**
MD11[6]—Message Data 11[6]	**H'FB0D**	**HCAN1**
MD11[7]—Message Data 11[7]	**H'FB0E**	**HCAN1**
MD11[8]—Message Data 11[8]	**H'FB0F**	**HCAN1**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD12[1]—Message Data 12[1]	**H'FB10**	**HCAN1**
MD12[2]—Message Data 12[2]	**H'FB11**	**HCAN1**
MD12[3]—Message Data 12[3]	**H'FB12**	**HCAN1**
MD12[4]—Message Data 12[4]	**H'FB13**	**HCAN1**
MD12[5]—Message Data 12[5]	**H'FB14**	**HCAN1**
MD12[6]—Message Data 12[6]	**H'FB15**	**HCAN1**
MD12[7]—Message Data 12[7]	**H'FB16**	**HCAN1**
MD12[8]—Message Data 12[8]	**H'FB17**	**HCAN1**

	Bit	7	6	5	4	3	2	1	0
MDx[1]	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD13[1]—Message Data 13[1]	**H'FB18**	**HCAN1**
MD13[2]—Message Data 13[2]	**H'FB19**	**HCAN1**
MD13[3]—Message Data 13[3]	**H'FB1A**	**HCAN1**
MD13[4]—Message Data 13[4]	**H'FB1B**	**HCAN1**
MD13[5]—Message Data 13[5]	**H'FB1C**	**HCAN1**
MD13[6]—Message Data 13[6]	**H'FB1D**	**HCAN1**
MD13[7]—Message Data 13[7]	**H'FB1E**	**HCAN1**
MD13[8]—Message Data 13[8]	**H'FB1F**	**HCAN1**

		7	6	5	4	3	2	1	0
MDx[1]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD14[1]—Message Data 14[1]	**H'FB20**	**HCAN1**
MD14[2]—Message Data 14[2]	**H'FB21**	**HCAN1**
MD14[3]—Message Data 14[3]	**H'FB22**	**HCAN1**
MD14[4]—Message Data 14[4]	**H'FB23**	**HCAN1**
MD14[5]—Message Data 14[5]	**H'FB24**	**HCAN1**
MD14[6]—Message Data 14[6]	**H'FB25**	**HCAN1**
MD14[7]—Message Data 14[7]	**H'FB26**	**HCAN1**
MD14[8]—Message Data 14[8]	**H'FB27**	**HCAN1**

MDx[1]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[2]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[3]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[4]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[5]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[6]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[7]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

MDx[8]	Bit	7	6	5	4	3	2	1	0
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

MD15[1]—Message Data 15[1]	**H'FB28**	**HCAN1**
MD15[2]—Message Data 15[2]	**H'FB29**	**HCAN1**
MD15[3]—Message Data 15[3]	**H'FB2A**	**HCAN1**
MD15[4]—Message Data 15[4]	**H'FB2B**	**HCAN1**
MD15[5]—Message Data 15[5]	**H'FB2C**	**HCAN1**
MD15[6]—Message Data 15[6]	**H'FB2D**	**HCAN1**
MD15[7]—Message Data 15[7]	**H'FB2E**	**HCAN1**
MD15[8]—Message Data 15[8]	**H'FB2F**	**HCAN1**

		7	6	5	4	3	2	1	0
MDx[1]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[2]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[3]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[4]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[5]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[6]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[7]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MDx[8]	Bit								
	Initial value	*	*	*	*	*	*	*	*
	Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

*: Undefined
x = 0 to 15

Bit	7	6	5	4	3	2	1	0
	—	—	IE	CMF	CST	CKS2	CKS1	CKS0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/(W)*	R/W	R/W	R/W	R/W

Clock Select

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/2
	1	0	Internal clock: counts on ø/4
		1	Internal clock: counts on ø/8
1	*	*	Internal clock: counts on ø/16

*: Don't care

Counter Start

0	PWCNT is stopped
1	PWCNT is started

Compare Match Flag

0	[Clearing conditions] • When 0 is written to CMF after reading CMF = 1 • When the DTC is activated by a compare match interrupt, and the DISEL bit in the DTC's MRB register is 0
1	[Setting condition] When PWCNT = PWCYR

Interrupt Enable

0	Interrupt disabled
1	Interrupt enabled

Note: * Only 0 can be written, to clear the flag.

PWOCR1—PWM Output Control Register 1 **H'FC02** **PWM1**

Bit	7	6	5	4	3	2	1	0
	OE1H	OE1G	OE1F	OE1E	OE1D	OE1C	OE1B	OE1A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output Enable

0	PWM output is disabled
1	PWM output is enabled

PWPR1—PWM Polarity Register 1 **H'FC04** **PWM1**

Bit	7	6	5	4	3	2	1	0
	OPS1H	OPS1G	OPS1F	OPS1E	OPS1D	OPS1C	OPS1B	OPS1A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output Polarity Select

0	PWM direct output
1	PWM inverse output

PWCYR1—PWM Cycle Register 1 **H'FC06** **PWM1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—										
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Set the PWM conversion cycle

PWBFR1A—PWM Buffer Register 1A **H'FC08** **PWM1**
PWBFR1C—PWM Buffer Register 1C **H'FC0A** **PWM1**
PWBFR1E—PWM Buffer Register 1E **H'FC0C** **PWM1**
PWBFR1G—PWM Buffer Register 1G **H'FC0E** **PWM1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	OTS	—	—	DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
Initial value	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
Read/Write	—	—	—	R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output Terminal Select
Bit 12 is the data transferred to bit 12 of PWDTR1

Duty
Bits 9 to 0 comprise the data transferred to bits 9 to 0 in PWDTR1

Register	OTS	Description
PWDTR1A	0	PWM1A output selected
	1	PWM1B output selected
PWDTR1C	0	PWM1C output selected
	1	PWM1D output selected
PWDTR1E	0	PWM1E output selected
	1	PWM1F output selected
PWDTR1G	0	PWM1G output selected
	1	PWM1H output selected

Note: When a PWCYR1 compare match occurs, data is transferred from PWBFR1A to PWDTR1A, from PWBFR1C to PWDTR1C, from PWBFR1E to PWDTR1E, and from PWBFR1G to PWDTR1G.

Bit	7	6	5	4	3	2	1	0
	—	—	IE	CMF	CST	CKS2	CKS1	CKS0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/(W)*	R/W	R/W	R/W	R/W

Clock Select

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/2
	1	0	Internal clock: counts on ø/4
		1	Internal clock: counts on ø/8
1	*	*	Internal clock: counts on ø/16

*: Don't care

Counter Start

0	PWCNT is stopped
1	PWCNT is started

Compare Match Flag

0	[Clearing conditions] • When 0 is written to CMF after reading CMF = 1 • When the DTC is activated by a compare match interrupt, and the DISEL bit in the DTC's MRB register is 0
1	[Setting condition] When PWCNT = PWCYR

Interrupt Enable

0	Interrupt disabled
1	Interrupt enabled

Note: * Only 0 can be written, to clear the flag.

PWOCR2—PWM Output Control Register 2 **H'FC12** **PWM2**

Bit	7	6	5	4	3	2	1	0
	OE2H	OE2G	OE2F	OE2E	OE2D	OE2C	OE2B	OE2A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output Enable

0	PWM output is disabled
1	PWM output is enabled

PWPR2—PWM Polarity Register 2 **H'FC14** **PWM2**

Bit	7	6	5	4	3	2	1	0
	OPS2H	OPS2G	OPS2F	OPS2E	OPS2D	OPS2C	OPS2B	OPS2A
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Output Polarity Select

0	PWM direct output
1	PWM inverse output

PWCYR2—PWM Cycle Register 2 **H'FC16** **PWM2**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	—	—	—										
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Set the PWM conversion cycle

PWBFR2A—PWM Buffer Register 2A **H'FC18** **PWM2**
PWBFR2B—PWM Buffer Register 2B **H'FC1A** **PWM2**
PWBFR2C—PWM Buffer Register 2C **H'FC1C** **PWM2**
PWBFR2D—PWM Buffer Register 2D **H'FC1E** **PWM2**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	—	—	—	TDS	—	—	DT9	DT8	DT7	DT6	DT5	DT4	DT3	DT2	DT1	DT0
Initial value	1	1	1	0	1	1	0	0	0	0	0	0	0	0	0	0
Read/Write	—	—	—	R/W	—	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Duty
Bits 9 to 0 compromise the data transferred to bits 9 to 0 in PWDTR2

Transfer Destination Select
Selects the PWDTR2 register to which data is to be transferred

Register	TDS	Description
PWBFR2A	0	PWDTR2A selected
	1	PWDTR2E selected
PWBFR2B	0	PWDTR2B selected
	1	PWDTR2F selected
PWBFR2C	0	PWDTR2C selected
	1	PWDTR2G selected
PWBFR2D	0	PWDTR2D selected
	1	PWDTR2H selected

Note: When a PWCYR2 compare match occurs, data is transferred from PWBFR2A to PWDTR2A or PWDTR2E, from PWBFR2B to PWDTR2B or PWDTR2F, from PWBFR2C to PWDTR2C or PWDTR2G, and from PWBFR2D to PWDTR2D or PWDTR2H.

PHDDR—Port H Data Direction Register **H'FC20** **Port**

Bit	7	6	5	4	3	2	1	0
	PH7DDR	PH6DDR	PH5DDR	PH4DDR	PH3DDR	PH2DDR	PH1DDR	PH0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PJDDR—Port J Data Direction Register **H'FC21** **Port**

Bit	7	6	5	4	3	2	1	0
	PJ7DDR	PJ6DDR	PJ5DDR	PJ4DDR	PJ3DDR	PJ2DDR	PJ1DDR	PJ0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

PHDR—Port H Data Register **H'FC24** **Port**

Bit	7	6	5	4	3	2	1	0
	PH7DR	PH6DR	PH5DR	PH4DR	PH3DR	PH2DR	PH1DR	PH0DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PJDR—Port J Data Register **H'FC25** **Port**

Bit	7	6	5	4	3	2	1	0
	PJ7DR	PJ6DR	PJ5DR	PJ4DR	PJ3DR	PJ2DR	PJ1DR	PJ0DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PORTH—Port H Register **H'FC28** **Port**

Bit	7	6	5	4	3	2	1	0
	PH7	PH6	PH5	PH4	PH3	PH2	PH1	PH0
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Note: * Determined by the state of PH7 to PH0.

PORTJ—Port J Register **H'FC29** **Port**

Bit	7	6	5	4	3	2	1	0
	PJ7	PJ6	PJ5	PJ4	PJ3	PJ2	PJ1	PJ0
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

Note: * Determined by the state of PJ7 to PJ0.

MSTPCRD—Module Stop Control Register D **H'FC60** **System**

Bit	7	6	5	4	3	2	1	0
	MSTPD7	MSTPD6	MSTPD5	MSTPD4	MSTPD3	MSTPD2	MSTPD1	MSTPD0
Initial value	1	1	Undefined	Undefined	Undefined	Undefined	Undefined	Undefined
Read/Write	R/W	R/W	—	—	—	—	—	—

Module Stop

0	PWM module stop mode is cleared
1	PWM module stop mode is set

SCRX—Serial Control Register X **H'FDB4** **IIC**

Bit :	7	6	5	4	3	2	1	0
	—	IICX1	IICX0	IICE	—	—	—	—
Initial value :	0	0	0	0	1	0	0	0
R/W :	R/W	R/W	R/W	R/W	—	R/W	R/W	R/W

I^2C master enable

0	Disables CPU access of I^2C bus interface data register and control register.
0	Enables CPU access of I^2C bus interface data register and control register.

I^2C transfer rate select 1, 0

Note: This register is valid only when an I^2C bus interface has been added as an H8S/2638 option.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	CLR3	CLR2	CLR1	CLR0
Initial value	0	0	0	0	1	1	1	1
R/W	R/(W)*1	R/(W)*1	R/(W)*1	R/(W)*1	W*2	W*2	W*2	W*2

Reserved bit

IIC clear 3 to 0

CLR3	CLR2	CLR1	CLR0	
0	0	—	—	Setting prohibited
	1	0	0	Setting prohibited
			1	IIC0 internal latch cleared
		1	0	IIC1 internal latch cleared
			1	IIC0 and IIC1 internal latch cleared
1	—	—	—	Invalid setting

Notes: *1 Should always be written with 0.
*2 Always read as 1.

This register is valid only when an I^2C bus interface has been added as an H8S/2638 option.

Bit	7	6	5	4	3	2	1	0
	SSBY	STS2	STS1	STS0	OPE	—	—	—
Initial value	0	1	0	1	1	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	—	—	—

Output Port Enable

0	In software standby mode, watch mode, and when making a direct transition, address bus and bus control signals are high-impedance
1	In software standby mode, watch mode, and when making a direct transition, the output state of the address bus and bus control signals is retained

Standby Timer Select 2 to 0

0	0	0	Standby time = 8192 states
		1	Standby time = 16384 states
	1	0	Standby time = 32768 states
		1	Standby time = 65536 states
1	0	0	Standby time = 131072 states
		1	Standby time = 262144 states
	1	0	Reserved
		1	Standby time = 16 states

Software Standby

0	Shifts to sleep mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode Shifts to sub-sleep mode when the SLEEP instruction is executed in sub-active mode
1	Shifts to software standby mode, sub-active mode, and watch mode when the SLEEP instruction is executed in high-speed mode or medium-speed mode Shifts to watch mode or high-speed mode when the SLEEP instruction is executed in sub-active mode

Bit	7	6	5	4	3	2	1	0
	MACS	—	INTM1	INTM0	NMIEG	—	—	RAME
Initial value	0	0	0	0	0	0	0	1
Read/Write	R/W	—	R/W	R/W	R/W	—	—	R/W

RAM Enable

0	On-chip RAM is disabled
1	On-chip RAM is enabled

NMI Edge Select

0	An interrupt is requested at the falling edge of NMI input
1	An interrupt is requested at the rising edge of NMI input

Interrupt Control Mode 1 and 0

INTM1	INTM0	Interrupt Control Mode	Description
0	0	0	Control of interrupts by I bit
	1	—	Setting prohibited
1	0	2	Control of interrupts by I2 to I0 bits and IPR
	1	—	Setting prohibited

MAC Saturation

0	Non-saturating calculation for MAC instruction
1	Saturating calculation for MAC instruction

Bit	7	6	5	4	3	2	1	0
	PSTOP	—	—	—	STCS	SCK2	SCK1	SCK0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	—	—	—	R/W	R/W	R/W	R/W

System Clock Select

0	0	0	Bus master in high-speed mode
		1	Medium-speed clock is ø/2
	1	0	Medium-speed clock is ø/4
		1	Medium-speed clock is ø/8
1	0	0	Medium-speed clock is ø/16
		1	Medium-speed clock is ø/32
	1	—	—

Frequency Multiplication Factor Switching Mode Select

0	Specified multiplication factor is valid after transition to software standby mode, watch mode*, or subactive mode*
1	Specified multiplication factor is valid immediately after STC bits are rewritten

ø Clock Output Disable

DDR	0	1	1
PSTOP	—	0	1
Hardware standby mode	High impedance	High impedance	High impedance
Software standby mode, watch mode*, and direct transition	High impedance	Fixed high	Fixed high
Sleep mode and subsleep mode*	High impedance	ø output	Fixed high
High-speed mode, medium-speed mode, and subactive mode*	High impedance	ø output	Fixed high

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are available in the U-mask and W-mask versions only.
These functions cannot be used with the other versions.

MDCR—Mode Control Register **H'FDE7** **System**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	MDS2	MDS1	MDS0
Initial value	1	0	0	0	0	—*	—*	—*
Read/Write	—	—	—	—	—	R	R	R

Mode Select 2 to 0
Indicate the input levels at pins MD2 to MD0

Note: * Determined by pins MD2 to MD0.

MSTPCRA—Module Stop Control Register A **H'FDE8** **System**

Bit	7	6	5	4	3	2	1	0
	MSTPA7	MSTPA6	MSTPA5	MSTPA4	MSTPA3	MSTPA2	MSTPA1	MSTPA0
Initial value	0	0	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Module Stop

0	Module stop mode is cleared
1	Module stop mode is set

MSTPCRB—Module Stop Control Register B **H'FDE9** **System**

Bit	7	6	5	4	3	2	1	0
	MSTPB7	MSTPB6	MSTPB5	MSTPB4	MSTPB3	MSTPB2	MSTPB1	MSTPB0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Module Stop

0	Module stop mode is cleared
1	Module stop mode is set

MSTPCRC—Module Stop Control Register C **H'FDEA** **System**

Bit	7	6	5	4	3	2	1	0
	MSTPC7	MSTPC6	MSTPC5	MSTPC4	MSTPC3	MSTPC2	MSTPC1	MSTPC0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Module Stop

0	Module stop mode is cleared
1	Module stop mode is set

PFCR—Pin Function Control Register **H'FDEB** **System**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	AE3	AE2	AE1	AE0
Initial value	0	0	0	0	1/0	1/0	1	1/0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Address Output Enable 3 to 0

0	0	0	0	A8–A23 address output disabled
			1	A8 address output enabled; A9–A23 address output disabled
		1	0	A8, A9 address output enabled; A10–A23 address output disabled
			1	A8–A10 address output enabled; A11–A23 address output disabled
	1	0	0	A8–A11 address output enabled; A12–A23 address output disabled
			1	A8–A12 address output enabled; A13–A23 address output disabled
		1	0	A8–A13 address output enabled; A14–A23 address output disabled
			1	A8–A14 address output enabled; A15–A23 address output disabled
1	0	0	0	A8–A15 address output enabled; A16–A23 address output disabled
			1	A8–A16 address output enabled; A17–A23 address output disabled
		1	0	A8–A17 address output enabled; A18–A23 address output disabled
			1	A8–A18 address output enabled; A19–A23 address output disabled
	1	0	0	A8–A19 address output enabled; A20–A23 address output disabled
			1	A8–A20 address output enabled; A21–A23 address output disabled (Initial value*)
		1	0	A8–A21 address output enabled; A22, A23 address output disabled
			1	A8–A23 address output enabled

Note: * In expanded mode with ROM, bits AE3 to AE0 are initialized to B'0000.
In ROMless expanded mode, bits AE3 to AE0 are initialized to B'1101.
Address pins A0 to A7 are made address outputs by setting the corresponding DDR bits to 1.

Bit	7	6	5	4	3	2	1	0
	DTON*	LSON*	NESEL*	SUBSTP*	RFCUT*	—	STC1	STC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Frequency Multiplication Factor

0	0	×1
	1	×2
1	0	×4
	1	Setting prohibited

Oscillation Circuit Feedback Resistance Control Bit

0	When the main clock is oscillating, sets the feedback resistance ON. When the main clock is stopped, sets the feedback resistance OFF
1	Sets the feedback resistance OFF

Subclock Enable

0	Enables subclock generation
1	Disables subclock generation

Noise Elimination Sampling Frequency Select

0	Sampling using 1/32 ×ø
1	Sampling using 1/4 ×ø

Low-Speed ON Flag

0	• When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, software standby mode, or watch mode* • When the SLEEP instruction is executed in sub-active mode, operation shifts to watch mode or shifts directly to high-speed mode • Operation shifts to high-speed mode when watch mode is cancelled
1	• When the SLEEP instruction is executed in high-speed mode, operation shifts to watch mode or sub-active mode • When the SLEEP instruction is executed in sub-active mode, operation shifts to sub-sleep mode or watch mode • Operation shifts to sub-active mode when watch mode is cancelled

Note: * Always set high-speed mode when shifting to watch mode or sub-active mode.

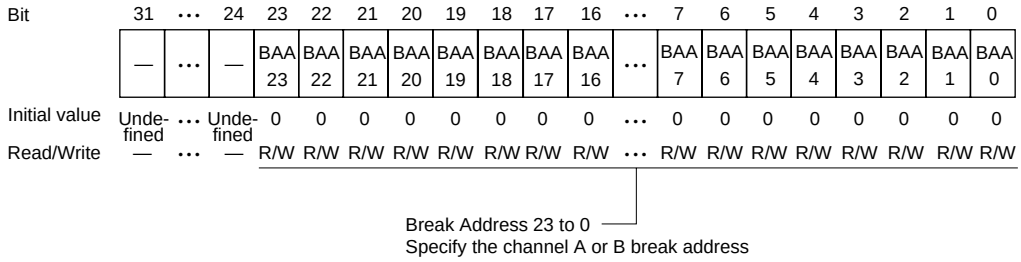
Direct Transition ON Flag

0	• When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts to sleep mode, software standby mode, or watch mode* • When the SLEEP instruction is executed in sub-active mode, operation shifts to sub-sleep mode or watch mode
1	• When the SLEEP instruction is executed in high-speed mode or medium-speed mode, operation shifts directly to sub-active mode*, or shifts to sleep mode or software standby mode • When the SLEEP instruction is executed in sub-active mode, operation shifts directly to high-speed mode, or shifts to sub-sleep mode

Note: * Bits 7 to 3 in LPWRCR are valid in the U-mask and W-mask versions; they are reserved bits in all other versions.
See section 23A.2.3 and 23B.2.3, Low-Power Control Register (LPWRCR), for more information.

BARA—Break Address Register A **H'FE00** **PBC**

BARB—Break Address Register B **H'FE04** **PBC**



Bit	7	6	5	4	3	2	1	0
	CMFA	CDA	BAMRA2	BAMRA1	BAMRA0	CSELA1	CSELA0	BIEA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Break Interrupt Enable

0	PC break interrupts are disabled
1	PC break interrupts are enabled

Break Condition Select

0	0	Instruction fetch is used as break condition
	1	Data read cycle is used as break condition
1	0	Data write cycle is used as break condition
	1	Data read/write cycle is used as break condition

Break Address Mask Register

0	0	0	All BARA bits are unmasked and included in break conditions
		1	BAA0 (lowest bit) is masked, and not included in break conditions
	1	0	BAA1–0 (lower 2 bits) are masked, and not included in break conditions
		1	BAA2–0 (lower 3 bits) are masked, and not included in break conditions
1	0	0	BAA3–0 (lower 4 bits) are masked, and not included in break conditions
		1	BAA7–0 (lower 8 bits) are masked, and not included in break conditions
	1	0	BAA11–0 (lower 12 bits) are masked, and not included in break conditions
		1	BAA15–0 (lower 16 bits) are masked, and not included in break conditions

CPU Cycle/DTC Cycle Select A

0	PC break is performed when CPU is bus master
1	PC break is performed when CPU or DTC is bus master

Condition Match Flag A

0	[Clearing condition] When 0 is written to CMFA after reading CMFA = 1
1	[Setting condition] When a condition set for channel A is satisfied

Note: * Only a 0 may be written to this bit to clear the flag.

ISCRH

Bit	15	14	13	12	11	10	9	8
	—	—	—	—	IRQ5SCB	IRQ5SCA	IRQ4SCB	IRQ4SCA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ISCRL

Bit	7	6	5	4	3	2	1	0
	IRQ3SCB	IRQ3SCA	IRQ2SCB	IRQ2SCA	IRQ1SCB	IRQ1SCA	IRQ0SCB	IRQ0SCA
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ5 to IRQ0 sense control A and B

IRQ5SCB to IRQ0SCB	IRQ5SCA to IRQ0SCA	Description
0	0	Interrupt request generated at $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ input at low level
	1	Interrupt request generated at falling edge of $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ input
1	0	Interrupt request generated at rising edge of $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ input
	1	Interrupt request generated at both falling and rising edges of $\overline{\text{IRQ5}}$ to $\overline{\text{IRQ0}}$ input

IER—IRQ Enable Register **H'FE14** **Interrupt Controller**

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ5E	IRQ4E	IRQ3E	IRQ2E	IRQ1E	IRQ0E
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

IRQ5 to IRQ0 Enable

0	IRQn interrupts disabled
1	IRQn interrupts enabled

(n = 5 to 0)

ISR—IRQ Status Register **H'FE15** **Interrupt Controller**

Bit	7	6	5	4	3	2	1	0
	—	—	IRQ5F	IRQ4F	IRQ3F	IRQ2F	IRQ1F	IRQ0F
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

IRQ5 to IRQ0 Flags

0	[Clearing conditions] • Cleared by reading IRQnF when set to 1, then writing 0 in IRQnF • When interrupt exception handling is executed while low-level detection is set (IRQnSCB = IRQnSCA = 0) and $\overline{\text{IRQn}}$ input is high • When IRQn interrupt exception handling is executed while falling, rising, or both-edge detection is set (IRQnSCB = 1 or IRQnSCA = 1) • When the DTC is activated by an IRQn interrupt, and the DISEL bit in MRB of the DTC is cleared to 0
1	[Setting conditions] • When $\overline{\text{IRQn}}$ input goes low when low-level detection is set (IRQnSCB = IRQnSCA = 0) • When a falling edge occurs in $\overline{\text{IRQn}}$ input while falling edge detection is set (IRQnSCB = 0, IRQnSCA = 1) • When a rising edge occurs in $\overline{\text{IRQn}}$ input while rising edge detection is set (IRQnSCB = 1, IRQnSCA = 0) • When a falling or rising edge occurs in $\overline{\text{IRQn}}$ input while both-edge detection is set (IRQnSCB = IRQnSCA = 1)

(n = 5 to 0)

Note: * Only 0 can be written, to clear the flag.

DTCERA—DTC Enable Register A	**H'FE16**	**DTC**
DTCERB—DTC Enable Register B	**H'FE17**	**DTC**
DTCERC—DTC Enable Register C	**H'FE18**	**DTC**
DTCERD—DTC Enable Register D	**H'FE19**	**DTC**
DTCERE—DTC Enable Register E	**H'FE1A**	**DTC**
DTCERF—DTC Enable Register F	**H'FE1B**	**DTC**
DTCERG—DTC Enable Register G	**H'FE1C**	**DTC**

Bit	7	6	5	4	3	2	1	0
	DTCE7	DTCE6	DTCE5	DTCE4	DTCE3	DTCE2	DTCE1	DTCE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

DTC Activation Enable

0	DTC activation by interrupt is disabled [Clearing conditions] • When data transfer ends with the DISEL bit set to 1 • When the specified number of transfers end
1	DTC activation by interrupt is enabled [Holding condition] When the DISEL bit is 0 and the specified number of transfers have not ended

Bit	7	6	5	4	3	2	1	0
	SWDTE	DTVEC6	DTVEC5	DTVEC4	DTVEC3	DTVEC2	DTVEC1	DTVEC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*1	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2	R/W*2

Set vector number for DTC software activation

DTC Software Activation Enable

0	DTC software activation is disabled [Clearing condition] When the DISEL bit is 0 and the specified number of transfers have not ended
1	DTC software activation is enabled [Holding conditions] • When data transfer ends with the DISEL bit set to 1 • When the specified number of transfers end • During software-activated deta transfer

Notes: *1 Only 1 can be written to the SWDTE bit.
*2 Bits DTVEC6 to DTVEC0 can be written to when SWDTE = 0.

Bit	7	6	5	4	3	2	1	0
	G3CMS1	G3CMS0	G2CMS1	G2CMS0	G1CMS1	G1CMS0	G0CMS1	G0CMS0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Group 0 Compare Match Select

0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

Group 1 Compare Match Select

0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

Group 2 Compare Match Select

0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

Group 3 Compare Match Select

0	0	Compare match in TPU channel 0
	1	Compare match in TPU channel 1
1	0	Compare match in TPU channel 2
	1	Compare match in TPU channel 3

Bit	7	6	5	4	3	2	1	0
	G3INV	G2INV	G1INV	G0INV	G3NOV	G2NOV	G1NOV	G0NOV
Initial value	1	1	1	1	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Group 0 Non-Overlap

0	Normal operation in pulse output group 0 (output values updated at compare match A in the selected TPU channel)
1	Non-overlapping operation in pulse output group 0 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Group 1 Non-Overlap

0	Normal operation in pulse output group 1 (output values updated at compare match A in the selected TPU channel)
1	Non-overlapping operation in pulse output group 1 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Group 2 Non-Overlap

0	Normal operation in pulse output group 2 (output values updated at compare match A in the selected TPU channel)
1	Non-overlapping operation in pulse output group 2 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Group 3 Non-Overlap

0	Normal operation in pulse output group 3 (output values updated at compare match A in the selected TPU channel)
1	Non-overlapping operation in pulse output group 3 (independent 1 and 0 output at compare match A or B in the selected TPU channel)

Group 0 Inversion

0	Inverted output for pulse output group 0 (low-level output at pin for a 1 in PODRL)
1	Direct output for pulse output group 0 (high-level output at pin for a 1 in PODRL)

Group 1 Inversion

0	Inverted output for pulse output group 1 (low-level output at pin for a 1 in PODRL)
1	Direct output for pulse output group 1 (high-level output at pin for a 1 in PODRL)

Group 2 Inversion

0	Inverted output for pulse output group 2 (low-level output at pin for a 1 in PODRH)
1	Direct output for pulse output group 2 (high-level output at pin for a 1 in PODRH)

Group 3 Inversion

0	Inverted output for pulse output group 3 (low-level output at pin for a 1 in PODRH)
1	Direct output for pulse output group 3 (high-level output at pin for a 1 in PODRH)

NDERH—Next Data Enable Register H **H'FE28** **PPG**

Bit	7	6	5	4	3	2	1	0
	NDER15	NDER14	NDER13	NDER12	NDER11	NDER10	NDER9	NDER8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next Data Enable

0	Pulse outputs PO15 to PO8 are disabled (NDR15 to NDR8 are not transferred to POD15 to POD8)
1	Pulse outputs PO15 to PO8 are enabled (NDR15 to NDR8 are transferred to POD15 to POD8)

NDERL—Next Data Enable Register L **H'FE29** **PPG**

Bit	7	6	5	4	3	2	1	0
	NDER7	NDER6	NDER5	NDER4	NDER3	NDER2	NDER1	NDER0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Next Data Enable

0	Pulse outputs PO7 to PO0 are disabled (NDR7 to NDR0 are not transferred to POD7 to POD0
1	Pulse outputs PO7 to PO0 are enabled (NDR7 to NDR0 are transferred to POD7 to POD0)

PODRH—Output Data Register H **H'FE2A** **PPG**

Bit	7	6	5	4	3	2	1	0
	POD15	POD14	POD13	POD12	POD11	POD10	POD9	POD8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * A bit that has been set for pulse output by NDER is read-only.

Bit	7	6	5	4	3	2	1	0
	POD7	POD6	POD5	POD4	POD3	POD2	POD1	POD0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Note: * A bit that has been set for pulse output by NDER is read-only.

Same Trigger for Pulse Output Groups

Address H'FE2C

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	NDR11	NDR10	NDR9	NDR8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address H'FE2E

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

Different Triggers for Pulse Output Groups

Address H'FE2C

Bit	7	6	5	4	3	2	1	0
	NDR15	NDR14	NDR13	NDR12	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Address H'FE2E

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR11	NDR10	NDR9	NDR8
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Note: For details, see section 11.2.4, Notes on NDR Access.

Same Trigger for Pulse Output Groups

Address H'FE2D

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	NDR3	NDR2	NDR1	NDR0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Address H'FE2F

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	—	—
Initial value	1	1	1	1	1	1	1	1
Read/Write	—	—	—	—	—	—	—	—

Different Triggers for Pulse Output Groups

Address H'FE2D

Bit	7	6	5	4	3	2	1	0
	NDR7	NDR6	NDR5	NDR4	—	—	—	—
Initial value	0	0	0	0	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	—	—	—	—

Address H'FE2F

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	NDR3	NDR2	NDR1	NDR0
Initial value	1	1	1	1	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Note: For details, see section 11.2.4, Notes on NDR Access.

P1DDR—Port 1 Data Direction Register **H'FE30** **Port**

Bit	7	6	5	4	3	2	1	0
	P17DDR	P16DDR	P15DDR	P14DDR	P13DDR	P12DDR	P11DDR	P10DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specify input or output for each of the pins in port 1

P3DDR—Port 3 Data Direction Register **H'FE32** **Port**

Bit	7	6	5	4	3	2	1	0
	—	—	P35DDR	P34DDR	P33DDR	P32DDR	P31DDR	P30DDR
Initial value	Undefined	Undefined	0	0	0	0	0	0
Read/Write	—	—	W	W	W	W	W	W

Specify input or output for each of the pins in port 3

PADDR—Port A Data Direction Register **H'FE39** **Port**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3DDR	PA2DDR	PA1DDR	PA0DDR
Initial value	Undefined	Undefined	Undefined	Undefined	0	0	0	0
Read/Write	—	—	—	—	W	W	W	W

Specify input or output for each of the pins in port A

PBDDR—Port B Data Direction Register **H'FE3A** **Port**

Bit	7	6	5	4	3	2	1	0
	PB7DDR	PB6DDR	PB5DDR	PB4DDR	PB3DDR	PB2DDR	PB1DDR	PB0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specify input or output for each of the pins in port B

PCDDR—Port C Data Direction Register **H'FE3B** **Port**

Bit	7	6	5	4	3	2	1	0
	PC7DDR	PC6DDR	PC5DDR	PC4DDR	PC3DDR	PC2DDR	PC1DDR	PC0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specify input or output for each of the pins in port C

PDDDR—Port D Data Direction Register **H'FE3C** **Port**

Bit	7	6	5	4	3	2	1	0
	PD7DDR	PD6DDR	PD5DDR	PD4DDR	PD3DDR	PD2DDR	PD1DDR	PD0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specify input or output for each of the pins in port D

PEDDR—Port E Data Direction Register **H'FE3D** **Port**

Bit	7	6	5	4	3	2	1	0
	PE7DDR	PE6DDR	PE5DDR	PE4DDR	PE3DDR	PE2DDR	PE1DDR	PE0DDR
Initial value	0	0	0	0	0	0	0	0
Read/Write	W	W	W	W	W	W	W	W

Specify input or output for each of the pins in port E

PFDDR—Port F Data Direction Register **H'FE3E** **Port**

Bit	7	6	5	4	3	2	1	0
	PF7DDR	PF6DDR	PF5DDR	PF4DDR	PF3DDR	—	—	PF0DDR
Modes 4, 5, 6								
Initial value	1	0	0	0	0	Undefined	Undefined	0
Read/Write	W	W	W	W	W	—	—	W
Mode 7								
Initial value	0	0	0	0	0	Undefined	Undefined	0
Read/Write	W	W	W	W	W	—	—	W

Specify input or output for each of the pins in port F

PAPCR—Port A MOS Pull-Up Control Register H'FE40 Port

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3PCR	PA2PCR	PA1PCR	PA0PCR
Initial value	Undefined	Undefined	Undefined	Undefined	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Control the MOS input pull-up function incorporated into port A

PBPCR—Port B MOS Pull-Up Control Register H'FE41 Port

Bit	7	6	5	4	3	2	1	0
	PB7PCR	PB6PCR	PB5PCR	PB4PCR	PB3PCR	PB2PCR	PB1PCR	PB0PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Control the MOS input pull-up function incorporated into port B

PCPCR—Port C MOS Pull-Up Control Register H'FE42 Port

Bit	7	6	5	4	3	2	1	0
	PC7PCR	PC6PCR	PC5PCR	PC4PCR	PC3PCR	PC2PCR	PC1PCR	PC0PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Control the MOS input pull-up function incorporated into port C

PDPCR—Port D MOS Pull-Up Control Register H'FE43 Port

Bit	7	6	5	4	3	2	1	0
	PD7PCR	PD6PCR	PD5PCR	PD4PCR	PD3PCR	PD2PCR	PD1PCR	PD0PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Control the MOS input pull-up function incorporated into port D

PEPCR—Port E MOS Pull-Up Control Register H'FE44 Port

Bit	7	6	5	4	3	2	1	0
	PE7PCR	PE6PCR	PE5PCR	PE4PCR	PE3PCR	PE2PCR	PE1PCR	PE0PCR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Control the MOS input pull-up function incorporated into port E

P3ODR—Port 3 Open Drain Control Register H'FE46 Port

Bit	7	6	5	4	3	2	1	0
	—	—	P35ODR	P34ODR	P33ODR	P32ODR	P31ODR	P30ODR
Initial value	Undefined	Undefined	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Control whether PMOS is on or off for each port 3 pin

PAODR—Port A Open Drain Control Register H'FE47 Port

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3ODR	PA2ODR	PA1ODR	PA0ODR
Initial value	Undefined	Undefined	Undefined	Undefined	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Control whether PMOS is on or off for each port A pin

PBODR—Port B Open Drain Control Register H'FE48 Port

Bit	7	6	5	4	3	2	1	0
	PB7ODR	PB6ODR	PB5ODR	PB4ODR	PB3ODR	PB2ODR	PB1ODR	PB0ODR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Control whether PMOS is on or off for each port B pin

Bit	7	6	5	4	3	2	1	0
	PC7ODR	PC6ODR	PC5ODR	PC4ODR	PC3ODR	PC2ODR	PC1ODR	PC0ODR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Control whether PMOS is on or off for each port C pin

Bit	7	6	5	4	3	2	1	0
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time Prescaler

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	Internal clock: counts on ø/1024
	1	0	Internal clock: counts on ø/256
		1	Internal clock: counts on ø/4096

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: Internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Counter Clear

0	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input capture
	1	0	TCNT cleared by TGRB compare match/input capture
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1
1	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRC compare match/input capture*2
	1	0	TCNT cleared by TGRD compare match/input capture*2
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Notes: *1 Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

*2 When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Bit	7	6	5	4	3	2	1	0
	—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Mode

0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	*	*	*	—

*: Don't care

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.
2. Phase counting mode cannot be set for channel 3. In this case, 0 should always be written to MD2.

Buffer Operation A

0	TGRA operates normally
1	TGRA and TGRC used together for buffer operation

Buffer Operation

0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

Bit	7	6	5	4	3	2	1	0
	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR3A I/O Control

0	0	0	0	TGR3A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3A is input capture register	Capture input source is TIOCA3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down

*: Don't care

TGR3B I/O Control

0	0	0	0	TGR3B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3B is input capture register	Capture input source is TIOCB3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down*1

*: Don't care

Note: *1 When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ø/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

Bit	7	6	5	4	3	2	1	0
	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR3C I/O Control

0	0	0	0	TGR3C is output compare register*1	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3C is input capture register*1	Capture input source is TIOCC3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down

*: Don't care

Note: *1 When the BFA bit in TMDR3 is set to 1 and TGR3C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

TGR3D I/O Control

0	0	0	0	TGR3D is output compare register*2	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR3D is input capture register*2	Capture input source is TIOCD3 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 4/count clock	Input capture at TCNT4 count-up/ count-down*1

*: Don't care

Note: *1 When bits TPSC2 to TPSC0 in TCR4 are set to B'000 and ø/1 is used as the TCNT4 count clock, this setting is invalid and input capture is not generated.

*2 When the BFB bit in TMDR3 is set to 1 and TGR3D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Bit	7	6	5	4	3	2	1	0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
Read/Write	R/W	—	—	R/W	R/W	R/W	R/W	R/W

TGR Interrupt Enable A

0	Interrupt requests (TGIA) by TGFA bit disabled
1	Interrupt requests (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt requests (TGIB) by TGFB bit disabled
1	Interrupt requests (TGIB) by TGFB bit enabled

TGR Interrupt Enable C

0	Interrupt requests (TGIC) by TGFC bit disabled
1	Interrupt requests (TGIC) by TGFC bit enabled

TGR Interrupt Enable D

0	Interrupt requests (TGID) by TGFD bit disabled
1	Interrupt requests (TGID) by TGFD bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

Bit	7	6	5	4	3	2	1	0
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Input Capture/Output Compare Flag C

0	[Clearing conditions] • When DTC is activated by TGIC interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register

Input Capture/Output Compare Flag D

0	[Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Note: * Can only be written with 0 for flag clearing.

TCNT3—Timer Counter 3 **H'FE86** **TPU3**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Up-counter

TGR3A—Timer General Register 3A **H'FE88** **TPU3**
TGR3B—Timer General Register 3B **H'FE8A** **TPU3**
TGR3C—Timer General Register 3C **H'FE8C** **TPU3**
TGR3D—Timer General Register 3D **H'FE8E** **TPU3**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time Prescaler

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	External clock: counts on TCLKC pin input
	1	0	Internal clock: counts on ø/1024
		1	Counts on TCNT5 overflow/underflow

Note: This setting is ignored when channel 4 is in phase counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is ignored when channel 4 is in phase counting mode.
Internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Counter Clear

0	0	TCNT clearing disabled
	1	TCNT cleared by TGRA compare match/input capture
1	0	TCNT cleared by TGRB compare match/input capture
	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*

Note: * Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MD3	MD2	MD1	MD0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Mode

0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	*	*	*	—

*: Don't care

Note: MD3 is a reserved bit. In a write, it should always be written with 0.

Bit	7	6	5	4	3	2	1	0
	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR4A I/O Control

0	0	0	0	TGR4A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR4A is input capture register	Capture input source is TIOCA4 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR3A compare match/ input capture	Input capture at generation of TGR3A compare match/input capture

*: Don't care

TGR4B I/O Control

0	0	0	0	TGR4B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR4B is input capture register	Capture input source is TIOCB4 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR3C compare match/ input capture	Input capture at generation of TGR3C compare match/input capture

*: Don't care

Bit	7	6	5	4	3	2	1	0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
Read/Write	R/W	—	R/W	R/W	—	—	R/W	R/W

TGR Interrupt Enable A

0	Interrupt requests (TGIA) by TGFA bit disabled
1	Interrupt requests (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt requests (TGIB) by TGFB bit disabled
1	Interrupt requests (TGIB) by TGFB bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

Bit	7	6	5	4	3	2	1	0
	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	1	1	0	0	0	0	0	0
Read/Write	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Underflow Flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

TCNT4—Timer Counter 4 **H'FE96** **TPU4**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Up/down-counter*

Note: * These counters can be used as up/down-counters only in phase counting mode or when counting overflow/underflow on another channel. In other cases they function as up-counters.

TGR4A—Timer General Register 4A **H'FE98** **TPU4**

TGR4B—Timer General Register 4B **H'FE9A** **TPU4**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time Prescaler

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	External clock: counts on TCLKC pin input
	1	0	Internal clock: counts on ø/256
		1	External clock: counts on TCLKD pin input

Note: This setting is ignored when channel 5 is in phase counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is ignored when channel 5 is in phase counting mode.
Internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Counter Clear

0	0	TCNT clearing disabled
	1	TCNT cleared by TGRA compare match/input capture
1	0	TCNT cleared by TGRB compare match/input capture
	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*

Note: * Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MD3	MD2	MD1	MD0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Mode

0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	*	*	*	—

*: Don't care

Note: MD3 is a reserved bit. In a write, it should always be written with 0.

Bit	7	6	5	4	3	2	1	0
	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR5A I/O Control

0	0	0	0	TGR5A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR5A is input capture register	Capture input source is TIOCA5 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

TGR5B I/O Control

0	0	0	0	TGR5B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR5B is input capture register	Capture input source is TIOCB5 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

Bit	7	6	5	4	3	2	1	0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
Read/Write	R/W	—	R/W	R/W	—	—	R/W	R/W

TGR Interrupt Enable A

0	Interrupt requests (TGIA) by TGFA bit disabled
1	Interrupt requests (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt requests (TGIB) by TGFB bit disabled
1	Interrupt requests (TGIB) by TGFB bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

Bit	7	6	5	4	3	2	1	0
	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	1	1	0	0	0	0	0	0
Read/Write	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Underflow Flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

TCNT5—Timer Counter 5 **H'FEA6** **TPU5**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Up/down-counter*

Note: * These counters can be used as up/down-counters only in phase counting mode or when counting overflow/underflow on another channel. In other cases they function as up-counters.

TGR5A—Timer General Register 5A **H'FEA8** **TPU5**
TGR5B—Timer General Register 5B **H'FEAA** **TPU5**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TSTR—Timer Start Register **H'FEB0** **TPU**

Bit	7	6	5	4	3	2	1	0
	—	—	CST5	CST4	CST3	CST2	CST1	CST0
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Counter Start

0	TCNTn count operation is stopped
1	TCNTn performs count operation

(n = 5 to 0)

Note: If 0 is written to the CST bit during operation with the TIOC pin designated for output, the counter stops but the TIOC pin output compare output level is retained. If TIOR is written to when the CST bit is cleared to 0, the pin output level will be changed to the set initial output value.

Bit	7	6	5	4	3	2	1	0
	—	—	SYNC5	SYNC4	SYNC3	SYNC2	SYNC1	SYNC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Timer Synchro

0	TCNTn operates independently (TCNT presetting/ clearing is unrelated to other channels)
1	TCNTn performs synchronous operation TCNT synchronous presetting/synchronous clearing is possible

(n = 5 to 0)

Note: 1. To set synchronous operation, the SYNC bits for at least two channels must be set to 1.
2. To set synchronous clearing, in addition to the SYNC bit , the TCNT clearing source must also be set by means of bits CCLR2 to CCLR0 in TCR.

IPRA—Interrupt Priority Register A	**H'FEC0**	**INT**
IPRB—Interrupt Priority Register B	**H'FEC1**	**INT**
IPRC—Interrupt Priority Register C	**H'FEC2**	**INT**
IPRD—Interrupt Priority Register D	**H'FEC3**	**INT**
IPRE—Interrupt Priority Register E	**H'FEC4**	**INT**
IPRF—Interrupt Priority Register F	**H'FEC5**	**INT**
IPRG—Interrupt Priority Register G	**H'FEC6**	**INT**
IPRH—Interrupt Priority Register H	**H'FEC7**	**INT**
IPRJ—Interrupt Priority Register J	**H'FEC9**	**INT**
IPRK—Interrupt Priority Register K	**H'FECA**	**INT**
IPRL—Interrupt Priority Register L	**H'FECB**	**INT**
IPRM—Interrupt Priority Register M	**H'FECC**	**INT**

Bit	7	6	5	4	3	2	1	0
	—	IPR6	IPR5	IPR4	—	IPR2	IPR1	IPR0
Initial value	0	1	1	1	0	1	1	1
Read/Write	—	R/W	R/W	R/W	—	R/W	R/W	R/W

Correspondence between Interrupt Sources and IPR Settings

Register	Bits	
	6 to 4	2 to 0
IPRA	IRQ0	IRQ1
IPRB	IRQ2	IRQ4
	IRQ3	IRQ5
IPRC	—*1	DTC
IPRD	Watchdog timer 0	—*1
IPRE	PC break	A/D converter, watchdog timer 1
IPRF	TPU channel 0	TPU channel 1
IPRG	TPU channel 2	TPU channel 3
IPRH	TPU channel 4	TPU channel 5
IPRJ	—*1	SCI channel 0
IPRK	SCI channel 1	SCI channel 2
IPRL	—*1	IIC (Option)*2
IPRM	PWM channel 1, 2, HCAN channel 1	HCAN channel 0

Notes: *1 Reserved. Read-only bits, always read as 1.
*2 I^2C bus interface is available as an option in the H8S/2638 and H8S/2639. The IIC bit becomes reserved bit when this optional feature is not used.

ABWCR—Bus Width Control Register **H'FED0** **Bus Controller**

Bit	7	6	5	4	3	2	1	0
	ABW7	ABW6	ABW5	ABW4	ABW3	ABW2	ABW1	ABW0
Modes 5 to 7								
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Mode 4								
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 Bus Width Control

0	Area n is designated for 16-bit access
1	Area n is designated for 8-bit access

(n = 7 to 0)

ASTCR—Access State Control Register **H'FED1** **Bus Controller**

Bit	7	6	5	4	3	2	1	0
	AST7	AST6	AST5	AST4	AST3	AST2	AST1	AST0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 7 to 0 Access State Control

0	Area n is designated for 2-state access Wait state insertion in area n external space is disabled
1	Area n is designated for 3-state access Wait state insertion in area n external space is enabled

(n = 7 to 0)

Bit	7	6	5	4	3	2	1	0
	W71	W70	W61	W60	W51	W50	W41	W40
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 4 Wait Control 1 and 0

0	0	Program wait not inserted when external space area 4 is accessed
	1	1 program wait state inserted when external space area 4 is accessed
1	0	2 program wait states inserted when external space area 4 is accessed
	1	3 program wait states inserted when external space area 4 is accessed

Area 5 Wait Control 1 and 0

0	0	Program wait not inserted when external space area 5 is accessed
	1	1 program wait state inserted when external space area 5 is accessed
1	0	2 program wait states inserted when external space area 5 is accessed
	1	3 program wait states inserted when external space area 5 is accessed

Area 6 Wait Control 1 and 0

0	0	Program wait not inserted when external space area 6 is accessed
	1	1 program wait state inserted when external space area 6 is accessed
1	0	2 program wait states inserted when external space area 6 is accessed
	1	3 program wait states inserted when external space area 6 is accessed

Area 7 Wait Control 1 and 0

0	0	Program wait not inserted when external space area 7 is accessed
	1	1 program wait state inserted when external space area 7 is accessed
1	0	2 program wait states inserted when external space area 7 is accessed
	1	3 program wait states inserted when external space area 7 is accessed

Bit	7	6	5	4	3	2	1	0
	W31	W30	W21	W20	W11	W10	W01	W00
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Area 0 Wait Control 1 and 0

0	0	Program wait not inserted when external space area 0 is accessed
	1	1 program wait state inserted when external space area 0 is accessed
1	0	2 program wait states inserted when external space area 0 is accessed
	1	3 program wait states inserted when external space area 0 is accessed

Area 1 Wait Control 1 and 0

0	0	Program wait not inserted when external space area 1 is accessed
	1	1 program wait state inserted when external space area 1 is accessed
1	0	2 program wait states inserted when external space area 1 is accessed
	1	3 program wait states inserted when external space area 1 is accessed

Area 2 Wait Control 1 and 0

0	0	Program wait not inserted when external space area 2 is accessed
	1	1 program wait state inserted when external space area 2 is accessed
1	0	2 program wait states inserted when external space area 2 is accessed
	1	3 program wait states inserted when external space area 2 is accessed

Area 3 Wait Control 1 and 0

0	0	Program wait not inserted when external space area 3 is accessed
	1	1 program wait state inserted when external space area 3 is accessed
1	0	2 program wait states inserted when external space area 3 is accessed
	1	3 program wait states inserted when external space area 3 is accessed

Bit	7	6	5	4	3	2	1	0
	ICIS1	ICIS0	BRSTRM	BRSTS1	BRSTS0	—	—	—
Initial value	1	1	0	1	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Burst Cycle Select 0

0	Max. 4 words in burst access
1	Max. 8 words in burst access

Burst Cycle Select 1

0	Burst cycle comprises 1 state
1	Burst cycle comprises 2 states

Burst ROM Enable

0	Area 0 is basic bus interface
1	Area 0 is burst ROM interface

Idle Cycle Insert 0

0	Idle cycle not inserted in case of successive external read and external write cycles
1	Idle cycle inserted in case of successive external read and external write cycles

Idle Cycle Insert 1

0	Idle cycle not inserted in case of successive external read cycles in different areas
1	Idle cycle inserted in case of successive external read cycles in different areas

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	—	—	WDBE	—
Initial value	0	0	0	0	1	0	0	0
Read/Write	R/W	R/W	—	R/W	R/W	R/W	R/W	—

Write Data Buffer Enable

0	Write data buffer function not used
1	Write data buffer function used

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	RAMS	RAM2	RAM1	RAM0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R/W	R/W	R/W	R/W	R/W	R/W

Flash Memory Area Selection

• H8S/2636

Addresses	Block Name	RAMS	RAM2	RAM1	RAM0
H'FFE000–H'FFE3FF	RAM area 1 kB	0	*	*	*
H'000000–H'0003FF	EB0 (1 kB)	1	0	0	
H'000400–H'0007FF	EB1 (1 kB)			1	
H'000800–H'000BFF	EB2 (1 kB)		1	0	
H'000C00–H'000FFF	EB3 (1 kB)			1	

*: Don't care

• H8S/2638

Addresses	Block Name	RAMS	RAM2	RAM1	RAM0
H'FFD000–H'FFDFFF	RAM area 4 kB	0	*	*	*
H'000000–H'000FFF	EB0 (4 kB)	1	0	0	0
H'001000–H'001FFF	EB1 (4 kB)				1
H'002000–H'002FFF	EB2 (4 kB)			1	0
H'003000–H'003FFF	EB3 (4 kB)				1
H'004000–H'004FFF	EB4 (4 kB)		1	0	0
H'005000–H'005FFF	EB5 (4 kB)				1
H'006000–H'006FFF	EB6 (4 kB)			1	0
H'007000–H'007FFF	EB7 (4 kB)				1

*: Don't care

RAM Select

0	Emulation not selected Program/erase-protection of all flash memory blocks is disabled
1	Emulation selected Program/erase-protection of all flash memory blocks is enabled

P1DR—Port 1 Data Register **H'FF00** **Port**

Bit	7	6	5	4	3	2	1	0
	P17DR	P16DR	P15DR	P14DR	P13DR	P12DR	P11DR	P10DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

P3DR—Port 3 Data Register **H'FF02** **Port**

Bit	7	6	5	4	3	2	1	0
	—	—	P35DR	P34DR	P33DR	P32DR	P31DR	P30DR
Initial value	Undefined	Undefined	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

PADR—Port A Data Register **H'FF09** **Port**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3DR	PA2DR	PA1DR	PA0DR
Initial value	Undefined	Undefined	Undefined	Undefined	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

PBDR—Port B Data Register **H'FF0A** **Port**

Bit	7	6	5	4	3	2	1	0
	PB7DR	PB6DR	PB5DR	PB4DR	PB3DR	PB2DR	PB1DR	PB0DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PCDR—Port C Data Register **H'FF0B** **Port**

Bit	7	6	5	4	3	2	1	0
	PC7DR	PC6DR	PC5DR	PC4DR	PC3DR	PC2DR	PC1DR	PC0DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PDDR—Port D Data Register **H'FF0C** **Port**

Bit	7	6	5	4	3	2	1	0
	PD7DR	PD6DR	PD5DR	PD4DR	PD3DR	PD2DR	PD1DR	PD0DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

PEDR—Port E Data Register **H'FF0D** **Port**

Bit	7	6	5	4	3	2	1	0
	PE7DR	PE6DR	PE5DR	PE4DR	PE3DR	PE2DR	PE1DR	PE0DR
Initial value	0	0	0	0	0	0	0	0
Read/Write	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
	PF7DR	PF6DR	PF5DR	PF4DR	PF3DR	—	—	PF0DR
Initial value	0	0	0	0	0	Undefined	Undefined	0
Read/Write	R/W	R/W	R/W	R/W	R/W	—	—	R/W

Bit	7	6	5	4	3	2	1	0
	CCLR2	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time Prescaler

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	External clock: counts on TCLKB pin input
	1	0	External clock: counts on TCLKC pin input
		1	External clock: counts on TCLKD pin input

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: Internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Counter Clear

0	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRA compare match/input capture
	1	0	TCNT cleared by TGRB compare match/input capture
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1
1	0	0	TCNT clearing disabled
		1	TCNT cleared by TGRC compare match/input capture*2
	1	0	TCNT cleared by TGRD compare match/input capture*2
		1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*1

Notes: *1 Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

*2 When TGRC or TGRD is used as a buffer register, TCNT is not cleared because the buffer register setting has priority, and compare match/input capture does not occur.

Bit	7	6	5	4	3	2	1	0
	—	—	BFB	BFA	MD3	MD2	MD1	MD0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W

Mode

0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	*	*	*	—

*: Don't care

Notes: 1. MD3 is a reserved bit. In a write, it should always be written with 0.
2. Phase counting mode cannot be set for channel 0. In this case, 0 should always be written to MD2.

Buffer Operation A

0	TGRA operates normally
1	TGRA and TGRC used together for buffer operation

Buffer Operation

0	TGRB operates normally
1	TGRB and TGRD used together for buffer operation

Bit	7	6	5	4	3	2	1	0
	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR0A I/O Control

0	0	0	0	TGR0A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0A is input capture register	Capture input source is TIOCA0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down

*: Don't care

TGR0B I/O Control

0	0	0	0	TGR0B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0B is input capture register	Capture input source is TIOCB0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down*1

*: Don't care

Note: *1 When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ø/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

Bit	7	6	5	4	3	2	1	0
	IOD3	IOD2	IOD1	IOD0	IOC3	IOC2	IOC1	IOC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR0C I/O Control

0	0	0	0	TGR0C is output compare register*1	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0C is input capture register*1	Capture input source is TIOCC0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down

*: Don't care

Note: *1 When the BFA bit in TMDR0 is set to 1 and TGR0C is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

TGR0D I/O Control

0	0	0	0	TGR0D is output compare register*2	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR0D is input capture register*2	Capture input source is TIOCD0 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is channel 1/count clock	Input capture at TCNT1 count-up/ count-down*1

*: Don't care

Notes: *1 When bits TPSC2 to TPSC0 in TCR1 are set to B'000 and ø/1 is used as the TCNT1 count clock, this setting is invalid and input capture is not generated.

*2 When the BFB bit in TMDR0 is set to 1 and TGR0D is used as a buffer register, this setting is invalid and input capture/output compare is not generated.

Note: When TGRC or TGRD is designated for buffer operation, this setting is invalid and the register operates as a buffer register.

Bit	7	6	5	4	3	2	1	0
	TTGE	—	—	TCIEV	TGIED	TGIEC	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
Read/Write	R/W	—	—	R/W	R/W	R/W	R/W	R/W

TGR Interrupt Enable A

0	Interrupt requests (TGIA) by TGFA bit disabled
1	Interrupt requests (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt requests (TGIB) by TGFB bit disabled
1	Interrupt requests (TGIB) by TGFB bit enabled

TGR Interrupt Enable C

0	Interrupt requests (TGIC) by TGFC bit disabled
1	Interrupt requests (TGIC) by TGFC bit enabled

TGR Interrupt Enable D

0	Interrupt requests (TGID) by TGFD bit disabled
1	Interrupt requests (TGID) by TGFD bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

Bit	7	6	5	4	3	2	1	0
	—	—	—	TCFV	TGFD	TGFC	TGFB	TGFA
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	—	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*

Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Input Capture/Output Compare Flag C

0	[Clearing conditions] • When DTC is activated by TGIC interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFC after reading TGFC = 1
1	[Setting conditions] • When TCNT = TGRC while TGRC is functioning as output compare register • When TCNT value is transferred to TGRC by input capture signal while TGRC is functioning as input capture register

Input Capture/Output Compare Flag D

0	[Clearing conditions] • When DTC is activated by TGID interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFD after reading TGFD = 1
1	[Setting conditions] • When TCNT = TGRD while TGRD is functioning as output compare register • When TCNT value is transferred to TGRD by input capture signal while TGRD is functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Note: * Can only be written with 0 for flag clearing.

TCNT0—Timer Counter 0 **H'FF16** **TPU0**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Up-counter

TGR0A—Timer General Register 0A **H'FF18** **TPU0**
TGR0B—Timer General Register 0B **H'FF1A** **TPU0**
TGR0C—Timer General Register 0C **H'FF1C** **TPU0**
TGR0D—Timer General Register 0D **H'FF1E** **TPU0**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time Prescaler

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	External clock: counts on TCLKB pin input
	1	0	Internal clock: counts on ø/256
		1	Counts on TCNT2 overflow/underflow

Note: This setting is ignored when channel 1 is in phase counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is ignored when channel 1 is in phase counting mode.
Internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Counter Clear

0	0	TCNT clearing disabled
	1	TCNT cleared by TGRA compare match/input capture
1	0	TCNT cleared by TGRB compare match/input capture
	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*

Note: * Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MD3	MD2	MD1	MD0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Mode

0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	*	*	*	—

*: Don't care

Note: MD3 is a reserved bit. In a write, it should always be written with 0.

Bit	7	6	5	4	3	2	1	0
	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR1A I/O Control

0	0	0	0	TGR1A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR1A is input capture register	Capture input source is TIOCA1 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR0A compare match/ input capture	Input capture at generation of channel 0/TGR0A compare match/ input capture

*: Don't care

TGR1B I/O Control

0	0	0	0	TGR1B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	0	0	0	TGR1B is input capture register	Capture input source is TIOCB1 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges
	1	*	*		Capture input source is TGR0C compare match/ input capture	Input capture at generation of TGR0C compare match/input capture

*: Don't care

Bit	7	6	5	4	3	2	1	0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
Read/Write	R/W	—	R/W	R/W	—	—	R/W	R/W

TGR Interrupt Enable A

0	Interrupt requests (TGIA) by TGFA bit disabled
1	Interrupt requests (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt requests (TGIB) by TGFB bit disabled
1	Interrupt requests (TGIB) by TGFB bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

Bit	7	6	5	4	3	2	1	0
	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	1	1	0	0	0	0	0	0
Read/Write	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Underflow Flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

TCNT1—Timer Counter 1 **H'FF26** **TPU1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Up/down-counter*

Note: * These counters can be used as up/down-counters only in phase counting mode or when counting overflow/underflow on another channel. In other cases they function as up-counters.

TGR1A—Timer General Register 1A **H'FF28** **TPU1**

TGR1B—Timer General Register 1B **H'FF2A** **TPU1**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	—	CCLR1	CCLR0	CKEG1	CKEG0	TPSC2	TPSC1	TPSC0
Initial value	0	0	0	0	0	0	0	0
Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Time Prescaler

0	0	0	Internal clock: counts on ø/1
		1	Internal clock: counts on ø/4
	1	0	Internal clock: counts on ø/16
		1	Internal clock: counts on ø/64
1	0	0	External clock: counts on TCLKA pin input
		1	External clock: counts on TCLKB pin input
	1	0	External clock: counts on TCLKC pin input
		1	Internal clock: counts on ø/1024

Note: This setting is ignored when channel 2 is in phase counting mode.

Clock Edge

0	0	Count at rising edge
	1	Count at falling edge
1	—	Count at both edges

Note: This setting is ignored when channel 2 is in phase counting mode.
Internal clock edge selection is valid when the input clock is ø/4 or slower. This setting is ignored if the input clock is ø/1, or when overflow/underflow of another channel is selected.

Counter Clear

0	0	TCNT clearing disabled
	1	TCNT cleared by TGRA compare match/input capture
1	0	TCNT cleared by TGRB compare match/input capture
	1	TCNT cleared by counter clearing for another channel performing synchronous clearing/synchronous operation*

Note: * Synchronous operation setting is performed by setting the SYNC bit in TSYR to 1.

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	MD3	MD2	MD1	MD0
Initial value	1	1	0	0	0	0	0	0
Read/Write	—	—	—	—	R/W	R/W	R/W	R/W

Mode

0	0	0	0	Normal operation
			1	Reserved
		1	0	PWM mode 1
			1	PWM mode 2
	1	0	0	Phase counting mode 1
			1	Phase counting mode 2
		1	0	Phase counting mode 3
			1	Phase counting mode 4
1	*	*	*	—

*: Don't care

Note: MD3 is a reserved bit. In a write, it should always be written with 0.

Bit	7	6	5	4	3	2	1	0
	IOB3	IOB2	IOB1	IOB0	IOA3	IOA2	IOA1	IOA0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

TGR2A I/O Control

0	0	0	0	TGR2A is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR2A is input capture register	Capture input source is TIOCA2 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

TGR2B I/O Control

0	0	0	0	TGR2B is output compare register	Output disabled	
			1		Initial output is 0 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
	1	0	0		Output disabled	
			1		Initial output is 1 output	0 output at compare match
		1	0			1 output at compare match
			1			Toggle output at compare match
1	*	0	0	TGR2B is input capture register	Capture input source is TIOCB2 pin	Input capture at rising edge
			1			Input capture at falling edge
		1	*			Input capture at both edges

*: Don't care

Bit	7	6	5	4	3	2	1	0
	TTGE	—	TCIEU	TCIEV	—	—	TGIEB	TGIEA
Initial value	0	1	0	0	0	0	0	0
Read/Write	R/W	—	R/W	R/W	—	—	R/W	R/W

TGR Interrupt Enable A

0	Interrupt requests (TGIA) by TGFA bit disabled
1	Interrupt requests (TGIA) by TGFA bit enabled

TGR Interrupt Enable B

0	Interrupt requests (TGIB) by TGFB bit disabled
1	Interrupt requests (TGIB) by TGFB bit enabled

Overflow Interrupt Enable

0	Interrupt requests (TCIV) by TCFV disabled
1	Interrupt requests (TCIV) by TCFV enabled

Underflow Interrupt Enable

0	Interrupt requests (TCIU) by TCFU disabled
1	Interrupt requests (TCIU) by TCFU enabled

A/D Conversion Start Request Enable

0	A/D conversion start request generation disabled
1	A/D conversion start request generation enabled

Bit	7	6	5	4	3	2	1	0
	TCFD	—	TCFU	TCFV	—	—	TGFB	TGFA
Initial value	1	1	0	0	0	0	0	0
Read/Write	R	—	R/(W)*	R/(W)*	—	—	R/(W)*	R/(W)*

Input Capture/Output Compare Flag A

0	[Clearing conditions] • When DTC is activated by TGIA interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFA after reading TGFA = 1
1	[Setting conditions] • When TCNT = TGRA while TGRA is functioning as output compare register • When TCNT value is transferred to TGRA by input capture signal while TGRA is functioning as input capture register

Input Capture/Output Compare Flag B

0	[Clearing conditions] • When DTC is activated by TGIB interrupt while DISEL bit of MRB in DTC is 0 • When 0 is written to TGFB after reading TGFB = 1
1	[Setting conditions] • When TCNT = TGRB while TGRB is functioning as output compare register • When TCNT value is transferred to TGRB by input capture signal while TGRB is functioning as input capture register

Overflow Flag

0	[Clearing condition] When 0 is written to TCFV after reading TCFV = 1
1	[Setting condition] When the TCNT value overflows (changes from H'FFFF to H'0000)

Underflow Flag

0	[Clearing condition] When 0 is written to TCFU after reading TCFU = 1
1	[Setting condition] When the TCNT value underflows (changes from H'0000 to H'FFFF)

Count Direction Flag

0	TCNT counts down
1	TCNT counts up

Note: * Can only be written with 0 for flag clearing.

TCNT2—Timer Counter 2 **H'FF36** **TPU2**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Up/down-counter*

Note: * These counters can be used as up/down-counters only in phase counting mode or when counting overflow/underflow on another channel. In other cases they function as up-counters.

TGR2A—Timer General Register 2A **H'FF38** **TPU2**

TGR2B—Timer General Register 2B **H'FF3A** **TPU2**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	—	—	CKS2	CKS1	CKS0
Initial value	0	0	0	1	1	0	0	0
Read/Write	R/(W)*	R/W	R/W	—	—	R/W	R/W	R/W

Clock Select 2 to 0

CKS2	CKS1	CKS0	Clock	Overflow Period* (where ø = 20 MHz)
0	0	0	ø/2	25.6 µs
		1	ø/64	819.2 µs
	1	0	ø/128	1.6 ms
		1	ø/512	6.6 ms
1	0	0	ø/2048	26.2 ms
		1	ø/8192	104.9 ms
	1	0	ø/32768	419.4 ms
		1	ø/131072	1.68 s

Note: * An overflow period is the time interval between the start of counting up from H'00 on the TCNT and the occurrence of a TCNT overflow.

Timer Enable

0	TCNT is initialized to H'00 and halted
1	TCNT counts

Timer Mode Select

0	Interval timer mode: WDT0 requests an interval timer interrupt (WOVI) from the CPU when the TCNT overflows
1	Watchdog timer mode: A reset is issued when the TCNT overflows if the RSTE bit of RSTCSR is set to 1*

Note: * For details see section 12.2.3, Reset Control/Status Register (RSTCSR).

Overflow Flag

0	[Clearing conditions] • Write 0 in the TME bit (Only applies to WDT1) • Read TCSR* when OVF = 1, then write 0 in OVF
1	[Setting condition] When TCNT overflows (changes from H'FF to H'00) (When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset)

Note: * When interval timer interrupts are disabled and OVF is polled, read the OVF = 1 state at least twice.

Note: * Only 0 can be written, to clear the flag.

TCSR0 register differs from other registers in being more difficult to write to. For details see section 12.2.4, Notes on Register Access.

TCNT0—Timer Counter 0 **H'FF74(W), H'FF75(R)** **WDT0**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Up-counter

Note: TCNT0 register differs from other registers in being more difficult to write to.
For details see section 12.2.4, Notes on Register Access.

RSTCSR—Reset Control/Status Register **H'FF76(W), H'FF77(R)** **WDT0**

Bit	7	6	5	4	3	2	1	0
	WOVF	RSTE	RSTS	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	—	—	—	—	—

Reset Select

0	Reset
1	Do not set

Reset Enable

0	Reset signal is not generated if TCNT overflows*
1	Reset signal is generated if TCNT overflows

Note: * The modules within the H8S/2646 are not reset, but TCNT and TCSR within the WDT are reset.

Watchdog Overflow Flag

0	[Clearing condition] Cleared by reading TCSR when WOVF = 1, then writing 0 to WOVF
1	[Setting condition] Set when TCNT overflows (changed from H'FF to H'00) during watchdog timer operation

Note: * Can only be written with 0 for flag clearing.

RSTCSR register differs from other registers in being more difficult to write to.
For details see section 12.2.4, Notes on Register Access.

SMR0—Serial Mode Register 0 **H'FF78** **SCI0**
SMR1—Serial Mode Register 1 **H'FF80** **SCI1**
SMR2—Serial Mode Register 2 **H'FF88** **SCI2**

Bit	7	6	5	4	3	2	1	0
	C/$\overline{A}$	CHR	PE	O/$\overline{E}$	STOP	MP	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select 1 and 0

0	0	ø clock
	1	ø/4 clock
1	0	ø/16 clock
	1	ø/64 clock

Multiprocessor Mode

0	Multiprocessor function disabled
1	Multiprocessor format selected

Stop Bit Length

0	1 stop bit:	In transmission, a single 1 bit (stop bit) is added to the end of a transmit character before it is sent.
1	2 stop bits:	In transmission, two 1 bits (stop bits) are added to the end of a transmit character before it is sent.

Parity Mode

0	Even parity*3
1	Odd parity*4

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled*2

Character Length

0	8-bit data
1	7-bit data*1

Communication Mode

0	Asynchronous mode
1	Synchronous mode

Notes: *1 When 7-bit data is selected, the MSB (bit 7) of TDR is not transmitted, and it is not possible to choose between LSB-first or MSB-first transfer.

*2 When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

*3 When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even.

In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is even.

*4 When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd.

In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.

Bit	:	7	6	5	4	3	2	1	0
		ICE	IEIC	MST	TRS	ACKE	BBSY	IRIC	SCP
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/(W)*	R/W

Start condition/stop condition prohibit

0	Writing 0 issues a start or stop condition, in combination with the BBSY flag
1	Reading always returns a value of 1 Writing is ignored

I²C Bus interface interrupt request flag

0	Waiting for transfer, or transfer in progress
1	Interrupt requested

Note: * For details see section 15.2.5, I²C Bus Control Register.

Bus busy

0	Bus is free [Clearing condition] When a stop condition is detected
1	Bus is free [Clearing condition] When a stop condition is detected

Acknowledge bit judgement selection

0	The value of the acknowledge bit is ignored, and continuous transfer is performed
1	If the acknowledge bit is 1, continuous transfer is interrupted

Master/slave select, transmit/receive select

0	0	Slave receive mode
	1	Slave transmit mode
1	0	Master receive mode
	1	Master transmit mode

Note: * For details see section 15.2.5, I²C Bus Control Register.

I²C Bus Interface Interrupt Enable

0	Interrupts disabled
1	Interrupts enabled

I²C Bus Interface Enable

0	I²C bus interface module disabled, with SCL and SDA signal pins set to port function I²C bus interface module internal states initialized SAR and SARX can be accessed
1	I²C bus interface module enabled for transfer operations (pins SCL and SCA are driving the bus) ICMR and ICDR can be accessed

Note: * Only 0 can be written, for flag clearing.
This register is valid only on the H8S/2638 or H8S/2639 with the I²C bus interface

SMR0—Serial Mode Register 0 **H'FF78** **Smart Card Interface 0**
SMR1—Serial Mode Register 1 **H'FF80** **Smart Card Interface 1**
SMR2—Serial Mode Register 2 **H'FF88** **Smart Card Interface 2**

Bit	7	6	5	4	3	2	1	0
	GM	BLK	PE	O/$\overline{E}$	BCP1	BCP0	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select 1 and 0

0	0	ø clock
	1	ø/4 clock
1	0	ø/16 clock
	1	ø/64 clock

Basic Clock Pulse

0	0	32 clock periods
	1	64 clock periods
1	0	372 clock periods
	1	256 clock periods

Parity Mode

0	Even parity*2
1	Odd parity*3

Parity Enable

0	Parity bit addition and checking disabled
1	Parity bit addition and checking enabled*1

Block Transfer Mode

0	Normal Smart Card interface mode operation • Error signal transmission/detection and automatic data retransmission performed • TXI interrupt generated by TEND flag • TEND flag set 12.5 etu after start of transmission (11.0 etu in GSM mode)
1	Block transfer mode operation • Error signal transmission/detection and automatic data retransmission not performed • TXI interrupt generated by TDRE flag • TEND flag set 11.5 etu after start of transmission (11.0 etu in GSM mode)

GSM Mode

0	Normal smart card interface mode operation • TEND flag generation 12.5 etu (11.5 etu in block transfer mode) after beginning of start bit • Clock output ON/OFF control only
1	GSM mode smart card interface mode operation • TEND flag generation 11.0 etu after beginning of start bit • High/Low fixing control possible in addition to clock output ON/OFF control (set by SCR)

Note: etu: Elementary time unit (time for transfer of 1 bit)

Notes: When the smart card interface is used, be sure to make the 1 setting shown for bit 5.

*1 When the PE bit is set to 1, the parity (even or odd) specified by the O/$\overline{E}$ bit is added to transmit data before transmission. In reception, the parity bit is checked for the parity (even or odd) specified by the O/$\overline{E}$ bit.

*2 When even parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is even.

In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is even.

*3 When odd parity is set, parity bit addition is performed in transmission so that the total number of 1 bits in the transmit character plus the parity bit is odd.

In reception, a check is performed to see if the total number of 1 bits in the receive character plus the parity bit is odd.

BRR0—Bit Rate Register 0 **H'FF79** **SCI0, Smart Card Interface 0**

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Set the serial transmit/receive bit rate

Note: For details see section 13.2.8, Bit Rate Register (BRR).

Bit	:	7	6	5	4	3	2	1	0
		ESTP	STOP	IRTR	AASX	AL	AAS	ADZ	ACKB
Initial value	:	0	0	0	0	0	0	0	0
R/W	:	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/(W)*	R/W

Acknowledge bit

0	When receiving, 0 is output at acknowledge output timing. When transmitting, this bit shows that an acknowledge (0) has not been sent from the receiving device.
1	When receiving, 1 is output at acknowledge output timing. When transmitting, this bit shows that an acknowledge (1) has been sent from the receiving device.

General call address confirmation flag

0	General call address not confirmed [Clearing] (1) When data is written to ICDR (when sending), or when data is read from ICDR (when receiving); (2) When 0 is written after reading ADZ=1; (3) In master mode.
1	General call address confirmation [Setting] • When general call address is detected is in slave receive mode and FSX = 0 or FS = 0).

Slave address confirmation flag

0	Slave address or general call address not confirmed [Clearing] (1) When data is written to ICDR (when sending), or when data is read from ICDR (when receiving); (2) When 0 is written after reading AAS=1; (3) In master mode.
1	Slave address or general call address confirmed [Setting] • When slave address or general call address is detected in slave receive mode and FS = 0.

Arbitration lost flag

0	Secure bus. [Clearing] (1) When data is written to ICDR (when sending), or when data is read (when receiving); (2) When 0 is written after reading AL=1.
1	Bus arbitration lost [Setting] (1) When there is a mismatch between internal SDA and SDA pin at rise in SCL in master transmit mode; (2) When the internal SCL level is HIGH at the fall in SCL in master transmit mode.

2nd slave address confirmation flag

0	2nd slave address not confirmed [Clearing] (1) When 0 is written after reading AASX=1; (2) When start conditions are detected; (3) In master mode.
1	2nd slave address confirmed [Setting] • When 2nd slave address is detected in slave receive mode and FSX = 0.

I²C bus interface continuous transmit and receive interrupt request flag

0	Transmit wait state, or transmitting [Clearing] (1) When 0 written after reading IRTR=1; (2) When IRIC flag is cleared to 0.
1	Continuous transmit state [Setting] • In I²C bus interface slave mode When 1 is set in TDRE or RDRF flag when AASX=1. • In other than I²C bus interface slave mode When TDRE or RDRF flag is set to 1.

Normal end condition detection flag

0	No normal end condition [Clearing] (1) When 0 is written after reading STOP=1; (2) When IRIC flag is cleared to 0.
1	Normal end condition detected in slave mode in I²C bus format [Setting] On detection of stop condition on completion of sending frame. • No meaning when in other than slave mode in I²C bus format

Error stop condition detection flag

0	No error stop condition [Clearing] (1) When 0 written after reading ESTP=1; (2) When IRIC flag is cleared to 0.
1	• Error stop condition detected in slave mode in I²C bus format [Setting] On detection of stop condition while sending frame. • No meaning when in other than slave mode in I²C bus format

Note: * Only 0 can be written to these bits (to clear these flags).
This register is valid only on the H8S/2638 or H8S/2639 with the I²C bus interface option added.

SCR0—Serial Control Register 0 **H'FF7A** **SCI0**
SCR1—Serial Control Register 1 **H'FF82** **SCI1**
SCR2—Serial Control Register 2 **H'FF8A** **SCI2**

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable 1 and 0

0	0	Asynchronous mode	Internal clock/SCK pin functions as I/O port
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
	1	Asynchronous mode	Internal clock/SCK pin functions as clock output*9
		Clocked synchronous mode	Internal clock/SCK pin functions as serial clock output
1	0	Asynchronous mode	External clock/SCK pin functions as clock input*10
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input
	1	Asynchronous mode	External clock/SCK pin functions as clock input*10
		Clocked synchronous mode	External clock/SCK pin functions as serial clock input

Transmit End Interrupt Enable

0	Transmit-end interrupt (TEI) request disabled*8
1	Transmit-end interrupt (TEI) request enabled*8

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception mode) [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled*7 Receive interrupt (RXI) requests, receive-error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received

Transmit Interrupt Enable

0	Transmit-data-empty interrupt (TXI) request disabled*1
1	Transmit-data-empty interrupt (TXI) request enabled

Receive Interrupt Enable

0	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request disabled*2
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

Receive Enable

0	Reception disabled*5
1	Reception enabled*6

Transmit Enable

0	Transmission disabled*3
1	Transmission enabled*4

Notes: *1 TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.

*2 RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or clearing the RIE bit to 0.

*3 The TDRE flag in SSR is fixed at 1.

*4 In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0.

SMR setting must be performed to decide the transfer format before setting the TE bit to 1.

*5 Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.

*6 Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode.

SMR setting must be performed to decide the transfer format before setting the RE bit to 1.

*7 When receive data including MPB = 0 is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and ORER flags in SSR , is not performed. When receive data including MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting is enabled.

*8 TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0.

*9 Outputs a clock of the same frequency as the bit rate.

*10 Inputs a clock with a frequency 16 times the bit rate.

SCR0—Serial Control Register 0 **H'FF7A** **Smart Card Interface 0**
SCR1—Serial Control Register 1 **H'FF82** **Smart Card Interface 1**
SCR2—Serial Control Register 2 **H'FF8A** **Smart Card Interface 2**

Bit	7	6	5	4	3	2	1	0
	TIE	RIE	TE	RE	MPIE	TEIE	CKE1	CKE0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Enable 1 and 0

SCMR	SMR	SCR Setting		SCK Pin Function
SMIF	C/$\overline{A}$, GM	CKE1	CKE0	
0	See the SCI			
1	0	0	0	Operates as port I/O pin
			1	Outputs clock as SCK output pin
	1		0	Operates as SCK output pin, with output fixed low
			1	Outputs clock as SCK output pin
		1	1	Operates as SCK output pin, with output fixed high
			0	Outputs clock as SCK output pin

Transmit End Interrupt Enable

0	Transmit-end interrupt (TEI) request disabled*8
1	Transmit-end interrupt (TEI) request enabled*8

Multiprocessor Interrupt Enable

0	Multiprocessor interrupts disabled (normal reception mode) [Clearing conditions] • When the MPIE bit is cleared to 0 • When data with MPB = 1 is received
1	Multiprocessor interrupts enabled*7 Receive interrupt (RXI) requests, receive-error interrupt (ERI) requests, and setting of the RDRF, FER, and ORER flags in SSR are disabled until data with the multiprocessor bit set to 1 is received

Transmit Interrupt Enable

0	Transmit-data-empty interrupt (TXI) request disabled*1
1	Transmit-data-empty interrupt (TXI) request enabled

Receive Interrupt Enable

0	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request disabled*2
1	Receive-data-full interrupt (RXI) request and receive-error interrupt (ERI) request enabled

Receive Enable

0	Reception disabled*5
1	Reception enabled*6

Transmit Enable

0	Transmission disabled*3
1	Transmission enabled*4

Notes: *1 TXI interrupt request cancellation can be performed by reading 1 from the TDRE flag, then clearing it to 0, or clearing the TIE bit to 0.

*2 RXI and ERI interrupt request cancellation can be performed by reading 1 from the RDRF flag, or the FER, PER, or ORER flag, then clearing the flag to 0, or clearing the RIE bit to 0.

*3 The TDRE flag in SSR is fixed at 1.

*4 In this state, serial transmission is started when transmit data is written to TDR and the TDRE flag in SSR is cleared to 0.

SMR setting must be performed to decide the transfer format before setting the TE bit to 1.

*5 Clearing the RE bit to 0 does not affect the RDRF, FER, PER, and ORER flags, which retain their states.

*6 Serial reception is started in this state when a start bit is detected in asynchronous mode or serial clock input is detected in clocked synchronous mode.

SMR setting must be performed to decide the transfer format before setting the RE bit to 1.

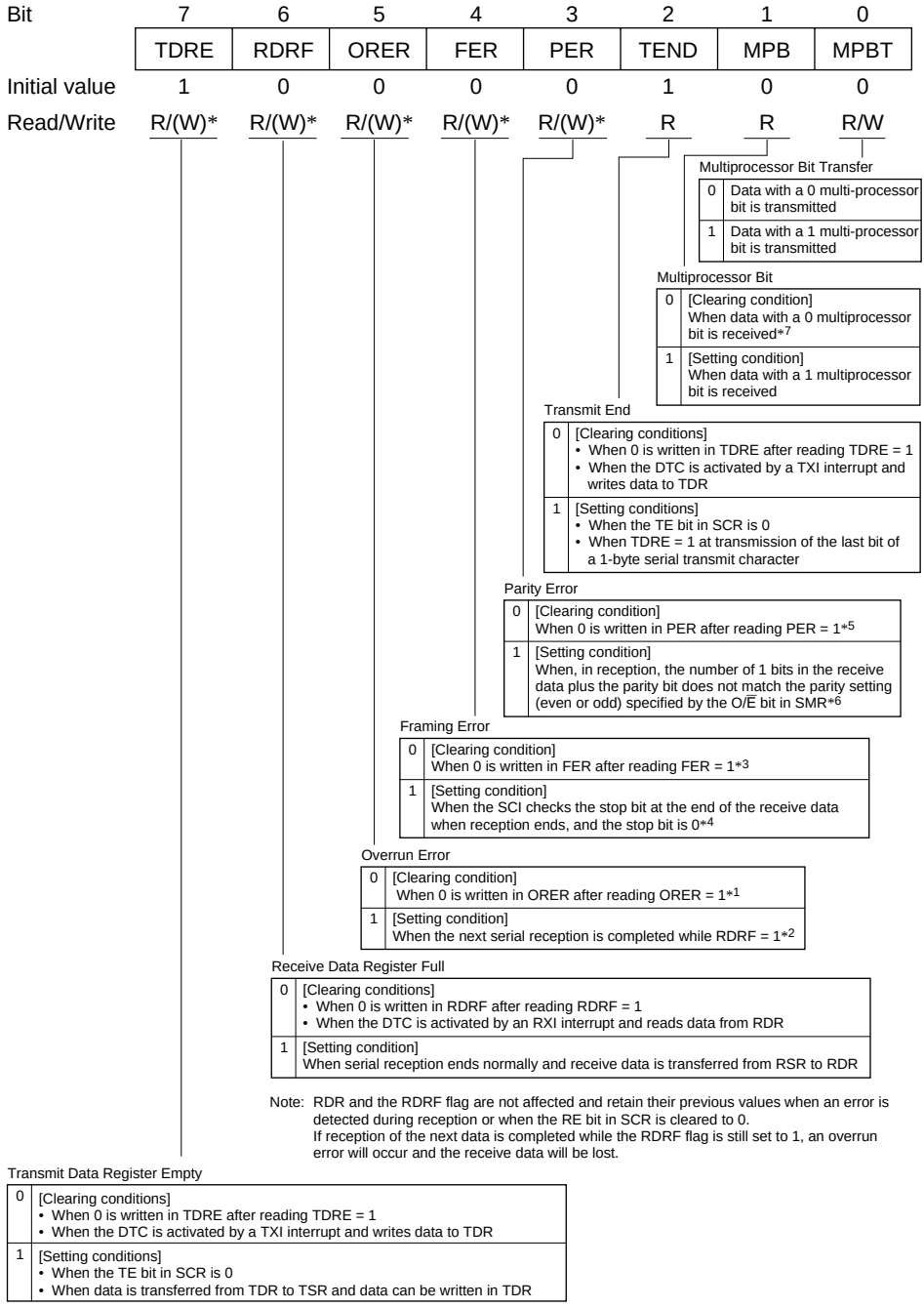
*7 When receive data including MPB = 0 is received, receive data transfer from RSR to RDR, receive error detection, and setting of the RDRF, FER, and ORER flags in SSR , is not performed. When receive data including MPB = 1 is received, the MPB bit in SSR is set to 1, the MPIE bit is cleared to 0 automatically, and generation of RXI and ERI interrupts (when the TIE and RIE bits in SCR are set to 1) and FER and ORER flag setting is enabled.

*8 TEI cancellation can be performed by reading 1 from the TDRE flag in SSR, then clearing it to 0 and clearing the TEND flag to 0, or clearing the TEIE bit to 0.

TDR0—Transmit Data Register 0 **H'FF7B** **SCI0, Smart Card Interface 0**

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Store serial transmit data


Bit 7 6 5 4 3 2 1 0
TDRE RDRF ORER FER PER TEND MPB MPBT
Initial value 1 0 0 0 0 1 0 0
Read/Write R/(W)* R/(W)* R/(W)* R/(W)* R/(W)* R R R/W
Multiprocessor Bit Transfer
0 Data with a 0 multi-processor bit is transmitted
1 Data with a 1 multi-processor bit is transmitted
Multiprocessor Bit
0 [Clearing condition] When data with a 0 multiprocessor bit is received*7
1 [Setting condition] When data with a 1 multiprocessor bit is received
Transmit End
0 [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1 [Setting conditions] • When the TE bit in SCR is 0 • When TDRE = 1 at transmission of the last bit of a 1-byte serial transmit character
Parity Error
0 [Clearing condition] When 0 is written in PER after reading PER = 1*5
1 [Setting condition] When, in reception, the number of 1 bits in the receive data plus the parity bit does not match the parity setting (even or odd) specified by the O/E bit in SMR*6
Framing Error
0 [Clearing condition] When 0 is written in FER after reading FER = 1*3
1 [Setting condition] When the SCI checks the stop bit at the end of the receive data when reception ends, and the stop bit is 0*4
Overrun Error
0 [Clearing condition] When 0 is written in ORER after reading ORER = 1*1
1 [Setting condition] When the next serial reception is completed while RDRF = 1*2
Receive Data Register Full
0 [Clearing conditions] • When 0 is written in RDRF after reading RDRF = 1 • When the DTC is activated by an RXI interrupt and reads data from RDR
1 [Setting condition] When serial reception ends normally and receive data is transferred from RSR to RDR
Note: RDR and the RDRF flag are not affected and retain their previous values when an error is detected during reception or when the RE bit in SCR is cleared to 0. If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.
Transmit Data Register Empty
0 [Clearing conditions] • When 0 is written in TDRE after reading TDRE = 1 • When the DTC is activated by a TXI interrupt and writes data to TDR
1 [Setting conditions] • When the TE bit in SCR is 0 • When data is transferred from TDR to TSR and data can be written in TDR

Notes: * Only 0 can be written, to clear the flag.

*1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*2 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

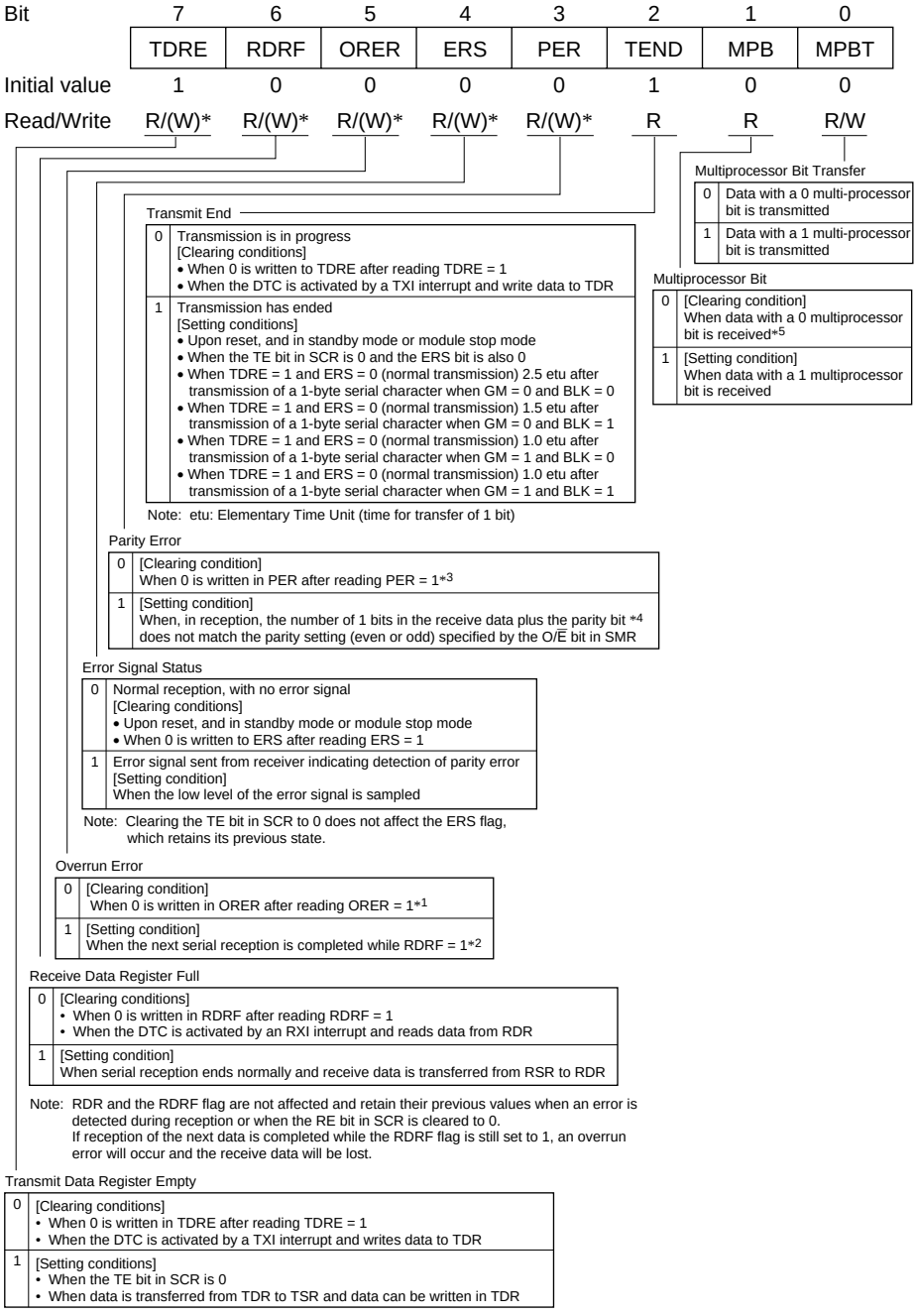
*3 The FER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*4 In 2-stop-bit mode, only the first stop bit is checked for a value of 0; the second stop bit is not checked. If a framing error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the FER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

*5 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*6 If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

*7 Retains its previous state when the RE bit in SCR is cleared to 0 with multiprocessor format.


Bit 7 6 5 4 3 2 1 0
TDRE RDRF ORER ERS PER TEND MPB MPBT
Initial value 1 0 0 0 0 1 0 0
Read/Write R/(W)* R/(W)* R/(W)* R/(W)* R/(W)* R R R/W
Multiprocessor Bit Transfer
0 Data with a 0 multi-processor bit is transmitted
1 Data with a 1 multi-processor bit is transmitted
Transmit End
0 Transmission is in progress
[Clearing conditions]
• When 0 is written to TDRE after reading TDRE = 1
• When the DTC is activated by a TXI interrupt and write data to TDR
1 Transmission has ended
[Setting conditions]
• Upon reset, and in standby mode or module stop mode
• When the TE bit in SCR is 0 and the ERS bit is also 0
• When TDRE = 1 and ERS = 0 (normal transmission) 2.5 etu after transmission of a 1-byte serial character when GM = 0 and BLK = 0
• When TDRE = 1 and ERS = 0 (normal transmission) 1.5 etu after transmission of a 1-byte serial character when GM = 0 and BLK = 1
• When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after transmission of a 1-byte serial character when GM = 1 and BLK = 0
• When TDRE = 1 and ERS = 0 (normal transmission) 1.0 etu after transmission of a 1-byte serial character when GM = 1 and BLK = 1
Note: etu: Elementary Time Unit (time for transfer of 1 bit)
Multiprocessor Bit
0 [Clearing condition]
When data with a 0 multiprocessor bit is received*5
1 [Setting condition]
When data with a 1 multiprocessor bit is received
Parity Error
0 [Clearing condition]
When 0 is written in PER after reading PER = 1*3
1 [Setting condition]
When, in reception, the number of 1 bits in the receive data plus the parity bit *4 does not match the parity setting (even or odd) specified by the O/E bit in SMR
Error Signal Status
0 Normal reception, with no error signal
[Clearing conditions]
• Upon reset, and in standby mode or module stop mode
• When 0 is written to ERS after reading ERS = 1
1 Error signal sent from receiver indicating detection of parity error
[Setting condition]
When the low level of the error signal is sampled
Note: Clearing the TE bit in SCR to 0 does not affect the ERS flag, which retains its previous state.
Overrun Error
0 [Clearing condition]
When 0 is written in ORER after reading ORER = 1*1
1 [Setting condition]
When the next serial reception is completed while RDRF = 1*2
Receive Data Register Full
0 [Clearing conditions]
• When 0 is written in RDRF after reading RDRF = 1
• When the DTC is activated by an RXI interrupt and reads data from RDR
1 [Setting condition]
When serial reception ends normally and receive data is transferred from RSR to RDR
Note: RDR and the RDRF flag are not affected and retain their previous values when an error is detected during reception or when the RE bit in SCR is cleared to 0.
If reception of the next data is completed while the RDRF flag is still set to 1, an overrun error will occur and the receive data will be lost.
Transmit Data Register Empty
0 [Clearing conditions]
• When 0 is written in TDRE after reading TDRE = 1
• When the DTC is activated by a TXI interrupt and writes data to TDR
1 [Setting conditions]
• When the TE bit in SCR is 0
• When data is transferred from TDR to TSR and data can be written in TDR

Notes: * Only 0 can be written, to clear the flag.

*1 The ORER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*2 The receive data prior to the overrun error is retained in RDR, and the data received subsequently is lost. Also, subsequent serial reception cannot be continued while the ORER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

*3 The PER flag is not affected and retains its previous state when the RE bit in SCR is cleared to 0.

*4 If a parity error occurs, the receive data is transferred to RDR but the RDRF flag is not set. Also, subsequent serial reception cannot be continued while the PER flag is set to 1. In clocked synchronous mode, serial transmission cannot be continued, either.

*5 Retains its previous state when the RE bit in SCR is cleared to 0 with multiprocessor format.

RDR0—Receive Data Register 0 **H'FF7D** **SCI0, Smart Card Interface 0**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Store serial receive data

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	SMIF
Initial value	1	1	1	1	0	0	1	0
Read/Write	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Operates as normal SCI (smart card interface function disabled)
1	Smart card interface function enabled

Smart Card Interface Data Invert

0	TDR contents are transmitted without modification Receive data is stored in RDR without modification
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Smart Card Interface Data Transfer Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

Bit	:	7	6	5	4	3	2	1	0
		ICDR7	ICDR6	ICDR5	ICDR4	ICDR3	ICDR2	ICDR1	ICDR0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

ICDRR

Bit	:	7	6	5	4	3	2	1	0
		ICDRR7	ICDRR6	ICDRR5	ICDRR4	ICDRR3	ICDRR2	ICDRR1	ICDRR0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	R	R	R	R	R	R	R	R

ICDRS

Bit	:	7	6	5	4	3	2	1	0
		ICDRS7	ICDRS6	ICDRS5	ICDRS4	ICDRS3	ICDRS2	ICDRS1	ICDRS0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	—	—	—	—	—	—	—	—

ICDRT

Bit	:	7	6	5	4	3	2	1	0
		ICDRT7	ICDRT6	ICDRT5	ICDRT4	ICDRT3	ICDRT2	ICDRT1	ICDRT0
Initial value	:	—	—	—	—	—	—	—	—
R/W	:	W	W	W	W	W	W	W	W

TDRE, RDRF (internal flag)

Bit	:							—	—
								TDRE	RDRF
Initial value	:							0	0
R/W	:							—	—

Note: This register is valid only on the H8S/2638 or H8S/2639 with the I²C bus interface option added.

SARX0—2nd Slave Address Register **H'FF7E** **IIC0**
SARX1—2nd Slave Address Register **H'FF86** **IIC1**

Bit	:	7	6	5	4	3	2	1	0
		SVAX6	SVAX5	SVAX4	SVAX3	SVAX2	SVAX1	SVAX0	FSX
Initial value	:	0	0	0	0	0	0	0	1
R/W	:	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

2nd slave address

Format select X

Note: This register is valid only on the H8S/2638 or H8S/2639 with the I^2C bus interface option added.

Bit	7	6	5	4	3	2	1	0
	MLS	WAIT	CKS2	CKS1	CKS0	BC2	BC1	BC0
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Bit counter

Bit 2	Bit 1	Bit 0	Bit/frame	
BC2	BC1	BC0	Clock sync serial format	PC bus format
0	0	0	8	9
		1	1	2
	1	0	2	3
		1	3	4
1	0	0	4	5
		1	5	6
	1	0	6	7
		1	7	8

Transmit clock select

SCRX bit 5, 6	Bit 5	Bit 4	Bit 3	Clock	Transfer rate				
IICX	CKS2	CKS1	CKS0		ø= 5 MHz	ø= 8 MHz	ø= 10 MHz	ø= 16 MHz	ø= 20 MHz
0	0	0	0	ø/28	179kHz	286 kHz	357 kHz	571 kHz*	714 kHz*
			1	ø/40	125kHz	200 kHz	250 kHz	400 kHz	500 kHz*
		1	0	ø/48	104kHz	167 kHz	208 kHz	333 kHz	417 kHz*
			1	ø/64	78.1kHz	125 kHz	156 kHz	250 kHz	313 kHz
	1	0	0	ø/80	62.5kHz	100 kHz	125 kHz	200 kHz	250 kHz
			1	ø/100	50.0kHz	80.0 kHz	100 kHz	160 kHz	200 kHz
		1	0	ø/112	44.6kHz	71.4 kHz	89.3 kHz	143 kHz	179 kHz
			1	ø/128	39.1kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
1	0	0	0	ø/56	89.3kHz	143 kHz	179 kHz	286 kHz	357 kHz
			1	ø/80	62.5kHz	100 kHz	125 kHz	200 kHz	250 kHz
		1	0	ø/96	52.1kHz	83.3 kHz	104 kHz	167 kHz	208 kHz
			1	ø/128	39.1kHz	62.5 kHz	78.1 kHz	125 kHz	156 kHz
	1	0	0	ø/160	31.3kHz	50.0 kHz	62.5 kHz	100 kHz	125 kHz
			1	ø/200	25.0kHz	40.0 kHz	50.0 kHz	80.0 kHz	100 kHz
		1	0	ø/224	22.3kHz	35.7 kHz	44.6 kHz	71.4 kHz	89.3 kHz
			1	ø/256	19.5kHz	31.3 kHz	39.1 kHz	62.5 kHz	78.1 kHz

Note: * These rates are outside the ranges stipulated in the I²C bus interface specifications (normal mode: max. 100 kHz, high-speed mode: max. 400 kHz).

Wait insert bit

0	Send data followed by acknowledge bit.
1	Insert wait between data and acknowledge bit.

MSB-first/LSB-first select

0	MSB first
1	LSB first

Note: This register is valid only on the H8S/2638 or H8S/2639 with the I²C bus interface option added.

Bit	7	6	5	4	3	2	1	0
	SVA6	SVA5	SVA4	SVA3	SVA2	SVA1	SVA0	FS
Initial value	0	0	0	0	0	0	0	0
R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Slave address

Format select

DDCSWR bit 6	SAR bit 0	SARX bit 0	Operating mode
SW	FS	FSX	
0	0	0	I²C bus format • SAR and SARX slave addresses recognized
		1	I²C bus format (initial value) • SAR slave address recognized • SARX slave address ignored
	1	0	I²C bus format • SAR slave address ignored • SARX slave address recognized
		1	Synchronous serial format • SAR and SARX slave addresses ignored
1	—	—	• Must not be set.

BRR1—Bit Rate Register 1 **H'FF81** **SCI1, Smart Card Interface 1**

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Set the serial transmit/receive bit rate

Note: For details see section 13.2.8, Bit Rate Register (BRR).

TDR1—Transmit Data Register 1 **H'FF83** **SCI1, Smart Card Interface 1**

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Store serial transmit data

RDR1—Receive Data Register 1 **H'FF85** **SCI1, Smart Card Interface 1**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Store serial receive data

SCMR1—Smart Card Mode Register 1 H'FF86 SCI1, Smart Card Interface 1

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	SMIF
Initial value	1	1	1	1	0	0	1	0
Read/Write	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Operates as normal SCI (smart card interface function disabled)
1	Smart card interface function enabled

Smart Card Interface Data Invert

0	TDR contents are transmitted without modification Receive data is stored in RDR without modification
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Smart Card Data Interface Transfer Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

BRR2—Bit Rate Register 2 H'FF89 SCI2, Smart Card Interface 2

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Set the serial transmit/receive bit rate

Note: For details see section 13.2.8, Bit Rate Register (BRR).

TDR2—Transmit Data Register 2 **H'FF8B** **SCI2, Smart Card Interface 2**

Bit	7	6	5	4	3	2	1	0
Initial value	1	1	1	1	1	1	1	1
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Store serial transmit data

RDR2—Receive Data Register 2 **H'FF8D** **SCI2, Smart Card Interface 2**

Bit	7	6	5	4	3	2	1	0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R

Store serial receive data

SCMR2—Smart Card Mode Register 2 **H'FF8E** **SCI2, Smart Card Interface 2**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	SDIR	SINV	—	SMIF
Initial value	1	1	1	1	0	0	1	0
Read/Write	—	—	—	—	R/W	R/W	—	R/W

Smart Card Interface Mode Select

0	Operates as normal SCI (smart card interface function disabled)
1	Smart card interface function enabled

Smart Card Interface Data Invert

0	TDR contents are transmitted without modification Receive data is stored in RDR without modification
1	TDR contents are inverted before being transmitted Receive data is stored in RDR in inverted form

Smart Card Interface Data Transfer Direction

0	TDR contents are transmitted LSB-first Receive data is stored in RDR LSB-first
1	TDR contents are transmitted MSB-first Receive data is stored in RDR MSB-first

ADDRA—A/D Data Register A **H'FF90** **A/D Converter**
ADDRB—A/D Data Register B **H'FF92** **A/D Converter**
ADDRC—A/D Data Register C **H'FF94** **A/D Converter**
ADDRD—A/D Data Register D **H'FF96** **A/D Converter**

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	AD9	AD8	AD7	AD6	AD5	AD4	AD3	AD2	AD1	AD0	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Read/Write	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R	R

Bit	7	6	5	4	3	2	1	0
	ADF	ADIE	ADST	SCAN	CH3	CH2	CH1	CH0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Channel Select 2 to 0

CH3	CH2	CH1	CH0	Single Mode (SCAN = 0)	Scan Mode (SCAN = 1)
0	0	0	0	AN0	AN0
			1	AN1	AN0, AN1
		1	0	AN2	AN0 to AN2
			1	AN3	AN0 to AN3
	1	0	0	AN4	AN4
			1	AN5	AN4, AN5
		1	0	AN6	AN4 to AN6
			1	AN7	AN4 to AN7
1	0	0	0	AN8	AN8
			1	AN9	AN8, AN9
		1	0	AN10	AN8 to AN10
			1	AN11	AN8 to AN11
	1	0	0	Setting prohibited	Setting prohibited
			1	Setting prohibited	Setting prohibited
		1	0	Setting prohibited	Setting prohibited
			1	Setting prohibited	Setting prohibited

Channel Select 3

0	AN8 to AN11 are group 0 analog input pins
1	AN0 to AN3 are group 0 analog input pins, AN4 to AN7 are group 1 analog input pins

Scan Mode

0	Single mode
1	Scan mode

A/D Start

0	A/D conversion stopped
1	• Single mode: A/D conversion is started. Cleared to 0 automatically when conversion on the specified channel ends • Scan mode: A/D conversion is started. Conversion continues sequentially on the selected channels until ADST is cleared to 0 by software, a reset, or a transition to standby mode or module stop mode

A/D Interrupt Enable

0	A/D conversion end interrupt (ADI) request disabled
1	A/D conversion end interrupt (ADI) request enabled

A/D End Flag

0	[Clearing conditions] • When 0 is written in the to ADF flag after reading ADF = 1 • When the DTC is activated by an ADI interrupt, and ADDR is read
1	[Setting conditions] • Single mode: When A/D conversion ends • Scan mode: When A/D conversion ends on all specified channels

Note: * Only 0 can be written, to clear the flag.

Bit	7	6	5	4	3	2	1	0
	TRGS1	TRGS0	—	—	CKS1	CKS0	—	—
Initial value	0	0	1	1	0	0	1	1
Read/Write	R/W	R/W	—	—	R/W	R/W	—	—

Clock Select

0	0	Conversion time = 530 states (max.)
	1	Conversion time = 266 states (max.)
1	0	Conversion time = 134 states (max.)
	1	Conversion time = 68 states (max.)

Timer Trigger Select

0	0	A/D conversion start by software is enabled
	1	A/D conversion start by TPU conversion start trigger is enabled
1	0	Setting prohibited
	1	A/D conversion start by external trigger pin ($\overline{\text{ADTRG}}$) is enabled

Bit	7	6	5	4	3	2	1	0
	OVF	WT/$\overline{IT}$	TME	PSS*2	RST/$\overline{NMI}$	CKS2	CKS1	CKS0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/(W)*1	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Clock Select 2 to 0

PSS	CKS2	CKS1	CKS0	Clock	Overflow Period*1 (where ø = 20 MHz) (where ø SUB*2 = 32.768 kHz)
0	0	0	0	ø/2	25.6 µs
			1	ø/64	819.2 µs
		1	0	ø/128	1.6 ms
			1	ø/512	6.6 ms
	1	0	0	ø/2048	26.2 ms
			1	ø/8192	104.9 ms
		1	0	ø/32768	419.4 ms
0	1	1	1	ø/131072	1.68 s
1	0	0	0	øSUB/2*2	15.6 ms
			1	øSUB/4*2	31.3 ms
		1	0	øSUB/8*2	62.5 ms
			1	øSUB/16*2	125 ms
	1	0	0	øSUB/32*2	250 ms
			1	øSUB/64*2	500 ms
		1	0	øSUB/128*2	1 s
			1	øSUB/256*2	2 s

Notes: *1 An overflow period is the time interval between the start of counting up from H'00 on the TCNT and the occurrence of a TCNT overflow.
*2 Subclock functions (subactive mode, subsleep mode, and watch mode) are not available in the HD64F2636F, but are available in the HD64F2636UF (U-mask version).

Reset or NMI

0	NMI request
1	Internal reset request

Prescaler Select

0	The TCNT counts frequency-division clock pulses of the ø based prescaler (PSM)
1	The TCNT counts frequency-division clock pulses of the ø SUB*-based prescaler (PSS)

Note: * Subclock functions (subactive mode, subsleep mode, and watch mode) are not available in the HD64F2636F, but are available in the HD64F2636UF (U-mask version).

Timer Enable

0	TCNT is initialized to H'00 and halted
1	TCNT counts

Timer Mode Select

0	Interval timer mode: WDT1 requests an interval timer interrupt (WOVI) from the CPU when the TCNT overflows
1	Watchdog timer mode: WDT1 requests a reset or an NMI interrupt from the CPU when the TCNT overflows

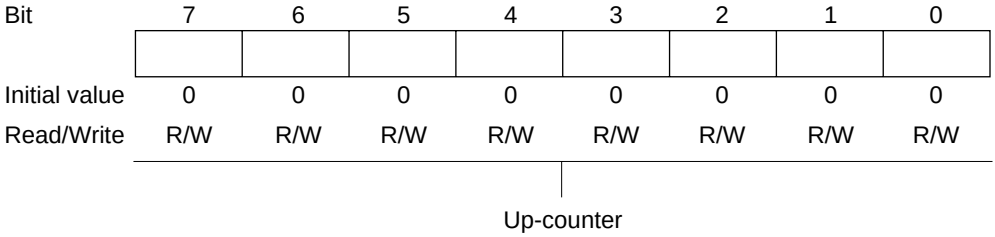
Overflow Flag

0	[Clearing conditions] • Write 0 in the TME bit (Only applies to WDT1) • Read TCSR* when OVF = 1, then write 0 in OVF
1	[Setting condition] When TCNT overflows (changes from H'FF to H'00) (When internal reset request generation is selected in watchdog timer mode, OVF is cleared automatically by the internal reset)

Note: * When interval timer interrupts are disabled and OVF is polled, read the OVF = 1 state at least twice.

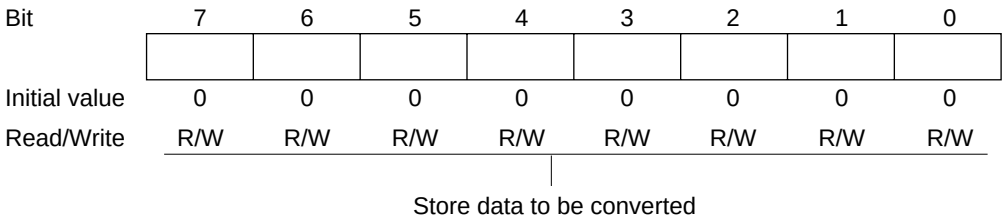
Notes: *1 Only 0 can be written, to clear the flag.
*2 Subclock functions (subactive mode, subsleep mode, and watch mode) are not available in the HD64F2636F and HD64F2638F, but are available in the HD64F2636UF and HD64F2638UF (U-mask version).

TCSR1 register differs from other registers in being more difficult to write to. For details see section 12.2.4, Notes on Register Access.



Note: TCNT1 register differs from other registers in being more difficult to write to.
For details see section 12.2.4, Notes on Register Access.

DADR0— D/A Data Register 0 **H'FFA4** **D/A0, 1**
DADR1— D/A Data Register 1 **H'FFA5** **D/A0, 1**



Bit	7	6	5	4	3	2	1	0
	DAOE1	DAOE0	DAE	—	—	—	—	—
Initial value	0	0	0	1	1	1	1	1
Read/Write	R/(W)*	R/W	R/W	—	—	—	—	—

D/A Enable

0	0	*	Disabled on channels 0 and 1
	1	0	Enabled on channel 0 Disabled on channel 1
		1	Enabled on channels 0 and 1
1	0	0	Disabled on channel 0Enabled on channel 1
		1	Enabled on channels 0 and 1
	1	*	Enabled on channels 0 and 1

*: Don't care

D/A Output Enable 0

0	Analog output DA0 is disabled
1	D/A conversion is enabled on channel 0. Analog output DA0 is enabled

D/A Output Enable 1

0	Analog output DA1 is disabled
1	D/A conversion is enabled on channel 1. Analog output DA1 is enabled

Bit	7	6	5	4	3	2	1	0
	FWE	SWE	ESU	PSU	EV	PV	E	P
Initial value	—*	0	0	0	0	0	0	0
Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Erase

0	Erase mode cleared
1	Transition to erase mode [Setting condition] When FWE = 1, SWE = 1, and ESU = 1

Program

0	Program mode cleared
1	Transition to program mode [Setting condition] When FWE = 1, SWE = 1, and PSU = 1

Program-Verify

0	Program-verify mode cleared
1	Transition to program-verify mode [Setting condition] When FWE = 1 and SWE = 1

Erase-verify

0	Erase-verify mode cleared
1	Transition to erase-verify mode [Setting condition] When FWE = 1 and SWE = 1

Program Setup

0	Program setup cleared
1	Program setup [Setting condition] When FWE = 1 and SWE = 1

Erase Setup

0	Erase setup cleared
1	Erase setup [Setting condition] When FWE = 1 and SWE = 1

Software Write Enable Bit

0	Writes disabled
1	Writes enabled [Setting condition] When FWE = 1

Flash Write Enable Bit

0	When a low level is input to the FWE pin (hardware-protected state)
1	When a high level is input to the FWE pin

Note: * Determined by the state of the FWE pin.

Bit	7	6	5	4	3	2	1	0
	FLER	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R	—	—	—	—	—	—	—

Flash Memory Error

0	Flash memory is operating normally Flash memory program/erase protection (error protection) is disabled [Clearing condition] Power-on reset or hardware standby mode
1	An error has occurred during flash memory programming/erasing Flash memory program/erase protection (error protection) is enabled [Setting condition] See section 21A.8.3, 21B.8.3, Error Protection

EBR1—Erase Block Register 1 **H'FFAA** **Flash Memory**
EBR2—Erase Block Register 2 **H'FFAB** **Flash Memory**

EBR1

Bit	15	14	13	12	11	10	9	8
	EB7	EB6	EB5	EB4	EB3	EB2	EB1	EB0
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

EBR2

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	EB11*	EB10*	EB9	EB8
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Specify the flash memory erase area

• H8S/2636

Block (Size)	Addresses
EB0 (1 kB)	H'000000–H'0003FF
EB1 (1 kB)	H'000400–H'0007FF
EB2 (1 kB)	H'000800–H'000BFF
EB3 (1 kB)	H'000C00–H'000FFF
EB4 (28 kB)	H'001000–H'007FFF
EB5 (16 kB)	H'008000–H'00BFFF
EB6 (8 kB)	H'00C000–H'00DFFF
EB7 (8 kB)	H'00E000–H'00FFFF
EB8 (32 kB)	H'010000–H'017FFF
EB9 (32 kB)	H'018000–H'01FFFF

• H8S/2638

Block (Size)	Addresses
EB0 (4 kB)	H'000000–H'000FFF
EB1 (4 kB)	H'001000–H'001FFF
EB2 (4 kB)	H'002000–H'002FFF
EB3 (4 kB)	H'003000–H'003FFF
EB4 (4 kB)	H'004000–H'004FFF
EB5 (4 kB)	H'005000–H'005FFF
EB6 (4 kB)	H'006000–H'006FFF
EB7 (4 kB)	H'007000–H'007FFF
EB8 (32 kB)	H'008000–H'00FFFF
EB9 (64 kB)	H'010000–H'01FFFF
EB10 (64 kB)	H'020000–H'02FFFF
EB11 (64 kB)	H'030000–H'03FFFF

Note: * These bits are only for the H8S/2638. On the H8S/2636, these bits are reserved.

FLPWCR—Flash Memory Power Control Register H'FFAC **Flash Memory**

Bit	7	6	5	4	3	2	1	0
	PDWND	—	—	—	—	—	—	—
Initial value	0	0	0	0	0	0	0	0
Read/Write	R/W	R	R	R	R	R	R	R

Power-Down Disable

0	Transition to flash memory power-down mode enabled
1	Transition to flash memory power-down mode disabled

PORT1—Port 1 Register **H'FFB0** **Port**

Bit	7	6	5	4	3	2	1	0
	P17	P16	P15	P14	P13	P12	P11	P10
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

State of the port 1 pins

Note: * Determined by state of pins P17 to P10.

PORT3—Port 3 Register **H'FFB2** **Port**

Bit	7	6	5	4	3	2	1	0
	—	—	P35	P34	P33	P32	P31	P30
Initial value	Undefined	Undefined	—*	—*	—*	—*	—*	—*
Read/Write	—	—	R	R	R	R	R	R

State of the port 3 pins

Note: * Determined by state of pins P35 to P30.

PORT4—Port 4 Register **H'FFB3** **Port**

Bit	7	6	5	4	3	2	1	0
	P47	P46	P45	P44	P43	P42	P41	P40
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

State of the port 4 pins

Note: * Determined by state of pins P47 to P40.

PORT9—Port 9 Register **H'FFB8** **Port**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	P93	P92	P91	P90
Initial value	Undefined	Undefined	Undefined	Undefined	—*	—*	—*	—*
Read/Write	—	—	—	—	R	R	R	R

State of the port 9 pins

Note: * Determined by state of pins P93 to P90.

PORTA—Port A Register **H'FFB9** **Port**

Bit	7	6	5	4	3	2	1	0
	—	—	—	—	PA3	PA2	PA1	PA0
Initial value	Undefined	Undefined	Undefined	Undefined	—*	—*	—*	—*
Read/Write	—	—	—	—	R	R	R	R

State of the port A pins

Note: * Determined by state of pins PA3 to PA0.

PORTB—Port B Register **H'FFBA** **Port**

Bit	7	6	5	4	3	2	1	0
	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

State of the port B pins

Note: * Determined by state of pins PB7 to PB0.

PORTC—Port C Register **H'FFBB** **Port**

Bit	7	6	5	4	3	2	1	0
	PC7	PC6	PC5	PC4	PC3	PC2	PC1	PC0
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

State of the port C pins

Note: * Determined by state of pins PC7 to PC0.

PORTD—Port D Register **H'FFBC** **Port**

Bit	7	6	5	4	3	2	1	0
	PD7	PD6	PD5	PD4	PD3	PD2	PD1	PD0
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

State of the port D pins

Note: * Determined by state of pins PD7 to PD0.

PORTE—Port E Register **H'FFBD** **Port**

Bit	7	6	5	4	3	2	1	0
	PE7	PE6	PE5	PE4	PE3	PE2	PE1	PE0
Initial value	—*	—*	—*	—*	—*	—*	—*	—*
Read/Write	R	R	R	R	R	R	R	R

State of the port E pins

Note: * Determined by state of pins PE7 to PE0.

Bit	7	6	5	4	3	2	1	0
	PF7	PF6	PF5	PF4	PF3	—	—	PF0
Initial value	—*	—*	—*	—*	—*	Undefined	Undefined	—*
Read/Write	R	R	R	R	R	—	—	R

State of the port F pins

Note: * Determined by state of pins PF7 to PF3, PF0.

C.1 Port 1 Block Diagrams

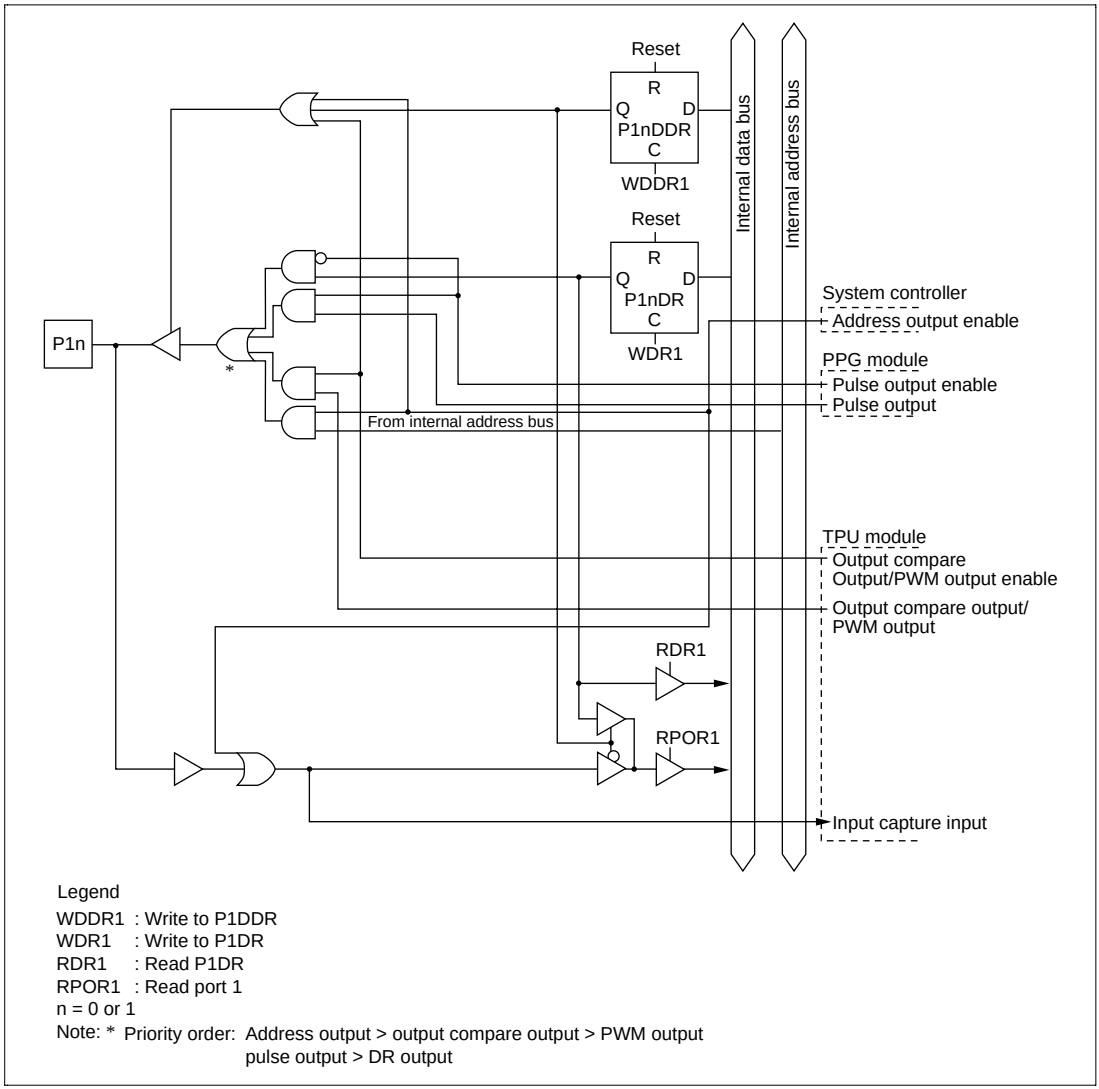


Figure C-1 (a) Port 1 Block Diagram (Pins P10 and P11)

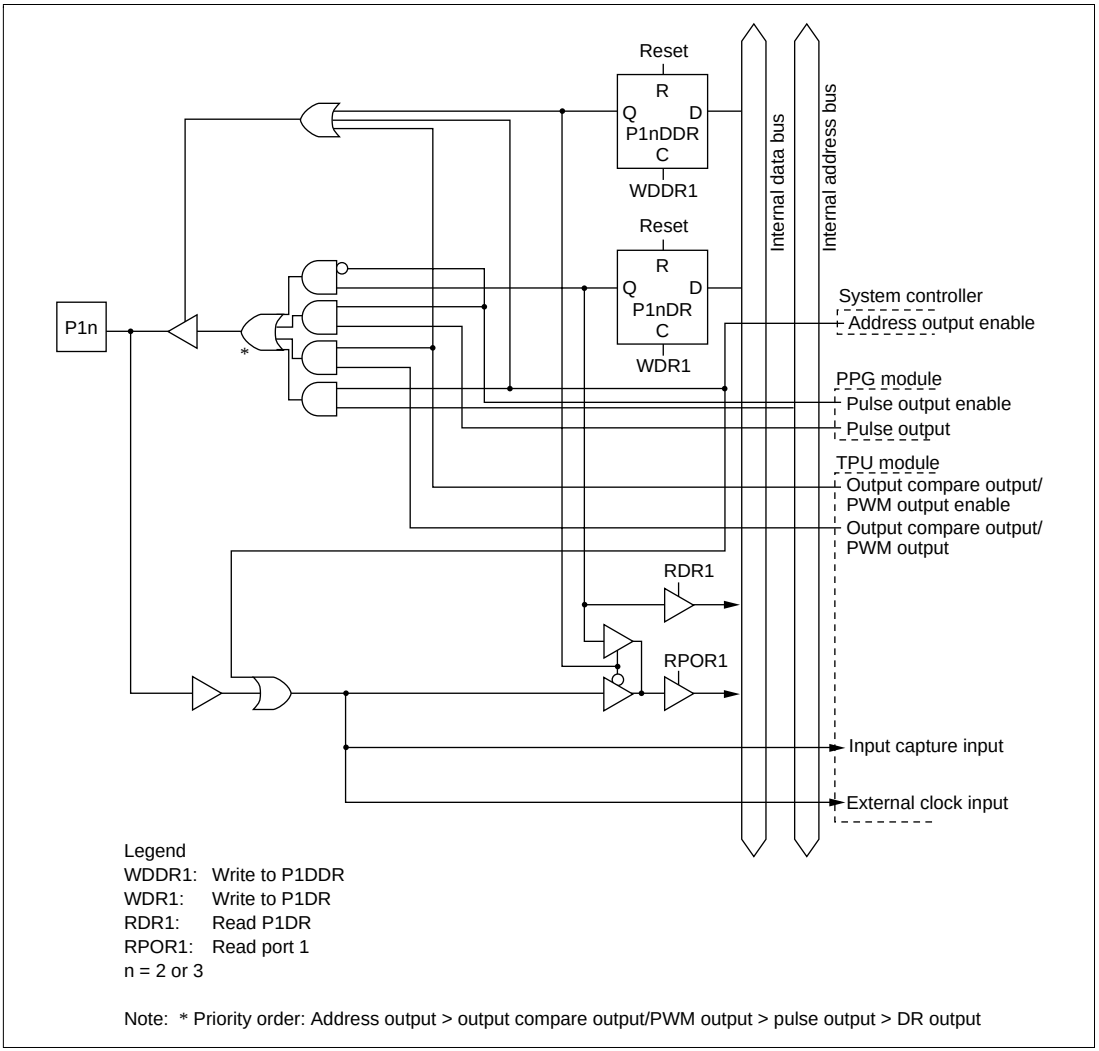


Figure C-1 (b) Port 1 Block Diagram (Pins P12 and P13)

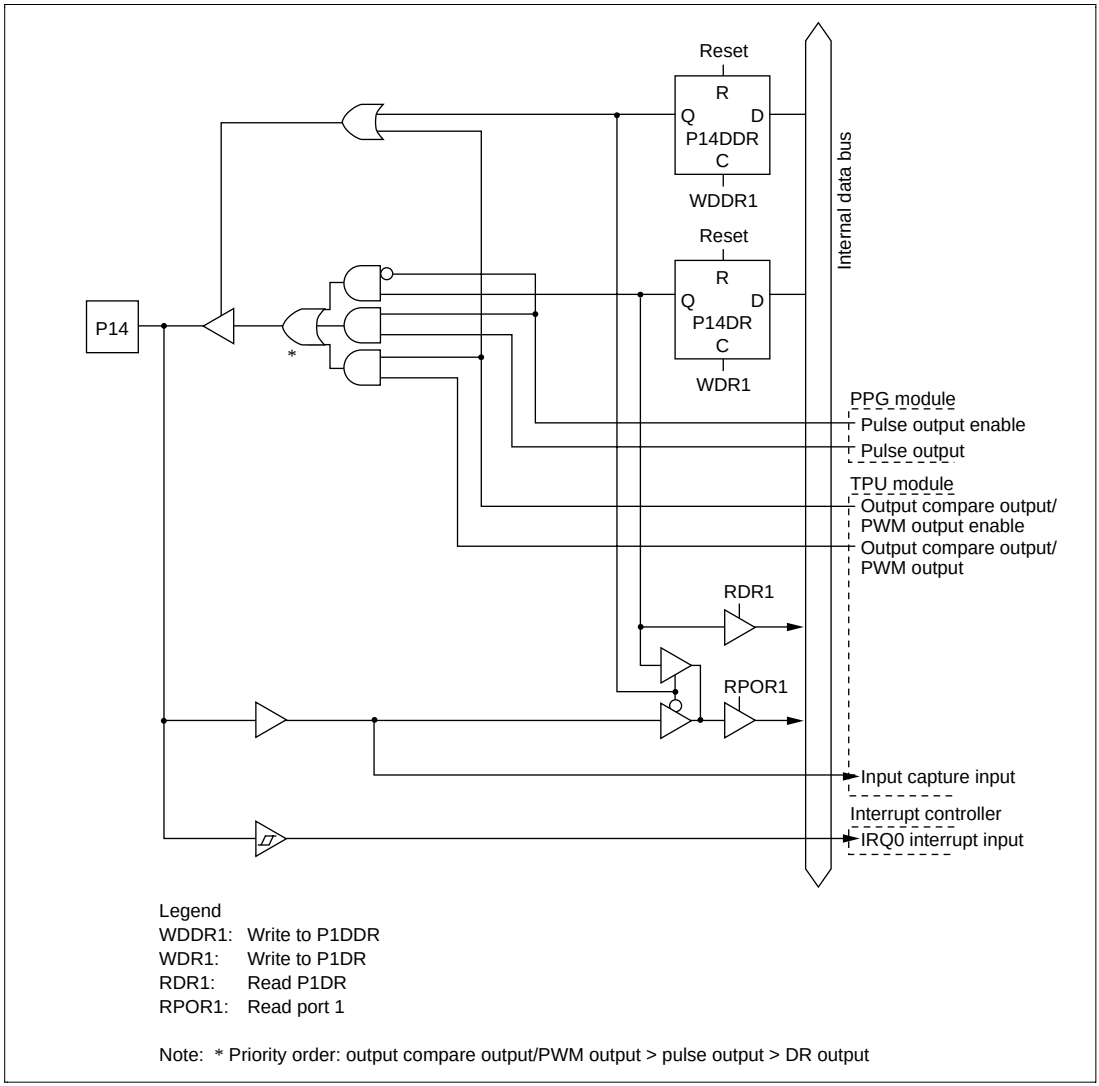


Figure C-1 (c) Port 1 Block Diagram (Pin P14)

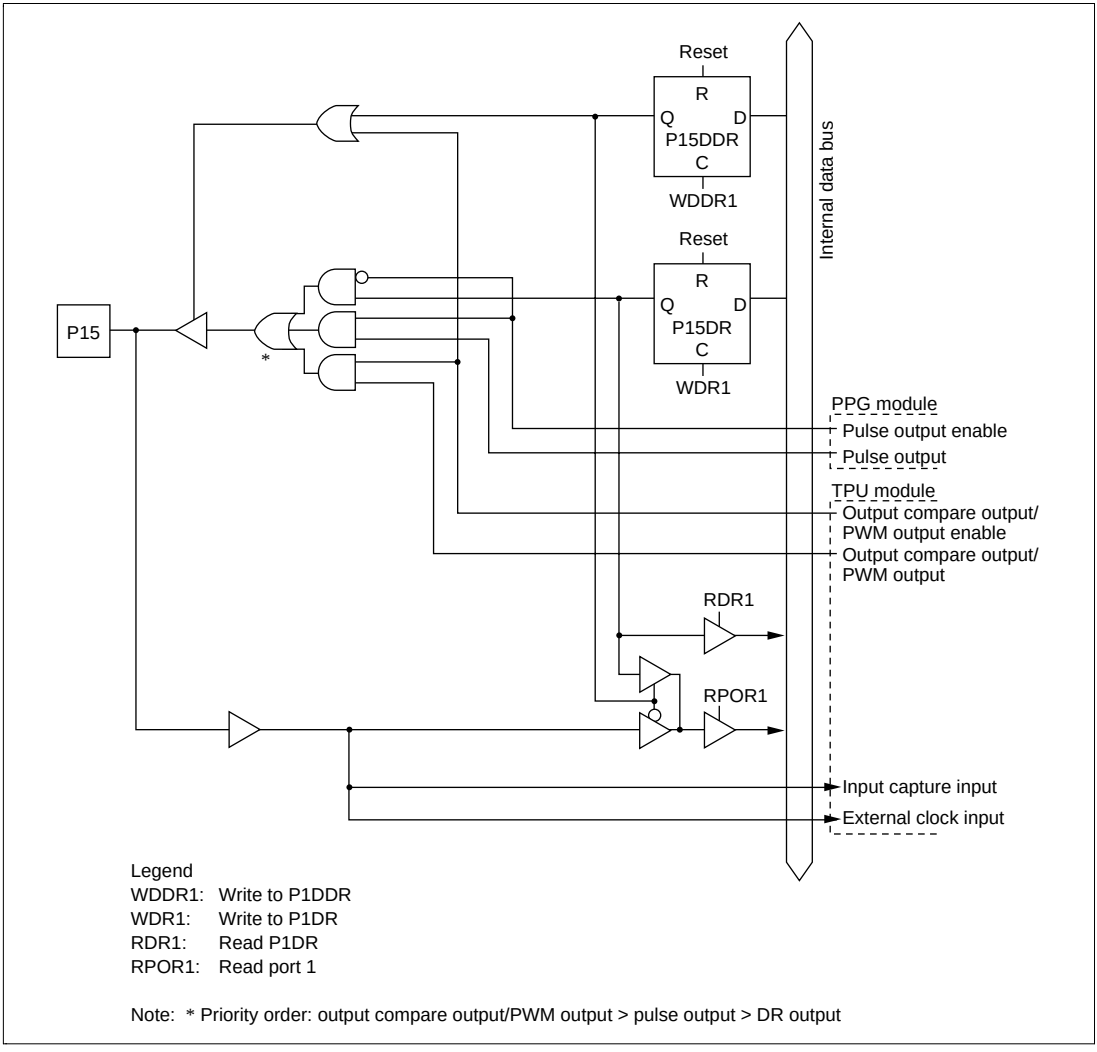


Figure C-1 (d) Port 1 Block Diagram (Pin P15)

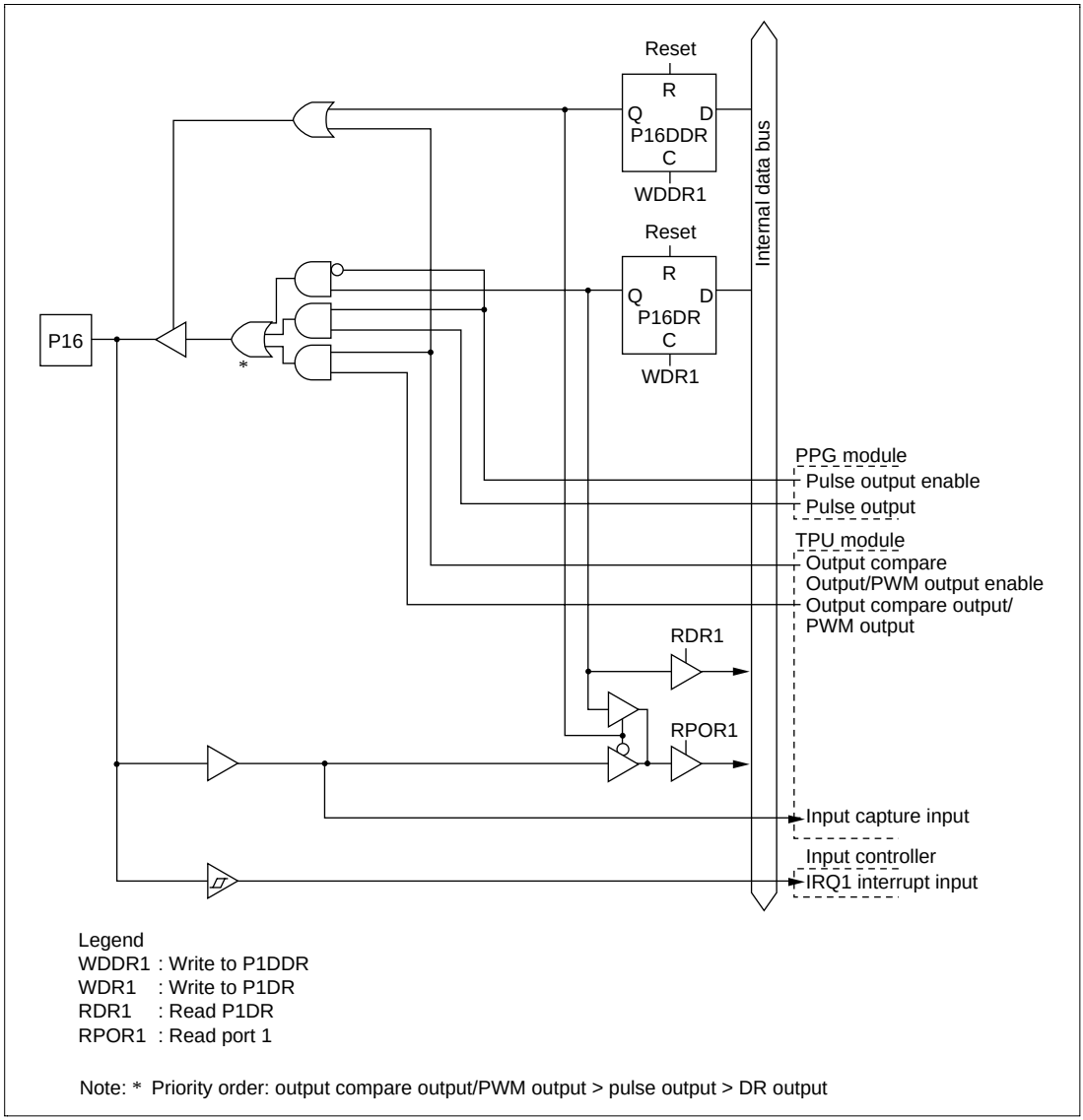


Figure C-1 (e) Port 1 Block Diagram (Pin P16)

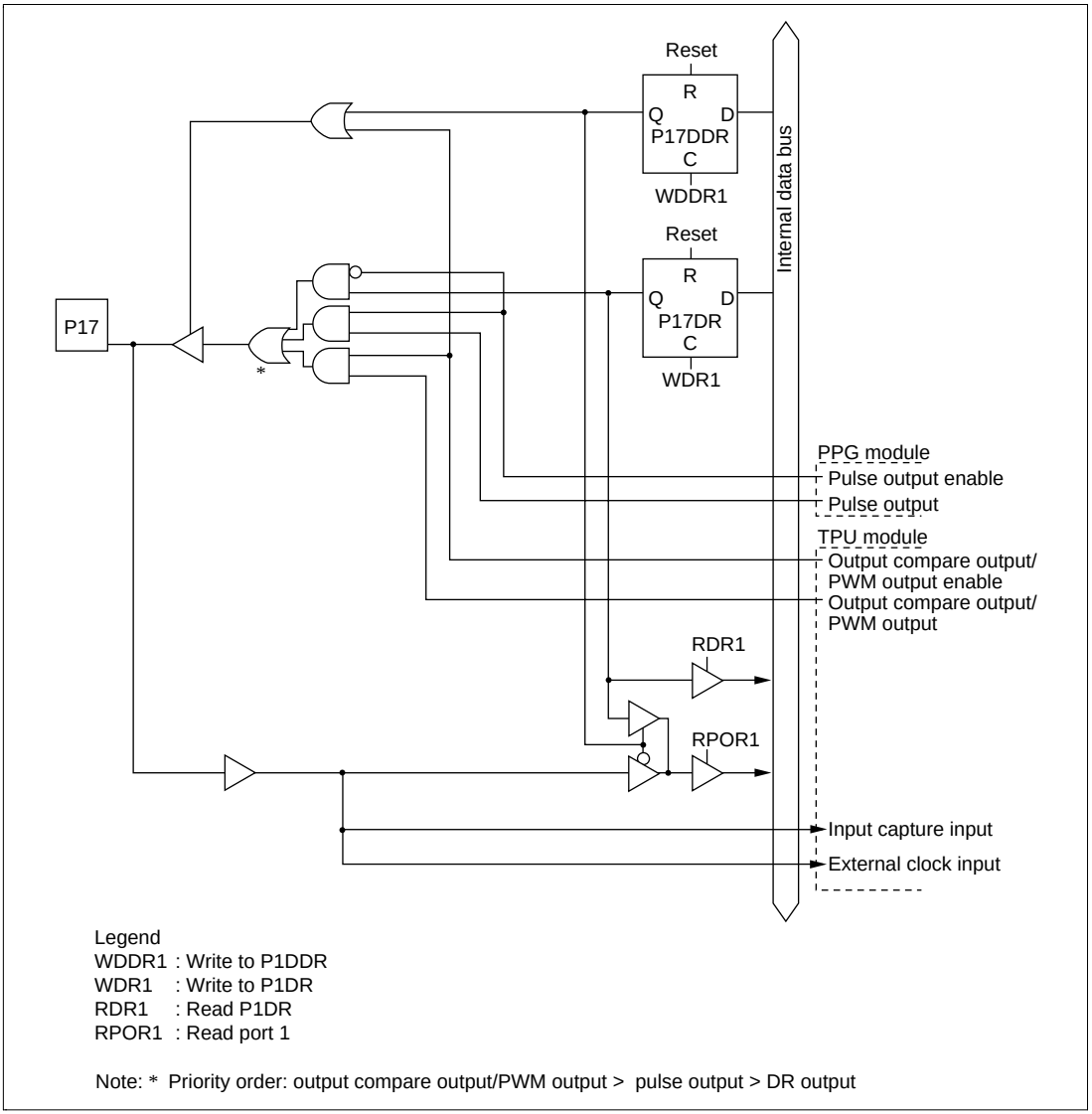


Figure C-1 (f) Port 1 Block Diagram (Pin P17)

C.2 Port 3 Block Diagrams

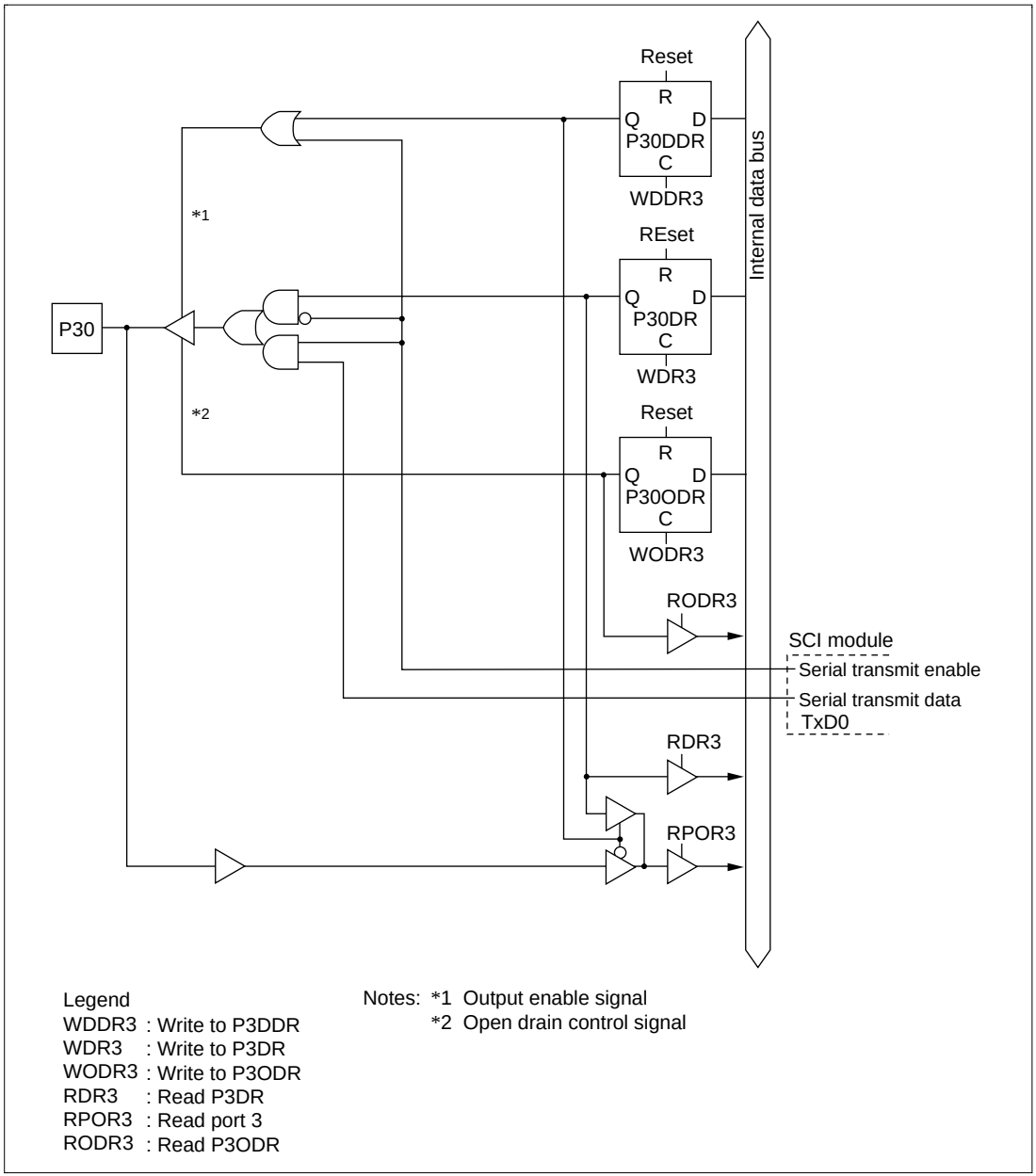


Figure C-2 (a) Port 3 Block Diagram (Pin P30)

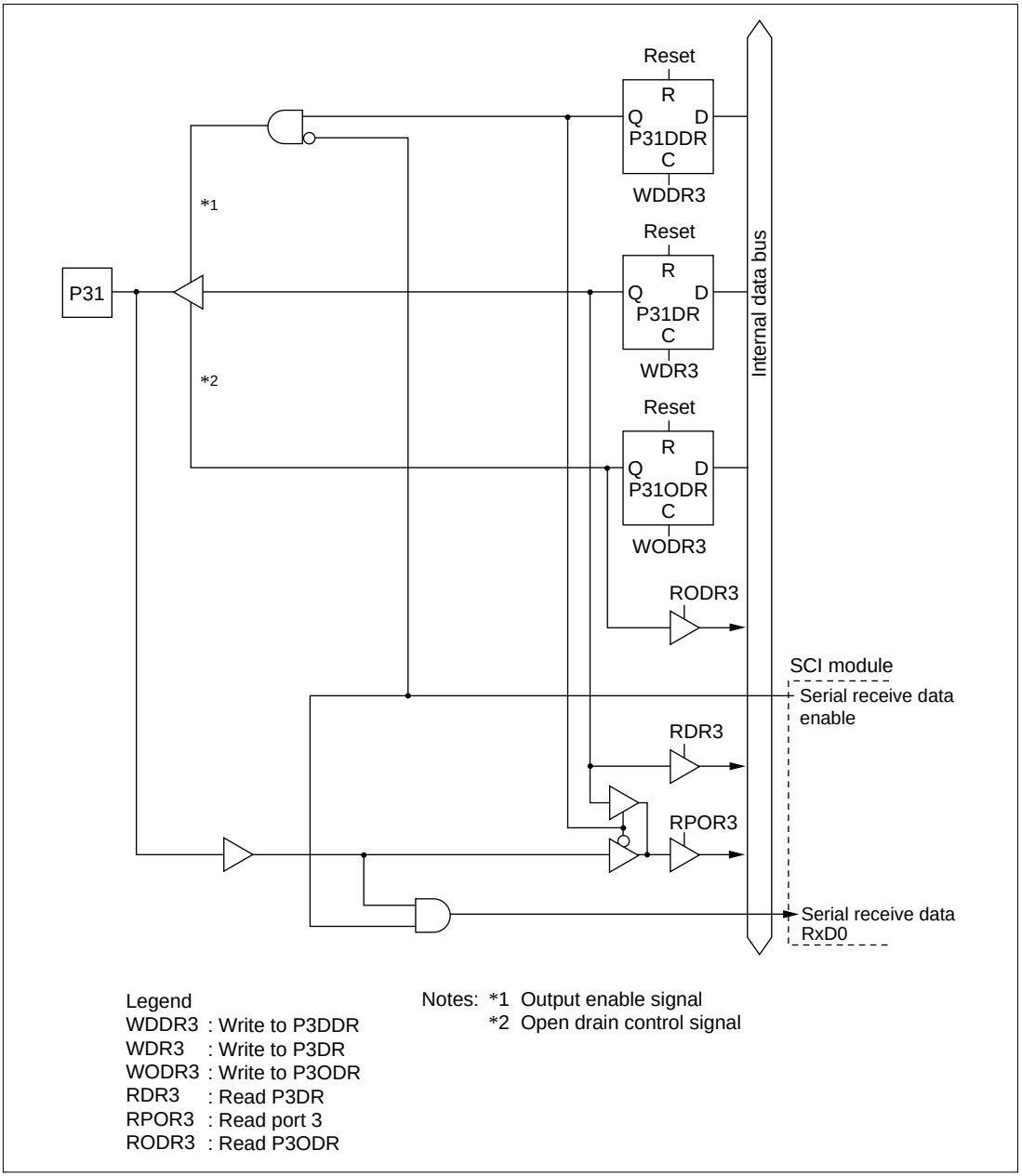
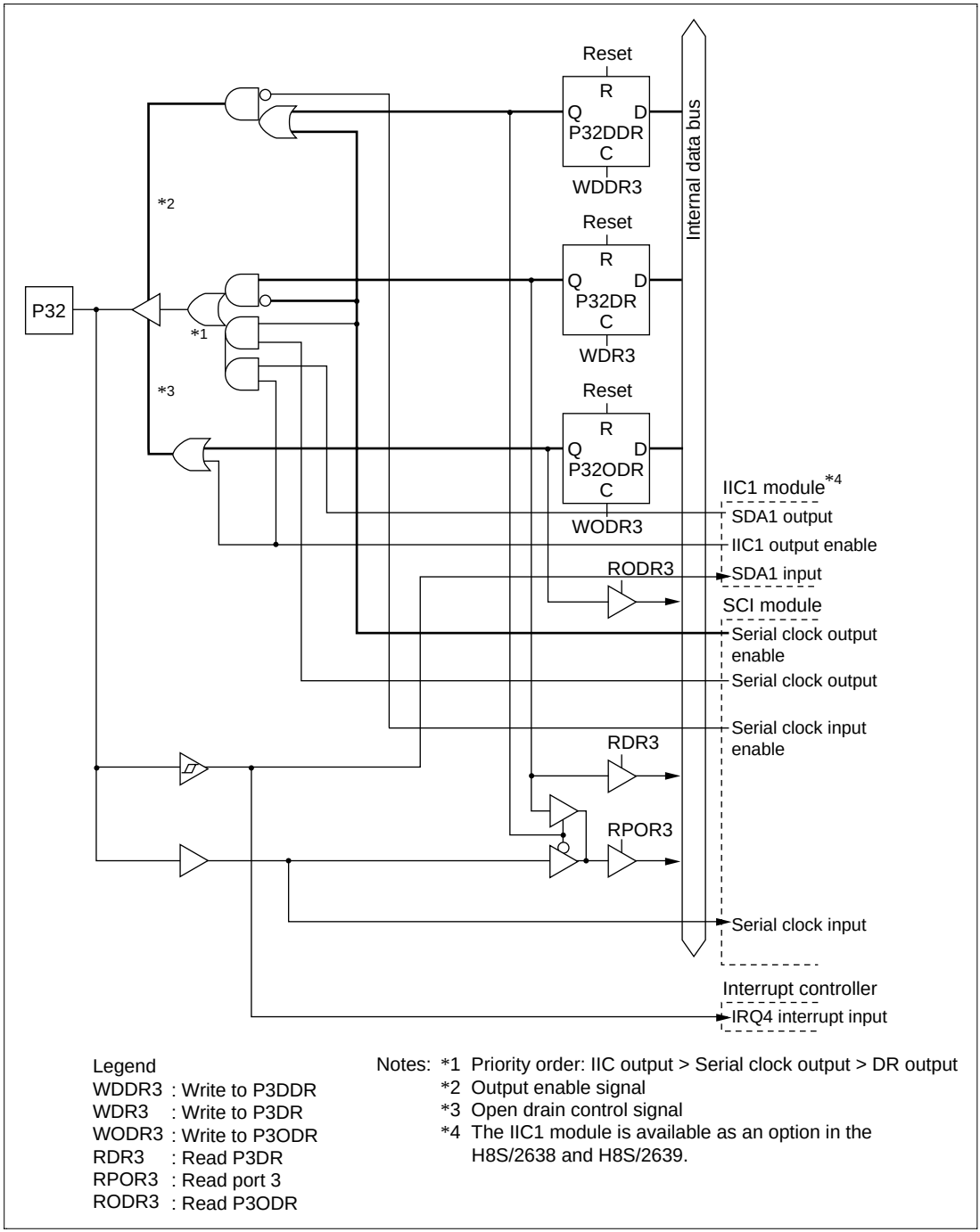


Figure C-2 (b) Port 3 Block Diagram (Pin P31)



Legend
WDDR3 : Write to P3DDR
WDR3 : Write to P3DR
WODR3 : Write to P3ODR
RDR3 : Read P3DR
RPOR3 : Read port 3
RODR3 : Read P3ODR

Notes: *1 Priority order: IIC output > Serial clock output > DR output
*2 Output enable signal
*3 Open drain control signal
*4 The IIC1 module is available as an option in the H8S/2638 and H8S/2639.

Figure C-2 (c) Port 3 Block Diagram (Pin P32)

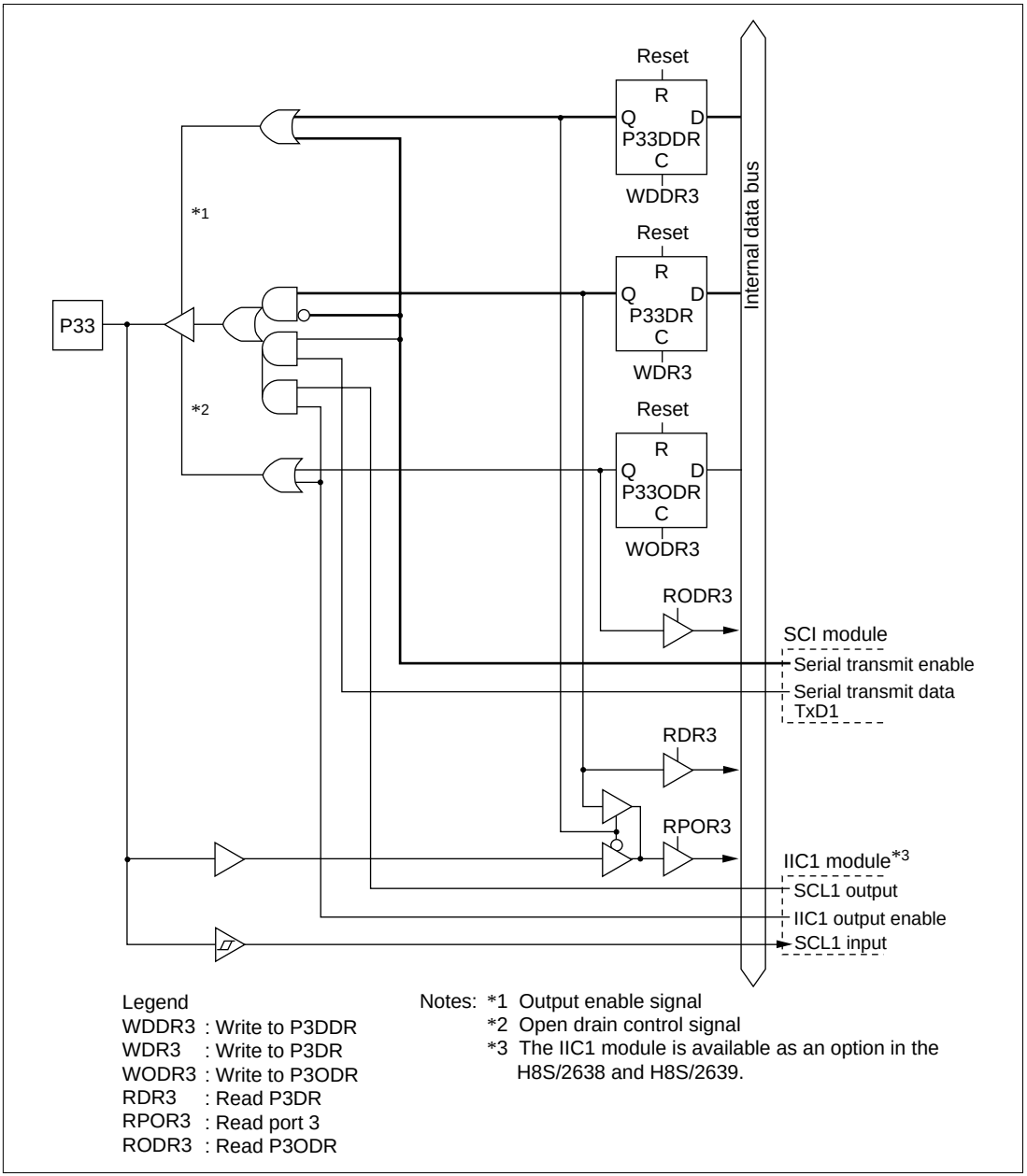


Figure C-2 (d) Port 3 Block Diagram (Pin P33)

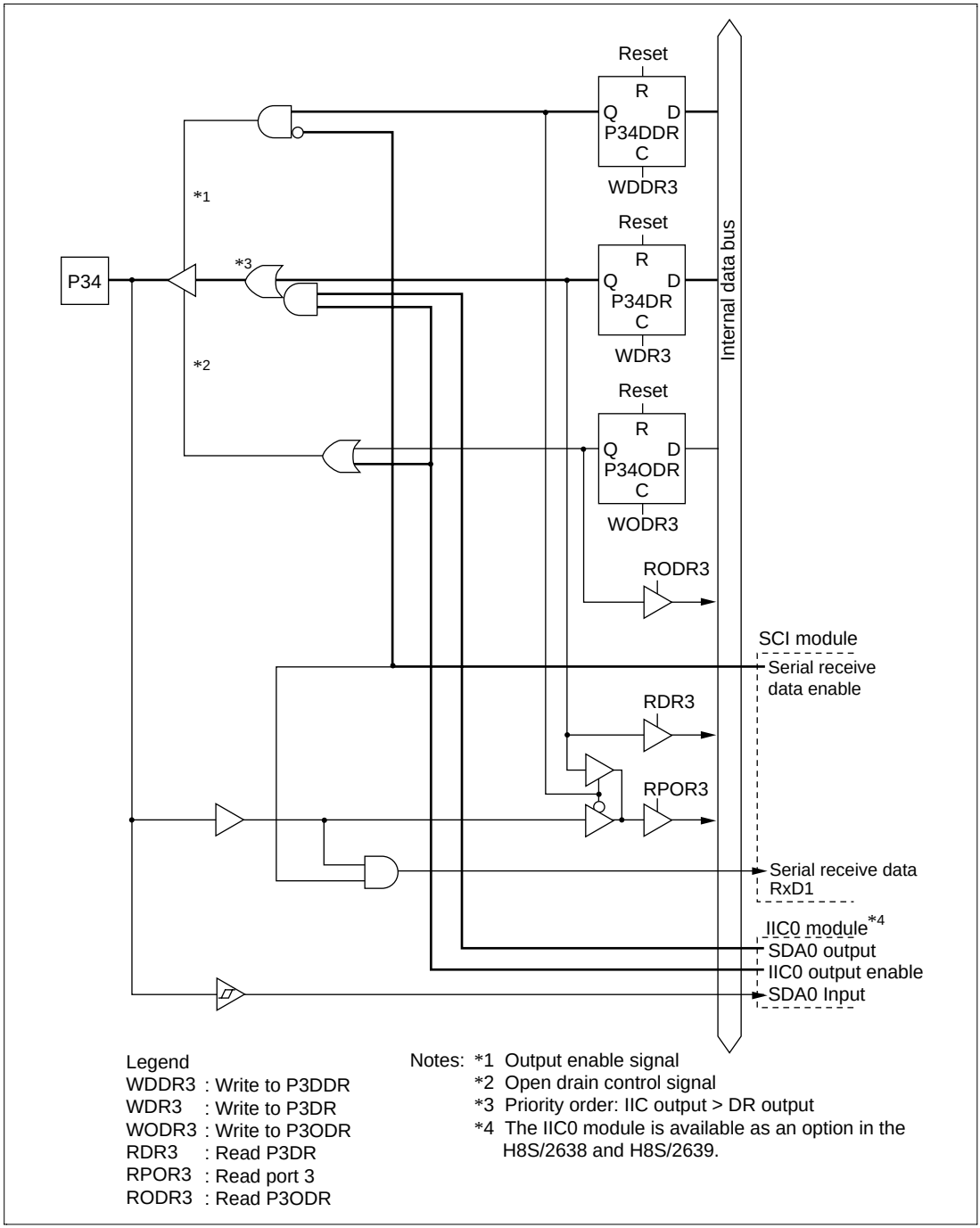
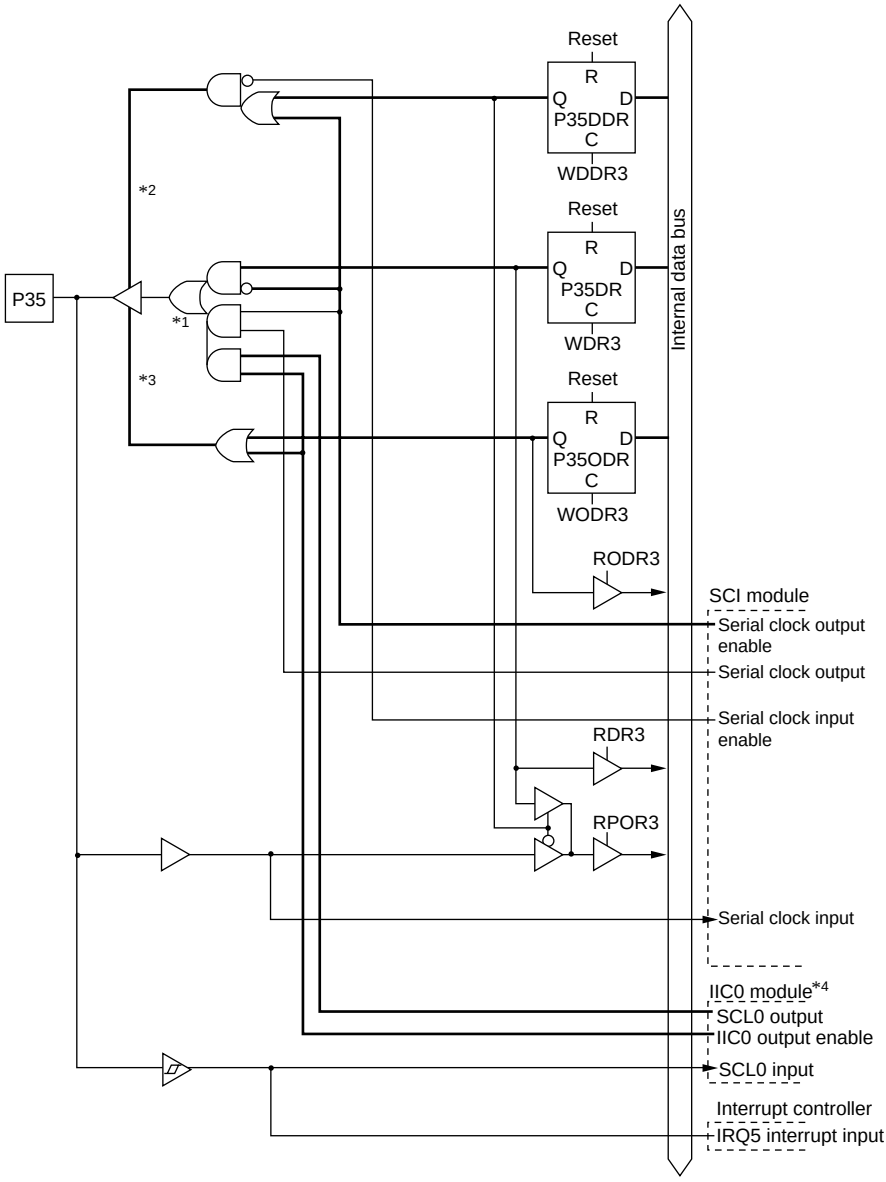


Figure C-2 (e) Port 3 Block Diagram (Pin P34)



Legend
WDDR3 : Write to P3DDR
WDR3 : Write to P3DR
WODR3 : Write to P3ODR
RDR3 : Read P3DR
RPOR3 : Read port 3
RODR3 : Read P3ODR

Notes: *1 Priority order: IIC output > Serial clock output > DR output
*2 Output enable signal
*3 Open drain control signal
*4 The IIC0 module is available as an option in the H8S/2638 and H8S/2639.

Figure C-2 (f) Port 3 Block Diagram (Pin P35)

C.3 Port 4 Block Diagram

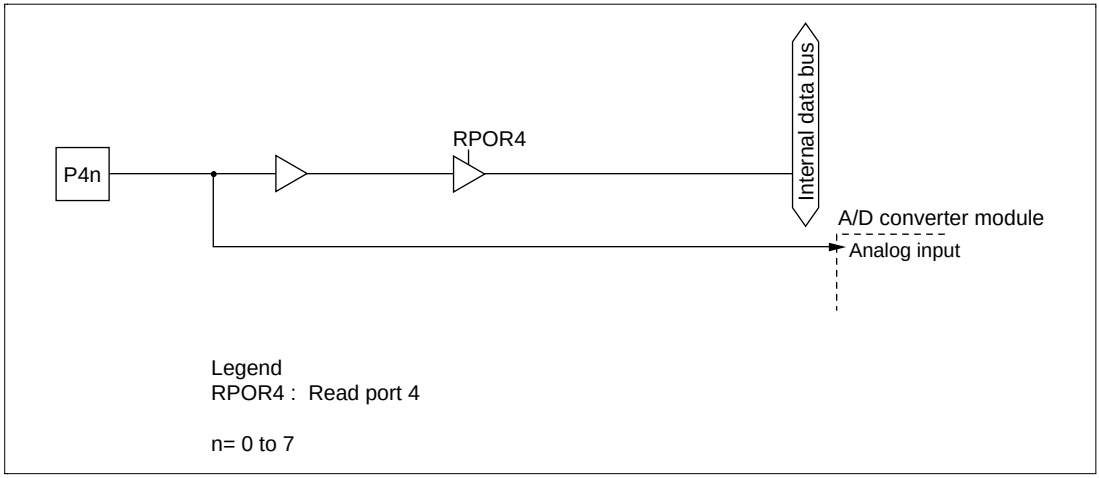


Figure C-3 Port 4 Block Diagram (Pin P4n)

C.4 Port 9 Block Diagram

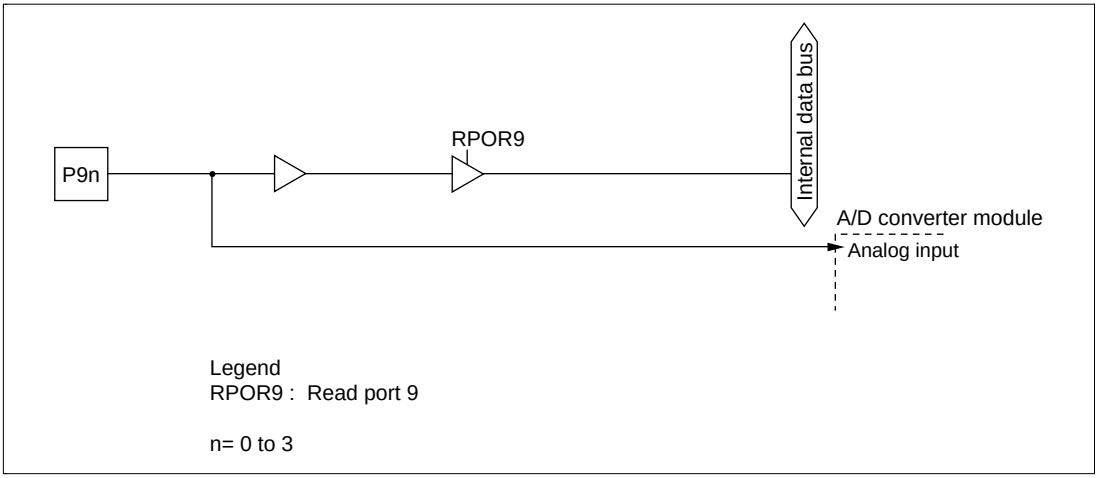


Figure C-4 Port 9 Block Diagram (Pin P9n)

C.5 Port A Block Diagram

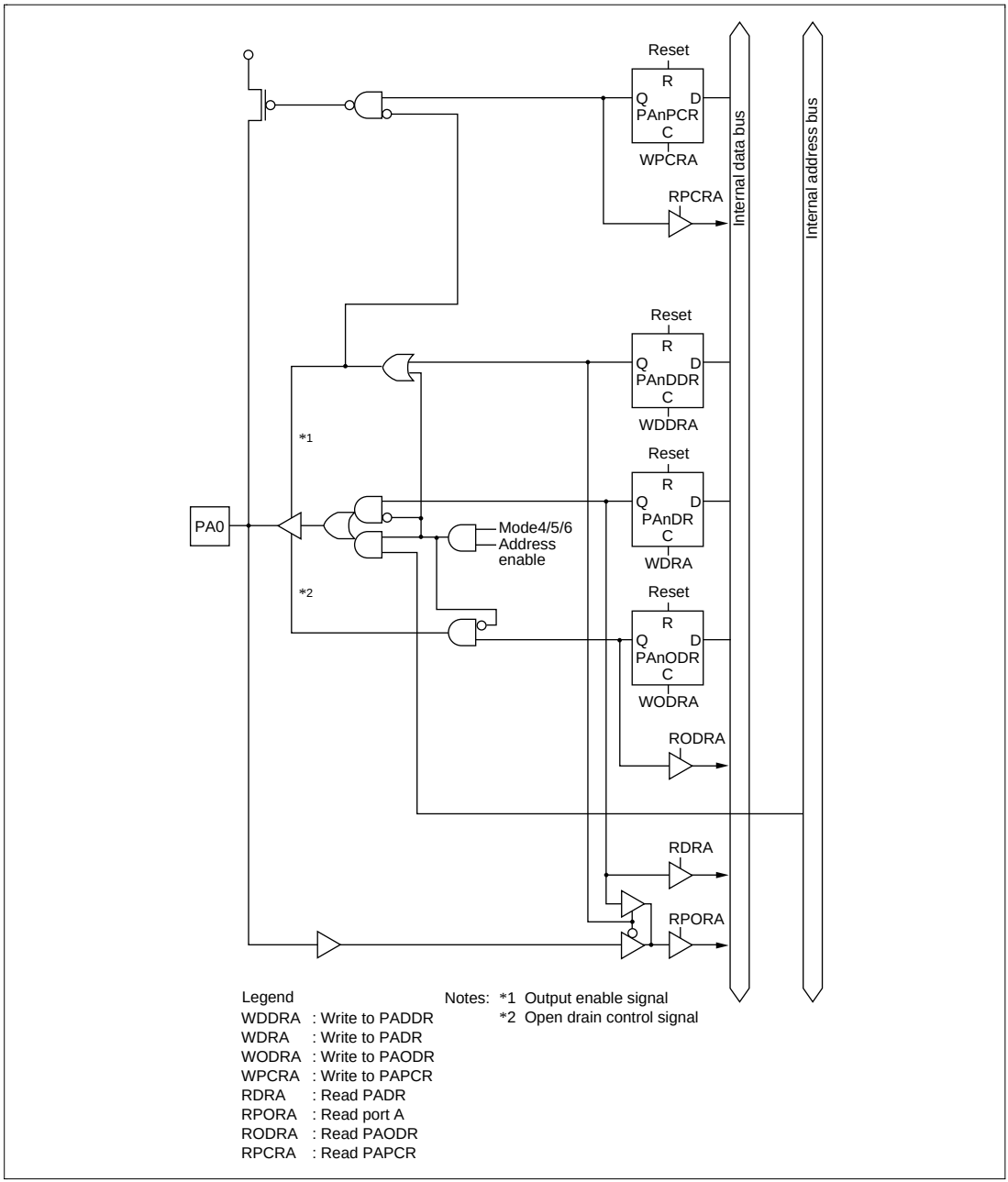


Figure C-5 (a) Port A Block Diagram (Pin PA0)

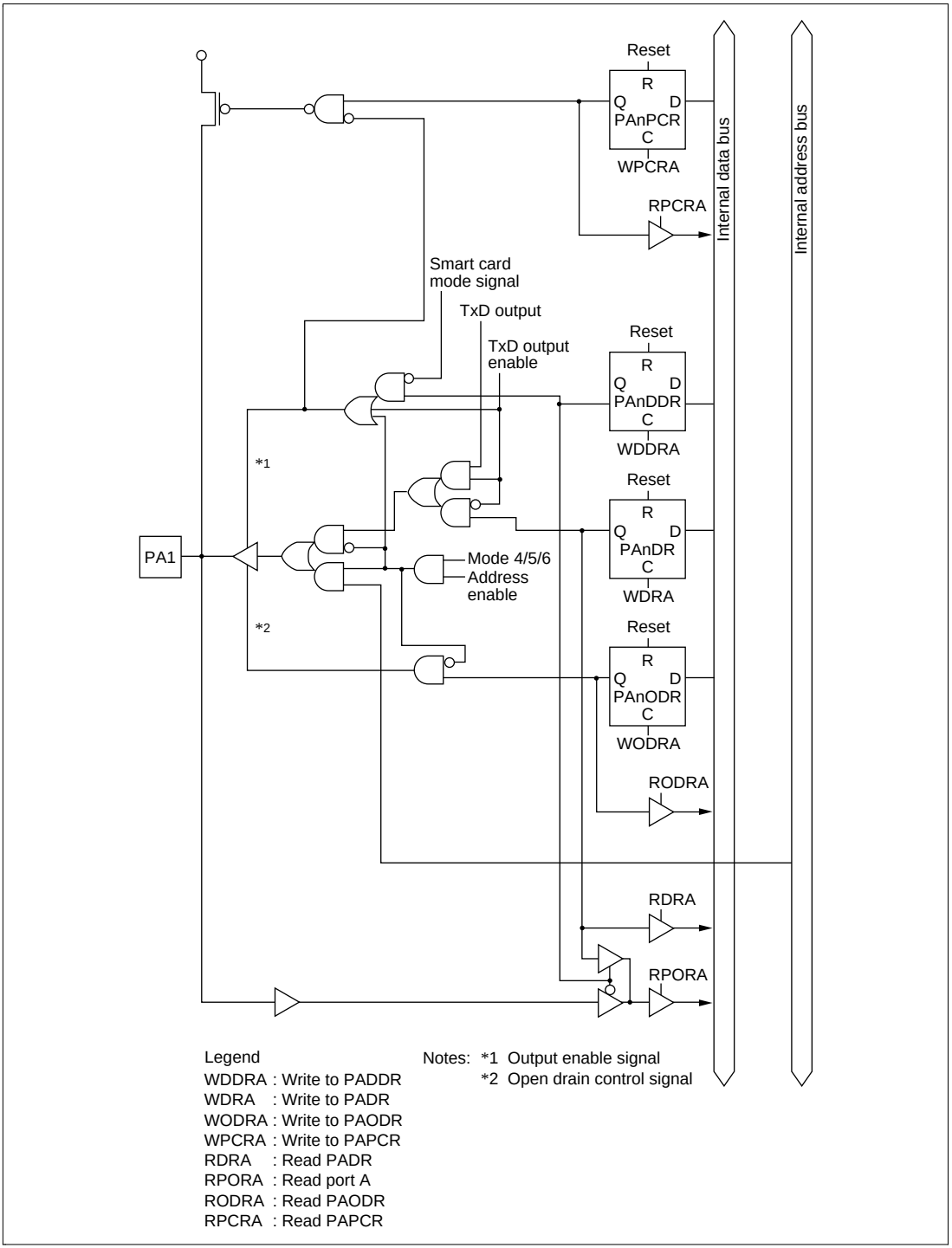


Figure C-5 (b) Port A Block Diagram (Pin PA1)

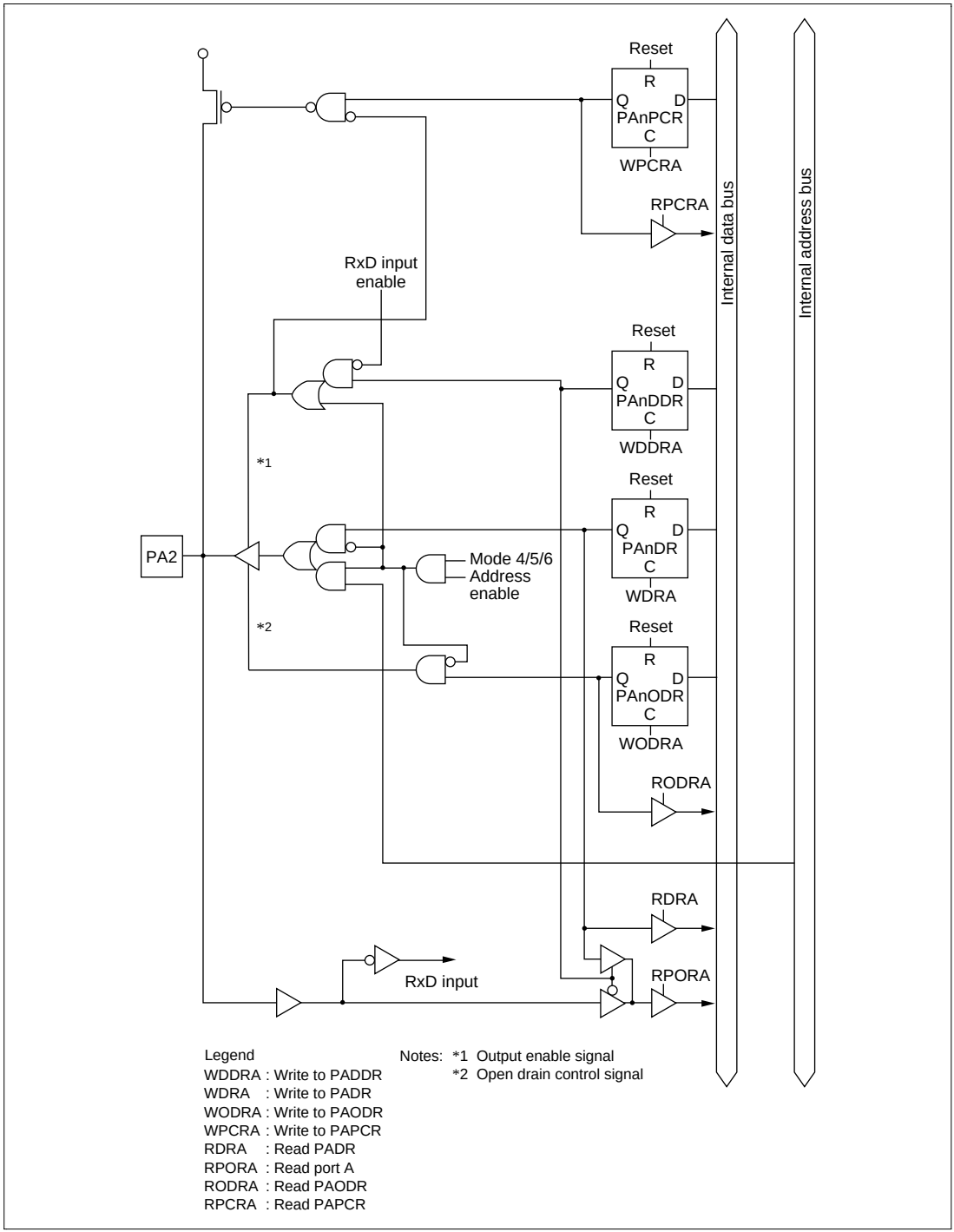
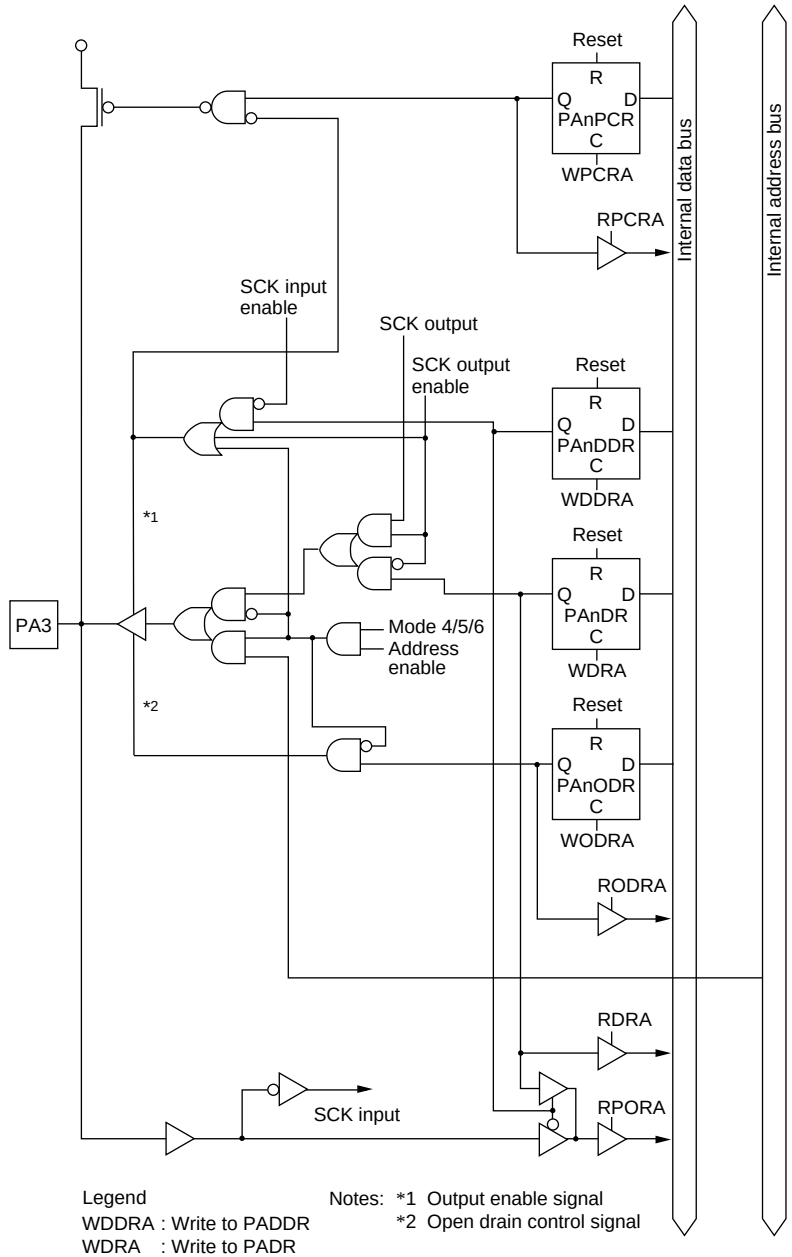


Figure C-5 (c) Port A Block Diagram (Pin PA2)



Legend
WDDRA : Write to PADDR
WDRA : Write to PADR
WODRA : Write to PAODR
WPCRA : Write to PAPCR
RDRA : Read PADR
RPORA : Read port A
RODRA : Read PAODR
RPCRA : Read PAPCR

Notes: *1 Output enable signal
*2 Open drain control signal

Figure C-5 (d) Port A Block Diagram (Pin PA3)

C.6 Port B Block Diagram

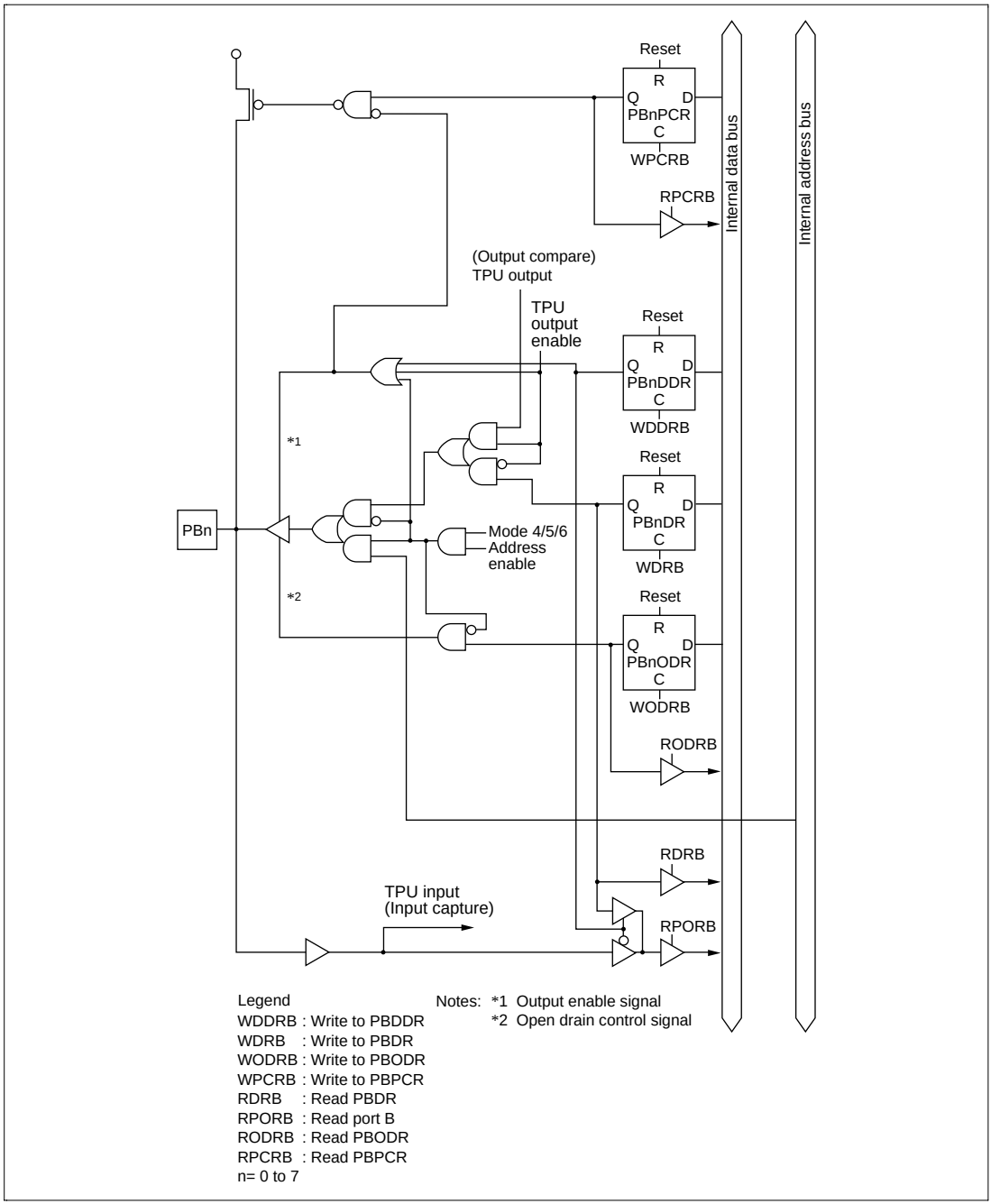


Figure C-6 Port B Block Diagram (Pin PBn)

C.7 Port C Block Diagram

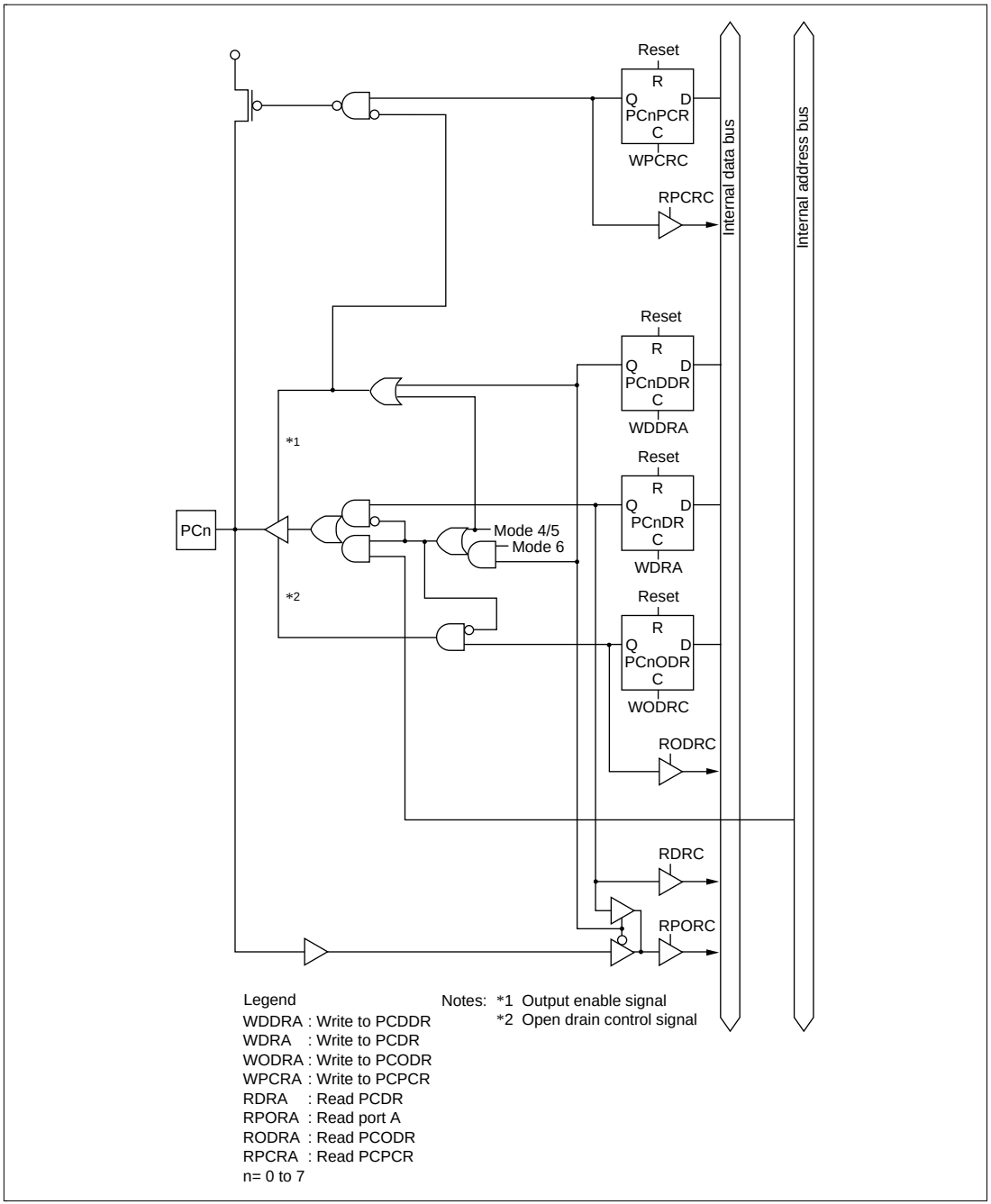


Figure C-7 Port C Block Diagram (Pin PCn)

C.8 Port D Block Diagram

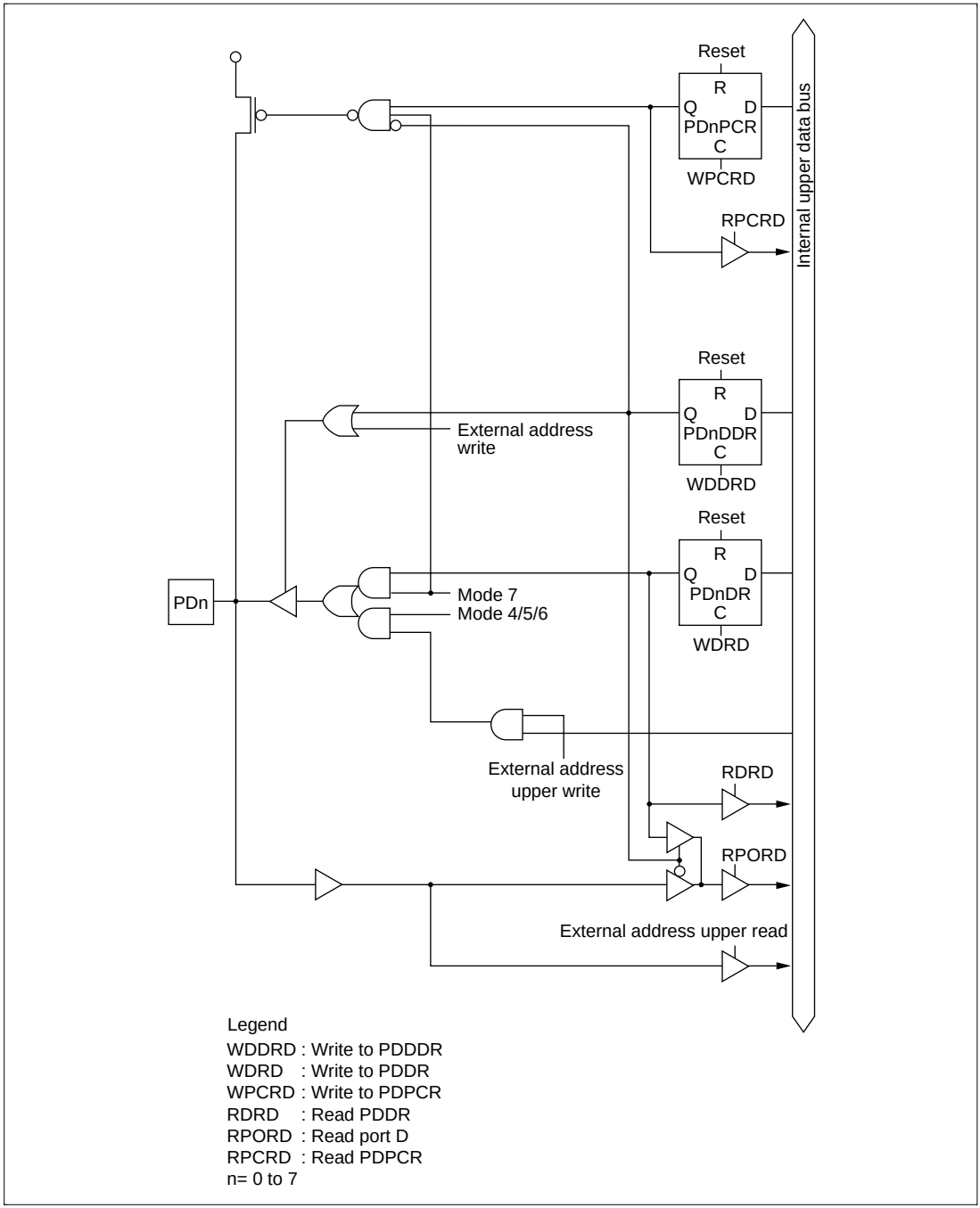


Figure C-8 Port D Block Diagram (Pin PDn)

C.9 Port E Block Diagram

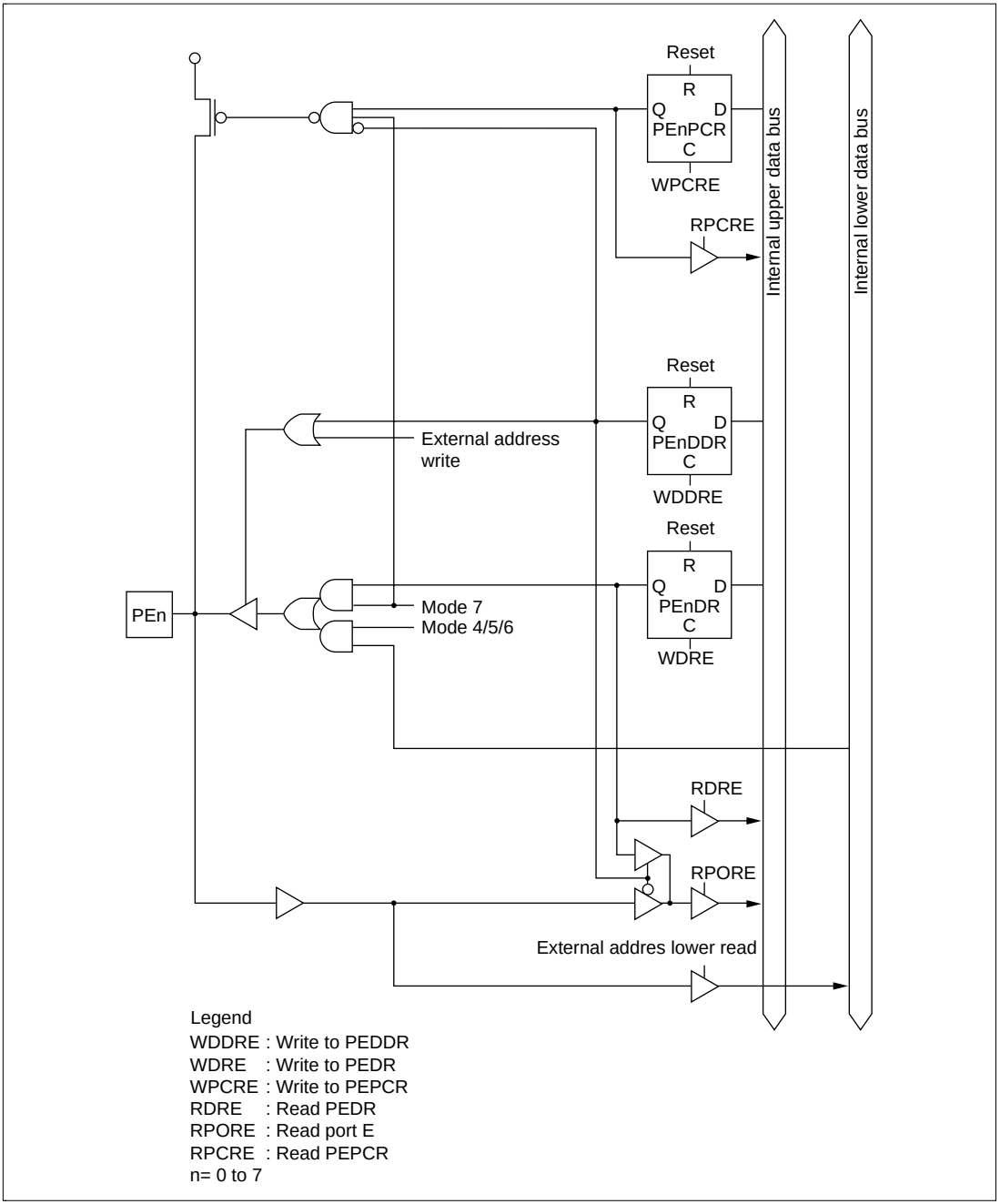


Figure C-9 Port E Block Diagram (Pin PEn)

C.10 Port F Block Diagrams

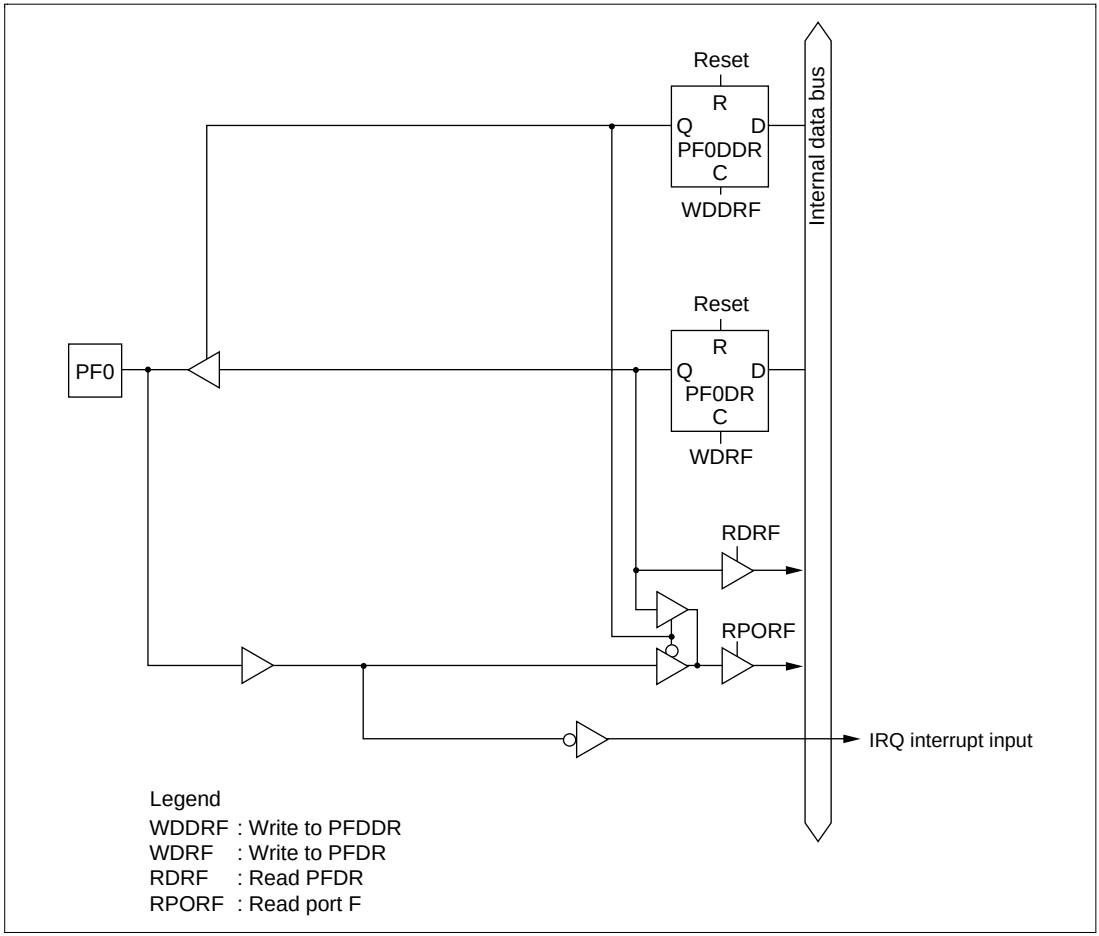


Figure C-10 (a) Port F Block Diagram (Pin PF0)

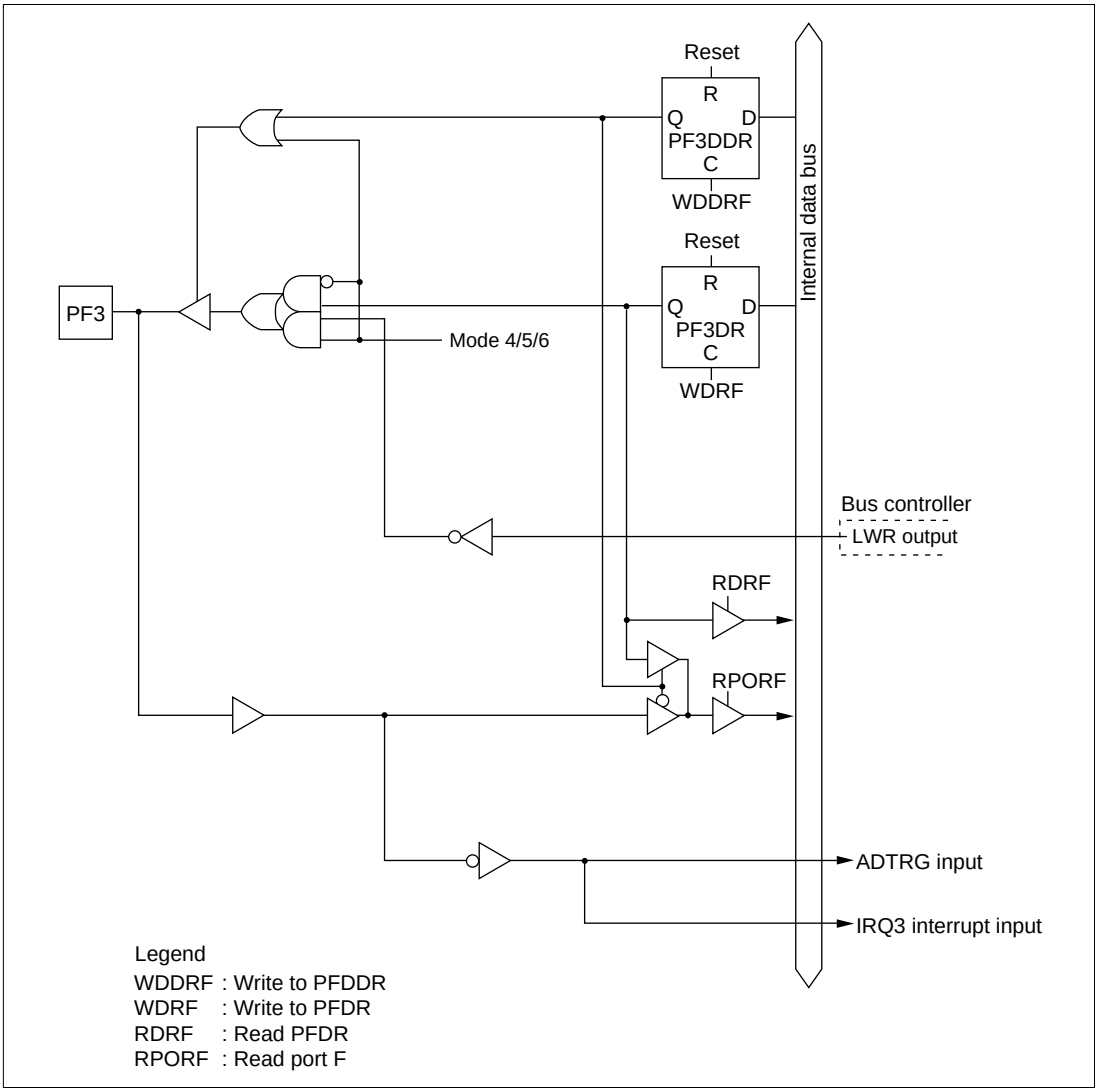


Figure C-10 (b) Port F Block Diagram (Pin PF3)

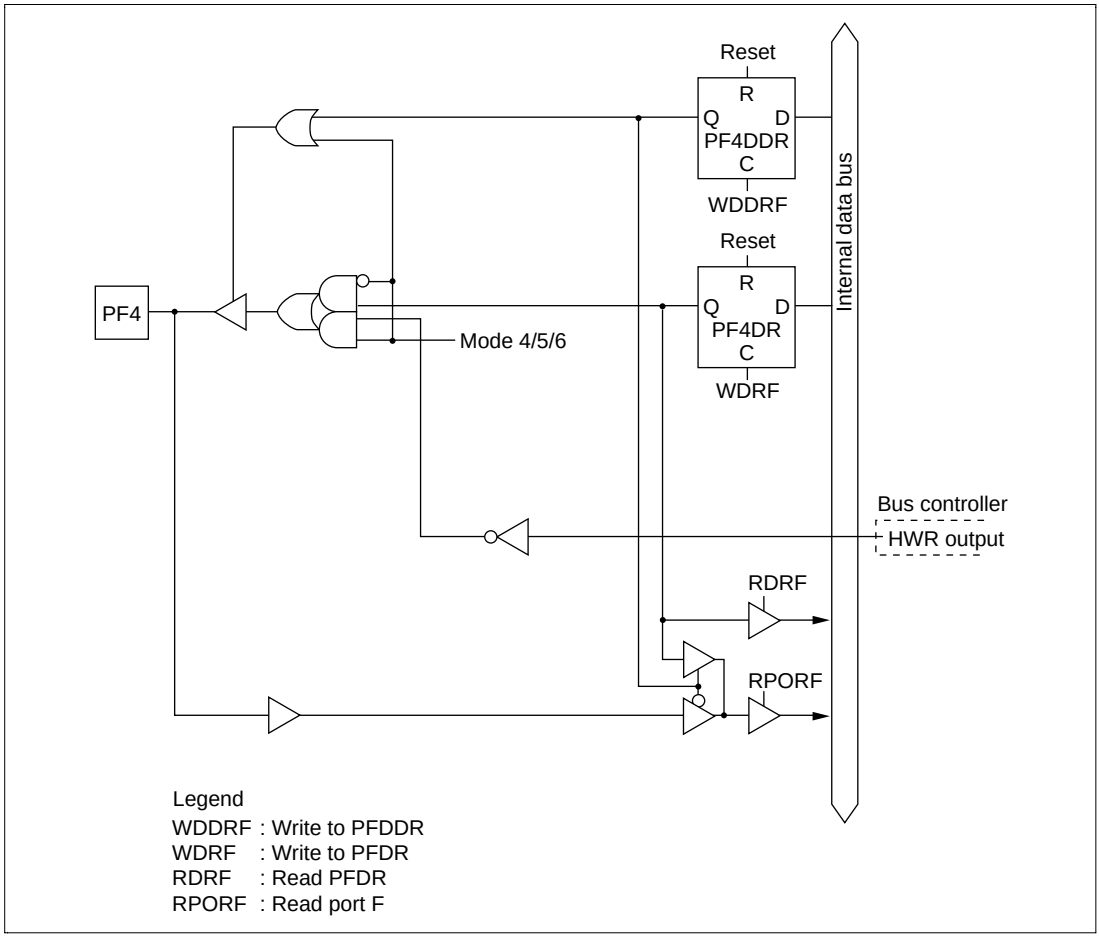


Figure C-10 (c) Port F Block Diagram (Pin PF4)

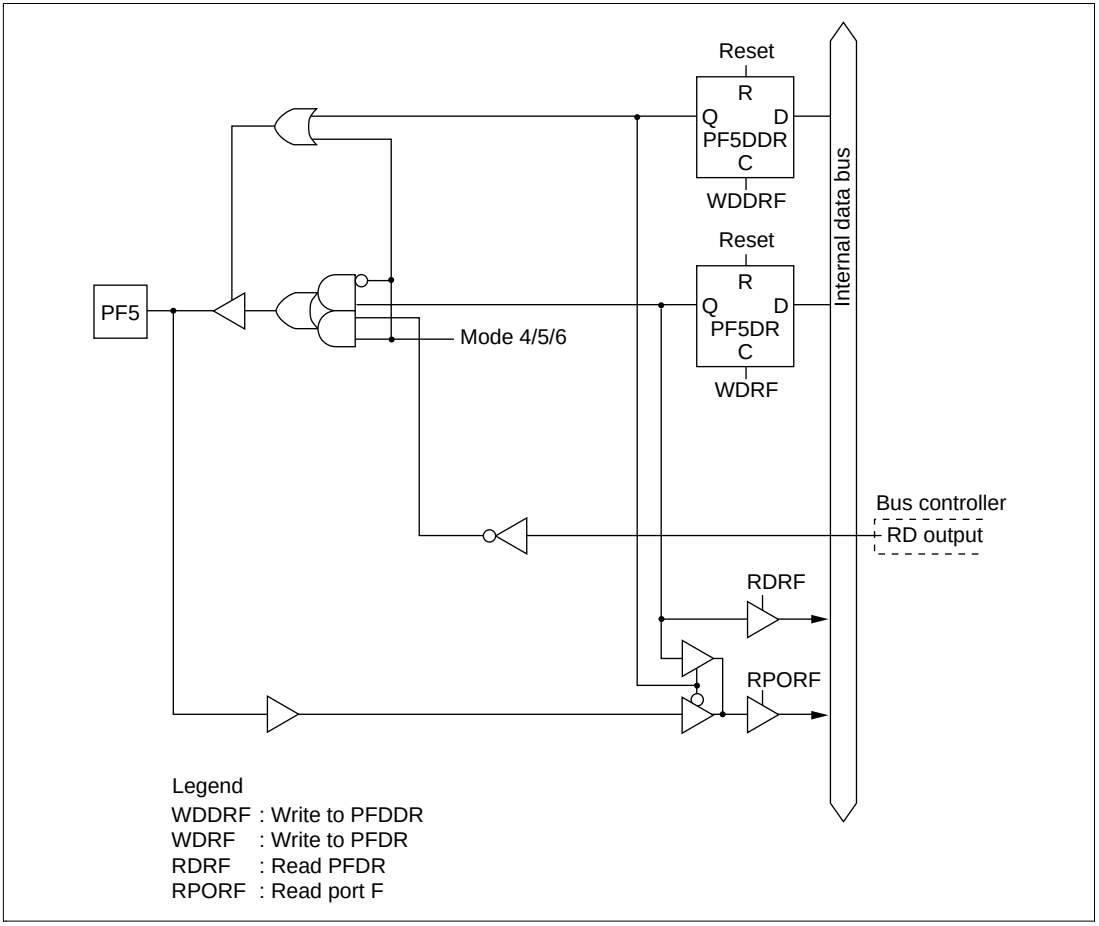


Figure C-10 (d) Port F Block Diagram (Pin PF5)

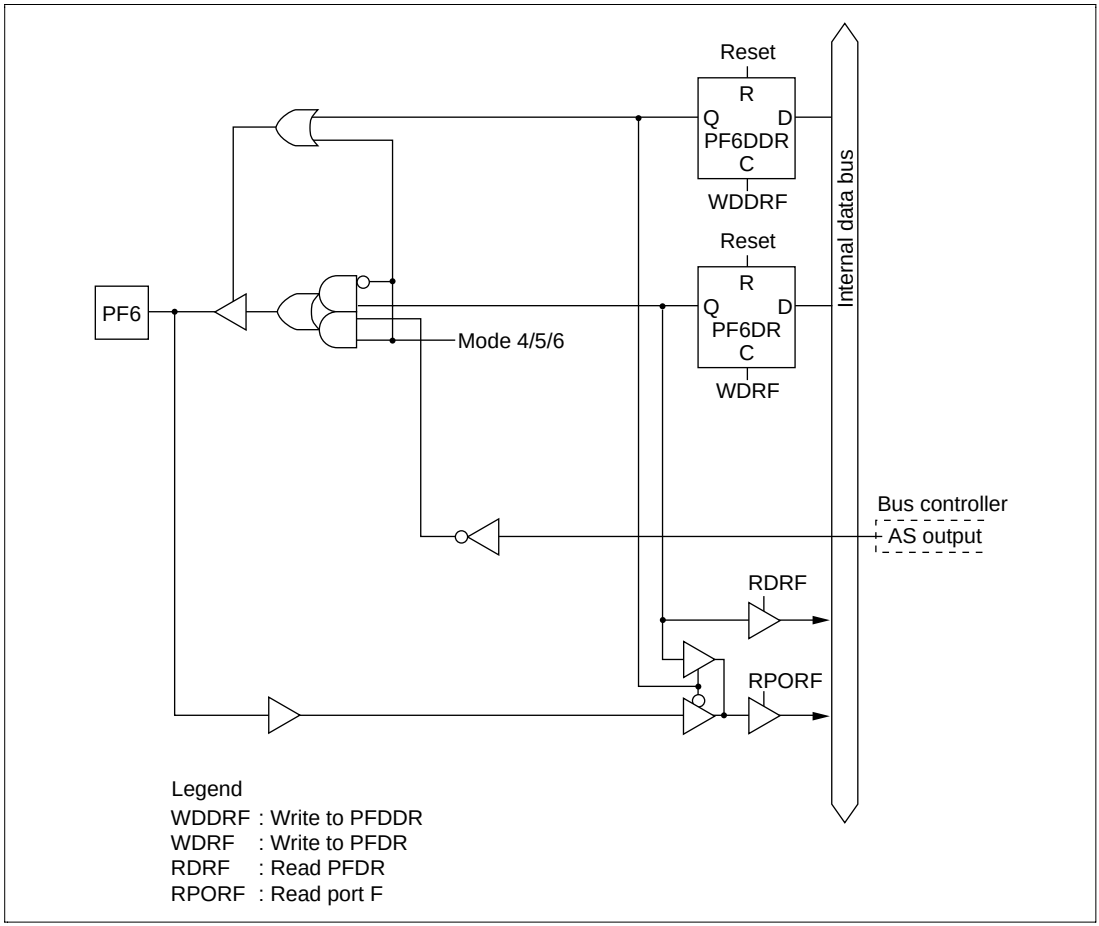


Figure C-10 (e) Port F Block Diagram (Pin PF6)

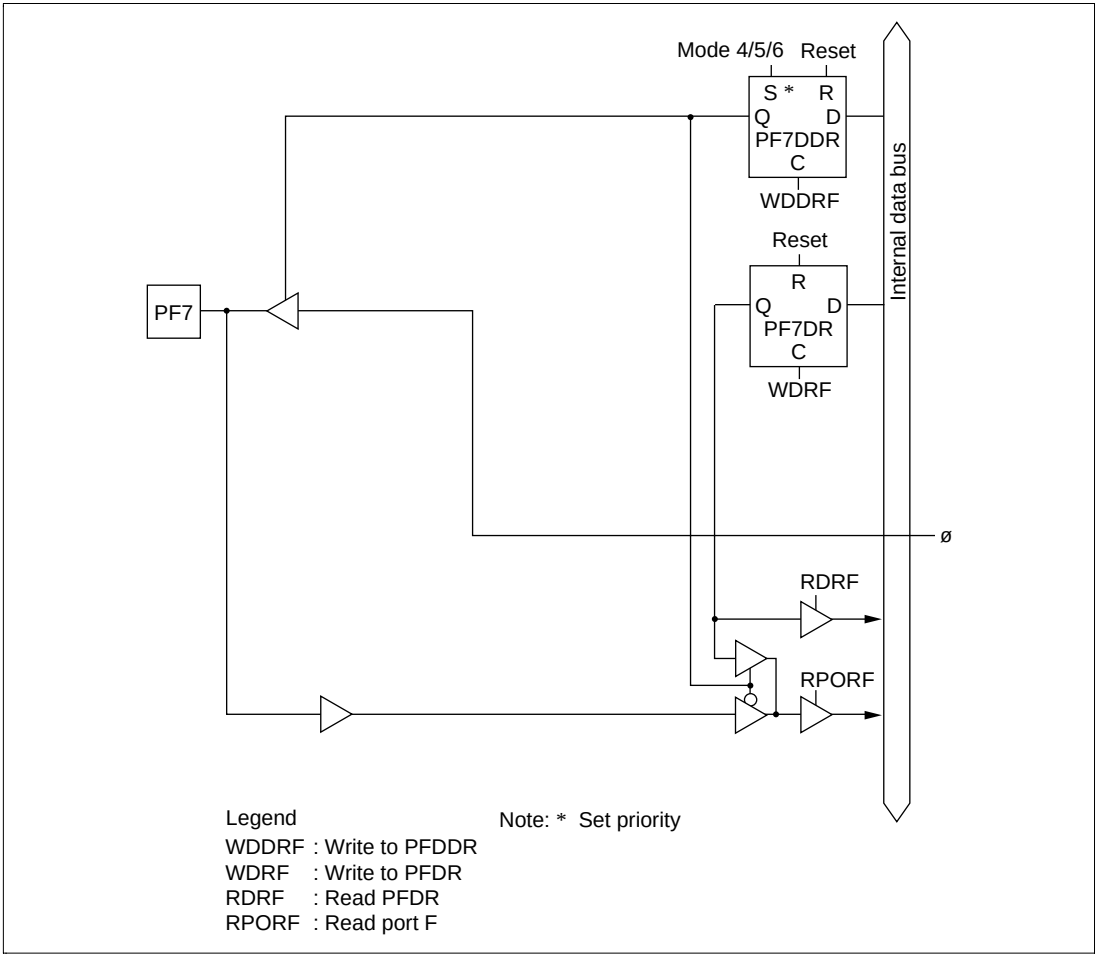


Figure C-10 (f) Port F Block Diagram (Pin PF7)

C.11 Port G Block Diagram

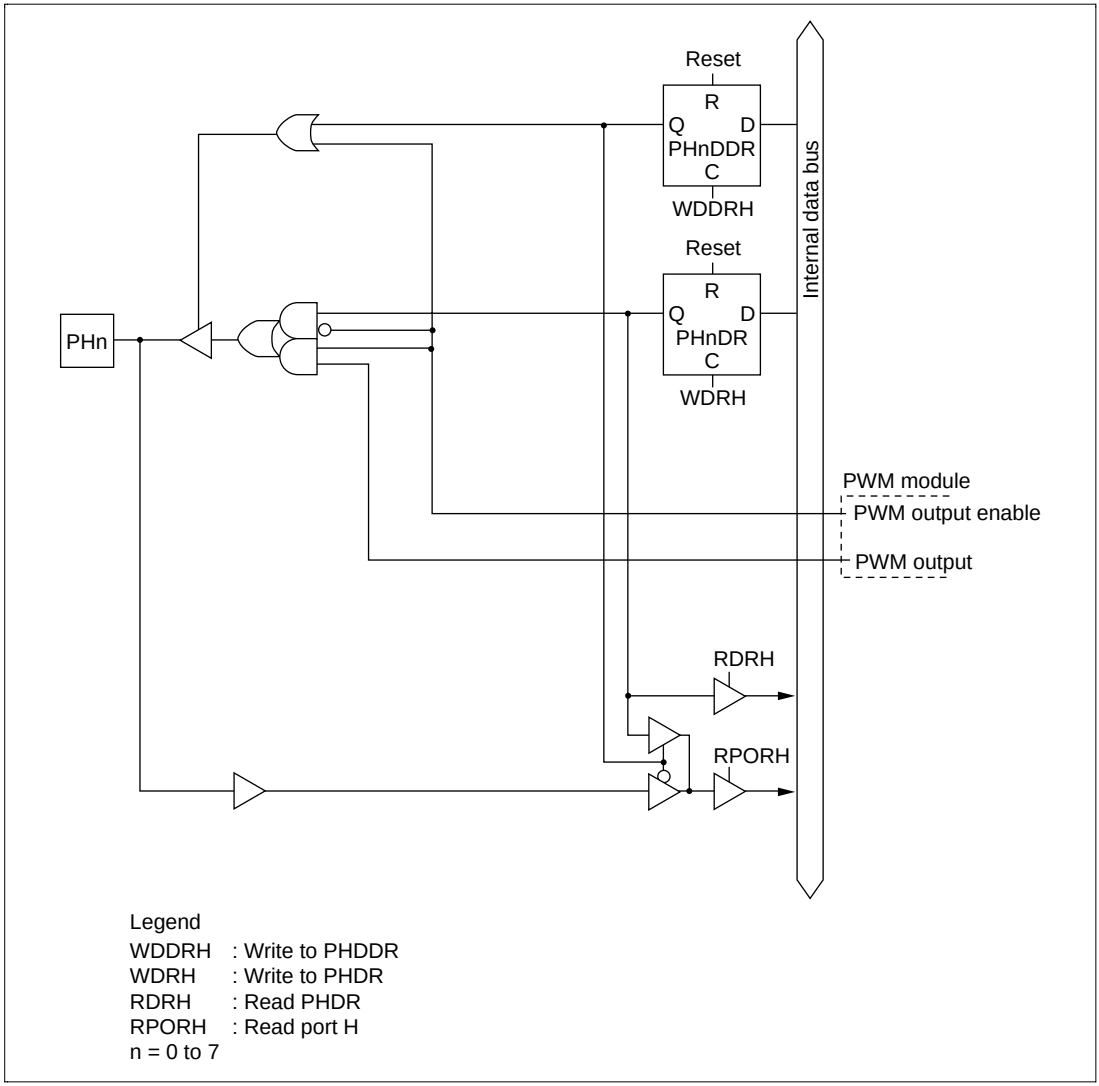


Figure C-11 Port H Block Diagram (Pin PHn)

C.12 Port J Block Diagram

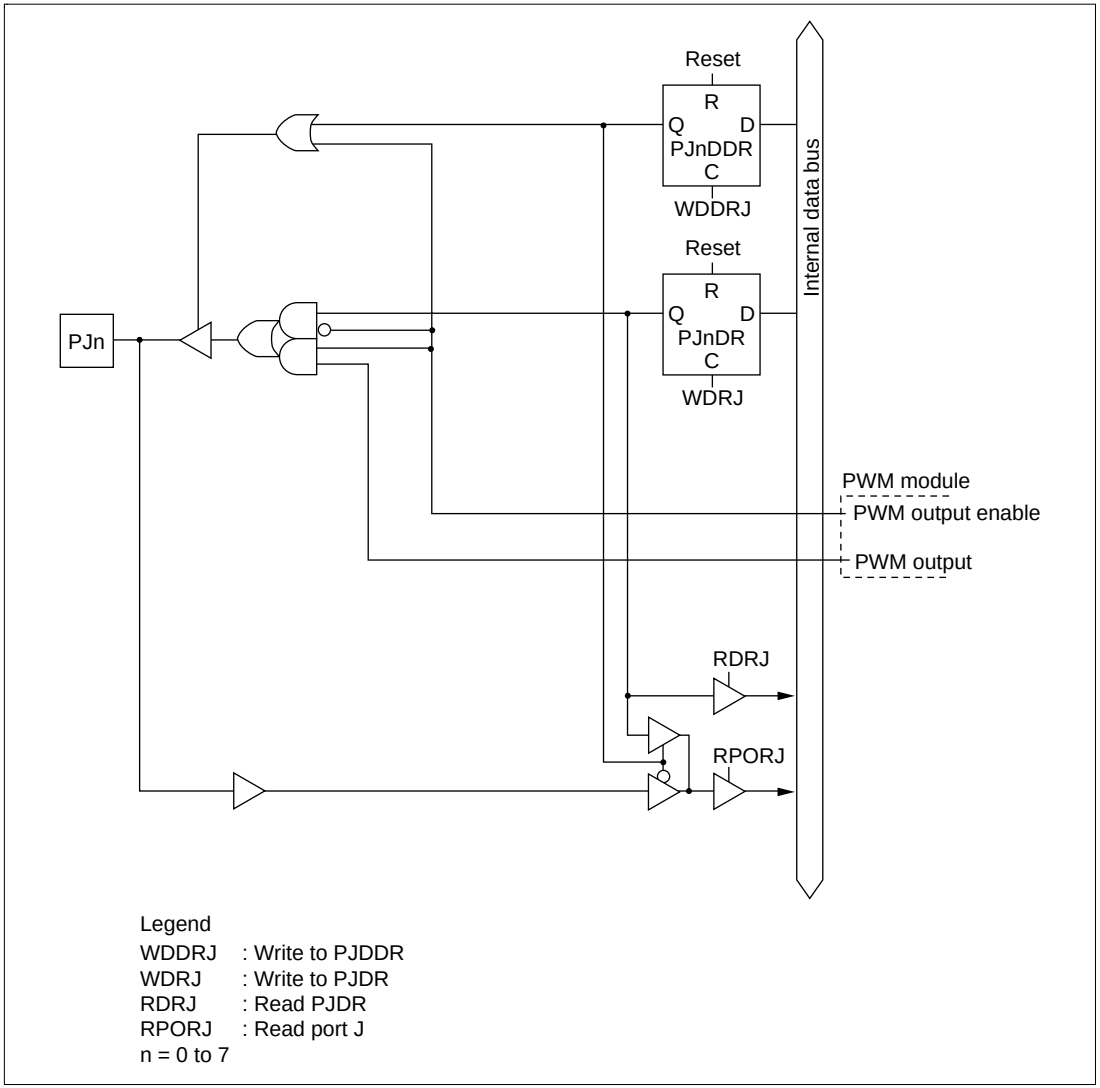


Figure C-12 Port J Block Diagram (Pin PJn)

Appendix D Pin States

D.1 Port States in Each Mode

Table D-1 I/O Port States in Each Processing State

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Program Execution State Sleep Mode
Port 1	4, 5	T	T	P10 to P13	P10 to P13
	6			[Address output, OPE = 0] T [Address output, OPE = 1] kept [Otherwise] kept	[Address output] A20 to A23 [Otherwise] I/O port
				P14 to P17 kept	P14 to P17 I/O port
	7			kept	P10 to P17 I/O port
Port 3	4 to 7	T	T	kept	I/O port
Port 4	4 to 7	T	T	T	Input port
Port 9	4 to 7	T	T	T	Input port
Port A	4, 5	L	T	[Address output, OPE = 0] T [Address output, OPE = 1] kept [Otherwise] kept	[Address output] A19 to A17 [Otherwise] I/O port
	6	T	T		
	7	T	T	kept	[Otherwise] I/O port
Port B	4, 5	L	T	[Address output, OPE = 0] T [Address output, OPE = 1] kept [Otherwise] kept	[Address output] A15 to A8 [Otherwise] I/O port
	6	T	T		
	7	T	T	kept	I/O port

Port Name Pin Name	MCU Operating Mode		Reset	Hardware Standby Mode	Software Standby Mode	Program Execution State Sleep Mode
Port C	4, 5		L	T	[OPE = 0] T [OPE = 1] kept	A7 to A0
	6		T	T	[DDR = 1, OPE = 0] T [DDR = 1, OPE = 1] kept [DDR = 0] kept	[DDR = 1] A7 to A0 [DDR = 0] I/O port
	7		T	T	kept	I/O port
Port D	4 to 6		T	T	T	Data bus
	7		T	T	kept	I/O port
Port E	4 to 6	8 bit bus	T	T	kept	I/O port
		16 bit bus	T	T	T	Data bus
	7		T	T	kept	I/O port
PF7/ø	4 to 6		Clock output	T	[DDR = 0] T [DDR = 1] H	[DDR = 0] T [DDR = 1] Clock output
	7		T	T	[DDR = 0] T [DDR = 1] H	[DDR = 0] T [DDR = 1] Clock output
PF6/AS	4 to 6		H	T	[OPE = 0] T [OPE = 1] H	AS
	7		T	T	kept	I/O port
PF5/RD PF4/HWR	4 to 6		H	T	[OPE = 0] T [OPE = 1] H	RD, HWR
	7		T	T	kept	I/O port

Port Name Pin Name	MCU Operating Mode	Reset	Hardware Standby Mode	Software Standby Mode	Program Execution State Sleep Mode
PF3	4 to 6	H	T	[OPE = 0] T [OPE = 1] H	LWR
	7	T	T	kept	I/O port
PF0	4 to 7	T	T	kept	I/O port
Port H	4 to 7	T	T	kept	I/O port
Port J	4 to 7	T	T	kept	I/O port
HTxD0, HTxD1	4 to 7	H	T	H	Tx output
HRxD0, HRxD1	4 to 7	Input	T	T	Rx output

Legend:

H : High level

L : Low level

T : High impedance

kept : Input port becomes high-impedance, output port retains state

DDR : Data direction register

OPE : Output port enable

Appendix E Timing of Transition to and Recovery from Hardware Standby Mode

Timing of Transition to Hardware Standby Mode

(1) To retain RAM contents with the RAME bit set to 1 in SYSCR, drive the $\overline{RES}$ signal low at least 10 states before the $\overline{STBY}$ signal goes low, as shown below. $\overline{RES}$ must remain low until $\overline{STBY}$ signal goes low (delay from $\overline{STBY}$ low to $\overline{RES}$ high: 0 ns or more).

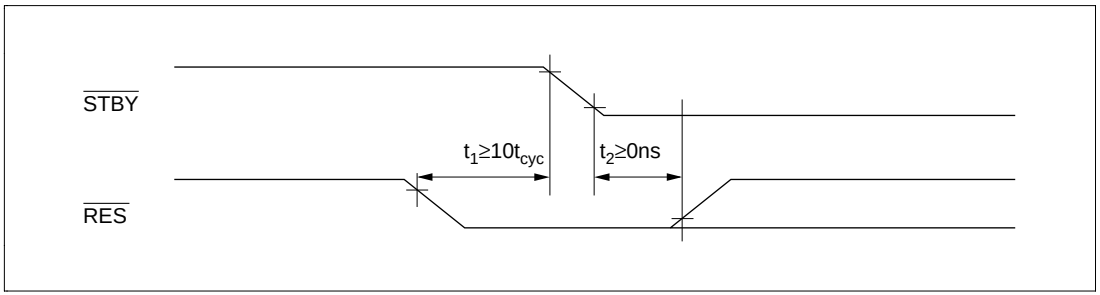


Figure E-1 Timing of Transition to Hardware Standby Mode

(2) To retain RAM contents with the RAME bit cleared to 0 in SYSCR, or when RAM contents do not need to be retained, $\overline{RES}$ does not have to be driven low as in (1).

Timing of Recovery from Hardware Standby Mode

Drive the $\overline{RES}$ signal low and the NMI signal high approximately 100 ns or more before $\overline{STBY}$ goes high to execute a reset.

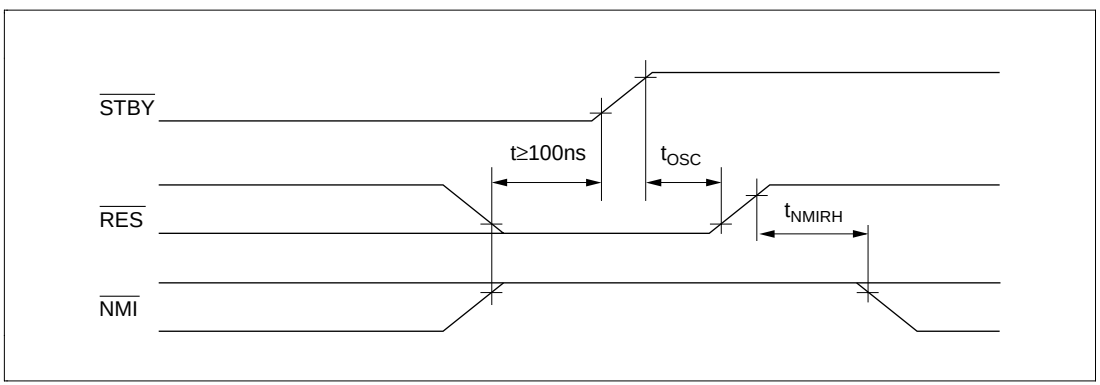


Figure E-2 Timing of Recovery from Hardware Standby Mode

Appendix F Product Code Lineup

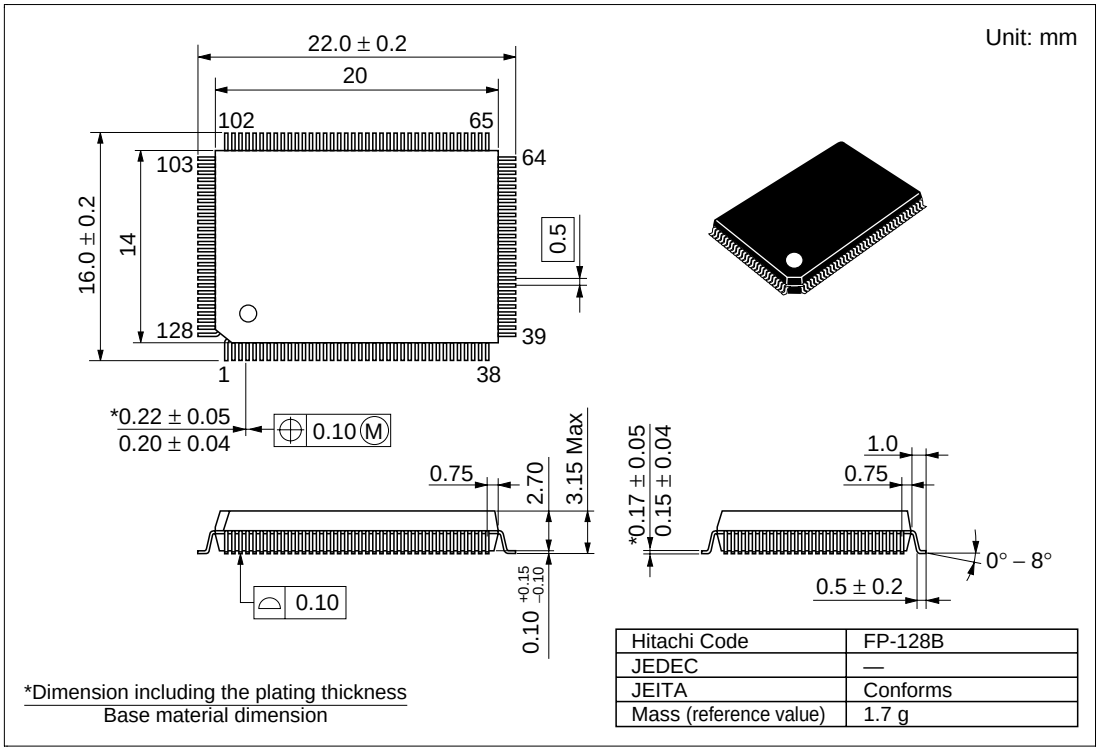
Table F-1 H8S/2636, H8S/2638 and H8S/2639 Product Code Lineup

Product Type		Product Code	Mark Code	Functions	Packages
H8S/2636	F-ZTAT version	HD64F2636	HD64F2636F	No subclock function	128-pin QFP (FP-128B)
			HD64F2636UF	Subclock function	128-pin QFP (FP-128B)
	Mask ROM version*	HD6432636	HD6432636F	No subclock function	128-pin QFP (FP-128B)
			HD6432636UF	Subclock function	128-pin QFP (FP-128B)
H8S/2638	F-ZTAT version	HD64F2638	HD64F2638F	No subclock function or I^2C bus interface	128-pin QFP (FP-128B)
			HD64F2638UF	Subclock function, no I^2C bus interface	128-pin QFP (FP-128B)
			HD64F2638WF	Subclock function and I^2C bus interface	128-pin QFP (FP-128B)
	Mask ROM version*	HD6432638	HD6432638F	No subclock function or I^2C bus interface	128-pin QFP (FP-128B)
			HD6432638UF	Subclock function, no I^2C bus interface	128-pin QFP (FP-128B)
			HD6432638WF	Subclock function and I^2C bus interface	128-pin QFP (FP-128B)
H8S/2639	F-ZTAT version	HD64F2639	HD64F2639UF	Subclock function, no I^2C bus interface	128-pin QFP (FP-128B)
			HD64F2639WF	Subclock function and I^2C bus interface	128-pin QFP (FP-128B)
	Mask ROM version*	HD6432639	HD6432639UF	Subclock function, no I^2C bus interface	128-pin QFP (FP-128B)
			HD6432639WF	Subclock function and I^2C bus interface	128-pin QFP (FP-128B)

Note: * Under planning

Appendix G Package Dimensions

Figure G-1 shows the package dimensions of the H8S/2636, H8S/2638 and H8S/2639.



Hitachi Code	FP-128B
JEDEC	—
JEITA	Conforms
Mass (reference value)	1.7 g

Figure G-1 FP-128B Package Dimensions

H8S/2639 Series, H8S/2638 Series, H8S/2636 Series Hardware Manual

Publication Date: 1st Edition, December 1999
3rd Edition, November 2001
Published by: Business Planning Division
Semiconductor & Integrated Circuits
Hitachi, Ltd.
Edited by: Technical Documentation Group
Hitachi Kodaira Semiconductor Co., Ltd.
Copyright © Hitachi, Ltd., 1999. All rights reserved. Printed in Japan.