USER'S MANUAL

NEC

V851™

32/16-BIT SINGLE-CHIP MICROCOMPUTER

HARDWARE

μPD703000
μPD703001
μPD70P3000

Document No. U10935EJ2V0UM00 (2nd edition)
Date Published September 1996 P
Printed in Japan

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.

The customer must judge the need for license:	μPD703000GC-xx-xxx-7EA
	μPD70P3000GC-xx-7EA
License not needed	: μPD703001GC-xx-7EA

The information in this document is subject to change without notice.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.

NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.

While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.

NEC devices are classified into the following three quality grades:

"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

- Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots
- Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)
- Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.

Anti-radioactive design is not implemented in this product.

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.)
Mountain View, California
Tel: 800-366-9782
Fax: 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.1.
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.
France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.
Spain Office
Madrid, Spain
Tel: 01-504-2787
Fax: 01-504-2860

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby Sweden
Tel: 8-63 80 820
Fax: 8-63 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: 253-8311
Fax: 250-3583

NEC Electronics Taiwan Ltd.
Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

NEC do Brasil S.A.
Sao Paulo-SP, Brasil
Tel: 011-889-1680
Fax: 011-889-1689

Major Revisions in This Edition

Page	Description
Throughout	Addition of description of 33-MHz operation
p.3	Addition of parts numbers to **1.4 Ordering Information**
p.8	Addition of description to **1.6.2 (2) Bus control unit (BCU)**
p.15	Partial change of **2.2 Pin Status**
p.20	Addition of description to **2.3.1 (8) (b) (vi) ST0, ST1**
p.23	Partial change of **2.4 Processing of Unused Pins**
p.26	Addition of text pointer to **3.2 CPU Register Set**
p.28	Addition of description to **Table 3-2 System Register Numbers**
p.46	Change of **Figure 3-10 Recommended Memory Map**
p.50	Change of **4.3 Number of Access Clocks**
p.70, 75	Addition of Note to **Figures 5-3** and **5-6 RETI Instruction Processing**
p.83, 86	Modification of Note in **Figures 5-10** and **5-12 RETI Instruction Processing**
p.88	Modification of **5.6.2 (2) To generate exception in service program**
p.96	Addition of description to **6.5.2 (1) Power save control register (PSC)**
p.104	Addition of Note to **6.5.5 (2) Releasing software STOP mode**
p.112	Addition of description and note to **7.2.1 Timer 1**
p.114	Addition of Note to **7.2.2 Timer 4**
p.134	Change of **Figure 7-17 PWM Output Timing (TM1)**
p.200	Modification of description of **9.4 Input Noise Filters**
p.212	Addition of **11.6 Notes on Releasing STOP Mode When External Clock is Used**

The mark ★ shows major revised points.

INTRODUCTION

Readers

This manual is intended for the users who wish to understand the functions of the V851 (μPD703000, 703001 and 70P3000) to design application systems using the V851.

Purpose

This manual presents information on the hardware functions of the V851.

Organization

Two volumes of the V851 User's Manual are available: hardware (this manual) and architecture (V850 Family™ User's Manual - Architecture) manuals. The organization of each manual is as follows:

Hardware	Architecture
• Pin function	• Data type
• CPU function	• Register set
• Internal peripheral function	• Instruction format and instruction set
• PROM mode	• Interrupt and exception
	• Pipeline operation

How to Read This Manual

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

- To find the details of a register where the name is known
  → Refer to **APPENDIX A REGISTER INDEX**.
- To understand the details of an instruction function
  → Refer to the **V850 Family User's Manual - Architecture**.
- To find out about the electrical characteristics of the V851
  → Refer to the **Data Sheet** separately available.
- To understand the overall functions of the V851
  → Read this manual according to the Table of Contents.

In this manual, the one-time PROM model is referred to as a PROM.

Legend

Data significance	: Left: higher digit, right: lower digit
Active low	: $\overline{xxx}$ (top bar over pin or signal name)
Memory map address	: Top: highest, bottom: lowest
Note	: Foot note
Caution	: Important information
Remark	: Supplement
Numeric representation	: Binary ... xxxx or xxxxB
	Decimal ... xxxx
	Hexadecimal ... xxxxH
Prefix indicating power of 2	: K (kilo): $2^{10} = 1024$
	M (mega): $2^{20} = 1024^2$

Related documents

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

Document Name \ Product Name		μPD703000, 703001	μPD70P3000
Data sheet		U10987E	U10988E
User's manual	Hardware	This manual	
	Architecture	U10243E	
Instruction list		U10229E	
Register list		—	

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF FIGURES

LIST OF TABLES

[MEMO]

CHAPTER 1 INTRODUCTION

The V851 is the first product of the NEC's V850 family single-chip microcontrollers for real-time control applications. This chapter briefly outlines the V851.

1.1 General

The V851 is a 32-/16-bit single-chip microcontroller that employs the CPU core of the V850 family of high-performance 32-bit single-chip microcontrollers for real-time control applications, and integrates peripheral functions such as ROM/RAM, real-time pulse unit, and serial interface.

The V851 is provided with multiplication instructions that are executed with a hardware multiplier, saturated operation instructions, and bit manipulation instructions that are ideal for digital servo control applications, in addition to the basic instructions that have a high real-time response speed and can be executed in 1 clock cycle. This microcontroller can be employed for many applications including real-time control systems such as engine control and ABS (Anti-lock Braking System); various automobile electronic systems; office machines including PPCs (Plain Paper Copiers), printers, and facsimiles; and factory automation systems such as NC (Numerical Control) machine tools and various controllers. In any of these applications, the V851 demonstrates an extremely high cost effectiveness.

1.2 Features

- ○ Number of instructions : 74
- ★ ○ Minimum instruction execution time : 30 ns (at 33 MHz)
- ○ General register : 32 bits × 32
- ○ Instruction set : Signed multiply (16 bits × 16 bits → 32 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
- ○ Memory space : 16 MB linear (common to program/data)
  Memory is divided in 1 MB unit blocks and wait states can be inserted into a bus cycle for every two blocks.
  Programmable wait function
  Idle state insertion function
- ○ External bus interface : 16-bit data bus (address/data multiplexed)
  Bus hold function
  External wait function
- ○ Internal memory : ROM/PROM: 32 KB
  RAM : 1 KB
- ○ Interrupt/exception : Non-maskable: 1 source
  Maskable: 14 sources (Eight levels of priorities can be set.)
  Illegal instruction code exception
- ○ I/O line : Input port: 1
  I/O port : 67
- ○ Real-time pulse unit : 16-bit timer/event counter: 1 ch
  16-bit timer: 1
  16-bit capture/compare register: 4
  16-bit interval timer: 1 ch
- ○ Serial interface : Asynchronous serial interface (UART): 1 ch
  Clocked serial interface (CSI): 1 ch
  Dedicated baud rate generator
- ○ Clock generator : ×5 function by PLL clock synthesizer
- ○ Power save function : HALT/IDLE/STOP mode
  Clock output stop function
- ○ CMOS technology

1.3 Application Fields

- For controlling systems using servo motor (such as PPCs, printers, and NC machine tools)
- For other control applications where high-speed response is required (such as engine control)

1.4 Ordering Information

Part Number	Package	Maximum Operating Frequency (MHz)	Internal ROM
μPD703000GC-25-xxx-7EA	100-pin plastic QFP (fine pitch) (14 × 14 mm)	25	Mask ROM
μPD703000GC-33-xxx-7EA	100-pin plastic QFP (fine pitch) (14 × 14 mm)	33	Mask ROM
μPD703001GC-25-7EA	100-pin plastic QFP (fine pitch) (14 × 14 mm)	25	None
μPD703001GC-33-7EA	100-pin plastic QFP (fine pitch) (14 × 14 mm)	33	None
μPD70P3000GC-25-7EA[Note]	100-pin plastic QFP (fine pitch) (14 × 14 mm)	25	One-time PROM
μPD70P3000GC-33-7EA[Note]	100-pin plastic QFP (fine pitch) (14 × 14 mm)	33	One-time PROM

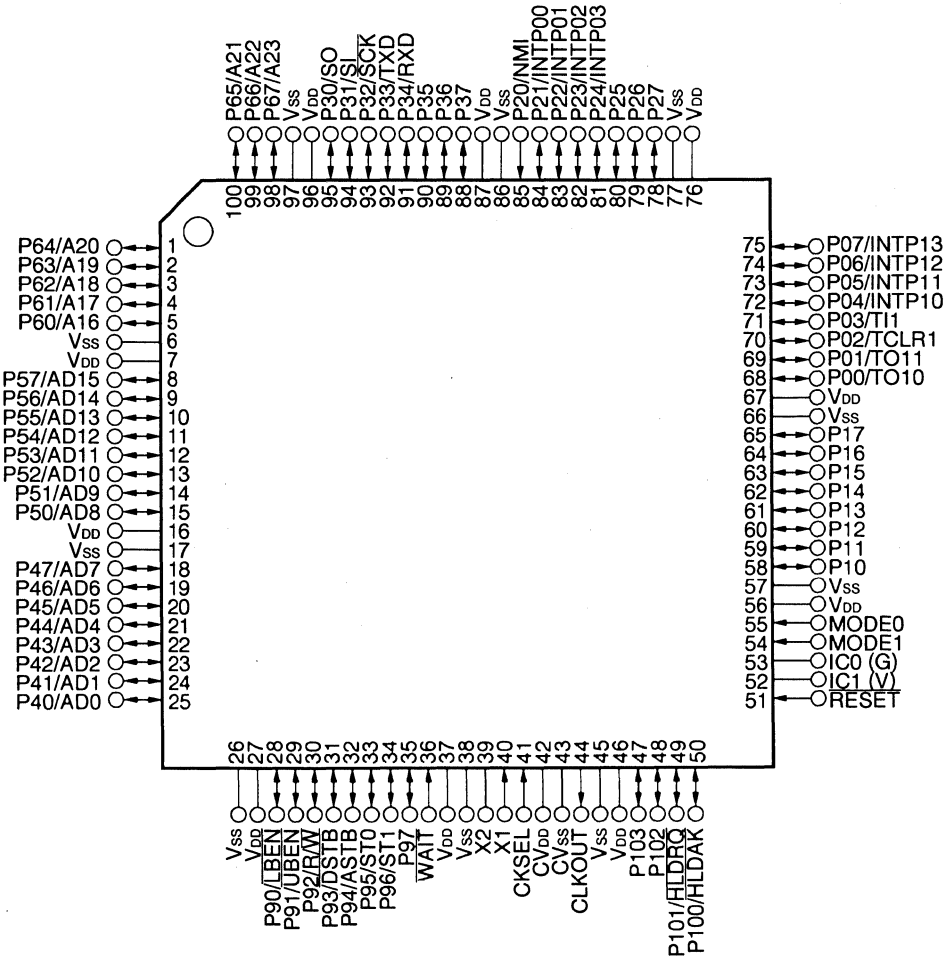
Note: Under development

Remark: xxx indicates a ROM code suffix.

★ 1.5 Pin Configuration (Top View)

1.5.1 Normal operation mode

- **100-pin plastic QFP (fine pitch)(14 × 14 mm)**
  μPD703000GC-xx-xxx-7EA
  μPD703001GC-xx-7EA
  μPD70P3000GC-xx-7EA



Cautions: **The content in parentheses indicates the processing of the pin not used in the normal operation mode.**

G: Connect this pin to VSS.

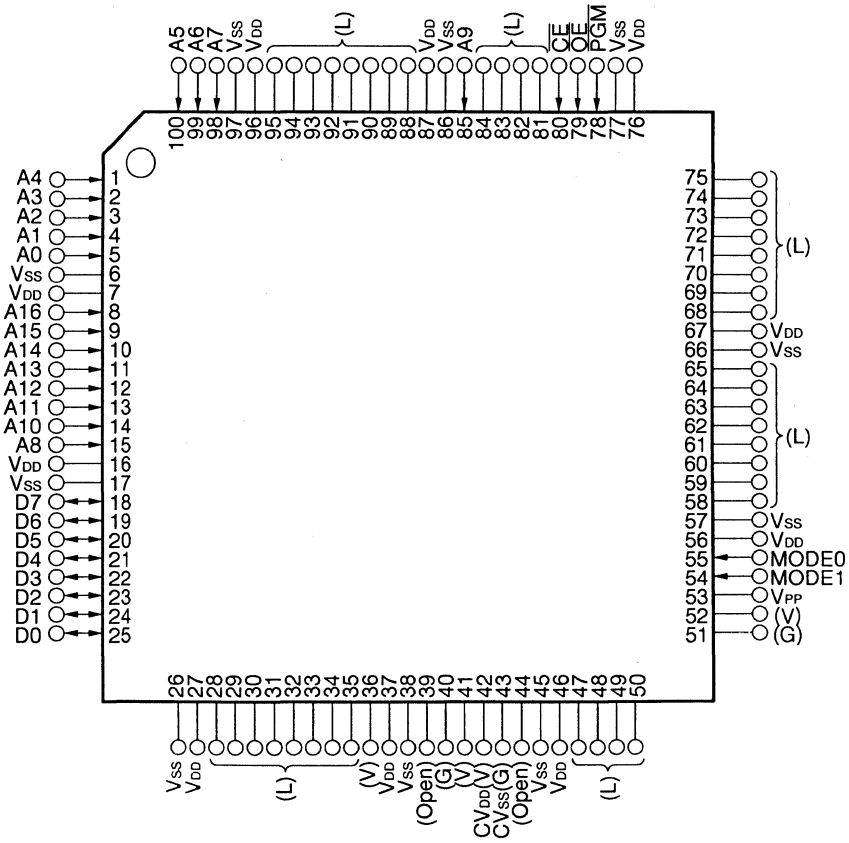
V: Connect this pin to VDD

Remark: xx indicates a maximum operating frequency.
xxx indicates a ROM code suffix.

P00-P07	:	Port0
P10-P17	:	Port1
P20-P27	:	Port2
P30-P37	:	Port3
P40-P47	:	Port4
P50-P57	:	Port5
P60-P67	:	Port6
P90-P97	:	Port9
P100-P103	:	Port10
TO10, TO11	:	Timer Output
TCLR1	:	Timer Clear
TI1	:	Timer Input
INTP00-INTP03, INTP10-INTP13	:	Interrupt Request From Peripherals
NMI	:	Non-maskable Interrupt Request
SO	:	Serial Output
SI	:	Serial Input
$\overline{SCK}$	:	Serial Clock
TXD	:	Transmit Data
RXD	:	Receive Data
AD0-AD15	:	Address/Data Bus
A16-A23	:	Address Bus
$\overline{LBEN}$	:	Lower Byte Enable
$\overline{UBEN}$	:	Upper Byte Enable
R/$\overline{W}$	:	Read/Write Status
$\overline{DSTB}$	:	Data Strobe
ASTB	:	Address Strobe
ST0, ST1	:	Status
$\overline{HLDAK}$	:	Hold Acknowledge
$\overline{HLDRQ}$	:	Hold Request
CLKOUT	:	Clock Output
CKSEL	:	Clock Select
$\overline{WAIT}$	:	Wait
MODE0, MODE1	:	Mode
$\overline{RESET}$	:	Reset
X1, X2	:	Crystal
CV_{DD}	:	Clock Generator Power Supply
CV_{SS}	:	Clock Generator Ground
V_{DD}	:	Power Supply
V_{SS}	:	Ground
IC0, IC1	:	Internally Connected

1.5.2 PROM programming mode

- **100-pin plastic QFP (fine pitch)(14 × 14 mm)**
  μPD70P3000GC-xx-7EA



Caution: **The content in parentheses indicates the processing of the pin not used in the PROM programming mode.**

L : Individually connect this pin to VSS via a resistor.

G : Directly connect this pin to VSS.

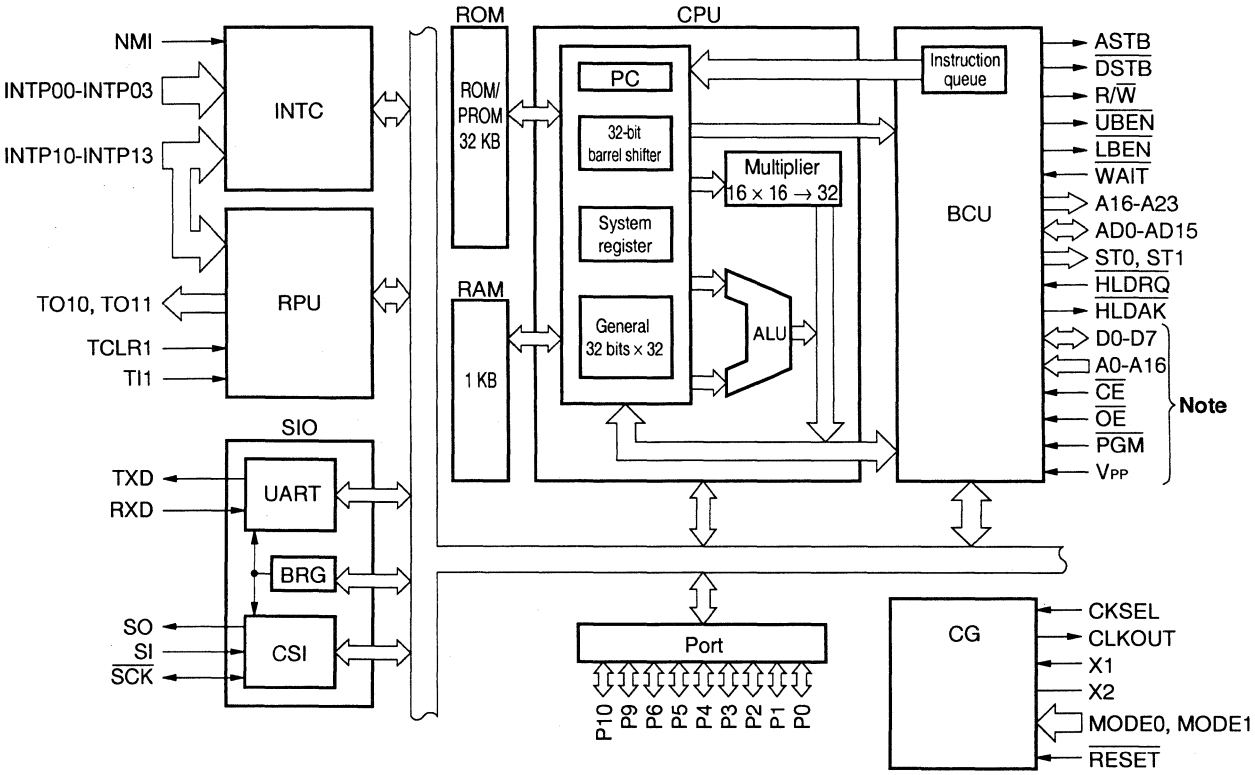
V : Directly connect this pin to VDD.

Open : Connect nothing to this pin.

A0-A16	: Address Bus	MODE0, MODE1	: Programming Mode Set
D0-D7	: Data Bus	VDD	: Power Supply
$\overline{CE}$	: Chip Enable	VSS	: Ground
$\overline{OE}$	: Output Enable	VPP	: Programming Power Supply
$\overline{PGM}$	: Programming Mode		

1.6 Function Block Configuration

1.6.1 Internal block diagram



Note: The pins used in the PROM programming mode.

1.6.2 Internal units

(1) CPU

Executes almost all the instruction processing such as address calculation, arithmetic/logic operation, and data transfer in 1 clock by using a 5-stage pipeline.

Dedicated hardware devices such as a multiplier (16 bits × 16 bits → 32 bits) and a barrel shifter (32 bits) are provided to increase the speed of processing complicated instructions.

(2) Bus control unit (BCU)

★ Initiates the necessary number of bus cycles based on the physical address obtained by the CPU. When the instruction of external memory is executed, if the CPU does not issue a request to start a bus cycle, generates a prefetch address to prefetch an instruction code. The prefetched instruction code is loaded to the internal instruction queue.

(3) ROM/PROM

ROM or PROM of 32-KB mapped starting from address 00000000H. Access is enabled/disabled by the MODE0 and MODE1 pins. With the PROM device, the programming mode is specified by these two pins. This ROM/PROM is accessed in 1 clock by the CPU when an instruction is fetched.

(4) RAM

1-KB RAM mapped starting from address FFFFE000H. This RAM can be accessed in 1 clock by the CPU when data is accessed.

(5) Interrupt controller (INTC)

Processes interrupt requests (NMI, INTP00-INTP03, INTP10-INTP13) from the internal peripheral hardware and external sources. Eight levels of priorities can be specified for these interrupt requests, and multiplexed processing control can be performed on an interrupt source.

(6) Clock generator (CG)

Supplies the CPU clock whose frequency is five times (when the internal PLL is used) or 1/2 times (when the PLL is not used) the frequency of the oscillator connected across the X1 and X2 pins. Input from an external clock source can also be referenced instead of using the oscillator.

(7) Real-time pulse unit (RPU)

Provides a 16-bit timer/event counter, a 16-bit interval timer, and capabilities for measuring pulse width and generation of programmable pulse outputs.

(8) Serial interface (SIO)

The serial interface consists of an asynchronous serial interface (UART) and a synchronous or clocked serial interface (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 output of the baud rate generator and system clock can be selected as the serial interface clock source.

(9) Ports

The V851 is provided with a total of 68 I/O port pins (of which one is an input port pin) that constitute ports 0 to 10. These port pins also function as various control pins.

Port	I/O	Function	
P0	8-bit I/O	General port	Timer I/O, external interrupt
P1			–
P2			External interrupt
P3			Serial interface
P4			External address/data bus
P5			
P6			External address bus
P9			External bus interface control signal I/O
P10	4-bit I/O		

[MEMO]

CHAPTER 2 PIN FUNCTIONS

The following table shows the names and functions of the V851's pins. These pins can be divided by function into port pins and other pins.

2.1 Pin Function List

2.1.1 Normal operation mode

(1) Port pins

(1/2)

Pin Name	I/O	Function	Alternate Function
P00	I/O	Port 0. 8-bit I/O port. Can be specified in input/output mode in 1-bit units.	TO10
P01			TO11
P02			TCLR1
P03			TI1
P04			INTP10
P05			INTP11
P06			INTP12
P07			INTP13
P10-P17	I/O	Port 1. 8-bit I/O port. Can be specified in input/output mode in 1-bit units.	–
P20	Input	Port 2. 8-bit I/O port. Can be specified in input/output mode in 1-bit units. P20 is fixed to input mode, however.	NMI
P21	I/O		INTP00
P22			INTP01
P23			INTP02
P24			INTP03
P25-P27			–
P30	I/O	Port 3. 8-bit I/O port. Can be specified in input/output mode in 1-bit units.	SO
P31			SI
P32			$\overline{SCK}$
P33			TXD
P34			RXD
P35-P37			–
P40-47	I/O	Port 4. 8-bit I/O port. Can be specified in input/output mode in 1-bit units.	AD0-AD7
P50-P57	I/O	Port 5. 8-bit I/O port. Can be specified in input/output mode in 1-bit units.	AD8-AD15

(2/2)

Pin Name	I/O	Function	Alternate Function
P60-P67	I/O	Port 6. 8-bit I/O port. Can be specified in input/output mode in 1-bit units.	A16-A23
P90	I/O	Port 9. 8-bit I/O port. Can be specified in input/output mode in 1-bit units.	$\overline{\text{LBEN}}$
P91			$\overline{\text{UBEN}}$
P92			R/$\overline{\text{W}}$
P93			$\overline{\text{DSTB}}$
P94			ASTB
P95			ST0
P96			ST1
P97			–
P100	I/O	Port 10. 4-bit I/O port. Can be specified in input/output mode in 1-bit units.	$\overline{\text{HLDAK}}$
P101			$\overline{\text{HLDRQ}}$
P102			–
P103			–

(2) Pins other than port pins

(1/2)

Pin Name	I/O	Function	Alternate Function
TO10	Output	Pulse signal output from timer 1.	P00
TO11			P01
TCLR1	Input	External clear signal input to timer 1.	P02
TI1		External count clock input to timer 1.	P03
INTP10	Input	External capture trigger input to timer 1. Also used to input external maskable interrupt request.	P04
INTP11			P05
INTP12			P06
INTP13			P07
NMI	Input	Non-maskable interrupt request input.	P20
INTP00	Input	External maskable interrupt request input.	P21
INTP01			P22
INTP02			P23
INTP03			P24
SO	Output	Serial transmit data output from CSI.	P30
SI	Input	Serial receive data input to CSI.	P31
$\overline{SCK}$	I/O	Serial clock I/O from/to CSI.	P32
TXD	Output	Serial transmit data output from UART.	P33
RXD	Input	Serial receive data input to UART.	P34
AD0-AD7	I/O	16-bit multiplexed address/data bus when external memory is used.	P40-P47
AD8-AD15			P50-P57
A16-A23	Output	Higher address bus when external memory is used.	P60-P67
$\overline{LBEN}$	Output	Lower byte enable signal output of external data bus.	P90
$\overline{UBEN}$		Higher byte enable signal output of external data bus.	P91
R/$\overline{W}$		External read/write status output.	P92
$\overline{DSTB}$		External data strobe signal output.	P93
ASTB		External address strobe signal output.	P94
ST0		External bus cycle status output.	P95
ST1			P96
$\overline{HLDAK}$	Output	Bus hold acknowledge output.	P100
$\overline{HLDRQ}$	Input	Bus hold request input.	P101
CLKOUT	Output	System clock output.	–
CKSEL	Input	Input specifying operation mode of clock generator.	–
$\overline{WAIT}$	Input	Control signal input inserting wait state to bus cycle.	–
MODE0, MODE1	Input	Specifies operation mode.	
$\overline{RESET}$	Input	System reset input.	
X1	Input	System clock oscillator connecting pins. Supply external clock to X1.	–
X2	–		–

(2/2)

Pin Name	I/O	Function	Alternate Function
CV_{DD}	–	Positive power supply for internal clock generator.	–
CV_{SS}	–	Ground for internal clock generator.	–
V_{DD}	–	Positive power supply	–
V_{SS}	–	Ground	–
IC0, IC1	–	Internally connected	–

2.1.2 PROM programming mode (μPD70P3000 only)

Control and timing of the V851 in the PROM mode are compatible with those of the μPD27C1001A. The functions of the pins of the V851 in the PROM mode are as follows:

Pin Name	Function in PROM Mode	Corresponding Pin of μPD27C1001A
P60-P67	Address input, low (A0-A7)	A0-A7
P50, P20, P51-P57	Address input, high (A8-A16)	A8, A9, A10-A16
P40-P47	Data input/output	D0-D7
P25	$\overline{CE}$ (chip enable) input	$\overline{CE}$
P26	$\overline{OE}$ (output enable) input	$\overline{OE}$
P27	$\overline{PGM}$ (program) input	$\overline{PGM}$
V_{PP}	Power supply for program write	V_{PP}
MODE0, MODE1	Operation mode specification	–

2.2 Pin Status

The operating status of each pin in each operation mode is as follows:

Pin \ Operating Status	Reset	STOP Mode	IDLE Mode	Bus Hold	Idle State	HALT Mode
AD0-AD15	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Hi-Z
A16-A23	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Retained	Retained
$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Retained	H
R/$\overline{\text{W}}$	Hi-Z	Hi-Z	Hi-Z	Hi-Z	H	H
$\overline{\text{DSTB}}$	Hi-Z	Hi-Z	Hi-Z	Hi-Z	H	H
ASTB	Hi-Z	Hi-Z	Hi-Z	Hi-Z	H	H
ST0, ST1	Hi-Z	Hi-Z	Hi-Z	Hi-Z	Idle status	Idle status
$\overline{\text{HLDRQ}}$	–	–	–	Operates	Operates	Operates
$\overline{\text{HLDAK}}$	Hi-Z	Hi-Z	Hi-Z	L	Operates	Operates
$\overline{\text{WAIT}}$	–	–	–	–	–	–

★

Hi-Z : high-impedance
Retained : Retains status in external bus cycle immediately before
L : low-level output
H : high-level output
– : input not sampled

2.3 Pin Function

2.3.1 Normal operation mode

(1) P00-P07 (Port0) ... 3-state I/O

These pins constitute an 8-bit I/O port, port 0. They also serves as control signal pins.
P00-P07 function not only as I/O port pins, but also as the I/O pins of the real-time pulse unit (RPU) and external interrupt request input pins. Each bit of port 0 can be specified in the port or control mode, by using port mode control register 0 (PMC0).

(a) Port mode

P00-P07 can be set in the input or output mode in 1-bit units by using port mode register 0 (PM0).

(b) Control mode

P00-P07 can be set in the port or control mode in 1-bit units by the PMC0 register.

(i) TO10, TO11 (Timer Output) ... output

These pins output pulse signals from timer 1.

(ii) TCLR1 (Timer Clear) ... input

This pin inputs an external clear signal to timer 1.

(iii) TI1 (Timer Input) ... input

This pin inputs an external count clock to timer 1.

(iv) INTP10-INTP13 (Interrupt Request From Peripherals) ... input

These pins are the external interrupt request input pins of timer 1.

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

These pins constitute an 8-bit I/O port, port 1, which can be set in the input or output mode in 1-bit units by using port mode register 1 (PM1).
The pins of port 1 function only as I/O pins and are not multiplexed with control pins.

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

These pins constitute an I/O port, port 2, which can be set in the input or output mode in 1-bit units except P20 which is fixed to the input mode. These pins function not only as port pins but also as external interrupt input pins.
Each bit of this port can be specified in the port or control mode by using port mode control register 2 (PMC2).
P25-P27 function only as I/O pins and are not multiplexed with control pins.

(a) Port mode

P21-P27 can be set in the input or output mode in 1-bit units by port mode register 2 (PM2).
PM20 is dedicated to the NMI input mode.

(b) Control mode

P20-P24 can be set in the port or control mode in 1-bit units by port mode control register 2 (PMC2).

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

This pin inputs a non-maskable interrupt request.

(ii) INTP00-INTP03 (Interrupt Request From Peripherals) ... input

These pins input external maskable interrupt requests.

(4) P30-P37 (Port 3) ... 3-state input

These pins constitute an 8-bit I/O port, port 3. They also function as control signal pins. P30-P37 function not only as I/O port pins but also as serial interface (UART, CSI) I/O pins in the control mode.
P35-P37 function only as I/O pins and are not multiplexed with control pins.

(a) Port mode

P30-P37 can be set in the input or output mode in 1-bit units by port mode register 3 (PM3).

(b) Control mode

P30-P37 can be set in the port or control mode in 1-bit units by the PMC3 register.

(i) SO (Serial Output) ... output

This pin outputs the serial transmit data of CSI.

(ii) SI (Serial Input) ... input

This pin inputs the serial receive data of CSI.

(iii) $\overline{\text{SCK}}$ (Serial Clock) ... 3-state I/O

This pin inputs/outputs the serial clock of CSI.

(iv) TXD (Transmit Data) ... output

This pin outputs the serial transmit data of UART.

(v) RXD (Receive Data) ... input

This pin inputs the serial receive data of UART.

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

These pins constitute an 8-bit I/O port, port 4. They also form a portion of the address/data bus connected to external memory.
P40-P47 function not only as I/O port pins but also as multiplexed address/data bus pins (AD0-AD7) in the control mode (external expansion mode) when an external memory is connected.
Each bit of this port can be set in the port or control mode, in 1-bit units, by using mode specification pins MODE0 and MODE1, and memory expansion mode register (MM).

(a) Port mode

P40-P47 can be set in the input or output port mode in 1-bit units by using port mode register 4 (PM4).

(b) Control mode (external expansion mode)

P40-P47 can be specified as AD0-AD7 by using the MODE0 and MODE1 pins and MM register.

(i) **AD0-AD7 (Address/Data0-7) ... 3-state I/O**

These pins constitute a multiplexed address/data bus when the external memory is accessed. They function as the A0-A7 output pins of a 24-bit address in the address timing (T1 state), and as the lower 8-bit data I/O bus pins of 16-bit data in the data timing (T2, TW, T3).

The output status of these pins changes in synchronization with the rising edge of the clock in each state of the bus cycle. AD0-AD7 go into a high-impedance state in the idle state (TI).

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

These pins constitute an 8-bit I/O port, port 5. They also form a portion of the address/data bus connected to external memory.

P50-P57 function not only as I/O port pins but also as multiplexed address/data bus pins (AD8-AD15) in the control mode (external expansion mode) when an external memory is connected.

Each bit of this port can be set in the port or control mode in 1-bit units by using mode specification pins MODE0 and MODE1, and memory expansion mode register (MM).

(a) Port mode

P50-P57 can be set in the input or output port mode in 1-bit units by using port mode register 5 (PM5).

(b) Control mode (external expansion mode)

P50-P57 can be specified as AD8-AD15 by using the MODE0 and MODE1 pins and MM register.

(i) **AD8-AD15 (Address/Data8-15) ... 3-state I/O**

These pins constitute a multiplexed address/data bus when the external memory is accessed. They function as the A8-A15 output pins of a 24-bit address in the address timing (T1 state), and as the higher 8-bit data I/O bus pins of 16-bit data in the data timing (T2, TW, T3).

The output status of these pins changes in synchronization with the rising of the clock in each state of the bus cycle. AD8-AD15 go into a high-impedance state in the idle state (TI).

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

These pins constitute an 8-bit I/O port, port 6. They also form a portion of the address/data bus connected to external memory.

P60-P67 function not only as I/O port pins but also as address bus pins (A16-A23) in the control mode (external expansion mode) when an external memory is connected. This port can be set in the port or control mode in 2-bit units by using mode specification pins MODE0 and MODE1, and memory expansion mode register (MM).

(a) Port mode

P60-P67 can be set in the input or output port mode in 1-bit units by using port mode register 6 (PM6).

(b) Control mode (external expansion mode)

P60-P67 can be specified as A16-A23 by using the MODE0 and MODE1 pins and MM register.

(i) **A16-A23 (Address/Data16-23) ... output**

These pins constitute the higher 8 bits of a 24-bit address bus when the external memory is accessed. The output status of these pins changes in synchronization with the rising edge of the clock in the T1 state. During the idle state (TI), the address of the bus cycle immediately before entering the idle state is retained.

(8) P90-P97 (Port 9) ... 3-state I/O

These pins constitute an 8-bit I/O port, port 9, and are also used to output control signals.

P90-P96 function not only as I/O port pins but also as control signal output pins in the control mode (external expansion mode) when an external memory is used.

This port can be set in the port or control mode in 5-, 2-, or 1-bit units by using mode specification pins MODE0 and MODE1, and memory expansion mode register (MM).

P97, however, is not multiplexed and can be used only as a port pin.

(a) Port mode

P90-P97 can be set in the input or output port mode in 1-bit units by using port mode register 9 (PM9).

(b) Control mode (external expansion mode)

P90-P96 can be used to output control signals when so specified by the MODE0 and MODE1 pins and MM register when an external memory is used.

(i) $\overline{\text{LBEN}}$ (Lower Byte Enable) ... output

This is the lower byte enable signal of the 16-bit external data bus.

This signal changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. The status of the bus signal remains unchanged in the idle state (TI).

(ii) $\overline{\text{UBEN}}$ (Upper Byte Enable) ... output

This is the upper byte enable signal of the 16-bit external data bus. It becomes inactive (high) in the byte access mode.

This signal changes in synchronization with the rising of the clock in the T1 state of the bus cycle. The status of the bus signal remains unchanged in the idle state (TI).

Access		$\overline{\text{UBEN}}$	$\overline{\text{LBEN}}$	A0
Word Access		0	0	0
Half-word Access		0	0	0
Byte Access	Even address	1	0	0
	Odd address	0	1	1

(iii) R/$\overline{\text{W}}$ (Read/Write Status) ... output

This is a status signal that indicates whether the bus cycle for external access is a read or write cycle. It goes high in the read cycle and low in the write cycle.

This signal changes in synchronization with the rising edge of the clock in the T1 state of the bus cycle. It goes high in the idle state (TI).

(iv) $\overline{\text{DSTB}}$ (Data Strobe) ... output

This is the access strobe signal of the external data bus.

It becomes active (low) in the T2 or TW state of the bus cycle, and becomes inactive (high) in the idle state (TI).

(v) ASTB (Address Strobe) ... output

This is the latch strobe signal of the external address bus.

It becomes active (low) in synchronization with the falling edge of the clock in the T1 state of the bus cycle, and becomes inactive (high) in synchronization with the falling edge of the clock in the T3 state. It goes high in the idle state (TI).

(vi) ST0, ST1 (Status0, 1) ... output

These are status signals which indicate the access type of the current bus cycle when external memory is referenced. The status changes in synchronization with the rising edge of the clock in the T1 and TI states of the bus cycle. In the table below, "Instruction fetch (branch)" cycle is the cycle that occurs only right after the execution is branched, and "Instruction fetch (continuous)" cycle is for the other cases of branch process.

ST1	ST0	Bus Cycle Status
0	0	Idle cycle
0	1	Instruction fetch (branch)
1	0	Operand data access
1	1	Instruction fetch (continuous)

★ The status "instruction fetch (branch)" is also indicated by the branch cycle immediately after system reset or on starting interrupt processing.

Immediately after system reset, the pins are set in the port mode and output no status, regardless of the setting in the ROM-less mode.

(9) P100-P103 (Port 10) ... 3-state I/O

Port 10 is a 4-bit I/O port that can be set in the input or output mode in 1-bit units. In addition to the function as a port, the pins constituting port 10 are used to input/output control signals, in the control mode, when an external bus master or ASIC is connected.

If port 10 is accessed in 8-bit units, the higher 4 bits are ignored if the access is write, and undefined if the access is read.

P102, P103, however, are not multiplexed and can be used only as port pins.

(a) Port mode

P100-P103 can be set in the input or output mode, in 1-bit units, by port mode register (PM10).

(b) Control mode

P100, P101 function as input and output pins for bus hold control signals when the function is enabled by mode control register 10 (PMC10).

(i) $\overline{\text{HLDAK}}$ (Hold Acknowledge) ... output

This is an acknowledge signal that indicates that the V851 has set the address bus, data bus, and control bus in the high-impedance state in response to a bus hold request.

As long as this signal is active, the address/data bus, and control signals remain in a high-impedance state.

(ii) $\overline{\text{HLDRQ}}$ (Hold Request) ... input

This input signal is used by an external device to request that the V851 relinquish control of the address, data bus, and control signals. This signal can be activated asynchronously with CLKOUT. When this signal becomes active, the V851 sets the address/data bus and control signals in the high-impedance state, after the current bus cycle completes. If there is no current bus activity, the address/data bus and control signals are immediately set to high-impedance. $\overline{\text{HLDAK}}$ is then made active and the bus and control lines are released.

(10) CLKOUT (Clock Output) ... output

This pin outputs the system clock, even during reset. The output of this pin can be fixed to a logic "0" when the clock output inhibit mode is set by PSC register.

(11) CKSEL (Clock Select) ... input

This pin specifies the operation mode of the clock generation circuit. Once set, the input value of this pin cannot be changed during operation.

CKSEL	Operation Mode
0	PLL mode
1	Direct mode

(12) $\overline{\text{WAIT}}$ (Wait) ... input

This control signal input pin inserts a data wait state to the bus cycle, and can be activated asynchronously to CLKOUT. This pin is sampled at the falling edge of the clock in the T2 and TW states of the bus cycle. If the set/hold time for the sampling timing is not satisfied, the wait state may not be inserted.

(13) MODE0, MODE1 (Mode0, 1) ... input

These pins specify the operation mode of the V851. Three operation modes can be selectable: single-chip mode, ROM-less mode, and PROM programming mode. The input value of these pins cannot be changed during normal operation.

MODE1	MODE0	Operation Mode	
0	0	ROM-less mode	
0	1	RFU (reserved)	
1	0	Single-chip mode	
1	1	PROM mode	V_{PP} = 5 V : read mode
			V_{PP} = 12.5 V : programming mode

(14) $\overline{\text{RESET}}$ (Reset) ... input

The $\overline{\text{RESET}}$ signal is an asynchronous input signal. A valid low-level signal on the $\overline{\text{RESET}}$ pin initiates a system reset, regardless of the clock operation. In addition to normal system initialization/start functions, the $\overline{\text{RESET}}$ signal is also used for exiting processor standby modes (HALT, IDLE, or STOP).

(15) X1, X2 (Crystal) ... input (X1 only)

An oscillator for system clock generation is connected across these pins.

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

(16) CV_{DD} (Clock Generator Power Supply)

This pin supplies positive power to the internal clock generator.

(17) CV_{SS} (Clock Generator Ground)

This is the ground pin of the internal clock generator.

(18) V_{DD} (Power Supply)

This pin supplies positive power. Connect all the V_{DD} pins to a positive power supply.

(19) V_{SS} (Ground)

This is a ground pin. Connect all the V_{SS} pins to ground.

(20) IC0 (Internally Connected)

This pin is internally connected and must be connected to V_{SS}.

★ **(21) IC1 (Internally Connected)**

This pin is internally connected and must be connected to V_{DD}.

2.3.2 PROM programming mode (μPD70P3000 only)

(1) A0-A16 ... input

These pins constitute an address bus that selects an address of the internal PROM (0000H-7FFFH).

(2) D0-D7 ... I/O

These pins constitute a data bus through which the internal PROM is written/read.

(3) $\overline{PGM}$ ... input

This pin inputs a program pulse and is activated when V_{PP} = 12.5 V, $\overline{CE}$ = 0, and $\overline{OE}$ = 1. Upon activation, the program on D0-D7 is written to an internal PROM cell selected by A0-A16.

(4) $\overline{CE}$ ... input

This is a chip enable input pin. When this signal is active, the program in PROM can be written/read.

(5) $\overline{OE}$ ... input

This is an output enable signal input pin and inputs a read strobe signal to the internal PROM. When the signal is activated while $\overline{CE}$=0, the contents of the PROM location selected by A0-A16, will appear at the outputs, D0-D7.

(6) V_{PP} ... input

This pin inputs a program pulse. When this pin is activated while V_{PP} = 12.5 V, $\overline{CE}$ = 0, and $\overline{OE}$ = 1, the program byte on D0-D7 can be written to the internal PROM cell selected by A0-A16.

(7) V_{DD}

Positive power supply pin

(8) V_{SS}

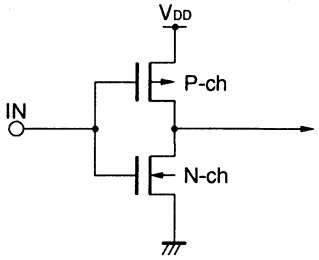
GND pin

2.4 Processing of Unused Pins

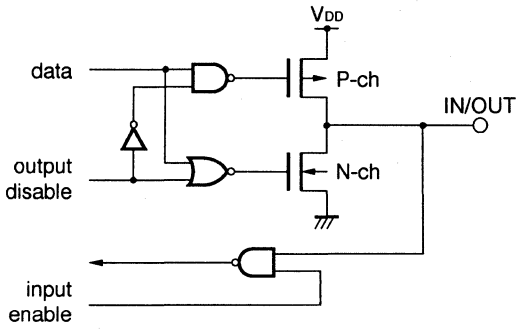
Pin	I/O Circuit Type	Recommended Connection
P00/TO10, P01/TO11	5	Input status : Individually connect to V_{DD} or V_{SS} via resistor Output status : Open
P02/TCLR1, P03/TI1, P04/INTP10-P07/INTP13	8	
P10-P17	5	
P20/NMI	2	Directly connect to V_{SS} via resistor
P21/INTP00-P24/INTP03	8	Input status : Individually connect to V_{DD} or V_{SS} via resistor Output status : Open
P25	5	
P26, P27	8	
P30/SO	5	
P31/SI, P32/$\overline{SCK}$	8	
P33/TXD, P34/RXD, P35	5	
P36, P37	8	
P40/AD0-P47/AD7	5	
P50/AD8-P57/AD15		
P60/A16-P67/A23		
P90/$\overline{LBEN}$		
P91/$\overline{UBEN}$		
P92/R/$\overline{W}$		
P93/$\overline{DSTB}$		
P94/ASTB		
P95/ST0, P96/ST1		
P97		
P100/$\overline{HLDAK}$		
P101/$\overline{HLDRQ}$		
P102		
P103		
CLKOUT	3	Open
CKSEL	2	Individually connect to V_{DD} or V_{SS} via resistor.
$\overline{WAIT}$	1	Directly connect to V_{DD} via resistor.
MODE0, MODE1	2	Individually connect to V_{DD} or V_{SS} via resistor.
$\overline{RESET}$		
IC0	–	Directly connect to V_{SS} via resistor.
IC1	–	Directly connect to V_{DD} via resistor. ★

2.5 I/O Circuit of Pin

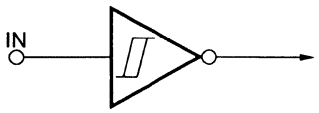
Type 1



Type 5

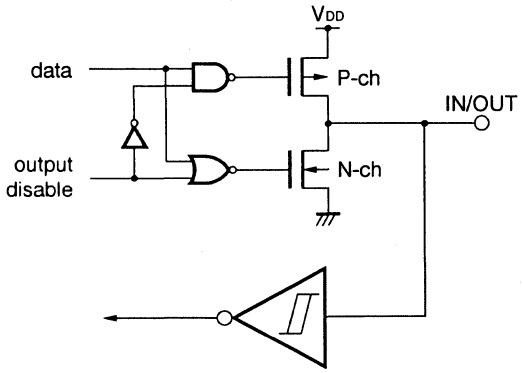


Type 2

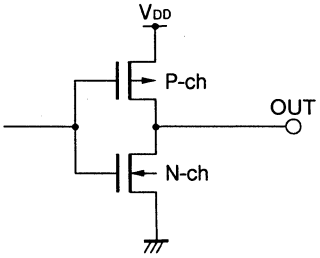


Schmitt trigger input with hysteresis characteristics

Type 8



Type 3



CHAPTER 3 CPU FUNCTIONS

The CPU of the V851 is based on a RISC architecture and executes most instructions in one clock cycle by using a 5-stage pipeline.

3.1 Features

- O Minimum instruction cycle: 30 ns (at 33 MHz) ★
- O Address space: 16 MB linear
- O Thirty-two 32-bit general registers
- O Internal 32-bit architecture
- O Five-stage pipeline control
- O Multiplication/division instructions
- O Saturated operation instructions
- O Single-cycle 32-bit shift instruction
- O Long/short instruction format
- O Internal memory
  - ROM/PROM: 32 KB
  - RAM: 1 KB
- O Four types of bit manipulation instructions
  - Set
  - Clear
  - Not
  - Test

3.2 CPU Register Set

The registers of the V851 can be classified into two categories: a general-purpose program register set and a dedicated system register set. All the registers are 32 bits wide. In this section, only summarized information is contained. For more details, refer to **"V850 Family User's Manual –Architecture–."**

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

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

3.2.1 Program register set

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

(1) General registers

Thirty-two general registers, r0-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-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 address generation
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	Used as register indicating beginning of text area[Note] ★
r6-r29	–	Address/data variable registers
r30	Element pointer	Base pointer register when memory is accessed
r31	Link pointer	Used by compiler when calling functions
PC	Program counter	Holds instruction address during program execution

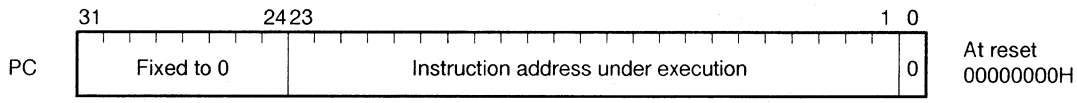
Note: Area in which program codes are located.

(2) Program counter

This register holds the address of the instruction under execution. The lower 24 bits of this register are valid, and bits 31-24 are fixed 0. If a carry occurs from bit 23 to 24, 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

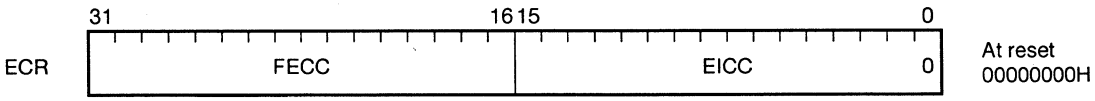
System registers control the status of the CPU and hold interrupt information. For values of all registers after reset, refer to section "**10.3 Initialize**".

Table 3-2. System Register Numbers

No.	System Register Name	Usage	Operation
0	EIPC	Status saving registers during interrupt	These registers save the PC and PSW when a trap or interrupt occurs. Because only one set of these registers are available, their contents must be saved when multiple interrupts are enabled. The higher 8 bits of EIPC, and the higher 24 bits of EIPSW are fixed to 0.
★ 1	EIPSW		
2	FEPC	Status saving registers for NMI	These registers save PC and PSW when NMI occurs. The higher 8 bits of FEPC, and the higher 24 bits of FEPSW are fixed to 0.
★ 3	FEPSW		
4	ECR	Interrupt source register	If trap, 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 exception code of NMI is set. The lower 16 bits are called EICC to which exception code of trap/interrupt is set (refer to **Figure 3-2**).
5	PSW	Program status word	Program status word is collection flags that indicate program status (instruction execution result) and CPU status (refer to **Figure 3-3**).
6-31	Reserved		

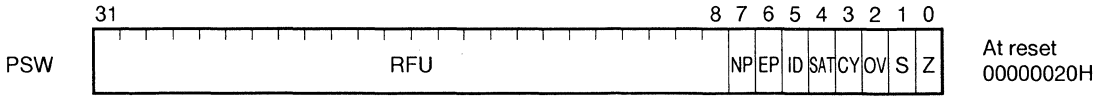
To read/write these system registers, specify a system register number indicated by the system register load/store instruction (LDSR or STSR instruction).

Figure 3-2. Interrupt Source Register (ECR)



Bit Position	Bit Name	Function
31-16	FECC	Fatal Error Cause Code Exception code of NMI (Refer to **Table 5-1. Interrupt List**)
15-0	EICC	Exception/Interrupt Cause Code Exception code of trap/interrupt (Refer to **Table 5-1. Interrupt List**)

Figure 3-3. Program Status Word (PSW)



Bit Position	Bit Name	Function
31-8	RFU	Reserved field (fixed to 0)
7	NP	NMI Pending Indicates that NMI processing is in progress. This flag is set when NMI is accepted, and disables multiple interrupts by masking NMI request.
6	EP	Exception Pending Indicates that trap processing is in progress. This flag is set when trap is generated.
5	ID	Interrupt Disable Indicates that accepting external interrupt request is disabled.
4	SAT	Saturated Math This flag is set if 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 result of operation is negative. It is reset if result is positive.
0	Z	Zero This flag is set if result of operation is zero (if result is not zero, it is reset).

3.3 Operation Modes

3.3.1 Operation modes

The V851 has the following operations modes. These modes are selected by the MODE0 and MODE1 pins.

(1) Single-chip mode

In single-chip mode, there is no external bus interface. After the system has been released from the reset status, the pins related to the bus interface are set for I/O port mode, execution branches to the reset entry address of the internal ROM/PROM, and instruction processing is started. However, access to external memory and peripheral devices can be enabled by setting the appropriate bits in the external memory expansion mode register (MM: refer to **3.4.6 (1)**).

(2) ROM-less mode

In ROM-less mode, all control signal relating to external bus operation are made available at the appropriate pins. After the system has been released from the reset status, the bus control signal pins are enabled, execution branches to the reset address of external memory, and instruction processing is started. Instruction fetch and data access from internal ROM/PROM are disabled.

(3) PROM programming mode

In PROM programming mode, the appropriate pins function to provide a μPD27C1001A compatible interface. By using a PROM programmer that supports the μPD27C1001A, the internal PROM of the V851 can be programmed.

(4) PROM read mode

In PROM read mode, the appropriate pins function to provide a μPD27C1001A compatible interface. By using a PROM programmer that supports the μPD27C1001A, the internal PROM of the V851 can be read.

3.3.2 Specifying operation mode

The operation mode of the V851 is specified by using the MODE0 and MODE1 pins. Set these pins in the application system. The MODE pins will also determin the V851 memory map configuration (refer to section "3.4.4". Do not change the setting of these pins during operation.

If the setting is changed during operation, the functionality is not guaranteed.

(1) In normal mode

MODE1	MODE0	Operation Mode
0	0	ROM-less mode
0	1	RFU (reserved)
1	0	Single-chip mode
1	1	RFU (reserved)

(2) In PROM mode

Pin Status			Operation Mode
V_{PP}	MODE1	MODE0	
5 V	0	0	RFU (reserved)
	0	1	
	1	0	
	1	1	PROM mode (read mode)
12.5 V	1	1	PROM mode (programming mode)

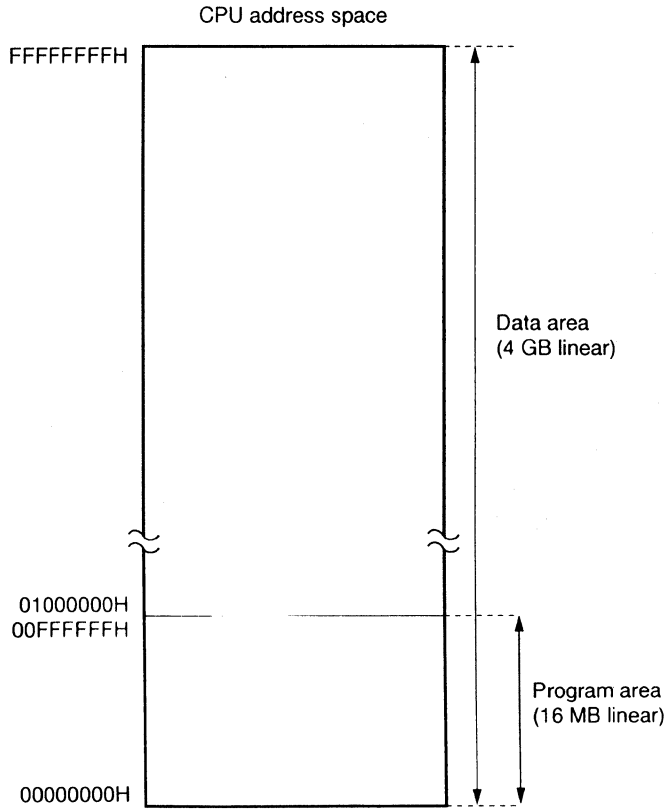
3.4 Address Space

3.4.1 CPU address space

The CPU of the V851 is of 32-bit architecture and supports up to 4 GB of linear address space (data space) during operand addressing (data access). When referencing instruction addresses, a linear address space (program space) of up to 16 MB is supported.

Figure 3-4 shows the CPU address space.

Figure 3-4. CPU Address Space

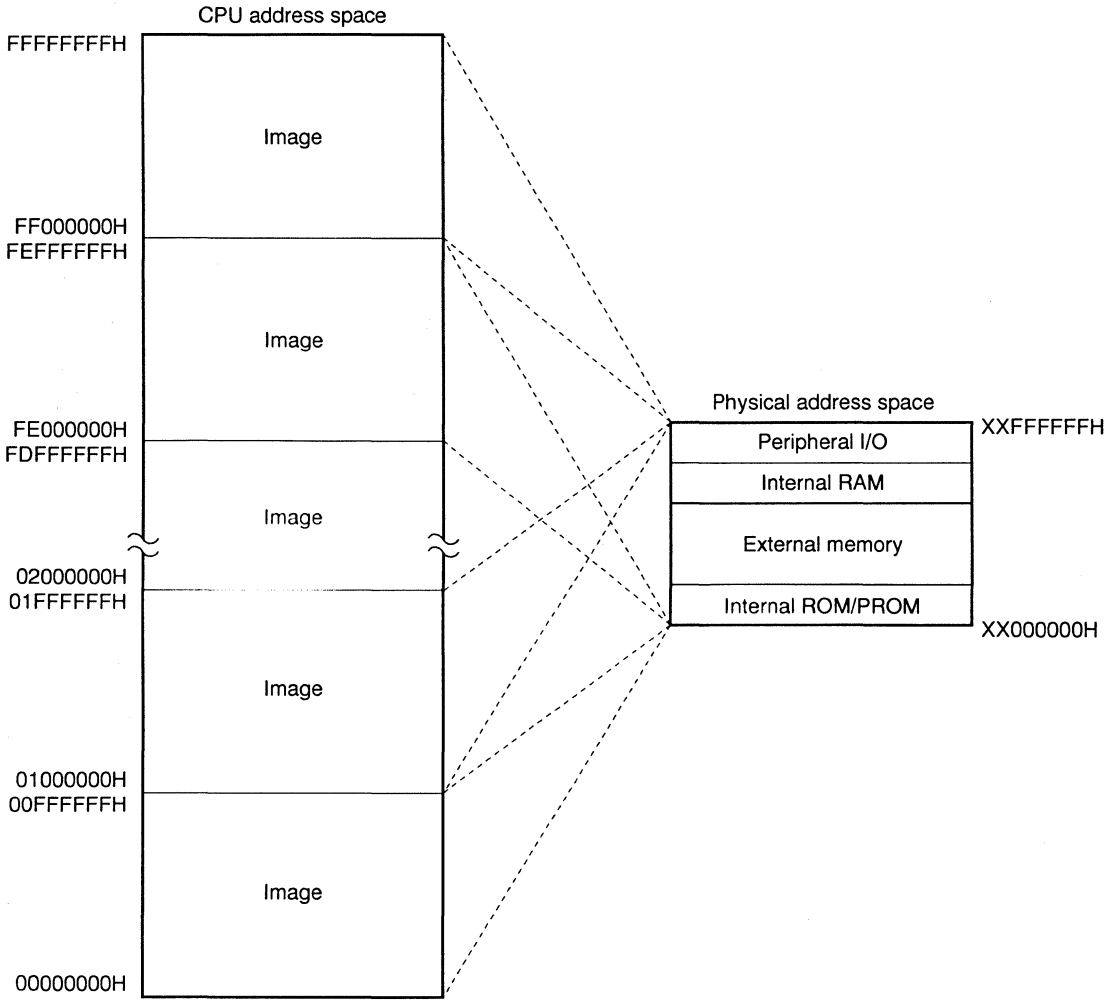


3.4.2 Image (Virtual Address Space)

The core CPU supports 4 GB of "virtual" addressing space, or 256 memory blocks, each containing 16-MB memory locations. In actuality, the same 16-MB block is accessed regardless of the values of bits 31-24 of the CPU address, since these bits are not seen on the external address bus. Figure 3-5 shows the image of the virtual addressing space.

Because the higher 8 bits of a 32-bit CPU address are ignored and the CPU address is only seen as a 24-bit external physical address, the physical location XX000000H is equally referenced by multiple address values 00000000H, 010000000H, 02000000H... through FF000000H. This mapping of CPU address to physical address applies to each of the 16-MB locations of the V851.

Figure 3-5. Image on Address Space



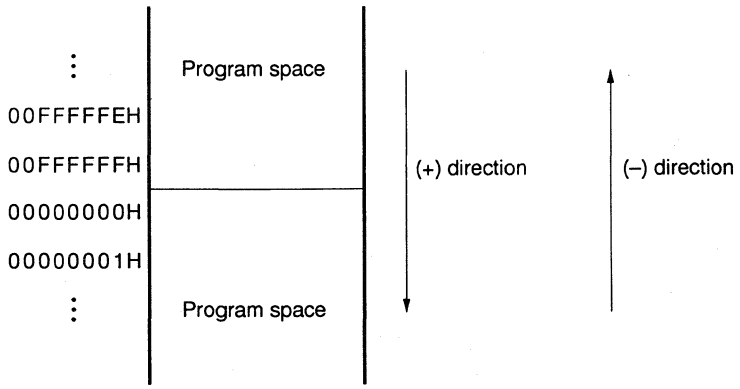
3.4.3 Wrap-around of CPU address space

(1) Program space

Of the 32 bits of the PC (program counter), the higher 8 bits are set to "0", and only the lower 24 bits are valid. Even if a carry or borrow occurs from bit 23 to 24 as a result of branch address calculation, the higher 8 bits ignore the carry or borrow and remain at "0".

Therefore, the lower-limit address of the program space, address 00000000H, and the upper-limit address 00FFFFFFH are contiguous addresses, and the program space is wrapped around at the boundary of these addresses.

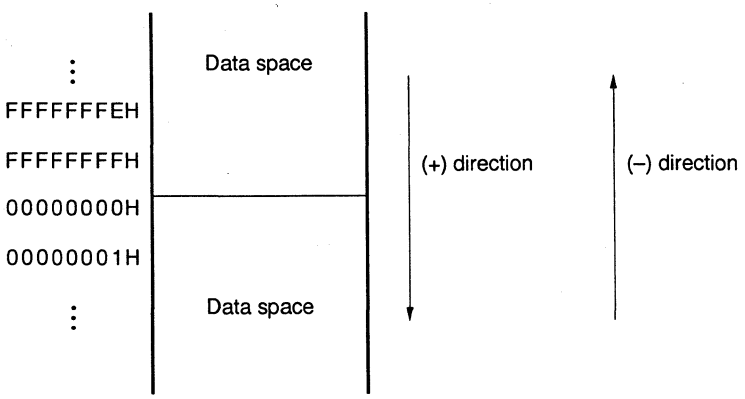
Caution: No instruction can be fetched from the 4-KB area of 00FFF000H-00FFFFFFH because this area is defined as peripheral I/O area. Therefore, do not execute any branch operation instructions in which the destination address will reside in any part of this area.



(2) Data space

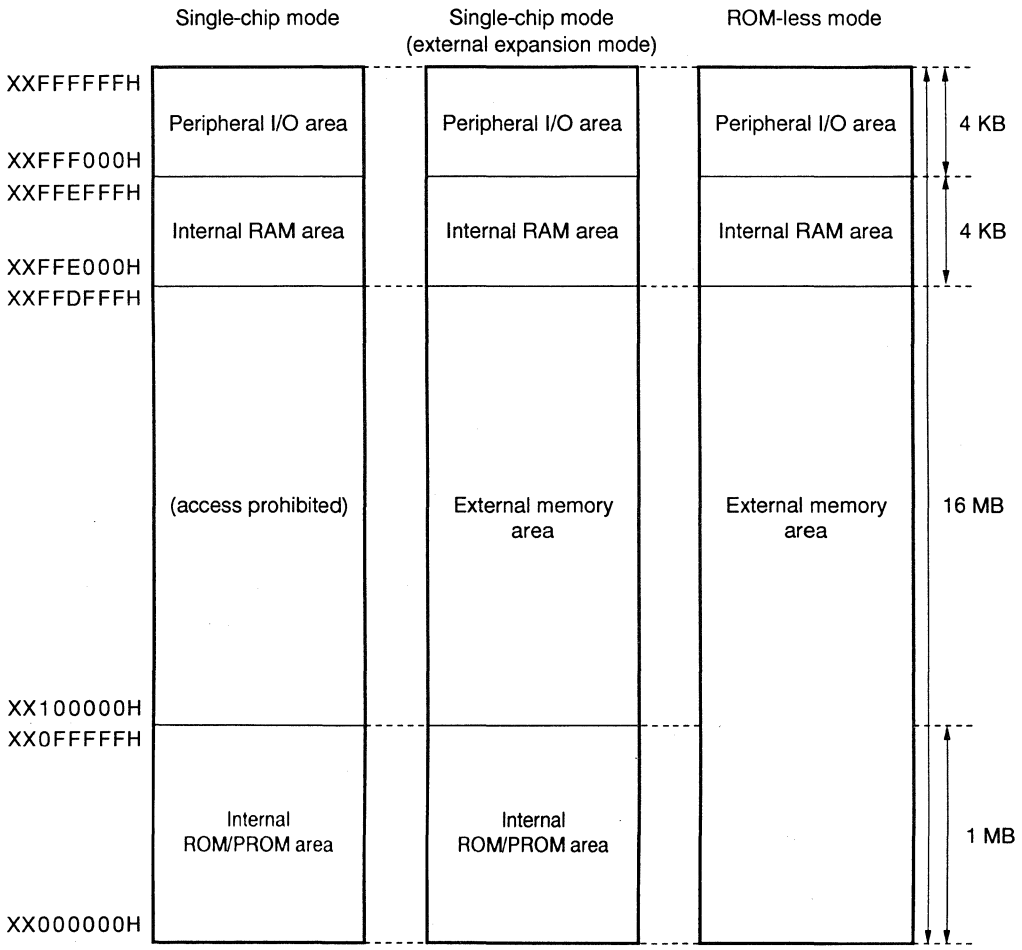
The result of operand address calculation that exceeds 32 bits is ignored.

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

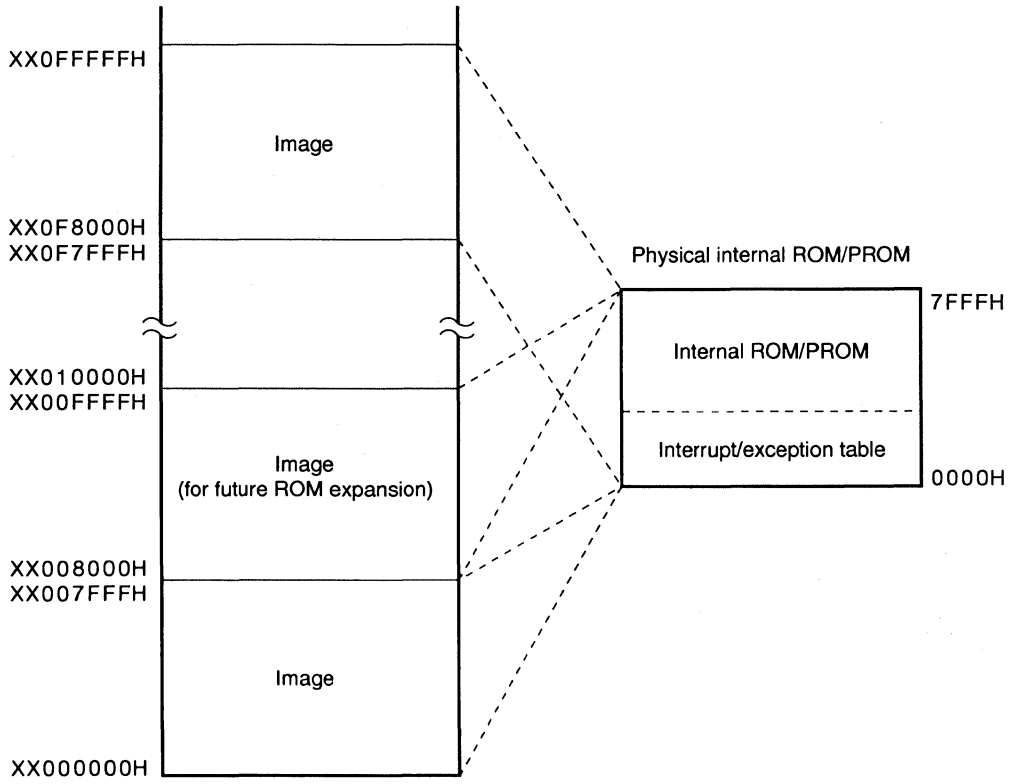
The V851 reserves areas as shown below. Each mode is specified by using the MODE0 and MODE1 pins (refer to **3.3 Operation Mode**).



3.4.5 Area

(1) Internal ROM/PROM area and interrupt/exception table

A 1-MB area corresponding to addresses 000000H-0FFFFFH is reserved for the internal ROM/PROM area. The V851 is provided with a 32-KB area of addresses 000000H-007FFFH as a physical internal ROM/PROM. The image of 000000H-007FFFH is seen in the rest of the area (008000H-0FFFFFH)

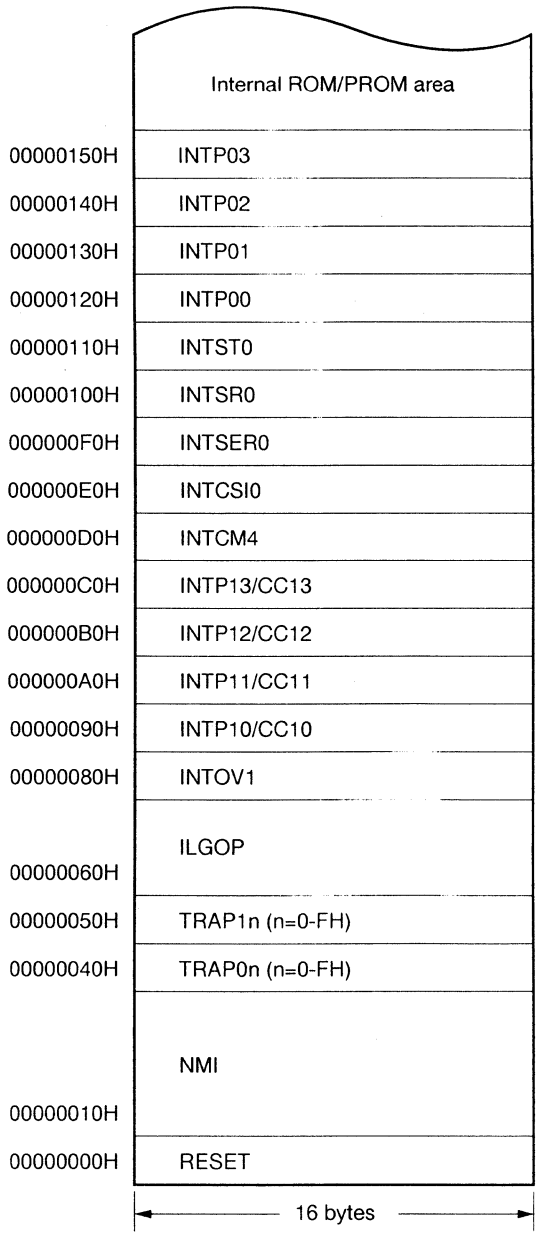


Interrupt/exception table

The V851 increases the interrupt response speed by assigning destination addresses corresponding to interrupts/exceptions.

The collection of these destination addresses is called an interrupt/exception table, which is located in the internal ROM/PROM area. When an interrupt/exception request is granted, execution jumps to the corresponding destination address, and the program written at that memory address is executed. Figure 3-6 shows the names of interrupts/exceptions, and the corresponding addresses.

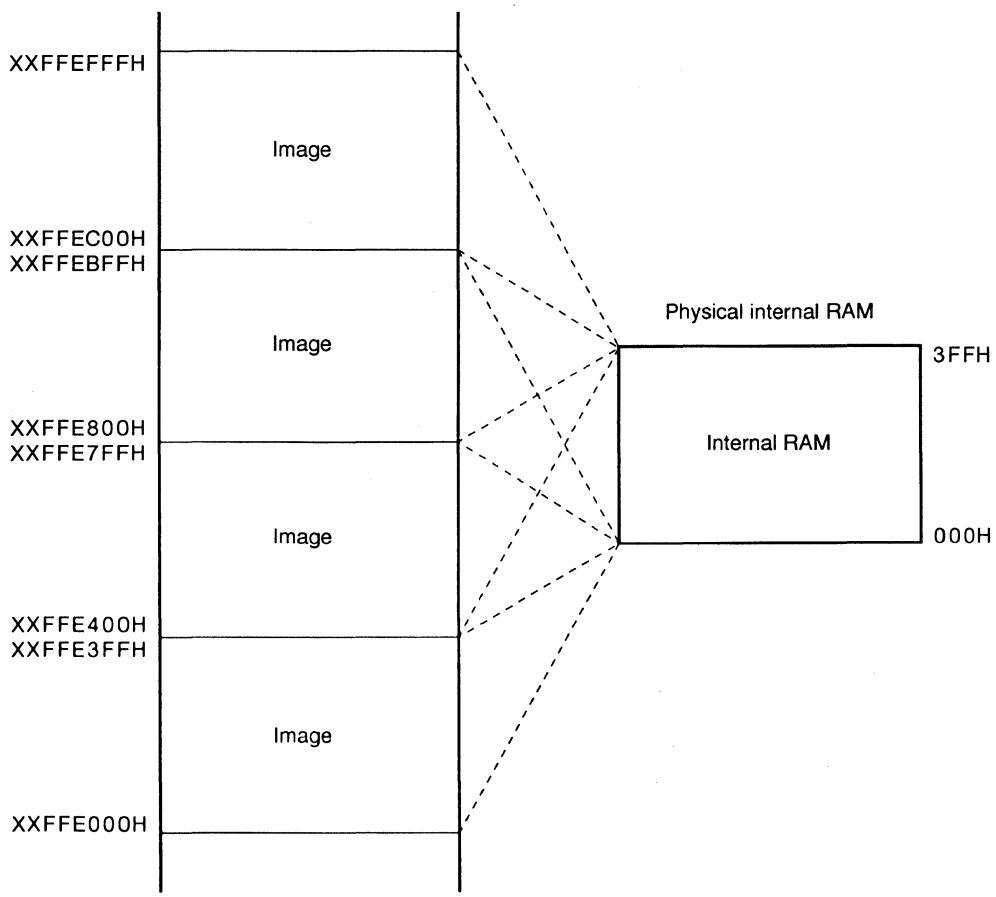
Figure 3-6. Interrupt/Exception Table



In the ROM-less mode, the internal ROM/PROM area is referenced as external memory area. To assure correct operation after reset, connecting an external memory to the reset routine is required.

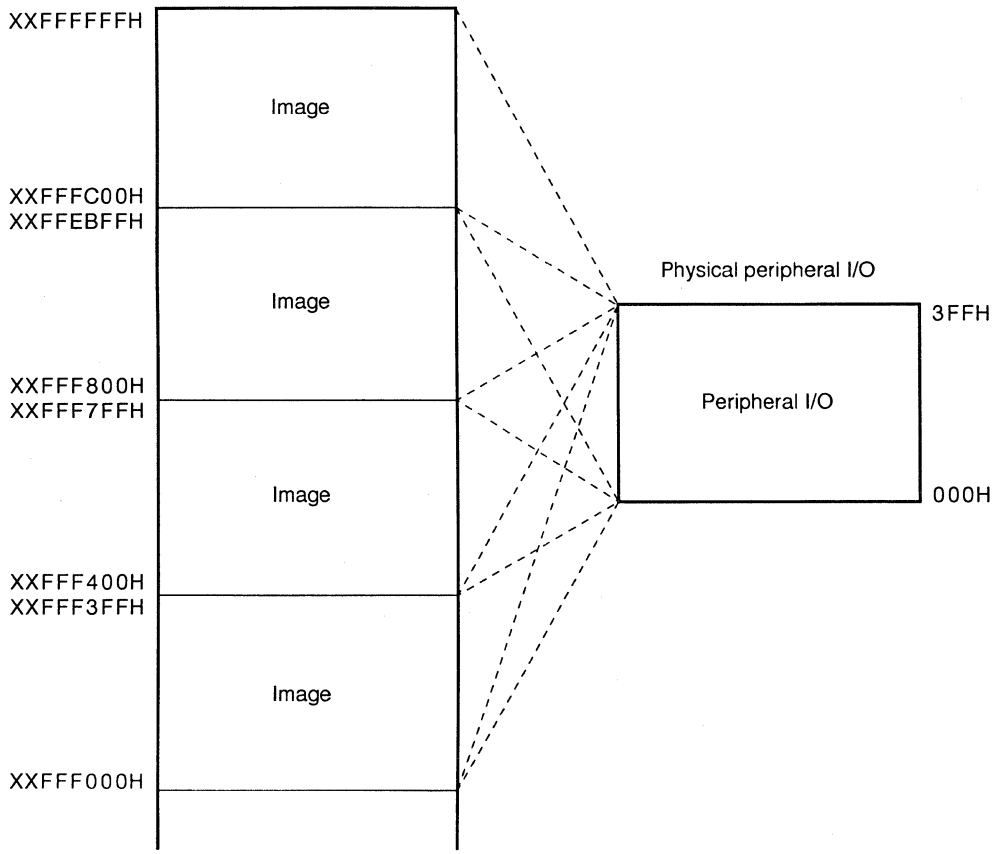
(2) Internal RAM area

A 4-KB area corresponding to addresses FFE000H through FFEFFFH is reserved as an internal RAM area. The V851 is provided with 1 KB of addresses FFE000H-FFE3FFH as a physical internal RAM area, and the image of FFE000H-FFE3FFH can be seen on the rest of the area (FFE400H-FFEFFFH).



(3) Peripheral I/O area

A 4-KB area of addresses FFF000H-FFFFFFH is reserved as a peripheral I/O area. The V851 is provided with a 1-KB area of addresses FFF000H-FFF3FFH as a physical peripheral I/O area, and the image of FFF000H-FFF3FFH can be seen on the rest of the area (FFF400H-FFFFFFH).



Peripheral I/O registers associated with the on-chip peripherals and the CPU are all memory-mapped to peripheral I/O area. Instruction fetches are not allowed in this area.

Cautions: 1. The least significant bit of an address is not decoded, since all registers reside on an even address. If an odd address (2n+1) in the peripheral I/O area is referenced, the register at the next lowest even address (2n) will be accessed.

2. The V851 does not have a peripheral I/O register than can be accessed in word units. If a register is accessed with a word operation, the effects will be limited to the halfword referenced by the instruction.

3. If a register that can be accessed in byte units is accessed in half-word units, the higher 24 bits become undefined, if the access is a read operation. If a write access is made, only the data in the lower 8 bits is written to the register.

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 V851 can use an area of up to ××100000H-××FFDFFFH in the single-chip mode and an area of up to ××000000H-××FFDFFFH in the ROM-less mode, for external memory accesses.

In the external memory area, 64 KB, 256 KB, 1 MB, 4 MB, or 16 MB of physical external memory can be allocated when the external expansion mode is specified. The same image as that of the physical external memory can be seen continuously on the external memory area, as shown in Figures 3-7 through 3-9, when the memory is not fully expanded (to 16 MB).

The internal RAM area, peripheral I/O area, and internal ROM/PROM area in the single-chip mode are not subject to external memory access.

Figure 3-7. External Memory Area (when expanded to 64 KB, 256 KB, or 1 MB)

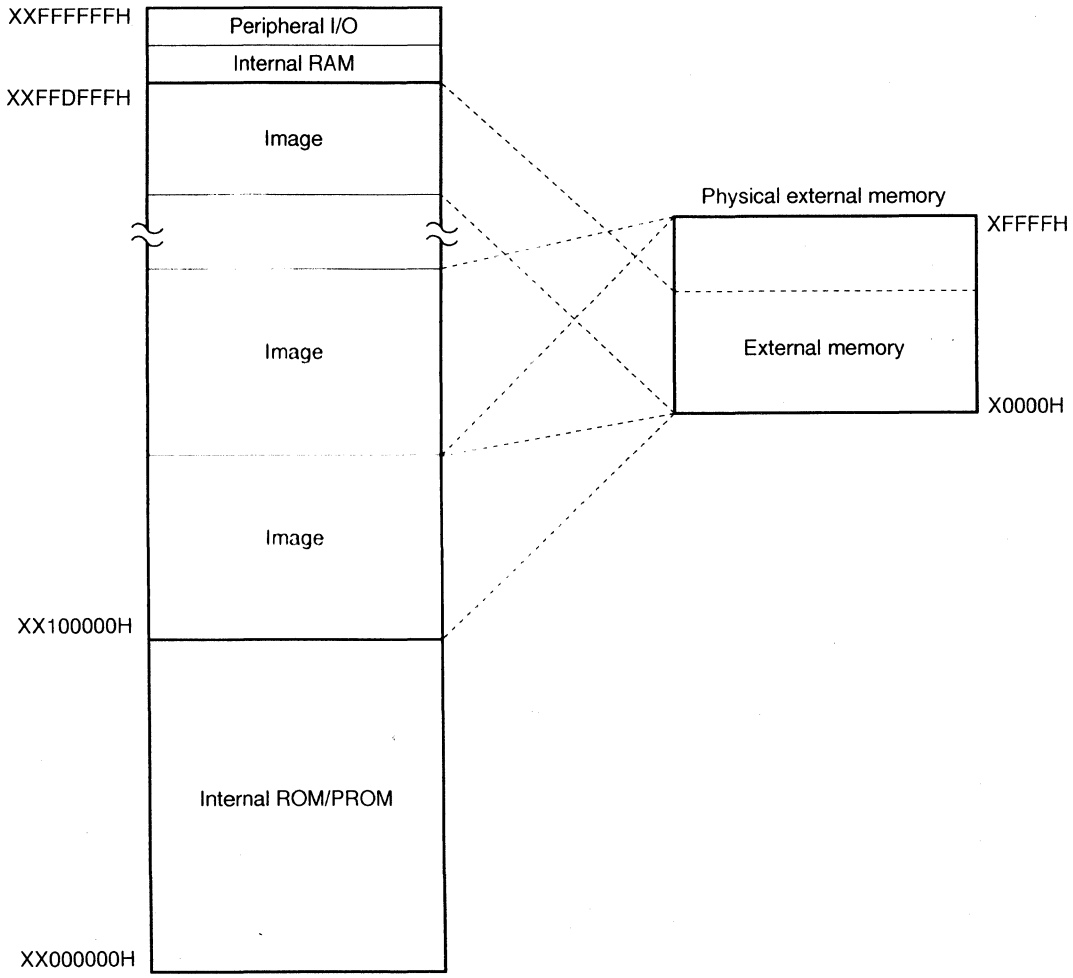


Figure 3-8. External Memory Area (when expanded to 4 MB)

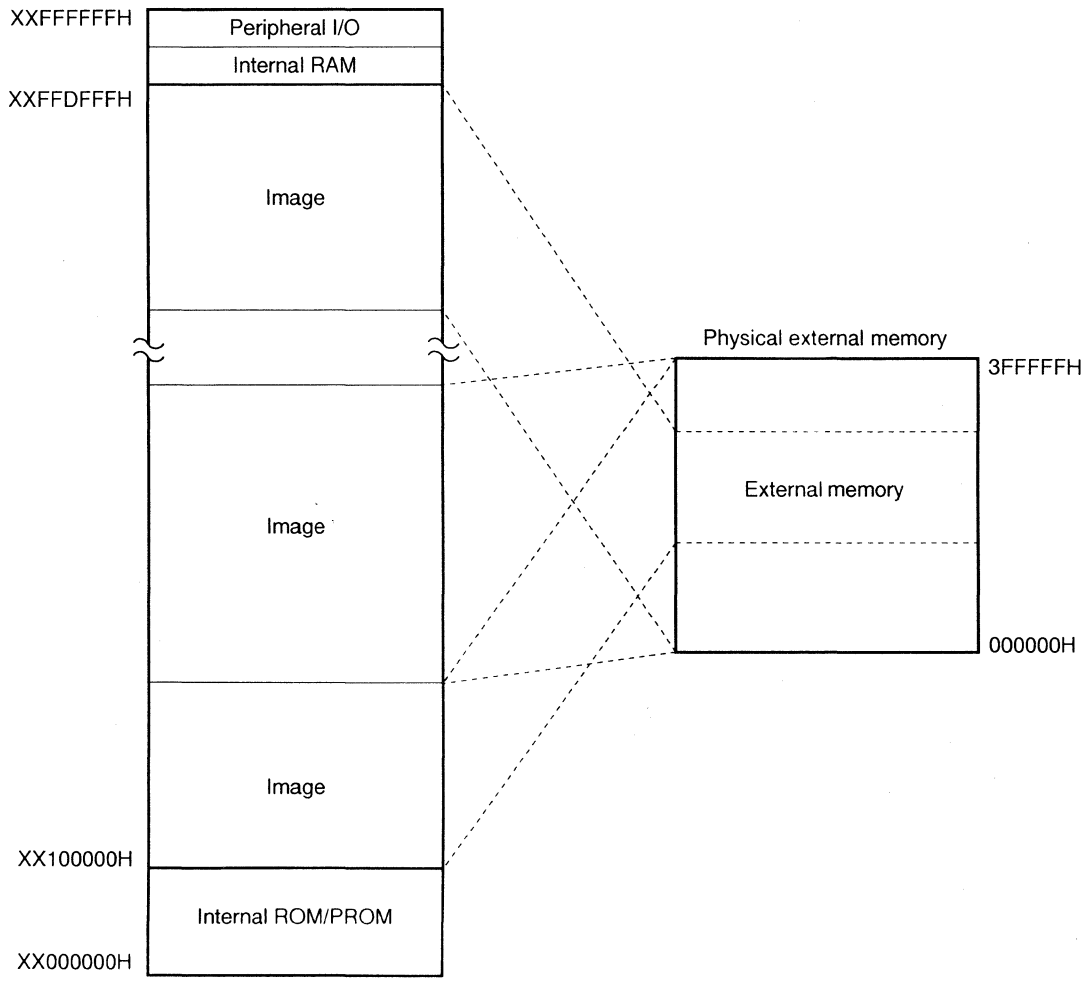
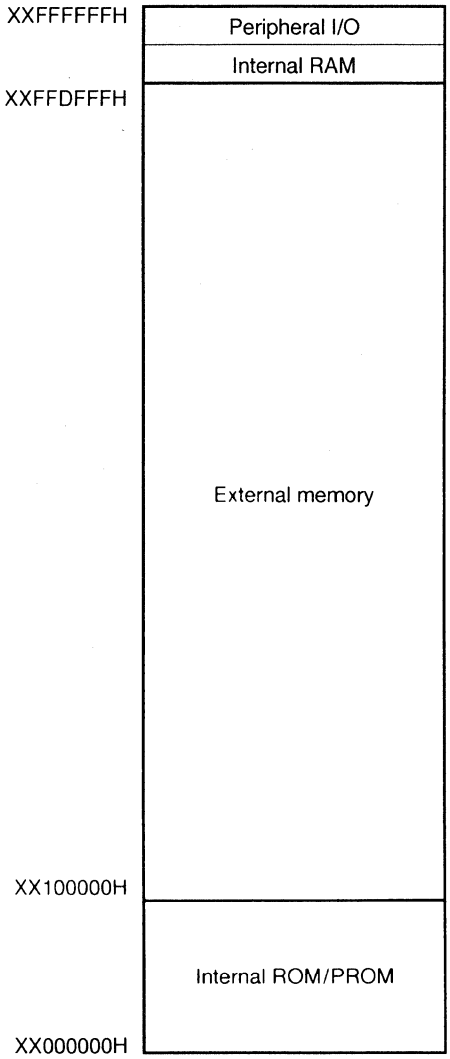


Figure 3-9. External Memory Area (when fully expanded)



3.4.6 External expansion mode

The V851 allows external devices to be connected to the external memory space by using the pins of ports 4 through 10. To connect an external device, the port pins must be set in the external expansion mode by using the MODE0 and MODE1 pins and memory expansion mode register (MM). The MODE0 and MODE1 pins specify an operation mode of the V851. When MODE0 = 0 and MODE1 = 0, the V851 is set in the ROM-less mode; when MODE0 = 0 and MODE1 = 1, the single-chip mode is used.

In ROM-less mode, the pins of ports 4 through 6, and P90 to P94 are set to external expansion mode during reset, thereby enabling the external bus signals and allowing communication with external memory devices.

In single-chip mode, the I/O port pins are set to port mode during reset, thus disabling the external bus signals and preventing any communications with external devices. The condition can be overridden by programming the MM register and setting the port pins to external expansion mode.

The memory area is also set by the MM register. However, the port 10 is set in the external expansion mode by using the PMC10 register (refer to **9.3.9 Port 10**).

(1) Memory expansion mode register (MM)

This register sets the mode of each pin of ports 4 through 9. In the external expansion mode, an external device can be connected to the external memory area of up to 16 MB. However, the external device cannot be connected to the internal RAM area, peripheral I/O area, and internal ROM/PROM area in the single-chip mode (access is restricted to external locations 100000H through FFE00H).

The MM register can be read/written in 8- or 1-bit units. Bit 4 of this register is set to 1.

	7	6	5	4	3	2	1	0
MM	0	0	0	1	MM3	MM2	MM1	MM0

Address: FFFFF04CH

At reset: 17H (in ROM-less mode), 10H (in single-chip mode)

Bit Position	Bit Name	Function
3	MM3	Memory Expansion Mode Specifies operation mode of P95 and P96 of port 9.
2-0	MM2-MM0	Memory Expansion Mode Specifies operation mode of ports 4, 5, 6, and 9 (P90-P94).

MM3:

MM3	Operation mode	P95	P96
0	Port mode	Port	
1	External expansion mode	ST0	ST1

MM2-MM0:

MM2	MM1	MM0	Address space	Port 4	Port 5	Port 6	Port 9 (P90-P94)
0	0	0	–	Port mode			
0	1	1	64-KB expansion	AD0-AD7	AD8-AD15		$\overline{LBEN}$, $\overline{UBEN}$, R/$\overline{W}$, $\overline{DSTB}$, ASTB
1	0	0	256-KB expansion			A16, A17	
1	0	1	1-MB expansion			A18, A19	
1	1	0	4-MB expansion			A20, A21	
1	1	1	16-MB expansion			A22, A23	
Others				RFU (reserved)			

Remark: For the details of the operation of each port pin, refer to **2.3 Pin Function**.

3.4.7 Recommended use of address space

The architecture of the V851 requires that a register that serves as a pointer be secured for address generation in addressing the memory. By optimizing the location of this pointer register, the maximum number of the general-purpose registers for variables can be reserved and the program size can be reduced (because the instruction that generates the pointer address is not necessary). ★

To enhance the efficiency of using the pointer in connection with the memory map of the V851, the following points are recommended:

(1) Program space

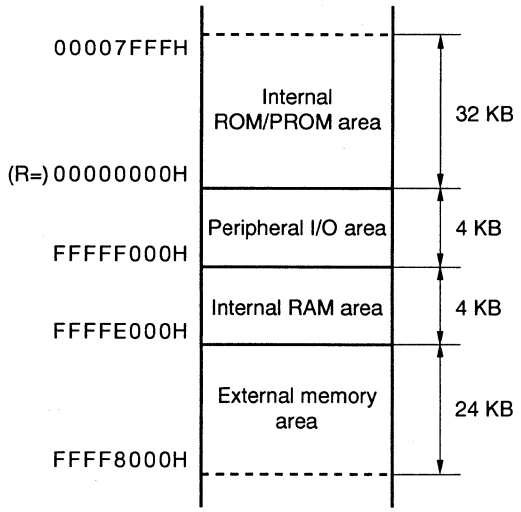
Of the 32 bits of the PC (program counter), the higher 8 bits are fixed to "0", and only the lower 24 bits are valid. Therefore, a contiguous 16-MB space, starting from address 00000000H, unconditionally corresponds to the memory map of the program space.

(2) Data space

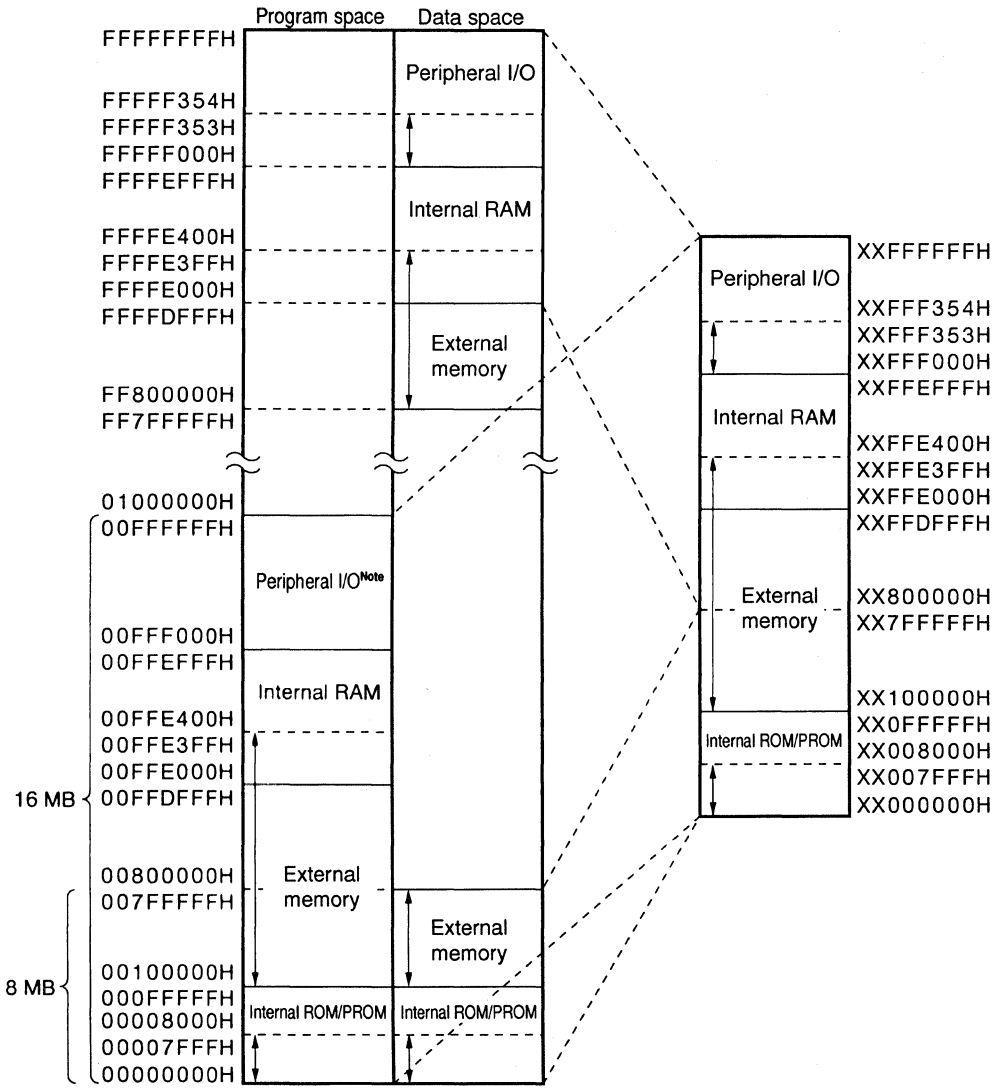
Efficient use of resources can be utilized through the wrap-around feature of the data space. The 16-MB of external address space can be mapped to the low 8-MB CPU address space (00000000F-007FFFFFH) and the high 8-MB CPU address space (FF800000H-FFFFFFFFH). Reference to address locations 00800000H-00FFFFFFH essentially remain the same, since the upper byte is ignored.

This mapping configuration, along with the use of the zero register (R0) in based addressing operations, allow for efficient access to internal resources.

For example, when R = r0 (zero register) is specified for the LD/ST disp16[R] instruction, an addressing range of 00000000H+/–32KB can be referenced with the sign-extended, 16-bit displacement value. Using the wrap-around mapping scheme described above, all resources including the internal ROM, RAM, and peripheral I/O can be accessed with one base register, which eliminates the need for additional base registers pointing to high memory locations.



★ **Figure 3-10. Recommended Memory Map**



Note: This area cannot be used as a program area.

Remark: The recommended area is indicated by the double-headed arrow (↕).

3.4.8 Peripheral I/O registers

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	O	O		Undefined
FFFFF002H	Port 1	P1		O	O		
FFFFF004H	Port 2	P2		O	O		
FFFFF006H	Port 3	P3		O	O		
FFFFF008H	Port 4	P4		O	O		
FFFFF00AH	Port 5	P5		O	O		
FFFFF00CH	Port 6	P6		O	O		
FFFFF012H	Port 9	P9		O	O		
FFFFF014H	Port 10	P10		O	O		
FFFFF020H	Port 0 mode register	PM0		O	O		FFH
FFFFF022H	Port 1 mode register	PM1		O	O		
FFFFF024H	Port 2 mode register	PM2		O	O		
FFFFF026H	Port 3 mode register	PM3		O	O		
FFFFF028H	Port 4 mode register	PM4		O	O		
FFFFF02AH	Port 5 mode register	PM5		O	O		
FFFFF02CH	Port 6 mode register	PM6		O	O		
FFFFF032H	Port 9 mode register	PM9		O	O		
FFFFF034H	Port 10 mode register	PM10		O	O		
FFFFF040H	Port 0 mode control register	PMC0		O	O		00H
FFFFF044H	Port 2 mode control register	PMC2		O	O		01H
FFFFF046H	Port 3 mode control register	PMC3		O	O		00H
FFFFF04CH	Memory expansion mode register	MM		O	O		10H/17H
FFFFF054H	Port 10 mode control register	PMC10		O	O		00H
FFFFF060H	Data wait control register	DWC				O	FFFFH
FFFFF062H	Bus cycle control register	BCC				O	AAAAH
FFFFF070H	Power save control register	PSC		O	O		00H
FFFFF078H	System status register	SYS		O	O		0000000XB
FFFFF084H	Baud rate generator register 0	BRG0		O	O		Undefined
FFFFF086H	Baud rate generator prescaler mode register 0	BPRM0		O	O		00H
FFFFF088H	Clocked serial interface mode register 0	CSIM0		O	O		
FFFFF08AH	Serial I/O shift register 0	SIO0		O	O		Undefined
FFFFF0C0H	Asynchronous serial interface mode register 00	ASIM00		O	O		80H
FFFFF0C2H	Asynchronous serial interface mode register 01	ASIM01		O	O		00H
FFFFF0C4H	Asynchronous serial interface status register 0	ASIS0	R	O	O		
FFFFF0C8H	Receive buffer 0 (9 bits)	RXB0				O	Undefined
FFFFF0CAH	Receive buffer 0L (lower 8 bits)	RXB0L		O	O		

Address	Function Register Name	Symbol	R/W	Bit Units for Manipulation			After reset
				1 bit	8 bits	16 bits	
FFFFF0CCH	Transmit shift register 0 (9 bits)	TXS0	W			O	Undefined
FFFFF0CEH	Transmit shift register 0L (lower 8 bits)	TXS0L			O		
FFFFF100H	Interrupt control register	OVIC1	R/W	O	O		47H
FFFFF102H	Interrupt control register	P1IC0		O	O		
FFFFF104H	Interrupt control register	P1IC1		O	O		
FFFFF106H	Interrupt control register	P1IC2		O	O		
FFFFF108H	Interrupt control register	P1IC3		O	O		
FFFFF10AH	Interrupt control register	CMIC4		O	O		
FFFFF10CH	Interrupt control register	CSIC0		O	O		
FFFFF10EH	Interrupt control register	SEIC0		O	O		
FFFFF110H	Interrupt control register	SRIC0		O	O		
FFFFF112H	Interrupt control register	STIC0		O	O		
FFFFF114H	Interrupt control register	P0IC0		O	O		
FFFFF116H	Interrupt control register	P0IC1		O	O		
FFFFF118H	Interrupt control register	P0IC2		O	O		
FFFFF11AH	Interrupt control register	P0IC3		O	O		
FFFFF166H	In-service priority register	ISPR	R	O	O		00H
FFFFF170H	Command register	PRCMD	W	O	O		Undefined
FFFFF180H	External interrupt mode register 0	INTM0	R/W	O	O		00H
FFFFF182H	External interrupt mode register 1	INTM1		O	O		
FFFFF184H	External interrupt mode register 2	INTM2		O	O		
FFFFF230H	Timer overflow status register	TOVS		O	O		
FFFFF240H	Timer unit mode register 1	TUM1				O	0000H
FFFFF242H	Timer control register 1	TMC1		O	O		00H
FFFFF244H	Timer output control register 1	TOC1		O	O		
FFFFF250H	Timer 1	TM1	R			O	0000H
FFFFF252H	Capture/compare register 10	CC10	R/W			O	Undefined
FFFFF254H	Capture/compare register 11	CC11				O	
FFFFF256H	Capture/compare register 12	CC12				O	
FFFFF258H	Capture/compare register 13	CC13				O	
FFFFF342H	Timer control register 4	TMC4		O	O		00H
FFFFF350H	Timer 4	TM4	R			O	0000H
FFFFF352H	Compare register 4	CM4	R/W			O	Undefined

CHAPTER 4 BUS CONTROL FUNCTION

The V851 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 data bus
- ○ External devices connected through multiplexed I/O port pins
- ○ Wait function
  - • Programmable wait function, capable of inserting up to 3 wait states per 2 blocks
  - • External wait control through $\overline{\text{WAIT}}$ input pin
- ○ Idle state insertion function
- ○ Bus mastership arbitration function
- ○ Bus hold function

4.2 Bus Control Pins

The following pins are used for interfacing to external devices:

External Bus Interface Function	Corresponding Port (pins)
Address/data bus (AD0-AD7)	Port 4 (P40-P47)
Address/data bus (AD8-AD15)	Port 5 (P50-P57)
Address bus (A16-A23)	Port 6 (P60-P67)
Read/write control ($\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$, R/$\overline{\text{W}}$, $\overline{\text{DSTB}}$)	Port 9 (P90-P93)
Address strobe (ASTB)	Port 9 (P94)
External wait control ($\overline{\text{WAIT}}$)	$\overline{\text{WAIT}}$
Bus cycle status (ST0, ST1)	Port 9 (P95-P96)
Bus hold control ($\overline{\text{HLDRQ}}$, $\overline{\text{HLDAK}}$)	Port 10 (P100-P101)

The bus interface function of each pin is enabled by the memory expansion mode register (MM). In ROM-less mode, the bus interface function of each pin is unconditionally enabled by the MODE0 and MODE1 inputs. For the details of specifying an operation mode of the external bus interface, refer to **3.4.6 (1) Memory expansion mode register (MM)**.

★ 4.3 Number of Access Clocks

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

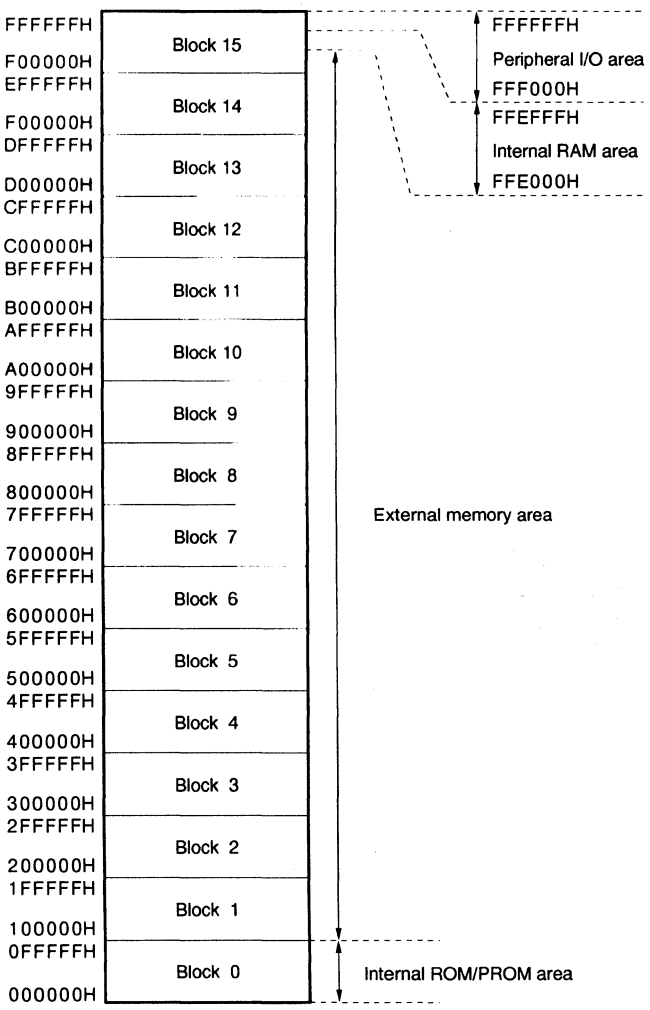
Bus Cycle Type	Resource (bus width)			
	Internal ROM/PROM (32 bits)	Internal RAM (32 bits)	Peripheral I/O (16 bits)	External Memory (16 bits)
Instruction fetch (continuous/normal mode)	1	3	Disable	3+n
Instruction fetch (branch)	1	3	Disable	3+n
Operand data access	3	1	3+n	3+n

Remarks: 1. Unit: clock/access

2. n: Number of wait states inserted

4.4 Memory Block Function

The 16-MB memory space is divided into memory blocks of 1-MB units. The programmable wait function and bus cycle operation mode can be independently controlled for every two memory blocks.



4.5 Wait Function

4.5.1 Programmable wait function

To facilitate interfacing with low-speed memories and I/O devices, up to 3 data wait states can be inserted in a bus cycle for two memory blocks. The number of wait states can be programmed by using data wait control register (DWC). Immediately after the system has been reset, three data wait states are automatically programmed for all memory blocks.

(1) Data wait control register (DWC)

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	At reset
DWC	DW71	DW70	DW61	DW60	DW51	DW50	DW41	DW40	DW31	DW30	DW21	DW20	DW11	DW10	DW01	DW00	FFFFF060H	FFFFH

Bit Position	Bit Name	Function
15-0	DWn1 DWn0 (n=0-7)	Data Wait Specifies number of wait states to be inserted

DWn1	DWn0	Number of wait states to be inserted
0	0	0
0	1	1
1	0	2
1	1	3

n	Blocks into which wait states are inserted
0	Blocks 0/1
1	Blocks 2/3
2	Blocks 4/5
3	Blocks 6/7
4	Blocks 8/9
5	Blocks 10/11
6	Blocks 12/13
7	Blocks 14/15

Cautions: 1. Block 0 is reserved for the internal ROM/PROM area in the single-chip mode. It is not subject to programmable wait control, regardless of the setting of DWC, and is always accessed without wait states.

2. The internal RAM area of block 15 is not subject to programmable wait control and is always accessed without wait states. The peripheral I/O area of this block is not subject to programmable wait control, either. The only wait control is dependent upon the execution of each peripheral function.

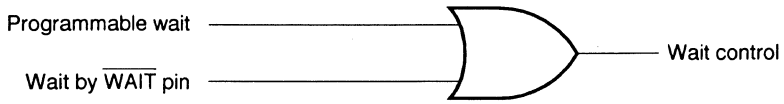
4.5.2 External wait function

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 sampling the external wait pin ($\overline{WAIT}$) to synchronize with the external device.

The external $\overline{WAIT}$ signal does not affect the access times of the internal ROM/PROM, internal RAM, and peripheral I/O areas. Input of the external $\overline{WAIT}$ signal can be done asynchronously to CLKOUT and is sampled at the falling edge of the clock in the T2 and TW states of a bus cycle. If the set-up and hold time of the $\overline{WAIT}$ input is not satisfied, the wait state may or may not be inserted in the next state.

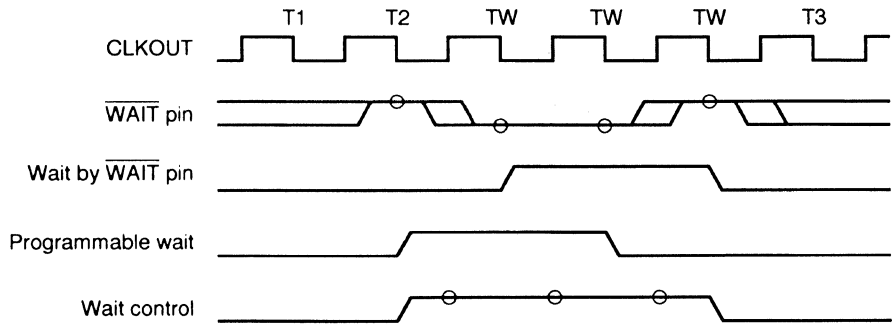
4.5.3 Relations between programmable wait and external wait

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 $\overline{WAIT}$ pin. In other words, the number of wait cycles is determined by the programmable wait value or the length of evaluation at the $\overline{WAIT}$ input pin.



For example, if the number of programmable wait states is 2 and the timing of the $\overline{WAIT}$ 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



Remark: ○ : sampling timing

4.6 Idle State Insertion Function

To facilitate interfacing with low-speed memory devices and meeting the data output float delay time (t_{DF}) on memory read accesses, one idle state (TI) can be inserted into the current bus cycle after the T3 state. The bus cycle following continuous bus cycles starts after one idle state.

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

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

(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	Address	At reset
BCC	BC71	0	BC61	0	BC51	0	BC41	0	BC31	0	BC21	0	BC11	0	BC01	0	FFFFF062H	AAAAH

Bit Position	Bit Name	Function
15, 13, 11, 9, 7, 5, 3, 1	BCn1 (n=0-7)	Bus Cycle Specifies insertion of idle state. 0: Not inserted 1: Inserted

n	Blocks into Which Idle State Is Inserted
0	Blocks 0/1
1	Blocks 2/3
2	Blocks 4/5
3	Blocks 6/7
4	Blocks 8/9
5	Blocks 10/11
6	Blocks 12/13
7	Blocks 14/15

Cautions: 1. Block 0 is reserved for the internal ROM/PROM area in the single-chip mode and therefore, no idle state can be inserted into this block.

2. The internal RAM area and peripheral I/O area of block 15 are not subject to insertion of the idle state.

3. Be sure to set bits 0, 2, 4, 6, 8, 10, 12, and 14 to 0. If 1 is set, the operation is not guaranteed. ★

4.7 Bus Hold Function

4.7.1 Outline of function

When P100 and P101 of port 10 are programmed to be in the control mode, the functions of the $\overline{\text{HLDRQ}}$ and $\overline{\text{HLDAK}}$ pins become valid.

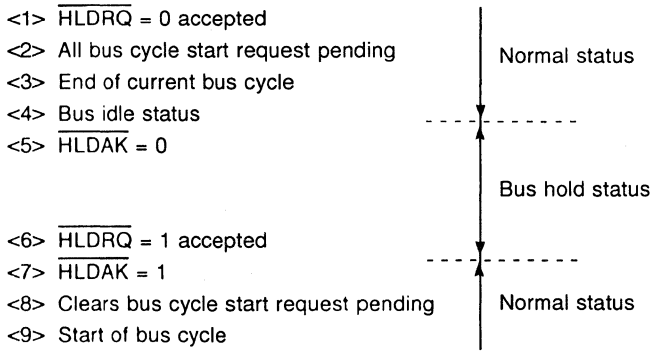
When the $\overline{\text{HLDRQ}}$ pin becomes active (low) indicating that other bus master is requesting for acquisition of the bus, the external address/data bus and strobe pins go into a high-impedance state, and the bus is released (bus hold status). When the $\overline{\text{HLDRQ}}$ pin becomes inactive (high) indicating that the request for the bus is cleared, these pins are driven again.

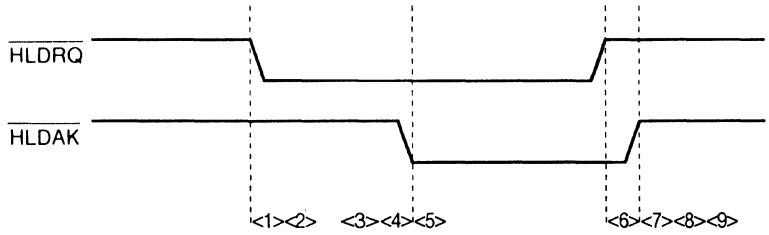
In the bus hold status, the $\overline{\text{HLDAK}}$ pin becomes active (low).

This feature can be used to design a system where two or more bus masters exist, such as when multi-processor configuration is used and when a DMA controller is connected.

4.7.2 Bus hold procedure

The procedure of bus hold function is illustrated below.





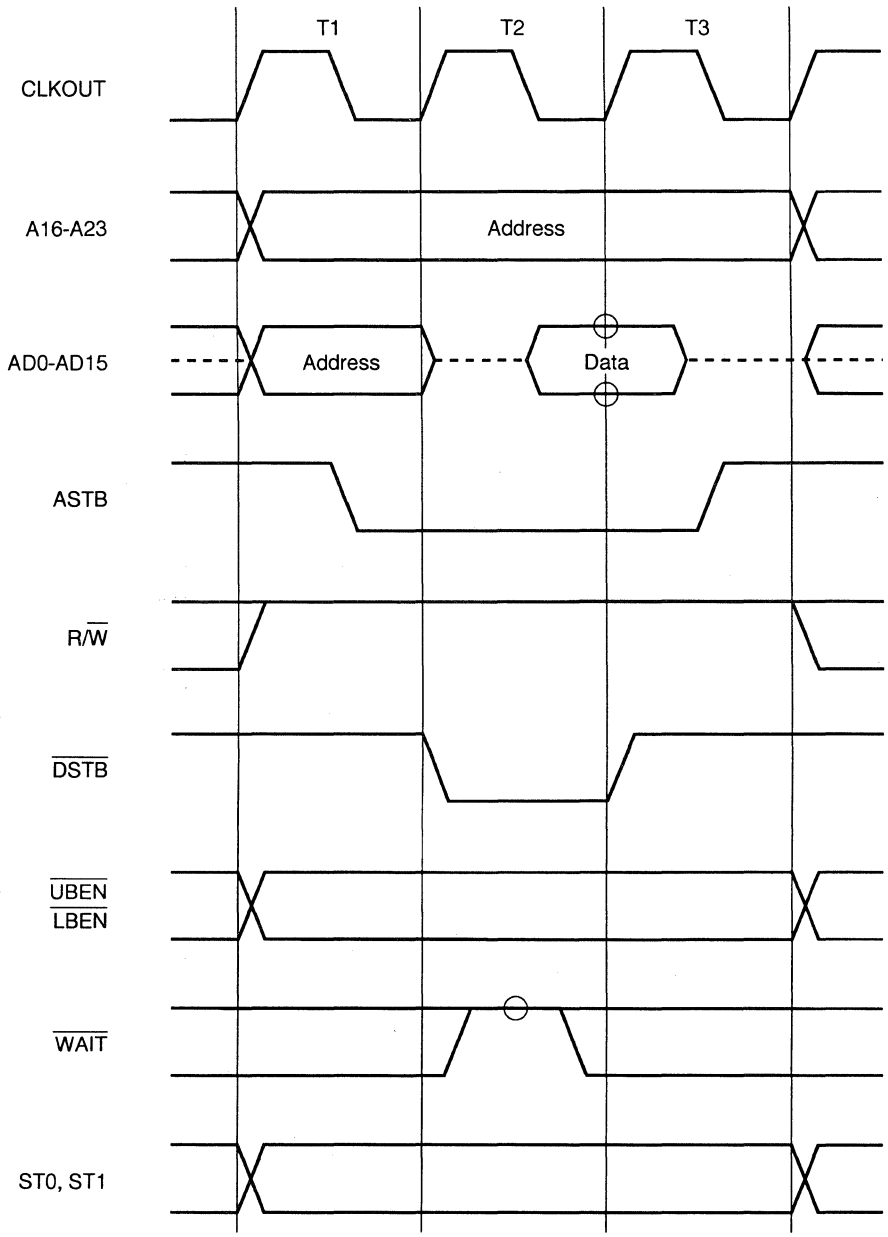
4.7.3 Operation in power save mode

In the STOP or IDLE mode, the system clock is stopped. Consequently, the bus hold status is not 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 status is set. When the $\overline{\text{HLDRQ}}$ pin becomes inactive, the $\overline{\text{HLDAK}}$ pin becomes inactive. As a result, the bus hold status is cleared, and the HALT mode is set again.

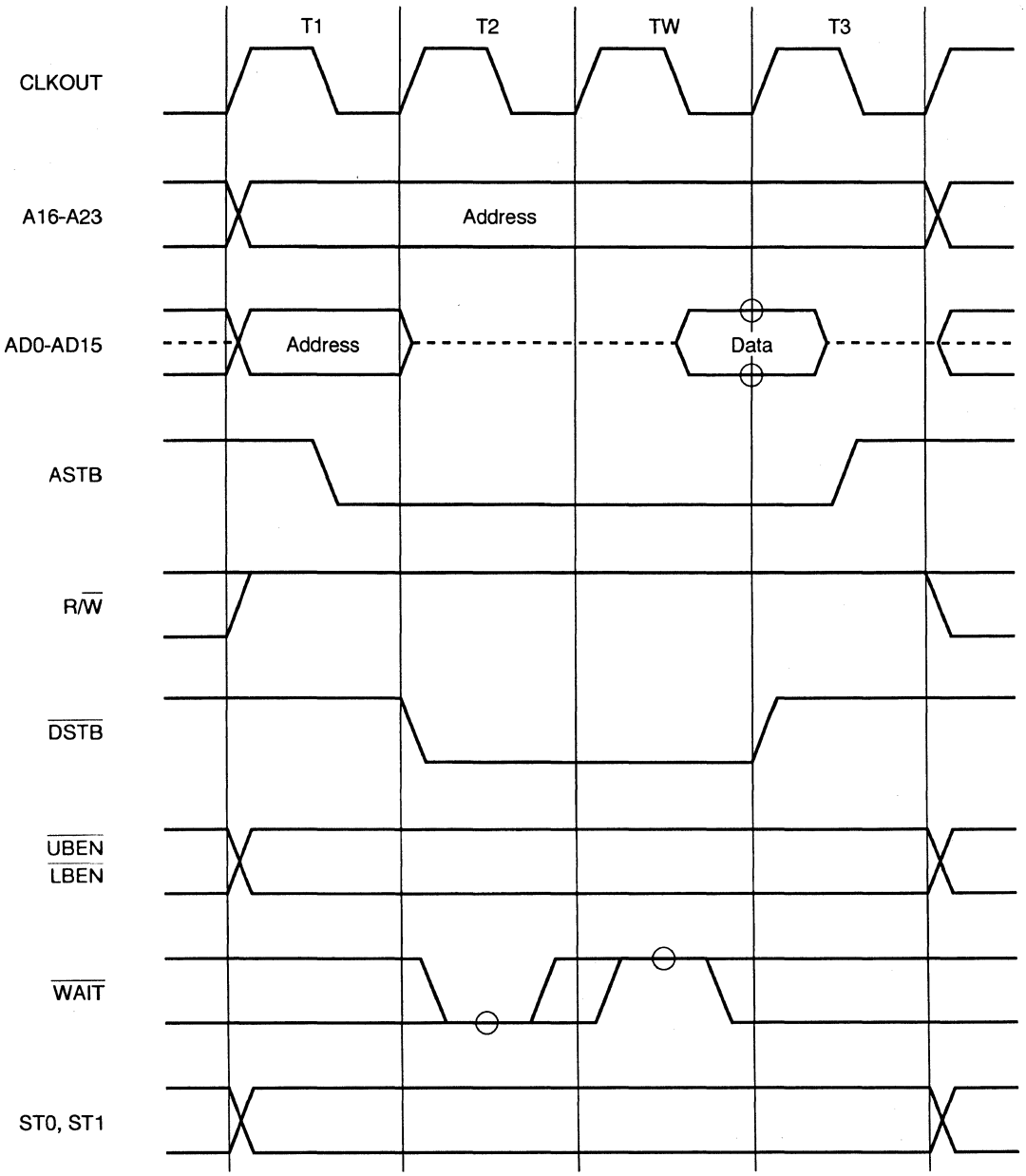
4.8 Bus Timing

(1) Memory read (0 wait)



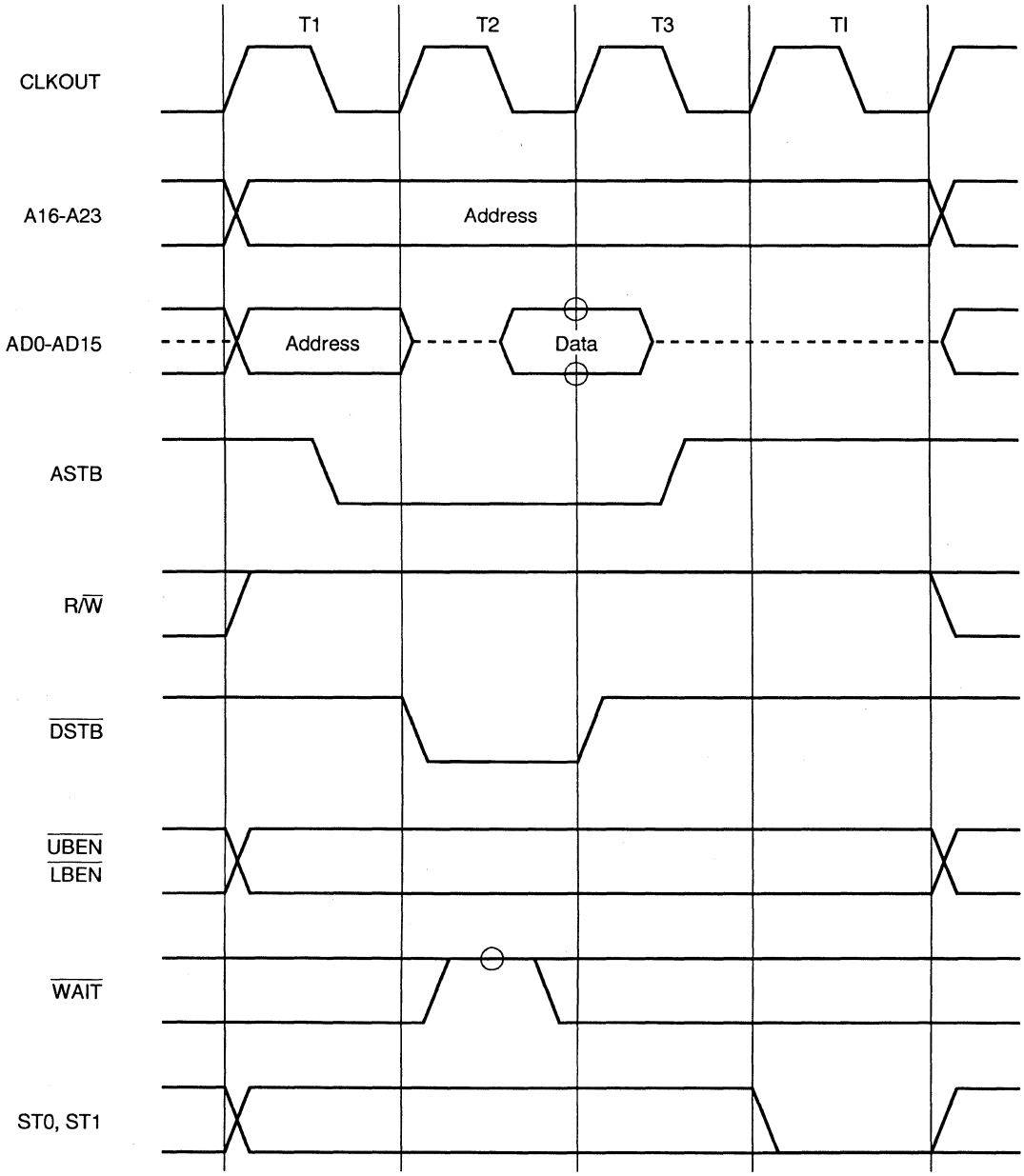
Remarks: **1.** ○ indicates the sampling timing when the number of programmable waits is set to 0.
2. The dotted line indicates the high-impedance state.

(2) Memory read (1 wait)



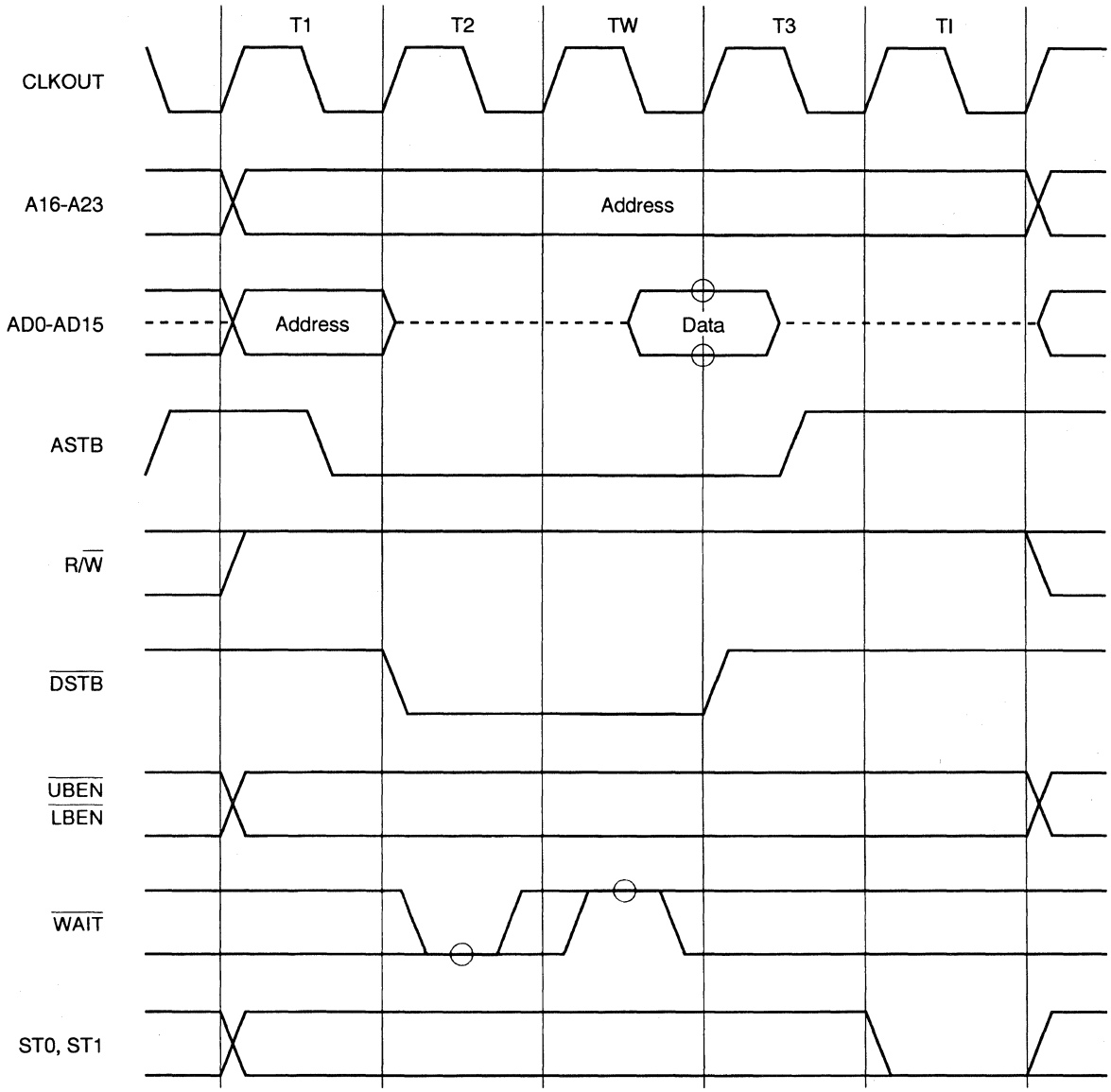
Remarks: **1.** ○ indicates the sampling timing when the number of programmable waits is set to 0.
2. The dotted line indicates the high-impedance state.

(3) Memory read (0 wait, idle state)



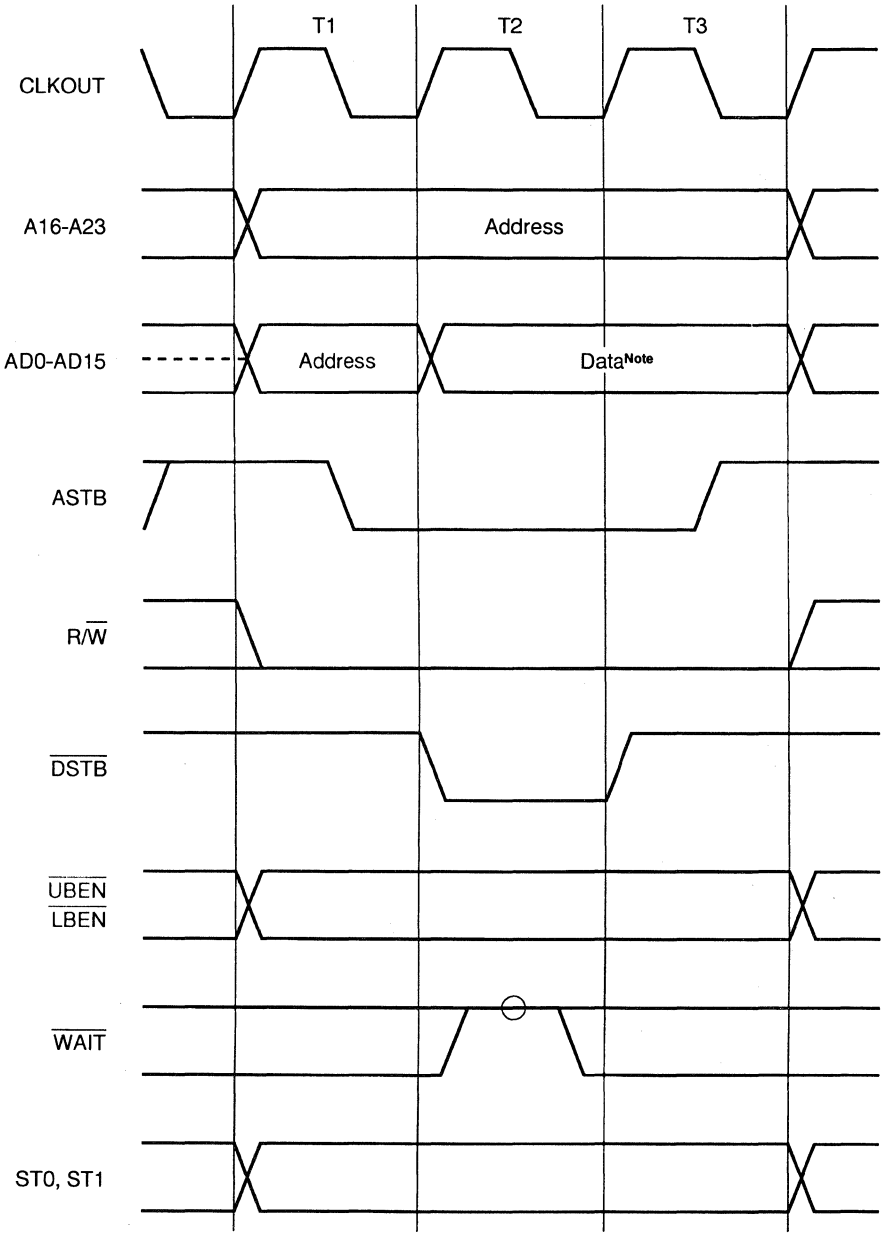
Remarks: **1.** ○ indicates the sampling timing when the number of programmable waits is set to 0.
2. The dotted line indicates the high-impedance state.

(4) Memory read (1 wait, idle state)



Remarks: **1.** ○ indicates the sampling timing when the number of programmable waits is set to 0.
2. The dotted line indicates the high-impedance state.

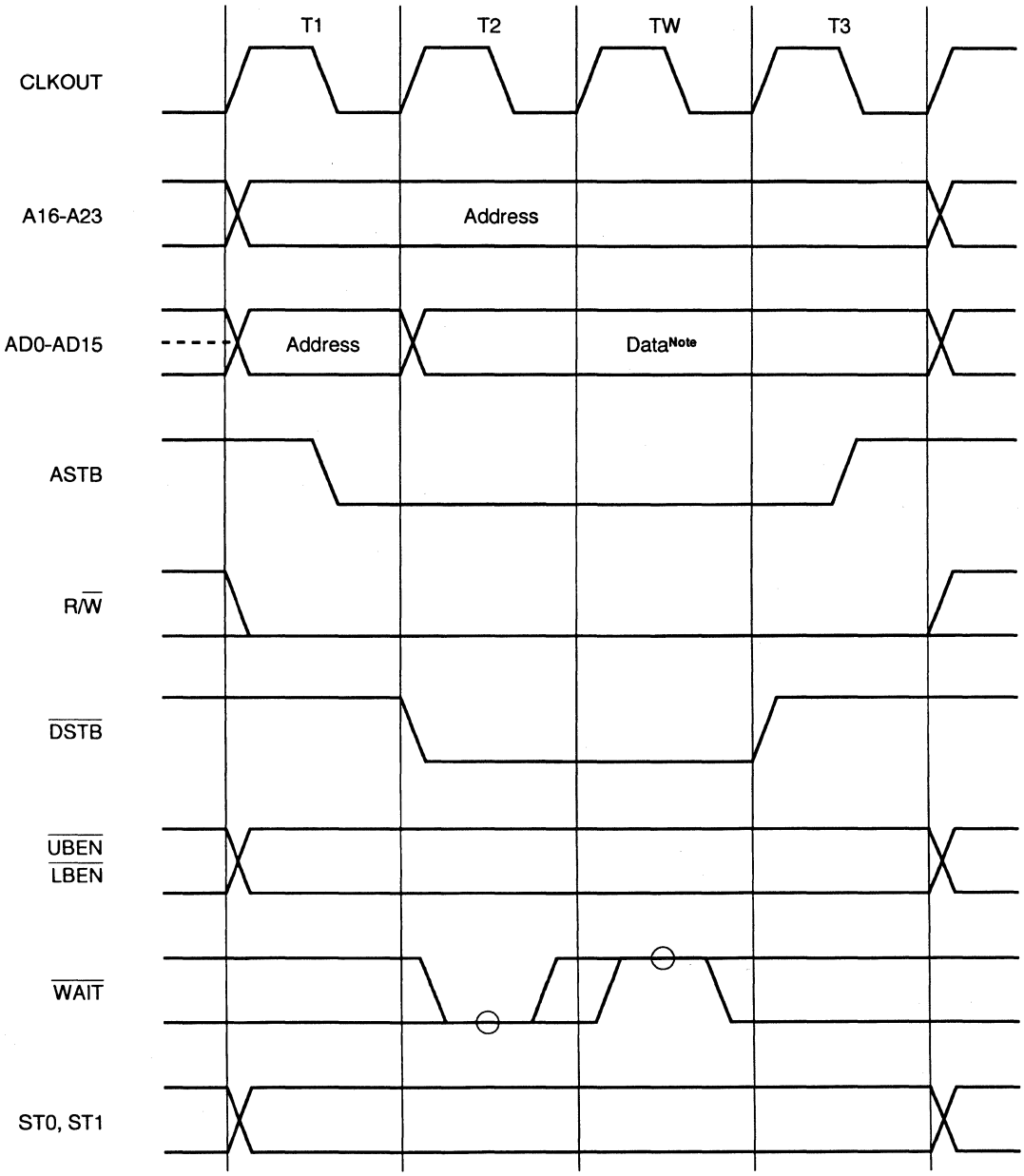
(5) Memory write (0 wait)



Note: AD0-AD7 output invalid data when odd address byte data is accessed.
AD8-AD15 output invalid data when even address byte data is accessed.

Remarks: **1.** ○ indicates the sampling timing when the number of programmable waits is set to 0.
2. The dotted line indicates the high-impedance state.

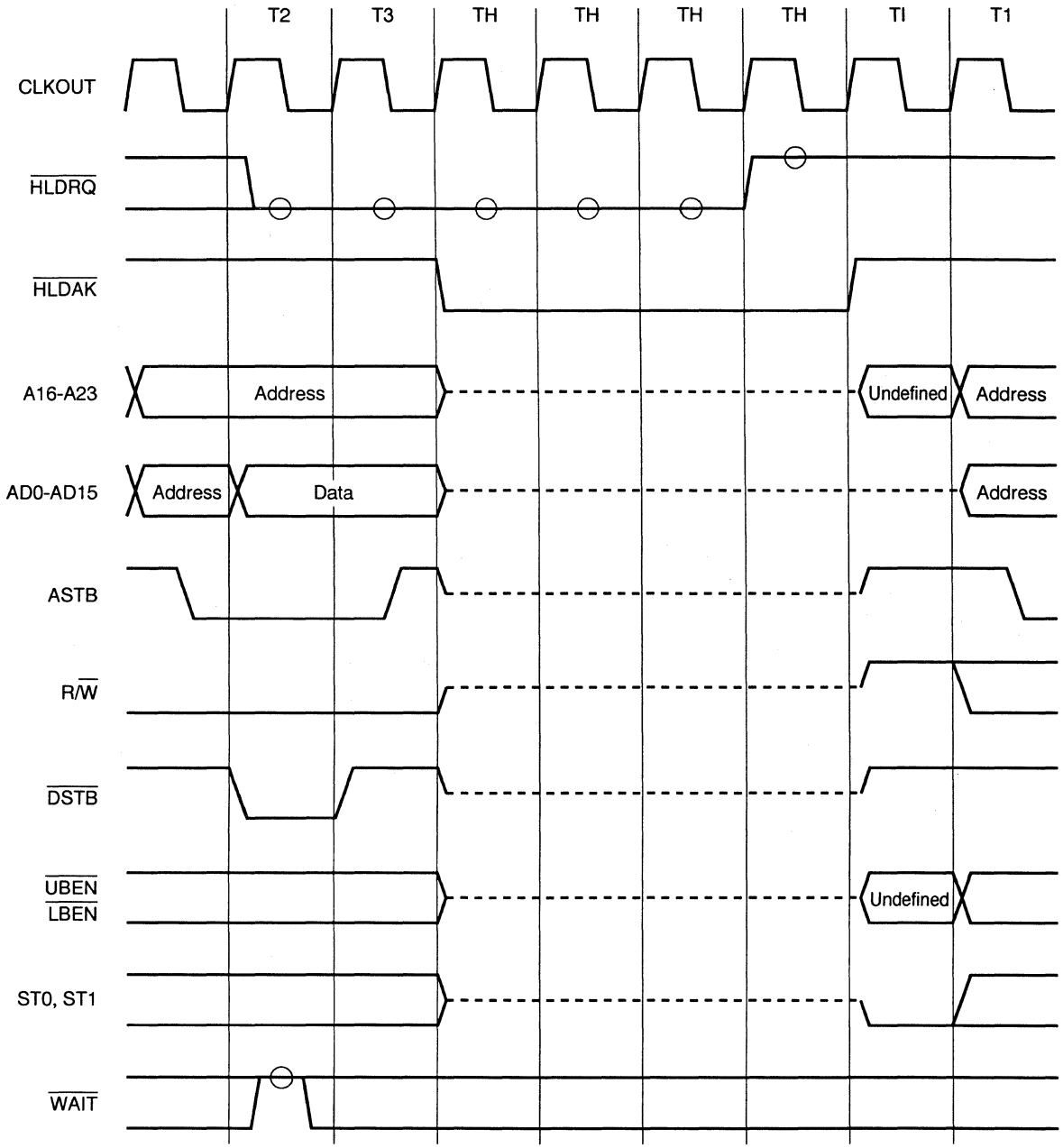
(6) Memory write (1 wait)



Note: AD0-AD7 output invalid data when odd address byte data is accessed.
AD8-AD15 output invalid data when even address byte data is accessed.

Remarks: **1.** ○ indicates the sampling timing when the number of programmable waits is set to 0.
2. The dotted line indicates the high-impedance state.

(7) Bus hold timing



Remarks: **1.** O indicates the sampling timing.

2. The dotted line indicates the high-impedance state.

4.9 Bus Priority

There are four external bus cycles: bus hold, operand data access, instruction fetch (branch), and instruction fetch (continuous). The bus hold cycle is given the highest priority, followed by operand data access, instruction fetch (branch), and instruction fetch (continuous) in that order.

The instruction fetch cycle may be inserted in between the read access and write access of read-modify-write access.

No instruction fetch cycle is inserted between the lower half-word access and higher half-word access of word operations.

Table 4-1. Bus Priority

External Bus Cycle	Priority
Bus hold	1
Operand data access	2
Instruction fetch (branch)	3
Instruction fetch (continuous)	4

4.10 Memory Boundary Operation Condition

4.10.1 Program space

(1) Do not execute branch to the peripheral I/O area or continuous fetch from the internal RAM area. Of course, it is impossible to fetch from external memory. If it is executed nevertheless, the NOP instruction code is continuously fetched.

(2) A prefetch operation straddling over the peripheral I/O area (invalid fetch) does not take place if a branch instruction exists at the upper-limit address of the internal RAM area.

4.10.2 Data space

Only the address aligned at the half-word (when the least significant bit of the address is "0")/word (when the lowest 2 bits of the address are "0") boundary is accessed for data half-word (16 bits)/word (32 bits) long.

Therefore, access that straddles over the memory or memory block boundary does not take place. For more details, refer to section **3.3 "Data Alignment"** of **V850 User's Manual —Architecture—**.

4.11 Internal Peripheral I/O Interface

Access to the internal peripheral I/O area is not output to the external bus. Therefore, the internal peripheral I/O area can be accessed in parallel with instruction fetch access.

Accesses to the internal peripheral I/O area takes, in most cases, three clock cycles. However accesses to the following timer/counter registers may take from 3 to 4 cycles.

Peripheral I/O Register	Access
TM1	Read
TM4	
CC10	Read/write
CC11	
CC12	
CC13	
CM4	Write

[MEMO]

CHAPTER 5 INTERRUPT/EXCEPTION PROCESSING FUNCTION

The V851 is provided with a dedicated interrupt controller (INTC) for interrupt processing and can process a total of 15 interrupt requests.

An interrupt is an event that occurs independently of program execution, and an exception is an event that occurs dependently on program execution. Generally, an exception takes precedence over an interrupt.

The V851 can process interrupt requests from the internal peripheral hardware and external sources. Moreover, exception processing can be started by an TRAP instruction (software exception) or by generation of an exception event (fetching of an illegal op code).

5.1 Features

○ Interrupt
- Non-maskable interrupt: 1 source
- Maskable interrupt : 14 sources
- 8 levels programmable priorities
- Multiple interrupt control according to priority
- Each maskable interrupt can be individually disabled.
- Rising and/or falling edge of external interrupt request signal can be specified.

○ Exception
- Software exception: 32 sources
- Exception trap : 1 source (illegal op code exception)

These interrupt/exception sources are listed in Table 5-1.

Table 5-1. Interrupt List

Type	Classification	Interrupt/Exception Source				Default Priority	Exception Code	Vector Address	Restored PC
		Name	Control Register	Generating Source	Generating Unit				
Reset	Interrupt	RESET	–	Reset input	–	–	0000H	00000000H	Undefined
Non-maskable	Interrupt	NMI	–	NMI input	–	–	0010H	00000010H	nextPC
Software exception	Exception	TRAP0n[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	INTOV1	OVIC1	Timer 1 overflow	RPU	0	0080H	00000080H	nextPC
	Interrupt	INTP10/INTCC10	P1IC0	INTP10 pin/CC10 coincidence	Pin/RPU	1	0090H	00000090H	nextPC
	Interrupt	INTP11/INTCC11	P1IC1	INTP11 pin/CC11 coincidence	Pin/RPU	2	00A0H	000000A0H	nextPC
	Interrupt	INTP12/INTCC12	P1IC2	INTP12 pin/CC12 coincidence	Pin/RPU	3	00B0H	000000B0H	nextPC
	Interrupt	INTP13/INTCC13	P1IC3	INTP13 pin/CC13 coincidence	Pin/RPU	4	00C0H	000000C0H	nextPC
	Interrupt	INTCM4	CMIC4	CM4 coincidence	RPU	5	00D0H	000000D0H	nextPC
	Interrupt	INTCSI0	CSIC0	CSI0 transmission/ reception completion	SIO	6	00E0H	000000E0H	nextPC
	Interrupt	INTSER0	SEIC0	UART0 reception error	SIO	7	00F0H	000000F0H	nextPC
	Interrupt	INTSR0	SRIC0	UART0 reception completion	SIO	8	0100H	00000100H	nextPC
	Interrupt	INTST0	STIC0 completion	UART0 transmission	SIO	9	0110H	00000110H	nextpc
	Interrupt	INTP00	P0IC0	INTP00 pin	Pin	10	0120H	00000120H	nextPC
	Interrupt	INTP01	P0IC1	INTP01 pin	Pin	11	0130H	00000130H	nextPC
	Interrupt	INTP02	P0IC2	INTP02 pin	Pin	12	0140H	00000140H	nextPC
	Interrupt	INTP03	P0IC3	INTP03 pin	Pin	13	0150H	00000150H	nextPC

Note: n: value of 0-FH

Remarks: 1. Default Priority: Priority that takes precedence when two or more maskable interrupt requests with the same priority level 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 saved when an interrupt is acknowledged during the DIVH (division) instruction execution is the value of the PC of the current instruction (DIVH).

2. The execution address of the illegal instruction when an illegal op code exception occurs is calculated as follows: (Restored PC – 4)

5.2 Non-Maskable Interrupt

The non-maskable interrupt is accepted unconditionally, even when interrupts are disabled (DI states) in the interrupt disabled (DI) status. The NMI is not subject to priority control and takes precedence over all the other interrupts.

The non-maskable interrupt request is input from the NMI pin. When the valid edge specified by the bit 0 (ESN0) of the external interrupt mode register 0 (INTM0) is detected on the NMI pin, the interrupt occurs.

While the service routine of the non-maskable interrupt is being executed, (PSW.NP = 1), the acceptance of another non-maskable interrupt request is kept pending. The pending NMI is accepted after the original service routine 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 routine for an NMI, the number of NMIs that will be acknowledged after PSW.NP goes to "0", is only one.

5.2.1 Accepting operation

If the non-maskable interrupt is generated by NMI input, the CPU performs the following processing, and transfers control to the handler routine:

(1) Saves the contents of PC to FEPC.
(2) Saves the current PSW to FEPSW.
(3) Writes exception code 0010H to the higher half-word (FECC) of ECR.
(4) Sets the NP and ID bits of PSW and clears the EP bit.
(5) Loads the vector address (00000010H) of the non-maskable interrupt routine to the PC, and transfers control.

Figure 5-1 illustrates how the non-maskable interrupt is processed.

Figure 5-1. Non-Maskable Interrupt Processing

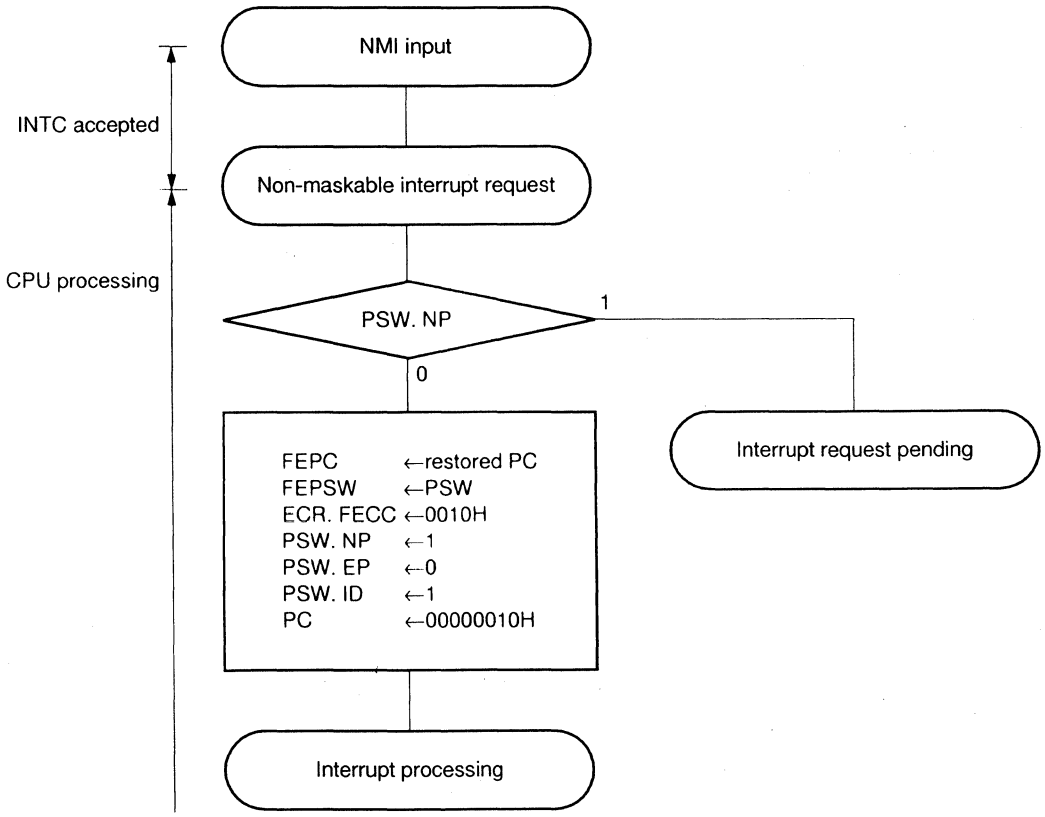
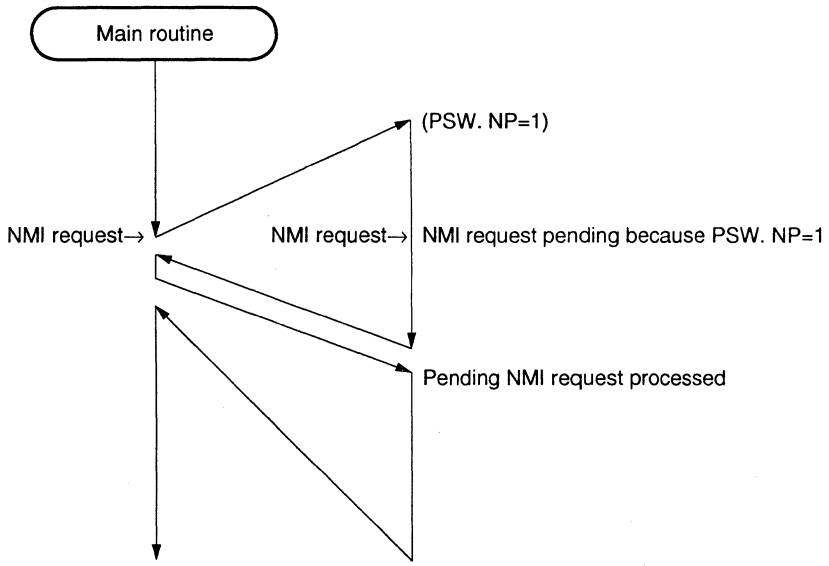
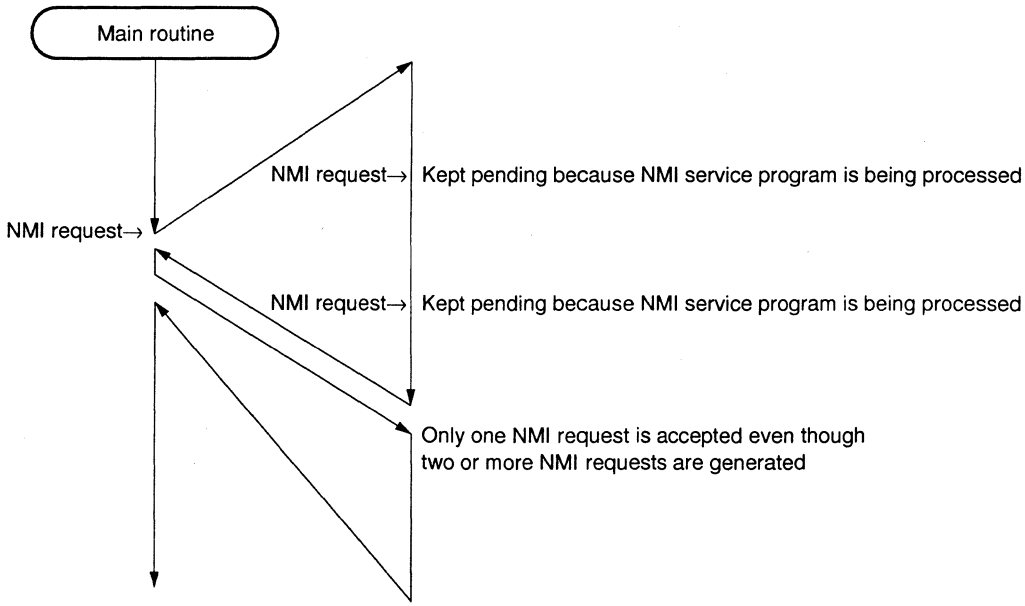


Figure 5-2. Accepting Non-Maskable Interrupt Request

(a) If a new NMI request is generated while an NMI service routine is executing:



(b) If a new NMI request is generated twice while an NMI service routine is executing:



5.2.2 Restore operation

Execution is restored from the non-maskable interrupt processing by the RETI instruction.

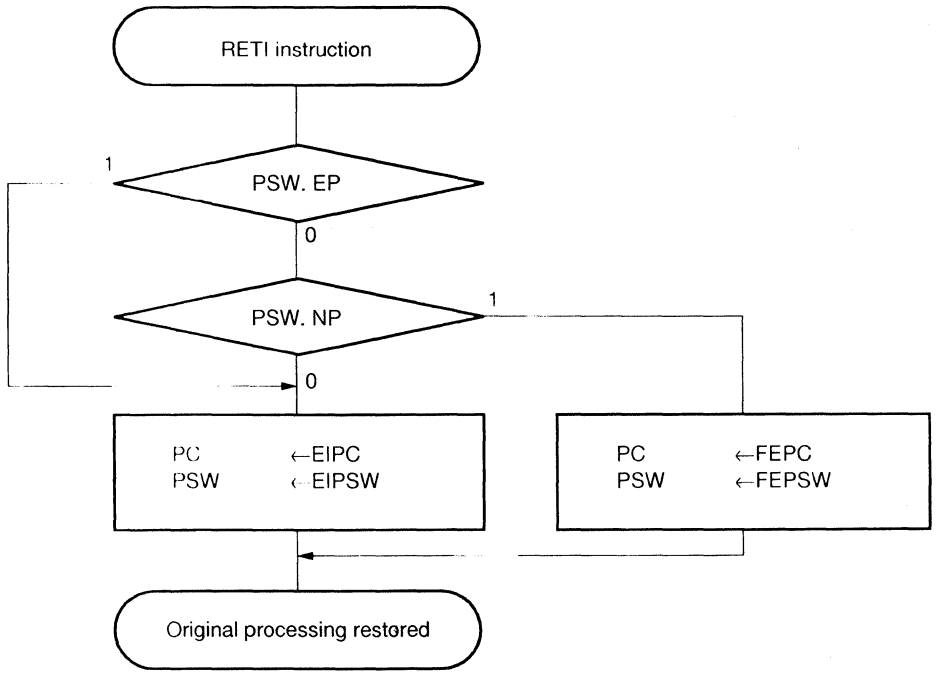
Operation of RETI instruction

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 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 5-3 illustrates how the RETI instruction is processed.

Figure 5-3. RETI Instruction Processing



★ **Caution: If the PSW.EP and PSW.NP bits are changed using the LDSR instruction during non-maskable interrupt service, it is necessary to reset PSW.EP to 0 and set PSW.NP to 1 using the LDSR instruction immediately before the RETI instruction to make sure that the PC and PSW are normally restored by the RETI instruction.**

5.2.3 External interrupt mode register 0 (INTM0)

INTM0 is a register that specifies the valid edge of the non-maskable interrupt (NMI). The valid edge of NMI can be specified as the rising or falling edge by the ESN0 bit of this register.

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

	7	6	5	4	3	2	1	0	Address	At reset
INTM0	0	0	0	0	0	0	0	ESN0	FFFFF180H	00H

Bit Position	Bit Name	Function
0	ESN0	Edge Select NMI Specifies valid edge of NMI pin 0: Falling edge 1: Rising edge

5.2.4 NP flag

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 accepted, and masks all interrupt requests to prohibit multiple interrupts from being acknowledged.

	31																				8	7	6	5	4	3	2	1	0	At reset
PSW	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	00000020H

Bit Position	Bit Name	Function
7	NP	NMI Pending Indicates that NMI interrupt processing is under execution 0: No NMI interrupt processing 1: NMI interrupt currently processing

5.3 Maskable Interrupts

Maskable interrupt requests can be masked by interrupt control registers. The V851 has 14 maskable interrupt sources.

If two or more maskable interrupt requests are generated at the same time, they are accepted according to the default priority. In addition to the default priority, eight levels of priorities can be specified by using the interrupt control registers, allowing programmable priority control.

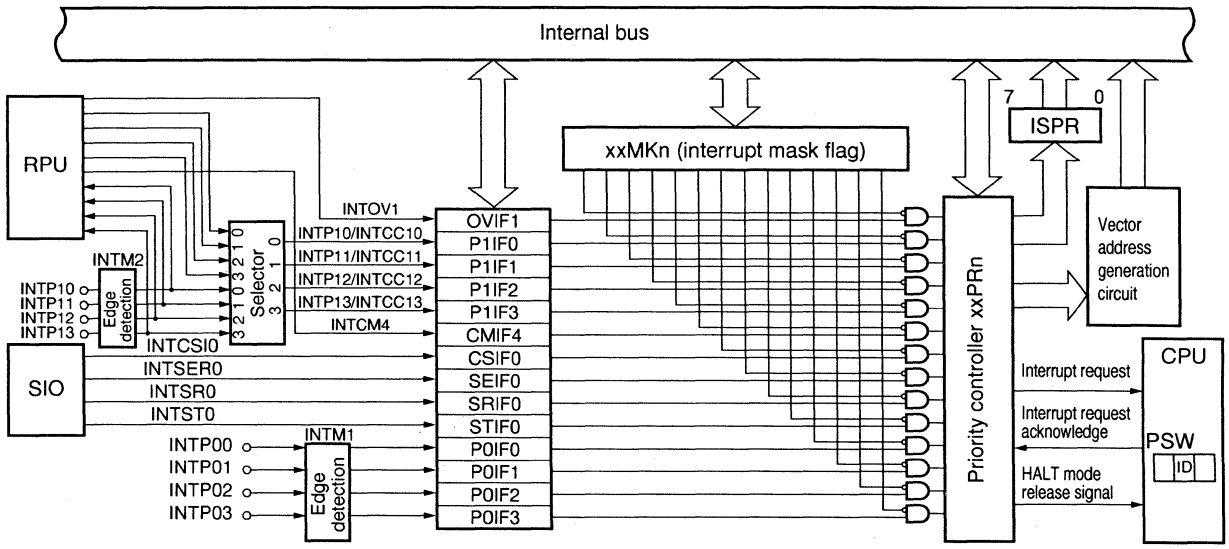
When an interrupt request has been acknowledged, the acceptance of other maskable interrupts is disabled and the interrupt disabled (DI) status is set.

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 to immediately interrupt the current service routine in progress. Note that only interrupts with a higher priority will have this capability; interrupts with the same priority level cannot be nested.

To use multiple interrupts, it is necessary to save EIPC and EIPSW to memory or a register before executing the EI instruction, and restore EIPC and EIPSW to the original values by executing the DI instruction before the RETI instruction.

5.3.1 Block diagram

Figure 5-4. Maskable Interrupt Block Diagram



Remark: xx: identification name of each peripheral unit
n : peripheral unit number

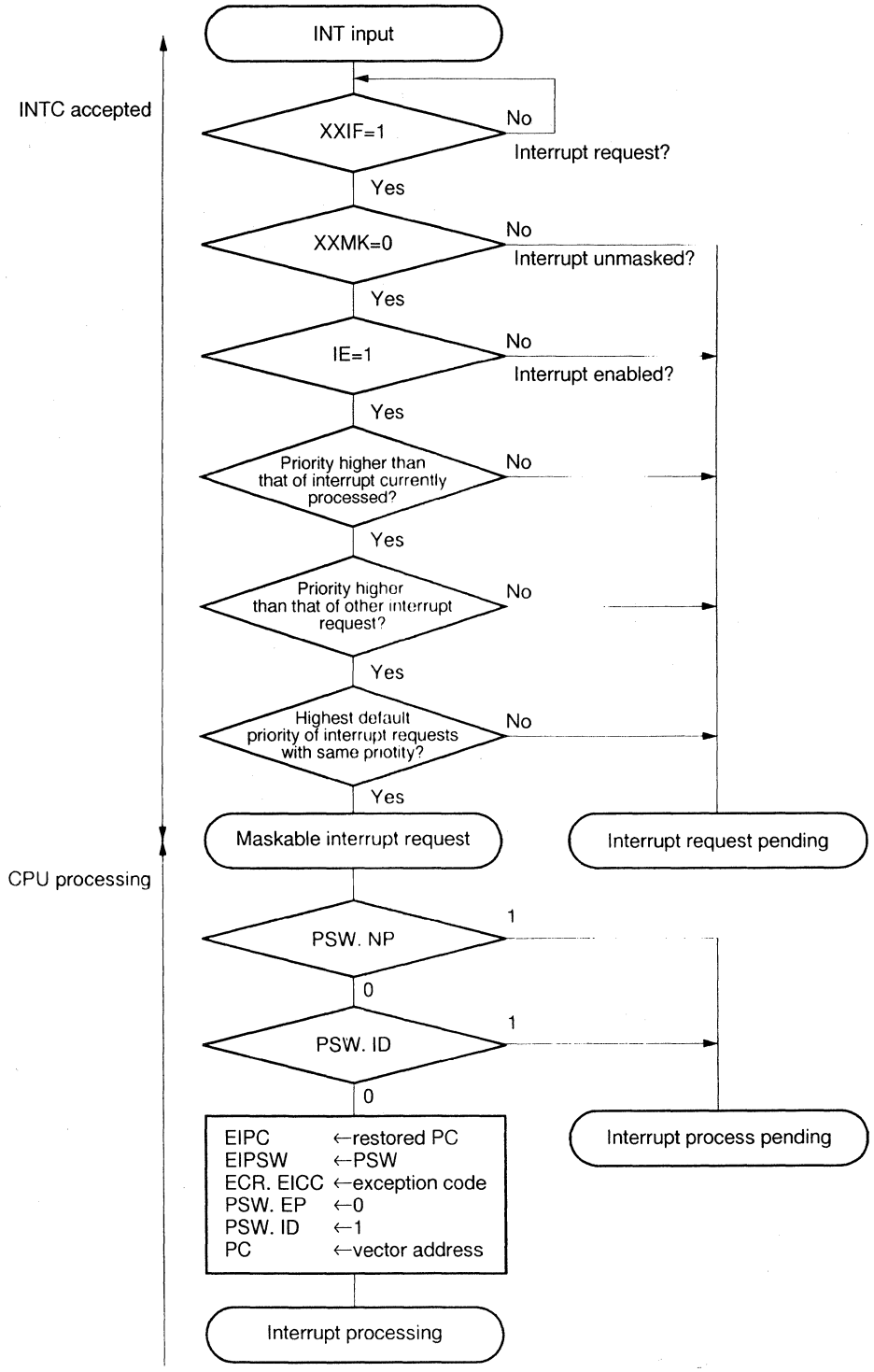
5.3.2 Operation

If a maskable interrupt occurs, the CPU performs the following processing, and transfers control to a vector routine:

(1) Saves the value of PC to EIPC.
(2) Saves the current PSW to EIPSW.
(3) Writes an exception code to the lower half-word of ECR (EICC).
(4) Sets the ID bit of PSW and clears the EP bit.
(5) Loads the corresponding vector address to the PC, and transfers control.

Figure 5-5 illustrates how the maskable interrupts are processed.

Figure 5-5. Maskable Interrupt Processing



The INT input masked by the interrupt control registers and the automatic interrupt mask that occurs while a previous interrupt is being processed (when PSW.NP = 1 or PSW.ID = 1) are internally monitored by the interrupt controller. When the interrupts are unmasked, or when PSW.NP = 0 and PSW.ID = 0 by using the RETI and LDSR instructions, the pending maskable interrupts can then be acknowledge, by priority, and processed.

5.3.3 Restore

To restore or return execution from the maskable interrupt service routine, the RETI instruction is used.

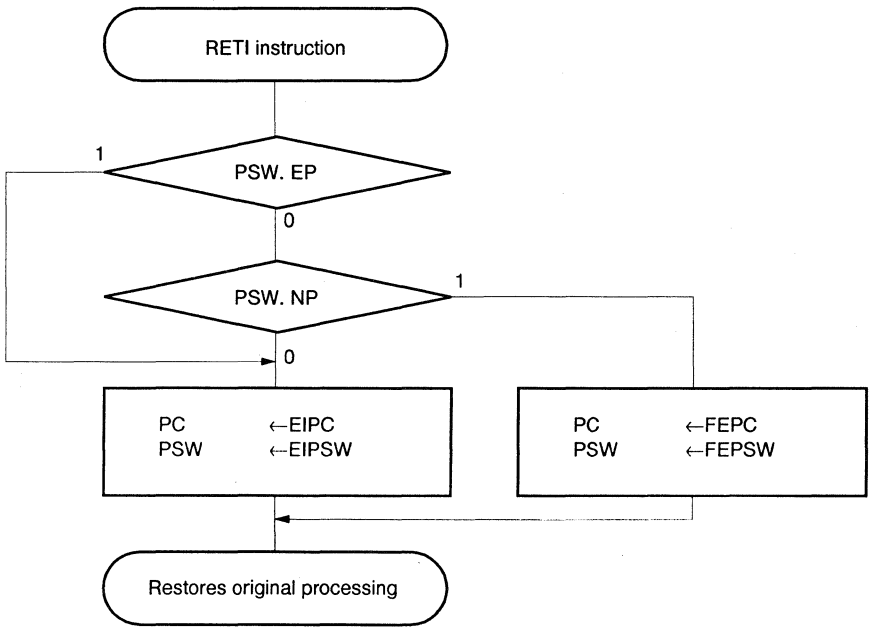
Operation of RETI instruction

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 PC and PSW from EIPC and EIPSW because the EP bit of PSW is 0 and the NP bit of PSW is 0.
(2) Transfers control to the address of the restored PC and PSW.

Figure 5-6 illustrates the processing of the RETI instruction.

Figure 5-6. RETI Instruction Processing



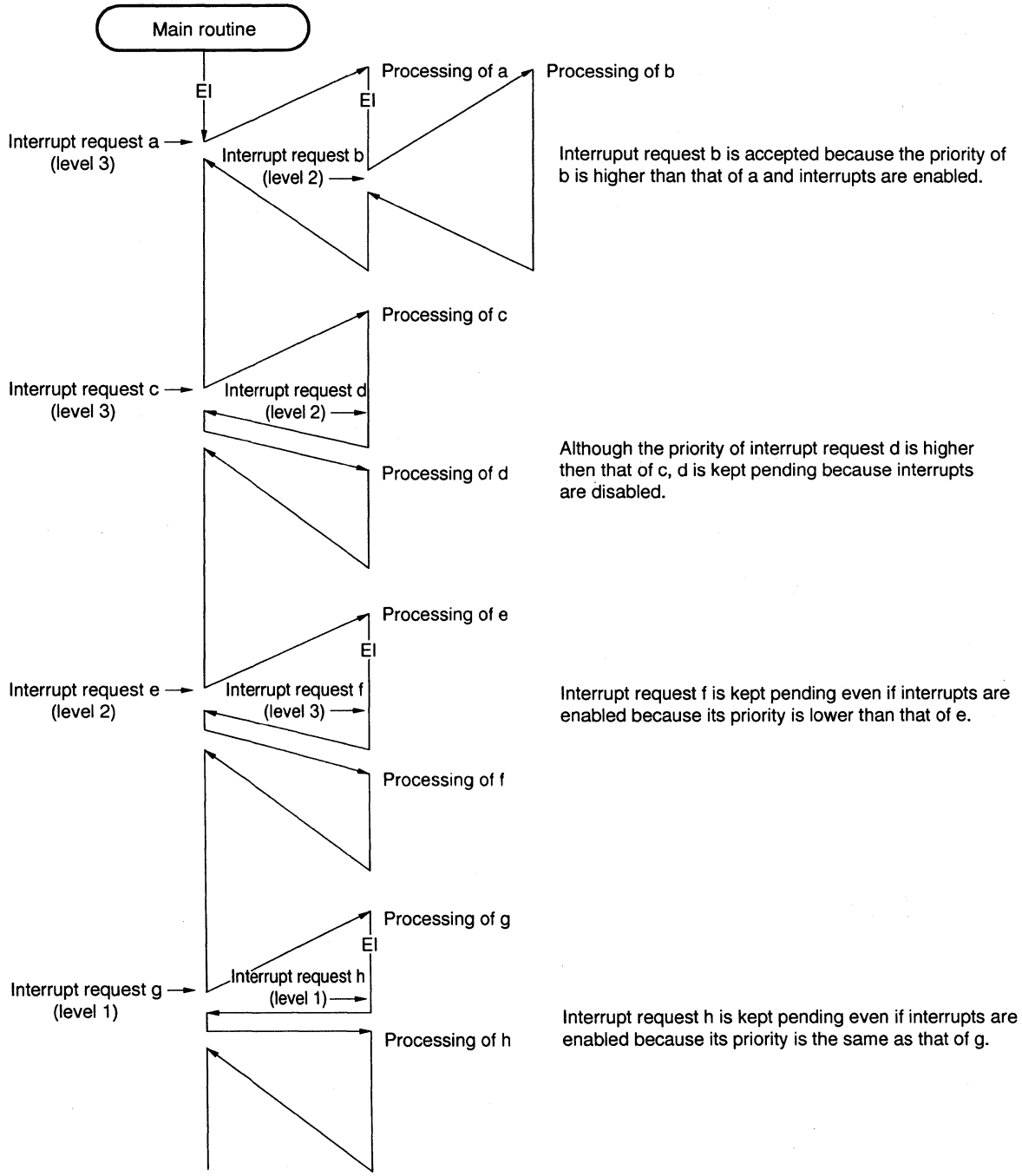
Caution: If the PSW.EP and PSW.NP bits are changed using the LDSR instruction during non-maskable interrupt service, it is necessary to reset PSW.EP to 0 and set PSW.NP to 1 using the LDSR instruction immediately before the RETI instruction to make sure that the PC and PSW are normally restored by the RETI instruction. ★

5.3.4 Priorities of maskable interrupts

There are two priority control criteria in the V851: control based on the default priority levels, and control based on programmable priority levels. The default priority levels are specified by default for each interrupt request type. The programmable priority is customized into eight levels by setting the priority specification flags (×PRn2 to PRn0, refer to the **table** in section **5.3.5**). The programmable priority levels override the default priority levels. Therefore, the order in which interrupts are serviced normally depends on each programmable priority levels. When two or more interrupts with the same programmable priority level occurred at the same time, the interrupt with the higher or highest default priority level will be serviced first. For more information, refer to **Table 5-1**.

Note that when an interrupt is acknowledged, the ID flag of 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 5-7. Example of Interrupt Nesting Process (1/2)

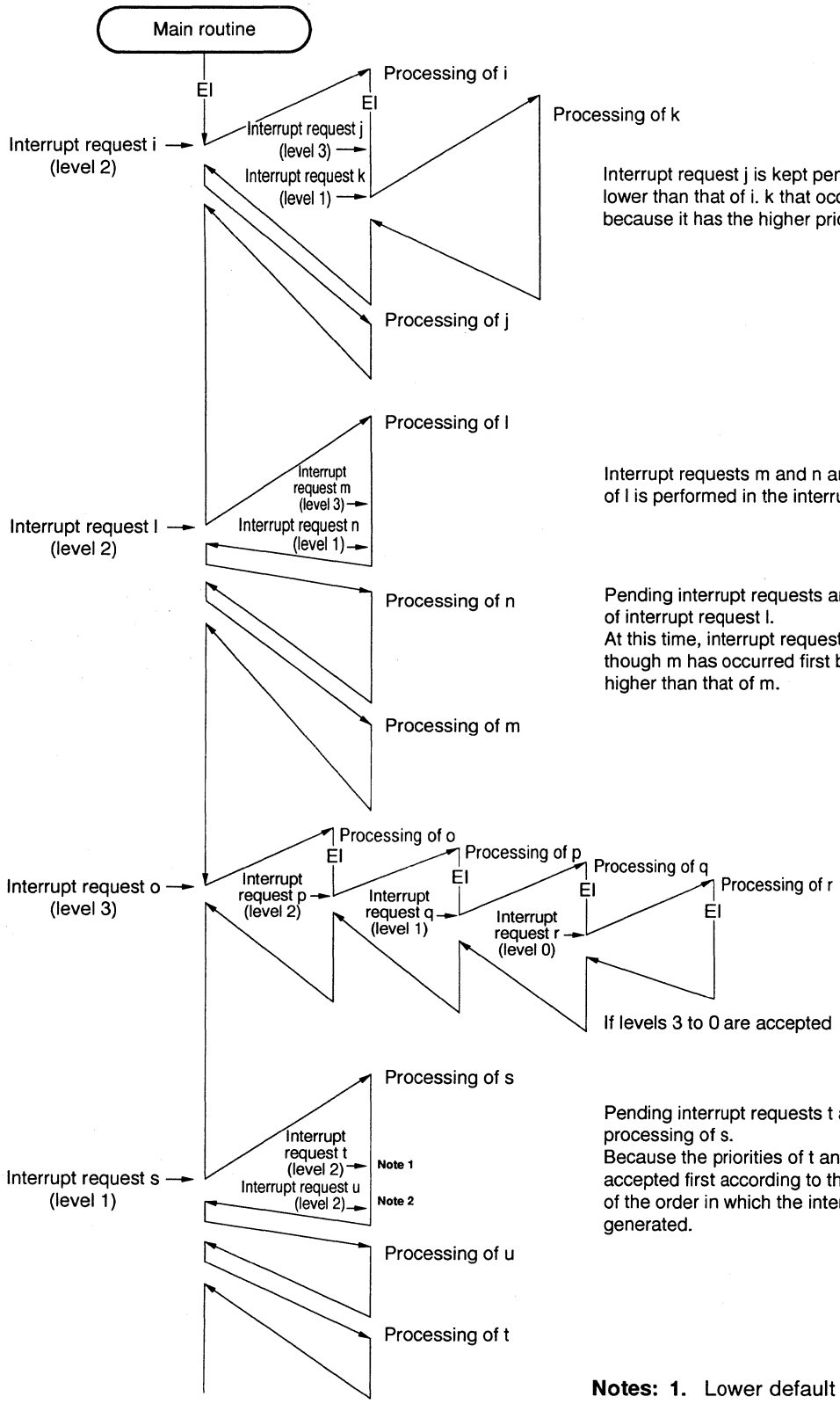


Remarks: **1.** a-u in the figure are the names of interrupt requests shown for the sake of explanation.

2. The default priority in the figure indicates the relative priority between two interrupt requests.

Caution: The values of EIPC and EIPSW must be saved before executing multiple interrupt.

Figure 5-7. Example of Interrupt Nesting Process (2/2)



Interrupt request j is kept pending because its priority is lower than that of i. k that occurs after j is accepted because it has the higher priority.

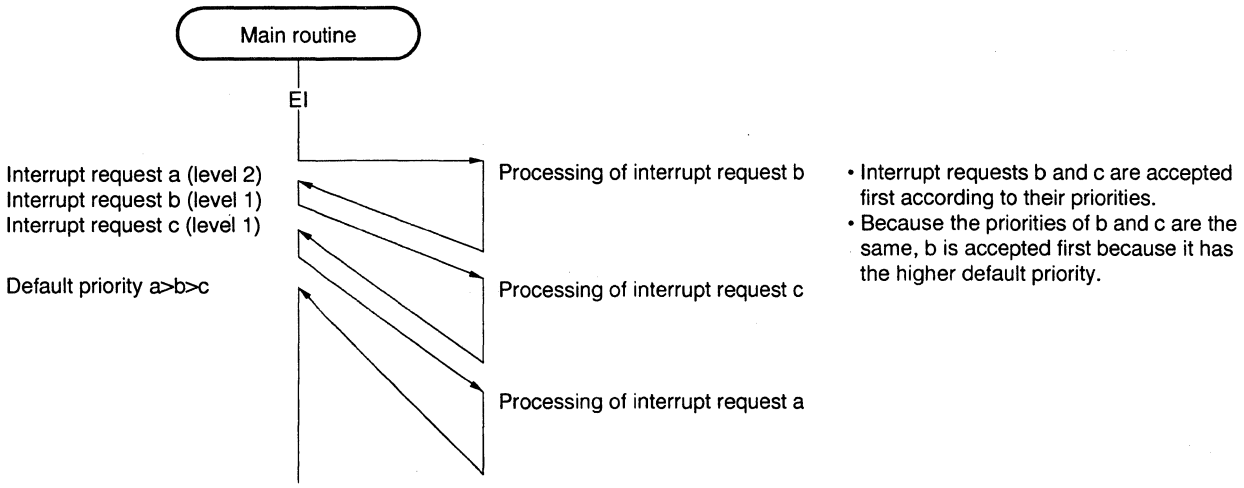
Interrupt requests m and n are kept pending because processing of l is performed in the interrupt disabled status.

Pending interrupt requests are accepted after processing of interrupt request l.
At this time, interrupt requests n is accepted first even though m has occurred first because the priority of n is higher than that of m.

Pending interrupt requests t and u are accepted after processing of s.
Because the priorities of t and u are the same, u is accepted first according to the default priority, regardless of the order in which the interrupt requests have been generated.

Notes: **1.** Lower default priority
2. Higher default priority

Figure 5-8. Example of Processing Interrupt Requests Simultaneously Generated



5.3.5 Interrupt control register (××ICn)

An interrupt control register is assigned to each maskable interrupt and holds the control conditions for each maskable interrupt request.

The interrupt control register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	At reset
XXICn	XXIFn	XXMKn	0	0	0	XXPRn2	XXPRn1	XXPRn0	FFFFF100H-FFFFF11AH	47H

Bit Position	Bit Name	Function
7	××IFn	Interrupt Request Flag Interrupt request flag 0: Interrupt request not issued 1: Interrupt request issued ××IFn flag is automatically reset by hardware when interrupt request is accepted.
6	××MKn	Mask Flag Interrupt mask flag 0: Enables interrupt processing 1: Disables interrupt processing (pending)
2-0	××PRn2-××PRn0	Priority Specifies eight levels of priorities for each interrupt

××PRn2	××PRn1	××PRn0	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: ××: identification name of each peripheral unit
n: peripheral unit number (0, 1, 2, ...)

5.3.6 External interrupt mode registers 1 and 2 (INTM1 and INTM2)

These registers specify the valid edges of external interrupt requests INTP00-INTP03 and INTP10-INTP13 that are input from external pins. INTM1 controls INTP00-INTP03, and INTP2 controls INTP10-INTP13.

The valid edge of each pin can be specified to be the rising, falling, and both rising and falling edges.

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

	7	6	5	4	3	2	1	0	Address	At reset
INTM1	ES031	ES030	ES021	ES020	ES011	ES010	ES001	ES000	FFFFF182H	00H

Bit Position	Bit Name	Function
7, 5, 3, 1 6, 4, 2, 0	ES0n1 ES0n0 (n=3-0)	Edge Select Specifies valid edge of INTP0n pin

ES0n1	ES0n0	Operation
0	0	Falling edge
0	1	Rising edge
1	0	RFU (reserved)
1	1	Both rising and falling edges

	7	6	5	4	3	2	1	0	Address	At reset
INTM2	ES131	ES130	ES121	ES120	ES111	ES110	ES101	ES100	FFFFF184H	00H

Bit Position	Bit Name	Function
7, 5, 3, 1 6, 4, 2, 0	ES1n1 ES1n0 (n=3-0)	Edge Select Specifies valid edge of INTP1n pin

ES1n1	ES1n0	Operation
0	0	Falling edge
0	1	Rising edge
1	0	RFU (reserved)
1	1	Both rising and falling edges

5.3.7 In-service priority register (ISPR)

This register holds the priority level of the maskable interrupt currently accepted. When an interrupt request is accepted, the bit of this register corresponding to the priority level of that interrupt is set to 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 reset to 0 by hardware. However, it is not reset when execution is returned from non-maskable processing or exception processing.

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

	7	6	5	4	3	2	1	0	Address	At reset
ISPR	ISPR7	ISPR6	ISPR5	ISPR4	ISPR3	ISPR2	ISPR1	ISPR0	FFFFF166H	00H

Bit Position	Bit Name	Function
7-0	ISPR7-ISPR0	In-Service Priority Flag Indicates priority of interrupt currently accepted 0: Interrupt request with priority n not accepted 1: Interrupt request with priority n accepted

Remark: n: 0-7 (priority level)

5.3.8 Maskable interrupt status flag

The interrupt disable status flag (ID) of the PSW controls the enabling and disabling of maskable interrupt requests. As a status flag, it also displays the current maskable interrupt acceptance condition.

	31																				8	7	6	5	4	3	2	1	0	At reset
PSW	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	00000020H

Bit Position	Bit Name	Function
5	ID	Interrupt Disable Enables or disables maskable interrupt processing. 0: Maskable interrupt accepting enabled 1: Maskable interrupt accepting disabled 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 interrupt and exceptions are acknowledged regardless of this flag. When a maskable interrupt is accepted, ID flag is automatically set to 1 by hardware.

5.4 Software Exception

The software exception is generated when the CPU executes the TRAP instruction, and can be always accepted.

TRAP instruction format: TRAP vector (where vector is 0-1FH)

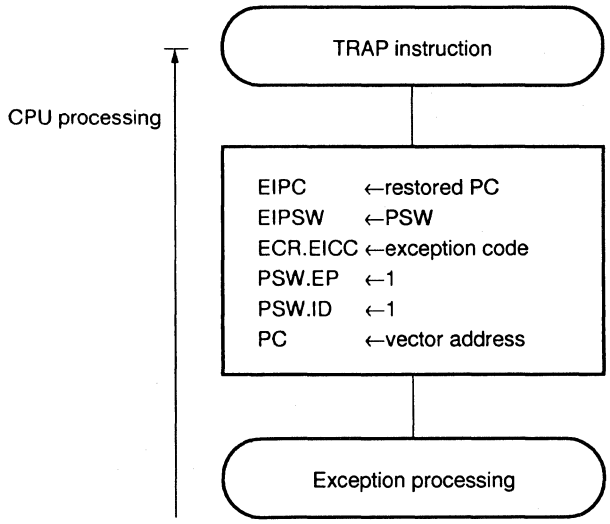
5.4.1 Operation

If the software exception occurs, the CPU performs the following processing, and transfers control to the handler routine:

(1) Saves the value of 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 PSW.
(5) Loads the vector address (00000040H or 00000050H) of the software exception routine in the PC, and transfers control.

Figure 5-9 illustrates how the software exception is processed.

Figure 5-9. Software Exception Processing



The vector address is determined by the operand of the TRAP instruction. If the operand is 0-0FH, the vector address is 00000040H; if the operand is 10H-1FH, it is 00000050H.

5.4.2 Restore

To restore or return execution from the software exception service routine, the RETI instruction is used.

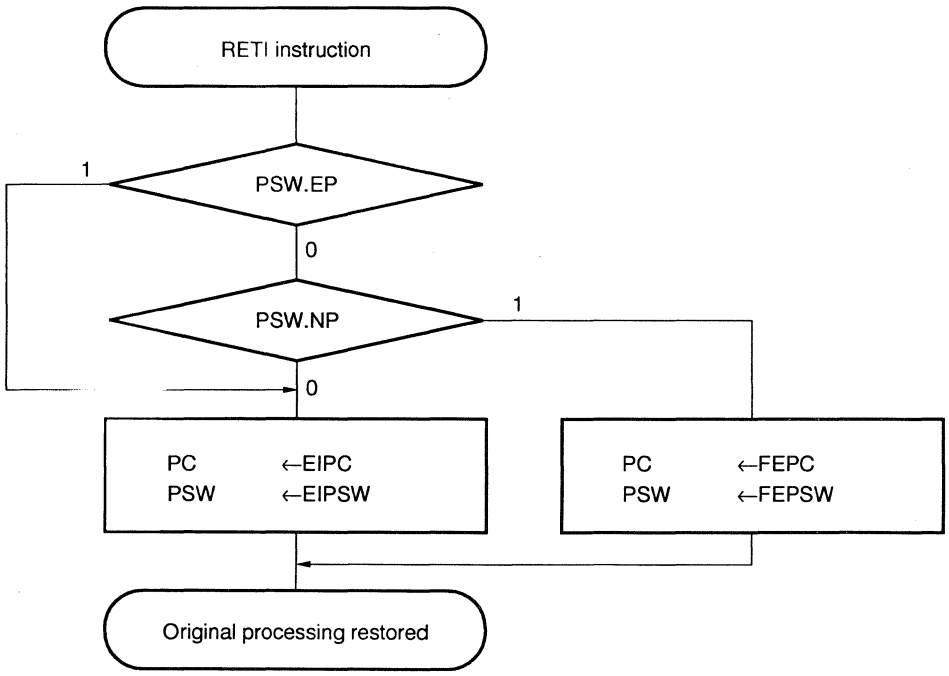
Operation of RETI instruction

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 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 5-10 illustrates the processing of the RETI instruction.

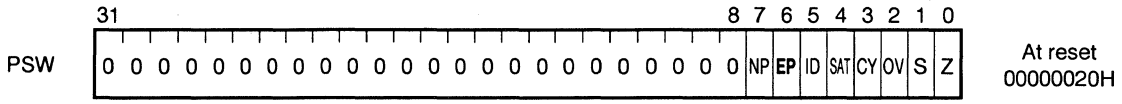
Figure 5-10. RETI Instruction Processing



Caution: If the PSW.EP and PSW.NP bits are changed using the LDSR instruction during software exception processing, it is necessary to reset PSW.EP to 0 and set PSW.NP to 1 using the LDSR instruction immediately before the RETI instruction to make sure that the PC and PSW are normally restored by the RETI instruction. ★

5.4.3 EP flag

The EP flag in the PSW is a status flag used to indicate that trap processing is in progress. It is set when a trap occurs.



Bit Position	Bit Name	Function
6	EP	Exception Pending Indicates that trap processing is in progress 0: Trap processing is not in progress 1: Trap processing is in progress

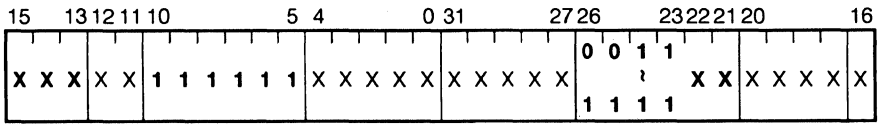
5.5 Exception Trap

The exception trap is an interrupt that is requested when illegal execution of an instruction takes place. In the V851, an illegal op code exception (ILGOP: ILeGal OPcode trap) is considered as an exception trap.

Illegal op code exception: occurs if the subop code field of an instruction to be executed next is not a valid op code.

5.5.1 Illegal op code definition

An illegal op code is defined to be a 32-bit word with bits 5-10 being 111111B and bits 23-26 being 0011B-1111B.



×: don't care

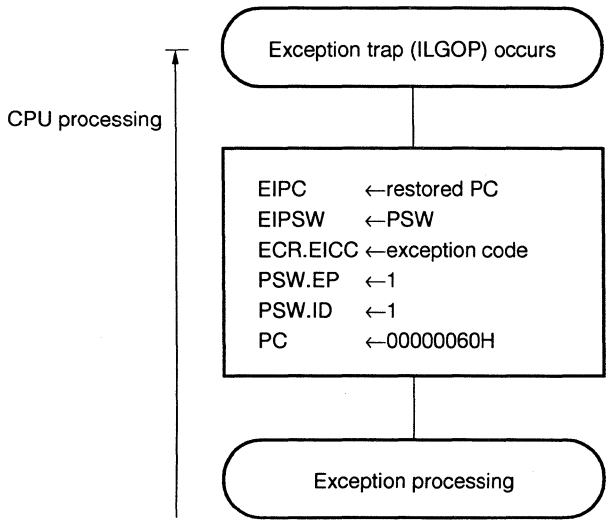
5.5.2 Operation

If an exception trap occurs, the CPU performs the following processing, and transfers control to the handler routine:

(1) Saves the value of PC to EIPC.
(2) Saves the current PSW to EIPSW.
(3) Writes an exception code (0060H) to the lower 16 bits (EICC) of ECR.
(4) Sets the EP and ID bits of PSW.
(5) Loads the vector address (00000060H) for the exception trap routine to the PC, and transfers control.

Figure 5-11 illustrates how the exception trap is processed.

Figure 5-11. Exception Trap Processing



5.5.3 Restore

To restore or return execution from the exception trap, the RETI instruction is used.

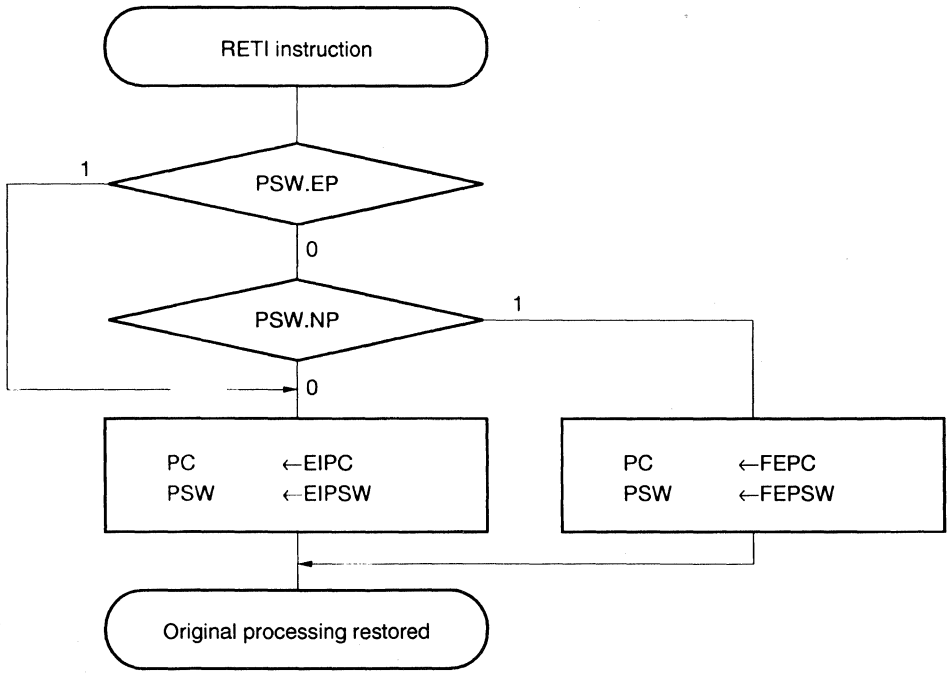
Operation of RETI instruction

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 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 5-12 illustrates the processing of the RETI instruction.

Figure 5-12. RETI Instruction Processing



★ **Caution: If the PSW.EP and PSW.NP bits are changed using the LDSR instruction during software exception processing, it is necessary to reset PSW.EP to 0 and set PSW.NP to 1 using the LDSR instruction immediately before the RETI instruction to make sure that the PC and PSW are normally restored by the RETI instruction.**

5.6 Priority Control

5.6.1 Priorities of interrupts and exceptions

	RESET	NMI	INT	TRAP	ILGOP
RESET		*	*	*	*
NMI	×		←	←	←
INT	×	↑		←	←
TRAP	×	↑	↑		←
ILGOP	×	↑	↑	↑	

RESET : reset
NMI : non-maskable interrupt
INT : maskable interrupt
TRAP : software exception
ILGOP illegal code exception
* : Item on the left ignores the item above.
× : Item on the left is ignored by the item above.
↑ : Item above is higher than the item on the left in priority.
← : Item on the left is higher than the item above in priority.

5.6.2 Multiple interrupt processing

Multiple interrupt processing is a function which allows the nesting of interrupts. If a higher priority interrupt is generated and accepted, it will be allowed to stop a current interrupt service routine in progress. Execution of the original routine will resume once the higher priority interrupt routine is completed.

If an interrupt with a lower or equal priority is generated and a service routine is currently in progress, the later interrupt will be kept pending.

Multiple interrupt processing control is performed while an interrupt service routine is currently in progress and the interrupts are kept enabled (ID=0). If a maskable interrupt or exception is generated and accepted while a prior interrupt routine is under progress, the higher priority interrupting routine must save the current contents of EIPC and EIPSW to allow proper restoration when the routine ends.

Programming examples used for interrupt nesting are shown in the following code fragments:

(1) To accept maskable interrupts in service routine

Service routine of maskable interrupt or exception

```
        ...
        ...
• Saves EIPC to memory or register
• Saves EIPSW to memory or register
• EI instruction (enables interrupt acceptance)
        ...
        ...                          ← Accepts interrupt such as INTP input
        ...
        ...
• DI instruction (disables interrupt acceptance)
• Restores saved value to EIPSW
• Restores saved value to EIPC
• RETI instruction
```

(2) To generate exception in service program

Service program of maskable interrupt or exception

```
        ...
        ...
• Saves EIPC to memory or register
• Saves EIPSW to memory or register
        ...
        ...                          ← Accepts exception such as TRAP instruction
        ...                          ← Accepts exception such as undefined instruction
        ...
• Restores saved value to EIPSW
• Restores saved value to EIPC
• RETI instruction
```

Priorities 0-7 (0 is the highest) can be programmed for each maskable interrupt request for multiple interrupt processing control. To set a priority level, write values to the xxPRn0-xxPRn2 bits of the interrupt request control register (xxICn) corresponding to each maskable interrupt request. At reset, the interrupt request is masked by the xxMKn bit, and the priority level is set to 7 by the xxPRn0-xxPRn2 bits.

Priorities of maskable interrupts

(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 interrupt processing is resumed after the interrupt processing of the higher priority has been completed and the RETI instruction has been executed.

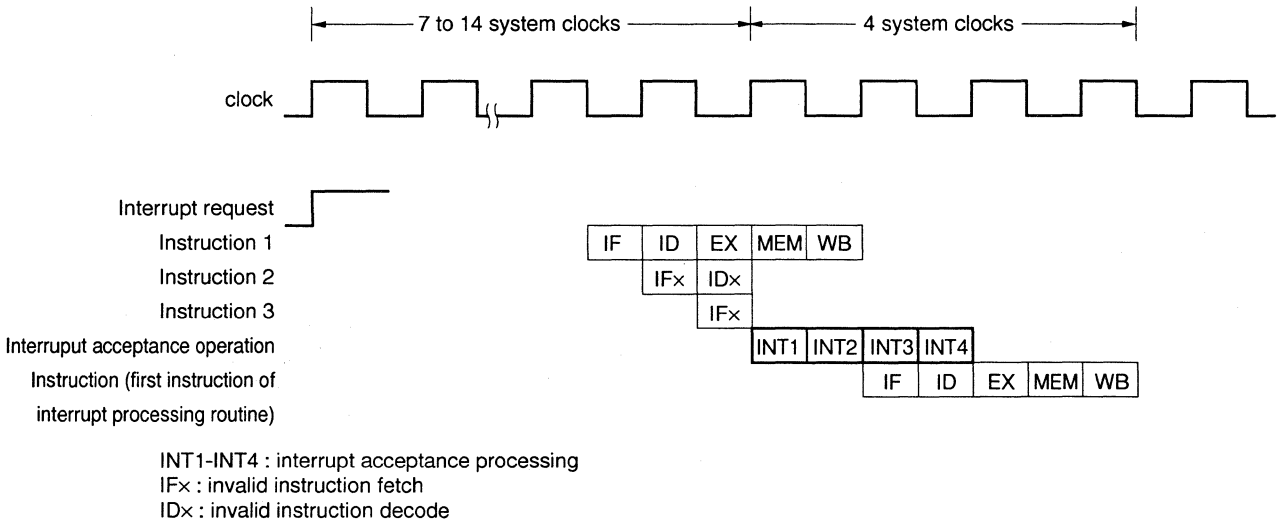
A pending interrupt request is accepted after the current interrupt processing has been completed and the RETI instruction has been executed.

★ **Caution: The maskable interrupt is not accepted but kept pending in the non-maskable interrupt routine (until the RETI instruction is executed).**

5.7 Interrupt Latency Time

The interrupt latency is defined as the time measured between the generation of the interrupt request and the execution of the first instruction in the corresponding interrupt processing routine. The following table describes the V851 interrupt latency time.

Figure 5-13. Pipeline Operation When Interrupt Request Is Accepted (outline) ★



INT1-INT4 : interrupt acceptance processing
IFx : invalid instruction fetch
IDx : invalid instruction decode

Interrupt Latency Time		Condition
Minimum	11 system clocks	Time to eliminate noise (2 system clocks) is also necessary for external interrupts, except when: • In IDLE/STOP mode • External bus is accessed • Two or more interrupt request non-sample instructions are executed in succession • Interrupt request control register is accessed
Maximum	18 system clocks	

★

5.8 Periods Where Interrupt Is Not Acknowledged

Interrupts are accepted during instruction execution. However, the interrupt is not accepted between the interrupt request non-sample instruction and the next instruction.

Interrupt request non-sample instruction

- EI instruction
- DI instruction
- LDSR reg2, 0x5 instruction (vs. PSW)

[MEMO]

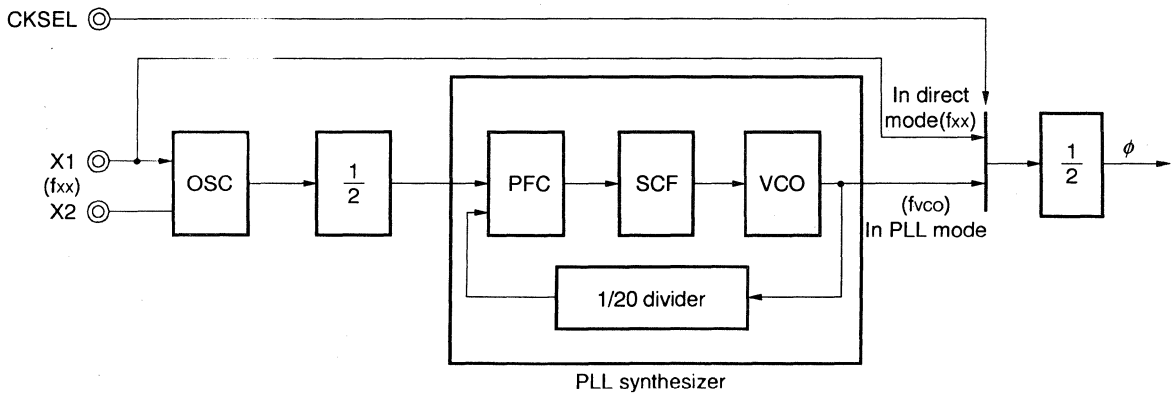
CHAPTER 6 CLOCK GENERATION FUNCTION

The clock generator produces and controls the internal system clock ϕ which is supplied to all the internal hardware units including the CPU.

6.1 Features

- ○ Multiplication function by PLL (Phase Locked Loop) synthesizer (×5)
- ○ Clock source
  - Oscillation through oscillator connection: fxx = $\frac{1}{5} \times \phi$ (PLL mode)
  - External clock: fxx = $\frac{1}{5} \times \phi$ (PLL mode)
  - External clock: fxx = $2 \times \phi$ (direct mode)
- ○ Power save mode
  - HALT mode
  - IDLE mode
  - Software STOP mode
- ○ Clock output inhibit function

6.2 Configuration



fvco : VCO oscillation frequency (= 10 · fxx)
ϕ : internal system clock frequency (= 1/2 · fvco: in PLL mode)
internal system clock frequency (= 1/2 · fxx: in direct mode)
OSC : oscillator (PLL mode only)
PFC : phase frequency comparator
SCF : switched capacitor filter
VCO : voltage-controlled oscillator

6.3 Selecting Input Clock

The clock generator consists of a clock oscillator and a PLL synthesizer. It can generate, for example, a 25-MHz system clock when a 5-MHz crystal resonator or ceramic resonator is connected across the X1 and X2 pins.

An external clock can be directly connected to the oscillator circuit. In this case, input the clock signal to the X1 pin, and leave the X2 pin open.

The clock generator has two operation modes: PLL and direct modes, and are selected by the CKSEL pin, as shown in the table below.

CKSEL	Operation Mode
0	PLL mode
1	Direct mode

Caution: The CKSEL pin level should never be changed during operation. The V851 may not operate correctly.

6.3.1 Direct mode

In the direct mode, an external clock with a frequency two times higher than that of the system clock is input. Because OSC and PLL synthesizer do not operate, the power dissipation can be significantly reduced. This mode is used mainly in applications where the V851 must operate on a relatively low frequency. To minimize the influence by noise, it is recommended that the frequency of the external clock, fxx, be kept to within 32 MHz (system clock ϕ = 16 MHz).

6.3.2 PLL mode

In the PLL mode, an external clock is input by connecting an external oscillator, which is multiplied by the PLL synthesizer to generate system clock (ϕ).

Because a frequency of up to 33 MHz can be generated based on an external oscillator of 3 to 5 MHz, a low-noise, power-saving system can be designed. The system clock (ϕ) with a frequency 5 times higher than the frequency fxx of the external oscillator or external clock (5 × fxx) can be generated.

The clock generator also provides a backup mode when operating in the PLL mode, thus improving system reliability. If the external oscillator or external clock source fails, the clock generator continues to provide the internal system clock ϕ based on the free-running frequency of the VCO. In this mode, the internal system clock ϕ operating at about 1 MHz.

Example of clock in PLL mode

System Clock Frequency ϕ	External Oscillator/External Clock Frequency (fxx)
32.768 MHz	6.5536 MHz
25.000 MHz	5.0000 MHz
20.000 MHz	4.0000 MHz
16.384 MHz	3.2768 MHz

6.4 PLL Stabilization

Following a power-on reset or when exiting the STOP mode, an amount of time will be required for the PLL to stabilize before using any of the V851 hardware functions which rely on execution speed. This required time is called PLL lock-up time, and is different (longer) than the oscillation stabilization time. Clock signals after the oscillation stabilization time have the required wave shape but the frequency might fluctuate. However clock signals after the PLL lock-up time are supplied at a specified frequency without fluctuation, satisfying the required wave shape. In addition, the status in which the frequency is not stable is called unlock status and the status in which it has been stabilized is called lock status.

Two system status flags are available to check with the stabilization of the PLL frequency: UNLOCK flag that indicates the stabilization status of the PLL frequency, and PRERR flag that indicates occurrence of a protection error (for the details of the PRERR flag, refer to **6.5.2 (2) Command register (PRCMD)**).

The SYS register, which contains these UNLOCK and PREERR flags, can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	At 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 read-only flag and indicates unlock status of PLL. It holds "0" as long as lock up status is maintained, and is not changed even if system is reset. 0 : Indicates lock status 1 : Indicates unlock status

Remark: For the description of the PRERR flag, refer to **6.5.2 (2) Command register (PRCMD)**.

If the unlock status condition should arise, due a power or clock source failure, the UNLOCK flag should be checked to verify that the PLL has stabilized before performing any execution speed dependent operations, such as real-time processing.

The static processing such as setting of the on-chip hardware units and initialization of the register data and memory data, however, can be executed before the UNLOCK flag is reset.

6.5 Power Save Control

6.5.1 General

The V851 is provided with the following power save or standby modes to reduce power consumption when CPU operation is not required.

(1) HALT mode

In this mode, the clock generator (oscillator and PLL synthesizer) continues operation but the operating clock of the CPU stops. The internal peripherals continue to function in reference to the internal system clock. Through intermittent operations between normal operation and HALT modes, total power consumption of the system can be reduced.

The HALT mode is entered by a dedicated instruction (HALT instruction).

(2) IDLE mode

In this mode, both the CPU clock and the internal system clock are stopped to further reduce power consumption. However, since the clock generator continues to run, normal operation can resume without having to wait for the oscillator and PLL circuits to stabilize.

The IDLE mode is entered by programming the PSC register.

The IDLE mode is categorized between the STOP and HALT modes in terms of clock stabilization time and power consumption, and is used in applications where the clock oscillation time should be eliminated but low power consumption is need.

(3) Software STOP mode

In this mode, the CPU clock, the internal system clock, and the clock generator are stopped, reducing power consumption to only leakage current. In this state, power consumption is minimized.

(a) In PLL mode

The software STOP mode is entered by programming the PSC register. As soon as the oscillator circuit stops, the clock output of the PLL synthesizer is stopped. After the software STOP mode has been released, it is necessary to allow for stabilization time of the oscillator and system clock. Moreover, the lock up or stabilization time of the PLL may also be necessary, depending on the application. However, when the processor operates on an external clock, the need for oscillation stabilization time of the oscillator will not be necessary.

(b) In direct mode

To stop the clock, fix the X1 pin to the low level.

The PLL lock up or stabilization time is not needed in the direct mode.

(4) Clock output inhibit

Output of the system clock from the CLKOUT pin is inhibited.

The operations of the clock generator in the normal, HALT, IDLE, and software STOP modes are shown in Table 6-1.

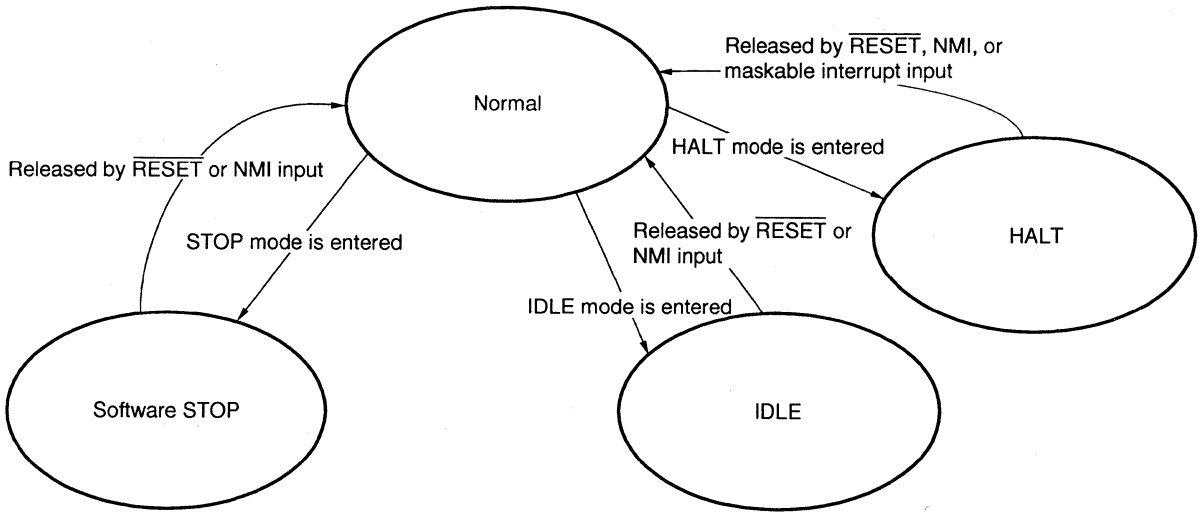
By combining and selecting the mode ideal for a specific application, the power consumption of the system can be effectively reduced.

Table 6-1. Operation of Clock Generator by Power Save Control

Clock Source		Standby Mode	Oscillator Circuit (OSC)	PLL Synthesizer	Clock Supply to Peripheral I/O	Clock Supply to CPU
PLL mode	Oscillation by crystal oscillator	Normal	O	O	O	O
		HALT	O	O	O	×
		IDLE	O	O	×	×
		STOP	×	×	×	×
	External clock	Normal	×	O	O	O
		HALT	×	O	O	×
		IDLE	×	O	×	×
		STOP	×	×	×	×
Direct mode		Normal	×	×	O	O
		HALT	×	×	O	×
		IDLE	×	×	×	×
		STOP	×	×	×	×

O : operates
× : stops

Status Transition Diagram



6.5.2 Control registers

(1) Power save control register (PSC)

This is an 8-bit register that controls the power save mode. It can be written only by a specific combination of instruction sequences so that its contents are not written by mistake due to erroneous program execution. This register can be read/written in 8- or 1-bit units.

	7	6	5	4	3	2	1	0	Address	At reset
PSC	DCLK1	DCLK0	TBCS	CESEL	0	IDLE	STP	0	FFFFF070H	00H

Bit Position	Bit Name	Function
7, 6	DCLKn (n=1, 0)	Disable CLKOUT Specifies operation mode of CLKOUT pin DCLK1 / DCLK0 / Mode: 0 / 0 / Normal output mode; 0 / 1 / RFU (reserved); 1 / 0 / RFU (reserved); 1 / 1 / Clock output inhibit mode
5	TBCS	Time Base Count Select Selects clock of time base counter 0: $fxx/2^8$ 1: $fxx/2^9$ For details, refer to explanation of "Time base counter (TBC)" in section 6.6 "Specifying Oscillation Stabilization Time".
4	CESEL	Crystal/External Select Specifies functions of X1 and X2 pins 0: Oscillator connected to X1 and X2 pins 1: External clock connected to X1 pin When CESEL = 1, cuts off feedback loop of oscillation circuit and does not make sure that oscillation stabilization time elapses after STOP mode has been released.
2	IDLE	IDLE Mode Specifies IDLE mode. When "1" is written to this bit, IDLE mode is entered. When IDLE mode is released, this bit is automatically reset to "0".
1	STP	STOP Mode Specifies software STOP mode. When "1" is written to this bit, STOP mode is entered. When STOP mode is released, this bit is automatically reset to "0".

Set data to the PSC register in the following sequence. ★

<1> Disable interrupts (by setting the NP bit of PSW to 1).
<2> Write any 8-bit data to the command register (PRCMD).
<3> Write set data to the PSC register (using the following instructions).
- Store instruction (ST/SST instruction)
- Bit manipulation instruction (SET1/CLR1/NOT1 instruction)

<4> Enable interrupts (by resetting the NP bit of PSW to 0).
<5> Insert NOP instructions (two or five instructions).

The PSC register can be read in any sequence.

Cautions: 1. If an interrupt is accepted between issuance of PRCMD (<2>) and writing to the PSC register immediately after that (<3>), nothing is written to the PSC register, and a protection error (in which case the PRERR bit of the SYS register is set to "1") may occur. Therefore, set the NP bit of PSW to 1 (<1>) and disable INT/NMI acceptance.
The same applies to use of a bit manipulation instruction to set the PSC register.
Insert NOP instructions (<5>) as dummy instructions so that the routine is executed correctly after the STOP/IDLE mode has been released. If the value of the ID bit of PSW does not change even if the instruction (<4>) that resets the NP bit to 0 is executed, insert two NOP instructions. Insert five NOP instructions if the value of the ID bit changes. Here is an example:

```
[Example]
LDSR  rX,5            ; NP bit = 1
ST.B  r0,PRCMD [r0] ; Writing to PRCMD
ST.B  rD,PSC [r0]     ; Setting of PSC register
LDSR  rY,5            ; NP bit = 0
NOP                   ; Dummy instruction (2 or 5 instructions)
        ⋮
NOP
(next instruction)    ; Execution routine after release of STOP/IDLE mode
        ⋮
```

rX: Value to be written to PSW
rY: Value to be written back to PSW
rD: Value to be set to PSC

To save the value of PSW, the value of PSW before the NP bit is set must be transferred to the rY register.

2. **The instructions after the store instruction (<4> Enabling interrupt, <5> NOP instructions) that are executed on the PSC register to set the software STOP mode or IDLE mode, are executed before each power save mode is set.**

(2) Command register (PRCMD)

The command register protects the PSC register from being illegally written so that the application system does not stop due to program hang-up.

Only data written first to the PSC register after the PRCMD register has been written becomes valid.

Because the register value can be rewritten only in a fixed sequence, illegal write operations are prevented.

The command register can be only written in 8-bit units (when this register is read, undefined data is read).

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

Bit Position	Bit Name	Function
7-0	REG7-REG0	Registration Code Registration code (any 8-bit data)

Occurrence of an illegal store operation can be checked by the PRERR flag of the system status register (SYS).

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

Bit Position	Bit Name	Function
4	PRERR	Protection Error Flag Indicates that PSC register is not written in the correct sequence and that a protection error has occurred. 0: Protection error does not occur 1: Protection error occurs

Remark: For the description of the UNLOCK flag, refer to 6.4 PLL Stabilization.

Operation conditions of PRERR flag

- Set condition (PRERR = "1") : <1> If the store instruction most recently executed to the peripheral I/O does not write data to the PRCMD register, but to PSC register
  ★ <2> If the first store instruction executed after the write operation to the PRCMD register is to the peripheral I/O register except PSC registers.
- Reset condition (PRERR = "0") : <1> When "0" is written to the PRERR flag of the SYS register.
  <2> At system reset

6.5.3 HALT mode

(1) Entering and operation status

In the HALT mode, the clock generator (oscillator circuit and PLL synthesizer) operates, while the operating clock of the CPU stops. The internal peripherals continue to function in reference to the internal system clock. By entering the HALT mode during the idle time of the CPU, the total power consumption of the system can be reduced.

This mode is entered by the HALT instruction.

In the HALT mode, program execution is stopped, but the contents of the registers and internal RAM immediately before entering the HALT mode are retained. The on-chip peripheral functions that are not dependent on the instruction processing of the CPU continue to operate.

Table 6-2 shows the status of each hardware unit in the HALT mode.

Table 6-2. Operating Status in HALT Mode

Function		Operating Status	
Clock Generator		Operates	
Internal System Clock		Operates	
CPU		Stops	
I/O Line		Retained	
Peripheral Function		Operates	
Internal Data		Status of internal data before setting of HALT mode, such as CPU registers, status, data, and internal RAM contents, are retained.	
External Expansion Mode	AD0-AD15	High impedance[Note]	
	A16-A23	Retained[Note]	High-impedance when $\overline{HLDAK}$ = 0
	$\overline{LBEN}$, $\overline{UBEN}$	1[Note]	
	R/$\overline{W}$		
	$\overline{DSTB}$		
	ASTB		
	ST0, ST1	00[Note]	
	$\overline{HLDAK}$	Operates	
CLKOUT		Clock output (when clock output is not inhibited)	

Note: The instruction fetch operation continues even after the HALT instruction has been executed, until the internal instruction prefetch queue becomes full. After the queue has become full, the operation is stopped in the status indicated in the above table.

(2) Releasing HALT mode

The HALT mode can be released by the non-maskable interrupt request, an unmasked maskable interrupt request, or a $\overline{\text{RESET}}$ signal input.

(a) Releasing by interrupt request

The HALT mode is unconditionally released by the NMI request 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 accepted. 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 accepted.

Operation after HALT mode has been released by interrupt request

Releasing Source	EI Status	DI Status
NMI request	Branches to vector address	
Maskable interrupt request	Branches to vector address or executes next instruction	Executes next instruction

(b) Releasing by $\overline{\text{RESET}}$ signal input

The operation same as the normal reset operation is performed.

6.5.4 IDLE mode

(1) Entering and operation status

In this mode, both the CPU clock and the internal system clock are stopped to further reduce power consumption. However, since the clock generator continues to run, normal operation can resume without having to wait for the oscillator and PLL circuit to stabilize.

The IDLE mode is entered when the PSC register is programmed by the store (ST/SST) instruction or bit manipulation (SET1/CLR1/NOT1) instruction.

Execution of the program is stopped in the IDLE mode, but the contents of the registers and internal RAM immediately before entering the IDLE mode are retained. The on-chip peripheral function are stopped in this mode. The external bus hold request ($\overline{\text{HLDRQ}}$) is not accepted. ★

Table 6-3 shows the hardware status in the IDLE mode.

Table 6-3. Operating Status in IDLE Mode

Function		Operating Status
Clock Generator		Operates
Internal System Clock		Stops
CPU		Stops
I/O Line		Retained
Peripheral Function		Stops
Internal Data		Status of all internal data immediately before IDLE mode is entered, such as CPU registers, status, data, and internal RAM contents, are retained.
External Expansion Mode	AD0-AD15	High-impedance
	A16-A23	
	$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$	
	R/$\overline{\text{W}}$	
	$\overline{\text{DSTB}}$	
	ASTB	
	ST0, ST1	
	$\overline{\text{HLDAK}}$	
CLKOUT		0

(2) Releasing IDLE mode

The IDLE mode is released by the NMI signal input or $\overline{\text{RESET}}$ signal input.

(a) Releasing by NMI signal input

The NMI request is accepted and serviced as soon as the IDLE mode has been released.

If the IDLE mode is entered in the NMI processing routine, however, only the IDLE mode is released, and the interrupt will not be accepted. The interrupt request will be retained and kept pending.

The interrupt processing that is started by the NMI signal input when the IDLE mode is released is treated in the same manner as a normal NMI interrupt that is processed (because there is only one vector address of the NMI interrupt). Therefore, if it is necessary to distinguish between the two types of NMI interrupts, a software flag should be defined in advance, and the flag must be set before setting the IDLE flag by the store/bit manipulation instruction. By checking this flag during the NMI interrupt processing, the NMI used to released the IDLE mode can be distinguished from the normal NMI.

(b) Releasing by $\overline{\text{RESET}}$ signal input

The operation same as the normal reset operation is performed.

6.5.5 Software STOP mode

(1) Entering and operaton status

In this mode, the CPU clock, the internal system clock, and the clock generator are stopped, reducing power consumption to only leakage current. In this state, power consumption is minimized.

The software STOP mode is entered by programming the PSC register using the store (ST/SST) or bit manipulation (SET1/CLR1/NOT1) instruction.

It is necessary to ensure the oscillation stabilization time of the oscillator circuit after the software STOP mode has been released, when the PLL mode (CKSEL pin = "0") and the oscillator connection mode (CESEL bit = "0") are set.

In the software STOP mode, program execution is stopped, but all the contents of the registers and internal RAM immediately before entering the STOP mode are retained. The on-chip peripheral function also stops operation.

Table 6-4 shows the hardware status in the software STOP mode.

Table 6-4. Operating Status in Software STOP Mode

Function		Operating Status
Clock Generator		Stops
Internal System Clock		Stops
CPU		Stops
I/O Line[Note]		Retained
Peripheral Function[Note]		Stops
Internal Data		Status of all internal data immediately before software STOP mode is set, such as CPU registers, status, data, and internal RAM contents, are retained.
External Expansion Mode	AD0-AD15	High-impedance
	A16-A23	
	$\overline{LBEN}$, $\overline{UBEN}$	
	R/$\overline{W}$	
	$\overline{DSTB}$	
	ASTB	
	ST0, ST1	
	$\overline{HLDAK}$	
CLKOUT		0

Note: When the value of V_{DD} is within the operating range.
Even if V_{DD} drops below the minimum operating voltage, the contents of the internal RAM can be retained if the data retention voltage V_{DDDR} is maintained.

(2) Releasing software STOP mode

The STOP mode is released by the NMI signal input or $\overline{\text{RESET}}$ signal input.

It is necessary to ensure the oscillation stabilization time when releasing from the STOP mode. This will depend on the operating status of the oscillator circuit (PLL mode (CKSEL pin = "0") and in the oscillator connection mode (CESEL bit = "0")).

(a) Releasing by NMI signal input

When the STOP mode is released by the NMI signal, the NMI request is also accepted.

If the STOP mode is set in an NMI processing routine, however, only the STOP mode is released, and the interrupt is not accepted. The interrupt request is retained and kept pending.

★ **Caution: When inputting an external clock to the X1 pin, supply the external clock at least 100 μs before releasing the STOP mode by using NMI input.**

NMI interrupt processing on releasing STOP mode

The interrupt processing that is started by the NMI signal input when the STOP mode is released is treated in the same manner as a normal NMI interrupt that is processed (because there is only one vector address of the NMI interrupt). Therefore, if it is necessary to distinguish between the two types of NMI interrupts, a software flag should be defined in advance, and the flag must be set before setting the STOP flag by the store/bit manipulation instruction. By checking this flag during the NMI interrupt processing, the NMI used to released the STOP mode can be distinguished from the normal NMI.

(b) Releasing by $\overline{\text{RESET}}$ signal input

The operation same as the normal reset operation is performed.

★ **Caution: When input an external clock to the X1 pin, make sure that the low-level width of the $\overline{\text{RESET}}$ pin is 100 μs or more when supplying the clock.**

6.6 Specifying Oscillation Stabilization Time

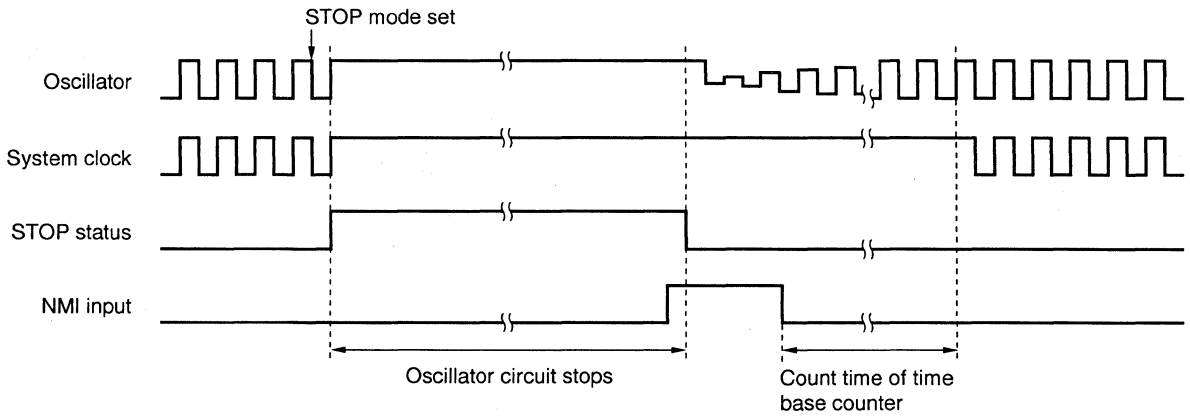
The time required for the oscillator circuit to become stabilized after the STOP mode has been released can be specified in the following two ways:

(1) By using internal time base counter (NMI signal input)

When the valid edge is input to the NMI pin, the STOP mode is released. When the inactive edge is input to the pin, the time base counter (TBC) starts counting, and the time required for the clock output from the oscillator circuit to become stabilized is specified by that count time.

Oscillation stabilization time = (Active level width after valid edge of NMI input has been detected) + (Count time of TBC)

After a specific time has elapsed, the system clock output is started, and execution branches to the vector address of the NMI interrupt.



During inactivity, the NMI pin should be kept at the inactive level (e.g. at a logic "1" when the valid edge is specified to be the falling edge).

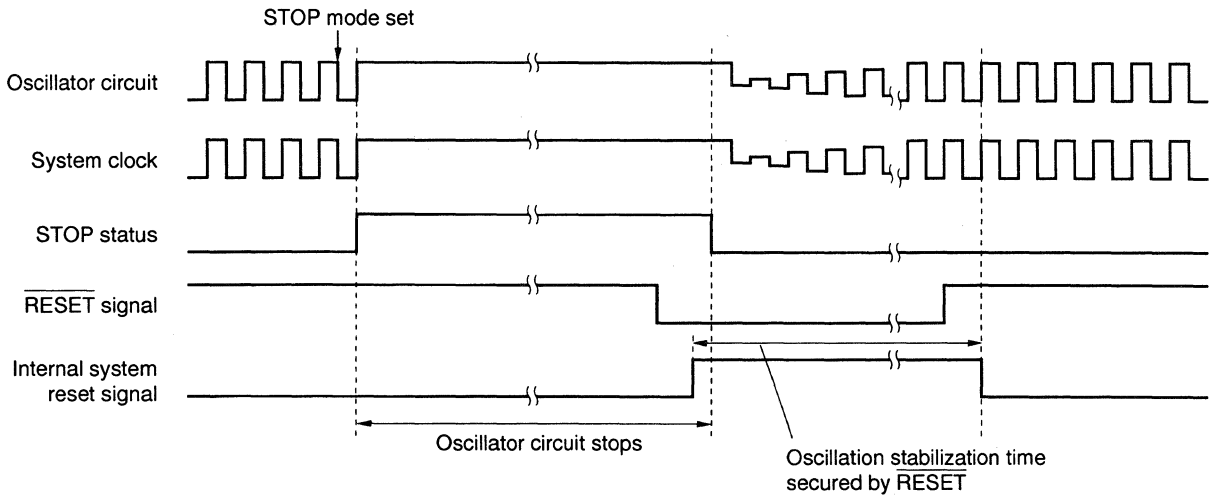
If an operation to enter the STOP mode is performed while a valid edge has been input to the NMI pin before the CPU accepts the interrupt, the STOP mode will immediately be released. Program execution is immediately started if the clock generator is in the direct mode (CKSEL = "1") and is driven by external clock (CESEL = 1). If the clock generator is in the PLL mode (CKSEL = "0) or is driven by an oscillator (CESEL = 0), program execution is started after the oscillation stabilization time specified in the time base counter has elapsed, following the valid edge input to the NMI pin.

(2) To specify time by signal level width ($\overline{\text{RESET}}$ signal input)

The STOP mode is released when the falling edge is input to the $\overline{\text{RESET}}$ pin.

The time required for the clock output from the oscillator circuit to become stabilized is specified by the low-level width of the signal input to the $\overline{\text{RESET}}$ pin.

After the rising edge has been input to the $\overline{\text{RESET}}$ pin, operation of the internal system clock begins, and execution branches to the vector address that is used when the system is reset.



Time base counter (TBC)

The time base counter is used to secure the oscillation stabilization time of the oscillator circuit when the software STOP mode is released.

The count clock of TBC is selected by the TBCS bit of the PSC register, and the following count time can be set:

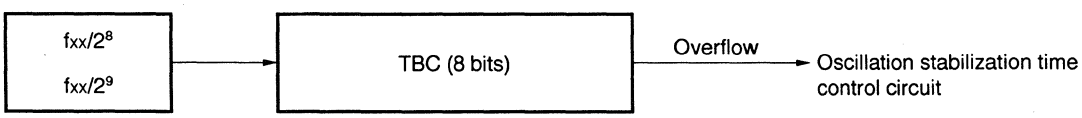
★ **Table 6-5. Example of Count Time**

TBCS	Count Clock	Count Time			
		fxx = 3.2768 MHz	fxx = 4.0000 MHz	fxx = 5.0000 MHz	fxx = 6.5536 MHz
		ϕ = 16.384 MHz	ϕ = 20.000 MHz	ϕ = 25.000 MHz	ϕ = 32.768 MHz
0	fxx/2^8	20.0 ms	16.3 ms	13.1 ms	10.0 ms
1	fxx/2^9	40.0 ms	32.7 ms	26.2 ms	20.0 ms

fxx : external oscillator frequency

ϕ : internal system clock frequency

Figure 6-1. Block Configuration



6.7 Clock Output Control

The operation mode of the CLKOUT pin can be selected by the DCLK0 and DCLK1 bits of the PSC register.

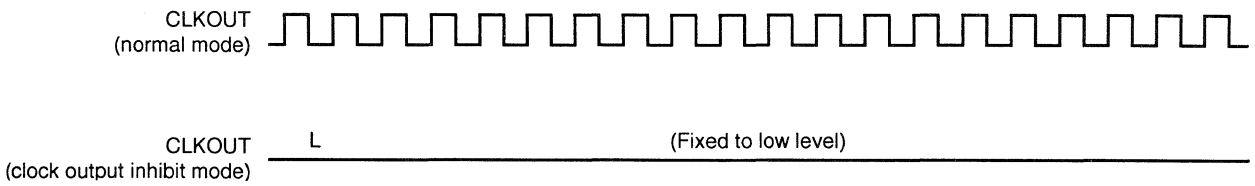
By using this operation mode in combination with the HALT, IDLE, or STOP mode, the power dissipation can be effectively reduced (for how to write these bits, refer to **6.5.2 Control registers**).

Clock output inhibit mode

The clock output from the CLKOUT pin is inhibited.

This mode is ideal for single-chip mode systems or systems that fetch instructions to external expansion devices or asynchronously accesses data.

Because the operation of CLKOUT is completely stopped in this mode, the power dissipation can be minimized and radiation noise from the CLKOUT pin can be suppressed.



[MEMO]

CHAPTER 7 TIMER/COUNTER FUNCTION (REAL-TIME PULSE UNIT)

7.1 Features

- ○ Measures pulse intervals and frequency, and outputs programmable pulse
  - 16-bit measurement possible
  - Generates pulses of various shapes (interval pulse, one-shot pulse)
- ○ Timer 1
  - 16-bit timer/event counter
  - Count clock source: 2 types (divided system clock and external pulse input)
  - Capture/compare register: 4
  - Count clear pin: TCLR1
  - Interrupt source: 5 types
  - External pulse output: 2
- ○ Timer 4
  - 16-bit interval timer
  - Count clock selected from divided system clock
  - Compare register: 1
  - Interrupt source: 1

7.2 Basic Configuration

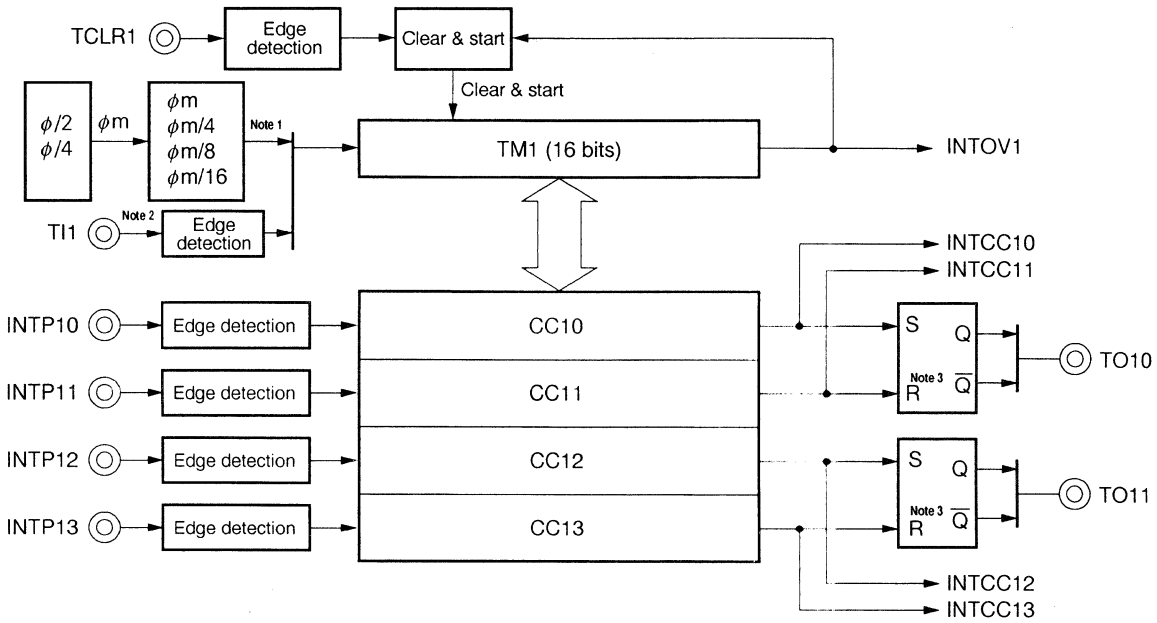
The basic configuration of the real-time pulse unit (RPU) is shown in the table below.

Table 7-1. Configuration of RPU

Timer	Count Clock	Register	Read/Write	Generated Interrupt Signal	Capture Trigger	Timer Output SR	Other Function
Timer 1	ϕ / 2 ϕ / 4 ϕ / 8 ϕ / 16 ϕ / 32 ϕ / 64 TI1 pin input	TM1	Read	INTOV1	–	–	External clear
		CC10	Read/write	INTCC10	INTP10	TO10 (S)	–
		CC11	Read/Write	INTCC11	INTP11	TO10 (R)	–
		CC12	Read/Write	INTCC12	INTP12	TO11 (S)	–
		CC13	Read/Write	INTCC13	INTP13	TO11 (R)	–
Timer 4	ϕ / 32 ϕ / 64 ϕ / 128 ϕ / 256	TM4	Read	–	–	–	–
		CM4	Read/write	INTCM4	–	–	–

Remark: ϕ : system clock
SR : set/reset

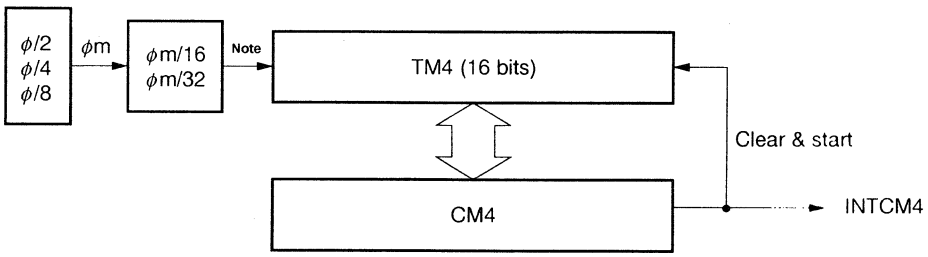
(1) Timer 1 (16-bit timer/event counter)



Notes: 1. Internal count clock frequency
2. External count clock frequency
3. Reset priority

Remark: ϕ indicates the system clock.

(2) Timer 4 (16-bit interval timer)



Note: Internal count clock

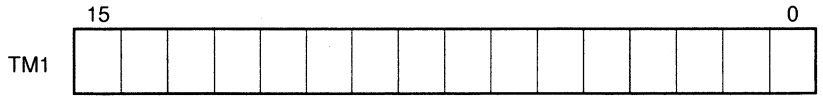
Remark: ϕ indicates the system clock.

7.2.1 Timer 1

(1) Timer 1 (TM1)

TM1 functions as a 16-bit free-running timer or event counter. Timer 1 is used to measure cycles and frequency, and also for programmable pulse generation.

TM1 can be only read in 16-bit units.



Address: FFFFF250H
At reset: 0000H

TM1 counts up the internal count clock or external count clock. The timer is started or stopped by the CE1 bit of timer control register 1 (TMC1).

Whether the internal or external count clock is used is specified by the TMC1 register.

(a) When external count clock is selected

TM1 operates as an event counter. The valid edge is specified by timer unit mode register 1 (TUM1), and TM1 counts up the signal input from the TI1 pin

★ **(b) When internal count clock is selected**

TM1 operates as a free running timer. The frequency of the count clock can be selected from the frequency divided by the prescaler, $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, $\phi/32$, or $\phi/64$, by using the TMC1 register.

When the timer overflows, an overflow interrupt can be generated. The timer can be stopped after an overflow has occurred, if so specified by the TUM1 register.

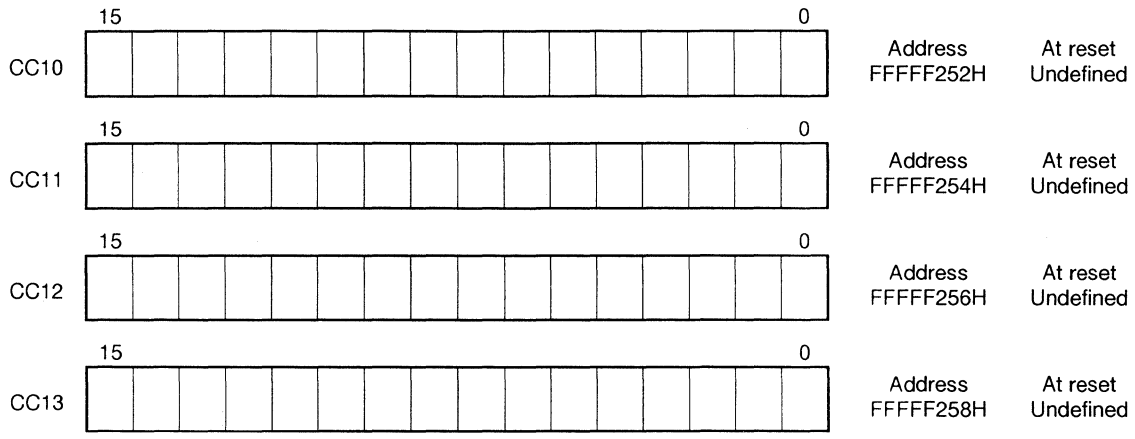
The timer can be cleared and started by external TCLR1 input. At this time, the prescaler is cleared at the same time. As a result, the time from the TCLR1 input to the first count up by the timer is held constant, according to the division ratio of the prescaler. The operation is set by the TUM1 register.

When the $\overline{\text{RESET}}$ signal is input, all the bits of TM1 are cleared to 0.

★ **Caution: Do not change the count clock frequency while the timer operates.**

(2) Capture/compare registers 10-13 (CC10-CC13)

Capture/compare registers are 16-bit registers and are connected to the TM1. These registers can be used as capture or compare registers depending on the specification of the timer unit mode register 1 (TUM1). They can be read/written in 16-bit units.



(a) When used as capture register

When a capture/compare register is used as a capture register, it detects the valid edge of the corresponding external interrupt (INTP10-INTP13) as a capture trigger. Timer 1 latches the count value in synchronization with the capture trigger (capture operation). The capture operation is performed asynchronously with the count clock. The latched value is held by the capture register, until the next capture operation is performed.

If the capture (latch) timing of the capture register contends with a register write operation by an instruction, the latter takes precedence, and the capture operation is ignored.

The valid edge of the external interrupt (rising, falling, or both edges) can be selected by external interrupt mode register (INTM2).

When a capture/compare register is used as a capture register, and when the valid edge of INTP10-INTP13 is detected, an interrupt is generated. During this time, no interrupt cannot be generated by the compare function of the register.

(b) When used as compare register

When a capture/compare register is used as a compare register, it compares its contents with the value of the timer at each clock tick. When the two values match, a coincidence signal INTCCn is generated. This coincidence signal can be used either to generate a maskable interrupt, to set/reset the timer output pins, or do both functions.

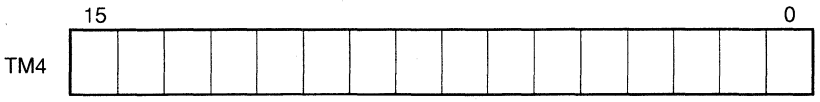
The interrupt source depends on the register mode, whether it is used as a capture or compare register. When used as a compare register, coincidence signal INTCCn or the valid edge of INTPn can be selected as an interrupt signal, depending on the specification of the TUM1 register.

When INTPn is selected, accepting an external interrupt (INTPn) and timer output by the set/reset output function of the compare register can be performed at the same time.

7.2.2 Timer 4

(1) Timer 4 (TM4)

TM4 is a 16-bit timer and is mainly used as an interval timer for software.
This timer can be only read in 16-bit units.



Address	At reset
FFFFF350H	0000H

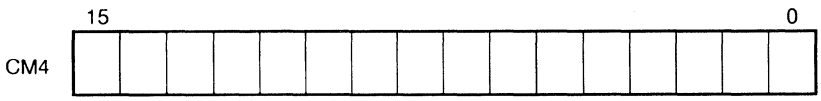
TM4 is started or stopped by the CE4 bit of the timer control register 4 (TMC4).
The count clock is selected by the TMC4 register from ϕ/32, ϕ/64, ϕ/128, or ϕ/256.
All the bits of TM4 are cleared to 0 by the $\overline{\text{RESET}}$ signal.

Cautions: 1. When the value of the timer coincides with the value of the compare register (CM4), the timer is cleared by the next clock tick. If the division ratio is large and results in a slow clock period, the timer value may not be cleared to zero yet, if the timer is read immediately after the occurrence of the coincidence signal interrupt.

★ **2. Do not change the count clock frequency while the timer operates.**

(2) Compare register 4 (CM4)

CM4 is a 16-bit register and is connected to TM4. This register can be read/written in 16-bit units.



Address	At reset
FFFFF352H	Undefined

CM4 compares its value with the value of TM4 at each clock tick of TM4, and generates an interrupt (INTCM4) when the two values match or coincide with each other. TM4 is cleared in synchronization with this coincidence.

7.3 Control Registers

(1) Timer unit mode register 1 (TUM1)

TUM1 is a register that controls the operation of timer 1, and specifies the operation mode of the capture/compare registers.

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	At reset
TUM1	0	0	OST	ECLR1	TES11	TES10	CES11	CES10	CMS13	CMS12	CMS11	CMS10	IMS13	IMS12	IMS11	IMS10	FFFFF240H	0000H

Bit Position	Bit Name	Function
13	OST	Overflow Stop Specifies operation of timer after occurrence of overflow. This flag is valid only for TM1. 0: Timer continues counting after overflow has occurred. 1: Timer holds 0000H and stops after overflow has occurred. At this time, CEI bit of TMC1 register remains "1". Timer resumes counting when following operation is performed: When ECLR1 = "0": Writing "1" to CE1 bit When ECLR1 = "1": Trigger input to timer clear pin (TCLR1)
12	ECLR1	External Input Timer Clear Enables clearing TM1 by external clear input (TCLR1) 0: TM1 is not cleared by external input 1: TM1 is cleared by external input After TM1 has been cleared, it starts counting.
11, 10	TES11, TES10	TI1 Edge Select Specifies valid edge of external clock input (TI1)
9, 8	CES11, CES10	TCLR1 Edge Select Specifies valid edge of external clear input (TCLR1)

TES11	TES10	Valid Edge
0	0	Falling edge
0	1	Rising edge
1	0	RFU (reserved)
1	1	Both rising and falling edges

CES11	CES10	Valid Edge
0	0	Falling edge
0	1	Rising edge
1	0	RFU (reserved)
1	1	Both rising and falling edge

Bit Position	Bit Name	Function
7-4	CMS10-CMS13	Capture/Compare Mode Select Selects operation mode of capture/compare registers (CC10-CC13) 0: Capture register. However, capture operation is performed only when CE1 of TMC1 register = "1". 1: Compare register
3-0	IMS10-IMS13	Interrupt Mode Select Selects INTPn or INTCCn as interrupt source (n = 10-13) 0: Uses coincidence signal of INTCCn of compare register as interrupt signal 1: Uses external input signal INTPn as interrupt signal

(2) Timer control register 1 (TMC1)

TMC1 controls operation of TM1.

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

	7	6	5	4	3	2	1	0	Address	At reset
TMC1	CE1	0	0	ETI	PRS11	PRS10	PRM11	0	FFFFF242H	00H

Bit Position	Bit Name	Function
7	CE1	Count Enable Controls timer operation. 0: Timer stops at "0000H" and does not operate. 1: Timer performs count operation. However, it does not start counting when TUM1.ECLR1 = "1", until TCLR1 signal is input. When TUM1.ECLR1 = "0", starting counting of timer by CE1 = "1" is triggered by writing "1" to CE1 bit. Therefore, timer is not started even when TUM1.ECLR1 = "0" after CE1 has been set with TUM1.ECLR1 = "1".
4	ETI	External TI1 Input Specifies external or internal count clock. 0: ϕ (internal) 1: TI1 (external)
3, 2	PRS11, PRS10	Prescaler Clock Select Selects internal count clock (ϕm is intermediate clock)
1	PRM11	Prescaler Clock Mode Selects intermediate clock ϕm of count clock (ϕ is system clock). 0: ϕ/2 1: ϕ/4

PRS11	PRS10	Count Clock
0	0	ϕm
0	1	ϕm/4
1	0	ϕm/8
1	1	ϕm/16

Caution: Do not change the count clock frequency while the timer operates.

(3) Timer control register 4 (TMC4)

TMC4 controls the operation of TM4.

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

	7	6	5	4	3	2	1	0	Address	At reset
TMC4	CE4	0	0	0	0	PRS40	PRM41	PRM40	FFFFF342H	00H

Bit Position	Bit Name	Function
7	CE4	Count Enable Controls operation of timer. 0: Timer stops at "0000H" and does not operate. 1: Timer performs count operation.
2	PRS40	Prescaler Clock Select Selects internal count clock (ϕm is intermediate clock). 0: ϕm/16 1: ϕm/32
1, 0	PRM41, PRM40	Prescaler Clock Mode Selects intermediate clock ϕm of count clock (ϕ is system clock).

PRM41	PRM40	ϕm
0	0	ϕ/2
0	1	ϕ/4
1	0	ϕ/8
1	1	RFU (reserved)

Caution: Do not change the count clock frequency while the timer operates.

(4) Timer output control register 1 (TOC1)

TOC1 controls the timer output from the TO10 and TO11 pins.

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

	7	6	5	4	3	2	1	0	Address	At reset
TOC1	ENTO11	ALV11	ENTO10	ALV10	0	0	0	0	FFFFF244H	00H

Bit Position	Bit Name	Function
7, 5	ENTO11, ENTO10	Enable TOxx pin Enables corresponding timer output (TO10, TO11). 0: Timer output is disabled. The inactive levels are output from TO10 and TO11 pins based on the ALV10 and ALV11 pins, respectively. For example, ALV11 is set to 0, TO11 outputs high. Even if coincidence signal is generated from corresponding compare register, levels of TO10 and TO11 pins do not change. 1: Timer output function is enabled. Timer output changes when coincidence signal is generated from corresponding compare register. After the timer output has been enabled before the first coincidence signal is generated, the inactive levels are output from TO10 and TO11 pins based on the ALV10 and ALV11 pins, respectively (For example, ALV11 is set to 0, TO11 outputs high during that period).
6, 4	ALV11, ALV10	Active Level TOxx pin Specifies active level of timer output. 0: Active-low 1: Active-high

Remark: F/F of TO10 and TO11 outputs give priority to reset.

Caution: The TO10 and TO11 outputs are not changed by the external interrupt signal (INTP1n). When using TO10 and TO11, specify a capture/compare register as a compare register (CMS1n = 1).

(5) External interrupt mode register 2 (INTM2) ★

The valid edge of external interrupt INTPn is detected as a capture trigger when CCn (n = 10 to 13) of TM1 is used as a capture register. This valid edge is specified by the INTM2 register (for details, refer to **5.3.6 External interrupt mode registers 1 and 2 (INTM1 and INTM2)**).

(6) Timer overflow status register (TOVS)

This register stores flags that indicate occurrence of an overflow from TM1 and TM4.

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

By testing and resetting the TOVS register via software, occurrence of an overflow can be polled.

	7	6	5	4	3	2	1	0	Address	At reset
TOVS	0	0	0	OVF4	0	0	OVF1	0	FFFFF230H	00H

Bit Position	Bit Name	Function
4, 1	OVFn	Overflow Flag TMn (n = 1, 4) overflow flag. 0: No overflow from TMn 1: Overflow from TMn The INTOV1 maskable interrupt request is also generated and TM1 continues counting. The OVF1 flag is cleared by software. Even though the OVF1 flag and the INTOV1 interrupt request flag are set by the same condition, these two flags are independent of each other. Setting the OVF1 flag in software does not generate an INTOV1 interrupt request. Likewise, setting the INTOV1 interrupt request flag by software does not set the OVF1 flag. Clearing the OVF1 flag by software does not clear the INTOV1 interrupt request and, likewise, when the INTOV1 request flag is cleared by hardware after the interrupt has been serviced, the OVF1 will not be cleared. If an overflow occurs when the TOVS register is being read, the overflow flags will not be updated and the condition will not be seen. However, this overflow condition will be reflected the next time the TOVS register is read.

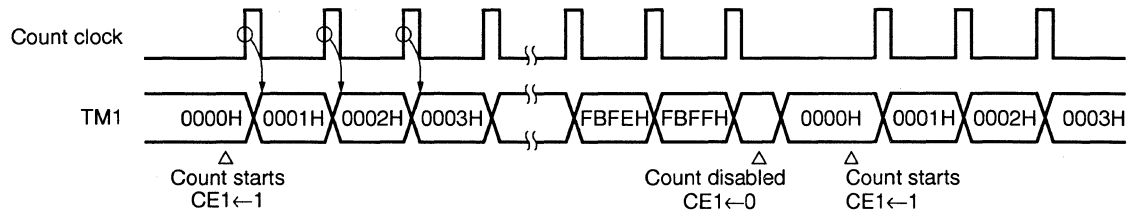
7.4 Timer 1 Operation

7.4.1 Count operation

Timer 1 functions as a 16-bit free-running timer or event counter, as specified by timer control register 1 (TMC1).

When it is used as a free-running timer, and when the count value of TM1 coincides with the value of any of the CC10-CC13 registers, an interrupt signal is generated, and timer output TO×× can be set/reset. In addition, a capture operation that holds the current count value of TM1 and loads it into one of the four registers CC10-CC13, 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 7-1. Basic Operation of Timer 1



7.4.2 Selecting count clock frequency

An internal or external count clock frequency can be input to timer 1. Which count clock frequency is used is specified by the ETI bit of the TMC1 register.

Caution: Do not change the count clock frequency while the timer operates.

(1) Internal count clock (ETI bit = 0)

An internal count clock frequency is selected by the PRM11, PRS11, and PRS10 bits of the TMC1 register, from $\phi/2$, $\phi/4$, $\phi/8$, $\phi/16$, $\phi/32$, and $\phi/64$.

PRS11	PRS10	PRM11	Count Clock
0	0	0	$\phi/2$
0	0	1	$\phi/4$
0	1	0	$\phi 8$
0	1	1	$\phi/16$
1	0	0	$\phi/16$
1	0	1	$\phi/32$
1	1	0	$\phi/32$
1	1	1	$\phi/64$

(2) External count clock (ETI bit = 1)

The signal input to the TI1 pin is counted. At this time, timer 1 can operate as an event counter. The valid edge of TI1 is specified by the TES11 and TES10 bits of the TUM1 register.

TES11	TES10	Valid Edge
0	0	Falling edge
0	1	Rising edge
1	0	RFU (reserved)
1	1	Both rising and falling edges

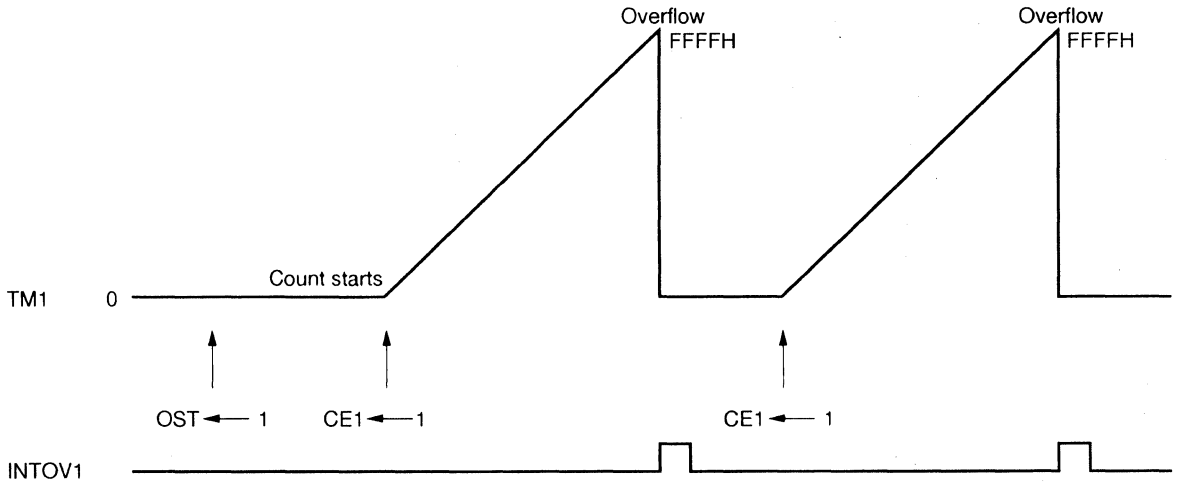
7.4.3 Overflow

If TM1 overflows as a result of counting the external events or internal count clock frequency, the OVF1 bit of the TOVS register is set to 1, and an overflow interrupt (INTOV) is generated.

After the overflow has occurred, the timer can be stopped by setting the OST bit of the TUM1 register to "1". If the timer is stopped due to overflow, the counting operation is not resumed until CE is set to "1" by software.

The operation is not affected even if CE1 is set to 1 during count operation.

Figure 7-2. Operation after Occurrence of Overflow (when ECLR1 = 0, OST = 1)



7.4.4 Clearing/starting timer by TCLR1 input

Timer 1 usually starts the count operation when the CE1 bit of the TMC1 register is set to 1. It is also possible to clear TM1 and start the count operation by using external input TCLR1.

When the valid edge is input to TCLR1 after ECLR1 = 1, OST = 0, and CE1 is set to 1, the count operation is started. If the valid edge is input to TCLR1 during operation, TM1 clears its value and then resumes the count operation (refer to Figure 7-3).

When the valid edge is input to TCLR1 after ECLR1 = 1, OST = 1, and CE1 is set from 0 to 1, the count operation is started. When TM1 overflows, the count operation is stopped once and is not resumed until the valid edge is input to TCLR1. If the valid edge of TCLR1 is detected during count operation, TM1 is cleared and continues counting (refer to **Figure 7-4**). The count operation is not resumed even if CE1 is set to 1 after overflow. ★

Figure 7-3. Clearing/Starting Timer by TCLR1 Input (when ECLR1 = 1, OST = 0)

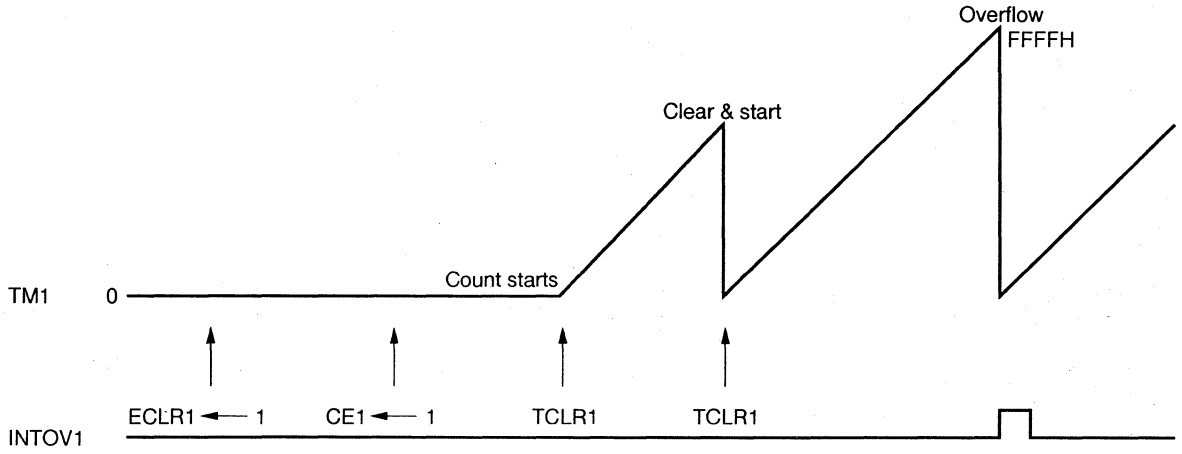
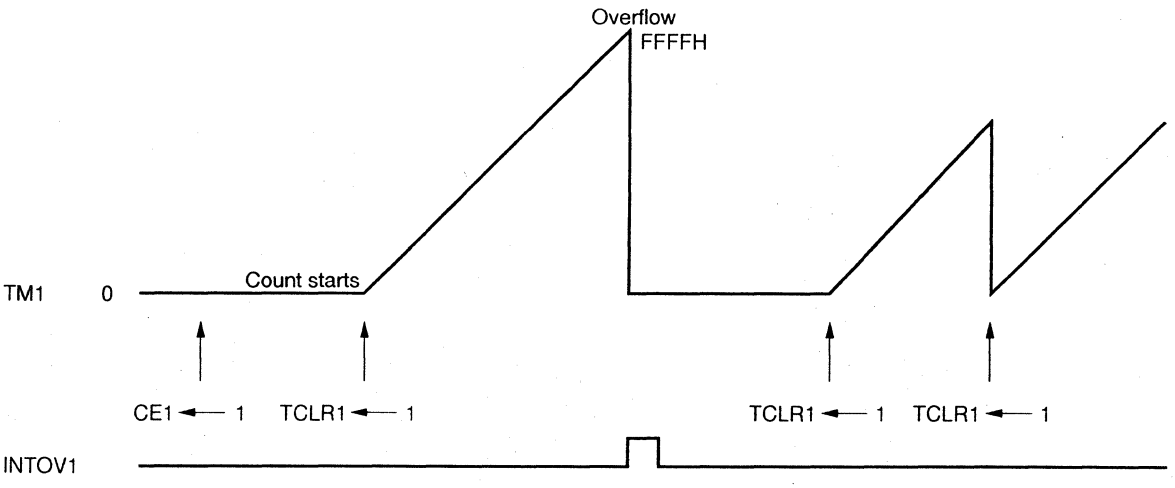


Figure 7-4. Relations between Clear/Start by TCLR1 Input and Overflow (when ECLR1 = 1, OST = 1) ★



7.4.5 Capture operation

A capture operation that captures and holds the count value of TM1 and loads it to a capture register in asynchronization with an external trigger can be performed. The valid edge from the external interrupt request input pin INTPn (n = 10-13) is used as the capture trigger. In synchronization with this capture trigger signal, the count value of TM1 during counting, is captured and loaded to the capture register. The value of the capture register is retained until the next capture trigger is generated.

Interrupt signal INTCCn is generated from INTPn input signal.

Table 7-2. Capture Trigger Signal to 16-Bit Capture Register (TM1)

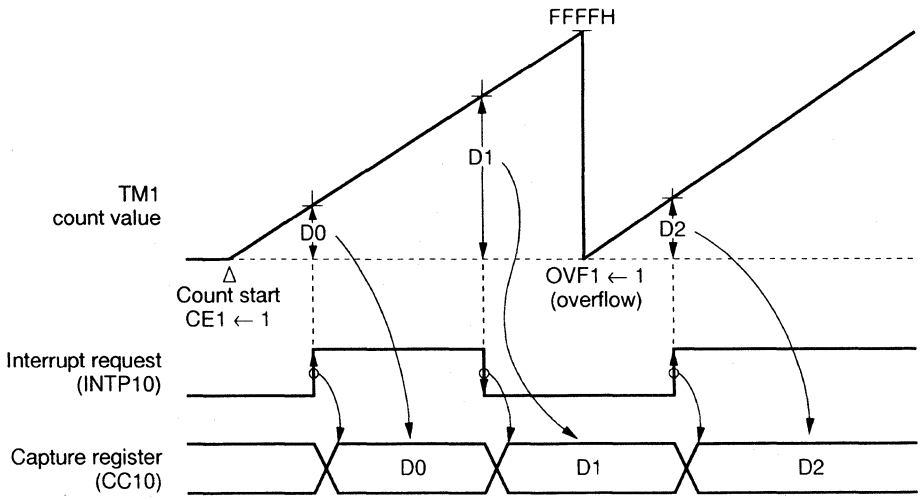
Capture Register	Capture Trigger Signal
CC10	INTP10
CC11	INTP11
CC12	INTP12
CC13	INTP13

Remark: CC10-CC13 are capture/compare registers. Whether these registers are used as capture or compare registers is specified by timer unit mode register 1 (TUM1).

The valid edge of the capture trigger is set by external interrupt mode register (INTM1).

When both the rising and falling edges are specified as the capture trigger, the width of an externally input pulse can be measured. If either the rising or falling edge is specified as the capture trigger, the frequency of the input pulse can be measured.

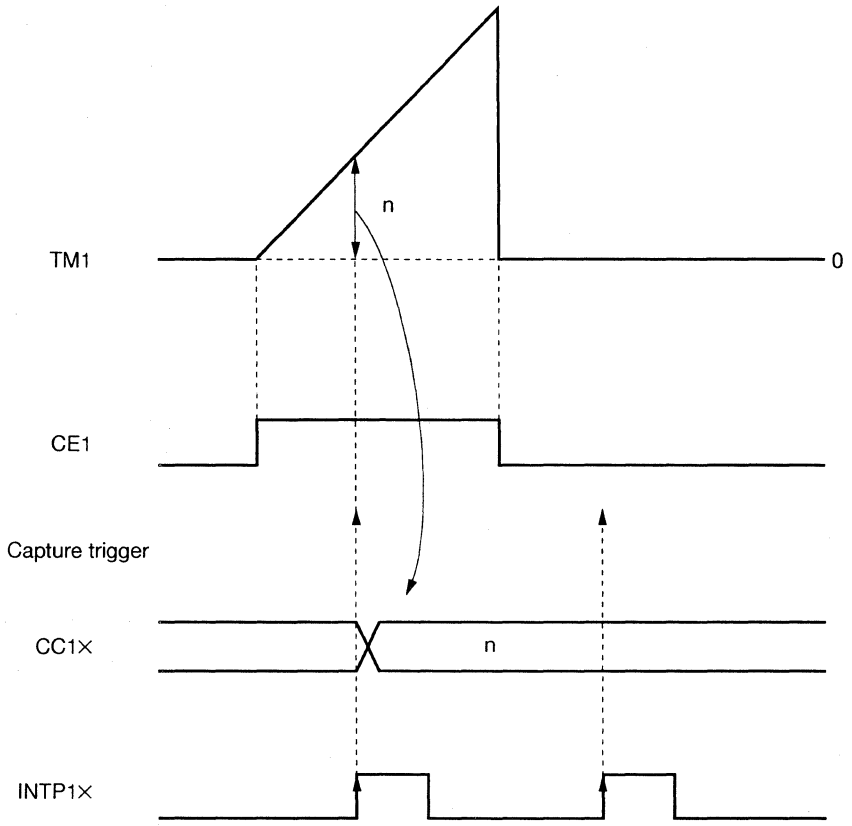
Figure 7-5. Example of TM1 Capture Operation (when both edges are specified)



Remark: Dn (n = 0, 1, 2, ...): count value of TM1

The capture operation is not performed even if the interrupt signal is input when CE1 is cleared to 0.

Figure 7-6. Example of TM1 Capture Operation



Remark: ×: 0-3

7.4.6 Compare operation

A comparison between the value in a compare register with the count value of TM1 can be performed.

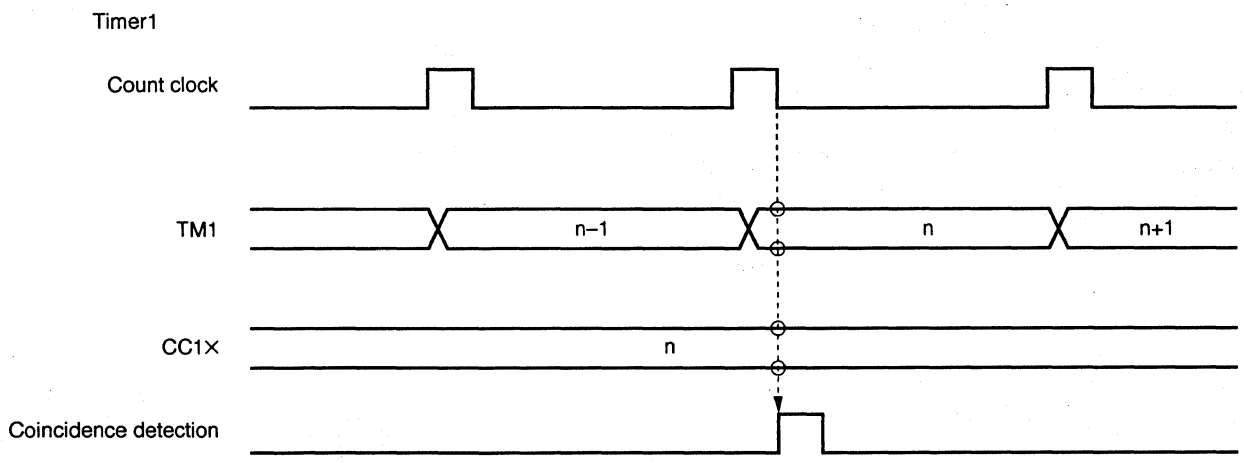
When the count value of TM1 coincides with the value of the compare register programmed in advance, a coincidence signal is sent to the output control circuit (refer to Figure 7-7). The levels of the timer output pins (TO10 and TO11) can be changed by the coincidence signal, and an interrupt request signal can be generated at the same time.

Table 7-3. Interrupt Request Signal from 16-Bit Compare Register (TM1)

Compare Register	Interrupt Request Signal
CC10	INTCC10
CC11	INTCC11
CC12	INTCC12
CC13	INTCC13

Remark: CC10-CC13 are capture/compare registers. Whether these registers are used as capture or compare registers is specified by timer unit mode register 1 (TUM1).

Figure 7-7. Example of Compare Operation



Remark: Note that the coincidence signal is generated immediately after TM1 is incremented as shown above.

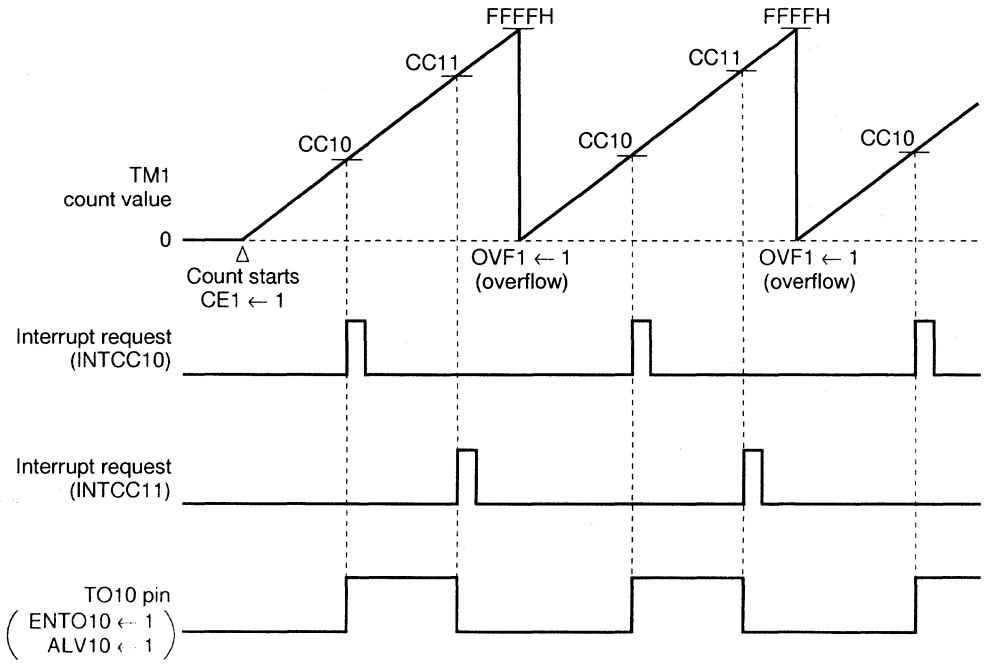
TM1 has two timer output pins: TO10 and TO11.

The count value of TM1 is compared with the value of CC10. When the two values coincide, the output level of the TO10 pin is set. The count value of TM1 is also compared with the value of CC11. When the two values coincide, the output level of the TO10 pin is reset.

Similarly, the count value of TM1 is compared with the value of CC12. When the two values coincide, the output level of the TO11 pin is set. The count value of TM1 is also compared with the value of CC13. When the two values coincide, the output level of TO11 pin is reset.

The output levels of the TO10 and TO11 pins can be specified by the TOC1 register.

Figure 7-8. Example of TM1 Compare Operation (set/reset output mode)



7.5 Timer 4 Operation

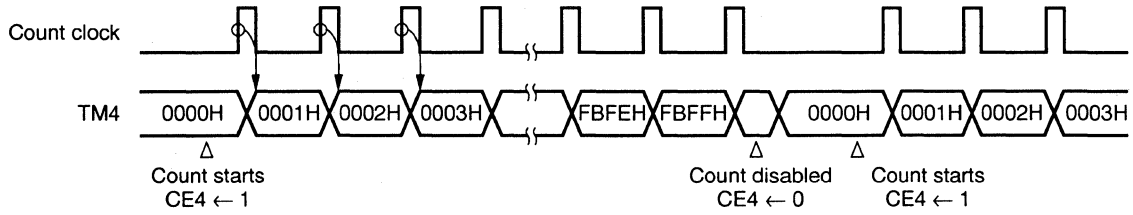
7.5.1 Count operation

Timer 4 functions as a 16-bit interval timer. The operation is specified by the timer control register 4 (TMC4).

The operation of timer 4 counts the internal count clocks (ϕ/32-ϕ/256) specified by the PRS40, PRM41, and PRM40 bits of the TMC4 register.

If the count value of TM4 coincides with the value of CM4, the value TM4 is cleared while simultaneously a coincidence interrupt (INTCM4) is generated.

Figure 7-9. Basic Operation of Timer 4



7.5.2 Selecting the count clock frequency

An internal count clock frequency is selected by the PRS40, PRM40, and PRM41 bits of the TMC4 register, from ϕ/32, ϕ/64, ϕ/128, and ϕ/256.

Caution: Do not change the count clock frequency while the timer operates.

PRS40	PRM40	PRM41	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)

7.5.3 Overflow

If TM4 overflows, the OVF4 bit of the TOVS register is set to 1.

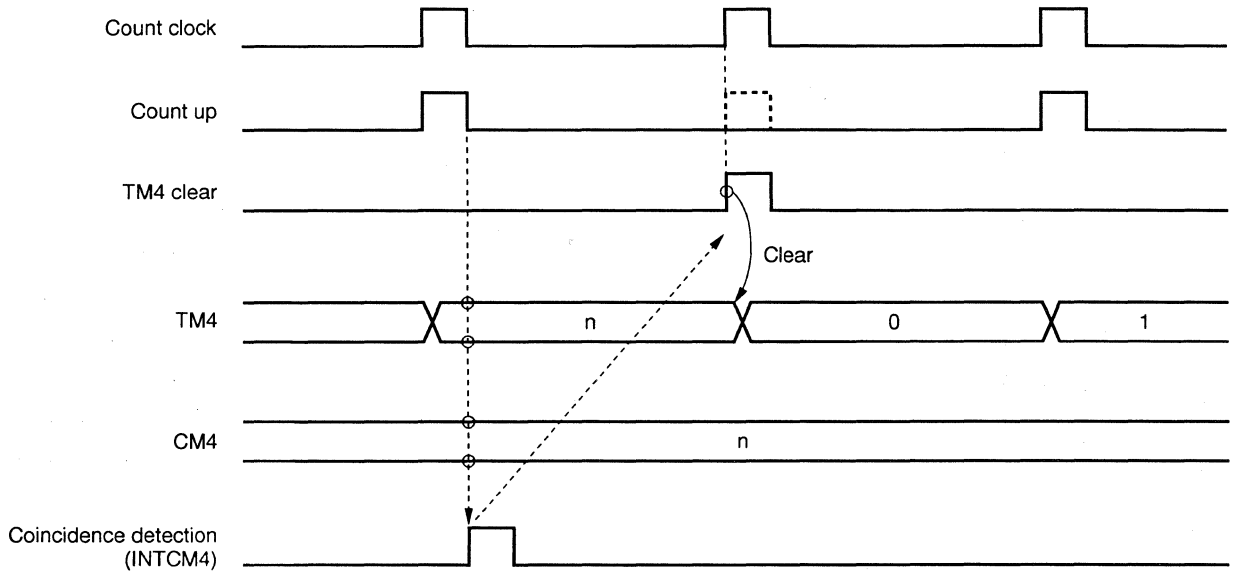
7.5.4 Compare operation

A comparison can be performed with the counter value of TM4 and the compare register (CM4).

When the count value of TM4 coincides with the value of the compare register, a coincidence interrupt (INTCM4) is generated. As a result, TM4 is cleared to 0 at the next count timing (refer to **Figure 7-10**). This function allows timer 4 to be used as an interval timer.

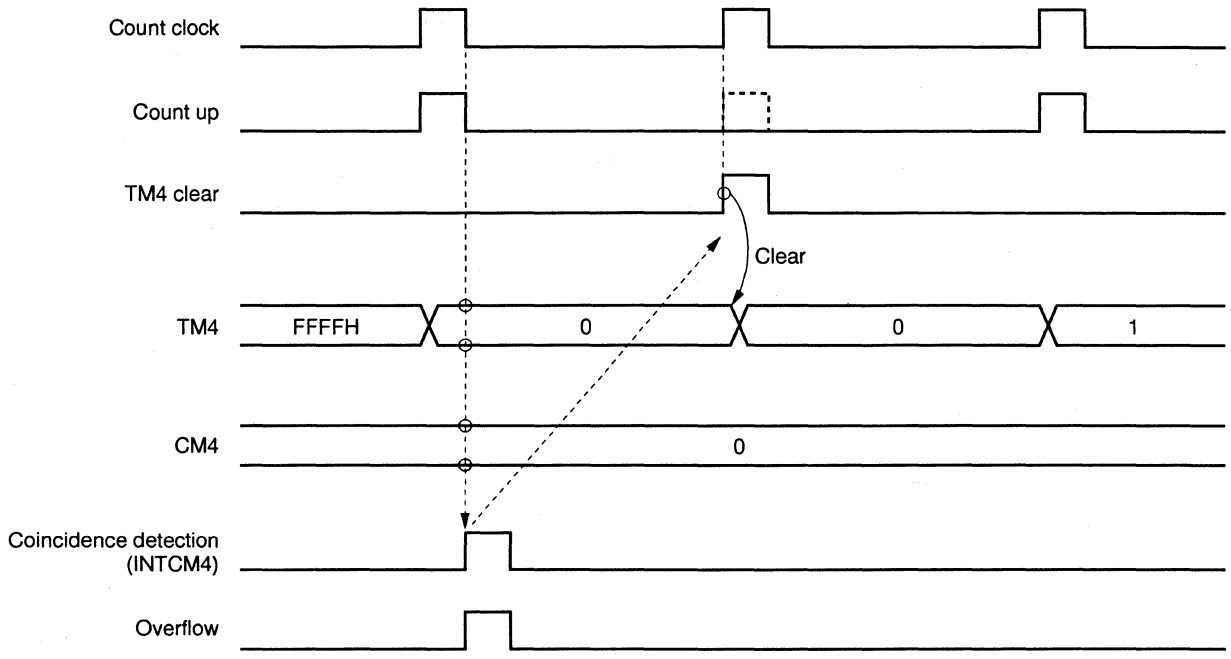
CM4 can be also set to 0. In this case, a coincidence is detected when TM4 overflows and is cleared to 0, and INTCM4 is generated. The value of TM4 is cleared to 0 at the next count timing, but INTCM4 is not generated when a coincidence occurs at this time (refer to **Figure 7-11**).

Figure 7-10. Operation with CM4 at 1-FFFFH



Remark: Interval time = (n+1) × count clock cycle
n = 1-65535 (FFFFH)

Figure 7-11. When CM4 Is Set to 0



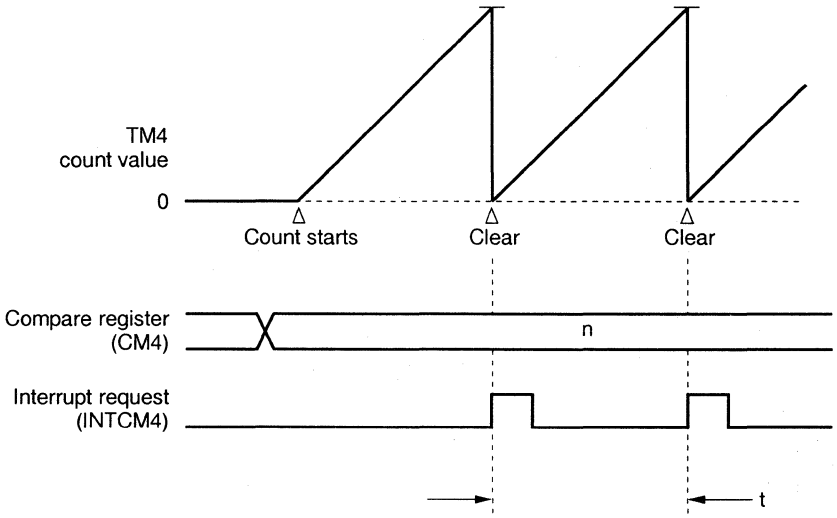
Remark: Interval time = (FFFFH + 2) × count clock cycle

7.6 Application Examples

(1) Operation as interval timer (timer 4)

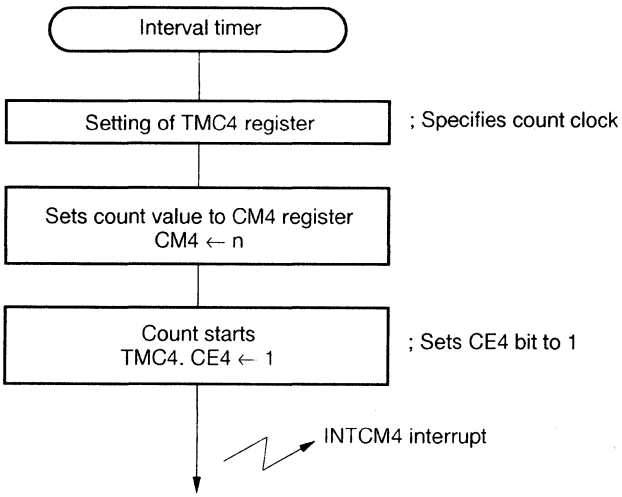
Timer 4 is used as an interval timer that repeatedly generates an interrupt request at time intervals specified by the count value set in advance to compare register CM4. Figure 7-12 shows the timing. Figure 7-13 illustrates the setting procedure.

Figure 7-12. Timing of Interval Timer Operation (timer 4)



Remark: n: value of CM4 register
t: interval time = (n+1) × count clock cycle

Figure 7-13. Setting Procedure of Interval Timer Operation (timer 4)



(2) Pulse width measurement (timer 1)

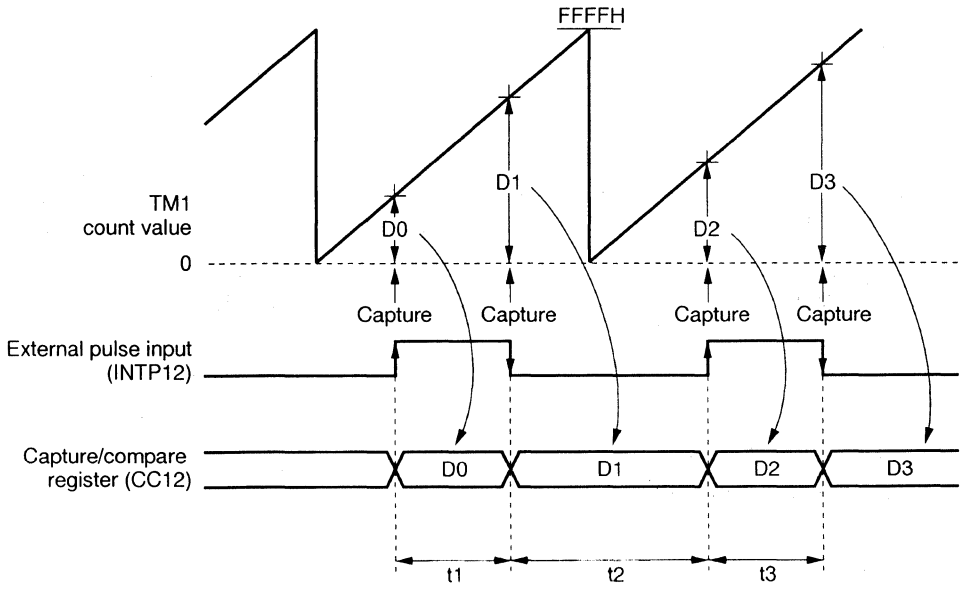
Timer 1 is used to measure pulse width.

In this example, the width of the high or low level of an external pulse input to the INTP12 pin is measured. The value of timer 1 (TM1) is captured to a capture/compare register (CC12) in synchronization with the valid edge of the INTP12 pin (both the rising and falling edges), as shown in Figure 7-14.

To calculate the pulse width, the difference between the count value of TM1 captured to the CC12 register on detection of valid edge n (Dn), and the count value on detection of valid edge (n - 1) (Dn - 1) is calculated. This difference is multiplied by the count clock.

Figure 7-15 shows the setting procedure.

Figure 7-14. Pulse Width Measurement Timing (timer 1)



t1= (D1–D0) × count clock cycle
t2= {(FFFFH–D1) +D2} × count clock cycle
t3= (D3–D2) × count clock cycle

Remark: Dn: count value of TM1 (n = 0, 1, 2, ...)

Figure 7-15. Setting Procedure for Pulse Width Measurement (timer 1)

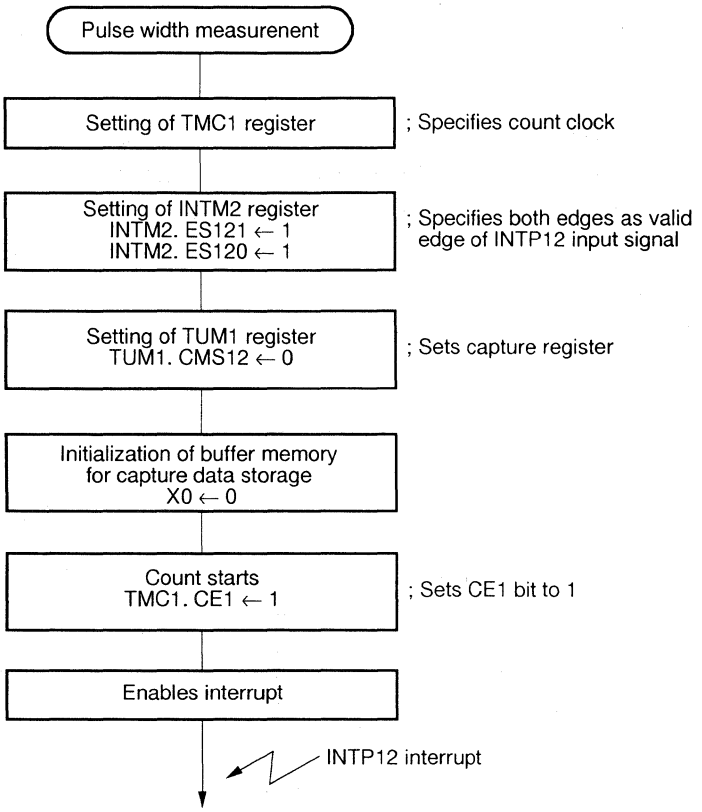
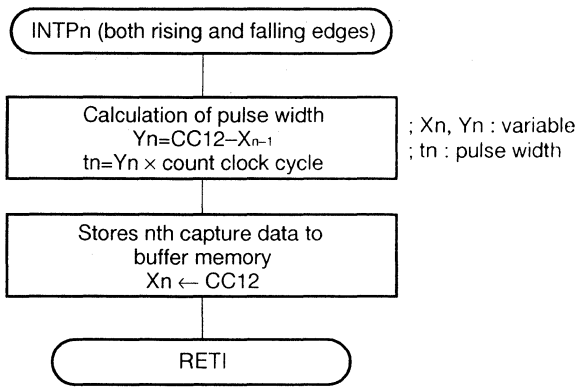


Figure 7-16. Interrupt Request Processing Routine Calculating Pulse Width (timer 1)



Caution: If an overflow occurs two times or more between (n–1)th capture and nth capture, the pulse width cannot be measured.

(3) PWM output (timer 1)

Any square wave can be output to timer output pins (TO10 and TO11) by combining the use of timer 1 and the timer output function.

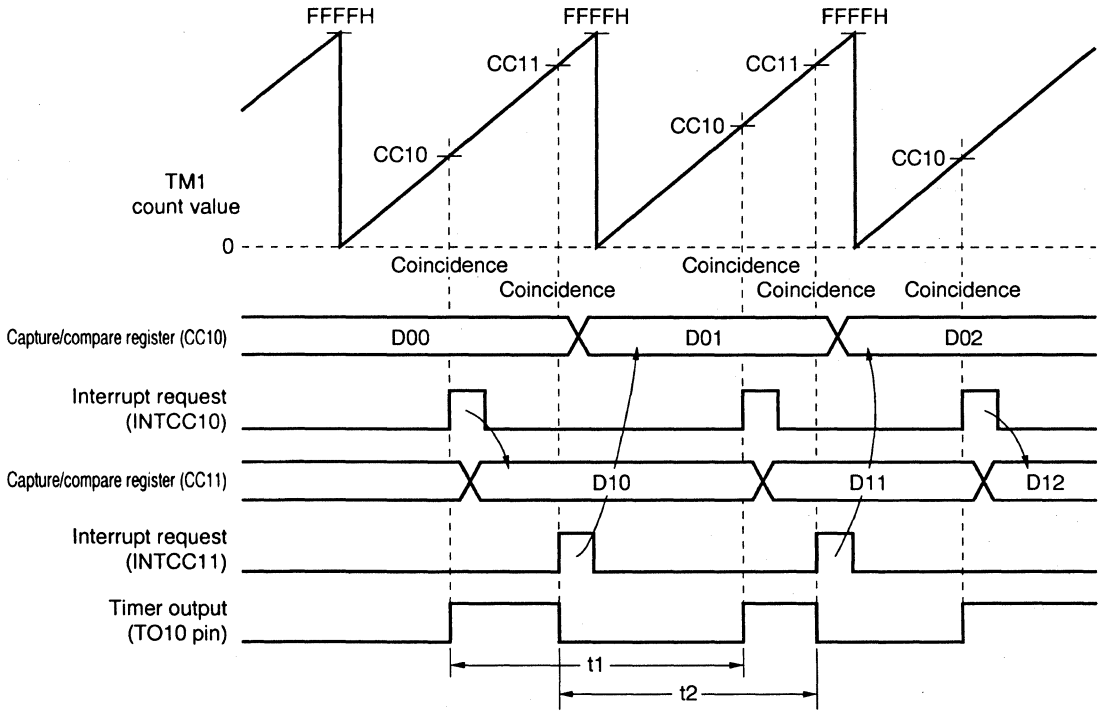
(a) Using timer 1

Two capture/compare registers, CC10 and CC11, are used in this example of PWM output. The output mode of the P21/TO10 pin has been programmed to the set/reset output mode.

A PWM signal with an accuracy of 16 bits can be output from the TO10 pin. Figure 7-17 shows the timing. When timer 1 is used as a 16-bit timer, the rising timing of the PWM output is determined by the value set to capture/compare register CC10, and the falling timing is determined by the value set to capture/compare register CC11.

Figure 7-18 shows the programming procedure at this time.

★ **Figure 7-17. PWM Output Timing (TM1)**



Remark: Dxx: set value of compare register

t1 = {(FFFFH – D00) + D01} × count clock cycle

t2 = {(FFFFH – D10) + D11} × count clock cycle

Figure 7-18. Programming Procedure of PWM Output (timer 1)

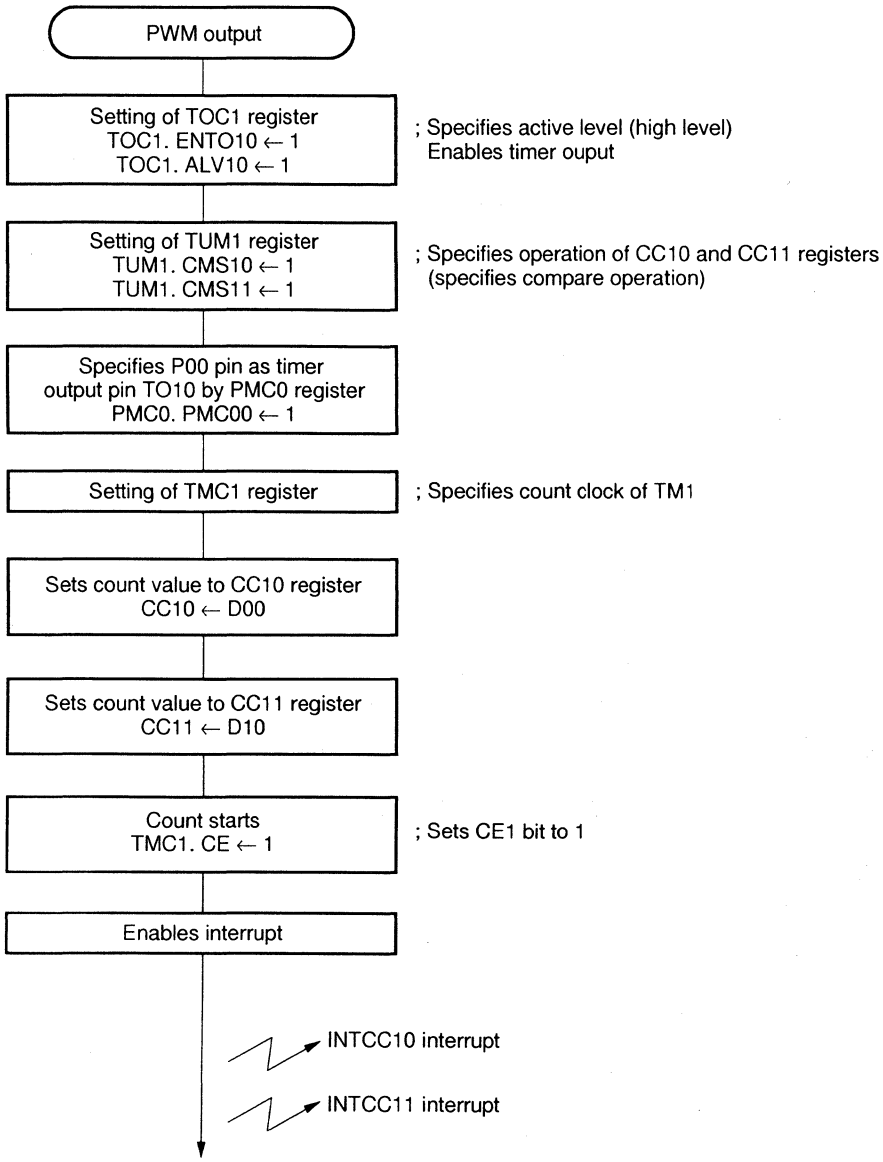
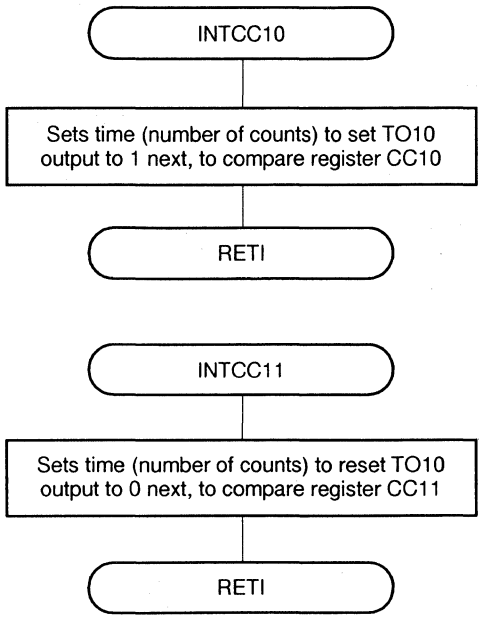


Figure 7-19. Interrupt Request Processing Routine, Modifying Compare Value (timer 1)



(4) Frequency measurement (timer 1)

Timer 1 can be used to measure the cycle or frequency of an external pulse input to the INTPn pin (n = 10-13).

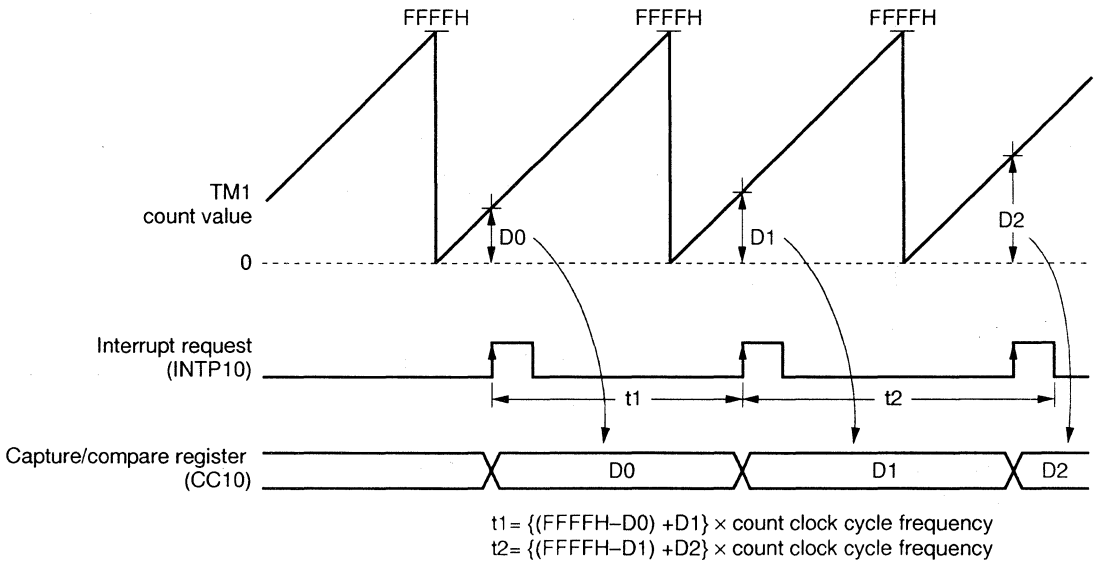
In this example, the frequency of the external pulse input to the INTP10 pin is measured with an accuracy of 16 bits, by combining the use of timer 1 and the capture/compare register CC10.

The valid edge of the INTP10 input signal is specified by the INTM2 register to be the rising edge.

To calculate the frequency, the difference between the count value of TM1 captured to the CC10 register at the nth rising edge (Dn), and the count value captured at the (n–1)th rising edge (Dn–1), is calculated, and the value multiplied by the count clock frequency.

Figure 7-21 shows the setting procedure at this time.

Figure 7-20. Frequency Measurement Timing(TM1)



Remark: Dn: count value of TM1 (n = 0, 1, 2, ...)

Figure 7-21. Set-up Procedure for Frequency Measurement (timer 1)

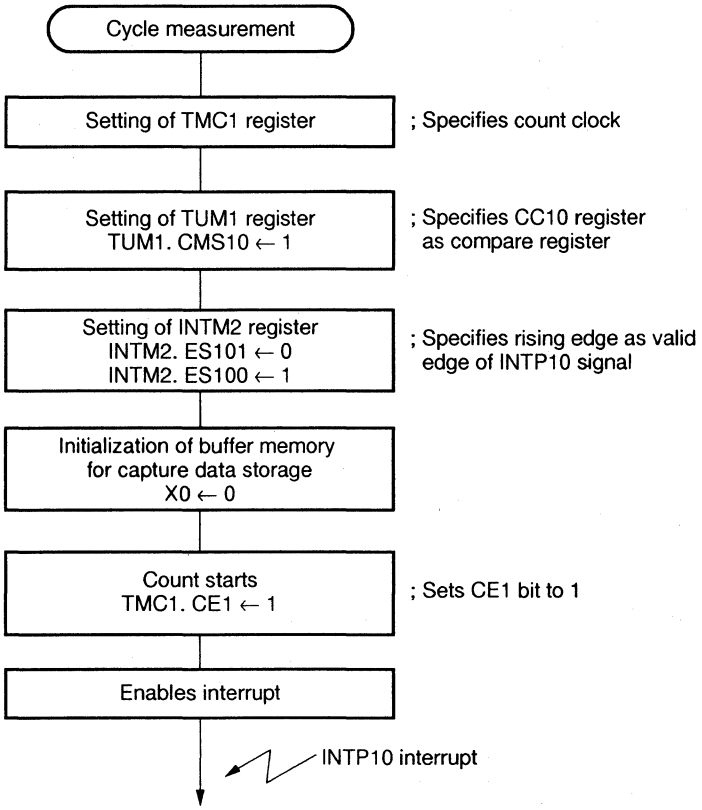
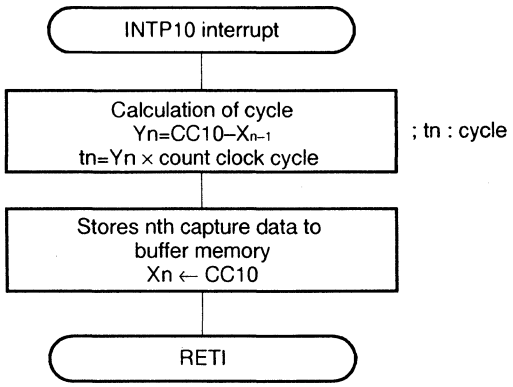


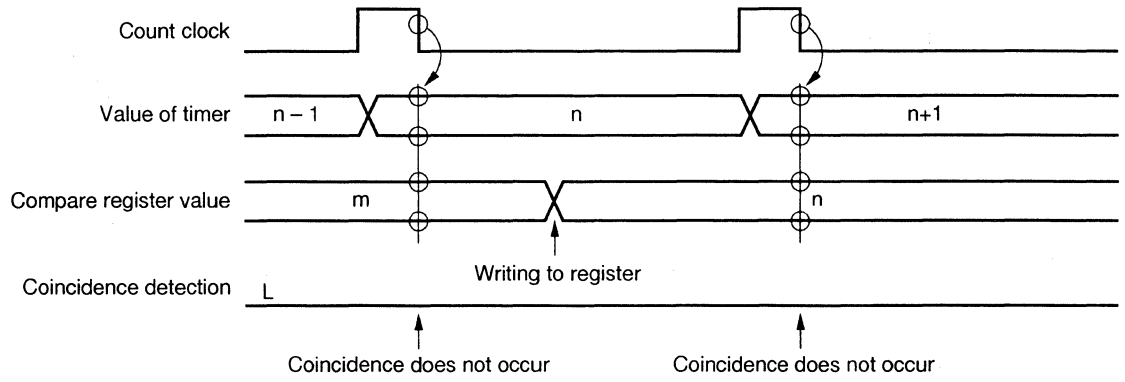
Figure 7-22. Interrupt Request Processing Routine Calculating Cycle (timer 1)



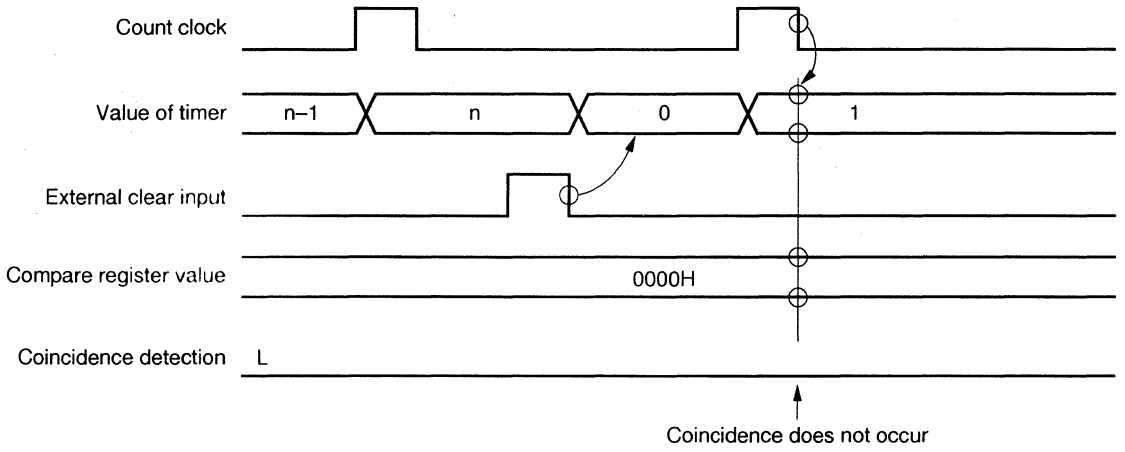
7.7 Note

Coincidence is detected by the compare register immediately after the timer value matches the compare register value, and does not take place in the following cases:

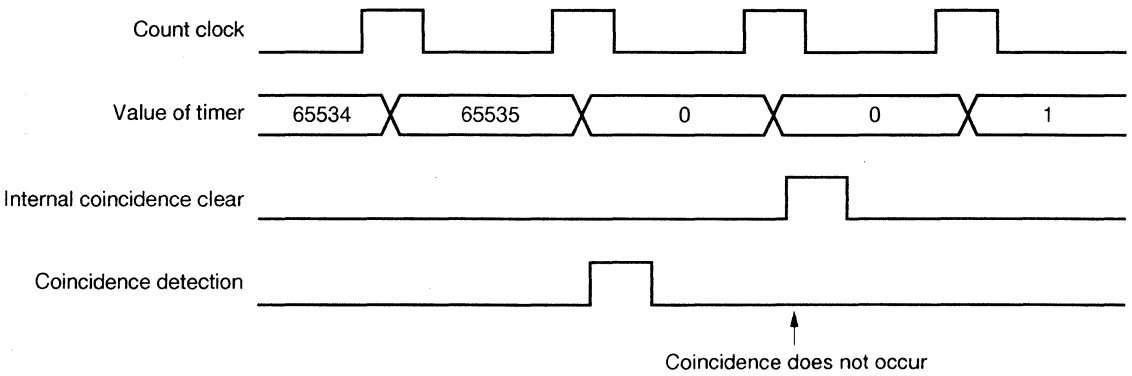
(1) When compare register is rewritten (TM1, TM4)



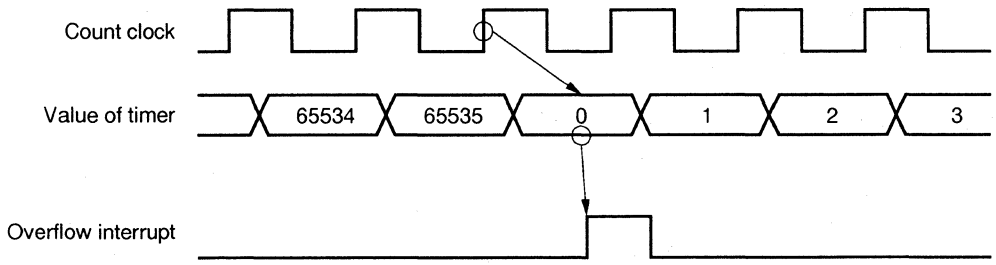
(2) When timer is cleared by external input (TM1)



(3) When timer is cleared (TM4)



When timer 1 is used as a free-running timer, the timer value is cleared to 0 when the timer overflows.



CHAPTER 8 SERIAL INTERFACE FUNCTION

8.1 Features

The V851 provides two types serial interfaces which operate as two independent peripheral units, each of which has one channel.

(1) Asynchronous serial interface (UART)
(2) Clocked serial interface (CSI)

The UART transmits/receives 1-byte serial data following a start bit and can perform full-duplex communication.

The CSI uses three signal lines to high-speed synchronous data transfers data (3-wire serial I/O): serial clock ($\overline{SCK}$), serial input (SI), and serial output (SO) lines.

8.2 Asynchronous Serial Interface (UART)

8.2.1 Features

★ ○ Transfer rate: 150 bps to 76800 bps (at ϕ = 33 MHz)
○ Full-duplex communication
○ Two-pin configuration: TXD: transmit data output pin
RXD: receive data input pin
○ Receive error detection function
- Parity error
- Framing error
- Overrun error

○ Three interrupt sources
- Receive error interrupt (INTSER0)
- Reception completion interrupt (INTSR0)
- Transmission completion interrupt (INTST0)

○ Character length of transmit/receive data is specified by ASIM00 and ASIM01 registers.
○ Character length: 7, 8 bits
9 bits (when extended)
○ Parity function: odd, even, 0, none
○ Transmit stop bit: 1, 2 bits
○ Internal baud rate generator

8.2.2 Configuration of asynchronous serial interface

The asynchronous serial interface is controlled by asynchronous serial interface mode register (ASIM) and asynchronous serial interface status register (ASIS). The receive data is stored in receive buffer (RXB), and the transmit data is written to transmit shift register (TXS).

Figure 8-1 shows the configuration of the asynchronous serial interface.

(1) Asynchronous serial interface mode registers (ASIM00, ASIM01)

ASIM00 and ASIM01 are 8-bit registers that specify the operation of the asynchronous serial interface.

(2) Asynchronous serial interface status register (ASIS0)

ASIS0 is a register containing flags that indicate receive errors, if any, and a transmit status flag. Each receive error flag is set to 1 when a receive error occurs, and is reset to 0 when data is read from the receive buffer (RXB0, RXB0L), or when new data is received (if the next data contains an error, the corresponding error flag is set).

The transmit status flag is set to 1 when transmission is started, and reset to 0 when transmission ends.

(3) Reception control parity check

The reception operation is controlled according to the contents programmed in the ASIM00 and ASIM01 registers. During the receive operation, errors such as parity error are also checked. If an error is found, the appropriate value is set to the ASIS0 register.

(4) Receive shift register

This shift register converts the serial data received on the RXD pin into parallel data. When it receives 1 byte of data, it transfers the receive data to the receive buffer.

The receive shift register cannot be accessed by the CPU.

(5) Receive buffer (RXB0, RXB0L)

RXB0 is a 9-bit buffer register that holds receive data. If data of 7 or 8 bits/character is received, 0 is stored to the most significant bit position of this register.

If this register is accessed in 16-bit units, RXB0 is specified. To access in lower 8-bit units, RXB0L is specified.

While reception is enabled, the receive data is transferred from the receive shift register to the receive buffer in synchronization with shift-in processing of 1 frame.

When the data is transferred to the receive buffer, a reception completion interrupt request (INTSR0) occurs.

(6) Transmit shift register (TXS0, TXS0L)

TXS0 is a 9-bit shift register used for transmit operation. When data is written to this register, the transmission operation is started.

A transmission complete interrupt request (INTST0) is generated after each complete data frame is trasmitted.

When this register is accessed in 16-bit units, TXS0 is specified. To access in lower 8-bit units, TXS0L is specified.

For related information, refer to section 8.2.5"Operation".

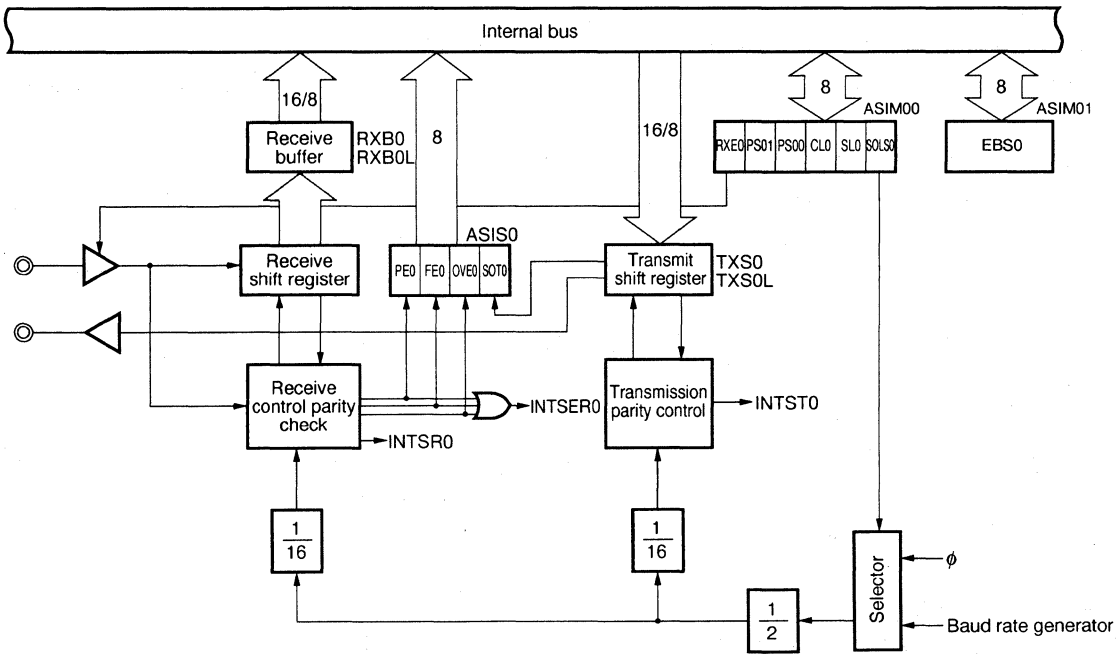
(7) Transmission parity control

A start bit, parity bit, and stop bit are appended to the data written to the TXS0 register, according to the contents programmed in the ASIM00 and ASIM01 registers, to control the transmission operation.

(8) Selector

Selects the source of the serial clock.

Figure 8-1. Block Diagram of Asynchronous Serial Interface



8.2.3 Mode registers and control registers

(1) Asynchronous serial interface mode registers (ASIM00 and ASIM01)

The 8-bit asynchronous serial interface mode register, ASIM00 and ASIM01, specify the serial clock source, the number of stop bits, the character length of one frame of data, and the type of parity bit control for the send and receive operations.

These registers specify the transfer mode of the UART.

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

	7	6	5	4	3	2	1	0	Address	At reset
ASIM00	1	RXE0	PS01	PS00	CL0	SL0	0	SCLS0	FFFFF0C0H	80H

Bit Position	Bit Name	Funciton
6	RXE0	Receive Enable Enables/disables reception. 0: Disables reception 1: Enables reception When reception is disabled, the receive shift register does not detect the start bit. Data is not shifted into the receive shift register and neither is any transfer to the receive buffer performed. Therefore, the previous contents of receive buffer are retained. When reception is enabled, the data is shifted into the receive shift register and transferred to the receive buffer when one complete frame has been received. A reception completion interrupt (INTSR0) is generated in synchronization with the transfer to the receive buffer.

<table>
<tr><th>Bit Position</th><th>Bit Name</th><th>Function</th></tr>
<tr><td>5, 4</td><td>PS01, PS00</td><td>Parity Select
Specifies parity bit.
<table>
<tr><th>PS01</th><th>PS00</th><th>Operation</th></tr>
<tr><td>0</td><td>0</td><td>No parity. Extended bit operation</td></tr>
<tr><td>0</td><td>1</td><td>0 parity
Transmission side → Transmits with parity bit 0
Reception side → Does not generate parity error on reception</td></tr>
<tr><td>1</td><td>0</td><td>Odd parity</td></tr>
<tr><td>1</td><td>1</td><td>Even parity</td></tr>
</table>
• Even parity
Parity bit is set to "1" when number of bits equal to one in received data is odd. If number of bits that are one is even, parity bit is cleared to 0. In this way, number of bits that are "1" in transmit data and parity bit is controlled to become even. During reception, number of bits that are "1" in receive data and parity bit are counted. If it is odd, parity error occurs.
• Odd parity
In contrast to even parity, number of bits included in transmit data and parity bit that are "1" is controlled to become odd.
During reception, parity error occurs if the number of "1"'s in the receive data and parity bit are added up to become even.
• 0 parity
Parity bit is cleared to "0" during transmission, regardless of transmit data.
During reception, the parity bit is not checked. Therefore, parity error does not occur regardless of whether parity bit is "0" or "1".
• No parity
No parity bit is appended to the transmit data.
Reception is performed on assumption that there is no parity bit. Because no parity bit is used, parity error does not occur.
Extended bit operation can be specified by EBS0 bit of ASIM01 register.</td></tr>
<tr><td>3</td><td>CL0</td><td>Character Length
Specifies character length of one frame.
0: 7 bits
1: 8 bits</td></tr>
<tr><td>2</td><td>SL0</td><td>Stop Bit Length
Specifies stop bit.
0: 1 bit
1: 2 bits</td></tr>
</table>

Bit Position	Bit Name	Function
0	SCLS0	Serial Clock Source Specifies serial clock. 0: Specified by baud rate generator and BPRM0 (baud rate generator prescaler mode register) 1: ϕ/2 • When SCLS0 = 1 ϕ/2 (system clock) is selected as serial clock source. In asynchronous mode, baud rate is expressed as follows because sampling rate of x16 is used: Baud rate = $\frac{\phi/2}{16}$ bps Value of baud rate when typical clock is used based on above expression is as follows:

ϕ	33 MHz	25 MHz	20 MHz	16 MHz	12.5 MHz	10 MHz	8 MHz	5 MHz
Baud rate	1031 K	781 K	625K	500 K	390 K	312 K	250 K	156 K

★

Caution: The operation of UART is not guaranteed if the bits 0-6 of this register are changed while UART is transmitting/receiving data.

	7	6	5	4	3	2	1	0	Address	At reset
ASIM01	0	0	0	0	0	0	0	EBS0	FFFFF0C2H	00H

Bit Position	Bit Name	Function
0	EBS0	Extended Bit Select Specifies extended bit operation of transmit/receive data when no parity is specified (PS01, PS00 = 00). 0: Disables extended bit operation 1: Enables extended bit operation When extended bit operation is enabled, 1 data bit is appended as most significant bit to 8-bit transmit/receive data, and therefore 9-bit data is communicated. Extended bit operation is valid only when no parity is specified by ASIM00 register. If zero, even, or odd parity is specified, specification by EBS0 bit is invalid, and extended bit is not appended.

(2) Asynchronous serial interface status register 0 (ASIS0)

This register contains three error flags that indicate the receive error status for each character received and the status of the transmit shift register.

The error flags always indicate the status of an error that has occurred most recently. If two or more errors occur before the current received data, only the status of the error that has occurred last is retained.

If a receive error occurs, read the receive buffer RXB0 or RXB0L after reading the ASIS0 register, and then clear the error flag.

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

	7	6	5	4	3	2	1	0	Address	At reset
ASIS0	SOT0	0	0	0	0	PE0	FE0	OVE0	FFFFF0C4H	00H

Bit Position	Bit Name	Function
7	SOT0	Status Of Transmission Status flag that indicates transmission operation status. Set (1) : Beginning of transmission of a data frame (writing to TXS register) Clear (0): End of transmission of a data frame (occurrence of INTST0) When serial data transfer begins, this flag will indicate if the transmit shift register is ready to be written or not.
2	PE0	Parity Error Status flag that indicates parity error. Set (1) : Transmit parity and receive parity do not match Clear (0): No error; this flag is automatically cleared to 0 when the data is read from the receive buffer.
1	FE0	Framing Error Status flag that indicates framing error. Set (1) : Stop bit is not detected Clear (0): No error; this flag is automatically cleared to 0 when the data is read from the receive buffer.
0	OVE0	Overrun Error Status flag that indicates overrun error. Set (1) : Overrun error; Contents of the receive shift register are transferred to the receive buffer before the previous data has been read by the CPU. This will cause an over writing of data and the previous informaiton will be lost. Clear (0): No error; this flag is automatically cleared to 0 when the data is read from the receive buffer. Because contents of receive shift register are transferred to receive buffer each time one frame of data has been received, if overrun error occurs, next receive data is written over contents of receive buffer, and previous receive data is discarded.

(3) Receive buffers (RXB0 and RXB0L)

RXB0 is a 9-bit buffer register that holds the receive data. When 7- or 8-bit/character is received, the higher bit of this register is 0.

When this register is accessed in 16-bit units, RXB0 is specified. To access in lower 8-bit units, RXB0L is specified.

When reception is enabled, the receive data is transferred from the receive shift register to the receive buffer when one complete frame of data (or character) has been received.

When the receive data is transferred to the receive buffer, a reception completion interrupt request (INTSR0) occurs.

When reception is disabled, the data is not shifted into the receive shift register and the reception completion interrupt is not generated. The previous contents of the receive buffer are retained.

RXB0 enables 16-bit read access only, and RXB0L enables 8-/1-bit read access only.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	At reset
RXB0	0	0	0	0	0	0	0	RXB0	RXB07	RXB06	RXB05	RXB04	RXB03	RXB02	RXB01	RXB00	FFFFF0C8H	Undefined

	7	6	5	4	3	2	1	0	Address	At reset
RXB0L	RXB07	RXB06	RXB05	RXB04	RXB03	RXB02	RXB01	RXB00	FFFFF0CAH	Undefined

Bit Position	Bit Name	Function
8	RXEB0	Receive Extended Buffer Extended bit when 9-bit/character is received. This bit is cleared to zero when 7- or 8-bit/character is received.
7-0	RXB0n (n=7-0)	Receive Buffer These bits store receive data. The RXB07 bit is cleared to zero when 7-bit/character is received.

(4) Transmit shift registers (TXS0, TXS0L)

TXS0 is a 9-bit shift register for data transmission. The transmit operation is started when data is written to this register.

Transmission complete interrupt request (INTST0) is generated after each complete data frame is transmitted.

When this register is accessed in 16-bit units, TXS0 is specified. To access in lower 8-bit units, TXS0L is specified.

TXS0 enables 16-bit write access only, and TXS0L enables 8-bit write access only.

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	Address	At 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	Address	At reset
TXS0L	TXS07	TXS06	TXS05	TXS04	TXS03	TXS02	TXS01	TXS00	FFFFF0CEH	Undefined

Bit Position	Bit Name	Function
8	TXED0	Transmit Extended Data Extended bit on transmission of 9-bit/character
7-0	TXS0n (n=7-0)	Transmit Shifter Writes transmit data.

Caution: Note that the UART of the V851 does not have a transmit buffer. This means that an interrupt request is generated in synchronization with the end of transmission of one frame of data. This operation is different from that of some other NEC microcontrollers which have transmit buffers. They generate interrupt requests on the completion of transmission (completion of transfer to buffer).

8.2.4 Interrupt request

UART generates the following three types of interrupt requests:

- Receive error interrupt
- Reception completion interrupt
- Transmission completion interrupt

Of these three, the receive error interrupt has the highest default priority, followed by the reception completion interrupt and transmission completion interrupt.

Table 8-1. Default Priority of Interrupts

Interrupt	Priority
Receive error	1
Reception completion	2
Transmission completion	3

(1) Receive error interrupt (INTSER0)

A receive error interrupt occurs as a result of ORing the three types of receive errors described in description of the ASIS0 register when reception is enabled.

This interrupt does not occur when reception is disabled.

(2) Reception completion interrupt (INTSR0)

The reception completion interrupt occurs if data is received in the receive shift register and then transferred to the receive buffer when reception is enabled.

This interrupt also occurs when a receive error occurs, but the receive error interrupt has the higher priority. The reception completion interrupt does not occur when reception is disabled.

(3) Transmission completion interrupt (INTST0)

Because the UART of the V851 does not have a transmit buffer, a transmission completion interrupt occurs when one frame of transmit data containing a 7-/8-/9-bit character is shifted out from the transmit shift register. The transmission completion interrupt is output when the last bit of data has been transmitted.

8.2.5 Operation

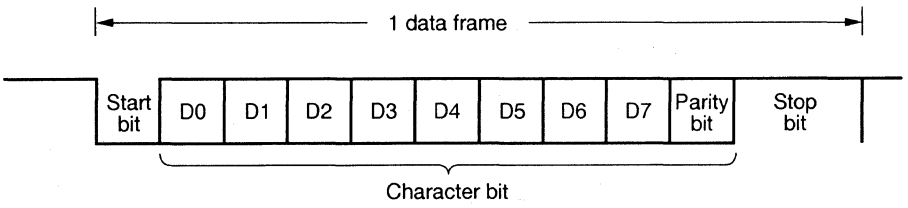
(1) Data format

Full-duplex serial data is transmitted/received.

One data frame of the transmit/receive data consists of a start bit, character bits, parity bit, and stop bit, as shown in Figure 8-2.

The length of the character bit, parity, and the length of the stop bit in one data frame are specified by the asynchronous serial interface mode registers (ASIM00 and ASIM01).

Figure 8-2. Format of Transmit/Receive Data of Asynchronous Serial Interface



- Start bit ..... 1 bit
- Character bit ..... 7/8/9 bits (with extended bit)
- ★ Parity/extended bit ..... Even/odd/0/none/extended bit
- Stop bit ..... 1/2 bits

(2) Transmission

Transmission is started when data is written to the transmit shift register (TXS0 or TXS0L). The next data is written to the TXS0 or TXS0L register by the service routine of the transmission completion interrupt (INTST0).

(a) Transmission enabled status

The UART of the V851 is always enabled to transmit data. Because the V851 does not have a pin that inputs a transmit enable signal, such as a CTS pin, a general input port is used when it is necessary to check whether the other party is ready to receive data.

(b) Starting transmission

Transmission is started by writing data to the transmit shift register (TXS0, TXS0L). The transmit data is transferred starting from the start bit with the LSB first. The start bit, parity bit, and stop bit are automatically appended.

(c) Transmission interrupt request

When one frame of data or character has been completely transferred, a transmission completion interrupt request (INTST0) occurs.

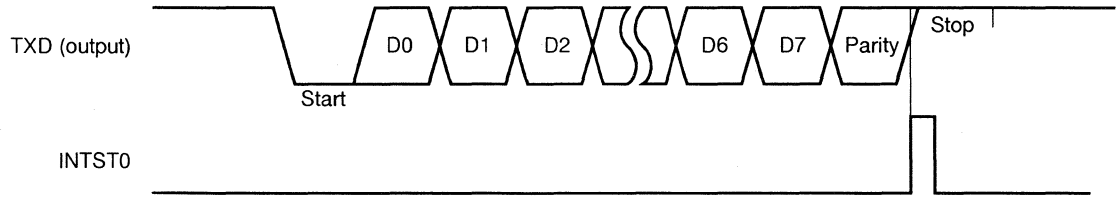
Unless the data to be transmitted next is written to the TXS0 or TXS0L register, the transmission is aborted. The communication rate drops unless the next transmit data is written to the TXS0 or TXS0L register immediately after transmission has been completed.

Cautions: 1. The transmission completion interrupt request (INTST0) is generated after each complete data frame is transmitted out of the transmit shift register. It is not generated by the empty state of TXS0 or TXS0L. Because of this, the INTST0 interrupt will not be generated immediately after reset.

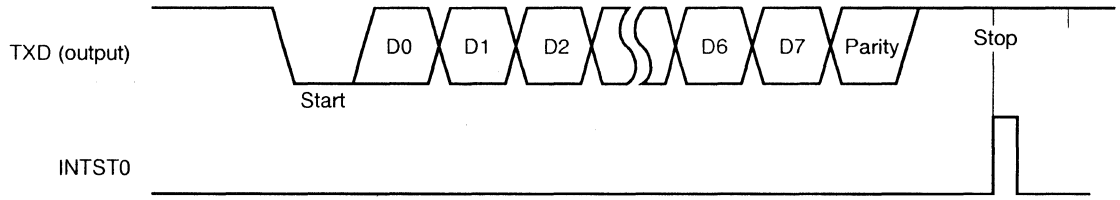
2. During the transmit operation, writing data into the TXS0 or TXS0L register is ignored (the data is discarded) until INTST0 is generated.

Figure 8-3. Asynchronous Serial Interface Transmission Completion Interrupt Timing

(a) Stop bit length: 1



(b) Stop bit length: 2



(3) Reception

When reception is enabled, sampling of the RXD pin is started, and reception of data begins when the start bit is detected. Each time one frame of data or character has been received, the reception completion interrupt (INTSR0) occurs. Usually, the receive data is transferred from the receive buffer (RXB0, RXB0L) to memory by this interrupt processing.

(a) Reception enabled status

Reception is enabled when the RXE0 bit of the ASIM00 register is set to 1.

RXE0 = 1: Reception is enabled
RXE0 = 0: Reception is disabled

When reception is disabled, the receive hardware stands by in the initial status.
At this time, the reception completion interrupt/receive error interrupt does not occur, and the contents of the receive buffer are retained.

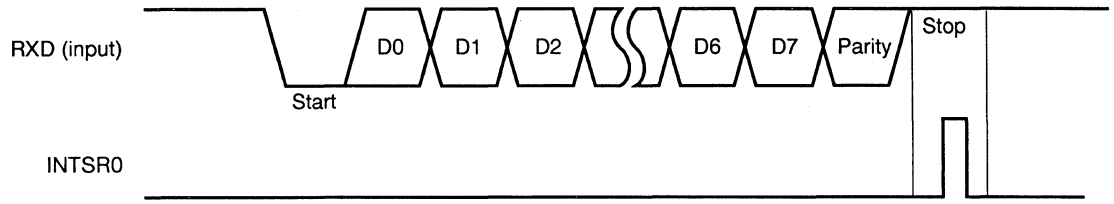
(b) Starting reception

Reception is started when the start bit is detected.
The RXD pin is sampled with the serial clock specified by the ASIM00 register. The RXD pin is sampled again eight clocks after the falling edge of the RXD pin has been detected. If the RXD pin is low at this time, it is recognized as the start bit, and reception is started. After that, the RXD pin is sampled in 16 clock ticks.
If the RXD pin is high eight clocks after the falling edge of the RXD pin has been detected, this falling edge is not recognized as the start bit. The serial clock counter is reinitialized, and the UART waits for the input of the next falling edge or valid start bit.

(c) Reception completion interrupt request

When one frame of data has been received with RXE0 = 1, the receive data in the shift register is transferred to RXB0, and a reception completion interrupt request (INTSR0) is generated.
If an error occurs, the receive data that contains an error is transferred to the receive buffer (RXB0, RXB0L), and the transmission completion interrupt (INTSR0) and receive error interrupt (INTSER0) occur simultaneously.
When the RXE0 bit is reset to 0 during reception, the receive operation is immediately disabled. The contents of the receive buffer (RXB0, RXB0L) and asynchronous serial interface status register (ASIS0) are not changed, and the reception completion interrupt (INTSR0) and receive error interrupt (INTSER0) will not be generated.

Figure 8-4. Asynchronous Serial Interface Reception Completion Interrupt Timing



(d) Receive error flag

Three error flags, parity error, framing error, and overrun error flags, are related with the reception operation.

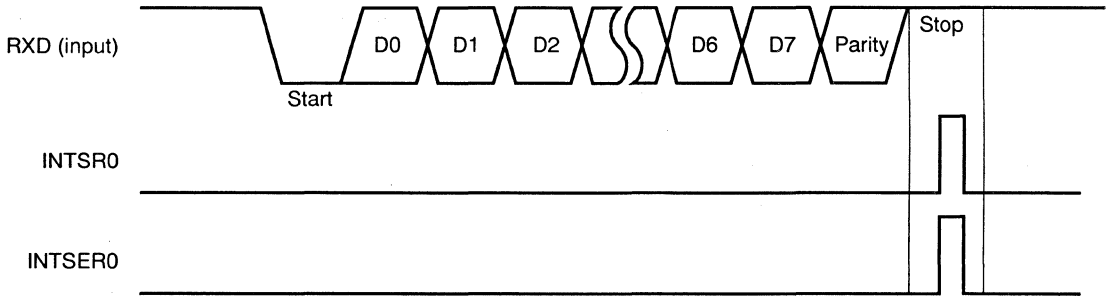
The receive error interrupt request occurs as a result of ORing these three error flags.

By reading the contents of the ASIS0 register. The error which caused the receive error interrupt (INTSER0) can be identified.

The contents of the ASIS0 register are reset to 0 when the receive buffer (RXB0, RXB0L) is read or the next data frame is received (if the next data contains an error, the corresponding error flag is set).

Receive Error	Error Cause
Parity Error	Parity specified during transmission does not coincide with parity of receive data
Framing error	Stop bit is not detected
Overrun error	Next data is completely received before data is read from receive buffer

Figure 8-5. Receive Error Timing



8.3 Clocked Serial Interface (CSI)

8.3.1 Features

★ ○ High transfer speed: 8.25 Mbps max. (with $\phi/4$, at ϕ = 33 MHz)
○ Half duplex communication
○ Character length: 8 bits
○ MSB first/LSB first selectable
○ External serial clock input/internal serial clock output selectable
○ 3 lines: SO : serial data output
SI : serial data input
$\overline{SCK}$: serial clock I/O
○ Interrupt source: 1
• Interrupt request signal (INTCSI0)

The clocked serial interface is controlled by the clocked serial interface mode register (CSIM0). The transmit/receive data is read/written from/to the SIO0 register.

(1) Clocked serial interface mode register (CSIM0)

CSIM0 is an 8-bit register that specifies the operation of the clocked serial interface.

(2) Shift register (SIO0)

SIO0 is an 8-bit register that converts serial data into parallel data, and vice versa. SIO0 is used for both transmission and reception.

Data is shifted in (received) or shifted out (transmitted) from the MSB or LSB side.

The actual transmitting and receiving of data is actually performed by writing data to and reading data from the SIO0 register.

(3) Serial clock selector

Selects the serial clock to be used.

(4) Serial clock control circuit

Controls supply of the serial clock to the shift register. When the internal clock is used, it also controls the clock output to the $\overline{SCK}$ pin.

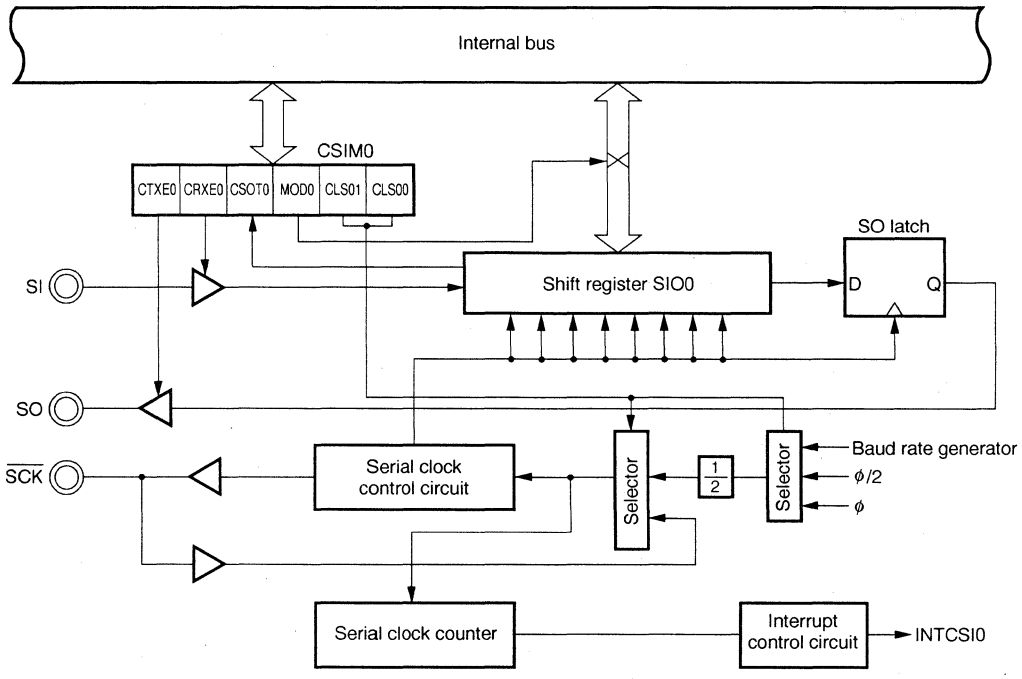
(5) Serial clock counter

Counts the serial clocks being output and the serial clocks received during transmission/reception to check whether 8-bit data has been transmitted or received.

(6) Interrupt signal generation control circuit

Controls whether an interrupt request is generated when the serial clock counter has counted eight serial clocks.

8.3.2 Configuration



8.3.3 Mode registers and control registers

(1) Clocked serial interface mode register 0 (CSIM0)

This register specifies the basic operation mode of CSI.

It can be read/written in 8- or 1-bit units (note, however, that bit 5 can only be read).

	7	6	5	4	3	2	1	0	Address	At reset
CSIM0	CTXE0	CRXE0	CSOT0	0	0	MOD0	CLS01	CLS00	FFFFF088H	00H

Bit Position	Bit Name	Function
7	CTXE0	CSI Transmit Enable Enables or disables transmission. 0: Disables transmission 1: Enables transmission When CTXE0 = "0", output buffers of both SO and SI pins go into high-impedance state.
6	CRXE0	CSI Receive Enable Disables or enables reception. 0: Disables reception 1: Enables reception If serial clock is received when transmission is enabled (CTXE0 = 1) and reception is disabled, "0" is input to shift register.
5	CSOT0	CSI Status Of Transmission Indicates that transfer operation is in progress. Set (1): Transfer start timing (writing to SIO0 register) Clear (0): Transfer end timing (INTCSI occurs) This bit is used to check whether writing to serial I/O shift register (SIO0) is permitted or not. Serial data transfer is started by enabling transmission (CTXE0 = 1).
2	MOD0	Bit Order Mode Specifies first bit. 0: MSB first 1: LSB first
1, 0	CLS01, CLS00	Clock Source Specifies serial clock.

CLS01	CLS00	Specifies serial clock		$\overline{SCK}$ pin
0	0	External clock		Input
0	1	Internal clock	Specified by BPRM0 register[Note1]	Output
1	0		$\phi/4$[Note2]	Output
1	1		$\phi/2$[Note2]	Output

Notes: **1.** For setting of BPRM0 register, refer to section 8.4 "Baud Rate Generator (BRG)".

2. $\phi/4$ and $\phi/2$ indicate one fourth ard a half of system clock frequency, respectively.

(2) Serial I/O shift register 0 (SIO0)

This register converts 8-bit serial data into parallel data, and vice versa. The actual transmitting and receiving of data is performed by writing data to and reading data from the SIO0 register.

A shift operation is performed when CTXE0 = "1" or CRXE0 = "1".

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

	7	6	5	4	3	2	1	0	Address	At reset
SIO0	SIO07	SIO06	SIO05	SIO04	SIO03	SIO02	SIO01	SIO00	FFFFF08AH	Undefined

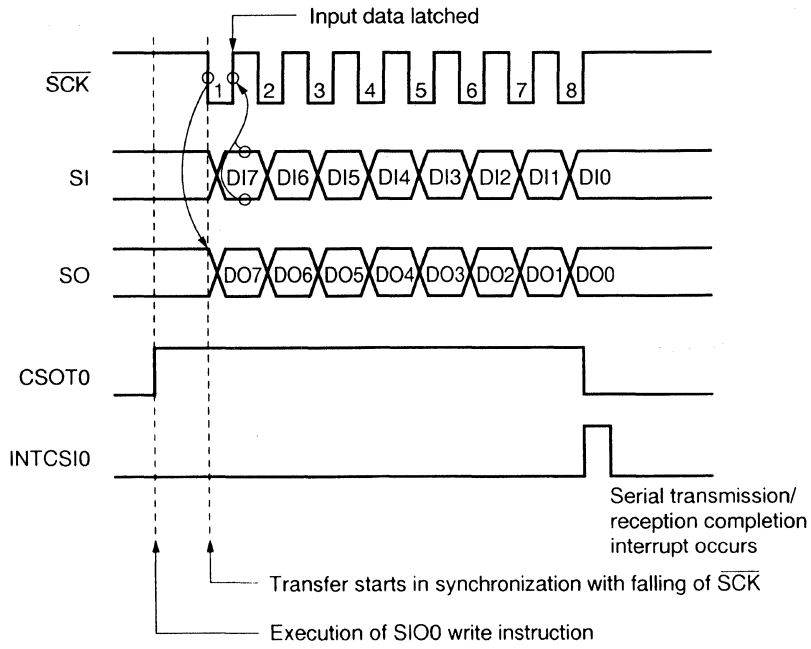
Bit Position	Bit Name	Function
7-0	SIO0n (n=7–0)	Serial I/O Data is shifted in (received) or out (transmitted) from MSB or LSB side.

8.3.4 Basic operation

(1) Transfer format

The CSI of the V851 performs interfacing by using three lines: one clock line and two data lines.
Serial transfer is started by executing an instruction that writes transfer data to the SIO0 register.
During transmission, the data is output from the SO pin in synchronization with the falling edge of $\overline{SCK}$.
During reception, the data input to the SI pin is latched in synchronization with the rising of $\overline{SCK}$. $\overline{SCK}$ stops when the serial clock counter overflows (at the rising of the 8th count), and $\overline{SCK}$ remains high until the next data transmission or reception is started. At the same time, an interrupt request signal (INTCSI0) is generated.

Caution: Data should be sent after changing the CTXE value for successful transfer. If CTXE is changed from 0 to 1 after the transmit data is sent to the shift register, serial transfer will not begin.



(2) Enabling transmission/reception

The CSI of the V851 has only one 8-bit shift register and does not have a buffer. Transmission and reception are therefore performed simultaneously.

(a) Transmission/reception enabling condition

When CTXE0 = 1, transmission is enabled.
When CRXE0 = 1, reception is enabled.
When CTXE0 = CRXE0 = 1, transmission/reception is enabled.

(i) Disabling SIO0 output by CTXE0

When CTXE0 = 0, the serial output pin goes into a high-impedance state.
When CTXE0 = 1, the data of the shift register is output.

(ii) Disabling SIO0 input by CRXE0

When CRXE0 = 0, the shift register input is "0".
When CRXE0 = 1, the serial input data is input to the shift register.

(iii) To check transmit data

To receive the transmit data and to check whether bus contention occurs, set CTXE0 and CRXE0 to 1.

(b) Starting transmission/reception

Transmission/reception is started by reading/writing the shift register (SIO0). Transmission/reception is controlled by setting the transmission enable bit (CTXE0) and reception enable bit (CRXE0) as follows:

CTXE0	CRXE0	Start Condition
0	0	Does not start
0	1	Reads shift register
1	0	Writes shift register
1	1	Writes shift register
0	0 → 1	Rewrites CRXE0 bit

In the above table, note that these bits should be set in advance of data transfer. For example, if CTXE0 is not changed from 0 to 1 before reading data from or writing data to the shift register, transfer will not begin. The bottom of the table means that, if CRXE0 is changed from 0 to 1 when CTXE0 is "0", the serial clock will be generated to initiate receive operation.

8.3.5 Transmission in 3-wire serial I/O mode

Transmission is started when data is written to the SIO0 register after transmission has been enabled by the clocked serial interface mode register (CSIM0).

(1) Starting transmission

Transmission is started by writing the transmit data to the shift register after the CTXE0 bit of the clocked serial interface mode register (CSIM0) has been set (the CRXE0 bit is cleared to "0").

If the CTXE0 bit is reset to 0, the SO pin goes into a high-impedance state.

(2) Transmitting data in synchronization with serial clock

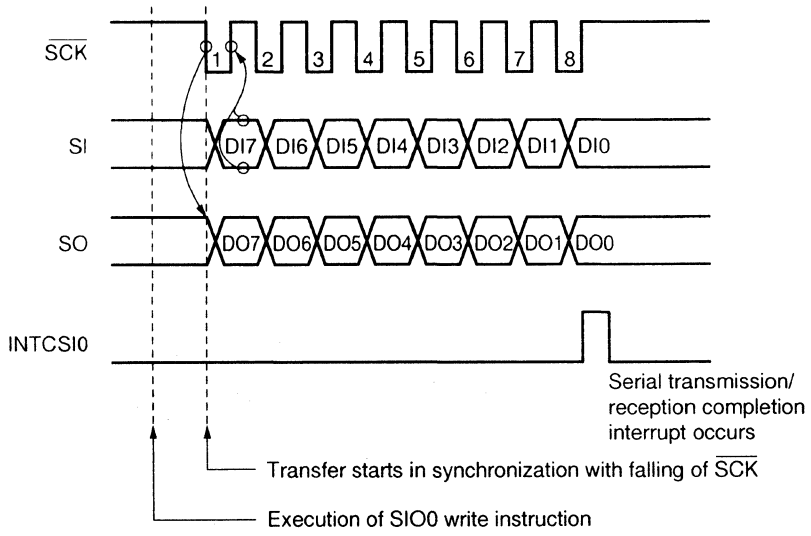
(a) When internal clock is selected as serial clock

When transmission is started, the serial clock is output from the $\overline{SCK}$ pin, and at the same time, data is sequentially output to the SO pin from SIO0 in synchronization with the falling edge of the serial clock.

(b) When external clock is selected as serial clock

When transmission is started, the data is sequentially output from SIO0 to the SO pin in synchronization with the falling of the serial clock input to the $\overline{SCK}$ pin immediately after transmission has been started. The shift operation is not performed even if the serial clock is input to the $\overline{SCK}$ pin if transmission is not enabled, and the output level of the SO pin will not change.

Figure 8-6. Timing of 3-Wire Serial I/O Mode (transmission)



8.3.6 Reception in 3-wire serial I/O mode

Reception is started if the status is changed from reception disabled to reception enabled status by the clocked serial interface mode register (CSIM) or if the SIO0 register is read by the CPU with reception enabled.

(1) Starting reception

Reception can be started in the following two ways:

<1> Changing the status of the CRXE0 bit of the CSIM0 register from "0" (reception disabled) to "1" (reception enabled)

<2> Reading the receive data from the shift register (SIO0) when the CRXE0 bit of the CSIM0 register is "1" (reception enabled)

Note that receive operation is initiated by the change of CRXE0 from 0 to 1. For example, if CRXE0 has already been set to "1", writing "1" to this bit does not initiate receive operation. In this case, CRXE0 must be set to "0" beforehand. When CRXE0 = 0, the shift register input is "0".

(2) Receiving data in synchronization with serial clock

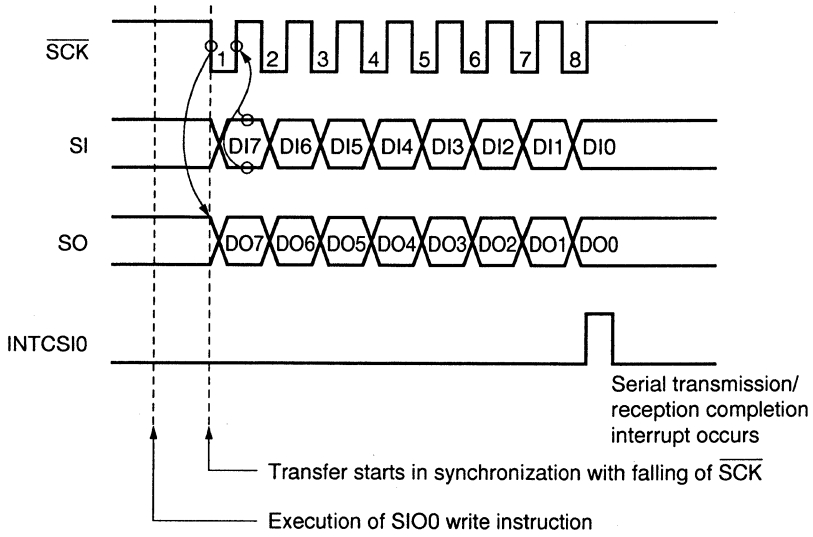
(a) When internal clock is selected as serial clock

When reception is started, the serial clock is output from the $\overline{SCK}$ pin, and at the same time, data is sequentially loaded from the SI pin to SIO0 in synchronization with the rising edge of the serial clock.

(b) When external clock is selected as serial clock

When reception is started, the data is sequentially loaded from the SI pin to SIO0 in synchronization with the rising of the serial clock input to the $\overline{SCK}$ pin immediately after reception has been started. The shift operation is not performed even if the serial clock is input to the $\overline{SCK}$ pin when reception is not enabled.

Figure 8-7. Timing of 3-Wire Serial I/O Mode (reception)



8.3.7 Transmission/reception in 3-wire serial I/O mode

Transmission and reception can be executed simultaneously if both transmission and reception are enabled by the clocked serial interface mode register (CSIM0).

(1) Starting transmission/reception

Transmission and reception can be performed simultaneously (transmission/reception operation) when both the CTXE0 and CRXE0 bits of the clocked serial interface mode register (CSIM0) are set to 1.
Transmission/reception can be started in the following two ways:

<1> By changing the status of the CRXE0 bit from "0" (reception disabled) to "1" (reception enabled) when the CTXE0 bit of the CSIM0 register is "1" (transmission enabled)
<2> By writing the transmit data to the shift register (SIO0) when both the CTXE0 and CRXE0 bits of the CSIM0 register are "1" (transmission/reception enabled)

Note that transmit/receive operation is initiated by the change of CRXE0 of CSIM0 register from 0 to 1. For example, if CRXE0 has already been set to "1", writing "1" to this bit does not initiate transmit/receive operation. In this case, CRXE0 must be set to "0" beforehand.

(2) Transmitting data in synchronization with serial clock

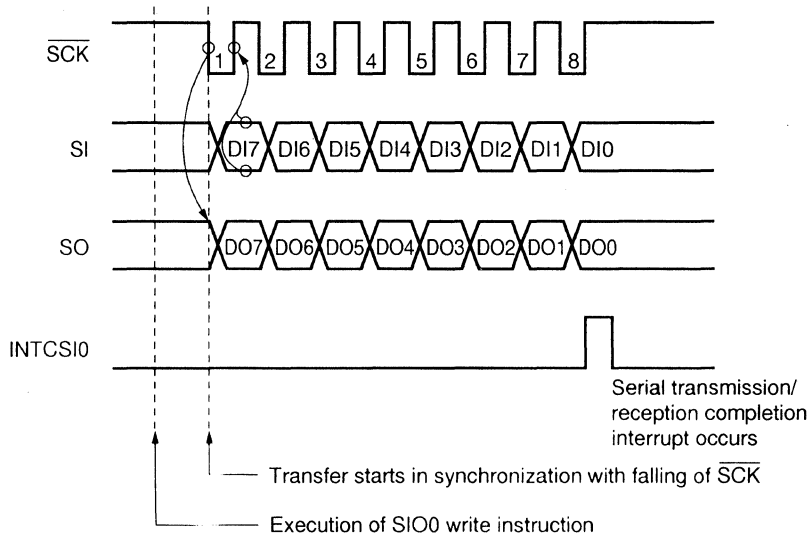
(a) When internal clock is selected as serial clock

When transmission/reception is started, the serial clock is output from the $\overline{SCK}$ pin, and at the same time, data is sequentially set to the SO pin from SIO0 in synchronization with the falling edge of the serial clock. Simultaneously, the data of the SI pin is sequentially loaded to SIO0 in synchronization with the rising edge of the serial clock.

(b) When external clock is selected as serial clock

When transmission/reception is started, the data is sequentially output from SIO0 to the SO pin in synchronization with the falling edge of the serial clock input to the $\overline{SCK}$ pin immediately after transmission/reception has been started. The data of the SI pin is sequentially loaded to SIO0 in synchronization with the rising edge of the serial clock. The shift operation is not performed even if the serial clock is input to the $\overline{SCK}$ pin when transmission/reception is not enabled, and the output level of the SO pin does not change.

Figure 8-8. Timing of 3-Wire Serial I/O Mode (transmission/reception)



Caution: When transmission/reception is started the first time, the CRXE0 bit always changes its states from "0" to "1".

Transmission/reception is therefore started immediately. In this case, the chances are that undefined data is output. Therefore, enable transmission/reception by writing the first transmit data to the SIO0 register in advance, when both transmission and reception are disabled (when both the CTXE0 and CRXE0 bits are reset to 0).

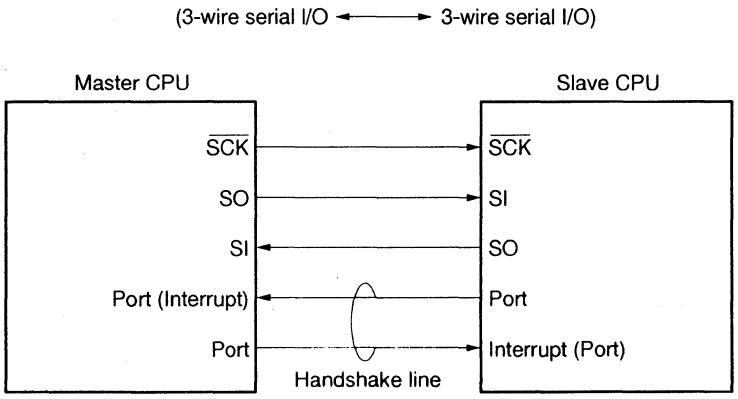
8.3.8 System Configuration Example

Data 8 bits long is transferred by using three signal lines: serial clock ($\overline{SCK}$), serial input (SI), and serial output (SO). This feature is effective for connecting peripheral I/Os and display controllers that have a conventional clocked serial interface.

To connect two or more devices, a handshake line is necessary.

Various devices can be connected, because it can be specified whether the data is transmitted starting from the MSB or LSB.

Figure 8-9. Example of CSI System Configuration



8.4 Baud Rate Generator (BRG)

8.4.1 Configuration and function

The internal baud rate, generator can provide the serial clock for the UART and CSI. The baud rate generator uses an 8-bit counter (TMBRG), prescaler, and a compare register (BRGO) to generate the serial clock.

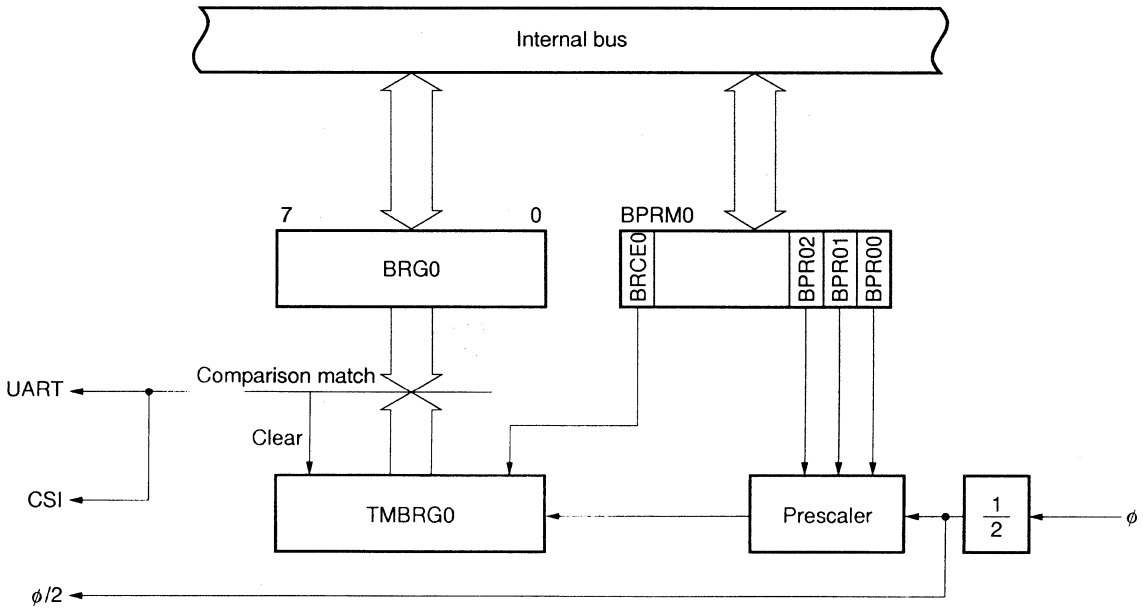
The serial interface can use the output of the internal baud rate generator or ϕ (system clock) as the serial clock.

The serial clock source for the UART is specified by the SCLS0 bit of the ASIM00 register. The serial clock source for the CSI is specified by the CLS00 and CLS01 bits of the CSIM0 register.

When the output of the baud rate generator is specified, the baud rate generator will be used as the clock source.

Because the serial clock for transmission/reception is shared by both the transmission and reception portions, the same baud rate is used for both transmission and reception.

Figure 8-10. Block Diagram



(1) Dedicated baud rate generator (BRG)

The dedicated baud rate generator BRG consists of an 8-bit timer (TMBRG) that generates a serial clock for transmission/reception, a compare register (BRG0), and a prescaler.

(a) Input clock

System clock ϕ is input to the BRG.

(b) Set-up value of BRG

(i) UART

If the dedicated baud rate generator is specified for UART, the actual baud rate can be calculated by the following expression, because a sampling rate of x16 is used:

$$\text{Baud rate} = \frac{\phi}{2 \times m \times 2^n \times 16 \times 2} \text{ [bps]}$$

where,

ϕ : system clock frequency [Hz]
m : BRG0 set-up value ($1 \leq m \leq 256$) (256 is set by writing 0 to the BRG register.)
n : BRG prescaler set-up value (n = 0, 1, 2, 3, 4)

(ii) CSI

If the dedicated baud rate generator is specified for CSI, the actual baud rate can be calculated by the following expression:

$$\text{Baud rate} = \frac{\phi}{2 \times m \times 2^n \times 2} \text{ [bps]}$$

where,

ϕ : system clock frequency [Hz]
m : BRG0 set-up value ($1 \leq m \leq 256$) (256 is set by writing 0 to the BRG register.)
n : BRG prescaler set-up value (n = 0, 1, 2, 3, 4)

Table 8-2 shows the set-up values of the baud rate generator when the typical clocks are used:

(c) Error of baud rate generator

The error of the baud rate generator is calculated as follows:

$$\text{Error [\%]} = \left(\frac{\text{Actual baud rate (baud rate with error)}}{\text{Desired baud rate (normal baud rate)}} - \right) \times 100$$

Example: (9520/9600–1) × 100 = –0.833 [%]
(5000/4800–1) × 100 = +4.167 [%]

(2) Allowable error range of baud rate generator

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.

Table 8-2. BRG Set-up Values ★

Baud Rate [bps]		ϕ = 33 MHz			ϕ = 25 MHz			ϕ = 16 MHz			ϕ = 12.5 MHz		
UART	CSI	BPR	BRG0	Error	BPR	BRG0	Error	BPR	BRG0	Error	BPR	BRG0	Error
110	1760	–	–	–	4	222	0.02 %	4	142	0.03 %	3	222	0.02 %
150	2400	4	215	0.07 %	4	163	0.15 %	3	208	0.16 %	3	163	0.15 %
300	4800	3	215	0.07 %	3	163	0.15 %	2	208	0.16 %	2	163	0.15 %
600	9600	2	215	0.07 %	2	163	0.15 %	1	208	0.16 %	1	163	0.15 %
1200	19200	1	215	0.07 %	1	163	0.15 %	0	208	0.16 %	0	163	0.15 %
2400	38400	0	215	0.07 %	0	163	0.15 %	0	104	0.16 %	0	81	0.47 %
4800	76800	0	107	0.39 %	0	81	0.47 %	0	52	0.16 %	0	41	0.76 %
9600	153600	0	54	0.54 %	0	41	0.76 %	0	26	0.16 %	0	20	1.73 %
10400	166400	0	50	0.84 %	0	38	1.16 %	0	24	1.16 %	0	19	1.16 %
19200	307200	0	27	0.54 %	0	20	1.73 %	0	13	1.16 %	0	10	1.73 %
38400	614400	0	13	3.29 %	0	10	1.73 %	0	7	6.99 %Note	0	5	1.73 %
76800	1228800	0	7	4.09 %	0	5	1.73 %	–	–	–	0	3	15.2 %Note
153600	2457600	0	3	11.9 %Note	0	2	27.2 %Note	–	–	–	–	–	–

Baud Rate [bps]		ϕ = 20 MHz			ϕ = 14.746 MHz			ϕ = 12.288 MHz			ϕ = 9.830 MHz		
UART	CSI	BPR	BRG0	Error	BPR	BRG0	Error	BPR	BRG0	Error	BPR	BRG0	Error
110	1760	4	178	0.25 %	4	131	0.07 %	3	218	0.08 %	3	175	0.26 %
150	2400	4	130	0.16 %	3	192	0.0 %	3	160	0.0 %	3	128	0.0 %
300	4800	3	130	0.16 %	2	192	0.0 %	2	160	0.0 %	2	128	0.0 %
600	9600	2	130	0.16 %	1	192	0.0 %	1	160	0.0 %	1	128	0.0 %
1200	19200	1	130	0.16 %	0	192	0.0 %	0	160	0.0 %	0	128	0.0 %
2400	38400	0	130	0.16 %	0	96	0.0 %	0	80	0.0 %	0	64	0.0 %
4800	76800	0	65	0.16 %	0	48	0.0 %	0	40	0.0 %	0	32	0.0 %
9600	153600	0	33	1.36 %	0	24	0.0 %	0	20	0.0 %	0	16	0.0 %
10400	166400	0	30	0.16 %	0	22	0.7 %	0	18	2.6 %	0	15	1.5 %
19200	307200	0	16	1.73 %	0	12	0.0 %	0	10	0.0 %	0	8	0.0 %
38400	614400	0	8	1.73 %	0	6	0.0 %	0	5	0.0 %	0	4	0.0 %
76800	1228800	0	4	1.73 %	0	3	0.0 %	0	3	16.7 %Note	0	2	0.0 %
153600	2457600	0	2	1.73 %	0	2	25.0 %Note	–	–	–	0	1	0.0 %

Note: Cannot be used because the error is too great.

8.4.2 Baud rate generator register 0 (BRG0)

This is an 8-bit compare register that sets a timer/count value for the dedicated baud rate generator.

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

	7	6	5	4	3	2	1	0	Address	At reset
BRG0	BRG07	BRG06	BRG05	BRG04	BRG03	BRG02	BRG01	BRG00	FFFFF084H	Undefined

Caution: The internal timer (TMBRG0) is cleared by writing the BRG0 register. Therefore, do not rewrite or program the BRG0 register during transmission/reception operation.

8.4.3 Baud rate generator prescaler mode register 0 (BPRM0)

This register controls the timer/count operation of the dedicated baud rate generator and selects a count clock.

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

	7	6	5	4	3	2	1	0	Address	At reset
BPRM0	BRCE0	0	0	0	0	BPR02	BPR01	BPR00	FFFFF086H	00H

Bit Position	Bit Name	Function
7	BRCE0	Baud Rate Generator Count Enable Controls count operation of BRG. 0: Stops count operation with cleared 1: Enables count operation
2-0	BPR02-BPR00	Baud Rate Generator Prescaler Specifies count clock input to TMBRG.

BPR02	BPR01	BPR00	Count clock
0	0	0	ϕ/2 (n=0)
0	0	1	ϕ/4 (n=1)
0	1	0	ϕ/8 (n=2)
0	1	1	ϕ/16 (n=3)
1	×	×	ϕ/32 (n=4)

n: set value of prescaler, ϕ: system clock

Caution: Do not change the count clock during transmission/reception operation.

CHAPTER 9 PORT FUNCTION

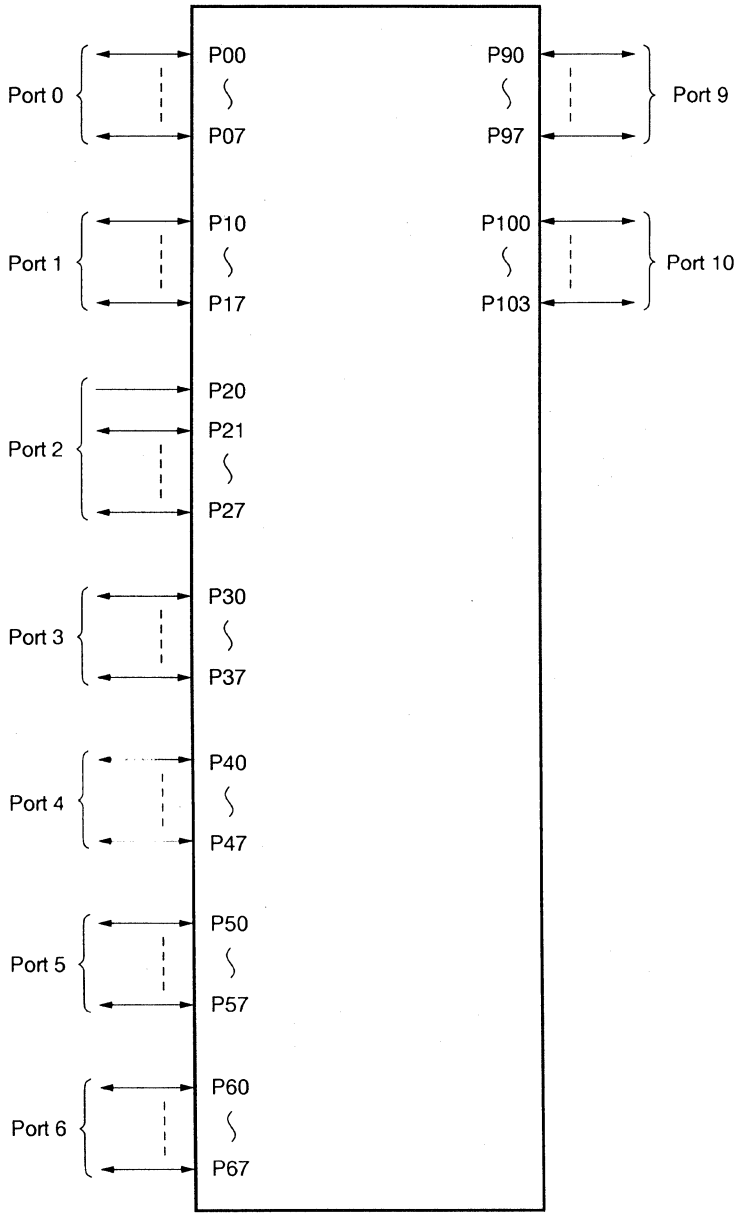
9.1 Features

The ports of the V851 have the following features:

- ○ Number of pins: input: 1
  I/O : 67
- ○ Multiplexed with I/O pins of other peripheral functions
- ○ Can be set in input/output mode in 1-bit units
- ○ Noise elimination
- ○ Edge detection

9.2 Basic Configuration of Port

The V851 is provided with a total of 68 input/output port pins (of which one is an input port pin) that make up ports 0 to 10. The configuration of the V851's ports is shown below.



Function of each port

The ports of the V851 have the functions shown in the table below.

Each port can be manipulated in 8- or 1-bit units and perform various types of control operations. In addition to port functions, the ports also have functions as internal hardware input/output pins, when placed in the control mode.

Port Name	Port Function	Function in Control Mode	Remarks
Port 0	8-bit I/O port (Can be set in input/output mode in 1-bit units)	Real-time pulse unit (RPU) input/output External interrupt request input	Can be set in port or control mode in 1-bit units
Port 1		–	Fixed to port mode
Port 2		External interrupt request input	Can be set in port or control mode in 1-bit units
Port 3		Serial interface (UART, CSI) input/output	
Port 4		Address/data bus (AD0-AD7) for external memory	Can be set in port or control mode in 8-bit units ★
Port 5		Address/data bus (AD8-AD15) for external memory	
Port 6		Address bus (A16-A23) for external memory	Can be set in port or control mode in 2-bit units
Port 9		Control signal output for external memory	Can be set in port or control mode in 5-, 2-, or 1-bit units
Port 10	4-bit I/O port (Can be set in input/output mode in 1-bit units)	Control signal input/output for system expansion	Can be set in port or control mode in 1-bit units

9.3 Port Pin Function

9.3.1 Port 0

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

	7	6	5	4	3	2	1	0		
P0	P07	P06	P05	P04	P03	P02	P01	P00	Address FFFFF000H	At reset Undefined

Bit Position	Bit Name	Function
7-0	P0n (n=7-0)	Port 0 I/O port

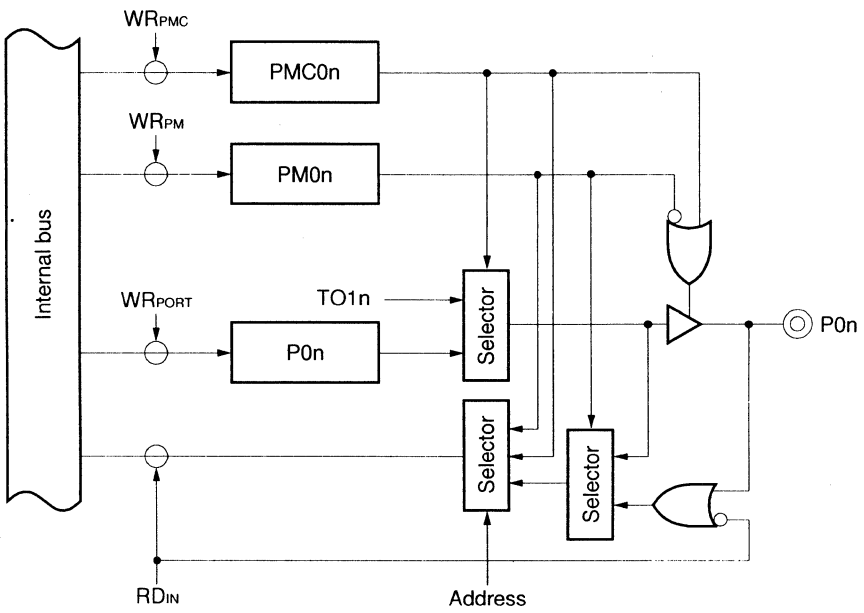
In addition to the function as a general I/O port, this port can also be used to input/output signals of the real-time pulse unit (RPU) and input external interrupt requests, when placed in the control mode.

Operation in control mode

Port		Control Mode	Remarks
Port 0	P00	TO10	Real-time pulse unit (RPU) output
	P01	TO11	
	P02	TCLR1	Real-time pulse unit (RPU) input
	P03	TI1	
	P04-P07	INTP10-INTP13	External interrupt input

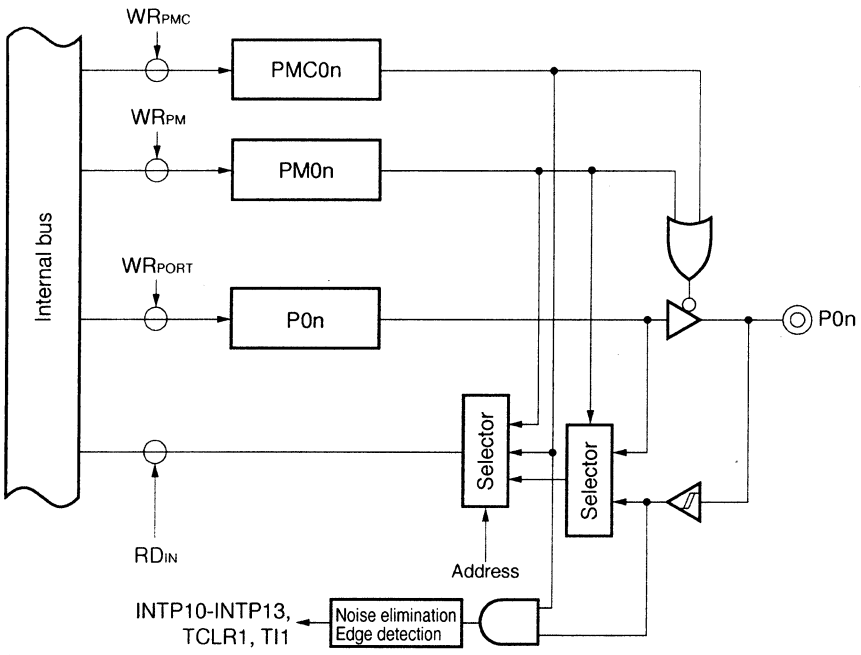
(1) Hardware configuration

Figure 9-1. Block Diagram of P00, P01 (Port 0)



Remark: n = 0, 1

Figure 9-2. Block Diagram of P02-P07 (Port 0)



Remark: n = 2-7

(2) Setting input/output mode and control mode

The input/output mode of port 0 is set by port mode register 0 (PM0). The control mode is set by port mode control register 0 (PMC0).

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	Address	At reset
PM0	PM07	PM06	PM05	PM04	PM03	PM02	PM01	PM00	FFFFF020H	FFH

Bit Position	Bit Name	Function
7-0	PM00-PM07	Port Mode Sets P00-P07 pins in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

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	Address	At reset
PMC0	PMC07	PMC06	PMC05	PMC04	PMC03	PMC02	PMC01	PMC00	FFFFF040H	00H

Bit Position	Bit Name	Function
7-4	PMC07-PMC04	Port Mode Control Indicates operation mode of P0n pin. 0: I/O port mode 1: External interrupt request input (INTP13-INTP10)
3	PMC03	Port Mode Control Indicates operation mode of P03 pin. 0: I/O port mode 1: TI1 input mode
2	PMC02	Port Mode Control Indicates operation mode of P02 pin. 0: I/O port mode 1: TCLR1 input mode
1	PMC01	Port Mode Control Indicates operation mode of P01 pin. 0: I/O port mode 1: TO11 output mode
0	PMC00	Port Mode Control Indicates operation mode of P00 pin. 0: I/O port mode 1: TO10 output mode

9.3.2 Port 1

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

	7	6	5	4	3	2	1	0		
P1	P17	P16	P15	P14	P13	P12	P11	P10	Address FFFFF002H	At reset Undefined

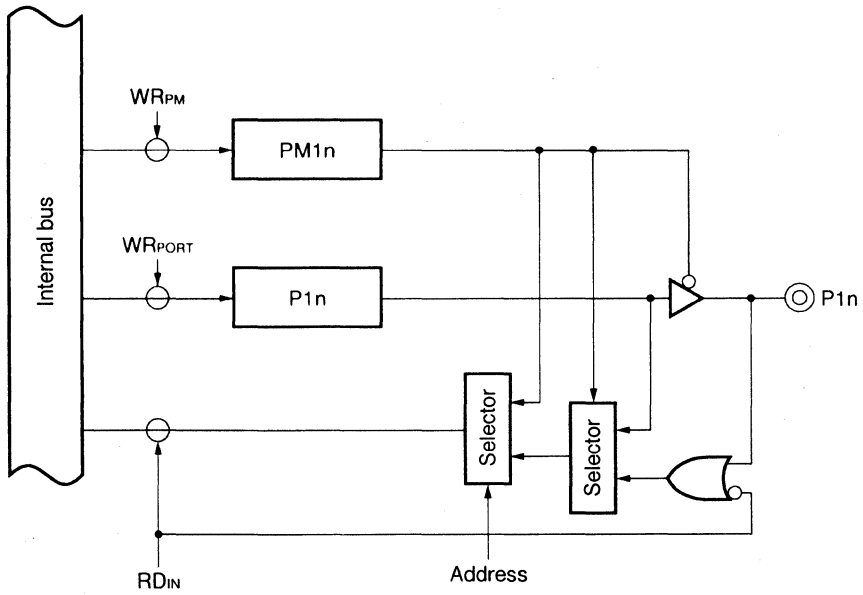
Bit Position	Bit Name	Function
7-0	P1n (n=7-0)	Port 1 I/O port

Port 1 is not multiplexed with other functions and is fixed in the port mode.

Port		Control Mode	Remarks
Port 1	P10-P17	–	Fixed in port mode

(1) Hardware configuration

Figure 9-3. Block Diagram of P10-P17 (Port 1)



Remark: n = 0-7

(2) Setting input/output mode

The input/output mode of port 1 is set by port mode register 1 (PM1).

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	Address	At reset
PM1	PM17	PM16	PM15	PM14	PM13	PM12	PM11	PM10	FFFFF022H	FFH

Bit Position	Bit Name	Function
7-0	PM1n (n=7-0)	Port Mode Sets P1n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

9.3.3 Port 2

Port 2 is an 8-bit input/output port that can be set in the input or output mode in 1-bit units.

	7	6	5	4	3	2	1	0	Address	At reset
P2	P27	P26	P25	P24	P23	P22	P21	P20	FFFFF004H	Undefined

Bit Position	Bit Name	Function
7-0	P2n (n=7-0)	Port 2 I/O port

In addition to the function as a port, this port can also be used to input external interrupt requests in the control mode.

P25-P27 are not multiplexed and are fixed in the control mode.

Operation in control mode

Port		Control Mode	Remarks
Port 2	P20	NMI	Non-maskable interrupt request input
	P21-24	INTP00-INTP03	External interrupt request input
	P25-P27	–	Fixed in port mode

(1) Hardware configuration

Figure 9-4. Block Diagram of P20 (Port 2)

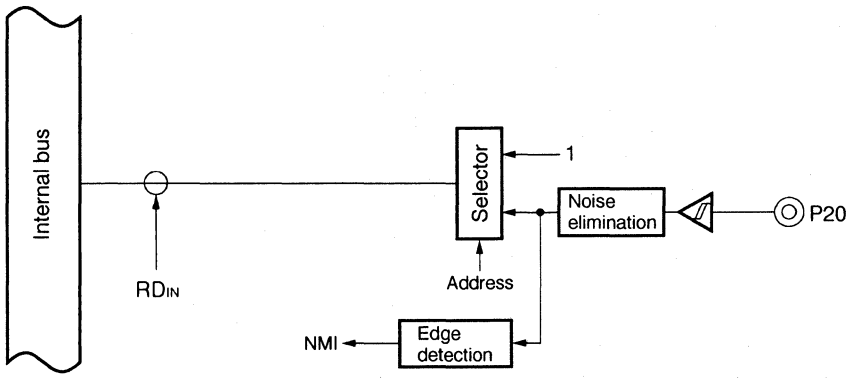
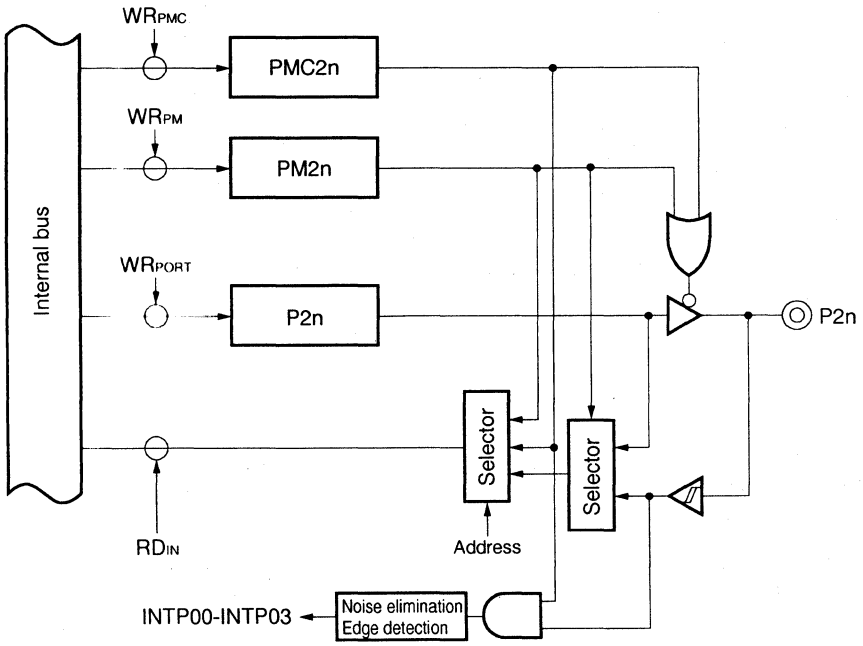


Figure 9-5. Block Diagram of P21-P24 (Port 2)



Remark: n = 1-4

Figure 9-6. Block Diagram of P25 (Port 2)

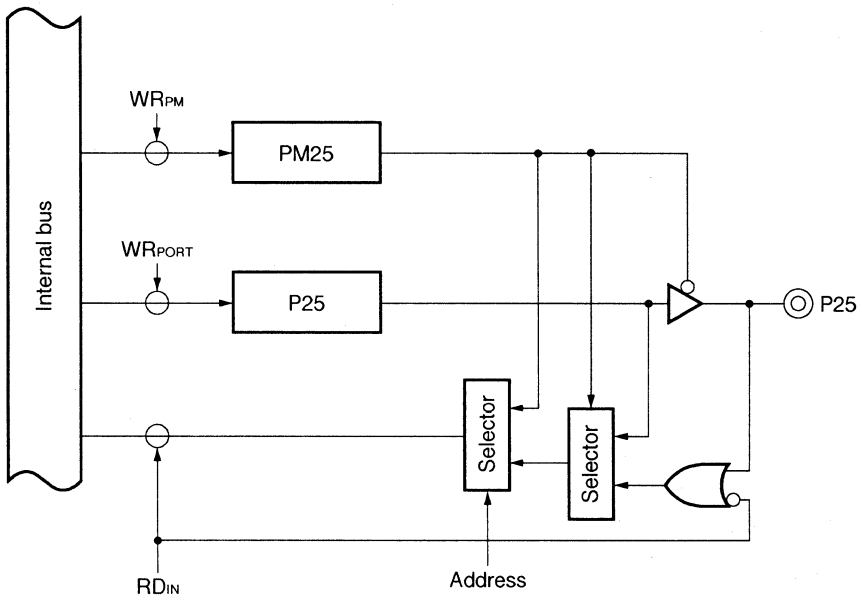
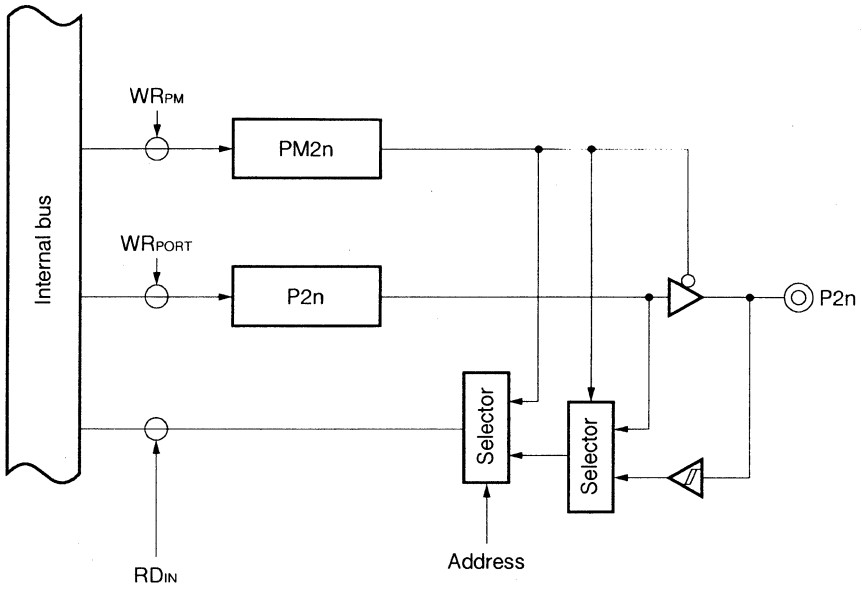


Figure 9-7. Block Diagram of P26, P27 (Port 2)



Remark: n = 6, 7

(2) Setting input/output mode and control mode

The input/output mode of port 2 is set by port mode register 2 (PM2). The control mode is set by port mode control register 2 (PMC2).

P20 is fixed in the NMI input mode.

Port 2 mode register (PM2)

This register can be read/written in 8- or 1-bit units. However, bit 0 is fixed to "1" by hardware. Even if "0" is written to this bit, it is ignored.

	7	6	5	4	3	2	1	0	Address	At reset
PM2	PM27	PM26	PM25	PM24	PM23	PM22	PM21	1	FFFFF024H	FFH

Bit Position	Bit Name	Function
7-1	PM2n (n=7-1)	Port Mode Sets P2n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

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. If "0" is written to this bit, it is ignored.

	7	6	5	4	3	2	1	0	Address	At reset
PMC2	0	0	0	PMC24	PMC23	PMC22	PMC21	1	FFFFF044H	01H

Bit Position	Bit Name	Function
4	PMC24	Port Mode Control Sets operation mode of P24 pin. 0: I/O port mode 1: INTP03 input mode
3	PMC23	Port Mode Control Sets operation mode of P23 pin. 0: I/O port mode 1: INTP02 input mode
2	PMC22	Port Mode Control Sets operation mode of P22 pin. 0: I/O port mode 1: INTP01 input mode
1	PMC21	Port Mode Control Sets operation mode of P21 pin. 0: I/O port mode 1: INTP00 input mode

9.3.4 Port 3

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

	7	6	5	4	3	2	1	0	Address	At reset
P3	P37	P36	P35	P34	P33	P32	P31	P30	FFFFF006H	Undefined

Bit Position	Bit Name	Function
7-0	P3n (n=7-0)	Port 3 I/O port

In addition to the function as a port, this port can also be used as the input/output lines of the serial interface (UART, CSI), when placed in the control mode.

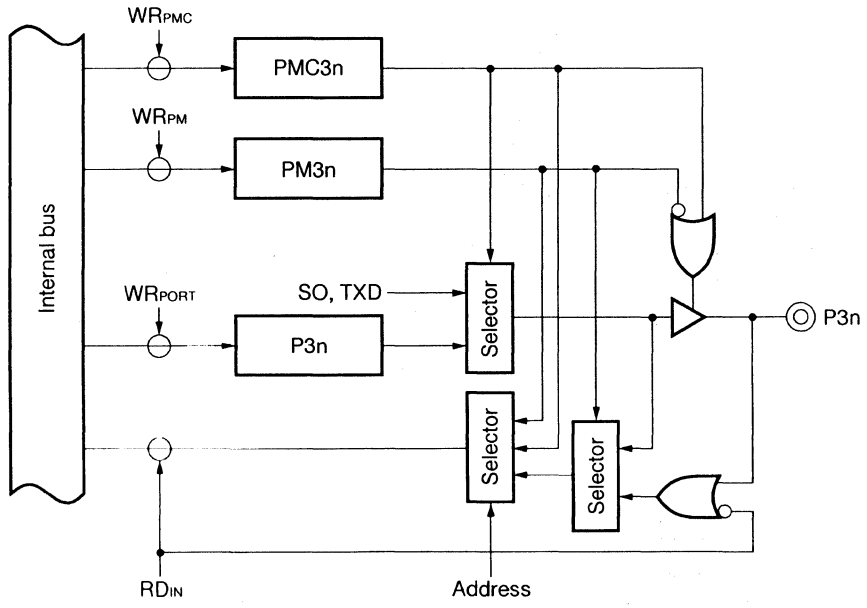
P35-P37 are not multiplexed and fixed in the port mode.

Operation in control mode

Port		Control Mode	Remarks
Port 3	P30	SO	I/O for serial interface (UART, CSI)
	P31	SI	
	P32	$\overline{\text{SCK}}$	
	P33	TXD	
	P34	RXD	
	P35-P37	–	Fixed in port mode

(1) Hardware configuration

Figure 9-8. Block Diagram of P30, P33 (Port 3)



Remark: n = 0, 3

Figure 9-9. Block Diagram of P31 (Port 3)

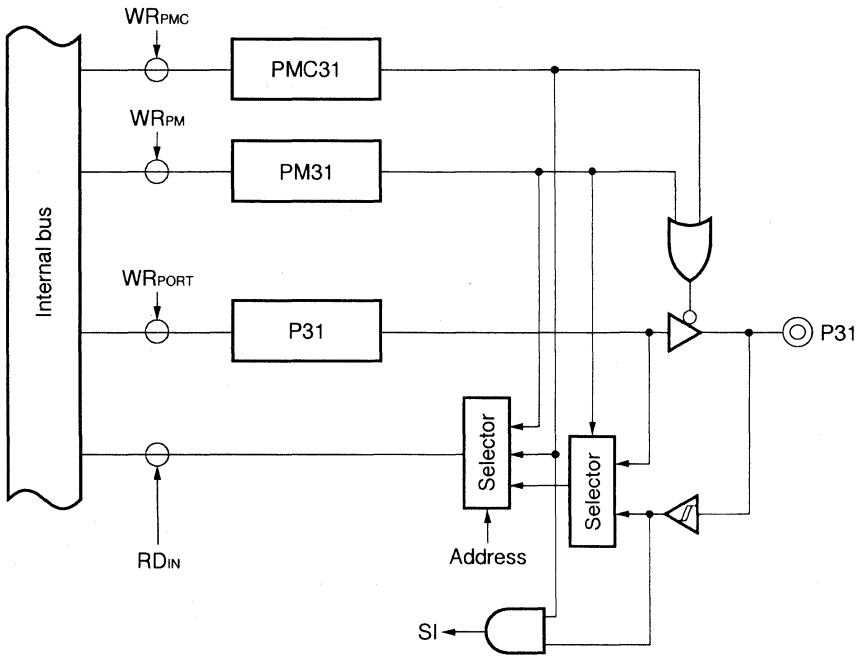


Figure 9-10. Block Diagram of P32 (Port 3)

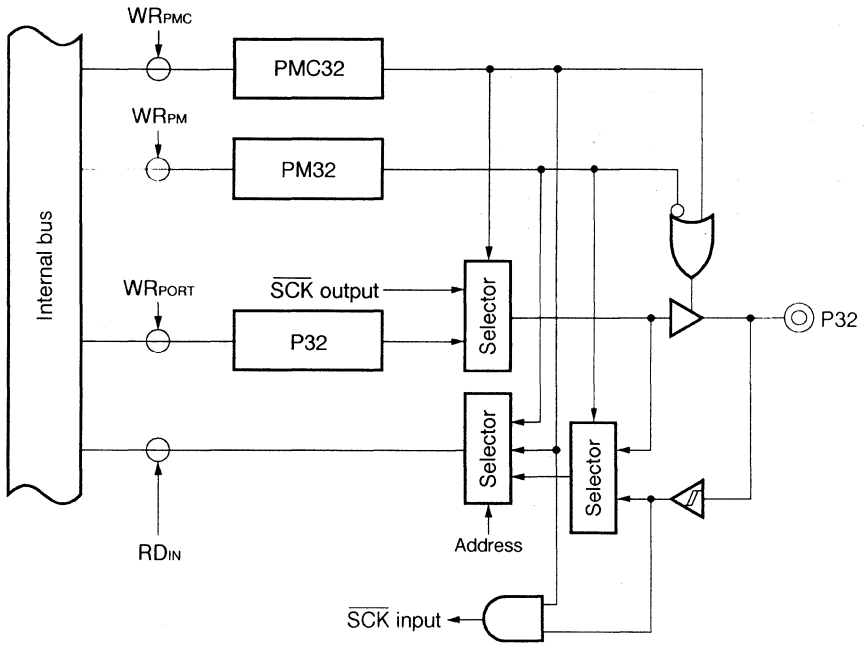


Figure 9-11. Block Diagram of P34 (Port 3)

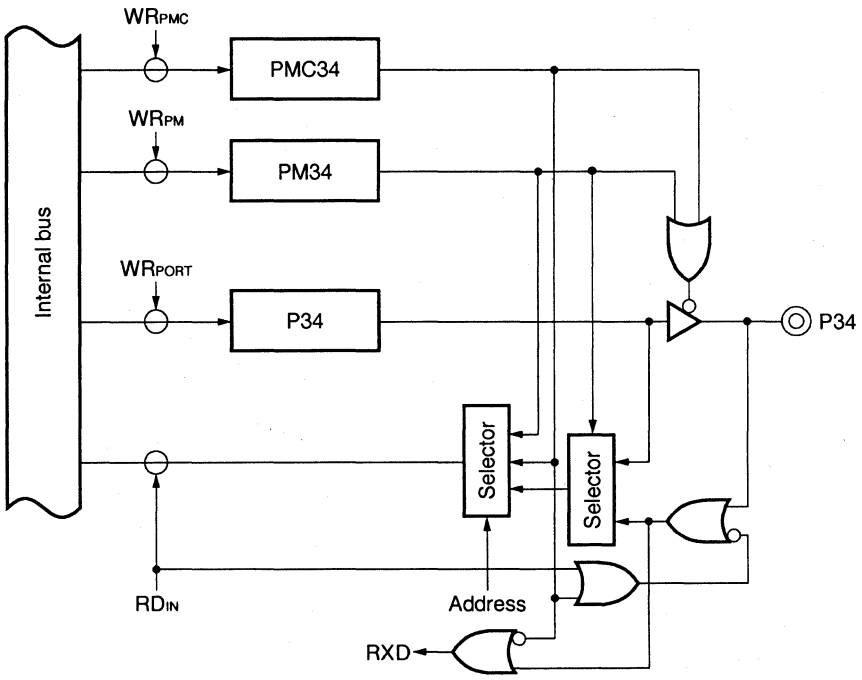


Figure 9-12. Block Diagram of P35 (Port 3)

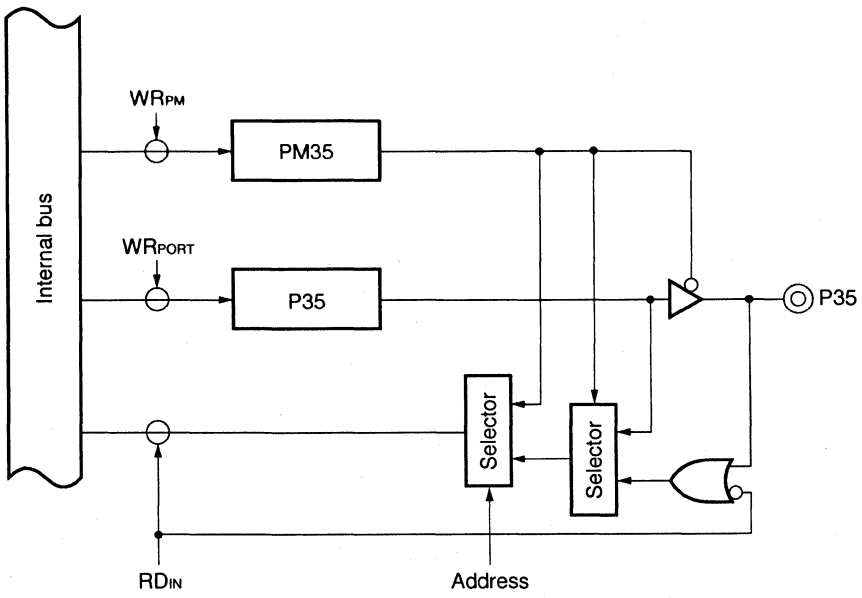
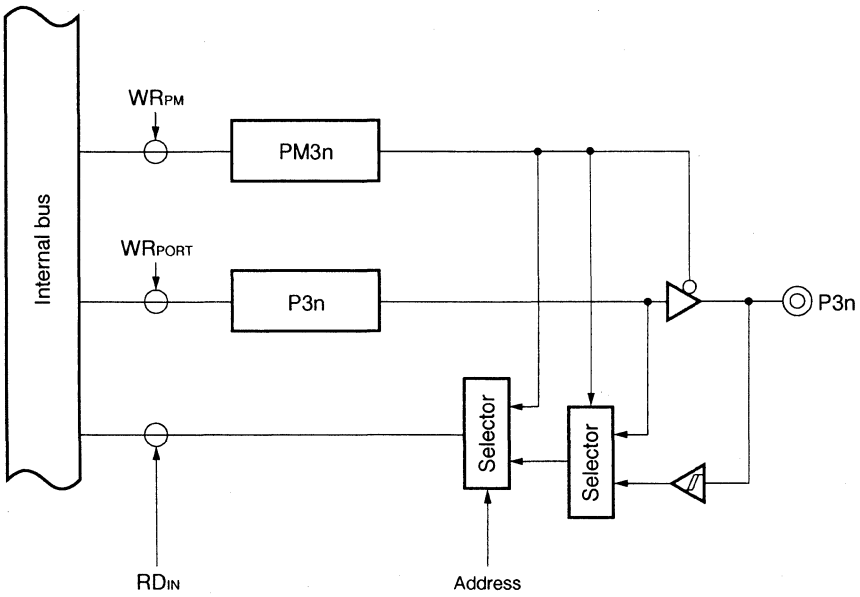


Figure 9-13. Block Diagram of P36, P37 (Port 3)



Remark: n = 6, 7

(2) Setting input/output mode and control mode

The input/output mode of port 3 is set by port mode register 3 (PM3). The control mode is set by port mode control register 3 (PMC3).

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	Address	At reset
PM3	PM37	PM36	PM35	PM34	PM33	PM32	PM31	PM30	FFFFF026H	FFH

Bit Position	Bit Name	Function
7-0	PM3n (n=7-0)	Port Mode Sets P3n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

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	At reset
PMC3	0	0	0	PMC34	PMC33	PMC32	PMC31	PMC30	FFFFF046H	00H

Bit Position	Bit Name	Function
4	PMC34	Port Mode Control Sets operation mode of P34 pin. 0: I/O port mode 1: RXD input mode
3	PMC33	Port Mode Control Sets operation mode of P33 pin. 0: I/O port mode 1: TXD output mode
2	PMC32	Port Mode Control Sets operation mode of P32 pin. 0: I/O port mode 1: $\overline{\text{SCK}}$ input/output mode
1	PMC31	Port Mode Control Sets operation mode of P31 pin. 0: I/O port mode 1: SI input mode
0	PMC30	Port Mode Control Sets operation mode of P30 pin. 0: I/O port mode 1: SO output mode

9.3.5 Port 4

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

	7	6	5	4	3	2	1	0		
P4	P47	P46	P45	P44	P43	P42	P41	P40	Address FFFFF008H	At reset Undefined

Bit Position	Bit Name	Function
7-0	P4n (n=7-0)	Port 4 I/O port

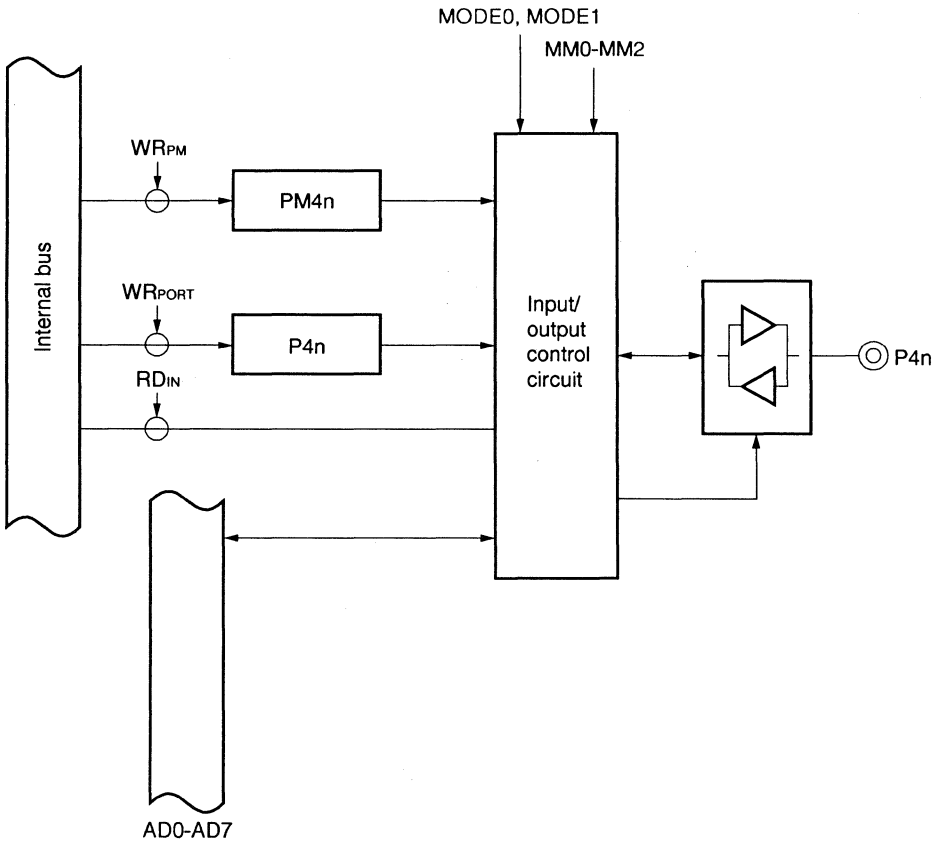
In addition to the function as a general I/O port, this port also serves as an external address/data bus, when placed in the control mode.

Operation in control mode

Port		Control Mode	Remarks
Port 4	P40-47	AD0-AD7	Address/data bus for external memory

(1) Hardware configuration

Figure 9-14. Block Diagram of P40-P47 (Port 4)



Remark: n = 0-7

(2) Setting input/output mode and control mode

The input/output mode of port 4 is set by port mode register 4 (PM4). To enable the external address/data bus function, the control mode (external expansion mode) is set by mode specification pins MODE0 and MODE1, and memory expansion mode register (MM: refer to **3.4.6 (1)**).

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	Address	At reset
PM4	PM47	PM46	PM45	PM44	PM43	PM42	PM41	PM40	FFFFF028H	FFH

Bit Position	Bit Name	Function
7-0	PM4n (n=7-0)	Port Mode Sets P4n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

Operation mode of port 4

Bit of MM Register			Operation Mode							
MM2	MM1	MM0	P40	P41	P42	P43	P44	P45	P46	P47
0	0	0	Port							
0	1	1	Address/data bus (AD0-AD7)							
1	0	0								
1	0	1								
1	1	0								
1	1	1								
Others			RFU (reserved)							

For the details of mode selection by the MODE0 and MODE1 pins, refer to **3.3.2 Specifying operation mode.**

When MODE0 and MODE1 = 00 (ROM-less mode), MM0-MM2 bits are initialized to 111 at system reset, enabling the external expansion mode. External expansion can be disabled by programming the MM0-MM2 bits and setting the port mode. If MM0-MM2 are cleared to 000, the subsequent external instruction cannot be fetched.

9.3.6 Port 5

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

	7	6	5	4	3	2	1	0		
P5	PM57	P56	P55	P54	P53	P52	P51	P50	Address FFFFF00AH	At reset Undefined

Bit Position	Bit Name	Function
7-0	P5n (n=7-0)	Port 5 I/O port

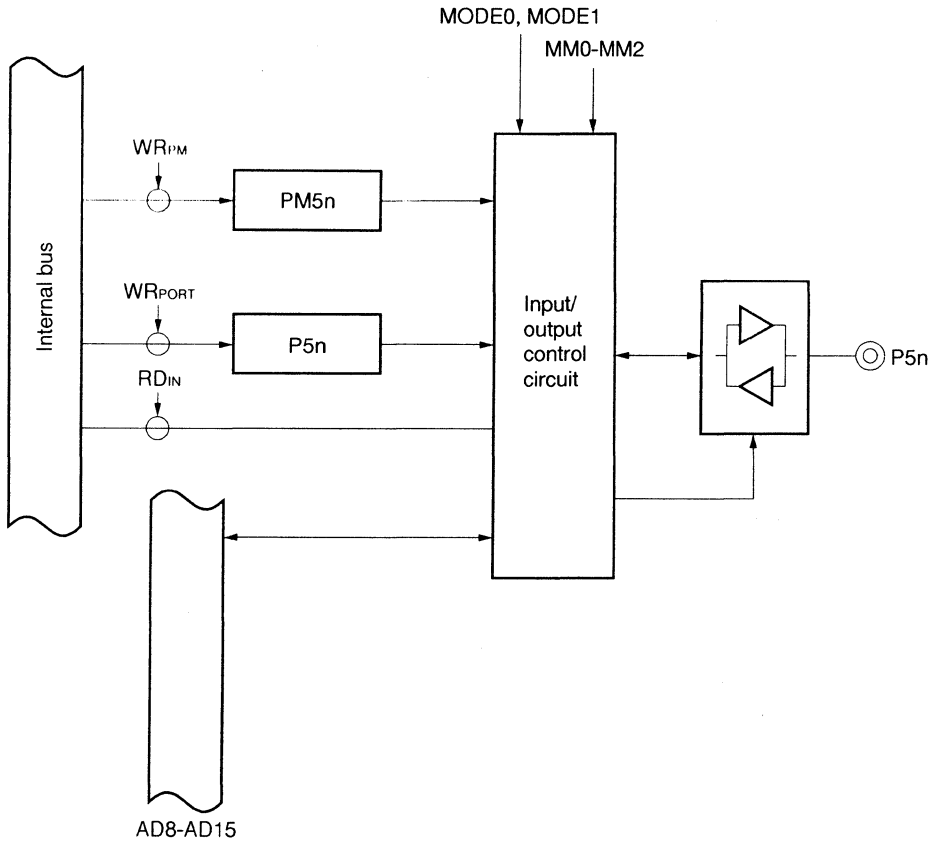
In addition to the function as a general I/O port, this port also serves as an external address/data bus, when placed in the control mode.

Operation in control mode

Port		Control Mode	Remarks
Port 5	P50-57	AD8-AD15	Address/data bus for external memory

(1) Hardware configuration

Figure 9-15. Block Diagram of P50-P57 (Port 5)



Remark: n = 0-7

(2) Setting input/output mode and control mode

The input/output mode of port 5 is set by port mode register 5 (PM5). To enable the external address/data bus function, the control mode (external expansion mode) is set by mode specification pins MODE0 and MODE1, and memory expansion mode register (MM: refer to **3.4.6 (1)**).

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	Address	At reset
PM5	PM57	PM56	PM55	PM54	PM53	PM52	PM51	PM50	FFFFF02AH	FFH

Bit Position	Bit Name	Function
7-0	PM5n (n=7-0)	Port Mode Sets P5n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

Operation mode of port 5

Bit of MM Register			Operation Mode							
MM2	MM1	MM0	P50	P51	P52	P53	P54	P55	P56	P57
0	0	0	Port							
0	1	1	Address/data bus (AD8-AD15)							
1	0	0								
1	0	1								
1	1	0								
1	1	1								
Others			RFU (reserved)							

9.3.7 Port 6

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

	7	6	5	4	3	2	1	0		
P6	P67	P66	P65	P64	P63	P62	P61	P60	Address FFFFF00CH	At reset Undefined

Bit Position	Bit Name	Function
7-0	P6n (n=7-0)	Port 6 I/O port

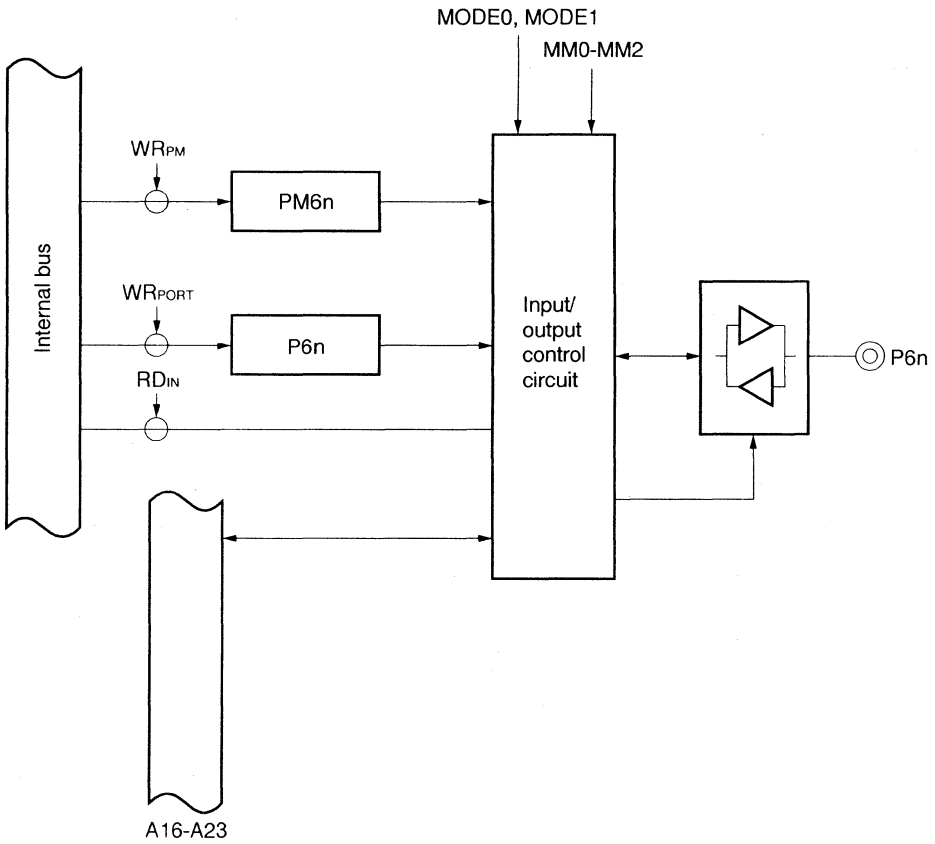
In addition to the function as a general I/O port, this port also serves as an external address bus, when placed in the control mode.

Operation in control mode

Port		Control Mode	Remarks
Port 6	P60-67	A16-A23	Address bus for external memory

(1) Hardware configuration

Figure 9-16. Block Diagram of P60-67 (Port 6)



Remark: n = 0-7

(2) Setting input/output mode and control mode

The input/output mode of port 6 is set by port mode register 6 (PM6). To enable the external address/data bus function, the control mode (external expansion mode) is set by mode specification pins MODE0 and MODE1, and memory expansion mode register (MM: refer to **3.4.6 (1)**).

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		
PM6	PM67	PM66	PM65	PM64	PM63	PM62	PM61	PM60	Address FFFFF02CH	At reset FFH

Bit Position	Bit Name	Function
7-0	PM6n (n=7-0)	Port Mode Sets P6n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

Operation mode of port 6

Bit of MM Register			Operation Mode							
MM2	MM1	MM0	P60	P61	P62	P63	P64	P65	P66	P67
0	0	0	Port							
0	1	1								
1	0	0	A16	A17	A18	A19	A20	A21	A22	A23
1	0	1								
1	1	0								
1	1	1								
Others			RFU (reserved)							

9.3.8 Port 9

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

	7	6	5	4	3	2	1	0		
P9	P97	P96	P95	P94	P93	P92	P91	P90	Address FFFFF012H	At reset Undefined

Bit Position	Bit Name	Function
7-0	P9n (n=7-0)	Port 9 I/O port

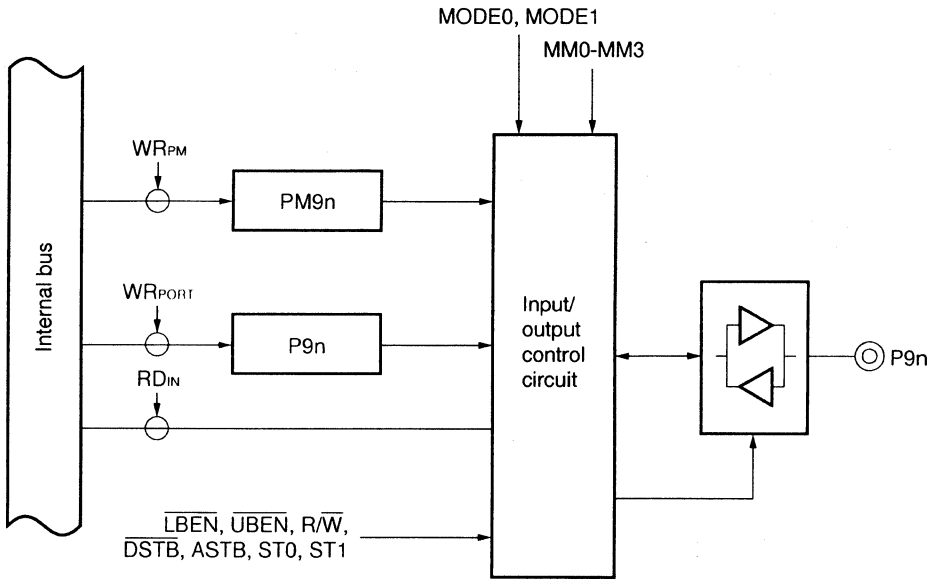
In addition to the function as a general I/O port, this port can also be used to output external bus control signals, when placed in the control mode.

Operation in control mode

Port		Control Mode	Remarks
Port 9	P90	$\overline{\text{LBEN}}$	Control signal output for external memory
	P91	$\overline{\text{UBEN}}$	
	P92	R/$\overline{\text{W}}$	
	P93	$\overline{\text{DSTB}}$	
	P94	ASTB	
	P95	ST0	
	P96	ST1	
	P97	–	Fixed in port mode

(1) Hardware configuration

Figure 9-17. Block Diagram of P90-P97 (Port 9)



Remark: n = 0-7

(2) Setting input/output mode and control mode

The input/output mode of port 9 is set by port mode register 9 (PM9). To enable the external bus control signals, the control mode (external expansion mode) is set by mode specification pins MODE0 and MODE1, and memory expansion mode register (MM: refer to **3.4.6 (1)**).

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		
PM9	PM97	PM96	PM95	PM94	PM93	PM92	PM91	PM90	Address FFFFF032H	At reset FFH

Bit Position	Bit Name	Function
7-0	PM9n (n=7-0)	Port Mode Sets P9n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

Operation mode of port 9

P90-P94

Bit of MM Register			Operation Mode				
MM2	MM1	MM0	P90	P91	P92	P93	P94
0	0	0	Port				
0	1	1	$\overline{\text{LBEN}}$	$\overline{\text{UBEN}}$	R/$\overline{\text{W}}$	$\overline{\text{DSTB}}$	ASTB
1	0	0					
1	0	1					
1	1	0					
1	1	1					
Others			RFU (reserved)				

P95, P96

MM3	Operation Mode	P95	P96
0	Port mode	Port	
1	External expansion mode	ST0	ST1

9.3.9 Port 10

Port 10 is a 4-bit input/output port that can be set in the input or output mode in 1-bit units.

	7	6	5	4	3	2	1	0		
P10	–	–	–	–	P103	P102	P101	P100	Address FFFFF014H	At reset Undefined

Bit Position	Bit Name	Function
3-0	P10n (n=3-0)	Port 10 I/O port

When port 10 is accessed in 8-bit units for write, the higher 4 bits are ignored. When it is accessed in 8-bit units for read, undefined data is read.

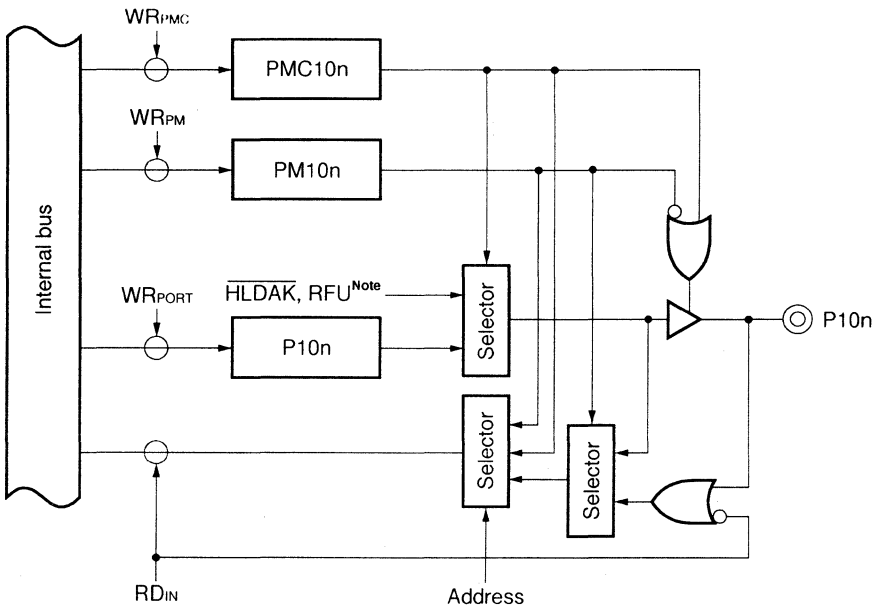
In addition to the function as a port, this port can also be used to input and output external contorl signals to a bus master or ASIC device, when placed in the control mode.

Operation in control mode

Port		Control Mode	Remarks
Port 10	P100	$\overline{\text{HLDAK}}$	Bus hold control signal input/output
	P101	$\overline{\text{HLDRQ}}$	
	P102, P103	–	Fixed in port mode

(1) Hardware configuration

Figure 9-18. Block Diagram of P100, P103 (Port 10)



Note: RFU is an undefined value.

Remark: n = 0, 3

Figure 9-19. Block Diagram of P101 (Port 10)

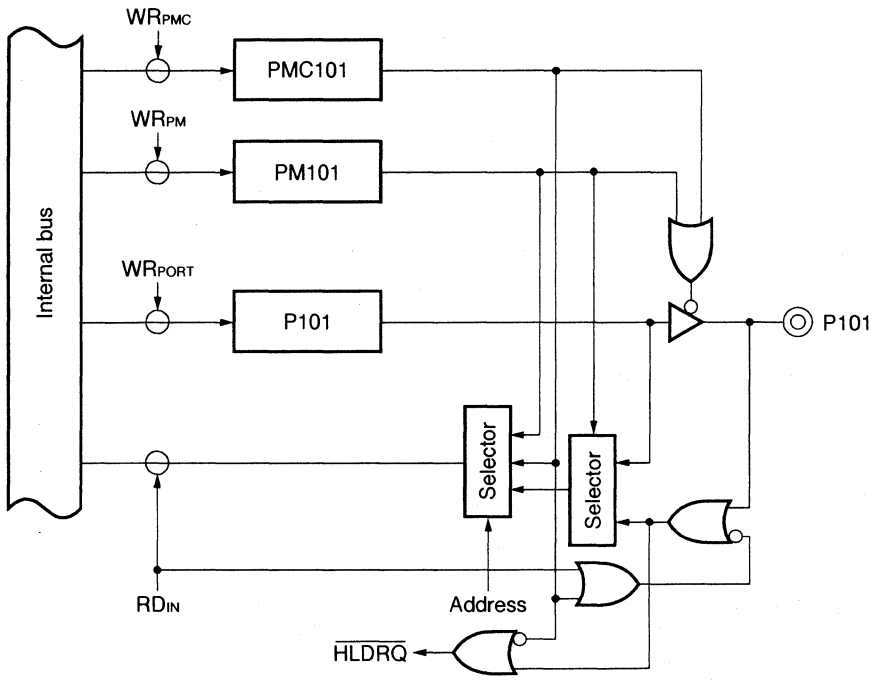
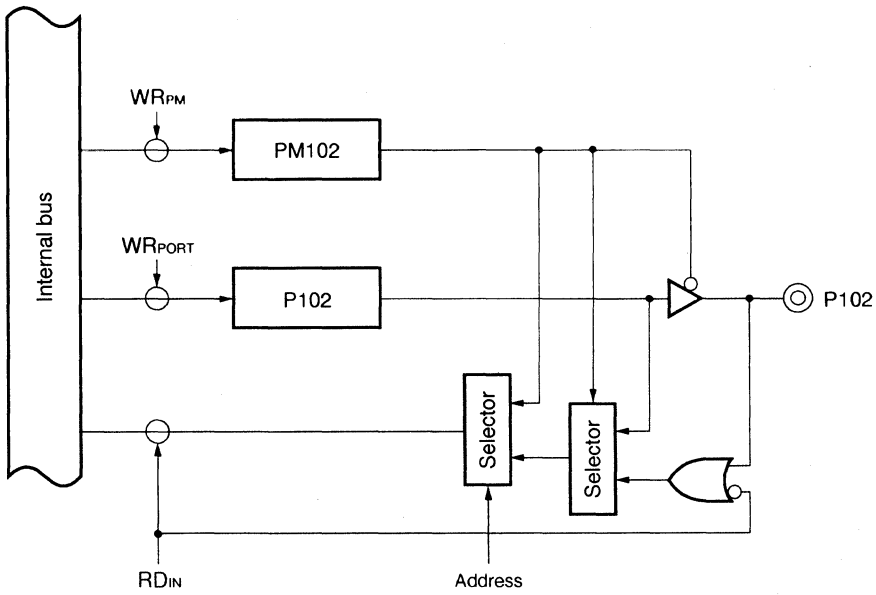


Figure 9-20. Block Diagram of P102 (Port 10)



(2) Setting input/output mode and control mode

The input/output mode of port 10 is set by port mode register 10 (PM10). The control mode is set by port mode control register 10 (PMC10).

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	Address	At reset
PM10	1	1	1	1	PM103	PM102	PM101	PM100	FFFFF034H	FFH

Bit Position	Bit Name	Function
3-0	PM10n (n=3-0)	Port Mode Sets P10n pin in input/output mode. 0: Output mode (output buffer ON) 1: Input mode (output buffer OFF)

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	At reset
PMC10	0	0	0	0	0	0	PMC101	PMC100	FFFFF054H	00H

Bit Position	Bit Name	Function
1	PMC101	Port Mode Control Sets operation mode of P101 pin. 0: I/O port mode 1: $\overline{\text{HLDRQ}}$ input mode
0	PMC100	Port Mode Control Sets operation mode of P100 pin. 0: I/O port mode 1: $\overline{\text{HLDAK}}$ output mode

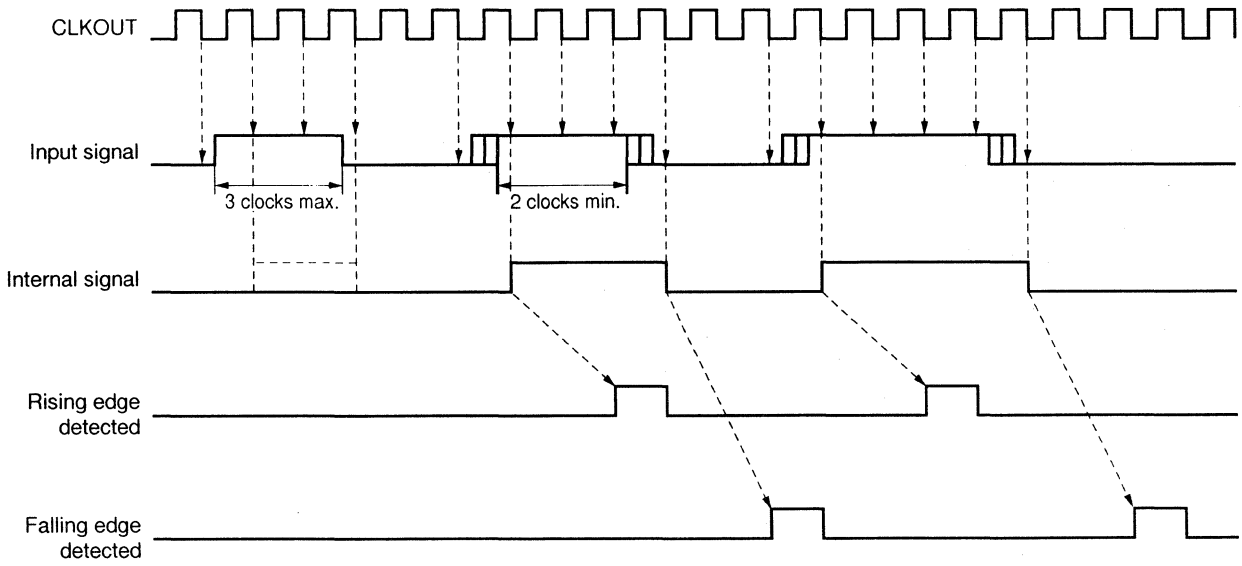
9.4 Input Noise Filters

Noise filters are provided to certain port pins when operating in the control mode. Spikes and voltage transitions which occur within the defined noise filtering times are ignored. One analog filter is used on the NMI input, while the rest are digital filters which operate via a timing control circuit. Filters are provided on the pins listed below.

Pin	Noise Filtering Time
★ P20/NMI[Note]	Analog delay (60 ns to 220 ns)
P02/TCLR1	2 to 3 system clocks
P03/TI1	
P04/INTP10	
P05/INTP11	
P06/INTP12	
P07/INTP13	
P21/INTP00	
P22/INTP01	
P23/INTP02	
P24/INTP03	

Note: The P20/NMI pin is used to release the STOP mode. In the STOP mode, the clock control timing circuit is not used because the clock is stopped.

Figure 9-21. Example of Noise Filtering Timing



CHAPTER 10 RESET FUNCTION

A valid low-level signal on the $\overline{\text{RESET}}$ pin initiates a system reset.

Program execution begins at the $\overline{\text{RESET}}$ vector address when the reset condition is removed and a high-level signal appears at the $\overline{\text{RESET}}$ pin.

10.1 Features

○ Analog noise filtering circuit (delay of 60 ns-220 ns) provided on reset pin ★

10.2 Pin Function

During the reset state, all the pins (except CLKOUT, $\overline{\text{RESET}}$, X2, V_{DD}, V_{SS}, CV_{DD}, and CV_{SS} pins are in the high-impedance state.

When an external memory is connected, a pull-up (or pull-down) resistor must be connected to each pin of ports 4, 5, 6, and 9. Otherwise, the memory contents may be lost if these pins go into a high-impedance state.

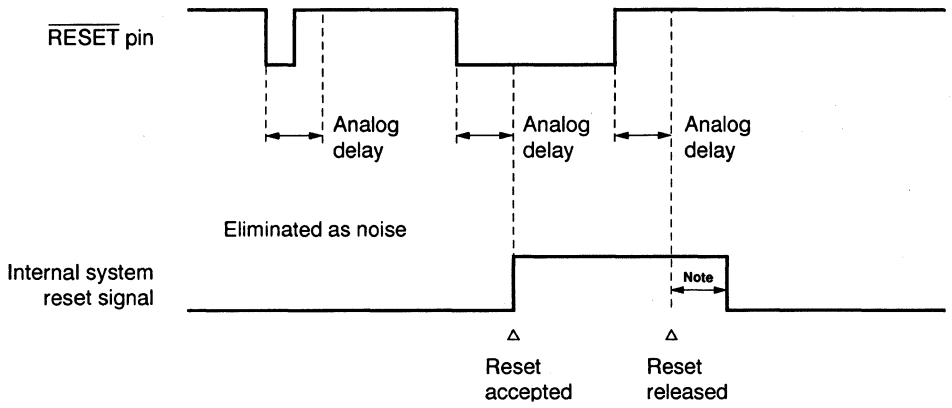
The internal system clock continues to generate the clock signal at CLKOUT pin while the device is in the reset state.

Table 10-1 shows the operating status of each pin during the reset period.

Table 10-1. Operating Status of Each Pin During Reset Period

Pin	Operating Status
AD0-AD15	Hi-Z
A16-A23	
$\overline{\text{LBEN}}$, $\overline{\text{UBEN}}$	
R/$\overline{\text{W}}$	
$\overline{\text{DSTB}}$	
ASTB	
ST0, ST1	
$\overline{\text{HLDRQ}}$	–
$\overline{\text{HLDAK}}$	Hi-Z
$\overline{\text{WAIT}}$	–
CLKOUT	Clock output

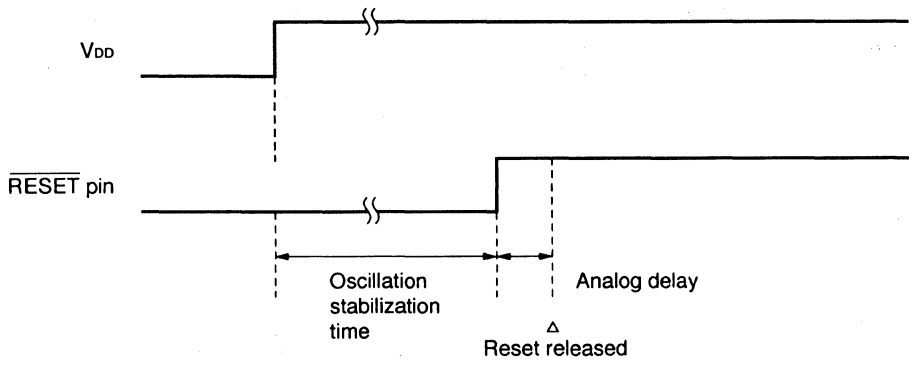
(1) Accepting reset signal



Note: The internal system reset signal remains active for the duration of at least 4 system clocks after the reset condition is removed from the RESET pin.

(2) Power-ON reset

On power-up, the RESET pin must be held low for at least 10 ms, until the power supply reaches its operating voltage and the clock has stabilized. The clock stabilization time consists of the time for the oscillator to stabilize (oscillation stabilization time) and the time for the PLL to lock at a specific frequency.



10.3 Initialize

Table 10-2 shows the initial value of each register after reset.

The contents of the registers must be initialized in the program as necessary. Especially, set the following registers as necessary because they are related to system setting:

- Power save control register (PSC) ... X1 and X2 pin function, CLKOUT pin operation, etc.
- Data wait control register (DWC) ... Number of data wait states

Caution: In Table 10-2, "Undefined" means an undefined value due to power-on reset or data corruption when a falling edge of RESET coincides with a data write operation. The previous status of data is retained by a falling edge of RESET due to the cases other than the above.

Table 10-2. Initial Values of Each Register at Reset

Register		Initial Value at Reset
r0		00000000H
r1-r31		Undefined
PC		00000000H
PSW		00000020H
EIPC		Undefined
EIPSW		Undefined
FEPC		Undefined
FEPSW		Undefined
ECR		00000000H
Internal RAM		Undefined
Port	Output latch (P0-P6, P9, P10)	Undefined
	Mode register (PM0-6, PM9, PM10)	FFH
	Mode control register (PMC0, PMC3, PMC10)	00H
	(PMC2)	01H
	Memory expansion mode register (MM)	10H or 17H
Clock generator	System status register (SYS)	0000000xB
Real-time pulse unit	Timer unit mode register (TUM1)	0000H
	Timer control register (TMC1, TMC4)	00H
	Timer output control register 1 (TOC1)	00H
	Timer (TM1, TM4)	0000H
	Capture/compare register (CC10-CC13)	Undefined
	Compare register 4 (CM4)	Undefined
	Timer overflow status register (TOVS)	00H
Serial interface	Asynchronous serial interface mode register 00 (ASIM00)	80H
	Asynchronous serial interface mode register 01 (ASIM01)	00H
	Asynchronous serial interface status register 0 (ASIS0)	00H
	Receive buffer (RXB0, RXB0L)	Undefined
	Transmit shift register (TXS0, TXS0L)	Undefined
	Clocked serial interface mode register 0 (CSIM0)	00H
	Serial I/O shift register 0 (SIO0)	Undefined
	Baud rate generator register 0 (BRG0)	Undefined
	Baud rate generator prescaler mode register 0 (BPRM0)	00H
Interrupt/exception processing function	Interrupt control register (xxCn)	47H
	In-service priority register (ISPR)	00H
	External interrupt mode register (INTM0, INTM1, INTM2)	00H
Memory management function	Data wait control register (DWC)	FFFFH
	Bus cycle control register (BCC)	AAAAH
Power save control	Command register (PRCMD)	Undefined
	Power save control register (PSC)	00H

Remark: "x" means don't care bit.

[MEMO]

CHAPTER 11 PROM MODE

The PROM model of the V851 has an internal 32K-byte one-time PROM. The internal ROM can be accessed in 1 clock, like the mask ROM model, to fetch instructions.

11.1 PROM Mode

The PROM mode is entered by the setting MODE0 and MODE1 pins.
Connect the pins not used in this mode as described in section 1.5.2 "PROM programming mode".

V_{PP}	MODE1	MODE0	Operation Mode
5.0 V	1	1	PROM mode (read mode)
12.5 V	1	1	PROM mode (programming mode)

V_{PP}: programming voltage

11.2 Operation Mode

Operation in the PROM programming mode is determined by the setting of the pins shown in the following table.

Operation Mode \ Pin		P25/$\overline{CE}$	P26/$\overline{OE}$	P27/$\overline{PGM}$	V_{PP}	V_{DD}	P47/D7-P40/D0
Read Mode	Read	L	L	H	+5.0 V	+5.0 V	Data output
	Output disable	L	H	×			Hi-Z[Note]
	Standby	H	×	×			
Programming Mode	Page data latch	H	L	H	+12.5 V	+6.5 V	Data input
	Page program	H	H	L			Hi-Z
	Byte program	L	H	L			Data input
	Program verify	L	L	H			Data output
	Program inhibit	×	L	L			Hi-Z[Note]
			H	H			

V_{PP}: programming voltage (12.5 V)
× : don't care

Note: In this case, the address input is invalid, and 1/0 can be input.

(1) Read mode

The read mode is set when $\overline{CE}$ = L and $\overline{OE}$ = L.

(2) Output disable mode

The data output goes into a high-impedance state when $\overline{OE}$ = H, and the output disable mode is set. If two or more μPD70P3000s are connected to the data bus, any one of the devices can be read by controlling the $\overline{OE}$ pin.

(3) Standby mode

The standby mode is set when $\overline{CE}$ = H.

In this mode, the data output goes into a high-impedance state regardless of the status of $\overline{OE}$.

(4) Page data latch mode

The page data latch mode is set when $\overline{CE}$ = H, $\overline{OE}$ = L, and $\overline{PGM}$ = H at the beginning of the page write mode.

In this mode, data of 1 page and 4 bytes is latched to the internal address/data latch circuit.

(5) Page write mode

Page write is executed in the page write mode by applying a 0.1-ms program pulse (active low) to the $\overline{PGM}$ pin with $\overline{CE}$ = H and $\overline{OE}$ = H, after an address and data of 1 page and 4 bytes have been latched. After that, the program can be verified when $\overline{CE}$ = L and $\overline{OE}$ = L.

If the program cannot be written by one program pulse, write and verify are repeatedly executed X times (X ≤ 10).

(6) Byte write mode

Byte write is executed by applying a 0.1-ms program pulse (active low) to the $\overline{PGM}$ pin with $\overline{CE}$ = L and $\overline{OE}$ = H. After that, the program can be verified when $\overline{OE}$ = L.

If the program cannot be written by one program pulse, write and verify are repeatedly executed X times (X ≤ 10).

(7) Program verify mode

The program verify mode is set by setting $\overline{CE}$ = L, $\overline{PGM}$ = H, and $\overline{OE}$ = L. Use this mode to the check if the program has been correctly written.

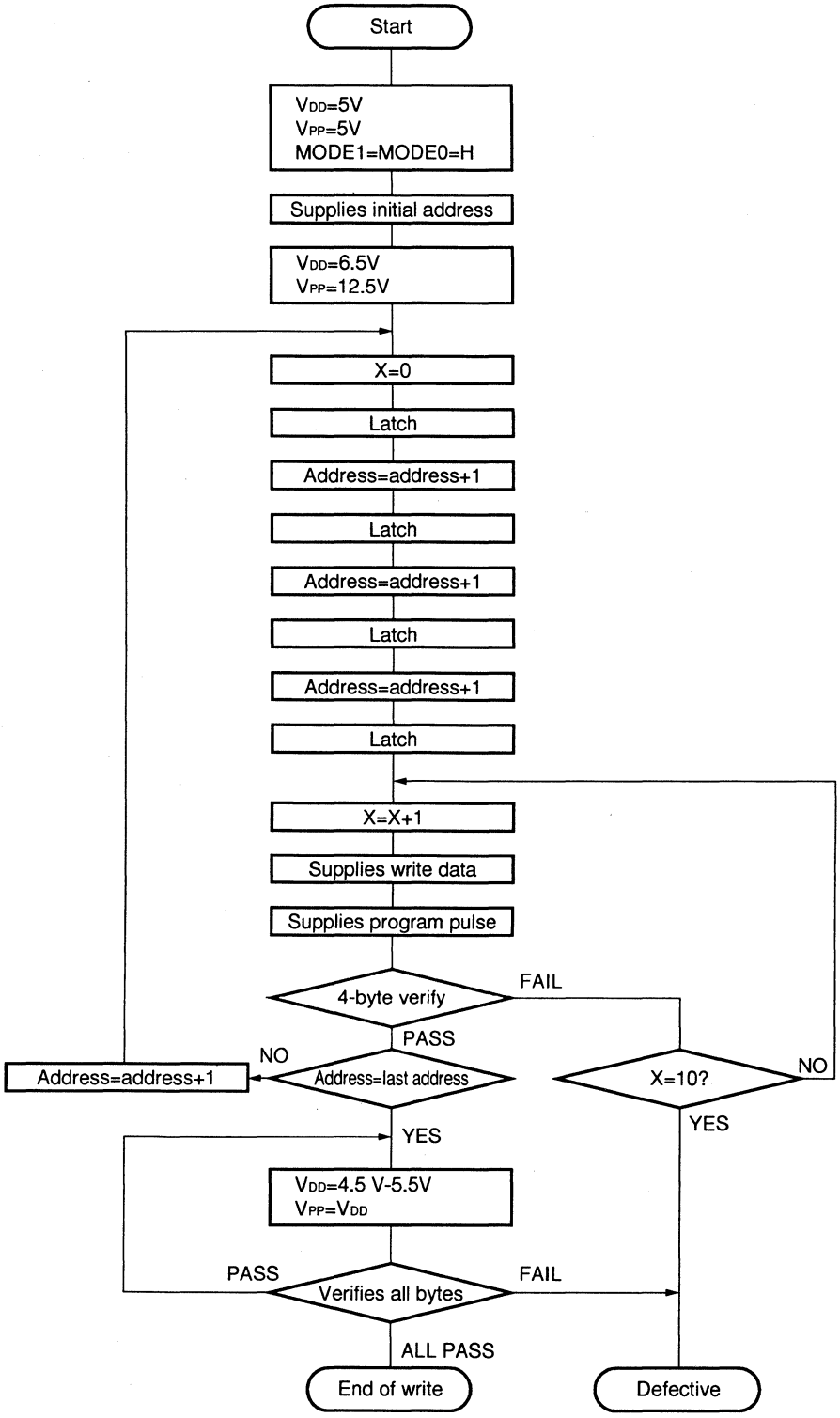
(8) Program inhibit mode

The program inhibit mode is used to write data to one of the μPD70P3000s whose $\overline{OE}$, V_{PP}, and D0-D7 pins are connected in parallel.

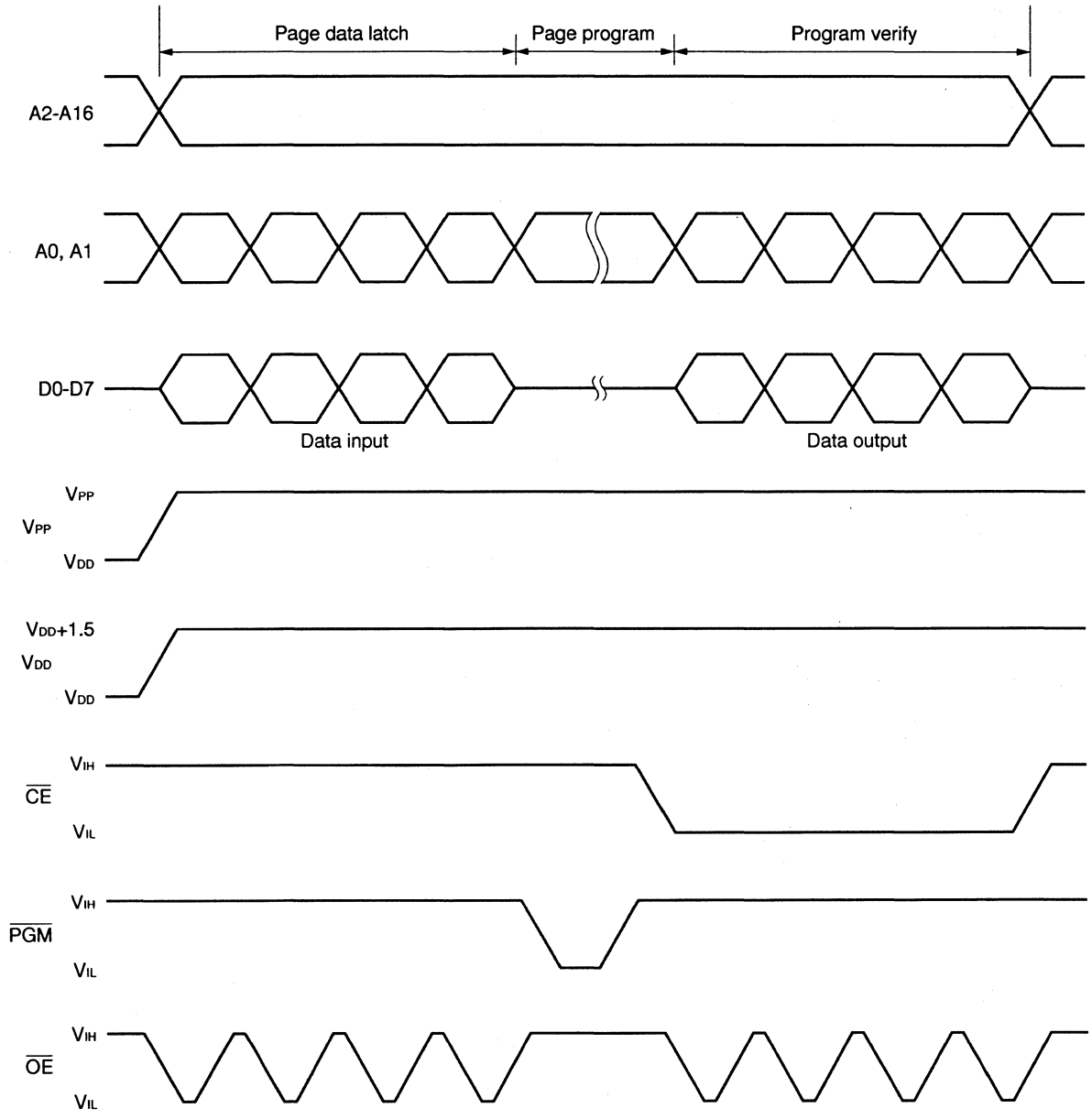
To write data, the page write mode or byte write mode is used. At this time, data cannot be written to a device whose $\overline{PGM}$ pin is high.

11.3 PROM Write Procedure

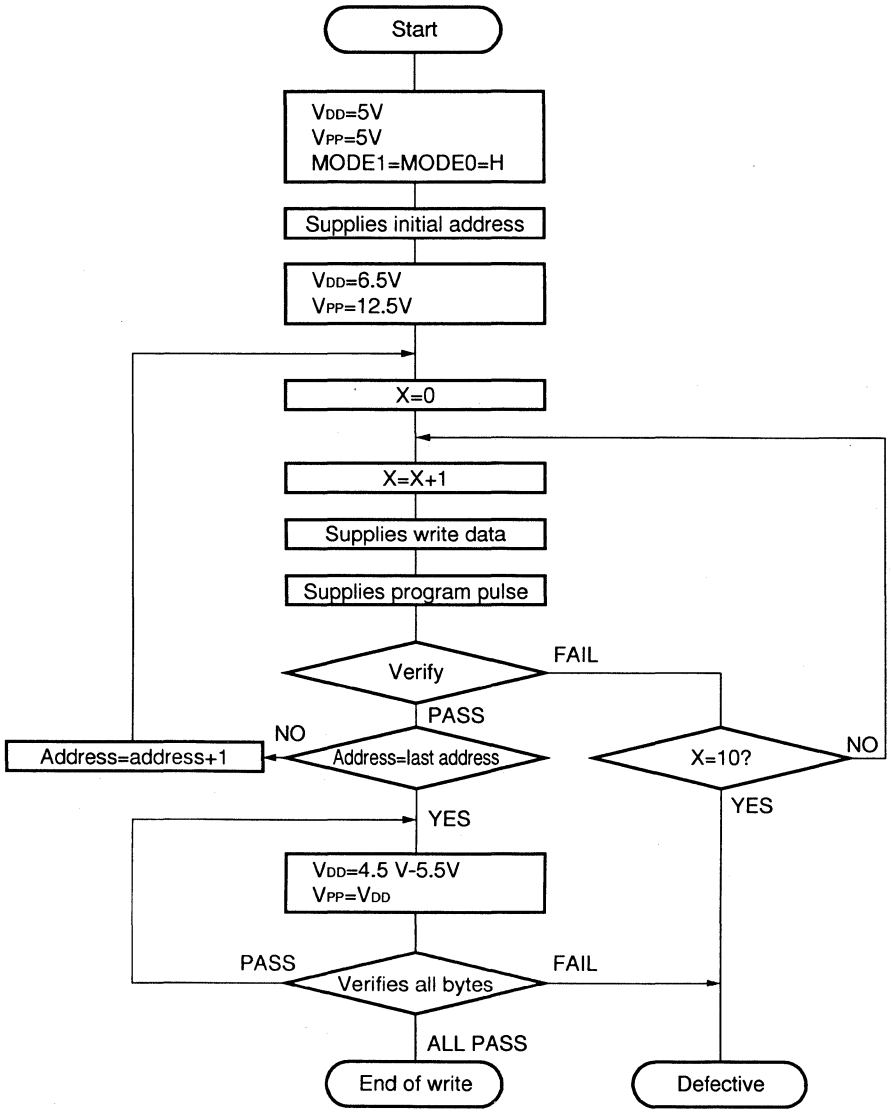
Page programming mode flowchart



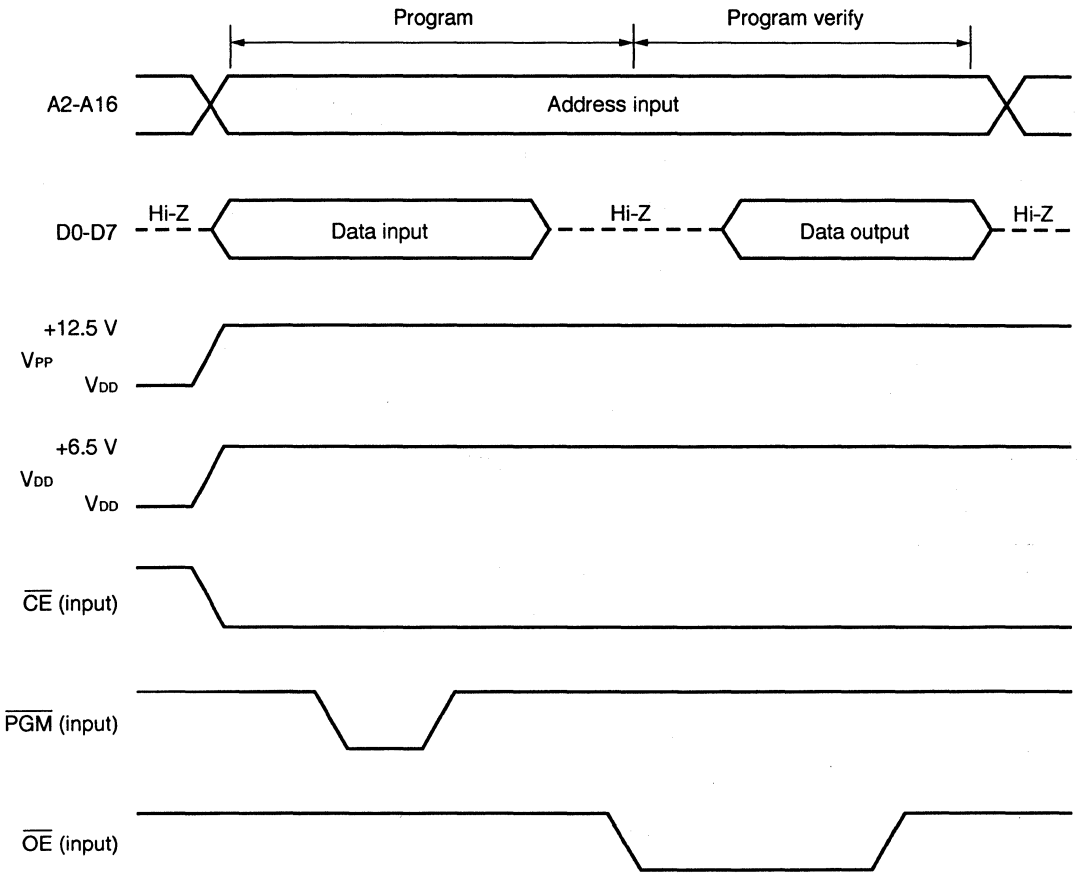
Page programming mode timing



Byte programming mode flowchart



Byte programming mode timing



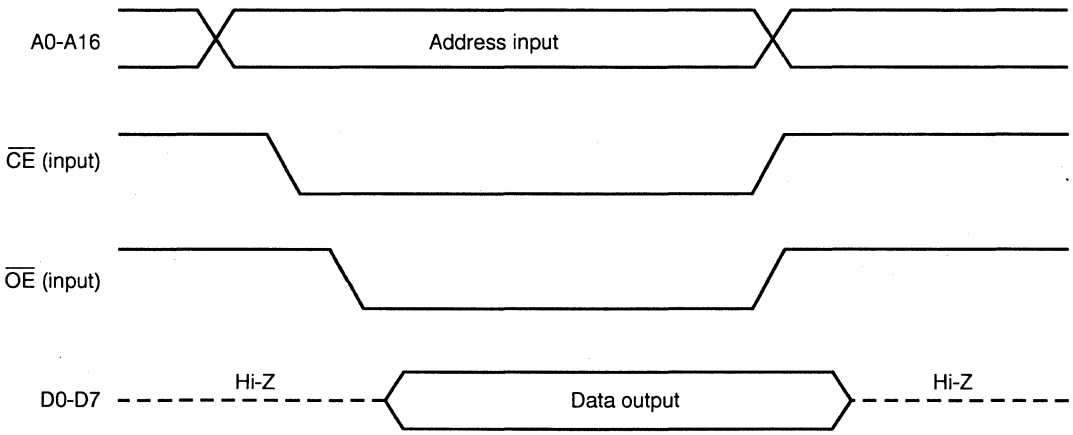
11.4 PROM Read Procedure

The contents of the PROM are read to the external data bus (D0-D7) in the following procedure:

(1) Fix V_{PP} = H, MODE0 = L, and MODE1 = L. Connect the unused pins as described in **1.5.2 PROM programming mode**.
(2) Supply +5 V to the V_{DD} and V_{PP} pins.
(3) Input the address of the data to be read to the A0-A16 pins.
(4) Read mode ($\overline{CE}$ = L, $\overline{OE}$ = L)
(5) The data is output to the D0-D7 pins.

Figure 11-1 shows the timing of (2) to (5) above.

Figure 11-1. PROM Read Timing



11.5 Screening of OTPROM Version

The one-time-programmable ROM (OTPROM) version, μPD70P3000GC-7EA, cannot be completely tested by NEC before shipment. It is recommended to perform screening to verify the PROM after the PROM has been stored under the following conditions:

Storage Temperature	Storage Time
125 °C	24 hours

★ 11.6 Notes on Releasing STOP Mode When External Clock Is Used

When an external clock is used, the external system controls clock supply. To release the STOP mode (by using $\overline{\text{RESET}}$ or NMI input), therefore, supply the clock at least 100 μs before the $\overline{\text{RESET}}$ or NMI signal is input.

APPENDIX A REGISTER INDEX

Symbol	Name	Unit	Page
ASIM00	Asynchronous serial interface mode register 00	UART	145
ASIM01	Asynchronous serial interface mode register 01	UART	147
ASIS0	Asynchronous serial interface status register 0	UART	148
BCC	Bus cycle control register	BCU	53
BPRM0	Baud rate generator prescaler mode register 0	BRG	170
BRG0	Baud rate generator register 0	BRG	170
CC10	Capture/compare register 10	RPU	113
CC11	Capture/compare register 11	RPU	113
CC12	Capture/compare register 12	RPU	113
CC13	Capture/compare register 13	RPU	113
CM4	Compare register 4	RPU	114
CMIC4	Interrupt control register	INTC	79
CSIC0	Interrupt control register	INTC	79
CSIM0	Clocked serial interface mode register 0	CSI	158
DWC	Data wait control register	BCU	51
ECR	Interrupt source register	CPU	28
EIPC	Interrupt status save register	CPU	28
EIPSW	Interrupt status save register	CPU	28
FEPC	NMI status save register	CPU	28
FEPSW	NMI status save register	CPU	28
INTM0	External interrupt mode register 0	INTC	71
INTM1	External interrupt mode register 1	INTC	80
INTM2	External interrupt mode register 2	INTC	80
ISPR	In-service priority register	INTC	81
MM	Memory expansion mode register	Port	44
OVIC1	Interrupt control register	INTC	79
P0	Port 0	Port	176
P1	Port 1	Port	178
P2	Port 2	Port	179
P3	Port 3	Port	183
P4	Port 4	Port	189
P5	Port 5	Port	191
P6	Port 6	Port	193
P9	Port 9	Port	194
P10	Port 10	Port	197
P0IC0	Interrupt control register	INTC	79
P0IC1	Interrupt control register	INTC	79

Symbol	Name	Unit	Page
P0IC2	Interrupt control register	INTC	79
P0IC3	Interrupt control register	INTC	79
P1IC0	Interrupt control register	INTC	79
P1IC1	Interrupt control register	INTC	79
P1IC2	Interrupt control register	INTC	79
P1IC3	Interrupt control register	INTC	79
PM0	Port 0 mode register	Port	176
PM1	Port 1 mode register	Port	179
PM2	Port 2 mode register	Port	182
PM3	Port 3 mode register	Port	187
PM4	Port 4 mode register	Port	190
PM5	Port 5 mode register	Port	192
PM6	Port 6 mode register	Port	194
PM9	Port 9 mode register	Port	196
PM10	Port 10 mode register	Port	199
PMC0	Port 0 mode control register	Port	177
PMC2	Port 2 mode control register	Port	183
PMC3	Port 3 mode control register	Port	188
PMC10	Port 10 mode control register	Port	199
PRCMD	Command register	CG	98
PSC	Power save control register	CG	96
PSW	Program status word	CPU	29,71,81,84
RXB0	Receive buffer 0	UART	149
RXB0L	Receive buffer 0L	UART	149
SEIC0	Interrupt control register	INTC	79
SIO0	Serial I/O shift register 0	CSI	159
SRIC0	Interrupt control register	INTC	79
STIC0	Interrupt control register	INTC	79
SYS	System status register	CG	93,98
TM1	Timer 1	RPU	112
TM4	Timer 4	RPU	114
TMC1	Timer control register 1	RPU	117
TMC4	Timer control register 4	RPU	118
TOC1	Timer output control register 1	RPU	119
TOVS	Timer overflow status register	RPU	120
TUM1	Timer unit mode register 1	RPU	115
TXS0	Transmit shift register 0	UART	150
TXS0L	Transmit shift register 0L	UART	150

APPENDIX B INSTRUCTION SET LIST

Legend

(1) Symbols used for operand description

Symbol	Description
reg1	General register (r0-r31): Used as source register
reg2	General register (r0-r31): Mainly used as destination register
immx	x-bit immediate
dispx	x-bit displacement
regID	System register number
bit#3	3-bit data for bit number specification
ep	Element pointer (r30)
cccc	4-bit data to indicate condition code
vector	5-bit data to specify trap vector (00H-1FH)

★

(2) Symbols used for operation description

Symbol	Description
←	Assignment
GR[ ]	General register
zero-extend(n)	Zero-extends n to word length
sign-extend(n)	Sign-extends n to word length
load-memory(a,b)	Reads data of size b from address a
store-memory(a,b,c)	Writes data b of size c to address a
load-memory-bit(a,b)	Reads bit b of address a
store-memory-bit(a,b,c)	Writes c to bit b of address a
saturated(n)	Performs saturated processing of n (n is 2's complement). If n is n ≥ 7FFFFFFFH as result of calculation, 7FFFFFFFH. If n is n ≤ 80000000H as result of calculation, 80000000H.
result	Reflects result on flag
Byte	Byte (8 bits)
Halfword	Half-word (16 bits)
Word	Word (32 bits)
+	Add
−	Subtract
||	Bit concatenation
×	Multiply
÷	Divide
AND	Logical product
OR	Logical sum
XOR	Exclusive logical sum

Symbol	Description
NOT	Logical negate
logically shift left by	Logical left shift
logically shift right by	Logical right shift
arithmetically shift right by	Arithmetic right shift

(3) Symbols used for execution clock description

Symbol	Description
i : issue	To execute another instruction immediately after instruction execution
r : repeat	To execute same instruction immediately after instruction execution
l : latency	To reference result of instruction execution by the next instruction

(4) Flag operation

Identifier	Description
(Blank)	Not affected
0	Cleared to 0
1	Set to 1
×	Set or cleared according to result
R	Previously saved value is restored

Condition code

Condition Name (cond)	Condition Code (cccc)	Conditional Expression	Description
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 No 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

Instruction Set (alphabetical order) (1/4)

Mnemonic	Operand	Code	Operation		Execution Clock i	Execution Clock r	Execution Clock I	Flag CY	Flag OV	Flag S	Flag Z	Flag 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	When condition satisfied	3	3	3					
			then PC←PC+sign-extned(disp9)	When condition not satisfied	1	1	1					
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)		4	4	4				×	
CMP	reg1, reg2	rrrrr001111RRRRR	result←GR[reg2]–GR[reg1]		1	1	1	×	×	×	×	
	imm5, reg2	rrrrr010011iiiii	result←GR[reg2]–sign-extend(imm5)		1	1	1	×	×	×	×	
DI		0000011111100000 0000000101100000	PSW.ID←1 (Maskable interrupt disabled)		1	1	1					
DIVH	reg1, reg2	rrrrr000010RRRRR	GR [reg2]←GR [reg2]÷GR [reg1]Note2 (signed division)		36	36	36		×	×	×	
EI		1000011111100000 0000000101100000	PSW.ID←0 (Maskable interrupt enabled)		1	1	1					
HALT		0000011111100000 0000000100100000	Stops		1	1	1					
JARL	disp22, reg2	rrrrr11110ddddd ddddddddddddddd0 **Note 3**	GR[reg2]←PC+4 PC←PC+sign-extend(disp22)		3	3	3					
JMP	[reg1]	00000000011RRRRR	PC←GR[reg1]		3	3	3					
JR	disp22	0000011110ddddd ddddddddddddddd0 **Note 3**	PC←PC+sign-extend(disp22)		3	3	3					
LD.B	disp16[reg1], reg2	rrrrr111000RRRRR dddddddddddddddd	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←sign-extend(Load-memory(adr, Byte))		1	1	2					
LD.H	disp16[reg1], reg2	rrrrr111001RRRRR ddddddddddddddd0 **Note 4**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]sign-extend(Load-memory(adr, Halfword))		1	1	2					
LD.W	disp16p[reg1], reg2	rrrrr111001RRRRR ddddddddddddddd1 **Note 4**	adr←GR[reg1]+sign-extend(disp16) GR[reg2]←Load-memory(adr, Word)		1	1	2					

Notes: 1. ddddddddd is the higher 8 bits of disp9.
2. Only the lower half-word is valid.
3. ddddddddddddddddddddd is the higher 21 bits of disp22.
4. ddddddddddddddd is the higher 15 bits of disp16.

Instruction Set (alphabetical order) (2/4)

Mnemonic	Operand	Code	Operation		Execution Clock i	Execution Clock r	Execution Clock l	Flag CY	Flag OV	Flag S	Flag Z	Flag SAT
★ LDSR	reg2, regID	rrrrr111111RRRRR	SR[regID]←GR[reg2]	regID = EIPC, FEPC	1	1	3					
		0000000000100000		regID = EIPSW, FEPSW			1					
		Note 1		regID = PSW			1	×	×	×	×	×
MOV	reg1, reg2	rrrrr000000RRRRR	GR[reg2]←GR[reg1]		1	1	1					
	imm5, reg2	rrrrr010000iiiii	GR[reg2]←sign-extend(imm5)		1	1	1					
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 ‖ 0^{16})		1	1	1					
MULH	reg1, reg2	rrrrr000111RRRRR	GR[reg2]←GR[reg2]Note2 ×GR[reg1]Note2 (Signed multiplication)		1	1	2					
	imm5, reg2	rrrrr010111iiiii	GR[reg2]←GR[reg2]Note2 ×sign-extend(imm5) (Signed multiplication)		1	1	2					
MULHI	imm16, reg1, reg2	rrrrr110111RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]Note2 ×imm16 (Signed multiplication)		1	1	2					
NOP		0000000000000000	Uses 1 clock cycle without doing anything		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)		4	4	4				×	
OR	reg1, reg2	rrrrr001000RRRRR	GR[reg2]←GR[reg2]OR GR[reg1]		1	1	1		0	×	×	
ORI	imm16, reg1, reg2	rrrrr110100RRRRR iiiiiiiiiiiiiiii	GR[reg2]←GR[reg1]OR zero-extend(imm16)		1	1	1		0	×	×	
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		4	4	4	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	×	×	

Notes: 1. The op code of this instruction uses the field of reg1 though the source register is shown as reg2 in the above table. Therefore, the meaning of register specification for mnemonic description and op code is different from that of the other instructions.

rrrrr = regID specification

RRRRR = reg2 specification

2. Only the lower half-word data is valid.

Instruction Set (alphabetical order) (3/4)

Mnemonic	Operand	Code	Operation	Execution Clock			Flag				
				i	r	l	CY	OV	S	Z	SAT
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	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					
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)	4	4	4				×	
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[reg1] 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	2					
SLD.H	disp8[ep], reg2	rrrrr1000ddddddd **Note 1**	adr←ep+zero-extend(disp8) GR[reg2]←sign-extend(Load-memory(adr, Halfword))	1	1	2					
SLD.W	disp8[ep], reg2	rrrrr1010ddddddd0 **Note 2**	adr←ep+zero-extend(disp8) GR[reg2]←Load-memory(adr, Word)	1	1	2					
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 1**	adr←ep+zero-extend(disp8) Store-memory(adr, GR[reg2], Halfword)	1	1	1						★
SST.W	reg2, disp8[ep]	rrrrr1010ddddddd1 **Note 2**	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						★

Notes: **1.** ddddddd is the higher 7 bits of disp8.

2. dddddd is the higher 6 bits of disp8.

Instruction Set (alphabetical order) (4/4)

Mnemonic	Operand	Code	Operation	Execution Clock			Flag				
				i	r	l	CY	OV	S	Z	SAT
★ ST.H	reg2, disp16[reg1]	r r r r r 1 1 1 0 1 1 R R R R R d d d d d d d d d d d d d d d 0 **Note**	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr, GR[reg2], Halfword)	1	1	1					
★ ST.W	reg2, disp16[reg1]	r r r r r 1 1 1 0 1 1 R R R R R d d d d d d d d d d d d d d d 1 **Note**	adr←GR[reg1]+sign-extend(disp16) Store-memory(adr, GR[reg2], Word)	1	1	1					
STSR	regID, reg2	r r r r r 1 1 1 1 1 1 R R R R R 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0	GR[reg2]←SR[regID]	1	1	1					
SUB	reg1, reg2	r r r r r 0 0 1 1 0 1 R R R R R	GR[reg2]←GR[reg2]–GR[reg1]	1	1	1	×	×	×	×	
SUBR	reg1, reg2	r r r r r 0 0 1 1 0 0 R R R R R	GR[reg2]←GR[reg1]–GR[reg2]	1	1	1	×	×	×	×	
TRAP	vector	0 0 0 0 0 1 1 1 1 1 1 i i i i i 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0	EIPC ←PC+4(Restored PC) EIPSW ←PSW ECR.EICC ←Interrupt code PSW.EP ←1 PSW.ID ←1 PC ←00000040H(vector=00H-0FH) 00000050H(vector=10H-1FH)	4	4	4					
TST	reg1, reg2	r r r r r 0 0 1 0 1 1 R R R R R	result←GR[reg2] AND GR[reg1]	1	1	1		0	×	×	
TST1	bit#3, disp16[reg1]	1 1 b b b 1 1 1 1 1 0 R R R R R d d d d d d d d d d d d d d d d	adr←GR[reg1]+sign-extend(disp16) Z flag←Not(Load-memory-bit(adr, bit#3))	3	3	3				×	
XOR	reg1, reg2	r r r r r 0 0 1 0 0 1 R R R R R	GR[reg2]←GR[reg2] XOR GR[reg1]	1	1	1		0	×	×	
XORI	imm16, reg1, reg2	r r r r r 1 1 0 1 0 1 R R R R R i i i i i i i i i i i i i i i i	GR[reg2]←GR[reg1] XOR zero-extend(imm16)	1	1	1		0	×	×	

Note: ddddddddddddddd is the higher 15 bits of disp16.

APPENDIX C INDEX

[B]

[C]

[D]

[E]

[F]

[L]

[M]

[P]

[S]

[T]

[U]

[MEMO]

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

Hong Kong, Philippines, Oceania
NEC Electronics Hong Kong Ltd.
Fax: +852-2886-9022/9044

Asian Nations except Philippines
NEC Electronics Singapore Pte. Ltd.
Fax: +65-250-3583

Europe
NEC Electronics (Europe) GmbH
Technical Documentation Dept.
Fax: +49-211-6503-274

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

Japan
NEC Corporation
Semiconductor Solution Engineering Division
Technical Information Support Dept.
Fax: 044-548-7900

South America
NEC do Brasil S.A.
Fax: +55-11-889-1689

Taiwan
NEC Electronics Taiwan Ltd.
Fax: 02-719-5951

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	❑	❑	❑	❑

CS 96.4