User’s Manual

V850E/MS1™

32-/16-Bit Single-Chip Microcontrollers

Hardware

μPD703100
μPD703100A
μPD703101
μPD703101A
μPD703102
μPD703102A
μPD70F3102
μPD70F3102A

Document No. U12688EJ4V0UM00 (4th edition)
Date Published January 2000 N CP(K)

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

Note:

Strong electric field, when exposed to a MOS device, can cause destruction of the gate oxide and ultimately degrade the device operation. Steps must be taken to stop generation of static electricity as much as possible, and quickly dissipate it once, when it has occurred. Environmental control must be adequate. When it is dry, humidifier should be used. It is recommended to avoid using insulators that easily build static electricity. Semiconductor devices must be stored and transported in an anti-static container, static shielding bag or conductive material. All test and measurement tools including work bench and floor should be grounded. The operator should be grounded using wrist strap. Semiconductor devices must not be touched with bare hands. Similar precautions need to be taken for PW boards with semiconductor devices on it.

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note:

No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS devices behave differently than Bipolar or NMOS devices. Input levels of CMOS devices must be fixed high or low by using a pull-up or pull-down circuitry. Each unused pin should be connected to V_{DD} or GND with a resistor, if it is considered to have a possibility of being an output pin. All handling related to the unused pins must be judged device by device and related specifications governing the devices.

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

Note:

Power-on does not necessarily define initial status of MOS device. Production process of MOS does not define the initial operation status of the device. Immediately after the power source is turned ON, the devices with reset function have not yet been initialized. Hence, power-on does not guarantee out-pin levels, I/O settings or contents of registers. Device is not initialized until the reset signal is received. Reset operation must be executed immediately after power-on for devices having reset function.

The export of these products from Japan is regulated by the Japanese government. The export of some or all of these products may be prohibited without governmental license. To export or re-export some or all of these products from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

License not needed:	μPD703100, 703100A, 70F3102, 70F3102A
The customer must judge the need for license:	μPD703101, 703101A, 703102, 703102A

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 408-588-6000
800-366-9782
Fax: 408-588-6130
800-729-9288

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.l.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 91-504-2787
Fax: 91-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 65-253-8311
Fax: 65-250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-2719-2377
Fax: 02-2719-5951

NEC do Brasil S.A.
Electron Devices Division
Rodovia Presidente Dutra, Km 214
07210-902-Guarulhos-SP Brasil
Tel: 55-11-6465-6810
Fax: 55-11-6465-6829

J99.1

[MEMO]

Major Revisions in This Edition

Page	Description
p. 98	Change of R/W and bit units for manipulation for PMX and PMCX in **3.4.8 Peripheral I/O registers**
p. 108	Addition of Caution to **4.5.2 (1) Bus size configuration register (BSC)**
p. 151	Modification of $\overline{\text{WAIT}}$ signal in **Figure 5-10 DRAM Access Timing During DMA Flyby Transfer**
p. 172	Addition of interrupt factor (INTAD) to **6.3.6 DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)**
p. 235	Deletion of part of explanation from **8.5.1 (3) (a) When in the PLL mode**
p. 235	Deletion of **8.5.1 (3) (b) When in the Direct mode**
p. 326	Modification of **Figure 11-3 Select Mode Operation Timing: 1-Buffer Mode (ANI1)**
p. 349	Change of block type of Port 2 in **12.2 (1) Function of each port**
p. 355	Modification of **Figure 12-3 Type C Block Diagram**
p. 367	Addition of **Figure 12-17 Type Q Block Diagram**
p. 374	Change of block types of P22 and P25 in **12.3.3 (1) Operation in control mode**
p. 375	Modification of Caution in **12.3.3 (2) (a) Port 2 mode register (PM2)**
p. 378	Deletion of Caution from **12.3.4 (2) (a) Port 3 mode register (PM3)**
p. 398	Deletion of Caution from **12.3.12 (2) (a) Port 11 mode register (PM11)**
p. 407	Addition of Caution and modification of explanation in **12.3.16 (2) (a) Port X mode register (PMX)**
p. 408	Addition of Caution and modification of explanation in **12.3.16 (2) (b) Port X mode control register (PMCX)**

The mark ★ shows major revised points.

[MEMO]

INTRODUCTION

Readers

This manual is intended for users who wish to understand the functions of the V850E/MS1 (μPD703100, 703100A, 703101, 703101A, 703102, 703102A, 70F3102, 70F3102A) to design application systems using the V850E/MS1.

Purpose

This manual is designed to help users understand the hardware functions of the V850E/MS1.

Organization

The **V850E/MS1 User's Manual** consists of two manuals: Hardware (this manual) and Architecture **(V850E/MS1 User's Manual Architecture)**. The organization of each manual is as follows:

Hardware

- Pin functions
- CPU function
- Internal peripheral functions
- Flash memory programming

Architecture

- Data type
- Register set
- Instruction format and instruction set
- Interrupts and exceptions
- Pipeline operation

How to Read This Manual

It is assumed that the readers of this manual have general knowledge of electrical engineering, logic circuits, and microcontrollers.

- To find the details of a register where the name is known
  → Refer to **APPENDIX A REGISTER INDEX**.

- To find the details of a function, etc. where the name is known
  → Refer to **APPENDIX C INDEX**.

- To understand the details of an instruction function
  → Refer to the **V850E/MS1 User's Manual Architecture**.

- To understand the overall functions of the V850E/MS1
  → Read this manual in the order of the **CONTENTS**.

Conventions	Data significance:	Higher digits on the left and lower digits on the right
	Active low representation:	$\overline{xxx}$ (overscore over pin or signal name)
	Memory map address:	Higher address on the top and lower address on the bottom
	Note:	Footnote for item marked with **Note** in the text
	Caution:	Information requiring particular attention
	Remark:	Supplementary information
	Numerical representation:	Binary ... xxxx or xxxxB
		Decimal ... xxxx
		Hexadecimal ... xxxxH
	Prefix indicating power of 2	
	(address space, memory	K (kilo) ... 2^{10} = 1,024
	capacity):	M (mega) ... 2^{20} = $1,024^{2}$
		G (giga) ... 2^{30} = $1,024^{3}$
	Data type:	Word ... 32 bits
		Halfword ... 16 bits
		Byte ... 8 bits

Related Documents

The related documents indicated in this publication may include preliminary versions. However, preliminary versions are not marked as such.

Document related to device

Document Name	Document No.
μPD703100-33, 703100-40, 703101-33, 703102-33 Data Sheet	U13995E
μPD703100A-33, 703100A-40, 703101A-33, 703102A-33 Data Sheet	U14168E
μPD70F3102-33 Data Sheet	U13844E
μPD70F3102A-33 Data Sheet	U13845E
V850E/MS1 User's Manual Hardware	This manual
V850E/MS1 User's Manual Architecture	U12197E
V850E/MS1 Application Note Hardware	U14214E

Documents related to development tools (User's Manuals)

Document Name		Document No.
IE-703102-MC (In-circuit Emulator)		U13875E
IE-703102-MC-EM1, IE-703102-MC-EM1-A (In-circuit Emulator Option Board)		U13876E
CA850 (C Compiler Package)	Operation	U13998E
	C Language	U13997E
	Assembly Language	U13828E
	Project Manager	U13996E
RX850 (Real-Time OS)	Basics	U13430E
	Installation	U13410E
RX850 Pro (Real-Time OS)	Fundamental	U13773E
	Installation	U13774E
ID850 (Integrated Debugger) (Ver.1.31)	Operation Windows™ Based	U13716E
ID850 (Integrated Debugger) (Ver.2.00 or later)	Operation Windows Based	U14217E
SM850 (System Simulator) (Ver.2.00 or later)	Operation Windows Based	U13759E
RD850[Note] (Task Debugger)		U11158E
RD850 (Ver.3.0) (Task Debugger)		U13737E
RD850 Pro (Ver.3.0) (Task Debugger)		U13916E
AZ850 (System Performance Analyzer)		U14410E
PG-FP3 (Flash Memory Programmer)		U13502E

Note Supporting ID850 (Ver.1.32)

[MEMO]

CONTENTS

LIST OF FIGURES (1/4)

LIST OF FIGURES (2/4)

LIST OF FIGURES (3/4)

LIST OF FIGURES (4/4)

LIST OF TABLES (1/2)

LIST OF TABLES (2/2)

[MEMO]

CHAPTER 1 INTRODUCTION

The V850E/MS1 is one of NEC's "V850 Family™" of single-chip microcontrollers. This chapter gives a simple outline of the V850E/MS1.

1.1 Outline

The V850E/MS1 is a 32-/16-bit single-chip microcontroller which uses the V850 Family's "V850E" CPU, and incorporates peripheral functions such as ROM, RAM, various types of memory controllers, a DMA controller, real-time pulse unit, serial interface and A/D converter, realizing large volume data processing and sophisticated real-time control.

(1) "V850E" CPU included

The "V850E" CPU supports the RISC instruction set, and through the use of basic instructions, each of which can be executed in 1 clock period, and an optimized pipeline, achieves a marked improvement in instruction execution speed. In addition, in order to make it ideal for use in digital servo control, a 32-bit hardware multiplier enables this CPU to support multiply instructions, saturated multiply instructions, bit operation instructions, etc.

Also, through 2-byte basic instructions and instructions compatible with high level languages, etc., the object code efficiency in a C compiler is increased, and the program size can be made more compact.

Further, since the on-chip interrupt controller provides a high speed interrupt response, including processing, this device is suited to high level real-time control fields.

(2) External memory interface function

The V850E/MS1 features various on-chip external memory interfaces including separately address configured (24 bits) and data (16 bits) buses, and SRAM and ROM interfaces, as well as on-chip memory controllers that can be directly linked to EDO DRAM, high-speed page DRAM, page ROM, etc., thereby raising the system performance and reducing the number of parts needed for application systems.

Also, through the DMA controller, CPU internal calculations and data transfers can be performed simultaneously with transfers with external memory, so it is possible to process large volumes of image data or voice data, etc., and through the high-speed execution of instructions using internal ROM and RAM, motor control, communications control and other real-time control tasks can be realized simultaneously.

(3) On-chip flash memory (μPD70F3102, 70F3102A)

The on-chip flash memory model (μPD70F3102, 70F3102A) has on-chip flash memory which is capable of high speed access, and since it is possible to rewrite a program with the V850E/MS1 mounted as is in the application system, system development time can be reduced and system maintainability after shipping can be markedly improved.

(4) A full range of middleware and development environment products

The V850E/MS1 can execute middleware such as JPEG, JBIG and MH/MR/MMR at high speed. Also, middleware that enables voice recognition, voice synthesis and other such processing is available; by including these middleware programs, a multimedia system can be easily realized.

A development environment system that includes an optimized C compiler, debugger, in-circuit emulator, simulator, system performance analyzer and other elements is also available.

1.2 Features

○ Number of instructions:	81
○ Minimum instruction execution time:	25 ns (at internal 40 MHz) … μPD703100-40, 703100A-40 30 ns (at internal 33 MHz) … other than above
○ General registers:	32 bits × 32
○ Instruction set:	Upwardly compatible with V850 CPU Signed multiplication (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits): 1 to 2 clocks Saturated operation instructions (with overflow/underflow detection function) 32-bit shift instructions: 1 clock Bit manipulation instructions Load/store instructions with long/short format Signed load instructions
○ Memory space:	32 MB linear address space (common program/data use) Chip select output function: 8 spaces Memory block division function: 2, 4, 8 MB/block Programmable wait function Idle state insertion function
○ External bus interface:	16-bit data bus (address/data multiplexed) 16-/8-bit bus sizing function Bus hold function External wait function

○ Internal memory:

Part Number	Internal ROM	Internal RAM
μPD703100, 703100A	None	4 Kbytes
μPD703101, 703101A	96 Kbytes (Mask ROM)	4 Kbytes
μPD703102, 703102A	128 Kbytes (Mask ROM)	4 Kbytes
μPD70F3102, 70F3102A	128 Kbytes (Flash memory)	4 Kbytes

○ Interrupt/exception:	External interrupts: 25 (including NMI) Internal interrupts: 47 sources Exceptions: 1 source Eight levels of priorities can be set.
○ Memory access controller:	DRAM controller (Compatible with EDO DRAM and high-speed page DRAM) Page-ROM controller

○ DMA controller:	4 channels Transfer units: 8 bits/16 bits Maximum transfer count: 65,536 (2^{16}) Transfer type: Flyby (1-cycle)/2-cycle Transfer mode: Single/Single step/Block DMA transfer terminate (terminal count) output signal
○ I/O lines:	Input ports: 9 I/O ports: 114
○ Real-time pulse unit:	16-bit timer/event counter: 6 channels 16-bit timers: 6 16-bit capture/compare registers: 24 16-bit interval timer: 2 channels
○ Serial interface:	Asynchronous serial interface (UART) Clocked serial interface (CSI) UART/CSI: 2 channels CSI: 2 channels Dedicated baud rate generator: 3 channels
○ A/D converter:	10-bit resolution A/D converter: 8 channels
○ Clock generator:	A multiply-by-five function via a PLL clock synthesizer. A divide-by-two function via external clock input.
○ Power save function:	HALT/IDLE/software STOP mode Clock output stop function
○ Package:	144-pin plastic LQFP: pin pitch 0.5 mm
○ CMOS technology:	All static circuits

1.3 Applications

- OA devices (printers, facsimiles, PPCs, etc.)
- Multimedia devices (digital still cameras, video printers, etc.)
- Consumer appliances (single lens reflex cameras, etc.)
- Industrial devices (motor control, NC machine tools, etc.)

1.4 Ordering Information

Part Number	Package	Maximum Operating Frequency	On-chip ROM	HV_{DD}
μPD703100AF1-40-FA1[Note]	157-pin plastic FBGA (14 × 14 mm)	40 MHz	None	3.0 to 3.6 V
μPD703100AGJ-40-8EU[Note]	144-pin plastic LQFP (Fine pitch) (20 × 20 mm)	40 MHz	None	3.0 to 3.6 V
μPD703100GJ-40-8EU[Note]	144-pin plastic LQFP (Fine pitch) (20 × 20 mm)	40 MHz	None	4.5 to 5.5 V
μPD703100AF1-33-FA1[Note]	157-pin plastic FBGA (14 × 14 mm)	33 MHz	None	3.0 to 3.6 V
μPD703100AGJ-33-8EU	144-pin plastic LQFP (Fine pitch) (20 × 20 mm)	33 MHz	None	3.0 to 3.6 V
μPD703100GJ-33-8EU[Note]	144-pin plastic LQFP (Fine pitch) (20 × 20 mm)	33 MHz	None	4.5 to 5.5 V
μPD703101AF1-33-×××-FA1[Note]	157-pin plastic FBGA (14 × 14 mm)	33 MHz	Mask ROM (96 KB)	3.0 to 3.6 V
μPD703101AGJ-33-×××-8EU	144-pin plastic LQFP (Fine pitch) (20 × 20 mm)	33 MHz	Mask ROM (96 KB)	3.0 to 3.6 V
μPD703101GJ-33-×××-8EU[Note]	144-pin plastic LQFP (Fine pitch) (20 × 20 mm)	33 MHz	Mask ROM (96 KB)	4.5 to 5.5 V
μPD703102AF1-33-×××-FA1[Note]	157-pin plastic FBGA (14 × 14 mm)	33 MHz	Mask ROM (128 KB)	3.0 to 3.6 V
μPD703102AGJ-33-×××-8EU	144-pin plastic LQFP (Fine pitch) (20 × 20 mm)	33 MHz	Mask ROM (128 KB)	3.0 to 3.6 V
μPD703102GJ-33-×××-8EU[Note]	144-pin plastic LQFP (Fine pitch) (20 × 20 mm)	33 MHz	Mask ROM (128 KB)	4.5 to 5.5 V
μPD70F3102AF1-33-FA1[Note]	157-pin plastic FBGA (14 × 14 mm)	33 MHz	Flash memory (128 KB)	3.0 to 3.6 V
μPD70F3102AGJ-33-8EU[Note]	144-pin plastic LQFP (Fine pitch) (20 × 20 mm)	33 MHz	Flash memory (128 KB)	3.0 to 3.6 V
μPD70F3102GJ-33-8EU[Note]	144-pin plastic LQFP (Fine pitch) (20 × 20 mm)	33 MHz	Flash memory (128 KB)	4.5 to 5.5 V

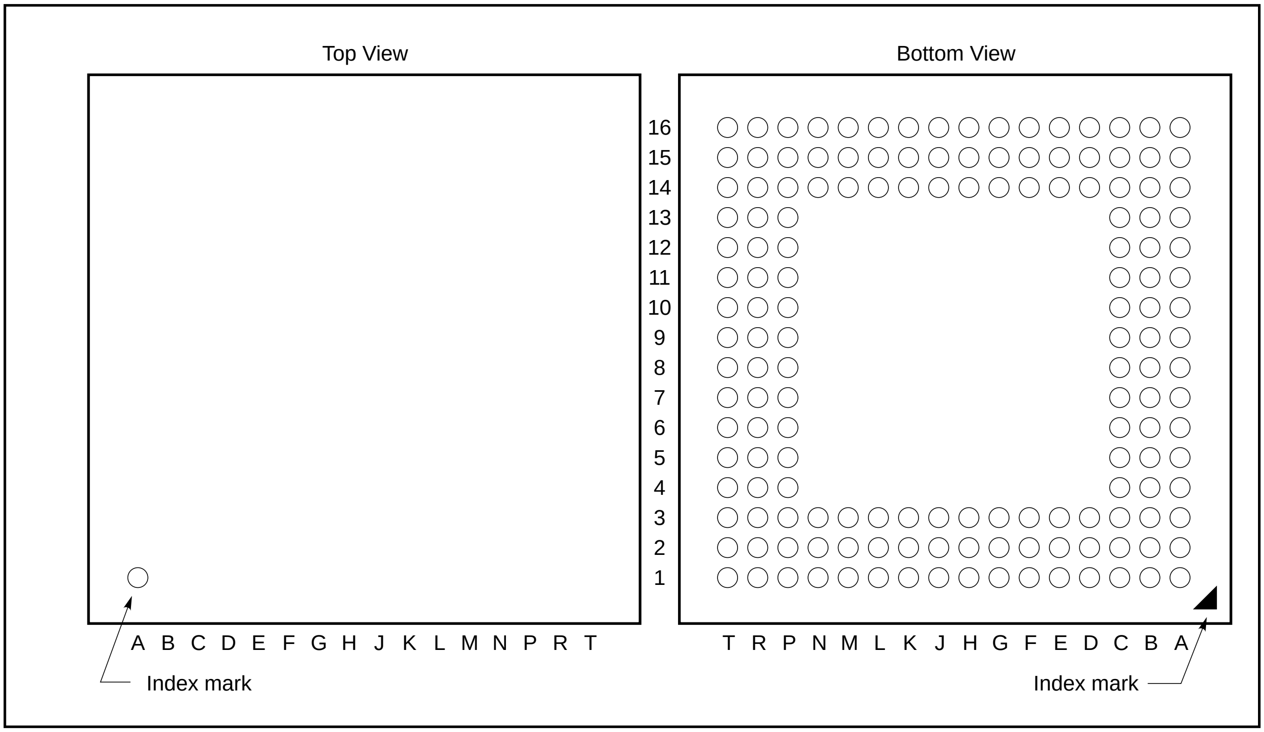
Note Under development

Remark ××× indicates ROM code suffix.

1.5 Pin Configuration (Top View)

157-pin plastic FBGA (14 × 14 mm)

- μPD703100AF1-40-FA1
- μPD703100AF1-33-FA1
- μPD703101AF1-33-×××-FA1
- μPD703102AF1-33-×××-FA1
- μPD70F3102AF1-33-FA1


Top View
Bottom View
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
A B C D E F G H J K L M N P R T
T R P N M L K J H G F E D C B A
Index mark
Index mark

(1/2)

Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
A1	—	B1	INTP103/$\overline{\text{DMARQ3}}$/P07	C1	INTP101/$\overline{\text{DMARQ1}}$/P05
A2	D0/P40	B2	D1/P41	C2	INTP102/$\overline{\text{DMARQ2}}$/P06
A3	D2/P42	B3	D3/P43	C3	V_{SS}
A4	D4/P44	B4	D5/P45	C4	V_{SS}
A5	D6/P46	B5	D7/P47	C5	HV_{DD}
A6	D8/P50	B6	D9/P51	C6	V_{SS}
A7	D10/P52	B7	D11/P53	C7	D12/P54
A8	D13/P55	B8	D14/P56	C8	D15/P57
A9	A0/PA0	B9	A1/PA1	C9	HV_{DD}
A10	A2/PA2	B10	A3/PA3	C10	A4/PA4
A11	A5/PA5	B11	A6/PA6	C11	A7/PA7
A12	A8/PB0	B12	A9/PB1	C12	V_{SS}
A13	A10/PB2	B13	A11/PB3	C13	A12/PB4
A14	A13/PB5	B14	A14/PB6	C14	A18/P62
A15	A15/PB7	B15	A17/P61	C15	A19/P63
A16	—	B16	A16/P60	C16	—

(2/2)

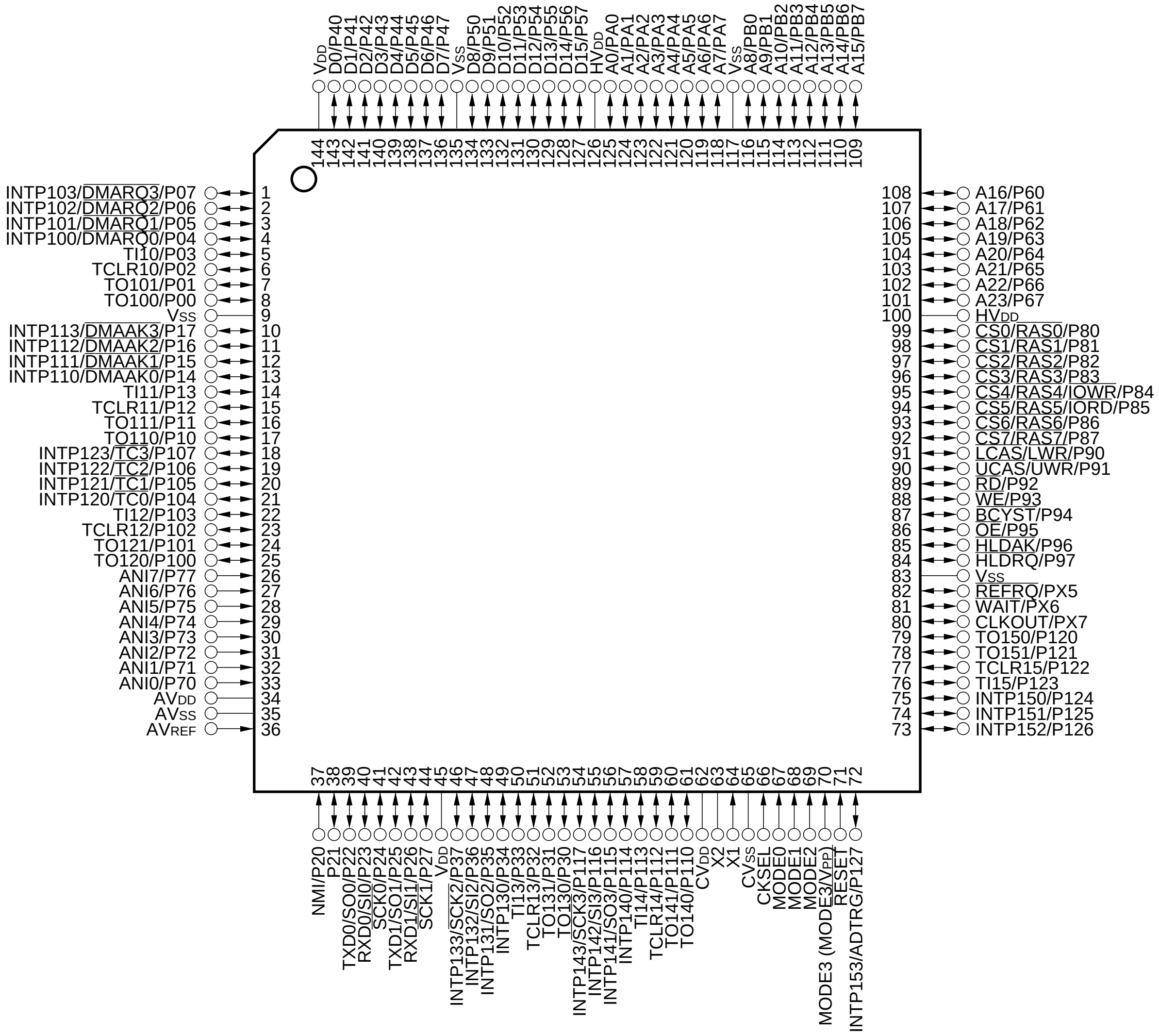
Pin Number	Pin Name	Pin Number	Pin Name	Pin Number	Pin Name
D1	TI10/P03	K1	TI12/P103	P14	$\overline{\text{RESET}}$
D2	INTP100/$\overline{\text{DMARQ0}}$/P04	K2	INTP120/$\overline{\text{TC0}}$/P104	P15	INTP151/P125
D3	HV_{DD}	K3	INTP121/$\overline{\text{TC1}}$/P105	P16	INTP150/P124
D4	—	K14	$\overline{\text{HLDAK}}$/P96	R1	AV_{SS}
D14	V_{SS}	K15	$\overline{\text{OE}}$/P95	R2	ANI0/P70
D15	A21/P65	K16	$\overline{\text{BCYST}}$/P94	R3	P21
D16	A20/P64	L1	TO120/P100	R4	$\overline{\text{SCK0}}$/P24
E1	TO101/P01	L2	TO121/P101	R5	$\overline{\text{SCK1}}$/P27
E2	TCLR10/P02	L3	TCLR12/P102	R6	INTP132/SI2/P36
E3	V_{SS}	L14	V_{SS}	R7	TI13/P33
E14	HV_{DD}	L15	$\overline{\text{REFRQ}}$/PX5	R8	TO130/P30
E15	A23/P67	L16	$\overline{\text{HLDRQ}}$/P97	R9	INTP141/SO3/P115
E16	A22/P66	M1	ANI5/P75	R10	TCLR14/P112
F1	INTP113/$\overline{\text{DMAAK3}}$/P17	M2	ANI6/P76	R11	TO140/P110
F2	TO100/P00	M3	ANI7/P77	R12	MODE0
F3	V_{DD}	M14	TO150/P120	R13	MODE1
F14	$\overline{\text{CS2}}$/$\overline{\text{RAS2}}$/P82	M15	$\overline{\text{WAIT}}$/PX6	R14	MODE2
F15	$\overline{\text{CS1}}$/$\overline{\text{RAS1}}$/P81	M16	CLKOUT/PX7	R15	INTP153/ADTRG/P127
F16	$\overline{\text{CS0}}$/$\overline{\text{RAS0}}$/P80	N1	ANI2/P72	R16	INTP152/P126
G1	INTP110/$\overline{\text{DMAAK0}}$/P14	N2	ANI3/P73	T1	—
G2	INTP111/$\overline{\text{DMAAK1}}$/P15	N3	ANI4/P74	T2	AV_{REF}
G3	INTP112/$\overline{\text{DMAAK2}}$/P16	N14	TI15/P123	T3	NMI/P20
G14	$\overline{\text{CS5}}$/$\overline{\text{RAS5}}$/$\overline{\text{IORD}}$/P85	N15	TCLR15/P122	T4	RXD0/SI0/P23
G15	$\overline{\text{CS4}}$/$\overline{\text{RAS4}}$/$\overline{\text{IOWR}}$/P84	N16	TO151/P121	T5	RXD1/SI1/P26
G16	$\overline{\text{CS3}}$/$\overline{\text{RAS3}}$/P83	P1	AV_{DD}	T6	INTP131/SO2/P35
H1	TO111/P11	P2	ANI1/P71	T7	TCLR13/P32
H2	TCLR11/P12	P3	TXD0/SO0/P22	T8	INTP143/$\overline{\text{SCK3}}$/P117
H3	TI11/P13	P4	TXD1/SO1/P25	T9	INTP140/P114
H14	$\overline{\text{LCAS}}$/$\overline{\text{LWR}}$/P90	P5	V_{DD}	T10	CV_{DD}
H15	$\overline{\text{CS7}}$/$\overline{\text{RAS7}}$/P87	P6	INTP133/$\overline{\text{SCK2}}$/P37	T11	X2
H16	$\overline{\text{CS6}}$/$\overline{\text{RAS6}}$/P86	P7	INTP130/P34	T12	X1
J1	INTP122/$\overline{\text{TC2}}$/P106	P8	TO131/P31	T13	CV_{SS}
J2	INTP123/$\overline{\text{TC3}}$/P107	P9	INTP142/SI3/P116	T14	MODE3 (MODE3/V_{PP})
J3	TO110/P10	P10	TI14/P113	T15	—
J14	$\overline{\text{WE}}$/P93	P11	TO141/P111	T16	—
J15	$\overline{\text{RD}}$/P92	P12	CKSEL	—	—
J16	$\overline{\text{UCAS}}$/$\overline{\text{UWR}}$/P91	P13	HV_{DD}	—	—

Remarks **1.** Leave the A1, A16, C16, D4, T1, T15, and T16 pins open.

2. Items in parentheses are pin names in the μPD70F3102, 70F3102A.

144-pin plastic LQFP (fine pitch) (20 × 20 mm)

- µPD703100GJ-40-8EU, 703100AGJ-40-8EU
- µPD703100GJ-33-8EU, 703100AGJ-33-8EU
- µPD703101GJ-33-×××-8EU, 703101AGJ-33-×××-8EU
- µPD703102GJ-33-×××-8EU, 703102AGJ-33-×××-8EU
- µPD70F3102GJ-33-8EU, 70F3102AGJ-33-8EU



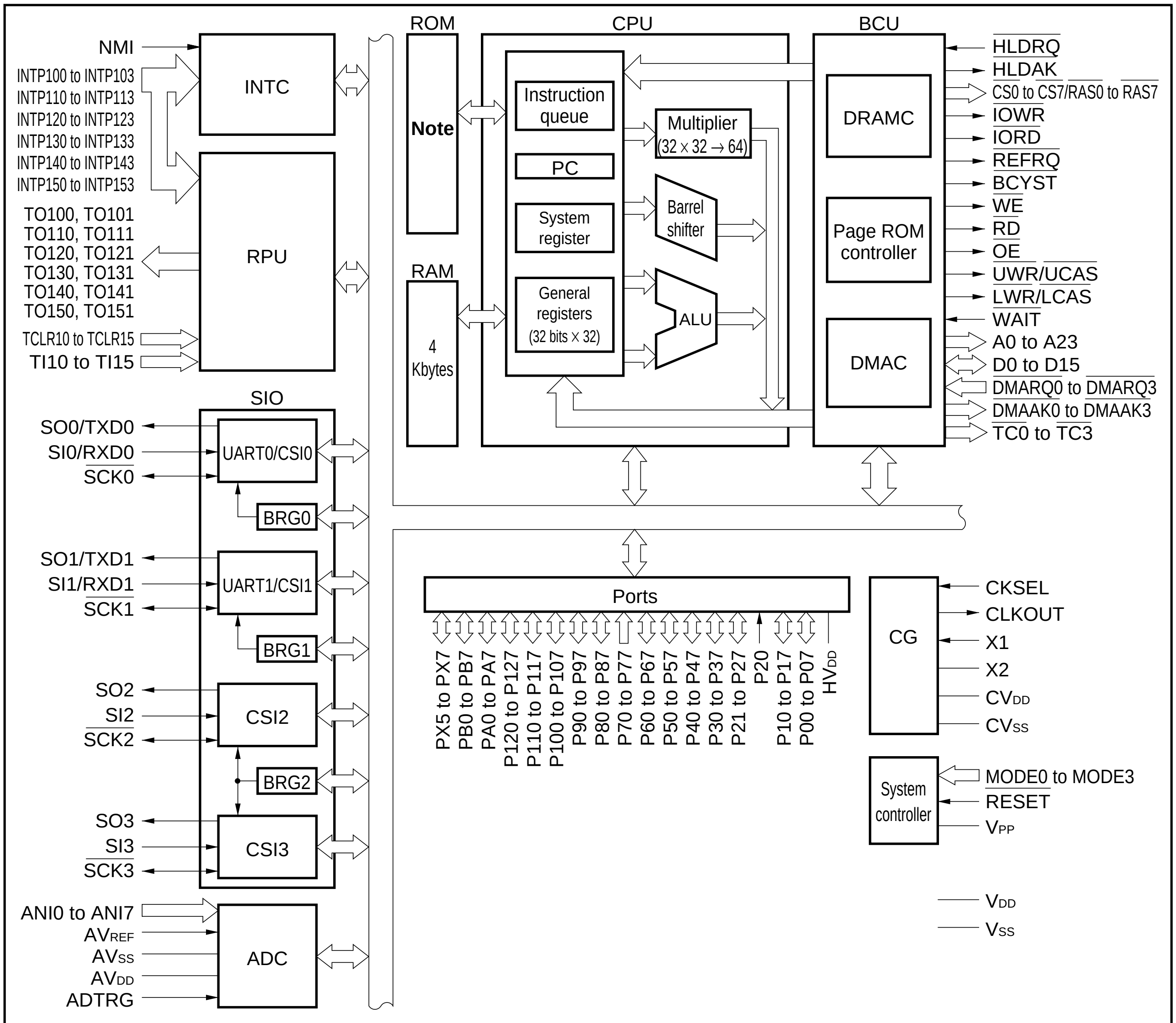
Remark Items in parentheses are pin names in the µPD70F3102, 70F3102A.

Pin Name

A0 to A23:	Address Bus	P60 to P67:	Port 6
ADTRG:	AD Trigger Input	P70 to P77:	Port 7
ANI0 to ANI7:	Analog Input	P80 to P87:	Port 8
AV_{DD}:	Analog Power Supply	P90 to P97:	Port 9
AV_{REF}:	Analog Reference Voltage	P100 to P107:	Port 10
AV_{SS}:	Analog Ground	P110 to P117:	Port 11
$\overline{\text{BCYST}}$:	Bus Cycle Start Timing	P120 to P127:	Port 12
CKSEL:	Clock Generator Operating Mode Select	PA0 to PA7:	Port A
CLKOUT:	Clock Output	PB0 to PB7:	Port B
$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$:	Chip Select	PX5 to PX7:	Port X
CV_{DD}:	Clock Generator Power Supply	$\overline{\text{RAS0}}$ to $\overline{\text{RAS7}}$:	Row Address Strobe
CV_{SS}:	Clock Generator Ground	$\overline{\text{RD}}$:	Read
D0 to D15:	Data Bus	$\overline{\text{REFRQ}}$:	Refresh Request
$\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$:	DMA Acknowledge	$\overline{\text{RESET}}$:	Reset
$\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$:	DMA Request	RXD0, RXD1:	Receive Data
$\overline{\text{HLDAK}}$:	Hold Acknowledge	$\overline{\text{SCK0}}$ to $\overline{\text{SCK3}}$:	Serial Clock
$\overline{\text{HLDRQ}}$:	Hold Request	SI0 to SI3:	Serial Input
HV_{DD}:	Power Supply for External Pins	SO0 to SO3:	Serial Output
INTP100 to INTP103,		$\overline{\text{TC0}}$ to $\overline{\text{TC3}}$:	Terminal Count Signal
INTP110 to INTP113,		TCLR10 to TCLR15:	Timer Clear
INTP120 to INTP123,		TI10 to TI15:	Timer Input
INTP130 to INTP133,		TO100, TO101,	
INTP140 to INTP143,		TO110, TO111,	
INTP150 to INTP153:	Interrupt Request from Peripherals	TO120, TO121,	
$\overline{\text{IORD}}$:	I/O Read Strobe	TO130, TO131,	
$\overline{\text{IOWR}}$:	I/O Write Strobe	TO140, TO141,	
$\overline{\text{LCAS}}$:	Lower Column Address Strobe	TO150, TO151:	Timer Output
$\overline{\text{LWR}}$:	Lower Write Strobe	TXD0, TXD1:	Transmit Data
MODE0 to MODE3:	Mode	$\overline{\text{UCAS}}$:	Upper Column Address Strobe
NMI:	Non-Maskable Interrupt Request	$\overline{\text{UWR}}$:	Upper Write Strobe
$\overline{\text{OE}}$:	Output Enable	V_{DD}:	Power Supply for Internal Unit
P00 to P07:	Port 0	V_{PP}:	Programming Power Supply
P10 to P17:	Port 1	V_{SS}:	Ground
P20 to P27:	Port 2	$\overline{\text{WAIT}}$:	Wait
P30 to P37:	Port 3	$\overline{\text{WE}}$:	Write Enable
P40 to P47:	Port 4	X1, X2:	Crystal
P50 to P57:	Port 5		

1.6 Function Block

1.6.1 Internal block diagram



Note μPD703100, 703100A: None

μPD703101, 703101A: 96 Kbytes (Mask ROM)

μPD703102, 703102A: 128 Kbytes (Mask ROM)

μPD70F3102, 70F3102A: 128 Kbytes (Flash memory)

1.6.2 Internal units

(1) CPU

The CPU uses five-stage pipeline control to enable single-clock execution of address calculations, arithmetic logic operations, data transfers, and almost all other instruction processing.

Other dedicated on-chip hardware, such as a multiplier (16 bits × 16 bits → 32 bits or 32 bits × 32 bits → 64 bits) and a barrel shifter (32 bits), help accelerate processing of complex instructions.

(2) Bus control unit (BCU)

The BCU starts a required external bus cycle based on the physical address obtained by the CPU. When an instruction is fetched from external memory space and the CPU does not send a bus cycle start request, the BCU generates a prefetch address and prefetches the instruction code. The prefetched instruction code is stored in an instruction queue in the CPU.

The BCU incorporates a DRAM controller (DRAMC), page ROM controller, and DMA controller (DMAC).

(a) DRAM controller (DRAMC)

This controller generates the $\overline{\text{RAS}}$, $\overline{\text{UCAS}}$ and $\overline{\text{LCAS}}$ signals (2CAS control) and controls DRAM access.

It is compatible with high-speed DRAM and EDO DRAM. When accessing DRAM, there are 2 types of cycle; normal access (off page) and page access (on page).

Also, it includes a refresh function that is compatible with the CBR refresh cycle.

(b) Page ROM controller

This controller is compatible with ROM that includes a page access function.

It performs address comparisons with the immediately preceding bus cycle and executes wait control for normal access (off page)/page access (on page). It can handle page widths of 8 to 64 bytes.

(c) DMA controller (DMAC)

This controller transfers data between memory and I/O in place of the CPU.

There are two address modes, flyby (1 cycle) transfer, and 2-cycle transfer. There are three bus modes, single transfer, single step transfer, and block transfer.

(3) ROM

The μPD703101 and 703101A have on-chip mask ROM (96 KB), the μPD703102 and 703102A have on-chip mask ROM (128 KB), and the μPD70F3102 and 70F3102A have on-chip flash memory (128 KB). The μPD703100 and 703100A do not include on-chip memory.

During instruction fetch, these memories can be accessed from the CPU in 1 clock cycles.

If the single-chip mode 0 or flash memory programming mode is set, memory mapping is done from address 00000000H, and if single-chip mode 1 is set, from address 00100000H. If ROM-less mode 0 or 1 is set, access is impossible.

(4) RAM

4 KB of RAM is mapped from address FFFFE000H. During instruction fetch, data can be accessed from the CPU in 1-clock cycles.

(5) Interrupt controller (INTC)

This controller handles hardware interrupt requests (NMI, INTP100 to INTP103, INTP110 to INTP113, INTP120 to INTP123, INTP130 to INTP133, INTP140 to INTP143, INTP150 to INTP153) from internal peripheral I/O and external hardware. Eight levels of interrupt priorities can be specified for these interrupt requests, and multiplexed servicing control can be performed for interrupt sources.

(6) Clock generator (CG)

This clock generator supplies frequencies that are 5 times the input clock (fxx) (used by the internal PLL) and 1/2 the input clock (when the internal PLL is not used) as an internal system clock (ϕ). As the input clock, an external oscillator is connected to pins X1 and X2 (only when an internal PLL synthesizer is used) or an external clock is input from pin X1.

(7) Real-time pulse unit (RPU)

This unit has a 6-channel 16-bit timer/event counter and 2-channel 16-bit interval timer on-chip, and it is possible to measure pulse widths or frequency and to output a programmable pulse.

(8) Serial interface (SIO)

The serial interface has a total of 4 channels of asynchronous serial interfaces (UART) and synchronous or clocked serial interfaces (CSI). Two of these channels can be switched between UART and CSI, and the other two channels are fixed to CSI.

UART transfers data by using the TXD and RXD pins and the CSI transfers data by using the SO, SI, and $\overline{\text{SCK}}$ pins.

The serial clock source can be selected from dedicated baud rate generator output or internal system clock.

(9) A/D converter (ADC)

This high-speed, high-resolution 10-bit A/D converter includes 8 analog input pins. Conversion uses the successive approximation method.

(10) Ports

As shown below, the following ports have general port functions and control pin functions.

Port	Port Function	Control Function
Port 0	8-bit I/O	Real-time pulse unit input/output, external interrupt input, DMA controller input
Port 1	8-bit I/O	Real-time pulse unit input/output, external interrupt input, DMA controller output
Port 2	1-bit input, 7-bit I/O	NMI input, serial interface input/output
Port 3	8-bit I/O	Real-time pulse unit input/output, external interrupt input, serial interface input/output
Port 4	8-bit I/O	External data bus
Port 5	8-bit I/O	External data bus
Port 6	8-bit I/O	External address bus
Port 7	8-bit input	A/D converter input
Port 8	8-bit I/O	External bus interface control signal output
Port 9	8-bit I/O	External bus interface control signal input/output
Port 10	8-bit I/O	Real-time pulse unit input/output, external interrupt input, DMA controller output
Port 11	8-bit I/O	Real-time pulse unit input/output, external interrupt input, serial interface input/output
Port 12	8-bit I/O	Real-time pulse unit input/output, external interrupt input, A/D converter external trigger input
Port A	8-bit I/O	External address bus
Port B	8-bit I/O	External address bus
Port X	3-bit I/O	Refresh request signal output, wait insertion signal input, internal system clock output

CHAPTER 2 PIN FUNCTIONS

The names and functions of this product's pins are listed below. These pins can be divided into port pins and non-port pins according to their functions.

2.1 List of Pin Functions

(1) Port pins (1/4)

Pin Name	I/O	Function	Alternate Function
P00	I/O	Port 0 8-bit input/output port Input/output mode can be specified in 1-bit units.	TO100
P01			TO101
P02			TCLR10
P03			TI10
P04			INTP100/$\overline{\text{DMARQ0}}$
P05			INTP101/$\overline{\text{DMARQ1}}$
P06			INTP102/$\overline{\text{DMARQ2}}$
P07			INTP103/$\overline{\text{DMARQ3}}$
P10	I/O	Port 1 8-bit input/output port Input/output mode can be specified in 1-bit units.	TO110
P11			TO111
P12			TCLR11
P13			TI11
P14			INTP110/$\overline{\text{DMAAK0}}$
P15			INTP111/$\overline{\text{DMAAK1}}$
P16			INTP112/$\overline{\text{DMAAK2}}$
P17			INTP113/$\overline{\text{DMAAK3}}$
P20	Input	Port 2 P20 is an input-only port. If a valid edge is input, it operates as an NMI input. Also, the status of the NMI input is shown by bit 0 of the P2 register. P21 to P27 are 7-bit input/output ports. Input/output mode can be specified in 1-bit units.	NMI
P21	I/O		—
P22			TXD0/SO0
P23			RXD0/SI0
P24			$\overline{\text{SCK0}}$
P25			TXD1/SO1
P26			RXD1/SI1
P27			$\overline{\text{SCK1}}$

(1) Port pins (2/4)

Pin Name	I/O	Function	Alternate Function
P30	I/O	Port 3 8-bit input/output port Input/output mode can be specified in 1-bit units.	TO130
P31			TO131
P32			TCLR13
P33			TI13
P34			INTP130
P35			INTP131/SO2
P36			INTP132/SI2
P37			INTP133/$\overline{\text{SCK2}}$
P40 to P47	I/O	Port 4 8-bit input/output port Input/output mode can be specified in 1-bit units.	D0 to D7
P50 to P57	I/O	Port 5 8-bit input/output port Input/output mode can be specified in 1-bit units.	D8 to D15
P60 to P67	I/O	Port 6 8-bit input/output port Input/output mode can be specified in 1-bit units.	A16 to A23
P70 to P77	Input	Port 7 8-bit input only port	ANI0 to ANI7
P80	I/O	Port 8 8-bit input/output port Input/output mode can be specified in 1-bit units.	$\overline{\text{CS0}}$/$\overline{\text{RAS0}}$
P81			$\overline{\text{CS1}}$/$\overline{\text{RAS1}}$
P82			$\overline{\text{CS2}}$/$\overline{\text{RAS2}}$
P83			$\overline{\text{CS3}}$/$\overline{\text{RAS3}}$
P84			$\overline{\text{CS4}}$/$\overline{\text{RAS4}}$/$\overline{\text{IOWR}}$
P85			$\overline{\text{CS5}}$/$\overline{\text{RAS5}}$/$\overline{\text{IORD}}$
P86			$\overline{\text{CS6}}$/$\overline{\text{RAS6}}$
P87			$\overline{\text{CS7}}$/$\overline{\text{RAS7}}$
P90	I/O	Port 9 8-bit input/output port Input/output mode can be specified in 1-bit units.	$\overline{\text{LCAS}}$/$\overline{\text{LWR}}$
P91			$\overline{\text{UCAS}}$/$\overline{\text{UWR}}$
P92			$\overline{\text{RD}}$
P93			$\overline{\text{WE}}$
P94			$\overline{\text{BCYST}}$
P95			$\overline{\text{OE}}$
P96			$\overline{\text{HLDAK}}$
P97			$\overline{\text{HLDRQ}}$

(1) Port pins (3/4)

Pin Name	I/O	Function	Alternate Function
P100	I/O	Port 10 8-bit input/output port Input/output mode can be specified in 1-bit units.	TO120
P101			TO121
P102			TCLR12
P103			TI12
P104			INTP120/$\overline{\text{TC0}}$
P105			INTP121/$\overline{\text{TC1}}$
P106			INTP122/$\overline{\text{TC2}}$
P107			INTP123/$\overline{\text{TC3}}$
P110	I/O	Port 11 8-bit input/output port Input/output mode can be specified in 1-bit units.	TO140
P111			TO141
P112			TCLR14
P113			TI14
P114			INTP140
P115			INTP141/SO3
P116			INTP142/SI3
P117			INTP143/$\overline{\text{SCK3}}$
P120	I/O	Port 12 8-bit input/output port Input/output mode can be specified in 1-bit units.	TO150
P121			TO151
P122			TCLR15
P123			TI15
P124			INTP150
P125			INTP151
P126			INTP152
P127			INTP153/ADTRG
PA0	I/O	Port A 8-bit input/output port Input/output mode can be specified in 1-bit units.	A0
PA1			A1
PA2			A2
PA3			A3
PA4			A4
PA5			A5
PA6			A6
PA7			A7

(1) Port pins (4/4)

Pin Name	I/O	Function	Alternate Function
PB0	I/O	Port B 8-bit input/out port Input/output mode can be specified in 1-bit units.	A8
PB1			A9
PB2			A10
PB3			A11
PB4			A12
PB5			A13
PB6			A14
PB7			A15
PX5	I/O	Port X 3-bit input/output port Input/output mode can be specified in 1-bit units.	$\overline{\text{REFRQ}}$
PX6			$\overline{\text{WAIT}}$
PX7			CLKOUT

(2) Non-port pins (1/4)

Pin Name	I/O	Function	Alternate Function
TO100	Output	Pulse signal output of timers 10 to 15	P00
TO101			P01
TO110			P10
TO111			P11
TO120			P100
TO121			P101
TO130			P30
TO131			P31
TO140			P110
TO141			P111
TO150			P120
TO151			P121
TCLR10	Input	External clear signal input of timers 10 to 15	P02
TCLR11			P12
TCLR12			P102
TCLR13			P32
TCLR14			P112
TCLR15			P122
TI10	Input	External count clock input of timers 10 to 15	P03
TI11			P13
TI12			P103
TI13			P33
TI14			P113
TI15			P123
INTP100	Input	External maskable interrupt request input, or timer 10 external capture trigger input	P04/$\overline{\text{DMARQ0}}$
INTP101			P05/$\overline{\text{DMARQ1}}$
INTP102			P06/$\overline{\text{DMARQ2}}$
INTP103			P07/$\overline{\text{DMARQ3}}$
INTP110	Input	External maskable interrupt request input, or timer 11 external capture trigger input	P14/$\overline{\text{DMAAK0}}$
INTP111			P15/$\overline{\text{DMAAK1}}$
INTP112			P16/$\overline{\text{DMAAK2}}$
INTP113			P17/$\overline{\text{DMAAK3}}$
INTP120	Input	External maskable interrupt request input, or timer 12 external capture trigger input	P104/$\overline{\text{TC0}}$
INTP121			P105/$\overline{\text{TC1}}$
INTP122			P106/$\overline{\text{TC2}}$
INTP123			P107/$\overline{\text{TC3}}$

(2) Non-port pins (2/4)

Pin Name	I/O	Function	Alternate Function
INTP130	Input	External maskable interrupt request input, or timer 13 external capture trigger input	P34
INTP131			P35/SO2
INTP132			P36/SI2
INTP133			P37/$\overline{\text{SCK2}}$
INTP140	Input	External maskable interrupt request input, or timer 14 external capture trigger input	P114
INTP141			P115/SO3
INTP142			P116/SI3
INTP143			P117/$\overline{\text{SCK3}}$
INTP150	Input	External maskable interrupt request input, or timer 15 external capture trigger input	P124
INTP151			P125
INTP152			P126
INTP153			P127/ADTRG
SO0	Input	CSI0 to CSI3 serial transmission data output (3-wire)	P22/TXD0
SO1			P25/TXD1
SO2			P35/INTP131
SO3			P115/INTP141
SI0	Input	CSI0 to CSI3 serial reception data input (3-wire)	P23/RXD0
SI1			P26/RXD1
SI2			P36/INTP132
SI3			P116/INTP142
$\overline{\text{SCK0}}$	I/O	CSI0 to CSI3 serial clock input/output (3-wire)	P24
$\overline{\text{SCK1}}$			P27
$\overline{\text{SCK2}}$			P37/INTP133
$\overline{\text{SCK3}}$			P117/INTP143
TXD0	Output	UART0 and UART1 serial transmission data output	P22/SO0
TXD1			P25/SO1
RXD0	Input	UART0 and UART1 serial reception data input	P23/SI0
RXD1			P26/SI1
D0 to D7	I/O	16-bit data bus for external memory	P40 to P47
D8 to D15			P50 to P57
A0 to A7	Output	24-bit address bus for external memory	PA0 to PA7
A8 to A15			PB0 to PB7
A16 to A23			P60 to P67
$\overline{\text{LWR}}$	Output	External data bus lower byte write enable signal output	P90/$\overline{\text{LCAS}}$
$\overline{\text{UWR}}$	Output	External data bus higher byte write enable signal output	P91/$\overline{\text{UCAS}}$
$\overline{\text{RD}}$	Output	External data bus read strobe signal output	P92

(2) Non-port pins (3/4)

Pin Name	I/O	Function	Alternate Function
$\overline{\text{WE}}$	Output	Write enable signal output for DRAM	P93
$\overline{\text{OE}}$	Output	Output enable signal output for DRAM	P95
$\overline{\text{LCAS}}$	Output	Column address strobe signal output for DRAM lower data	P90/$\overline{\text{LWR}}$
$\overline{\text{UCAS}}$	Output	Column address strobe signal output for DRAM higher data	P91/$\overline{\text{UWR}}$
$\overline{\text{RAS0}}$ to $\overline{\text{RAS3}}$	Output	Row address strobe signal output for DRAM	P80/$\overline{\text{CS0}}$ to P83/$\overline{\text{CS3}}$
$\overline{\text{RAS4}}$			P84/$\overline{\text{CS4}}$/$\overline{\text{IOWR}}$
$\overline{\text{RAS5}}$			P85/$\overline{\text{CS5}}$/$\overline{\text{IORD}}$
$\overline{\text{RAS6}}$			P86/$\overline{\text{CS6}}$
$\overline{\text{RAS7}}$			P87/$\overline{\text{CS7}}$
$\overline{\text{BCYST}}$	Output	Strobe signal output that shows the start of the bus cycle	P94
$\overline{\text{CS0}}$ to $\overline{\text{CS3}}$	Output	Chip select signal output	P80/$\overline{\text{RAS0}}$ to P83/$\overline{\text{RAS3}}$
$\overline{\text{CS4}}$			P84/$\overline{\text{RAS4}}$/$\overline{\text{IOWR}}$
$\overline{\text{CS5}}$			P85/$\overline{\text{RAS5}}$/$\overline{\text{IORD}}$
$\overline{\text{CS6}}$			P86/$\overline{\text{RAS6}}$
$\overline{\text{CS7}}$			P87/$\overline{\text{RAS7}}$
$\overline{\text{WAIT}}$	Input	Control signal input that inserts a wait in the bus cycle	PX6
$\overline{\text{REFRQ}}$	Output	Refresh request signal output for DRAM	PX5
$\overline{\text{IOWR}}$	Output	DMA write strobe signal output	P84/$\overline{\text{RAS4}}$/$\overline{\text{CS4}}$
$\overline{\text{IORD}}$	Output	DMA read strobe signal output	P85/$\overline{\text{RAS5}}$/$\overline{\text{CS5}}$
$\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$	Input	DMA request signal input	P04/INTP100 to P07/INTP103
$\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$	Output	DMA acknowledge signal output	P14/INTP110 to P17/INTP113
$\overline{\text{TC0}}$ to $\overline{\text{TC3}}$	Output	DMA termination (terminal count) signal output	P104/INTP120 to P107/INTP123
$\overline{\text{HLDAK}}$	Output	Bus hold acknowledge output	P96
$\overline{\text{HLDRQ}}$	Input	Bus hold request input	P97
ANI0 to ANI7	Input	Analog inputs to the A/D converter	P70 to P77
NMI	Input	Non-maskable interrupt request input	P20
CLKOUT	Output	System clock output	PX7
CKSEL	Input	Input which specifies the clock generator's operating mode	—
MODE0 to MODE2	Input	Operation mode specification	—
MODE3			V_{PP}[Note]

Note μPD70F3102 and 70F3102A only

(2) Non-port pins (4/4)

Pin Name	I/O	Function	Alternate Function
$\overline{\text{RESET}}$	Input	System reset input	—
X1	Input	Connects the system clock oscillator. In the case of an external source supplying the clock, it is input to X1.	—
X2	—		—
ADTRG	Input	A/D converter external trigger input	P127/INTP153
AV$_{\text{REF}}$	Input	Reference voltage applied to A/D converter	—
AV$_{\text{DD}}$	—	Positive power supply to A/D converter	—
AV$_{\text{SS}}$	—	Ground for A/D converter	—
CV$_{\text{DD}}$	—	Supplies a positive power supply for the dedicated clock generator.	—
CV$_{\text{SS}}$	—	Ground potential for the dedicated clock generator	—
V$_{\text{DD}}$	—	Supplies the positive power supply (internal unit power supply).	—
HV$_{\text{DD}}$	—	Supplies the positive power supply (external pin power supply).	—
V$_{\text{SS}}$	—	Ground potential	—
V$_{\text{PP}}$[Note]	—	High-voltage application pin during program write/verify	MODE3

Note μPD70F3102 and 70F3102A only

2.2 Pin Status

The state of each pin after reset, in a power save mode (software STOP, IDLE, HALT), during bus hold (TH), and in the idle state (TI), is shown below.

Operating State Pin	Reset	Software STOP Mode	IDLE Mode	HALT Mode	Bus Hold (TH)	Idle State (TI)
D0 to D15	Hi-Z	HI-Z (output) — (input)	HI-Z (output) — (input)	Operating	Hi-Z	Hi-Z
A0 to A23	Hi-Z	Hi-Z	Hi-Z	Operating	Hi-Z	Hold
$\overline{\text{WE}}$, $\overline{\text{OE}}$, $\overline{\text{RD}}$, $\overline{\text{BCYST}}$	Hi-Z	Hi-Z	Hi-Z	Operating	Hi-Z	H
$\overline{\text{UWR}}$, $\overline{\text{LWR}}$, $\overline{\text{IORD}}$, $\overline{\text{IOWR}}$, $\overline{\text{CS0}}$ to $\overline{\text{CS7}}$	Hi-Z	H	H	Operating	Hi-Z	H
$\overline{\text{RAS0}}$ to $\overline{\text{RAS7}}$	Hi-Z	Operating	Operating	Operating	Hi-Z	Hold[Note 2]
$\overline{\text{UCAS}}$, $\overline{\text{LCAS}}$	Hi-Z	Operating	Operating	Operating	Hi-Z	H
$\overline{\text{REFRQ}}$	Hi-Z	Operating	Operating	Operating	Operating	H
$\overline{\text{HLDRQ}}$	—	—	—	Operating	Operating	Operating
$\overline{\text{HLDAK}}$	Hi-Z	Hi-Z	Hi-Z	Operating	L	Operating
$\overline{\text{WAIT}}$	—	—	—	Operating	—	—
CLKOUT	**Note 1**	L	L	Operating	Operating	Operating
$\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$	—	—	—	Operating	Operating	Operating
$\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$	Hi-Z	H	H	Operating	H	H
$\overline{\text{TC0}}$ to $\overline{\text{TC3}}$	Hi-Z	H	H	Operating	Operating	Operating
INTP100 to INTP103, INTP110 to INTP113, INTP120 to INTP123, INTP130 to INTP133, INTP140 to INTP143, INTP150 to INTP153	—	—	—	Operating	Operating	Operating
NMI	—	Operating	Operating	Operating	Operating	Operating
P00 to P07, P10 to P17, P20 to P27, P30 to P37, P40 to P47, P50 to P57, P60 to P67, P70 to P77, P80 to P87, P90 to P97, P100 to P107, P110 to P117, P120 to P127, PA0 to PA7, PB0 to PB7, PX5 to PX7	Hi-Z	Hold (output) — (input)	Hold (output) — (input)	Operating	Operating	Operating
TCLR10 to TCLR15	—	—	—	Operating	Operating	Operating
TI10 to TI15	—	—	—	Operating	Operating	Operating
TO100, TO101, TO110, TO111, TO120, TO121, TO130, TO131, TO140, TO141, TO150, TO151	Hi-Z	Hold	Hold	Operating	Operating	Operating

Operating State Pin	Reset	Software STOP Mode	IDLE Mode	HALT Mode	Bus Hold (TH)	Idle State (TI)
SI0 to SI3	—	—	—	Operating	Operating	Operating
SO0 to SO3	Hi-Z	Hold	Hold	Operating	Operating	Operating
$\overline{\text{SCK0}}$ to $\overline{\text{SCK3}}$	Hi-Z	Hold (output) — (input)	Hold (output) — (input)	Operating	Operating	Operating
RXD0, RXD1	—	—	—	Operating	Operating	Operating
TXD0, TXD1	Hi-Z	Hold	Hold	Operating	Operating	Operating
ANI0 to ANI7, ADTRG	—	—	—	Operating	Operating	Operating

Notes 1. When in single-chip mode 0: Hi-Z
At other times: Operating

2. In the idle state (TI) just before and just after bus hold, H

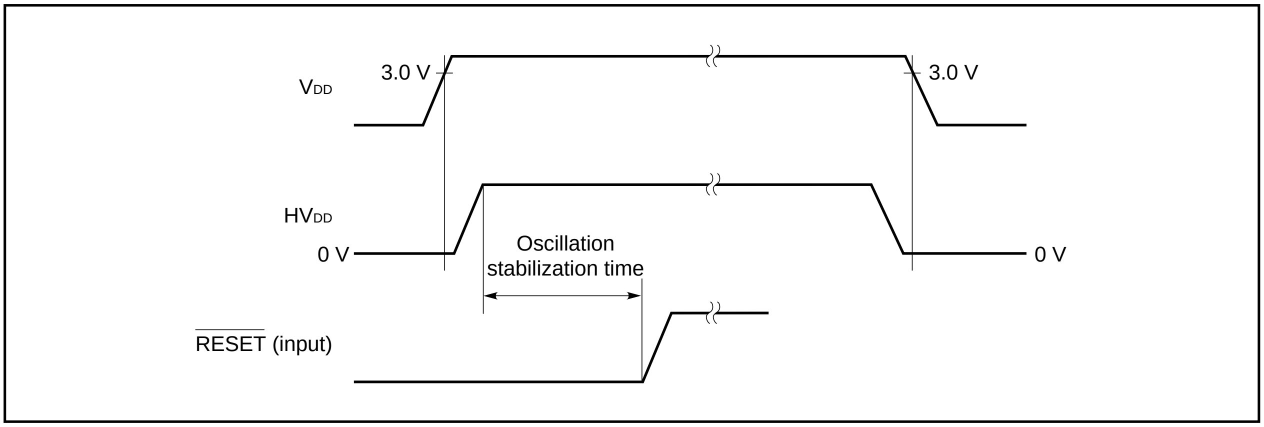
Remark Hi-Z: High-impedance
Hold: State during immediately preceding external bus cycle is held
H: High-level output
L: Low-level output
—: No sampling of input

Cautions when turning on/off power supply

The V850E/MS1 is configured with two power supply pins: the internal unit power supply pin (V_{DD}) and the external pin power supply pin (HV_{DD}). If the voltage exceeds its operation guaranteed range, the input/output state of the I/O pins may become undefined. If this input/output undefined state causes problems in the system, the pin status can be made high impedance by taking the following countermeasures.

- **When turning on the power**
  Apply 0 V to the HV_{DD} pin until the voltage of the V_{DD} pin is within the operation guaranteed range (3.0 to 3.6 V).

- **When turning off the power**
  Apply a voltage within the operation guaranteed range (3.0 to 3.6 V) to the V_{DD} pin until the voltage of the HV_{DD} pin becomes 0 V.



2.3 Description of Pin Functions

(1) P00 to P07 (Port 0) ··· 3-state I/O

Port 0 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode it operates as the input/output for the real-time pulse unit (RPU), the external interrupt request input and the DMA request input.

The operation mode can be set as port or control in 1-bit units, specified by the port 0 mode control register (PMC0).

(a) Port mode

P00 to P07 can be set to input or output in bit units by the port 0 mode register (PM0).

(b) Control mode

P00 to P07 can be set in the port/control mode in bit units by the PMC0 register.

(i) TO100, TO101 (Timer Output) ··· output

Output the pulse signals for timer 1.

(ii) TCLR10 (Timer Clear) ··· input

This is an input pin for external clear signals for timer 1.

(iii) TI10 (Timer Input) ··· input

This is an input pin for an external counter clock for timer 1.

(iv) INTP100 to INTP103 (Interrupt Request from Peripherals) ··· input

These are input pins for external interrupt requests for timer 1.

(v) $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$ (DMA Request) ··· input

These are DMA service request signals. They correspond to DMA channels 0 to 3, respectively, and operate independently of each other. The priority order is fixed at $\overline{\text{DMARQ0}}$ > $\overline{\text{DMARQ1}}$ > $\overline{\text{DMARQ2}}$ > $\overline{\text{DMARQ3}}$.

This signal is sampled when the CLKOUT signal falls. Maintain the active level until a DMA request is received.

(2) P10 to P17 (Port 1) ··· 3-state I/O

Port 1 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode it operates as the input/output for the real-time pulse unit (RPU), the external interrupt request input and the DMA request input.

The operation mode can be set as port or control in 1-bit units, specified by the port 1 mode control register (PMC1).

(a) Port mode

P10 to P17 can be set to input or output in bit units by the port 1 mode register (PM1).

(b) Control Mode

P10 to P17 can be set in the port/control mode in bit units by the PMC1 register.

(i) TO110, TO111 (Timer Output) ··· output

Output the pulse signals for timer 1.

(ii) TCLR11 (Timer Clear) ··· input

This is an input pin for external clear signals for timer 1.

(iii) TI11 (Timer Input) ··· input

This is an input pin for an external counter clock for timer 1.

(iv) INTP110 to INTP113 (Interrupt Request from Peripherals) ··· input

These are input pins for external interrupt requests for timer 1.

(v) $\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$ (DMA Acknowledge) ··· output

This signal shows that a DMA service request was acknowledged.

They correspond to DMA channels 0 to 3, respectively, and operate independently of each other.

These signals become active only when external memory is being accessed. When DMA transfers are being executed between internal RAM and internal peripheral I/O, they do not become active.

These signals are activated on the falling of the CLKOUT signal in the T0, T1R, or T1FH state of the DMA cycle, and are retained at the active level during DMA transfers.

(3) P20 to P27 (Port 2) ··· 3-state I/O

Port 2, except for P20, which is an input-only pin, is an input/output port which can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode it operates as the input/output for the serial interface (UART0/CSI0, UART1/CST1).

The operation mode can be set as port or control in 1-bit units, specified by the port 2 mode control register (PMC2).

(a) Port mode

P21 to P27 can be set to input or output in bit units by the port 2 mode register (PM2). P20 is an exclusive input port, and if a valid edge is input, it operates as an NMI input.

(b) Control mode

P22 to P27 can be set in the port/control mode in bit units by the PMC2 register.

(i) NMI (Non-Maskable Interrupt Request) ··· input

This is the input pin for non-maskable interrupt requests.

(ii) TXD0, TXD1 (Transmit Data) ··· output

Output UART0, UART1 serial transmit data.

(iii) RXD0, RXD1 (Receive Data) ··· input

Input UART0, UART1 serial receive data.

(iv) SO0, SO1 (Serial Output) ··· output

Output CSI0, CSI1 serial transmit data.

(v) SI0, SI1 (Serial Input) ··· input

Input CSI0, CSI1 serial receive data.

(vi) $\overline{\text{SCK0}}$, $\overline{\text{SCK1}}$ (Serial Clock) ··· 3-state I/O

These are the input/output pins for the CSI0, CSI1 serial clock.

(4) P30 to P37 (Port 3) ··· 3-state I/O

Port 3 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode it operates as the input/output for the real-time pulse unit (RPU), the external request input and the serial interface (CSI2) input/output. The operation mode can be set as port or control in 1-bit units, specified by the port 3 mode control register (PMC3).

(a) Port mode

P33 to P37 can be set to input or output in bit units by the port 3 mode register (PM3).

(b) Control mode

P30 to P37 can be set in the port/control mode in bit units by the PMC3 register.

(i) TO130, TO131 (Timer Output) ··· output

Output pulse signals for timer 1.

(ii) TCLR13 (Timer Clear) ··· input

This is an input pin for external clear signals for timer 1.

(iii) TI13 (Timer Input) ··· input

This is an input pin for an external counter clock for timer 1.

(iv) INTP130 to INTP133 (Interrupt Request from Peripherals) ··· input

These are input pins for external interrupt requests for timer 1.

(v) SO2 (Serial Output)··· output

Outputs CSI2 serial transmit data.

(vi) SI2 (Serial Input)··· input

Inputs CSI2 serial receive data.

(vii) $\overline{\text{SCK2}}$ (Serial Clock)··· 3-state I/O

This is the input/output pin for the CSI2 serial clock.

(5) P40 to P47 (Port 4) ··· 3-state I/O

Port 4 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode (external expansion mode) it operates as a data bus (D0 to D7) when memory is externally expanded.

The operation mode is specified by the mode specification pins (MODE0 to MODE3) and the memory expansion mode register (MM).

(a) Port mode

P40 to P47 can be set to input or output in bit units by the port 4 mode register (PM4).

(b) Control mode (External expansion mode)

P40 to P47 can be set as D0 to D7 by using the MODE0 to MODE3 pins and MM register.

(i) D0 to D7 (Data) ··· 3-state I/O

These pins constitute the data bus that is used for external access. They operate as the lower 8-bit input/output bus pins for 16-bit data. The output changes in synchronization with the falling of the clock in the T1 state CLKOUT signal of the bus cycle. In the idle state (TI), the impedance becomes high.

(6) P50 to P57 (Port 5) ··· 3-state I/O

Port 5 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as an I/O port, in the control mode (external expansion mode) it operates as a data bus (D8 to D15) when memory is externally expanded.

The operation mode is specified by the mode specification pins (MODE0 to MODE3) and the memory expansion mode register (MM).

(a) Port mode

P50 to P57 can be set to input or output in bit units by the port 5 mode register (PM5).

(b) Control mode (External expansion mode)

P50 to P57 can be set as D8 to D15 by using the MODE0 to MODE3 pins and MM register.

(i) D8 to D15 (Data) ··· 3-state I/O

These pins constitute the data bus that is used for external access. They operate as the higher 8-bit input/output bus pins for 16-bit data. The output changes in synchronization with the falling of the clock in the T1 state CLKOUT signal of the bus cycle. In the idle state (TI), the impedance becomes high.

(7) P60 to P67 (Port 6) ··· 3-state I/O

Port 6 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode (external expansion mode) it operates as an address bus (A16 to A23) when memory is externally expanded.

The operation mode can be set as port or control in 2-bit units, specified by the mode specification pins (MODE0 to MODE3) and the memory expansion mode register (MM).

(a) Port mode

P60 to P67 can be set to input or output in bit units by the port 6 mode register (PM6).

(b) Control mode (External expansion mode)

P60 to P67 can be set as A16 to A23 by using the MODE0 to MODE3 pins and MM register.

(i) A16 to A23 (Address) ··· output

These pins constitute the higher 8-bits of a 24-bits address bus when the external memory is accessed. The output changes in synchronization with the falling edge of the CLKOUT signal in the T1 state of the bus cycle. In the idle state (TI), the previous bus cycle’s address is held.

(8) P70 to P77 (Port 7) ··· input

Port 7 is an 8-bit input-only port in which all pins are fixed as input pins.

Besides functioning as a port, in the control mode it operates as analog input for the A/D converter. However, the input port and analog input pin cannot be switched.

(a) Port mode

P70 to P77 are input-only pins.

(b) Control mode

P70 to P77 function alternately pins ANI0 to ANI7, but these alternate functions are not switchable.

(i) ANI0 to ANI7 (Analog Input) ··· input

These are analog input pins for the A/D converter.

Connect a capacitor between these pins and AV_{SS} to prevent noise-related operation faults. Also, do not apply voltage that is outside the range for AV_{SS} and AV_{REF} to pins that are being used as inputs for the A/D converter. If it is possible for noise above the AV_{REF} range or below the AV_{SS} to enter, clamp these pins using a diode that has a small V_F value.

(9) P80 to P87 (Port 8) ··· 3-state I/O

Port 8 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode it operates as a control signal output when memory and peripheral I/O are externally expanded.

The operation mode can be set as port or control in 1-bit units, specified by the port 8 mode control register (PMC8).

(a) Port mode

P80 to P87 can be set to input or output in bit units by the port 8 mode register (PM8).

(b) Control mode

P80 to P87 can be set in the port/control mode in bit units by the PMC8 register.

(i) $\overline{\text{CS0}}$ to $\overline{\text{CS7}}$ (Chip Select) ··· 3-state output

This is the chip select signal for SRAM, external ROM, external peripheral I/O, page ROM and the synchronous flash memory area.

The $\overline{\text{CSn}}$ signal is assigned to memory block n (n = 0 to 7).

It becomes active at the time the bus cycle when the corresponding memory block is accessed starts.

In the idle state (TI), it becomes inactive.

(ii) $\overline{\text{RAS0}}$ to $\overline{\text{RAS7}}$ (Row Address Strobe) ··· 3-state output

This is the strobe signal for the row address for the DRAM area and the strobe signal for the CBR refresh cycle.

The $\overline{\text{RASn}}$ signal is assigned to memory block n (n = 0 to 7).

During on-page disable, after the DRAM access bus cycle ends, it becomes inactive.

During on-page enable, even after the DRAM access bus cycle ends, it is kept in the active state.

During the reset period and during a hold period, it is in the high impedance state, so connect it to HV$_{DD}$ via a resistor.

(iii) $\overline{\text{IORD}}$ (I/O Read) ··· 3-state output

This is the read strobe signal for external I/O during DMA flyby transfer. It indicates whether the bus cycle currently being executed is a read cycle for external I/O during flyby transfer, or a read cycle for the SRAM area.

In order to make it possible to connect directly to memory or external I/O during DMA flyby transfer, $\overline{\text{UWR}}$ or $\overline{\text{LWR}}$ rises before $\overline{\text{IORD}}$ rises.

Furthermore, this external I/O can be accessed even when it is assigned to the SRAM area.

(iv) $\overline{\text{IOWR}}$ (I/O Write) ··· 3-state output

This is the write strobe signal for external I/O during DMA flyby transfer. It indicates whether the bus cycle currently being executed is a write cycle for external I/O during flyby transfer, or a write cycle for the SRAM area.

In order to make it possible to connect directly to memory or external I/O during DMA flyby transfer, $\overline{\text{IOWR}}$ rises before $\overline{\text{RD}}$ rises.

Furthermore, this external I/O can be accessed even when it is assigned to the SRAM area.

(10) P90 to P97 (Port 9) ··· 3-state I/O

Port 9 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode it operates as a control signal output and bus hold control signal input/output when memory is externally expanded.

The operation mode can be set as port or control in 1-bit units, specified by the port 9 mode control register (PMC9).

(a) Port mode

P90 to P97 can be set to input or output in bit units by the port 9 mode register (PM9).

(b) Control mode

P90 to P97 can be set in the port/control mode in bit units by the PMC9 register.

(i) $\overline{\text{LCAS}}$ (Lower Column Address Strobe) ··· 3-state output

This is the strobe signal for column address for DRAM and the strobe signal for the CBR refresh cycle.

In the data bus, the lower byte is valid.

(ii) $\overline{\text{UCAS}}$ (Upper Column Address Strobe) ··· 3-state output

This is the strobe signal for column address for DRAM and the strobe signal for the CBR refresh cycle.

In the data bus, the higher byte is valid.

(iii) $\overline{\text{LWR}}$ (Lower Byte Write Strobe) ··· 3-state output

This strobe signal shows whether the bus cycle currently being executed is a write cycle for the SRAM, external ROM, external peripheral I/O, or page ROM.

In the data bus, the lower byte becomes valid. If the bus cycle is a lower memory write, it becomes active at the rise of the T1 state's CLKOUT signal and becomes inactive at the rise of the T2 state's CLKOUT signal.

(iv) $\overline{\text{UWR}}$ (Upper Byte Write Strobe) ··· 3-state output

This strobe signal shows whether the bus cycle currently being executed is a write cycle for the SRAM, external ROM, external peripheral I/O, or page ROM.

In the data bus, the higher byte becomes valid. If the bus cycle is a higher memory write, it becomes active at the rise of the T1 state's CLKOUT signal and becomes inactive at the rise of the T2 state's CLKOUT signal.

(v) $\overline{\text{RD}}$ (Read Strobe) ··· 3-state output

This strobe signal shows that the bus cycle currently being executed is a read cycle for the SRAM, external ROM, external peripheral I/O, page ROM or synchronous flash memory area.

In the idle state (TI), it becomes inactive.

(vi) $\overline{\text{WE}}$ (Write Enable) ··· 3-state output

This signal shows that the bus cycle currently being executed is a write cycle for the SRAM area. In the idle state (TI), it becomes inactive.

(vii) $\overline{\text{BCYST}}$ (Bus Cycle Start Timing) ··· 3-state output

This outputs a status signal showing the start of the bus cycle. It becomes active for 1 clock cycle from the start of each cycle.

In the idle state (TI), it becomes inactive.

(viii) $\overline{\text{OE}}$ (Output Enable) ··· 3-state output

This signal shows that the bus cycle currently being executed is a read cycle for the DRAM area.

In the idle state (TI), it becomes inactive.

(ix) $\overline{\text{HLDAK}}$ (Hold Acknowledge) ··· output

In this mode, this pin is the output pin for the acknowledge signal that indicates high impedance status for the address bus, data bus, and control bus when the V850E/MS1 receives a bus hold request.

While this signal is active, the impedance of the address bus, data bus and control bus becomes high and the bus mastership is transferred to the external bus master.

(x) $\overline{\text{HLDRQ}}$ (Hold Request) ··· input

In this mode, this pin is the input pin by which an external device requests the V850E/MS1 to release the address bus, data bus, and control bus. This pin accepts asynchronous input for the CLKOUT signal. When this pin is active, the address bus, data bus, and control bus are set to high impedance. This occurs either when the V850E/MS1 completes execution of the current bus cycle or immediately if no bus cycle is being executed, then the $\overline{\text{HLDAK}}$ signal is set as active and the bus is released.

In order to make the bus hold state secure, keep the $\overline{\text{HLDRQ}}$ signal active until the $\overline{\text{HLDAK}}$ signal is output.

(11) P100 to P107 (Port 10) ··· 3-state I/O

Port 10 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode it operates as an input/output for real time pulse unit (RPU), external interrupt request input and DMA termination signal (terminal count) from DMA controller.

The operation mode can be set as port or control in 1-bit units, specified by the port 10 mode control register (PMC10).

(a) Port mode

P100 to P107 can be set to input or output in bit units by the port 10 mode register (PM10).

(b) Control mode

P100 to P107 can be set in the port/control mode in bit units by the PMC10 register.

(i) TO120, TO121 (Timer Output) ··· output

Output the pulse signal of timer 1.

(ii) TCLR12 (Timer Clear) ··· input

This is an input pin for external clear signals for timer 1.

(iii) TI12 (Timer Input) ··· input

This is an input pin for an external counter clock for timer 1.

(iv) INTP120 to INTP123 (Interrupt Request from Peripherals) ··· input

These are input pins for external interrupt requests for timer 1.

(v) $\overline{\text{TC0}}$ to $\overline{\text{TC3}}$ (Terminal Count) ··· output

This signal shows that DMA transfer by the DMA controller is terminated.

This signal becomes active for 1 clock cycle at the fall of the CLKOUT signal.

(12) P110 to P117 (Port 11) ··· 3-state I/O

Port 11 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode it operates as an input/output for real-time pulse unit (RPU), external interrupt request, input and serial interface (CSI3) input/output.

The operation mode can be set as port or control in 1-bit units, specified by the port 11 mode control register (PMC11).

(a) Port mode

P110 to P117 can be set to input or output in bit units by the port 11 mode register (PM11).

(b) Control mode

P110 to P117 can be set in the port/control mode in bit units by the PMC11 register.

(i) TO140, TO141 (Timer Output) ··· output

Output the pulse signal of timer 1.

(ii) TCLR14 (Timer Clear) ··· input

This is an input pin for external clear signals for timer 1.

(iii) TI14 (Timer Input) ··· input

This is an input pin for an external counter clock for timer 1.

(iv) INTP140 to INTP143 (Interrupt Request from Peripherals) ··· input

These are input pins for external interrupt requests for timer 1.

(v) SO3 (Serial Output 3)··· output

Outputs the CSI3 serial transfer data.

(vi) SI3 (Serial Input 3)··· input

Inputs the CSI3 serial receive data.

(vii) $\overline{\text{SCK3}}$ (Serial Clock 3)··· 3-state I/O

This is the input/output pin for the CSI3 serial clock.

(13) P120 to P127 (Port 12) ··· 3-state I/O

Port 12 is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode it operates as an input/output for real-time pulse unit (RPU), external interrupt request input and external trigger input to A/D converter.

The operation mode can be set as port or control in 1-bit units, specified by the port 12 mode control register (PMC12).

(a) Port mode

P120 to P127 can be set to input or output in bit units by the port 12 mode register (PM12).

(b) Control mode

P120 to P127 can be set in the port/control mode in bit units by the PMC12 register.

(i) TO150, TO151 (Timer Output) ··· output

Output the pulse signal of timer 1.

(ii) TCLR15 (Timer Clear) ··· input

This is an input pin for external clear signals for timer 1.

(iii) TI15 (Timer Input) ··· input

This is an input pin for an external counter clock for timer 1.

(iv) INTP150 to INTP153 (Interrupt Request from Peripherals) ··· input

These are input pins for external interrupt requests for timer 1.

(v) ADTRG (AD Trigger Input)··· input

This is the A/D converter external trigger input pin.

(14) PA0 to PA7 (Port A) ··· 3-state I/O

Port A is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode (external expansion mode) it operates as an address bus (A0 to A7) when memory is externally expanded.

The operation mode is specified by the mode specification pins (MODE0 to MODE3) and the memory expansion mode register (MM).

(a) Port mode

PA0 to PA7 can be set to input or output in bit units by the port A mode register (PMA).

(b) Control mode (External expansion mode)

PA0 to PA7 can be set as A0 to A7 by using the MODE0 to MODE3 pins and MM register.

(i) A0 to A7 (Address) ··· output

These pins constitute the address bus that is used for external access. The output changes in synchronization with the falling of the CLKOUT signal in the T1 state of the bus cycle. In the idle state (TI), the previous bus cycle's address is held.

(15) PB0 to PB7 (Port B) ··· 3-state I/O

Port B is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode (external expansion mode) it operates as an address bus (A8 to A15) when memory is externally expanded.

The operation mode can be set as port or control in 2-bit or 4-bit units, specified by the mode specification pins (MODE0 to MODE3) and memory expansion mode register (MM).

(a) Port mode

PB0 to PB7 can be set to input or output in bit units by the port B mode register (PMB).

(b) Control mode (External expansion mode)

PB0 to PB7 can be set as A8 to A15 by using the MODE0 to MODE3 pins and MM register.

(i) A8 to A15 (Address) ··· output

These pins constitute the address bus when the external memory is accessed.

The output changes in synchronization with the rising edge of the CLKOUT signal in the T1 state of the bus cycle. In the idle state (TI), the impedance becomes high.

(16) PX5 to PX7 (Port X) ··· 3-state I/O

Port X is an 8-bit input/output port that can be set to input or output in 1-bit units.

Besides functioning as a port, in the control mode it operates as a refresh request signal output for DRAM, wait insertion signal input and system clock output.

The operation mode can be set as port or control in 1-bit units, specified by the port X mode control register (PMCX).

(a) Port mode

PX5 to PX7 can be set to input or output in bit units by the port X mode register (PMX).

(b) Control mode

PX5 to PX7 can be set in the port/control mode in bit units by the PMCX register.

(i) $\overline{\text{REFRQ}}$ (Refresh Request) ··· 3-state output

This is the refresh request signal for DRAM.

In cases where the address is decoded by an external circuit and the connected DRAM is increased, or in cases where external SIMMs are connected, this signal is used for $\overline{\text{RAS}}$ control during the refresh cycle.

This signal becomes active during the refresh cycle. Also, during bus hold, it becomes active when a refresh request is generated and informs the external bus master that a refresh request was generated.

(ii) $\overline{\text{WAIT}}$ (Wait) ··· input

This is the control signal input pin that inserts a data wait in the bus cycle, and it can be input asynchronously with respect to the CLKOUT signal. When the CLKOUT signal falls, sampling is executed. When the set/hold time is not terminated within the sampling timing, the wait insertion may not be executed.

(iii) CLKOUT (Clock Output) ··· output

This is the internal system clock output pin. When in single-chip mode 1 and ROM-less modes 0 and 1, output from the CLKOUT pin can be executed even during reset.

When in single-chip mode 0, it changes to the port mode during reset, so output from the CLKOUT pin cannot be executed. Set the port X mode control register (PMCX) to control mode to execute CLKOUT output.

(17) CKSEL (Clock Generator Operating Mode Select) ··· input

This is the input pin that specifies the clock generator’s operation mode.

Make sure the input level does not change during operation.

(18) MODE0 to MODE3 (Mode) ··· input

These are the input pins that specify the operation mode. Operation modes can be roughly divided into normal operation mode and flash memory programming mode. In the normal operation mode, there are single-chip modes 0 and 1, and ROM-less modes 0 and 1 (for details, refer to **3.3 Operation Modes**). The operation mode is determined by sampling the status of each of the MODE0 to MODE3 pins during reset.

Note that this status must be fixed so that the input level does not change during operation.

(a) μPD703100, 703100A

MODE3	MODE2	MODE1	MODE0	Operation Mode	
L	L	L	L	Normal operation mode	ROM-less mode 0
L	L	L	H		ROM-less mode 1
Other than above				Setting prohibited	

(b) μPD703101, 703101A, 703102, 703102A

MODE3	MODE2	MODE1	MODE0	Operation Mode	
L	L	L	L	Normal operation mode	ROM-less mode 0
L	L	L	H		ROM-less mode 1
L	L	H	L		Single-chip mode 0
L	L	H	H		Single-chip mode 1
Other than above				Setting prohibited	

(c) μPD70F3102, 70F3102A

MODE3/VPP	MODE2	MODE1	MODE0	Operation Mode	
0 V	L	L	L	Normal operation mode	ROM-less mode 0
0 V	L	L	H		ROM-less mode 1
0 V	L	H	L		Single-chip mode 0
0 V	L	H	H		Single-chip mode 1
7.8 V	L	H	L	Flash memory programming mode	
Other than above				Setting prohibited	

Remark L: Low-level input
H: High-level input

(19) $\overline{\text{RESET}}$ (Reset) ··· input

$\overline{\text{RESET}}$ input is asynchronous input for a signal that has a constant low-level width regardless of the operating clock's status. When this signal is input, a system reset is executed as the first priority ahead of all other operations.

In addition to being used for ordinary initialization/start operations, this pin can also be used to release a power save mode (HALT, IDLE, or software STOP).

(20) X1, X2 (Crystal) ··· input

These pins are used to connect the resonator that generates the system clock.

An external clock source can be referenced by connecting the external clock input to the X1 pin and leaving the X2 pin open.

(21) CV_{DD} (Power Supply for Clock Generator)

This pin supplies positive power to the clock generator.

(22) CV_{SS} (Ground for Clock Generator)

This is the ground pin of the clock generator.

(23) V_{DD} (Power Supply for Internal Unit)

These are the positive power supply pins for each internal unit. All the V_{DD} pins should be connected to a positive power source (3.3 V).

(24) HV_{DD} (Power Supply for External Pins)

These are the positive power supply pins for external pins. All the HV_{DD} pins should be connected to a positive power source (5 V to 3.3 V).

(25) V_{SS} (Ground)

These are ground pins. All the V_{SS} pins should be connected to ground.

(26) AV_{DD} (Analog V_{DD})

This is the analog power supply pin for the A/D converter.

(27) AV_{SS} (Analog V_{SS})

This is the ground pin for the A/D converter.

(28) AV_{REF} (Analog Reference Voltage) ··· input

This is the reference voltage supply pin for the A/D converter.

(29) V_{PP} (Programming Power Supply)

This is the positive power supply pin used for flash memory programming mode.

This pin is used for μPD70F3102 and 70F3102A.

2.4 Pin Input/Output Circuits and Recommended Connection of Unused Pins

If connecting to V_{DD} or V_{SS} via resistors, it is recommended that 1 to 10 kΩ resistors be connected.

Pin Name	Input/Output Circuit Type	Recommended Connection of Unused Pins
P00/TO100, P01/TO101	5	Input: Independently connect to HV_{DD} or V_{SS} via a resistor. Output: Leave open.
P02,TCLR10, P03/TI10	5-K	
P04/INTP100/$\overline{\text{DMARQ0}}$ to P07/INTP103/$\overline{\text{DMARQ3}}$		
P10/TO110, P11/TO111	5	
P12/TCLR11, P13/TI11	5-K	
P14/INTP110/$\overline{\text{DMAAK0}}$ to P17/INTP113/$\overline{\text{DMAAK3}}$		
P20/NMI	2	Connect directly to V_{SS}.
P21	5	Input: Independently connect to HV_{DD} or V_{SS} via a resistor. Output: Leave open.
P22/TXD0/SO0		
P23/RXD0/SI0	5-K	
P24/$\overline{\text{SCK0}}$		
P25/TXD1/SO1	5	
P26/RXD1/SI1	5-K	
P27/$\overline{\text{SCK1}}$		
P30/TO130, P31/TO131	5	
P32/TCLR13, P33/TI13	5-K	
P34/INTP130		
P35/INTP131/SO2		
P36/INTP132/SI2		
P37/INTP133/$\overline{\text{SCK2}}$		
P40/D0 to P47/D7	5	
P50/D8 to P57/D15		
P60/A16 to P67/A23		
P70/ANI0 to P77/ANI7	9	Connect directly to V_{SS}.
P80/$\overline{\text{CS0}}$/$\overline{\text{RAS0}}$ to P83/$\overline{\text{CS3}}$/$\overline{\text{RAS3}}$	5	Input: Independently connect to HV_{DD} or V_{SS} via a resistor. Output: Leave open.
P84/$\overline{\text{CS4}}$/$\overline{\text{RAS4}}$/$\overline{\text{IOWR}}$, P85/$\overline{\text{CS5}}$/$\overline{\text{RAS5}}$/$\overline{\text{IORD}}$		
P86/$\overline{\text{CS6}}$/$\overline{\text{RAS6}}$, P87/$\overline{\text{CS7}}$/$\overline{\text{RAS7}}$		
P90/$\overline{\text{LCAS}}$/$\overline{\text{LWR}}$		
P91/$\overline{\text{UCAS}}$/$\overline{\text{UWR}}$		
P92/$\overline{\text{RD}}$		
P93/$\overline{\text{WE}}$		
P94/$\overline{\text{BCYST}}$		
P95/$\overline{\text{OE}}$		
P96/$\overline{\text{HLDAK}}$		
P97/$\overline{\text{HLDRQ}}$		
P100/TO120, P101/TO121		

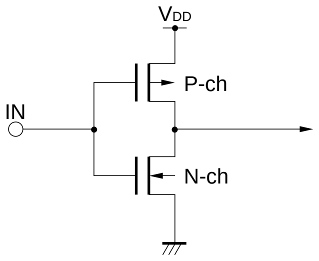
Pin Name	Input/Output Circuit Type	Recommended Connection of Unused Pins
P102/TCLR12, P103/TI12	5-K	Input: Independently connect to HV_{DD} or V_{SS} via a resistor. Output: Leave open.
P104/INTP120/$\overline{\text{TC0}}$ to P107/INTP123/$\overline{\text{TC3}}$		
P110/TO140, P111/TO141	5	
P112/TCLR14, P113/TI14	5-K	
P114/INTP140		
P115/INTP141/SO3		
P116/INTP142/SI3		
P117/INTP143/$\overline{\text{SCK3}}$		
P120/TO150, P121/TO151	5	
P122/TCLR15, P123/TI15	5-K	
P124/INTP150 to P126/INTP152		
P127/INTP153/ADTRG		
PA0/A0 to PA7/A7	5	
PB0/A8 to PB7/A15		
PX5/$\overline{\text{REFRQ}}$		
PX6/$\overline{\text{WAIT}}$		
PX7/CLKOUT		
CKSEL	1	Connect directly to HV_{DD}.
$\overline{\text{RESET}}$	2	—
MODE0 to MODE2		
MODE3[Note 1]		Connect to V_{SS} via a resistor (R_{VPP}).
MODE3/V_{PP}[Note 2]		
AV_{REF}, AV_{SS}	—	Connect directly to V_{SS}.
AV_{DD}	—	Connect directly to HV_{DD}.

Notes 1. μPD703100, 703100A, 703101, 703101A, 703102, 703102A only

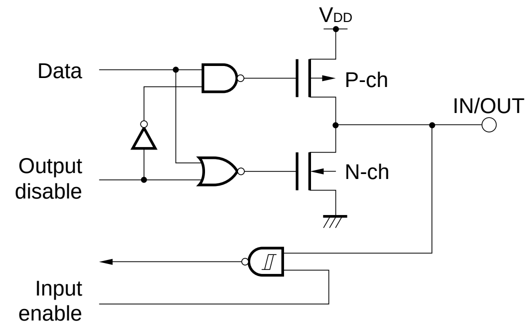
2. μPD70F3102, 70F3102A only

2.5 Pin Input/Output Circuits

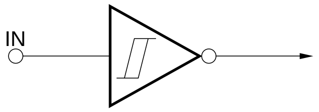
Type 1



Type 5-K

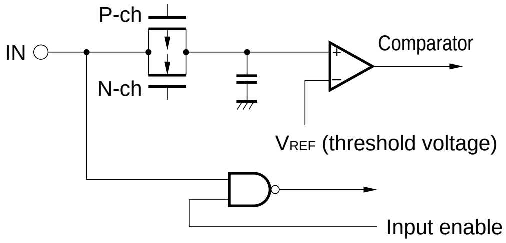


Type 2

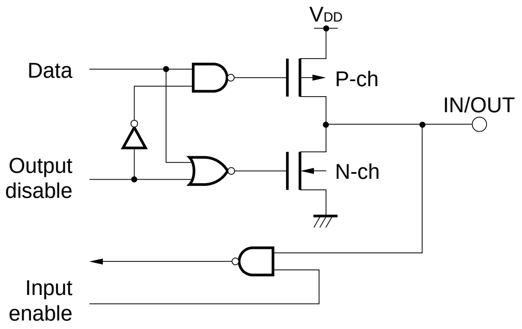


Schmitt-triggered input with hysteresis characteristics

Type 9



Type 5



Caution Note that V$_{DD}$ in the circuit diagram is replaced by HV$_{DD}$.

[MEMO]

CHAPTER 3 CPU FUNCTION

The CPU of the V850E/MS1 is based on RISC architecture and executes almost all the instructions in one clock cycle, using 5-stage pipeline control.

3.1 Features

- Minimum instruction execution time: 25 ns (at internal 40 MHz operation) … μPD703100-40, 703100A-40
  30 ns (at internal 33 MHz operation) … other than above
- Memory space Program space: 64 MB Linear
  Data space: 4 GB Linear
- Thirty-two 32-bit general-purpose registers
- Internal 32-bit architecture
- Five-stage pipeline control
- Multiplication/division instructions
- Saturated operation instructions
- One-clock 32-bit shift instruction
- Long/short instruction format
- Four types of bit manipulation instructions
  - Set
  - Clear
  - Not
  - Test

3.2 CPU Register Set

The registers of the V850E/MS1 can be classified into two categories: a general-purpose program register set and a dedicated system register set. The size of the registers is 32 bits.

For details, refer to **V850E/MS1 User's Manual Architecture**.

(1) Program register set

31	0
r0	Zero Register
r1	Reserved for Address Generation
r2	Interrupt Stack Pointer
r3	Stack Pointer (SP)
r4	Global Pointer (GP)
r5	Text Pointer (TP)
r6	
r7	
r8	
r9	
r10	
r11	
r12	
r13	
r14	
r15	
r16	
r17	
r18	
r19	
r20	
r21	
r22	
r23	
r24	
r25	
r26	
r27	
r28	
r29	
r30	Element Pointer (EP)
r31	Link Pointer (LP)

31	0
PC	Program Counter

(2) System register set

31	0
EIPC	Exception/Interrupt PC
EIPSW	Exception/Interrupt PSW

31	0
FEPC	Fatal Error PC
FEPSW	Fatal Error PSW

31	0
ECR	Exception Cause Register

31	0
PSW	Program Status Word

31	0
CTPC	CALLT Caller PC
CTPSW	CALLT Caller PSW

31	0
DBPC	ILGOP Caller PC
DBPSW	ILGOP Caller PSW

31	0
CTBP	CALLT Base Pointer

3.2.1 Program register set

The program register set includes general-purpose registers and a program counter.

(1) General-purpose registers

Thirty-two general-purpose registers, r0 to r31, are available. Any of these registers can be used as a data variable or address variable.

However, r0 and r30 are implicitly used by instructions, and care must be exercised when using these registers. Also, r1 to r5 and r31 are implicitly used by the assembler and C compiler. Therefore, before using these registers, their contents must be saved so that they are not lost. The contents must be restored to the registers after the registers have been used.

Table 3-1. Program Registers

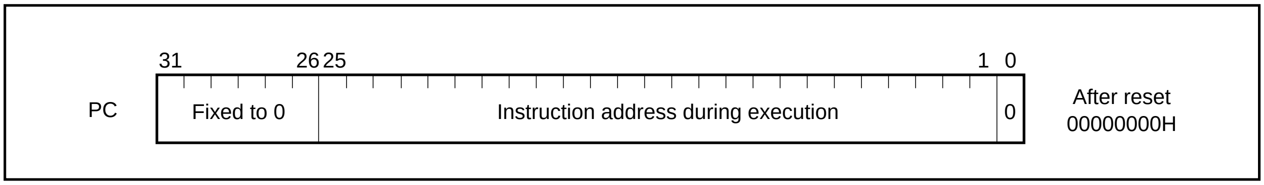
Name	Usage	Operation
r0	Zero register	Always holds 0
r1	Assembler-reserved register	Working register for generating 32-bit immediate data
r2	Interrupt stack pointer	Stack pointer for interrupt handler
r3	Stack pointer	Used to generate stack frame when function is called
r4	Global pointer	Used to access global variable in data area
r5	Text pointer	Register to indicate the start of the text area (where program code is located)
r6 to r29	—	Address/data variable registers
r30	Element pointer	Base pointer when memory is accessed
r31	Link pointer	Used by compiler when calling function
PC	Program counter	Holds instruction address during program execution

(2) Program counter

This register holds the instruction address during program execution. The lower 26 bits of this register are valid, and bits 31 to 26 are fixed to 0. If a carry occurs from bit 25 to 26, it is ignored.

Bit 0 is fixed to 0, and branching to an odd address cannot be performed.

Figure 3-1. Program Counter (PC)



3.2.2 System register set

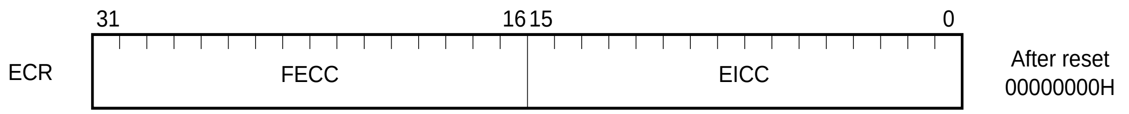
System registers control the status of the CPU and hold interrupt information.

Table 3-2. System Register Numbers

No.	System Register Name	Usage	Operation
0	EIPC	Status saving register during interrupt	These registers save the PC and PSW when a software exception or interrupt occurs. Because only one set of these registers is available, their contents must be saved when multiple interrupts are enabled.
1	EIPSW		
2	FEPC	Status saving register during NMI	These registers save the PC and PSW when an NMI occurs.
3	FEPSW		
4	ECR	Interrupt source register	If an exception, maskable interrupt, or NMI occurs, this register will contain information referencing the interrupt source. The higher 16 bits of this register are called FECC, to which the exception code of the NMI is set. The lower 16 bits are called EICC, to which the exception code of the exception/interrupt is set. Refer to **Figure 3-2**.
5	PSW	Program status word	The program status word is a collection of flags that indicate the program status (instruction execution result) and CPU status. Refer to **Figure 3-3**.
16	CTPC	Status saving register during CALLT execution	If the CALLT instruction is executed, this register saves the PC and PSW.
17	CTPSW		
18	DBPC	Status saving register during exception trap	If an exception trap is generated due to detection of an illegal instruction code, this register saves the PC and PSW.
19	DBPSW		
20	CTBP	CALLT base pointer	This is used to specify the table address and generate the target address.
6 to 15	Reserved		
21 to 31			

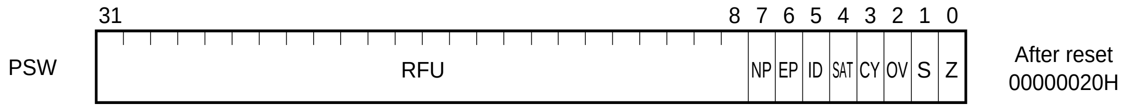
To read/write these system registers, specify the system register number indicated by a system register load/store instruction (LDSR or STSR instruction).

Figure 3-2. Interrupt Source Register (ECR)



Bit Position	Bit Name	Function
31 to 16	FECC	Fatal Error Cause Code Exception code of NMI (refer to **Table 7-1 Interrupt List**)
15 to 0	EICC	Exception/Interrupt Cause Code Exception code of exception/interrupt (refer to **Table 7-1 Interrupt List**)

Figure 3-3. Program Status Word (PSW)



Bit Position	Flag	Function
31 to 8	RFU	Reserved field (fixed to 0).
7	NP	NMI Pending Indicates that NMI processing is in progress. This flag is set when an NMI is accepted, and disables multiple interrupts.
6	EP	Exception Pending Indicates that exception processing is in progress. This flag is set when an exception is generated. Moreover, interrupt requests can be accepted when this bit is set.
5	ID	Interrupt Disable Indicates that accepting maskable interrupt request is disabled.
4	SAT	Saturated Math This flag is set if the result of executing saturated operation instruction overflows (if overflow does not occur, value of previous operation is held).
3	CY	Carry This flag is set if carry or borrow occurs as result of operation (if carry or borrow does not occur, it is reset).
2	OV	Overflow This flag is set if overflow occurs during operation (if overflow does not occur, it is reset).
1	S	Sign This flag is set if the result of operation is negative (it is reset if the result is positive).
0	Z	Zero This flag is set if the result of operation is zero (if the result is not zero, it is reset).

3.3 Operation Modes

3.3.1 Operation modes

The V850E/MS1 has the following operation modes. Mode specification is carried out by MODE0 to MODE3.

(1) Normal operation mode

(a) Single-chip modes 0, 1

Access to the internal ROM is enabled.

In single-chip mode 0, after system reset is cancelled, each pin related to the bus interface enters the port mode, branches to the reset entry address of the internal ROM and starts instruction processing. The external expansion mode, which connects an external device to external memory area, is enabled by setting the memory expansion mode register (MM: refer to **3.4.6 (1)**) with an instruction.

In single-chip mode 1, after system reset is cancelled, each pin related to the bus interface enters the control mode, branches to the external device (memory) reset entry address and starts instruction processing.

The internal ROM area is mapped from address 100000H.

(b) ROM-less modes 0, 1

After system reset is cancelled, each pin related to the bus interface enters the control mode, branches to the external device (memory) reset entry address and starts instruction processing. Fetching of instructions and data access from internal ROM becomes impossible.

In ROM-less mode 0, the data bus is a 16-bit data bus and in ROM-less mode 1, the data bus is an 8-bit data bus.

(2) Flash memory programming mode (μPD70F3102 and 70F3102A only)

If this mode is specified, it becomes possible for the flash programmer to run a program to the internal flash memory.

3.3.2 Operation mode specification

The operation mode is specified according to the status of pins MODE0 to MODE3. In an application system fix the specification of these pins and do not change them during operation.

Operation is not guaranteed if these pins are changed during operation.

(a) μPD703100, 703100A

MODE3	MODE2	MODE1	MODE0	Operation Mode		External Data Bus Width	Remarks
L	L	L	L	Normal operation mode	ROM-less mode 0	16 bits	—
L	L	L	H		ROM-less mode 1	8 bits	
Other than above				Setting prohibited		—	

(b) μPD703101, 703101A, 703102, 703102A

MODE3	MODE2	MODE1	MODE0	Operation Mode		External Data Bus Width	Remarks
L	L	L	L	Normal operation mode	ROM-less mode 0	16 bits	—
L	L	L	H		ROM-less mode 1	8 bits	—
L	L	H	L		Single-chip mode 0	—	Internal ROM area is allocated from address 000000H.
L	L	H	H		Single-chip mode 1	16 bits	Internal ROM area is allocated from address 100000H.
Other than above				Setting prohibited		—	—

(c) μPD70F3102, 70F3102A

MODE3/ V_{PP}	MODE2	MODE1	MODE0	Operation Mode		External Data Bus Width	Remarks
0 V	L	L	L	Normal operation mode	ROM-less mode 0	16 bits	—
0 V	L	L	H		ROM-less mode 1	8 bits	—
0 V	L	H	L		Single-chip mode 0	—	Internal ROM area is allocated from address 000000H.
0 V	L	H	H		Single-chip mode 1	16 bits	Internal ROM area is allocated from address 100000H.
7.8 V	L	H	L	Flash memory programming mode		—	—
Other than above				Setting prohibited		—	—

Remark L: Low-level input
H: High-level input

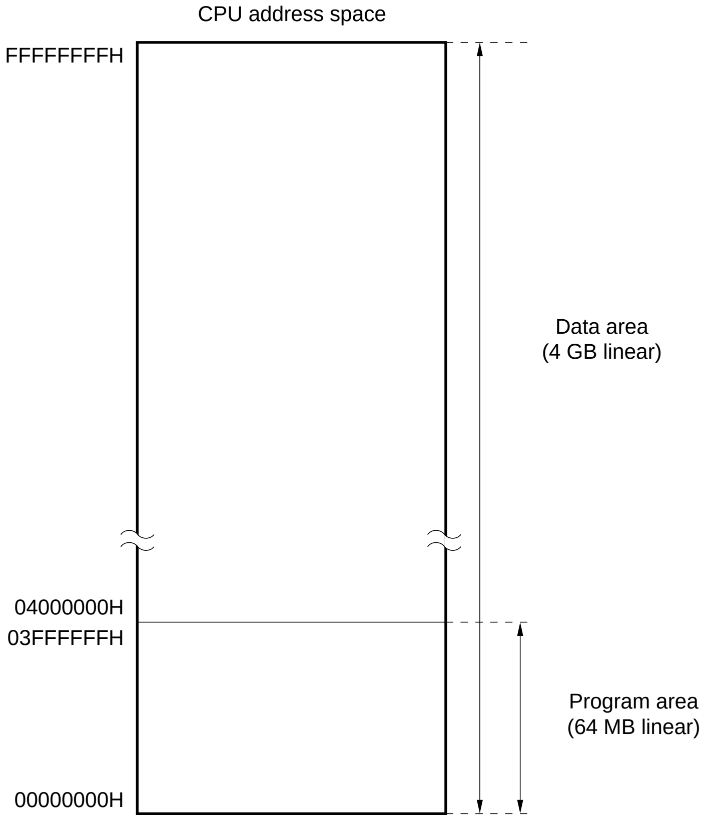
3.4 Address Space

3.4.1 CPU address space

The CPU of the V850E/MS1 is of 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). Also, in instruction address addressing, a maximum of 64 MB of linear address space (program space) is supported.

Figure 3-4 shows the CPU address space.

Figure 3-4. CPU Address Space

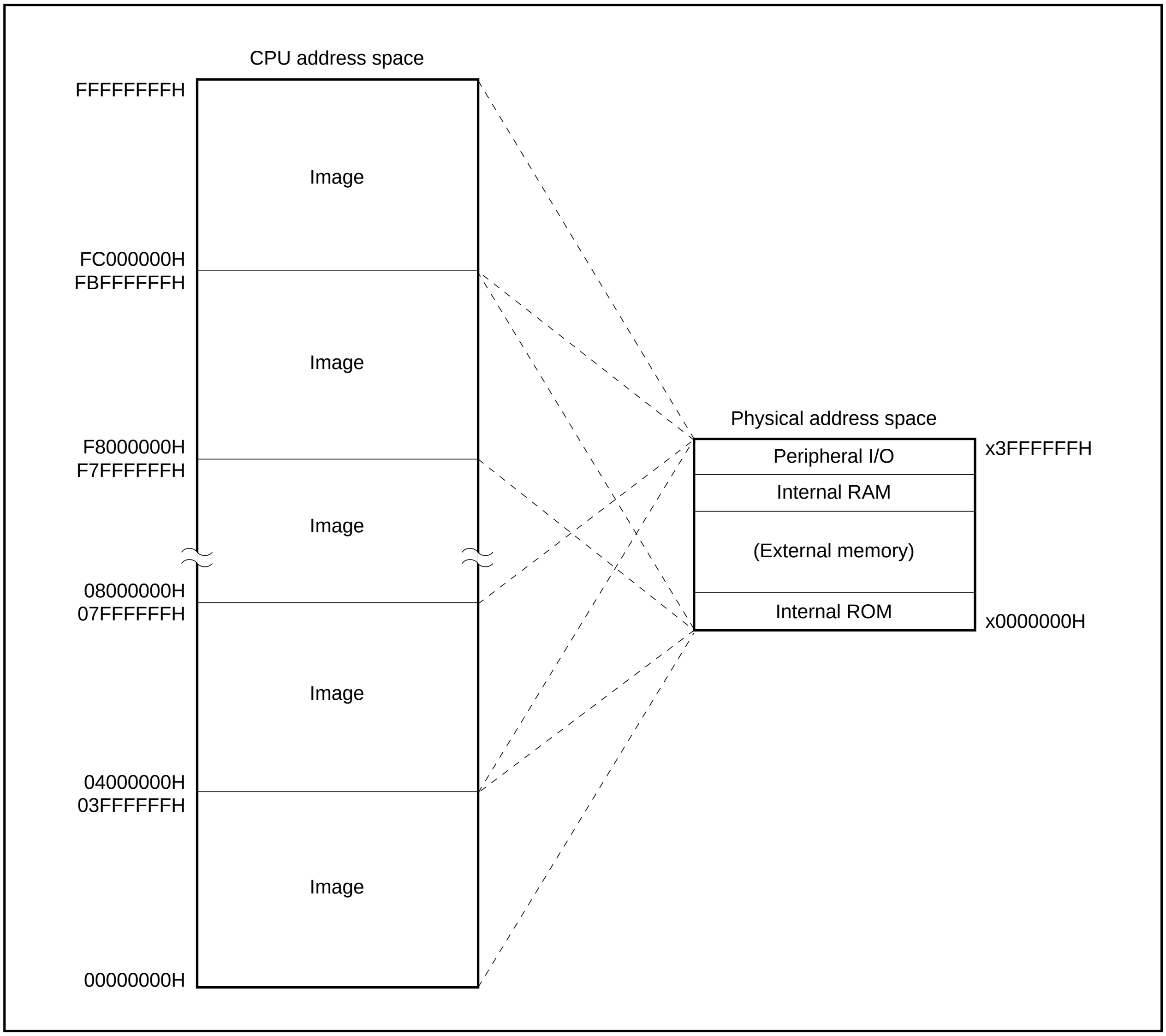


3.4.2 Image

The core CPU supports 4 GB of “virtual” addressing space, or 64 memory blocks, each containing 64 MB physical address space. In actuality, the same 64 MB physical address space is accessed regardless of the values of bits 31 to 26 of the CPU address. Figure 3-5 shows the image of the virtual addressing space.

Because the higher 6 bits of a 32-bit CPU address are disregarded and access is made to a 26-bit physical address, physical address x0000000H can be seen as CPU address 00000000H, and in addition, can be seen as address 04000000H, address 08000000H, address F8000000H or address FC000000H.

Figure 3-5. Image on Address Space



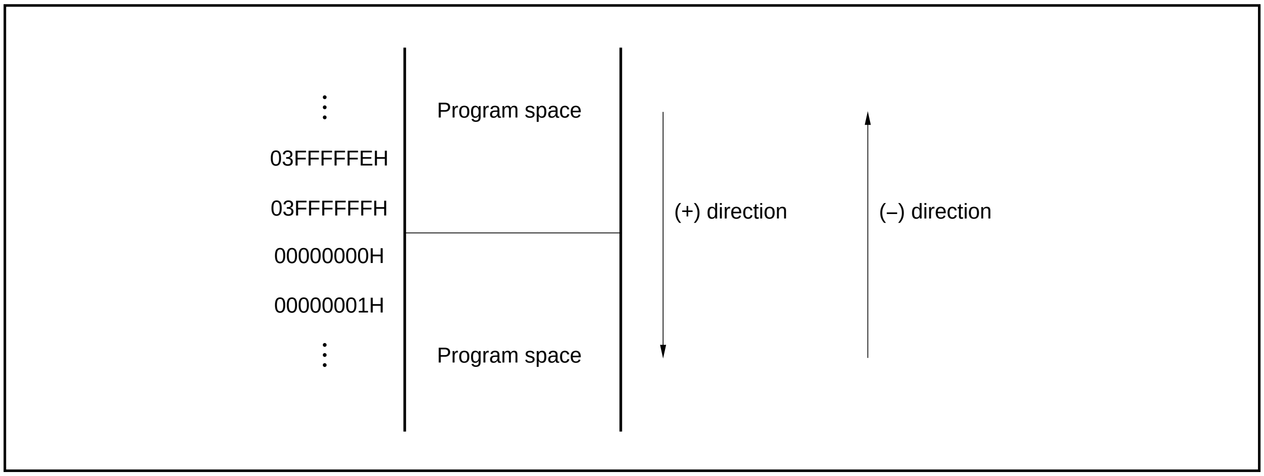
3.4.3 Wrap-around of CPU address space

(1) Program space

Of the 32 bits of the PC (program counter), the higher 6 bits are set to 0, and only the lower 26 bits are valid. Even if a carry or borrow occurs from bit 25 to 26 as a result of branch address calculation, the higher 6 bits ignore the carry or borrow.

Therefore, the lower-limit address of the program space, address 00000000H, and the upper-limit address 03FFFFFFH become contiguous addresses. Wrap-around refers to the situation that the lower-limit address and upper-limit address become contiguous like this.

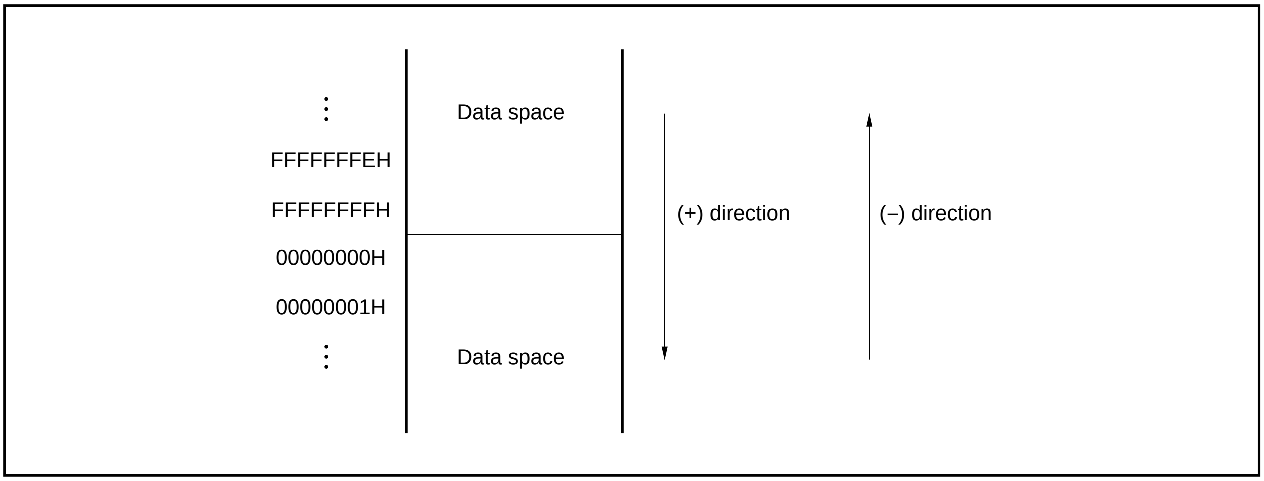
Caution No instruction can be fetched from the 4 KB area of 03FFF000H to 03FFFFFFH because this area is defined as the peripheral I/O area. Therefore, do not execute any branch address calculation in which the result will reside in any part of this area.



(2) Data space

The result of an operand address calculation that exceeds 32 bits is ignored.

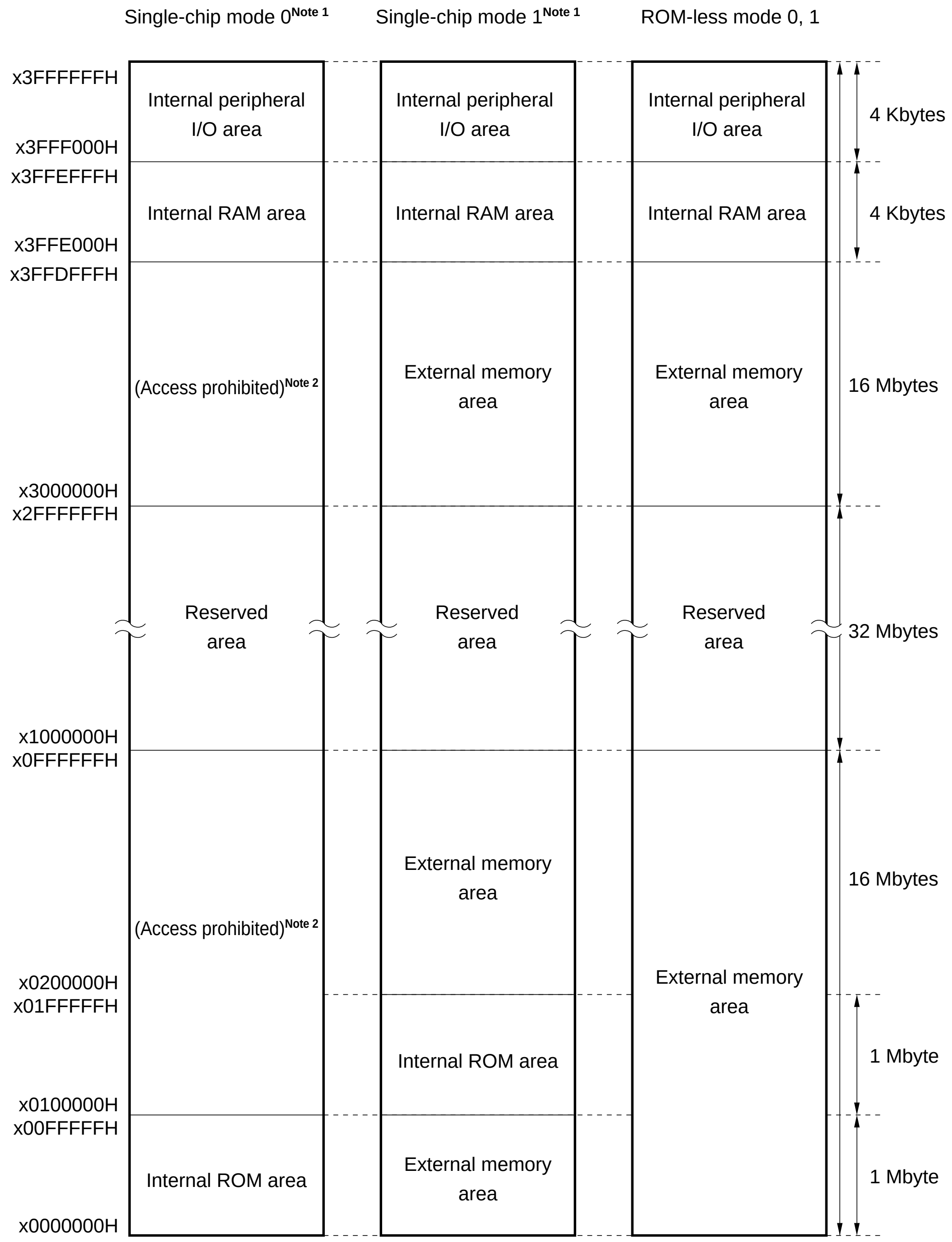
Therefore, the lower-limit address of the program space, address 00000000H, and the upper-limit address FFFFFFFFH are contiguous addresses, and the data space is wrapped around at the boundary of these addresses.



3.4.4 Memory map

The V850E/MS1 reserves areas as shown below.

Each mode is specified by the MM register and the MODE0 to MODE3 pins.



Notes 1. μPD703101, 703101A, 703102, 703102A, 70F3102, and 70F3102A only

2. If the external expansion mode is set, this area can be accessed as external memory area.

3.4.5 Area

(1) Internal ROM area (μPD703101, 703101A, 703102, 703102A, 70F3102, and 70F3102A only)

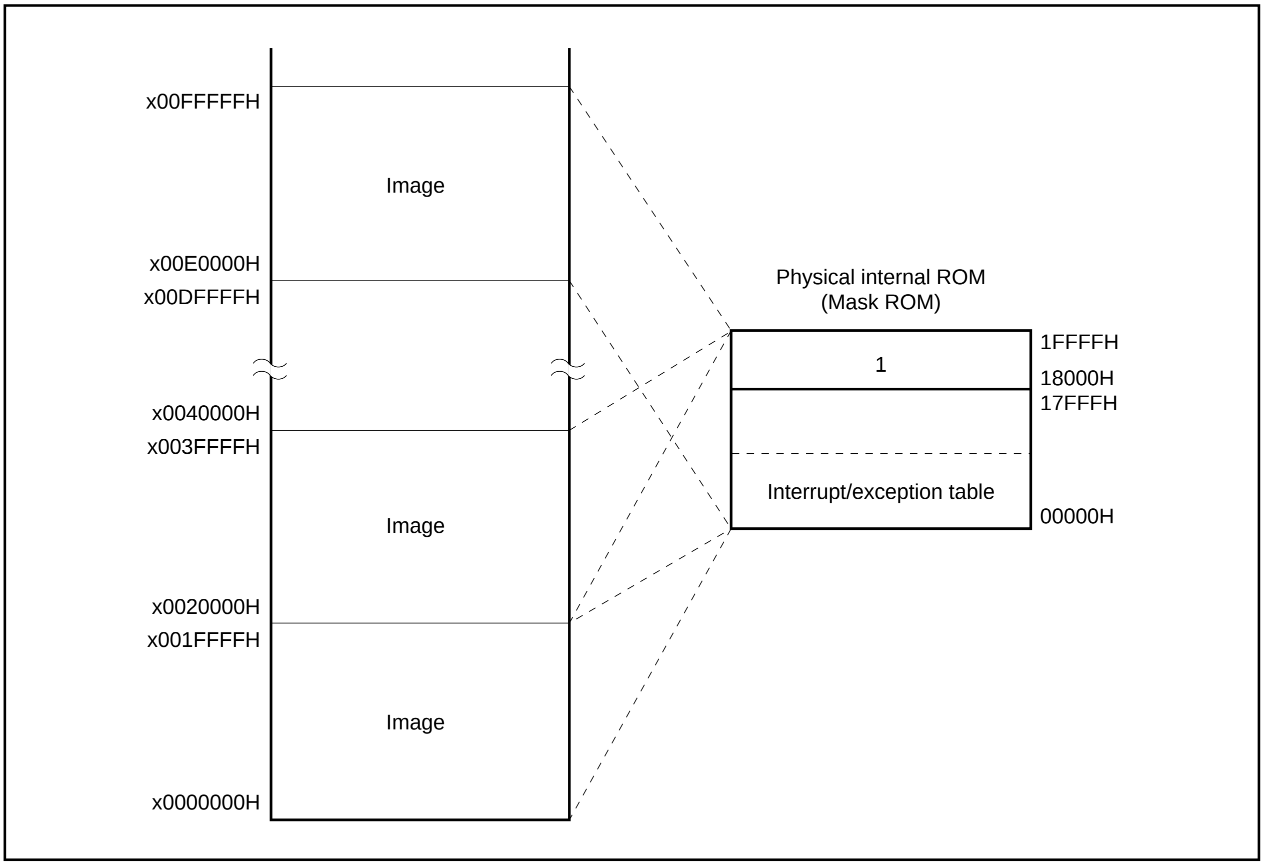
(a) Memory map

1 MB of internal ROM area, addresses 00000H to FFFFFH, is reserved.

<1> μPD703101, 703101A

96 KB of memory, addresses 00000H to 17FFFH, is provided as physical internal ROM (mask ROM).

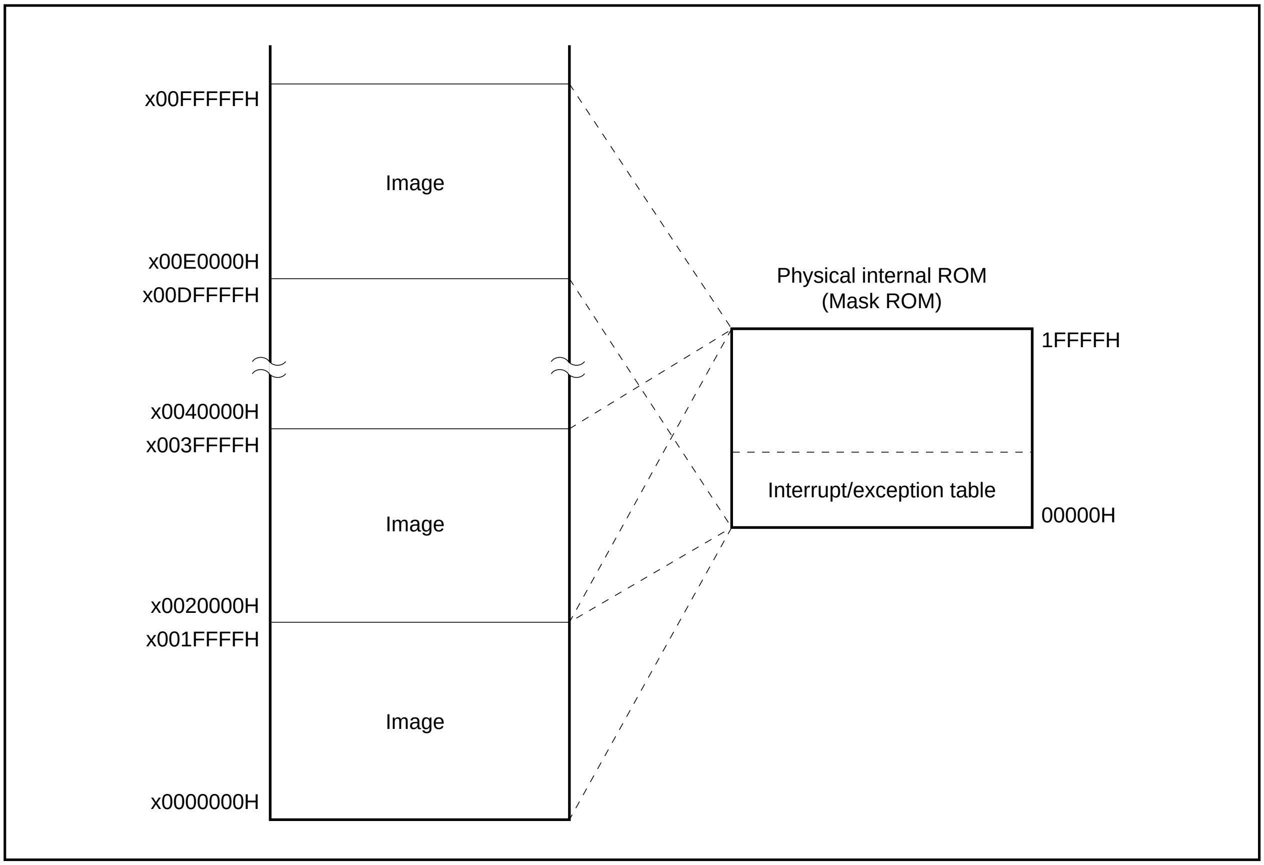
Also, in the remaining area (20000H to FFFFFH), the image of 00000H to 1FFFFH can be seen (however, addresses 18000H to 1FFFFH are fixed at 1).



<2> μPD703102, 703102A

128 KB of memory, addresses 00000H to 1FFFFH, is provided as physical internal ROM (mask ROM).

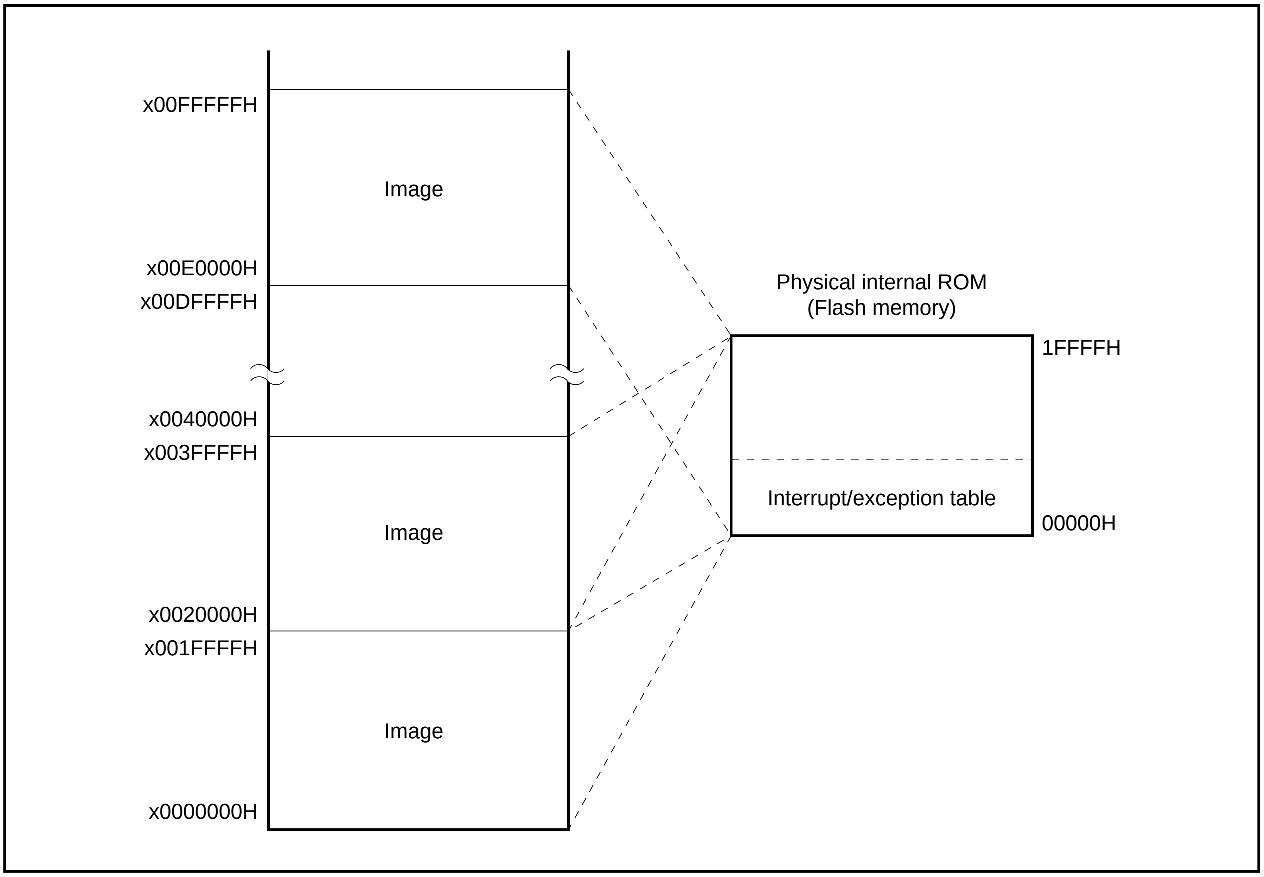
Also, in the remaining area (20000H to FFFFFH), the image of 00000H to 1FFFFH can be seen.



<3> μPD70F3102, 70F3102A

128 KB of memory, addresses 00000H to 1FFFFH, is provided as physical internal ROM (flash memory).

Also, in the remaining area (20000H to FFFFFH), the image of 00000H to 1FFFFH can be seen.



(b) Interrupt/exception table

The V850E/MS1 increases the interrupt response speed by assigning handler addresses corresponding to interrupts/exceptions.

The collection of these handler addresses is called an interrupt/exception table, which is located in the internal ROM area. When an interrupt/exception request is granted, execution jumps to the handler address, and the program written at that memory is executed. Table 3-3 shows the sources of interrupts/exceptions, and the corresponding addresses.

Remark When in ROM-less modes 0 and 1, or in the case of the μPD703100 or 703100A, the internal ROM area becomes an external memory area. In order to restore correct operation after reset, provide a handler address to the reset routine in address 0 of the external memory.

Table 3-3. Interrupt/Exception Table (1/2)

Start Address of Interrupt/Exception Table	Interrupt/Exception Source
00000000H	RESET
00000010H	NMI
00000040H	TRAP0n (n = 0 to FH)
00000050H	TRAP1n (n = 0 to FH)
00000060H	ILGOP
00000080H	INTOV10
00000090H	INTOV11
000000A0H	INTOV12
000000B0H	INTOV13
000000C0H	INTOV14
000000D0H	INTOV15
00000100H	INTP100/INTCC100
00000110H	INTP101/INTCC101
00000120H	INTP102/INTCC102
00000130H	INTP103/INTCC103
00000140H	INTP110/INTCC110
00000150H	INTP111/INTCC111
00000160H	INTP112/INTCC112
00000170H	INTP113/INTCC113
00000180H	INTP120/INTCC120
00000190H	INTP121/INTCC121
000001A0H	INTP122/INTCC122
000001B0H	INTP123/INTCC123
000001C0H	INTP130/INTCC130
000001D0H	INTP131/INTCC131
000001E0H	INTP132/INTCC132
000001F0H	INTP133/INTCC133
00000200H	INTP140/INTCC140
00000210H	INTP141/INTCC141
00000220H	INTP142/INTCC142
00000230H	INTP143/INTCC143
00000240H	INTP150/INTCC150
00000250H	INTP151/INTCC151
00000260H	INTP152/INTCC152
00000270H	INTP153/INTCC153
00000280H	INTCM40

Table 3-3. Interrupt/Exception Table (2/2)

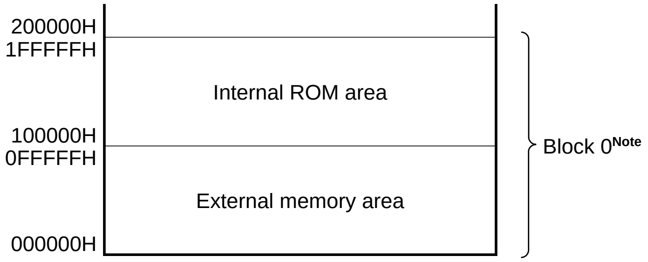
Start Address of Interrupt/Exception Table	Interrupt/Exception Source
00000290H	INTCM41
000002A0H	INTDMA0
000002B0H	INTDMA1
000002C0H	INTDMA2
000002D0H	INTDMA3
00000300H	INTCSI0
00000310H	INTSER0
00000320H	INTSR0
00000330H	INTST0
00000340H	INTCSI1
00000350H	INTSER1
00000360H	INTSR1
00000370H	INTST1
00000380H	INTCSI2
000003C0H	INTCSI3
00000400H	INTAD

(c) Internal ROM area relocation function

If set in single-chip mode 1, the internal ROM area is located beginning from address 100000H, so booting from external memory becomes possible.

Therefore, in order to restore correct operation after reset, provide a handler address to the reset routine in address 0 of the external memory.

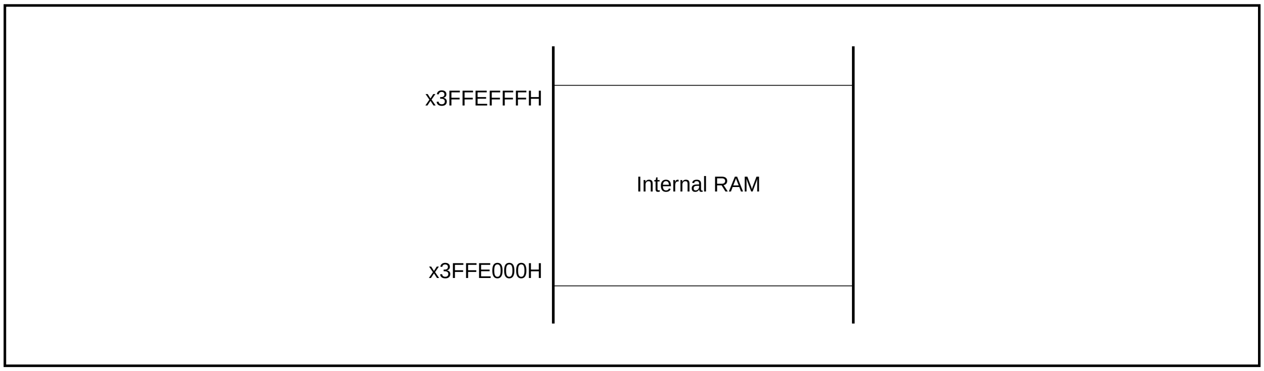
Figure 3-6. Internal ROM Area in Single-Chip Mode 1



Note Refer to **4.3 Memory Block Function**

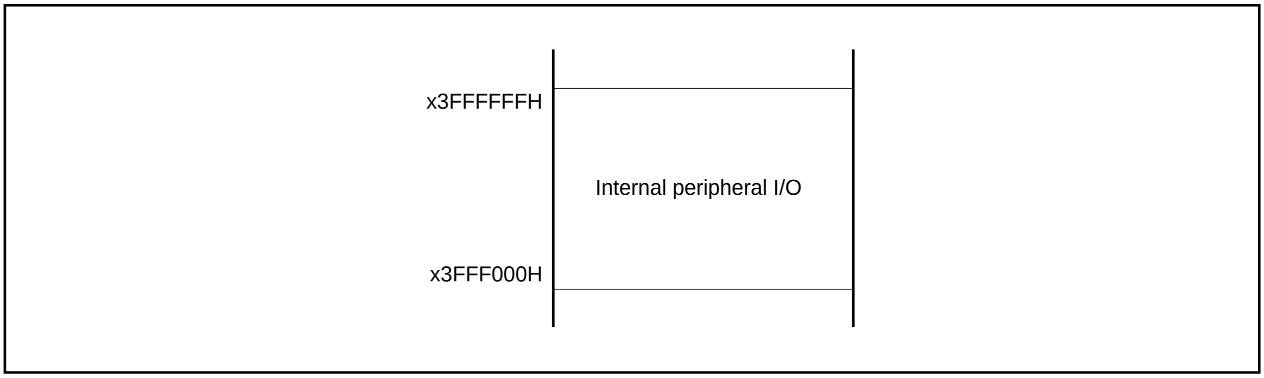
(2) Internal RAM area

4 KB of memory, addresses 3FFE000H to 3FFEFFFH, is provided as a physical internal RAM area.



(3) Internal peripheral I/O area

4 KB of memory, addresses 3FFF000H to 3FFFFFFH, is provided as an internal peripheral I/O area.



Peripheral I/O registers associated with the operation mode specification and the state monitoring for the internal peripheral I/O are all memory-mapped to the internal peripheral I/O area. Program fetches are not allowed in this area.

Cautions 1. The least significant bit of an address is not decoded. If byte access is executed in the register at an odd address (2n + 1), the register at the even address (2n) will be accessed because of the hardware specification.

2. In the V850E/MS1, no registers exist which are capable of word access, but if word access is executed in the register, for the word area, disregarding the bottom 2 bits of the address, halfword access is performed twice in the order of lower, then higher.

3. For registers in which byte access is possible, if halfword access is executed, the higher 8 bits become non-specific during the read operation, and the lower 8 bits of data are written to the register during the write operation.

4. Addresses that are not defined as registers are reserved for future expansion. If these addresses are accessed, the operation is undefined and not guaranteed.

(4) External memory area

The following areas can be used as external memory area. However, the reserved area from x1000000H to x2FFFFFFH is excluded.

(a) μPD703101, 703101A, 703102, 703102A, 70F3102, 70F3102A

When in single-chip mode 0:	x0100000H to x3FFDFFFH
When in single-chip mode 1:	x0000000H to x00FFFFFH, x0200000H to x3FFDFFFH
When in ROM-less modes 0 and 1:	x0000000H to x3FFDFFFH

(b) μPD703100, 703100A

x0000000H to x3FFDFFFH

Access to the external memory area uses the chip select signal assigned to each memory block (refer to **4.4 Bus Cycle Type Control Function**).

Note that the internal ROM, internal RAM and internal peripheral I/O areas cannot be accessed as external memory areas.

3.4.6 External expansion mode

The V850E/MS1 allows external devices to be connected to the external memory space by using the pins of ports 4, 5, 6, A, and B. Setting the external expansion mode is carried out by selecting each pin of ports 4, 5, 6, A, and B in the control mode by means of the MM register.

Note that the status at reset time differs as shown below in accordance with the operating mode specification set by pins MODE0 to MODE3 (refer to **3.3 Operation Modes** for details of the operation modes).

(1) Status at reset time in each operation mode

(a) In the case of ROM-less mode 0

At reset time, each pin of ports 4, 5, 6, A, and B enters the control mode, so the external expansion mode is set without changing the MM register (the external data bus width is 16 bits).

(b) In the case of ROM-less mode 1

At reset time, each pin of ports 4, 5, 6, A, and B enters the control mode, so the external expansion mode is set without changing the setting of the MM register (the external data bus width is 8 bits).

(c) In the case of single-chip mode 0

At reset time, since the internal ROM area is accessed, each pin of ports 4, 5, 6, A, and B enters the port mode and external devices cannot be used.

Set the MM register to change to the external expansion mode.

(d) In the case of single-chip mode 1

Internal ROM area is allocated from address 100000H (Refer to **3.4.5 (1) (c) Internal ROM area relocation function**). For that reason, at reset time, each pin of ports 4, 5, 6, A, and B enters the control mode, and is set in the external expansion mode without changing the settings of the MM register (the external data bus width becomes 16 bits).

(2) Memory expansion mode register (MM)

This register sets the mode of each pin of ports 4, 5, 6, A, and B. In the external expansion mode, an external device can be connected to an external memory area of up to 32 MB. However, an external device cannot be connected to the internal RAM area, internal peripheral I/O area, and internal ROM area in the single-chip modes 0, 1 (even if connected physically, it does not become an access target.).

The MM register can be read/written in 8- or 1-bit units. However, bits 4 to 7 are fixed to 0.

	7	6	5	4	3	2	1	0		
MM	0	0	0	0	MM3	MM2	MM1	MM0	Address FFFFF04CH	After reset **Note**

Note When in ROM-less mode 0: 07H When in single-chip mode 0: 00H
When in ROM-less mode 1: 0FH When in single-chip mode 1: 07H

Bit Position	Bit Name	Function
3 to 0	MM3 to MM0	Memory Expansion Mode Set the function of ports 4, 5, 6, A, and B.

MM3	MM2	MM1	MM0	Port 4	Port 5	Port A	Port B			Port 6			
0	0	0	0	P40 to P47	P50 to P57	PA0 to PA7	PB0 to PB3	PB4, PB5	PB6, PB7	P60, P61	P62, P63	P64, P65	P66, P67
0	0	0	1	D0 to D7	D8 to D15	A0 to A7	A8 to A11						
0	0	1	0					A12, A13					
0	0	1	1						A14, A15				
0	1	0	0							A16, A17			
0	1	0	1								A18, A19		
0	1	1	0									A20, A21	
0	1	1	1										A22, A23
1	0	0	0	P40 to P47	P50 to P57	PA0 to PA7	PB0 to PB3	PB4, PB5	PB6, PB7	P60, P61	P62, P63	P64, P65	P66, P67
1	0	0	1	D0 to D7		A0 to A7	A8 to A11						
1	0	1	0					A12, A13					
1	0	1	1						A14, A15				
1	1	0	0							A16, A17			
1	1	0	1								A18, A19		
1	1	1	0									A20, A21	
1	1	1	1										A22, A23

Caution **Write to the MM register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the MM register is complete. However, it is possible to access an external memory area whose initialization is complete.**

Remarks **1.** For details of the operation of each port's pins, refer to **2.3 Description of Pin Functions**.

2. The function of each port at system reset time is as shown below.

Operation Mode	MM Register	Port 4	Port 5	Port A	Port B	Port 6
ROM-less mode 0	07H	D0 to D7	D8 to D15	A0 to A7	A8 to A15	A16 to A23
ROM-less mode 1	0FH		P50 to P57			
Single-chip mode 0	00H	P40 to P47	P50 to P57	PA0 to PA7	PB0 to PB7	P60 to P67
Single-chip mode 1	07H	D0 to D7	D8 to D15	A0 to A7	A8 to A15	A16 to A23

3.4.7 Recommended use of address space

The architecture of the V850E/MS1 requires that a register that serves as a pointer be secured for address generation when accessing the operand data in the data space. An instruction can be used to directly access operand data at the address in this pointer register ±32 KB. However, the general-purpose registers that can be used as a pointer register are limited. Therefore, by minimizing the deterioration of address calculation performance when changing the pointer value, the number of usable general-purpose registers for handling variables is maximized, and the program size can be saved.

To enhance the efficiency of using the pointer in connection with the memory map of the V850E/MS1, the following points are recommended:

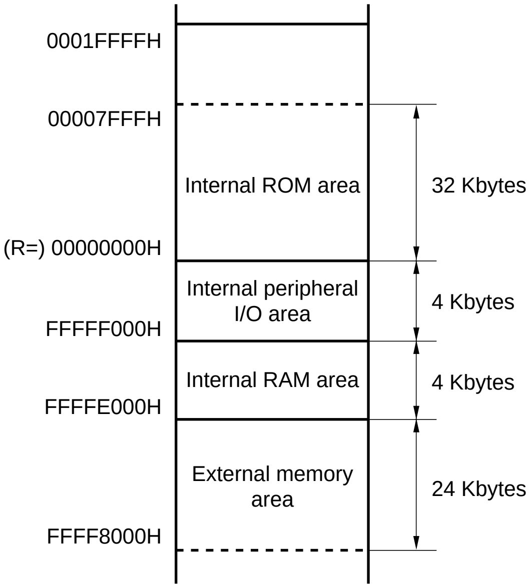
(1) Program space

Of the 32 bits of the PC (program counter), the higher 6 bits are fixed to 0, and only the lower 26 bits are valid. Therefore, a contiguous 64 MB space, starting from address 00000000H, unconditionally corresponds to the memory map of the program space.

(2) Data space

For the efficient use of resources using the wrap-around feature of the data space, the continuous 16 MB address spaces 00000000H to 00FFFFFFH and FF000000H to FFFFFFFFH of the 4 GB CPU are used as the data space. With the V850E/MS1, the 64 MB physical address space is seen as 64 images in the 4 GB CPU address space. The highest bit (bit 25) of this 26-bit address is assigned as address sign-extended to 32 bits.

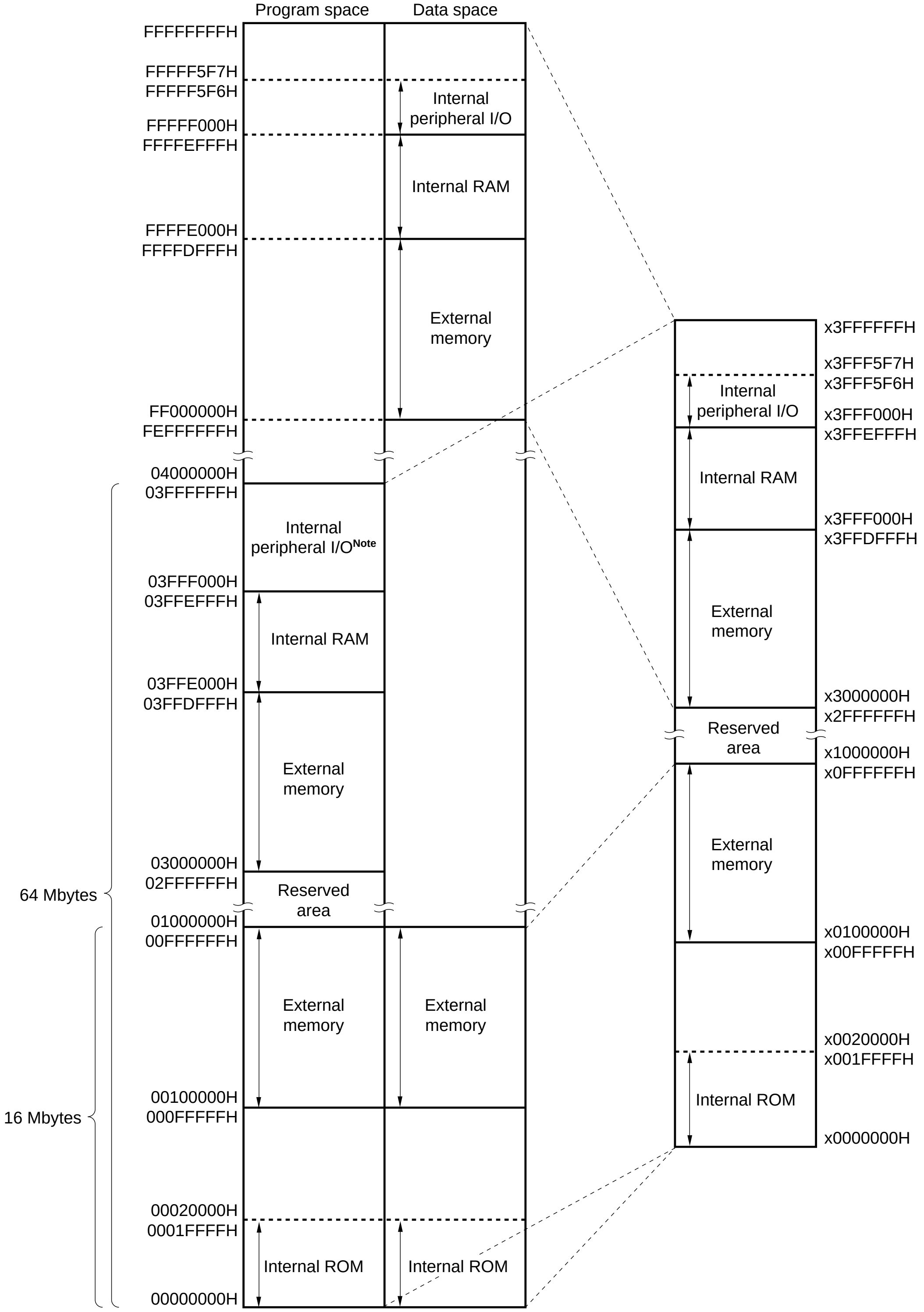
Example Application of wrap-around



When R = r0 (zero register) is specified for the LD/ST disp16 [R] instruction, an addressing range of 00000000H ±32 KB can be referenced with the sign-extended, 16-bit displacement value. By mapping the external memory in the 24 KB area in the figure, all resources including internal hardware can be accessed with one pointer.

The zero register (r0) is a register set to 0 by hardware, and eliminates the need for additional registers for the pointer.

Figure 3-7. Recommended Memory Map



Note This area cannot be used as a program area.

Remarks **1.** The arrows indicate the recommended area.

2. This is a recommended memory map when the μPD703102 is set to single-chip mode 0, and used as external expansion mode.

3.4.8 Peripheral I/O registers

(1/8)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 bit	8 bits	16 bits	
FFFFF000H	Port 0	P0	R/W	○	○		Undefined
FFFFF002H	Port 1	P1		○	○		
FFFFF004H	Port 2	P2		○	○		
FFFFF006H	Port 3	P3		○	○		
FFFFF008H	Port 4	P4		○	○		
FFFFF00AH	Port 5	P5		○	○		
FFFFF00CH	Port 6	P6		○	○		
FFFFF00EH	Port 7	P7	R	○	○		
FFFFF010H	Port 8	P8	R/W	○	○		
FFFFF012H	Port 9	P9		○	○		
FFFFF014H	Port 10	P10		○	○		
FFFFF016H	Port 11	P11		○	○		
FFFFF018H	Port 12	P12		○	○		
FFFFF01CH	Port A	PA		○	○		
FFFFF01EH	Port B	PB		○	○		
FFFFF020H	Port 0 mode register	PM0		○	○		FFH
FFFFF022H	Port 1 mode register	PM1		○	○		
FFFFF024H	Port 2 mode register	PM2		○	○		
FFFFF026H	Port 3 mode register	PM3		○	○		
FFFFF028H	Port 4 mode register	PM4		○	○		
FFFFF02AH	Port 5 mode register	PM5		○	○		
FFFFF02CH	Port 6 mode register	PM6		○	○		
FFFFF030H	Port 8 mode register	PM8		○	○		
FFFFF032H	Port 9 mode register	PM9		○	○		
FFFFF034H	Port 10 mode register	PM10		○	○		
FFFFF036H	Port 11 mode register	PM11		○	○		
FFFFF038H	Port 12 mode register	PM12		○	○		
FFFFF03CH	Port A mode register	PMA		○	○		
FFFFF03EH	Port B mode register	PMB		○	○		
FFFFF040H	Port 0 mode control register	PMC0		○	○		00H
FFFFF042H	Port 1 mode control register	PMC1		○	○		
FFFFF044H	Port 2 mode control register	PMC2		○	○		01H
FFFFF046H	Port 3 mode control register	PMC3		○	○		00H
FFFFF04CH	Memory expansion mode register	MM		○	○		00H/07H/ 0FH

(2/8)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 bit	8 bits	16 bits	
FFFFF050H	Port 8 mode control register	PMC8	R/W	○	○		00H/FFH
FFFFF052H	Port 9 mode control register	PMC9		○	○		
FFFFF054H	Port 10 mode control register	PMC10		○	○		00H
FFFFF056H	Port 11 mode control register	PMC11		○	○		
FFFFF058H	Port 12 mode control register	PMC12		○	○		
FFFFF060H	Data wait control register 1	DWC1				○	FFFFH
FFFFF062H	Bus cycle control register	BCC				○	5555H
FFFFF064H	Bus cycle type control register	BCT				○	0000H
FFFFF066H	Bus size configuration register	BSC				○	5555H/ 0000H
FFFFF06AH	Data wait control register 2	DWC2		○	○		FFH
FFFFF06CH	Fly-by transfer data wait control register	FDW		○	○		00H
FFFFF070H	Power save control register	PSC		○	○		
FFFFF072H	Clock control register	CKC		○	○		
FFFFF078H	System status register	SYS		○	○		0000000×B
FFFFF084H	Baud rate generator compare register 0	BRGC0		○	○		Undefined
FFFFF086H	Baud rate generator prescaler mode register 0	BPRM0		○	○		00H
FFFFF088H	Clocked serial interface mode register 0	CSIM0		○	○		
FFFFF08AH	Serial I/O shift register 0	SIO0		○	○		Undefined
FFFFF094H	Baud rate generator compare register 1	BRGC1		○	○		
FFFFF096H	Baud rate generator prescaler mode register 1	BPRM1		○	○		00H
FFFFF098H	Clocked serial interface mode register 1	CSIM1		○	○		
FFFFF09AH	Serial I/O shift register 1	SIO1		○	○		Undefined
FFFFF0A4H	Baud rate generator compare register 2	BRGC2		○	○		
FFFFF0A6H	Baud rate generator prescaler mode register 2	BPRM2		○	○		00H
FFFFF0A8H	Clocked serial interface mode register 2	CSIM2		○	○		
FFFFF0AAH	Serial I/O shift register 2	SIO2		○	○		Undefined
FFFFF0B8H	Clocked serial interface mode register 3	CSIM3		○	○		00H
FFFFF0BAH	Serial I/O shift register 3	SIO3		○	○		Undefined
FFFFF0C0H	Asynchronous serial interface mode register 00	ASIM00		○	○		80H
FFFFF0C2H	Asynchronous serial interface mode register 01	ASIM01		○	○		00H
FFFFF0C4H	Asynchronous serial interface status register 0	ASIS0	R	○	○		
FFFFF0C8H	Receive buffer 0 (9 bits)	RXB0				○	Undefined
FFFFF0CAH	Receive buffer 0L (lower 8 bits)	RXB0L		○	○		
FFFFF0CCH	Transmit shift register 0 (9 bits)	TXS0	W			○	
FFFFF0CEH	Transmit shift register 0L (lower 8 bits)	TXS0L			○		

(3/8)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 bit	8 bits	16 bits	
FFFFF0D0H	Asynchronous serial interface mode register 10	ASIM10	R/W	○	○		80H
FFFFF0D2H	Asynchronous serial interface mode register 11	ASIM11		○	○		00H
FFFFF0D4H	Asynchronous serial interface status register 1	ASIS1	R	○	○		
FFFFF0D8H	Receive buffer 1 (9 bits)	RXB1				○	Undefined
FFFFF0DAH	Receive buffer 1L (lower 8 bits)	RXB1L		○	○		
FFFFF0DCH	Transmit shift register 1 (9 bits)	TXS1	W			○	
FFFFF0DEH	Transmit shift register 1L (lower 8 bits)	TXS1L			○		
FFFFF100H	Interrupt control register	OVIC10	R/W	○	○		47H
FFFFF102H	Interrupt control register	OVIC11		○	○		
FFFFF104H	Interrupt control register	OVIC12		○	○		
FFFFF106H	Interrupt control register	OVIC13		○	○		
FFFFF108H	Interrupt control register	OVIC14		○	○		
FFFFF10AH	Interrupt control register	OVIC15		○	○		
FFFFF10CH	Interrupt control register	CMIC40		○	○		
FFFFF10EH	Interrupt control register	CMIC41		○	○		
FFFFF110H	Interrupt control register	P10IC0		○	○		
FFFFF112H	Interrupt control register	P10IC1		○	○		
FFFFF114H	Interrupt control register	P10IC2		○	○		
FFFFF116H	Interrupt control register	P10IC3		○	○		
FFFFF118H	Interrupt control register	P11IC0		○	○		
FFFFF11AH	Interrupt control register	P11IC1		○	○		
FFFFF11CH	Interrupt control register	P11IC2		○	○		
FFFFF11EH	Interrupt control register	P11IC3		○	○		
FFFFF120H	Interrupt control register	P12IC0		○	○		
FFFFF122H	Interrupt control register	P12IC1		○	○		
FFFFF124H	Interrupt control register	P12IC2		○	○		
FFFFF126H	Interrupt control register	P12IC3		○	○		
FFFFF128H	Interrupt control register	P13IC0		○	○		
FFFFF12AH	Interrupt control register	P13IC1		○	○		
FFFFF12CH	Interrupt control register	P13IC2		○	○		
FFFFF12EH	Interrupt control register	P13IC3		○	○		
FFFFF130H	Interrupt control register	P14IC0		○	○		
FFFFF132H	Interrupt control register	P14IC1		○	○		
FFFFF134H	Interrupt control register	P14IC2		○	○		
FFFFF136H	Interrupt control register	P14IC3		○	○		

(4/8)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 bit	8 bits	16 bits	
FFFFF138H	Interrupt control register	P15IC0	R/W	○	○		47H
FFFFF13AH	Interrupt control register	P15IC1		○	○		
FFFFF13CH	Interrupt control register	P15IC2		○	○		
FFFFF13EH	Interrupt control register	P15IC3		○	○		
FFFFF140H	Interrupt control register	DMAIC0		○	○		
FFFFF142H	Interrupt control register	DMAIC1		○	○		
FFFFF144H	Interrupt control register	DMAIC2		○	○		
FFFFF146H	Interrupt control register	DMAIC3		○	○		
FFFFF148H	Interrupt control register	CSIC0		○	○		
FFFFF14AH	Interrupt control register	CSIC1		○	○		
FFFFF14CH	Interrupt control register	CSIC2		○	○		
FFFFF14EH	Interrupt control register	CSIC3		○	○		
FFFFF150H	Interrupt control register	SEIC0		○	○		
FFFFF152H	Interrupt control register	SRIC0		○	○		
FFFFF154H	Interrupt control register	STIC0		○	○		
FFFFF156H	Interrupt control register	SEIC1		○	○		
FFFFF158H	Interrupt control register	SRIC1		○	○		
FFFFF15AH	Interrupt control register	STIC1		○	○		
FFFFF15CH	Interrupt control register	ADIC		○	○		
FFFFF166H	In-service priority register	ISPR	R	○	○		00H
FFFFF170H	Command register	PRCMD	W		○		Undefined
FFFFF180H	External interrupt mode register 0	INTM0	R/W	○	○		00H
FFFFF182H	External interrupt mode register 1	INTM1		○	○		
FFFFF184H	External interrupt mode register 2	INTM2		○	○		
FFFFF186H	External interrupt mode register 3	INTM3		○	○		
FFFFF188H	External interrupt mode register 4	INTM4		○	○		
FFFFF18AH	External interrupt mode register 5	INTM5		○	○		
FFFFF18CH	External interrupt mode register 6	INTM6		○	○		
FFFFF1A0H	DMA source address register 0H	DSA0H				○	Undefined
FFFFF1A2H	DMA source address register 0L	DSA0L				○	
FFFFF1A4H	DMA destination address register 0H	DDA0H				○	
FFFFF1A6H	DMA destination address register 0L	DDA0L				○	
FFFFF1A8H	DMA source address register 1H	DSA1H				○	
FFFFF1AAH	DMA source address register 1L	DSA1L				○	
FFFFF1ACH	DMA destination address register 1H	DDA1H				○	
FFFFF1AEH	DMA destination address register 1L	DDA1L				○	

(5/8)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 bit	8 bits	16 bits	
FFFFF1B0H	DMA source address register 2H	DSA2H	R/W			○	Undefined
FFFFF1B2H	DMA source address register 2L	DSA2L				○	
FFFFF1B4H	DMA destination address register 2H	DDA2H				○	
FFFFF1B6H	DMA destination address register 2L	DDA2L				○	
FFFFF1B8H	DMA source address register 3H	DSA3H				○	
FFFFF1BAH	DMA source address register 3L	DSA3L				○	
FFFFF1BCH	DMA destination address register 3H	DDA3H				○	
FFFFF1BEH	DMA destination address register 3L	DDA3L				○	
FFFFF1E0H	DMA byte count register 0	DBC0				○	
FFFFF1E2H	DMA byte count register 1	DBC1				○	
FFFFF1E4H	DMA byte count register 2	DBC2				○	
FFFFF1E6H	DMA byte count register 3	DBC3				○	
FFFFF1F0H	DMA addressing control register 0	DADC0				○	0000H
FFFFF1F2H	DMA addressing control register 1	DADC1				○	
FFFFF1F4H	DMA addressing control register 2	DADC2				○	
FFFFF1F6H	DMA addressing control register 3	DADC3				○	
FFFFF200H	DRAM configuration register 0	DRC0				○	3FC1H
FFFFF202H	DRAM configuration register 1	DRC1				○	
FFFFF204H	DRAM configuration register 2	DRC2				○	
FFFFF206H	DRAM configuration register 3	DRC3				○	
FFFFF210H	Refresh control register 0	RFC0				○	0000H
FFFFF212H	Refresh control register 1	RFC1				○	
FFFFF214H	Refresh control register 2	RFC2				○	
FFFFF216H	Refresh control register 3	RFC3				○	
FFFFF218H	Refresh wait control register	RWC		○	○		00H
FFFFF220H	DRAM type configuration register	DTC				○	0000H
FFFFF224H	Page-ROM configuration register	PRC		○	○		E0H
FFFFF230H	Timer overflow status register	TOVS		○	○		00H
FFFFF240H	Timer unit mode register 10	TUM10				○	0000H
FFFFF242H	Timer control register 10	TMC10		○	○		00H
FFFFF244H	Timer output control register 10	TOC10		○	○		
FFFFF250H	Timer 10	TM10	R			○	0000H
FFFFF252H	Capture/compare register 100	CC100	R/W			○	Undefined
FFFFF254H	Capture/compare register 101	CC101				○	
FFFFF256H	Capture/compare register 102	CC102				○	
FFFFF258H	Capture/compare register 103	CC103				○	

(6/8)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 bit	8 bits	16 bits	
FFFFF260H	Timer unit mode register 11	TUM11	R/W			○	0000H
FFFFF262H	Timer control register 11	TMC11		○	○		00H
FFFFF264H	Timer output control register 11	TOC11		○	○		
FFFFF270H	Timer 11	TM11	R			○	0000H
FFFFF272H	Capture/compare register 110	CC110	R/W			○	Undefined
FFFFF274H	Capture/compare register 111	CC111				○	
FFFFF276H	Capture/compare register 112	CC112				○	
FFFFF278H	Capture/compare register 113	CC113				○	
FFFFF280H	Timer unit mode register 12	TUM12				○	0000H
FFFFF282H	Timer control register 12	TMC12		○	○		00H
FFFFF284H	Timer output control register 12	TOC12		○	○		
FFFFF290H	Timer 12	TM12	R			○	0000H
FFFFF292H	Capture/compare register 120	CC120	R/W			○	Undefined
FFFFF294H	Capture/compare register 121	CC121				○	
FFFFF296H	Capture/compare register 122	CC122				○	
FFFFF298H	Capture/compare register 123	CC123				○	
FFFFF2A0H	Timer unit mode register 13	TUM13				○	0000H
FFFFF2A2H	Timer control register 13	TMC13		○	○		00H
FFFFF2A4H	Timer output control register 13	TOC13		○	○		
FFFFF2B0H	Timer 13	TM13	R			○	0000H
FFFFF2B2H	Capture/compare register 130	CC130	R/W			○	Undefined
FFFFF2B4H	Capture/compare register 131	CC131				○	
FFFFF2B6H	Capture/compare register 132	CC132				○	
FFFFF2B8H	Capture/compare register 133	CC133				○	
FFFFF2C0H	Timer unit mode register 14	TUM14				○	0000H
FFFFF2C2H	Timer control register 14	TMC14		○	○		00H
FFFFF2C4H	Timer output control register 14	TOC14		○	○		
FFFFF2D0H	Timer 14	TM14	R			○	0000H
FFFFF2D2H	Capture/compare register 140	CC140	R/W			○	Undefined
FFFFF2D4H	Capture/compare register 141	CC141				○	
FFFFF2D6H	Capture/compare register 142	CC142				○	
FFFFF2D8H	Capture/compare register 143	CC143				○	
FFFFF2E0H	Timer unit mode register 15	TUM15				○	0000H
FFFFF2E2H	Timer control register 15	TMC15		○	○		00H
FFFFF2E4H	Timer output control register 15	TOC15		○	○		
FFFFF2F0H	Timer 15	TM15	R			○	0000H

(7/8)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 bit	8 bits	16 bits	
FFFFF2F2H	Capture/compare register 150	CC150	R/W			○	Undefined
FFFFF2F4H	Capture/compare register 151	CC151				○	
FFFFF2F6H	Capture/compare register 152	CC152				○	
FFFFF2F8H	Capture/compare register 153	CC153				○	
FFFFF342H	Timer control register 40	TMC40		○	○		00H
FFFFF346H	Timer control register 41	TMC41		○	○		
FFFFF350H	Timer 40	TM40	R			○	0000H
FFFFF352H	Compare register 40	CM40	R/W			○	Undefined
FFFFF354H	Timer 41	TM41	R			○	0000H
FFFFF356H	Compare register 41	CM41	R/W			○	Undefined
FFFFF380H	A/D converter mode register 0	ADM0		○	○		00H
FFFFF382H	A/D converter mode register 1	ADM1		○	○		07H
FFFFF390H	A/D conversion result register 0	ADCR0	R			○	Undefined
FFFFF392H	A/D conversion result register 0H	ADCR0H		○	○		
FFFFF394H	A/D conversion result register 1	ADCR1				○	
FFFFF396H	A/D conversion result register 1H	ADCR1H		○	○		
FFFFF398H	A/D conversion result register 2	ADCR2				○	
FFFFF39AH	A/D conversion result register 2H	ADCR2H		○	○		
FFFFF39CH	A/D conversion result register 3	ADCR3				○	
FFFFF39EH	A/D conversion result register 3H	ADCR3H		○	○		
FFFFF3A0H	A/D conversion result register 4	ADCR4				○	
FFFFF3A2H	A/D conversion result register 4H	ADCR4H		○	○		
FFFFF3A4H	A/D conversion result register 5	ADCR5				○	
FFFFF3A6H	A/D conversion result register 5H	ADCR5H		○	○		
FFFFF3A8H	A/D conversion result register 6	ADCR6				○	
FFFFF3AAH	A/D conversion result register 6H	ADCR6H		○	○		
FFFFF3ACH	A/D conversion result register 7	ADCR7				○	
FFFFF3AEH	A/D conversion result register 7H	ADCR7H		○	○		
FFFFF41AH	Port X	PX	R/W	○	○		
★ FFFFF43AH	Port X mode register	PMX	W		○		FFH
★ FFFFF45AH	Port X mode control register	PMCX			○		00H/E0H
FFFFF580H	Port/control select register 0	PCS0	R/W	○	○		00H
FFFFF582H	Port/control select register 1	PCS1		○	○		
FFFFF586H	Port/control select register 3	PCS3		○	○		
FFFFF590H	Port/control select register 8	PCS8		○	○		
FFFFF594H	Port/control select register 10	PCS10		○	○		

(8/8)

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After Reset
				1 bit	8 bits	16 bits	
FFFFF596H	Port/control select register 11	PCS11	R/W	○	○		00H
FFFFF5D0H	DMA disable status register	DDIS	R	○	○		
FFFFF5D2H	DMA restart register	DRST	R/W	○	○		
FFFFF5E0H	DMA trigger factor register 0	DTFR0		○	○		
FFFFF5E2H	DMA trigger factor register 1	DTFR1		○	○		
FFFFF5E4H	DMA trigger factor register 2	DTFR2		○	○		
FFFFF5E6H	DMA trigger factor register 3	DTFR3		○	○		
FFFFF5F0H	DMA channel control register 0	DCHC0		○	○		
FFFFF5F2H	DMA channel control register 1	DCHC1		○	○		
FFFFF5F4H	DMA channel control register 2	DCHC2		○	○		
FFFFF5F6H	DMA channel control register 3	DCHC3		○	○		

3.4.9 Specific registers

Specific registers are registers that are protected from being written with illegal data due to erroneous program execution, etc. The write access of these specific registers is executed in a specific sequence, and if abnormal store operations occur, the system status register (SYS) is notified. The V850E/MS1 has two specific registers, clock control register (CKC) and the power save control register (PSC). For details of the CKC register, refer to **8.3.3** and for details of the PSC register, refer to **8.5.2**.

The access sequence to the specific registers is shown below.

The following sequence shows the data setting of the specific registers.

<1> Provide data in the desired general-purpose register to be set in the specific register.

<2> Write the general-purpose register prepared in <1> in the command register (PRCMD).

<3> Write to the specific register using the general-purpose register prepared in <1> (do this using the following instructions).

- Store instruction (ST/SST instruction)
- Bit operation instruction (SET1/CLR1/NOT1 instruction)

<4> If the system moves to the IDLE or software STOP mode, insert a NOP instruction (1 instruction).

Example

```
<1>  MOV    0x04, r10
<2>  ST.B   r10, PRCMD [r0]
<3>  ST.B   r10, PSC [r0]
<4>  NOP
```

No special sequence is required when reading the specific registers.

Caution Do not write to the PRCMD register or to a specific register by DMA transfer.

Remarks

1. A store instruction to a command register will not be received with an interrupt.
This presupposes that this is done with the continuous store instructions in <1> and <2> above in the program. If another instruction is placed between <1> and <2>, when an interrupt is received by that instruction, the above sequence may not be established, and cause a malfunction, so caution is necessary.

2. The data written in the PRCMD register is dummy data, but use the same general-purpose register for writing to the PRCMD register (<2> in the example above) as was used in setting data in the specific register (<3> in the example above). Addressing is the same in the case where a general-purpose register is used.

3. It is necessary to insert 1 or more NOP instructions just after a store instruction to the PSC register for setting it in the software STOP or IDLE mode. When releasing each power save mode by interrupt, or when resetting after executing interrupt processing, start execution from the next instruction without executing the instruction just after the store instruction.

[Example of Description]

```
ST reg_code, PRCMD   ; PRCMD write
                       (reg_code: Registration code)
ST data, PSC         ; Setting of the PSC register
NOP                  ; Dummy instruction (1 instruction)
(next instruction)   ; Execution routine after releasing the software
                       STOP/IDLE mode
      ⋮                          ⋮
```

The case where bit operation instructions are used in the PSC register settings is the same.

(1) Command register (PRCMD)

The command register (PRCMD) is a register used when write-accessing the specific register to prevent incorrect writing to the specific registers due to the erroneous program execution.

This register can be written in 8-bit units. It becomes undefined in a read cycle.

Occurrence of illegal store operations can be checked by the PRERR bit of the SYS register.

	7	6	5	4	3	2	1	0		
PRCMD	REG7	REG6	REG5	REG4	REG3	REG2	REG1	REG0	Address FFFFF170H	After reset Undefined

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7 to 0</td><td>REG7 to REG0</td><td>Registration Code
<table>
<tr><th>Specific Register</th><th>Registration Code</th></tr>
<tr><td>CKC</td><td>Any 8-bit data</td></tr>
<tr><td>PSC</td><td>Any 8-bit data</td></tr>
</table>
</td></tr>
</table>

(2) System status register (SYS)

This register is assigned status flags showing the operating state of the entire system. This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
SYS	0	0	0	PRERR	0	0	0	UNLOCK	FFFFF078H	0000000×B

Bit Position	Bit Name	Function
4	PRERR	Protection Error Flag This is a cumulative flag that shows that writing to a specific register was not done in the correct sequence and that a protection error occurred [Note]. 0: Protection error did not occur 1: Protection error occurred
0	UNLOCK	Unlock Status Flag This is an exclusive read out flag. It shows that the PLL is in the unlocked state (for details, refer to **8.4 PLL Lockup**). 0: Locked. 1: Unlocked.

Note Operation conditions of PRERR flag

- Set conditions (PRERR = “1”)
  - <1> If the store instruction most recently executed to peripheral I/O does not write data to the PRCMD register, but to the specific register.
  - <2> If the first store instruction executed after the write operation to the PRCMD register is to a peripheral I/O register other than the specific registers.
- Reset conditions: (PRERR = “0”)
  - <1> When “0” is written to the PRERR flag of the SYS register.
  - <2> At system reset.

CHAPTER 4 BUS CONTROL FUNCTION

The V850E/MS1 is provided with an external bus interface function by which external memories such as ROM and RAM, and I/O can be connected.

4.1 Features

- 16-bit/8-bit data bus sizing function
- 8-space chip select output function
- Wait function
  - Programmable wait function, capable of inserting up to 7 wait states for each memory block
  - External wait function via $\overline{\mathrm{WAIT}}$ pin
- Idle state insertion function
- Bus mastership arbitration function
- Bus hold function
- Capable of connecting to external devices via alternate function pins

4.2 Bus Control Pins

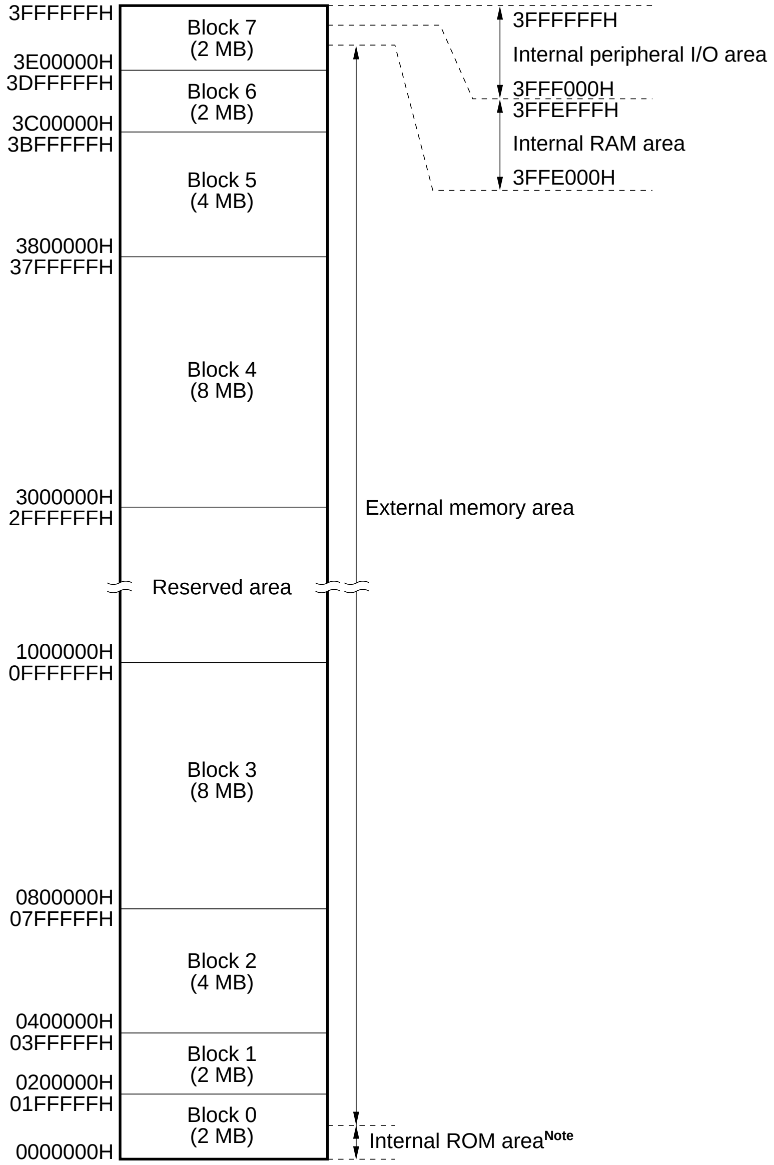
The following pins are used for connecting to external devices:

Bus Control Pin (Function When in the Control Mode)	Function When in the Port Mode	Register Which Performs Port/Control Mode Switching
Data bus (D0 to D7)	P40 to P47 (Port 4)	MM
Data bus (D8 to D15)	P50 to P57 (Port 5)	MM
Address bus (A0 to A7)	PA0 to PA7 (Port A)	MM
Address bus (A8 to A15)	PB0 to PB7 (Port B)	MM
Address bus (A16 to A23)	P60 to P67 (Port 6)	MM
Chip select ($\overline{\mathrm{CS0}}$ to $\overline{\mathrm{CS7}}$, $\overline{\mathrm{RAS0}}$ to $\overline{\mathrm{RAS7}}$, $\overline{\mathrm{IORD}}$, $\overline{\mathrm{IOWR}}$)	P80 to P87 (Port 8)	PMC8
Read/write control ($\overline{\mathrm{LCAS}}$, $\overline{\mathrm{UCAS}}$, $\overline{\mathrm{LWR}}$, $\overline{\mathrm{UWR}}$, $\overline{\mathrm{RD}}$, $\overline{\mathrm{WE}}$, $\overline{\mathrm{OE}}$)	P90 to P93, P95 (Port 9)	PMC9
Bus cycle start ($\overline{\mathrm{BCYST}}$)	P94 (Port 9)	PMC9
External wait control ($\overline{\mathrm{WAIT}}$)	PX6 (Port X)	PMCX
Bus hold control ($\overline{\mathrm{HLDAK}}$, $\overline{\mathrm{HLDRQ}}$)	P96, P97 (Port 9)	PMC9
DRAM refresh control ($\overline{\mathrm{REFRQ}}$)	PX5 (Port X)	PMCX
Internal system clock (CLKOUT)	PX7 (Port X)	PMCX

Remark In the case of single-chip mode 1 and ROM-less modes 0 and 1, when the system is reset, each bus control pin becomes unconditionally valid (however, D8 to D15 are valid only in single-chip mode 1 and ROM-less mode 0). For details, refer to **3.4.6 External expansion mode**.

4.3 Memory Block Function

The 64 MB memory space is divided into memory blocks of 2 MB, 4 MB, and 8 MB units. The programmable wait function and bus cycle operation mode can be independently controlled for each individual memory block.



Note When in single-chip mode 1 and ROM-less modes 0 and 1, this becomes an external memory area. When in single-chip mode 1, addresses 0100000H to 01FFFFF become internal ROM area.

4.4 Bus Cycle Type Control Function

In the V850E/MS1, the following external devices can be connected directly to each memory block.

- SRAM, external ROM, external I/O
- Page ROM
- DRAM

Connected external devices are specified by the bus cycle type configuration register (BCT).

4.4.1 Bus cycle type configuration register (BCT)

This register can be read /written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCT	BT71	BT70	BT61	BT60	BT51	BT50	BT41	BT40	BT31	BT30	BT21	BT20	BT11	BT10	BT01	BT00
Memory block	7		6		5		4		3		2		1		0	

Address FFFFF064H

After reset 0000H

Bit Position	Bit Name	Function
15 to 0	BTn1, BTn0 (n = 7 to 0)	Bus Cycle Type Specifies the external device connected to memory block n.

BTn1	BTn0	External Device Connected Directly to Memory Block n
0	0	SRAM, external ROM, external I/O
0	1	Page ROM
1	0	DRAM[Note]
1	1	Setting prohibited

Note Using the DTC register, one DRAM access type setting can be selected out of 4 types for each memory block (refer to **5.3.5 DRAM type configuration register (DTC)**).

Caution Write to the BCT register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the BCT register is complete. However, it is possible to access an external memory area whose initialization is complete.

The chip select signal ($\overline{\text{CS0}}$/$\overline{\text{RAS0}}$ to $\overline{\text{CS7}}$/$\overline{\text{RAS7}}$) is output as follows in correspondence with blocks 0 to 7.

External Device Memory Block	SRAM, External ROM, External I/O Page ROM	DRAM
Block 0[Note 1]	$\overline{\text{CS0}}$	$\overline{\text{RAS0}}$
Block 1	$\overline{\text{CS1}}$	$\overline{\text{RAS1}}$
Block 2	$\overline{\text{CS2}}$	$\overline{\text{RAS2}}$
Block 3	$\overline{\text{CS3}}$	$\overline{\text{RAS3}}$
Block 4	$\overline{\text{CS4}}$	$\overline{\text{RAS4}}$
Block 5	$\overline{\text{CS5}}$	$\overline{\text{RAS5}}$
Block 6	$\overline{\text{CS6}}$	$\overline{\text{RAS6}}$
Block 7[Note 2]	$\overline{\text{CS7}}$	$\overline{\text{RAS7}}$

Notes 1. Except internal ROM area.

2. Except internal RAM area and internal peripheral I/O area.

4.5 Bus Access

4.5.1 Number of access clocks

The number of basic clocks necessary for accessing each resource is as follows.

Bus Cycle Configuration				Instruction Fetch		Operand Data Access	
Resource (Bus Width)				Normal Access	Burst Access	Normal Access	Burst Access
Internal ROM (32 bits)				1	—	3	—
Internal RAM (32 bits)				1 or 2	—	1	—
Internal peripheral I/O (16 bits)				—	—	3 + n	—
External device	SRAM, external ROM, external I/O (16/8 bits)			2 + n	—	2 + n	—
		During DMA flyby transfer		—	—	2 + n	—
	Page ROM (16/8 bits)			2 + n	2 + n	2 + n	2 + n
	High-speed page DRAM (16/8 bits)			3 + n	2 + n	3 + n	2 + n
		During DMA flyby transfer	During read	—	—	3 + n	2 + n
			During write	—	—	3 + n	3 + n
	EDO DRAM (16/8 bits)			3 + n	1 + n	3 + n	1 + n
		During DMA flyby transfer	During read	—	—	3 + n	2 + n
			During write	—	—	3 + n	3 + n

Remarks 1. Unit: Clock/access

2. n: Number of wait insertions

(1) Internal peripheral I/O interface

The contents of the access to internal peripheral I/O are not output to the external bus. Therefore, during instruction fetch access, internal peripheral I/O access can be performed in parallel.

Internal peripheral I/O access is basically 3-clock access. However, on some occasions, access to internal peripheral I/O registers with timer/counter functions also involves a wait.

Internal Peripheral I/O Register	Access	Waits	Clock Cycles
CC1n0 to CC1n3, TM1n (n = 0 to 5)	Read	1	4
	Write	0/1	3/4
CM40, CM41	Read	0	3
	Write	0/1	3/4
TM40, TM41	Read	0/1	3/4
	Write	0	3
Other	Read	0	3
	Write	0	3

4.5.2 Bus sizing function

The V850E/MS1 is provided with a bus sizing function that is used to control the data bus width of each memory block.

The data bus width is specified by using the bus size configuration register (BSC).

(1) Bus size configuration register (BSC)

This register can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
BSC	BS71	BS70	BS61	BS60	BS51	BS50	BS41	BS40	BS31	BS30	BS21	BS20	BS11	BS10	BS01	BS00	FFFFF066H	**Note**
Memory block	7		6		5		4		3		2		1		0			

Note When in single-chip modes 0, 1: 5555H
When in ROM-less mode 0: 5555H
When in ROM-less mode 1: 0000H

Bit Position	Bit Name	Function
15 to 0	BSn1, BSn0 (n = 7 to 0)	Data Bus Width Sets the data bus width of memory block n. BSn1 / BSn0 / Data Bus Width of Memory Block n 0 / 0 / 8 bits 0 / 1 / 16 bits 1 / Optional / RFU (Reserved)

BSn1	BSn0	Data Bus Width of Memory Block n
0	0	8 bits
0	1	16 bits
1	Optional	RFU (Reserved)

Cautions 1. Write to the BSC register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the BSC register is complete. However, it is possible to access an external memory area whose initialization is complete.

2. The in-circuit emulator (IE-703102-MC) for the V850E/MS1 does not support 8-bit width external ROM emulation.

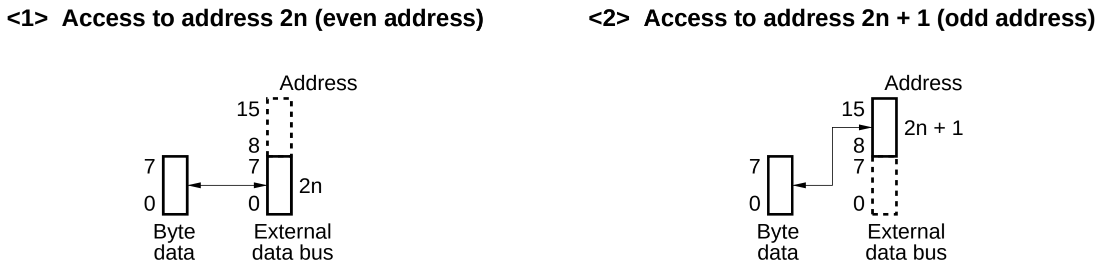
★ **3. When 8-bit data bus width is selected, only the write signal $\overline{\text{LWR}}$ becomes active, $\overline{\text{UWR}}$ does not become active.**

4.5.3 Bus width

V850E/MS1 carries out peripheral I/O access and external memory access in 8, 16, or 32 bits. The following shows the operation for each access. All data is accessed in order from the lower side.

(1) Byte access (8 bits)

(a) When the data bus width is 16 bits

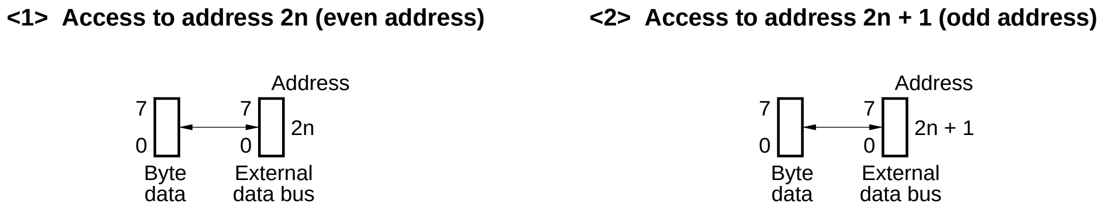


<1> Access to address 2n (even address)

<2> Access to address 2n + 1 (odd address)

Remark n = 0, 1, 2, 3, ···

(b) When the data bus width is 8 bits



<1> Access to address 2n (even address)

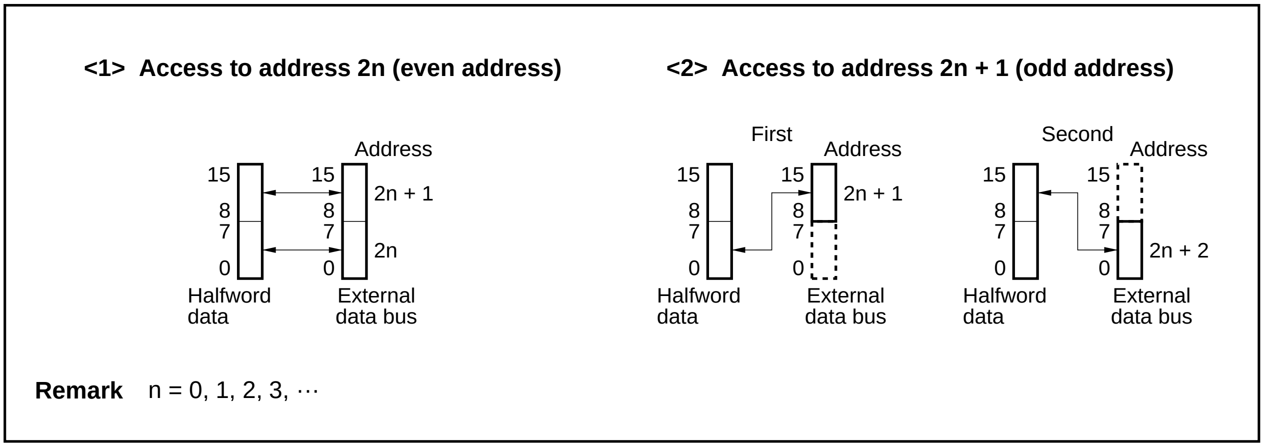
<2> Access to address 2n + 1 (odd address)

Remark n = 0, 1, 2, 3, ···

(2) Halfword access (16 bits)

In halfword access to external memory, data is exchanged as is, or accessed in the order of lower byte, then higher byte.

(a) When the data bus width is 16 bits

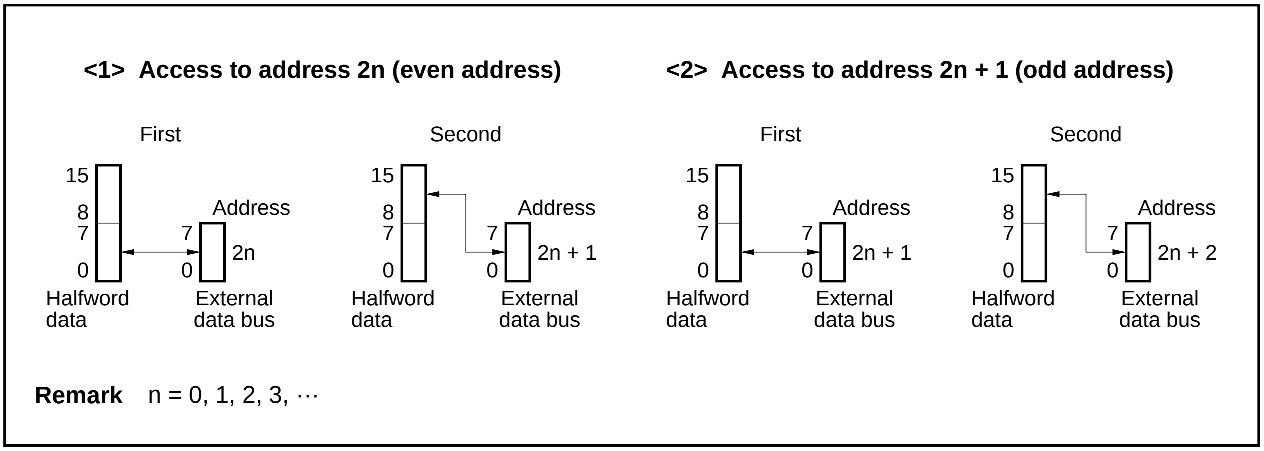


<1> Access to address 2n (even address)

<2> Access to address 2n + 1 (odd address)

Remark n = 0, 1, 2, 3, ⋯

(b) When the data bus width is 8 bits



<1> Access to address 2n (even address)

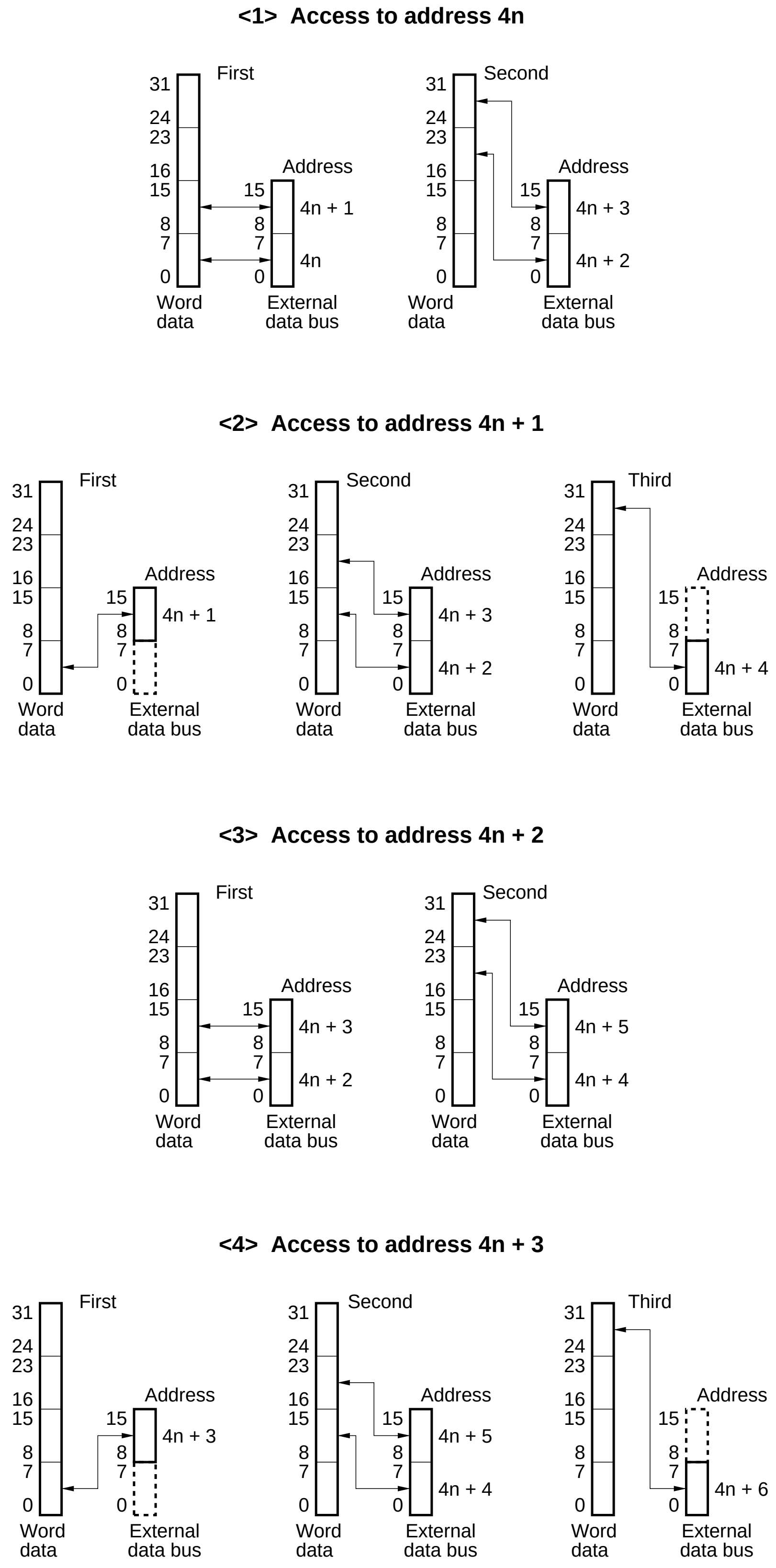
<2> Access to address 2n + 1 (odd address)

Remark n = 0, 1, 2, 3, ⋯

(3) Word access (32 bits)

In word access to external memory, data is accessed in order from the lower halfword, then the higher halfword, or in order from the lowest byte to the highest byte.

(a) When the data bus width is 16 bits



<1> Access to address 4n

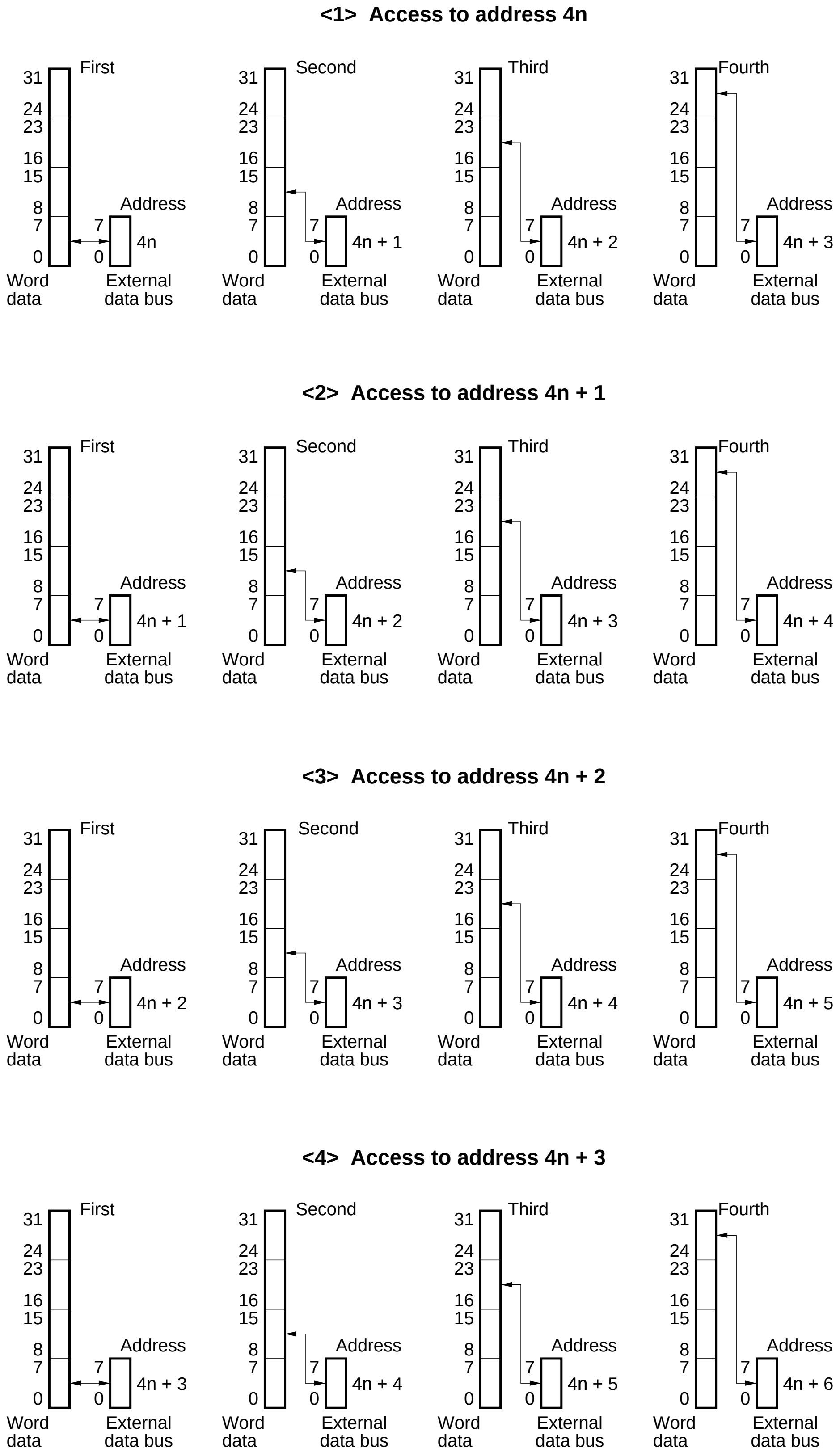
<2> Access to address 4n + 1

<3> Access to address 4n + 2

<4> Access to address 4n + 3

Remark n = 0, 1, 2, 3, ⋯

(b) When the data bus width is 8 bits



<1> Access to address 4n

<2> Access to address 4n + 1

<3> Access to address 4n + 2

<4> Access to address 4n + 3

Remark n = 0, 1, 2, 3, ⋯

4.6 Wait Function

4.6.1 Programmable wait function

With the aim of realizing easy interfacing with low-speed memory or with I/Os, it is possible to insert up to 7 data wait states with respect to the starting bus cycle for each memory block.

The number of wait states can be set by data wait control registers 1 and 2 (DWC1, DWC2) and can be specified by program. Just after system reset, all blocks have 7 data wait states inserted.

(1) Data wait control registers 1, 2 (DWC1, DWC2)

It is possible to read/write the DWC1 register in 16-bit units and the DWC2 register in 8/1-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
DWC1	DW71	DW70	DW61	DW60	DW51	DW50	DW41	DW40	DW31	DW30	DW21	DW20	DW11	DW10	DW01	DW00	Address FFFFF060H	After reset FFFFH
Memory block	7		6		5		4		3		2		1		0			

	7	6	5	4	3	2	1	0		
DWC2	DW72	DW62	DW52	DW42	DW32	DW22	DW12	DW02	Address FFFFF06AH	After reset FFH
Memory block	7	6	5	4	3	2	1	0		

Register Name	Bit Position	Bit Name	Function
DWC1	15 to 0	DWn1, DWn0 (n = 7 to 0)	Data Wait Specifies the number of wait states inserted in memory block n. Registers DWC1 and DWC2 are set in combination.
DWC2	7 to 0	DWn2 (n = 7 to 0)	

DWn2	DWn1	DWn0	Number of Wait States Inserted in Memory Block n
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Cautions 1. The internal ROM area and internal RAM area are not subject to programmable waits and ordinarily no wait access is carried out. Neither is the internal peripheral I/O area subject to programmable wait states, with wait control performed only by each peripheral function.

2. In the following cases, the settings of registers DWC1 and DWC2 are invalid (wait control is performed by each memory controller).

- **DRAM access**
- **Page ROM on-page access**

3. Write to the DWC1 and DWC2 registers after reset, and then do not change the set values. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the DWC1 and DWC2 registers is complete. However, it is possible to access an external memory area whose initialization is complete.

4.6.2 External wait function

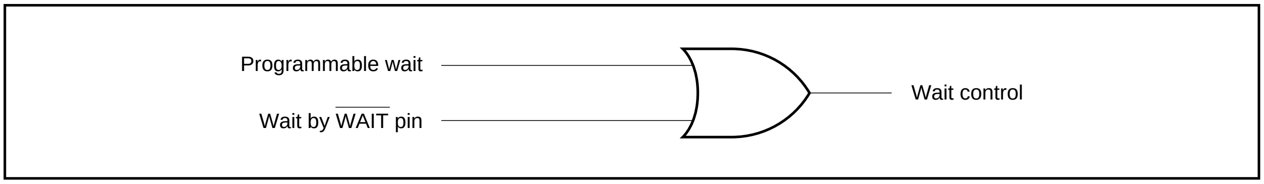
When an extremely slow device, I/O, or asynchronous system is connected, any number of wait states can be inserted in a bus cycle by the external wait pin ($\overline{\text{WAIT}}$) to synchronize with the external device.

Just as with programmable waits, access to internal ROM, internal RAM and internal peripheral I/O areas cannot be controlled by external waits.

Input of the external $\overline{\text{WAIT}}$ signal can be done asynchronously to CLKOUT and is sampled at the falling edge of the clock in the T1 and TW states of a bus cycle. If the setup/hold time in the sampling timing is not satisfied, a wait may or may not be inserted in the next state.

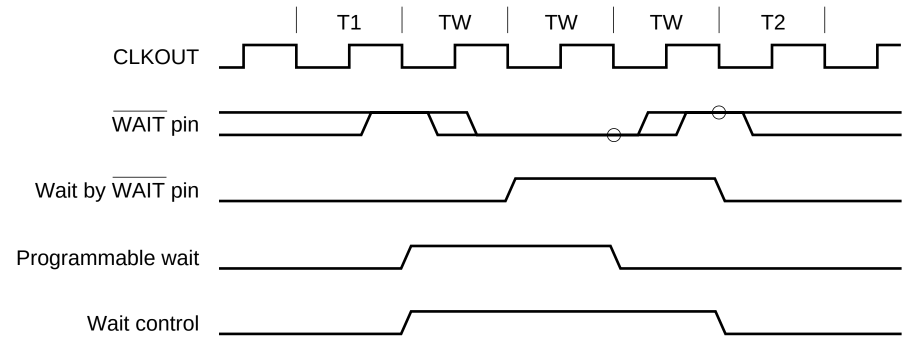
4.6.3 Relationship between programmable wait and external wait

A wait cycle is inserted as a result of an OR operation between the wait cycle specified by the set value of programmable wait and the wait cycle controlled by the $\overline{\text{WAIT}}$ pin. In other words, the number of wait cycles is determined by whichever has the most cycles.



For example, if the programmable wait is two waits, and the timing of the $\overline{\text{WAIT}}$ pin input signal is as illustrated below, three wait states will be inserted in the bus cycle.

Figure 4-1. Example of Inserting Wait States


T1
TW
TW
TW
T2
CLKOUT
$\overline{\text{WAIT}}$ pin
Wait by $\overline{\text{WAIT}}$ pin
Programmable wait
Wait control

Remark ○: Sampling timing

4.6.4 Bus cycles in which the wait function is valid

In the V850E/MS1, the number of waits can be specified according to the type of memory specified for each memory block.

The registers which set the bus cycles and waits in which the wait function is valid are as shown below.

Table 4-1. Bus Cycles in Which the Wait Function Is Valid (1/2)

Bus Cycle			Type of Wait	Programmable Wait Setting		Wait by WAIT Pin
				Higher Order: Register Lower Order: Bit	Number of Waits	
SRAM, external ROM, external I/O cycle			Data access wait	DWC1, DWC2 DWxx	0 to 7	○
Page ROM cycle	Off-page		Data access wait	DWC1, DWC2 DWxx	0 to 7	○
	On-page		Data access wait	PRC PRW0 to PRW2	0 to 7	○
EDO DRAM, high-speed page DRAM cycle	Read access	Off-page	RAS pre-charge	DRCn RPC0n, RPC1n	0 to 3	×
			Row address hold	DRCn RHC0n, RHC1n	0 to 3	×
			Data access wait	DRCn DAC0n, DAC1n	0 to 3	**Note**
		On-page	CAS pre-charge	DRCn CPC0n, CPC1n	0 to 3	×
			Data access wait	DRCn DAC0n, DAC1n	0 to 3	×
	Write access	Off-page	RAS pre-charge	DRCn RPC0n, RPC1n	0 to 3	×
			Row address hold	DRCn RHC0n, RHC1n	0 to 3	**Note**
			Data access wait	DRCn DAC0n, DAC1n	0 to 3	×
		On-page	CAS pre-charge	DRCn CPC0n, CPC1n	0 to 3	×
			Data access wait	DRCn DAC0n, DAC1n	0 to 3	×
CBR refresh cycle			RAS pre-charge	RWC RRW0, RRW1	0 to 3	×
			RAS active width	RWC RCW0 to RCW2	0 to 7	×

Note EDO DRAM cycle: ×
High-speed page DRAM cycle: ○

Remarks 1. ○: Valid ×: Invalid

2. n = 0 to 3
xx = 00 to 02, 10 to 12, 20 to 22, 30 to 32, 40 to 42, 50 to 52, 60 to 62, 70 to 72

Table 4-1. Bus Cycles in Which the Wait Function Is Valid (2/2)

Bus Cycle			Type of Wait		Programmable Wait Setting		Wait by $\overline{\text{WAIT}}$ Pin
					Higher Order: Register Lower Order: Bit	Number of Waits	
CBR self-refresh cycle			RAS pre-charge		RWC RRW0, RRW1	0 to 3	×
			RAS active width		RWC RCW0 to RCW2	0 to 7	×
			Self-refresh release width		RWC SRW0 to SRW2	0 to 14	×
DMA flyby transfer cycle	External I/O ↔ SRAM		Data access wait	TW	DWC1, DWC2 DWxx	0 to 7	○
				TF	FDW FDWm	0, 1	×
	DRAM → External I/O	Off-page	RAS pre-charge		DRCn RPC0n, RPC1n	0 to 3	×
			Row address hold		DRCn RHC0n, RHC1n	0 to 3	×
			Data access wait	TW	DRCn DAC0n, DAC1n	0 to 3	○
				TF	FDW FDWm	0, 1	×
		On-page	CAS pre-charge		DRCn CPC0n, CPC1n	0 to 3	×
			Data access wait	TW	DRCn DAC0n, DAC1n	0 to 3	○
				TF	FDW FDWm	0, 1	×
	External I/O → DRAM	Off-page	RAS pre-charge		DRCn RPC0n, RPC1n	0 to 3	×
			Row address hold		DRCn RHC0n, RHC1n	0 to 3	○
			Data access wait	TW	DRCn DAC0n, DAC1n	0 to 3	×
				TF	FDW FDWm	0, 1	×
		On-page	CAS pre-charge		DRCn CPC0n, CPC1n	1 to 3	○
			Data access wait	TW	DRCn DAC0n, DAC1n	0 to 3	×
				TF	FDW FDWm	0, 1	×

Remarks **1.** ○: Valid ×: Invalid

2. n = 0 to 3

m = 0 to 7

xx = 00 to 02, 10 to 12, 20 to 22, 30 to 32, 40 to 42, 50 to 52, 60 to 62, 70 to 72

4.7 Idle State Insertion Function

To facilitate interfacing with low-speed memory devices, an idle state (TI) can be inserted into the current bus cycle after the T2 state in order to meet the data output float delay time (t_{DF}) on memory read accesses for each memory block. The bus cycle following the T2 state starts after the idle state is inserted.

Specifying insertion of the idle state is programmable by setting the bus cycle control register (BCC).

Immediately after the system reset is cancelled, idle state insertion is automatically programmed for all memory blocks.

The idle state is inserted only if the read cycle is followed by a write cycle.

(1) Bus cycle control register (BCC)

This register can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
BCC	BC71	BC70	BC61	BC60	BC51	BC50	BC41	BC40	BC31	BC30	BC21	BC20	BC11	BC10	BC01	BC00	Address FFFFF062H	After reset 5555H
Memory block	7		6		5		4		3		2		1		0			

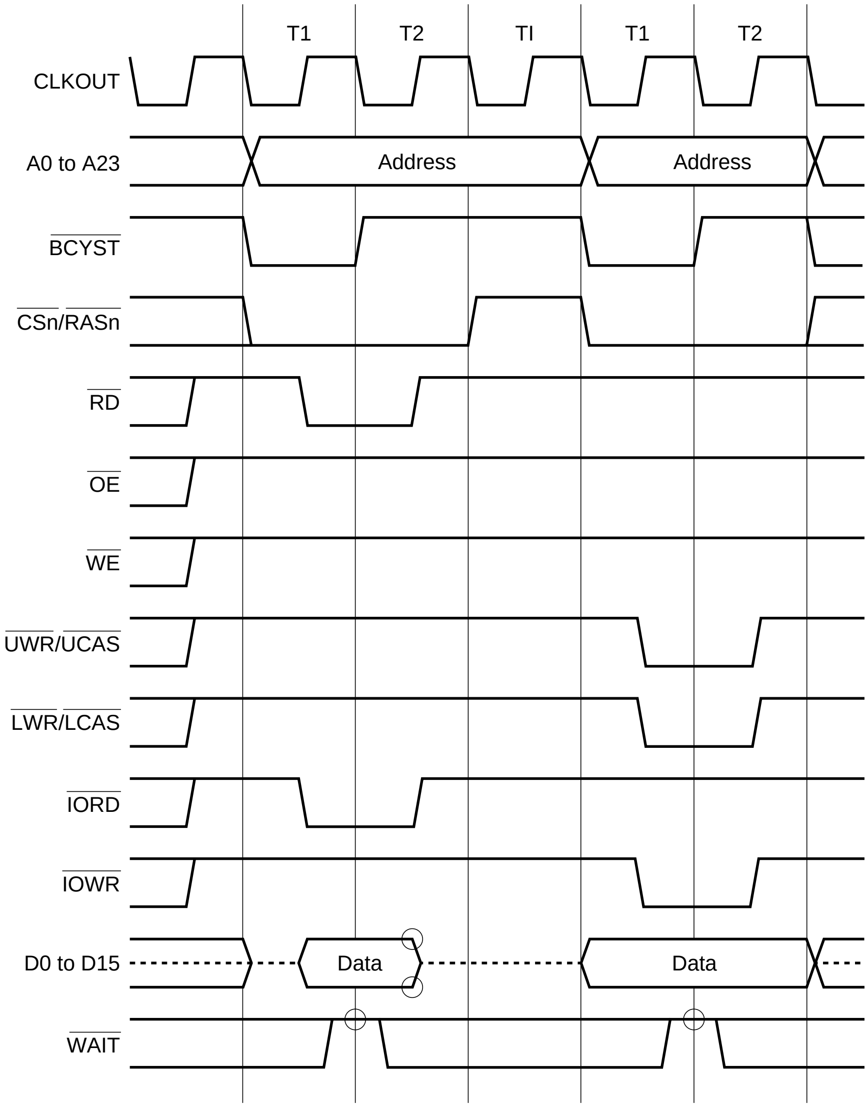
Bit Position	Bit Name	Function
15 to 0	BCn1, BCn0 (n = 7 to 0)	Bus Cycle Specifies insertion of an idle state in memory block n.

BCn1	BCn0	Idle State in Memory Block n
0	0	Not inserted
0	1	Inserted
1	Optional	RFU (Reserved)

Cautions 1. The internal ROM area, internal RAM area and internal peripheral I/O area are not subject to insertion of an idle state.

2. Write to the BCC register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the BCC register is complete. However, it is possible to access an external memory area whose initialization is complete.

(2) Idle state insertion timing



Remarks **1.** The circle indicates the sampling timing.

2. The broken lines indicate high impedance.

3. n = 0 to 7

4.8 Bus Hold Function

4.8.1 Outline of function

If pins P96 and P97 are specified in the control mode, the $\overline{\text{HLDAK}}$ and $\overline{\text{HLDRQ}}$ functions become valid.

If it is determined that the $\overline{\text{HLDRQ}}$ pin has become active (low level) as a bus acquisition request from another bus master, the external address/data bus and each strobe pin are shifted to high impedance and released (bus hold state). If the $\overline{\text{HLDRQ}}$ pin becomes inactive (high level) and the bus acquisition request is canceled, driving of these pins begins again.

During the bus hold interval, internal operations in the V850E/MS1 continue until there is external memory access.

The bus hold state can be known by the $\overline{\text{HLDAK}}$ pin becoming active (low level).

In a multiprocessor configuration, etc., a system that has multiple bus masters can be configured.

Note that bus hold requests are not received with the following timings.

Caution The $\overline{\text{HLDRQ}}$ function is invalid during the reset period. When the $\overline{\text{RESET}}$ pin and $\overline{\text{HLDRQ}}$ pin are made active simultaneously, and then the $\overline{\text{RESET}}$ pin is made inactive, the $\overline{\text{HLDAK}}$ pin becomes active after a one-clock idle cycle has been inserted. Note that for a power-on reset, even if the $\overline{\text{RESET}}$ pin and $\overline{\text{HLDRQ}}$ pin are made active simultaneously, and then the $\overline{\text{RESET}}$ pin is made inactive, the $\overline{\text{HLDAK}}$ pin does not become active. When a bus master other than the V850E/MS1 is externally connected, execute arbitration at the moment of power-on using the $\overline{\text{RESET}}$ signal.

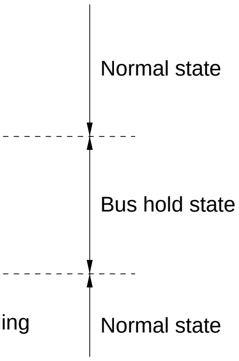
State	Data Bus Width	Access Configuration	Timing in Which Bus Hold Request Will Not Be Received
CPU bus lock	16 bits	Word access to even address	Between 1st and 2nd times
		Word access to odd address	Between 1st and 2nd times
			Between 2nd and 3rd times
		Halfword access to odd address	Between 1st and 2nd times
	8 bits	Word access	Between 1st and 2nd times
			Between 2nd and 3rd times
			Between 3rd and 4th times
		Halfword access	Between 1st and 2nd times
Read modify write access to bit operation instruction	—	—	Between read access and write access

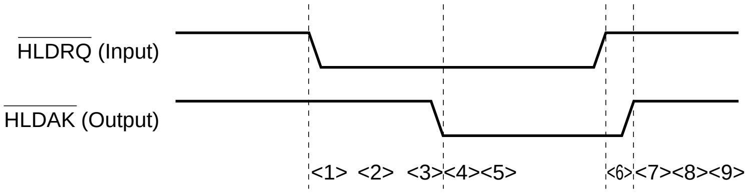
4.8.2 Bus hold procedure

The procedure of the bus hold function is illustrated below.

<1> $\overline{\text{HLDRQ}}$ = 0 accepted
<2> All bus cycle start request pending
<3> End of current bus cycle
<4> Transition to bus idle state
<5> $\overline{\text{HLDAK}}$ = 0

<6> $\overline{\text{HLDRQ}}$ = 1 accepted
<7> $\overline{\text{HLDAK}}$ = 1
<8> Clears bus cycle start request pending
<9> Start of bus cycle



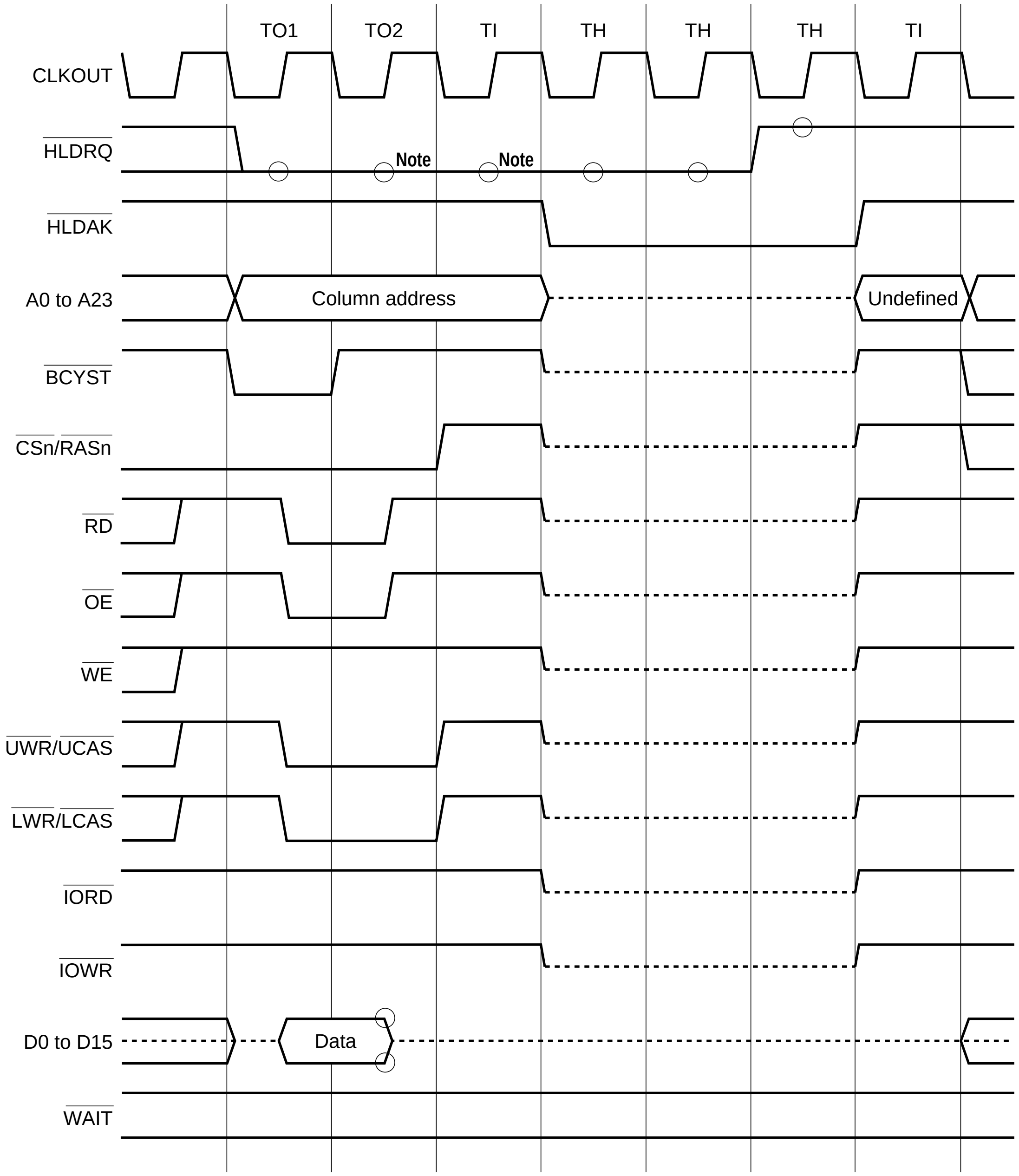


4.8.3 Operation in power save mode

In the STOP or IDLE mode, the internal system clock is stopped. Consequently, the bus hold state is not accepted and set even if the $\overline{\text{HLDRQ}}$ pin becomes active.

In the HALT mode, the $\overline{\text{HLDAK}}$ pin immediately becomes active when the $\overline{\text{HLDRQ}}$ pin becomes active, and the bus hold state is set. When the $\overline{\text{HLDRQ}}$ pin becomes inactive, the $\overline{\text{HLDAK}}$ pin becomes inactive. As a result, the bus hold state is cleared, and the HALT mode is set again.

4.8.4 Bus hold timing



Note If HLDRQ signal is inactive (high level) at this sampling timing, bus hold state is not entered.

Remarks **1.** The circle indicates the sampling timing.

2. The broken lines indicate high impedance.

3. n = 0 to 7

4. Timing from DRAM access to bus hold state.

4.9 Bus Priority Order

There are five external bus cycles: bus hold, instruction fetch, operand data access, DMA cycle and refresh cycle.

Bus hold has the highest priority, then the refresh cycle, DMA cycle, instruction fetch and operand data access, in descending order.

Between read access and write access in read modify write access, an instruction fetch may be inserted.

Also, between bus access and bus access during CPU bus lock, an instruction fetch may be inserted.

Table 4-2. Bus Priority Order

Priority Order	External Bus Cycle	Bus Master
High	Bus hold	External device
	Refresh cycle	DRAM controller
	DMA cycle	DMA controller
	Instruction fetch	CPU
Low	Operand data access	CPU

4.10 Boundary Operation Conditions

4.10.1 Program space

(1) Branching to the peripheral I/O area or successive fetch from the internal RAM area to the internal peripheral I/O area is prohibited. In terms of hardware, fetching the NOP op code continues, and fetching from the external memory is not performed.

(2) If a branch instruction exists at the upper limit of the internal RAM area, a pre-fetch operation (invalid fetch) that straddles over the internal peripheral I/O area does not occur when instruction fetch is performed.

(3) In burst fetch mode, if an instruction fetch is performed for contiguous memory blocks, the burst fetch is terminated at the upper limit of a block, and the start-up cycle is started at the lower limit of the next block.

(4) Burst fetch is valid only in the external memory area. In memory block 7, it is terminated when the internal address count value has reached the upper limit of the external memory area.

4.10.2 Data space

The V850E/MS1 incorporates an address misalign function.

Through this function, regardless of the data format (word data, halfword data), data can be placed in all addresses. However, in the case of word data and halfword data, if data is not subject to boundary alignment, the bus cycle will be generated at least 2 times and bus efficiency will drop.

(1) In the case of halfword length data access

When the address's lowest bit is a 1, the byte length bus cycle will be generated 2 times.

(2) In the case of word length data access

(a) When the address's lowest bit is a 1, bus cycles will be generated in the order of byte length bus cycle, halfword length bus cycle, and byte length bus cycle.

(b) When the address's lower 2 bits are 10, the halfword length bus cycle will be generated 2 times.

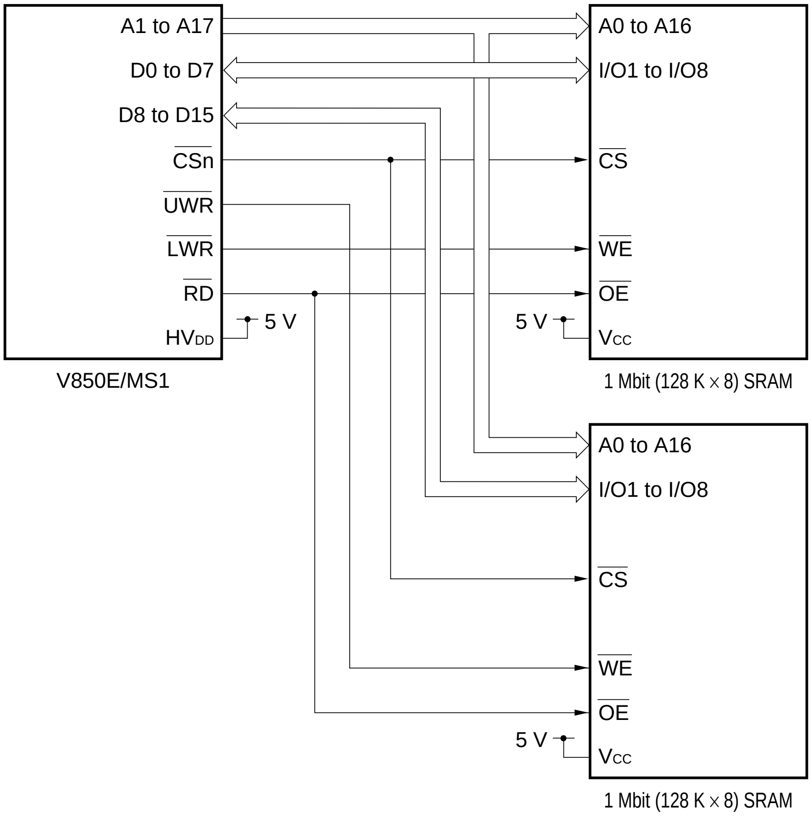
[MEMO]

5.1 SRAM, External ROM, External I/O Interface

5.1.1 SRAM connections

An example of connection to SRAM is shown below.

Figure 5-1. Example of Connection to SRAM

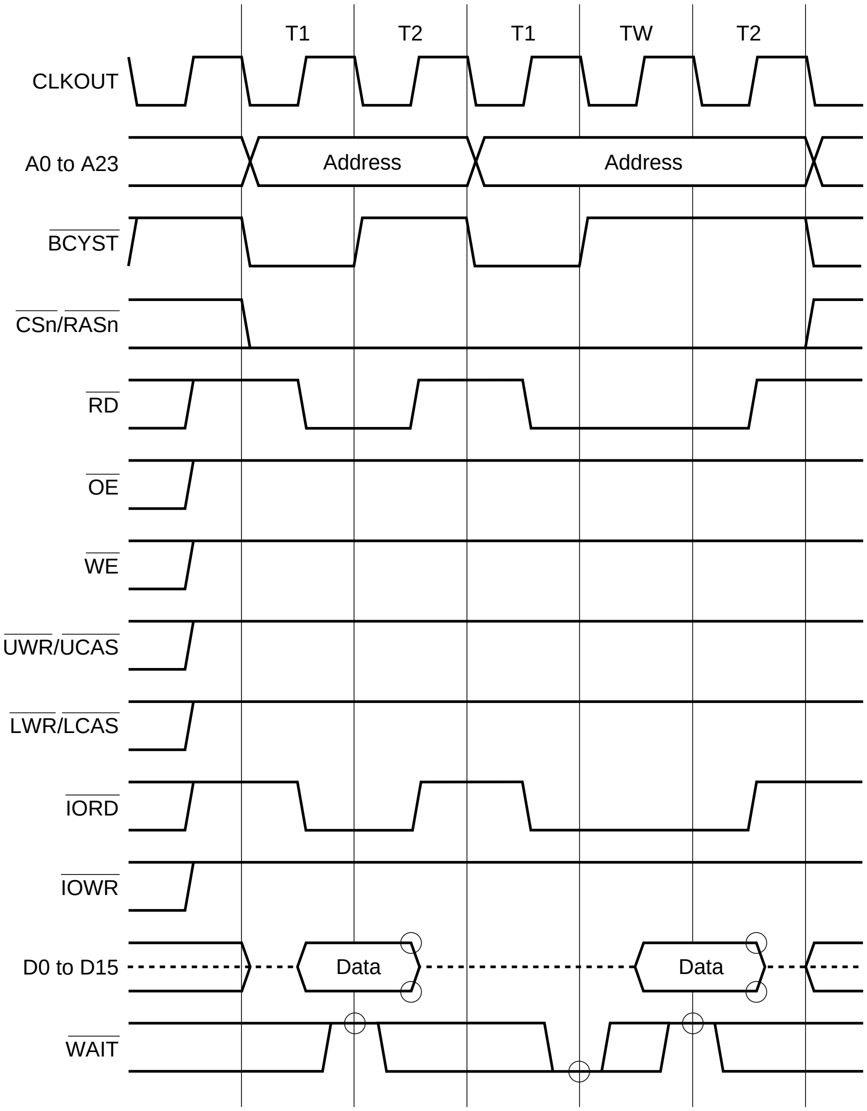


Remark n = 0 to 7

5.1.2 SRAM, external ROM, external I/O access

Figure 5-2. SRAM, External ROM, External I/O Access Timing (1/4)

(a) During read



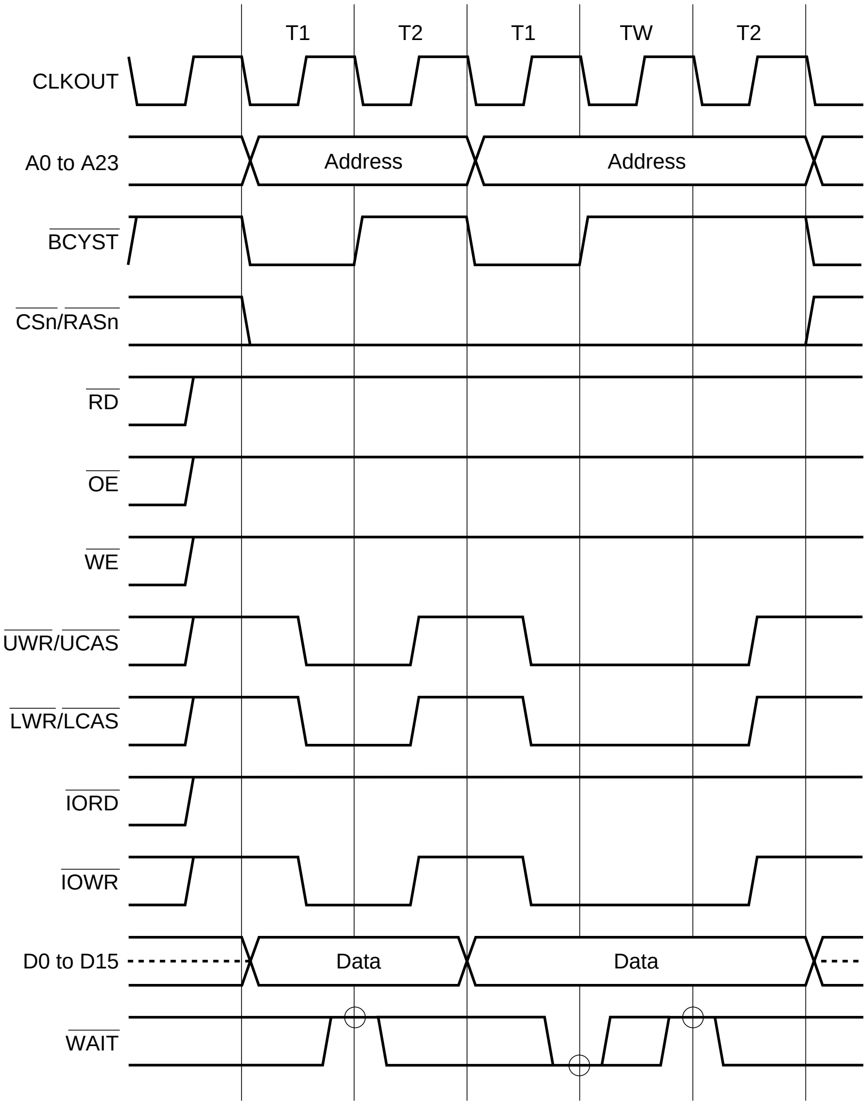
Remarks **1.** The circle indicates the sampling timing.

2. The broken lines indicate high impedance.

3. n = 0 to 7

Figure 5-2. SRAM, External ROM, External I/O Access Timing (2/4)

(b) During write



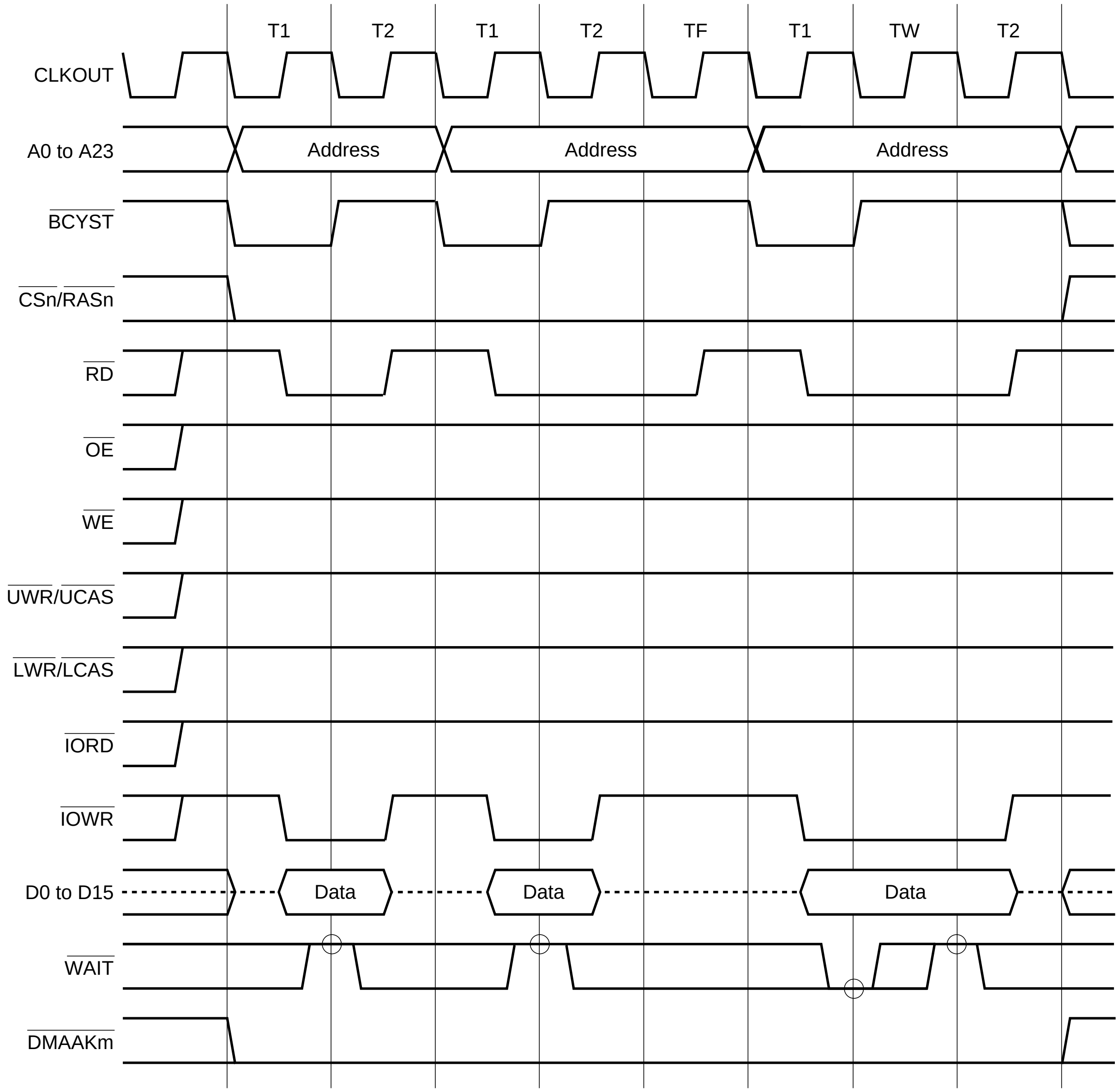
Remarks 1. The circle indicates the sampling timing.

2. The broken lines indicate high impedance.

3. n = 0 to 7

Figure 5-2. SRAM, External ROM, External I/O Access Timing (3/4)

(c) During DMA flyby transfer (SRAM → External I/O)



Remarks **1.** The circle indicates the sampling timing.

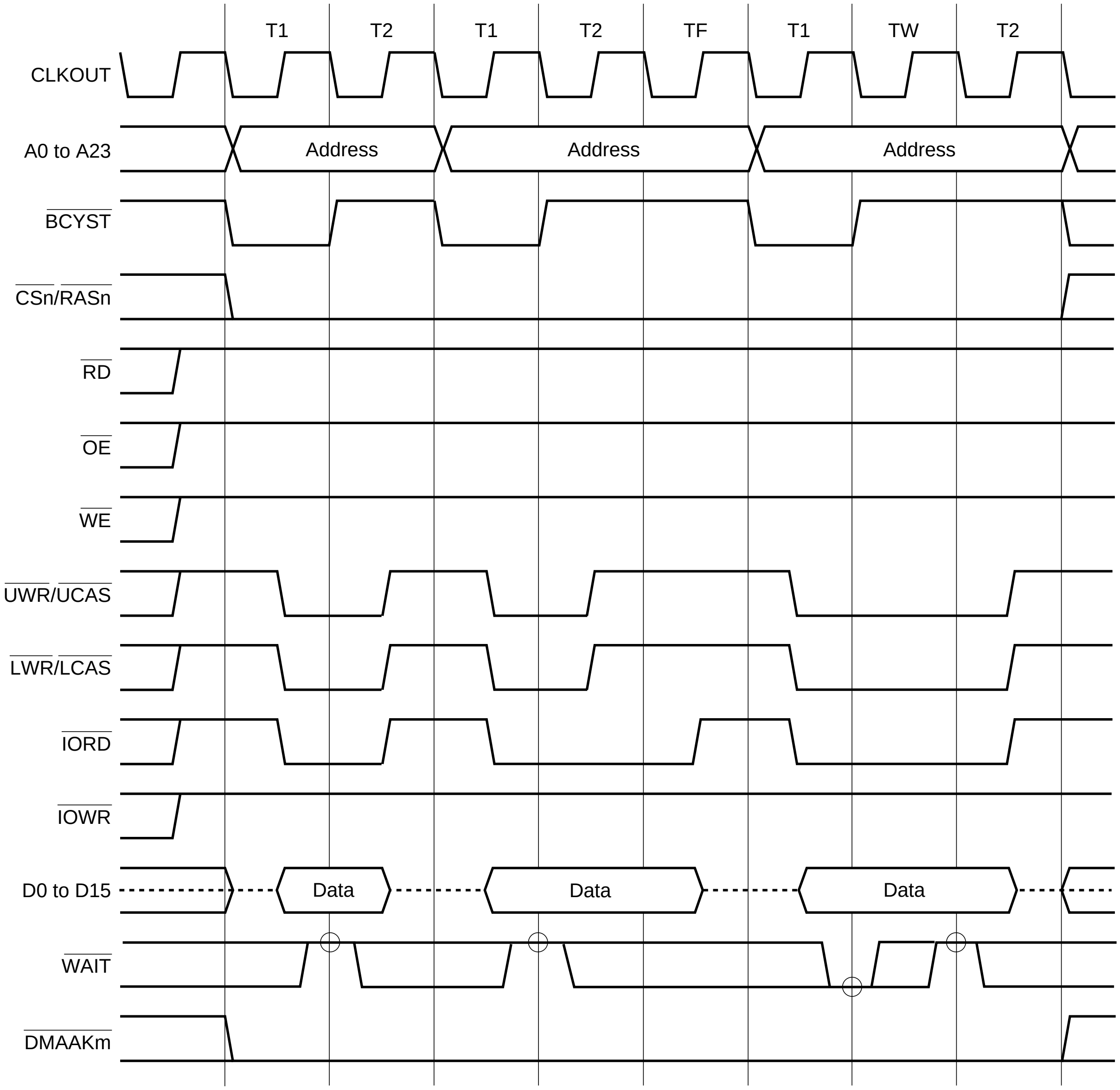
2. The broken lines indicate high impedance.

3. n = 0 to 7

4. m = 0 to 3

Figure 5-2. SRAM, External ROM, External I/O Access Timing (4/4)

(d) During DMA flyby transfer (External I/O → SRAM)



Remarks 1. The circle indicates the sampling timing.
2. The broken lines indicate high impedance.
3. n = 0 to 7
4. m = 0 to 3

5.2 Page ROM Controller (ROMC)

The page ROM controller (ROMC) is for access to ROM (page ROM) with a page access function.

Comparison of addresses with the immediately previous bus cycle is carried out and wait control for normal access (off-page) and page access (on-page) is executed. This controller is capable of handling page widths of from 8 to 64 bytes.

5.2.1 Features

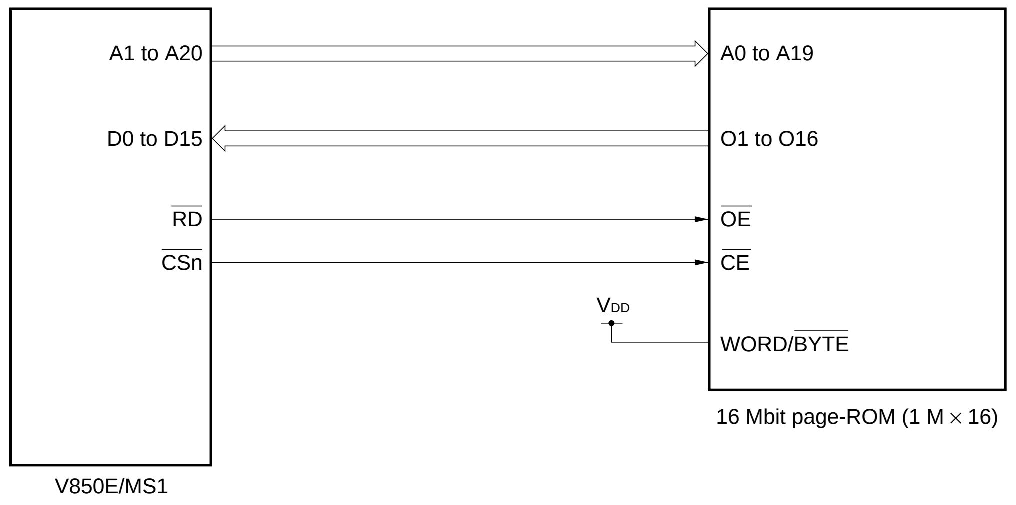
- It can connect directly to 8-bit/16-bit page ROM.
- When the bus width is 16 bits, it can handle 4/8/16/32-word page access.
  When the bus width is 8 bits, it can handle 8/16/32/64-word page access.
- Individual wait settings (0 to 7 waits) for off-page and on-page are possible.

5.2.2 Page ROM connections

Examples of page ROM connections are shown below.

Figure 5-3. Example of Page ROM Connections (1/2)

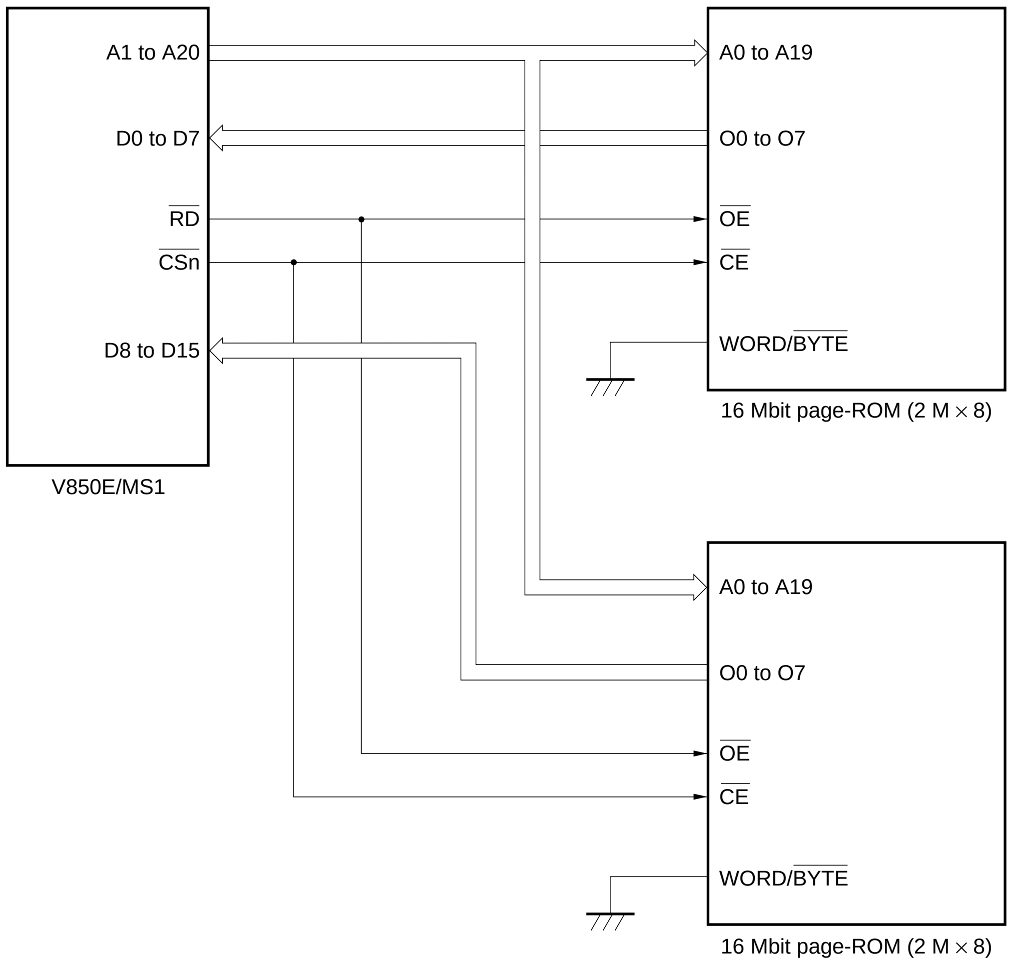
(a) In the case of 16 Mbit (1 M × 16) page ROM



Remark n = 0 to 7

Figure 5-3. Example of Page ROM Connections (2/2)

(b) In the case of 16 Mbit (2 M × 8) page ROM



Remark n = 0 to 7

5.2.3 On-page/off-page judgment

Whether a page ROM cycle is on-page or off-page is judged by latching the address of the previous cycle and comparing it with the address of the current cycle.

Using the page ROM configuration register (PRC), one of the addresses (A3 to A5) is set as the masking address (no comparison is made) according to the configuration of the connected page ROM and the number of continuously readable bits.

Figure 5-4. On-Page/Off-Page Judgment for Page ROM Connection (1/2)

(a) In the case of 16 Mbit (1 M × 16) page ROM (4-word page access)

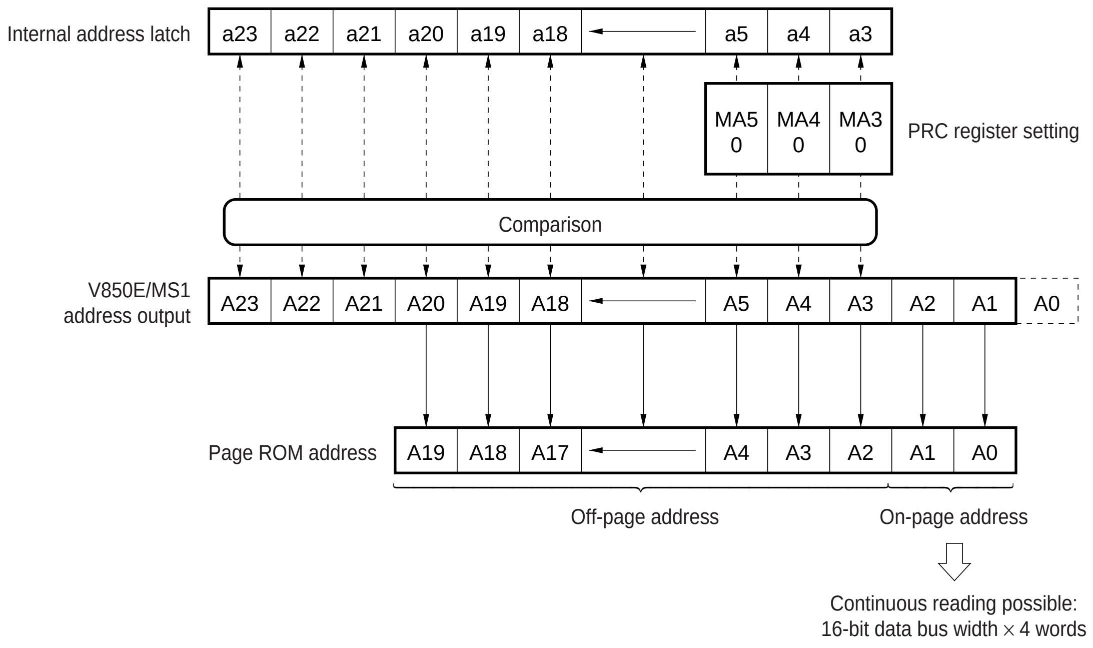
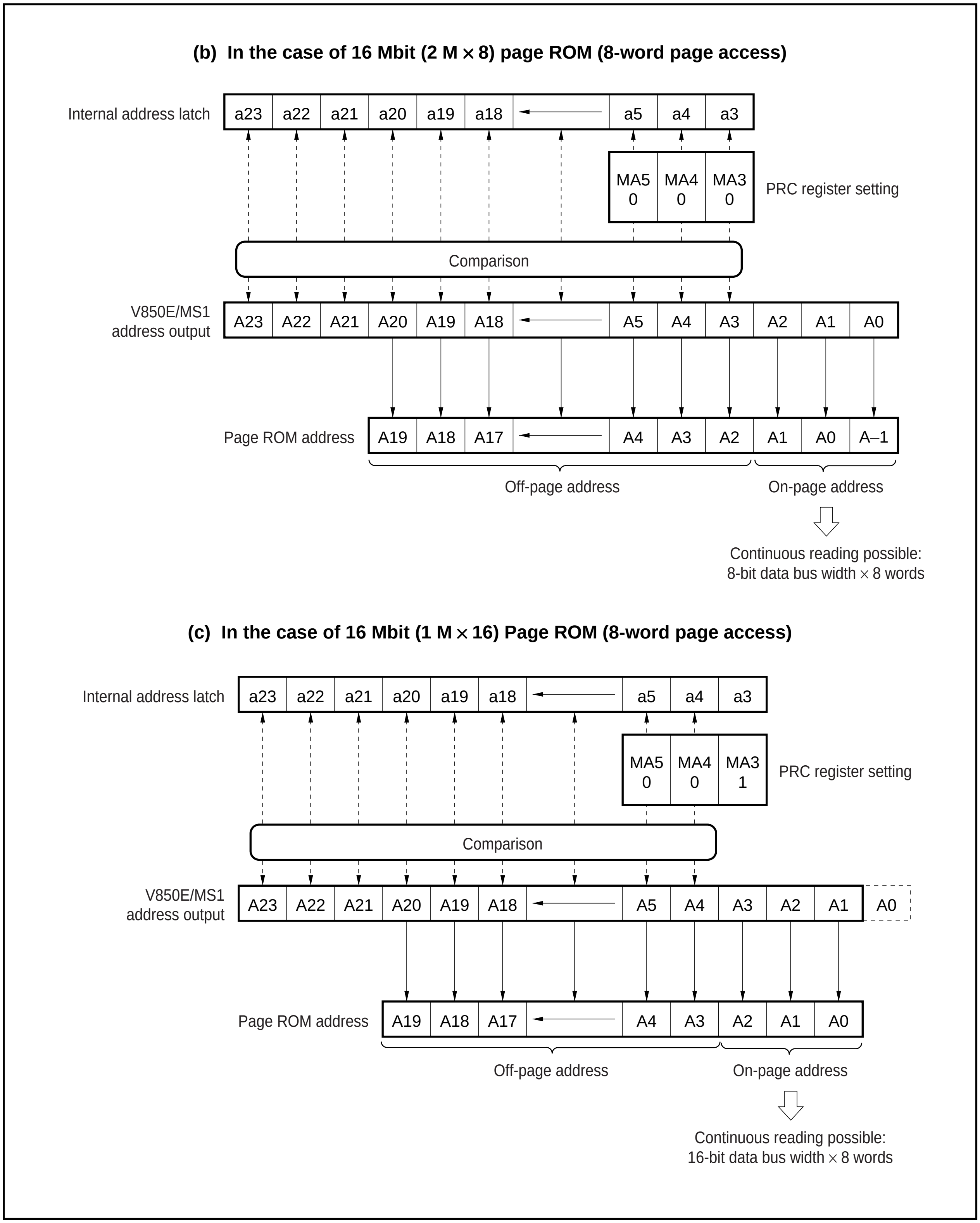


Figure 5-4. On-Page/Off-Page Judgment for Page ROM Connection (2/2)



(b) In the case of 16 Mbit (2 M × 8) page ROM (8-word page access)

(c) In the case of 16 Mbit (1 M × 16) Page ROM (8-word page access)

5.2.4 Page ROM configuration register (PRC)

This specifies whether page ROM on-page access is enabled or disabled. Also, if on-page access is enabled, the masked addresses (no comparison is made) out of the addresses (A3 to A5) corresponding to the configuration of the connected page ROM and the number of bits that can be read continuously, as well as the number of waits corresponding to the internal system clock, are set.

This register can be read/written in 8- or 1-bit units.

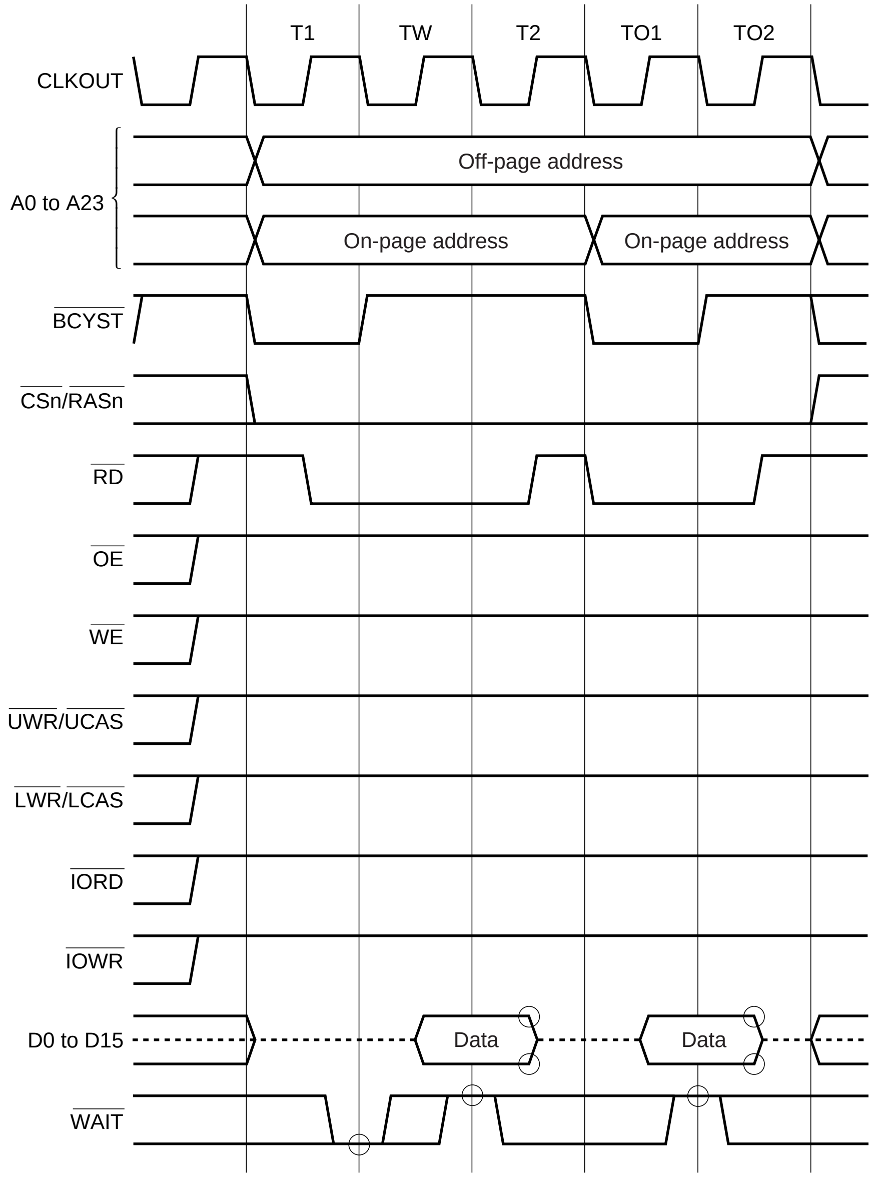
	7	6	5	4	3	2	1	0		
PRC	PAE	PRW2	PRW1	PRW0	0	MA5	MA4	MA3	Address FFFFF224H	After reset 70H

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7</td><td>PAE</td><td>Page ROM On-page Access Enable
Specifies whether page ROM on-page access is enabled or disabled.
0: Disable
1: Enable</td></tr>
<tr><td>6 to 4</td><td>PRW2 to PRW0</td><td>Page-ROM On-page Access Wait Control
Sets the number of waits corresponding to the internal system clock.
The waits set by this bit are inserted only for on-page access. For off-page access, the waits set by registers DWC1 and DWC2 are inserted (refer to <b>4.6 Wait Function</b>).
<table>
<tr><th>PRW2</th><th>PRW1</th><th>PRW0</th><th>Number of Inserted Wait Cycles</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>7</td></tr>
</table></td></tr>
<tr><td>2 to 0</td><td>MA5 to MA3</td><td>Mask Address
Each address (A5 to A3) corresponding to MA5 to MA3 is masked (by 1). The masked address is not subject to comparison during on/off-page judgment. It is set according to the number of continuously readable bits.
<table>
<tr><th>MA5</th><th>MA4</th><th>MA3</th><th>Number of Continuously Readable Bits</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>4 words × 16 bits (8 words × 8 bits)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>8 words × 16 bits (16 words × 8 bits)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>16 words × 16 bits (32 words × 8 bits)</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>32 words × 16 bits (64 words × 8 bits)</td></tr>
</table></td></tr>
</table>

Caution Write to the PRC register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the PRC register is complete. However, it is possible to access an external memory area whose initialization is complete.

5.2.5 Page ROM access

Figure 5-5. Page ROM Access Timing



Remarks **1.** The circle indicates the sampling timing.

2. The broken lines indicate high impedance.

3. n = 0 to 7

5.3 DRAM Controller

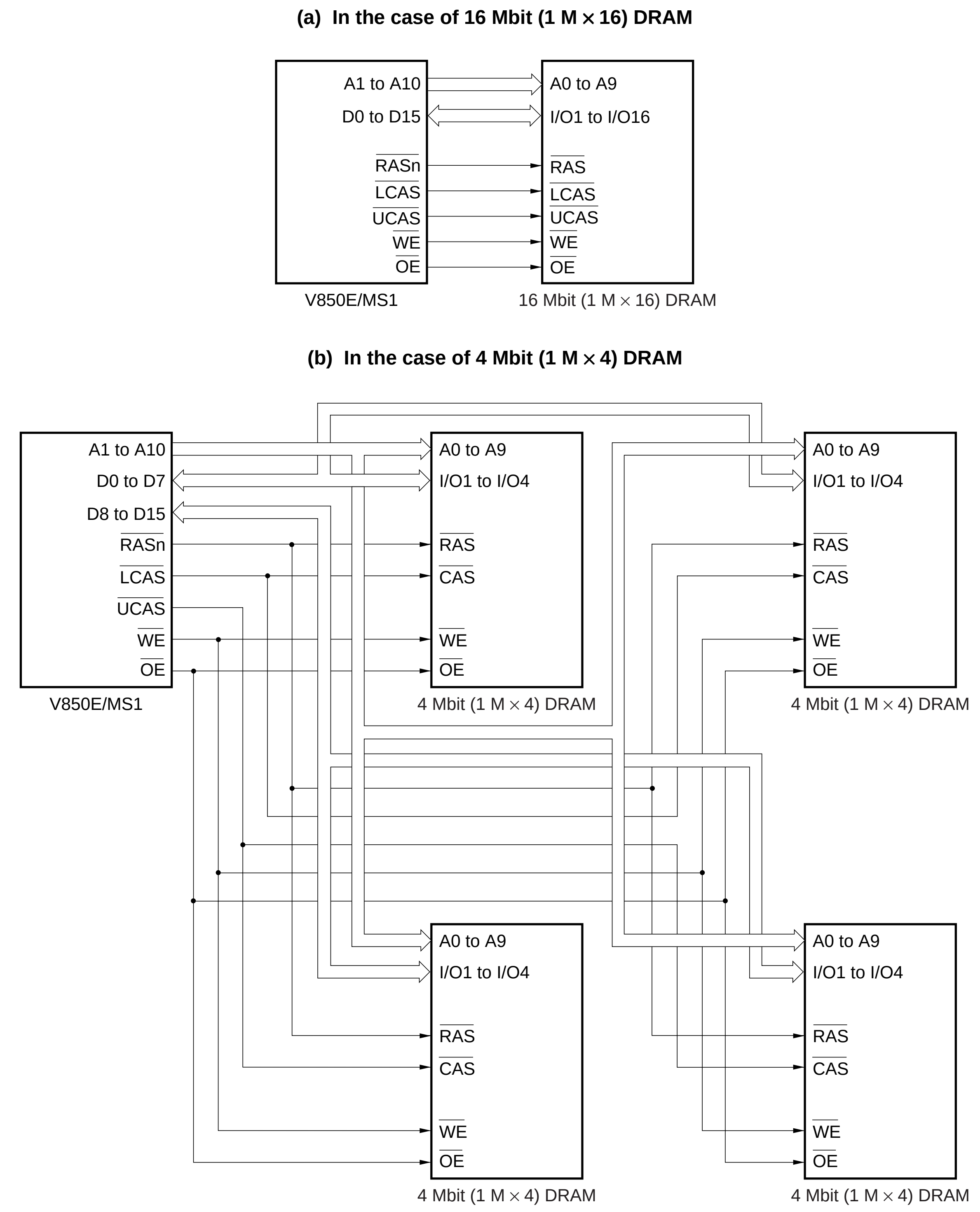
5.3.1 Features

- Generates the $\overline{\text{RAS}}$, $\overline{\text{LCAS}}$ and $\overline{\text{UCAS}}$ signals.
- Can be connected directly to high-speed page DRAM and EDO DRAM.
- Supports the RAS hold mode.
- 4 types of DRAM can be assigned to 8 memory block spaces.
- Can handle 2CAS type DRAM
- Can be switched between row and column address multiplex widths.
- Waits (0 to 3 waits) can be inserted at the following timings.
  - Row address precharge wait
  - Row address hold wait
  - Data access wait
  - Column address precharge wait
- Supports CBR refresh and CBR self-refresh.

5.3.2 DRAM connections

Examples of connections to DRAM are shown below.

Figure 5-6. Examples of Connections to DRAM



(a) In the case of 16 Mbit (1 M × 16) DRAM

(b) In the case of 4 Mbit (1 M × 4) DRAM

Remark n = 0 to 7

5.3.3 Address multiplex function

Depending on the value of the DAW0n and DAW1n bits in DRAM configuration register n (DRCn), the row address, column address output in the DRAM cycle is multiplexed as shown in Figure 5-7 (n = 0 to 3). In Figure 5-7, a0 to a23 show the addresses output from the CPU and A0 to A23 show the V850E/MS1's address pins. For example, when DAW0n and DAW1n = 11, it indicates that a12 to a22 are output from the address pins (A1 to A11) as row addresses and a1 to a11 are output as column addresses.

Table 5-1 shows the relationship between connectable DRAM and the address multiplex width. Depending on the DRAM being connected, DRAM space is from 128 KB to 8 MB.

Figure 5-7. Row Address/Column Address Output

Address pin	A23 to A18	A17	A16	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
Row address (DAW1n, DAW0n = 11)	a23 to a18	a17	a16	a15	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11
Row address (DAW1n, DAW0n = 10)	a23 to a18	a17	a16	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10
Row address (DAW1n, DAW0n = 01)	a23 to a18	a17	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9
Row address (DAW1n, DAW0n = 00)	a23 to a18	a25	a24	a23	a22	a21	a20	a19	a18	a17	a16	a15	a14	a13	a12	a11	a10	a9	a8
Column address	a23 to a18	a17	a16	a15	a14	a13	a12	a11	a10	a9	a8	a7	a6	a5	a4	a3	a2	a1	a0

Table 5-1. Example of DRAM and Address Multiplex Width

Address Multiplex Width	DRAM Capacity (Bits) and Configuration					DRAM Space (Bytes)
	256 K	1 M	4 M	16 M	64 M	
8 bits	64 K × 4	—	—	—	—	128 K
9 bits	—	256 K × 4	256 K × 16	—	—	512 K
	—	—	512 K × 8	—	—	1 M
	—	—	—	—	4 M × 16	8 M
10 bits	—	—	1 M × 4	1 M × 16	—	2 M
	—	—	—	2 M × 8	—	4 M
	—	—	—	—	4 M × 16	8 M
11 bits	—	—	—	4 M × 4	—	8 M

5.3.4 DRAM configuration registers 0 to 3 (DRC0 to DRC3)

This sets the type of DRAM to be connected.

These registers can be read/written in 16-bit units.

Caution If the object of access is a DRAM area, the wait set in registers DWC1 and DWC2 becomes invalid. In this case, waits are controlled by registers DRC0 to DRC3.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DRC0	PAE10	PAE00	RPC10	RPC00	RHC10	RHC00	DAC10	DAC00	CPC10	CPC00	0	RHD0	0	0	DAW10	DAW00	FFFFF200H	3FC1H
DRC1	PAE11	PAE01	RPC11	RPC01	RHC11	RHC01	DAC11	DAC01	CPC11	CPC01	0	RHD1	0	0	DAW11	DAW01	FFFFF202H	3FC1H
DRC2	PAE12	PAE02	RPC12	RPC02	RHC12	RHC02	DAC12	DAC02	CPC12	CPC02	0	RHD2	0	0	DAW12	DAW02	FFFFF204H	3FC1H
DRC3	PAE13	PAE03	RPC13	RPC03	RHC13	RHC03	DAC13	DAC03	CPC13	CPC03	0	RHD3	0	0	DAW13	DAW03	FFFFF206H	3FC1H

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15, 14</td><td>PAE1n,
PAE0n</td><td>DRAM On-page Access Mode Control
Controls the on-page access cycle.
<table>
<tr><th>PAE1n</th><th>PAE0n</th><th>Access Mode</th></tr>
<tr><td>0</td><td>0</td><td>On-page access disabled.</td></tr>
<tr><td>0</td><td>1</td><td>High-speed page DRAM</td></tr>
<tr><td>1</td><td>0</td><td>EDO DRAM</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>13, 12</td><td>RPC1n,
RPC0n</td><td>Row Address Precharge Control
Specifies the number of wait states inserted as row address precharge time.
<table>
<tr><th>RPC1n</th><th>RPC0n</th><th>Number of Wait States Inserted</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>1</td><td>3</td></tr>
</table></td></tr>
</table>

Remark n = 0 to 3

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11, 10</td><td>RHC1n,
RHC0n</td><td>Row Address Hold Wait Control
Specifies the number of wait states inserted as row address hold time.
<table>
<tr><th>RHC1n</th><th>RHC0n</th><th>Number of Wait States Inserted</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>1</td><td>3</td></tr>
</table></td></tr>
<tr><td>9, 8</td><td>DAC1n,
DAC0n</td><td>Data Access Programmable Wait Control
Specifies the number of wait states inserted as data access time in DRAM access.
<table>
<tr><th>DAC1n</th><th>DAC0n</th><th>Number of Wait States Inserted</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>1</td><td>3</td></tr>
</table></td></tr>
<tr><td>7, 6</td><td>CPC1n,
CPC0n</td><td>Column Address Pre-charge Control
Specifies the number of wait states inserted as column address precharge time.
<table>
<tr><th>CPC1n</th><th>CPC0n</th><th>Number of Wait States Inserted</th></tr>
<tr><td>0</td><td>0</td><td>0[Note]</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>1</td><td>3</td></tr>
</table>
Note 1 wait is inserted during DRAM write access in DMA flyby transfer.</td></tr>
<tr><td>4</td><td>RHDn</td><td>RAS Hold Disable
Sets the RAS hold mode.
If access to DRAM during on-page operation is not continuous, and access enters another space midway, the $\overline{\text{RASm}}$ signal (m = 0 to 7) is maintained in the active state (low level) during the time the other space is being accessed in the RAS hold mode state. In this way, if access continues in the same DRAM row address following access of the other space, on-page operation can be continued.
0: RAS hold mode enabled
1: RAS hold mode disabled</td></tr>
</table>

Remark n = 0 to 3

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>1, 0</td><td>DAW1n,
DAW0n</td><td>DRAM Address Multiplex Width Control
This sets the address multiplex width (refer to <b>5.3.3 Address multiplex function</b>).
<table>
<tr><th>DAW1n</th><th>DAW0n</th><th>Address Multiplex Width</th></tr>
<tr><td>0</td><td>0</td><td>8 bits</td></tr>
<tr><td>0</td><td>1</td><td>9 bits</td></tr>
<tr><td>1</td><td>0</td><td>10 bits</td></tr>
<tr><td>1</td><td>1</td><td>11 bits</td></tr>
</table>
</td></tr>
</table>

Caution Write to the DRCn register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the DRCn register is complete. However, it is possible to access an external memory area whose initialization is complete.

Remark n = 0 to 3

5.3.5 DRAM type configuration register (DTC)

This controls the relationship between DRAM configuration register n (DRCn) and memory block m (n = 0 to 3, m = 0 to 7).

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DTC	DC 71	DC 70	DC 61	DC 60	DC 51	DC 50	DC 41	DC 40	DC 31	DC 30	DC 21	DC 20	DC 11	DC 10	DC 01	DC 00	FFFFF220H	0000H
Memory block	7		6		5		4		3		2		1		0			

Bit Position	Bit Name	Function
15 to 0	DCm1, DCm0	DRAM Type Configuration Specifies the DRAM configuration register n (DRCn) corresponding to memory block m. Furthermore, it has no meaning if the memory block m is not specified in the DRAM area.

DCm1	DCm0	DRAM Configuration Register n (DRCn) Corresponding to Memory Block m
0	0	DRC0
0	1	DRC1
1	0	DRC2
1	1	DRC3

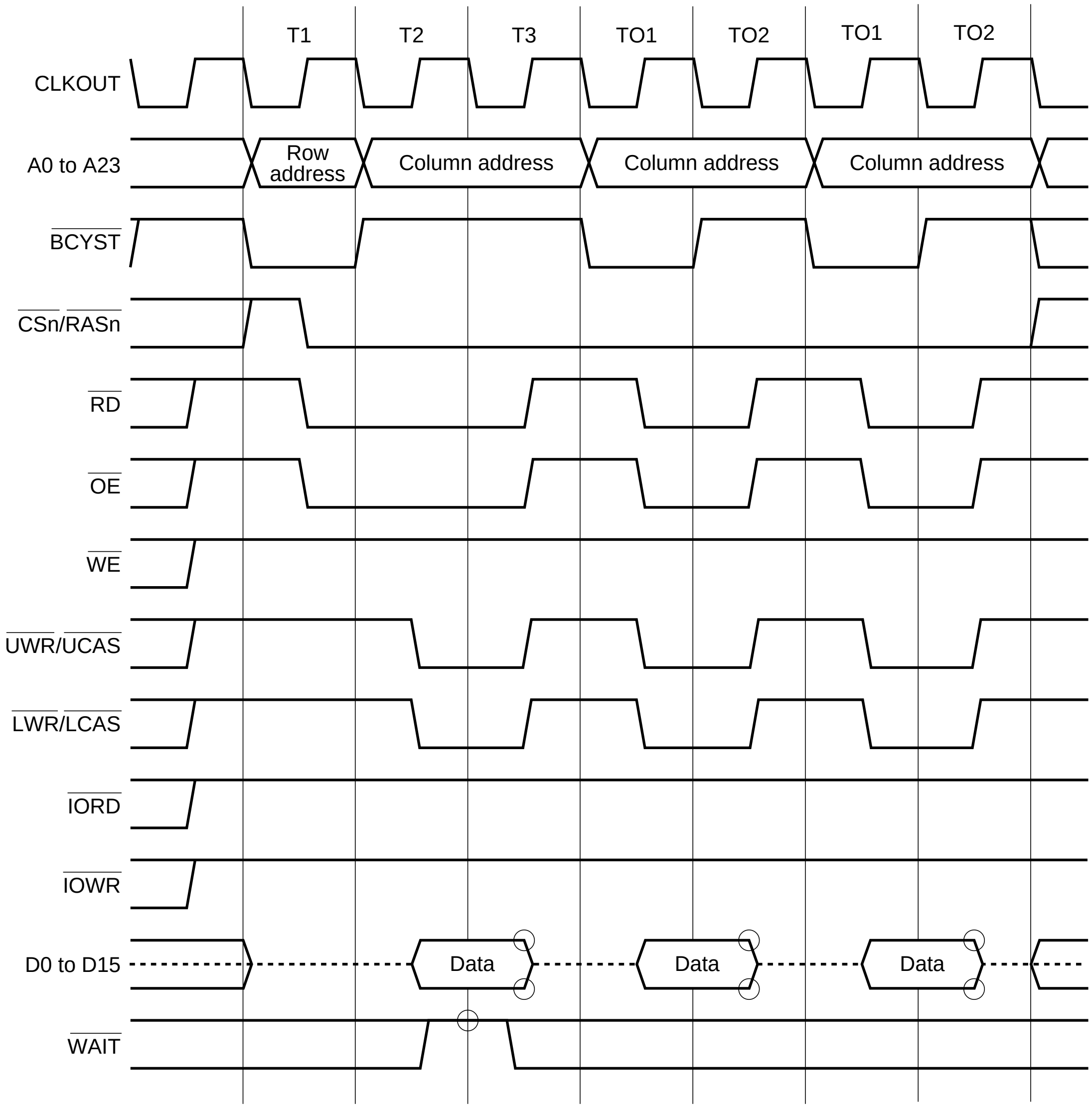
Caution Write to the DTC register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the DTC register is complete. However, it is possible to access an external memory area whose initialization is complete.

Remark n = 0 to 3
m = 0 to 7

5.3.6 DRAM access

Figure 5-8. High-Speed Page DRAM Access Timing (1/4)

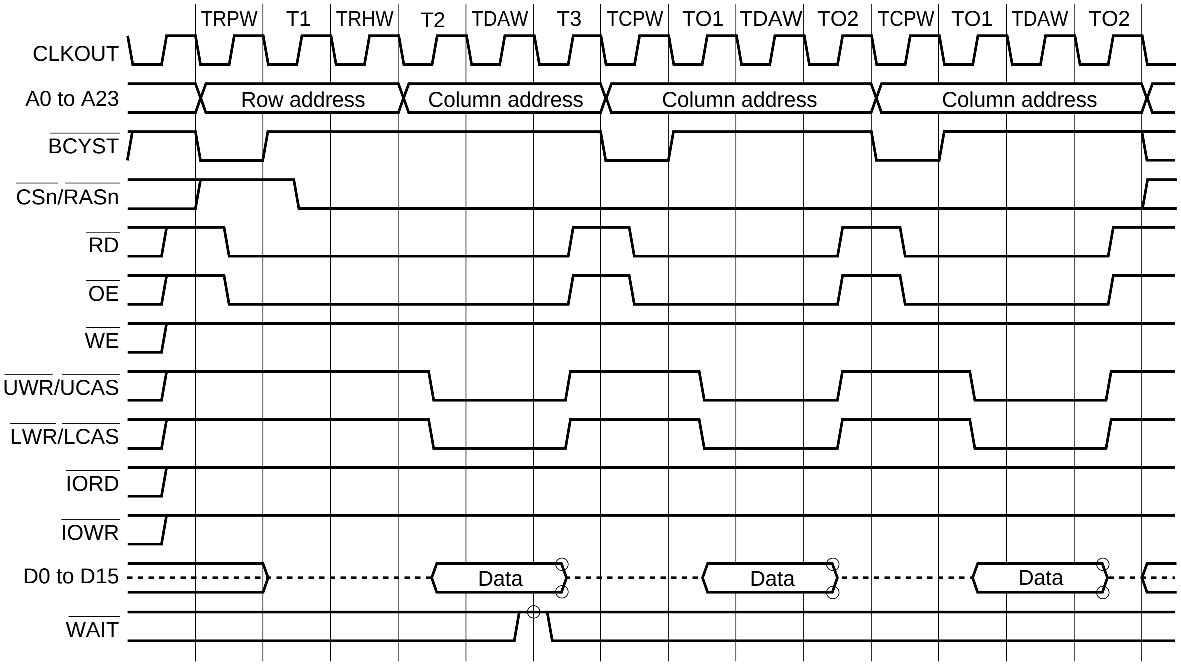
(a) Read timing 1



Remarks **1.** This is the timing in the case of no waits.
2. The circle indicates the sampling timing.
3. The broken lines indicate high impedance.
4. n = 0 to 7

Figure 5-8. High-Speed Page DRAM Access Timing (2/4)

(b) Read timing 2



Remarks **1.** This is the timing in the following cases (×× = 00 to 03, 10 to 13).

Number of waits according to bit RPC×× (TRPW): 1

Number of waits according to bit RHC×× (TRHW): 1

Number of waits according to bit DAC×× (TDAW): 1

Number of waits according to bit CPC×× (TCPW): 1

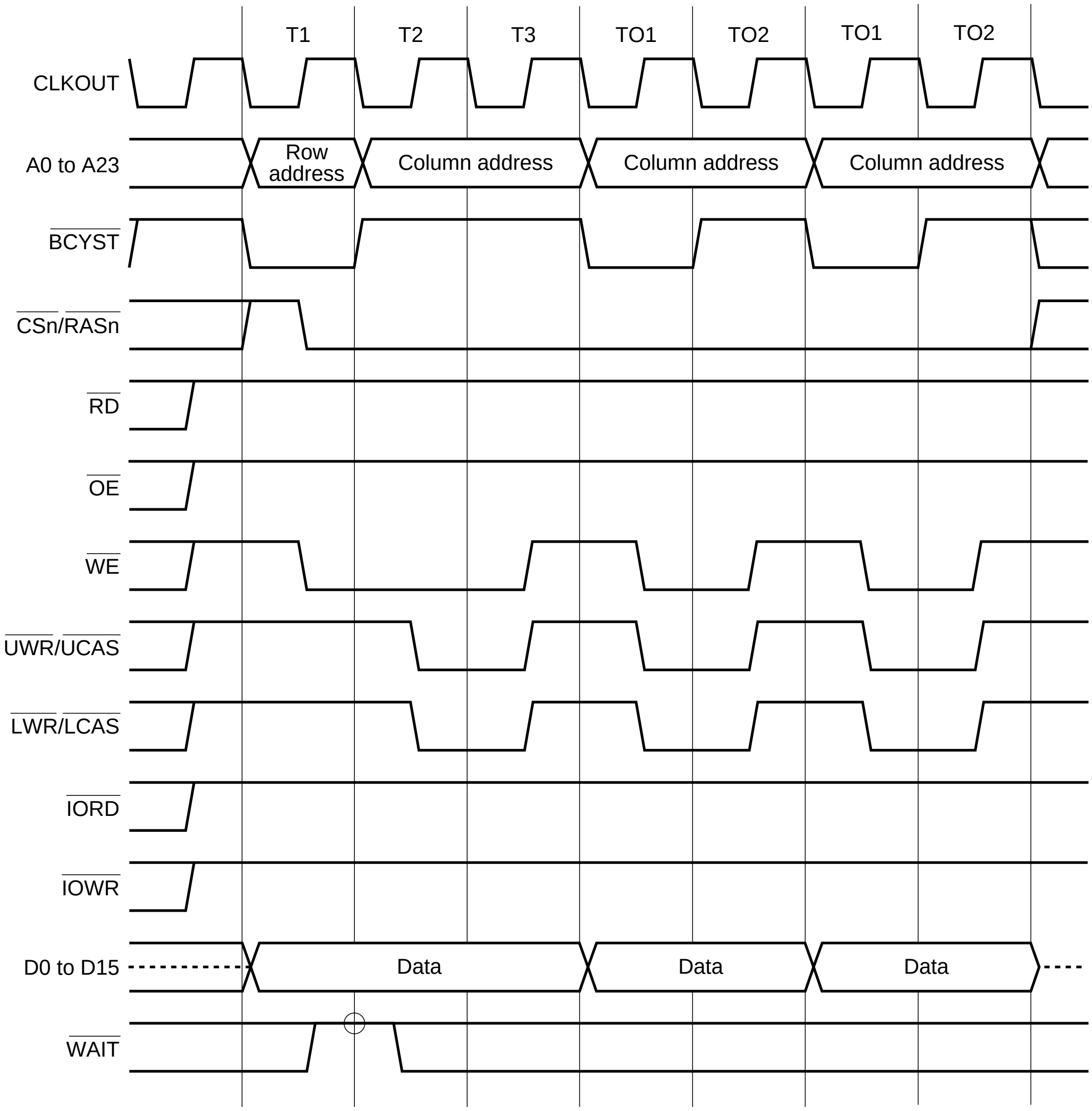
2. The circle indicates the sampling timing.

3. The broken lines indicate high impedance.

4. n = 0 to 7

Figure 5-8. High-Speed Page DRAM Access Timing (3/4)

(c) Write timing 1



Remarks **1.** This is the timing in the case of no waits.

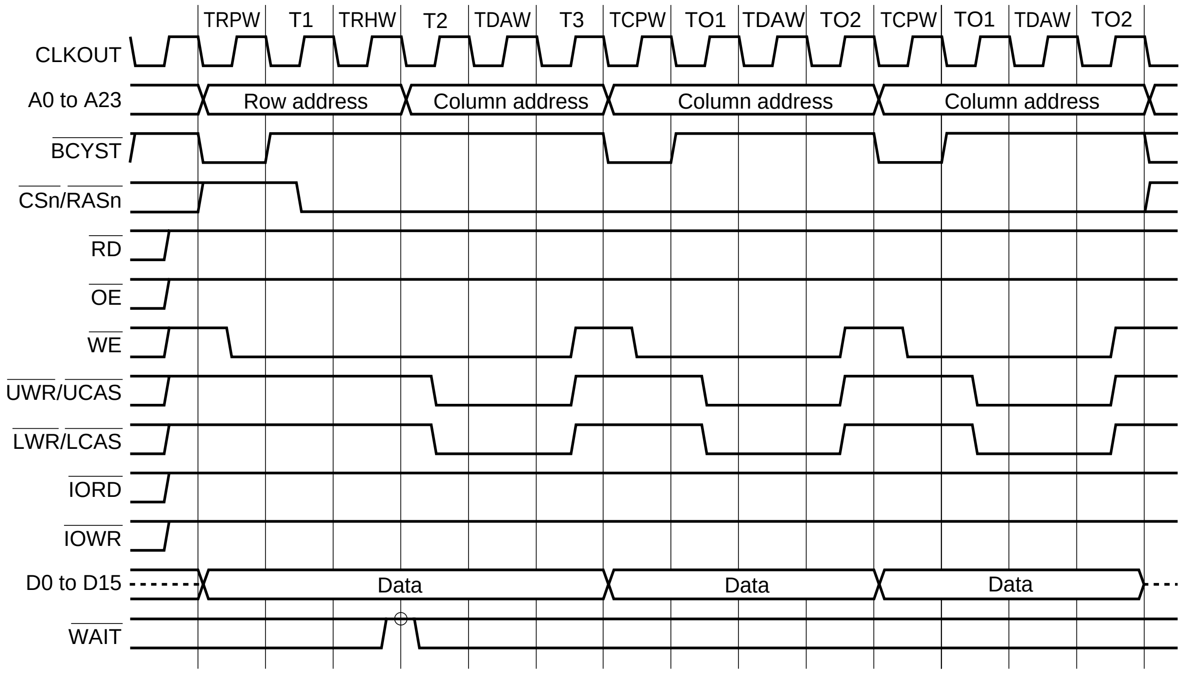
2. The circle indicates the sampling timing.

3. The broken lines indicate high impedance.

4. n = 0 to 7

Figure 5-8. High-Speed Page DRAM Access Timing (4/4)

(d) Write timing 2



Remarks 1. This is the timing in the following cases (×× = 00 to 03, 10 to 13).
Number of waits according to bit RPC×× (TRPW): 1
Number of waits according to bit RHC×× (TRHW): 1
Number of waits according to bit DAC×× (TDAW): 1
Number of waits according to bit CPC×× (TCPW): 1

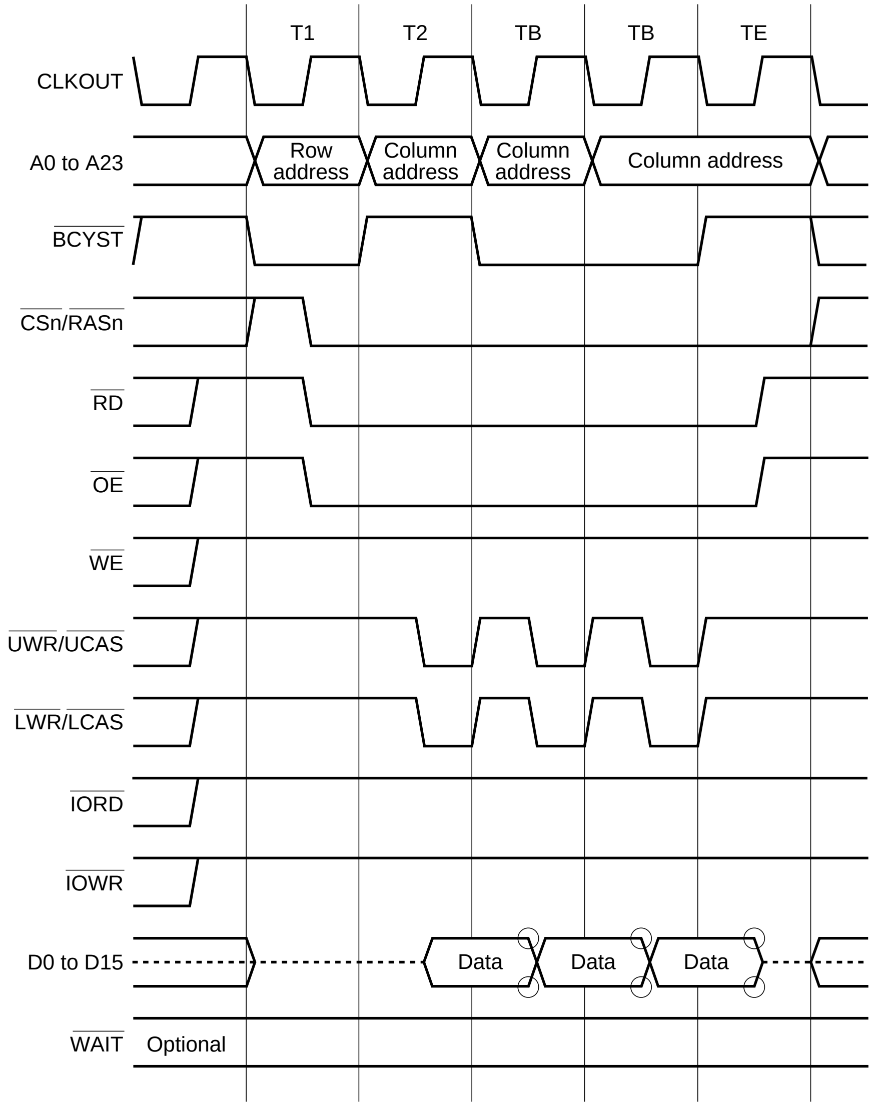
2. The circle indicates the sampling timing.

3. The broken lines indicate high impedance.

4. n = 0 to 7

Figure 5-9. EDO DRAM Access Timing (1/4)

(a) Read timing 1



Remarks 1. This is the timing in the case of no waits.

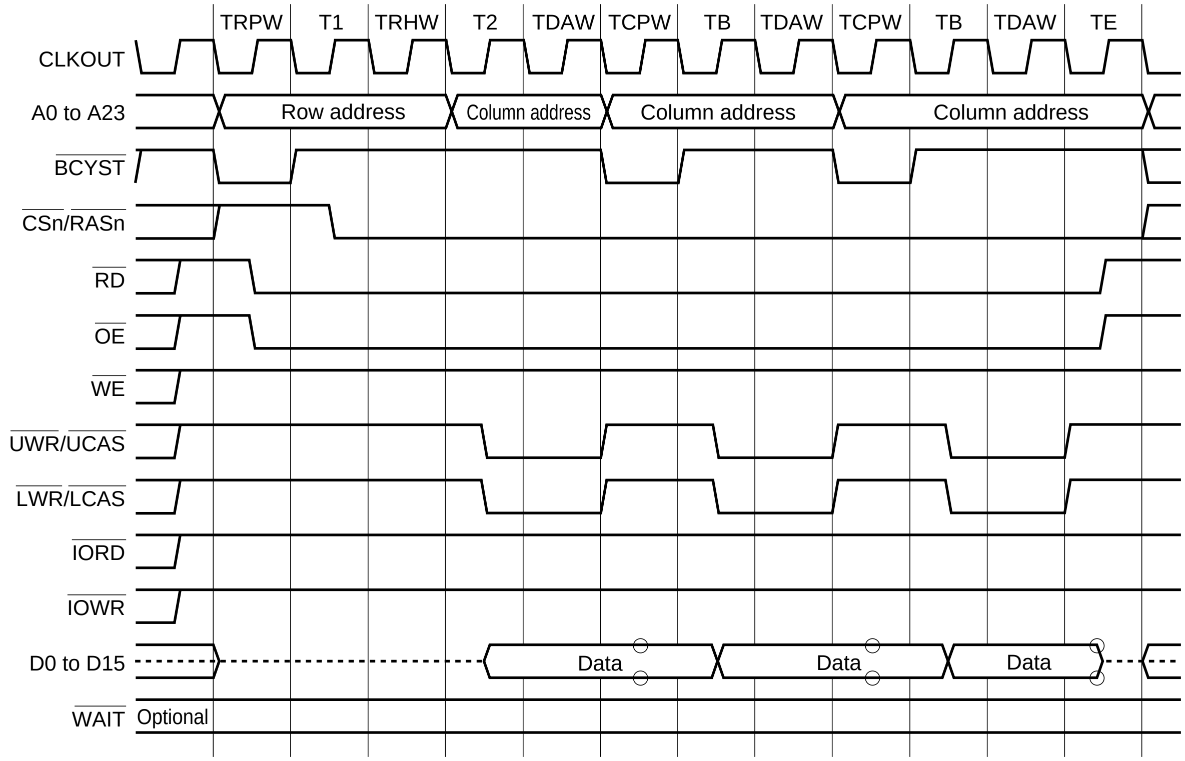
2. The circle indicates the sampling timing.

3. The broken lines indicate high impedance.

4. n = 0 to 7

Figure 5-9. EDO DRAM Access Timing (2/4)

(b) Read timing 2



Remarks **1.** This is the timing in the following cases (×× = 00 to 03, 10 to 13).

Number of waits according to bit RPC×× (TRPW): 1

Number of waits according to bit RHC×× (TRHW): 1

Number of waits according to bit DAC×× (TDAW): 1

Number of waits according to bit CPC×× (TCPW): 1

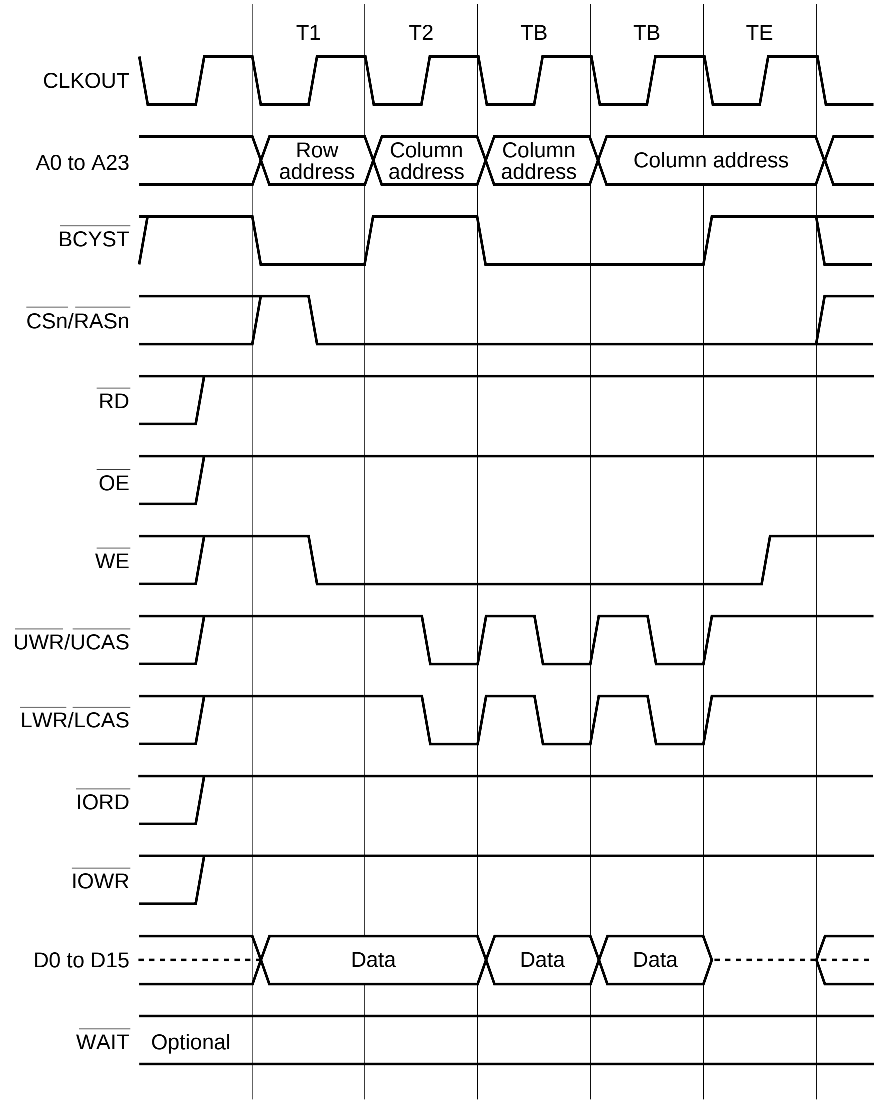
2. The circle indicates the sampling timing.

3. The broken lines indicate high impedance.

4. n = 0 to 7

Figure 5-9. EDO DRAM Access Timing (3/4)

(c) Write timing 1



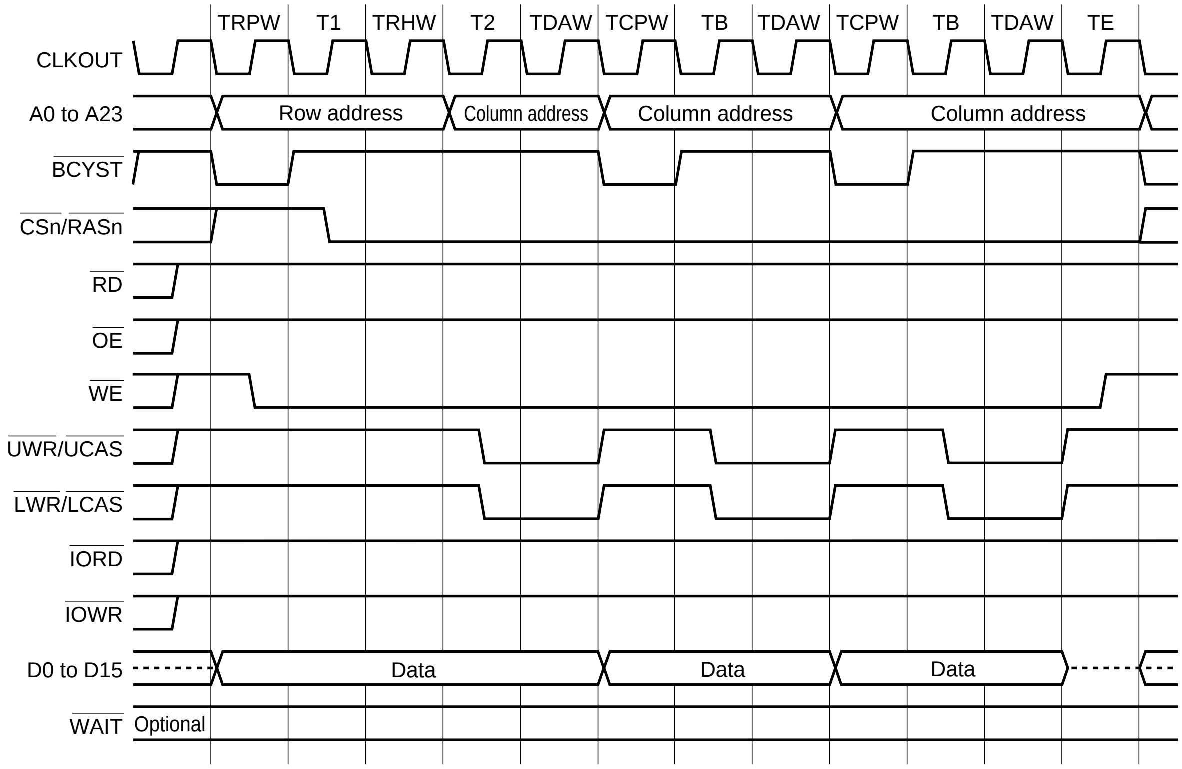
Remarks **1.** This is the timing in the case of no waits.

2. The broken lines indicate high impedance.

3. n = 0 to 7

Figure 5-9. EDO DRAM Access Timing (4/4)

(d) Write timing 2



Remarks 1. This is the timing in the following cases (×× = 00 to 03, 10 to 13).

Number of waits according to bit RPC×× (TRPW): 1

Number of waits according to bit RHC×× (TRHW): 1

Number of waits according to bit DAC×× (TDAW): 1

Number of waits according to bit CPC×× (TCPW): 1

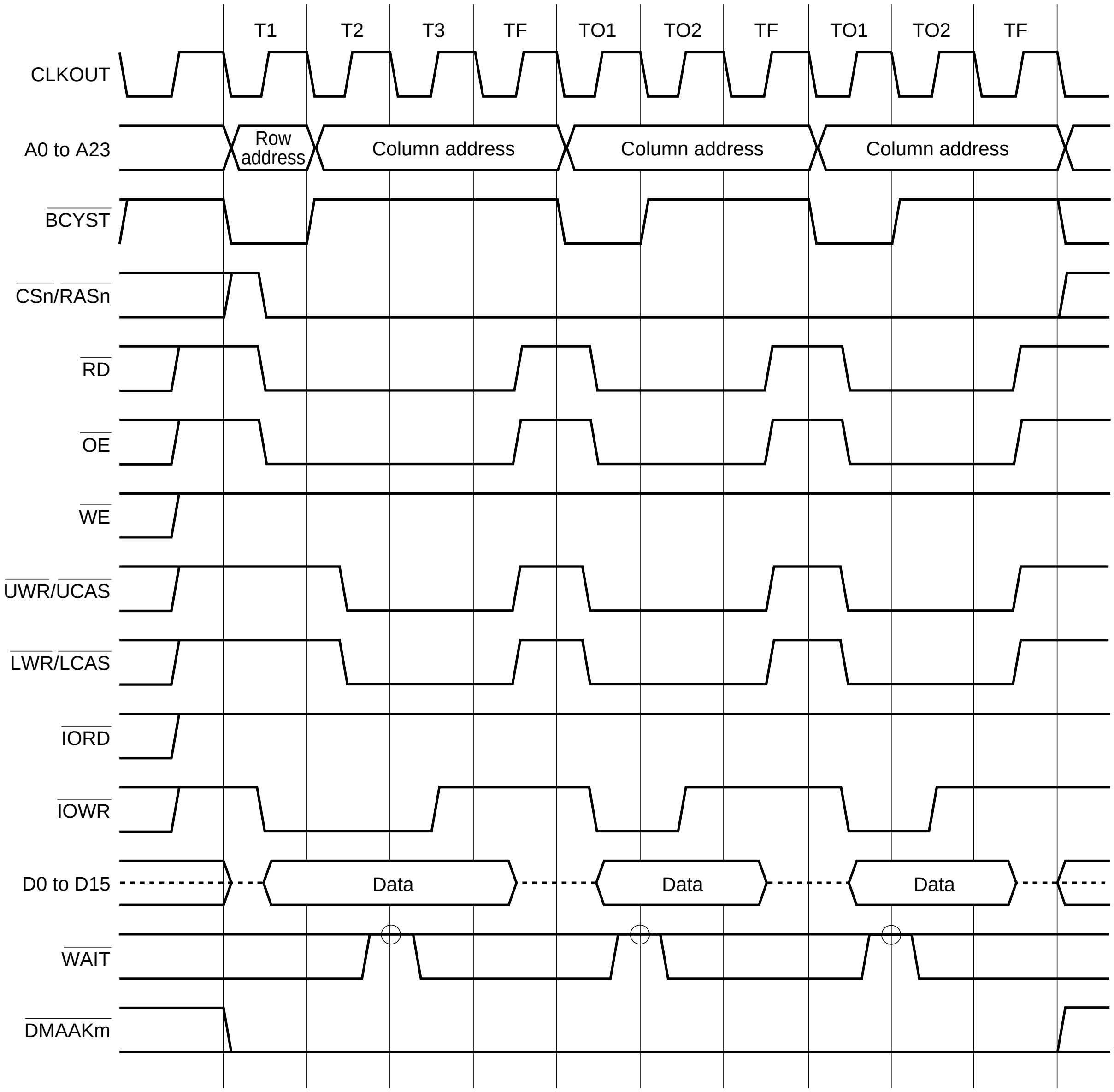
2. The broken lines indicate high impedance.

3. n = 0 to 7

5.3.7 DRAM access during DMA flyby transfer

Figure 5-10. DRAM Access Timing During DMA Flyby Transfer (1/2)

(a) In the case of DRAM → External I/O



Remarks **1.** This is the timing in the case where wait (TF) insertion setting was carried out according to the FDW register.

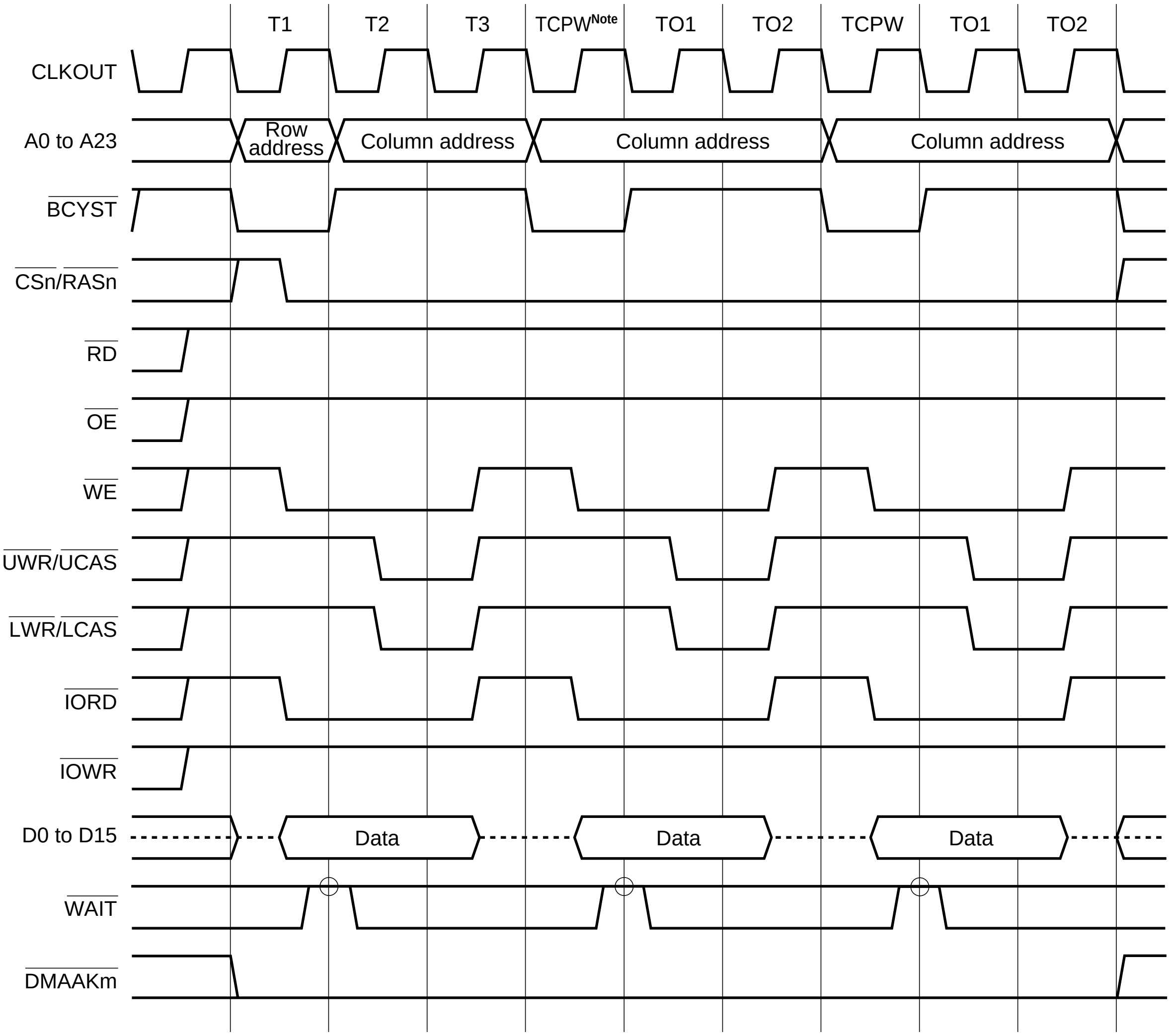
2. The circle indicates the sampling timing.

3. The broken lines indicate high impedance.

4. n = 0 to 7
m = 0 to 3

Figure 5-10. DRAM Access Timing During DMA Flyby Transfer (2/2)

(b) In the case of external I/O → DRAM



Note A minimum of 1 clock cycle is inserted for the TCPW cycle regardless of the CPC0m and CPC1m bit settings in the DRCm register.

Remarks 1. This is the timing in the case where the number of waits according to the CPC×× bit (TCPW) is 1 (×× = 00 to 03, 10 to 13).

2. In the case of external I/O → DRAM, the FDW register setting is invalid.

3. The circle indicates the sampling timing.

4. The broken lines indicate high impedance.

5. n = 0 to 7
m = 0 to 3

5.3.8 Refresh control function

V850E/MS1 can create a CBR (CAS-before-RAS) refresh cycle. The refresh cycle is set with the refresh control register (RFC).

When another bus master occupies the external bus, the DRAM controller cannot occupy the external bus. In this case, the DRAM controller sends a refresh request to the bus master by changing the $\overline{\text{REFRQ}}$ signal to active (low level).

During the refresh interval, the address bus maintains the state it was in just before the refresh cycle.

(1) Refresh control registers 0 to 3 (RFC0 to RFC3)

These set whether refresh is enabled or disabled, and the refresh interval. The refresh interval is determined by the following calculation formula.

Refresh interval (μs) = Refresh count clock (T_{RCY}) × Interval factor

The refresh count clock and interval factor are determined by the RENn bit and RIn bit, respectively, of the RFCn register.

Note that n corresponds to the register number (0 to 3) of DRAM configuration registers 0 to 3 (DRC0 to DRC3).

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
RFC0	REN0	0	0	0	0	0	RCC01	RCC00	0	0	RI05	RI04	RI03	RI02	RI01	RI00	FFFFF210H	0000H
RFC1	REN1	0	0	0	0	0	RCC11	RCC10	0	0	RI15	RI14	RI13	RI12	RI11	RI10	FFFFF212H	0000H
RFC2	REN2	0	0	0	0	0	RCC21	RCC20	0	0	RI25	RI24	RI23	RI22	RI21	RI20	FFFFF214H	0000H
RFC3	REN3	0	0	0	0	0	RCC31	RCC30	0	0	RI35	RI34	RI33	RI32	RI31	RI30	FFFFF216H	0000H

Bit Position	Bit Name	Function
15	RENn	Refresh Enable Specifies whether CBR refresh is enabled or disabled. 0: Refresh disabled 1: Refresh enabled

Remark n = 0 to 3

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>9, 8</td><td>RCCn1,
RCCn0</td><td>Refresh Count Clock
Specifies the refresh count clock (T_{RCY}).
<table>
<tr><th>RCCn1</th><th>RCCn0</th><th>Refresh Count Clock (T_{RCY})</th></tr>
<tr><td>0</td><td>0</td><td>32/ϕ</td></tr>
<tr><td>0</td><td>1</td><td>128/ϕ</td></tr>
<tr><td>1</td><td>0</td><td>256/ϕ</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table>
</td></tr>
<tr><td>5 to 0</td><td>RIn5 to
RIn0</td><td>Refresh Interval
Sets the interval factor of the interval timer for generation of refresh timing.
<table>
<tr><th>RIn5</th><th>RIn4</th><th>RIn3</th><th>RIn2</th><th>RIn1</th><th>RIn0</th><th>Interval Factor</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>2</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>3</td></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>4</td></tr>
<tr><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td><td>⋮</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td><td>64</td></tr>
</table>
</td></tr>
</table>

Caution After refresh enable, if changing the refresh count clock or the interval factor, first clear the RENn bit (0) (refresh disable state), then perform reset.

Remark n = 0 to 3

ϕ = Internal system clock frequency

Example An example of the DRAM refresh interval and an example of setting the interval factor are shown below.

Table 5-2. Example of DRAM Refresh Interval

DRAM Capacity (bits)	Refresh Cycle (Cycles/ms)	Refresh Interval (µs)
256 K	256/4	15.6
1 M	512/8	15.6
	512/64	125
4 M	512/128	250
	1 K/16	15.6
	1 K/128	125
16 M	1 K/256	250
	2 K/256	125
	4 K/64	15.6
	4 K/256	62.5
64 M	4 K/64	15.6

Table 5-3. Example of Interval Factor Settings

Specified Refresh Interval Value (µs)	Refresh Count Clock (T_{RCY})	Interval Factor Value[Notes 1, 2]			
		When ϕ = 16 MHz	When ϕ = 20 MHz	When ϕ = 33 MHz	When ϕ = 40 MHz
15.6	32/ϕ	7 (14)	9 (14.4)	15 (14.5)	19 (15.2)
	128/ϕ	1 (8)	2 (12.8)	3 (11.6)	4 (12.8)
	256/ϕ	—	1 (12.8)	1 (7.8)	2 (12.8)
62.5	32/ϕ	30 (60)	38 (60.8)	63 (61.1)	—
	128/ϕ	7 (56)	9 (57.6)	15 (58.2)	19 (60.8)
	256/ϕ	3 (48)	4 (51.2)	7 (54.3)	9 (57.6)
125	32/ϕ	—	—	—	—
	128/ϕ	15 (120)	19 (121.6)	32 (124.1)	39 (124.8)
	256/ϕ	7 (112)	9 (115.2)	16 (124.1)	19 (121.6)
250	32/ϕ	—	—	—	—
	128/ϕ	31 (248)	38 (243.2)	64 (248.2)	—
	256/ϕ	15 (240)	19 (243.2)	32 (248.2)	39 (249.6)

Notes 1. The interval factor is set by bits RIn0 to RIn5 of the RFCn register (n = 0 to 3).

2. The values in parentheses are the calculated value (µs) for the refresh interval.
Refresh Interval (µs) = Refresh count clock (T_{RCY}) × Interval factor

Remark ϕ: Internal system clock frequency

(2) Refresh wait control register (RWC)

This specifies insertion of wait states during the refresh cycle. The register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
RWC	RRW1	RRW0	RCW2	RCW1	RCW0	SRW2	SRW1	SRW0	FFFFF218H	00H

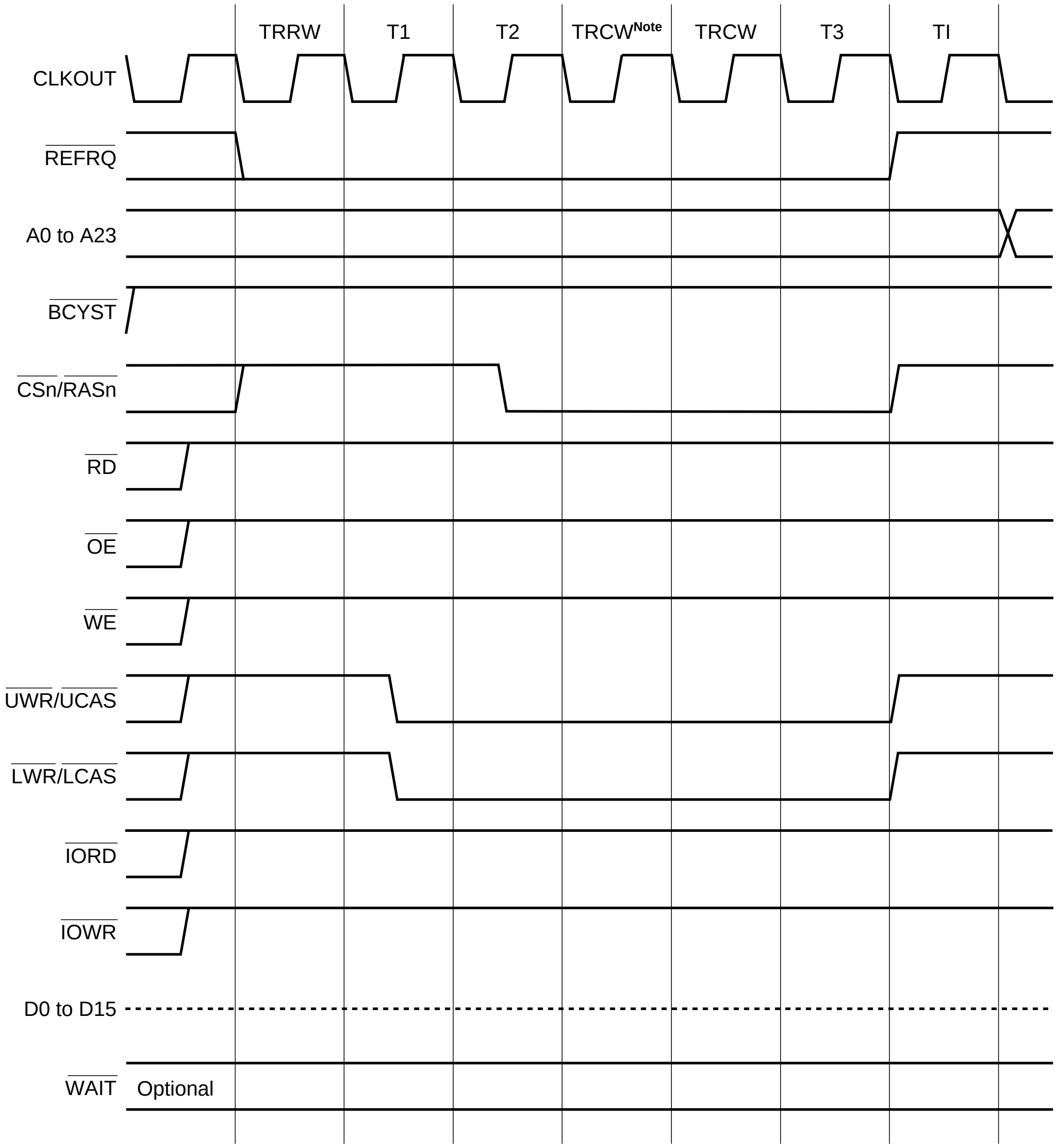
<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>RRW1,
RRW0</td><td>Refresh RAS Wait Control
Specifies the number of wait states inserted as hold time for the $\overline{\text{RASm}}$ signal’s high level width during CBR refresh.
<table>
<tr><th>RRW1</th><th>RRW0</th><th>Number of Insertion Wait States</th></tr>
<tr><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>1</td><td>1</td></tr>
<tr><td>1</td><td>0</td><td>2</td></tr>
<tr><td>1</td><td>1</td><td>3</td></tr>
</table></td></tr>
<tr><td>5 to 3</td><td>RCW2 to
RCW0</td><td>Refresh Cycle Wait Control
Specifies the number of wait states inserted as hold time for the $\overline{\text{RASm}}$ signal’s low level width during CBR refresh.
<table>
<tr><th>RCW2</th><th>RCW1</th><th>RCW0</th><th>Number of Insertion Wait States</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>7</td></tr>
</table></td></tr>
<tr><td>2 to 0</td><td>SRW2 to
SRW0</td><td>Self-refresh Release Wait Control
Specifies the number of wait states inserted as CBR self-refresh release time.
<table>
<tr><th>SRW2</th><th>SRW1</th><th>SRW0</th><th>Number of Insertion Wait States</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>1</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>2</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>3</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>4</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>5</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>6</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>7</td></tr>
</table></td></tr>
</table>

Caution Write to the RWC register after reset, and then do not change the set value. Also, do not access an external memory area other than the one for this initialization routine until the initial setting of the RWC register is complete. However, it is possible to access an external memory area whose initialization is complete.

Remark m = 0 to 7

(3) Refresh timing

Figure 5-11. CBR Refresh Timing



Note A minimum of 1 clock cycle is inserted for the TRCW cycle regardless of the RCW0 to RCW2 bit settings in the RWC register.

Remarks 1. This is the timing in the case where the number of waits (TRCW) according to the bits RCW0 to RCW2 is 1.

2. n = 0 to 7

5.3.9 Self-refresh functions

In the case of IDLE mode and software STOP mode, the DRAM controller generates a CBR self-refresh cycle.

However, the $\overline{\text{RASn}}$ pulse width of DRAM should meet the specifications to enter a self-refresh operation mode (n = 0 to 7).

To release the self-refresh cycle, follow either of two methods below.

(1) Release by NMI input

(a) In the case of self-refresh cycle with IDLE mode

Set the $\overline{\text{RASn}}$, $\overline{\text{LCAS}}$, $\overline{\text{UCAS}}$ signals to inactive (high level) immediately to release the self-refresh cycle.

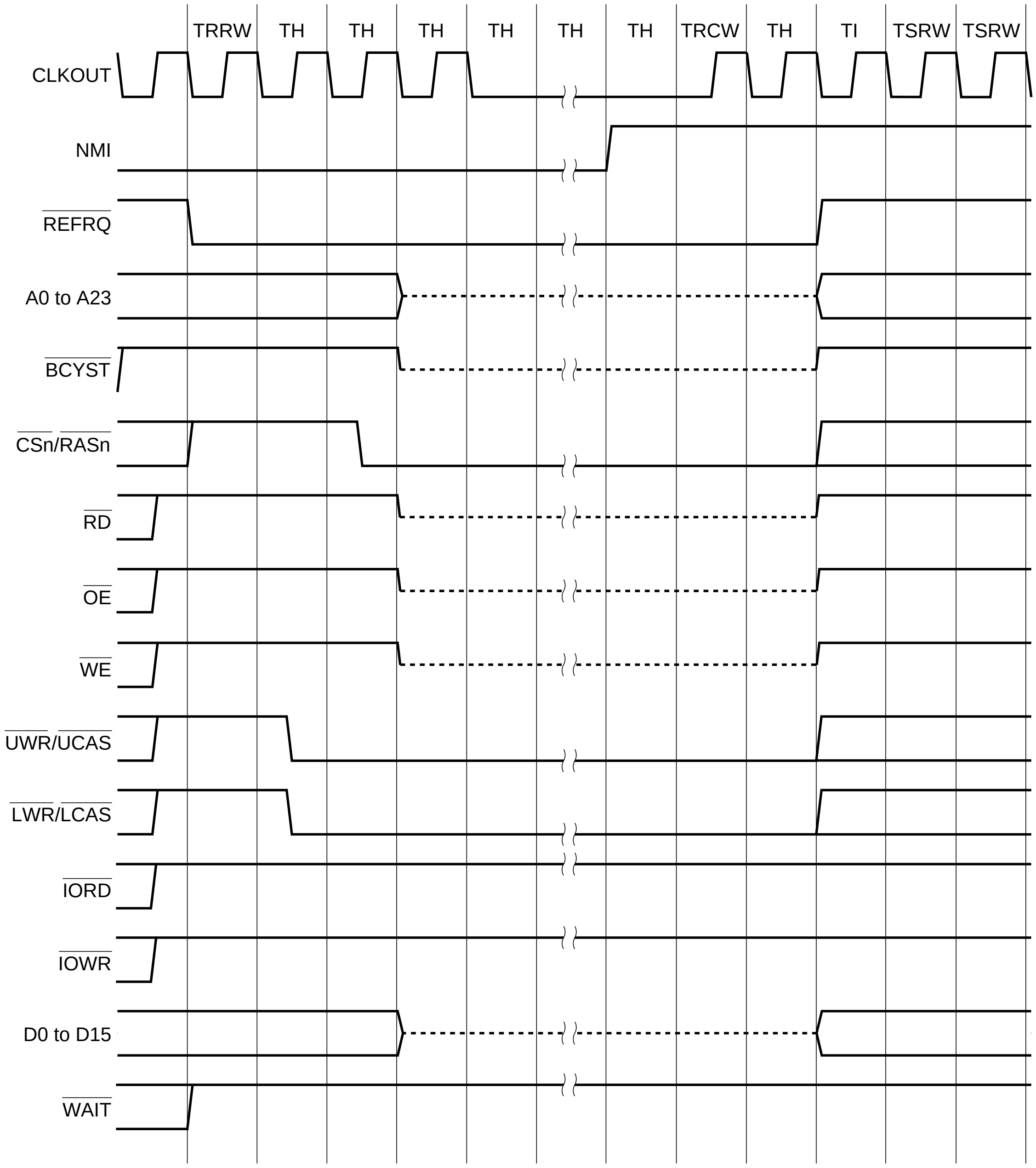
(b) In the case of self-refresh cycle with software STOP mode

Set the $\overline{\text{RASn}}$, $\overline{\text{LCAS}}$, $\overline{\text{UCAS}}$ signals to inactive (high level) after stabilizing oscillation to release the self-refresh cycle.

(2) Release by $\overline{\text{RESET}}$ input

Figure 5-12. CBR Self-Refresh Timing (1/2)

(a) In the case of release according to the NMI input (in the IDLE Mode)



Remarks 1. This is the timing in the following cases.

Number of waits according to bits RRW0 and RRW1 (TRRW): 1

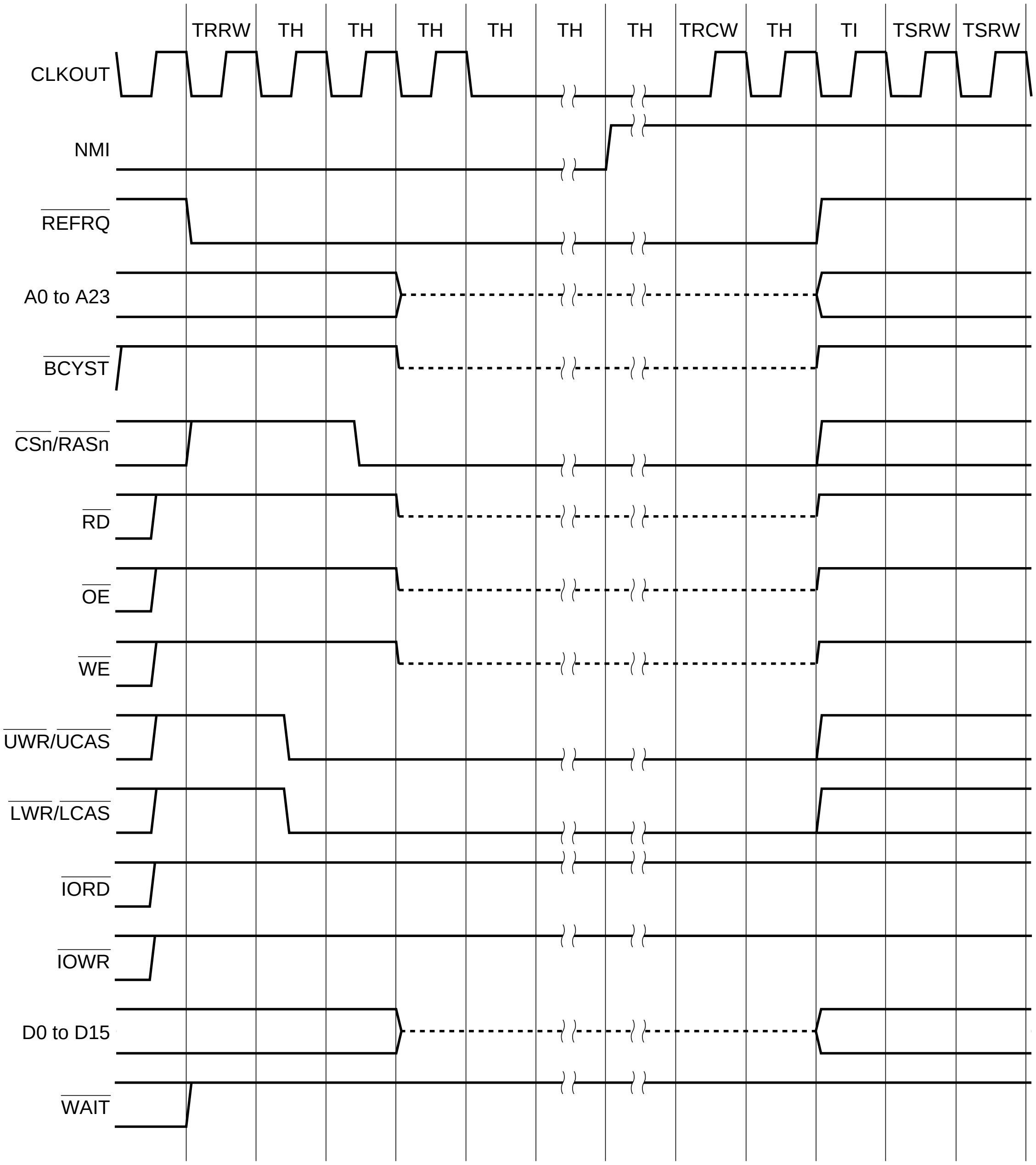
Number of waits according to bits RCW0 to RCW2 (TRCW): 1

Number of waits according to bits SRW0 to SRW2 (TSRW): 2

2. n = 0 to 7

Figure 5-12. CBR Self-Refresh Timing (2/2)

(b) In the case of release according to the NMI input (in the software STOP Mode)



Remarks **1.** This is the timing in the following cases.

Number of waits according to bits RRW0 and RRW1 (TRRW): 1

Number of waits according to bits RCW0 to RCW2 (TRCW): 1

Number of waits according to bits SRW0 to SRW2 (TSRW): 2

2. n = 0 to 7

CHAPTER 6 DMA FUNCTIONS (DMA CONTROLLER)

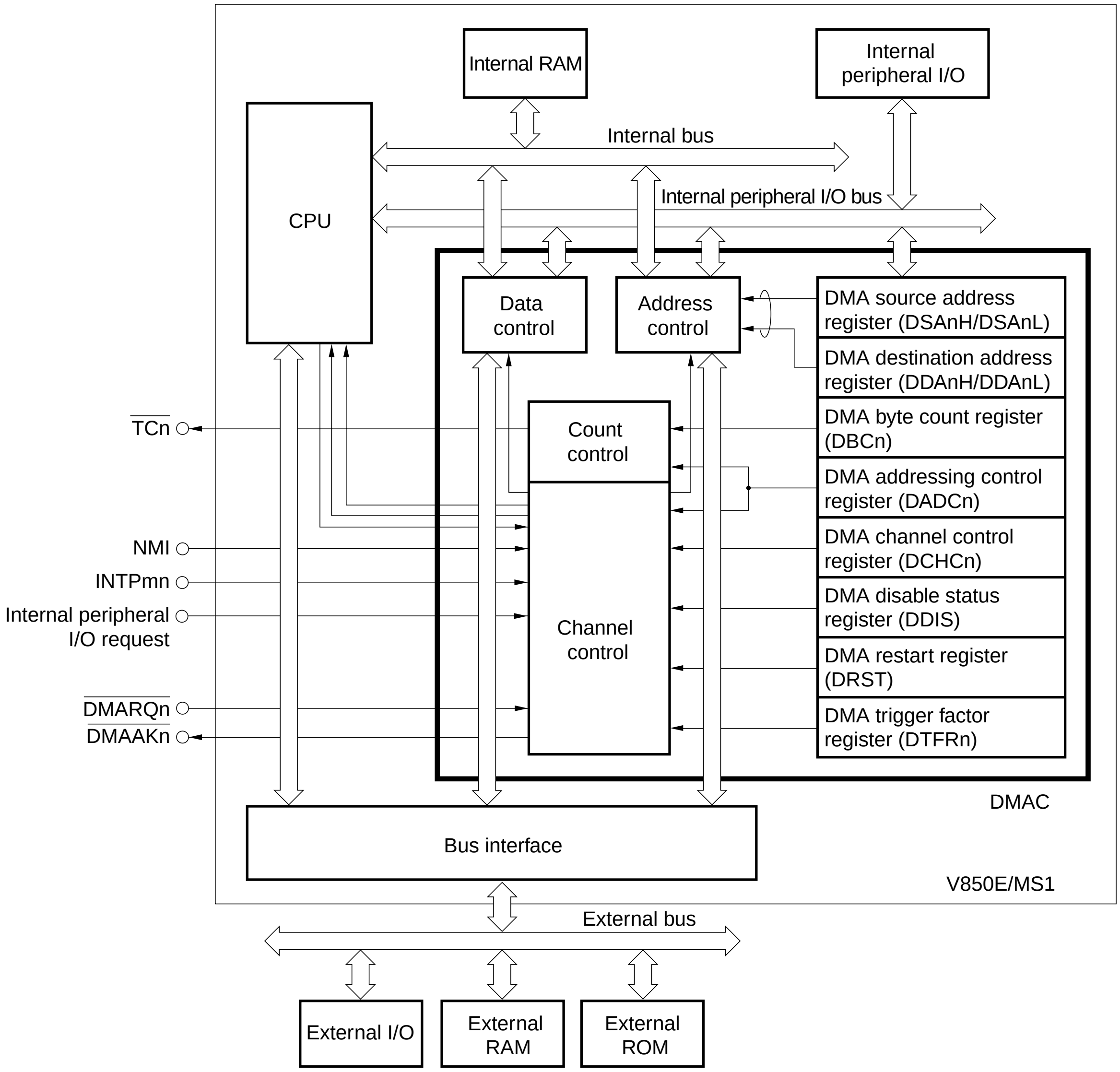
The V850E/MS1 includes a DMA (Direct Memory Access) controller (DMAC), which executes and controls DMA transfer.

The DMAC (DMA controller) transfers data between memory and I/O, or within memory, based on DMA requests issued by the internal peripheral I/O (serial interface and real-time pulse unit), $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$ pins, or software triggers.

6.1 Features

- 4 independent DMA channels
- Transfer unit: 8/16 bits
- Maximum transfer count: 65,536 (2^{16})
- Two types of transfer
  - Flyby (one-cycle) transfer
  - Two-cycle transfer
- Three transfer modes
  - Single transfer mode
  - Single-step transfer mode
  - Block transfer mode
- Transfer requests
  - $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$ pin (× 4)
  - Requests from internal peripheral I/O (serial interface and real-time pulse unit)
  - Requests from software
- Transfer objects
  - Memory to I/O and vice versa
  - Memory to memory
- DMA transfer end output signal ($\overline{\text{TC0}}$ to $\overline{\text{TC3}}$)

6.2 Configuration


Internal RAM
Internal peripheral I/O
Internal bus
Internal peripheral I/O bus
CPU
Data control
Address control
DMA source address register (DSAnH/DSAnL)
DMA destination address register (DDAnH/DDAnL)
DMA byte count register (DBCn)
DMA addressing control register (DADCn)
DMA channel control register (DCHCn)
DMA disable status register (DDIS)
DMA restart register (DRST)
DMA trigger factor register (DTFRn)
Count control
Channel control
$\overline{\text{TCn}}$
NMI
INTPmn
Internal peripheral I/O request
$\overline{\text{DMARQn}}$
$\overline{\text{DMAAKn}}$
DMAC
Bus interface
V850E/MS1
External bus
External I/O
External RAM
External ROM

Remark m = 10 to 15
n = 0 to 3

6.3 Control Registers

6.3.1 DMA source address registers 0 to 3 (DSA0 to DSA3)

These registers are used to set the DMA source addresses (26 bits each) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DSAnH and DSAnL.

During DMA transfer, the registers store the next DMA source addresses.

When flyby transfer between external memory and external I/O is specified with the TTYP bits of DMA addressing control register n (DADCn), the external memory addresses are set with the DSAn register. The setting made with DMA destination address register n (DDAn) is ignored.

(1) DMA source address registers 0H to 3H (DSA0H to DSA3H)

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DSA0H	0	0	0	0	0	0	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF1A0H	Undefined
DSA1H	0	0	0	0	0	0	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF1A8H	Undefined
DSA2H	0	0	0	0	0	0	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF1B0H	Undefined
DSA3H	0	0	0	0	0	0	SA25	SA24	SA23	SA22	SA21	SA20	SA19	SA18	SA17	SA16	FFFFF1B8H	Undefined

Bit Position	Bit Name	Function
9 to 0	SA25 to SA16	Source Address Sets the DMA source address (A25 to A16). During DMA transfer, it stores the next DMA source address. During flyby transfer between external memory and external I/O, it stores a memory address.

(2) DMA source address registers 0L to 3L (DSA0L to DSA3L)

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DSA0L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF1A2H	Undefined
DSA1L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF1AAH	Undefined
DSA2L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF1B2H	Undefined
DSA3L	SA15	SA14	SA13	SA12	SA11	SA10	SA9	SA8	SA7	SA6	SA5	SA4	SA3	SA2	SA1	SA0	FFFFF1BAH	Undefined

Bit Position	Bit Name	Function
15 to 0	SA15 to SA0	Source Address Sets the DMA source address (A15 to A0). During DMA transfer, it stores the next DMA source address. During flyby transfer between external memory and external I/O, it stores a memory address.

6.3.2 DMA destination address registers 0 to 3 (DDA0 to DDA3)

These registers are used to set the DMA destination addresses (26 bits each) for DMA channel n (n = 0 to 3). They are divided into two 16-bit registers, DDAnH and DDAnL.

During DMA transfer, the registers store the next DMA destination addresses.

When flyby transfer between external memory and external I/O is specified with the TTYP bits of DMA addressing control register n (DADCn), the setting of these registers are ignored. But when flyby transfer between internal RAM and internal peripheral I/O has been set, the DMA destination address registers (DDA0 to DDA3) must be set.

(1) DMA destination address registers 0H to 3H (DDA0H to DDA3H)

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DDA0H	0	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF1A4H	Undefined
DDA1H	0	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF1ACH	Undefined
DDA2H	0	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF1B4H	Undefined
DDA3H	0	0	0	0	0	0	DA25	DA24	DA23	DA22	DA21	DA20	DA19	DA18	DA17	DA16	FFFFF1BCH	Undefined

Bit Position	Bit Name	Function
9 to 0	DA25 to DA16	Destination Address Sets the DMA destination address (A25 to A16). During DMA transfer, it stores the next DMA destination address. This is disregarded during flyby transfer between external memory and external I/O, but be sure to set this register during flyby transfer between internal RAM and internal peripheral I/O.

(2) DMA destination address registers 0L to 3L (DDA0L to DDA3L)

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DDA0L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	FFFFF1A6H	Undefined
DDA1L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	FFFFF1AEH	Undefined
DDA2L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	FFFFF1B6H	Undefined
DDA3L	DA15	DA14	DA13	DA12	DA11	DA10	DA9	DA8	DA7	DA6	DA5	DA4	DA3	DA2	DA1	DA0	FFFFF1BEH	Undefined

Bit Position	Bit Name	Function
15 to 0	DA15 to DA0	Destination Address Sets the DMA destination address (A15 to A0). During DMA transfer, it stores the next DMA destination address. This is disregarded during flyby transfer between external memory and external I/O, but be sure to set this register during flyby transfer between internal RAM and internal peripheral I/O.

6.3.3 DMA byte count registers 0 to 3 (DBC0 to DBC3)

These 16-bit registers are used to set the byte transfer counts for DMA channel n (n = 0 to 3).

They store the remaining transfer counts during DMA transfer.

These registers are decremented by 1 for byte transfer and by two for 16-bit transfer. Transfer ends when a borrow occurs. Thus, “transfer count –1” should be set for byte transfer and “(transfer count –1) × 2” for 16-bit transfer.

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DBC0	BC 15	BC 14	BC 13	BC 12	BC 11	BC 10	BC 9	BC 8	BC 7	BC 6	BC 5	BC 4	BC 3	BC 2	BC 1	BC 0	FFFFF1E0H	Undefined
DBC1	BC 15	BC 14	BC 13	BC 12	BC 11	BC 10	BC 9	BC 8	BC 7	BC 6	BC 5	BC 4	BC 3	BC 2	BC 1	BC 0	FFFFF1E2H	Undefined
DBC2	BC 15	BC 14	BC 13	BC 12	BC 11	BC 10	BC 9	BC 8	BC 7	BC 6	BC 5	BC 4	BC 3	BC 2	BC 1	BC 0	FFFFF1E4H	Undefined
DBC3	BC 15	BC 14	BC 13	BC 12	BC 11	BC 10	BC 9	BC 8	BC 7	BC 6	BC 5	BC 4	BC 3	BC 2	BC 1	BC 0	FFFFF1E6H	Undefined

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>15 to 0</td><td>BC15 to BC0</td><td>Byte Count
Sets the byte transfer count. During DMA transfer, it stores the remaining byte transfer count.
<table>
<tr><th>DBCn</th><th>States</th></tr>
<tr><td>0000H</td><td>Byte transfer count 1 or the remaining byte transfer count</td></tr>
<tr><td>0001H</td><td>Byte transfer count 2 or the remaining byte transfer count</td></tr>
<tr><td>:</td><td>:</td></tr>
<tr><td>FFFFH</td><td>Byte transfer count 65,536 (2^{16}) or the remaining byte transfer count</td></tr>
</table>
</td></tr>
</table>

Remark n = 0 to 3

6.3.4 DMA addressing control registers 0 to 3 (DADC0 to DADC3)

These 16-bit registers are used to control the DMA transfer operation modes for DMA channel n (n = 0 to 3).

These registers can be read/written in 16-bit units.

Caution During DMA transfer, do not perform writing to these registers.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
DADC0	0	0	0	0	0	0	0	DS	SAD1	SAD0	DAD1	DAD0	TM1	TM0	TTYP	TDIR	FFFFF1F0H	0000H
DADC1	0	0	0	0	0	0	0	DS	SAD1	SAD0	DAD1	DAD0	TM1	TM0	TTYP	TDIR	FFFFF1F2H	0000H
DADC2	0	0	0	0	0	0	0	DS	SAD1	SAD0	DAD1	DAD0	TM1	TM0	TTYP	TDIR	FFFFF1F4H	0000H
DADC3	0	0	0	0	0	0	0	DS	SAD1	SAD0	DAD1	DAD0	TM1	TM0	TTYP	TDIR	FFFFF1F6H	0000H

Bit Position	Bit Name	Function
8	DS	Data Size Sets the transfer data size for DMA transfer. 0: 8 bits 1: 16 bits
7, 6	SAD1, SAD0	Source Address count Direction Sets the count direction of the source address for DMA channel n.

SAD1	SAD0	Count Direction
0	0	Increment
0	1	Decrement
1	0	Fixed
1	1	Setting prohibited

Remark n = 0 to 3

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>5, 4</td><td>DAD1,
DAD0</td><td>Destination Address count Direction
Sets the count direction of the destination address for DMA channel n.<table>
<tr><th>DAD1</th><th>DAD0</th><th>Count Direction</th></tr>
<tr><td>0</td><td>0</td><td>Increment</td></tr>
<tr><td>0</td><td>1</td><td>Decrement</td></tr>
<tr><td>1</td><td>0</td><td>Fixed</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>3, 2</td><td>TM1, TM0</td><td>Transfer Mode
Sets the transfer mode during DMA transfer.<table>
<tr><th>TM1</th><th>TM0</th><th>Transfer Mode</th></tr>
<tr><td>0</td><td>0</td><td>Single transfer mode</td></tr>
<tr><td>0</td><td>1</td><td>Single-step transfer mode</td></tr>
<tr><td>1</td><td>0</td><td>Block transfer mode</td></tr>
<tr><td>1</td><td>1</td><td>Setting prohibited</td></tr>
</table></td></tr>
<tr><td>1</td><td>TTYP</td><td>Transfer Type
Sets the DMA transfer type.
0: Two-cycle transfer
1: Flyby transfer</td></tr>
<tr><td>0</td><td>TDIR</td><td>Transfer Direction
Sets the transfer direction during transfer between I/O and memory. The setting is valid during flyby transfer only and ignored during two-cycle transfer.
0: Memory → I/O (read)
1: I/O → memory (write)</td></tr>
</table>

Remark n = 0 to 3

6.3.5 DMA channel control registers 0 to 3 (DCHC0 to DCHC3)

These 8-bit registers are used to control the DMA transfer operation mode for DMA channel n (n = 0 to 3).

These registers can be read/written in 8-bit units. (However, bit 7 is read-only and bits 2 and 1 are write-only. When the DMA channel control registers are read, bits 2 and 1 are always 0.)

	7	6	5	4	3	2	1	0	Address	After reset
DCHC0	TC0	0	0	0	0	INIT0	STG0	EN0	FFFFF5F0H	00H
DCHC1	TC1	0	0	0	0	INIT1	STG1	EN1	FFFFF5F2H	00H
DCHC2	TC2	0	0	0	0	INIT2	STG2	EN2	FFFFF5F4H	00H
DCHC3	TC3	0	0	0	0	INIT3	STG3	EN3	FFFFF5F6H	00H

Bit Position	Bit Name	Function
7	TCn	Terminal Count This status bit indicates whether DMA transfer through DMA channel n has ended or not. This bit can only be read. It is set (1) when DMA transfer ends with a terminal count and reset (0) when it is read. 0: DMA transfer has not ended. 1: DMA transfer has ended.
2	INITn	Initialize If this bit is set (1), the DMA transfer is forcibly terminated.
1	STGn	Software Trigger In DMA transfer enable state (TCn bit = 0, ENn bit = 1), if this bit is set (1), DMA transfer can be started by software.
0	ENn	Enable Specifies whether DMA transfer through DMA channel n is to be enabled or disabled. It is reset (0) when DMA transfer ends with a terminal count. It is also reset (0) when transfer is forcibly ended by means of setting (1) NMI input or INITn bit. 0: DMA transfer disabled. 1: DMA transfer enabled.

Remark n = 0 to 3

6.3.6 DMA trigger factor registers 0 to 3 (DTFR0 to DTFR3)

These 8-bit registers are used to control the DMA transfer start trigger through interrupt requests from peripheral I/O.

The interrupt requests that are set with these registers start DMA transfer.

These registers can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
DTFR0	0	0	IFC05	IFC04	IFC03	IFC02	IFC01	IFC00	FFFFF5E0H	00H
DTFR1	0	0	IFC15	IFC14	IFC13	IFC12	IFC11	IFC10	FFFFF5E2H	00H
DTFR2	0	0	IFC25	IFC24	IFC23	IFC22	IFC21	IFC20	FFFFF5E4H	00H
DTFR3	0	0	IFC35	IFC34	IFC33	IFC32	IFC31	IFC30	FFFFF5E6H	00H

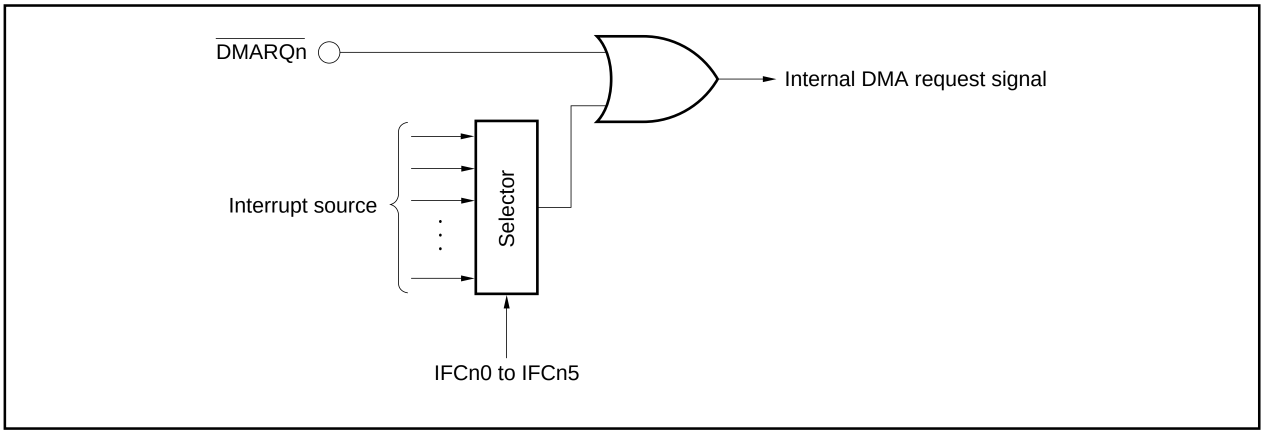
Bit Position	Bit Name	Function
5 to 0	IFCn5 to IFCn0	Interrupt Factor Code This code indicates the source of the DMA transfer trigger.

IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source
0	0	0	0	0	0	DMA request from internal peripheral I/O disabled.
0	0	0	0	0	1	INTCM40
0	0	0	0	1	0	INTCM41
0	0	0	0	1	1	INTCSI0
0	0	0	1	0	0	INTSR0
0	0	0	1	0	1	INTST0
0	0	0	1	1	0	INTCSI1
0	0	0	1	1	1	INTSR1
0	0	1	0	0	0	INTST1
0	0	1	0	0	1	INTCSI2
0	0	1	0	1	0	INTCSI3
0	0	1	0	1	1	INTP100/INTCC100
0	0	1	1	0	0	INTP101/INTCC101
0	0	1	1	0	1	INTP102/INTCC102
0	0	1	1	1	0	INTP103/INTCC103
0	0	1	1	1	1	INTP110/INTCC110
0	1	0	0	0	0	INTP111/INTCC111
0	1	0	0	0	1	INTP112/INTCC112
0	1	0	0	1	0	INTP113/INTCC113

Bit Position	Bit Name	Function						
		IFCn5	IFCn4	IFCn3	IFCn2	IFCn1	IFCn0	Interrupt Source
5 to 0	IFCn5 to IFCn0	0	1	0	0	1	1	INTP120/INTCC120
		0	1	0	1	0	0	INTP121/INTCC121
		0	1	0	1	0	1	INTP122/INTCC122
		0	1	0	1	1	0	INTP123/INTCC123
		0	1	0	1	1	1	INTP130/INTCC130
		0	1	1	0	0	0	INTP131/INTCC131
		0	1	1	0	0	1	INTP132/INTCC132
		0	1	1	0	1	0	INTP133/INTCC133
		0	1	1	0	1	1	INTP140/INTCC140
		0	1	1	1	0	0	INTP141/INTCC141
		0	1	1	1	0	1	INTP142/INTCC142
		0	1	1	1	1	0	INTP143/INTCC143
		0	1	1	1	1	1	INTP150/INTCC150
		1	0	0	0	0	0	INTP151/INTCC151
		1	0	0	0	0	1	INTP152/INTCC151
		1	0	0	0	1	0	intp153/intcc153
★		1	0	0	0	1	1	INTAD
		Other than above						Setting prohibited

Remark n = 0 to 3

Remark The relationship between the $\overline{\text{DMARQn}}$ signal and the interrupt source which becomes the DMA transfer start trigger is as follows (n = 0 to 3).



6.3.7 DMA disable status register (DDIS)

This register holds the contents of the ENn bit of the DCHCn register during NMI input (n = 0 to 3). It is read-only, in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
DDIS	0	0	0	0	CH3	CH2	CH1	CH0	Address FFFFF5D0H	After reset 00H

Bit Position	Bit Name	Function
3 to 0	CHn (n = 3 to 0)	NMI Interruption Status Reflects the contents of the ENn bit of the DCHCn register during NMI input. The contents of this register are held until the next NMI input or until the next system reset.

6.3.8 DMA restart register (DRST)

This register is used to restart DMA transfer that was forcibly interrupted during NMI input. The RENn bit of this register and the ENn bit of the DCHCn register are linked to each other (n = 0 to 3). After NMI is completed, the DDIS register is referred to and the DMA channel that was interrupted is confirmed, then by setting the RENn bit in the corresponding channel (1), DMA transfer can be restarted. The register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
DRST	0	0	0	0	REN3	REN2	REN1	REN0	Address FFFFF5D2H	After reset 00H

Bit Position	Bit Name	Function
3 to 0	RENn (n = 3 to 0)	Restart Enable This sets DMA transfer enable/disable in DMA channel n. If DMA transfer is completed in accordance with the terminal count, it is reset (0). It is also reset (0) when DMA is forcibly terminated by NMI input or by setting of the INITn bit (1) in the DCHCn register. 0: DMA transfer disabled. 1: DMA transfer enabled.

6.3.9 Flyby transfer data wait control register (FDW)

To prevent illegal writing during flyby transfer, this register sets the insertion of wait states (TF) for securing the time from when the write signal ($\overline{\text{IOWR}}$, $\overline{\text{UWR}}$, $\overline{\text{LWR}}$, $\overline{\text{WE}}$) becomes inactive until the read signal ($\overline{\text{RD}}$, $\overline{\text{IORD}}$, $\overline{\text{OE}}$) becomes inactive. This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
FDW	FDW7	FDW6	FDW5	FDW4	FDW3	FDW2	FDW1	FDW0	Address FFFFF06CH	After reset 00H
Memory Block	7	6	5	4	3	2	1	0		

Bit Position	Bit Name	Function
7 to 0	FDWn (n = 7 to 0)	Flyby Data Wait Sets wait state insertion for memory block n. 0: Wait state not inserted. 1: Wait state inserted.

Caution Write to the FDW register after reset, and then do not change the value. Also, do not access an external memory area until the initial setting of the FDW register is complete. (However, the memory area 0000000H to 01FFFFFH is excluded.)

Remark Setting of the FDW register is valid during the DMA transfers shown below.

Type of Memory Object of Transfer	SRAM, Page ROM	DRAM
Memory → I/O	Valid	Valid
I/O → Memory	Valid	Invalid

6.4 DMA Bus States

6.4.1 Types of bus states

The DMAC bus cycle consists of the following 25 states:

(1) TI state

The TI state is idle state, during which no access request is issued.

The $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$ signals are sampled at the falling edge of the CLKOUT signal.

(2) T0 state

DMA transfer ready state. (A DMA transfer request has been issued, causing bus mastership to be acquired for the first DMA transfer).

(3) T1R state

The bus enters the T1R state at the beginning of a read operation in two-cycle transfer mode. Address driving starts. After entering the T1R state, the bus invariably enters the T2R state.

(4) T1RI state

T1RI is a state in which the bus is waiting for the acknowledge in response to an external memory read request. After entering the last T1RI state, the bus invariably enters the T2R state.

(5) T2R state

The T2R state corresponds to the last state of a read operation in two-cycle transfer mode, or to a wait state. In the last T2R state, read data is sampled. After entering the last T2R state, the bus invariably enters the T1W state.

(6) T2RI state

Internal peripheral I/O or internal RAM DMA transfer ready state (Bus mastership is acquired for DMA transfer to internal peripheral I/O or internal RAM). After entering the last T2RI state, the bus invariably enters the T1W state.

(7) T1W state

The bus enters the T1W state at the beginning of a write operation in two-cycle transfer mode. Address driving starts. After entering the T1W state, the bus invariably enters the T2W state.

(8) T1WI state

T1WI is a state in which the bus is waiting for the acknowledge signal in response to an external memory write request. After entering the last T1WI state, the bus invariably enters the T2W state.

(9) T2W state

The T2W state corresponds to the last state of a write operation in two-cycle transfer mode, or to a wait state. In the last T2W state, the write strobe signal is made inactive.

(10) T1F state

The bus enters the T1F state at the beginning of a flyby transfer from internal peripheral I/O to internal RAM. The read cycle from internal peripheral I/O is started. After entering the T1F state, the bus invariably enters the T2F state.

(11) T2F state

The T2F state corresponds to the middle state of a flyby transfer from internal peripheral I/O to internal RAM. The write cycle to internal RAM is started. After entering the T2F state, the bus invariably enters the T3F state.

(12) T3F state

The T3F state corresponds to the last state of a flyby transfer from internal peripheral I/O to internal RAM, or a wait state. In the last T3F state, the write strobe signal is made inactive.

(13) T1FR state

The bus enters the T1FR state at the beginning of a flyby transfer from internal RAM to internal peripheral I/O. The read cycle from internal RAM is started. After entering the T1FR state, the bus invariably enters the T2FR state.

(14) T2FR state

The T2FR state corresponds to the middle state of a flyby transfer from internal RAM to internal peripheral I/O. The write cycle to internal peripheral I/O is started. After entering the T2FR state, the bus invariably enters the T3FR state.

(15) T3FR state

T3FR is a state in which it is judged whether a flyby transfer from internal RAM to internal peripheral I/O is continued or not. If the next transfer is executed in block transfer mode, the bus enters the T1FRB state after the T3FR state, otherwise, the bus enters the T4 state.

(16) T1FRB state

The bus enters the T1FRB state at the beginning of a flyby block transfer from internal RAM to internal peripheral I/O. The read cycle from internal RAM is started.

(17) T1FRBI state

The T1FRBI state corresponds to a wait state of a flyby block transfer from internal RAM to internal peripheral I/O.

A wait state requested by peripheral hardware is generated, and the bus enters the T2FRB state.

(18) T2FRB state

The T2FRB state corresponds to the middle state of a flyby block transfer from internal RAM to internal peripheral I/O. The write cycle to internal peripheral I/O is started. After entering the T2FRB state, the bus invariably enters the T3FRB state.

(19) T3FRB state

T3FRB is a state in which it is judged whether a flyby transfer from internal RAM to internal peripheral I/O is continued or not. If the next transfer is executed in block transfer mode, the bus enters the T1FRB state after the T3FRB state, otherwise, the bus enters the T4 state.

(20) T4 state

The T4 state corresponds to a wait state of a flyby transfer from internal RAM to internal peripheral I/O. A wait state requested by peripheral hardware is generated, and the bus enters the T3 state.

(21) T1FH state

The T1FH state corresponds to the standard state of a flyby transfer between external memory and external I/O, and is the executing cycle of this transfer. After entering the T1FH state, the bus enters the T2FH state.

(22) T1FHI state

The T1FHI state corresponds to the last state of a flyby transfer between external memory and external I/O, and is a state in which the bus is waiting for end of DMA flyby transfer. After entering the T1FHI state, the bus is released, and enters the TE state.

(23) T2FH state

T2FH is a state in which it is judged whether a flyby transfer between external memory and external I/O is continued or not. If the next transfer is executed in block transfer mode, the bus enters the T1FH state after the T2FH state, otherwise, when a wait is issued, the bus enters the T1FHI state. When a wait is not issued, the bus is released, and enters the TE state.

(24) T3 state

The bus enters the T3 state when a DMA transfer has been completed, and the bus has been released. After entering the T3 state, the bus invariably enters the TE state.

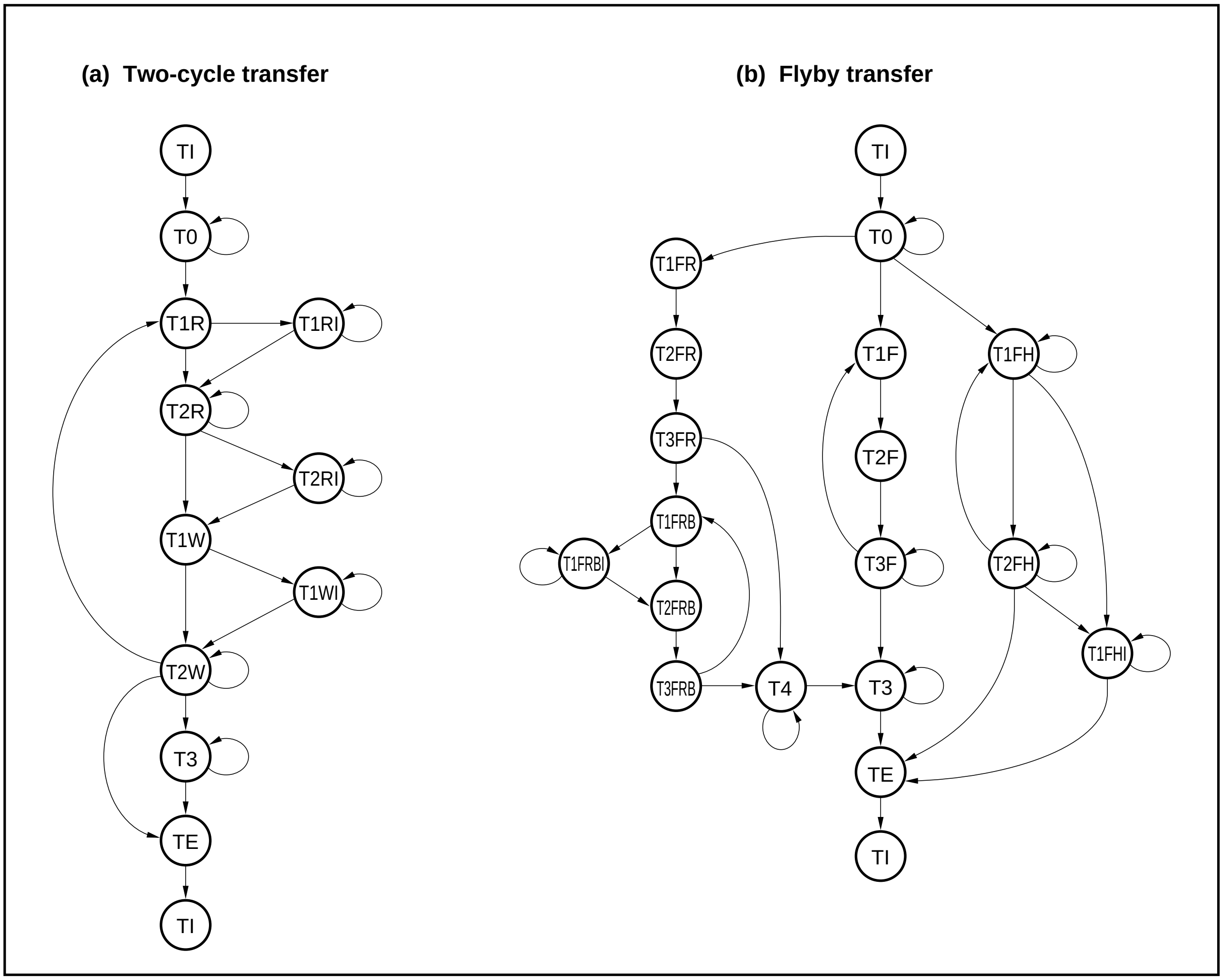
(25) TE state

The TE state corresponds to the output state. In the TE state, the DMAC outputs the DMA transfer end signal ($\overline{\text{TCn}}$), and initializes miscellaneous internal signals (n = 0 to 3). After entering the TE state, the bus invariably enters the TI state.

6.4.2 DMAC state transition

Except block transfer mode, each time the processing for a DMA service is completed, the bus is released (the bus enters bus release mode).

Figure 6-1. DMAC Bus Cycle State Transition Diagram



6.5 Transfer Mode

6.5.1 Single transfer mode

In single transfer mode, the DMAC releases the bus at each byte/halfword transfer. If there is a subsequent DMA transfer request, transfer is performed again. This operation continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence.

Figures 6-2 and 6-3 show examples of single transfer. Figure 6-3 shows an example of single transfer in which a higher priority DMA request is issued. DMA channels 0 to 2 are in block transfer mode and channel 3 is in single transfer mode.

Figure 6-2. Single Transfer Example 1

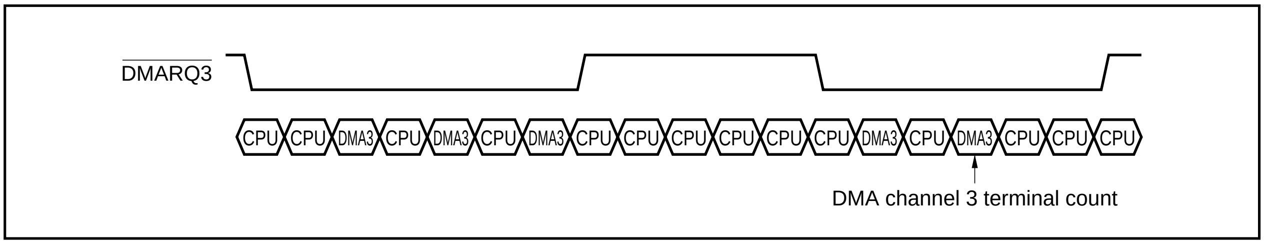
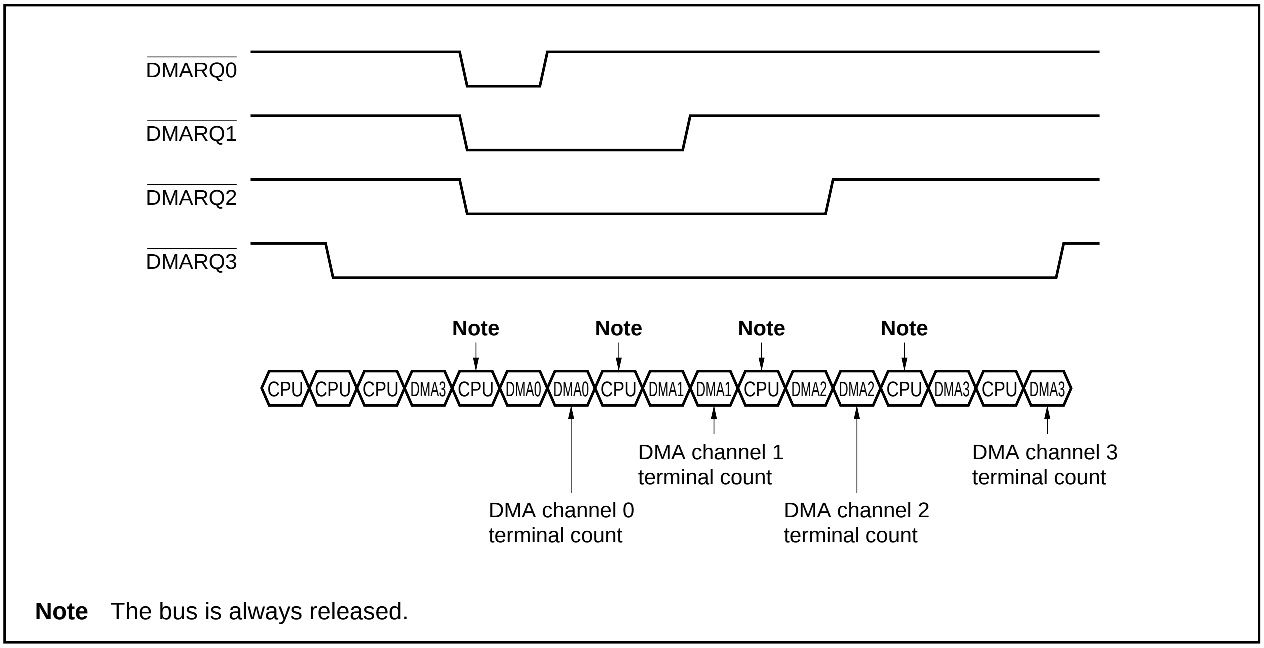


Figure 6-3. Single Transfer Example 2


$\overline{\text{DMARQ0}}$
$\overline{\text{DMARQ1}}$
$\overline{\text{DMARQ2}}$
$\overline{\text{DMARQ3}}$
Note
Note
Note
Note
CPU
CPU
CPU
DMA3
CPU
DMA0
DMA0
CPU
DMA1
DMA1
CPU
DMA2
DMA2
CPU
DMA3
CPU
DMA3
DMA channel 1 terminal count
DMA channel 3 terminal count
DMA channel 0 terminal count
DMA channel 2 terminal count

Note The bus is always released.

6.5.2 Single-step transfer mode

In single-step transfer mode, DMAC releases the bus at each byte/halfword transfer. Once a request signal ($\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$) is received, this operation continues until a terminal count occurs.

When the DMAC has released the bus, if another higher priority DMA transfer request is issued, the higher priority DMA request always takes precedence.

Figures 6-4 and 6-5 show examples of single-step transfer.

Figure 6-4. Single-Step Transfer Example 1

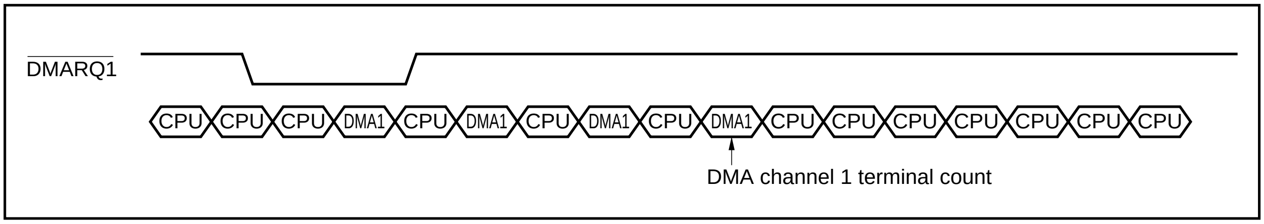
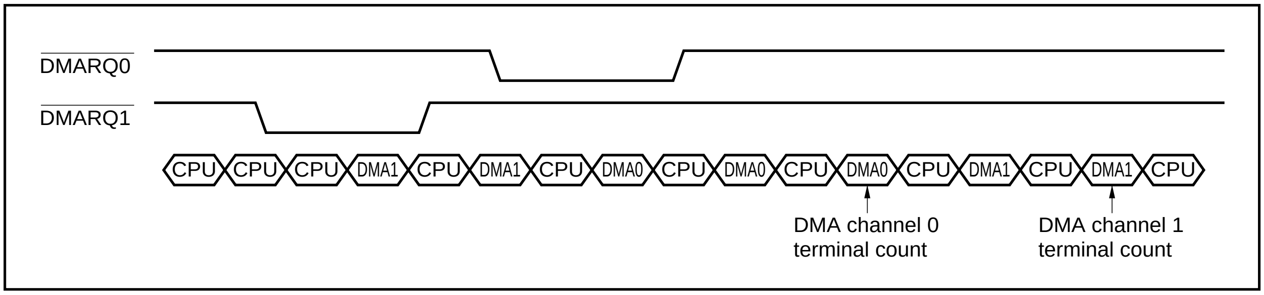

DMARQ1
CPU CPU CPU DMA1 CPU DMA1 CPU DMA1 CPU DMA1 CPU CPU CPU CPU CPU CPU CPU
DMA channel 1 terminal count

Figure 6-5. Single-Step Transfer Example 2


DMARQ0
DMARQ1
CPU CPU CPU DMA1 CPU DMA1 CPU DMA0 CPU DMA0 CPU DMA0 CPU DMA1 CPU DMA1 CPU
DMA channel 0 terminal count
DMA channel 1 terminal count

6.5.3 Block transfer mode

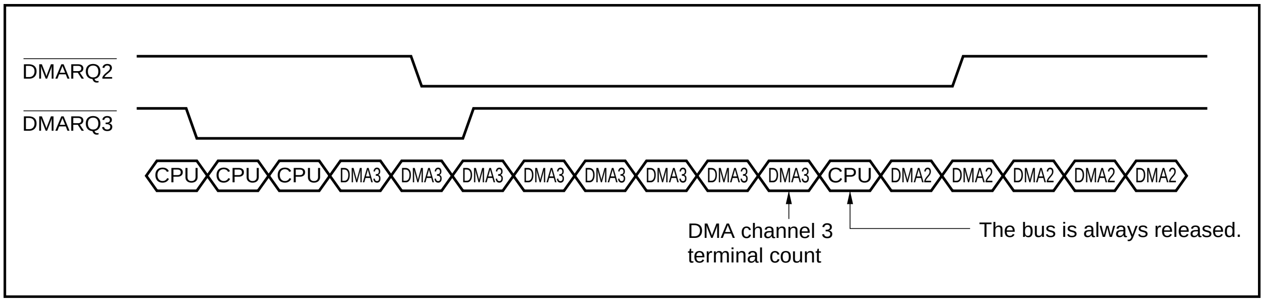
In block transfer mode, once transfer starts, the transfer continues without the bus being released, until a terminal count occurs. No other DMA requests are accepted during block transfer.

After the block transfer ends and DMAC releases the bus, another DMA transfer can be accepted.

Figures 6-6 shows an example of block transfer. In this block transfer example, a high priority DMA request is issued. DMA channels 2 and 3 are in block transfer mode.

Note that caution is required when in block transfer mode. For details, refer to **6.19 Precautions**.

Figure 6-6. Block Transfer Example


DMARQ2
DMARQ3
CPU CPU CPU DMA3 DMA3 DMA3 DMA3 DMA3 DMA3 DMA3 DMA3 CPU DMA2 DMA2 DMA2 DMA2 DMA2
DMA channel 3 terminal count
The bus is always released.

6.6 Transfer Types

6.6.1 Two-cycle transfer

In two-cycle transfer, data transfer is performed in two-cycles, source to DMAC then DMAC to destination.

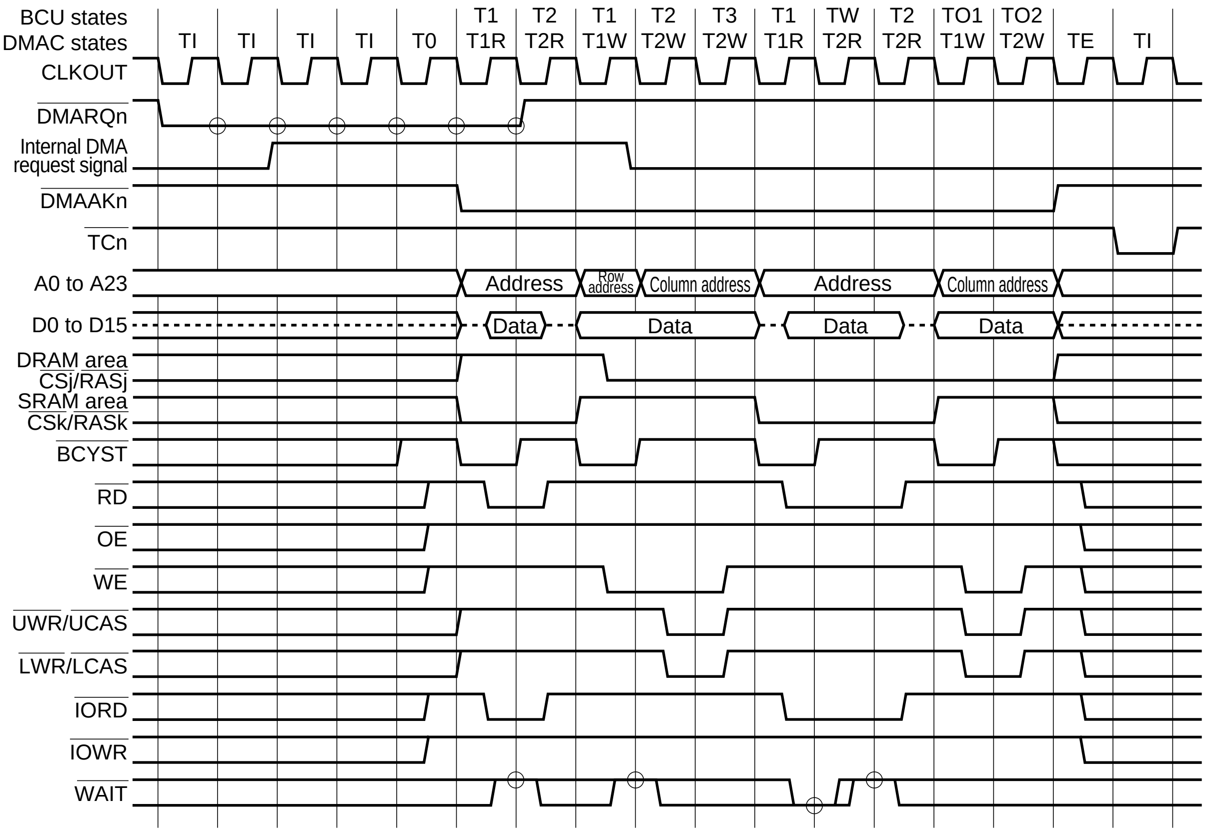
In the first cycle, the source address is output to perform reading from the source to DMAC. In the second cycle, the destination address is output to perform writing from DMAC to the destination.

Figure 6-7 shows examples of two-cycle transfer.

Note that caution is required when in two-cycle transfer. For details, refer to **6.19 Precautions**.

Figure 6-7. Timing of Two-Cycle Transfer (1/4)

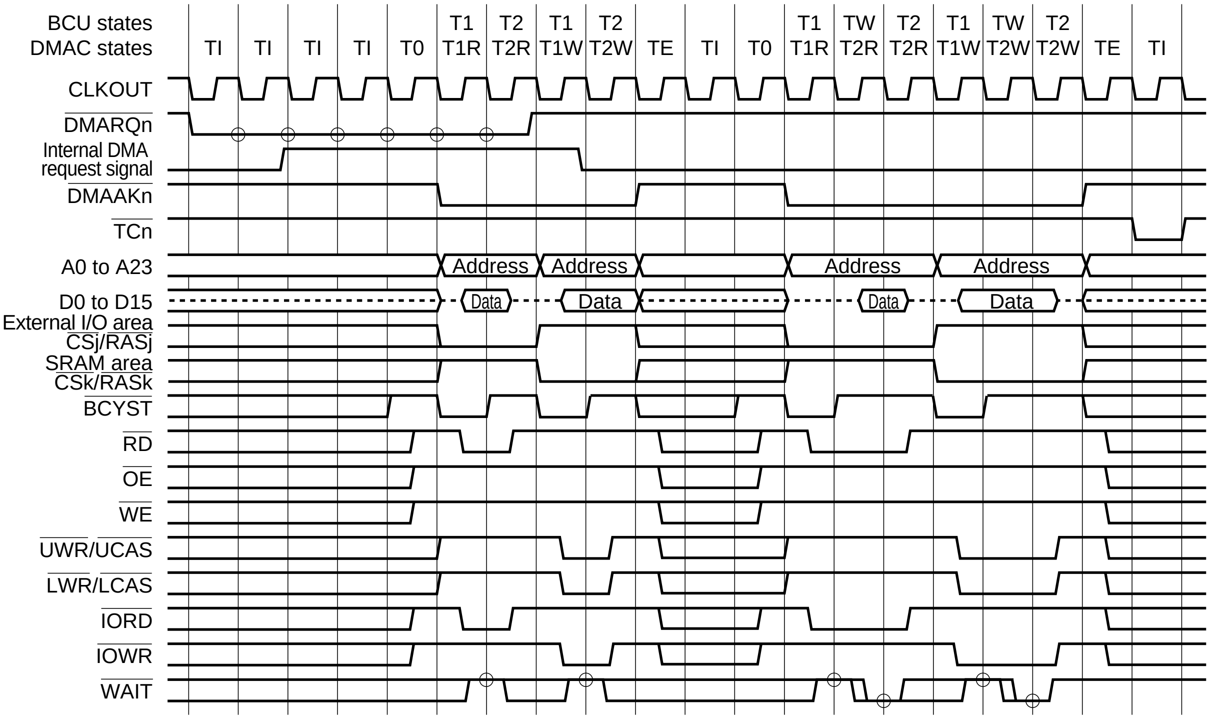
(a) Block transfer mode (SRAM → DRAM)



Remarks
1. The circles indicate the sampling timing.
2. Broken lines indicate high impedance.
3. n = 0 to 3
   j = 0 to 7, k = 0 to 7 (However, j ≠ k.)

Figure 6-7. Timing of Two-Cycle Transfer (2/4)

(b) Single-step transfer mode (External I/O → SRAM)



Remarks **1.** The circles indicate the sampling timing.

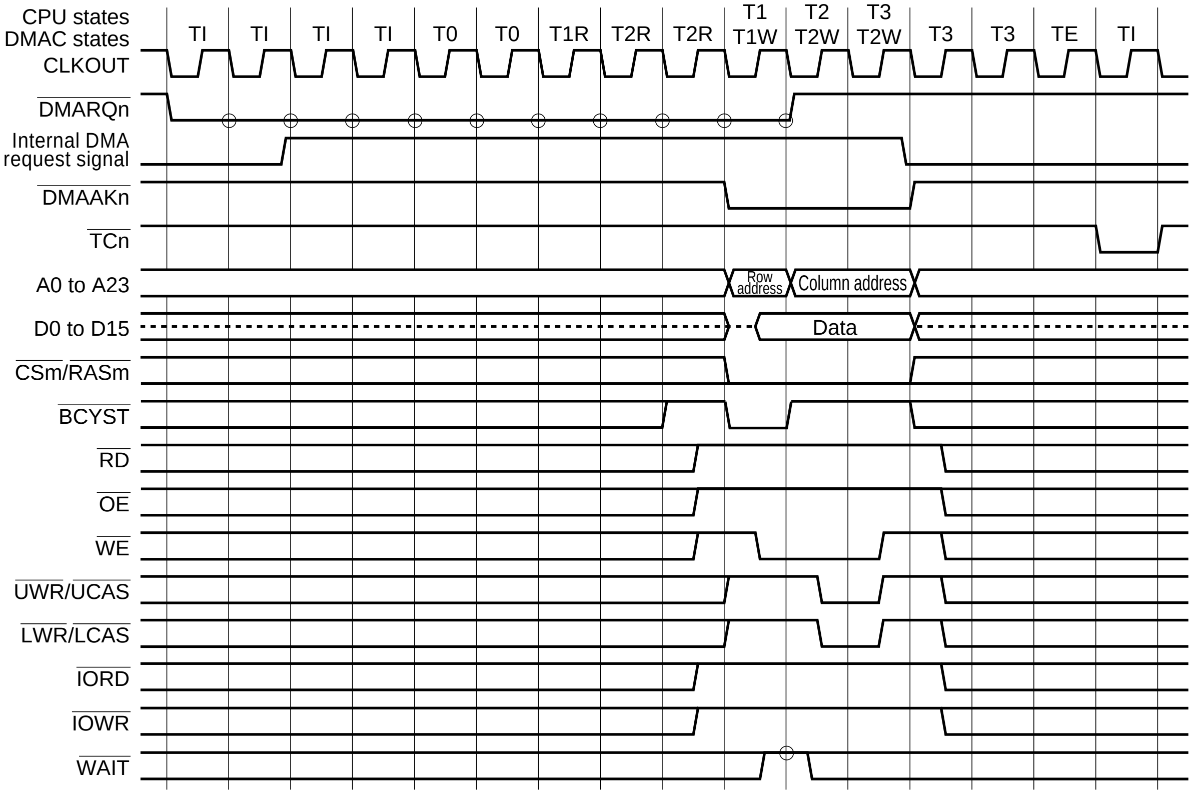
2. Broken lines indicate high impedance.

3. n = 0 to 3

j = 0 to 7, k = 0 to 7 (However, j ≠ k.)

Figure 6-7. Timing of Two-Cycle Transfer (3/4)

(c) Single transfer mode (Internal peripheral I/O → DRAM)



Remarks **1.** The circles indicate the sampling timing.

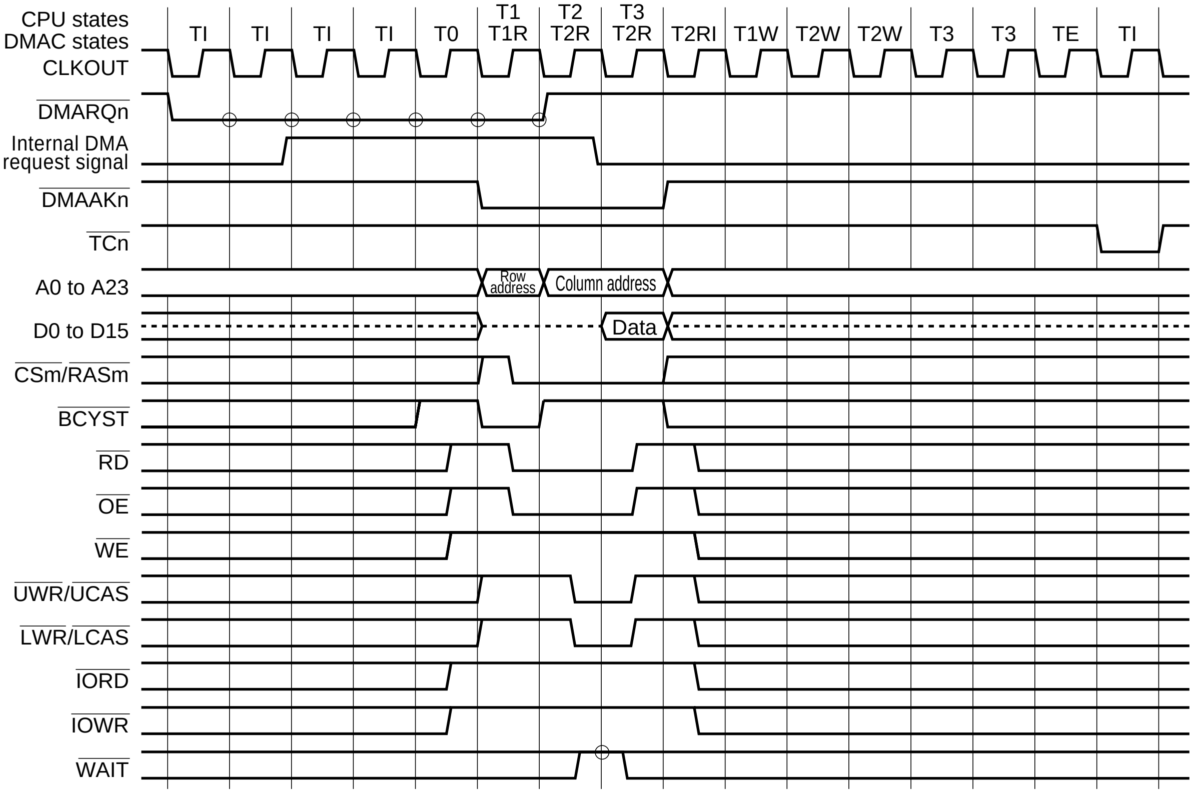
2. Broken lines indicate high impedance.

3. n = 0 to 3

m = 0 to 7

Figure 6-7. Timing of Two-Cycle Transfer (4/4)

(d) Single transfer mode (DRAM → Internal peripheral I/O)



Remarks **1.** The circles indicate the sampling timing.

2. Broken lines indicate high impedance.

3. n = 0 to 3

m = 0 to 7

6.6.2 Flyby transfer

The V850E/MS1 supports flyby transfer between external memory and external I/O, and internal RAM and internal peripheral I/O.

(1) Flyby transfer between external memory and external I/O

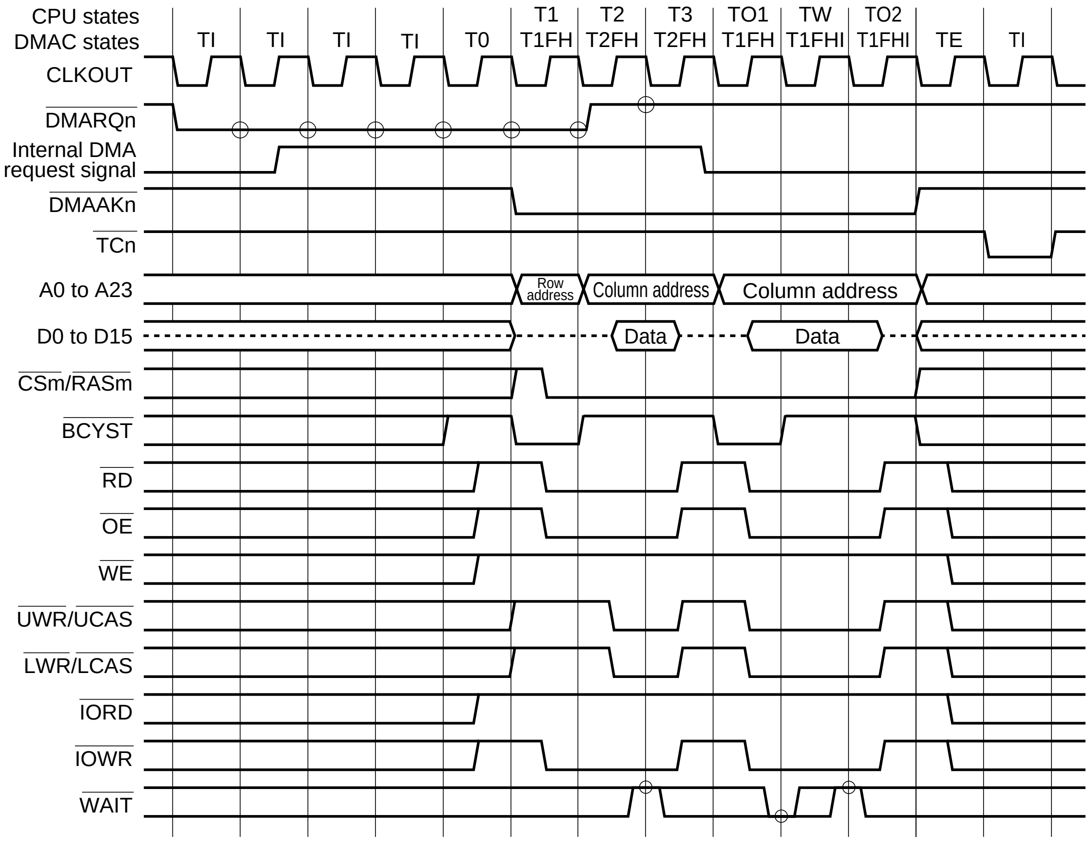
This data transfer between memory and I/O is performed in one cycle. To achieve single-cycle transfer, the memory address is always output irrespective of whether it is that of the source or the destination, and the read/write strobe signals for the memory and I/O are made active at the same time.

The external I/O is selected with the $\overline{\text{DMAAK0}}$ to $\overline{\text{DMAAK3}}$ signal.

Figure 6-8 shows examples of flyby DMA transfer for an external device.

Figure 6-8. Timing of Flyby Transfer (DRAM → External I/O) (1/3)

(a) Block transfer mode



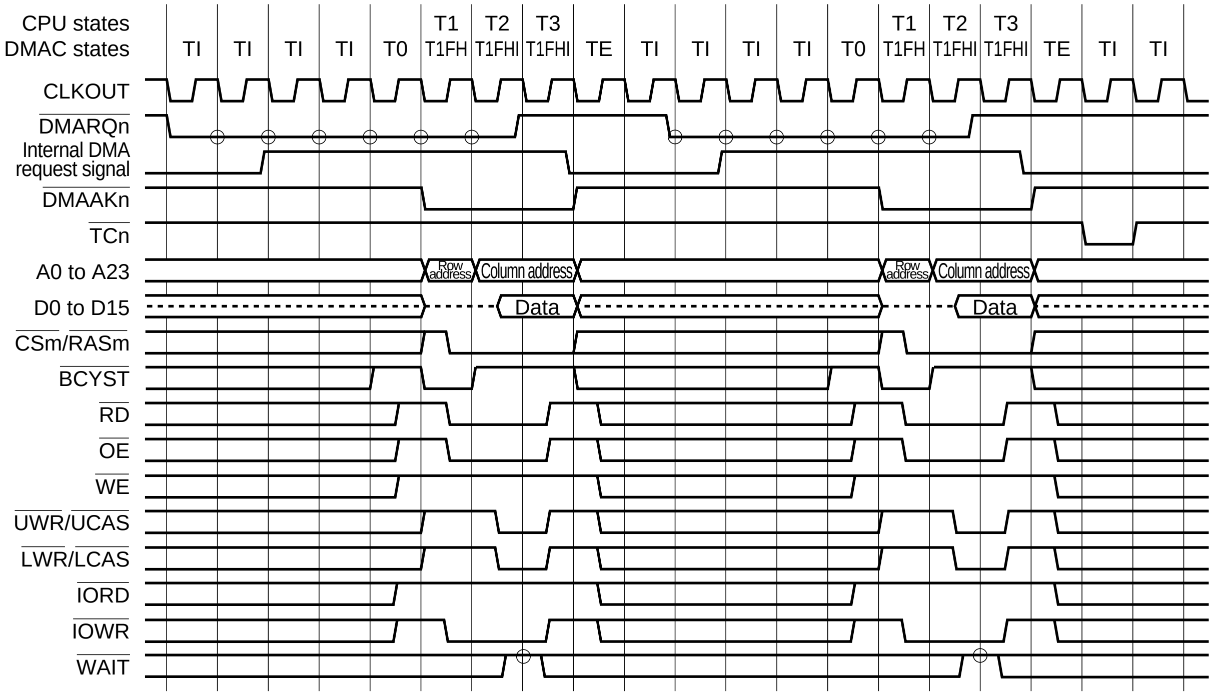
Remarks 1. The circles indicate the sampling timing.

2. Broken lines indicate high impedance.

3. n = 0 to 3
m = 0 to 7

Figure 6-8. Timing of Flyby Transfer (DRAM → External I/O) (2/3)

(b) Single transfer mode



Remarks **1.** The circles indicate the sampling timing.

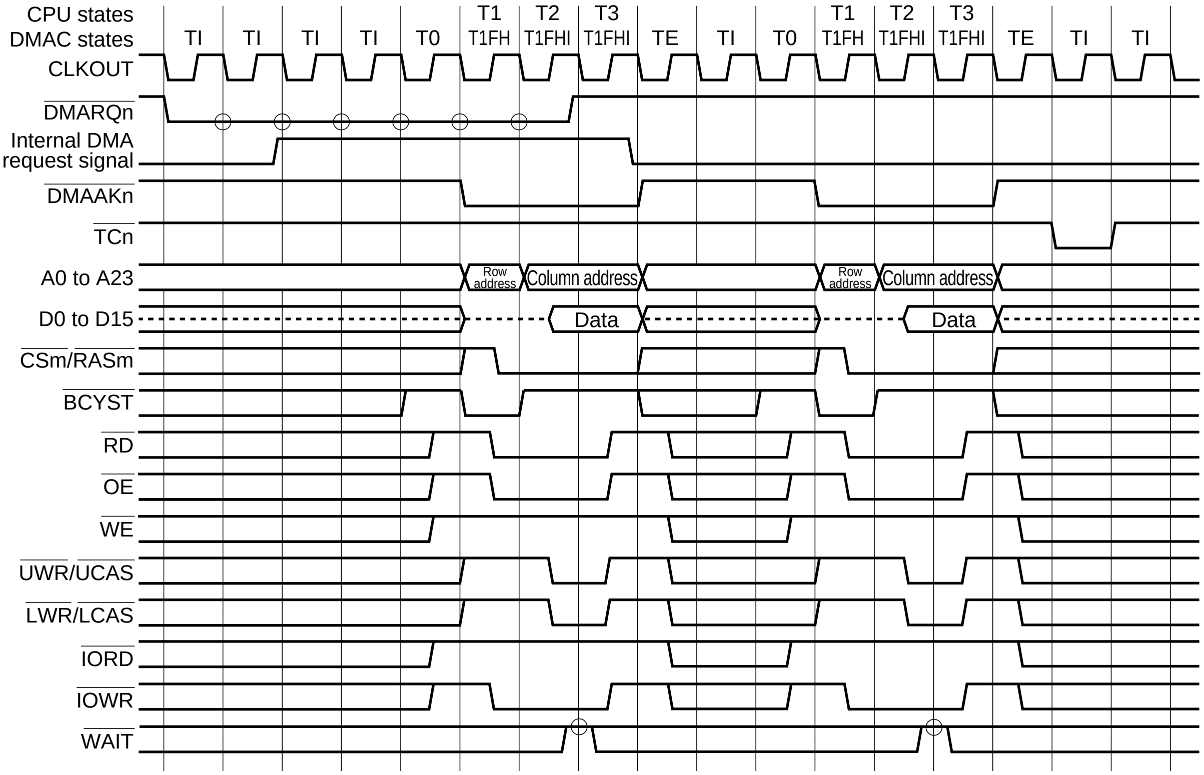
2. Broken lines indicate high impedance.

3. n = 0 to 3

m = 0 to 7

Figure 6-8. Timing of Flyby Transfer (DRAM → External I/O) (3/3)

(c) Single-step transfer mode



Remarks **1.** The circles indicate the sampling timing.

2. Broken lines indicate high impedance.

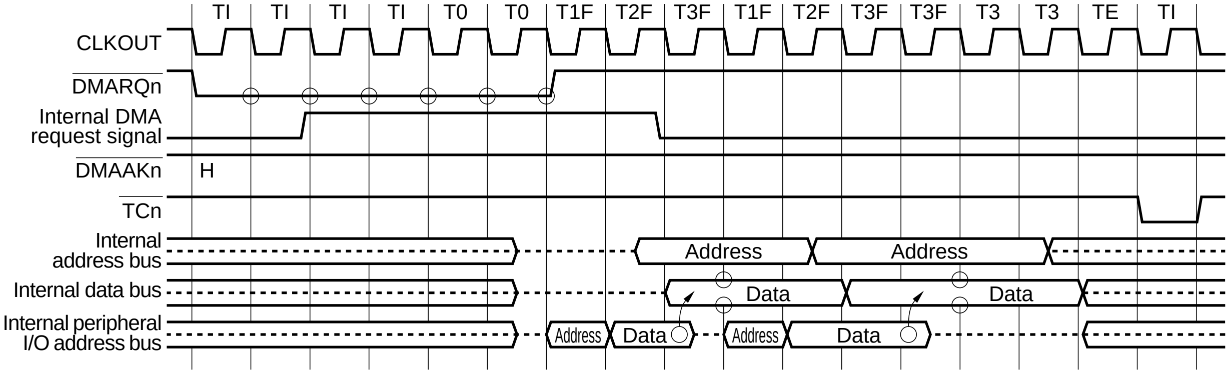
3. n = 0 to 3
m = 0 to 7

(2) Flyby transfer between internal RAM and internal peripheral I/O

Internal RAM and internal peripheral I/O are mapped on different address spaces. Therefore, different addresses are always output, and the read/write strobe signals for internal RAM and internal peripheral I/O are controlled at the same time.

Figure 6-9 shows an example of flyby DMA transfer (block transfer mode) between internal RAM and internal peripheral I/O.

Figure 6-9. Timing of Flyby Transfer (Internal Peripheral I/O → Internal RAM)



Remarks **1.** The circles indicate the sampling timing.

2. Broken lines indicate high impedance.

3. n = 0 to 3

4. With this timing, the external bus operates independently of the internal bus, so there is no influence on the external bus.

6.7 Transfer Objects

6.7.1 Transfer type and transfer objects

Table 6-1 lists the relationship between transfer type and transfer object.

Cautions 1. Among the transfer destinations and sources shown in Table 6-1, when an "×" is indicated for a combination, that operation is not guaranteed.

2. Make the data bus width of the transfer destination and source the same (for two-cycle transfer and flyby transfer).

Table 6-1. Relationship Between Transfer Type and Transfer Object

(a) Two-cycle transfer

		Destination			
		Internal peripheral I/O	External I/O	Internal RAM	External memory
Source	Internal peripheral I/O	×	×	○	○
	External I/O	×	×	○	○
	Internal RAM	○	○	○	○
	External memory	○	○	○	○

(b) Flyby transfer

		Destination			
		Internal peripheral I/O	External I/O	Internal RAM	External memory
Source	Internal peripheral I/O	×	×	○	×
	External I/O	×	×	×	○
	Internal RAM	○	×	×	×
	External memory	×	○	×	×

Remark ○: Possible
×: Impossible

6.7.2 External bus cycle during DMA transfer

The external bus cycle during DMA transfer is as follows.

Table 6-2. External Bus Cycle During DMA Transfer

Transfer Type	Transfer Object	External Bus Cycle	
Two-cycle transfer	Internal peripheral I/O, Internal RAM	None[Note]	—
	External I/O	Yes	SRAM cycle
	External memory	Yes	Memory access cycle set in the BCT register
Flyby transfer	Between internal RAM and internal peripheral I/O	None[Note]	—
	Between external memory and external I/O	Yes	The memory access DMA flyby transfer cycle set by the BCT register as external memory

Note Other external bus cycles, such as a CPU-based bus cycle, can be started.

6.8 DMA Channel Priorities

The DMA channel priorities are fixed, as follows:

DMA channel 0 > DMA channel 1 > DMA channel 2 > DMA channel 3

These priorities are valid in the TI state only. In block transfer mode, the channel used for transfer is never switched.

In single-step transfer mode, if a higher priority DMA transfer request is issued while the bus is released (in the TI state), the higher priority DMA transfer request is accepted.

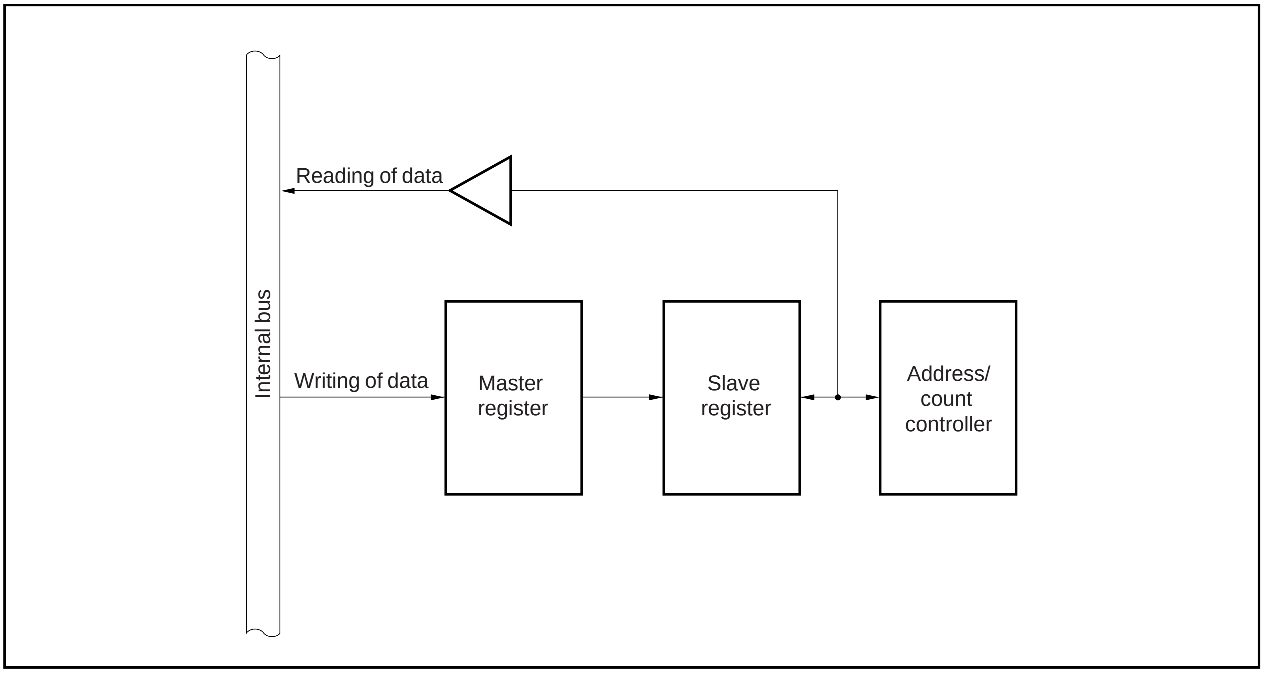
6.9 Next Address Setting Function

The DMA source address registers (DSAnH, DSAnL) DMA destination address registers (DDAnH, DDAnL) and DMA byte count register (DBCn) are buffer registers with a 2-stage FIFO configuration (n = 0 to 3).

When the terminal count is issued, these registers are rewritten with the value that was set just previously. Therefore, during DMA transfer, these registers' contents do not become valid even if they are rewritten. When starting DMA transfer with the rewritten contents of these registers, set the ENn bit (1) of the DCHCn register.

Figure 6-10 shows the buffer register configuration.

Figure 6-10. Buffer Register Configuration



6.10 DMA Transfer Start Factors

There are 3 types of DMA transfer start factors, as shown below.

(1) Request from an external pin ($\overline{\text{DMARQn}}$)

Although requests from the $\overline{\text{DMARQn}}$ pin are sampled each time the CLKOUT signal falls, sampling should be continued until the $\overline{\text{DMAAKn}}$ signal becomes active (n = 0 to 3).

If a state in which the ENn bit of the DCHCn register = 1 and the TCn bit = 0 is set, the $\overline{\text{DMARQn}}$ signal in the T1 state becomes active. If the $\overline{\text{DMARQn}}$ signal becomes active in the T1 state, it changes to the T0 state and DMA transfer starts.

(2) Request from software

If the STGn, ENn and TCn bits of the DCHCn register are set as follows, DMA transfer starts (n = 0 to 3).

- STGn bit = 1
- ENn bit = 1
- TCn bit = 0

(3) Request from internal peripheral I/O

If, when the ENn and TCn bits of the DCHCn register are set as shown below, an interrupt request is issued from the internal peripheral I/O that is set in the DTFRn register, DMA transfer starts (n = 0 to 3).

- ENn bit = 1
- TCn bit = 0

6.11 Interrupting DMA Transfer

6.11.1 Interruption factors

DMA transfer is interrupted if the following factors occur.

- Bus hold
- Refresh cycle

If the factor that is interrupting DMA transfer disappears, DMA transfer promptly restarts.

6.11.2 Forcible interruption

DMA transfer can be forcibly interrupted by an NMI input during DMA transfer.

At such a time, the DMAC resets the ENn bit of the DCHCn register of all channels (0) and activates the DMA transfer disabled state, after which the DMA transfer being executed when the NMI was input is terminated (n = 0 to 3).

When in the single step mode or block transfer mode, the DMA transfer request is held in the DMAC. If the ENn bit is reset (1), DMA transfer restarts from the point where it was interrupted.

When in the single transfer mode, if the ENn bit is set (1), the next DMA transfer request is received and DMA transfer starts.

6.12 Terminating DMA Transfer

6.12.1 DMA transfer end interrupt

When DMA transfer ends and the TC bit of the corresponding DCHCn register is set (1), a DMA transfer end interrupt (INTDMAn) is issued (n = 0 to 3) to the interrupt controller (INTC).

6.12.2 Terminal count output

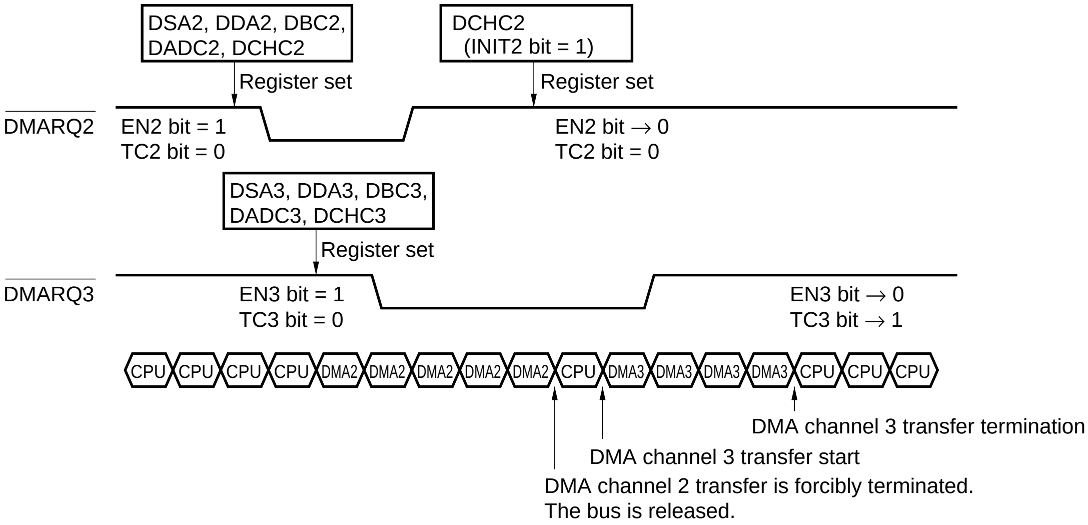
In the TI state directly after the cycle when DMA transfer ends (TE state), the $\overline{\text{TCn}}$ signal output becomes active for 1 clock cycle.

6.12.3 Forcible termination

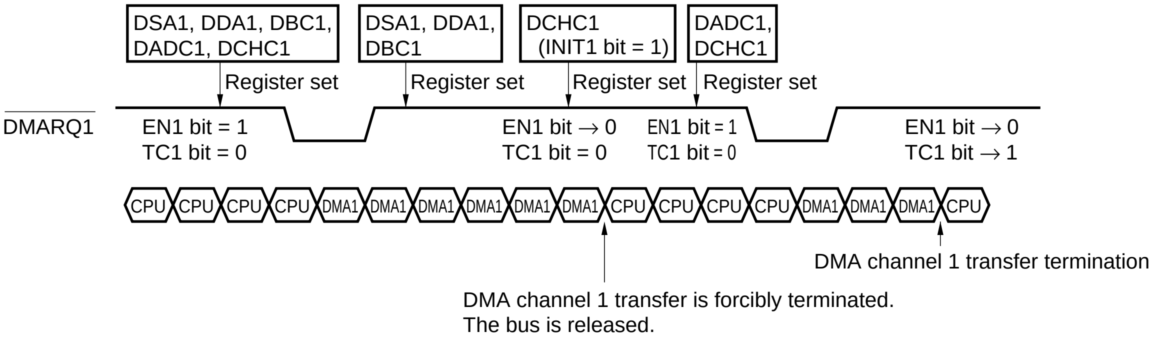
In addition to forcible interruption of DMA transfer by NMI input, DMA transfer can also be terminated forcibly by the INITn bit of the DCHCn register. Examples of the forcible termination operation are shown below (n = 0 to 3).

Figure 6-11. Example of Forcible Termination of DMA Transfer

(a) During block transfer through DMA channel 2, transfer through DMA channel 3 is started.



(b) During block transfer through DMA channel 1, transfer is terminated, and a different conditional transfer is executed.



Remark During DMA transfer, the next condition can be set, because the DSAn, DDAn, DBCn registers are buffered registers, but the setting to the DADCn register is ignored (refer to **6.9 Next Address Setting Function**).

6.13 Boundary of Memory Area

The transfer operation is not guaranteed if the source or the destination address is over the area of DMA objects (external memory, internal RAM, external I/O, or internal peripheral I/O) during DMA transfer.

6.14 Transfer of Misalign Data

16-bit DMA transfer of misalign data is not supported. If the source or the destination address is set to an odd address, the LSB bit of the address is forcibly accepted as “0”.

6.15 Clocks of DMA Transfer

Table 6-3 lists the overhead before and after DMA transfer and minimum execution clock for DMA transfer.

Table 6-3. Minimum Execution Clock in DMA Cycle

From accepting $\overline{\text{DMARQn}}$ to falling edge of $\overline{\text{DMAAKn}}$	4 clocks
External memory access	Refer to miscellaneous memory and I/O cycle
Internal RAM access	2 clocks
Internal peripheral I/O access	3 clocks
From rising edge of $\overline{\text{DMAAKn}}$ to falling edge of $\overline{\text{TCn}}$	1 clock

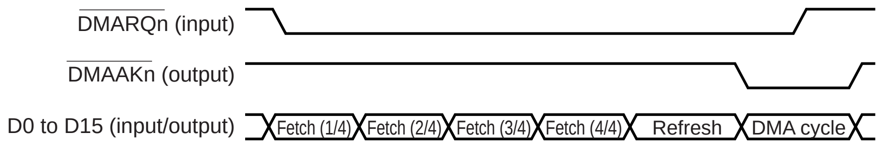
Remark n = 0 to 3

6.16 Maximum Response Time to DMA Request

Under the conditions shown below, the response time to a DMA request becomes the maximum time (this is the state permitted by the DRAM refresh cycle).

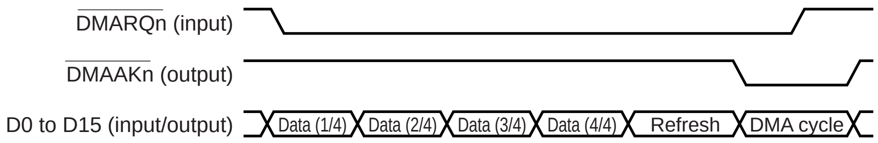
(1) Condition 1

Condition	Instruction fetch from external memory at the 8-bit data bus width
Response time	Tinst × 4 + Tref



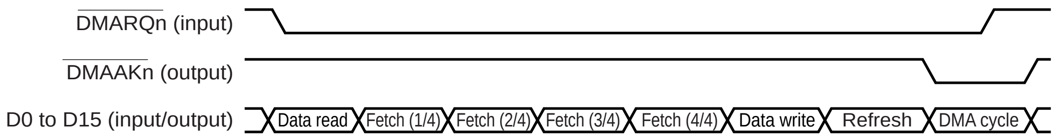
(2) Condition 2

Condition	Word data access with external memory at the 8-bit data bus width
Response time	Tdata × 4 + Tref



(3) Condition 3

Condition	Instruction fetch from external memory at the 8-bit data bus width. Execution of the bit manipulation instruction (SET1, CLR1, NOT1).
Response time	Tinst × 4 + Tdata × 2 + Tref



Remarks 1. Tinst: The number of clocks per bus cycle during instruction fetch.
Tdata: The number of clocks per bus cycle during data access.
Tref: The number of clocks per refresh cycle.

2. n = 0 to 3

6.17 One Time Single Transfer with $\overline{\text{DMARQ0}}$ to $\overline{\text{DMARQ3}}$

To execute one time single transfer to external memory via $\overline{\text{DMARQn}}$ signal input, $\overline{\text{DMARQn}}$ should be inactive within the clock time shown in Table 6-4 from when $\overline{\text{DMAAKn}}$ becomes active (n = 0 to 3). If $\overline{\text{DMARQn}}$ is active for more than the clock time shown in Table 6-4, single transfers are continuously executed.

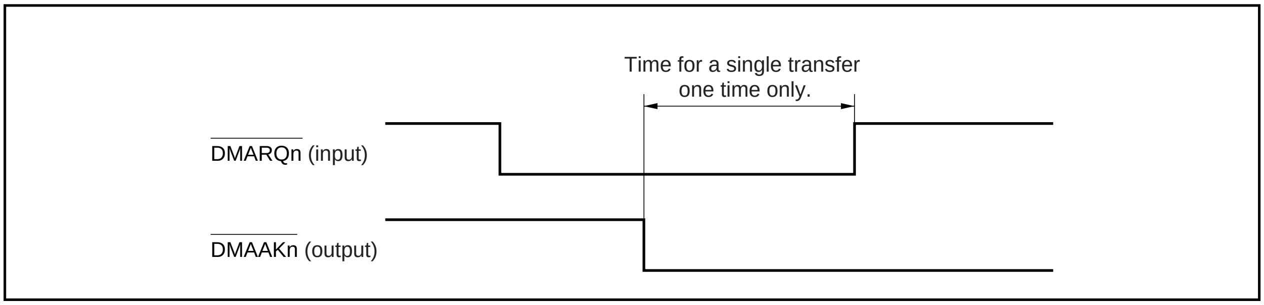

Time for a single transfer
one time only.
$\overline{\text{DMARQn}}$ (input)
$\overline{\text{DMAAKn}}$ (output)

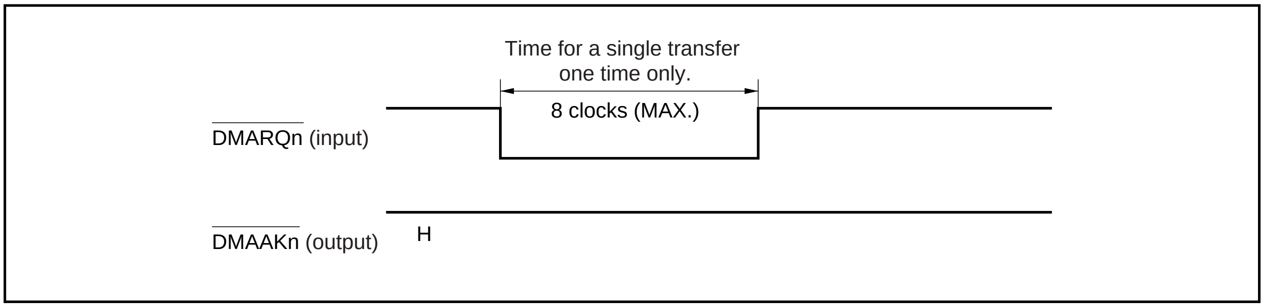
Table 6-4. $\overline{\text{DMAAKn}}$ Active → $\overline{\text{DMARQn}}$ Inactive Time for Single Transfer to External Memory

Transfer Type	Source	Destination	$\overline{\text{DMAAKn}}$ Signal Active → $\overline{\text{DMARQn}}$ Inactive Time (Max.)[Note]
Two-cycle transfer	DRAM (off page)	All objects	5 clocks
	DRAM (on page)	All objects	4 clocks
	SRAM or external I/O	All objects	4 clocks
	Internal RAM or internal peripheral I/O	DRAM (off page)	7 clocks
	Internal RAM or internal peripheral I/O	DRAM (on page)	6 clocks
	Internal RAM	SRAM or external I/O	6 clocks
	Internal peripheral I/O	SRAM	6 clocks
Flyby transfer	DRAM (off page) ↔ External I/O		3 clocks
	DRAM (on page) ↔ External I/O		2 clocks
	SRAM ↔ External I/O		2 clocks

Note When inserting waits, add the number of waits together.

Remark n = 0 to 3

Also, if a single transfer is executed between internal RAM and internal peripheral I/O, it is necessary that the $\overline{\text{DMARQn}}$ signal be inactivated within 8 clock cycles after it is activated. If 8 clock cycles are exceeded, transfer may continue. Note that the $\overline{\text{DMAAKn}}$ signal does not become active at this time.


Time for a single transfer one time only.
8 clocks (MAX.)
$\overline{\text{DMARQn}}$ (input)
$\overline{\text{DMAAKn}}$ (output)
H

6.18 Bus Arbitration for CPU

The CPU can access any external memory, external I/O, internal RAM, and internal peripheral I/O not undergoing DMA transfer.

While data is being transferred between external memory and external I/O, the CPU can access internal RAM and internal peripheral I/O.

While data transfer is being executed between internal RAM and internal peripheral I/O, the CPU can access external memory and external I/O.

6.19 Precaution

If a DMA transfer which satisfies all the following conditions is interrupted by NMI input, the $\overline{\text{DMAAKn}}$ signal may become active and remain so until the next DMA transfer (n = 0 to 3).

- Two-cycle transfer
- Block transfer mode
- Transfer from external memory to external memory, or from external I/O to external I/O
- The destination side is EDO DRAM, with no-wait on-page access.

Note that device operations other than the $\overline{\text{DMAAKn}}$ signal are not influenced.

Change the $\overline{\text{DMAAKn}}$ signal to inactive by executing the routine shown below in the NMI handler, etc.

```
LD.B   DDIS[r0], reg  ; Confirm the interrupted DMA channel by NMI input.
ST.B   reg, DRST[r0]  ; Restart transfer in the interrupted channel.
ST.B   r0, DRST[r0]   ; By immediately interrupting transfer again, after DMA transfer only once, the DMAAKn
                        signal becomes inactive.
```

[MEMO]

CHAPTER 7 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V850E/MS1 is provided with a dedicated interrupt controller (INTC) for interrupt processing and can process a total of 48 interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event that is dependent on program execution. Generally, an exception takes precedence over an interrupt.

The V850E/MS1 can process interrupt requests from the internal peripheral hardware and external sources. Moreover, exception processing can be started by the TRAP instruction (software exception) or by the generation of an exception event (fetching of an illegal op code), which is known as an exception trap.

7.1 Features

○ Interrupts
- Non-maskable interrupts: 1 source
- Maskable interrupts: 47 sources
- 8 levels of programmable priorities
- Mask specification for interrupt requests according to priority
- Mask can be specified for each maskable interrupt request.
- Noise elimination, edge detection, and valid edge of external interrupt request signal can be specified.

○ Exceptions
- Software exceptions: 32 sources
- Exception trap: 1 source (illegal op code exception)

Interrupt/exception sources are listed in Table 7-1.

Table 7-1. Interrupt List (1/3)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Controlling Register	Source	Generating Unit				
Reset	Interrupt	RESET	—	$\overline{\text{RESET}}$ input	Pin	—	0000H	00000000H	Undefined
Non-maskable	Interrupt	NMI	—	NMI input	Pin	—	0010H	00000010H	nextPC
Software exception	Exception	TRAP0[Note]	—	TRAP instruction	—	—	004n[Note]H	00000040H	nextPC
	Exception	TRAP1n[Note]	—	TRAP instruction	—	—	005n[Note]H	00000050H	nextPC
Exception trap	Exception	ILGOP	—	Illegal op code	—	—	0060H	00000060H	nextPC
Maskable	Interrupt	INTOV10	OVIC10	Timer 10 overflow	RPU	0	0080H	00000080H	nextPC
	Interrupt	INTOV11	OVIC11	Timer 11 overflow	RPU	1	0090H	00000090H	nextPC
	Interrupt	INTOV12	OVIC12	Timer 12 overflow	RPU	2	00A0H	000000A0H	nextPC
	Interrupt	INTOV13	OVIC13	Timer 13 overflow	RPU	3	00B0H	000000B0H	nextPC
	Interrupt	INTOV14	OVIC14	Timer 14 overflow	RPU	4	00C0H	000000C0H	nextPC
	Interrupt	INTOV15	OVIC15	Timer 15 overflow	RPU	5	00D0H	000000D0H	nextPC
	Interrupt	INTP100/ INTCC100	P10IC0	Match of INTP100 pin/CC100	Pin/RPU	6	0100H	00000100H	nextPC
	Interrupt	INTP101/ INTCC101	P10IC1	Match of INTP101 pin/CC101	Pin/RPU	7	0110H	00000110H	nextPC
	Interrupt	INTP102/ INTCC102	P10IC2	Match of INTP102 pin/CC102	Pin/RPU	8	0120H	00000120H	nextPC
	Interrupt	INTP103/ INTCC103	P10IC3	Match of INTP103 pin/CC103	Pin/RPU	9	0130H	00000130H	nextPC
	Interrupt	INTP110/ INTCC110	P11IC0	Match of INTP110 pin/CC110	Pin/RPU	10	0140H	00000140H	nextPC
	Interrupt	INTP111/ INTCC111	P11IC1	Match of INTP111 pin/CC111	Pin/RPU	11	0150H	00000150H	nextPC
	Interrupt	INTP112/ INTCC112	P11IC2	Match of INTP112 pin/CC112	Pin/RPU	12	0160H	00000160H	nextPC
	Interrupt	INTP113/ INTCC113	P11IC3	Match of INTP113 pin/CC113	Pin/RPU	13	0170H	00000170H	nextPC
	Interrupt	INTP120/ INTCC120	P12IC0	Match of INTP120 pin/CC120	Pin/RPU	14	0180H	00000180H	nextPC
	Interrupt	INTP121/ INTCC121	P12IC1	Match of INTP121 pin/CC121	Pin/RPU	15	0190H	00000190H	nextPC
	Interrupt	INTP122/ INTCC122	P12IC2	Match of INTP122 pin/CC122	Pin/RPU	16	01A0H	000001A0H	nextPC
	Interrupt	INTP123/ INTCC123	P12IC3	Match of INTP123 pin/CC123	Pin/RPU	17	01B0H	000001B0H	nextPC
	Interrupt	INTP130/ INTCC130	P13IC0	Match of INTP130 pin/CC130	Pin/RPU	18	01C0H	000001C0H	nextPC
	Interrupt	INTP131/ INTCC131	P13IC1	Match of INTP131 pin /CC131	Pin/RPU	19	01D0H	000001D0H	nextPC

Note n = 0 to FH

Table 7-1. Interrupt List (2/3)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Controlling Register	Source	Generating Unit				
Maskable	Interrupt	INTP132/ INTCC132	P13IC2	Match of INTP132 pin/CC132	Pin/RPU	20	01E0H	000001E0H	nextPC
	Interrupt	INTP133/ INTCC133	P13IC3	Match of INTP133 pin/CC133	Pin/RPU	21	01F0H	000001F0H	nextPC
	Interrupt	INTP140/ INTCC140	P14IC0	Match of INTP140 pin/CC140	Pin/RPU	22	0200H	00000200H	nextPC
	Interrupt	INTP141/ INTCC141	P14IC1	Match of INTP141 pin/CC141	Pin/RPU	23	0210H	00000210H	nextPC
	Interrupt	INTP142/ INTCC142	P14IC2	Match of INTP142 pin/CC142	Pin/RPU	24	0220H	00000220H	nextPC
	Interrupt	INTP143/ INTCC143	P14IC3	Match of INTP143 pin/CC143	Pin/RPU	25	0230H	00000230H	nextPC
	Interrupt	INTP150/ INTCC150	P15IC0	Match of INTP150 pin/CC150	Pin/RPU	26	0240H	00000240H	nextPC
	Interrupt	INTP151/ INTCC151	P15IC1	Match of INTP151 pin/CC151	Pin/RPU	27	0250H	00000250H	nextPC
	Interrupt	INTP152/ INTCC152	P15IC2	Match of INTP152 pin/CC152	Pin/RPU	28	0260H	00000260H	nextPC
	Interrupt	INTP153/ INTCC153	P15IC3	Match of INTP153 pin/CC153	Pin/RPU	29	0270H	00000270H	nextPC
	Interrupt	INTCM40	CMIC40	CM40 match signal	RPU	30	0280H	00000280H	nextPC
	Interrupt	INTCM41	CMIC41	CM41 match signal	RPU	31	0290H	00000290H	nextPC
	Interrupt	INTDMA0	DMAIC0	DMA channel 0 transfer completion	DMAC	32	02A0H	000002A0H	nextPC
	Interrupt	INTDMA1	DMAIC1	DMA channel 1 transfer completion	DMAC	33	02B0H	000002B0H	nextPC
	Interrupt	INTDMA2	DMAIC2	DMA channel 2 transfer completion	DMAC	34	02C0H	000002C0H	nextPC
	Interrupt	INTDMA3	DMAIC3	DMA channel 3 transfer completion	DMAC	35	02D0H	000002D0H	nextPC
	Interrupt	INTCSI0	CSIC0	CSI0 transmission/ reception completion	SIO	36	0300H	00000300H	nextPC
	Interrupt	INTSER0	SEIC0	UART0 reception error	SIO	37	0310H	00000310H	nextPC
	Interrupt	INTSR0	SRIC0	UART0 reception completion	SIO	38	0320H	00000320H	nextPC
	Interrupt	INTST0	STIC0	UART0 transmission completion	SIO	39	0330H	00000330H	nextPC

Table 7-1. Interrupt List (3/3)

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Handler Address	Restored PC
		Name	Controlling Register	Source	Generating Unit				
Maskable	Interrupt	INTCSI1	CSIC1	CSI1 transmission/ reception completion	SIO	40	0340H	00000340H	nextPC
	Interrupt	INTSER1	SEIC1	UART1 reception error	SIO	41	0350H	00000350H	nextPC
	Interrupt	INTSR1	SRIC1	UART1 reception completion	SIO	42	0360H	00000360H	nextPC
	Interrupt	INTST1	STIC1	UART1 transmission completion	SIO	43	0370H	00000370H	nextPC
	Interrupt	INTCSI2	CSIC2	CSI2 transmission/ reception completion	SIO	44	0380H	00000380H	nextPC
	Interrupt	INTCSI3	CSIC3	CSI3 transmission/ reception completion	SIO	45	03C0H	000003C0H	nextPC
	Interrupt	INTAD	ADIC	A/D conversion completion	ADC	46	0400H	00000400H	nextPC

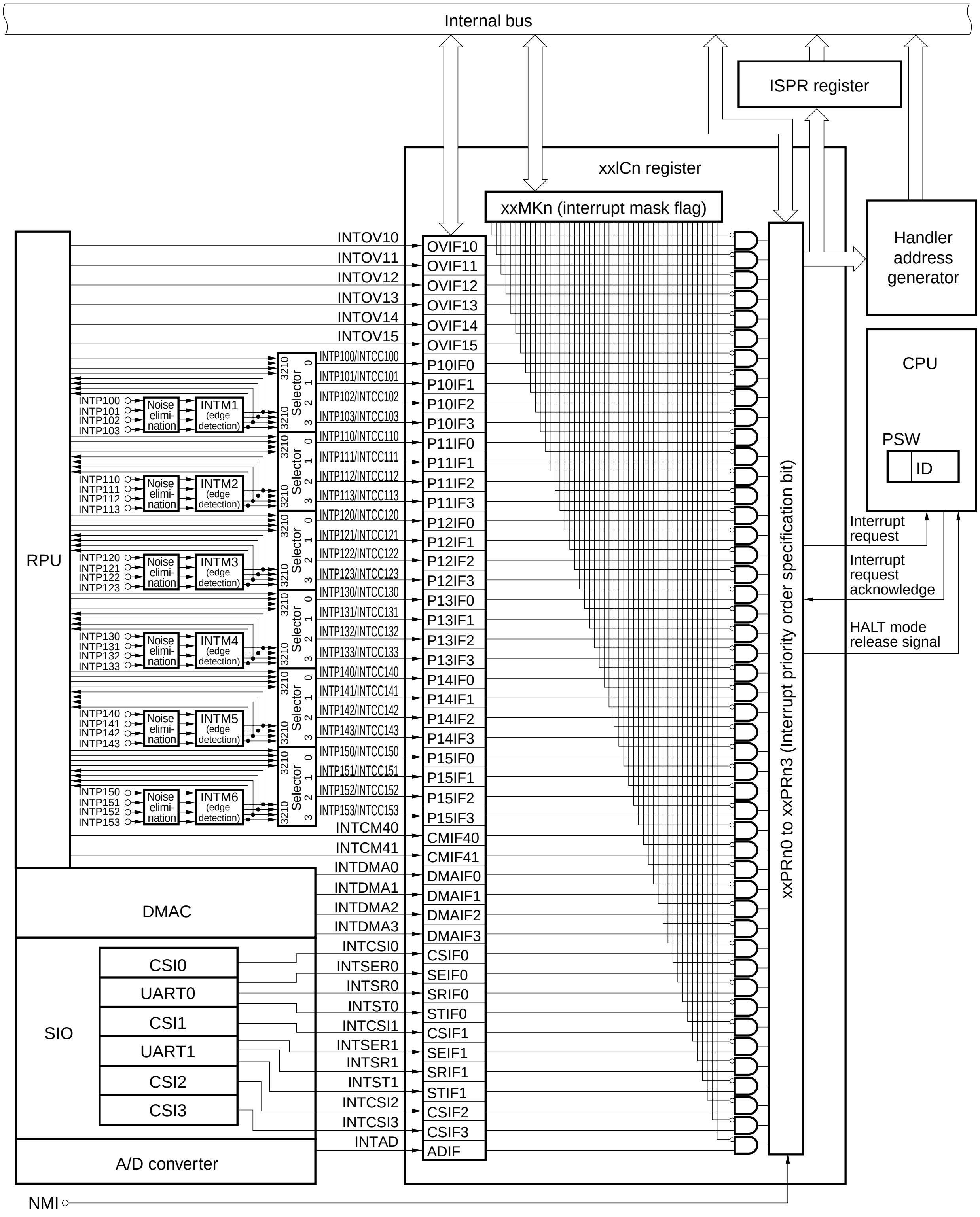
Caution INTP1mn (external interrupt) and INTCC1mn (compare register match interrupt) share a control register (m = 0 to 5, n = 0 to 3). Set the valid interrupt request using bits 3 to 0 (IMS1mn) of timer unit mode registers 10 to 15 (TUM10 to TUM15) (see 9.3 (1) Timer unit mode registers 10 to 15 (TUM10 to TUM15)).

Remarks 1. Default priority: The priority order when two or more maskable interrupt requests occur at the same time. The highest priority is 0.

Restored PC: The value of the PC saved to EIPC or FEPC when interrupt/exception processing is started. However, the value of the PC, which is saved when an interrupt is acknowledged during division (DIV, DIVH, DIVU, and DIVHU) instruction execution, is the value of the PC of the current instruction (DIV, DIVH, DIVU, and DIVHU).

2. The execution address of the illegal instruction when an illegal op code exception occurs is d

Figure 7-1. Block Diagram of Interrupt Control Function



Remark xx: OV, CM, P10 to P15, DMA, CS, SE, SR, ST, AD

n: None, or 10 to 15, 40, 41, 0 to 3

7.2 Non-Maskable Interrupt

A non-maskable interrupt request is acknowledged unconditionally, even when interrupts are in the interrupt disabled (DI) status. An NMI is not subject to priority control and takes precedence over all other interrupts.

A non-maskable interrupt request is input from the NMI pin. When the valid edge specified by bit 0 (ESN0) of the external interrupt mode register 0 (INTM0) is detected on the NMI pin, the interrupt occurs.

While the service program of the non-maskable interrupt is being executed (PSW.NP = 1), the acknowledgement of another non-maskable interrupt requests is held pending. The pending NMI is acknowledged after the original service program of the non-maskable interrupt under execution has been terminated (by the RETI instruction), or when PSW.NP is cleared to 0 by the LDSR instruction. Note that if two or more NMI requests are input during the execution of the service program for an NMI, the number of NMIs that will be acknowledged after PSW.NP goes to ‘‘0’’, is only one.

Remark PSW.NP: The NP bit of the PSW register.

7.2.1 Operation

If a non-maskable interrupt is generated, the CPU performs the following processing, and transfers control to the handler routine:

(1) Saves the restored PC to FEPC.
(2) Saves the current PSW to FEPSW.
(3) Writes the exception code (0010H) to the higher halfword (FECC) of ECR.
(4) Sets the NP and ID bits of PSW and clears the EP bit.
(5) Sets the handler address (00000010H) corresponding to the non-maskable interrupt to the PC, and transfers control.

The processing configuration of a non-maskable interrupt is shown in Figure 7-2.

Figure 7-2. Processing Configuration of Non-Maskable Interrupt

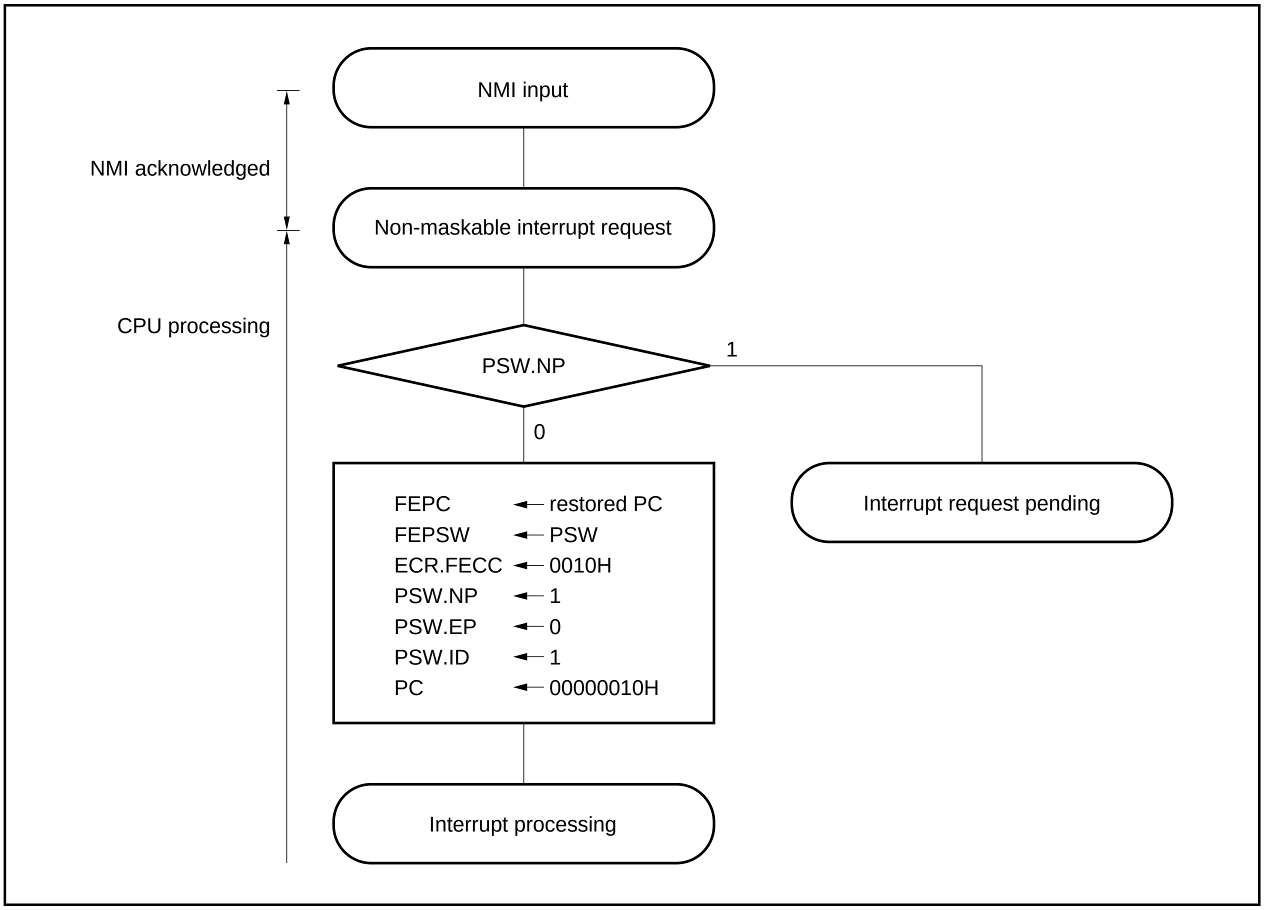
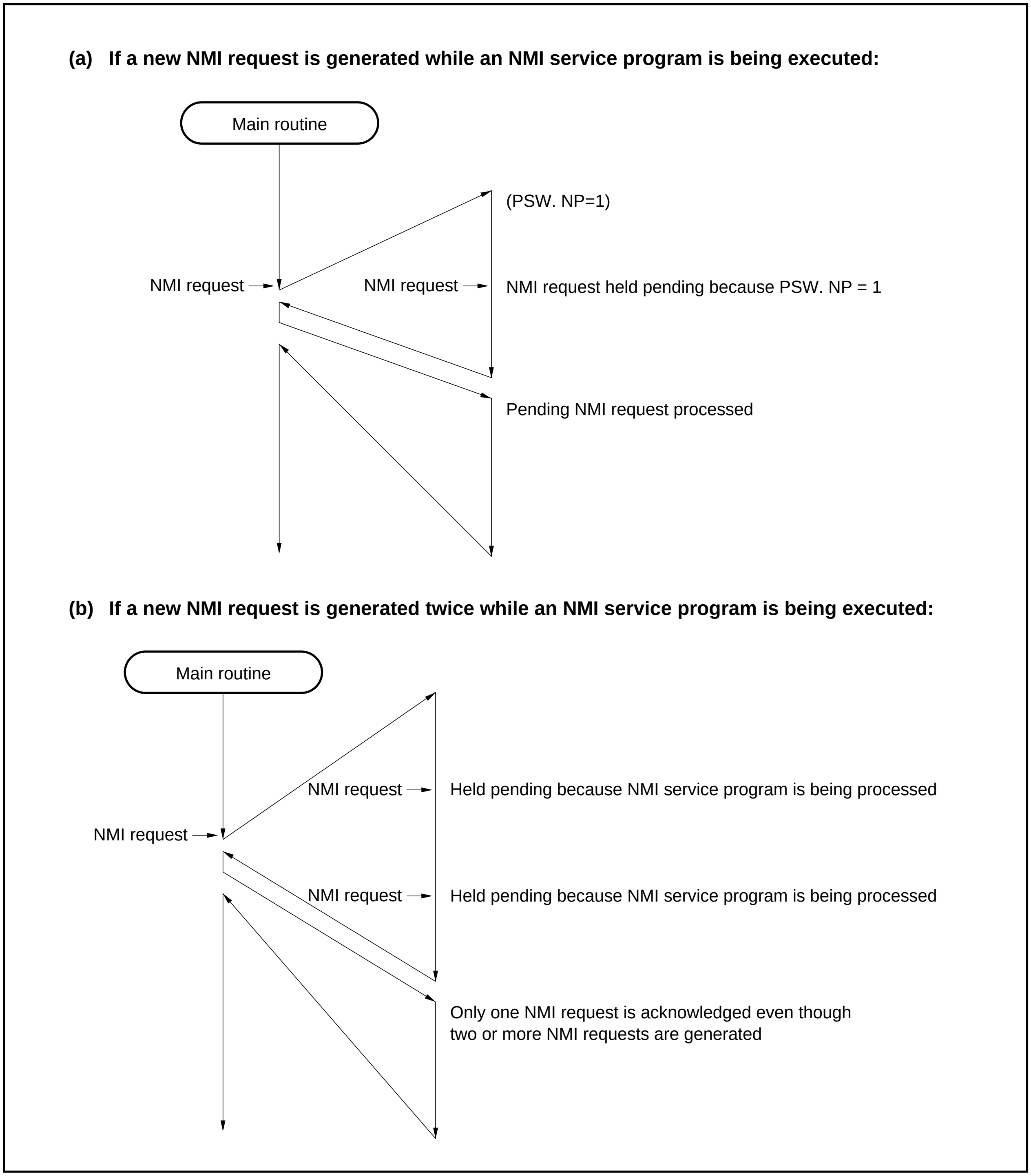


Figure 7-3. Acknowledging Non-Maskable Interrupt Request



(a) If a new NMI request is generated while an NMI service program is being executed:

Main routine
(PSW. NP=1)
NMI request
NMI request
NMI request held pending because PSW. NP = 1
Pending NMI request processed

(b) If a new NMI request is generated twice while an NMI service program is being executed:

Main routine
NMI request
Held pending because NMI service program is being processed
NMI request
NMI request
Held pending because NMI service program is being processed
Only one NMI request is acknowledged even though two or more NMI requests are generated

7.2.2 Restore

Execution is restored from the non-maskable interrupt processing by the RETI instruction.

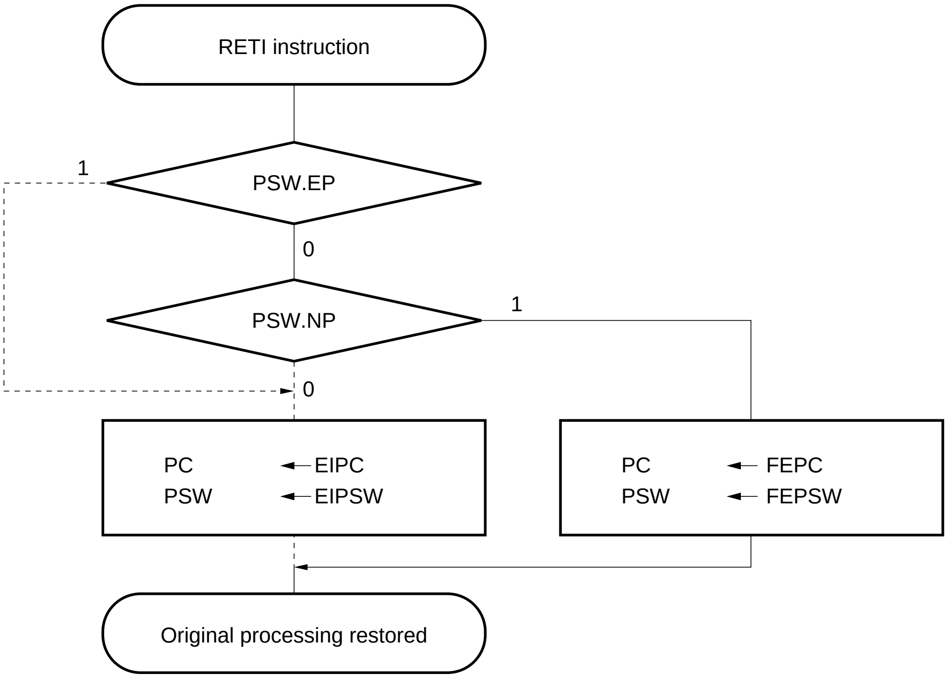
When the RETI instruction is executed, the CPU performs the following processing, and transfers control to the address of the restored PC.

(1) Restores the values of the PC and PSW from FEPC and FEPSW, respectively, because the EP bit of PSW is 0 and the NP bit of PSW is 1.

(2) Transfers control back to the address of the restored PC and PSW.

Figure 7-4 illustrates how the RETI instruction is processed.

Figure 7-4. RETI Instruction Processing



Caution When the PSW.EP bit and PSW.NP bit are changed by the LDSR instruction during non-maskable interrupt processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 1 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

7.2.3 Non-maskable interrupt status flag (NP)

The NP flag is bit 7 of the PSW.

The NP flag is a status flag that indicates that non-maskable interrupt (NMI) processing is under execution. This flag is set when the NMI interrupt has been acknowledged, and masks all interrupt requests and exceptions to prohibit multiple interrupts from being acknowledged.

	31 … 8	7	6	5	4	3	2	1	0	
PSW	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	NP	EP	ID	SAT	CY	OV	S	Z	After reset 00000020H

Bit Position	Bit Name	Function
7	NP	NMI Pending Indicates that NMI interrupt processing is in progress. 0: No NMI interrupt processing 1: NMI interrupt currently being processed

7.2.4 Noise elimination

NMI pin noise is eliminated with analog delay. The delay time is 60 to 220 ns. The signal input that changes within the delay time is not internally acknowledged.

The NMI pin is used for releasing the software STOP mode. In the software STOP mode, the internal system clock is not used for noise elimination because the internal system clock is stopped.

7.2.5 Edge detection function

INTM0 is a register that specifies the valid edge of the non-maskable interrupt (NMI). The NMI valid edge can be specified to be either the rising edge or the falling edge by the ESN0 bit.

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
INTM0	0	0	0	0	0	0	0	ESN0	Address FFFFF180H	After reset 00H

Bit Position	Bit Name	Function
0	ESN0	Edge Select NMI Specifies the NMI pin’s valid edge. 0: Falling edge 1: Rising edge

7.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V850E/MS1 has 47 maskable interrupt sources.

If two or more maskable interrupt requests are generated at the same time, they are acknowledged according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers (programmable priority control).

When an interrupt request has been acknowledged, the acknowledgement of other maskable interrupt requests is disabled and the interrupt disabled (DI) status is set.

When the EI instruction is executed in an interrupt processing routine, the interrupt enabled (EI) status is set which enables interrupts having a higher priority than the interrupt requests in progress (specified by the interrupt control register). Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be nested.

However, if multiplexed interrupts are executed, the following processing is necessary.

<1> Save EIPC and EIPSW in memory or a general-purpose register before executing the EI instruction.

<2> Execute the DI instruction before executing the RETI instruction, then reset EIPC and EIPSW with the values saved in <1>.

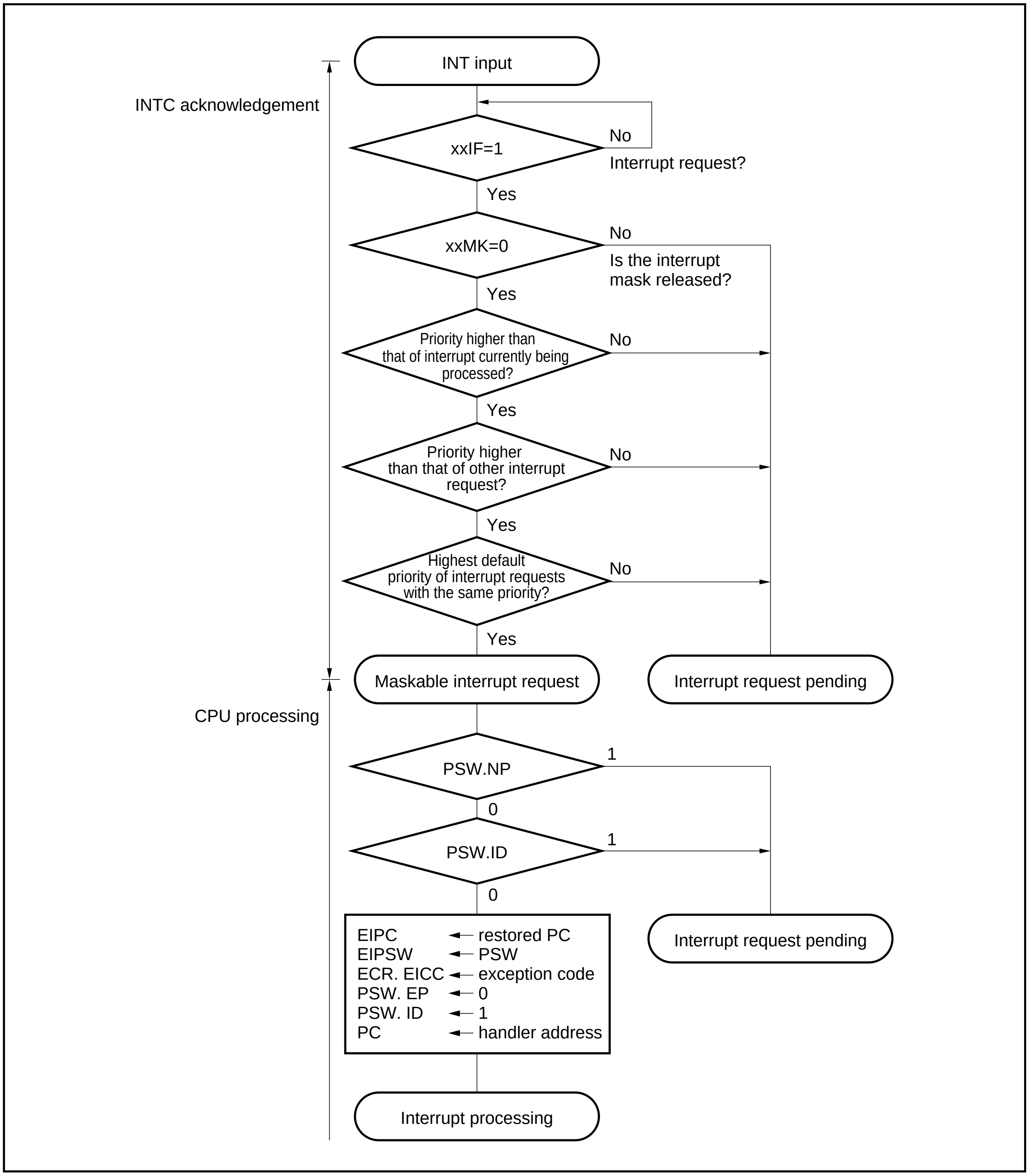
7.3.1 Operation

If a maskable interrupt occurs by INT input, the CPU performs the following processing, and transfers control to a handler routine:

(1) Saves the restored PC to EIPC.
(2) Saves the current PSW to EIPSW.
(3) Writes an exception code to the lower halfword of ECR (EICC).
(4) Sets the ID bit of the PSW and clears the EP bit.
(5) Sets the handler address corresponding to each interrupt to the PC, and transfers control.

The processing configuration of a maskable interrupt is shown in Figure 7-5.

Figure 7-5. Maskable Interrupt Processing


INT input
INTC acknowledgement
xxIF=1
No
Interrupt request?
Yes
xxMK=0
No
Is the interrupt mask released?
Yes
Priority higher than that of interrupt currently being processed?
No
Yes
Priority higher than that of other interrupt request?
No
Yes
Highest default priority of interrupt requests with the same priority?
No
Yes
Maskable interrupt request
Interrupt request pending
CPU processing
PSW.NP
1
0
PSW.ID
1
0
EIPC ← restored PC
EIPSW ← PSW
ECR. EICC ← exception code
PSW. EP ← 0
PSW. ID ← 1
PC ← handler address
Interrupt request pending
Interrupt processing

The INT input masked by the interrupt controllers and the INT input that occurs while another interrupt is being processed (when PSW.NP = 1 or PSW.ID = 1) are held pending internally by the interrupt controller. When the interrupts are unmasked, or when PSW.NP = 0 and PSW.ID = 0 are set by the RETI and LDSR instructions, input of the pending INT starts the new maskable interrupt processing.

7.3.2 Restore

To restore from the maskable interrupt processing, the RETI instruction is used.

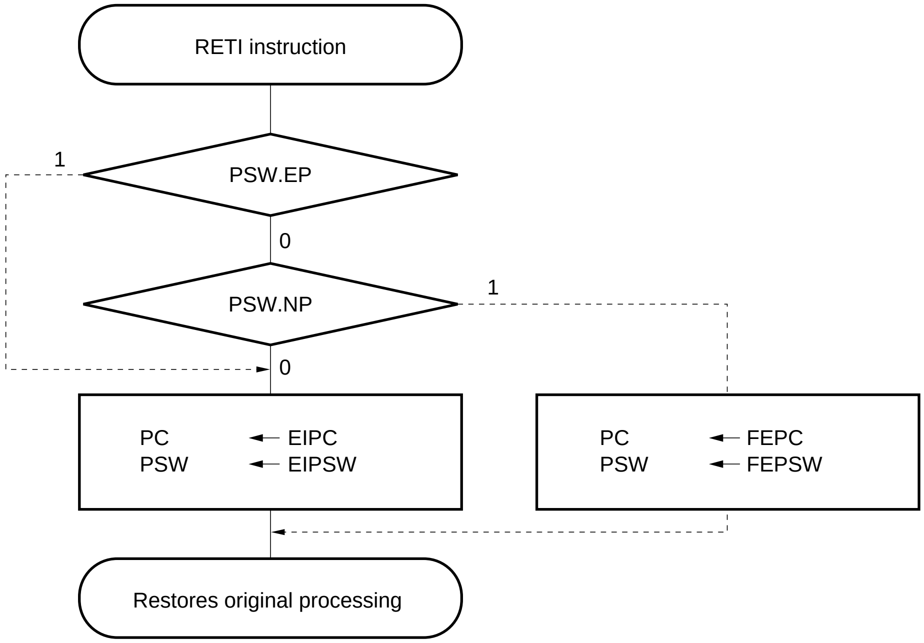
When the RETI instruction is executed, the CPU performs the following steps, and transfers control to the address of the restored PC.

(1) Restores the values of the PC and PSW from EIPC and EIPSW because the EP bit of the PSW is 0 and the NP bit of the PSW is 0.

(2) Transfers control to the address of the restored PC and PSW.

Figure 7-6 illustrates the processing of the RETI instruction.

Figure 7-6. RETI Instruction Processing



Caution When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during maskable interrupt processing, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 0 and PSW.NP back to 0 using the LDSR instruction immediately before the RETI instruction.

Remark The solid line shows the CPU processing flow.

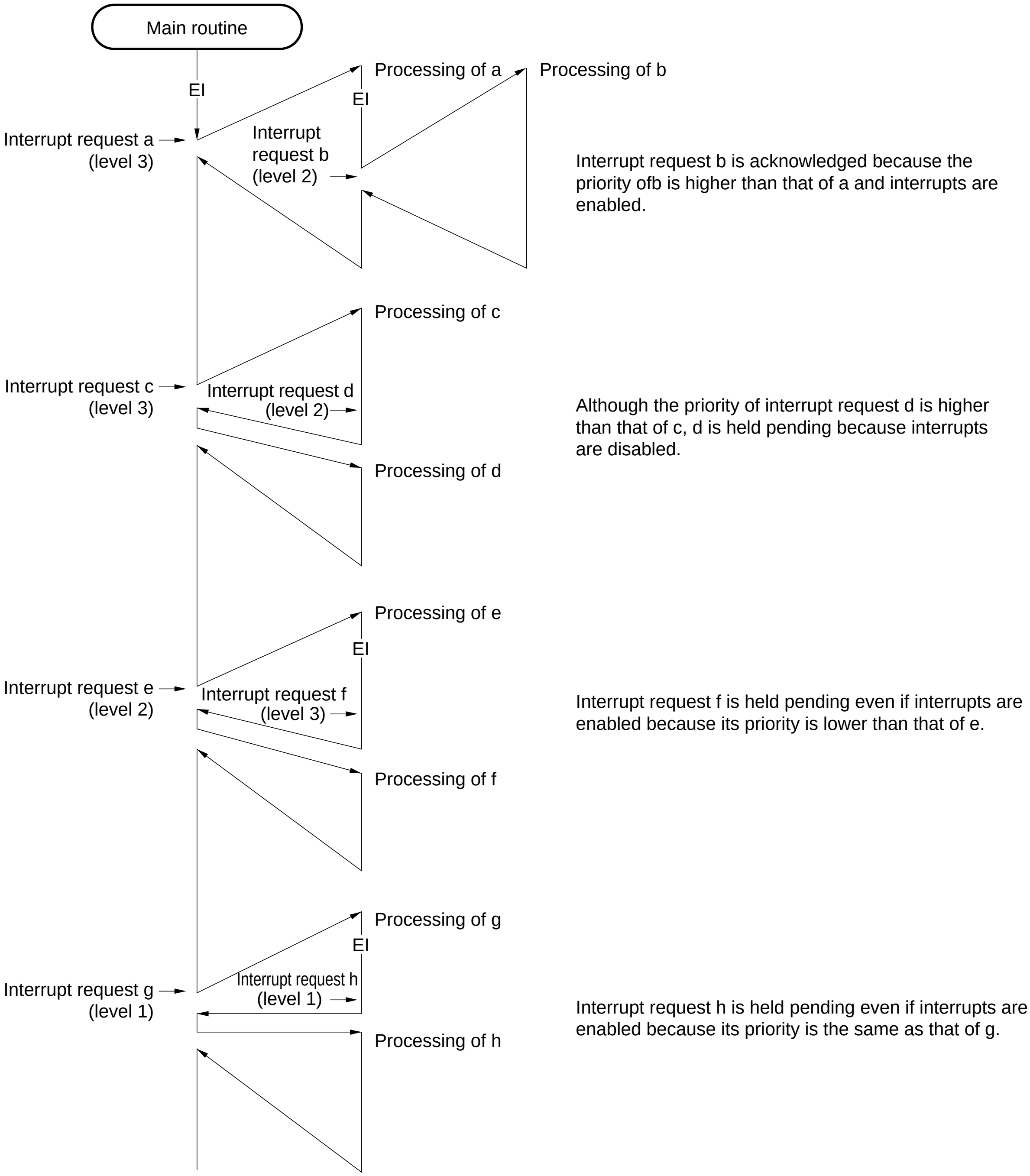
7.3.3 Priorities of maskable interrupts

The V850E/MS1 provides multiple interrupt servicing whereby an interrupt is acknowledged while another interrupt is being serviced. Multiple interrupts can be controlled by priority levels.

There are two types of priority level control: control based on the default priority levels, and control based on the programmable priority levels which are specified by the interrupt priority level specification bit (xxPRn) of the interrupt control register (xxICn). When two or more interrupts having the same priority level specified by the xxPRn bit are generated at the same time, interrupts are serviced in order depending on the priority level allocated to each interrupt request type (default priority level) beforehand. For more information, refer to Table 7-1. The programmable priority control customizes interrupt requests into eight levels by setting the priority level specification flag.

Note that when an interrupt request is acknowledged, the ID flag of the PSW is automatically set to 1. Therefore, when multiple interrupts are to be used, clear the ID flag to 0 beforehand (for example, by placing the EI instruction into the interrupt service program) to set the interrupt enable mode.

Figure 7-7. Example of Processing in Which Another Interrupt Request Is Issued While Interrupt Is Being Processed (1/2)

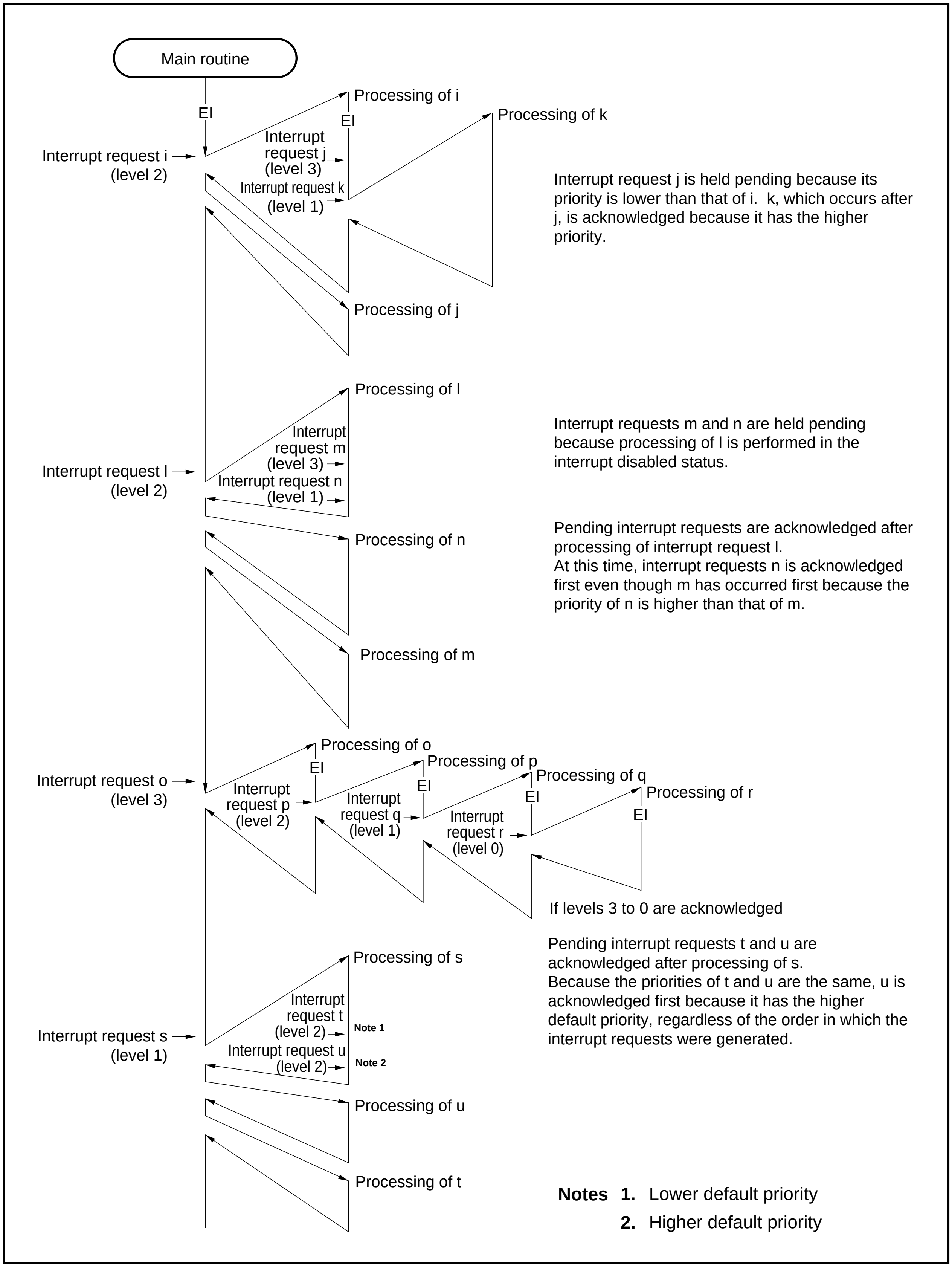

Main routine
EI
Interrupt request a
(level 3)
Processing of a
EI
Interrupt
request b
(level 2)
Processing of b
Interrupt request b is acknowledged because the priority ofb is higher than that of a and interrupts are enabled.
Processing of c
Interrupt request c
(level 3)
Interrupt request d
(level 2)
Although the priority of interrupt request d is higher than that of c, d is held pending because interrupts are disabled.
Processing of d
Processing of e
EI
Interrupt request e
(level 2)
Interrupt request f
(level 3)
Interrupt request f is held pending even if interrupts are enabled because its priority is lower than that of e.
Processing of f
Processing of g
EI
Interrupt request g
(level 1)
Interrupt request h
(level 1)
Interrupt request h is held pending even if interrupts are enabled because its priority is the same as that of g.
Processing of h

Remarks **1.** a to u in the figure are the names of interrupt requests shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt requests.

Caution The values of the EIPC and EIPSW registers must be saved before executing multiple interrupts.

Figure 7-7. Example of Processing in Which Another Interrupt Request Is Issued While Interrupt Is Being Processed (2/2)


Main routine
EI
Interrupt request i (level 2)
Processing of i
EI
Interrupt request j (level 3)
Interrupt request k (level 1)
Processing of k
Processing of j
Processing of l
Interrupt request l (level 2)
Interrupt request m (level 3)
Interrupt request n (level 1)
Processing of n
Processing of m
Interrupt request o (level 3)
Processing of o
EI
Interrupt request p (level 2)
Processing of p
EI
Interrupt request q (level 1)
Processing of q
EI
Interrupt request r (level 0)
Processing of r
EI
Interrupt request s (level 1)
Processing of s
Interrupt request t (level 2)
Note 1
Interrupt request u (level 2)
Note 2
Processing of u
Processing of t

Interrupt request j is held pending because its priority is lower than that of i. k, which occurs after j, is acknowledged because it has the higher priority.

Interrupt requests m and n are held pending because processing of l is performed in the interrupt disabled status.

Pending interrupt requests are acknowledged after processing of interrupt request l.
At this time, interrupt requests n is acknowledged first even though m has occurred first because the priority of n is higher than that of m.

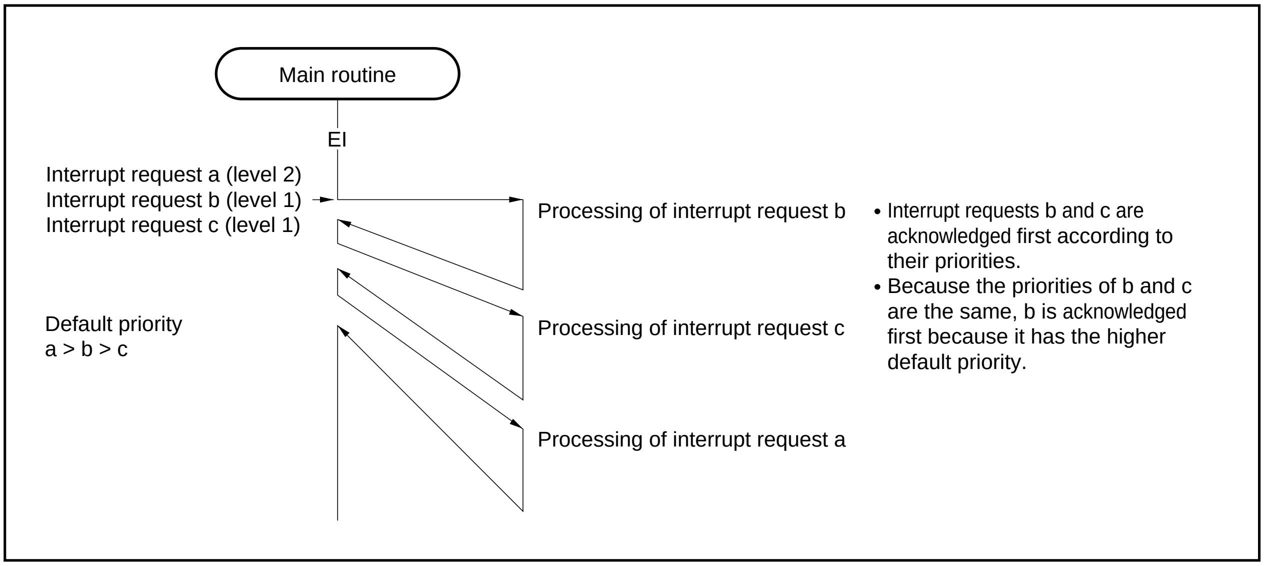
If levels 3 to 0 are acknowledged

Pending interrupt requests t and u are acknowledged after processing of s.
Because the priorities of t and u are the same, u is acknowledged first because it has the higher default priority, regardless of the order in which the interrupt requests were generated.

Notes 1. Lower default priority

2. Higher default priority

Figure 7-8. Example of Processing Interrupt Requests Simultaneously Generated


Main routine
EI
Interrupt request a (level 2)
Interrupt request b (level 1)
Interrupt request c (level 1)
Default priority
a > b > c
Processing of interrupt request b
Processing of interrupt request c
Processing of interrupt request a
• Interrupt requests b and c are acknowledged first according to their priorities.
• Because the priorities of b and c are the same, b is acknowledged first because it has the higher default priority.

7.3.4 Interrupt control register (xxICn)

An interrupt control register is assigned to each interrupt request (maskable interrupt) and sets the control conditions for each maskable interrupt request.

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
xxICn	xxIFn	xxMKn	0	0	0	xxPRn2	xxPRn1	xxPRn0	Address FFFFF100H to FFFFF15CH	After reset 47H

Bit Position	Bit Name	Function
7	xxIFn	Interrupt Request Flag This is an interrupt request flag. 0: Interrupt request not issued 1: Interrupt request issued The flag xxlFn is reset automatically by the hardware if an interrupt request is received.
6	xxMKn	Mask Flag This is an interrupt mask flag. 0: Enables interrupt processing 1: Disables interrupt processing (pending)
2 to 0	xxPRn2 to xxPRn0	Priority 8 levels of priority order are specified in each interrupt.

xxPRn2	xxPRn1	xxPRn0	Interrupt Priority Specification Bit
0	0	0	Specifies level 0 (highest).
0	0	1	Specifies level 1.
0	1	0	Specifies level 2.
0	1	1	Specifies level 3.
1	0	0	Specifies level 4.
1	0	1	Specifies level 5.
1	1	0	Specifies level 6.
1	1	1	Specifies level 7 (lowest).

Remark xx: Identification name of each peripheral unit (OV, P10 to P15, CM, CS, SE, SR, ST, AD, DMA)
n: Peripheral unit number (None, or 0 to 3, 10 to 15, 40, 41).

Address and bit of each interrupt control register is as follows:

Table 7-2. Interrupt Control Register Addresses and Bits (1/2)

Address	Register	Bit							
		7	6	5	4	3	2	1	0
FFFFF100H	OVIC10	OVIF10	OVMK10	0	0	0	OVPR102	OVPR101	OVPR100
FFFFF102H	OVIC11	OVIC11	OVMK11	0	0	0	OVPR112	OVPR111	OVPR110
FFFFF104H	OVIC12	OVIF12	OVMK12	0	0	0	OVPR122	OVPR121	OVPR120

Table 7-2. Interrupt Control Register Addresses and Bits (2/2)

Address	Register	Bit							
		7	6	5	4	3	2	1	0
FFFFF106H	OVIC13	OVIF13	OVMK13	0	0	0	OVPR132	OVPR131	OVPR130
FFFFF108H	OVIC14	OVIF14	OVMK14	0	0	0	OVPR142	OVPR141	OVPR140
FFFFF10AH	OVIC15	OVIF15	OVMK15	0	0	0	OVPR152	OVPR151	OVPR150
FFFFF10CH	CMIC40	CMIF40	CMMK40	0	0	0	CMPR402	CMPR401	CMPR400
FFFFF10EH	CMIC41	CMIF41	CMMK41	0	0	0	CMPR412	CMPR411	CMPR410
FFFFF110H	P10IC0	P10IF0	P10MK0	0	0	0	P10PR02	P10PR01	P10PR00
FFFFF112H	P10IC1	P10IF1	P10MK1	0	0	0	P10PR12	P10PR11	P10PR10
FFFFF114H	P10IC2	P10IF2	P10MK2	0	0	0	P10PR22	P10PR21	P10PR20
FFFFF116H	P10IC3	P10IF3	P10MK3	0	0	0	P10PR32	P10PR31	P10PR30
FFFFF118H	P11IC0	P11IF0	P11MK0	0	0	0	P11PR02	P11PR01	P11PR00
FFFFF11AH	P11IC1	P11IF1	P11MK1	0	0	0	P11PR12	P11PR11	P11PR10
FFFFF11CH	P11IC2	P11IF2	P11MK2	0	0	0	P11PR22	P11PR21	P11PR20
FFFFF11EH	P11IC3	P11IF3	P11MK3	0	0	0	P11PR32	P11PR31	P11PR30
FFFFF120H	P12IC0	P12IF0	P12MK0	0	0	0	P12PR02	P12PR01	P12PR00
FFFFF122H	P12IC1	P12IF1	P12MK1	0	0	0	P12PR12	P12PR11	P12PR10
FFFFF124H	P12IC2	P12IF2	P12MK2	0	0	0	P12PR22	P12PR21	P12PR20
FFFFF126H	P12IC3	P12IF3	P12MK3	0	0	0	P12PR32	P12PR31	P12PR30
FFFFF128H	P13IC0	P13IF0	P13MK0	0	0	0	P13PR02	P13PR01	P13PR00
FFFFF12AH	P13IC1	P13IF1	P13MK1	0	0	0	P13PR12	P13PR11	P13PR10
FFFFF12CH	P13IC2	P13IF2	P13MK2	0	0	0	P13PR22	P13PR21	P13PR20
FFFFF12EH	P13IC3	P13IF3	P13MK3	0	0	0	P13PR32	P13PR31	P13PR30
FFFFF130H	P14IC0	P14IF0	P14MK0	0	0	0	P14PR02	P14PR01	P14PR00
FFFFF132H	P14IC1	P14IF1	P14MK1	0	0	0	P14PR12	P14PR11	P14PR10
FFFFF134H	P14IC2	P14IF2	P14MK2	0	0	0	P14PR22	P14PR21	P14PR20
FFFFF136H	P14IC3	P14IF3	P14MK3	0	0	0	P14PR32	P14PR31	P14PR30
FFFFF138H	P15IC0	P15IF0	P15MK0	0	0	0	P15PR02	P15PR01	P15PR00
FFFFF13AH	P15IC1	P15IF1	P15MK1	0	0	0	P15PR12	P15PR11	P15PR10
FFFFF13CH	P15IC2	P15IF2	P15MK2	0	0	0	P15PR22	P15PR21	P15PR20
FFFFF13EH	P15IC3	P15IF3	P15MK3	0	0	0	P15PR32	P15PR31	P15PR30
FFFFF140H	DMAIC0	DMAIF0	DMAMK0	0	0	0	DMAPR02	DMAPR01	DMAPR00
FFFFF142H	DMAIC1	DMAIF1	DMAMK1	0	0	0	DMAPR12	DMAPR11	DMAPR10
FFFFF144H	DMAIC2	DMAIF2	DMAMK2	0	0	0	DMAPR22	DMAPR21	DMAPR20
FFFFF146H	DMAIC3	DMAIF3	DMAMK3	0	0	0	DMAPR32	DMAPR31	DMAPR30
FFFFF148H	CSIC0	CSIF0	CSMK0	0	0	0	CSPR02	CSPR01	CSPR00
FFFFF14AH	CSIC1	CSIF1	CSMK1	0	0	0	CSPR12	CSPR11	CSPR10
FFFFF14CH	CSIC2	CSIF2	CSMK2	0	0	0	CSPR22	CSPR21	CSPR20
FFFFF14EH	CSIC3	CSIF3	CSMK3	0	0	0	CSPR32	CSPR31	CSPR30
FFFFF150H	SEIC0	SEIF0	SEMK0	0	0	0	SEPR02	SEPR01	SEPR00
FFFFF152H	SRIC0	SRIF0	SRMK0	0	0	0	SRPR02	SRPR01	SRPR00
FFFFF154H	STIC0	STIF0	STMK0	0	0	0	STPR02	STPR01	STPR00
FFFFF156H	SEIC1	SEIF1	SEMK1	0	0	0	SEPR12	SEPR11	SEPR10
FFFFF158H	SRIC1	SRIF1	SRMK1	0	0	0	SRPR12	SRPR11	SRPR10
FFFFF15AH	STIC1	STIF1	STMK1	0	0	0	STPR12	STPR11	STPR10
FFFFF15CH	ADIC	ADIF	ADMK	0	0	0	ADPR2	ADPR1	ADPR0

7.3.5 In-service priority register (ISPR)

This register holds the priority level of the maskable interrupt currently acknowledged. When an interrupt request is acknowledged, the bit of this register corresponding to the priority level of that interrupt request is set (1) and remains set while the interrupt is serviced.

When the RETI instruction is executed, the bit corresponding to the interrupt request having the highest priority is automatically cleared (0) by hardware. However, it is not cleared (0) when execution is returned from non-maskable interrupt servicing or exception processing.

This register is read-only in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0	Address FFFFF166H	After reset 00H

Bit Position	Bit Name	Function
7 to 0	ISPR7 to ISPR0	In-Service Priority Flag Indicates priority of interrupt currently acknowledged. 0: Interrupt request with priority n not acknowledged 1: Interrupt request with priority n acknowledged

Remark n = 0 to 7 (priority level)

7.3.6 Maskable interrupt status flag (ID)

The ID flag is bit 5 of the PSW.

This controls the maskable interrupt's operating state, and stores control information on enabling/disabling acknowledgement of interrupt requests.

	31																							8	7	6	5	4	3	2	1	0	
PSW	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	NP	EP	ID	SAT	CY	OV	S	Z	After reset 00000020H

Bit Position	Bit Name	Function
5	ID	Interrupt Disable Indicates whether maskable interrupt processing is enabled or disabled. 0: Maskable interrupt acknowledgement enabled 1: Maskable interrupt acknowledgement disabled (pending) It is set to 1 by the DI instruction and reset to 0 by the EI instruction. Its value is also modified by the RETI instruction or LDSR instruction when referencing the PSW. Non-maskable interrupts and exceptions are acknowledged regardless of this flag. When a maskable interrupt is acknowledged, the ID flag is automatically set to 1 by hardware. The interrupt request generated during the acknowledgement disabled period (ID = 1) is acknowledged when the xxIFn bit of xxICn is set to 1, and the ID flag is cleared to 0.

7.3.7 Noise elimination

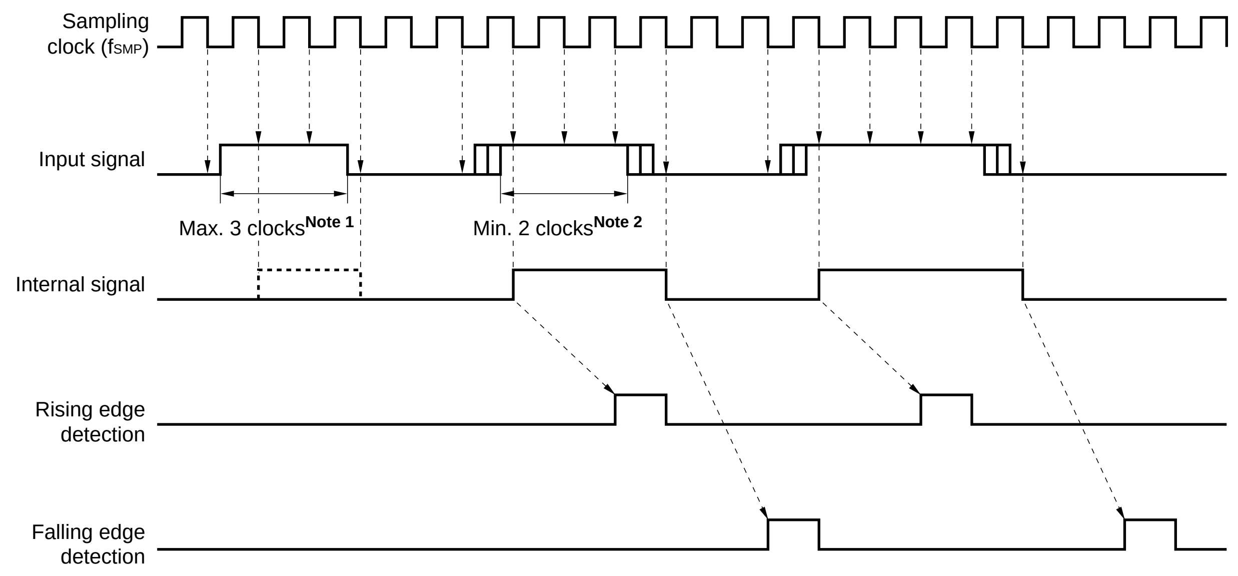
Digital noise elimination circuits are added to each of the INTPn0 to INTPn3, TIn, TCLRn and ADTRG pins (n = 10 to 15). Using these circuits, these pins' input level is sampled each sampling clock cycle (f_{SMP}). If the same level cannot be detected 3 times consecutively in the sampling results, that input pulse is removed as noise.

The noise elimination time at each pin is shown below.

Pin	Sampling Clock (f_{SMP})	Noise Elimination Time
TCLR10 to TCLR15	ϕ	2×ϕ to 3×ϕ
TI10 to TI15	ϕ	
INTP100 to INTP103, INTP110 to INTP113, INTP120 to INTP123, INTP130 to INTP133, INTP140 to INTP143, INTP150 to INTP152, INTP153/ADTRG	ϕ	

Remark ϕ: Internal system clock

Figure 7-9. Example of Noise Elimination Timing



Notes 1. Pulse width of unrecognizable noise.

2. Pulse width of recognizable signals.

Cautions 1. If the input pulse width is between 2 and 3 sampling clocks, whether the input pulse is detected as a valid edge or eliminated as a noise is indefinite.

2. To securely detect the level as a pulse, the same level input of 3 sampling clocks or more is required.

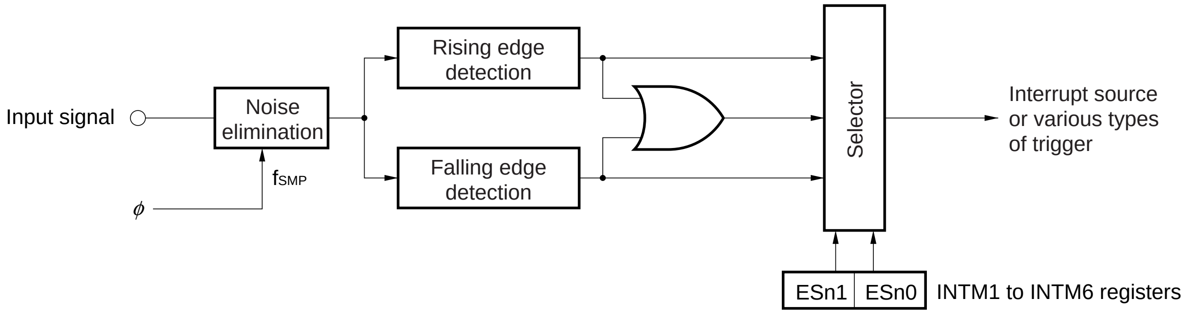
3. When noise is generated in synchronization with a sampling clock, this may not be recognized as noise. In this case, eliminate the noise by attaching a filter to the input pin.

7.3.8 Edge detection function

The valid edge of pins INTPn0 to INTPn3 and ADTRG can be selected by program. The valid edge that can be selected is one of the following (n = 10 to 15).

- Rising edge
- Falling edge
- Both the rising and falling edges

Edge detected INTPn0 to INTPn3 and ADTRG signals become interrupt factors or capture triggers. The block diagram of the edge detectors for these pins is shown below.



Remark n = 00 to 03, 10 to 13, 20 to 23, 30 to 33, 40 to 43, 50 to 53

ϕ: Internal system clock

f_{SMP}: Sampling clock

Valid edges are specified in external interrupt mode registers 1 to 6 (INTM1 to INTM6).

(1) External interrupt mode registers 1 to 6 (INTM1 to INTM6)

These are registers that specify the valid edge for external interrupt requests (INTP100 to INTP103, INTP110 to INTP113, INTP120 to INTP123, INTP130 to INTP133, INTP140 to INTP143, INTP150 to INTP152, INTP153/ADTRG), by external pins. The correspondence between each register and the external interrupt requests which that register controls is shown below.

- INTM1: INTP100 to INTP103
- INTM2: INTP110 to INTP113
- INTM3: INTP120 to INTP123
- INTM4: INTP130 to INTP133
- INTM5: INTP140 to INTP143
- INTM6: INTP150 to INTP152, INTP153/ADTRG

INTP153 is used for both an A/D converter external trigger input (ADTRG) and a pin. Therefore, if the ES531 and ES530 bits of INTM6 are set in the external trigger mode by bits TRG0 to TRG2 of A/D converter mode register 1 (ADM1), they specify the active edge of the external trigger input (ADTRG).

The valid edge can be specified independently for each pin, as the rising edge, the falling edge or both the rising and falling edges.

These registers can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
INTM1	ES031	ES030	ES021	ES020	ES011	ES010	ES001	ES000	FFFFF182H	00H
Control pins	INTP103		INTP102		INTP101		INTP100			
INTM2	ES131	ES130	ES121	ES120	ES111	ES110	ES101	ES100	FFFFF184H	00H
Control pins	INTP113		INTP112		INTP111		INTP110			
INTM3	ES231	ES230	ES221	ES220	ES211	ES210	ES201	ES200	FFFFF186H	00H
Control pins	INTP123		INTP122		INTP121		INTP120			
INTM4	ES331	ES330	ES321	ES320	ES311	ES310	ES301	ES300	FFFFF188H	00H
Control pins	INTP133		INTP132		INTP131		INTP130			
INTM5	ES431	ES430	ES421	ES420	ES411	ES410	ES401	ES400	FFFFF18AH	00H
Control pins	INTP143		INTP142		INTP141		INTP140			
INTM6	ES531	ES530	ES521	ES520	ES511	ES510	ES501	ES500	FFFFF18CH	00H
Control pins	INTP153/ADTRG		INTP152		INTP151		INTP150			

Bit Position	Bit Name	Function
7 to 0	ESmn1, ESmn0 (m = 5 to 0, n = 3 to 0)	Edge Select Specifies the valid edge of the INTP1mn pins and ADTRG pin.

ESmn1	ESmn0	Operation
0	0	Falling edge
0	1	Rising edge
1	0	RFU (reserved)
1	1	Both the rising and falling edges

7.4 Software Exception

A software exception is generated when the CPU executes the TRAP instruction, and can be always acknowledged.

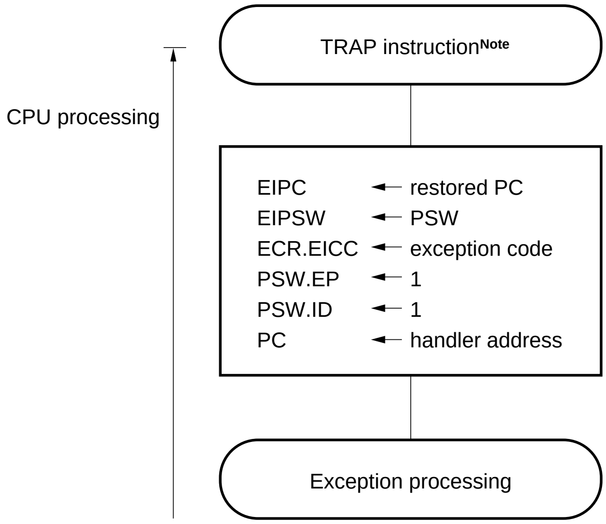
7.4.1 Operation

If a software exception occurs, the CPU performs the following processing, and transfers control to the handler routine:

(1) Saves the restored PC to EIPC.
(2) Saves the current PSW to EIPSW.
(3) Writes an exception code to the lower 16 bits (EICC) of ECR (interrupt source).
(4) Sets the EP and ID bits of the PSW.
(5) Sets the handler address (00000040H or 00000050H) corresponding to the software exception to the PC, and transfers control.

Figure 7-10 illustrates how a software exception is processed.

Figure 7-10. Software Exception Processing



Note TRAP Instruction Format: TRAP vector (however the vector is the value 0 to 1FH.)

The handler address is determined by the TRAP instruction's operand (vector). If the vector is 0 to 0FH, it becomes 00000040H, and if the vector is 10H to 1FH, it becomes 00000050H.

7.4.2 Restore

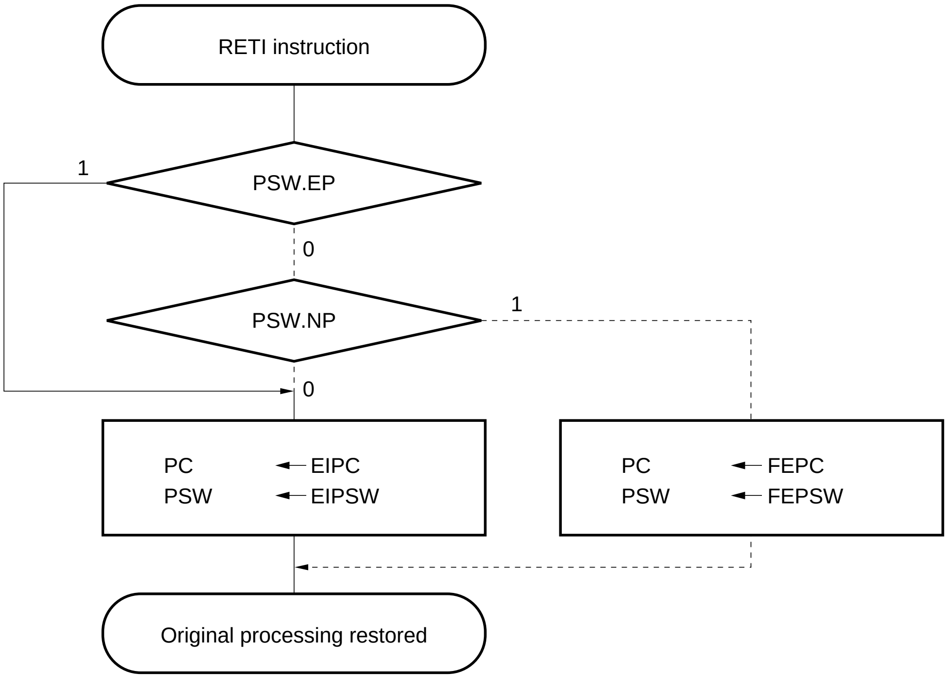
To restore from the software exception processing, the RETI instruction is used.

By executing the RETI instruction, the CPU carries out the following processing and shifts control to the restored PC's address.

(1) Loads the restored PC and PSW from EIPC and EIPSW because the EP bit of PSW is 1.
(2) Transfers control to the address of the restored PC and PSW.

Figure 7-11 illustrates the processing of the RETI instruction.

Figure 7-11. RETI Instruction Processing



Caution **When the PSW.EP bit and the PSW.NP bit are changed by the LDSR instruction during the software exception process, in order to restore the PC and PSW correctly during recovery by the RETI instruction, it is necessary to set PSW.EP back to 1 using the LDSR instruction immediately before the RETI instruction.**

Remark The solid line shows the CPU processing flow.

7.4.3 Exception status flag (EP)

The EP flag is a status flag used to indicate that exception processing is in progress. It is set when an exception occurs.

	31 … 8	7	6	5	4	3	2	1	0	
PSW	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	NP	EP	ID	SAT	CY	OV	S	Z	After reset 00000020H

Bit Position	Bit Name	Function
6	EP	Exception Pending Shows that exception processing is in progress. 0: Exception processing not in progress. 1: Exception processing in progress.

7.5 Exception Trap

The exception trap is an interrupt that is requested when illegal execution of an instruction takes place. In the V850E/MS1, an illegal op code exception (ILGOP: ILleGal Opcode trap) is considered an exception trap.

An illegal op code exception is generated in the case where the sub op code of the following instruction is an illegal op code when execution of that instruction is attempted.

7.5.1 Illegal op code definition

The illegal op code has a 32-bit long instruction format: bits 10 to 5 are 111111B and bits 26 to 23 are 0111B to 1111B, with bit 16 defined as an optional instruction code, 0B.

15 11	10 5	4 0	31 27	26 23	22 17	16
× × × × ×	1 1 1 1 1 1	× × × × ×	× × × × ×	0 1 1 1 to 1 1 1 1	× × × × × ×	0

×: don’t care

Caution Since it is possible to assign this instruction to an illegal op code in the future, it is recommended that it not be used.

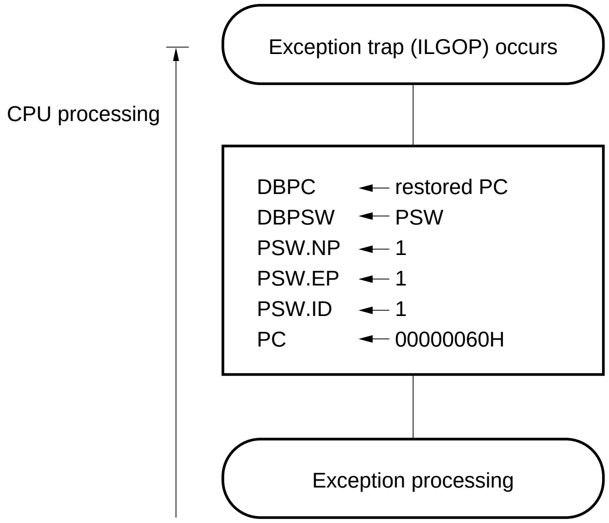
7.5.2 Operation

If an exception trap occurs, the CPU performs the following processing, and transfers control to the handler routine:

(1) Saves the restored PC to DBPC.
(2) Saves the current PSW to DBPC.
(3) Sets the NP, EP and ID bits of PSW.
(4) Sets the handler address (00000060H) corresponding to the exception trap to the PC, and transfers control.

Figure 7-12 illustrates how the exception trap is processed.

Figure 7-12. Exception Trap Processing



7.5.3 Restore

Recovery from an exception trap is not possible. Perform system reset by $\overline{\text{RESET}}$ input.

7.6 Multiple Interrupt Processing Control

Multiple interrupt processing control is a process by which the interrupt request currently being processed can be interrupted during processing if there is an interrupt request with a higher priority level, and the higher priority interrupt request is acknowledged and processed first.

If there is an interrupt request with a lower priority level than the interrupt request currently being processed, that interrupt request is held pending.

Maskable interrupt multiple processing control is executed when an interrupt has an enable status (ID = 0). Thus, if multiple interrupts are executed, it is necessary to have an interrupt enable status (ID = 0) even for an interrupt processing routine.

If a maskable interrupt or a software exception is generated in a maskable interrupt or software exception service program, it is necessary to save EIPC and EIPSW.

This is accomplished by the following procedure.

(1) To acknowledge maskable interrupts in a service program

Service program of maskable interrupt or exception

```
          ...
          ...
•  EIPC saved to memory or register
•  EIPSW saved to memory or register
•  EI instruction (enables interrupt acknowledgement)
          ...
          ...                                          ←  Maskable interrupt acknowledgement
          ...
          ...
•  DI instruction (disables interrupt acknowledgement)
•  Saved value restored to EIPSW
•  Saved value restored to EIPC
•  RETI instruction
```

(2) To generate an exception in a service program

Service program of maskable interrupt or exception

```
            ...
            ...
•  EIPC saved to memory or register
•  EIPSW saved to memory or register
            ...
•  TRAP instruction                      ←  Exception such as TRAP instruction acknowledged.
            ...
•  Saved value restored to EIPSW
•  Saved value restored to EIPC
•  RETI instruction
```

The priority order for multiple interrupt processing control has 8 levels, from 0 to 7 for each maskable interrupt request (0 is the highest priority), which can be set as desired via software. The priority order level is set with the xxPRn0 to xxPRn2 bits of the interrupt control request register (xxlCn), which is provided for each maskable interrupt request. At system reset time, an interrupt request is masked by the xxMKn bit and the priority order is set to level 7 by the xxPRn0 to xxPRn2 bits.

The priority order of maskable interrupts is as follows.

(High) Level 0 > Level 1 > Level 2 > Level 3 > Level 4 > Level 5 > Level 6 > Level 7 (Low)

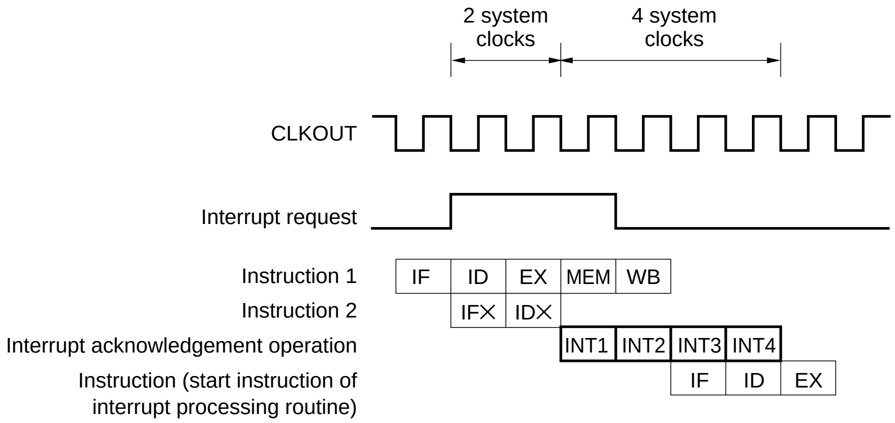
Interrupt processing that has been suspended as a result of multiple processing control is resumed after the interrupt processing of the higher priority has been completed and the RETI instruction has been executed.
A pending interrupt request is acknowledged after the current interrupt processing has been completed and the RETI instruction has been executed.

Caution In the non-maskable interrupt processing routine (time until the RETI instruction is executed), maskable interrupts are not acknowledged but are held pending.

7.7 Interrupt Latency Time

The following table describes the V850E/MS1 interrupt latency time (from interrupt generation to start of interrupt processing).

Figure 7-13. Pipeline Operation at Interrupt Request Acknowledgement (Outline)



Remark INT1 to INT4: Interrupt acknowledgement processing
IF×: Invalid instruction fetch
ID×: Invalid instruction decode

Interrupt Latency Time (Internal System Clock)			Condition
	Internal interrupt	External interrupt	
Minimum	5	7	The following cases are exceptions. • In IDLE/software STOP mode • External bus is accessed • Two or more interrupt request non-sample instructions are executed in succession • Access to interrupt control register
Maximum	11	13	

7.8 Periods in Which Interrupt Is Not Acknowledged

An interrupt is acknowledged while an instruction is being executed. However, no interrupt will be acknowledged between an interrupt non-sample instruction and the next instruction.

The interrupt request non-sampling instructions are as follows.

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (vs. PSW)
- The store instruction for the interrupt control register (xxICn) and command register (PRCMD)

[MEMO]

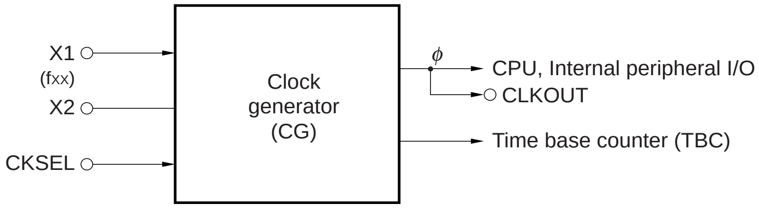
CHAPTER 8 CLOCK GENERATOR FUNCTIONS

The clock generator (CG) generates and controls the internal system clock (ϕ) which is supplied to each internal unit, of which the CPU is the primary unit.

8.1 Features

- ○ Multiplier function using a PLL (phase locked loop) synthesizer
- ○ Clock Source
  - Oscillation by connecting an oscillator: $f_{XX} = \phi/5$
  - External clock: $f_{XX} = 2 \times \phi, \phi/5$
- ○ Power save control
  - HALT mode
  - IDLE mode
  - Software STOP mode
  - Clock output inhibit function
- ○ Internal system clock output function

8.2 Configuration



Remark ϕ: Internal system clock frequency
f_{XX}: External resonator or external clock frequency

8.3 Input Clock Selection

The clock generator is configured from an oscillator and a PLL synthesizer. If, for example an 8 MHz crystal resonator or ceramic resonator is connected to pins X1 and X2, an internal system clock (ϕ) of 40 MHz can be generated.

Also, an external clock can be input directly to the oscillator. In this case, input a clock signal to the X1 pin only and leave the X2 pin open.

Two types of mode, a PLL mode and a direct mode, are provided as the basic operation modes for the clock generator. Selection of the operation mode is done by the CKSEL pin. The input of this pin latches at reset time.

CKSEL	Operation Mode
0	PLL mode
1	Direct mode

Caution Fix the input level of the CKSEL pin before use. If it is switched during operation, there is a possibility of malfunction occurring.

8.3.1 Direct mode

In the direct mode, an external clock with double the internal system clock's frequency is input. Since the oscillator and PLL synthesizer are not operating, a large amount of power can be saved. Mainly, the V850E/MS1 is used in application systems where it operates at relatively low frequencies. In consideration of EMI countermeasures, if the external clock frequency (f_{XX}) is 32 MHz (internal system clock (ϕ) = 16 MHz) or greater, the PLL mode is recommended.

Caution In the direct mode, be sure to input an external clock (do not connect an external resonator).

8.3.2 PLL mode

In the PLL mode, by connecting an external resonator or inputting an external clock and multiplying this clock by the PLL synthesizer, an internal system clock (ϕ) is generated.

At reset time, an internal system clock (ϕ) which is 5 times the frequency of the input clock's frequency (f_{XX}) ($5 \times f_{XX}$), is generated.

In the PLL mode, if the clock supply from an external resonator or external clock source stops, the internal system clock (ϕ) continues to operate based on the self-propelled frequency of the clock generator's internal voltage controlled oscillator (VCO). In this case, ϕ = approx. 1 MHz (target). However, do not devise an application method in which you expect to use this self-propelled frequency.

Example Clock used when in the PLL mode

System Clock Frequency (ϕ) [MHz]	External Resonator/External Clock Frequency (f_{XX}) [MHz]
40.000	8.0000
32.768	6.5536
25.000	5.0000
20.000	4.0000
16.384	3.2768

8.3.3 Clock control register (CKC)

When in the PLL mode, this is an 8-bit register which controls the internal system clock frequency (ϕ), and it can be written to only by a specific combination of instruction sequences so that it cannot be rewritten easily by mistake due to program runaway.

This register can be read/written in 8- or 1-bit units.

Caution When in the direct mode, do not change the setting of this register.

	7	6	5	4	3	2	1	0		
CKC	0	0	0	0	0	0	CKDIV1	CKDIV0	Address FFFFF072H	After reset 00H

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>1, 0</td><td>CKDIV1,
CKDIV0</td><td>Clock Divide
Sets the internal system clock frequency (ϕ) when in the PLL mode.
<table>
<tr><th>CKDIV1</th><th>CKDIV0</th><th>Internal System Clock (ϕ)</th></tr>
<tr><td>0</td><td>0</td><td>5 × f_{XX}</td></tr>
<tr><td>0</td><td>1</td><td>Setting prohibited</td></tr>
<tr><td>1</td><td>0</td><td>f_{XX}</td></tr>
<tr><td>1</td><td>1</td><td>f_{XX}/2</td></tr>
</table>
</td></tr>
</table>

The sequence of setting data to this register is the same as for the power save control register (PSC). However, the restrictions shown in **Remark 2** of **3.4.9 Specific registers** do not apply. For details, refer to **8.5.2 Control registers**.

Example Clock generator setting

Operation Mode	CKSEL Pin	CKC Register		Input Clock (f_{XX})	Internal System Clock (ϕ)
		CKDIV1 Bit	CKDIV0 Bit		
Direct mode	High-level input	0	0	16 MHz	8 MHz
PLL mode	Low-level input	0	0	8 MHz	40 MHz
		1	0	8 MHz	8 MHz
		1	1	8 MHz	4 MHz
Other than above				Setting prohibited	

8.4 PLL Lockup

Lockup time (frequency stabilization time) is the amount of time from immediately after the software STOP mode is released after the power is turned on, until the phase locks at the proper frequency and becomes stable. The state until this stabilization occurs is called the unlocked state and the stabilized state is called the locked state.

There is an UNLOCK flag which reflects the PLL's frequency stabilization state, and a PRERR flag which shows when a protection error occurs, in the system status register (SYS).

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
SYS	0	0	0	PRERR	0	0	0	UNLOCK	FFFFF078H	0000000×B

Bit Position	Bit Name	Function
0	UNLOCK	Unlock Status Flag This is an exclusive read flag and shows the PLL's unlocked state. As long as the lockup state is maintained, it is kept at 0, and is not initialized when system reset occurs. 0: Indicates that the PLL is in a locked state. 1: Indicates that the PLL is not locked (in an unlocked state).

Remark For an explanation of the PRERR flag, refer to **3.4.9 (2) System status register (SYS)**.

If the clock stops, the power fails, or some other factor occurs to cause the unlocked state, in control processing which depends on software execution speed such as real-time processing, be sure to begin processing after judging the UNLOCK flag by software immediately after operation starts, and after waiting for the clock to stabilize again.

On the other hand, for static processing such as setting of internal hardware, or initialization of register data and memory data, it is possible to execute these without waiting for the UNLOCK flag to be reset.

The relationship between the oscillation stabilization time (the time from when the resonator starts to oscillate until the input waveform stabilizes) when a resonator is used, and the PLL lockup time (the time until the frequency is stabilized) is shown below.

Oscillation stabilization time < PLL lockup time

8.5 Power Saving Control

8.5.1 Outline

The V850E/MS1 standby function comprises the following three modes:

(1) HALT mode

In this mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the CPU's operation clock stops. Supply of the clock to the other internal peripheral functions is continued. Through intermittent operation by combining with the normal operating mode, the system's total power consumption can be reduced.

The system is switched to the HALT mode via an exclusive instruction (the HALT instruction).

(2) IDLE mode

In this mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but supply of the internal system clock is stopped, which causes the system overall to stop.

When releasing the system from the IDLE mode, it is not necessary to secure the oscillation stabilization time of the oscillator, so it is possible to switch to normal operation at high speed.

The system enters the IDLE mode in accordance with the settings in the PSC register (specific register).

The IDLE mode is positioned midway between the software STOP mode and the HALT mode in relation to clock stabilization time and current consumption and is used for cases where the low current consumption mode is used and where it is desired to eliminate the clock stabilization time after it is released.

(3) Software STOP mode

In this mode, the clock generator (oscillator and PLL synthesizer) is stopped and the system overall is stopped, thus entering an ultra-low power consumption state where only leak current is lost.

It is possible to enter the software STOP mode by setting the PSC register (specific register).

★ **(a) When in the PLL Mode**

By setting the register by software, you can enter the software STOP mode. At the same time the oscillator stops, the PLL synthesizer's clock output stops. After releasing the software STOP mode, it is necessary to secure oscillation stabilization time for the oscillator for a period of time until the system clock stabilizes. Also, depending on the program, PLL lockup time may be required.

(4) Clock output inhibit mode

Internal system clock output from the CLKOUT pin is prohibited.

The operation of the clock generator in normal operation, and in the HALT, IDLE, and software STOP modes is shown in Table 8-1.

By combining each of the modes and by switching modes according to the required usage, it is possible to realize an effective low power consumption system.

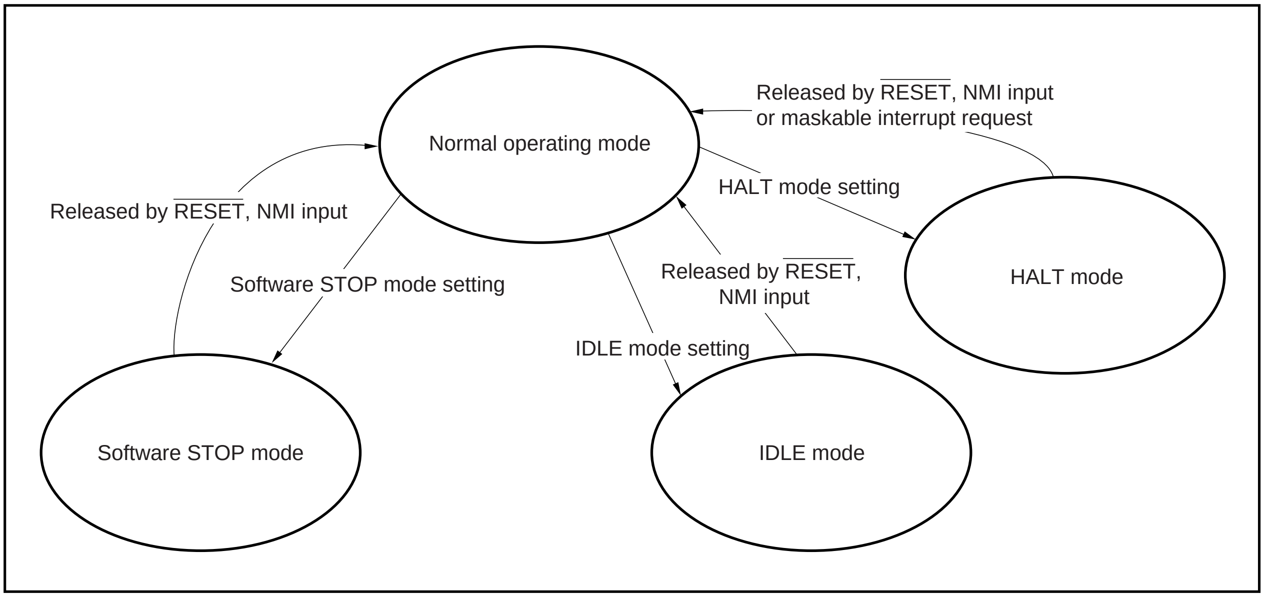
Table 8-1. Clock Generator Operation by Power Save Control

Clock Source		Power Save Mode	Oscillator (OSC)	PLL Synthesizer	Supply of Clock to Internal Peripheral I/O	Supply of Clock to the CPU
PLL mode	Oscillation by resonator	(During normal operation)	○	○	○	○
		HALT mode	○	○	○	×
		IDLE mode	○	○	×	×
		Software STOP mode	×	×	×	×
	External clock	(During normal operation)	×	○	○	○
		HALT mode	×	○	○	×
		IDLE mode	×	○	×	×
		Software STOP mode	×	×	×	×
Direct mode		(During normal operation)	×	×	○	○
		HALT mode	×	×	○	×
		IDLE mode	×	×	×	×
		Software STOP mode	×	×	×	×

○: Operating

×: Stopped

Figure 8-1. Power Save Mode State Transition Diagram


Normal operating mode
Released by $\overline{\text{RESET}}$, NMI input or maskable interrupt request
HALT mode setting
HALT mode
Released by $\overline{\text{RESET}}$, NMI input
Software STOP mode setting
Released by $\overline{\text{RESET}}$, NMI input
IDLE mode setting
Software STOP mode
IDLE mode

8.5.2 Control registers

(1) Power save control register (PSC)

This is an 8-bit register that controls the power save mode.

This is one of the specific registers and is active only when accessed by a specific sequence during a write operation. For details, refer to **3.4.9 Specific registers**.

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PSC	DCLK1	DCLK0	TBCS	CESEL	0	IDLE	STP	0	Address FFFFF070H	After reset 00H

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>DCLK1,
DCLK0</td><td>Disable CLKOUT
This specifies the CLKOUT pin’s operating mode.
<table>
<tr><th>DCLK1</th><th>DCLK0</th><th>Mode</th></tr>
<tr><td>0</td><td>0</td><td>Normal output mode</td></tr>
<tr><td>0</td><td>1</td><td>RFU (reserved)</td></tr>
<tr><td>1</td><td>0</td><td>RFU (reserved)</td></tr>
<tr><td>1</td><td>1</td><td>Clock output inhibit mode</td></tr>
</table></td></tr>
<tr><td>5</td><td>TBCS</td><td>Time Base Count Select
Selects the time base counter clock.
0: $f_{XX}/2^8$
1: $f_{XX}/2^9$
Details are shown in <b>8.6.2 Time base counter (TBC</b>).</td></tr>
<tr><td>4</td><td>CESEL</td><td>Crystal/External Select
Specifies the function of pins X1 and X2.
0: An oscillator is connected to pins X1 and X2.
1: An external clock is connected to pin X1.
If CESEL = 1, the oscillator’s feedback loop is cut and current leakage is prevented when in the software STOP mode. Also, the oscillation stabilization time count by the time base counter (TBC) after the software STOP mode is released is not carried out.</td></tr>
<tr><td>2</td><td>IDLE[Note]</td><td>IDLE Mode
Specifies the IDLE mode.
It enters the IDLE state if 1 is written.
It is automatically reset (0) if the IDLE mode is released.</td></tr>
<tr><td>1</td><td>STP[Note]</td><td>STOP Mode
Specifies the software STOP mode.
It enters the STOP state if 1 is written.
It is automatically reset (0) if the software STOP mode is released.</td></tr>
</table>

Note If the IDLE bit is set at 1 and the STP bit is also set at 1, the system enters the software STOP mode.

8.5.3 HALT mode

(1) Setting and operating state

In this mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but the CPU's operation clock stops. Supply of the clock to other internal peripheral I/O functions is continued and their operation continues. By setting the HALT mode during the time when CPU is idle, the system's total power consumption can be reduced.

Switching to the HALT mode is accomplished by executing the HALT instruction.

In the HALT mode, program execution stops, but all the contents of all the registers, internal RAM, and ports are held in the state they were in just before the HALT mode was entered. Also, internal peripheral I/O (other than the ports) that is not dependent on CPU instruction processing continues operation. The state of each hardware unit when in the HALT mode is shown in Table 8-2.

Remark Even after HALT instruction execution, instruction fetch operations continue until the internal instruction prefetch queue becomes full. When the prefetch queue becomes full, it stops in the state shown in Table 8-2.

Table 8-2. Operating States When in HALT Mode

Function		Operating State
Clock generator		Operating
Internal system clock		Operating
CPU		Stop
Port		Hold
Internal peripheral I/O (except ports)		Operating
Internal data		All the CPU's registers, status, data, internal RAM contents and other internal data, etc. are retained in the state they were in before entering the HALT mode.
When in external expansion mode	D0 to D15	Operating
	A0 to A23	
	$\overline{\text{RD}}$, $\overline{\text{WE}}$, $\overline{\text{OE}}$, $\overline{\text{BCYST}}$	
	$\overline{\text{LWR}}$, $\overline{\text{UWR}}$, $\overline{\text{IORD}}$, $\overline{\text{IOWR}}$	
	$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$	
	$\overline{\text{RAS0}}$ to $\overline{\text{RAS7}}$	
	$\overline{\text{LCAS}}$, $\overline{\text{UCAS}}$	
	$\overline{\text{REFRQ}}$	
	$\overline{\text{HLDRQ}}$	
	$\overline{\text{HLDAK}}$	
	$\overline{\text{WAIT}}$	
CLKOUT		Clock output (when not in clock output inhibit)

(2) Releasing HALT mode

The HALT mode can be released by NMI pin input, an unmasked maskable interrupt request, or a $\overline{\text{RESET}}$ signal input.

(a) Release by NMI pin input, maskable interrupt request

The HALT mode is unconditionally released by NMI pin input or an unmasked maskable interrupt request regardless of the priority. However, if the HALT mode is set in an interrupt processing routine, the operation will differ as follows:

(i) If an interrupt request with a priority lower than that of the interrupt request under execution is generated, the HALT mode is released, but the newly generated interrupt request is not acknowledged. The new interrupt request will be kept pending.

(ii) If an interrupt request with a priority higher (including NMI request) than the interrupt request under execution is generated, the HALT mode is released, and the interrupt request is also acknowledged.

Table 8-3. Operations after HALT Mode Is Released by Interrupt Request

Releasing Source	Interrupt Enable (EI) State	Interrupt Disable (DI) State
NMI request	Branch to handler address	
Maskable interrupt request	Branch to the handler address or execute the next instruction.	Execute the next instruction.

(b) Release by $\overline{\text{RESET}}$ pin input

This operation is the same as a normal reset operation.

8.5.4 IDLE mode

(1) Settings and operating state

In this mode, the clock generator (oscillator and PLL synthesizer) continues to operate, but supply of the internal system clock is stopped, which causes the system overall to stop.

When releasing the system from the IDLE mode, it is not necessary to secure the oscillation stabilization time of the oscillator, so it is possible to switch to normal operation at high speed.

The IDLE mode is entered by the setting of the PSC register (specific register), set through a store instruction (ST/SST instruction) or a bit operation instruction (SET1/CLR1/NOT1 instruction) (refer to **3.4.9 Specific registers**).

In the IDLE mode, program execution is stopped, but all the contents of all the registers, internal RAM, and ports are held. Operation of the internal peripheral I/O (except the ports) is also stopped.

The state of each hardware unit when in IDLE mode is as shown in Table 8-4.

Table 8-4. Operating States When in IDLE Mode

Function		Operating State
Clock generator		Operating
Internal system clock		Stop
CPU		Stop
Port		Hold
Internal peripheral I/O (except ports)		Stop
Internal data		All the CPU's registers, status, data, internal RAM contents and other internal data, etc. are retained in the state they were in before entering the HALT mode.
When in external expansion mode	D0 to D15	High-impedance
	A0 to A23	
	$\overline{\text{RD}}$, $\overline{\text{WE}}$, $\overline{\text{OE}}$, $\overline{\text{BCYST}}$	
	$\overline{\text{LWR}}$, $\overline{\text{UWR}}$, $\overline{\text{IORD}}$, $\overline{\text{IOWR}}$	High-level output
	$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$	
	$\overline{\text{RAS0}}$ to $\overline{\text{RAS7}}$	Operating
	$\overline{\text{LCAS}}$, $\overline{\text{UCAS}}$	
	$\overline{\text{REFRQ}}$	
	$\overline{\text{HLDRQ}}$	Input (no sampling)
	$\overline{\text{HLDAK}}$	High-impedance
	$\overline{\text{WAIT}}$	Input (no sampling)
CLKOUT		Low-level output

(2) Releasing IDLE mode

The IDLE Mode is released by NMI pin input or $\overline{\text{RESET}}$ pin input.

(a) Release by NMI pin input

This is acknowledged as a NMI request together with a release of the IDLE mode.

However, in cases where setting the system in the IDLE mode is included in the NMI processing routine, the IDLE mode is released only, and this interrupt is not acknowledged. The interrupt request itself is held pending.

The interrupt processing that is started when the IDLE mode is released by NMI pin input is treated in the same way as ordinary NMI interrupt processing in an emergency, etc. (since the NMI interrupt handler's address is unique). Consequently, in cases where it is necessary to distinguish between the two in a program, it is necessary to prepare the software status in advance and set the status before setting the PSC register using the store instruction or a bit operation instruction. By checking this status in NMI interrupt processing, it is possible to distinguish it from an ordinary NMI.

(b) Release by $\overline{\text{RESET}}$ pin input

This is the same as an ordinary reset operation.

8.5.5 Software STOP mode

(1) Settings and operating state

In this mode, the clock generator (oscillator and PLL synthesizer) is stopped. The system overall is stopped, and it enters an ultra-low power consumption state where only device leakage current is lost.

It is possible to enter the software STOP mode by setting the PSC register (specific register) using a store instruction (ST/SST instruction) or a bit manipulation instruction (SET1/CLR1/NOT1 instruction) in software (refer to **3.4.9 Specific registers**).

In the case of the PLL mode and oscillator connection mode (CESEL bit of the PSC register = 0), it is necessary to secure the oscillation stabilization of the oscillator after releasing the software STOP mode.

In the software STOP mode, program execution stops, but all the contents of all the registers, internal RAM, and ports are held in the state they were in just before entering the software STOP mode. Operation of the internal peripheral I/O (except the ports) is also stopped.

The status of each hardware unit during the software STOP mode is as shown in Table 8-5.

Caution In the case of the direct mode (CKSEL pin = 1) or external clock connection mode (CESEL bit of the PSC register = 1), the software STOP mode cannot be used.

Table 8-5. Operating States When in Software STOP Mode

Function		Operating State
Clock generator		Stop
Internal system clock		Stop
CPU		Stop
Port[Note]		Hold
Internal peripheral I/O (except ports)		Stop
Internal data[Note]		All the CPU’s registers, status, data, internal RAM contents, other internal data, etc. are retained in the state they were in before entering the HALT mode.
When in external expansion mode	D0 to D15	High-impedance
	A0 to A23	
	$\overline{\text{RD}}$, $\overline{\text{WE}}$, $\overline{\text{OE}}$, $\overline{\text{BCYST}}$	
	$\overline{\text{LWR}}$, $\overline{\text{UWR}}$, $\overline{\text{IORD}}$, $\overline{\text{IOWR}}$	High-level output
	$\overline{\text{CS0}}$ to $\overline{\text{CS7}}$	
	$\overline{\text{RAS0}}$ to $\overline{\text{RAS7}}$	Operating
	$\overline{\text{LCAS}}$, $\overline{\text{UCAS}}$	
	$\overline{\text{REFRQ}}$	
	$\overline{\text{HLDRQ}}$	Input (no sampling)
	$\overline{\text{HLDAK}}$	High-impedance
	$\overline{\text{WAIT}}$	Input (no sampling)
CLKOUT		Low-level output

Note If the V_{DD} value is within the operable range.
However, even when it drops below the minimum operable voltage, if the data hold voltage V_{DDDR} is maintained, the contents of internal RAM only are held.

(2) Releasing software STOP mode

The software STOP mode is released by NMI pin input or $\overline{\text{RESET}}$ pin input.

Also, when releasing the software STOP mode in the PLL mode and the oscillator connection mode (CESEL bit of the PSC register = 0), it is necessary to secure oscillation stabilization time for the oscillator.

Note that depending on the program, PLL lockup time may also be necessary. For details, refer to **8.4 PLL Lockup**.

(a) Release by NMI Pin Input

An NMI pin input is acknowledged as an NMI request as well as a release of the software STOP mode.

However, if setting in the software STOP mode is included in an NMI processing routine, the software STOP mode only is released and the interrupt is not acknowledged. The interrupt request itself is held pending.

The interrupt processing started when the STOP mode is released by an NMI pin input is treated in the same way as ordinary NMI interrupt processing in an emergency, etc. (since the NMI interrupt handler address is unique). Consequently, in cases where it is necessary to distinguish between the two, it is necessary to prepare the software status in advance and set the status before setting the PSC register using the store instruction or a bit operation instruction. By checking this status in NMI interrupt processing, it is possible to distinguish it from an ordinary NMI.

(b) Release by $\overline{\text{RESET}}$ Pin Input

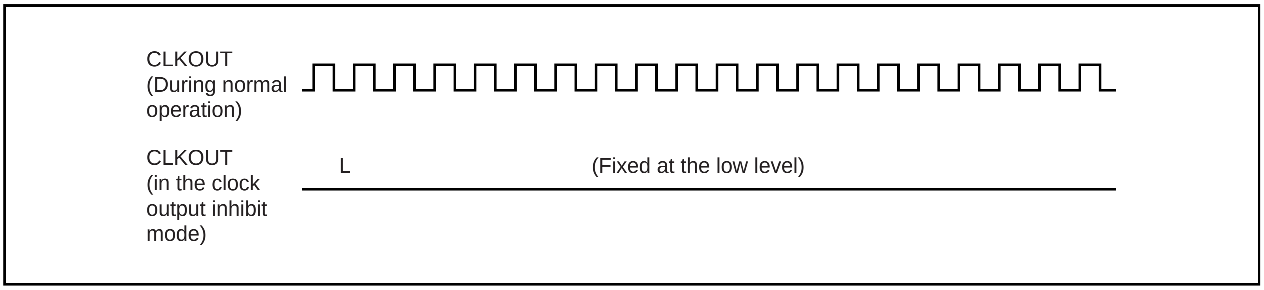
This is the same as an ordinary reset operation.

8.5.6 Clock output inhibit mode

If the DCLK0 bit and DCLK1 bit of the PSC register are set to 1, the system enters the clock output inhibit mode, in which clock output from the CLKOUT pin is disabled.

This is most appropriate in single-chip mode 0 and 1 systems, or in systems which access instruction fetches or data from external expansion devices asynchronously.

In this mode, since the CLKOUT signal output's operation is completely stopped, much lower power consumption and suppression of radiation noise from the CLKOUT pin is possible. Also, by combining this mode with the HALT, IDLE, and software STOP mode, more effective power saving becomes possible (refer to **8.5.2 Control registers**).



Remark When in flash memory programming mode, the CLKOUT signal is not output regardless of the PSC register setting.

8.6 Securing Oscillation Stabilization Time

8.6.1 Specifying securing of oscillation stabilization time

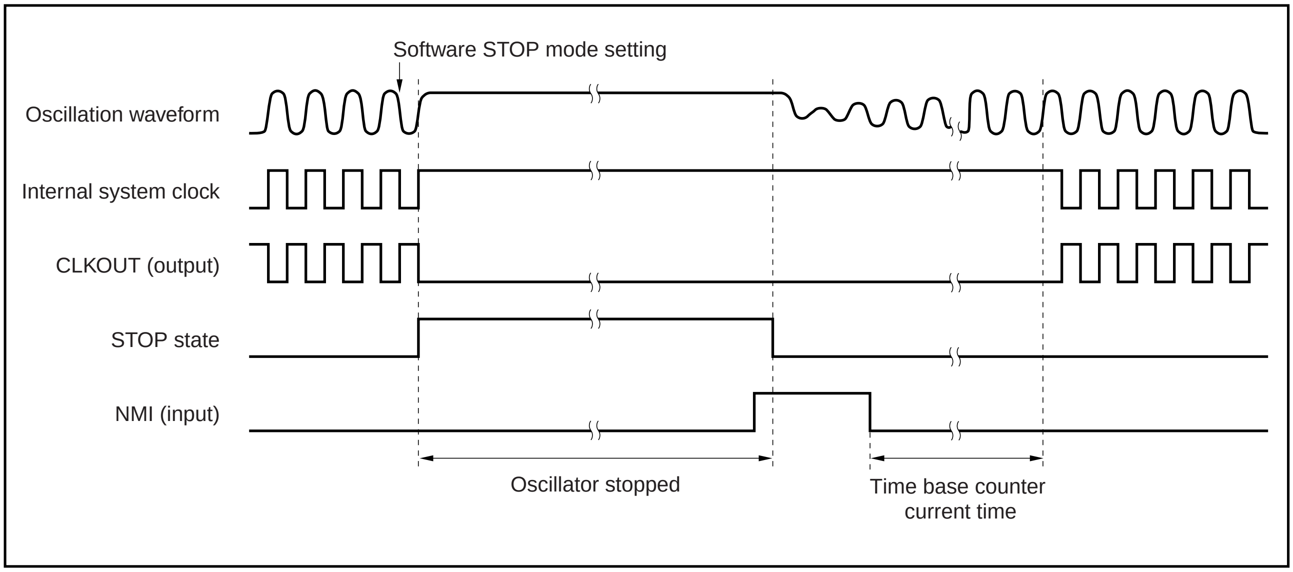
There are 2 methods for specifying securing of time for stabilizing the oscillator in the stop mode after releasing the software STOP mode.

(1) If securing time by the internal time base counter (NMI pin input)

If the active edge of the NMI pin is input, the software STOP mode is released. When the inactive edge is input to the pin, the time base counter (TBC) starts counting, and at that count time, the time until the clock output from the oscillator stabilizes is secured.

Oscillation stabilization time ≅ (Active level width after NMI input active edge detection) + (TBC count time)

After the proper time, start internal system clock output and branch to the NMI interrupt handler address.



The NMI pin should normally be set at the inactive level (for example, so that it changes to high level when the active edge is specified to be falling).

Furthermore, if an operation is executed which sets the system in the STOP mode for a time until an interrupt is received from the CPU from the NMI active edge input timing, the software STOP mode is quickly released.

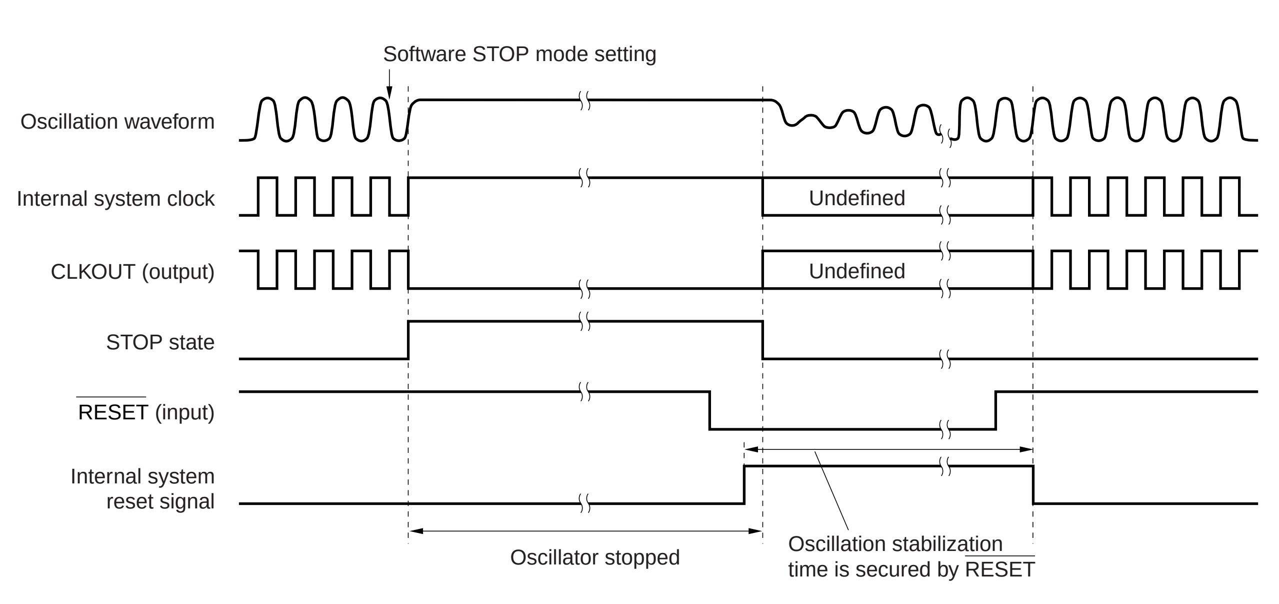
In the case of the PLL mode and the resonator connection mode (CESEL bit of PSC register = 0), program execution starts after the oscillation stabilization time is secured by the time base counter after input of the NMI pin's inactive edge.

(2) If securing time by the signal level width ($\overline{\text{RESET}}$ pin input)

By inputting the falling edge to the $\overline{\text{RESET}}$ pin, the software STOP mode is released.

At the signal low level width input to the pin, enough time is secured until the clock output from the oscillator stabilizes.

After inputting the rising edge to the $\overline{\text{RESET}}$ pin, supply of the internal system clock begins and the system branches to the handler address that was set at system reset time.


Software STOP mode setting
Oscillation waveform
Internal system clock
Undefined
CLKOUT (output)
Undefined
STOP state
$\overline{\text{RESET}}$ (input)
Internal system reset signal
Oscillator stopped
Oscillation stabilization time is secured by $\overline{\text{RESET}}$

8.6.2 Time base counter (TBC)

The time base counter (TBC) is used to secure the oscillation stabilization time of the oscillator when the software STOP mode is released.

- **Resonator connection time (PLL Mode, and CESEL bit of the PSC Register = 0)**
  After releasing the software STOP mode, the oscillation stabilization time is counted by the TBC and after counting is ended, program execution begins.

The TBC count clock is selected by the TBCS bit in the PSC register, and it is possible to set the following count times (refer to **8.5.2 (1) Power save control register (PSC)**).

Table 8-6. Example of Count Time ($\phi = 5 \times f_{XX}$)

TBCS Bit	Count Clock	Count Time			
		f_{XX} = 3.2768 MHz	f_{XX} = 5.0000 MHz	f_{XX} = 6.5536 MHz	f_{XX} = 8.0000 MHz
		ϕ = 16.384 MHz	ϕ = 25.000 MHz	ϕ = 32.768 MHz	ϕ = 40.000 MHz
0	$f_{XX}/2^8$	20.0 ms	13.1 ms	10.0 ms	8.1 ms
1	$f_{XX}/2^9$	40.0 ms	26.2 ms	20.0 ms	16.3 ms

f_{XX}: External resonator frequency
ϕ: Internal system clock frequency

CHAPTER 9 TIMER/COUNTER FUNCTION (REAL-TIME PULSE UNIT)

9.1 Features

- ○ Measures the pulse interval and frequency and outputs a programmable pulse.
  - 16-bit measurements are possible.
  - Pulse multiple states can be generated (interval pulse, one shot pulse)
- ○ Timer 1
  - 16-bit timer/event counter
  - Count clock sources: 2 types (internal system clock division selection, external pulse input)
  - Capture/compare common registers: 24
  - Count clear pins: TCLR10 to TCLR15
  - Interrupt sources: 30 types
  - External pulse outputs: 12
- ○ Timer 4
  - 16-bit interval timer
  - The count clock is selected from the internal system clock divisions.
  - Compare registers: 2
  - Interrupt sources: 2 types

9.2 Basic Configuration

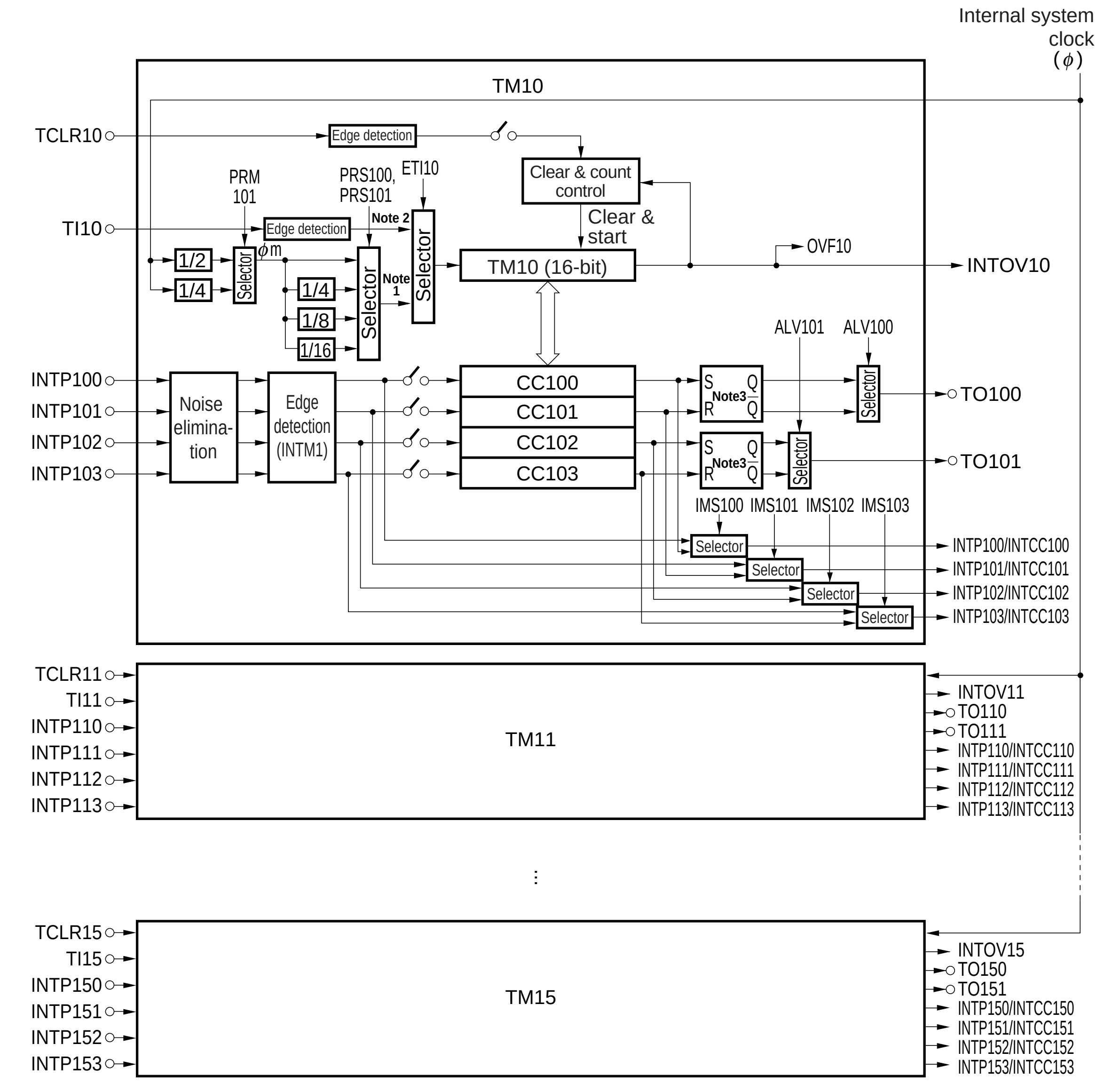
The basic configuration is shown below.

Table 9-1. RPU Configuration List

Timer	Count Clock	Register	Read/Write	Interrupt Signals Generated	Capture Trigger	Timer Output S/R	Other Functions
Timer 1	ϕ/2 ϕ/4 ϕ/8 ϕ/16 ϕ/32 ϕ/64 TI1n Pin Input (n = 0 to 5)	TM10	Read	INTOV10	—	—	External clear
		CC100	Read/write	INTCC100	INTP100	TO100 (S)	—
		CC101	Read/write	INTCC101	INTP101	TO100 (R)	—
		CC102	Read/write	INTCC102	INTP102	TO101 (S)	—
		CC103	Read/write	INTCC103	INTP103	TO101 (R)	—
		TM11	Read	INTOV11	—	—	External clear
		CC110	Read/write	INTCC110	INTP110	TO110 (S)	A/D conversion start trigger
		CC111	Read/write	INTCC111	INTP111	TO110 (R)	A/D conversion start trigger
		CC112	Read/write	INTCC112	INTP112	TO111 (S)	A/D conversion start trigger
		CC113	Read/write	INTCC113	INTP113	TO111 (R)	A/D conversion start trigger
		TM12	Read	INTOV12	—	—	External clear
		CC120	Read/write	INTCC120	INTP120	TO120 (S)	—
		CC121	Read/write	INTCC121	INTP121	TO120 (R)	—
		CC122	Read/write	INTCC122	INTP122	TO121 (S)	—
		CC123	Read/write	INTCC123	INTP123	TO121 (R)	—
		TM13	Read	INTOV13	—	—	External clear
		CC130	Read/write	INTCC130	INTP130	TO130 (S)	—
		CC131	Read/write	INTCC131	INTP131	TO130 (R)	—
		CC132	Read/write	INTCC132	INTP132	TO131 (S)	—
		CC133	Read/write	INTCC133	INTP133	TO131 (R)	—
		TM14	Read	INTOV14	—	—	External clear
		CC140	Read/write	INTCC140	INTP140	TO140 (S)	—
		CC141	Read/write	INTCC141	INTP141	TO140 (R)	—
		CC142	Read/write	INTCC142	INTP142	TO141 (S)	—
		CC143	Read/write	INTCC143	INTP143	TO141 (R)	—
		TM15	Read	INTOV15	—	—	External clear
		CC150	Read/write	INTCC150	INTP150	TO150 (S)	—
		CC151	Read/write	INTCC151	INTP151	TO150 (R)	—
		CC152	Read/write	INTCC152	INTP152	TO151 (S)	—
		CC153	Read/write	INTCC153	INTP153	TO151 (R)	—
Timer 4	ϕ/32 ϕ/64 ϕ/128 ϕ/256	TM40	Read	—	—	—	—
		CM40	Read/write	INTCM40	—	—	—
		TM41	Read	—	—	—	—
		CM41	Read/write	INTCM41	—	—	—

Remark ϕ: Internal system clock
S/R: Set/reset

(1) Timer 1 (16-bit timer/event counter)

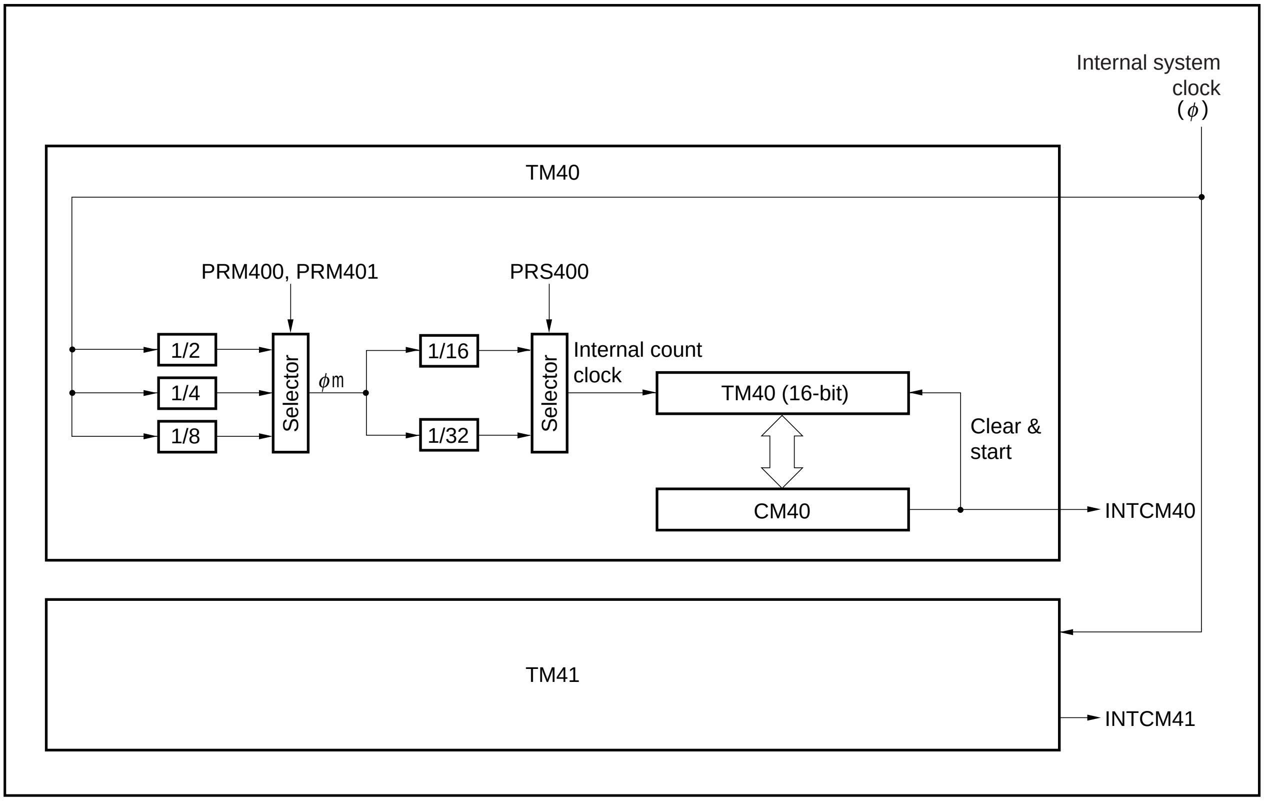


Notes **1.** Internal count clock

2. External count clock

3. Reset priority

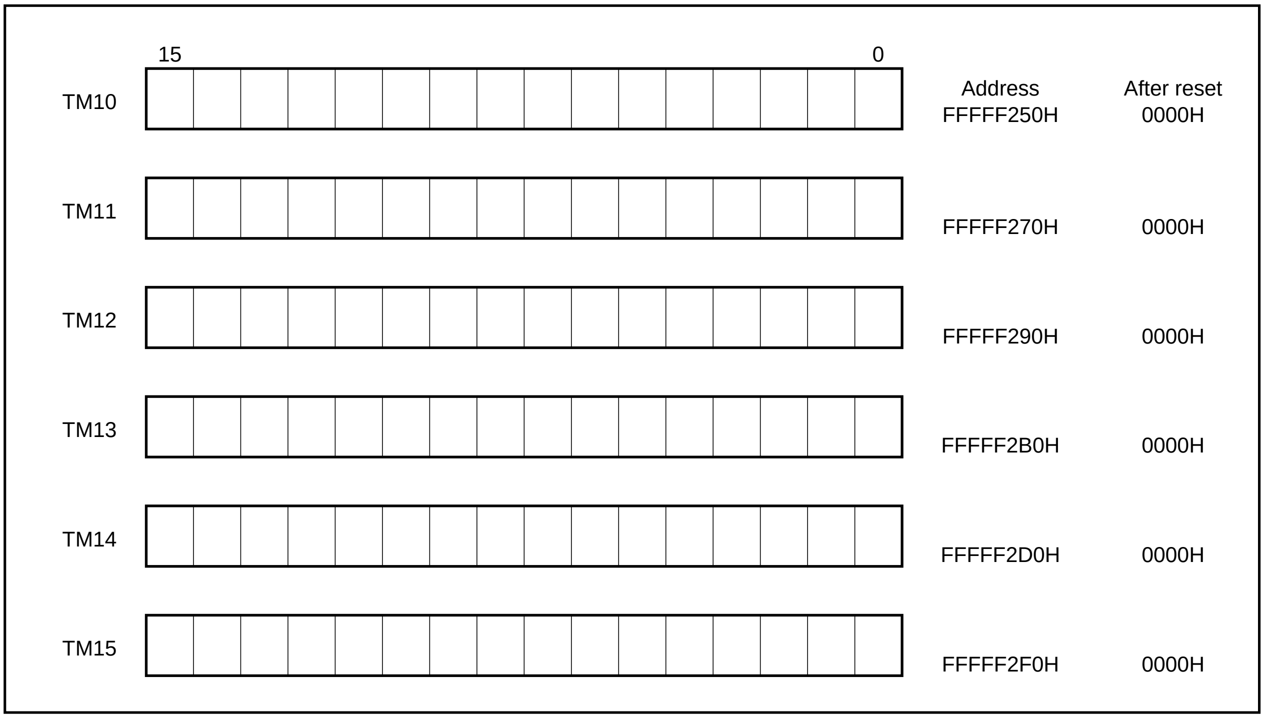
(2) Timer 4 (16-bit interval timer)



9.2.1 Timer 1

(1) Timers 10 to 15 (TM10 to TM15)

TM1n functions as a 16-bit free running timer or as an event counter for an external signal. Mainly, besides period measurement and frequency measurement, it can be used as a pulse output (n = 0 to 5). TM1n is read-only, in 16-bit units.



	Address	After reset
TM10	FFFFF250H	0000H
TM11	FFFFF270H	0000H
TM12	FFFFF290H	0000H
TM13	FFFFF2B0H	0000H
TM14	FFFFF2D0H	0000H
TM15	FFFFF2F0H	0000H

TM1n carries out count-up operations of the internal count clock or of an external count clock. Starting and stopping of the timer is controlled by the CE1n bit of timer control register 1n (TMC1n).

Selection of internal or external count clocks is performed by the TMC1n register.

(a) Selection of an external count clock

TM1n operates as an event counter. The active edge is specified by the timer unit mode register 1n (TUM1n) and through input of pin TI1n, TM1n is counted up.

(b) Selection of an internal count clock

TM1n operates as a free running timer. The counter clock can be selected from among the divisions performed by the prescaler, $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, $\phi/32$, or $\phi/64$, through the TMC1n register.

If the timer overflows, an overflow interrupt can be generated. Also, the timer can be stopped after an overflow through the TUM1n register specification.

The timer can also be cleared and started using the external input TCLR1n. When this is done, the prescaler is cleared at the same time, so the time from TCLR1n input to timer count-up is constant corresponding to the prescaler's dividing ratio. The operation setting is carried out by the TUM1n register.

Caution The count clock cannot be changed during timer operation.

(2) Capture/compare registers 1n0 to 1n3 (CC1n0 to CC1n3) (n = 0 to 5)

The capture/compare registers are 16-bit registers to which TM1n is connected. They can be used as either a capture register or a compare register in accordance with the specification in timer unit mode register 1n (TUM1n). These registers can be read/written in 16-bit units.

	15															0	Address	After reset
CC100 to CC103																	FFFFF252H to FFFFF258H	Undefined
CC110 to CC113																	FFFFF272H to FFFFF278H	Undefined
CC120 to CC123																	FFFFF292H to FFFFF298H	Undefined
CC130 to CC133																	FFFFF2B2H to FFFFF2B8H	Undefined
CC140 to CC143																	FFFFF2D2H to FFFFF2D8H	Undefined
CC150 to CC153																	FFFFF2F2H to FFFFF2F8H	Undefined

(a) Set as a capture register

If set as a capture register, these registers detect the active edge of the corresponding signals in external interrupts INTP1n0 to INTP1n3 as a capture trigger. Timer 1n is synchronized with the capture trigger and latches a count value (capture operation). The capture operation is performed out of synch with the count clock. The latched value is held in the capture register until the next capture operation is performed.

If the capture (latch) timing to the capture register and writing to the register in response to an instruction are in contention, the latter has the priority and the capture operation is disregarded.

Also, specification of the active edge of external interrupts (rising, falling, or both edges) can be selected by the external interrupt mode register (INTM1 to INTM6).

When there is a specification in the capture register, an interrupt is issued when the active edge of INTP1n0 to INTP1n3 signals is detected. When this is done, an interrupt cannot be issued by INTCC1n0 to INTCC1n3, which are the compare register’s matching signals.

(b) Set as a compare register

If set as a compare register, these registers perform a comparison of the timer and register values at each count clock of the timer, and issue an interrupt if the values match.

The compare registers are provided with a set/reset output function. In synch with matching signal generation, the corresponding timer output (TO1n0, TO1n1) is set or reset.

The interrupt source differs with the function of the register.

If specified a compare register, these registers can be made interrupt signals by selecting, through the specification of the TUM1n register, active edge detection of either the INTCC1n0 to INTCC1n3 signals, which are the matching signals, or the INTP1n0 to INTP1n3 signals.

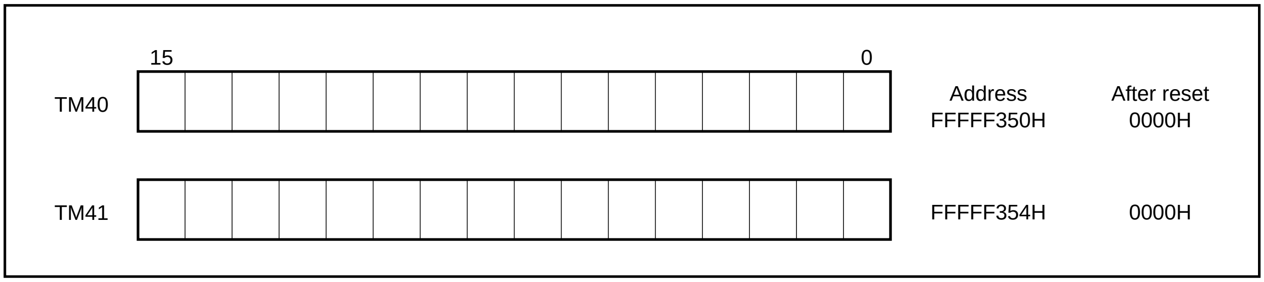
Furthermore, if the INTP1n0 to INTP1n3 signals are selected, acknowledgement of an external interrupt request and timer output by the compare register’s set/reset output function can be carried out in parallel.

9.2.2 Timer 4

(1) Timers 40, 41 (TM40, TM41)

TM4n is a 16-bit timer. It can mainly be used as an interval timer for software (n = 0, 1).

TM4n is read-only in 16-bit units.



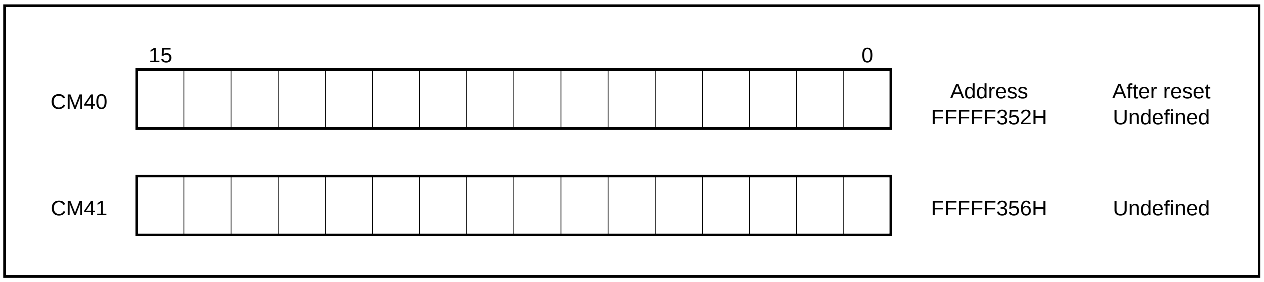
Starting and stopping of TM4n is controlled by the CE4n bit of timer control register 4n (TMC4n).

The count clock can be selected from $\phi/32$, $\phi/64$, $\phi/128$, or $\phi/256$ divisions of the prescaler via register TMC4n.

Caution Since the timer is cleared at the next count clock after a compare match is issued, when the division ratio is large, even if the timer's value is read immediately after the match interrupt is issued, the timer's value may not be 0.
Also, the count clock cannot be changed during timer operation.

(2) Compare registers 40, 41 (CM40, CM41)

CM4n is a 16-bit register and is connected to TM4n. This register can be read/written in 16-bit units.



This register compares TM4n and CM4n each TM4n count clock and if they match, issues an interrupt (INTCM4n). TM4n is cleared in synchronization with this match.

9.3 Control Registers

(1) Timer unit mode registers 10 to 15 (TUM10 to TUM15)

The TUM1n register is a register which controls the operation of timer 1 and specifies the capture/compare register operation mode (n = 0 to 5).

These registers can be read/written in 16-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TUM10	0	0	OST0	ECLR10	TES101	TES100	CES101	CES100	CMS103	CMS102	CMS101	CMS100	IMS103	IMS102	IMS101	IMS100	FFFFF240H	0000H
TUM11	0	0	OST1	ECLR11	TES111	TES110	CES111	CES110	CMS113	CMS112	CMS111	CMS110	IMS113	IMS112	IMS111	IMS110	FFFFF260H	0000H
TUM12	0	0	OST2	ECLR12	TES121	TES120	CES121	CES120	CMS123	CMS122	CMS121	CMS120	IMS123	IMS122	IMS121	IMS120	FFFFF280H	0000H
TUM13	0	0	OST3	ECLR13	TES131	TES130	CES131	CES130	CMS133	CMS132	CMS131	CMS130	IMS133	IMS132	IMS131	IMS130	FFFFF2A0H	0000H
TUM14	0	0	OST4	ECLR14	TES141	TES140	CES141	CES140	CMS143	CMS142	CMS141	CMS140	IMS143	IMS142	IMS141	IMS140	FFFFF2C0H	0000H
TUM15	0	0	OST5	ECLR15	TES151	TES150	CES151	CES150	CMS153	CMS152	CMS151	CMS150	IMS153	IMS152	IMS151	IMS150	FFFFF2E0H	0000H

Bit Position	Bit Name	Function
13	OSTn	Overflow Stop Specifies the timer's operation after overflow. This flag is valid only in TM1n. 0: Timer continues to count up after timer overflow. 1: Timer holds 0000H and is in the stopped state after timer overflow. When this happens, the CE1 bit in the TMC1n register remains at 1. Counting up resumes with the next operation. When ECLR1n = 0: 1 write operation to the CE1n bit. When ECLR1n = 1: Trigger input to the timer clear pin (TCLR1n).
12	ECLR1n	External Input Timer Clear Clearing of the timer is enabled by the TM1n external clear input (TCLR1n). 0: Timer is not cleared by an external input. 1: TM1n is cleared by an external input. Counting up starts after clearing.

Remark n = 0 to 5

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>11, 10</td><td>TES1n1,
TES1n0</td><td>TI1n Edge Select
Specifies the active edge of the external clock input (TI1n).
<table>
<tr><th>TES1n1</th><th>TES1n0</th><th>Active Edge</th></tr>
<tr><td>0</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>RFU (reserved)</td></tr>
<tr><td>1</td><td>1</td><td>Both the rising and falling edges</td></tr>
</table></td></tr>
<tr><td>9, 8</td><td>CES1n1,
CES1n0</td><td>TCLR1n Edge Select
Specifies the active edge of the external clear input (TCLR1n).
<table>
<tr><th>CES1n1</th><th>CES1n0</th><th>Active Edge</th></tr>
<tr><td>0</td><td>0</td><td>Falling edge</td></tr>
<tr><td>0</td><td>1</td><td>Rising edge</td></tr>
<tr><td>1</td><td>0</td><td>RFU (reserved)</td></tr>
<tr><td>1</td><td>1</td><td>Both the rising and falling edges</td></tr>
</table></td></tr>
<tr><td>7 to 4</td><td>CMS1nm
(m = 3 to 0)</td><td>Capture/Compare Mode Select
Selects the capture/compare register's (CC1nm) operation mode.
0: Operates as a capture register. However, the capture operation when it is specified as a capture register is performed only when the CE1n bit of the TMC1n register = 1.
1: Operates as a compare register.</td></tr>
<tr><td>3 to 0</td><td>IMS1nm
(m = 3 to 0)</td><td>Interrupt Mode Select
Selects either INTP1nm or INTCC1nm as the interrupt source.
0: Makes the compare register's matching signal INTCC1nm the interrupt request signal.
1: It makes the external input signal INTP1nm the interrupt request signal.</td></tr>
</table>

Remark n = 0 to 5

Remarks **1.** If the A/D converter is set in the timer trigger mode, the compare register’s match interrupt becomes the A/D conversion start trigger, starting the conversion operation. When this happens, the compare register’s match interrupt functions as a compare register match interrupt to the CPU. In order for a compare register match interrupt not to be issued to the CPU, disable interrupts with the interrupt mask bits (P11MK0 to P11MK3) of the interrupt control register (P11IC0 to P11IC3).

2. If the A/D converter is set in the external trigger mode, the external trigger input becomes the A/D converter starting trigger, starting the conversion operation. When this happens, the external trigger input also functions as Timer 1’s capture trigger and as an external interrupt. In order for it not to issue capture triggers or external interrupts, set Timer 1 in the compare register and disable interrupts with the interrupt control register’s interrupt mask bit.

If Timer 1 is not set in the compare register, and if interrupts are not disabled in the interrupt control register, the following will happen.

(a) If the TUM15 register’s interrupt mask bit (IMS153) is 0

It also functions as the compare register’s match interrupt with respect to the CPU.

(b) If the TUM15 register’s interrupt mask bit (IMS153) is 1

The A/D converter’s external trigger input also functions as an external interrupt to the CPU.

(2) Timer control registers 10 to 15 (TMC10 to TMC15)

TMC10 to 15 control the respective operations of TM10 to TM15.

These registers can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
TMC10	CE10	0	0	ETI10	PRS101	PRS100	PRM101	0	FFFFF242H	00H
TMC11	CE11	0	0	ETI11	PRS111	PRS110	PRM111	0	FFFFF262H	00H
TMC12	CE12	0	0	ETI12	PRS121	PRS120	PRM121	0	FFFFF282H	00H
TMC13	CE13	0	0	ETI13	PRS131	PRS130	PRM131	0	FFFFF2A2H	00H
TMC14	CE14	0	0	ETI14	PRS141	PRS140	PRM141	0	FFFFF2C2H	00H
TMC15	CE15	0	0	ETI15	PRS151	PRS150	PRM151	0	FFFFF2E2H	00H

Bit Position	Bit Name	Function
7	CE1n	Count Enable Controls timer operation. 0: The timer is stopped in the 0000H state and does not operate. 1: The timer performs a count operation. However, when the ECLR1n bit of the TUM1n register is 1, the timer does not start counting up until there is a TCLR1n input. When the ECLR1n bit is 0, the operation of setting (1) in the CE1n bit becomes the count start trigger. Thus, after the CE1n bit is set (1) when the ECLR1n bit = 1, the timer will not start even if the ECLR1n bit is made 0.
4	ETI1n	External TI1n Input Specifies whether switching of the count clock is external or internal. 0: Specifies the ϕ system (internal). 1: Specifies TI1n (external).

Caution Do not change the count clock during timer operation.

Remark n = 0 to 5

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>3, 2</td><td>PRS1n1,
PRS1n0</td><td>Prescaler Clock Select
Selects the internal count clock (ϕm is the intermediate clock).
<table>
<tr><th>PRS1n1</th><th>PRS1n0</th><th>Internal Count Clock</th></tr>
<tr><td>0</td><td>0</td><td>ϕm</td></tr>
<tr><td>0</td><td>1</td><td>ϕm/4</td></tr>
<tr><td>1</td><td>0</td><td>ϕm/8</td></tr>
<tr><td>1</td><td>1</td><td>ϕm/16</td></tr>
</table>
</td></tr>
<tr><td>1</td><td>PRM1n1</td><td>Prescaler Clock Mode
Selects the intermediate count clock (ϕm). (ϕ is the internal system clock).
0: ϕ/2
1: ϕ/4</td></tr>
</table>

Caution Do not change the count clock during timer operation.

Remark n = 0 to 5

(3) Timer control registers 40, 41 (TMC40, TMC41)

TMC40 and TMC41 control the operation of TM40 and TM41, respectively.

These registers can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
TMC40	CE40	0	0	0	0	PRS400	PRM401	PRM400	FFFFF342H	00H
TMC41	CE41	0	0	0	0	PRS410	PRM411	PRM410	FFFFF346H	00H

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7</td><td>CE4n</td><td>Count Enable
Controls timer operations.
0: The timer is stopped in the 0000H state and does not operate.
1: The timer performs a count operation.</td></tr>
<tr><td>2</td><td>PRS4n0</td><td>Prescaler Clock Select
Selects the internal count clock (ϕm is the intermediate clock).
0: ϕm/16
1: ϕm/32</td></tr>
<tr><td>1, 0</td><td>PRM4n1,
PRM4n0</td><td>Prescaler Clock Mode
Selects the intermediate count clock ((ϕm). (ϕ is the internal system clock).
<table>
<tr><th>PRM4n1</th><th>PRM4n0</th><th>ϕm</th></tr>
<tr><td>0</td><td>0</td><td>ϕ/2</td></tr>
<tr><td>0</td><td>1</td><td>ϕ/4</td></tr>
<tr><td>1</td><td>0</td><td>ϕ/8</td></tr>
<tr><td>1</td><td>1</td><td>RFU (reserved)</td></tr>
</table>
</td></tr>
</table>

Caution Do not change the count clock during timer operation.

Remark n = 0, 1

(4) Timer output control registers 10 to 15 (TOC10 to TOC15)

The TOC1n register controls the timer output from the TO1n0 and TO1n1 pins (n = 0 to 5).

These registers can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
TOC10	ENTO101	ALV101	ENTO100	ALV100	0	0	0	0	FFFFF244H	00H
TOC11	ENTO111	ALV111	ENTO110	ALV110	0	0	0	0	FFFFF264H	00H
TOC12	ENTO121	ALV121	ENTO120	ALV120	0	0	0	0	FFFFF284H	00H
TOC13	ENTO131	ALV131	ENTO130	ALV130	0	0	0	0	FFFFF2A4H	00H
TOC14	ENTO141	ALV141	ENTO140	ALV140	0	0	0	0	FFFFF2C4H	00H
TOC15	ENTO151	ALV151	ENTO150	ALV150	0	0	0	0	FFFFF2E4H	00H

Bit Position	Bit Name	Function
7, 5	ENTO1n1, ENTO1n0	Enable TO pin Enables output of each corresponding timer (TO1n0, TO1n1). 0: Timer output is disabled. The reverse phase level (inactive level) of the ALV1n0 and ALV1n1 bits is output from the TO1n0 and TO1n1 pins. Even if a match signal is generated by the corresponding compare register, the level of the TO1n0 and TO1n1 pins does not change. 1: Timer output is enabled. If a match signal is generated from the corresponding compare register, the timer's output changes. From the timer that timer output is enabled until match signals are first generated, the reverse phase level (inactive level) of the ALV1n0 and ALV1n1 bits is output.
6, 4	ALV1n1, ALV1n0	Active Level TO pin Specifies the timer output's active level. 0: The active level is the low level. 1: The active level is the high level.

Remarks 1. The TO1n0 and TO1n1 output flip-flop is reset priority.

2. n = 0 to 5

Caution The TO1n0 and TO1n1 output is not changed by an external interrupt signal (INTP1n0 to INTP1n3). When the TO1n0 and TO1n1 signals are used, specify the capture/compare register as the compare register (CMS1n0 to CMS1n3 bit of the TUM1n register = 1).

(5) External interrupt mode registers 1 to 6 (INTM1 to INTM6)

If CC1n0 to CC1n3 of TM1n are used as a capture register, the active edge of the external interrupt INTP1n0 to INTP1n3 signals is detected as a capture trigger (for details, refer to **CHAPTER 7 INTERRUPT/EXCEPTION PROCESSING FUNCTION**) (n = 0 to 5).

(6) Timer overflow status register (TOVS)

This interrupts overflow flags from TM10 to TM15, TM40, and TM41.

The register can be read/written in 8- or 1-bit units.

By setting and resetting the TOVS register through software, polling of overflow occurrences can be accomplished.

	7	6	5	4	3	2	1	0	Address	After reset
TOVS	OVF41	OVF40	OVF15	OVF14	OVF13	OVF12	OVF11	OVF10	FFFFF230H	00H

Bit Position	Bit Name	Function
7 to 0	OVF41, OVF40, OVF15 to OVF10	Overflow Flag This is the overflow flag for TM41, TM40 and TM1n. 0: No overflow is generated. 1: Overflow is generated. **Caution Interrupt requests (INTOV1n) for the interrupt controller are generated in synch with an overflow from TM1n, but because interrupt operations and the TOVS register are independent, the overflow flag (OVF1n) from TM1n can be operated by software just like other overflow flags.** **At this time, the interrupt request flag (OVF1n) corresponding to INTOV1n is not affected.** During CPU access interval, transfers to the TOVS register cannot be made. Therefore, even if an overflow is generated during a readout from the TOVS register, the flag's value does not change and it is reflected in the next read operation.

Remark n = 0 to 5

9.4 Timer 1 Operation

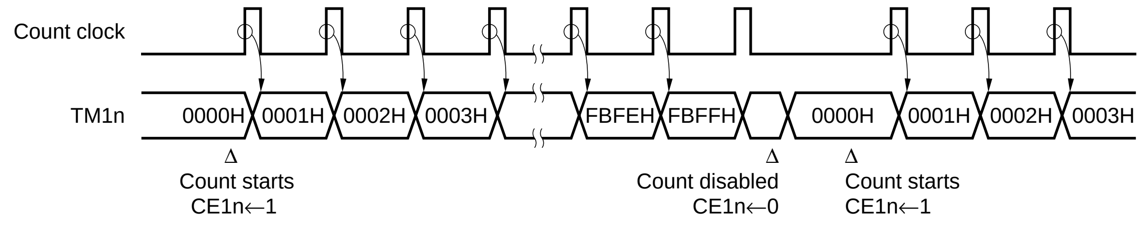
9.4.1 Count operation

Timer 1 functions as a 16-bit free-running timer or an event counter for an external signal.

Whether the timer operates as a free-running timer or event counter is specified by timer control register 1n (TMC1n) (n = 0 to 5).

When it is used as a free-running timer, and when the count values of TM1n match with the value of any of the CC1n0 to CC1n3 registers, an interrupt signal is generated, and timer output signal TO1n0 and TO1n1 can be set/reset. In addition, a capture operation that holds the current count value of TM1n and loads it into one of the four registers CC1n0 to CC1n3, is performed in synchronization with the valid edge detected from the corresponding external interrupt request pin as an external trigger. The captured value is retained until the next capture trigger is generated.

Figure 9-1. Basic Operation of Timer 1



Remark n = 0 to 5

9.4.2 Count clock selection

The count clock input to Timer 1 is either internal or external, and can be selected by the ETI1n bit in the TMC1n register (n = 0 to 5).

Caution Do not change the count clock during timer operation.

(1) Internal count clock (ETI1n bit = 0)

An internal count clock can be selected from among 6 possible clock rates, ϕ/2, ϕ/4, ϕ/8, ϕ/16, ϕ/32, or ϕ/64, by the setting of the PRS1n1, PRS1n0, and PRM1n1 bits of the TMC1n register.

PRS1n1	PRS1n0	PRM1n1	Internal Count Clock
0	0	0	ϕ/2
0	0	1	ϕ/4
0	1	0	ϕ/8
0	1	1	ϕ/16
1	0	0	ϕ/16
1	0	1	ϕ/32
1	1	0	ϕ/32
1	1	1	ϕ/64

Remark n = 0 to 5

(2) External count clock (ETI1n bit = 1)

This counts the signals input to the TI1n pin. At this time, Timer 1 can be operated as an event counter.
The TI1n active edge can be set by the TES1n1 and TES1n0 bits of the TUM1n register.

TES1n1	TES1n0	Active Edge
0	0	Rising edge
0	1	Falling edge
1	0	RFU (reserved)
1	1	Both the rising and falling edges

Remark n = 0 to 5

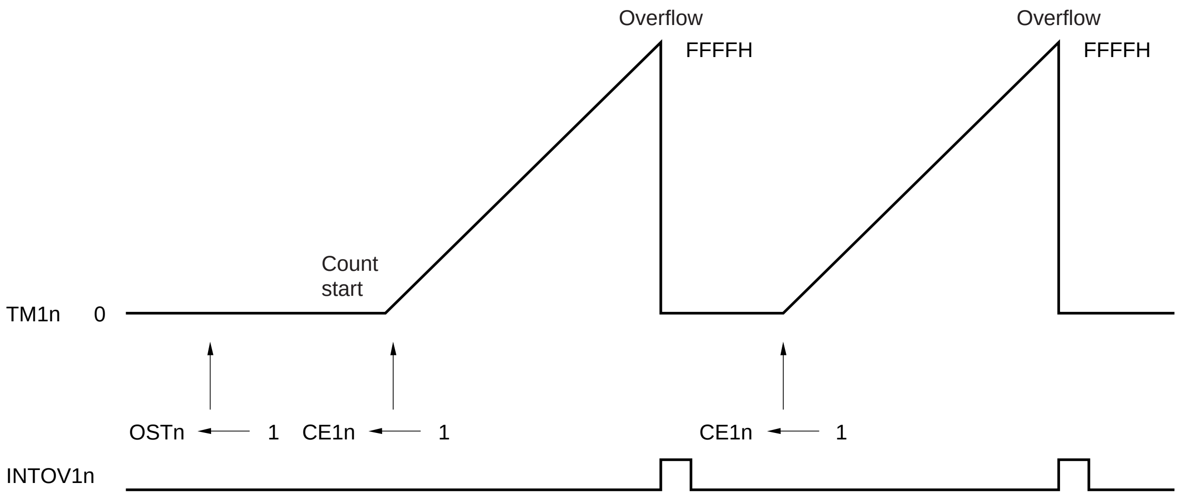
9.4.3 Overflow

When the TM1n register counts the count clock to FFFFH and overflow occurs as a result, a flag is set in the OVF1n bit of the TOVS register and an overflow interrupt (INTOV1n) is generated (n = 0 to 5).

Also, by setting the OSTn bit (1) in the TUM1n register, the timer can be stopped after overflow. If the timer is stopped due to an overflow, the count operation does not resume until the CE1n bit in the TMC1n register is set (1).

Note that even if the CE1n bit is set (1) during a count operation, it has no influence on operation.

Figure 9-2. Operation after Overflow (If ECLR1n = 0 and OSTn = 1)



Remark n = 0 to 5

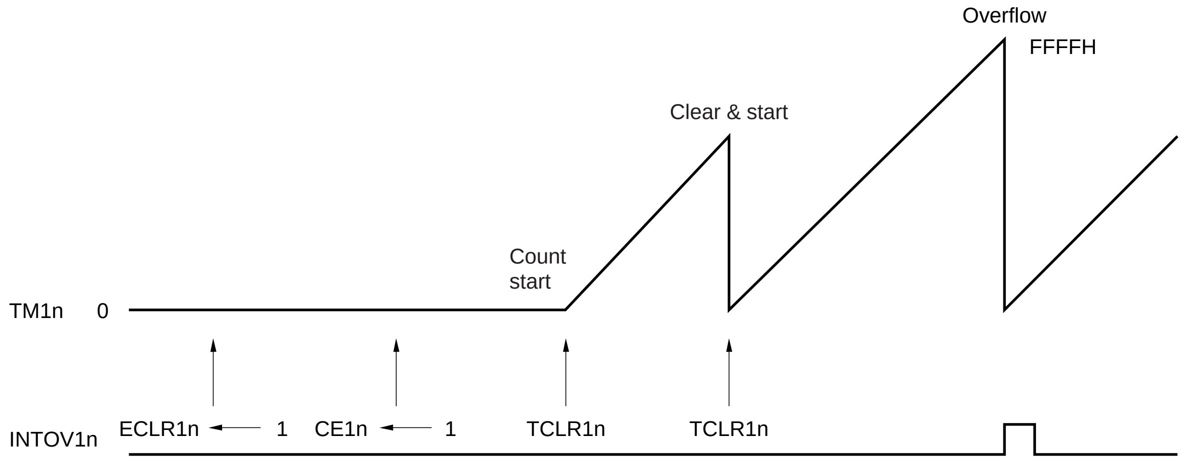
9.4.4 Clearing/starting timer by TCLR1n signal input

Timer 1 ordinarily starts a counting operation when the CE1n bit in the TMC1n register is set (1), but TM1n can be cleared and a count operation started by input of the TCLR1n signal (n = 0 to 5).

If the ECLR1n bit of the TUM1n register is set to 1, and the OSTn bit is set to 0, if the active edge is input to the TCLR1n signal after the CE1n bit is set (1), the counting operation starts. Also, if the active edge is input to the TCLR1n signal during operation, the TM1n's value is cleared and the count operation resumes (refer to **Figure 9-3**).

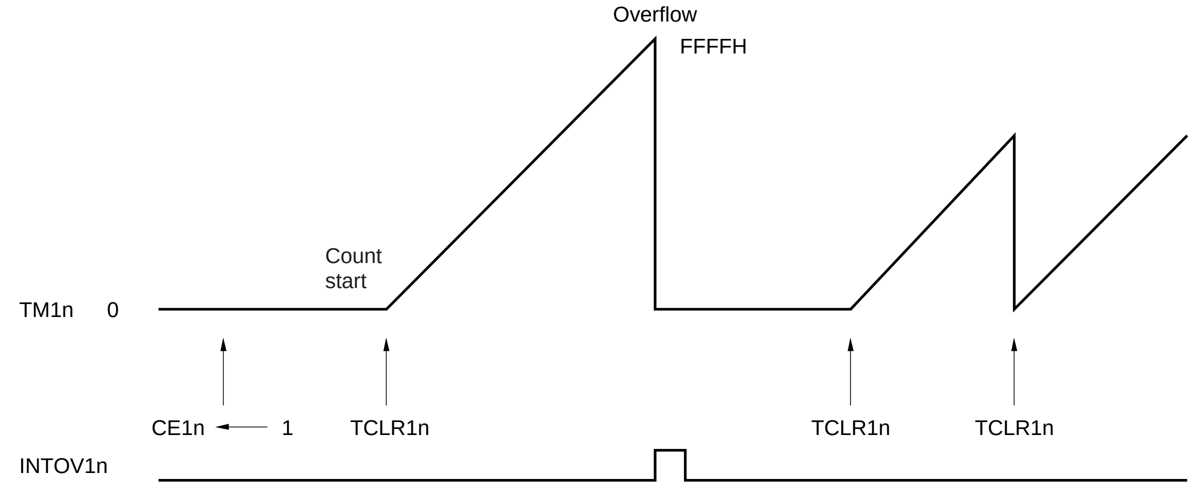
If the ECLR1n bit of the TUM1n register is set to 1, and the OSTn bit is set to 1, the counting operation starts if the active edge is input to the TCLR1n signal after the CE1n bit is set (1). If TM1n overflows, the count operation stops once and it does not resume the count operation until the active edge is input again to the TCLR1n signal. If the active edge of the TCLR1n signal is detected during a counting operation, TM1n is cleared and the count operation continues (refer to **Figure 9-4**). Note that if the CE1n bit is set (1) after an overflow, the count operation does not resume.

Figure 9-3. Timer Clear/Start Operation by TCLR1n Signal Input (If ECLR1n = 1 and OSTn = 0)



Remark n = 0 to 5

Figure 9-4. Relationship Between Clear/Start by TCLR1n Signal Input and Overflow Operation (If ECLR1n = 1 and OSTn = 1)



Remark n = 0 to 5

9.4.5 Capture operation

In synch with an external trigger, a capture operation is performed in which the TM1n count value is captured and held in the capture register asynchronous to the count clock (n = 0 to 5). The active edge detected from external interrupt request input pins INTP1n0 to INTP1n3 is used as the external trigger (capture trigger). In synch with that capture trigger signal, the count value of TM1n, as it is counting, is captured and held in the capture register. The value in the capture register is held until the next capture trigger is generated.

Also, interrupt requests (INTCC1n0 to INTCC1n3) are generated from the INTP1n0 to INTP1n3 signal inputs.

Table 9-2. Capture Trigger Signals (TM1n) to 16-Bit Capture Registers

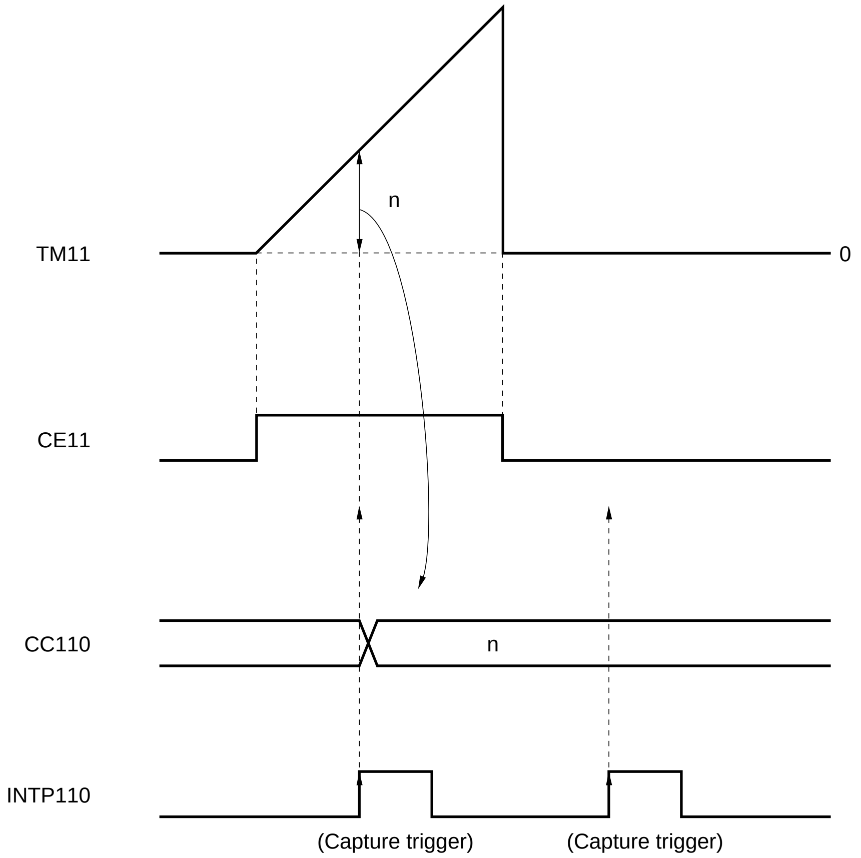
Capture Register	Capture Trigger Signal
CC1n0	INTP1n0
CC1n1	INTP1n1
CC1n2	INTP1n2
CC1n3	INTP1n3

Remarks 1. CC1n0 to CC1n3 are the capture/compare registers. Which register is used is specified in timer unit mode register 1n (TUM1n).

2. n = 0 to 5

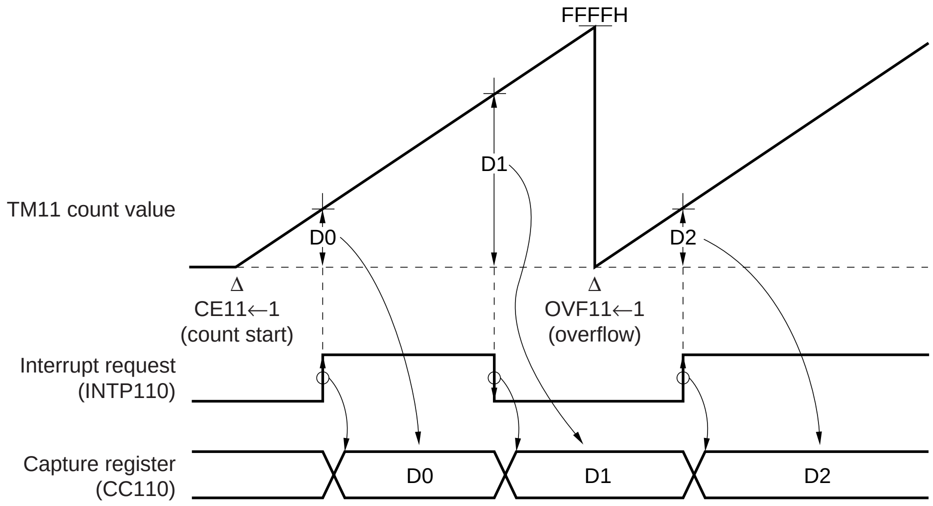
The capture trigger's active edge is set by the external interrupt mode register (INTM1 to INTM6). If both the rising and falling edges are made capture triggers, the input pulse width from an external source can be measured. Also, if the edge from one side is used as the capture trigger, the input pulse's period can be measured.

Figure 9-5. Example of Capture Operation



Remark When the CE11 bit = 0, no capture operation is performed even if INTP110 is input.

Figure 9-6. Example of TM11 Capture Operation (When Both Edges Are Specified)



Remark D0 to D2: TM11 count value

9.4.6 Compare operation

Compare operations in which the value set in the compare register is compared with the TM1n count value are performed (n = 0 to 5).

If the TM1n count value matches the value that has been previously set in the compare register, a match signal is sent to the output control circuit (refer to **Figure 9-7**). The timer output pins (TO1n0, TO1n1) are changed by the match signal and simultaneously issue interrupt request signals.

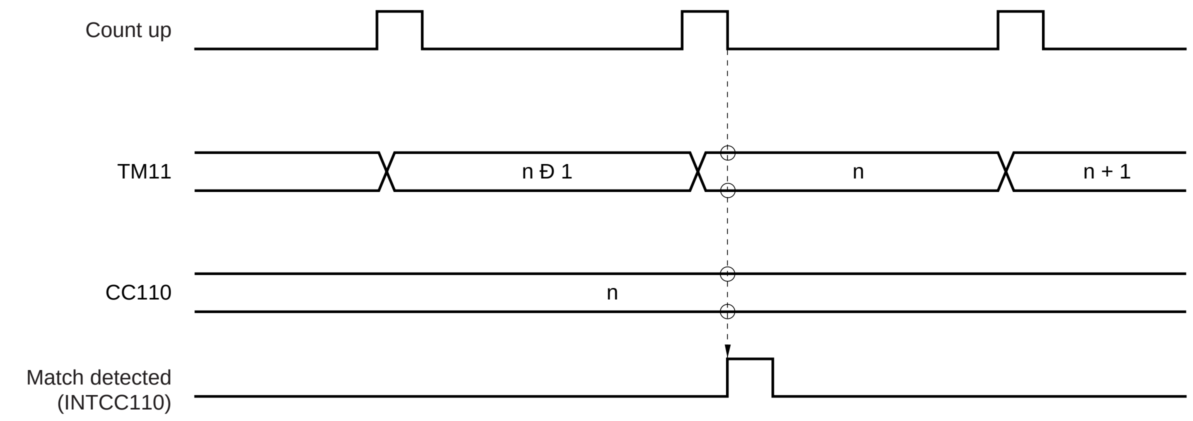
Table 9-3. Interrupt Request Signals (TM1n) from 16-Bit Compare Registers

Compare Register	Interrupt Request Signal
CC1n0	INTCC1n0
CC1n1	INTCC1n1
CC1n2	INTCC1n2
CC1n3	INTCC1n3

Remarks 1. CC1n0 to CC1n3 are capture/compare registers. Which register will be used is specified by the timer unit mode register 1n (TUM1n).

2. n = 0 to 5

Figure 9-7. Example of Compare Operation



Remark A Match is detected immediately after counting up, then a match detection signal is generated.

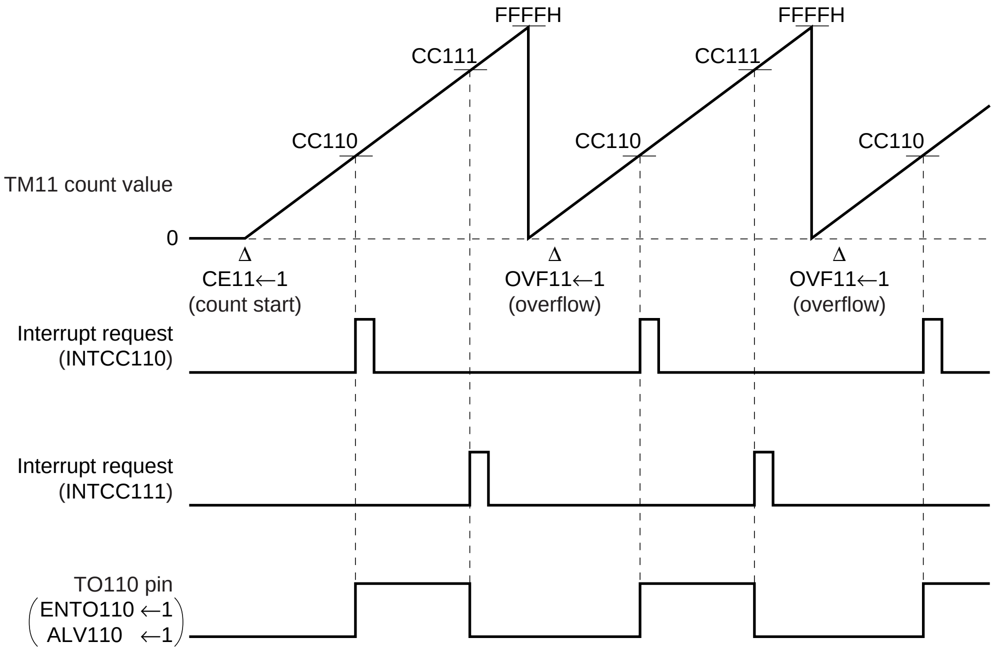
Timer 1 has 12 timer output pins (TO1n0, TO1n1).

The TM1n count value and the CC1n0 value are compared and if they match, the output level of the TO1n0 pin is set. Also, the TM1n count value and the CC1n1 value are compared, and if they match, the TO1n0 pin's output level is reset.

In the same way, the TM1n count value and the CC1n2 value are compared, and if they match, the TO1n1 pin's output level is set. Also, the TM1n counter value and the CC1n3 value are compared, and if they match, the TO1n1 pin's output level is set.

The output level of pins TO1n0 and TO1n1 can also be specified by the TOC1n register.

Figure 9-8. Example of TM11 Compare Operation (Set/Reset Output Mode)



9.5 Timer 4 Operation

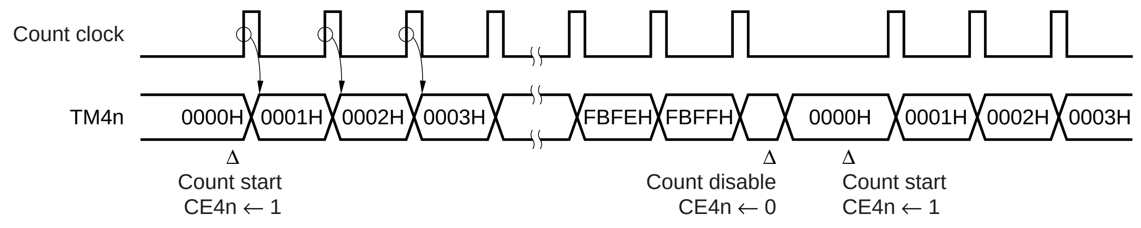
9.5.1 Count operation

Timer 4 functions as a 16-bit interval timer. Setting of its operation is specified in timer control register 4n (TMC4n) (n = 0, 1).

In a timer 4 count operation, the internal count clock (ϕ/32 to ϕ/256) specified by the PRS4n0, PRM4n1, and PRM4n0 bits of the TMC4n register is counted up.

If the count results in TM4n match the value in CM4n, TM4n is cleared. At the same time, a matching interrupt (INTCM4n) is generated.

Figure 9-9. Basic Operation of Timer 4



Remark n = 0, 1

9.5.2 Count clock selection

Using the setting of the TMC4n register's PRS4n0, PRM4n1, and PRM4n0 bits, one of four possible internal count clocks, ϕ/32, ϕ/64, ϕ/128 or ϕ/256, can be selected (n = 0, 1).

Caution Do not change the count clock during timer operation.

PRS4n0	PRM4n1	PRM4n0	Internal Count Clock
0	0	0	ϕ/32
0	0	1	ϕ/64
0	1	0	ϕ/128
0	1	1	RFU (reserved)
1	0	0	ϕ/64
1	0	1	ϕ/128
1	1	0	ϕ/256
1	1	1	RFU (reserved)

Remark n = 0, 1

9.5.3 Overflow

If the TM4n overflows as a result of counting the internal count clock, the OVF4n bit of the TOVS register is set (1) (n = 0, 1).

9.5.4 Compare operation

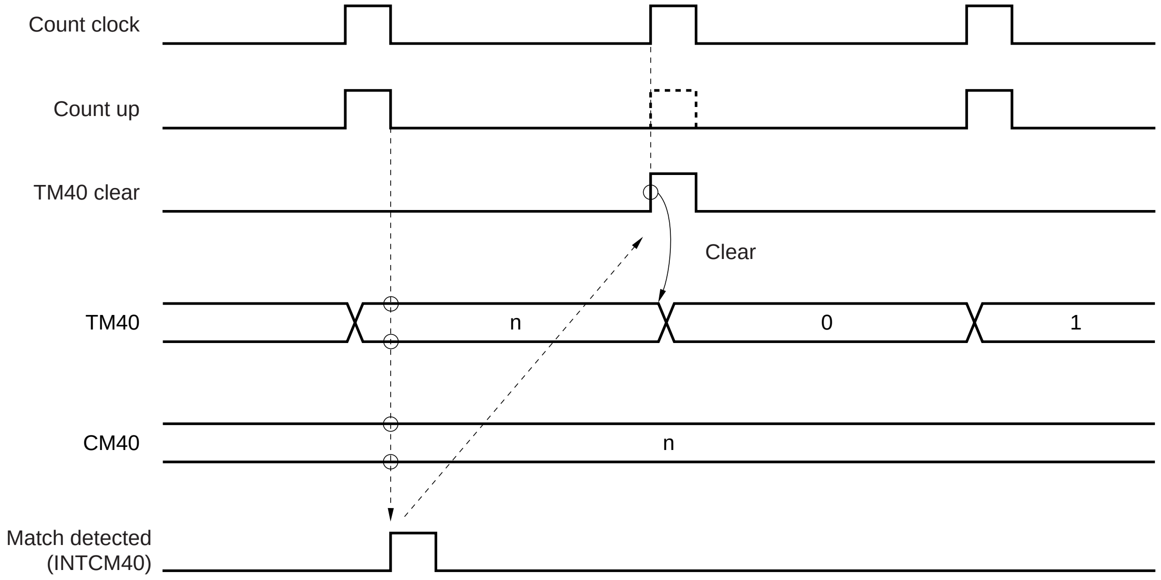
In Timer 4, a compare operation which compares the value set in the compare register (CM4n) with the TM4n count value is performed (n = 0, 1).

If values are found to match in the compare operation, an interrupt (INTCM4n) is issued. By issuing an interrupt, TM4n is cleared (0) with the following timing (refer to **Figure 9-10 (a)**). Through this function, Timer 4 is used as an interval timer.

CM4n can also be set to 0. In this case, if TM4n overflows and becomes 0, a value match is detected and INTCM4n is issued. Using the following count timing, the TM4n value is cleared (0), but with this match, INTCM4n is not issued (refer to **Figure 9-10 (b)**).

Figure 9-10. Example of TM40 Compare Operation (1/2)

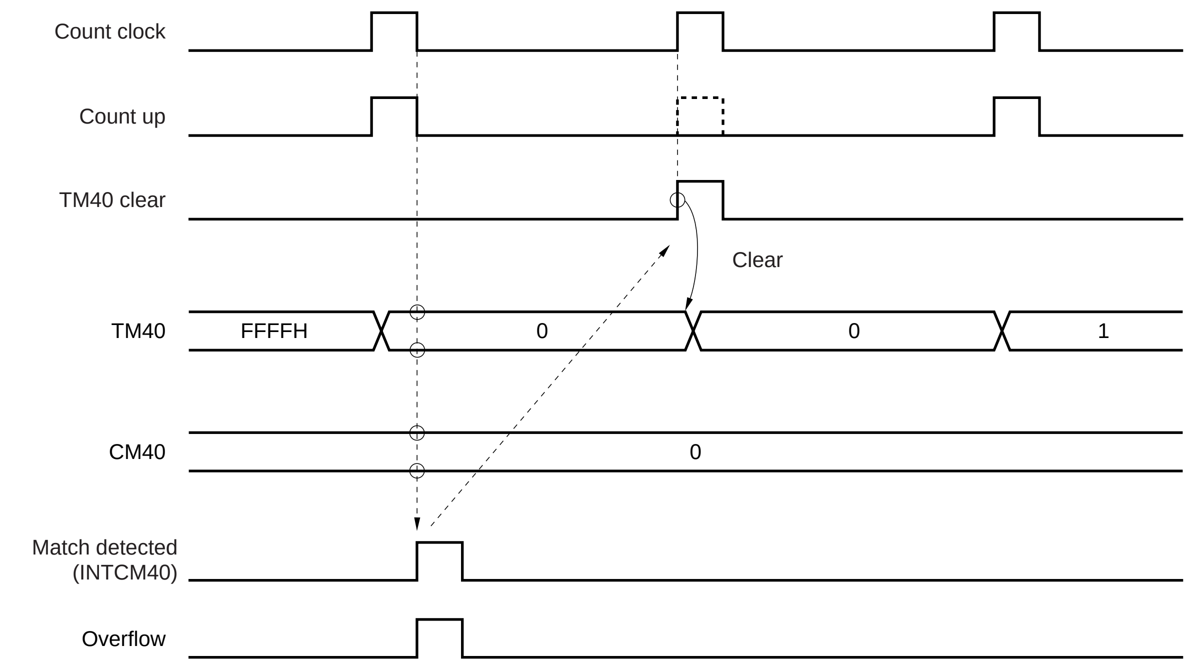
(a) If FFFFH is set in CM40



Remark Interval time = (n + 1) × count clock cycle
n = 1 to 65,535 (FFFFH)

Figure 9-10. Example of TM40 Compare Operation (2/2)

(b) If 0 is set in CM40



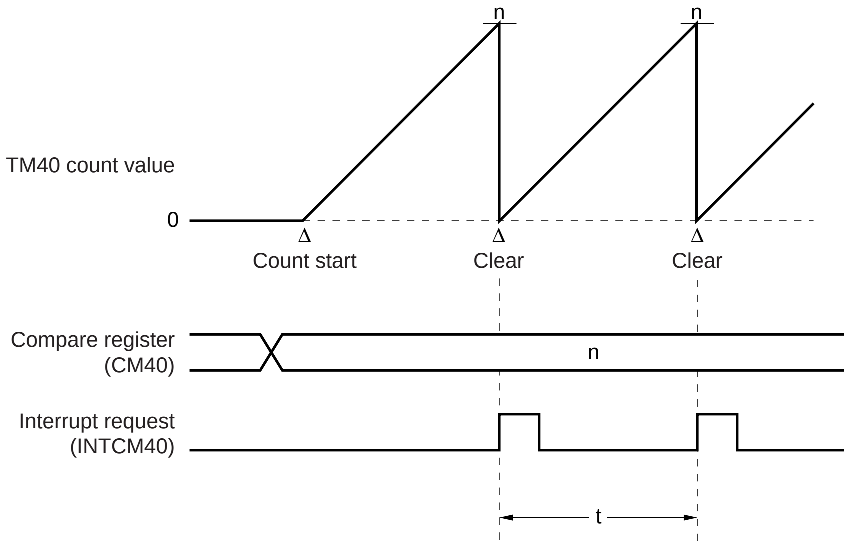
Remark Interval time = (FFFFH + 1) × count clock cycle

9.6 Application Example

(1) Operation as an interval timer (Timer 4)

In this example, timer 4 is used as an interval timer that repeatedly issues an interrupt at intervals specified by the count time preset in the compare register (CM4n) (n = 0, 1).

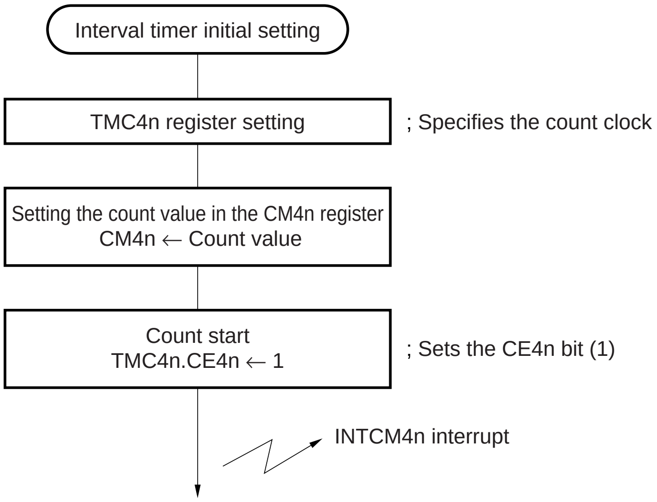
Figure 9-11. Example of Timing in Interval Timer Operation



Remark n: Value in the CM40 register

t: Interval time = (n + 1) × count clock cycle

Figure 9-12. Example of Interval Timer Operation Setting Procedure



Remark n = 0, 1

(2) Operation for pulse width measurement (Timer 1)

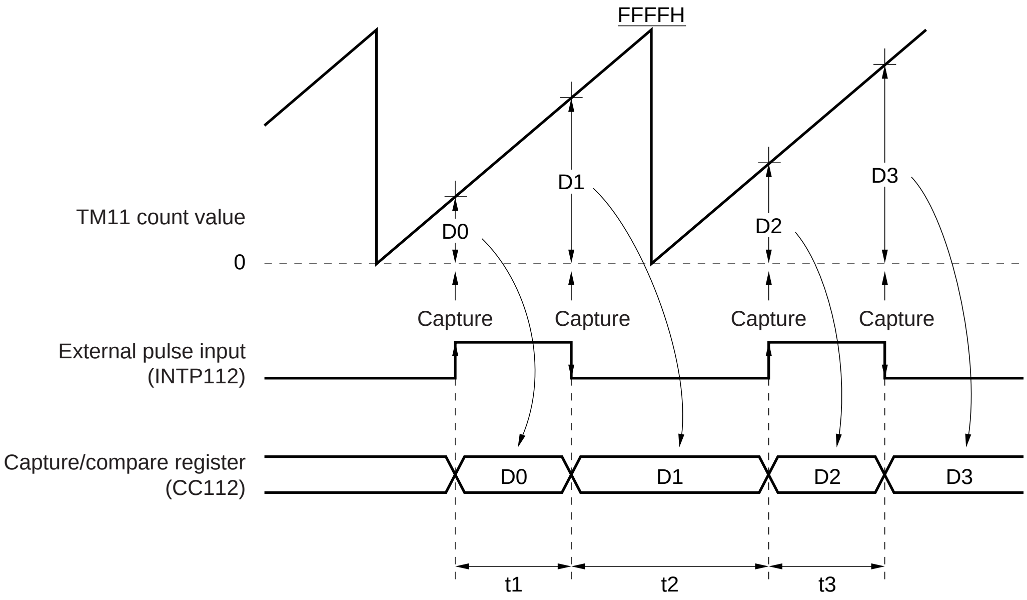
In measuring the pulse width, timer 1 is used.

Here, an example is given of measurement of high level or low level width of an external pulse input to the INTP112 pin.

As shown in Figure 9-13, in synch with the active edge (specified as both the rising edge and falling edge) of the INTP112 pin's input, the value of the counting timer 1 (TM11) is fetched to and held in the capture/compare register (CC112).

The pulse width is calculated by determining the difference between the count value of TM11 captured in the CC112 register through active edge detection the nth time and the count value (Dn – 1) captured through active edge detection the (n – 1)th time, then multiplying this value by the count clock.

Figure 9-13. Example of Pulse Measurement Timing



t1 = (D1 – D0) × count clock cycle

t2 = {(10000H – D1) + D2} × count clock cycle

t3 = (D3 – D2) × count clock cycle

Remark D0 to D3: TM11 count values

Figure 9-14. Example of Pulse Width Measurement Setting Procedure

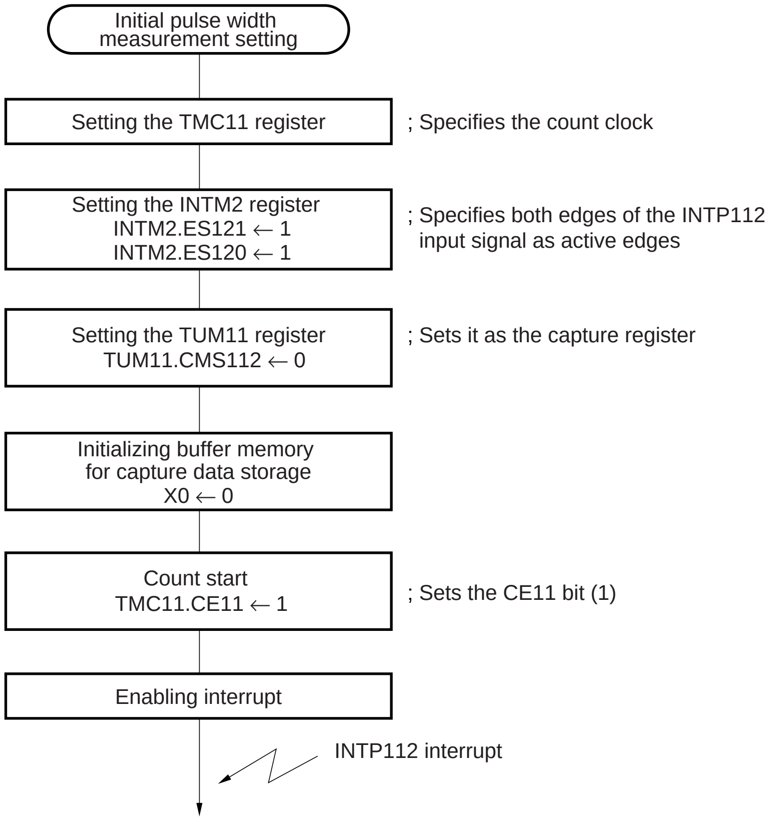
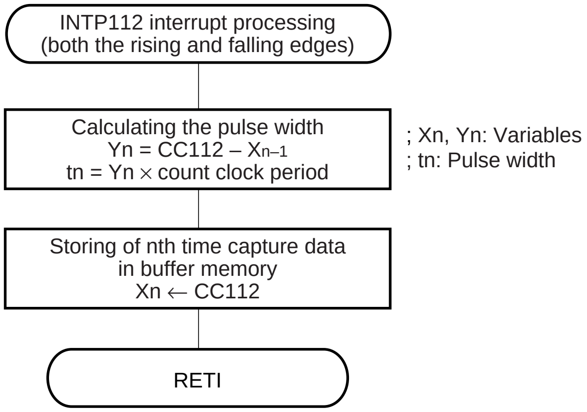

Initial pulse width measurement setting
Setting the TMC11 register
; Specifies the count clock
Setting the INTM2 register
INTM2.ES121 ← 1
INTM2.ES120 ← 1
; Specifies both edges of the INTP112 input signal as active edges
Setting the TUM11 register
TUM11.CMS112 ← 0
; Sets it as the capture register
Initializing buffer memory for capture data storage
X0 ← 0
Count start
TMC11.CE11 ← 1
; Sets the CE11 bit (1)
Enabling interrupt
INTP112 interrupt

Figure 9-15. Example of Interrupt Request Processing Routine Which Calculates the Pulse Width


INTP112 interrupt processing (both the rising and falling edges)
Calculating the pulse width
Yn = CC112 – Xn–1
tn = Yn × count clock period
; Xn, Yn: Variables
; tn: Pulse width
Storing of nth time capture data in buffer memory
Xn ← CC112
RETI

Caution If 2 or more overflows occur between the (n – 1)th capture and the nth capture, the pulse width cannot be measured.

(3) Operation as a PWM output (Timer 1)

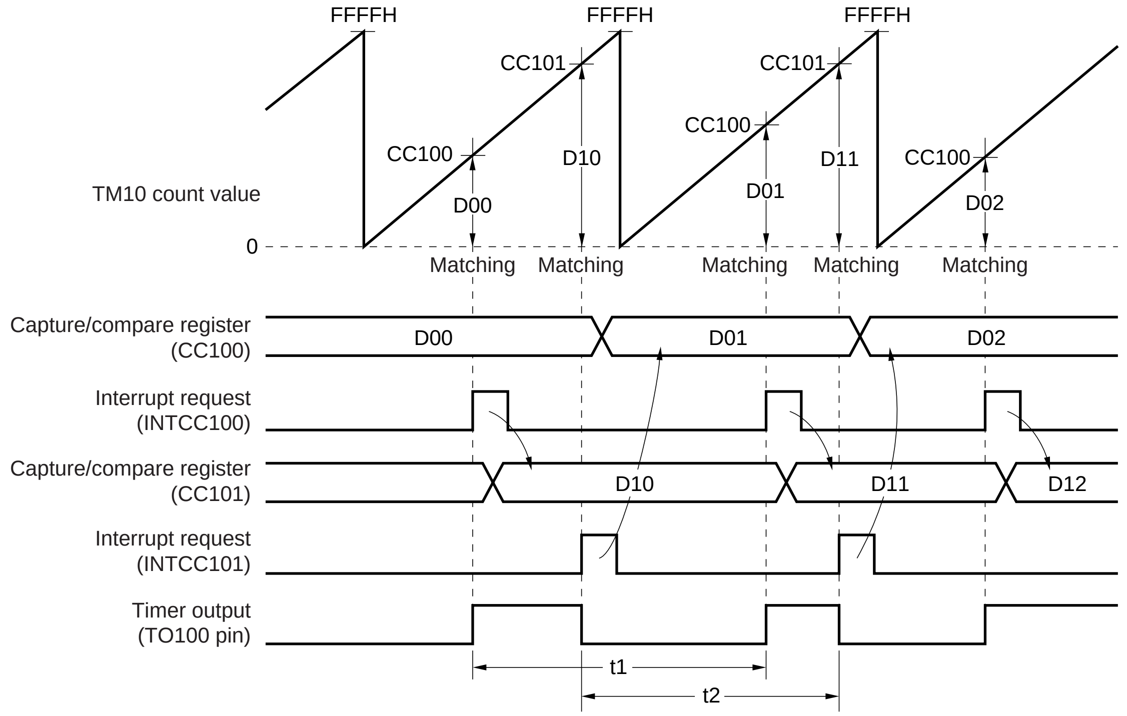
Through a combination of timer 1 and the timer output function, the desired rectangular wave can be output to the timer output pins (TO1n0, TO1n1) and used as a PWM output (n = 0 to 5).

Here an example is shown using the capture/compare registers CC100 and CC101.

In this case, a PWM signal with 16-bit precision can be output from the TO100 pin. The timing is shown in Figure 9-16.

If used as a 16-bit timer, the PWM output's rise timing set in the capture/compare register (CC100) is determined as shown in Figure 9-16, and the fall timing is determined by the value set in the capture/compare register (CC101).

Figure 9-16. Example of PWM Output Timing



Remark D00 to D02, D10 to D12: Compare register setting values

t1 = {(10000H – D00) + D01} × count clock period

t2 = {(10000H – D10) + D11} × count clock period

Figure 9-17. Example of PWM Output Setting Procedure

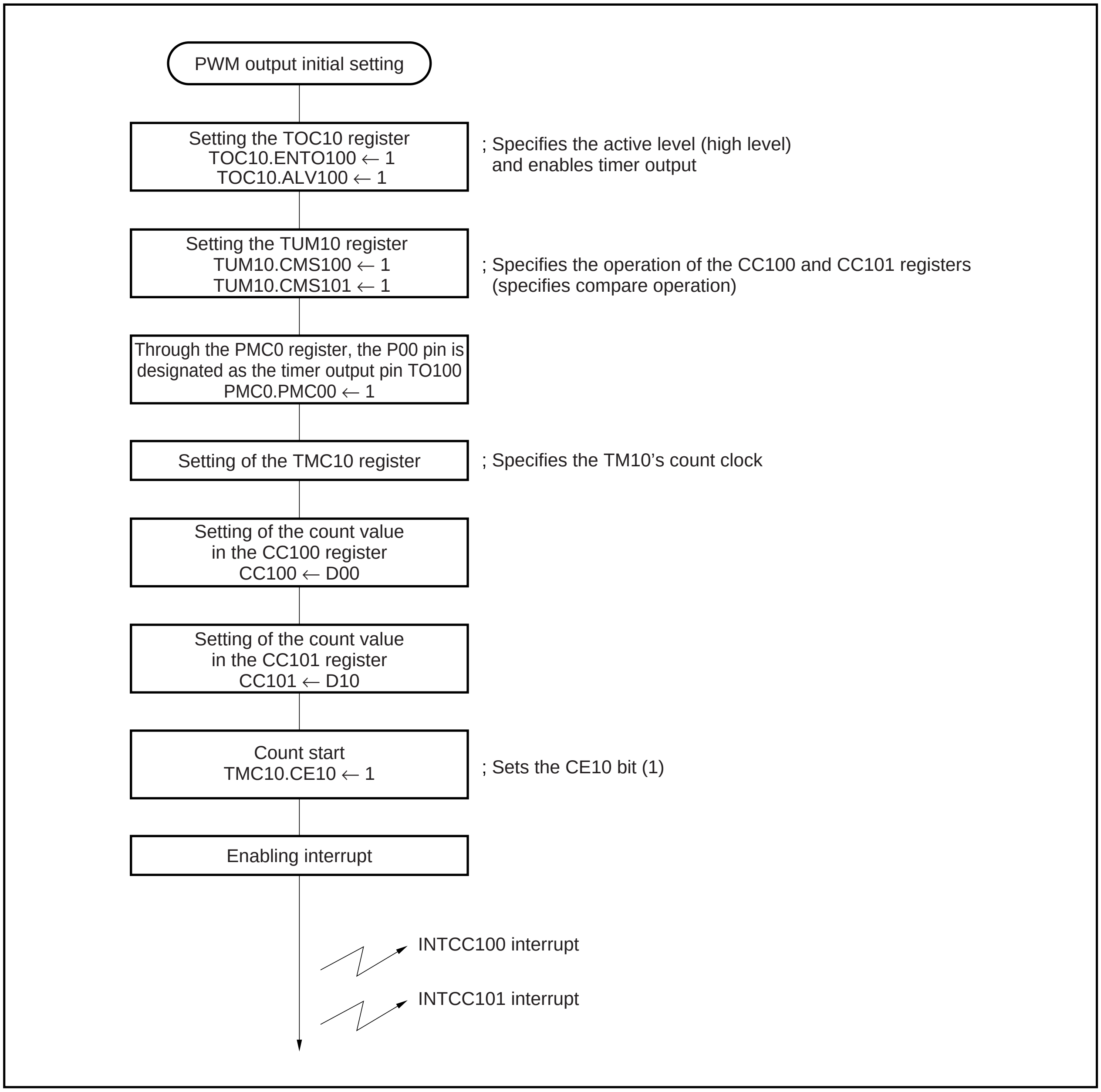
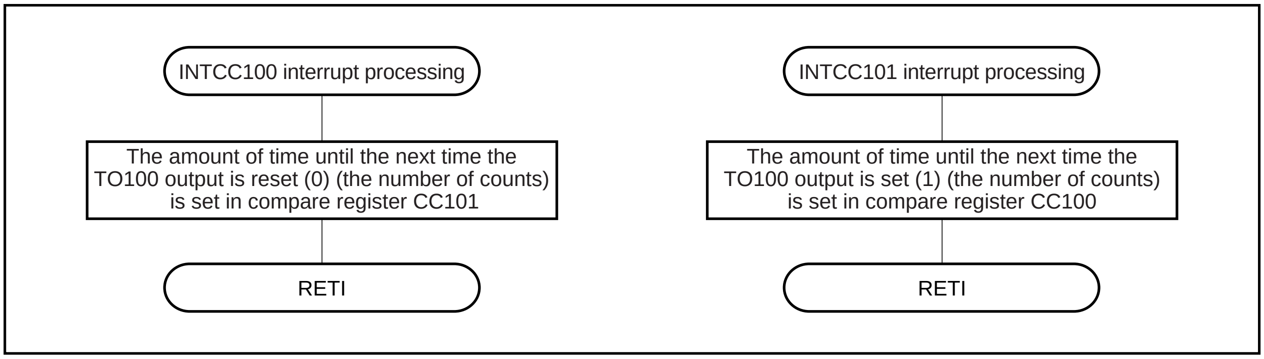

PWM output initial setting
Setting the TOC10 register
TOC10.ENTO100 ← 1
TOC10.ALV100 ← 1
; Specifies the active level (high level) and enables timer output
Setting the TUM10 register
TUM10.CMS100 ← 1
TUM10.CMS101 ← 1
; Specifies the operation of the CC100 and CC101 registers (specifies compare operation)
Through the PMC0 register, the P00 pin is designated as the timer output pin TO100
PMC0.PMC00 ← 1
Setting of the TMC10 register
; Specifies the TM10's count clock
Setting of the count value in the CC100 register
CC100 ← D00
Setting of the count value in the CC101 register
CC101 ← D10
Count start
TMC10.CE10 ← 1
; Sets the CE10 bit (1)
Enabling interrupt
INTCC100 interrupt
INTCC101 interrupt

Figure 9-18. Example of Interrupt Request Processing Routine for Rewriting Compare Value


INTCC100 interrupt processing
The amount of time until the next time the TO100 output is reset (0) (the number of counts) is set in compare register CC101
RETI
INTCC101 interrupt processing
The amount of time until the next time the TO100 output is set (1) (the number of counts) is set in compare register CC100
RETI

(4) Operation for frequency measurement (Timer 1)

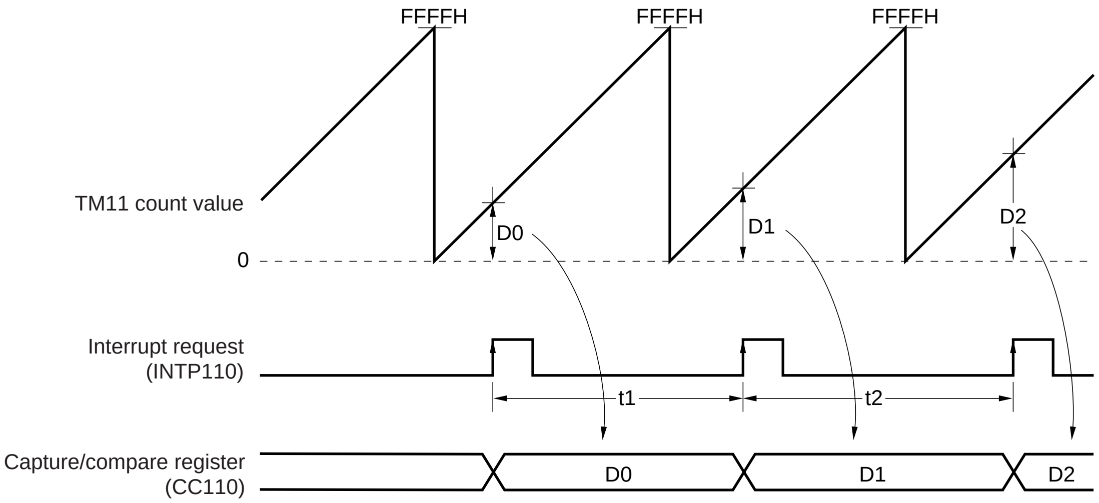
Timer 1 can measure the frequency of an external pulse's input to pins INTP1n0 to INTP1n3 (n = 0 to 5).

Here, an example is shown where timer 1 and the capture/compare register CC110 are combined to measure the frequency of an external pulse input to the INTP110 pin with 16-bit precision.

The active edge of the INTP110 input signal is specified to be the rising edge by the INTM2 register.

The frequency is calculated by determining the difference between the TM11 count value (Dn) captured in the CC110 register from the nth rising edge, and the count value (Dn–1) captured from the rising edge the (n – 1)th time, then multiplying this value by the count clock.

Figure 9-19. Example of Frequency Measurement Timing



t1 = {(10000H – D0) + D1} × count clock cycle frequency

t2 = {(10000H – D0) + D2} × count clock cycle frequency

Remark D0 to D2: TM11 count value

Figure 9-20. Example of Frequency Measurement Setting Procedure

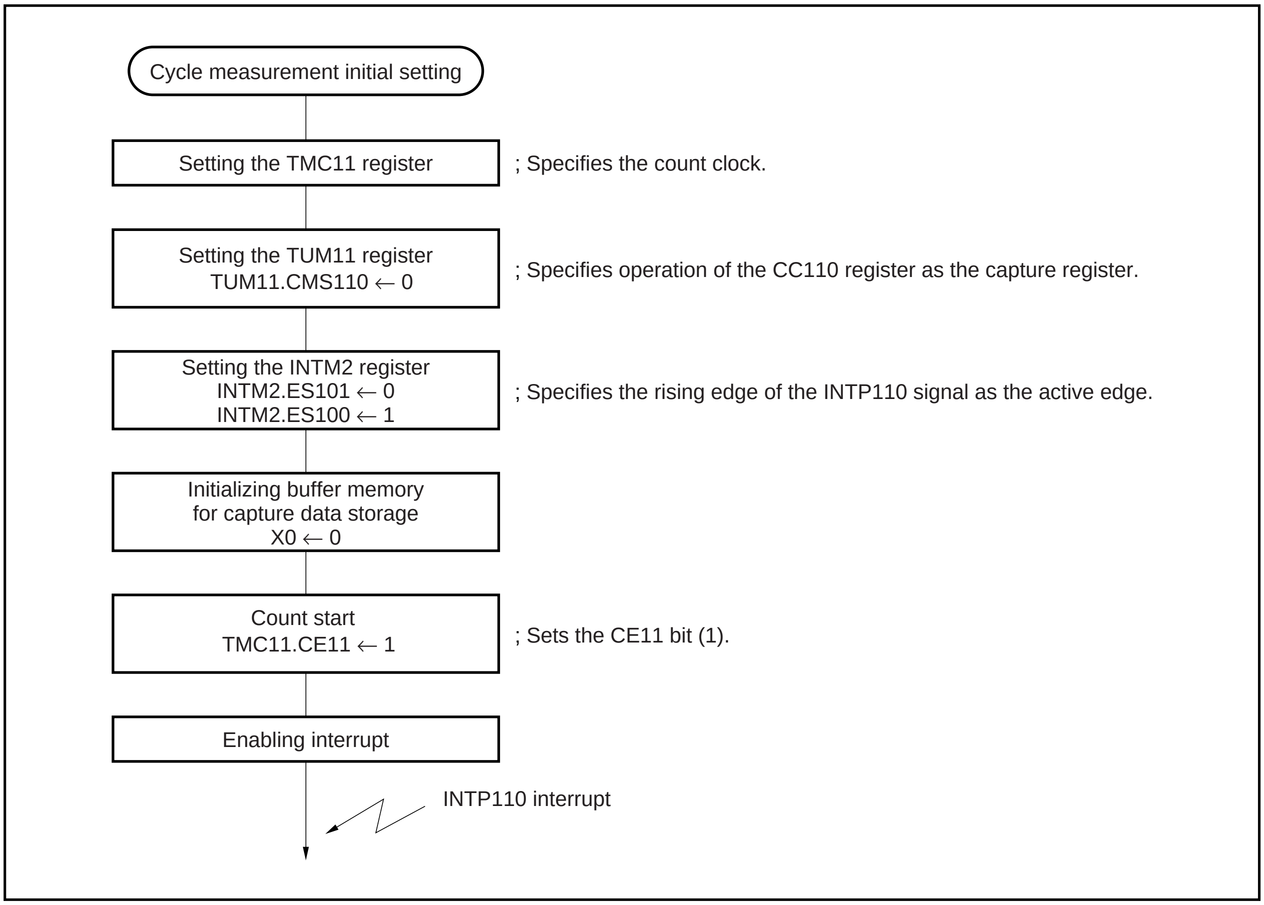
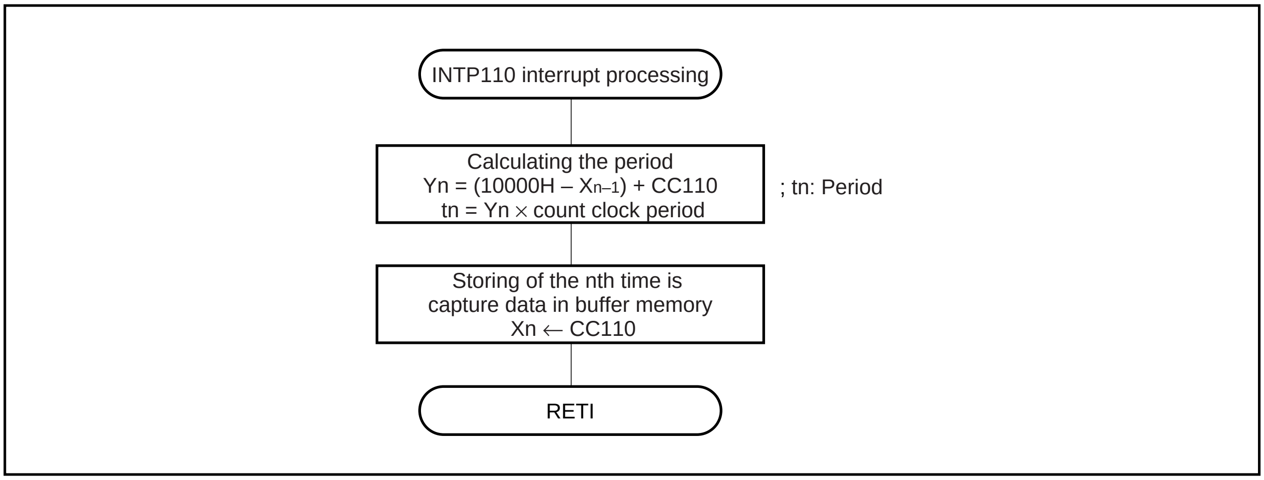

Cycle measurement initial setting
Setting the TMC11 register
; Specifies the count clock.
Setting the TUM11 register
TUM11.CMS110 ← 0
; Specifies operation of the CC110 register as the capture register.
Setting the INTM2 register
INTM2.ES101 ← 0
INTM2.ES100 ← 1
; Specifies the rising edge of the INTP110 signal as the active edge.
Initializing buffer memory
for capture data storage
X0 ← 0
Count start
TMC11.CE11 ← 1
; Sets the CE11 bit (1).
Enabling interrupt
INTP110 interrupt

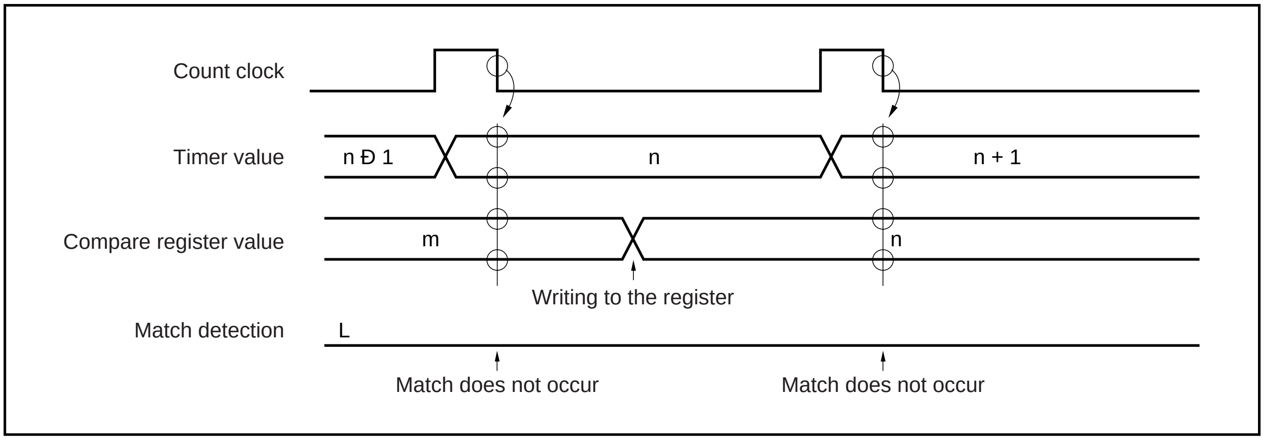
Figure 9-21. Example of Interrupt Request Processing Routine Which Calculates the Frequency


INTP110 interrupt processing
Calculating the period
$Y_n = (10000H - X_{n-1}) + CC110$
$t_n = Y_n \times$ count clock period
; tn: Period
Storing of the nth time is
capture data in buffer memory
Xn ← CC110
RETI

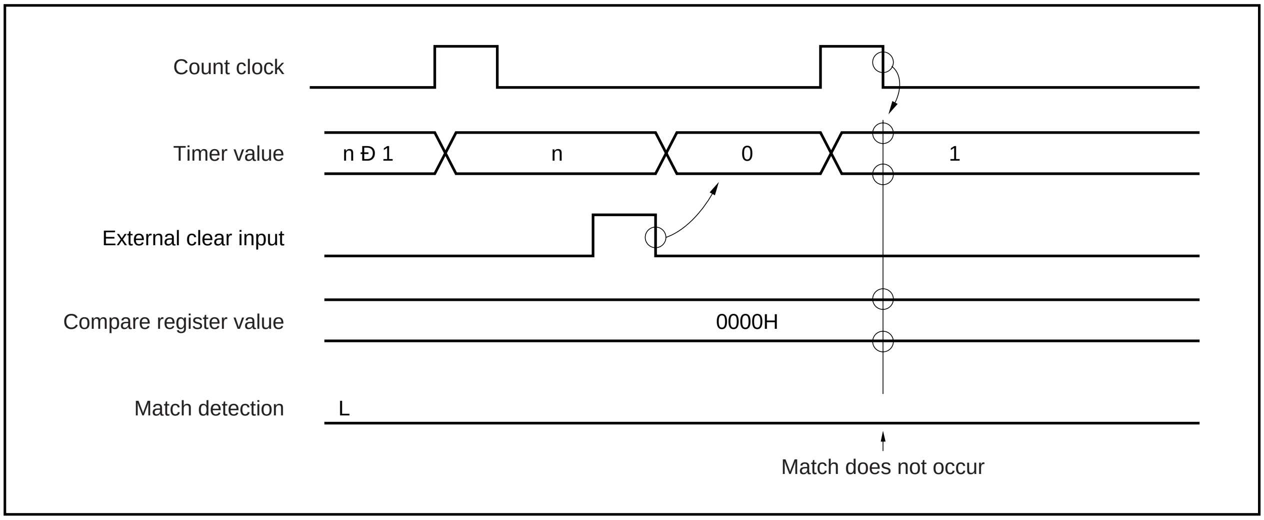
9.7 Precaution

Match detection by the compare register is always performed immediately after timer count up. In the following cases, a match does not occur.

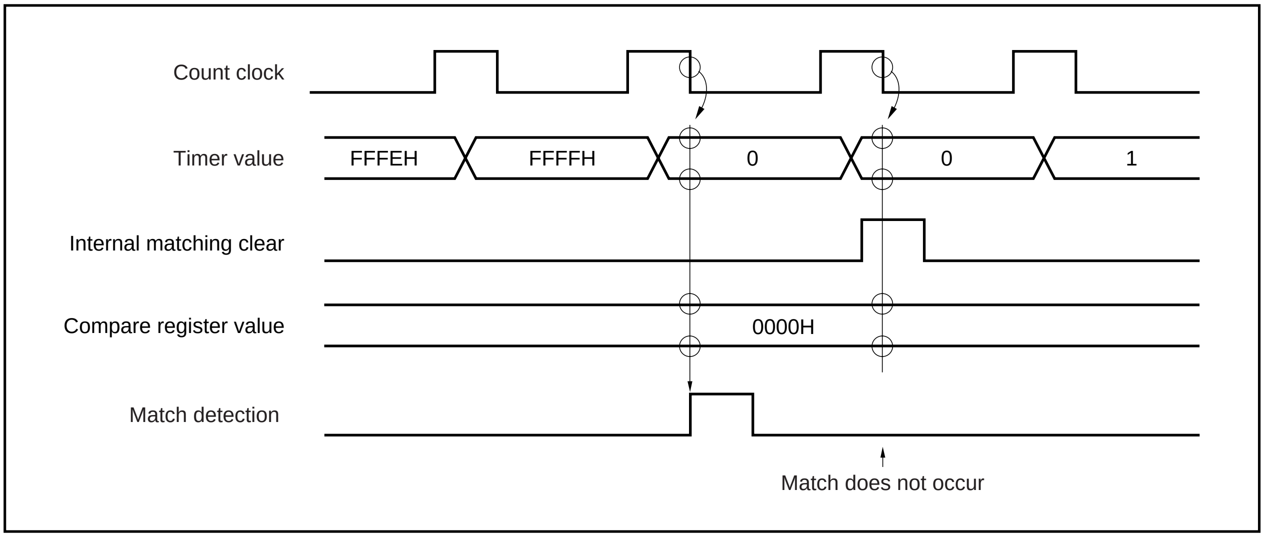
(1) When rewriting the compare register (TM10 to TM15, TM40, TM41)



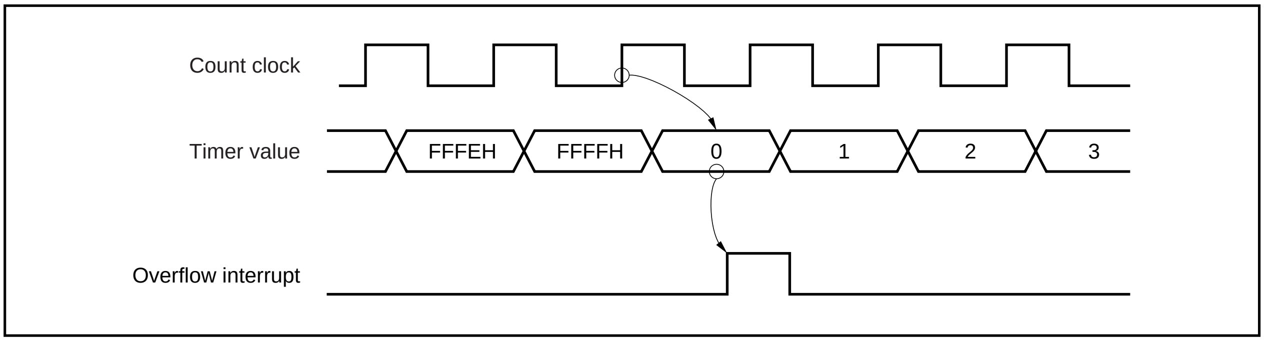
(2) During external clear (TM10 to TM15)



(3) When the timer is cleared (TM40, TM41)



Remark When operating timer 1 as the free-running timer, the timer's value becomes 0 when timer overflow occurs.



CHAPTER 10 SERIAL INTERFACE FUNCTION

10.1 Features

Two types of serial interfaces with 6 transmit/receive channels are provided as the serial interface function, and up to 4 channels can be used simultaneously.

The following two types of interface configuration are provided.

(1)	Asynchronous serial interface (UART0, UART1):	2 channels
(2)	Clocked serial interface (CSI0 to CSI3):	4 channels

UART0 and UART1 use the method of transmitting and receiving 1 byte of serial data following the start bit, and full duplex communication is possible.

CSI0 to CSI3 carry out data transfer with 3 types of signal lines, a serial clock ($\overline{\text{SCK0}}$ to $\overline{\text{SCK3}}$), serial input (SI0 to SI3), and serial output (SO0 to SO3) (3-wire serial I/O).

Caution UART0 and CSI0, and UART1 and CSI1 share the same pins, the use of which is specified with the ASIM00 and ASIM10 registers.

10.2 Asynchronous Serial Interfaces 0, 1 (UART0, UART1)

10.2.1 Features

○ Transfer rate 150 bps to 76,800 bps (using the exclusive baud rate generator when the internal system clock is 33 MHz)
Maximum 4.125 Mbps (using the ϕ/2 clock when the internal system clock is 33 MHz)

○ Full duplex communication On-chip receive buffer (RXBn)

○ 2-pin configuration TXDn: Transmit data output pin
RXDn: Receive data input pin

○ Receive error detection functions

- Parity error
- Framing error
- Overrun error

○ Interrupt sources: 3 types

- Receive error interrupt (INTSERn)
- Reception complete interrupt (INTSRn)
- Transmission complete interrupt (INTSTn)

○ The character length of transmit/receive data is specified by the ASIMn0 and ASIMn1 registers.

○ Character length 7, 8 bits
9 bits (when adding an expansion bit)

○ Parity function: odd, even, 0, none

○ Transmission stop bit: 1, 2 bits

○ On-chip dedicated baud rate generator

○ Serial clock ($\overline{\text{SCKn}}$) output function

Remark n = 0, 1

10.2.2 Configuration

UARTn is controlled by the asynchronous serial interface mode registers (ASIMn0, ASIMn1) and the asynchronous serial interface status registers (ASISn) (n = 0, 1). Receive data is held in the receive buffer (RXBn) and transmit data is written in the transmit shift registers (TXSn).

The asynchronous serial interface is configured as shown in Figure 10-1.

(1) Asynchronous serial interface mode registers (ASIM00, ASIM01, ASIM10, ASIM11)

The ASIMn0 and ASIMn1 registers are 8-bit registers that specify asynchronous serial interface operations.

(2) Asynchronous serial interface status registers (ASIS0, ASIS1)

The ASISn registers are registers of flags that show the contents of errors when a receive error occurs and transmission status flags. Each receive error flag is set (1) when a receive error occurs and is cleared (0) by reading of data from the receive buffer (RXBn) or reception of the next new data (if there is an error in the next data, that error flag will not be cleared (0) but left set (1)).

The transmit status flag is set (1) when transmission starts and is cleared (0) when transmission ends.

(3) Receive control parity check

Receive operations are controlled according to the contents set in the ASIMn0 and ASIMn1 registers. Also, errors such as parity errors are checked during receive operations. If an error is detected, a value corresponding to the error content is set in the ASISn register.

(4) Receive shift register

This is a shift register that converts serial data input to the RXDn pin to parallel data. When 1 byte of data is received, the receive data is transferred to the receive buffer.

This register cannot be directly manipulated.

(5) Receive buffers (RXB0, RXB0L, RXB1, RXB1L)

RXBn are 9-bit buffer registers that hold receive data, and when 7 or 8-bit character data is received, a 0 is stored in the higher bits.

During 16-bit access of these registers, specify RXB0 and RXB1, and during lower 8-bit access, specify RXB0L and RXB1L.

In the receive enabled state, 1 frame of receive data is transmitted to the receive buffer from the receive shift register in synchronization with the termination of shift-in processing.

Also, a reception complete interrupt request (INTSRn) is generated when data is transmitted to the receive buffer.

(6) Transmit shift register (TXS0, TXS0L, TXS1, TXS1L)

TXSn are 9-bit shift registers for transmit processing. Writing of data to these registers starts a transmit operation.

A transmission complete interrupt request (INTSTn) is generated in synchronization with termination of transmission of 1 frame, which includes TXSn data.

During 16-bit access of these registers, specify TXS0 and TXS1, and during lower 8-bit access, specify TXS0L and TXS1L.

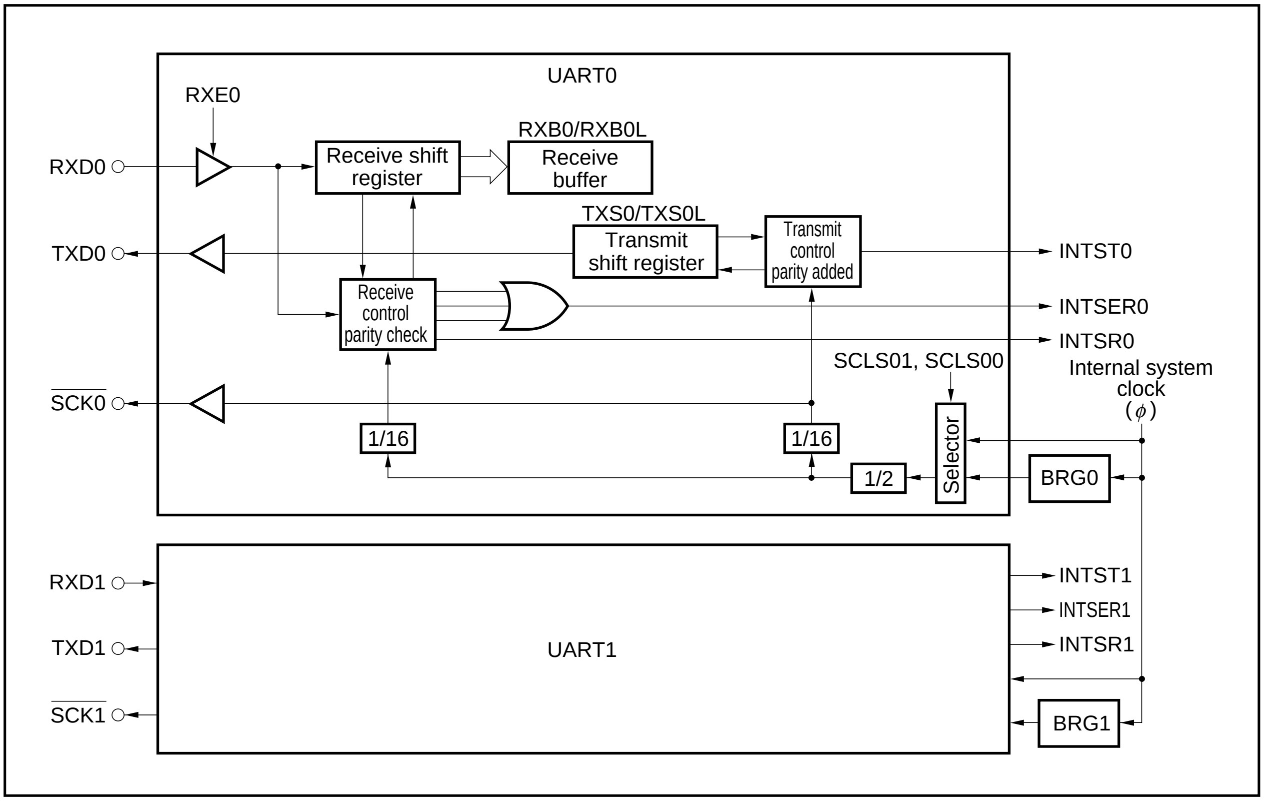
(7) Adding transmit control parity

In accordance with the contents set in the ASIMn0 and ASIMn1 registers, start bits, parity bits, stop bits, etc. are added to the data written to the TXSn or TXSnL register, and transmit operation control is carried out.

(8) Selector

This selects the serial clock source.

Figure 10-1. Block Diagram of Asynchronous Serial Interface



10.2.3 Control registers

(1) Asynchronous serial interface mode registers 00, 01, 10, 11 (ASIM00, ASIM01, ASIM10, ASIM11)

These registers specify the UART0 and UART1 transfer mode.

These registers can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
ASIM00	TXE0	RXE0	PS01	PS00	CL0	SL0	SCLS01	SCLS00	FFFFF0C0H	80H
ASIM10	TXE1	RXE1	PS11	PS10	CL1	SL1	SCLS11	SCLS10	FFFFF0D0H	80H

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7, 6</td><td>TXEn,
RXEn</td><td>Transmit/Receive Enable
Specifies the transmission/reception enable status/disable status.
<table>
<tr><th>TXEn</th><th>RXEn</th><th>Operation</th></tr>
<tr><td>0</td><td>0</td><td>Transmission/reception disabled (CSIn selected)</td></tr>
<tr><td>0</td><td>1</td><td>Reception enabled</td></tr>
<tr><td>1</td><td>0</td><td>Transmission enabled</td></tr>
<tr><td>1</td><td>1</td><td>Transmission/reception enabled</td></tr>
</table>
When reception is disabled, the receive shift register does not detect the start bit. The receive buffer contents are held without shift-in processing or transmit processing to the receive buffer being performed.
While in the reception enabled state, the receive shift operation is started in synchronization with detection of the start bit and after 1 frame of data has been received, the contents of the receive shift register are transmitted to the receive buffer. Also, the reception complete interrupt (INTSRn) is generated in synchronization with transmission to the receive buffer. The TXDn pin becomes high impedance when transmission is disabled and a high level is output if it is not transmitting when transmission is enabled.</td></tr>
</table>

Remark n = 0, 1

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>5, 4</td><td>PSn1, PSn0</td><td>Parity Select
Specifies the parity bit length.
<table>
<tr><th>PSn1</th><th>PSn0</th><th>Operation</th></tr>
<tr><td>0</td><td>0</td><td>No parity, expansion bit operation</td></tr>
<tr><td>0</td><td>1</td><td>Specifies 0 parity
Transmission side → Transmits with parity bit at 0.
Reception side → Does not generate parity errors during receiving.</td></tr>
<tr><td>1</td><td>0</td><td>Specifies odd parity.</td></tr>
<tr><td>1</td><td>1</td><td>Specifies even parity.</td></tr>
</table>
• <b>Even parity</b>
If the number of bits whose values are 1 in the transmit data is odd, a parity bit is set (1). If the number of bits whose values are 1 is even, the parity bit is cleared (0). In this way, the number of bits in the transmit data and the parity bit which are 1 is controlled so that it is an even number. During receiving the number of bits in the receive data and parity bit which are 1 is counted, and if it is an odd number, a parity error is generated.
• <b>Odd parity</b>
This is the opposite of even parity, with the number of bits in the transmit data and parity bit being controlled so that it is an odd number.
During receiving, if the number of bits in the receive data and parity bit which are 1 turns out to be an even number, a parity error is generated.
• <b>0 parity</b>
During transmission, the parity bit is cleared (0) regardless of the transmit data.
During reception, since no parity bit check is performed, no parity error is generated.
• <b>No parity</b>
No parity bit is added to transmit data.
During reception, data are received as having no parity bit. Since there is no parity bit, parity errors are not generated.
Expansion bit operations can be specified with the EBSn bit in the ASIMn1 register.</td></tr>
<tr><td>3</td><td>CLn</td><td>Character Length
Specifies the character length of 1 frame.
0: 7 bits
1: 8 bits</td></tr>
</table>

Remark n = 0, 1

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>2</td><td>SLn</td><td>Stop Bit Length
Specifies the stop bit length.
0: 1 bit
1: 2 bits</td></tr>
<tr><td>1, 0</td><td>SCLSn1,
SCLSn0</td><td>Serial Clock Source
Specifies the serial clock.
<table>
<tr><th>SCLSn1</th><th>SCLSn0</th><th>Serial Clock</th></tr>
<tr><td>0</td><td>0</td><td>Baud rate generator output</td></tr>
<tr><td>0</td><td>1</td><td>ϕ/2 (× 16 sampling rate)</td></tr>
<tr><td>1</td><td>0</td><td>ϕ/2 (× 8 sampling rate)</td></tr>
<tr><td>1</td><td>1</td><td>ϕ/2 (× 4 sampling rate)</td></tr>
</table>
• <b>In the case of SCLSn1, SCLSn0 = 00</b>
ϕ/2 is selected as the serial clock source. (ϕ: internal system clock). In the asynchronous mode, ×16, ×8 and ×4 sampling rates are used, so the baud rate is expressed by the following formula.
$\text{Baud rate} = \frac{\phi/2}{\text{sampling rate}}\text{bps}$
Based on the formula above, the baud rate value in the case where a representative clock is used is shown below.
<table>
<tr><th>Sampling Rate[Note]
Internal
System Clock (ϕ)</th><th>×16
(01)</th><th>×8
(10)</th><th>×4
(11)</th></tr>
<tr><td>40 MHz</td><td>1,250 K</td><td>2,500 K</td><td>—</td></tr>
<tr><td>33 MHz</td><td>1,031 K</td><td>2,062 K</td><td>4,125 K</td></tr>
<tr><td>25 MHz</td><td>781 K</td><td>1,562 K</td><td>3,125 K</td></tr>
<tr><td>20 MHz</td><td>625 K</td><td>1,250 K</td><td>2,500 K</td></tr>
<tr><td>16 MHz</td><td>500 K</td><td>1,000 K</td><td>2,000 K</td></tr>
<tr><td>12.5 MHz</td><td>390 K</td><td>781 K</td><td>1,562 K</td></tr>
<tr><td>10 MHz</td><td>312 K</td><td>625 K</td><td>1,250 K</td></tr>
<tr><td>8 MHz</td><td>250 K</td><td>500 K</td><td>1,000 K</td></tr>
<tr><td>5 MHz</td><td>156 K</td><td>312 K</td><td>625 K</td></tr>
</table>
<b>Note</b> Values in ( ) are the set values for the SCLSn1 and SCLSn0 bits
• <b>If SCLSn1, SCLSn0 = 00</b>
The baud rate generator output is selected as the serial clock source. For details concerning the baud rate generator, refer to <b>10.4 Dedicated Baud Rate Generators 0 to 2 (BRG0 to BRG2)</b>.</td></tr>
</table>

Caution UARTn operation is not guaranteed if this register is changed during UARTn transmission or reception. Furthermore, if this register is changed during UARTn transmission or reception, a transmission complete interrupt (INTSTn) is generated during transmission, and a reception complete interrupt (INTSRn) is generated during reception.

Remark n = 0, 1

	7	6	5	4	3	2	1	0	Address	After reset
ASIM01	0	0	0	0	0	0	0	EBS0	FFFFF0C2H	00H
ASIM11	0	0	0	0	0	0	0	EBS1	FFFFF0D2H	00H

Bit Position	Bit Name	Function
0	EBSn	Extended Bit Select Specifies transmit/receive data expansion bit operation when no parity operation is specified (PSn1, PSn0 = 00). 0: Expansion bit operation disabled. 1: Expansion bit operation enabled. When expansion bit is specified, 1 data bit is added to the high-order of 8-bit transmit/receive data, and communications by 9-bit data are enabled. Expansion bit operation is enabled only in the case where no parity operations have been specified in the ASIMn0 register. If 0 parity, or even/odd parity operation is specified, the EBSn bit specification is made invalid and the expansion bit adding operation is not performed.

Caution UARTn operation when this register has been changed during UARTn transmission/ reception is not guaranteed.

Remark n = 0, 1

(2) Asynchronous serial interface status registers 0, 1 (ASIS0, ASIS1)

These registers are configured with 3-bit error flags (PEn, FEn, OVEn), which show the error status when UARTn reception is terminated, and a transmit status flag (SOTn) (n = 0,1).

The status flag that shows a receive error always shows the state of the error that occurred most recently. That is, if the same error occurred several times before reading of receive data, this flag would hold the status of the error that occurred most recently.

If a receive error occurs, after reading the ASISn register, read the receive buffer (RXBn or RXBnL) and clear the error flag.

These are read-only registers in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
ASIS0	SOT0	0	0	0	0	PE0	FE0	OVE0	FFFFF0C4H	00H
ASIS1	SOT1	0	0	0	0	PE1	FE1	OVE1	FFFFF0D4H	00H

Bit Position	Bit Name	Function
7	SOTn	Status Of Transmission This is a status flag that shows the transmission operation’s state. Set (1): Transmission start timing (writing to the TXSn or TXSnL register) Clear (0): Transmission end timing (generation of the INTSTn interrupt) When about to start serial data transmission, use this as a means of judging whether writing to the transmit shift register is enabled or not.
2	PEn	Parity Error This is a status flag that shows a parity error. Set (1): When transmit parity and receive parity do not match. Clear (0): Data are read from the receive buffer and processed.
1	FEn	Framing Error This is a status flag that shows a framing error. Set (1): When a stop bit was not detected. Clear (0): Data are read from the receive buffer and processed.
0	OVEn	Overrun Error This is a status flag that shows an overrun error. Set (1): When UARTn has finished the next receiving processing before fetching receive data from the receive buffer. Clear (0): Data are read from the receive buffer and processed. Furthermore, due to the configuration where 1 frame at a tie is received, then the contents of the receive shift register are transmitted to the receive buffer, when an overrun error has occurred, the next receive data is written over the data existing in the receive buffer, and the previous receive data is discarded.

Remark n = 0, 1

(3) Receive buffers 0, 0L, 1, 1L (RXB0, RXB0L, RXB1, RXB1L)

RXBn are 9-bit buffer registers that hold receive data, with a 0 stored in the higher bits when 7 or 8-bit character data is received (n = 0, 1).

During 16-bit access of these registers, specify RXB0 and RXB1, and during lower 8-bit access, specify RXB0L and RXB1L.

While in the reception enabled state, receive data is transmitted from the receive shift register to the receive buffer in synchronization with the end of shift-in processing of 1 frame.

Also, a reception complete interrupt request (INTSRn) is generated by transfer of receive data to the receive buffer.

In the reception disabled state, transmission of receive data to the receive buffer is not performed even if shift-in processing of 1 frame is completed, and the contents of the receive buffer are held.

Also, a reception complete interrupt request is not generated.

RXB0 and RXB1 are read-only registers in 16-bit units, and RXB0L and RXB1L are read-only registers in 8- or 1-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
RXB0	0	0	0	0	0	0	0	RXEB0	RXB07	RXB06	RXB05	RXB04	RXB03	RXB02	RXB01	RXB00	FFFFF0C8H	Undefined

	7	6	5	4	3	2	1	0		
RXB0L	RXB07	RXB06	RXB05	RXB04	RXB03	RXB02	RXB01	RXB00	FFFFF0CAH	Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RXB1	0	0	0	0	0	0	0	RXEB1	RXB17	RXB16	RXB15	RXB14	RXB13	RXB12	RXB11	RXB10	FFFFF0D8H	Undefined

	7	6	5	4	3	2	1	0		
RXB1L	RXB17	RXB16	RXB15	RXB14	RXB13	RXB12	RXB11	RXB10	FFFFF0DAH	Undefined

Bit Position	Bit Name	Function
8	RXEBn	Receive Extended Buffer This is the expansion bit during reception of 9-bit/character data. A 0 can be read in reception of 7 and 8-bit/character data.
7 to 0	RXBn7 to RXBn0	Receive Buffer This stores receive data. A 0 can be read when RXBn7 is receiving 7-bit/character data.

Remark n = 0, 1

(4) Transmit shift registers 0, 0L, 1, 1L (TXS0, TXS0L, TXS1, TXS1L)

TXSn are 9-bit shift registers for transmission processing and when transmission is enabled, transmission operations are started (n = 0, 1) by writing of data to these registers.

When transmission is disabled, the values are disregarded even if writing is performed.

A transmission complete interrupt request (INTSTn) is generated in synchronization with the end of transmission of 1 frame including TXS data.

During 16-bit access of these registers, specify TXS0 and TXS1, and during lower 8-bit access, specify TXS0L and TXS1L.

TXS0 and TXS1 are write-only registers in 16-bit units, and TXS0L and TXS1L are write-only registers in 8-bit units.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
TXS0	0	0	0	0	0	0	0	TXED0	TXS07	TXS06	TXS05	TXS04	TXS03	TXS02	TXS01	TXS00	FFFFF0CCH	Undefined

	7	6	5	4	3	2	1	0		
TXS0L	TXS07	TXS06	TXS05	TXS04	TXS03	TXS02	TXS01	TXS00	FFFFF0CEH	Undefined

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
TXS1	0	0	0	0	0	0	0	TXED1	TXS17	TXS16	TXS15	TXS14	TXS13	TXS12	TXS11	TXS10	FFFFF0DCH	Undefined

	7	6	5	4	3	2	1	0		
TXS1L	TXS17	TXS16	TXS15	TXS14	TXS13	TXS12	TXS11	TXS10	FFFFF0DEH	Undefined

Bit Position	Bit Name	Function
8	TXEDn	Transmit Extended Data This is the expansion bit during transmission of 9-bit/character data.
7 to 0	TXSn7 to TXSn0 (n = 0, 1)	Transmit Shifter This writes transmission data.

Cautions 1. UARTn does not have a transmit buffer, so there is no interrupt request at the end of transmission (to the buffer), and an interrupt request (INTSTn) is generated in synchronization with the end of transmission of 1 frame of data.

2. If the UARTn registers are changed during transmission, UARTn operation is not guaranteed.

Remark n = 0, 1

10.2.4 Interrupt request

UARTn generates the following three types of interrupt requests (n = 0, 1).

- Receive error interrupt (INTSERn)
- Reception complete interrupt (INTSRn)
- Transmission complete interrupt (INTSTn)

The priority order of these three interrupts is, from high to low: receive error interrupt, reception complete interrupt, transmission complete interrupt.

Table 10-1. Default Priority of Interrupt

Interrupt	Priority
Receive error	1
Reception complete	2
Transmission complete	3

(1) Receive error interrupt (INTSERn)

In the reception enabled state, a receive error interrupt is generated by ORing the three receive errors.

In the reception disabled state, no receive error interrupt is generated.

(2) Reception completion interrupt (INTSRn)

In the reception enabled state, a reception complete interrupt is generated when data is shifted into the receive shift register and transferred to the receive buffer.

This reception complete interrupt request is also generated when a receive error has occurred, but the receive error interrupt has a higher servicing priority.

In the reception disabled state, no reception complete interrupt is generated.

(3) Transmission completion interrupt (INTSTn)

As this UARTn has no transmit buffer, a transmission complete interrupt is generated when one frame of transmit data containing a 7-, 8-, or 9-bit character is shifted out of the transmit shift register.

A transmission complete interrupt is output at the start of transmission of the last bit of transmit data.

10.2.5 Operation

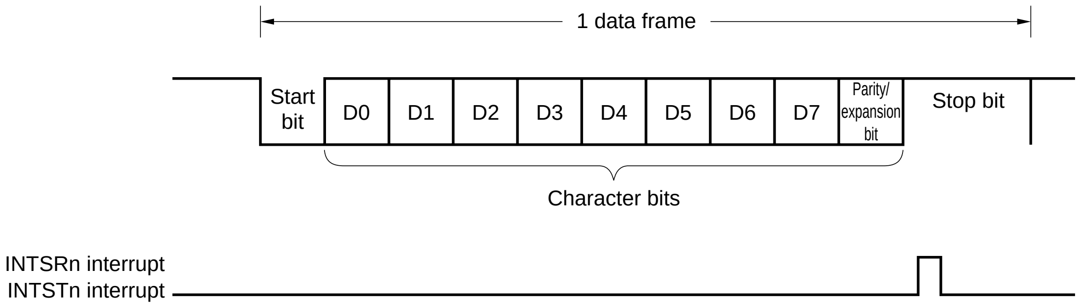
(1) Data format

Transmission and reception of full duplex serial data are performed.

As shown in Figure 10-2, 1 data frame consists of a start bit, character bits, a parity bit, and a stop bit as the format of transmit/receive data.

Specification of the character bit length within 1 data frame, parity selection and specification of the stop bit length are performed by the asynchronous serial interface mode register (ASIMn0, ASIMn1) (n = 0, 1).

Figure 10-2. Transmission/Reception Data Format of Asynchronous Serial Interface



- Start bit...........................1 bit
- Character bits ....7 bits/8 bits
- Parity/expansion bit........Even parity/odd parity/0 parity/no parity/expansion bit
- Stop bit...........................1 bit/2 bits

Remark n = 0, 1

(2) Transmission

Transmission starts when data is written to the transmit shift register (TXSn or TXSnL). With the transmission complete interrupt (INTSTn) processing routine, the next data is written to the TXSn or TXSnL register (n = 0, 1).

(a) Transmit enable state

This is set with the TXEn bit of the ASIMn0 register.

TXEn = 1: Transmit enabled state
TXEn = 0: Transmit disabled state

However, when setting the transmit enabled state, be sure to set both the CTXEn and CRXEn bits of the clocked serial interface mode register (CSIMn) of the channel in use to 0.

Note that since UARTn does not have CTS (transmit enabled signal) input pins, when the opposite party wants to confirm the reception enabled state, use a port.

(b) Starting a transmit operation

In the transmit enabled state, if data is written to the transmit shift register (TXSn or TXSnL), the transmit operation starts. Transmit data is transmitted from the start bit to the LSB header. A start bit, parity/expansion bit and stop bit are added automatically.

In the transmit disabled state, data is not written to the transmit shift register. Even if writing is done, the values are disregarded.

(c) Transmission interrupt request

If the transmit shift register (TXSn or TXSnL) becomes empty, a transmission complete interrupt request (INTSTn) is generated.

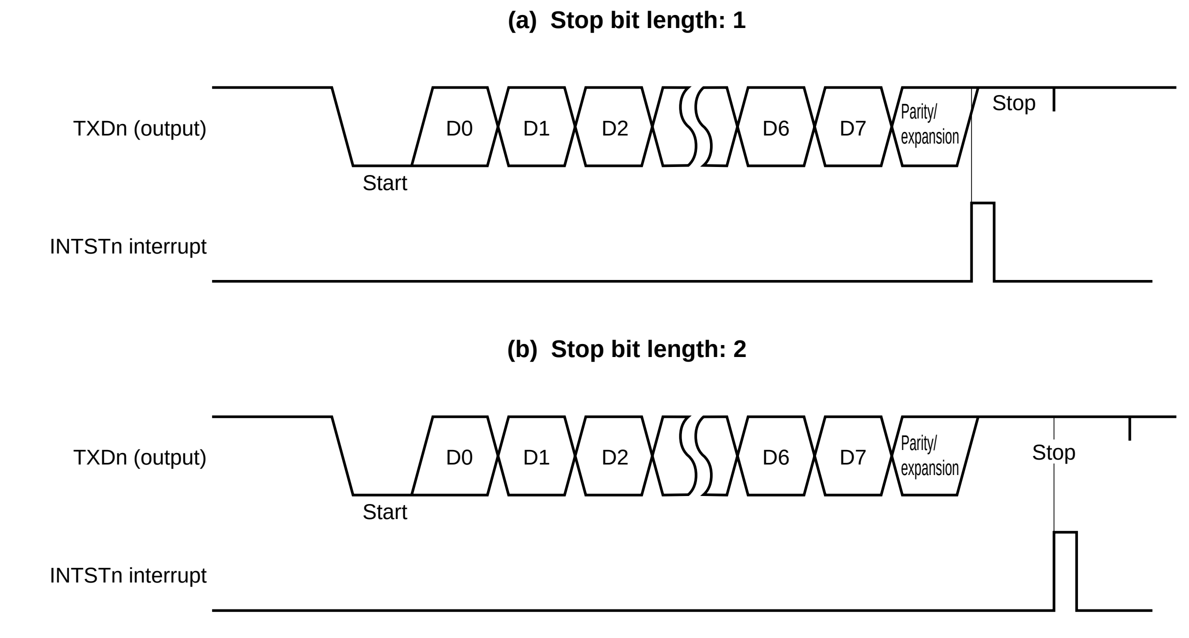
If the next transmit data is not written to the TXSn or TXSnL register, the transmit operation is interrupted.

After 1 transmission is ended, the transmission rate drops if the next transmit data is not written to the TXSn or TSXnL register immediately.

Cautions 1. Normally, when the transmit shift register (TXSn or TXSnL) has become empty, a transmission complete interrupt (INTSTn) is generated. However, when $\overline{\text{RESET}}$ is input, if the transmit shift register (TXSn or TXSnL) has become empty, a transmission complete interrupt (INTSTn) is not generated.

2. During a transmit operation before INTSTn generation, even if data is written to the TXSn or TXSnL register, the written data is invalid.

Figure 10-3. Asynchronous Serial Interface Transmission Completion Interrupt Timing



Remark n = 0, 1

(3) Reception

If reception is enabled, sampling of the RXDn pin is started and if a start bit is detected, data reception begins. When 1 frame of data reception is completed, the reception complete interrupt (INTSRn) is generated. Normally, with this interrupt processing, receive data is transmitted from the receive buffer (RXBn or RXBnL) to memory (n = 0, 1).

(a) Receive enabled state

Reception is enabled when the RXEn bit of the ASIMn0 register is set to 1.

RXEn = 1: Receive enabled state
RXEn = 0: Receive disabled state

However, when reception is enabled, be sure to set both the CTXEn and CRXEn bits of the clocked serial interface mode register (CSIMn) of the channel in use to 0.

In the receive disabled state, the reception hardware stands by in the initial state.

At this time, no reception complete interrupts or reception error interrupts are generated, and the contents of the receive buffer are retained.

(b) Start of receive operation

The receive operation is started by detection of the start bit.

The RXDn pin is sampled using the serial clock from the baud rate generator (BRGn). When an RXDn pin low level is detected, the RXDn pin is sampled again after 8 serial clock cycles. If it is low this is recognized as a start bit, the receive operation is started and the RXDn pin input is subsequently sampled at intervals of 16 serial clock cycles.

If the RXDn pin input is found to be high when sampled again 8 serial clock cycles after an RXDn pin low level is detected, this low level is not recognized as a start bit, the operation is stopped by initializing the serial clock counter for sample timing generation, and the unit waits for the next low-level input.

(c) Reception complete interrupt request

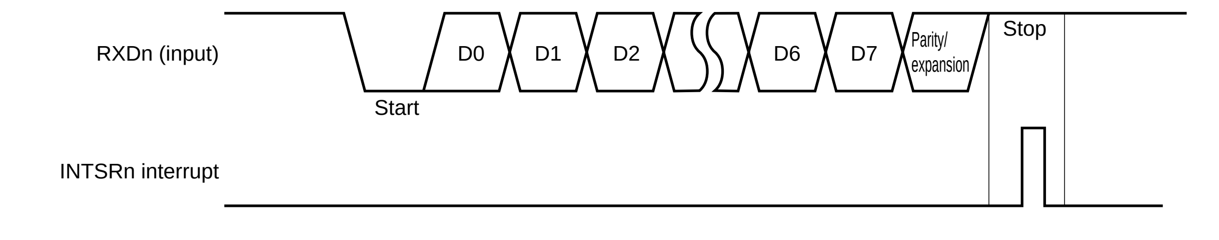
When RXEn = 1, after one frame of data has been received, the receive data in the shift register is transferred to RXBn and RXBnL a reception complete interrupt request (INTSRn) is generated.

Also, even if an error occurs, the receive data where the error occurred is transmitted to the receive buffer (RXBn or RXBnL) and a reception complete interrupt (INTSRn) and receive error interrupt (INTSERn) are generated simultaneously.

Furthermore, if the RXEn bit is reset (0) during a receive operation, the receive operation is stopped immediately. At this time, the contents of the receive buffer (RXBn or RXBnL) and the asynchronous serial interface status register (ASISn) do not change and the reception complete interrupt (INTSRn) and receive error interrupt (INTSERn) are not generated.

When RXEn = 0 and reception is disabled, a reception complete interrupt request is not generated.

Figure 10-4. Asynchronous Serial Interface Reception Complete Interrupt Timing



Remark n = 0, 1

(d) Receive error flag

In synchronization with the receive operation, three types of error flags, the parity error flag, framing error flag, and overrun error flag, are affected.

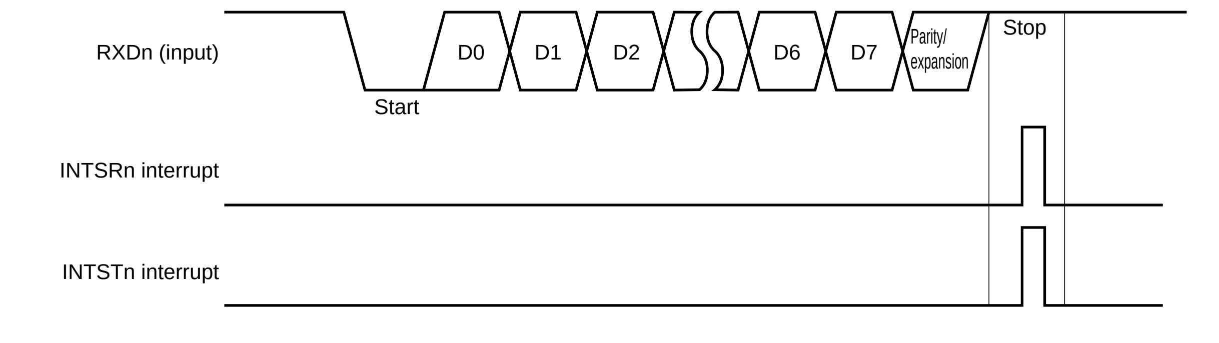
A receive error interrupt request is generated by ORing these three error flags.

By reading out the contents of the ASISn register in the receive error interrupt (INTSERn), which error occurred during reception can be detected.

As for the contents of the ASISn register, either the receive buffer (RXBn or RXBnL) are read or it is reset (0) by reception of the next data (if there is an error in the next receive data, that error flag is set).

Receiving Error	Cause
Parity error	The parity specification during transmission does not match with the parity of the receive data.
Framing error	A stop bit was not detected.
Overrun error	Reception of the next data was completed before data was read from the receive buffer.

Figure 10-5. Receive Error Timing



Remark n = 0, 1

10.3 Clocked Serial Interfaces 0 to 3 (CSI0 to CSI3)

10.3.1 Features

○ High transfer rate Max. 10 Mbps (when the internal system clock is operating at 40 MHz)
… μPD703100-40, 703100A-40
Max. 8.25 Mbps (when the internal system clock is operating at 33 MHz)
… other than above

○ Half-duplex communications

○ Character length: 8 bits

○ It is possible to switch MSB first or LSB first for data.

○ Either external serial clock input or internal serial clock output can be selected.

○ 3-wire type SOn: Serial data output
SIn: Serial data input
$\overline{\text{SCKn}}$: Serial clock input/output

○ Interrupt source 1 type

- Transmission/reception complete interrupt (INTCSIn)

Remark n = 0 to 3

10.3.2 Configuration

CSIn are controlled by the clocked serial interface mode registers (CSIMn). Transmission/reception data can be read from and written to the SIOn registers (n = 0 to 3).

(1) Clocked serial interface mode registers (CSIM0 to CSIM3)

The CSIMn registers are 8-bit registers that specify CSIn operations.

(2) Serial I/O shift registers (SIO0 to SIO3)

The SIOn registers are 8-bit registers that convert serial data to parallel data. SIOn is used for both transmission and reception.

Data is shifted in (received) or shifted out (transmitted) either from the MSB side or the LSB side.

Actual transmitting/receiving operations are controlled by reading from or writing to SIOn.

(3) Selector

This selects the serial clock to be used.

(4) Serial clock controller

This performs control of supply to the serial clock shift register. Also, when the internal clock is used, it controls the clock that outputs to the $\overline{\text{SCKn}}$ pin.

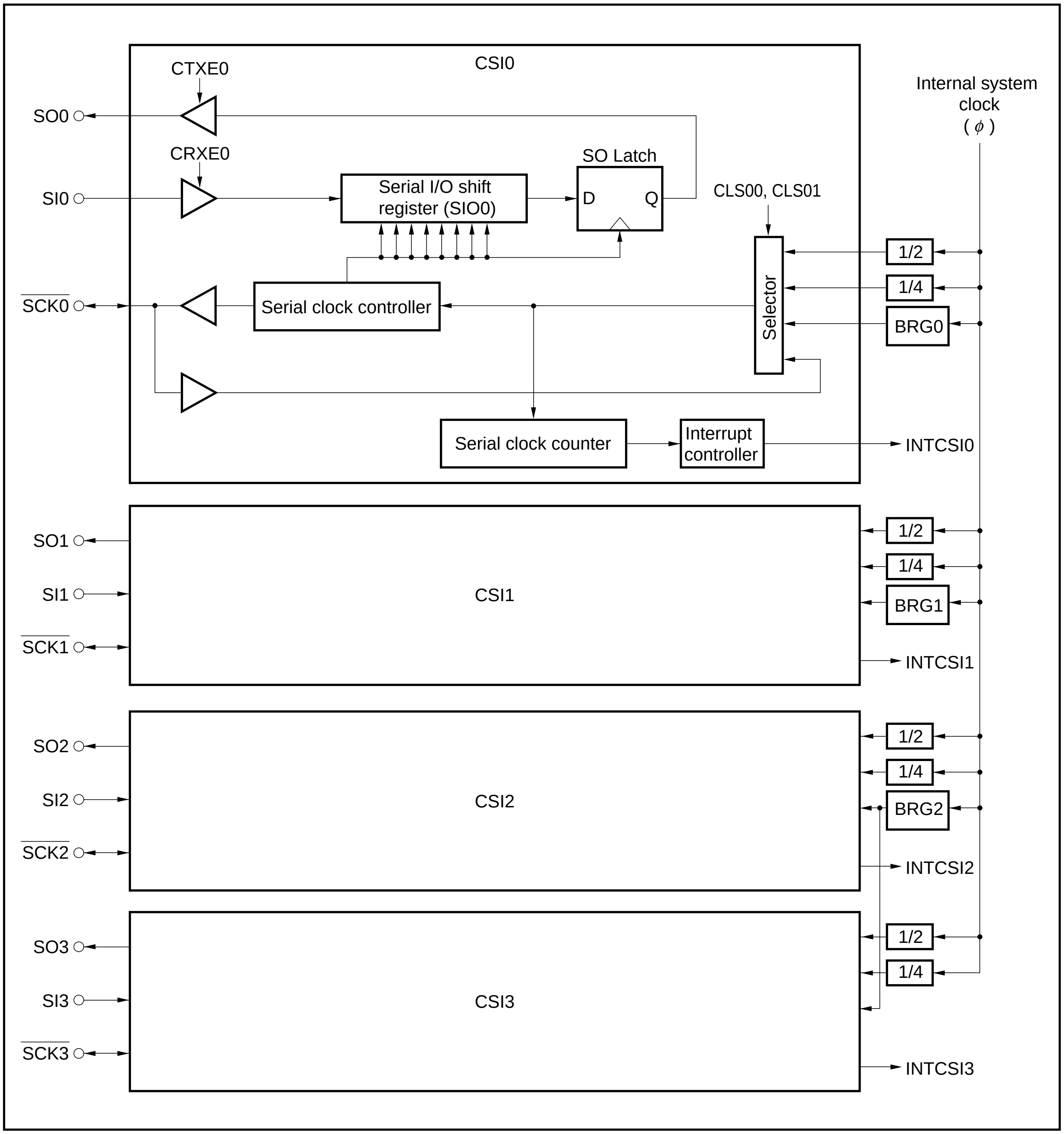
(5) Serial clock counter

Counts the serial clock that outputs, or is input during transmit/receive operations, and determines if 8-bit data were transmitted or received.

(6) Interrupt control circuit

This circuit controls whether or not an interrupt request is generated when the serial clock counter counts 8 clocks.

Figure 10-6. Block Diagram of Clocked Serial Interface



10.3.3 Control registers

(1) Clocked serial interface mode registers 0 to 3 (CSIM0 to CSIM3)

These registers specify the basic operation mode of CSI0 to CSI3.

These registers can be read/written in 8- or 1-bit units (however, for bit 5, only reading is possible).

	7	6	5	4	3	2	1	0	Address	After reset
CSIM0	CTXE0	CRXE0	CSOT0	0	0	MOD0	CLS01	CLS00	FFFFF088H	00H
CSIM1	CTXE1	CRXE1	CSOT1	0	0	MOD1	CLS11	CLS10	FFFFF098H	00H
CSIM2	CTXE2	CRXE2	CSOT2	0	0	MOD2	CLS21	CLS20	FFFFF0A8H	00H
CSIM3	CTXE3	CRXE3	CSOT3	0	0	MOD3	CLS31	CLS30	FFFFF0B8H	00H

Bit Position	Bit Name	Function
7	CTXEn	CSI Transmit Enable Specifies the transmit enabled state/disabled state. 0: Transmission disabled state 1: Transmission enabled state When CTXEn = 0, the impedance of both the SOn and SIn pins becomes high.
6	CRXEn	CSI Receive Enable Specifies the receive enabled/disabled state. 0: Reception disabled state 1: Reception enabled state When transmission is enabled (CTXEn = 1) and reception is disabled, if a serial clock is being input, 0 is input to the shift register. If reception is disabled (CRXEn = 0) while receiving data, the SIOn register's contents become undefined.
5	CSOTn	CSI Status Of Transmission Shows that a transmit operation is in progress. Set (1): Transmit start timing (writing to the SIOn register) Clear (0): Transmit end timing (INTCSIn generated) If set in the transmission enabled state (CTXEn = 1), when the attempt is made to start serial data transmission, this is used as a means of judging whether or not writing to serial I/O shift register n (SIOn) is enabled.
2	MODn	Mode Specifies the operating mode. 0: MSB first 1: LSB first

Remark n = 0 to 3

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>1, 0</td><td>CLSn1,
CLSn0</td><td>Clock Source
Specifies the serial clock.
<table>
<tr><th>CLSn1</th><th>CLSn0</th><th colspan="2">Serial Clock Specification</th><th>SCK Pin</th></tr>
<tr><td>0</td><td>0</td><td colspan="2">External clock</td><td>Input</td></tr>
<tr><td>0</td><td>1</td><td rowspan="3">Internal clock</td><td>Specified by the BPRMm register[Note 1]</td><td>Output</td></tr>
<tr><td>1</td><td>0</td><td>ϕ/4[Note 2]</td><td>Output</td></tr>
<tr><td>1</td><td>1</td><td>ϕ/2[Note 2]</td><td>Output</td></tr>
</table>
<b>Notes 1.</b> Refer to <b>10.4 Dedicated Baud Rate Generators 0 to 2 (BRG0 to BRG2)</b> concerning setting of the BPRMm registers (m = 0 to 2).
<b>2.</b> ϕ/4 and ϕ/2 are divider signals (ϕ: Internal system clock).</td></tr>
</table>

Cautions 1. When setting the CLSn1 and CLSn0 bits, do so in the transmission/reception disabled (CTXEn bit = CRXEn bit = 0) state. If the CLSn1 and CLSn0 bits are set in a state other than transmission/reception disabled, subsequent operation may not be normal.

2. If the values set in bits 0 to 2 of these registers are changed while CSIn is transmitting or receiving, the operation of CSIn is not guaranteed.

Remark n = 0 to 3

(2) Serial I/O shift registers 0 to 3 (SIO0 to SIO3)

These registers convert 8-bit serial data to 8-bit parallel data and convert 8-bit parallel data to 8-bit serial data.

The actual transmit/receive operation is controlled by reading from or writing to the SIOn registers.

Shift operation is performed when CTXEn = 1 or CRXEn = 1.

These registers can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
SIO0	SIO07	SIO06	SIO05	SIO04	SIO03	SIO02	SIO01	SIO00	FFFFF08AH	Undefined
SIO1	SIO17	SIO16	SIO15	SIO14	SIO13	SIO12	SIO11	SIO10	FFFFF09AH	Undefined
SIO2	SIO27	SIO26	SIO25	SIO24	SIO23	SIO22	SIO21	SIO20	FFFFF0AAH	Undefined
SIO3	SIO37	SIO36	SIO35	SIO34	SIO33	SIO32	SIO31	SIO30	FFFFF0BAH	Undefined

Bit Position	Bit Name	Function
7 to 0	SIOn7 to SIOn0 (n = 0 to 3)	Serial I/O Data shift in (receiving) or shift out (transmitting) from the MSB or from the LSB.

Caution CSIn operation is not guaranteed if this register is changed during CSIn operation.

10.3.4 Basic operation

(1) Transfer format

CSIn transmits/receives data with three lines: one clock line and two data lines (n = 0 to 3).

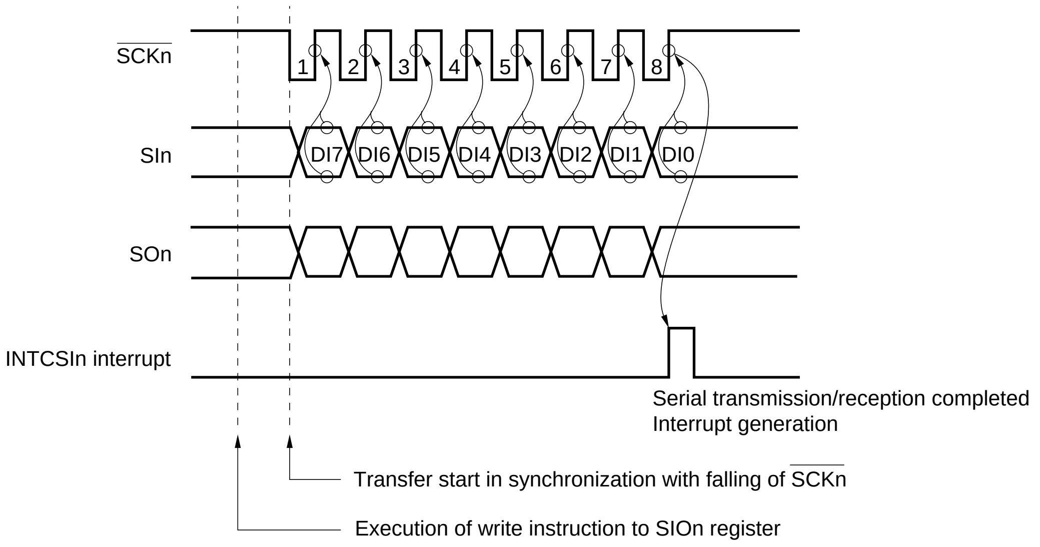
A serial transfer starts when an instruction that writes transfer data to the SIOn register is executed.

In the case of transmission, data is output from the SOn pin at each falling edge of $\overline{\text{SCKn}}$.

In the case of reception, data is latched through the SIn pin at each rising edge of $\overline{\text{SCKn}}$.

$\overline{\text{SCKn}}$ stops when the serial clock counter overflows (at the rising edge of the 8th count), and $\overline{\text{SCKn}}$ remains high until the next data transmission or reception is started. At the same time, a transmission/reception complete interrupt (INTCSIn) is generated.

Caution Even if CTXEn bit is changed from 0 to 1 after the transmit data is written to the SIOnL registers, serial transfer will not begin.


$\overline{\text{SCKn}}$
1
2
3
4
5
6
7
8
SIn
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SOn
INTCSIn interrupt
Serial transmission/reception completed
Interrupt generation
Transfer start in synchronization with falling of $\overline{\text{SCKn}}$
Execution of write instruction to SIOn register

Remark n = 0 to 3

(2) Transmission/reception enabled

CSIn each have only one 8-bit shift register and do not have any buffers, so basically, they conduct transmission and reception simultaneously (n = 0 to 3).

(a) Transmission/reception enable conditions

Setting of the CSIn transmission and reception enable conditions is accomplished by the CTXEn and CRXEn bits of the CSIMn registers.

However, it is necessary to set TXE0 bit = RXE0 bit = 0 in the ASIM00 register in the case of CSI0 and to set TXE1 bit = RXE1 bit = 0 in the ASIM10 register in the case of CSI1.

CTXEn	CRXEn	Transmit/Receive Operation
0	0	Transmission/reception disabled
0	1	Reception enabled
1	0	Transmission enabled
1	1	Transmission/reception enabled

Remark n = 0 to 3

Remarks **1.** If the CTXEn bit = 0, CSIn becomes as follows.

- CSI0, CSI1: The serial output becomes high impedance or UARTn output (TXDn).
- CSI2, CSI3: The serial output becomes high impedance.

If the CTXEn bit = 1, the shift register data is output.

2. If the CRXEn bit = 0, the shift register input becomes 0.
If the CRXEn bit = 1, the serial input is input to the shift register.

3. In order to receive transmit data itself and check if a bus conflict is occurring, set CTXEn bit = CRXEn bit = 1.

(3) Starting transmit/receive operations

Transmit or receive operations are started by reading/writing the SIOn registers. Transmission/reception start control is carried out by setting the CTXEn and CRXEn bits of the CSIMn registers as shown below (n = 0 to 3).

CTXEn	CRXEn	Start Condition
0	0	Doesn’t start
0	1	Reads the SIOn register
1	0	Writes to the SIOn register
1	1	Writes to the SIOn register
0	0 → 1	Rewrites the CRXEn bit

Remark n = 0 to 3

When the CTXEn bit is 0, the SIOn register is read/write, and even if it is set (1) afterward, transfer does not start.

Also, when the CTXEn bit is 0, if the CRXEn bit is changed from 0 to 1, the serial clock is generated and receive operation starts.

10.3.5 Transmission by CSI0 to CSI3

After changing the settings to enable transmission by clocked serial interface mode register n (CSIMn), writing to the SIOn registers starts the transmit operation (n = 0 to 3).

(1) Starting the transmit operation

Starting the transmit operation is accomplished by setting the CTXEn bit of clocked serial interface mode register n (CSIMn) (setting the CRXEn bit to 0), and writing transmit data to shift register n (SIOn).
Note that when the CTXEn bit = 0, the impedance of the SOn pin becomes high.

(2) Transmitting data in synchronization with the serial clock

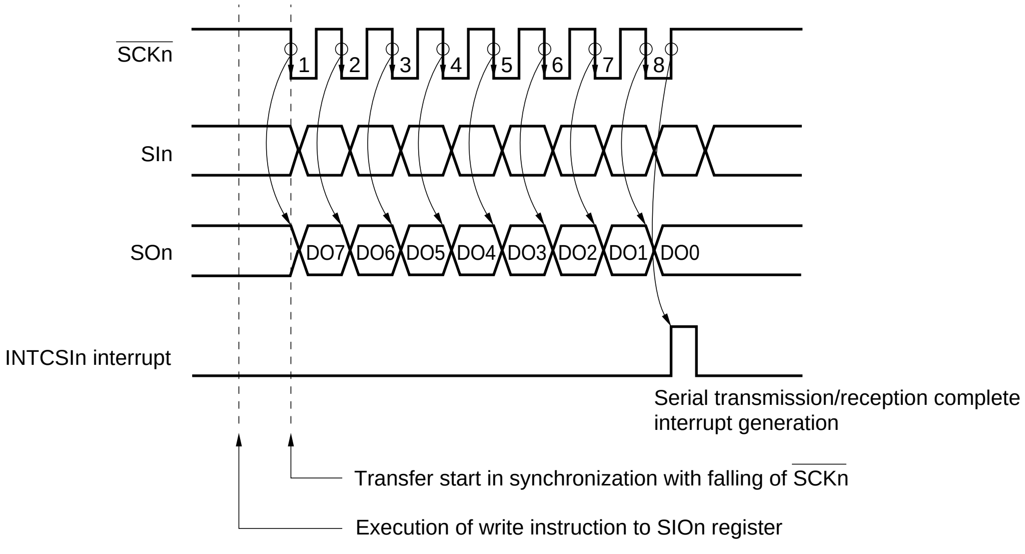
(a) If the internal clock is selected as the serial clock

When transmission is started, the serial clock is output from the $\overline{\text{SCKn}}$ pin and at the same time, data from the SIOn register is output sequentially to the SOn pin in synchronization with the fall of the serial clock.

(b) If an external clock is selected as the serial clock

When transmission is started, data from the SIOn register is output sequentially to the SOn pin in synchronization with the fall of the serial clock input to the $\overline{\text{SCKn}}$ pin after transmission starts. When transmission is not started, the shift operation is not performed even if the serial clock is input to the $\overline{\text{SCKn}}$ pin and the SOn pin's output level does not change.

Figure 10-7. Timing of 3-Wire Serial I/O Mode (Transmission)


$\overline{\text{SCKn}}$
1
2
3
4
5
6
7
8
SIn
SOn
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
INTCSIn interrupt
Serial transmission/reception complete interrupt generation
Transfer start in synchronization with falling of $\overline{\text{SCKn}}$
Execution of write instruction to SIOn register

Remark n = 0 to 3

10.3.6 Reception by CSI0 to CSI3

When the reception disabled setting is changed to reception enabled for clocked serial interface mode register n (CSIMn), and data is read from the SIOn register in the reception enabled state, a receive operation is started (n = 0 to 3).

(1) Starting the receive operation

The following 2 methods can be used to start receive operations.

<1> If the CRXEn bit of the CSIMn register is changed from the reception disabled state (0) to the reception enabled state (1)

<2> If the CRXEn bit of the CSIMn register reads receive data from shift register n (SIOn) when in the reception enabled state (1)

When the CRXEn bit of the CSIMn register is set (1), even if 1 is written again, a receive operation is not started. Note that when the CRXEn bit = 0, the shift register input becomes 0.

(2) Receiving data in synchronization with the serial clock

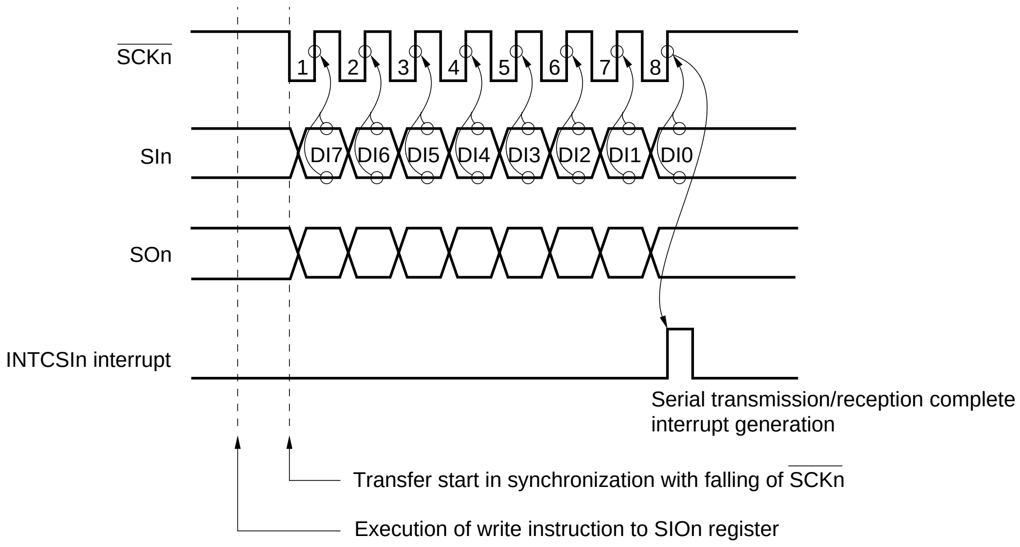
(a) If the internal clock is selected as the serial clock

When reception is started, the serial clock is output from the $\overline{\text{SCKn}}$ pin and at the same time, data from the SIn pin is fetched sequentially to the SIOn register in synchronization with the rise of the serial clock.

(b) If an external clock is selected as the serial clock

When reception is started, data from the SIn pin is fetched sequentially to the SIOn register in synchronization with the rise of the serial clock input to the $\overline{\text{SCKn}}$ pin after reception starts. When reception has not started, the shift operation is not performed even if the serial clock is input to the $\overline{\text{SCKn}}$ pin.

Figure 10-8. Timing of 3-Wire Serial I/O Mode (Reception)


$\overline{\text{SCKn}}$
1
2
3
4
5
6
7
8
SIn
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SOn
INTCSIn interrupt
Serial transmission/reception complete interrupt generation
Transfer start in synchronization with falling of $\overline{\text{SCKn}}$
Execution of write instruction to SIOn register

Remark n = 0 to 3

10.3.7 Transmission and reception by CSI0 to CSI3

If both transmission and reception by clocked serial interface mode register n (CSIMn) are enabled, transmit and receive operations can be carried out simultaneously (n = 0 to 3).

(1) Starting transmit and receive operations

When both the CTXEn bit and CRXEn bit of clocked serial interface mode register n (CSIMn) are set (1), both transmit operations and receive operations can be performed simultaneously (transmit/receive operations).

Transmit and receive operations are started when both the CTXEn and CRXEn bits of the CSIMn register are set to 1, enabling transmission and reception and when transmit data is written to shift register n (SIOn).

If the CRXEn bit of the CSIMn register is 1, even if data is written again, a transmit/receive operation is not started.

(2) Transmitting data in synchronization with the serial clock

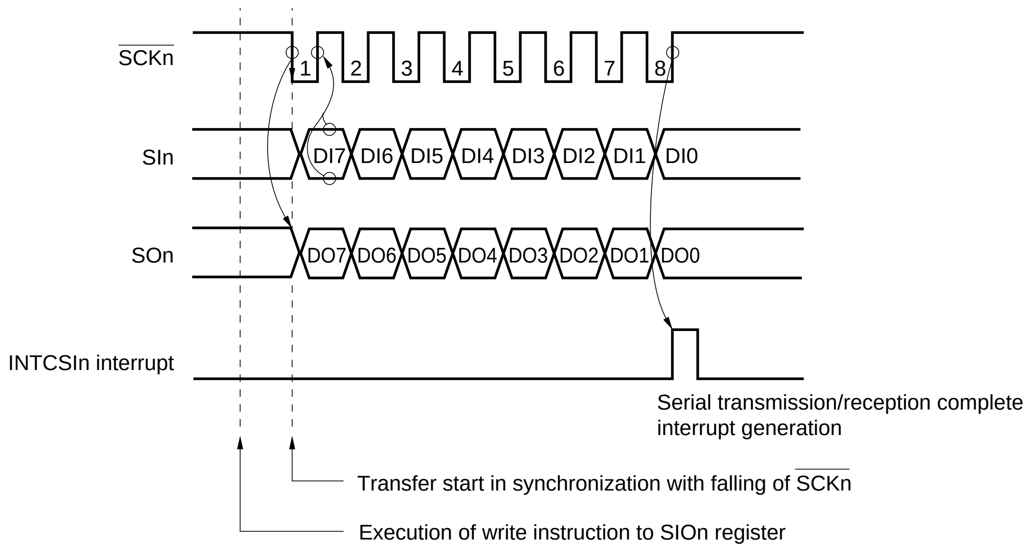
(a) If the internal clock is selected as the serial clock

When transmission/reception is started, the serial clock is output from the $\overline{\text{SCKn}}$ pin and at the same time, data from the SIOn register is output sequentially to the SOn pin in synchronization with the fall of the serial clock. Also, data from the SIn pin is fetched sequentially to the SIOn register in synchronization with the rise of the serial clock.

(b) If an external clock is selected as the serial clock

When transmission/reception is started, data from the SIOn register is output sequentially to the SOn pin in synchronization with the fall of the serial clock input to the $\overline{\text{SCKn}}$ pin after transmission/reception starts. Also, data from the SIn pin is fetched sequentially to the SIOn register in synchronization with the rise of the serial clock. When transmission/reception is not started, even if the serial clock is input to the $\overline{\text{SCKn}}$ pin, shift operations are not performed and the output level of the SOn pin does not change.

Figure 10-9. Timing of 3-Wire Serial I/O Mode (Transmission/Reception)


SCKn
1
2
3
4
5
6
7
8
SIn
DI7
DI6
DI5
DI4
DI3
DI2
DI1
DI0
SOn
DO7
DO6
DO5
DO4
DO3
DO2
DO1
DO0
INTCSIn interrupt
Serial transmission/reception complete interrupt generation
Transfer start in synchronization with falling of SCKn
Execution of write instruction to SIOn register

Remark n = 0 to 3

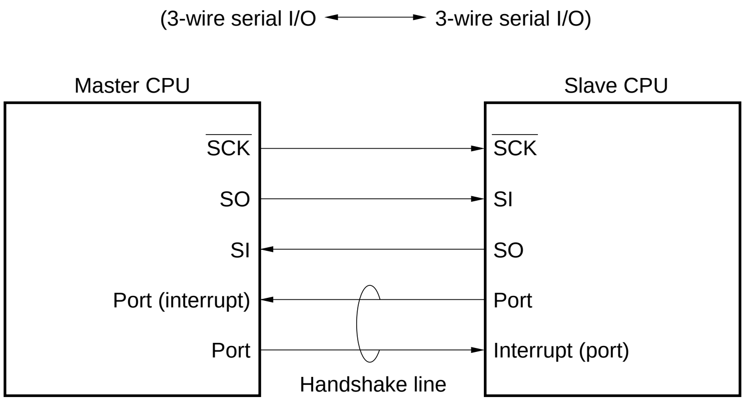
10.3.8 Example of system configuration

Using 3 signal lines, the serial clock (SCKn), serial input (SIn) and serial output (SOn), transfer of 8-bit data is carried out. This is effective in cases where connections are made to peripheral I/O with the old type of clocked serial interface built in, or with a display controller, etc. (n = 0 to 3).

If connecting to multiple devices, a line for handshake is necessary.

Since either the MSB or the LSB can be selected as the communication's header bit, it is possible to communicate with various types of device.

Figure 10-10. Example of CSI System Configuration



10.4 Dedicated Baud Rate Generators 0 to 2 (BRG0 to BRG2)

10.4.1 Configuration and function

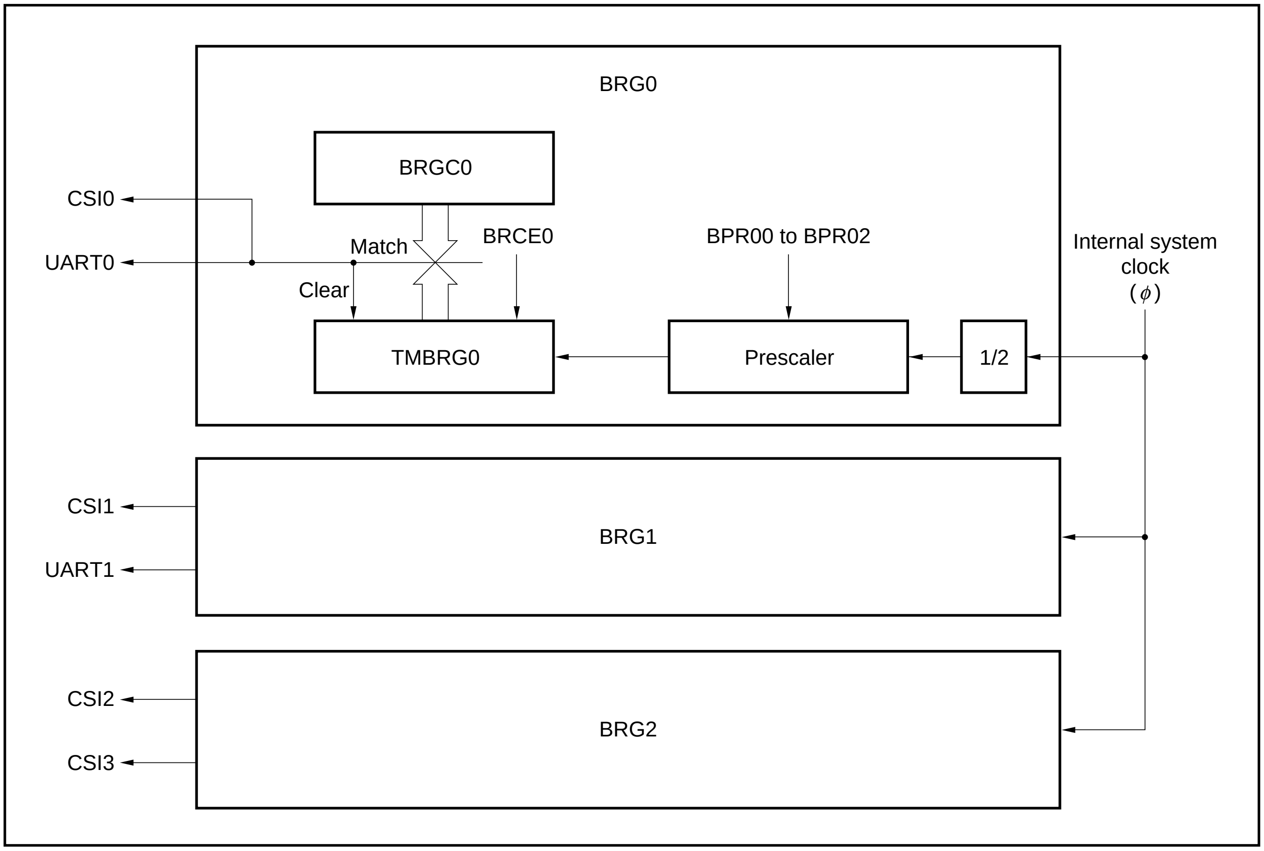
A dedicated baud rate generator output or the internal system clock (ϕ) can be selected for the serial interface serial clock for each channel.

The serial clock source is specified with the ASIM00 and ASIM10 registers for UART0 and UART1, and with the CSIM0 to CSIM3 registers for CSI0 to CSI3.

If the dedicated baud rate generator output is specified, BRG0 to BRG2 are selected as the clock source.

Since 1 serial clock is used in common for 1 channel of transmission and reception, the baud rate is the same for both transmission and for reception.

Figure 10-11. Block Diagram of Dedicated Baud Rate Generator



(1) Dedicated baud rate generators 0 to 2 (BRG0 to BRG2)

Dedicated baud rate generator BRGn (n = 0 to 2) consists of a dedicated 8-bit timer (TMBRGn) which generates the transmission/reception shift clock plus a compare register (BRGCn) and prescaler.

(a) Input clock

Internal system clock (ϕ) is input to the BRGn.

(b) Value set to BRGn

(i) UART0, UART1

When the dedicated baud rate generator is specified as the serial source clock with the UART0, UART1, a sampling rate of ×16 is used, and therefore the baud rate is given by the following expression.

$$\text{Baud rate} = \frac{\phi}{2 \times j \times 2^{k} \times 16 \times 2}\,[\text{bps}]$$

ϕ Internal system clock frequency [Hz]

j: Timer count value = BRGCn register setting value ($1 \leq j \leq 256$Note)

k: Prescaler setting value = BPRMn register setting value (k = 0, 1, 2, 3, 4)

Note The j = 256 setting results in writing 0 to the BRGCn register.

(ii) CSI0 to CSI3

If BRG0 to BRG2 are specified as the serial clock source in CSI0 to CSI3, the actual baud rate is expressed by the following formula.

$$\text{Baud rate} = \frac{\phi}{2 \times j \times 2^{k} \times 2}\,[\text{bps}]$$

ϕ: Internal system clock frequency [Hz]

j: Timer count value = BRGCn register setting value ($1 \leq j \leq 256$Note)

k: Prescaler setting value = BPRMn register setting value (k = 0, 1, 2, 3, 4)

Note The j = 256 setting results in writing 0 to the BRGCn register.

BRGn setting values when representative clock frequencies are used are shown below.

Table 10-2. Baud Rate Generator Setup Values

Baud Rate [bps]		ϕ = 33 MHz			ϕ = 25 MHz			ϕ = 16 MHz			ϕ = 12.5 MHz		
UART0, UART1	CSI0 to CSI3	BPR	BRG	Error	BPR	BRG	Error	BPR	BRG	Error	BPR	BRG	Error
110	1,760	—	—	—	4	222	0.02%	4	142	0.03%	3	222	0.02%
150	2,400	4	215	0.07%	4	163	0.15%	3	208	0.16%	3	163	0.15%
300	4,800	3	215	0.07%	3	163	0.15%	2	208	0.16%	2	163	0.15%
600	9,600	2	215	0.07%	2	163	0.15%	1	208	0.16%	1	163	0.15%
1,200	19,200	1	215	0.07%	1	163	0.15%	0	208	0.16%	0	163	0.15%
2,400	38,400	0	215	0.07%	0	163	0.15%	0	104	0.16%	0	81	0.47%
4,800	768,00	0	107	0.39%	0	81	0.47%	0	52	0.16%	0	41	0.76%
9,600	153,600	0	54	0.54%	0	41	0.76%	0	26	0.16%	0	20	1.73%
10,400	166,400	0	50	0.84%	0	38	1.16%	0	24	0.16%	0	19	1.16%
19,200	307,200	0	27	0.54%	0	20	1.73%	0	13	0.16%	0	10	1.73%
38,400	614,400	0	13	3.29%	0	10	1.73%	0	7	6.99%[Note]	0	5	1.73%
76,800	1,228,800	0	7	4.09%	0	5	1.73%	—	—	—	0	3	15.2%[Note]
153,600	2,457,600	0	3	11.90%[Note]	0	2	27.2%[Note]	—	—	—	—	—	—

Baud Rate [bps]		ϕ = 40 MHz			ϕ = 20 MHz			ϕ = 14.764 MHz			ϕ = 12.288 MHz		
UART0, UART1	CSI0 to CSI3	BPR	BRG	Error	BPR	BRG	Error	BPR	BRG	Error	BPR	BRG	Error
110	1,760	—	—	—	4	178	0.25%	4	131	0.07%	3	218	0.08%
150	2,400	—	—	—	4	130	0.16%	3	192	0.0%	3	160	0.0%
300	4,800	4	130	0.16%	3	130	0.16%	2	192	0.0%	2	160	0.0%
600	9,600	4	65	0.16%	2	130	0.16%	1	192	0.0%	1	160	0.0%
1,200	19,200	3	65	0.16%	1	130	0.16%	0	192	0.0%	0	160	0.0%
2,400	38,400	2	65	0.16%	0	130	0.16%	0	96	0.0%	0	80	0.0%
4,800	76,800	1	65	0.16%	0	65	0.16%	0	48	0.0%	0	40	0.0%
9,600	153,600	0	65	0.16%	0	33	1.36%	0	24	0.0%	0	20	0.0%
10,400	166,400	0	60	0.16%	0	30	0.16%	0	22	0.7%	0	18	2.6%
19,200	307,200	0	32	1.73%	0	16	1.73%	0	12	0.0%	0	10	0.0%
38,400	614,400	0	16	1.73%	0	8	1.73%	0	6	0.0%	0	5	0.0%
76,800	1,228,800	0	8	1.73%	0	4	1.73%	0	3	0.0%	0	3	16.7%[Note]
153,600	2,457,600	0	4	1.73%	0	2	1.73%	0	2	25.0%[Note]	—	—	—

Note Cannot be used because the error is too great.

Remark BPR: Prescaler setting value (Set in the BPRMn register (n = 0 to 2))
BRG: Timer count value (Set in the BRGCn register (n = 0 to 2))
ϕ: Internal system clock frequency

(c) Baud rate error

The baud rate generator error is calculated as follows:

$$\text{Error [\%]} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (normal baud rate)}} - 1 \right) \times 100$$

Example: (9,520/9,600 – 1) × 100 = –0.833 [%]
(5,000/4,800 – 1) × 100 = +4.167 [%]

(2) Allowable error range of baud rate

The allowable error range depends on the number of bits of one frame.

The basic limit is ±5% of baud rate error and ±4.5% of sample timing with an accuracy of 16 bits. However, the practical limit should be ±2.3% of baud rate error, assuming that both the transmission and reception sides contain an error.

10.4.2 Baud rate generator compare registers 0 to 2 (BRGC0 to BRGC2)

These are 8-bit compare registers used to set the timer count value for the BRG0 to BRG2.

These registers can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
BRGC0	BRG07	BRG06	BRG05	BRG04	BRG03	BRG02	BRG01	BRG00	FFFFF084H	Undefined
BRGC1	BRG17	BRG16	BRG15	BRG14	BRG13	BRG12	BRG11	BRG10	FFFFF094H	Undefined
BRGC2	BRG27	BRG26	BRG25	BRG24	BRG23	BRG22	BRG21	BRG20	FFFFF0A4H	Undefined

Caution Do not change the values in the BRGCn (n = 0 to 2) register by software during a transmit/receive operation, because writing this register causes the internal timer (TMBRGn) to be cleared.

10.4.3 Baud rate generator prescaler mode registers 0 to 2 (BPRM0 to BPRM2)

These registers control BRG0 to BRG2 timer count operations and select the count clock.

These registers can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
BPRM0	BRCE0	0	0	0	0	BPR02	BPR01	BPR00	FFFFF086H	00H
BPRM1	BRCE1	0	0	0	0	BPR12	BPR11	BPR10	FFFFF096H	00H
BPRM2	BRCE2	0	0	0	0	BPR22	BPR21	BPR20	FFFFF0A6H	00H

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>7</td><td>BRCEn</td><td>Baud Rate Generator Count Enable
Controls the BRGn count operations.
0: Stops count operations in the cleared state.
1: Enables the count operation.</td></tr>
<tr><td>2 to 0</td><td>BPRn2 to BPRn0</td><td>Baud Rate Generator Prescaler
Specifies the count clock input to the internal timer (TMBRGn).
<table>
<tr><th>BPRn2</th><th>BPRn1</th><th>BPRn0</th><th>Count Clock</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>$\phi/2$ (m = 0)</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>$\phi/4$ (m = 1)</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>$\phi/8$ (m = 2)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>$\phi/16$ (m = 3)</td></tr>
<tr><td>1</td><td>don’t care</td><td>don’t care</td><td>$\phi/32$ (m = 4)</td></tr>
</table>
m: Prescaler setting value ϕ: Internal system clock frequency</td></tr>
</table>

Caution Do not change the count clock during a transmit/receive operation.

Remark n = 0 to 2

CHAPTER 11 A/D CONVERTER

11.1 Features

○ Analog input: 8 channels
○ 10-bit A/D converter
○ On-chip A/D conversion result register (ADCR0 to ADCR7)
10 bits × 8
○ A/D conversion trigger mode
A/D trigger mode
Timer trigger mode
External trigger mode
○ Successive approximation method

11.2 Configuration

The A/D converter of the V850E/MS1 adopts the successive approximation method, and uses the A/D converter mode registers (ADM0, ADM1), and ADCRn register to perform A/D conversion operations (n = 0 to 7).

(1) Input circuit

Selects the analog input (ANI0 to ANI7) according to the mode set to the ADM0 and ADM1 registers and sends the input to the sample and hold register.

(2) Sample and hold circuit

The sample and hold circuit samples each of the analog input signals sequentially sent from the input circuit, and sends the sample to the voltage comparator. This circuit also holds the sampled analog input signal voltage during A/D conversion.

(3) Voltage comparator

The voltage comparator compares the analog input signal with the output voltage of the series resistor string.

(4) Series resistor string

The series resistor string is used to generate voltages to match analog inputs.

The series resistor string is connected between the reference voltage pin (AV_{REF}) for the A/D converter and the GND pin (AV_{SS}) for the A/D converter. To make 1,024 equal voltage steps between these 2 pins, it is configured from 1,023 equal resistors and 2 resistors with 1/2 of the resistance value.

The voltage tap of the series resistor string is selected by a tap selector controlled by a successive approximation register (SAR).

(5) Successive approximation register (SAR)

The SAR is a 10-bit register in which is set series resistor string voltage tap data, which have values that match analog input voltage values, 1 bit at a time beginning with the most significant bit (MSB).

If the data is set in the SAR all the way to the least significant bit (LSB) (A/D conversion completed), the contents of that SAR (conversion results) are held in the A/D conversion results register (ADCRn).

(6) A/D conversion results register (ADCRn)

The ADCR is a 10-bit register that holds A/D conversion results. Each time A/D conversion is completed, conversion results are loaded from the successive approximation register (SAR).

$\overline{\text{RESET}}$ input makes its contents undefined.

(7) Controller

Selects the analog input, generates the sample and hold circuit operation timing, and controls the conversion trigger according to the mode set to the ADM0 and ADM1 registers.

(8) ANI0 to ANI7 pins

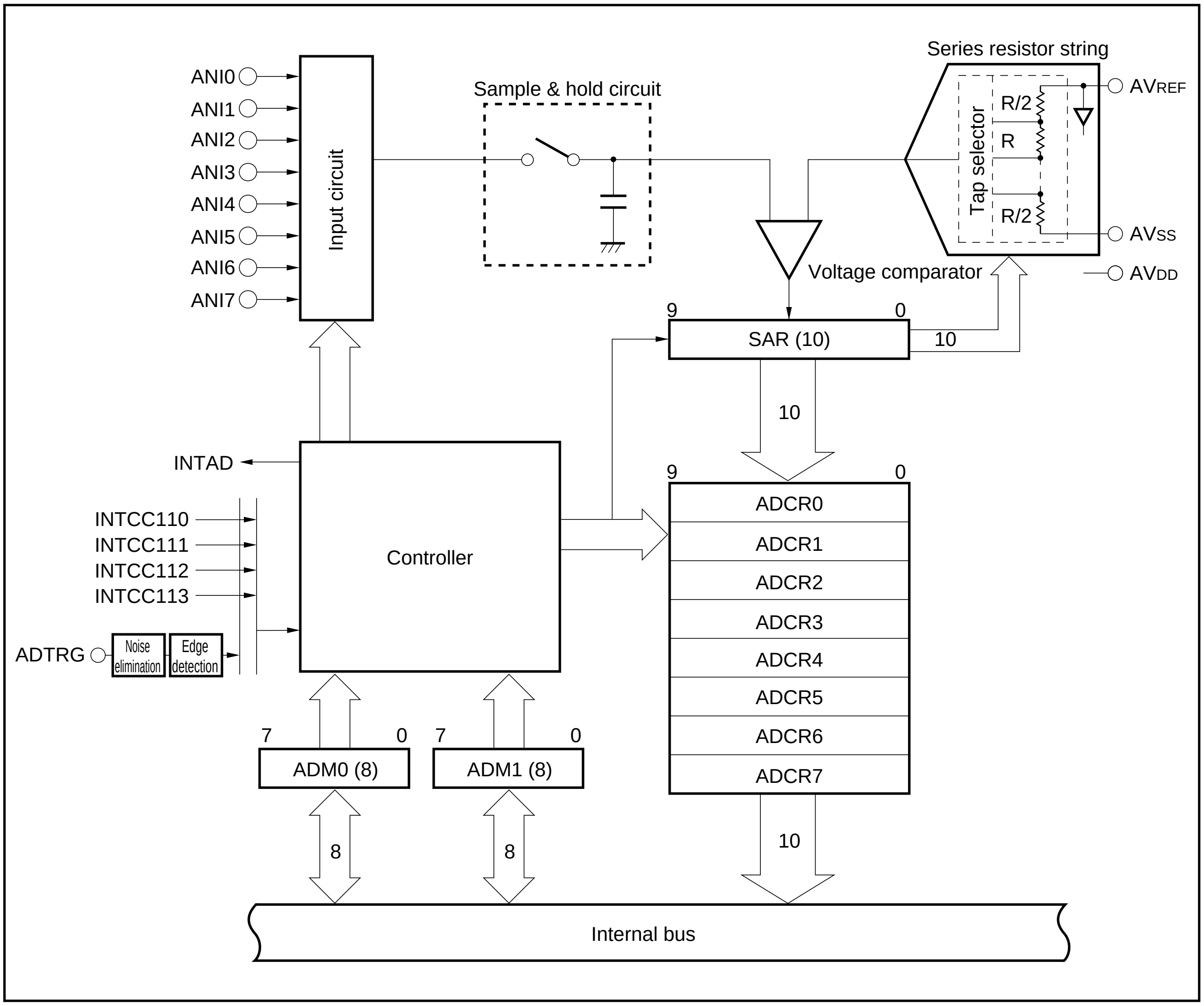
8-channel analog input pin for the A/D converter. Inputs the analog signal to be A/D converted.

Caution Make sure that the voltages input to ANI0 through ANI7 do not exceed the rated values. If a voltage higher than V_{DD} or lower than V_{SS} (even within the range of the absolute maximum ratings) is input to a channel, the conversion value of the channel is undefined, and the conversion values of the other channels may also be affected.

(9) AV_{REF} pin

Pin for inputting the reference voltage of the A/D converter. Converts signals input to the ANIn pin to digital signals based on the voltage applied between AV_{REF} and AV_{SS}.

Figure 11-1. A/D Converter Block Diagram



Cautions 1. When noise is generated from the analog input pins (ANI0 to ANI7) and the reference voltage input pin (AV$_{REF}$), it may cause an illegal conversion result.

In order to avoid this illegal conversion result influencing the system, software processing is required.

An example of the necessary software processing is as follows.

- **Use the average value of the A/D conversion results after obtaining several A/D conversion results.**
- **When an exceptional conversion result is obtained after performing A/D conversion several times consecutively, omit it and use the rest of the conversion results.**
- **When an A/D conversion result that indicates a system malfunction is obtained, be sure to recheck the abnormal generation before performing malfunction processing.**

2. Make sure not to append the voltage that extends the value between AV$_{SS}$ to AV$_{REF}$ to the pins used as A/D converter input pins.

11.3 Control Registers

(1) A/D converter mode register 0 (ADM0)

The ADM0 register is an 8-bit register that executes the selection of the analog input pin, specification of operation mode, and conversion operations.

This register can be read/written in 8- or 1-bit units, However, when the data is written to the ADM0 register during A/D conversion operations, the conversion operation is initialized and conversion is executed from the beginning. Bit 6 cannot be written and writing executed is ignored.

	7	6	5	4	3	2	1	0	Address	After reset
ADM0	CE	CS	BS	MS	0	ANIS2	ANIS1	ANIS0	FFFFF380H	00H

Bit Position	Bit Name	Function
7	CE	Convert Enable Enables or disables A/D conversion operation. 0: Disabled 1: Enabled
6	CS	Converter Status Indicates the status of A/D converter. This bit is read only. 0: Stops 1: Operates
5	BS	Buffer Select Specifies buffer mode in the select mode. 0: 1-buffer mode 1: 4-buffer mode
4	MS	Mode Select Specifies operation mode of A/D converter. 0: Scan mode 1: Select mode
2 to 0	ANIS2 to ANIS0	Analog Input Select Specifies analog input pin to A/D convert.

ANIS2	ANIS1	ANIS0	Select Mode		Scan Mode	
			A/D trigger mode	Timer trigger mode	A/D trigger mode	Timer trigger mode[Note]
0	0	0	ANI0	ANI0	ANI0	1
0	0	1	ANI1	ANI1	ANI0, ANI1	2
0	1	0	ANI2	ANI2	ANI0 to ANI2	3
0	1	1	ANI3	ANI3	ANI0 to ANI3	4
1	0	0	ANI4	Setting prohibited	ANI0 to ANI4	4 + ANI4
1	0	1	ANI5	Setting prohibited	ANI0 to ANI5	4 + ANI4, ANI5
1	1	0	ANI6	Setting prohibited	ANI0 to ANI6	4 + ANI4 to ANI6
1	1	1	ANI7	Setting prohibited	ANI0 to ANI7	4 + ANI4 to ANI7

Note In the timer trigger mode (4-trigger mode) during the scan mode, because the scanning sequence of the ANI0 to ANI3 pins is specified by the sequence in which the match signals are generated from the compare register, the number of trigger inputs should be specified instead of a certain analog input pin. When ANIS2 is set to 1, the scan mode shifts to A/D trigger mode after counting the trigger four times, and then starts converting.

Cautions 1. When the CE bit is 1 in the timer trigger mode and external trigger mode, the trigger signal standby state is set. To clear the CE bit, write 0 or reset.

In the A/D trigger mode, the conversion trigger is set by writing 1 to the CE bit. After the operation, when the mode is changed to the timer trigger mode or external trigger mode without clearing the CE bit, the trigger input standby state is set immediately after the change.

2. It takes 3 clocks for CS bit to become 1 after A/D conversion starts.

(2) A/D converter mode register 1 (ADM1)

The ADM1 register is an 8-bit register that specifies the conversion operation time and trigger mode.

This register can be read/written in 8- or 1-bit units. However, when the data is written to the ADM1 register during an A/D conversion operation, the conversion operation is initialized and conversion is executed from the beginning again.

	7	6	5	4	3	2	1	0	Address	After reset
ADM1	0	TRG2	TRG1	TRG0	0	FR2	FR1	FR0	FFFFF382H	07H

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>6 to 4</td><td>TRG2 to
TRG0</td><td>Trigger Mode
Specifies trigger mode.
<table>
<tr><th>TRG2</th><th>TRG1</th><th>TRG0</th><th>Trigger Mode</th></tr>
<tr><td>0</td><td>0</td><td>don't care</td><td>A/D trigger mode</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>Timer trigger mode (1-trigger mode)</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>Timer trigger mode (4-trigger mode)</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>External trigger mode</td></tr>
<tr><td colspan="3">Other than above</td><td>Setting prohibited</td></tr>
</table>
<b>Remark</b> The valid edge of the external input signal during the external trigger mode is specified by bits 7 and 6 (ES531, ES530) of the external interrupt mode register (INTM6). For details, refer to <b>7.3.8 (1) External interrupt mode registers 1 to 6 (INTM1 to INTM6)</b>.</td></tr>
<tr><td>2 to 0</td><td>FR2 to
FR0</td><td>Frequency
Specifies conversion operation time. These bits control the conversion time to be same value irrespective of oscillation frequency.
<table>
<tr><th rowspan="2">FR2</th><th rowspan="2">FR1</th><th rowspan="2">FR0</th><th rowspan="2">Number of Conversion Clocks</th><th colspan="4">Conversion Operation Time (μs)Note</th></tr>
<tr><th>φ =
40 MHz</th><th>φ =
33 MHz</th><th>φ =
25 MHz</th><th>φ =
16 MHz</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>48 clocks</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0</td><td>0</td><td>1</td><td>72 clocks</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>0</td><td>1</td><td>0</td><td>96 clocks</td><td>—</td><td>—</td><td>—</td><td>6.00</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>120 clocks</td><td>—</td><td>—</td><td>—</td><td>7.50</td></tr>
<tr><td>1</td><td>0</td><td>0</td><td>168 clocks</td><td>—</td><td>5.09</td><td>6.72</td><td>—</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>192 clocks</td><td>—</td><td>5.82</td><td>7.68</td><td>—</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>240 clocks</td><td>6.00</td><td>7.27</td><td>9.60</td><td>—</td></tr>
<tr><td>1</td><td>1</td><td>1</td><td>336 clocks</td><td>8.40</td><td>—</td><td>—</td><td>—</td></tr>
</table>
<b>Note</b> Figures under Conversion Operation Time are target values.
<b>Remark</b> φ: Internal system clock frequency
—: Setting prohibited</td></tr>
</table>

(3) A/D conversion result registers (ADCR0 to ADCR7, ADCR0H to ADCR7H)

The ADCRn register is a 10-bit register holding the A/D conversion results. It is provided with eight 10-bit registers (n = 0 to 7).

This register is read-only, in 16- or 8-bit units.

During 16-bit access to this register, the ADCRn register is specified, and during higher 8-bit access, the ADCRnH register is specified.

When reading the 10-bit data of A/D conversion results from the ADCRn register, only the lower 10 bits are valid and the higher 6 bits are always read as 0.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	After reset
ADCRn	0	0	0	0	0	0	ADn9	ADn8	ADn7	ADn6	ADn5	ADn4	ADn3	ADn2	ADn1	ADn0	FFFFF390H to FFFFF3ACH	Undefined

	7	6	5	4	3	2	1	0		
ADCRnH	ADn9	ADn8	ADn7	ADn6	ADn5	ADn4	ADn3	ADn2	FFFFF392H to FFFFF3AEH	Undefined

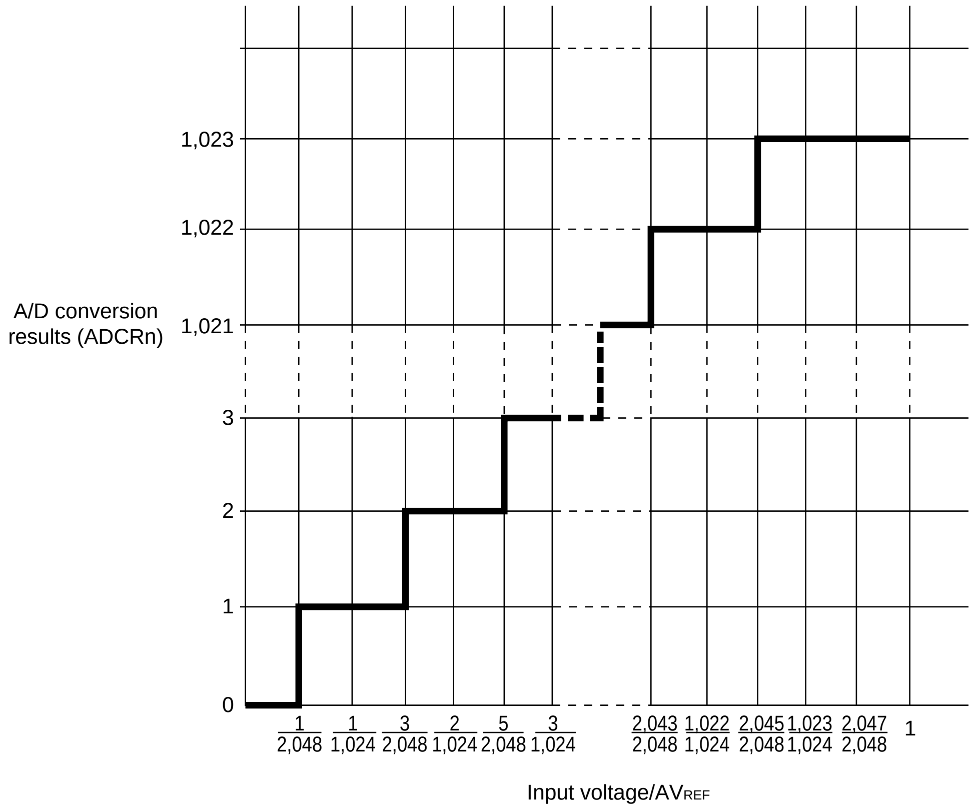
Remark n = 0 to 7

The correspondence between each analog input pin and the ADCRn register (except the 4-buffer mode) is shown below.

Analog Input Pin	ADCRn Register
ANI0	ADCR0, ADCR0H
ANI1	ADCR1, ADCR1H
ANI2	ADCR2, ADCR2H
ANI3	ADCR3, ADCR3H
ANI4	ADCR4, ADCR4H
ANI5	ADCR5, ADCR5H
ANI6	ADCR6, ADCR6H
ANI7	ADCR7, ADCR7H

Figure 11-2 shows the relationship between the analog input voltage and the A/D conversion results.

Figure 11-2. Relationship Between Analog Input Voltage and A/D Conversion Results



Remark n = 0 to 7

11.4 A/D Converter Operation

11.4.1 Basic operation of A/D converter

A/D conversion is executed in the following order.

(1) The selection of the analog input and specification of the operation mode, trigger mode, etc. should be set in the ADM0 and ADM1 registers[Note 1].
When the CE bit of the ADM0 register is set (1), A/D conversion starts in the A/D trigger mode. In the timer trigger mode and external trigger mode, the trigger standby state[Note 2] is set.

(2) The voltage generated from the voltage tap of the series resistor string and analog input are compared by the comparator.

(3) When the comparison of the 10 bits ends, the conversion results are stored in the ADCRn register. When A/D conversion is performed for the specified number of times, the A/D conversion end interrupt (INTAD) is generated (n = 0 to 7).

Notes 1. When the ADM0 and ADM1 registers are changed during an A/D conversion operation, the A/D conversion operation before the change is stopped and the conversion results are not stored in the ADCRn register.

2. In the timer trigger mode and external trigger mode, if the CE bit of the ADM0 register is set to 1, the mode changes to the trigger standby state. The A/D conversion operation is started by the trigger signal, and the trigger standby state is returned when the A/D conversion operation ends.

11.4.2 Operation mode and trigger mode

The A/D converter can specify various conversion operations by specifying the operation mode and trigger mode. The operation mode and trigger mode are set by the ADM0 and ADM1 registers.

The following shows the relationship between the operation mode and trigger mode.

Trigger Mode		Operation Mode		Setting Value		Analog Input
				ADM0 register	ADM1 register	
A/D trigger		Select	1 buffer	xx010xxxB	000x0xxxB	ANI0 to ANI7
			4 buffers	xx110xxxB	000x0xxxB	
		Scan		xxx00xxxB	000x0xxxB	
Timer trigger	1 trigger	Select	1 buffer	xx010xxxB	00100xxxB	ANI0 to ANI3
			4 buffers	xx110xxxB	00100xxxB	
		Scan		xxx00xxxB	00100xxxB	
	4 triggers	Select	1 buffer	xx010xxxB	00110xxxB	
			4 buffers	xx110xxxB	00110xxxB	
		Scan		xxx00xxxB	00110xxxB	
External trigger		Select	1 buffer	xx010xxxB	01100xxxB	
			4 buffers	xx110xxxB	01100xxxB	
		Scan		xxx00xxxB	01100xxxB	

(1) Trigger mode

There are three types of trigger modes that serve as the start timing of the A/D conversion processing: A/D trigger mode, timer trigger mode, and external trigger mode. The ANI0 to ANI3 pins are able to specify all of these modes, but pins ANI4 to ANI7 can only specify the A/D trigger mode. The timer trigger mode consists of the 1-trigger mode and 4-trigger mode as the sub-trigger mode. These trigger modes are set by the ADM1 register.

(a) A/D trigger mode

Generates the conversion timing of the analog input for the ANI0 to ANI7 pins inside the A/D converter unit. ANI4 to ANI7 pins are always set in this mode.

(b) Timer trigger mode

Specifies the conversion timing of the analog input set for the ANI0 to ANI3 pins using the values set to the TM11 compare register. This mode can only be specified by pins ANI0 to ANI3.

This register creates the analog input conversion timing by generating the match interrupts of the four capture/compare registers (CC110 to CC113) connected to the 16-bit TM11.

There are two types of sub-trigger modes: 1-trigger mode and 4-trigger mode.

- **1-trigger mode**
  Mode that uses one match interrupt from timer 11 as the A/D conversion start timing.

- **4-trigger mode**
  Mode that uses four match interrupts from timer 11 as the A/D conversion start timing.

(c) External trigger mode

Mode that specifies the conversion timing of the analog input to the ANI0 to ANI3 pins using the ADTRG pin. This mode can be specified only with ANI0 to ANI3 pins.

(2) Operation mode

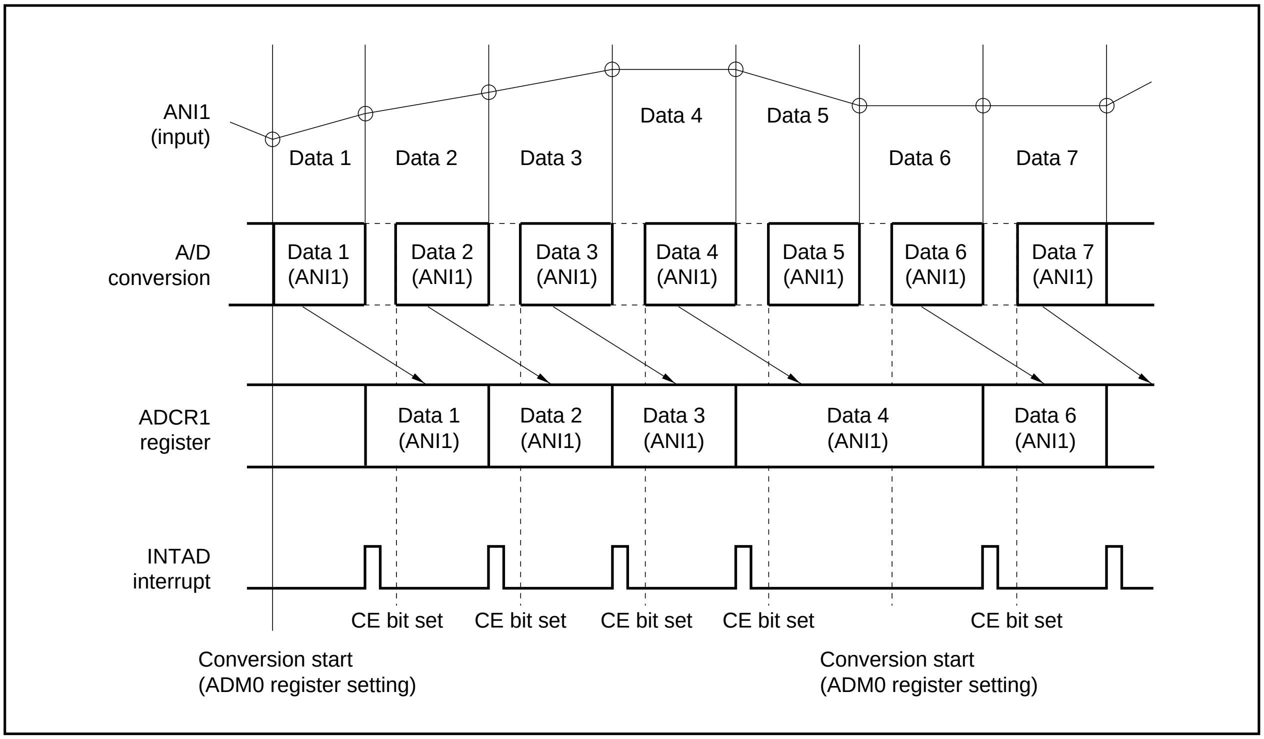
There are two types of operation modes that set the ANI0 to ANI7 pins: select mode and scan mode. The select mode has sub-modes including the 1-buffer mode and 4-buffer mode. These modes are set by the ADM0 register.

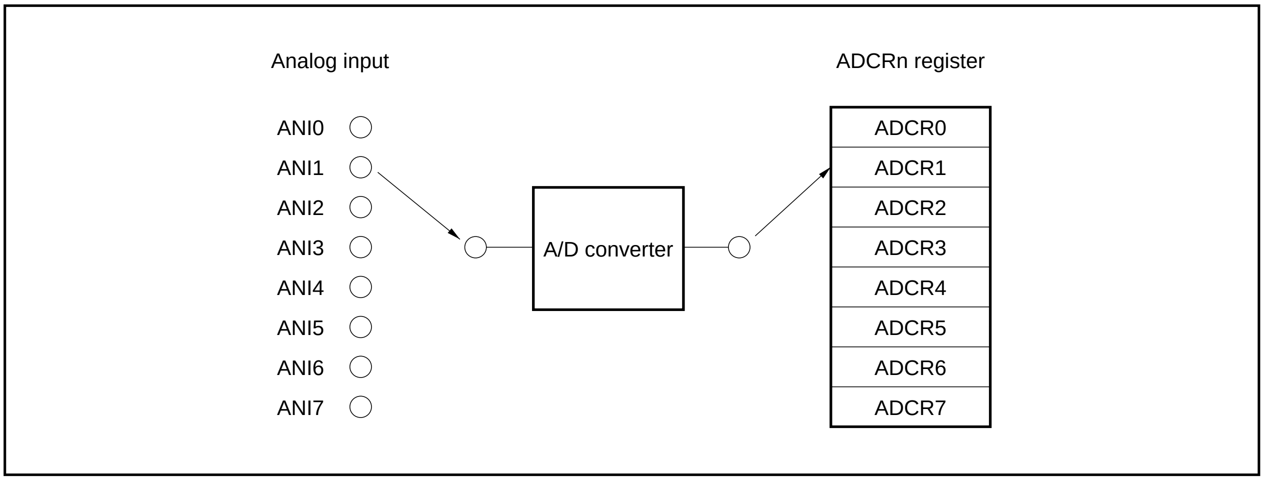
(a) Select mode

One analog input specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input (ANIn). For this mode, the 1-buffer mode and 4-buffer mode are provided for storing the A/D conversion results (n = 0 to 7).

- **1-buffer mode**
  One analog input specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input (ANIn). The ANIn and ADCRn registers correspond one to one, and an A/D conversion end interrupt (INTAD) is generated each time one A/D conversion ends.

★ **Figure 11-3. Select Mode Operation Timing: 1-Buffer Mode (ANI1)**

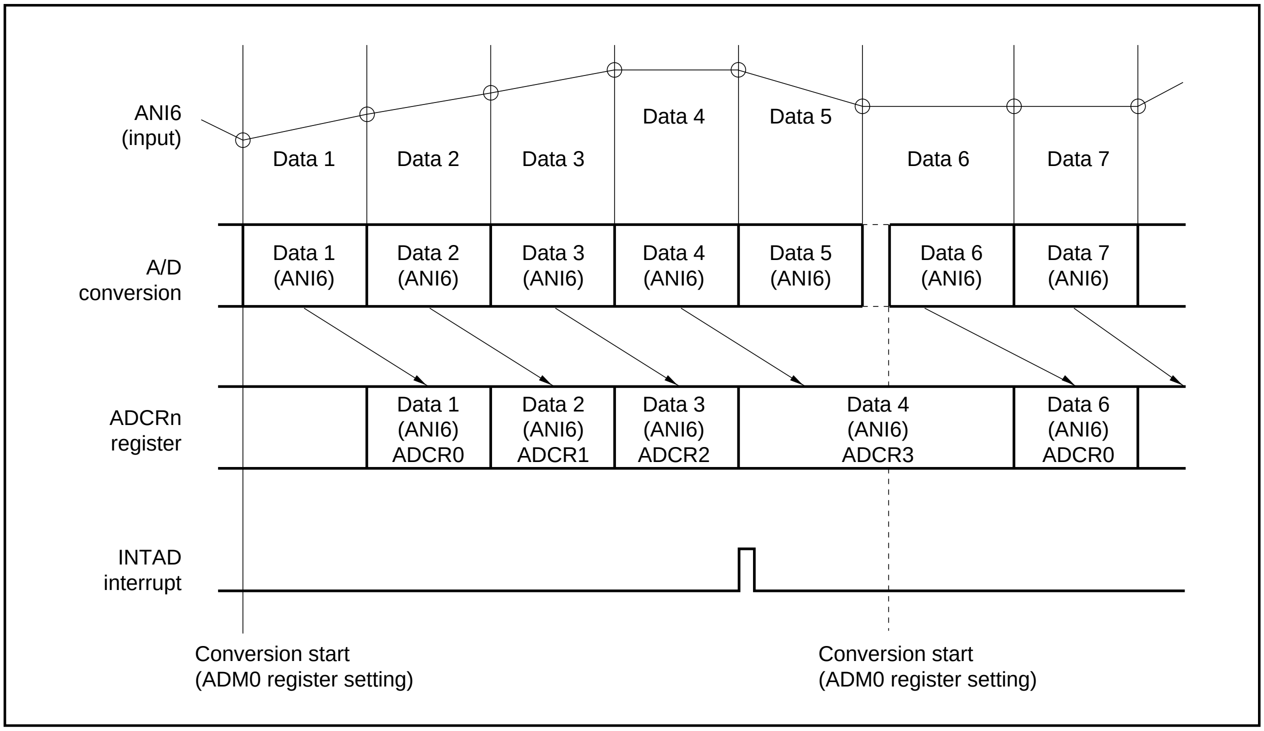


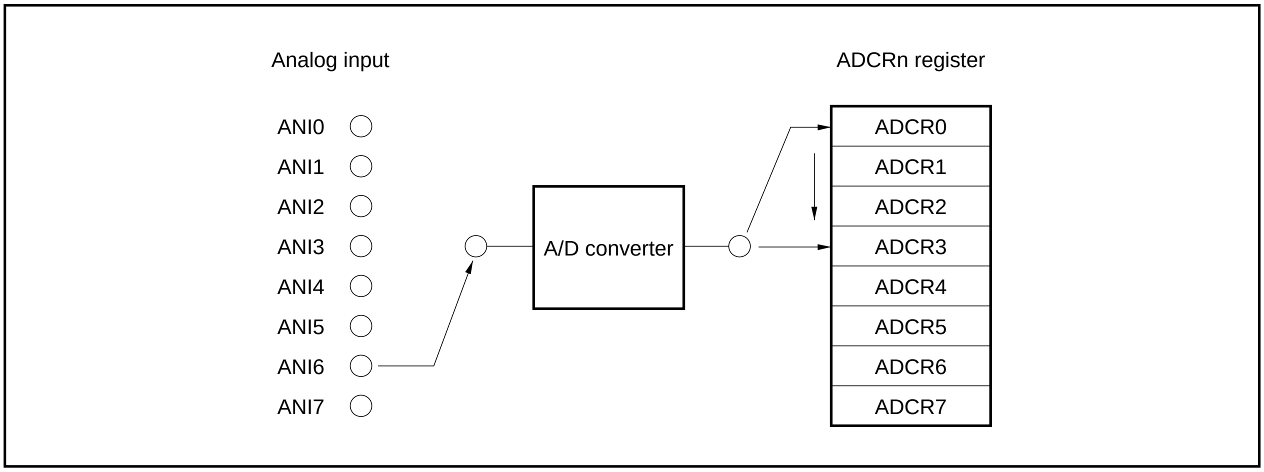


- **4-buffer mode**

  One analog input is A/D converted four times and the results are stored in the ADCR0 to ADCR3 registers. The A/D conversion end interrupt (INTAD) is generated when the four A/D conversions end.

Figure 11-4. Select Mode Operation Timing: 4-Buffer Mode (ANI6)

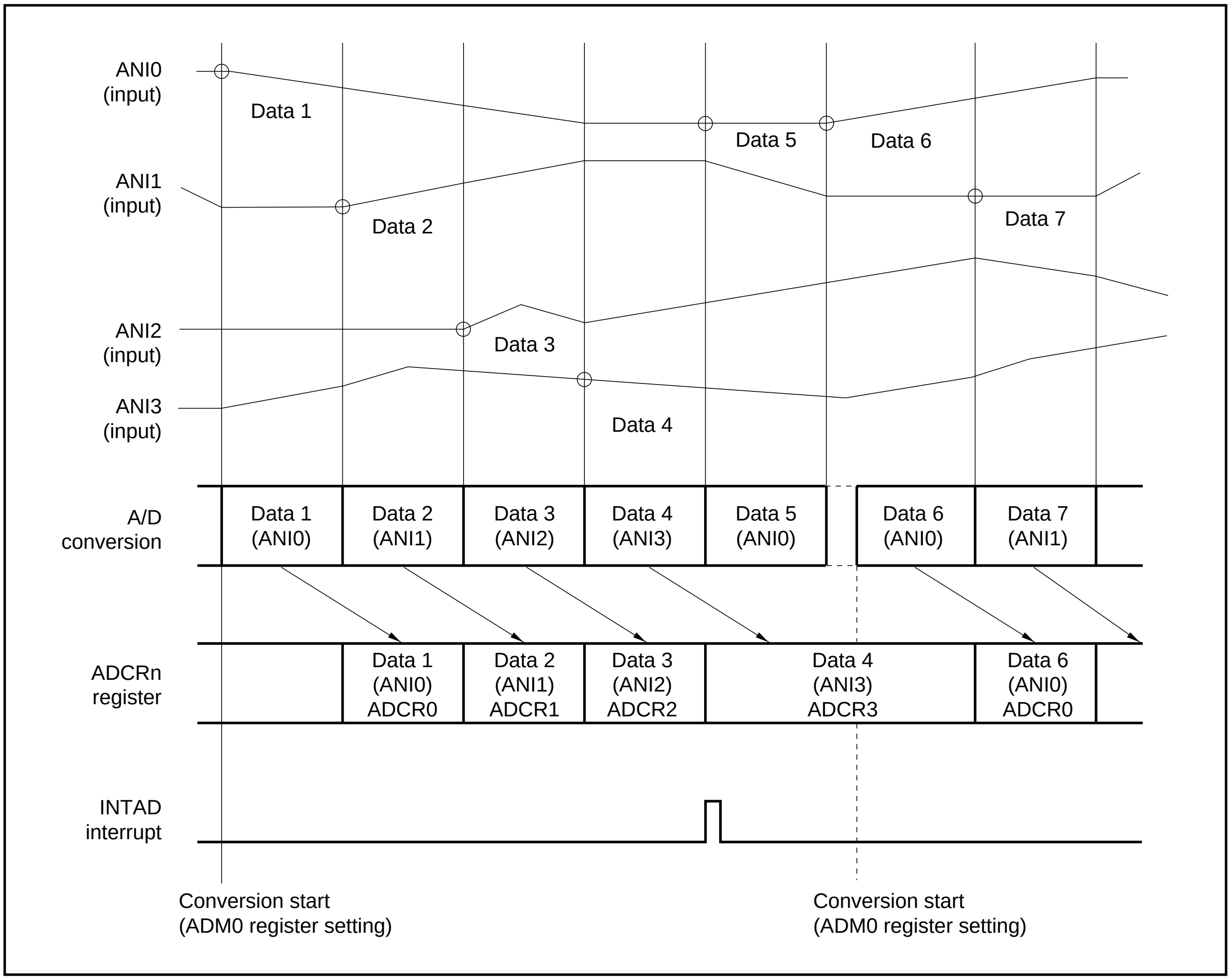


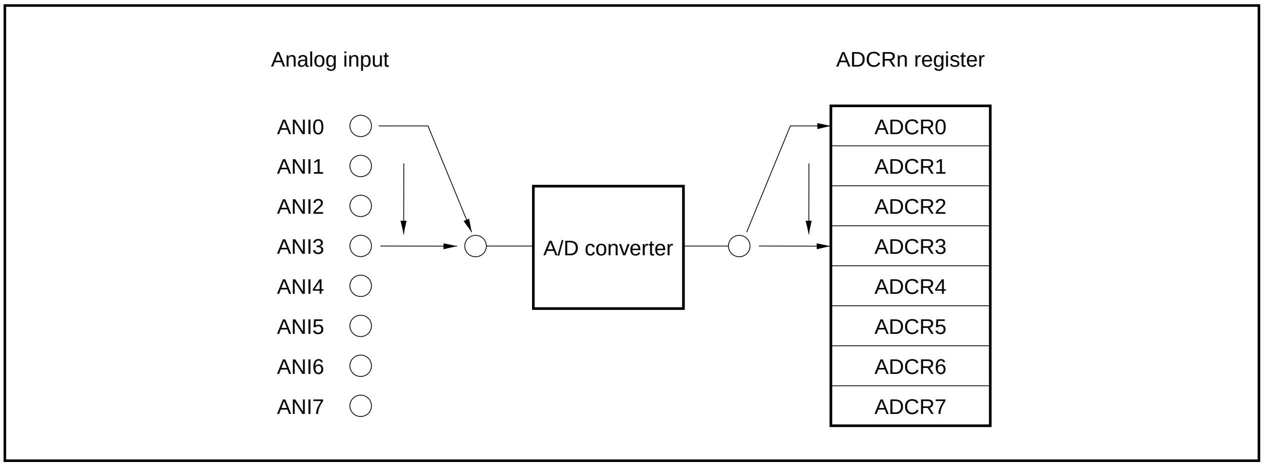


(b) Scan mode

Selects the analog inputs specified by the ADM0 register sequentially from the ANI0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRn register corresponding to the analog input. When the conversion of the specified analog input ends, the INTAD interrupt is generated.

Figure 11-5. Scan Mode Operation Timing: 4-Channel Scan (ANI0 to ANI3)





11.5 Operation in A/D Trigger Mode

When the CE bit of the ADM0 register is set to 1, A/D conversion starts.

11.5.1 Select mode operations

The analog input specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input. For the select mode, the 1-buffer mode and 4-buffer mode are supported according to the storing method of the A/D conversion results (n = 0 to 7).

(1) 1-buffer mode (A/D trigger select: 1-buffer)

One analog input is A/D converted once. The conversion results are stored in one ADCRn register. The analog input and ADCRn register correspond one to one.

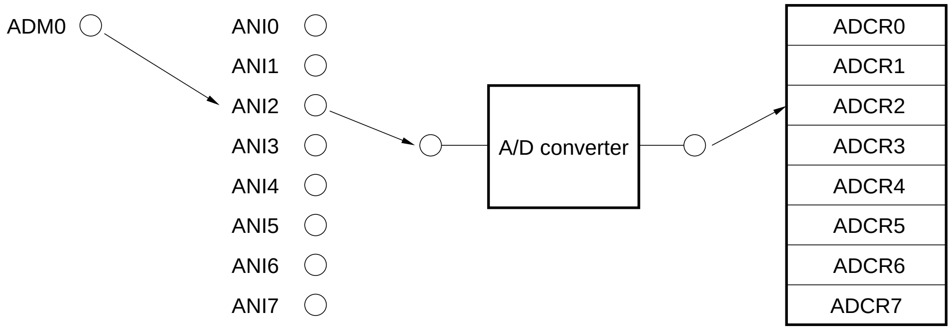
Each time an A/D conversion is executed, an INTAD interrupt is generated and the AD conversion terminates.

Analog Input	A/D Conversion Results Register
ANIn	ADCRn

(n = 0 to 7)

If 1 is written to the CE bit of the ADM0 register, A/D conversion can be restarted. This is most appropriate for applications in which the results of each first time A/D conversion are read.

Figure 11-6. Example of 1-Buffer Mode (A/D Trigger Select 1-Buffer) Operation



(1) CE bit of ADM0 is set to 1 (enable)

(2) ANI2 A/D conversion

(3) Conversion result is stored in ADCR2

(4) INTAD interrupt generation

(2) 4-buffer mode (A/D trigger select: 4-buffer)

One analog input is A/D converted four times and the results are stored in the four ADCR0 to ADCR3 registers. When four A/D conversions end, an INTAD interrupt is generated and A/D conversion terminates.

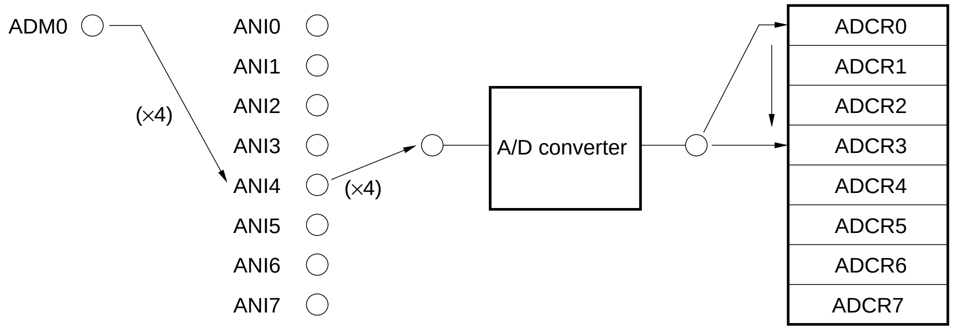
Analog Input	A/D Conversion Result Register
ANIn	ADCR0
ANIn	ADCR1
ANIn	ADCR2
ANIn	ADCR3

(n = 0 to 7)

If 1 is written in the CE bit of the ADM0 register, A/D conversion can be restarted.

This is most appropriate for applications that determine the average A/D conversion results.

Figure 11-7. Example of 4-Buffer Mode (A/D Trigger Select 4-Buffer) Operation



(1) CE bit of ADM0 is set to 1 (enable)

(2) ANI4 A/D conversion

(3) Conversion result is stored in ADCR0

(4) ANI4 A/D conversion

(5) Conversion result is stored in ADCR1

(6) ANI4 A/D conversion

(7) Conversion result is stored in ADCR2

(8) ANI4 A/D conversion

(9) Conversion result is stored in ADCR3

(10) INTAD interrupt generation

11.5.2 Scan mode operations

The analog inputs specified by the ADM0 register are selected sequentially from the ANI0 pin, and A/D conversion is executed. The A/D conversion results are stored in the ADCRn register corresponding to the analog input (n = 0 to 7).

When the conversion of all the specified analog input ends, the INTAD interrupt is generated, and A/D conversion terminates.

Analog Input	A/D Conversion Result Register
ANIn	ADCR0
|	|
ANIn[Note]	ADCRn

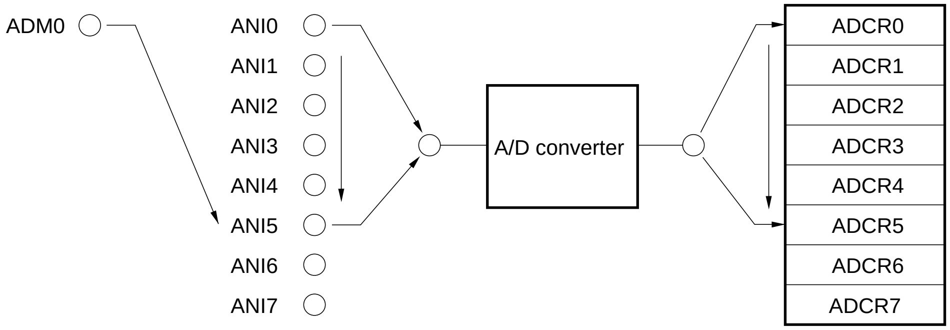
(n = 0 to 7)

Note Set in the ANIS0 to ANIS2 bits of the ADM0 register.

If 1 is written in the CE bit of the ADM0 register, A/D conversion can be restarted.

This is most appropriate for applications that are constantly monitoring multiple analog inputs.

Figure 11-8. Example of Scan Mode (A/D Trigger Scan) Operation



(1) CE bit of ADM0 is set to 1 (enable)

(2) ANI0 A/D conversion

(3) Conversion result is stored in ADCR0

(4) ANI1 A/D conversion

(5) Conversion result is stored in ADCR1

(6) ANI2 A/D conversion

(7) Conversion result is stored in ADCR2

(8) ANI3 A/D conversion

(9) Conversion result is stored in ADCR3

(10) ANI4 A/D conversion

(11) Conversion result is stored in ADCR4

(12) ANI5 A/D conversion

(13) Conversion result is stored in ADCR5

(14) INTAD interrupt generation

11.6 Operation in Timer Trigger Mode

The A/D converter is the match interrupt signal of the TM11 compare register, and can set conversion timings to a maximum of four channel analog inputs (ANI0 to ANI3).

TM11 and four capture/compare registers (CC110 to CC113) are used for the timer for specifying the analog conversion trigger.

The following two modes are provided according to the value set in the TUM11 register.

(1) 1-shot mode

To use the 1-shot mode, the OST bit of the TUM11 register should be set to 1 (1-shot mode).

When the A/D conversion period is longer than the TM11 period, the TM11 generates an overflow, holds 0000H, and stops. Thereafter, TM11 does not output the match interrupt signal (A/D conversion trigger) of the compare register, and the A/D converter also enters the A/D conversion standby state. The TM11 count operation restarts when the valid edge of the TCLR11 pin input is detected or when 1 is written to the CE11 bit of the TMC11 register.

(2) Loop mode

To use the loop mode, the OST bit of the TUM11 register should be set to 0 (normal mode).

When the TM11 generates an overflow, the TM11 starts counting from 0000H again, and the match interrupt signal (A/D conversion trigger) of the compare register is repeatedly output and A/D conversion is also repeated.

11.6.1 Select mode operations

One analog input (ANI0 to ANI3) specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input. For the select mode, the 1-buffer mode and 4-buffer mode are provided according to the storing method of the A/D conversion results (n = 0 to 3).

(1) 1-buffer mode operations (Timer trigger select: 1-buffer)

One analog input is A/D converted once and the conversion results are stored in one ADCRn register.

There are two modes in 1-buffer modes, the 1-trigger mode and 4-trigger mode, according to the number of triggers.

(a) 1-trigger mode (Timer trigger select: 1-buffer, 1-trigger)

One analog input is A/D converted once using the trigger of the match interrupt signal (INTCC110) and the results are stored in one ADCRn register.

An INTAD interrupt is generated for each A/D conversion and A/D conversion terminates.

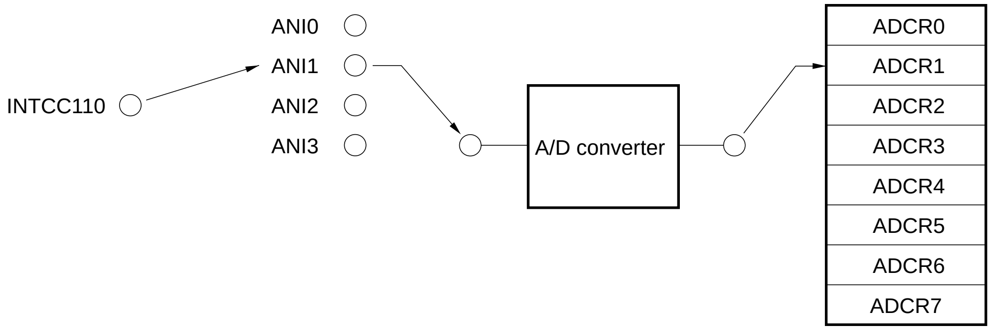
Trigger	Analog Input	A/D Conversion Result Register
INTCC110 interrupt	ANIn	ADCRn

(n = 0 to 3)

When the TM11 is set to the 1-shot mode, A/D conversion ends after one conversion. To restart A/D conversion, input the valid edge to the TCLR11 pin or write 1 to the CE11 bit of the TMC11 register.

When set to the loop mode, unless the CE bit of the ADM0 register is set to 0, A/D conversion is repeated each time the match interrupt is generated.

Figure 11-9. Example of 1-Trigger Mode (Timer Trigger Select 1-Buffer 1-Trigger) Operation



(1) CE bit of ADM0 is set to 1 (enable)
(2) CC110 compare generation
(3) ANI1 A/D conversion
(4) Conversion result is stored in ADCR1
(5) INTAD interrupt generation

(b) 4-trigger mode (Timer trigger select: 1-buffer, 4-trigger)

One analog input is A/D converted four times using four match interrupt signals (INTCC110 to INTCC113) as triggers and the results are stored in one ADCRn register. The INTAD interrupt is generated with each A/D conversion, and the CS bit of the ADM0 register is reset (0). The results of one A/D conversion are held by the ADCRn register until the next A/D conversion ends. Perform transmission of the conversion results to the memory and other operations using the INTAD interrupt after each A/D conversion ends.

Trigger	Analog Input	A/D Conversion Result Register
INTCC110 interrupt	ANIn	ADCRn
INTCC111 interrupt	ANIn	ADCRn
INTCC112 interrupt	ANIn	ADCRn
INTCC113 interrupt	ANIn	ADCRn

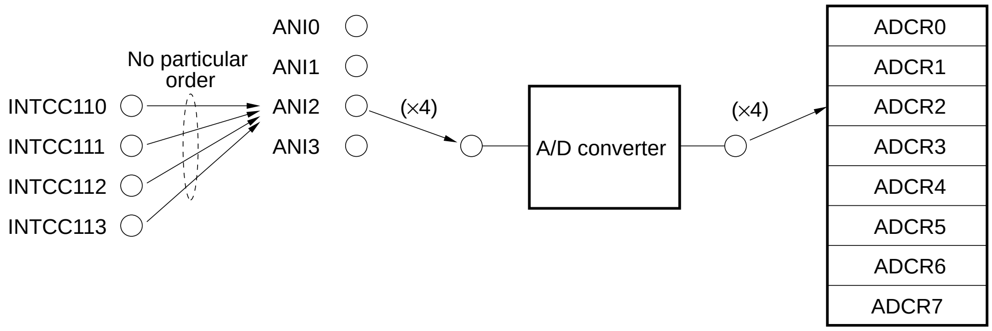
(n = 0 to 3)

When the TM11 is set to the 1-shot mode, A/D conversion ends after four conversions. To restart A/D conversion, input the valid edge to the TCLR11 pin or write 1 to the CE11 bit of the TMC11 register to restart the TM11. When the first match interrupt after TM11 is restarted is generated, the CS bit is set (1) and A/D conversion is started.

When set to the loop mode, unless the CE bit of the ADM0 register is set to 0, A/D conversion is repeated each time the match interrupt is generated.

The match interrupts (INTCC110 to INTCC113) can be generated in any order. The same trigger, even when it enters several times consecutively, is accepted as a trigger each time.

Figure 11-10. Example of 4-Trigger Mode (Timer Trigger Select 1-Buffer 4-Trigger) Operation



(1) CE bit of ADM0 is set to 1 (enable)
(2) CC112 compare generation (random)
(3) ANI2 A/D conversion
(4) Conversion result is stored in ADCR2
(5) INTAD interrupt generation
(6) CC111 compare generation (random)
(7) ANI2 A/D conversion
(8) Conversion result is stored in ADCR2
(9) INTAD interrupt generation
(10) CC113 compare generation (random)
(11) ANI2 A/D conversion
(12) Conversion result is stored in ADCR2
(13) INTAD interrupt generation
(14) CC110 compare generation (random)
(15) ANI2 A/D conversion
(16) Conversion result is stored in ADCR2
(17) INTAD interrupt generation

(2) 4-buffer mode operations (Timer trigger select: 4-buffer)

One analog input is A/D converted four times, and the results are stored in the ADCR0 to ADCR3 registers. There are two 4-buffer modes, 1-trigger mode and 4-trigger mode, according to the number of triggers.

This mode is suitable for applications that calculate the average of the A/D conversion result.

(a) 1-trigger mode

One analog input is A/D converted four times using the match interrupt signal (INTCC110) as a trigger, and the results are stored in the ADCR0 to ADCR3 registers.

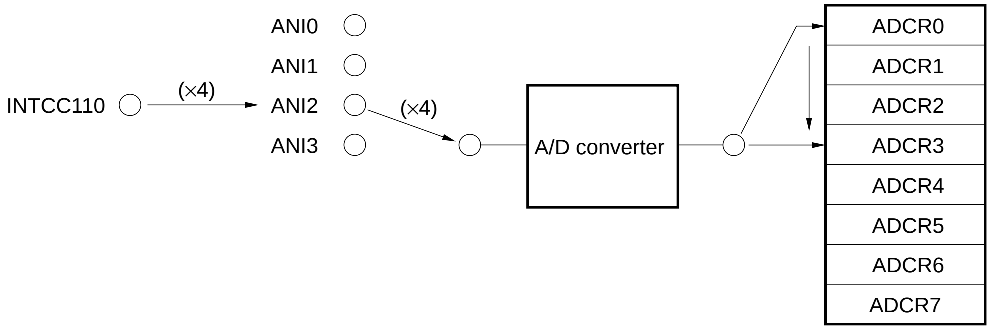
An INTAD interrupt is generated when the four A/D conversions end and A/D conversion terminates.

Trigger	Analog Input	A/D Conversion Result Register
INTCC110 interrupt	ANIn	ADCR0
INTCC110 interrupt	ANIn	ADCR1
INTCC110 interrupt	ANIn	ADCR2
INTCC110 interrupt	ANIn	ADCR3

(n = 0 to 3)

When the TM11 is set to the 1-shot mode, and less than four match interrupts are generated, if the CE bit is set to 0, the INTAD interrupt is not generated and the standby state is set.

Figure 11-11. Example of 1-Trigger Mode (Timer Trigger Select 4-Buffer 1-Trigger) Operation



(1) CE bit of ADM0 is set to 1 (enable)
(2) CC110 compare generation
(3) ANI2 A/D conversion
(4) Conversion result is stored in ADCR0
(5) CC110 compare generation
(6) ANI2 A/D conversion
(7) Conversion result is stored in ADCR1
(8) CC110 compare generation
(9) ANI2 A/D conversion
(10) Conversion result is stored in ADCR2
(11) CC110 compare generation
(12) ANI2 A/D conversion
(13) Conversion result is stored in ADCR3
(14) INTAD interrupt generation

(b) 4-trigger mode

One analog input is A/D converted four times using four match interrupt signals (INTCC110 to INTCC113) as triggers and the results are stored in four ADCRn registers. The INTAD interrupt is generated when the four A/D conversions end, the CS bit is reset (0), and A/D conversion terminates.

Trigger	Analog Input	A/D Conversion Result Register
INTCC110 interrupt	ANIn	ADCR0
INTCC111 interrupt	ANIn	ADCR1
INTCC112 interrupt	ANIn	ADCR2
INTCC113 interrupt	ANIn	ADCR3

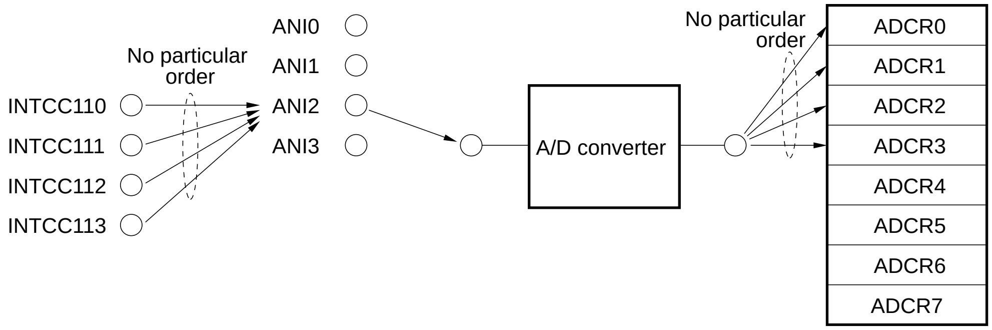
(n = 0 to 3)

When the TM11 is set to the 1-shot mode, A/D conversion ends after four conversions. To restart A/D conversion, input the valid edge to the TCLR11 pin or write 1 to the CE11 bit of the TMC11 register to restart the TM11. When the first match interrupt after TM11 is restarted is generated, the CS bit is set (1) and A/D conversion is started.

When set to the loop mode, unless the CE bit is set to 0, A/D conversion is repeated each time the match interrupt is generated.

Whichever the order of occurrence of match interrupts (INTCC110 to INTCC113), there is no problem, and the conversion results are stored in the ADCRn register corresponding to the input trigger. Also, even in cases where the same trigger is input continuously, it is received as a trigger.

Figure 11-12. Example of 4-Trigger Mode (Timer Trigger Select 4-Buffer 4-Trigger) Operation



(1) CE bit of ADM0 is set to 1 (enable)
(2) CC111 compare generation (random)
(3) ANI2 A/D conversion
(4) Conversion result is stored in ADCR1
(5) CC113 compare generation (random)
(6) ANI2 A/D conversion
(7) Conversion result is stored in ADCR3
(8) CC112 compare generation (random)
(9) ANI2 A/D conversion
(10) Conversion result is stored in ADCR2
(11) CC110 compare generation (random)
(12) ANI2 A/D conversion
(13) Conversion result is stored in ADCR0
(14) INTAD interrupt generation

11.6.2 Scan mode operations

The analog inputs specified by the ADM0 register are selected sequentially from the ANI0 pin and A/D converted for the specified number of times using the match interrupt signal as a trigger.

In the conversion operation, first the analog input lower channels (ANI0 to ANI3) are A/D converted for the specified number of times. In the ADM0 register, if the lower channels (ANI0 to ANI3) of the analog input are set so that they are scanned, and when the set number of A/D conversions ends, the INTAD interrupt is generated and A/D conversion ends.

When the higher channels (ANI4 to ANI7) of the analog input are set so that they are scanned in the ADM0 register, after the conversion of the lower channel ends, the mode is shifted to the A/D trigger mode, and the remaining A/D conversions are executed.

The conversion results are stored in the ADCRn register corresponding to the analog input. When the conversion of all the specified analog inputs has ended, the INTAD interrupt is generated and A/D conversion terminates (n = 0 to 7).

There are two scan modes, 1-trigger mode and 4-trigger mode, according to the number of triggers.

This is most appropriate for applications that are constantly monitoring multiple analog inputs.

(1) 1-trigger mode (Timer trigger scan: 1-trigger)

The analog inputs are A/D converted for the specified number of times using the match interrupt signal (INTCC110) as a trigger.

The analog input and ADCRn register correspond one to one.

When all the A/D conversions specified have ended, the INTAD interrupt is generated and A/D conversion ends.

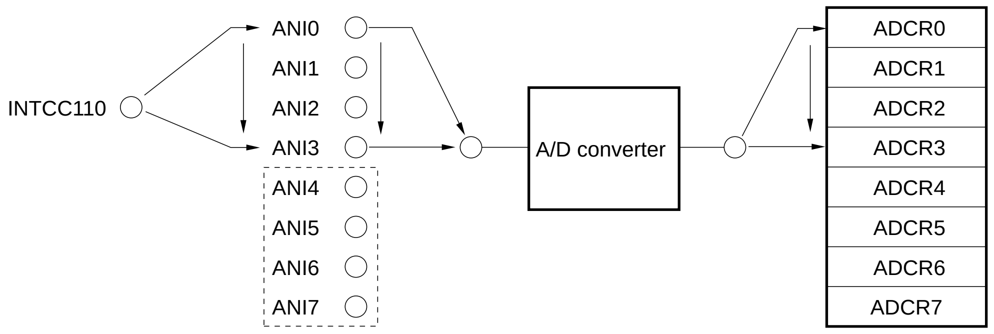
Trigger	Analog Input	A/D Conversion Result Register
INTCC110 interrupt	ANI0	ADCR0
INTCC110 interrupt	ANI1	ADCR1
INTCC110 interrupt	ANI2	ADCR2
INTCC110 interrupt	ANI3	ADCR3
(A/D trigger mode)	ANI4	ADCR4
	ANI5	ADCR5
	ANI6	ADCR6
	ANI7	ADCR7

When the match interrupt is generated after all the specified A/D conversions end, A/D conversion is restarted.

When the TM11 is set to the 1-shot mode, and less than a specified number of match interrupts are generated, if the CE bit is set to 0, the INTAD interrupt is not generated and the standby state is set.

Figure 11-13. Example of 1-Trigger Mode (Timer Trigger Scan 1-Trigger) Operation

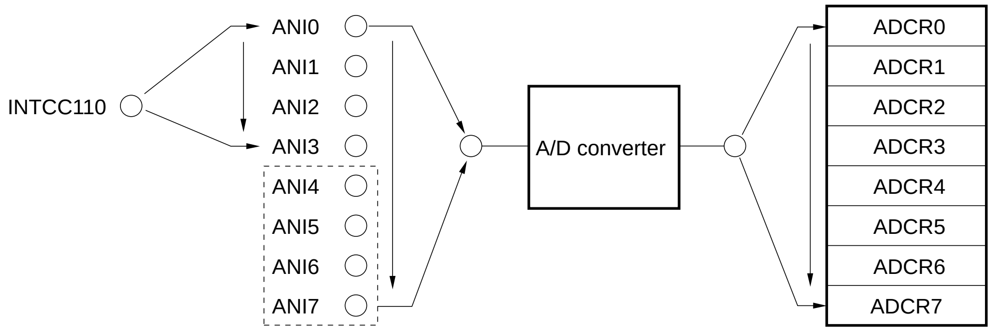
(a) Setting when scanning ANI0 to ANI3



(1) CE bit of ADM0 is set to 1 (enable)
(2) CC110 compare generation
(3) ANI0 A/D conversion
(4) Conversion result is stored in ADCR0
(5) CC110 compare generation
(6) ANI1 A/D conversion
(7) Conversion result is stored in ADCR1
(8) CC110 compare generation
(9) ANI2 A/D conversion
(10) Conversion result is stored in ADCR2
(11) CC110 compare generation
(12) ANI3 A/D conversion
(13) Conversion result is stored in ADCR3
(14) INTAD interrupt generation

Caution The analog input enclosed in the broken lines cannot be used with INTCC11n as the trigger (n = 0 to 3). When a setting is made to scan ANI0 to ANI7, ANI4 to ANI7 are converted in A/D trigger mode (see (b)).

(b) Setting when scanning ANI0 to ANI7



(1) to (13) Same as (a)
(14) ANI4 A/D conversion
(15) Conversion result is stored in ADCR4
(16) ANI5 A/D conversion
(17) Conversion result is stored in ADCR5
(18) ANI6 A/D conversion
(19) Conversion result is stored in ADCR6
(20) ANI7 A/D conversion
(21) Conversion result is stored in ADCR7
(22) INTAD interrupt generation

(2) 4-trigger mode

The analog inputs are A/D converted for the number of times specified using the match interrupt signal (INTCC110 to INTCC113) as a trigger.

The analog input and ADCRn register correspond one to one.

When all the A/D conversions specified have ended, the INTAD interrupt is generated and A/D conversion ends.

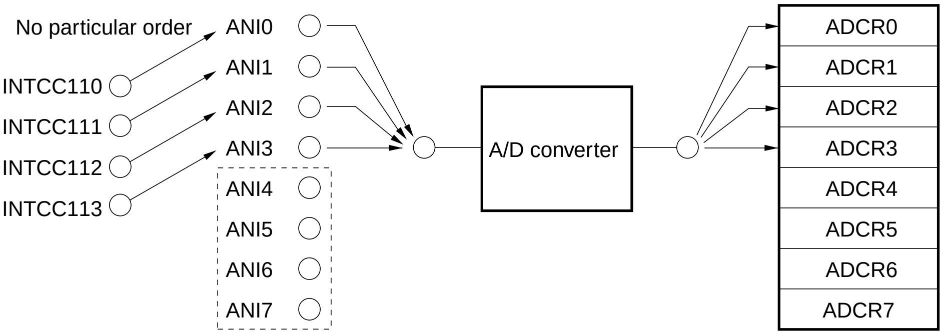
Trigger	Analog Input	A/D Conversion Result Register
INTCC110 interrupt	ANI0	ADCR0
INTCC111 interrupt	ANI1	ADCR1
INTCC112 interrupt	ANI2	ADCR2
INTCC113 interrupt	ANI3	ADCR3
(A/D trigger mode)	ANI4	ADCR4
	ANI5	ADCR5
	ANI6	ADCR6
	ANI7	ADCR7

To restart conversion when TM11 is set to the 1-shot mode, restart TM11. If set to the loop mode and the CE bit is 1, A/D conversion is restarted when a match interrupt is generated after conversion ends.

The match interrupt can be generated in any order. However, because the trigger signal and the analog input correspond one to one, the scanning sequence is determined according to the order in which the match signals of the compare register are generated.

Figure 11-14. Example of 4-Trigger Mode (Timer Trigger Scan 4-Trigger) Operation

(a) Setting when scanning ANI0 to ANI3



(1) CE bit of ADM0 is set to 1 (enable)

(2) CC111 compare generation (random)

(3) ANI1 A/D conversion

(4) Conversion result is stored in ADCR1

(5) CC113 compare generation (random)

(6) ANI3 A/D conversion

(7) Conversion result is stored in ADCR3

(8) CC110 compare generation (random)

(9) ANI0 A/D conversion

(10) Conversion result is stored in ADCR0

(11) CC112 compare generation (random)

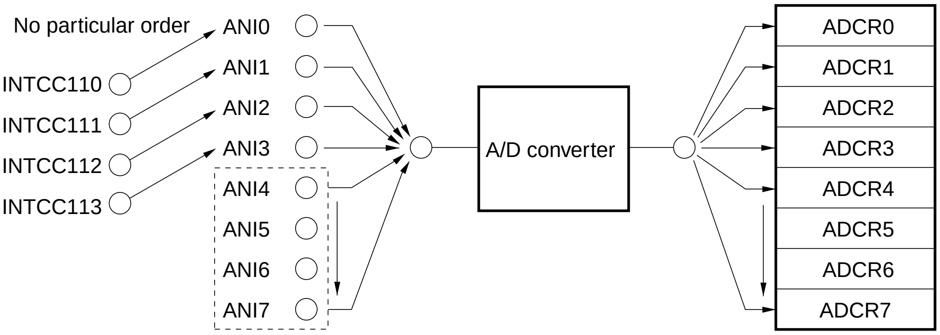
(12) ANI2 A/D conversion

(13) Conversion result is stored in ADCR2

(14) INTAD interrupt generation

Caution The analog input enclosed in the broken lines cannot be used with INTCC11n as the trigger (n = 0 to 3). When a setting is made to scan ANI0 to ANI7, ANI4 to ANI7 are converted in A/D trigger mode (see (b)).

(b) Setting when scanning ANI0 to ANI7



(1) to (13) Same as (a)

(14) ANI4 A/D conversion

(15) Conversion result is stored in ADCR4

(16) ANI5 A/D conversion

(17) Conversion result is stored in ADCR5

(18) ANI6 A/D conversion

(19) Conversion result is stored in ADCR6

(20) ANI7 A/D conversion

(21) Conversion result is stored in ADCR7

(22) INTAD interrupt generation

11.7 Operation in External Trigger Mode

In the external trigger mode, the analog inputs (ANI0 to ANI3) are A/D converted by the ADTRG pin input timing.

The ADTRG pin is also used as the P127 and INTP153 pins. To set the external trigger mode, set the PMC127 bit of the PMC12 register to 1 and bits TRG2 to TRG0 of the ADM1 register to 110.

For the valid edge of the external input signal in the external trigger mode, the rising edge, falling edge, or both rising and falling edges can be specified using bits ES531 and ES530 of the INTM6 register. For details, refer to **7.3.8 (1) External interrupt mode registers 1 to 6 (INTM1 to INTM6)**.

11.7.1 Select mode operations (external trigger select)

One analog input (ANI0 to ANI3) specified by the ADM0 register is A/D converted. The conversion results are stored in the ADCRn register corresponding to the analog input. There are two select modes, 1-buffer mode and 4-buffer mode, storing the conversion results (n = 0 to 3).

(1) 1-buffer mode (External trigger select: 1-buffer)

One analog input is A/D converted using the ADTRG signal as a trigger. The conversion results are stored in one ADCRn register. The analog input and the A/D conversion results register correspond one to one. INTAD interrupts are generated after each A/D conversion, and A/D conversion ends.

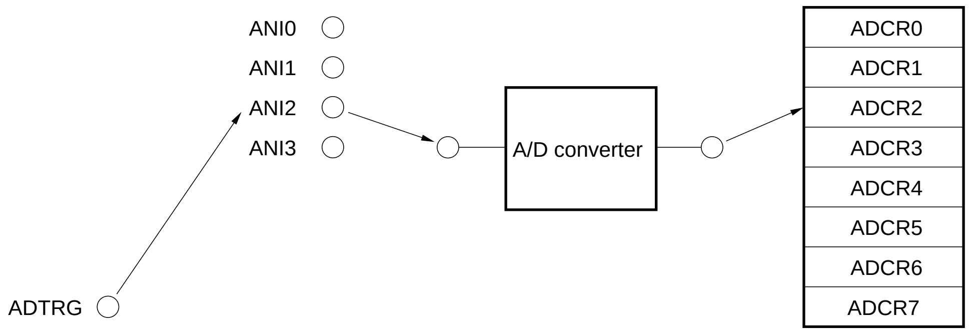
Trigger	Analog Input	A/D Conversion Result Register
ADTRG signal	ANIn	ADCRn

(n = 0 to 3)

While the CE bit of the ADM0 register is 1, the A/D conversion is repeated every time a trigger is input from the ADTRG pin.

This is most appropriate for applications that read the results each time there is an A/D conversion.

Figure 11-15. Example of 1-Buffer Mode (External Trigger Select 1-Buffer) Operation



(1) CE bit of ADM0 is set to 1 (enable)

(2) External trigger generation

(3) ANI2 A/D conversion

(4) Conversion result is stored in ADCR2

(5) INTAD interrupt generation

(2) 4-buffer mode (External trigger select: 4-buffer)

One analog input is A/D converted four times using the ADTRG signal as a trigger and the results are stored in the ADCR0 to ADCR3 registers. The INTAD interrupt is generated and conversion ends when the four A/D conversions end.

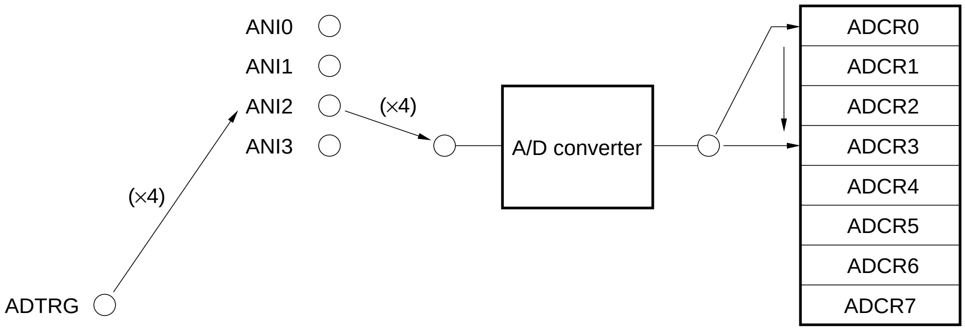
Trigger	Analog Input	A/D Conversion Result Register
ADTRG signal	ANIn	ADCR0
ADTRG signal	ANIn	ADCR1
ADTRG signal	ANIn	ADCR2
ADTRG signal	ANIn	ADCR3

(n = 0 to 3)

While the CE bit of the ADM0 register is 1, A/D conversion is repeated every time a trigger is input from the ADTRG pin.

This is most appropriate for applications that determine the average A/D conversion results.

Figure 11-16. Example of 4-Buffer Mode (External Trigger Select 4-Buffer) Operation



(1) CE bit of ADM0 is set to 1 (enable)
(2) External trigger generation
(3) ANI2 A/D conversion
(4) Conversion result is stored in ADCR0
(5) External trigger generation
(6) ANI2 A/D conversion
(7) Conversion result is stored in ADCR1
(8) External trigger generation
(9) ANI2 A/D conversion
(10) Conversion result is stored in ADCR2
(11) External trigger generation
(12) ANI2 A/D conversion
(13) Conversion result is stored in ADCR3
(14) INTAD interrupt generation

11.7.2 Scan mode operations (external trigger scan)

The analog inputs specified by the ADM0 register are selected sequentially from the ANI0 pin using the ADTRG signal as a trigger, and A/D converted. The A/D conversion results are stored in the ADCRn register corresponding to the analog input (n = 0 to 7).

When the lower 4 channels (ANI0 to ANI3) of the analog input are set so that they are scanned in the ADM0 register, the INTAD interrupt is generated when the number of A/D conversions specified end, and A/D conversion ends.

When the higher 4 channels (ANI4 to ANI7) of the analog input are set so that they are scanned in the ADM0 register, after the conversion of the lower 4 channels ends, the mode is shifted to the A/D trigger mode, and the remaining A/D conversions are executed. The conversion results are stored in the ADCRn register corresponding to the analog input.

Trigger	Analog Input	A/D Conversion Result Register
ADTRG signal	ANI0	ADCR0
ADTRG signal	ANI1	ADCR1
ADTRG signal	ANI2	ADCR2
ADTRG signal	ANI3	ADCR3
(A/D trigger mode)	ANI4	ADCR4
	ANI5	ADCR5
	ANI6	ADCR6
	ANI7	ADCR7

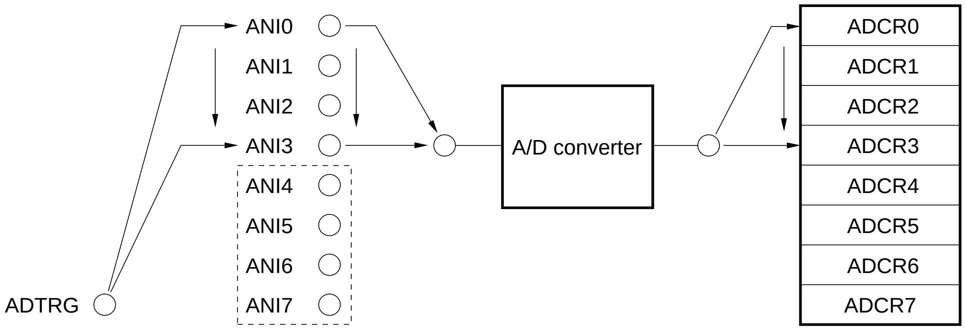
When the conversion of all the specified analog inputs ends, the INTAD interrupt is generated and A/D conversion ends.

When a trigger is input to the ADTRG pin while the CE bit of the ADM0 register is 1, the A/D conversion is started again.

This is most appropriate for applications that are constantly monitoring multiple analog inputs.

Figure 11-17. Example of Scan Mode (External Trigger Scan) Operation

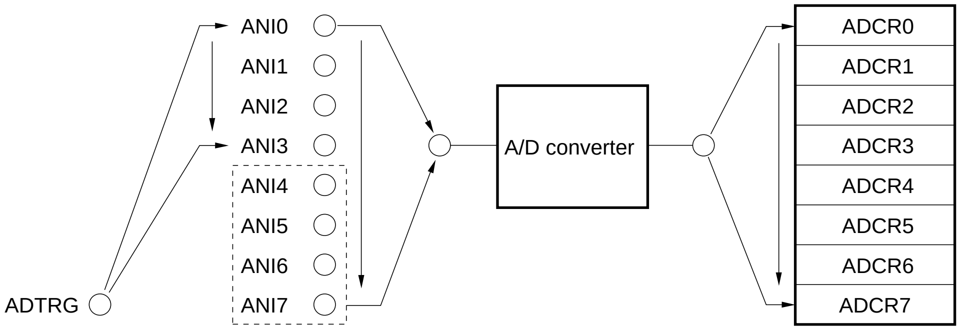
(a) Setting when scanning ANI0 to ANI3



(1) CE bit of ADM0 is set to 1 (enable)
(2) External trigger generation
(3) ANI0 A/D conversion
(4) Conversion result is stored in ADCR0
(5) External trigger generation
(6) ANI1 A/D conversion
(7) Conversion result is stored in ADCR1
(8) External trigger generation
(9) ANI2 A/D conversion
(10) Conversion result is stored in ADCR2
(11) External trigger generation
(12) ANI3 A/D conversion
(13) Conversion result is stored in ADCR3
(14) INTAD interrupt generation

Caution The analog input enclosed in the broken lines cannot be used with ADTRG as the trigger. When a setting is made to scan ANI0 to ANI7, ANI4 to ANI7 are converted in A/D trigger mode (see (b)).

(b) Setting when scanning ANI0 to ANI7



(1) to (13) Same as (a)
(14) ANI4 A/D conversion
(15) Conversion result is stored in ADCR4
(16) ANI5 A/D conversion
(17) Conversion result is stored in ADCR5
(18) ANI6 A/D conversion
(19) Conversion result is stored in ADCR6
(20) ANI7 A/D conversion
(21) Conversion result is stored in ADCR7
(22) INTAD interrupt generation

11.8 Operating Precautions

11.8.1 Stopping conversion operation

When 0 is written to the CE bit of the ADM0 register during a conversion operation, the conversion operation stops and the conversion results are not stored in the ADCRn register (n = 0 to 7).

11.8.2 External/timer trigger interval

Set the interval (input time interval) of the trigger in the external or timer trigger mode longer than the conversion time specified by the FR2 to FR0 bits of the ADM1 register.

(1) When interval = 0

When several triggers are input simultaneously, the analog input with the smaller ANIn pin number is converted. The other trigger signals input simultaneously are ignored, and the number of trigger inputs is not counted. Therefore, the generation of interrupts and storage of results in the ADCRn register will become abnormal (n = 0 to 7).

(2) When 0 < interval ≤ conversion operation time

When the timer trigger is input during a conversion operation, the conversion operation stops and the conversion starts according to the last timer trigger input.

When a conversion operation stops, the conversion results are not stored in the ADCRn register. However, the number of trigger inputs is counted, and when the interrupt is generated, the value at which conversion ended is stored in the ADCRn register.

11.8.3 Operation of standby mode

(1) HALT mode

The A/D conversion operation continues. When released by the NMI input, the ADM0 and ADM1 registers and ADCRn register hold the value (n = 0 to 7).

(2) IDLE mode, STOP mode

As clock supply to the A/D converter is stopped, no conversion operations are performed. When these modes are released using the NMI input, the ADM0 and ADM1 registers and the ADCRn register hold the value. However, when the IDLE and software STOP modes are set during a conversion operation, the conversion operation stops. At this time, if released using the NMI input, the conversion operation resumes, but the conversion result written to the ADCRn register will become undefined.

In the IDLE and software STOP modes, operation of the comparator is also stopped to reduce the power consumption, and to further reduce current consumption, set the voltage of the AV_{REF} to V_{SS}.

11.8.4 Compare match interrupt when in timer trigger mode

The compare register’s match interrupt becomes an A/D conversion start trigger and starts the conversion operation. When this happens, the compare register’s match interrupt functions even if it is a compare register match interrupt directed to the CPU. In order to prevent match interrupts from the compare register being directed to the CPU, disable interrupts by the interrupt mask bits (P11MK0 to P11MK3) of the interrupt control register (P11IC0 to P11IC3).

11.8.5 Timer 1 functions when in external trigger mode

The external trigger input becomes an A/D conversion start trigger. At this time, the external trigger input also functions as a timer 15 (TM15) capture trigger external interrupt. In order to prevent it from generating capture trigger external interrupts, set TM15 as a compare register and disable interrupts by the interrupt mask bit of the interrupt control register.

The operation if TM15 is not set as a compare register and interrupts are not disabled by the interrupt control register is as follows.

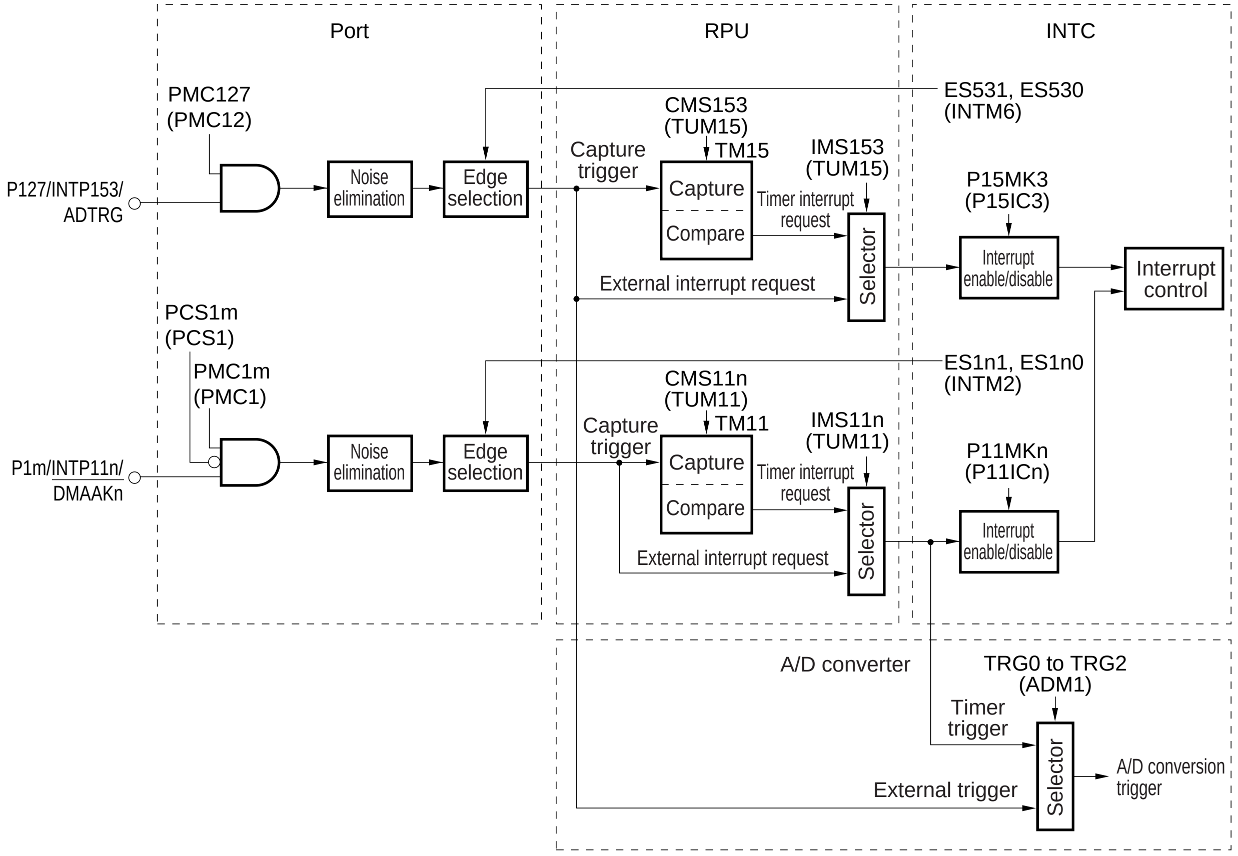
(a) If the TUM15 register's interrupt mask bit (IMS153) is 0

It also functions to generate compare register match interrupts to the CPU.

(b) If the TUM15 register's interrupt mask bit (IMS153) is 1

The A/D converter's external trigger input also functions as an external interrupt to the CPU.

Figure 11-18. Relationship of A/D Converter and Port, INTC and RPU



Remarks 1. m = 4 to 7, n = 0 to 3

2. Items in parentheses ( ) show the register names.

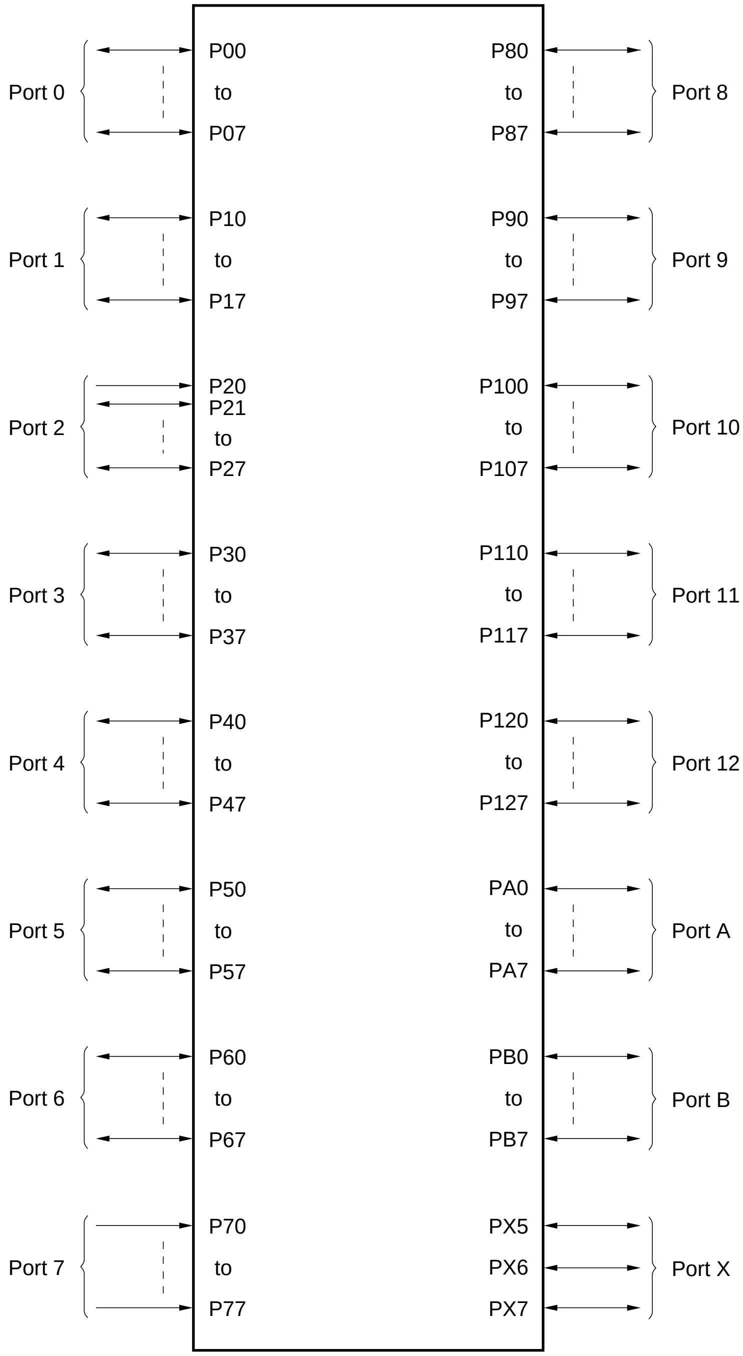
CHAPTER 12 PORT FUNCTIONS

12.1 Features

- Number of ports Input-only ports 9
I/O ports 114
- Function alternately as the input/output pins of other peripheral functions.
- It is possible to specify input and output in bit units.

12.2 Port Configuration

This product incorporates a total of 123 input/output ports (including 9 input-only ports) named ports 0 through 12, and A, B and X. The port configuration is shown below.



(1) Function of each port

The port functions of this product are shown below.

8/1-bit operations are possible on all ports, allowing various kinds of control to be performed. In addition to their port functions, these pins also function as internal peripheral I/O input/output pins in the control mode.

Port Name	Pin Name	Port Function	Function in Control Mode	Block Type[Note]
Port 0	P00 to P07	8-bit I/O	Input/output of real-time pulse unit (RPU) External interrupt input DMA control (DMAC) input	A, B, M
Port 1	P10 to P17	8-bit I/O	Input/output of real-time pulse unit (RPU) External interrupt input DMA control (DMAC) output	A, B, K
★ Port 2	P20 to P27	1-bit input, 7-bit I/O	NMI input Serial interface (UART0/CSI0, UART1/CSI1) input/output	C, D, I, J, Q
Port 3	P30 to P37	8-bit I/O	Input/output of real-time pulse unit (RPU) External interrupt input Serial interface (CSI2) input/output	A, B, K, M, N
Port 4	P40 to P47	8-bit I/O	External data bus (D0 to D7)	E
Port 5	P50 to P57	8-bit I/O	External data bus (D8 to D15)	E
Port 6	P60 to P67	8-bit I/O	External address bus (A16 to A23)	F
Port 7	P70 to P77	8-bit input	A/D converter (ADC) analog input	G
Port 8	P80 to P87	8-bit I/O	External bus interface control signal output	O, P
Port 9	P90 to P97	8-bit I/O	External bus interface control signal input/output	H, O
Port 10	P100 to P107	8-bit I/O	Input/output of real-time pulse unit (RPU) External interrupt input DMA control (DMAC) output	A, B, K
Port 11	P110 to P117	8-bit I/O	Input/output of real-time pulse unit (RPU) External interrupt input Serial interface (CSI3) input/output	A, B, K, M, N
Port 12	P120 to P127	8-bit I/O	Input/output of real-time pulse unit (RPU) External interrupt input A/D converter (ADC) external trigger input	A, B
Port A	PA0 to PA7	8-bit I/O	External address bus (A0 to A7)	F
Port B	PB0 to PB7	8-bit I/O	External address bus (A8 to A15)	F
Port X	PX5 to PX7	3-bit I/O	Refresh request signal output Wait insertion signal input Internal system clock output	A, L

Note Refer to **12.2 (3) Block diagram of port**.

Caution When switching to the control mode, be sure to set ports that operate as output pins, or as input/output pins in the control mode, by the following procedure.

<1> Set the inactive level for the signal output in the control mode in the relevant bits of port n (Pn) (n = 0 to 6, 8 to 12, A, B, X).

<2> Switch to the control mode from the port n mode control register (PMCn).

If <1> above is not performed, when switching from the port mode to the control mode, the contents of port n (Pn) will be output instantaneously.

(2) Function when each port's pins are reset and register which sets the port/control mode

(1/3)

Port Name	Pin Name	Pin Function After Reset				Register Which Sets the Mode
		Single-chip Mode 0	Single-chip Mode 1	ROM-less Mode 0	ROM-less Mode 1	
Port 0	P00/TO100	P00 (input mode)				PMC0
	P01/TO101	P01 (input mode)				
	P02/TCLR10	P02 (input mode)				
	P03/TI10	P03 (input mode)				
	P04/INTP100/$\overline{\mathrm{DMARQ0}}$	P04 (input mode)				PMC0, PCS0[Note]
	P05/INTP101/$\overline{\mathrm{DMARQ1}}$	P05 (input mode)				
	P06/INTP102/$\overline{\mathrm{DMARQ2}}$	P06 (input mode)				
	P07/INTP103/$\overline{\mathrm{DMARQ3}}$	P07 (input mode)				
Port 1	P10/TO110	P10 (input mode)				PMC1
	P11/TO111	P11 (input mode)				
	P12/TCLR11	P12 (input mode)				
	P13/TI11	P13 (input mode)				
	P14/INTP110/$\overline{\mathrm{DMAAK0}}$	P14 (input mode)				PMC1, PCS1[Note]
	P15/INTP111/$\overline{\mathrm{DMAAK1}}$	P15 (input mode)				
	P16/INTP112/$\overline{\mathrm{DMAAK2}}$	P16 (input mode)				
	P17/INTP113/$\overline{\mathrm{DMAAK3}}$	P17 (input mode)				
Port 2	P20/NMI	NMI				—
	P21	P21 (input mode)				
	P22/TXD0/SO0	P22 (input mode)				PMC2, ASIM00
	P23/RXD0/SI0	P23 (input mode)				
	P24/$\overline{\mathrm{SCK0}}$	P24 (input mode)				PMC2[Note]
	P25/TXD1/SO1	P25 (input mode)				PMC2, ASIM10
	P26/RXD1/SI1	P26 (input mode)				
	P27/$\overline{\mathrm{SCK1}}$	P27 (input mode)				PMC2[Note]
Port 3	P30/TO130	P30 (input mode)				PMC3
	P31/TO131	P31 (input mode)				
	P32/TCLR13	P32 (input mode)				
	P33/TI13	P33 (input mode)				
	P34/INTP130	P34 (input mode)				
	P35/INTP131/SO2	P35 (input mode)				PMC3, PCS3
	P36/INTP132/SI2	P36 (input mode)				
	P37/INTP133/$\overline{\mathrm{SCK2}}$	P37 (input mode)				

Note Selects the pin function when in the control mode.

(2/3)

Port Name	Pin Name	Pin Function After Reset				Register Which Sets the Mode
		Single-chip Mode 0	Single-chip Mode 1	ROM-less Mode 0	ROM-less Mode 1	
Port 4	P40/D0 to P47/D7	P40 to P47 (input mode)	D0 to D7			MM
Port 5	P50/D8 to P57/D15	P50 to P57 (input mode)	D8 to D15		P50 to P57 (input mode)	MM
Port 6	P60/A16 to P67/A23	P60 to P67 (input mode)	A16 to A23			MM
Port 7	P70/ANI0 to P77/ANI7	P70/ANI0 to P77/ANI7				—
Port 8	P80/$\overline{\text{CS0}}$/$\overline{\text{RAS0}}$	P80 (input mode)	$\overline{\text{CS0}}$/$\overline{\text{RAS0}}$			PMC8
	P81/$\overline{\text{CS1}}$/$\overline{\text{RAS1}}$	P81 (input mode)	$\overline{\text{CS1}}$/$\overline{\text{RAS1}}$			
	P82/$\overline{\text{CS2}}$/$\overline{\text{RAS2}}$	P82 (input mode)	$\overline{\text{CS2}}$/$\overline{\text{RAS2}}$			
	P83/$\overline{\text{CS3}}$/$\overline{\text{RAS3}}$	P83 (input mode)	$\overline{\text{CS3}}$/$\overline{\text{RAS3}}$			
	P84/$\overline{\text{CS4}}$/$\overline{\text{RAS4}}$/$\overline{\text{IOWR}}$	P84 (input mode)	$\overline{\text{CS4}}$/$\overline{\text{RAS4}}$			PMC8, PCS8[Note]
	P85/$\overline{\text{CS5}}$/$\overline{\text{RAS5}}$/$\overline{\text{IORD}}$	P85 (input mode)	$\overline{\text{CS5}}$/$\overline{\text{RAS5}}$			
	P86/$\overline{\text{CS6}}$/$\overline{\text{RAS6}}$	P86 (input mode)	$\overline{\text{CS6}}$/$\overline{\text{RAS6}}$			PMC8
	P87/$\overline{\text{CS7}}$/$\overline{\text{RAS7}}$	P87 (input mode)	$\overline{\text{CS7}}$/$\overline{\text{RAS7}}$			
Port 9	P90/$\overline{\text{LCAS}}$/$\overline{\text{LWR}}$	P90 (input mode)	$\overline{\text{LCAS}}$/$\overline{\text{LWR}}$			PMC9
	P91/$\overline{\text{UCAS}}$/$\overline{\text{UWR}}$	P91 (input mode)	$\overline{\text{UCAS}}$/$\overline{\text{UWR}}$			
	P92/$\overline{\text{RD}}$	P92 (input mode)	$\overline{\text{RD}}$			
	P93/$\overline{\text{WE}}$	P93 (input mode)	$\overline{\text{WE}}$			
	P94/$\overline{\text{BCYST}}$	P94 (input mode)	$\overline{\text{BCYST}}$			PMC9
	P95/$\overline{\text{OE}}$	P95 (input mode)	$\overline{\text{OE}}$			PMC9
	P96/$\overline{\text{HLDAK}}$	P96 (input mode)	$\overline{\text{HLDAK}}$			
	P97/$\overline{\text{HLDRQ}}$	P97 (input mode)	$\overline{\text{HLDRQ}}$			
Port 10	P100/TO120	P100 (input mode)				PMC10
	P101/TO121	P101 (input mode)				
	P102/TCLR12	P102 (input mode)				
	P103/TI12	P103 (input mode)				
	P104/INTP120/$\overline{\text{TC0}}$	P104 (input mode)				PMC10, PCS10[Note]
	P105/INTP121/$\overline{\text{TC1}}$	P105 (input mode)				
	P106/INTP122/$\overline{\text{TC2}}$	P106 (input mode)				
	P107/INTP123/$\overline{\text{TC3}}$	P107 (input mode)				

Note Selects the pin function when in the control mode.

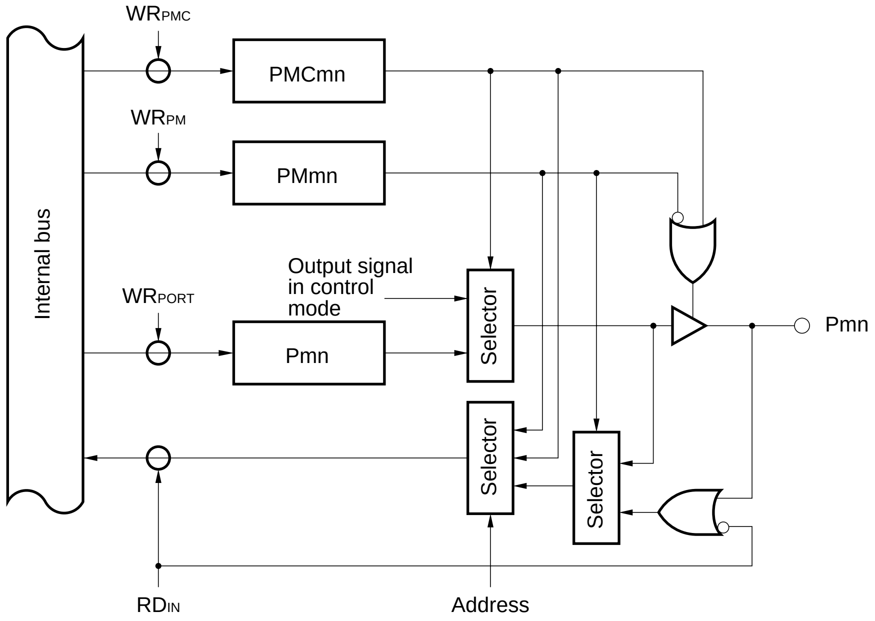
(3/3)

Port Name	Pin Name	Pin Function After Reset				Register Which Sets the Mode
		Single-chip Mode 0	Single-chip Mode 1	ROM-less Mode 0	ROM-less Mode 1	
Port 11	P110/TO140	P110 (input mode)				PMC11
	P111/TO141	P111 (input mode)				
	P112/TCLR14	P112 (input mode)				
	P113/TI14	P113 (input mode)				
	P114/INTP140	P114 (input mode)				
	P115/INTP141/SO3	P115 (input mode)				PMC11, PCS11[Note]
	P116/INTP142/SI3	P116 (input mode)				
	P117/INTP143/$\overline{\text{SCK3}}$	P117 (input mode)				
Port 12	P120/TO150	P120 (input mode)				PMC12
	P121/TO151	P121 (input mode)				
	P122/TCLR15	P122 (input mode)				
	P123/TI15	P123 (input mode)				
	P124/INTP150	P124 (input mode)				
	P125/INTP151	P125 (input mode)				
	P126/INTP152	P126 (input mode)				
	P127/INTP153/ADTRG	P127 (input mode)				PMC12, ADM1[Note]
Port A	PA0/A0 to PA7/A7	PA0 to PA7 (input mode)	A0 to A7			MM
Port B	PB0/A8 to PB7/A15	PB0 to PB7 (input mode)	A8 to A15			MM
Port X	PX5/$\overline{\text{REFRQ}}$	PX5 (input mode)	$\overline{\text{REFRQ}}$			PMCX
	PX6/$\overline{\text{WAIT}}$	PX6 (input mode)	$\overline{\text{WAIT}}$			
	PX7/CLKOUT	PX7 (input mode)	CLKOUT			

Note Selects the pin function when in the control mode.

(3) Block diagram of port

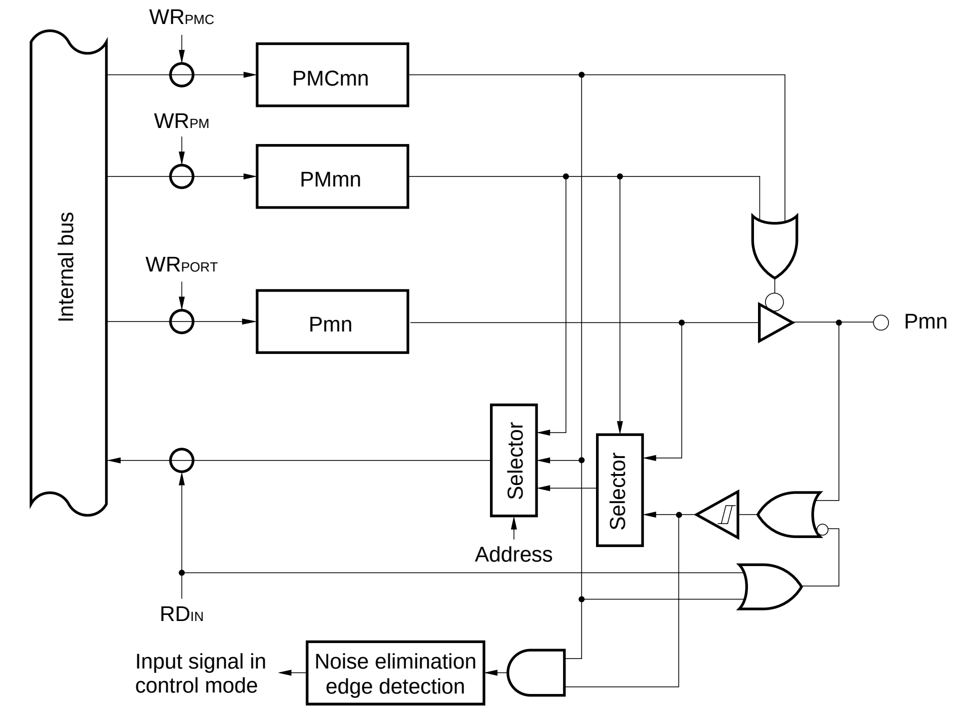
Figure 12-1. Type A Block Diagram



Remark m: Port number

n: Bit number

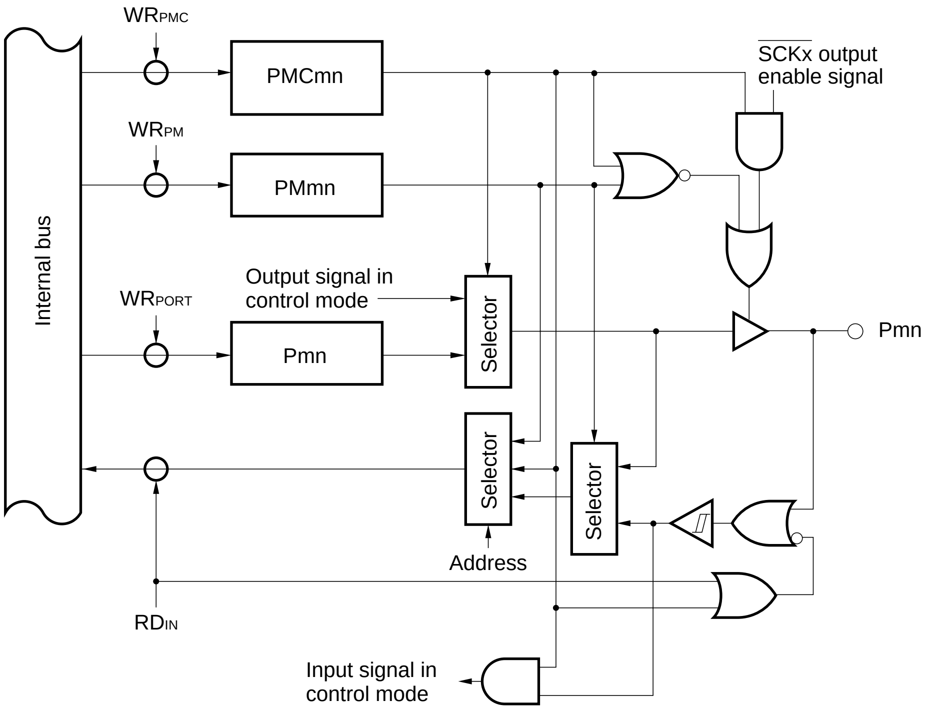
Figure 12-2. Type B Block Diagram



Remark m: Port number

n: Bit number

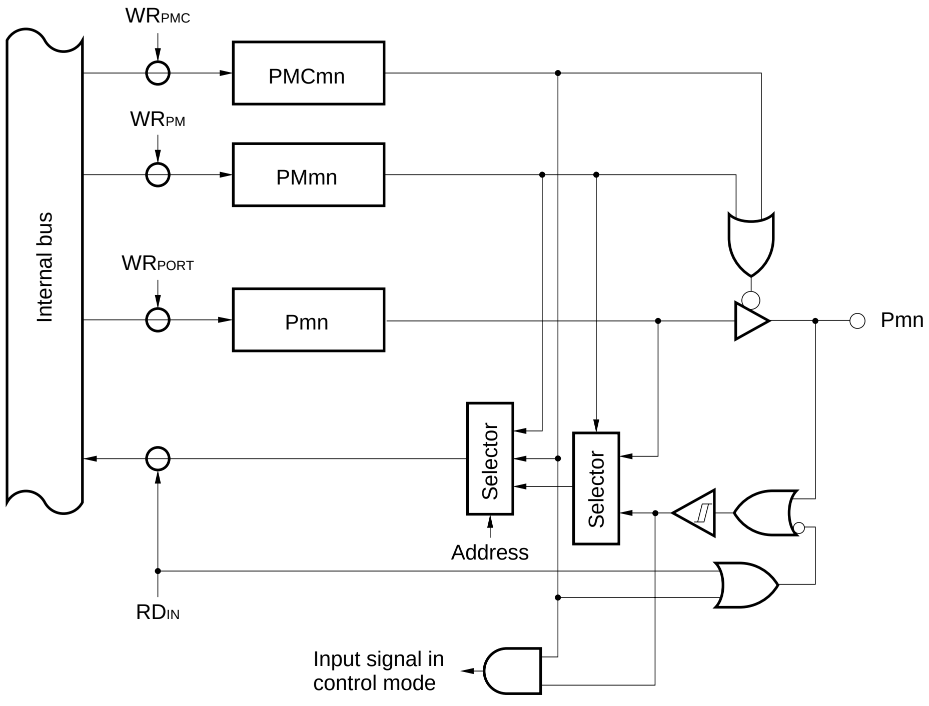
★ **Figure 12-3. Type C Block Diagram**


WRPMC
PMCmn
SCKx output enable signal
WRPM
PMmn
Internal bus
Output signal in control mode
WRPORT
Selector
Pmn
Pmn
Selector
Selector
Address
RDIN
Input signal in control mode

Remark mn: 24, 27

x: 0 (when mn = 24), 1 (when mn = 27)

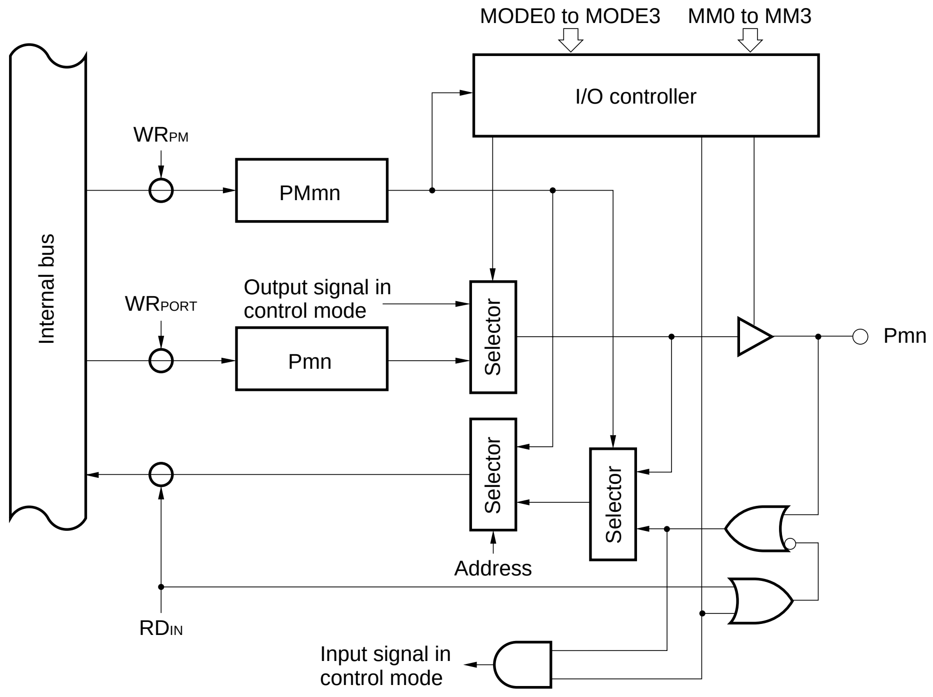
Figure 12-4. Type D Block Diagram



Remark m: Port number

n: Bit number

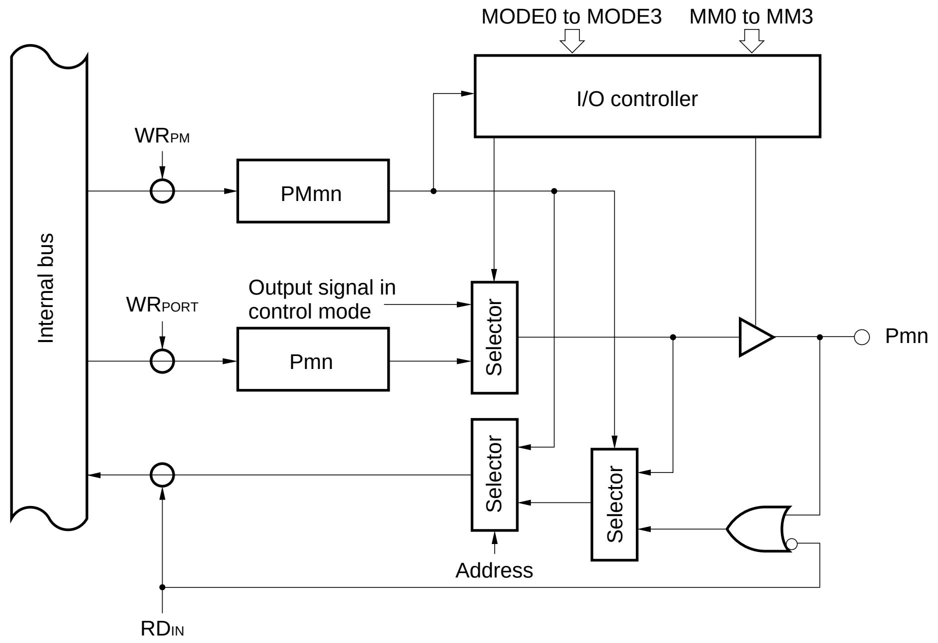
Figure 12-5. Type E Block Diagram



Remark m: Port number

n: Bit number

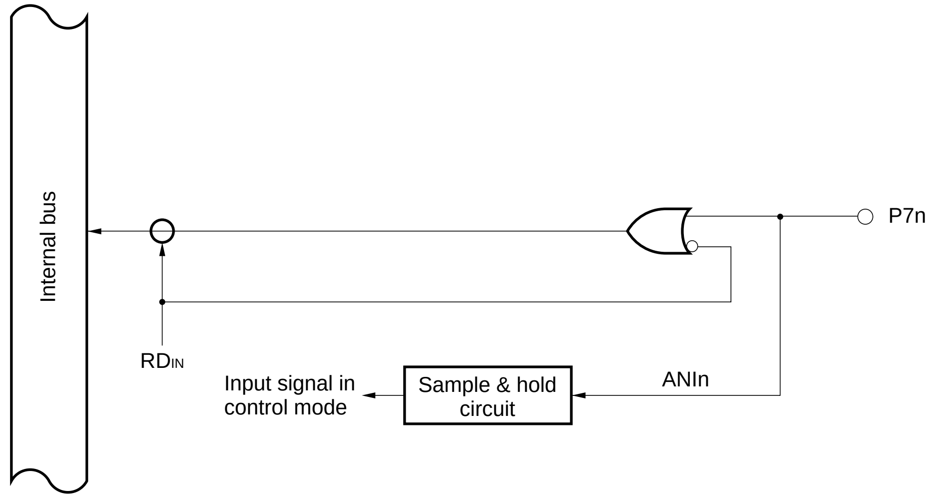
Figure 12-6. Type F Block Diagram



Remark m: Port number

n: Bit number

Figure 12-7. Type G Block Diagram



Remark n = 0 to 7

Figure 12-8. Type H Block Diagram

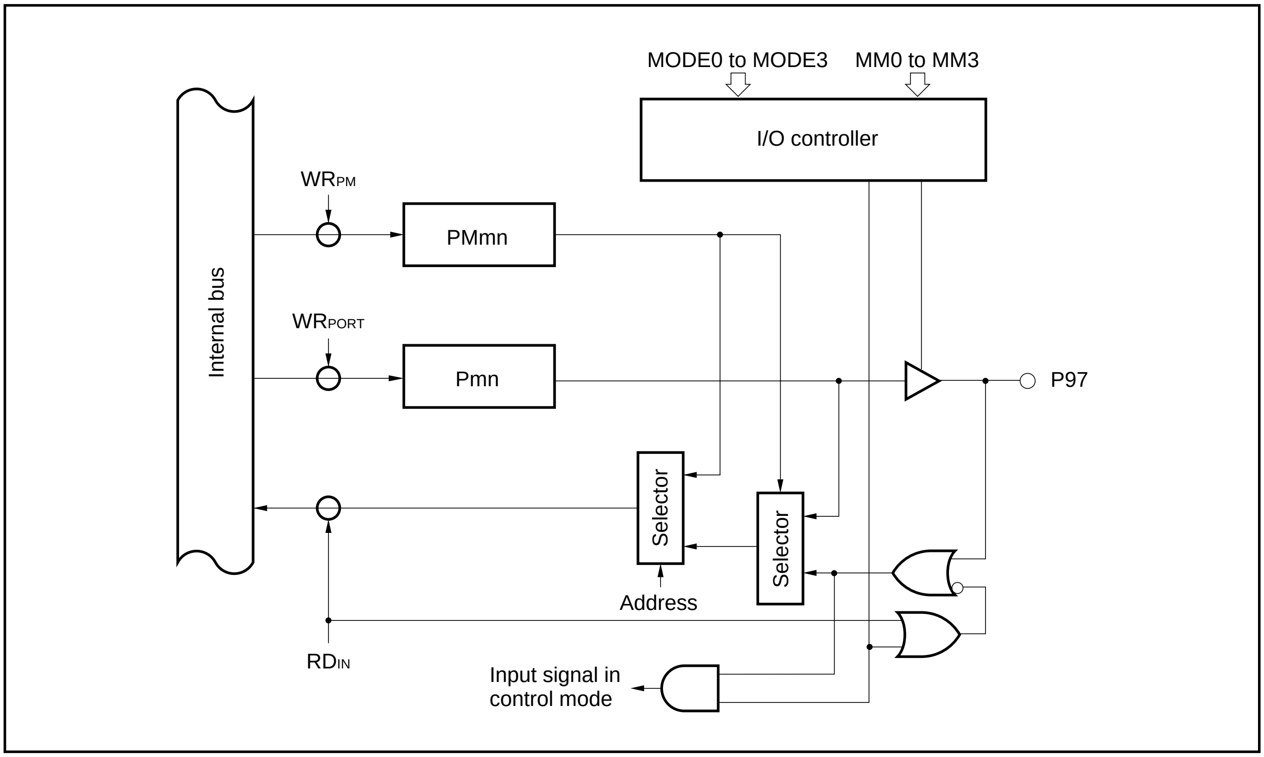


Figure 12-9. Type I Block Diagram

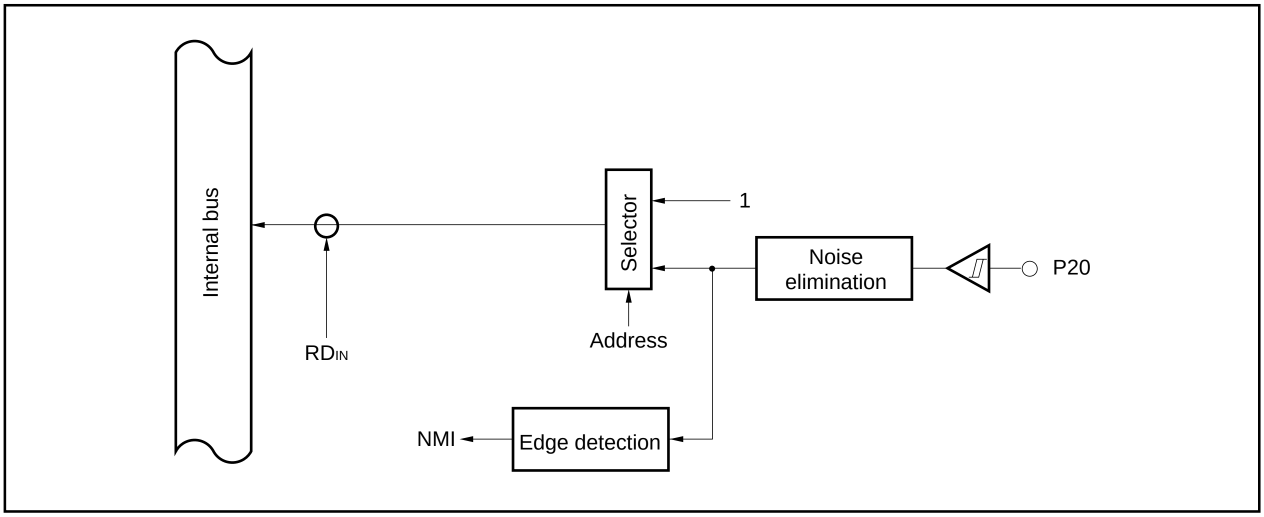
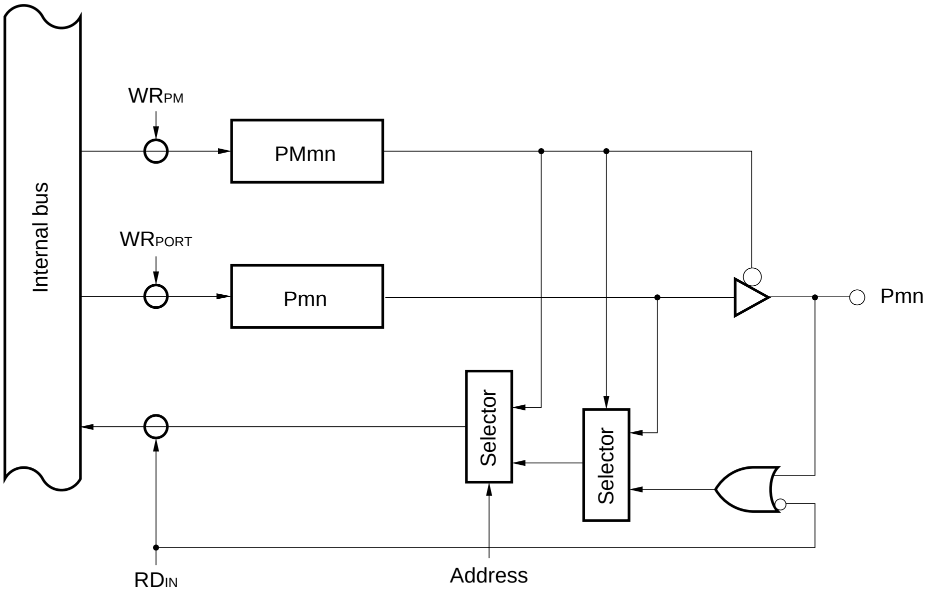


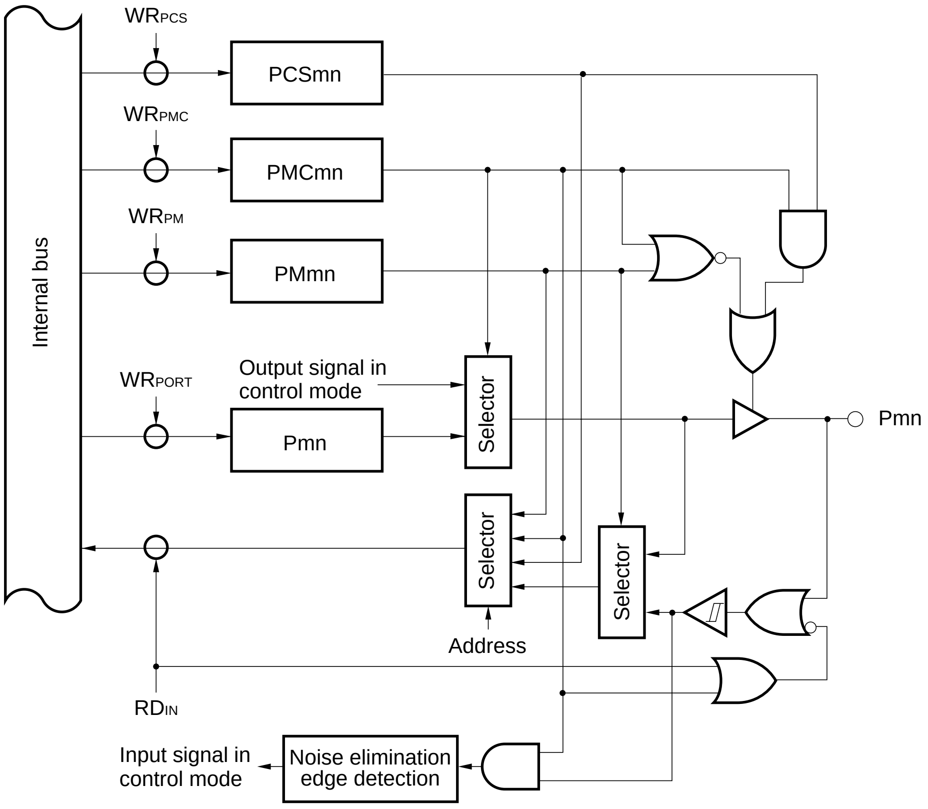
Figure 12-10. Type J Block Diagram



Remark m: Port number

n: Bit number

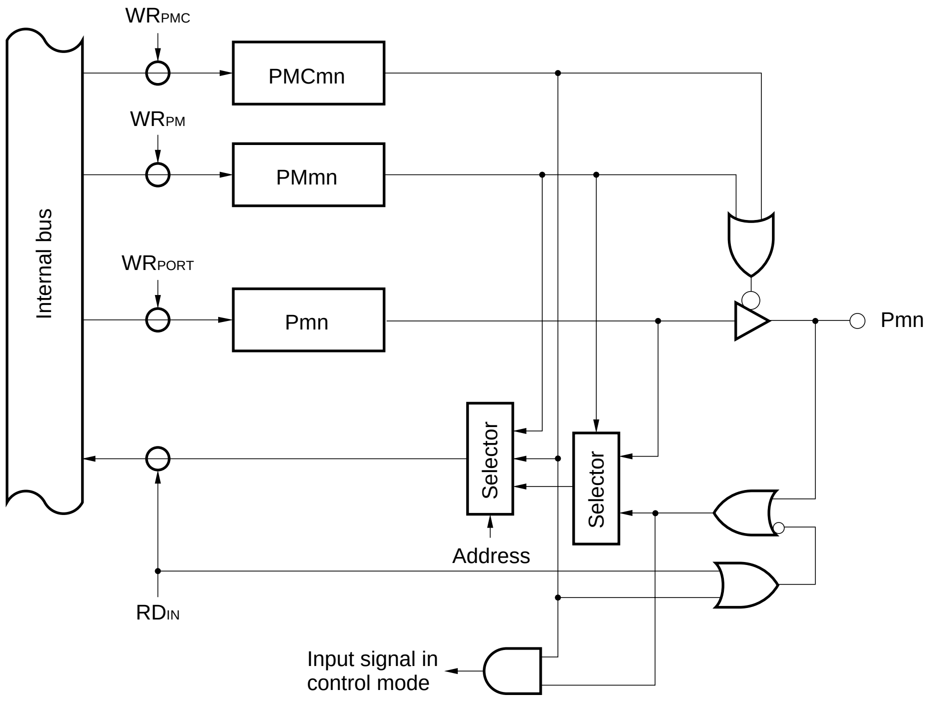
Figure 12-11. Type K Block Diagram


Internal bus
WRPCS
PCSmn
WRPMC
PMCmn
WRPM
PMmn
WRPORT
Output signal in control mode
Pmn
Selector
Pmn
Selector
Selector
Address
RDIN
Input signal in control mode
Noise elimination edge detection

Remark m: Port number

n: Bit number

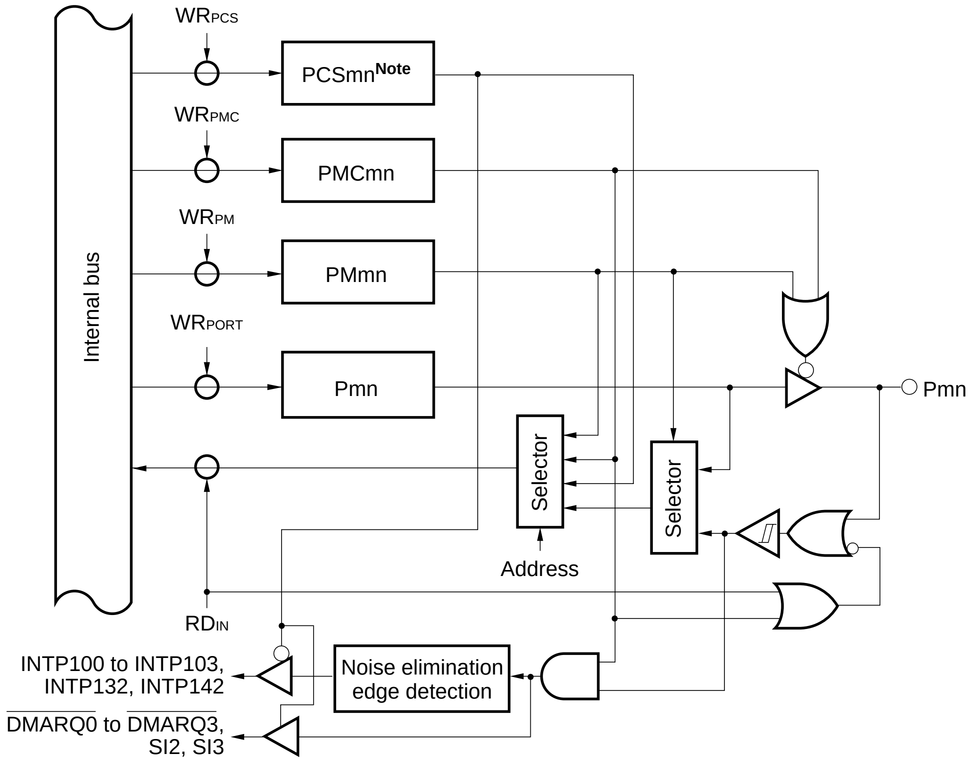
Figure 12-12. Type L Block Diagram



Remark m: Port number

n: Bit number

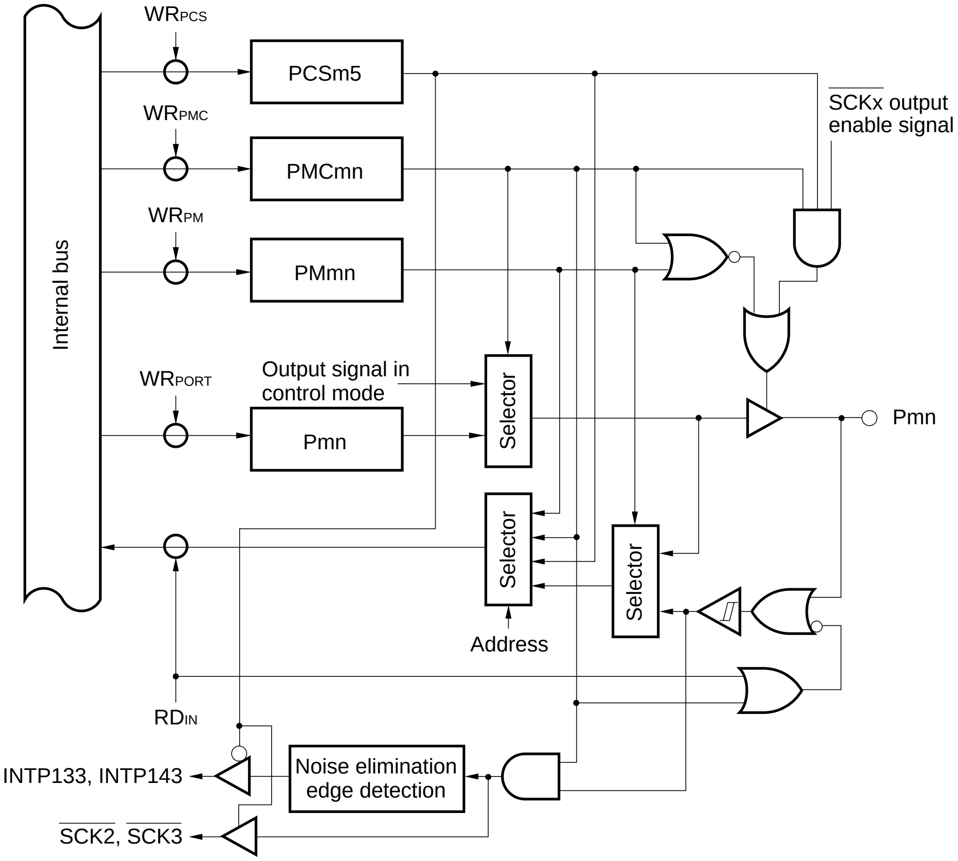
Figure 12-13. Type M Block Diagram



Note When mn = 36: PCS35
When mn = 116: PCS115

Remark mn: 04 to 07, 36, 116

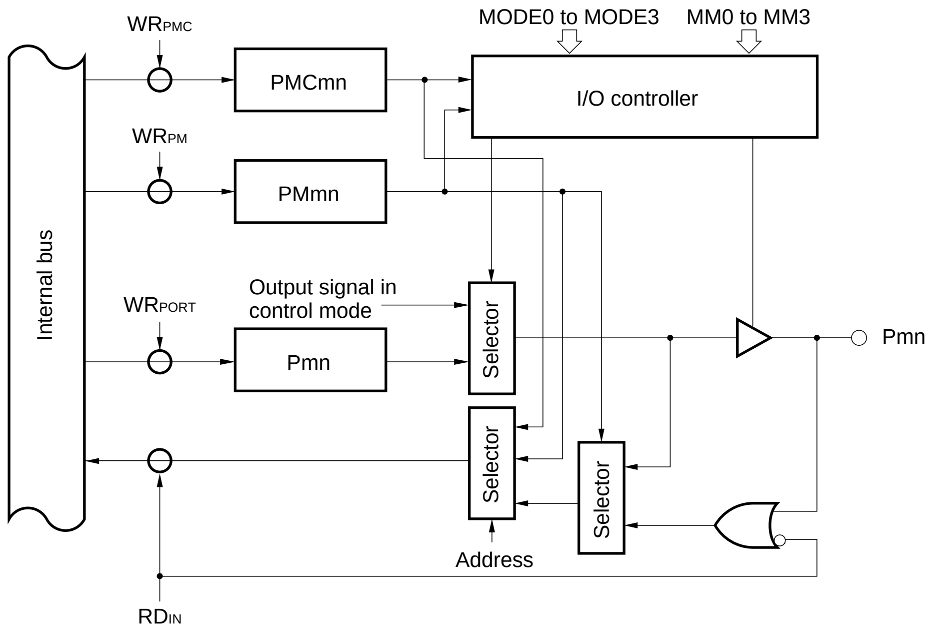
Figure 12-14. Type N Block Diagram



Remark mn: 37, 117

x: 2 (when mn = 37), 3 (when mn = 117)

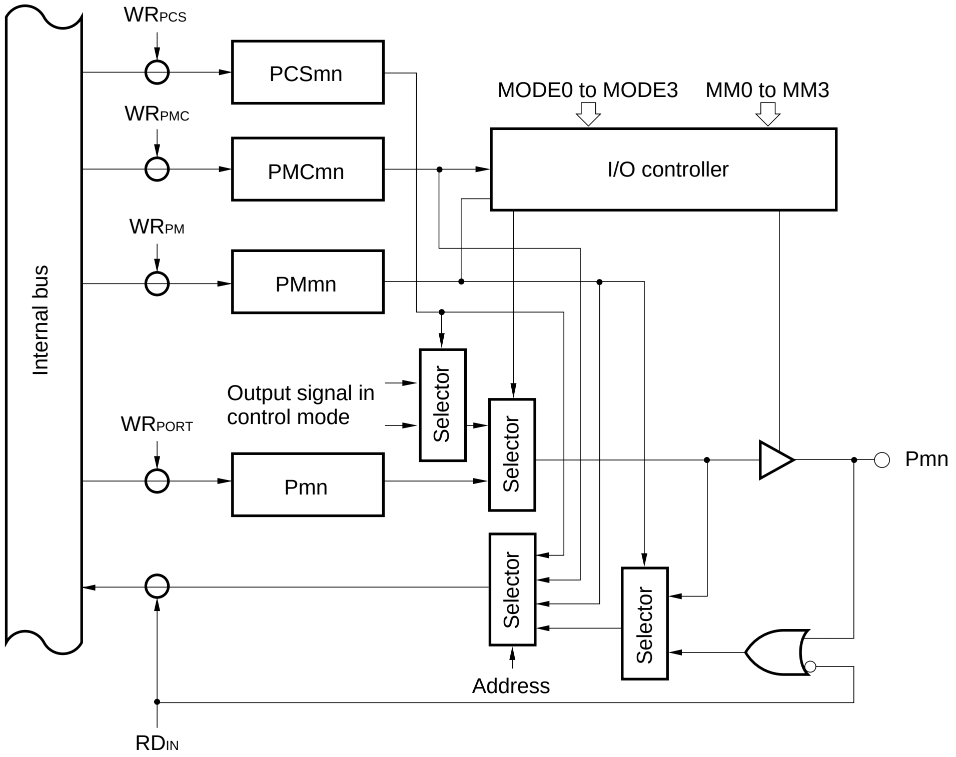
Figure 12-15. Type O Block Diagram



Remark m: Port number

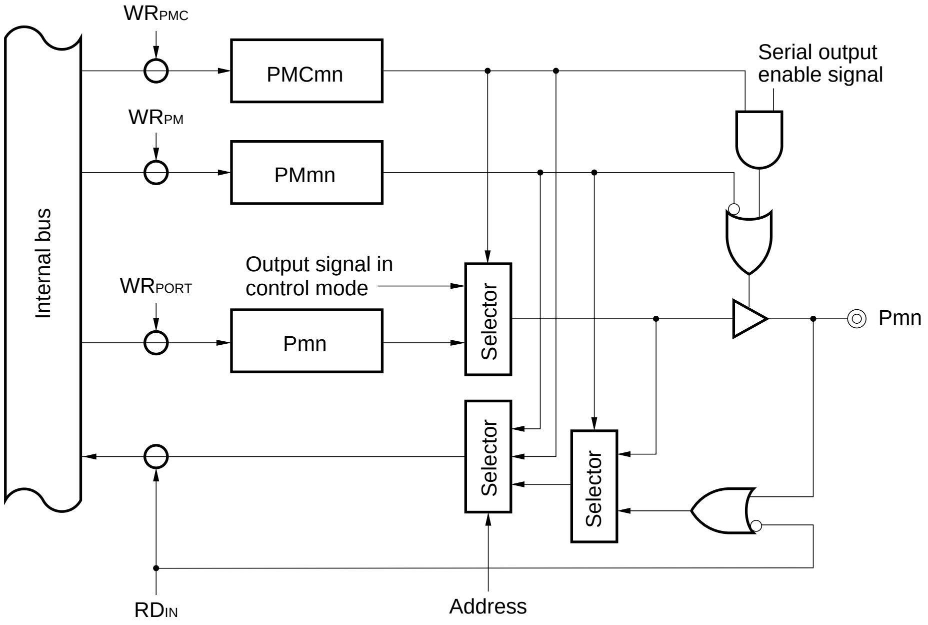
n: Bit number

Figure 12-16. Type P Block Diagram



Remark m: Port number
n: Bit number

★ **Figure 12-17. Type Q Block Diagram**



Remark m: Port number

n: Bit number

12.3 Port Pin Functions

12.3.1 Port 0

Port 0 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
P0	P07	P06	P05	P04	P03	P02	P01	P00	Address FFFFF000H	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	P0n (n = 7 to 0)	Port 0 Input/output port

In addition to their function as port pins, the port 0 pins can also operate as real-time pulse unit (RPU) inputs/outputs, external interrupt request inputs, and DMA request inputs in the control mode.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 0	P00	TO100	Real-time pulse unit (RPU) output	A
	P01	TO101		
	P02	TCLR10	Real-time pulse unit (RPU) input	B
	P03	TI10		
	P04 to P07	INTP100/$\overline{\text{DMARQ0}}$ to INTP103/$\overline{\text{DMARQ3}}$	External interrupt request input/DMA request input	M

(2) Input/output mode/control mode setting

Port 0 input/output mode setting is performed by means of the port 0 mode register (PM0), and control mode setting is performed by means of the port 0 mode control register (PMC0) and port/control select register 0 (PCS0).

(a) Port 0 mode register (PM0)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	Address FFFFF020H	After reset FFH

Bit Position	Bit Name	Function
7 to 0	PM0n (n = 7 to 0)	Port Mode Specifies the input/output mode of pin P0n. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Port 0 mode control register (PMC0)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PMC0	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00	Address FFFFF040H	After reset 00H

Bit Position	Bit Name	Function
7 to 4	PMC0n (n = 7 to 4)	Port Mode Control Specifies the operation mode of pin P0n. Sets in combination with the PCS0 register. 0: Input/output port mode 1: External interrupt request (INTP103 to INTP100) input mode/DMA request ($\overline{\text{DMARQ3}}$ to $\overline{\text{DMARQ0}}$) input mode
3	PMC03	Port Mode Control Sets operation mode of P03 pin. 0: Input/output port mode 1: TI10 input mode
2	PMC02	Port Mode Control Sets operation mode of P02 pin. 0: Input/output port mode 1: TCLR10 input mode
1	PMC01	Port Mode Control Sets operation mode of P01 pin. 0: Input/output port mode 1: TO101 output mode
0	PMC00	Port Mode Control Sets operation mode of P00 pin. 0: Input/output port mode 1: TO100 output mode

(c) Port/control select register 0 (PCS0)

This register can be read/written in 8- or 1-bit units. However, bits 3 to 0 are fixed at 0, so even if 1 is written, it is disregarded.

	7	6	5	4	3	2	1	0	Address	After reset
PCS0	PCS07	PCS06	PCS05	PCS04	0	0	0	0	FFFFF580H	00H

Bit Position	Bit Name	Function
7	PCS07	Port Control Select Specifies the operating mode when pin P07 is in the control mode. 0: INTP103 input mode 1: $\overline{\text{DMARQ3}}$ input mode
6	PCS06	Port Control Select Specifies the operating mode when pin P06 is in the control mode. 0: INTP102 input mode 1: $\overline{\text{DMARQ2}}$ Input mode
5	PCS05	Port Control Select Specifies the operating mode when pin P05 is in the control mode. 0: INTP101 input mode 1: $\overline{\text{DMARQ1}}$ input mode
4	PCS04	Port Control Select Specifies the operating mode when pin P04 is in the control mode. 0: INTP100 input mode 1: $\overline{\text{DMARQ0}}$ input mode

Caution When the port mode is specified by the PMC0 register, the settings of this register are ignored.

12.3.2 Port 1

Port 1 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
P1	P17	P16	P15	P14	P13	P12	P11	P10	Address FFFFF002H	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	P1n (n = 7 to 0)	Port 1 Input/output port

In addition to their function as port pins, the port 1 pins can also operate as real-time pulse unit (RPU) inputs/outputs, external interrupt request inputs, and DMA acknowledge outputs in the control mode.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 1	P10	TO110	Real-time pulse unit (RPU) output	A
	P11	TO111		
	P12	TCLR11	Real-time pulse unit (RPU) input	B
	P13	TI11		
	P14 to P17	INTP110/$\overline{\text{DMAAK0}}$ to INTP113/$\overline{\text{DMAAK3}}$	External interrupt input/DMA acknowledge output	K

(2) Input/output mode/control mode setting

Port 1 input/output mode setting is performed by means of the port 1 mode register (PM1), and control mode setting is performed by means of the port 1 mode control register (PMC1) and port/control select register 1 (PCS1).

(a) Port 1 mode register (PM1)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	Address FFFFF022H	After reset FFH

Bit Position	Bit Name	Function
7 to 0	PM1n (n = 7 to 0)	Port Mode Sets P1n in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Port 1 mode control register (PMC1)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PMC1	PMC17	PMC16	PMC15	PMC14	PMC13	PMC12	PMC11	PMC10	Address FFFFF042H	After reset 00H

Bit Position	Bit Name	Function
7 to 4	PMC1n (n = 7 to 4)	Port Mode Control Sets operation mode of P1n pin. Set in combination with PCS1. 0: Input/output port mode 1: External interrupt request (INTP113 to INTP110) input mode/ DMA acknowledge ($\overline{\text{DMAAK3}}$ to $\overline{\text{DMAAK0}}$) output mode
3	PMC13	Port Mode Control Sets operation mode of P13 pin. 0: Input/output port mode 1: TI11 input mode
2	PMC12	Port Mode Control Sets operation mode of P12 pin. 0: Input/output port mode 1: TCLR11 input mode
1	PMC11	Port Mode Control Sets operation mode of P11 pin. 0: Input/output port mode 1: TO111 output mode
0	PMC10	Port Mode Control Sets operation mode of P10 pin. 0: Input/output port mode 1: TO110 output mode

(c) Port/control select register 1 (PCS1)

This register can be read/written in 8- or 1-bit units. However, bits 3 to 0 are fixed at 0, so even if 1 is written, it is disregarded.

	7	6	5	4	3	2	1	0
PCS1	PCS17	PCS16	PCS15	PCS14	0	0	0	0

Address
FFFFF582H

After reset
00H

Bit Position	Bit Name	Function
7	PCS17	Port Control Select Specifies the operating mode when pin P17 is in the control mode. 0: INTP113 input mode 1: $\overline{\text{DMAAK3}}$ output mode
6	PCS16	Port Control Select Specifies the operating mode when pin P16 is in the control mode. 0: INTP112 input mode 1: $\overline{\text{DMAAK2}}$ output mode
5	PCS15	Port Control Select Specifies the operating mode when pin P15 is in the control mode. 0: INTP111 input mode 1: $\overline{\text{DMAAK1}}$ output mode
4	PCS14	Port Control Select Specifies the operating mode when pin P14 is in the control mode. 0: INTP110 input mode 1: $\overline{\text{DMAAK0}}$ output mode

Caution When the port mode is specified by the PMC1 register, the settings of this register are ignored.

12.3.3 Port 2

Port 2 is an 8-bit input/output port that can be set to input or output in 1-bit units. However, P20 always operates as an NMI input if the edge is input.

	7	6	5	4	3	2	1	0		
P2	P27	P26	P25	P24	P23	P22	P21	P20	Address FFFFF004H	After reset Undefined

Bit Position	Bit Name	Function
7 to 1	P2n (n = 7 to 1)	Port 2 Input/output port
0	P20	Fix to NMI input mode.

In addition to their function as port pins, the port 2 pins can also operate as serial interface (UART0/CSI0, UART1/CSI1) inputs/outputs in the control mode. Note that pin P21 does not have an alternate function and operates only in the port mode.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 2	P20	NMI	Non-maskable interrupt request input	I
	P21	—	Fixed to port mode	J
★	P22	TXD0/SO0	Input/output for serial interface (UART0/CSI0, UART1/CSI1)	Q
	P23	RXD0/SI0		D
	P24	$\overline{\text{SCK0}}$		C
★	P25	TXD1/SO1		Q
	P26	RXD1/SI1		D
	P27	$\overline{\text{SCK1}}$		C

(2) Input/output mode/control mode setting

Port 2 input/output mode setting is performed by means of the port 2 mode register (PM2), and control mode setting is performed by means of the port 2 mode control register (PMC2). Pin P20 is fixed to NMI input mode.

(a) Port 2 mode register (PM2)

This register can be read/written in 8- or 1-bit units. However, bit 0 is fixed at 1 by hardware, so writing 0 to this bit is ignored.

	7	6	5	4	3	2	1	0	Address	After reset
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	1	FFFFF024H	FFH

Bit Position	Bit Name	Function
7 to 1	PM2n (n = 7 to 1)	Port Mode Sets P2n in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

Caution When the serial interface is used, use the following bits in the state when they are set to 1 (initial value).

★ **When UART0 is used:**	**PM22**
★ **When UART1 is used:**	**PM25**
When CSI0 is used:	**PM24 to PM22**
When CSI1 is used:	**PM27 to PM25**

(b) Port 2 mode control register (PMC2)

This register can be read/written in 8- or 1-bit units. However, bit 0 is fixed to 1 by hardware, so writing 0 to this bit is ignored. Bit 1 is fixed to 0, so writing 1 to this bit is ignored.

	7	6	5	4	3	2	1	0		
PMC2	PMC27	PMC26	PMC25	PMC24	PMC23	PMC22	0	1	Address FFFFF044H	After reset 01H

Bit Position	Bit Name	Function
7	PMC27	Port Mode Control Sets operation mode of P27 pin. 0: Input/output port mode 1: $\overline{\text{SCK1}}$ input/output mode
6	PMC26	Port Mode Control Sets operation mode of P26 pin. 0: Input/output port mode 1: RXD1/SI1 input mode
5	PMC25	Port Mode Control Sets operation mode of P25 pin. 0: Input/output port mode 1: TXD1/SO1 output mode
4	PMC24	Port Mode Control Sets operation mode of P24 pin. 0: Input/output port mode 1: $\overline{\text{SCK0}}$ input/output mode
3	PMC23	Port Mode Control Sets operation mode of P23 pin. 0: Input/output port mode 1: RXD0/SI0 input mode
2	PMC22	Port Mode Control Sets operation mode of P22 pin. 0: Input/output port mode 1: TXD0/SO0 output mode

Remark UART0 and CSI0, and UART1 and CSI1 share the same pins respectively. Either one of these is selected according to the ASIM00 and ASIM10 registers (refer to **10.2.3 Control registers**).

12.3.4 Port 3

Port 3 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
P3	P37	P36	P35	P34	P33	P32	P31	P30	Address FFFFF006H	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	P3n (n = 7 to 0)	Port 3 Input/output port

In addition to their function as port pins, the port 3 pins can also operate as the input/output signals of the real-time pulse unit (RPU), the input signals of external interrupt, and the input/output lines of the serial interface (CSI2) when in the control mode.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 3	P30	TO130	Real-time pulse unit (RPU) output	A
	P31	TO131		
	P32	TCLR13	Real-time pulse unit (RPU) input	B
	P33	TI13		
	P34	INTP130	External interrupt input	
	P35	INTP131/SO2	External interrupt input/serial interface (CSI2) input/output	K
	P36	INTP132/SI2		M
	P37	INTP133/$\overline{\text{SCK2}}$		N

(2) Input/output mode/control mode setting

Port 3 input/output mode setting is performed by means of the port 3 mode register (PM3), and control mode setting is performed by means of the port 3 mode control register (PMC3) and port/control select register 3 (PCS3).

★ **(a) Port 3 mode register (PM3)**

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	Address FFFFF026H	After reset FFH

Bit Position	Bit Name	Function
7 to 0	PM3n (n = 7 to 0)	Port Mode Sets P3n in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Port 3 mode control register (PMC3)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PMC3	PMC37	PMC36	PMC35	PMC34	PMC33	PMC32	PMC31	PMC30	FFFFF046H	00H

Bit Position	Bit Name	Function
7 to 5	PMC3n (n = 7 to 5)	Port Mode Control Sets operation mode of P3n pin. Set in combination with PCS3. 0: Input/output port mode 1: External interrupt request (INTP133 to INTP131) input mode/CSI2 ($\overline{\text{SCK2}}$, SI2, SO2) input/output mode
4	PMC34	Port Mode Control Sets operation mode of P34 pin. 0: Input/output port mode 1: INTP130 input mode
3	PMC33	Port Mode Control Sets operation mode of P33 pin. 0: Input/output port mode 1: TI13 input mode
2	PMC32	Port Mode Control Sets operation mode of P32 pin. 0: Input/output port mode 1: TCLR13 input mode
1	PMC31	Port Mode Control Sets operation mode of P31 pin. 0: Input/output port mode 1: TO131 output mode
0	PMC30	Port Mode Control Sets operation mode of P30 pin. 0: Input/output port mode 1: TO130 output mode

(c) Port/control select register 3 (PCS3)

This register can be read/written in 8- or 1-bit units. However, except for bit 5, all the bits are fixed at 0, so even if 1 is written, it is disregarded.

	7	6	5	4	3	2	1	0		
PCS3	0	0	PCS35	0	0	0	0	0	Address FFFFF586H	After reset 00H

Bit Position	Bit Name	Function
5	PCS35	Port Control Select Specifies the operating mode when pins P37 to P35 are in the control mode. 0: INTP133 input mode (P37) INTP132 input mode (P36) INTP131 input mode (P35) 1: $\overline{\text{SCK2}}$ input/output mode (P37) SI2 input mode (P36) SO2 output mode (P35)

Caution When the port mode is specified by the PMC3 register, the settings of this register are ignored.

12.3.5 Port 4

Port 4 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
P4	P47	P46	P45	P44	P43	P42	P41	P40	Address FFFFF008H	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	P4n (n = 7 to 0)	Port 4 Input/output port

In addition to their function as port pins, the port 4 pins can also operate in the control mode (external expansion mode) as a data bus used when memory is expanded externally.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 4	P40 to P47	D0 to D7	Data bus in memory expansion	E

(2) Input/output mode/control mode setting

Port 4 input/output mode setting is performed by means of the port 4 mode register (PM4), and control mode (external expansion mode) setting is performed by means of the mode specification pins (MODE0 to MODE3) and the memory expansion mode register (MM: refer to **3.4.6 (1)**).

(a) Port 4 mode register (PM4)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	Address FFFFF028H	After reset FFH

Bit Position	Bit Name	Function
7 to 0	PM4n (n = 7 to 0)	Port Mode Sets P4n in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Operation mode of port 4

Bit of MM Register				Operation Mode							
MM3	MM2	MM1	MM0	P40	P41	P42	P43	P44	P45	P46	P47
don't care	0	0	0	Port (P40 to P47)							
	0	0	1	Data bus (D0 to D7)							
	0	1	0								
	0	1	1								
	1	0	0								
	1	0	1								
	1	1	0								
	1	1	1								

For the details of mode selection by the MODE0 to MODE3 pins, refer to **3.3.2 Operating mode specification**.

In ROM-less modes 0 or 1, or single-chip mode 1, the MM0 to MM3 bits are initialized to 111× at system reset, enabling the external expansion mode. External expansion can be disabled by programming the MM0 to MM3 bits and setting the port mode. If MM0 to MM3 are set to 000×, the subsequent external instruction cannot be fetched.

Remark ×: don't care

12.3.6 Port 5

Port 5 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
P5	P57	P56	P55	P54	P53	P52	P51	P50	FFFFF00AH	Undefined

Bit Position	Bit Name	Function
7 to 0	P5n (n = 7 to 0)	Port 5 Input/output port

In addition to their function as port pins, the port 5 pins can also operate in the control mode (external expansion mode) as a data bus used when memory is expanded externally.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 5	P50 to P57	D8 to D15	Data bus in memory expansion	E

(2) Input/output mode/control mode setting

Port 5 input/output mode setting is performed by means of the port 5 mode register (PM5), and control mode (external expansion mode) setting is performed by means of the mode specification pins (MODE0 to MODE3) and the memory expansion mode register (MM: refer to **3.4.6 (1)**).

(a) Port 5 mode register (PM5)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	Address FFFFF02AH	After reset FFH

Bit Position	Bit Name	Function
7 to 0	PM5n (n = 7 to 0)	Port Mode Sets P5n in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Operation mode of port 5

Bit of MM Register				Operation Mode							
MM3	MM2	MM1	MM0	P50	P51	P52	P53	P54	P55	P56	P57
0	0	0	0	Port (P50 to P57)							
0	0	0	1	Data bus (D8 to D15)							
0	0	1	0								
0	0	1	1								
0	1	0	0								
0	1	0	1								
0	1	1	0								
0	1	1	1								
1	don’t care			Port (50 to P57)							

For the details of mode selection by the MODE0 to MODE3 pins, refer to **3.3.2 Operating mode specification**.

In ROM-less mode 0 or single-chip mode 1, the MM0 to MM3 bits are initialized to 1110 at system reset, enabling the external expansion mode. External expansion can be disabled by programming the MM0 to MM3 bits and setting the port mode. If MM0 to MM3 are set to ×××1 or 0000, the subsequent external instruction cannot be fetched.

Remark ×: don’t care

12.3.7 Port 6

Port 6 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
P6	P67	P66	P65	P64	P63	P62	P61	P60	Address FFFFF00CH	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	P6n (n = 7 to 0)	Port 6 Input/output port

In addition to their function as port pins, the port 6 pins can also operate in the control mode (external expansion mode) as an address bus used when memory is expanded externally.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 6	P60 to P67	A16 to A23	Address bus in memory expansion	F

(2) Input/output mode/control mode setting

Port 6 input/output mode setting is performed by means of the port 6 mode register (PM6), and control mode (external expansion mode) setting is performed by means of the mode specification pins (MODE0 to MODE3) and the memory expansion mode register (MM: refer to **3.4.6 (1)**).

(a) Port 6 mode register (PM6)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	FFFFF02CH	FFH

Bit Position	Bit Name	Function
7 to 0	PM6n (n = 7 to 0)	Port Mode Sets P6n in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Operation mode of port 6

Bit of MM Register				Operation Mode							
MM3	MM2	MM1	MM0	P60	P61	P62	P63	P64	P65	P66	P67
don’t care	0	0	0	Port (P60 to P67)							
	0	0	1								
	0	1	0								
	0	1	1								
	1	0	0	A16	A17	P62	P63	P64	P65	P66	P67
	1	0	1			A18	A19				
	1	1	0					A20	A21		
	1	1	1							A22	A23

For the details of mode selection by the MODE0 to MODE3 pins, refer to **3.3.2 Operating mode specification**.

In ROM-less modes 0 or 1, or single-chip mode 1, the MM0 to MM3 bits are initialized to 111× at system reset, enabling the external expansion mode. External expansion can be disabled by programming the MM0 to MM3 bits and setting the port mode.

Remark ×: don’t care

12.3.8 Port 7

Port 7 is an 8-bit input only port and all pins of port 7 are fixed in the input mode.

	7	6	5	4	3	2	1	0		
P7	P77	P76	P75	P74	P73	P72	P71	P70	Address FFFFF00EH	After reset Undefined

In addition to their function as port pins, the port 7 pins can also operate as analog inputs for A/D converter.

This port is used also as the analog input pins (ANI0 to ANI7), but the port and analog input pins cannot be switched. By reading the port, the state of each pin can be read.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 7	P70 to P77	ANI0 to ANI7	Analog input for A/D converter	G

12.3.9 Port 8

Port 8 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
P8	P87	P86	P85	P84	P83	P82	P81	P80	Address FFFFF010H	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	P8n (n = 7 to 0)	Port 8 Input/output port

In addition to their function as port pins, in the control mode, the port 8 pins operate as chip select signal outputs, row address strobe signal outputs for DRAM, and read/write strobe signal outputs for external I/O.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 8	P80 to P83	$\overline{\mathrm{CS0}}$/$\overline{\mathrm{RAS0}}$ to $\overline{\mathrm{CS3}}$/$\overline{\mathrm{RAS3}}$	Chip select signal output Row address signal output	O
	P84	$\overline{\mathrm{CS4}}$/$\overline{\mathrm{RAS4}}$/$\overline{\mathrm{IOWR}}$	Chip select signal output Row address signal output Write strobe signal output	P
	P85	$\overline{\mathrm{CS5}}$/$\overline{\mathrm{RAS5}}$/$\overline{\mathrm{IORD}}$	Chip select signal output Row address signal output Read strobe signal output	
	P86, P87	$\overline{\mathrm{CS6}}$/$\overline{\mathrm{RAS6}}$, $\overline{\mathrm{CS7}}$/$\overline{\mathrm{RAS7}}$	Chip select signal output Row address signal output	O

(2) Input/output mode/control mode setting

Port 8 input/output mode setting is performed by means of the port 8 mode register (PM8), and control mode (external expansion mode) setting is performed by means of the mode specification pins (MODE0 to MODE3) and the port 8 mode control register (PMC8).

(a) Port 8 mode register (PM8)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PM8	PM87	PM86	PM85	PM84	PM83	PM82	PM81	PM80	Address FFFFF030H	After reset FFH

Bit Position	Bit Name	Function
7 to 0	PM8n (n = 7 to 0)	Port Mode Sets P8n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Port 8 mode control register (PMC8)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PMC8	PMC87	PMC86	PMC85	PMC84	PMC83	PCM82	PMC81	PMC80	Address FFFFF050H	After reset **Note**

Note Single-chip mode 0: 00H
Single-chip mode 1: FFH
ROM-less mode 0, 1: FFH

Bit Position	Bit Name	Function
7	PMC87	Port Mode Control Sets operation mode of P87 pin. 0: Input/output port mode 1: $\overline{\text{CS7}}$/$\overline{\text{RAS7}}$ output mode
6	PMC86	Port Mode Control Sets operation mode of P86 pin. 0: Input/output port mode 1: $\overline{\text{CS6}}$/$\overline{\text{RAS6}}$ output mode
5	PMC85	Port Mode Control Sets operation mode of P85 pin. Set in combination with PCS8. 0: Input/output port mode 1: $\overline{\text{CS5}}$/$\overline{\text{RAS5}}$ output mode/$\overline{\text{IORD}}$ output mode
4	PMC84	Port Mode Control Sets operation mode of P84 pin. Set in combination with PCS8. 0: Input/output port mode 1: $\overline{\text{CS4}}$/$\overline{\text{RAS4}}$ output mode/$\overline{\text{IOWR}}$ output mode
3	PMC83	Port Mode Control Sets operation mode of P83 pin. 0: Input/output port mode 1: $\overline{\text{CS3}}$/$\overline{\text{RAS3}}$ output mode
2	PMC82	Port Mode Control Sets operation mode of P82 pin. 0: Input/output port mode 1: $\overline{\text{CS2}}$/$\overline{\text{RAS2}}$ output mode
1	PMC81	Port Mode Control Sets operation mode of P81 pin. 0: Input/output port mode 1: $\overline{\text{CS1}}$/$\overline{\text{RAS1}}$ output mode
0	PMC80	Port Mode Control Sets operation mode of P80 pin. 0: Input/output port mode 1: $\overline{\text{CS0}}$/$\overline{\text{RAS0}}$ output mode

(c) Port/control select register 8 (PCS8)

This register can be read/written in 8- or 1-bit units. However, all the bits except for bits 5 and 4 are fixed at 0, so even if 1 is written, it is disregarded.

	7	6	5	4	3	2	1	0	Address	After reset
PCS8	0	0	PCS85	PCS84	0	0	0	0	FFFFF590H	00H

Bit Position	Bit Name	Function
5	PCS85	Port Control Select Specifies the operating mode when pin P85 is in the control mode. 0: $\overline{\text{CS5}}$/$\overline{\text{RAS5}}$ output mode 1: $\overline{\text{IORD}}$ output mode
4	PCS84	Port Control Select Specifies the operating mode when pin P84 is in the control mode. 0: $\overline{\text{CS4}}$/$\overline{\text{RAS4}}$ output mode 1: $\overline{\text{IOWR}}$ output mode

Caution When the port mode is specified by the PMC8 register, the settings of this register are ignored.

12.3.10 Port 9

Port 9 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
P9	P97	P96	P95	P94	P93	P92	P91	P90	Address FFFFF012H	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	P9n (n = 7 to 0)	Port 9 Input/output port

In addition to their function as port pins, the port 9 pins can also operate in the control mode (external expansion mode) as control signal outputs and bus hold control signal output used when memory is expanded externally.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 9	P90	$\overline{\text{LWR}}$/$\overline{\text{LCAS}}$	Control signal output in memory expansion	O
	P91	$\overline{\text{UWR}}$/$\overline{\text{UCAS}}$		
	P92	$\overline{\text{RD}}$		
	P93	$\overline{\text{WE}}$		
	P94	$\overline{\text{BCYST}}$		
	P95	$\overline{\text{OE}}$		
	P96	$\overline{\text{HLDAK}}$	Bus hold acknowledge signal output	
	P97	$\overline{\text{HLDRQ}}$	Bus hold request signal input	H

(2) Input/output mode/control mode setting

Port 9 input/output mode setting is performed by means of the port 9 mode register (PM9), and control mode (external expansion mode) setting is performed by means of the mode specification pins (MODE0 to MODE3) and the port 9 mode control register (PMC9).

(a) Port 9 mode register (PM9)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	FFFFF032H	FFH

Bit Position	Bit Name	Function
7 to 0	PM9n (n = 7 to 0)	Port Mode Sets P9n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Port 9 mode control register (PMC9)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PMC9	PMC97	PMC96	PMC95	PMC94	PMC93	PCM92	PMC91	PMC90	FFFFF052H	**Note**

Note Single-chip mode 0: 00H
Single-chip mode 1: FFH
ROM-less mode 0, 1: FFH

Bit Position	Bit Name	Function
7	PMC97	Port Mode Control Sets operation mode of P97 pin. 0: Input/output port mode 1: $\overline{\text{HLDRQ}}$ input mode
6	PMC96	Port Mode Control Sets operation mode of P96 pin. 0: Input/output port mode 1: $\overline{\text{HLDAK}}$ output mode
5	PMC95	Port Mode Control Sets operation mode of P95 pin. 0: Input/output port mode 1: $\overline{\text{OE}}$ output mode
4	PMC94	Port Mode Control Sets operation mode of P94 pin. 0: Input/output port mode 1: $\overline{\text{BCYST}}$ output mode
3	PMC93	Port Mode Control Sets operation mode of P93 pin. 0: Input/output port mode 1: $\overline{\text{WE}}$ output mode
2	PMC92	Port Mode Control Sets operation mode of P92 pin. 0: Input/output port mode 1: $\overline{\text{RD}}$ output mode
1	PMC91	Port Mode Control Sets operation mode of P91 pin. 0: Input/output port mode 1: $\overline{\text{UWR}}$/$\overline{\text{UCAS}}$ output mode
0	PMC90	Port Mode Control Sets operation mode of P90 pin. 0: Input/output port mode 1: $\overline{\text{LWR}}$/$\overline{\text{LCAS}}$ output mode

12.3.11 Port 10

Port 10 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
P10	P107	P106	P105	P104	P103	P102	P101	P100	Address FFFFF014H	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	P10n (n = 7 to 0)	Port 10 Input/output port

In addition to their function as port pins, the port 10 pins can also operate as real-time pulse unit (RPU) inputs/outputs, external interrupt inputs, and DMA (terminal count) outputs in the control mode.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 10	P100	TO120	Real-time pulse unit (RPU) output	A
	P101	TO121		
	P102	TCLR12	Real-time pulse unit (RPU) input	B
	P103	TI12		
	P104 to P107	INTP120/$\overline{\text{TC0}}$ to INTP123/$\overline{\text{TC3}}$	External interrupt input/DMA (terminal count) output	K

(2) Input/output mode/control mode setting

Port 10 input/output mode setting is performed by means of the port 10 mode register (PM10), and control mode setting is performed by means of the port 10 mode control register (PMC10) and port/control select register 10 (PCS10).

(a) Port 10 mode register (PM10)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PM10	PM107	PM106	PM105	PM104	PM103	PM102	PM101	PM100	Address FFFFF034H	After reset FFH

Bit Position	Bit Name	Function
7 to 0	PM10n (n = 7 to 0)	Port Mode Sets P10n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Port 10 mode control register (PMC10)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PMC10	PMC107	PMC106	PMC105	PMC104	PMC103	PMC102	PMC101	PMC100	FFFFF054H	00H

Bit Position	Bit Name	Function
7 to 4	PMC10n (n = 7 to 4)	Port Mode Control Sets operation mode of P10n pin. Set in combination with PCS10. 0: Input/output port mode 1: External interrupt request (INTP123 to INTP120) input mode/DMA terminal signal ($\overline{\text{TC3}}$ to $\overline{\text{TC0}}$) output mode
3	PMC103	Port Mode Control Sets operation mode of P103 pin. 0: Input/output port mode 1: TI12 input mode
2	PMC102	Port Mode Control Sets operation mode of P102 pin. 0: Input/output port mode 1: TCLR12 input mode
1	PMC101	Port Mode Control Sets operation mode of P101 pin. 0: Input/output port mode 1: TO121 output mode
0	PMC100	Port Mode Control Sets operation mode of P100 pin. 0: Input/output port mode 1: TO120 output mode

(c) Port/control select register 10 (PCS10)

This register can be read/written in 8- or 1-bit units. However, bits 3 to 0 are fixed at 0, so even if 1 is written, it is disregarded.

	7	6	5	4	3	2	1	0		
PCS10	PCS107	PCS106	PCS105	PCS104	0	0	0	0	Address FFFFF594H	After reset 00H

Bit Position	Bit Name	Function
7	PCS107	Port Control Select Specifies the operating mode when pin P107 is in the control mode. 0: INTP123 input mode 1: $\overline{\text{TC3}}$ output mode
6	PCS106	Port Control Select Specifies the operating mode when pin P106 is in the control mode. 0: INTP122 input mode 1: $\overline{\text{TC2}}$ output mode
5	PCS105	Port Control Select Specifies the operating mode when pin P105 is in the control mode. 0: INTP121 input mode 1: $\overline{\text{TC1}}$ output mode
4	PCS104	Port Control Select Specifies the operating mode when pin P104 is in the control mode. 0: INTP120 input mode 1: $\overline{\text{TC0}}$ output mode

Caution When the port mode is specified by the PMC10 register, the settings of this register are ignored.

12.3.12 Port 11

Port 11 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
P11	P117	P116	P115	P114	P113	P112	P111	P110	Address FFFFF016H	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	P11n (n = 7 to 0)	Port 11 Input/output port

In addition to their function as port pins, the port 11 pins can also operate as real-time pulse unit (RPU) inputs/outputs, external interrupt request inputs, and serial interface (CSI3) inputs/outputs in the control mode.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 11	P110	TO140	Real-time pulse unit (RPU) output	A
	P111	TO141		
	P112	TCLR14	Real-time pulse unit (RPU) input	B
	P113	TI14		
	P114	INTP140	External interrupt input	
	P115	INTP141/SO3	External interrupt input/serial interface (CSI3) input/output	K
	P116	INTP142/SI3		M
	P117	INTP143/$\overline{\text{SCK3}}$		N

(2) Input/output mode/control mode setting

Port 11 input/output mode setting is performed by means of the port 11 mode register (PM11), and control mode setting is performed by means of the port 11 mode control register (PMC11) and port/control select register 11 (PCS11).

★ **(a) Port 11 mode register (PM11)**

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PM11	PM117	PM116	PM115	PM114	PM113	PM112	PM111	PM110	FFFFF036H	FFH

Bit Position	Bit Name	Function
7 to 0	PM11n (n = 7 to 0)	Port Mode Sets P11n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Port 11 mode control register (PMC11)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PMC11	PMC117	PMC116	PMC115	PMC114	PMC113	PMC112	PMC111	PMC110	FFFFF056H	00H

Bit Position	Bit Name	Function
7 to 5	PMC11n (n = 7 to 5)	Port Mode Control Sets operation mode of P11n pin. Set in combination with PCS11. 0: Input/output port mode 1: External interrupt request (INTP143 to INTP141) input mode/CSI3 ($\overline{\text{SCK3}}$, SI3, SO3) input/output mode
4	PMC114	Port Mode Control Sets operation mode of P114 pin. 0: Input/output port mode 1: INTP140 input mode
3	PMC113	Port Mode Control Sets operation mode of P113 pin. 0: Input/output port mode 1: TI14 input mode
2	PMC112	Port Mode Control Sets operation mode of P112 pin. 0: Input/output port mode 1: TCLR14 input mode
1	PMC111	Port Mode Control Sets operation mode of P111 pin. 0: Input/output port mode 1: TO141 output mode
0	PMC110	Port Mode Control Sets operation mode of P110 pin. 0: Input/output port mode 1: TO140 output mode

(c) Port/control select register 11 (PCS11)

This register can be read/written in 8- or 1-bit units. However, except for bit 5, all bits are fixed at 0, so even if 1 is written, it is disregarded.

	7	6	5	4	3	2	1	0	Address	After reset
PCS11	0	0	PCS115	0	0	0	0	0	FFFFF596H	00H

Bit Position	Bit Name	Function
5	PCS115	Port Control Select Specifies the operating mode when pins P117 to P115 are in the control mode. 0: INTP143 input mode (P117) INTP142 input mode (P116) INTP141 input mode (P115) 1: $\overline{\text{SCK3}}$ input/output mode (P117) SI3 input mode (P116) SO3 output mode (P115)

Caution When the port mode is specified by the PMC11 register, the settings of this register are ignored.

12.3.13 Port 12

Port 12 is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
P12	P127	P126	P125	P124	P123	P122	P121	P120	Address FFFFF018H	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	P12n (n = 7 to 0)	Port 12 Input/output port

In addition to their function as port pins, the port 12 pins can also operate as real-time pulse unit (RPU) inputs/outputs, external interrupt request inputs, and A/D converter trigger input in the control mode.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port 12	P120	TO150	Real-time pulse unit (RPU) output	A
	P121	TO151		
	P122	TCLR15	Real-time pulse unit (RPU) input	B
	P123	TI15		
	P124 to P126	INTP150 to INTP152	External interrupt input	
	P127	INTP153/ADTRG	External interrupt input/AD converter external trigger input	

(2) Input/output mode/control mode setting

Port 12 input/output mode setting is performed by means of the port 12 mode register (PM12), and control mode setting is performed by means of the port 12 mode control register (PMC12).

(a) Port 12 mode register (PM12)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PM12	PM127	PM126	PM125	PM124	PM123	PM122	PM121	PM120	Address FFFFF038H	After reset FFH

Bit Position	Bit Name	Function
7 to 0	PM12n (n = 7 to 0)	Port Mode Sets P12n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Port 12 mode control register (PMC12)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PMC12	PMC127	PMC126	PMC125	PMC124	PMC123	PMC122	PMC121	PMC120	FFFFF058H	00H

Bit Position	Bit Name	Function
7	PMC127	Port Mode Control Sets operation mode of P127 pin. 0: Input/output port mode 1: External interrupt request (INTP153) input mode A/D converter external trigger (ADRTG) input mode[Note]
6 to 4	PMC12n (n = 6 to 4)	Port Mode Control Sets operation mode of P12n pin. 0: Input/output port mode 1: External interrupt request (INTP152 to INTP150) input mode
3	PMC123	Port Mode Control Sets operation mode of P123 pin. 0: Input/output port mode 1: TI15 input mode
2	PMC122	Port Mode Control Sets operation mode of P122 pin. 0: Input/output port mode 1: TCLR15 input mode
1	PMC121	Port Mode Control Sets operation mode of P121 pin. 0: Input/output port mode 1: TO151 output mode
0	PMC120	Port Mode Control Sets operation mode of P120 pin. 0: Input/output port mode 1: TO150 output mode

Note If the TRG bit of the A/D converter mode register (ADM1) is set in the external trigger mode when bit PMC127 = 1, it functions as an A/D converter external trigger input (ADTRG).

12.3.14 Port A

Port A is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
PA	PA7	PA6	PA5	PA4	PA3	PA2	PA1	PA0	Address FFFFF01CH	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	PAn (n = 7 to 0)	Port A Input/output port

In addition to their function as port pins, the port A pins can also operate in the control mode (external expansion mode) as an address bus used when memory is expanded externally.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port A	PA0 to PA7	A0 to A7	Address bus in memory expansion	F

(2) Input/output mode/control mode setting

Port A input/output mode setting is performed by means of the port A mode register (PMA), and control mode (external expansion mode) setting is performed by means of the mode specification pins (MODE0 to MODE3) and the memory expansion mode register (MM: refer to **3.4.6 (1)**).

(a) Port A mode register (PMA)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PMA	PMA7	PMA6	PMA5	PMA4	PMA3	PMA2	PMA1	PMA0	Address FFFFF03CH	After reset FFH

Bit Position	Bit Name	Function
7 to 0	PMAn (n = 7 to 0)	Port Mode Sets PAn pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Operation mode of port A

Bit of MM Register				Operation Mode							
MM3	MM2	MM1	MM0	PA0	PA1	PA2	PA3	PA4	PA5	PA6	PA7
don’t care	0	0	0	Port (PA0 to PA7)							
	0	0	1	Address bus (A0 to A7)							
	0	1	0								
	0	1	1								
	1	0	0								
	1	0	1								
	1	1	0								
	1	1	1								

For the details of mode selection by the MODE0 to MODE3 pins, refer to **3.3.2 Operating mode specification**.

In ROM-less modes 0 or 1, or single-chip mode 1, the MM0 to MM3 bits are initialized to 111× at system reset, enabling the external address output mode. If MM0 to MM3 are set to 000× by the program, the port mode can be changed to, but the subsequent external instruction cannot be fetched from data bus.

Remark ×: don’t care

12.3.15 Port B

Port B is an 8-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0		
PB	PB7	PB6	PB5	PB4	PB3	PB2	PB1	PB0	Address FFFFF01EH	After reset Undefined

Bit Position	Bit Name	Function
7 to 0	PBn (n = 7 to 0)	Port B Input/output port

In addition to their function as port pins, the port B pins can also operate in the control mode (external expansion mode) as an address bus used when memory is expanded externally.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port B	PB0 to PB7	A8 to A15	Address bus in memory expansion	F

(2) Input/output mode/control mode setting

Port B input/output mode setting is performed by means of the port B mode register (PMB), and control mode (external expansion mode) setting is performed by means of the mode specification pins (MODE0 to MODE3) and the memory expansion mode register (MM: refer to **3.4.6 (1)**).

(a) Port B mode register (PMB)

This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0		
PMB	PMB7	PMB6	PMB5	PMB4	PMB3	PMB2	PMB1	PMB0	Address FFFFF03EH	After reset FFH

Bit Position	Bit Name	Function
7 to 0	PMBn (n = 7 to 0)	Port Mode Sets PBn pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

(b) Operation mode of port B

Bit of MM Register				Operation Mode							
MM3	MM2	MM1	MM0	PB0	PB1	PB2	PB3	PB4	PB5	PB6	PB7
don’t care	0	0	0	Port (PB0 to PB7)							
	0	0	1	A8	A9	A10	A11	PB4	PB5	PB6	PB7
	0	1	0					A12	A13		
	0	1	1							A14	A15
	1	0	0								
	1	0	1								
	1	1	0								
	1	1	1								

For the details of mode selection by the MODE0 to MODE3 pins, refer to **3.3.2 Operating mode specification**.

In ROM-less modes 0 or 1, or single-chip mode 1, the MM0 to MM3 bits are initialized to 111× at system reset, enabling the external address output mode. If MM0 to MM3 are set to 000× by the program, the port mode can be changed to, but the subsequent external instruction cannot be fetched from data bus. Also, if MM0 to MM3 are set to 100x or 010x, the subsequent external address output from port B is disabled.

Remark ×: don’t care

12.3.16 Port X

Port X is a 3-bit input/output port that can be set to input or output in 1-bit units.

	7	6	5	4	3	2	1	0	Address	After reset
PX	PX7	PX6	PX5	—	—	—	—	—	FFFFF41AH	Undefined

Bit Position	Bit Name	Function
7 to 5	PXn (n = 7 to 5)	Port X Input/output port

In addition to their function as port pins, the port X pins can also operate as DRAM refresh request signal output, wait control input, and internal system clock output in the control mode. Lower 5 bits of port X are always undefined in the case of 8-bit access.

(1) Operation in control mode

Port		Control Mode	Remark	Block Type
Port X	PX5	$\overline{\text{REFRQ}}$	DRAM refresh request signal output	A
	PX6	$\overline{\text{WAIT}}$	Wait control input	L
	PX7	CLKOUT	Internal system clock output	A

(2) Input/output mode/control mode setting

Port X input/output mode setting is performed by means of the port X mode register (PMX), and control mode setting is performed by means of the port X mode control register (PMCX).

(a) Port X mode register (PMX)

★ This register is write-only, in 8-bit units. However, the lower 5 bits are fixed at 1 by hardware, so even if 0 is written, it is disregarded.

	7	6	5	4	3	2	1	0	Address	After reset
PMX	PMX7	PMX6	PMX5	1	1	1	1	1	FFFFF43AH	FFH

Bit Position	Bit Name	Function
7 to 5	PMXn (n = 7 to 5)	Port Mode Sets PXn pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

★ **Caution Do not change the port mode using a bit manipulation instruction (CLR1, NOT1, SET1, TST1).**

(b) Port X mode control register (PMCX)

★ This register is write-only, in 8-bit units. However, the lower 5 bits are fixed at 0 by hardware, so even if 1 is written, it is disregarded.

	7	6	5	4	3	2	1	0		
PMCX	PMCX7	PMCX6	PMCX5	0	0	0	0	0	Address FFFFF45AH	After reset **Note**

Note Single-chip mode 0: 00H
Single-chip mode 1: E0H
ROM-less mode 0, 1: E0H

Bit Position	Bit Name	Function
7	PMCX7	Port Mode Control Sets operation mode of PX7 pin. 0: Input/output port mode 1: CLKOUT output mode
6	PMCX6	Port Mode Control Sets operation mode of PX6 pin. 0: Input/output port mode 1: $\overline{\text{WAIT}}$ input mode
5	PMCX5	Port Mode Control Sets operation mode of PX5 pin. 0: Input/output port mode 1: $\overline{\text{REFRQ}}$ output mode

★ **Caution Do not change the operation mode using a bit manipulation instruction (CLR1, NOT1, SET1, TST1).**

CHAPTER 13 RESET FUNCTIONS

When a low-level signal is input to the $\overline{\text{RESET}}$ pin, a system reset is effected and the hardware is initialized.

When the $\overline{\text{RESET}}$ signal level changes from low to high, the reset state is released and program execution is started. Register contents must be initialized as required in the program.

13.1 Features

The reset pin ($\overline{\text{RESET}}$) incorporates a noise eliminator which uses analog delay (≅ 60 ns) to prevent malfunction due to noise.

13.2 Pin Functions

During a system reset, most pins (all but the CLKOUT[Note], $\overline{\text{RESET}}$, X2, HV_{DD}, V_{DD}, V_{SS}, CV_{DD}, CV_{SS}, AV_{DD}, AV_{SS}, and AV_{REF} pins) enter the high impedance state. Therefore, when memory is connected externally, a pull-up or pull-down resistor must be connected to the specified pins of ports 4, 5, 6, 8, 9, A, B, and X. If no resister is connected there, memory contents may be lost when these pins enter high impedance state. For the same reason, the output pins of the internal peripheral I/O functions and output ports should be handled in the same manner.

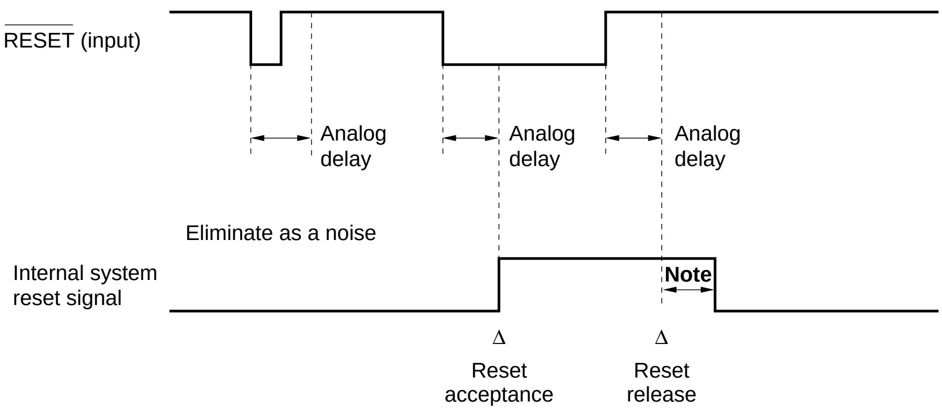
Note In ROM-less modes 0 and 1, and in single-chip mode 1, the CLKOUT signal is output even during reset. In single-chip mode 0, the CLKOUT signal is not output until the PMCX register is set.

Table 13-1 shows the operating state of each output pin and each input/output pin during reset.

Table 13-1. Operating State of Each Pin During Reset

Pin Name		Pin State			
		When in Single-Chip Mode 0	When in Single-Chip Mode 1	When in ROM-less Mode 0	When in ROM-less Mode 1
D0 to D7, A0 to A23, $\overline{\text{CS0}}$ to $\overline{\text{CS7}}$, $\overline{\text{RAS0}}$ to $\overline{\text{RAS7}}$, $\overline{\text{LCAS}}$, $\overline{\text{LWR}}$, $\overline{\text{UCAS}}$, $\overline{\text{UWR}}$, $\overline{\text{RD}}$, $\overline{\text{WE}}$, $\overline{\text{BCYST}}$, $\overline{\text{OE}}$, $\overline{\text{HLDAK}}$, $\overline{\text{REFRQ}}$		(Port mode)	High-impedance		
D8 to D15		(Port mode)	High-impedance		(Port mode)
$\overline{\text{WAIT}}$, $\overline{\text{HLDRQ}}$		(Port mode)	(Input)		
CLKOUT		(Port mode)	Operating		
Port pin	Ports 0 to 3, 10 to 12	(Input)			
	Ports 4, 6, 8, 9, A, B, X	(Input)	(Control mode)		
	Port 5	(Input)	(Control mode)		(Input)

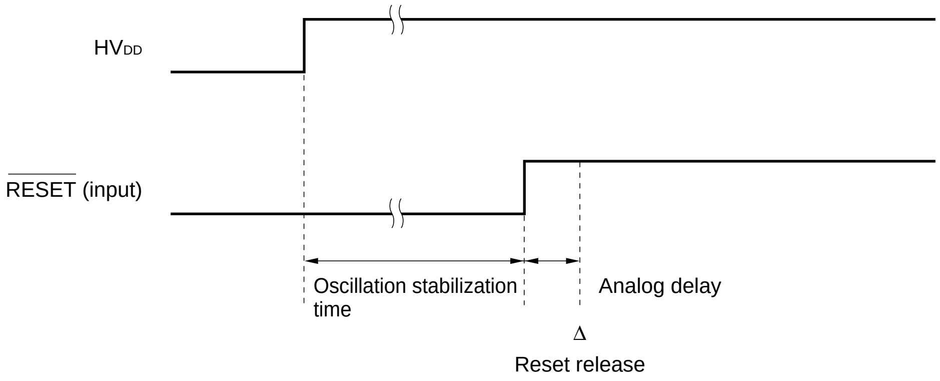
(1) Receiving the reset signal



Note The internal system reset signal continues in the active state for at least 4 system clock cycles after reset clear timing by the $\overline{\text{RESET}}$ signal.

(2) Reset during power on

In the reset operation during power on (when the power is turned on), in accordance with the low-level width of the $\overline{\text{RESET}}$ signal, it is necessary to secure an oscillation stabilization time of 10 ms or greater from power rise to the reception of the reset.


HV$_{DD}$
$\overline{\text{RESET}}$ (input)
Oscillation stabilization time
Analog delay
Δ
Reset release

13.3 Initialization

The initial values of the CPU, internal RAM and internal peripheral I/O after reset are shown in Table 13-2.

Initialize the contents of each register as necessary during program operation. Particularly, the registers shown below are related to system settings, so set them as necessary.

○ Power save control register (PSC): Sets the functions of pins X1 and X2, the operation of the CLKOUT pin, etc.

○ Data wait control register (DWC): Sets the number of data wait states.

Table 13-2. Initial Values of CPU, Internal RAM, and Internal Peripheral I/O after Reset (1/2)

Internal Hardware		Register Name	Initial Value After Reset
CPU	Program registers	General-purpose register (r0)	00000000H
		General-purpose registers (r1 to r31)	Undefined
		Program counter (PC)	00000000H
	System registers	Status saving register during interrupt (EIPC, EIPSW)	Undefined
		Status saving register during NMI (FEPC, FEPSW)	Undefined
		Interrupt control register (ECR)	00000000H
		Program status word (PSW)	00000020H
		Status saving register during CALLT execution (CTPC, CTPSW)	Undefined
		Status saving register during exception trap (DBPC, DBPSW)	Undefined
		CALLT base pointer (CTBP)	Undefined
Internal RAM		—	Undefined
Internal peripheral I/O		Command register (PRCMD)	Undefined
	Bus control functions	Data wait control register (DWC1)	FFFFH
		Data wait control register (DWC2)	FFH
		Bus cycle control register (BCC)	5555H
		Bus cycle type configuration register (BCT)	0000H
		Bus size configuration register (BSC)	5555H/0000H
	Memory control functions	DRAM configuration registers (DRC0 to DRC3)	3FC1H
		DRAM type configuration register (DTC)	0000H
		Page ROM configuration register (PRC)	E0H
		Refresh control registers (RFC0 to RFC3)	0000H
		Refresh wait control register (RWC)	00H
	DMA functions	Control registers (DADC0 to DADC3)	0000H
		Source address registers (DSA0H to DSA3H, DSA0L to DSA3L)	Undefined
		Channel control registers (DCHC0 to DCHC3)	00H
		Destination address registers (DDA0H to DDA3H, DDA0L to DDA3L)	Undefined
		Trigger factor registers (DTFR0 to DTFR3)	00H
		Byte count registers (DBC0 to DBC3)	Undefined
		Fly-by transfer data wait control register (FDW)	00H
		DMA disable status register (DDIS)	00H
		DMA restart register (DRST)	00H
	Interrupt/exception control functions	In-service priority register (ISPR)	00H
		External interrupt mode registers (INTM0 to INTM6)	00H
		Interrupt control registers (OVIC10 to OVIC15, CMIC40, CMIC41, P10IC0 to P10IC3, P11IC0 to P11IC3, P12IC0 to P12IC3, P13IC0 to P13IC3, P14IC0 to P14IC3, P15IC0 to P15IC3, DMAIC0 to DMAIC3, CSIC0 to CSIC3, SEIC0, STIC0, SRIC0, SRIC1, SEIC1, STIC1, ADIC)	47H

Table 13-2. Initial Values of CPU, Internal RAM, and Internal Peripheral I/O after Reset (2/2)

Internal Hardware		Register Name	Initial Value After Reset
Internal peripheral I/O	Clock generator functions	System status register (SYS)	0000000×B
		Clock control register (CKC)	00H
		Power save control register (PSC)	00H
	Timer/counter functions	Capture/compare registers (CC100 to CC103, CC110 to CC113, CC120 to CC123, CC130 to CC133, CC140 to CC143, CC150 to CC153)	Undefined
		Compare registers (CM40, CM41)	Undefined
		Timer overflow status register (TOVS)	00H
		Timer control register (TMC10 to TMC15, TMC40, TMC41)	00H
		Timer unit mode register (TUM10 to TUM15)	0000H
		Timers (TM10 to TM15, TM40, TM41)	0000H
		Timer output control registers (TOC10 to TOC15)	00H
	Serial interface functions	Asynchronous serial interface status registers (ASIS0, ASIS1)	00H
		Asynchronous serial interface mode registers (ASIM00, ASIM10)	80H
		Asynchronous serial interface mode registers (ASIM01, ASIM11)	00H
		Receive buffers (RXB0, RXB1, RXB0L, RXB1L)	Undefined
		Transmit shift registers (TXS0, TXS1, TXS0L, TXS1L)	Undefined
		Clocked serial interface mode registers (CSIM0 to CSIM3)	00H
		Serial I/O shift registers (SIO0 to SIO3)	Undefined
		Baud rate generator compare registers (BRGC0 to BRGC2)	Undefined
		Baud rate generator prescaler mode registers (BPRM0 to BPRM2)	00H
	A/D converters	Mode register (ADM0)	00H
		Mode register (ADM1)	07H
		A/D conversion result registers (ADCR0 to ADCR7, ADCR0H to ADCR7H)	Undefined
	Port functions	Ports (P0 to P12, PA, PB, PX)	Undefined
		Port/control select registers (PCS0, PCS1, PCS3, PCS8, PCS10, PCS11)	00H
		Mode registers (PM0 to PM12, PMA, PMB, PMX)	FFH
		Mode control registers (PMC0, PMC1, PMC3, PMC10 to PMC12)	00H
		Mode control register (PMC2)	01H
		Mode control registers (PMC8, PCM9)	00H/FFH
		Mode control register (PMCX)	00H/E0H
		Memory expansion mode register (MM)	00H/07H/0FH

Caution "Undefined" in the above table is undefined during power-on reset, or undefined as a result of data destruction when $\overline{\text{RESET}}$ is input and the data writing timing has been synchronized. For other $\overline{\text{RESET}}$s, data is held in the same state it was in before the $\overline{\text{RESET}}$ operation.

Remark ×: Undefined

CHAPTER 14 FLASH MEMORY (μPD70F3102, 70F3102A)

The μPD70F3102 and 70F3102A are V850E/MS1 on-chip flash memory products with a 128KB flash memory. In the instruction fetch to this flash memory, 4 bytes can be accessed by a single clock, just as in the mask ROM version.

Writing to flash memory can be performed with the device mounted on the target system (on board). A dedicated flash programmer is connected to the target system to perform writing.

The following can be considered as the development environment and applications of flash memory.

- Software can be altered after the V850E/MS1 is solder-mounted on the target system.
- Small-scale production of various models is made easier by differentiating software.
- Data adjustment in starting mass production is made easier.

14.1 Features

- 4-byte/1-clock access (in instruction fetch access)
- All area one-shot erase
- Erase in 4KB block units
- Communication through serial interface from the dedicated flash programmer
- Erase/write voltage: V_{PP} = 7.8 V
- On-board programming
- Number of rewrites: 100 times (target)

14.2 Writing by Flash Programmer

Writing can be performed either on-board or off-board by the dedicated flash programmer.

(1) On-board programming

The contents of the flash memory are rewritten after the V850E/MS1 is mounted on the target system. Mount connectors, etc., on the target system to connect the dedicated flash programmer.

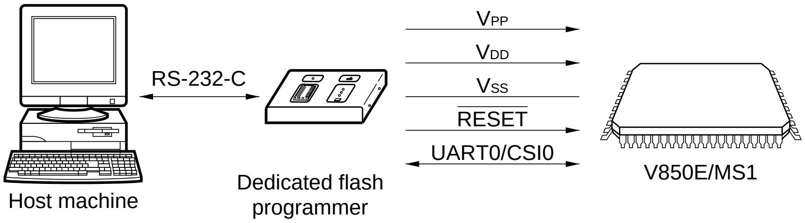
(2) Off-board programming

Writing to flash memory is performed by the dedicated program adapter (FA Series), etc., before mounting the V850E/MS1 on the target system.

Remark The FA Series is a product of Naito Densei Machida Mfg. Co., Ltd.

14.3 Programming Environment

The following shows the environment required for writing programs to the flash memory of the V850E/MS1.



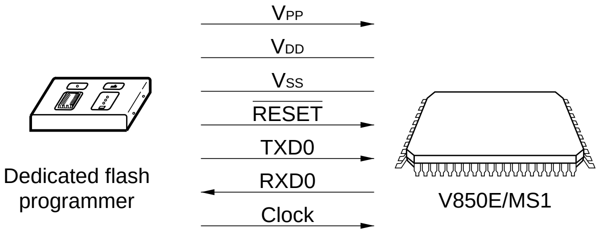
A host machine is required for controlling the dedicated flash programmer.

UART0 or CSI0 is used for the interface between the dedicated flash programmer and the V850E/MS1 to perform writing, erasing, etc. A dedicated program adapter (FA Series) is required for off-board writing.

14.4 Communication System

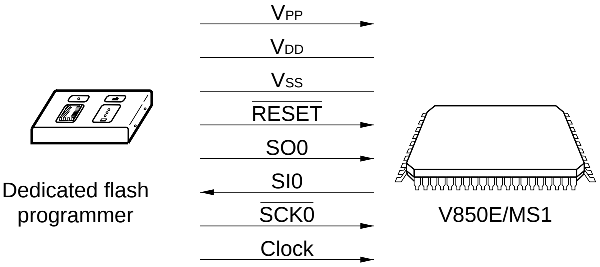
(1) UART0

Transfer rate: 4,800 to 76,800 bps (LSB first)



(2) CSI0

Transfer rate: up to 10 Mbps (MSB first)



The dedicated flash programmer outputs the transfer clock, and the V850E/MS1 operates as a slave.

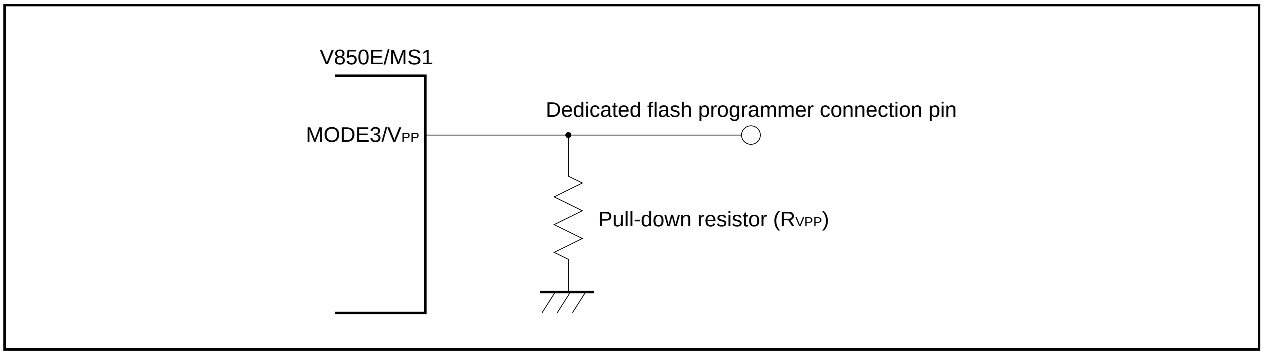
14.5 Pin Handling

When performing on-board writing, install a connector on the target system to connect to the dedicated flash programmer. Also, install a function on-board to switch from the normal operation mode (single-chip modes 0 and 1 or ROM-less modes 0 and 1) to the flash memory programming mode.

When switched to the flash memory programming mode, all the pins not used for the flash memory programming become the same status as that immediately after reset of single-chip mode 0. Therefore, all the ports become output high-impedance status, so that pin handling is required when the external device does not acknowledge the output high-impedance status.

14.5.1 MODE3/V_{PP} pin

In the normal operation mode, 0 V is input to the MODE3/V_{PP} pin. In the flash memory programming mode, 7.8 V writing voltage is supplied to the MODE3/V_{PP} pin. The following shows an example of the connection of the MODE3/V_{PP} pin.


V850E/MS1
MODE3/VPP
Dedicated flash programmer connection pin
Pull-down resistor (RVPP)

14.5.2 Serial interface pin

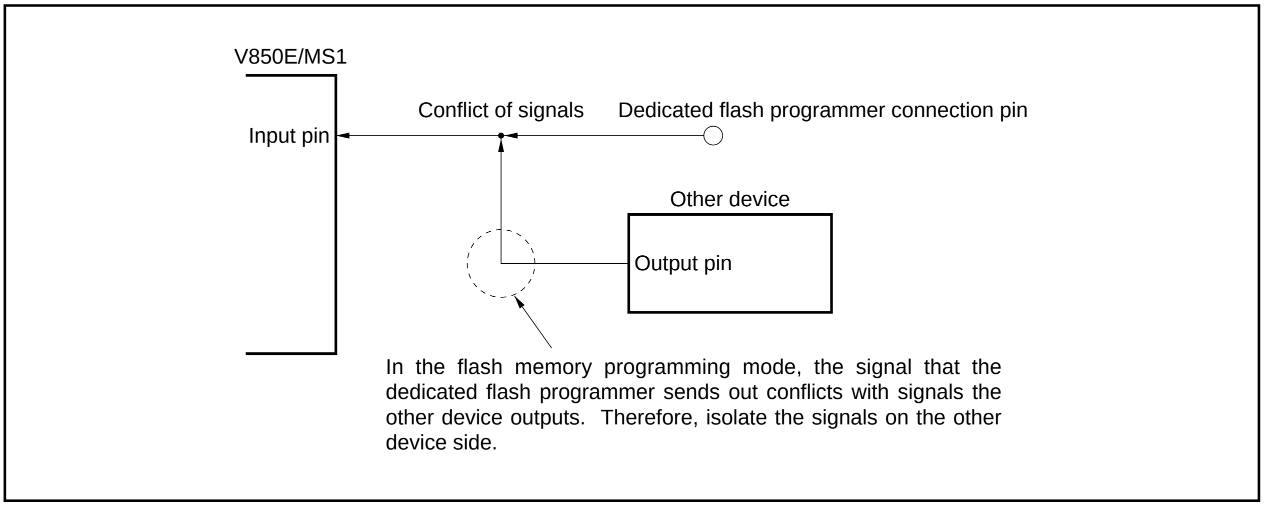
The following shows the pins used by each serial interface.

Serial Interface	Pins Used
CSI0	SO0, SI0, $\overline{\text{SCK0}}$
UART0	TXD0, RXD0

When connecting a dedicated flash programmer to a serial interface pin that is connected to other devices on-board, care should be taken to avoid the conflict of signals and the malfunction of other devices, etc.

(1) Conflict of signals

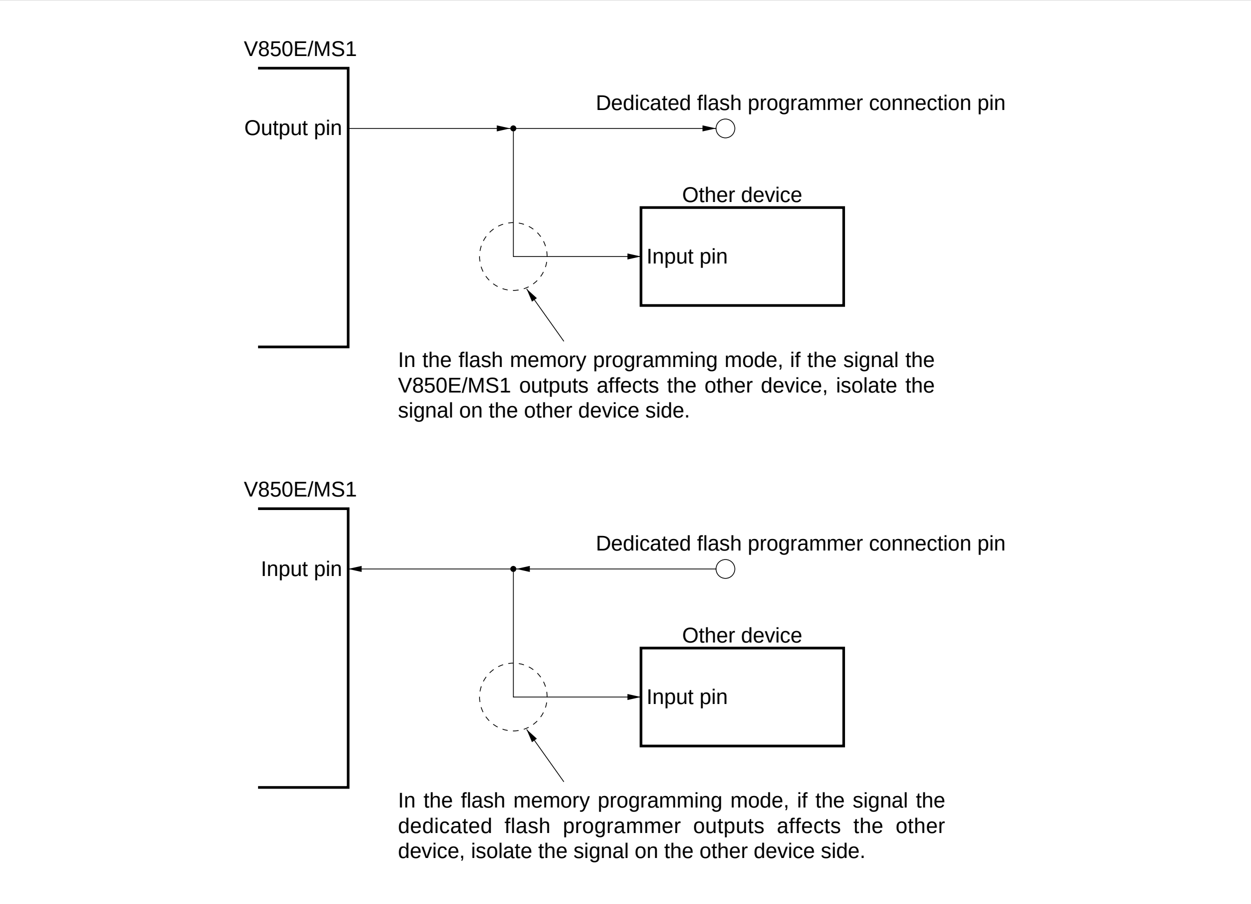
When connecting a dedicated flash programmer (output) to a serial interface pin (input) which is connected to another device (output), conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the other device or set the other device to the output high-impedance status.



In the flash memory programming mode, the signal that the dedicated flash programmer sends out conflicts with signals the other device outputs. Therefore, isolate the signals on the other device side.

(2) Malfunction of the other device

When connecting a dedicated flash programmer (output or input) to a serial interface pin (input or output) connected to another device (input), the signal output to the other device may cause the device to malfunction. To avoid this, isolate the connection to the other device or make the setting so that the input signal to the other device is ignored.


V850E/MS1
Output pin
Dedicated flash programmer connection pin
Other device
Input pin

In the flash memory programming mode, if the signal the V850E/MS1 outputs affects the other device, isolate the signal on the other device side.

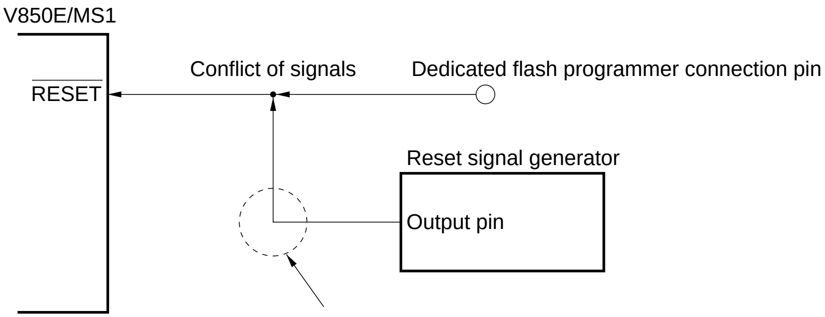
V850E/MS1
Input pin
Dedicated flash programmer connection pin
Other device
Input pin

In the flash memory programming mode, if the signal the dedicated flash programmer outputs affects the other device, isolate the signal on the other device side.

14.5.3 $\overline{\text{RESET}}$ pin

When connecting the reset signals of the dedicated flash programmer to the $\overline{\text{RESET}}$ pin that is connected to the reset signal generation circuit on-board, conflict of signals occurs. To avoid the conflict of signals, isolate the connection to the reset signal generator.

When reset signal is input from the user system during the flash memory programming mode, programming operation will not be performed correctly. Therefore, do not input signals other than the reset signals from the dedicated flash programmer.


V850E/MS1
$\overline{\text{RESET}}$
Conflict of signals
Dedicated flash programmer connection pin
Reset signal generator
Output pin

In the flash memory programming mode, the signal the reset signal generation circuit outputs conflicts with the signal the dedicated flash programmer outputs. Therefore, isolate the signals on the reset signal generator side.

14.5.4 NMI pin

Do not change the input signal to the NMI pin during the flash memory programming mode. If the NMI pin is changed during the flash memory programming mode, the programming may not be performed correctly.

14.5.5 MODE0 to MODE2 pins

If MODE0 to MODE2 are set as follows and a write voltage (7.8 V) is applied to the MODE3/V_{PP} pin and reset is canceled, these pins change to the flash memory programming mode.

- MODE0: Low-level input
- MODE1: High-level input
- MODE2: Low-level input

14.5.6 Port pin

When the flash memory programming mode is set, all the port pins except the pins which communicate with the dedicated flash programmer become output high-impedance status. The treatment of these port pins is not necessary. If problems such as disabling output high-impedance status should occur to the external devices connected to the port, connect them to V_{DD} or V_{SS} through resistors.

14.5.7 $\overline{\text{WAIT}}$ pin

Input high- or low-level signals relative to HV_{DD} to $\overline{\text{WAIT}}$ pin.

14.5.8 Other signal pins

Connect X1, X2, and AV_{REF} to the same status as that in the normal operation mode.

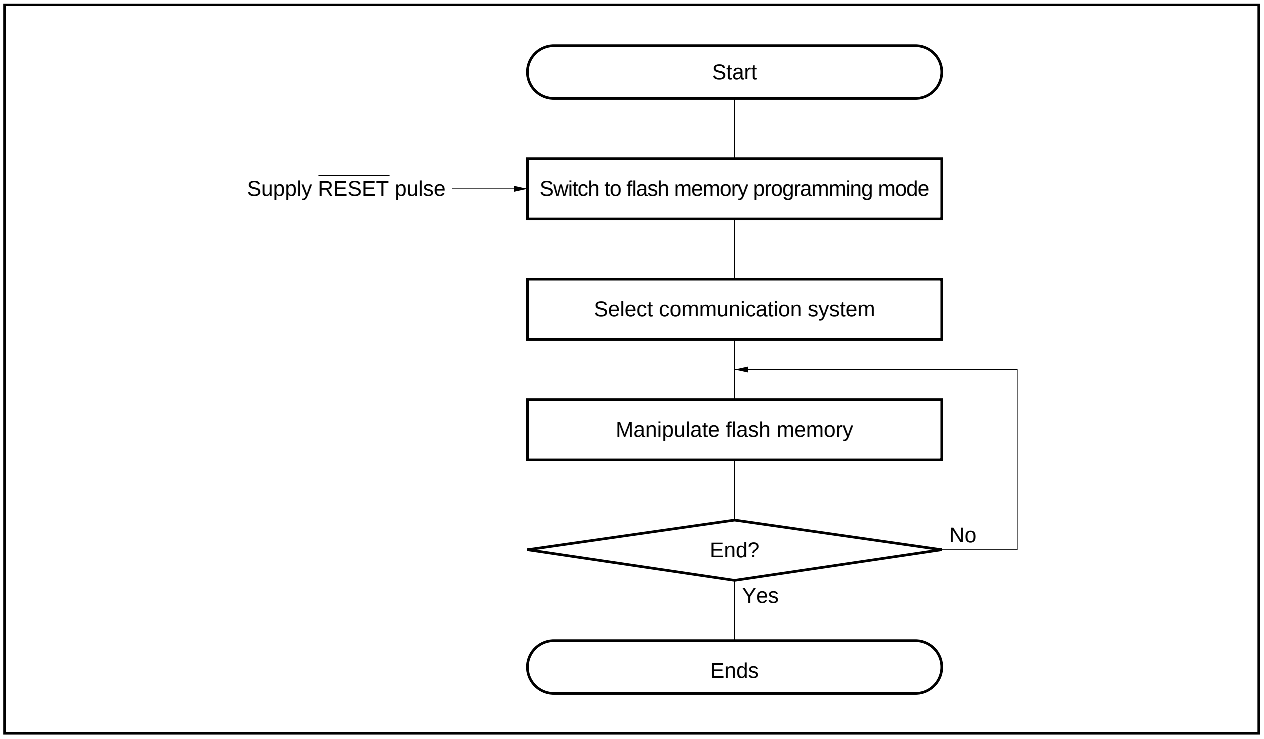
14.5.9 Power supply

Supply the power supply (V_{DD}, HV_{DD}, V_{SS}, AV_{DD}, AV_{SS}, CV_{DD}, and CV_{SS}) the same as that in normal operation mode. Connect V_{DD} and GND of the dedicated flash programmer to V_{DD} and V_{SS}. (V_{DD} of the dedicated flash programmer is provided with power supply monitoring function.)

14.6 Programming Method

14.6.1 Flash memory control

The following shows the procedure for manipulating the flash memory.

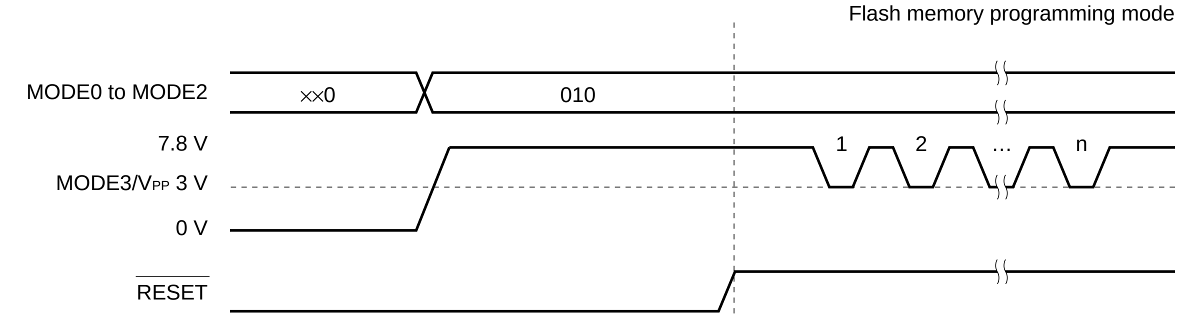

Start
Supply $\overline{\text{RESET}}$ pulse
Switch to flash memory programming mode
Select communication system
Manipulate flash memory
End?
No
Yes
Ends

14.6.2 Flash memory programming mode

When rewriting the contents of flash memory using the dedicated flash programmer, set the V850E/MS1 in the flash memory programming mode. When switching modes, set the MODE0 to MODE2 and MODE3/V_{PP} pins before releasing reset.

When performing on-board writing, change modes using a jumper, etc.

- MODE0: Low-level input
- MODE1: High-level input
- MODE2: Low-level input
- MODE3/V_{PP}: 7.8 V



Remark ×: don’t care

14.6.3 Selection of communication mode

In the V850E/MS1, a communication mode is selected by inputting pulses (16 pulses max.) to V_{PP} pin after switching to the flash memory programming mode. The V_{PP} pulse is generated by the dedicated flash programmer.

The following shows the relationship between the number of pulses and the communication modes.

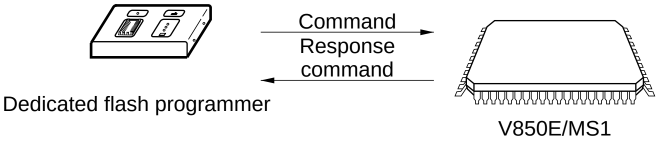
Table 14-1. List of Communication Modes

V_{PP} Pulse	Communication Mode	Remarks
0	CSI0	V850E/MS1 performs slave operation, MSB first
8	UART0	Communication rate: 9600 bps (after reset), LSB first
Others	RFU (reserved)	Setting prohibited

Caution When UART0 is selected, the receive clock is calculated based on the reset command sent from the dedicated flash programmer after receiving the V_{PP} pulse.

14.6.4 Communication command

The V850E/MS1 communicates with the dedicated flash programmer by means of commands. A command sent from the dedicated flash programmer to the V850E/MS1 is called a “command”. The response signal sent from the V850E/MS1 to the dedicated flash programmer is called a “response command”.



The following shows the commands of the firmware for flash memory control of the V850E/MS1. All of these commands are issued from the dedicated flash programmer, and the V850E/MS1 performs the various processing corresponding to the commands.

Category	Command Name	Function
Verify	One-shot verify command	Compares the contents of the entire memory and the input data.
	Block verify command	Compares the contents of the specified memory block and the input data.
Erase	One-shot erase command	Erases the contents of the entire memory.
	Block erase command	Erases the contents of the specified memory block setting 4 Kbytes as one memory block.
	Write back command	Writes back the contents that is over-erased.
Blank check	One-shot blank check command	Checks the erase state of the entire memory.
	Block blank check command	Checks the erase of the specified memory block.
Data write	High-speed write command	Writes data by the specification of the write address and the number of bytes to be written, and executes verify check.
	Continuous write command	Writes data from the address following the high-speed write command executed immediately before, and executes verify check.
System setting and control	Status read out command	Acquires the status of operations.
	Oscillating frequency setting command	Sets the oscillating frequency.
	Erasing time setting command	Sets the erasing time of one-shot erase.
	Writing time setting command	Sets the writing time of data write.
	Write back time setting command	Sets the write back time.
	Baud rate setting command	Sets the baud rate when using UART0.
	Silicon signature command	Reads outs the silicon signature information.
	Reset command	Escapes from each state.

The V850E/MS1 sends back response commands to the commands issued from the dedicated flash programmer. The following shows the response commands the V850E/MS1 sends out.

Response Command Name	Function
ACK (acknowledge)	Acknowledges command/data, etc.
NAK (not acknowledge)	Acknowledges illegal command/data, etc.

[MEMO]

APPENDIX A REGISTER INDEX

(1/8)

(2/8)

Register Symbol	Register Name	Unit	Page
CC100	Capture/compare register 100	RPU	252
CC101	Capture/compare register 101	RPU	252
CC102	Capture/compare register 102	RPU	252
CC103	Capture/compare register 103	RPU	252
CC110	Capture/compare register 110	RPU	252
CC111	Capture/compare register 111	RPU	252
CC112	Capture/compare register 112	RPU	252
CC113	Capture/compare register 113	RPU	252
CC120	Capture/compare register 120	RPU	252
CC121	Capture/compare register 121	RPU	252
CC122	Capture/compare register 122	RPU	252
CC123	Capture/compare register 123	RPU	252
CC130	Capture/compare register 130	RPU	252
CC131	Capture/compare register 131	RPU	252
CC132	Capture/compare register 132	RPU	252
CC133	Capture/compare register 133	RPU	252
CC140	Capture/compare register 140	RPU	252
CC141	Capture/compare register 141	RPU	252
CC142	Capture/compare register 142	RPU	252
CC143	Capture/compare register 143	RPU	252
CC150	Capture/compare register 150	RPU	252
CC151	Capture/compare register 151	RPU	252
CC152	Capture/compare register 152	RPU	252
CC153	Capture/compare register 153	RPU	252
CKC	Clock control register	CG	233
CM40	Compare register 40	RPU	253
CM41	Compare register 41	RPU	253
CMIC40	Interrupt control register	INTC	217
CMIC41	Interrupt control register	INTC	217
CSIC0	Interrupt control register	INTC	217
CSIC1	Interrupt control register	INTC	217
CSIC2	Interrupt control register	INTC	217
CSIC3	Interrupt control register	INTC	217
CSIM0	Clocked serial interface mode register 0	CSI0	301
CSIM1	Clocked serial interface mode register 1	CSI1	301
CSIM2	Clocked serial interface mode register 2	CSI2	301
CSIM3	Clocked serial interface mode register 3	CSI3	301

(3/8)

Register Symbol	Register Name	Unit	Page
CTBP	CALLT base pointer	CPU	72
CTPC	Status saving register during CALLT execution	CPU	72
CTPSW	Status saving register during CALLT execution	CPU	72
DADC0	DMA addressing control register 0	DMAC	168
DADC1	DMA addressing control register 1	DMAC	168
DADC2	DMA addressing control register 2	DMAC	168
DADC3	DMA addressing control register 3	DMAC	168
DBC0	DMA byte count register 0	DMAC	167
DBC1	DMA byte count register 1	DMAC	167
DBC2	DMA byte count register 2	DMAC	167
DBC3	DMA byte count register 3	DMAC	167
DBPC	Status saving register during exception trap	CPU	72
DBPSW	Status saving register during exception trap	CPU	72
DCHC0	DMA channel control register 0	DMAC	170
DCHC1	DMA channel control register 1	DMAC	170
DCHC2	DMA channel control register 2	DMAC	170
DCHC3	DMA channel control register 3	DMAC	170
DDA0H	DMA destination address register 0H	DMAC	165
DDA0L	DMA destination address register 0L	DMAC	166
DDA1H	DMA destination address register 1H	DMAC	165
DDA1L	DMA destination address register 1L	DMAC	166
DDA2H	DMA destination address register 2H	DMAC	165
DDA2L	DMA destination address register 2L	DMAC	166
DDA3H	DMA destination address register 3H	DMAC	165
DDA3L	DMA destination address register 3L	DMAC	166
DDIS	DMA disable status register	BCU	173
DMAIC0	Interrupt control register	INTC	217
DMAIC1	Interrupt control register	INTC	217
DMAIC2	Interrupt control register	INTC	217
DMAIC3	Interrupt control register	INTC	217
DRC0	DRAM configuration register 0	BCU	139
DRC1	DRAM configuration register 1	BCU	139
DRC2	DRAM configuration register 2	BCU	139
DRC3	DRAM configuration register 3	BCU	139
DRST	DMA restart register	BCU	173
DSA0H	DMA source address register 0H	DMAC	163
DSA0L	DMA source address register 0L	DMAC	164

(4/8)

Register Symbol	Register Name	Unit	Page
DSA1H	DMA source address register 1H	DMAC	163
DSA1L	DMA source address register 1L	DMAC	164
DSA2H	DMA source address register 2H	DMAC	163
DSA2L	DMA source address register 2L	DMAC	164
DSA3H	DMA source address register 3H	DMAC	163
DSA3L	DMA source address register 3L	DMAC	164
DTC	DRAM type configuration register	BCU	142
DTFR0	DMA trigger factor register 0	DMAC	171
DTFR1	DMA trigger factor register 1	DMAC	171
DTFR2	DMA trigger factor register 2	DMAC	171
DTFR3	DMA trigger factor register 3	DMAC	171
DWC1	Data wait control register 1	BCU	113
DWC2	Data wait control register 2	BCU	113
ECR	Interrupt source register	CPU	72
EIPC	Status saving register during interrupt	CPU	72
EIPSW	Status saving register during interrupt	CPU	72
FDW	Flyby transfer data wait control register	BCU	174
FEPC	Status saving register during NMI	CPU	72
FEPSW	Status saving register during NMI	CPU	72
INTM0	External interrupt mode register 0	INTC	208
INTM1	External interrupt mode register 1	INTC	221
INTM2	External interrupt mode register 2	INTC	221
INTM3	External interrupt mode register 3	INTC	221
INTM4	External interrupt mode register 4	INTC	221
INTM5	External interrupt mode register 5	INTC	221
INTM6	External interrupt mode register 6	INTC	221
ISPR	In-service priority register	INTC	218
MM	Memory expansion mode register	Port	87
OVIC10	Interrupt control register	INTC	216
OVIC11	Interrupt control register	INTC	216
OVIC12	Interrupt control register	INTC	216
OVIC13	Interrupt control register	INTC	217
OVIC14	Interrupt control register	INTC	217
OVIC15	Interrupt control register	INTC	217
P0	Port 0	Port	368
P1	Port 1	Port	371
P2	Port 2	Port	374

(5/8)

Register Symbol	Register Name	Unit	Page
P3	Port 3	Port	377
P4	Port 4	Port	380
P5	Port 5	Port	382
P6	Port 6	Port	384
P7	Port 7	Port	386
P8	Port 8	Port	387
P9	Port 9	Port	391
P10	Port 10	Port	394
P10IC0	Interrupt control register	INTC	217
P10IC1	Interrupt control register	INTC	217
P10IC2	Interrupt control register	INTC	217
P10IC3	Interrupt control register	INTC	217
P11	Port 11	Port	397
P11IC0	Interrupt control register	INTC	217
P11IC1	Interrupt control register	INTC	217
P11IC2	Interrupt control register	INTC	217
P11IC3	Interrupt control register	INTC	217
P12	Port 12	Port	401
P12IC0	Interrupt control register	INTC	217
P12IC1	Interrupt control register	INTC	217
P12IC2	Interrupt control register	INTC	217
P12IC3	Interrupt control register	INTC	217
P13IC0	Interrupt control register	INTC	217
P13IC1	Interrupt control register	INTC	217
P13IC2	Interrupt control register	INTC	217
P13IC3	Interrupt control register	INTC	217
P14IC0	Interrupt control register	INTC	217
P14IC1	Interrupt control register	INTC	217
P14IC2	Interrupt control register	INTC	217
P14IC3	Interrupt control register	INTC	217
P15IC0	Interrupt control register	INTC	217
P15IC1	Interrupt control register	INTC	217
P15IC2	Interrupt control register	INTC	217
P15IC3	Interrupt control register	INTC	217
PA	Port A	Port	403
PB	Port B	Port	405
PC	Program counter	CPU	71

(6/8)

Register Symbol	Register Name	Unit	Page
PCS0	Port/control select register 0	Port	370
PCS1	Port/control select register 1	Port	373
PCS3	Port/control select register 3	Port	380
PCS8	Port/control select register 8	Port	390
PCS10	Port/control select register 10	Port	396
PCS11	Port/control select register 11	Port	400
PM0	Port 0 mode register	Port	368
PM1	Port 1 mode register	Port	371
PM2	Port 2 mode register	Port	375
PM3	Port 3 mode register	Port	378
PM4	Port 4 mode register	Port	381
PM5	Port 5 mode register	Port	383
PM6	Port 6 mode register	Port	385
PM8	Port 8 mode register	Port	388
PM9	Port 9 mode register	Port	392
PM10	Port 10 mode register	Port	394
PM11	Port 11 mode register	Port	398
PM12	Port 12 mode register	Port	401
PMA	Port A mode register	Port	403
PMB	Port B mode register	Port	405
PMC0	Port 0 mode control register	Port	369
PMC1	Port 1 mode control register	Port	372
PMC2	Port 2 mode control register	Port	376
PMC3	Port 3 mode control register	Port	379
PMC8	Port 8 mode control register	Port	389
PMC9	Port 9 mode control register	Port	393
PMC10	Port 10 mode control register	Port	395
PMC11	Port 11 mode control register	Port	399
PMC12	Port 12 mode control register	Port	402
PMCX	Port X mode control register	Port	408
PMX	Port X mode register	Port	407
PRC	Page ROM configuration register	BCU	134
PRCMD	Command register	CPU	101
PSC	Power save control register	CPU	237
PSW	Program status word	CPU	73
PX	Port X	Port	407
r0 to r31	General register	CPU	71

(7/8)

Register Symbol	Register Name	Unit	Page
RFC0	Refresh control register 0	BCU	153
RFC1	Refresh control register 1	BCU	153
RFC2	Refresh control register 2	BCU	153
RFC3	Refresh control register 3	BCU	153
RWC	Refresh wait control register	BCU	156
RXB0	Receive buffer 0 (9 bits)	UART0	292
RXB0L	Receive buffer 0L (Lower order 8 bits)	UART0	292
RXB1	Receive buffer 1 (9 bits)	UART1	292
RXB1L	Receive buffer 1L (Lower order 8 bits)	UART1	292
SEIC0	Interrupt control register	INTC	217
SEIC1	Interrupt control register	INTC	217
SIO0	Serial I/O shift register 0	CSI0	303
SIO1	Serial I/O shift register 1	CSI1	303
SIO2	Serial I/O shift register 2	CSI2	303
SIO3	Serial I/O shift register 3	CSI3	303
SRIC0	Interrupt control register	INTC	217
SRIC1	Interrupt control register	INTC	217
STIC0	Interrupt control register	INTC	217
STIC1	Interrupt control register	INTC	217
SYS	System status register	CPU	102
TM10	Timer 10	RPU	251
TM11	Timer 11	RPU	251
TM12	Timer 12	RPU	251
TM13	Timer 13	RPU	251
TM14	Timer 14	RPU	251
TM15	Timer 15	RPU	251
TM40	Timer 40	RPU	253
TM41	Timer 41	RPU	253
TMC10	Timer control register 10	RPU	257
TMC11	Timer control register 11	RPU	257
TMC12	Timer control register 12	RPU	257
TMC13	Timer control register 13	RPU	257
TMC14	Timer control register 14	RPU	257
TMC15	Timer control register 15	RPU	257
TMC40	Timer control register 40	RPU	259
TMC41	Timer control register 41	RPU	259
TOC10	Timer output control register 10	RPU	260

(8/8)

Register Symbol	Register Name	Unit	Page
TOC11	Timer output control register 11	RPU	260
TOC12	Timer output control register 12	RPU	260
TOC13	Timer output control register 13	RPU	260
TOC14	Timer output control register 14	RPU	260
TOC15	Timer output control register 15	RPU	260
TOVS	Timer overflow status register	RPU	261
TUM10	Timer unit mode register 10	RPU	254
TUM11	Timer unit mode register 11	RPU	254
TUM12	Timer unit mode register 12	RPU	254
TUM13	Timer unit mode register 13	RPU	254
TUM14	Timer unit mode register 14	RPU	254
TUM15	Timer unit mode register 15	RPU	254
TXS0	Transmit shift register 0 (9 bits)	UART0	293
TXS0L	Transmit shift register 0L (Lower order 8 bits)	UART0	293
TXS1	Transmit shift register 1 (9 bits)	UART1	293
TXS1L	Transmit shift register 1L (Lower order 8 bits)	UART1	293

APPENDIX B INSTRUCTION SET LIST

B.1 General Examples

(1) Register symbols used to describe operands

Register Symbol	Explanation
reg1	General registers (r0 to r31): Used as source registers.
reg2	General registers (r0 to r31): Used mainly as destination registers.
reg3	General registers (r0 to r31): Used mainly to store the remainders of division results and the higher order 3 bits of multiplication results.
immX	X bit immediate
dispX	X bit displacement
regID	System register number
bit#3	3-bit data for specifying the bit number
ep	Element pointer (r30)
cccc	4-bit data which show the conditions code
vector	5-bit data which specify the trap vector (00H to 1FH)
listX	X item register list

(2) Register symbols used to describe op codes

Register Symbol	Explanation
R	1-bit data of a code which specifies reg1 or regID
r	1-bit data of the code which specifies reg2
w	1-bit data of the code which specifies reg3
d	1-bit displacement data
i	1-bit immediate data
cccc	4-bit data which show the conditions code
bbb	3-bit data for specifying the bit number
L	1-bit data which specifies a register list

(3) Register symbols used in operation (1/2)

Register Symbol	Explanation
←	Input for
GR [ ]	General register
SR [ ]	System register
zero-extend (n)	Expand n with zeros until word length.
sign-extend (n)	Expand n with signs until word length.
load-memory (a, b)	Read data from address a until size b.
store-memory (a, b, c)	Write data b in address a to size c.
load-memory-bit (a, b)	Read bit b of address a.

(3) Register symbols used in operation (2/2)

Register Symbol	Explanation
store-memory-bit (a, b, c)	Write bit b of address a to c.
saturated (n)	Execute saturated processing of n (n is a 2’s complement). If, as a result of calculations, n ≥ 7FFFFFFFH, let it be 7FFFFFFFH. n ≤ 80000000H, let it be 80000000H.
result	Reflects the results in a flag.
Byte	Byte (8 bits)
Half-word	Half word (16 bits)
Word	Word (32 bits)
+	Addition
–	Subtraction
ll	Bit concatenation
×	Multiplication
÷	Division
%	Remainder from division results
AND	Logical product
OR	Logical sum
XOR	Exclusive OR
NOT	Logical negation
logically shift left by	Logical shift left
logically shift right by	Logical shift right
arithmetically shift right by	Arithmetic shift right

(4) Register symbols used in an execution clock

Register Symbol	Explanation
i : issue	If executing another instruction immediately after executing the first instruction.
r : repeat	If repeating execution of the same instruction immediately after executing the first instruction.
l : latency	If referring to the results of instruction execution immediately after execution using another instruction.

(5) Register symbols used in flag operations

Identifier	Explanation
(Blank)	No change
0	Clear to 0
X	Set or cleared in accordance with the results.
R	Previously saved values are restored.

(6) Condition codes

Condition Name (cond)	Condition Code (cccc)	Condition Formula	Explanation
V	0 0 0 0	OV = 1	Overflow
NV	1 0 0 0	OV = 0	No overflow
C/L	0 0 0 1	CY = 1	Carry Lower (Less than)
NC/NL	1 0 0 1	CY = 0	No carry Not lower (Greater than or equal)
Z/E	0 0 1 0	Z = 1	Zero Equal
NZ/NE	1 0 1 0	Z = 0	Not zero Not equal
NH	0 0 1 1	(CY or Z) = 1	Not higher (Less than or equal)
H	1 0 1 1	(CY or Z) = 0	Higher (Greater than)
N	0 1 0 0	S = 1	Negative
P	1 1 0 0	S = 0	Positive
T	0 1 0 1	—	Always (Unconditional)
SA	1 1 0 1	SAT = 1	Saturated
LT	0 1 1 0	(S xor OV) = 1	Less than signed
GE	1 1 1 0	(S xor OV) = 0	Greater than or equal signed
LE	0 1 1 1	((S xor OV) or Z) = 1	Less than or equal signed
GT	1 1 1 1	((S xor OV) or Z) = 0	Greater than signed

B.2 Instruction Set (in Alphabetical Order)

(1/6)

Mnemonic	Operand	Op Code	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
ADD	reg1,reg2	rrrrr001110RRRRR	GR[reg2]←GR[reg2]+GR[reg1]	1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010010iiiii	GR[reg2]←GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	
ADDI	imm16,reg1,reg2	rrrrr110000RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1	×	×	×	×	
AND	reg1,reg2	rrrrr001010RRRRR	GR[reg2]←GR[reg2]AND GR[reg1]	1	1	1		0	×	×	
ANDI	imm16,reg1,reg2	rrrrr110110RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]AND zero-extend(imm16)	1	1	1		0	0	×	
Bcond	disp9	ddddd1011dddcccc **Note 1**	if conditions are satisfied then PC←PC+sign-extend(disp9) When conditions are satisfied	2 **Note 2**	2 **Note 2**	2 **Note 2**					
			When conditions are not satisfied	1	1	1					
BSH	reg2,reg3	rrrrr11111100000 wwwww01101000010	GR[reg3]←GR[reg2] (23 : 16) ll GR[reg2] (31 : 24) ll GR[reg2] (7 : 0) ll GR[reg2] (15 : 8)	1	1	1	×	0	×	×	
BSW	reg2,reg3	rrrrr11111100000 wwwww01101000000	GR[reg3]←GR[reg2] (7 : 0) ll GR[reg2] (15 : 8) ll GR [reg2] (23 : 16) ll GR[reg2] (31 : 24)	1	1	1	×	0	×	×	
CALLT	imm6	0000001000iiiiii	CTPC←PC+2(return PC) CTPSW←PSW adr←CTBP+zero-extend(imm6 logically shift left by 1) PC←CTBP+zero-extend(Load-memory(adr,Half-word))	4	4	4					
CLR1	bit#3, disp16[reg1]	10bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,0)	3 **Note 3**	3 **Note 3**	3 **Note 3**				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100100	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,0)	3 **Note 3**	3 **Note 3**	3 **Note 3**				×	
CMOV	cccc,imm5,reg2,reg3	rrrrr111111iiiii wwwww011000cccc0	if conditions are satisfied then GR[reg3]←sign-extended(imm5) else GR[reg3]←GR[reg2]	1	1	1					
	cccc,reg1,reg2,reg3	rrrrr111111RRRRR wwwww011001cccc0	if conditions are satisfied then GR[reg3]←GR[reg1] else GR[reg3]←GR[reg2]	1	1	1					
CMP	reg1,reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
	imm5,reg2	rrrrr010011iiiii	result←GR[reg2]–sign-extend(imm5)	1	1	1	×	×	×	×	
CTRET		0000011111100000 0000000101000100	PC←CTPC PSW←CTPSW	3	3	3	R	R	R	R	R
DI		0000011111100000 0000000101100000	PSW.ID←1	1	1	1					

(2/6)

Mnemonic	Operand	Op Code	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
DISPOSE	imm5,list12	0000011001iiiiiL LLLLLLLLLLL00000	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded	N+1 Note 4	N+1 Note 4	N+1 Note 4					
	imm5,list12,[reg1]	0000011001iiiiiL LLLLLLLLLLLRRRRR **Note 5**	sp←sp+zero-extend(imm5 logically shift left by 2) GR[reg in list12]←Load-memory(sp,Word) sp←sp+4 repeat 2 steps above until all regs in list12 is loaded PC←GR[reg1]	N+3 Note 4	N+3 Note 4	N+3 Note 4					
DIV	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000000	GR[reg2]←GR[reg2}÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVH	reg1,reg2	rrrrr000010RRRRR	GR[reg2]←GR[reg2]÷GR[reg1] Note 6	35	35	35		×	×	×	
	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000000	GR[reg2]←GR[reg2]÷GR[reg1] Note 6 GR[reg3]←GR[reg2]%GR[reg1]	35	35	35		×	×	×	
DIVHU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01010000010	GR[reg2]←GR[reg2]÷GR[reg1] Note 6 GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
DIVU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01011000010	GR[reg2]←GR[reg2]÷GR[reg1] GR[reg3]←GR[reg2]%GR[reg1]	34	34	34		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0	1	1	1					
HALT		0000011111100000 0000000100100000	Stop	1	1	1					
HSW	reg2,reg3	rrrrr11111100000 wwwww01101000100	GR[reg3]←GR[reg2](15 : 0) ll GR[reg2] (31 : 16)	1	1	1	×	0	×	×	
JARL	disp22,reg2	rrrrr11110dddddd ddddddddddddddd0 **Note 7**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)	2	2	2					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]	3	3	3					
JR	disp22	0000011110dddddd ddddddddddddddd0 **Note 7**	PC←PC+sign-extend(disp22)	2	2	2					
LD.B	disp16[reg1],reg2	rrrrr111000RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	n Note 9					
LD.BU	disp16[reg1],reg2	rrrrr11110bRRRRR ddddddddddddddd1 **Notes 8, 10**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	n Note 11					
LD.H	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr,Half-word))	1	1	n Note 9					

(3/6)

Mnemonic	Operand	Op Code	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
LD.HU	disp16[reg1],reg2	rrrrr111111RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←zero-extend(Load-memory(adr,Half-word))	1	1	n Note11					
LD.W	disp16[reg1],reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-exend(disp16) GR[reg2]←Load-memory(adr,Word)	1	1	n Note 9					
LDSR	reg2,regID	rrrrr111111RRRRR 0000000000100000 **Note 12**	SR[regID]←GR[reg2] Other than regID=PSW	1	1	1					
			regID=PSW				×	×	×	×	×
MOV	reg1,reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]	1	1	1					
	imm5,reg2	rrrrr010000iiiii	GR[reg2]←sign-extend(imm5)	1	1	1					
	imm32,reg1	00000110001RRRRR iiiiiiiiiiiiiiii iiiiiiiiiiiiiiii	GR[reg1]←imm32	2	2	2					
MOVEA	imm16,reg1,reg2	rrrrr110001RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+sign-extend(imm16)	1	1	1					
MOVHI	imm16,reg1,reg2	rrrrr110010RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]+(imm16 ll 0^{16})	1	1	1					
MUL	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100000	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]	1	2 Note14	2					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII00	GR[reg3] ll GR[reg2]←GR[reg2]xsign-extend(imm9) **Note 13**	1	2 Note14	2					
MULH	reg1,reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2]$^{\text{Note 6}}$xGR[reg1]$^{\text{Note 6}}$	1	1	2					
	imm5,reg2	rrrrr010111iiiii	GR[reg2]←GR[reg2]$^{\text{Note 6}}$xsign-extend(imm5)	1	1	2					
MULHI	imm16,reg1,reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]$^{\text{Note 6}}$ximm16	1	1	2					
MULU	reg1,reg2,reg3	rrrrr111111RRRRR wwwww01000100010	GR[reg3] ll GR[reg2]←GR[reg2]xGR[reg1]	1	2 Note 14	2					
	imm9,reg2,reg3	rrrrr111111iiiii wwwww01001IIII10	GR[reg3] ll GR[reg2]←GR[reg2]xzero-extend(imm9) **Note 13**	1	2 Note 14	2					
NOP		0000000000000000	Pass at least one clock cycle doing nothing.	1	1	1					
NOT	reg1,reg2	rrrrr000001RRRRR	GR[reg2]←NOT(GR[reg1])	1	1	1		0	×	×	
NOT1	bit#3,disp16[reg1]	01bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,Z flag)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100010	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,Z flag)	3 Note 3	3 Note 3	3 Note 3				×	
OR	reg1,reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]	1	1	1		0	×	×	

(4/6)

Mnemonic	Operand	Op Code	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
ORI	imm16,reg1,reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)	1	1	1		0	×	×	
PREPARE	list12,imm5	00000111110iiiiiL LLLLLLLLLLL00001	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5)	N+1 Note 4	N+1 Note 4	N+1 Note 4					
	list12,imm5, sp/imm[Note 15]	00000111110iiiiiL LLLLLLLLLLLff011 imm16/imm32 Note 16	Store-memory(sp–4,GR[reg in list12],Word) sp←sp–4 repeat 1 step above until all regs in list12 is stored sp←sp-zero-extend(imm5) ep←sp/imm	N+2 Note 4 Note 17	N+2 Note 4 Note 17	N+2 Note 4 Note 17					
RETI		0000011111100000 0000000101000000	if PSW.EP=1 then PC ←EIPC PSW ←EIPSW else if PSW.NP=1 then PC ←FEPC PSW ←FEPSW else PC ←EIPC PSW ←EIPSW	3	3	3	R	R	R	R	R
SAR	reg1,reg2	rrrrr111111RRRRR 0000000010100000	GR[reg2]←GR[reg2]arithmetically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010101iiiii	GR[reg2]←GR[reg2]arithmetically shift right by zero-extend (imm5)	1	1	1	×	0	×	×	
SASF	cccc,reg2	rrrrr1111110cccc 0000001000000000	if conditions are satisfied then GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000001H else GR[reg2]←(GR[reg2]Logically shift left by 1) OR 00000000H	1	1	1					
SATADD	reg1,reg2	rrrrr000110RRRRR	GR[reg2]←saturated(GR[reg2]+GR[reg1])	1	1	1	×	×	×	×	×
	imm5,reg2	rrrrr010001iiiii	GR[reg2]←saturated(GR[reg2]+sign-extend(imm5)	1	1	1	×	×	×	×	×
SATSUB	reg1,reg2	rrrrr000101RRRRR	GR[reg2]←saturated(GR[reg2]–GR[reg1])	1	1	1	×	×	×	×	×
SATSUBI	imm16,reg1,reg2	rrrrr110011RRRRR iiiiiiiiiiiiiiii	GR[reg2]←saturated(GR[reg1]–sign-extend(imm16)	1	1	1	×	×	×	×	×
SATSUBR	reg1,reg2	rrrrr000100RRRRR	GR[reg2]←saturated(GR[reg1]–GR[reg2])	1	1	1	×	×	×	×	×
SETF	cccc,reg2	rrrrr1111110cccc 0000000000000000	If conditions are satisfied then GR[reg2]←00000001H else GR[reg2]←00000000H	1	1	1					

(5/6)

Mnemonic	Operand	Op Code	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SET1	bit#3,disp16[reg1]	00bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit(adr,bit#3)) Store-memory-bit(adr,bit#3,1)	3 Note 3	3 Note 3	3 Note 3				×	
	reg2,[reg1]	rrrrr111111RRRRR 0000000011100000	adr←GR[reg1] Z flag←Not(Load-memory-bit(adr,reg2)) Store-memory-bit(adr,reg2,1)	3 Note 3	3 Note 3	3 Note 3				×	
SHL	reg1,reg2	rrrrr111111RRRRR 0000000011000000	GR[reg2]←GR[reg2] logically shift left by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010110iiiii	GR[reg2]←GR[reg2] logically shift left by zero-extend(imm5)	1	1	1	×	0	×	×	
SHR	reg1,reg2	rrrrr111111RRRRR 0000000010000000	GR[reg2]←GR[reg2] logically shift right by GR[reg1]	1	1	1	×	0	×	×	
	imm5,reg2	rrrrr010100iiiii	GR[reg2]←GR[reg2] logically shift right by zero-extend(imm5)	1	1	1	×	0	×	×	
SLD.B	disp7[ep],reg2	rrrrr0110ddddddd	adr←ep+zero-extend(disp7) GR[reg2]←sign-extend(Load-memory(adr,Byte))	1	1	n Note 9					
SLD.BU	disp4[ep],reg2 **Note 18**	rrrrr0000110dddd	adr←ep+zero-extend(disp4) GR[reg2]←zero-extend(Load-memory(adr,Byte))	1	1	n Note 9					
SLD.H	disp8[ep],reg2	rrrrr1000ddddddd **Note 19**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr,Half-word))	1	1	n Note 9					
SLD.HU	disp5[ep],reg2 **Notes 18, 20**	rrrrr0000111dddd	adr←ep+zero-extend(disp5) GR[reg2]←zero-extend(Load-memory(adr,Half-word))	1	1	n Note 9					
SLD.W	disp8[ep],reg2	rrrrr1010dddddd0 **Note 21**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr,Word)	1	1	n Note 9					
SST.B	reg2,disp7[ep]	rrrrr0111ddddddd	adr←ep+zero-extend(disp7) Store-memory(adr,GR[reg2],Byte)	1	1	1					
SST.H	reg2,disp8[ep]	rrrrr1001ddddddd **Note 19**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Half-word)	1	1	1					
SST.W	reg2,disp8[ep]	rrrrr1010dddddd1 **Note 21**	adr←ep+zero-extend(disp8) Store-memory(adr,GR[reg2],Word)	1	1	1					
ST.B	reg2,disp16[reg1]	rrrrr111010RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr,GR[reg2],Byte)	1	1	1					
ST.H	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd0 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Half-word)	1	1	1					
ST.W	reg2,disp16[reg1]	rrrrr111011RRRRR ddddddddddddddd1 **Note 8**	adr←GR[reg1]+sign-extend(disp16) Store-memory (adr,GR[reg2], Word)	1	1	1					
STSR	regID,reg2	rrrrr111111RRRRR 0000000001000000	GR[reg2]←SR[regID]	1	1	1					

(6/6)

Mnemonic	Operand	Op Code	Operation	Execution Clock			Flags				
				i	r	l	CY	OV	S	Z	SAT
SUB	reg1,reg2	rrrrr001101RRRRR	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1,reg2	rrrrr001100RRRRR	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
SWITCH	reg1	00000000010RRRRR	adr←(PC+2) + (GR [reg1] logically shift left by 1) PC←(PC+2) + (sign-extend (Load-memory (adr,Half-word))) logically shift left by 1	5	5	5					
SXB	reg1	00000000101RRRRR	GR[reg1]←sign-extend (GR[reg1] (7 : 0))	1	1	1					
SXH	reg1	00000000111RRRRR	GR[reg1]←sign-extend (GR[reg1] (15 : 0))	1	1	1					
TRAP	vector	00000111111iiiii 0000000100000000	EIPC ←PC+4 (Return PC) EIPSW ←PSW ECR.EICC ←Interrupt Code PSW.EP ←1 PSW.ID ←1 PC ←00000040H (when vector is 00H to 0FH) 00000050H (when vector is 10H to 1FH)	3	3	3					
TST	reg1,reg2	rrrrr001011RRRRR	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3,disp16[reg1]	11bbb111110RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) Z flag←Not (Load-memory-bit (adr,bit#3))	3 Note 3	3 Note 3	3 Note 3				×	
	reg2, [reg1]	rrrrr111111RRRRR 0000000011100110	adr←GR[reg1] Z flag←Not (Load-memory-bit (adr,reg2))	3 Note 3	3 Note 3	3 Note 3				×	
XOR	reg1,reg2	rrrrr001001RRRRR	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16,reg1,reg2	rrrrr110101RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1] XOR zero-extend (imm16)	1	1	1		0	×	×	
ZXB	reg1	00000000100RRRRR	GR[reg1]←zero-extend (GR[reg1] (7 : 0))	1	1	1					
ZXH	reg1	00000000110RRRRR	GR[reg1]←zero-extend (GR[reg1] (15 : 0))	1	1	1					

Notes

1. dddddddd: Higher 8 bits of disp9.
2. 3 clocks if the final instruction includes PSW write access.
3. If there is no wait state (3 + the number of read access wait states).
4. N is the total number of list 12 read registers. (According to the number of wait states. Also, if there are no wait states, N is the number of list 12 registers.)
5. RRRRR: other than 00000.
6. The lower halfword data only are valid.
7. ddddddddddddddddddddd: The higher 21 bits of disp22.
8. ddddddddddddddd: The higher 15 bits of disp16.
9. According to the number of wait states (1 if there are no wait states).
10. b: bit 0 of disp16.
11. According to the number of wait states (2 if there are no wait states).

Notes 12. In this instruction, for convenience of mnemonic description, the source register is made reg2, but the reg1 field is used in the op code. Therefore, the meaning of register specification in the mnemonic description and in the op code differs from other instructions.
r r r r r = regID specification
RRRRR = reg2 specification

13. i i i i i: Lower 5 bits of imm9.
I I I I: Lower 4 bits of imm9.

14. In the case of r = w (the lower 32 bits of the results are not written in the register) or w = r0 (the higher 32 bits of the results are not written in the register), 1.

15. sp/imm: specified by bits 19 and 20 of the sub op code.

16. ff = 00: Load sp in ep.
01: Load sign expanded 16-bit immediate data (bits 47 to 32) in ep.
10: Load 16-bit logically left shifted 16-bit immediate data (bits 47 to 32) in ep.
11: Load 32-bit immediate data (bits 63 to 32) in ep.

17. If imm = imm32, N + 3 blocks.

18. r r r r r : Other than 00000.

19. ddddddd: Higher 7 bits of disp8.

20. dddd: Higher 4 bits of disp5.

21. dddddd: Higher 6 bits of disp8.

APPENDIX C INDEX

NEC

Facsimile Message

From:

____
Name

____
Company

____
Tel. FAX

____
Address

Although NEC has taken all possible steps to ensure that the documentation supplied to our customers is complete, bug free and up-to-date, we readily accept that errors may occur. Despite all the care and precautions we've taken, you may encounter problems in the documentation. Please complete this form whenever you'd like to report errors or suggest improvements to us.

Thank you for your kind support.

North America
NEC Electronics Inc.
Corporate Communications Dept.
Fax: 1-800-729-9288
1-408-588-6130

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

South America
NEC do Brasil S.A.
Fax: +55-11-6465-6829

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Korea
NEC Electronics Hong Kong Ltd.
Seoul Branch
Fax: 02-528-4411

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-2719-5951

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Japan
NEC Semiconductor Technical Hotline
Fax: 044-548-7900

I would like to report the following error/make the following suggestion:

Document title: ______________________________

Document number: ____________________ Page number: __________

____

____

____

If possible, please fax the referenced page or drawing.

Document Rating	Excellent	Good	Acceptable	Poor
Clarity	❏	❏	❏	❏
Technical Accuracy	❏	❏	❏	❏
Organization	❏	❏	❏	❏