To all our customers

# Regarding the change of names mentioned in the document, such as Mitsubishi Electric and Mitsubishi XX, to Renesas Technology Corp.

The semiconductor operations of Hitachi and Mitsubishi Electric were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Mitsubishi Electric, Mitsubishi Electric Corporation, Mitsubishi Semiconductors, and other Mitsubishi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Note : Mitsubishi Electric will continue the business operations of high frequency & optical devices and power devices.

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

RENESAS
Renesas Technology Corp.

# 3874 Group

SINGLE-CHIP 8-BIT CMOS MICROCOMPUTER

## DESCRIPTION

The 3874 group is the 8-bit microcomputer based on the 740 family core technology.

The 3874 group includes data link layer communication control circuit, A-D converters, D-A converter, automatic data transfer serial I/O, UART, and watchdog timer etc.

The various microcomputers in the 3874 group include variations of internal memory size and packaging. For details, refer to the section on part numbering.

For details on availability of microcomputers in the 3874 group, refer to the section on group expansion.

## FEATURES

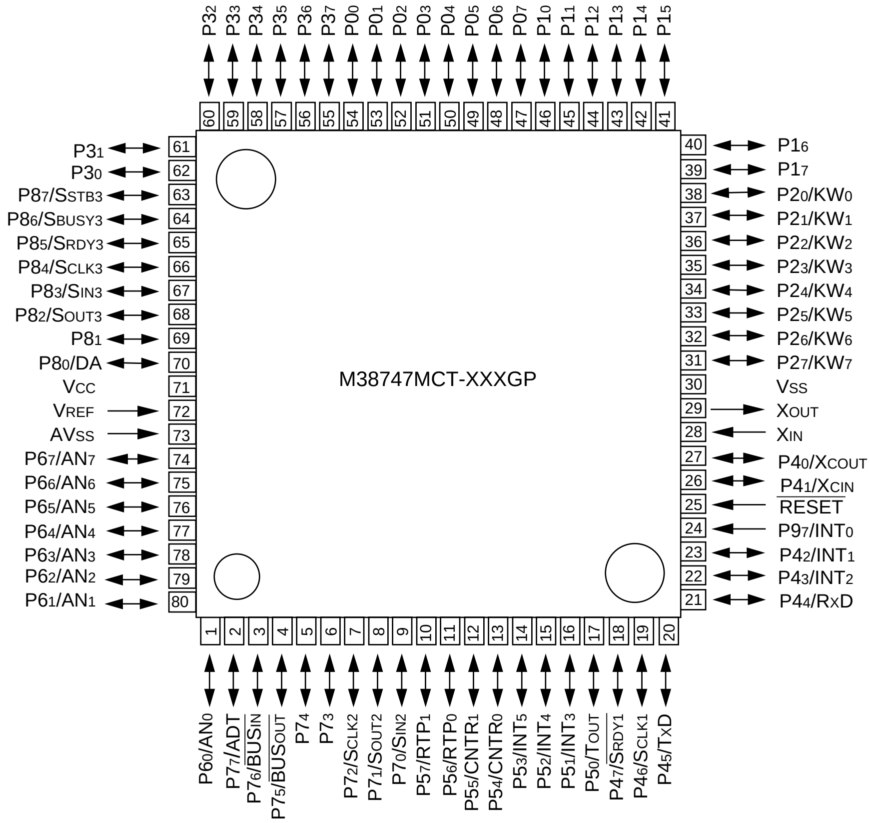
- Basic machine-language instructions ...................................... 71
- Minimum instruction execution time................................. 0.32 µs
  (at 6.4 MHz oscillation frequency, in double-speed mode)
- Memory size
  ROM .............................................................. 16 K to 60 K bytes
  RAM .............................................................. 1024 to 2048 bytes
- Programmable input/output ports ............................................ 72
- Input port .................................................................................... 1
- Interrupts ................................................ 27 sources, 16 vectors
  (Interrupt source discrimination register exists, included key input interrupt)
- Timer 1, timer 2, timer 3 ................................................ 8-bit ✕ 3
- Timer X, timer Y........................................................... 16-bit ✕ 2
- Serial I/O1 .................... 8-bit ✕ 1(UART or Clock-synchronized)
- Serial I/O2 .................................. 8-bit ✕ 1(Clock-synchronized)
- Serial I/O3 ..................................................................... 8-bit ✕ 1
  (Clock-synchronized automatic data transfer/arbitrary bit transfer function available)
- A-D converter ................................................ 8-bit ✕ 8 channels
- D-A converter .................................................. 8-bit ✕ 1 channel
- Data link layer communication control circuit ............................ 1
- Clock generating circuit..................................... Built-in 2 circuits
  (connect to external ceramic resonator or quartz-crystal oscillator)
- Watchdog timer ............................................................ 20-bit ✕ 1
- Power source voltage ............................................... 3.0 to 5.5 V
- Power dissipation
  In double-speed mode ......................................................90 mW
  In high-speed mode .........................................................60 mW
  (at 32 kHz oscillation frequency, at 5 V power source voltage)
  In low-speed mode ......................................................... 180 µW
  (at 32 kHz oscillation frequency, at 3 V power source voltage)
- Operating temperature range...................................–40 to 85°C
  (Extended operating temperature version and automotive version)

## APPLICATION

Automotive comfort control for audio system, air conditioning etc., automotive body electronics control, household appliances, and other consumer applications, etc.

## PIN CONFIGURATION

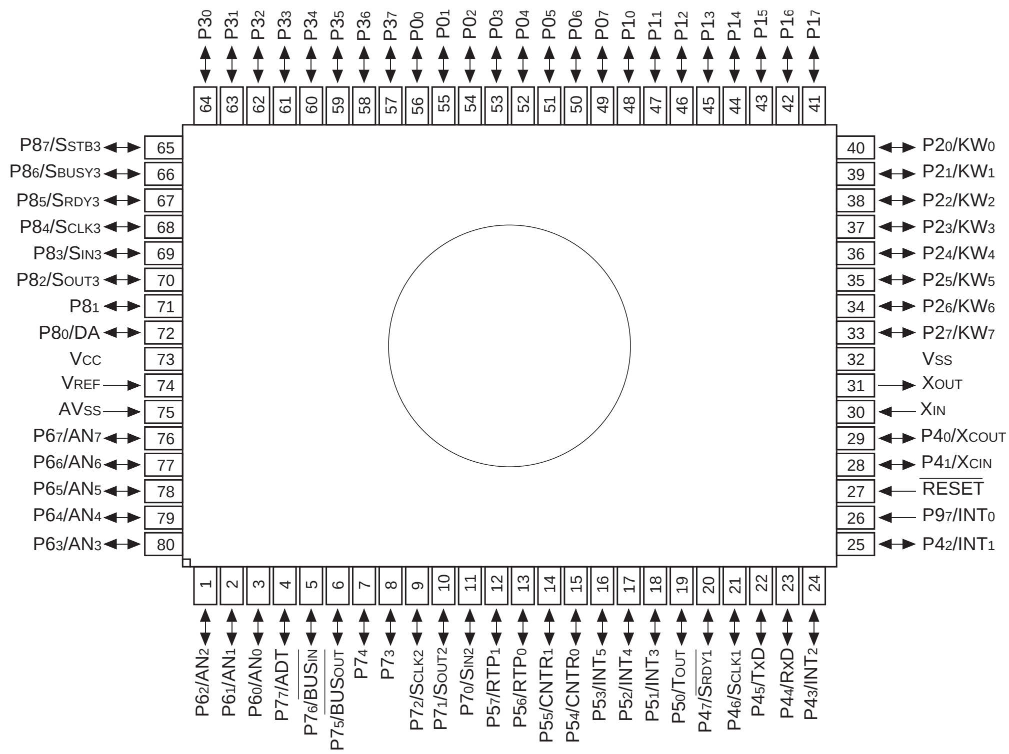
PIN CONFIGURATION (TOP VIEW)




**Package type : 80P6S-A**

Fig. 1 M38747MCT-XXXGP pin configuration

## PIN CONFIGURATION (TOP VIEW)




**Package type : 80D0**

Fig. 2 M38749EFFS pin configuration

## FUNCTIONAL BLOCK

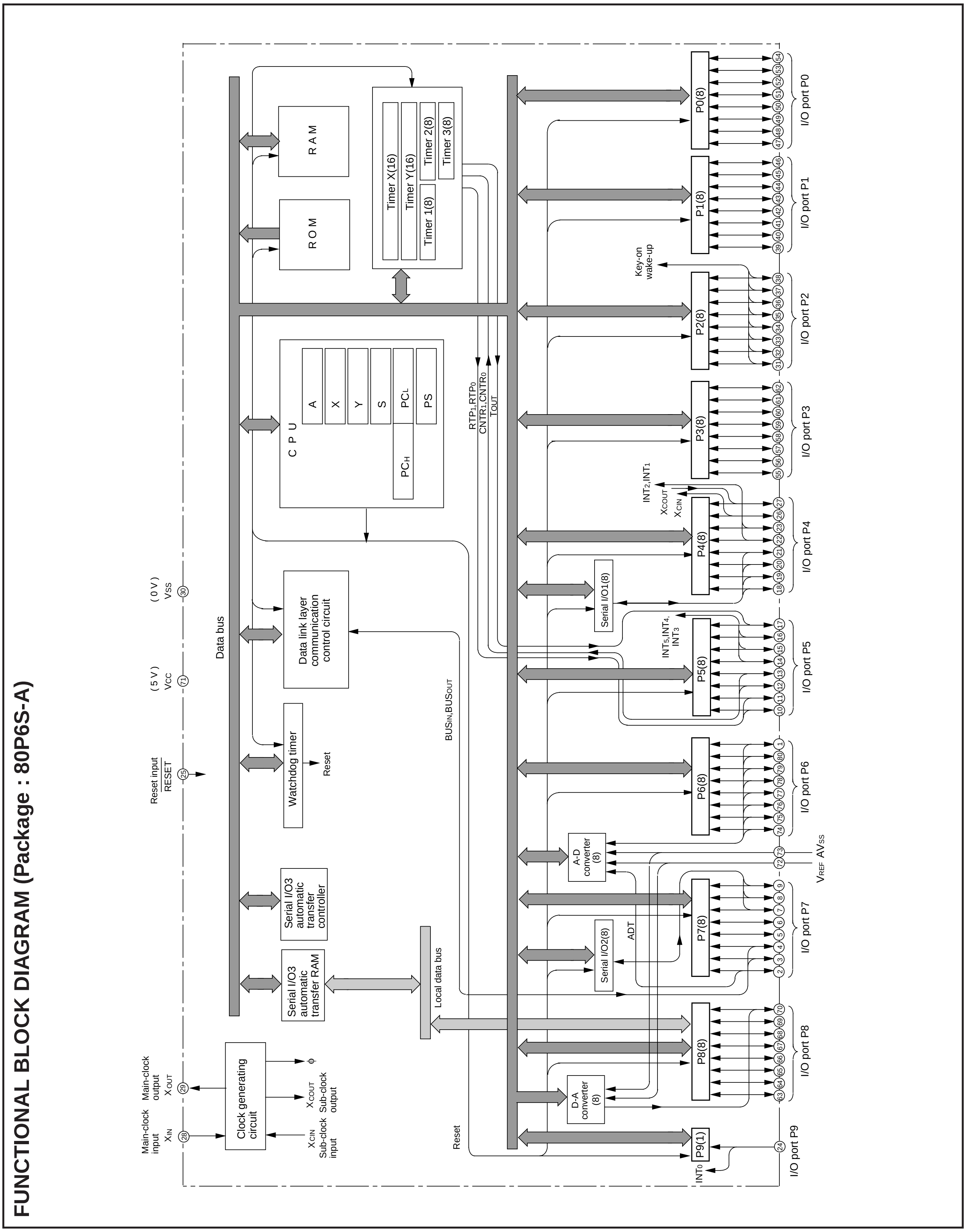



Fig. 3 Functional block diagram

## PIN DESCRIPTION

**Table 1 Pin description (1)**

| Pin | Name | Functions | Function except a port function |
|---|---|---|---|
| $V_{CC}$, $V_{SS}$ | Power source input | •Apply voltage of 3.0 V – 5.5 V to Vcc, and 0 V to Vss. | |
| $V_{REF}$ | Reference voltage input | •Reference voltage input pin for A-D and D-A converters. | |
| $AV_{SS}$ | Analog power source input | •Analog power source input pin for A-D and D-A converters.<br>•Connect to $V_{SS}$. | |
| $\overline{RESET}$ | Reset input | •Reset input pin for active “L.” | |
| $X_{IN}$ | Clock input | •Input and output pins for the clock generating circuit.<br>•Connect a ceramic resonator or quartz-crystal oscillator between the $X_{IN}$ and $X_{OUT}$ pins to set the oscillation frequency.<br>•When an external clock is used, connect the clock source to the $X_{IN}$ pin and leave the $X_{OUT}$ pin open.<br>•Feedback resistor is built in between $X_{IN}$ pin and $X_{OUT}$ pin. | |
| $X_{OUT}$ | Clock output | | |
| $P0_0$–$P0_7$ | I/O port P0 | •8-bit CMOS I/O port.<br>•I/O direction register allows each pin to be individually programmed as either input or output.<br>•CMOS compatible input level.<br>•CMOS 3-state output structure. | |
| $P1_0$–$P1_7$ | I/O port P1 | | |
| $P2_0$–$P2_7$ | I/O port P2 | | |
| $P3_0$–$P3_7$ | I/O port P3 | | |
| $P4_0/X_{COUT}$, $P4_1/X_{CIN}$ | I/O port P4 | •8-bit I/O port with the same function as port P0.<br>•CMOS compatible input level.<br>•CMOS 3-state output structure. | •Sub-clock generating circuit I/O pins connect a resonator. (This circuit cannot be operated by an external clock.) |
| $P4_2/INT_1$, $P4_3/INT_2$ | | | •Interrupt input pins |
| $P4_4/R_XD$, $P4_5/T_XD$, $P4_6/\overline{S_{CLK1}}$, $P4_7/S_{RDY1}$ | | | •Serial I/O1 function pins |
| $P5_0/T_{OUT}$ | I/O port P5 | •8-bit I/O port with the same function as port P0.<br>•CMOS compatible input level.<br>•CMOS 3-state output structure. | •Timer 2 output pin |
| $P5_1/INT_3$–$P5_3/INT_5$ | | | •Interrupt input pins |
| $P5_4/CNTR_0$, $P5_5/CNTR_1$ | | | •Timer X, timer Y function pins |
| $P5_6/RTP_0$, $P5_7/RTP_1$ | | | •Real time port function pins |

**Table 2 Pin description (2)**

| Pin | Name | Functions | Function except a port function |
|---|---|---|---|
| $P6_0/AN_0$–<br>$P6_7/AN_7$ | I/O port P6 | •8-bit I/O port with the same function as port P0.<br>•CMOS compatible input level.<br>•CMOS 3-state output structure. | •A-D converter input pins |
| $P7_0/S_{IN2}$,<br>$P7_1/S_{OUT2}$,<br>$P7_2/S_{CLK2}$ | I/O port P7 | •8-bit I/O port with the same function as port P0.<br>•CMOS compatible input level.<br>•CMOS 3-state output structure. | •Serial I/O2 function pins |
| $P7_3$, $P7_4$ | | | |
| $P7_5/\overline{BUS_{OUT}}$,<br>$P7_6/\overline{BUS_{IN}}$ | | | •Data link layer communication con-trol pins |
| $P7_7/ADT$ | | | •A-D trigger input pin |
| $P8_0/DA$ | I/O port P8 | •8-bit I/O port with the same function as port P0. | •D-A converter output pin |
| $P8_1$ | | | |
| $P8_2/S_{OUT3}$,<br>$P8_3/S_{IN3}$,<br>$P8_4/S_{CLK3}$,<br>$P8_5/S_{RDY3}$ | | | •Serial I/O3 function pins |
| $P8_6/S_{BUSY3}$,<br>$P8_7/S_{STB3}$ | | | |
| $P9_7/INT_0$ | Input port P9 | •1-bit input port.<br>•CMOS compatible input level. | •Interrupt input pin |

## PART NUMBERING

Product M3874 7 M C T- XXX GP

Package type
GP : 80P6S-A
FS : 80D0

ROM number
Omitted in some types.

D– : Extended operating temperature version
F– : Extended operating speed version of “D–”
T– : Automotive version

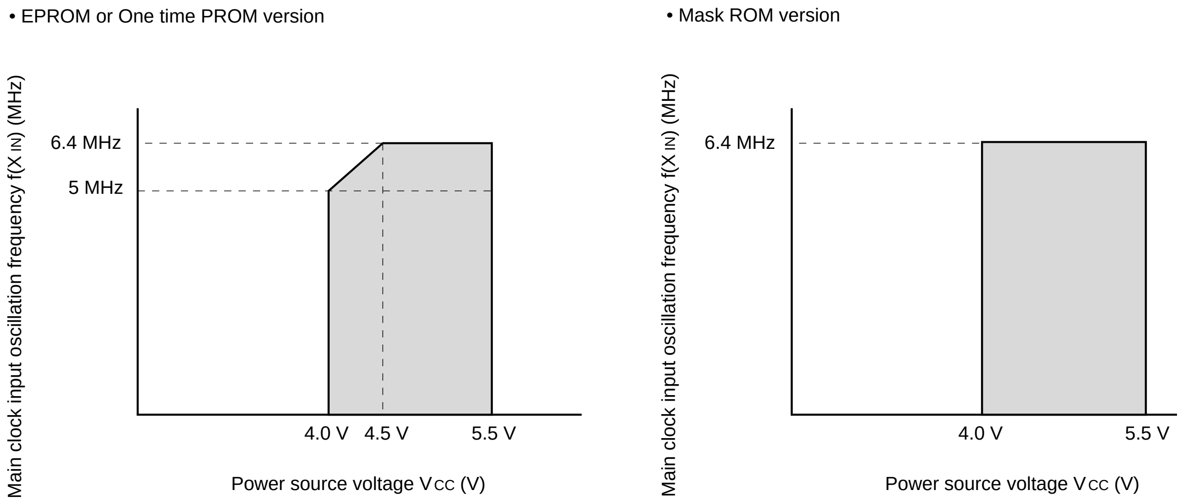
ROM/PROM size
4 : 16384 bytes
5 : 20480 bytes
6 : 24576 bytes
7 : 28672 bytes
8 : 32768 bytes
9 : 36864 bytes
A : 40960 bytes
B : 45056 bytes
C : 49152 bytes
D : 53248 bytes
E : 57344 bytes
F : 61440 bytes
The first 128 bytes and the last 2 bytes of ROM are reserved areas ; they cannot be used.

Memory type
M : Mask ROM version
E : EPROM or One Time PROM version

RAM size
7 : 1024 bytes
8 : 1536 bytes
9 : 2048 bytes

Fig. 4 Part numbering

## 3874 group main clock input oscillation frequency in double-speed mode




In low-speed mode, middle-speed mode, and high-speed mode, characteristic of main clock input oscillation frequency guarantee limit is not different.

Fig. 5 Main clock input oscillation frequency in double-speed mode

## GROUP EXPANSION (Extended operating temperature version)

The 3874 group (extended operating temperature version) is designed for automotive comfort and amusement control such as audio, air-conditioner etc., household appliances, and other consumer applications.
Mitsubishi plans to expand the 3874 group (extended operating temperature version) as follows:

### Memory Type

Support for mask ROM, One Time PROM, and EPROM versions

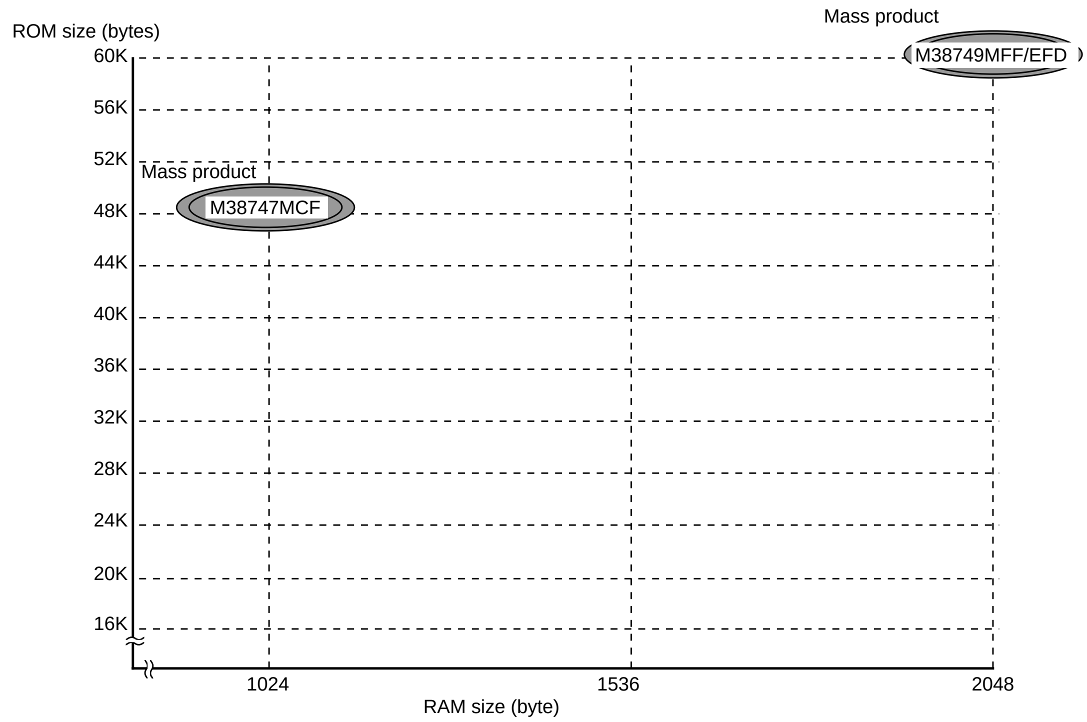
### Memory Size

ROM/PROM size ............................................... 48 K to 60 K bytes
RAM size .......................................................... 1024 to 2048 bytes

### Packages

80P6S-A .................................. 0.65 mm-pitch plastic molded QFP
80D0 ....................... 0.8 mm-pitch ceramic LCC (EPROM version)

**Memory Expansion Plan of 3874 group (Extended operating temperature version)**




Products under development or planning : the development schedule and specification may be revised without notice.
Planning products may be stopped during the development.

Fig. 6 Memory expansion plan (Extended operating temperature version)

Currently planning products are listed below.

**Table 3 Support products** As of March 1998

| Product name | (P) ROM size (bytes) ROM size for User in ( ) | RAM size (bytes) | Package | Remarks |
|---|---|---|---|---|
| M38749EFDGP | 61440 (61310) | 2048 | 80P6S-A | One Time PROM version (blank) |
| M38749EFFS | | | 80D0 | EPROM version (for software development, operating temperature = –20 to 85°C) |
| M38749MFF | | | 80P6S-A | Mask ROM version |
| M38747MCF | 49152 (49022) | 1024 | | |

## GROUP EXPANSION (Automotive version)

The 3874 group (automotive version) is designed for automotive body electronics control.
Mitsubishi plans to expand the 3874 group (automotive version) as follows:

### Memory Type

Support for mask ROM and One Time PROM versions

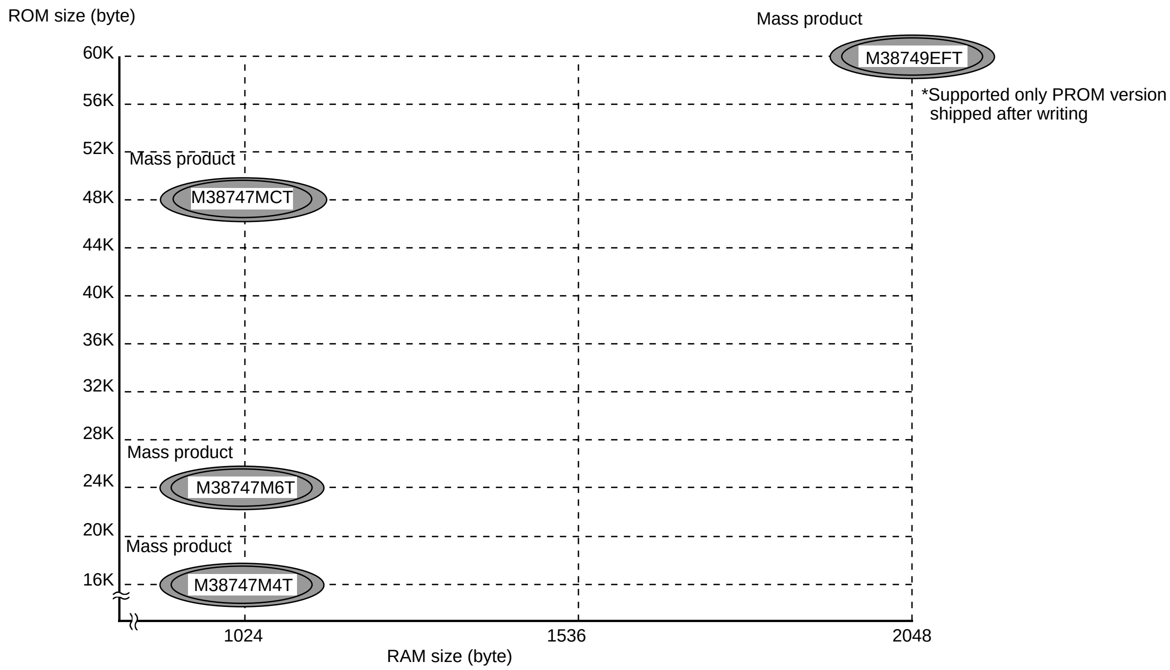
### Memory Size

ROM/PROM size ............................................... 16 K to 60 K bytes
RAM size ......................................................... 1024 to 2048 bytes

### Packages

80P6S-A .................................. 0.65 mm-pitch plastic molded QFP

**Memory Expansion Plan of 3874 group (Automotive version)**




Products under development or planning : the development schedule and specification may be revised without notice.
Planning products may be stopped during the development.

Fig. 7 Memory expansion plan (Automotive correspondence version)

Currently planning products are listed below.

**Table 4 Support products** As of March 1998

| Product name | (P) ROM size (bytes) ROM size for User in ( ) | RAM size (bytes) | Package | Remarks |
|---|---|---|---|---|
| M38749EFT | 61440 (61310) | 2048 | 80P6S-A | One Time PROM version |
| M38747MCT | 49152 (49022) | 1048 | | Mask ROM version |
| M38747M6T | 24576 (24446) | | | |
| M38747M4T | 16384 (16254) | | | |

## FUNCTIONAL DESCRIPTION
## CENTRAL PROCESSING UNIT (CPU)

The 3874 group uses the standard 740 Family instruction set. Refer to the table of 740 Family addressing modes and machine instructions or the 740 Family Software Manual for details on the instruction set.

Machine-resident 740 Family instructions are as follows:

The FST and SLW instructions cannot be used.

The STP, WIT, MUL, and DIV instructions can be used.

## [CPU Mode Register (CPUM)] $003B_{16}$

The CPU mode register contains the stack page selection bit and the internal system clock selection bit etc.

The CPU mode register is allocated at address $003B_{16}$.

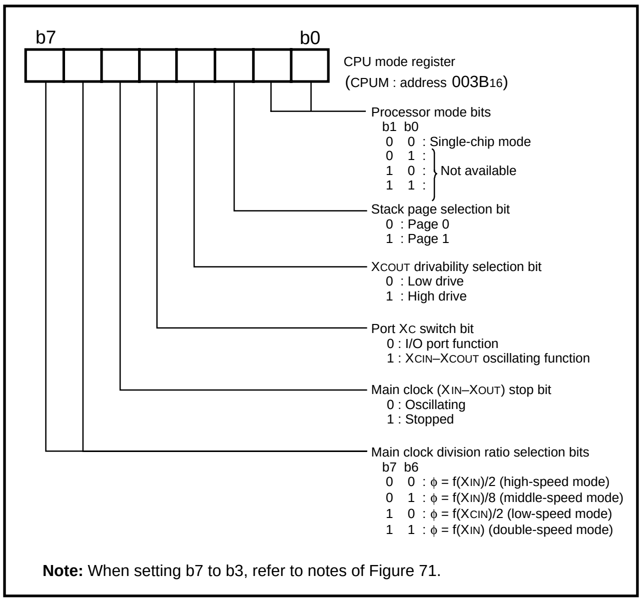



Fig. 8 Structure of CPU mode register

## MEMORY

### Special Function Register (SFR) Area

The Special Function Register area in the zero page contains control registers such as I/O ports and timers.

### RAM

RAM is used for data storage and for stack area of subroutine calls and interrupts.

### ROM

The first 128 bytes and the last 2 bytes of ROM are reserved for device testing and the rest is user area for storing programs.

### Interrupt Vector Area

The interrupt vector area contains reset and interrupt vectors.

### Zero Page

Access to this area with only 2 bytes is possible in the zero page addressing mode.

### Special Page

Access to this area with only 2 bytes is possible in the special page addressing mode.

RAM area

| RAM size (bytes) | Address $XXXX_{16}$ |
|---|---|
| 1024 | $043F_{16}$ |
| 1536 | $063F_{16}$ |
| 2048 | $083F_{16}$ |

ROM area

| ROM size (bytes) | Address $YYYY_{16}$ | Address $ZZZZ_{16}$ |
|---|---|---|
| 16384 | $C000_{16}$ | $C080_{16}$ |
| 20480 | $B000_{16}$ | $B080_{16}$ |
| 24576 | $A000_{16}$ | $A080_{16}$ |
| 28672 | $9000_{16}$ | $9080_{16}$ |
| 32768 | $8000_{16}$ | $8080_{16}$ |
| 36864 | $7000_{16}$ | $7080_{16}$ |
| 40960 | $6000_{16}$ | $6080_{16}$ |
| 45056 | $5000_{16}$ | $5080_{16}$ |
| 49152 | $4000_{16}$ | $4080_{16}$ |
| 53248 | $3000_{16}$ | $3080_{16}$ |
| 57344 | $2000_{16}$ | $2080_{16}$ |
| 61440 | $1000_{16}$ | $1080_{16}$ |

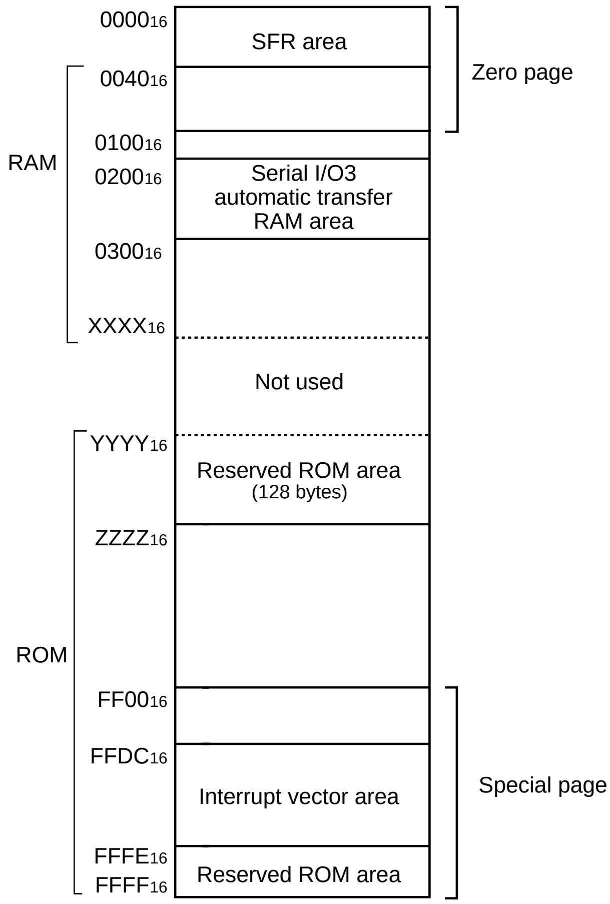



Fig. 9 Memory map diagram

| Address | Register | Address | Register |
|---|---|---|---|
| $0000_{16}$ | Port P0 (P0) | $0020_{16}$ | Timer X (low-order) (TXL) |
| $0001_{16}$ | Port P0 direction register (P0D) | $0021_{16}$ | Timer X (high-order) (TXH) |
| $0002_{16}$ | Port P1 (P1) | $0022_{16}$ | Timer Y (low-order) (TYL) |
| $0003_{16}$ | Port P1 direction register (P1D) | $0023_{16}$ | Timer Y (high-order) (TYH) |
| $0004_{16}$ | Port P2 (P2) | $0024_{16}$ | Timer 1 (T1) |
| $0005_{16}$ | Port P2 direction register (P2D) | $0025_{16}$ | Timer 2 (T2) |
| $0006_{16}$ | Port P3 (P3) | $0026_{16}$ | Timer 3 (T3) |
| $0007_{16}$ | Port P3 direction register (P3D) | $0027_{16}$ | Timer X mode register (TXM) |
| $0008_{16}$ | Port P4 (P4) | $0028_{16}$ | Timer Y mode register (TYM) |
| $0009_{16}$ | Port P4 direction register (P4D) | $0029_{16}$ | Timer 123 mode register (T123M) |
| $000A_{16}$ | Port P5 (P5) | $002A_{16}$ | Communication mode register (BUSM) |
| $000B_{16}$ | Port P5 direction register (P5D) | $002B_{16}$ | Transmit control register (TXDCON) |
| $000C_{16}$ | Port P6 (P6) | $002C_{16}$ | Transmit status register (TXDSTS) |
| $000D_{16}$ | Port P6 direction register (P6D) | $002D_{16}$ | Receive control register (RXDCON) |
| $000E_{16}$ | Port P7 (P7) | $002E_{16}$ | Receive status register (RXDSTS) |
| $000F_{16}$ | Port P7 direction register (P7D) | $002F_{16}$ | Bus interrupt source discrimination control register (BICOND) |
| $0010_{16}$ | Port P8 (P8) | $0030_{16}$ | Control field selection register (CFSEL) |
| $0011_{16}$ | Port P8 direction register (P8D) | $0031_{16}$ | Control field register (CF) |
| $0012_{16}$ | Port P9 (P9) | $0032_{16}$ | Transmit/Receive FIFO (TRFIFO) |
| $0013_{16}$ | Serial I/O3 register/Transfer counter (SIO3) | $0033_{16}$ | PULL UP register (PULLU) |
| $0014_{16}$ | Serial I/O3 control register 1 (SIO3CON1) | $0034_{16}$ | A-D control register (ADCON) |
| $0015_{16}$ | Serial I/O3 control register 2 (SIO3CON2) | $0035_{16}$ | A-D/D-A conversion register (AD) |
| $0016_{16}$ | Serial I/O3 control register 3 (SIO3CON3) | $0036_{16}$ | Interrupt source discrimination register 2 (IREQD2) |
| $0017_{16}$ | Serial I/O3 automatic transfer data pointer (SIO3DP) | $0037_{16}$ | Interrupt source discrimination control register 2 (ICOND2) |
| $0018_{16}$ | Transmit/Receive buffer register (TB/RB) | $0038_{16}$ | Interrupt source discrimination register 1 (IREQD1) |
| $0019_{16}$ | Serial I/O1 status register (SIO1STS) | $0039_{16}$ | Interrupt source discrimination control register 1 (ICOND1) |
| $001A_{16}$ | Serial I/O1 control register (SIO1CON) | $003A_{16}$ | Interrupt edge selection register (INTEDGE) |
| $001B_{16}$ | UART control register (UARTCON) | $003B_{16}$ | CPU mode register (CPUM) |
| $001C_{16}$ | Baud rate generator (BRG) | $003C_{16}$ | Interrupt request register 1 (IREQ1) |
| $001D_{16}$ | Serial I/O2 control register (SIO2CON) | $003D_{16}$ | Interrupt request register 2 (IREQ2) |
| $001E_{16}$ | Watchdog timer control register (WDTCON) | $003E_{16}$ | Interrupt control register 1 (ICON1) |
| $001F_{16}$ | Serial I/O2 register (SIO2) | $003F_{16}$ | Interrupt control register 2 (ICON2) |

Fig. 10 Memory map of special function register (SFR)

## I/O PORTS

The I/O ports P0–P8 have direction registers which determine the input/output direction of each individual pin. Each bit in a direction register corresponds to one pin, and each pin can be set to be input port or output port.

When “0” is written to the bit corresponding to a pin, that pin becomes an input pin. When “1” is written to that bit, that pin becomes an output pin.

If data is read from a pin which is set to output, the value of the port output latch is read, not the value of the pin itself. Pins set to input are floating. If a pin set to input is written to, only the port output latch is written to and the pin remains floating.

**Table 5 I/O port function (1)**

| Pin | Name | Input/Output | I/O Structure | Non-Port Function | Related SFRs | Ref.No. |
|---|---|---|---|---|---|---|
| $P0_0$–$P0_7$ | Port P0 | Input/output, individual bits | •CMOS compatible input level<br>•CMOS 3-state output | | | (1) |
| $P1_0$–$P1_7$ | Port P1 | Input/output, individual bits | •CMOS compatible input level<br>•CMOS 3-state output | | | |
| $P2_0$–$P2_7$ | Port P2 | Input/output, individual bits | •CMOS compatible input level<br>•CMOS 3-state output | •Key input (key-on wake-up) interrupt input | •PULL UP register | (2) |
| $P3_0$–$P3_7$ | Port P3 | Input/output, individual bits | •CMOS compatible input level<br>•CMOS 3-state output | | •CPU mode register | (1) |
| $P4_0$/$X_{COUT}$ | Port P4 | Input/output, individual bits | •CMOS compatible input level<br>•CMOS 3-state output | •Sub-clock generating circuit I/O | •CPU mode register | (3) |
| $P4_1$/$X_{CIN}$ | | | | | | (4) |
| $P4_2$/$INT_1$,<br>$P4_3$/$INT_2$ | | | | •External interrupt input | •Interrupt edge selection register | (5) |
| $P4_4$/$R_XD$ | | | | •Serial I/O1 function I/O | •Serial I/O1 control register<br>•Serial I/O1 status register<br>•UART control register<br>•PULL UP register | (6) |
| $P4_5$/$T_XD$ | | | | | | (7) |
| $P4_6$/$S_{CLK1}$ | | | | | | (8) |
| $P4_7$/$\overline{S_{RDY1}}$ | | | | | | (9) |
| $P5_0$/$T_{OUT}$ | Port P5 | Input/output, individual bits | •CMOS compatible input level<br>•CMOS 3-state output | •Timer 2 output | •Timer 123 mode register | (10) |
| $P5_1$/$INT_3$,<br>$P5_2$/$INT_4$,<br>$P5_3$/$INT_5$ | | | | •External interrupt input | •Interrupt edge selection register | (5) |
| $P5_4$/$CNTR_0$ | | | | •Timer X function I/O | •Timer X mode register | (11) |
| $P5_5$/$CNTR_1$ | | | | •Timer Y function I/O | •Timer Y mode register | (12) |
| $P5_6$/$RTP_0$ | | | | •Real time port function output | •Timer X mode register | (13) |
| $P5_7$/$RTP_1$ | | | | •Real time port function output | •Timer Y mode register | |
| $P6_0$/$AN_0$–<br>$P6_7$/$AN_7$ | Port P6 | Input/output, individual bits | •CMOS compatible input level<br>•CMOS 3-state output | •A-D converter input | •A-D control register | (14) |

**Table 6  I/O port function (2)**

| Pin | Name | Input/Output | I/O Function | Non-Port Function | Related SFRs | Ref.No. |
|---|---|---|---|---|---|---|
| $P7_0/S_{IN2}$ | Port P7 | Input/output,<br>individual bits | •CMOS compatible input level<br>•CMOS 3-state output | •Serial I/O2 function I/O | •Serial I/O2 control register<br>•PULL UP register | (15) |
| $P7_1/S_{OUT2}$ | | | | | | (16) |
| $P7_2/S_{CLK2}$ | | | | | | (17) |
| $P7_3$,$P7_4$ | | | | | | (1) |
| $P7_5/\overline{BUS_{OUT}}$ | | | | •Data link layer communication control I/O | •Communication mode register<br>•Transmit control register<br>•Transmit status register<br>•Receive control regiser<br>•Receive status register<br>•Bus interrupt source discrimination control register<br>•Control field selection register<br>•Control field register<br>•Transmit/Receive FIFO | (18) |
| $P7_6/\overline{BUS_{IN}}$ | | | | | | (19) |
| $P7_7$/ADT | | | | •A-D trigger input | •A-D control register | (20) |
| $P8_0$/DA | Port P8 | Input/output,<br>individual bits | •CMOS compatible input level<br>•CMOS 3-state output | •D-A function output | •A-D control register | (21) |
| $P8_1$ | | | | | | (1) |
| $P8_2/S_{OUT3}$ | | | | •Serial I/O3 function I/O | •Serial I/O3 register/ Transfer counter<br>•Serial I/O3 control register 1<br>•Serial I/O3 control register 2<br>•Serial I/O3 control register 3<br>•Serial I/O3 automatic transfer data pointer | (22) |
| $P8_3/S_{IN3}$ | | | | | | (23) |
| $P8_4/S_{CLK3}$ | | | | | | (24) |
| $P8_5/S_{RDY3}$ | | | | | | (25) |
| $P8_6/S_{BUSY3}$ | | | | | | (26) |
| $P8_7/S_{STB3}$ | | | | | | (27) |
| $P9_7/INT_0$ | Port P9 | Input | •CMOS compatible input level | •External interrupt input | •Interrupt edge selection register | (28) |

**Note:** Make sure that the input level at each pin is either 0 V or Vcc during execution of the STP instruction.
When an input level is at an intermediate potential, a current will flow from Vcc to Vss through the input-stage gate.

## Pull-up Control

P2₀–P2₆, TxD, SCLK1, SOUT2, and SCLK2 can perform pull-up control by setting “1” to the pull-up register (address $0033_{16}$).
$P2_0$–$P2_7$'s pull-up is valid in the input mode, and TxD, SCLK1, SOUT2, and SCLK2s' pull-up is valid in the output mode.

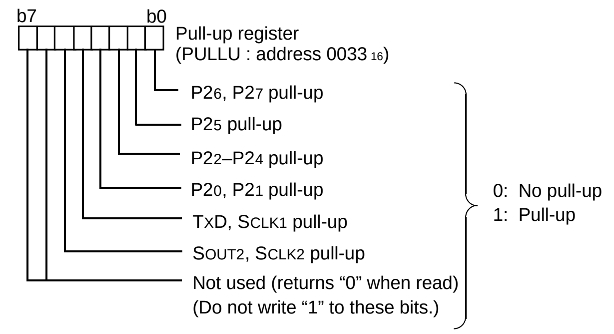



Fig.11 Structure of Pull-up Register

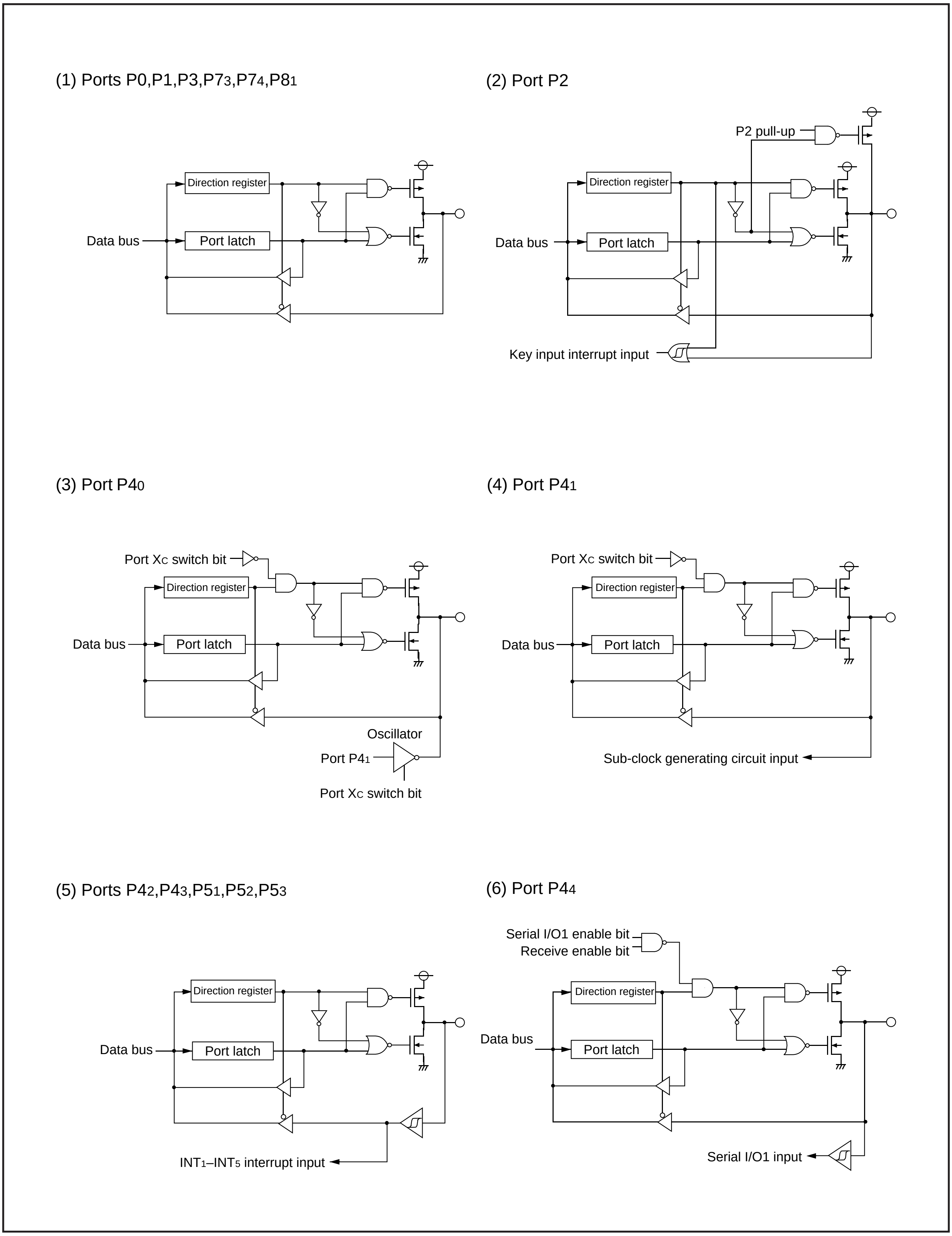
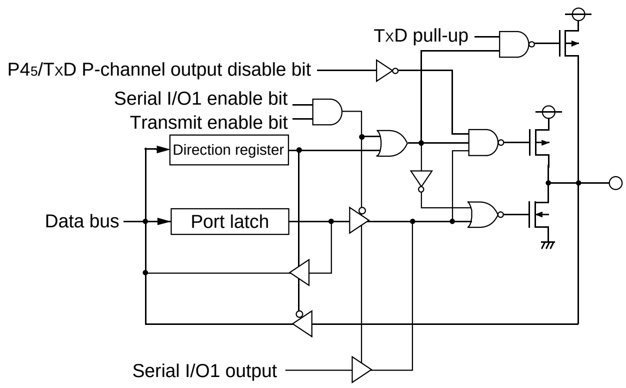
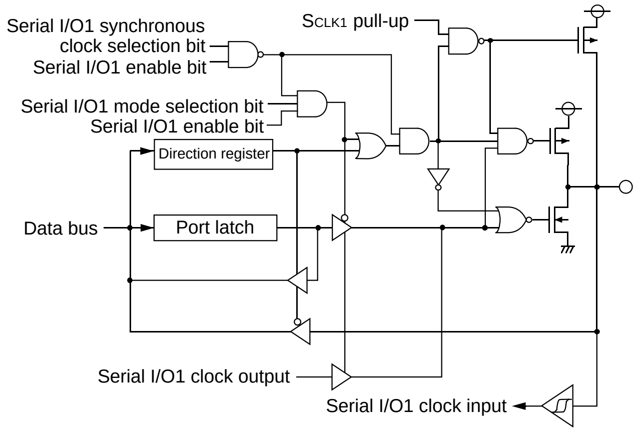



Fig. 12 Port block diagram (1)

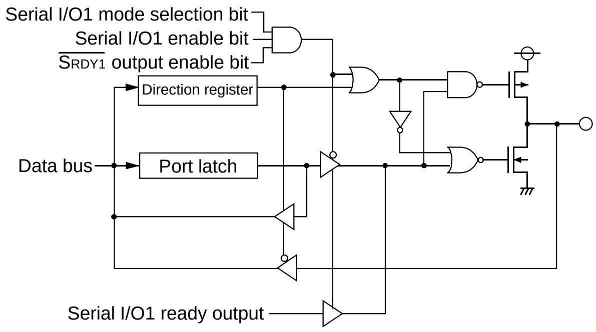
(7) Port $P4_5$



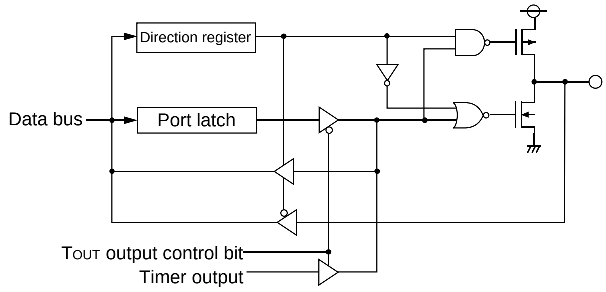

(8) Port $P4_6$



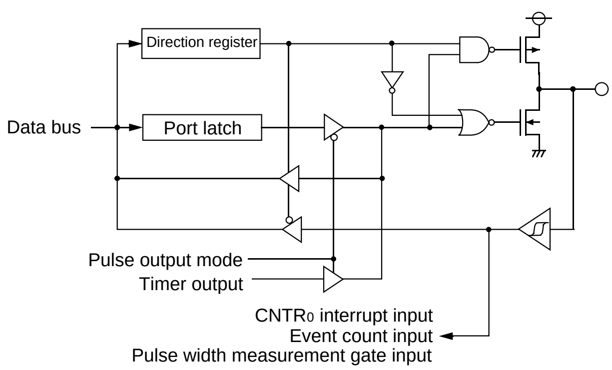

(9) Port $P4_7$




(10) Port $P5_0$




(11) Port $P5_4$




(12) Port $P5_5$

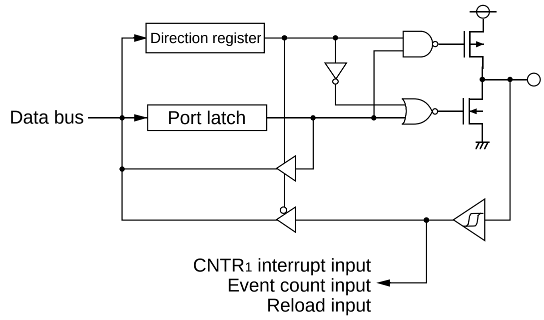
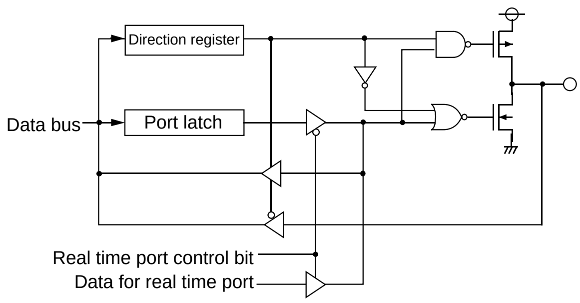
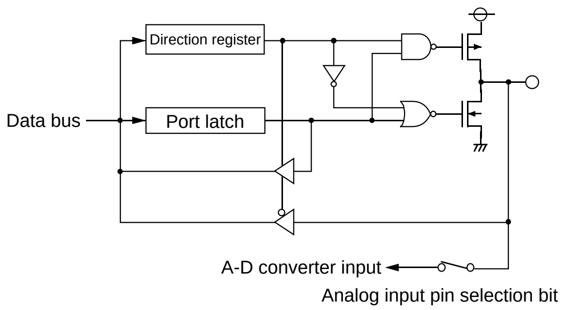



Fig. 13 Port block diagram (2)

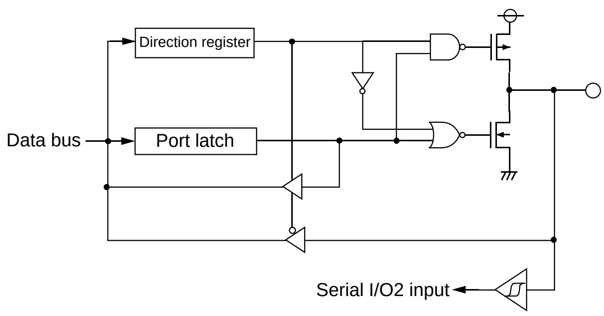
(13) Ports P5₆, P5₇




(14) Port P6




(15) Port P7₀




(16) Port P7₁

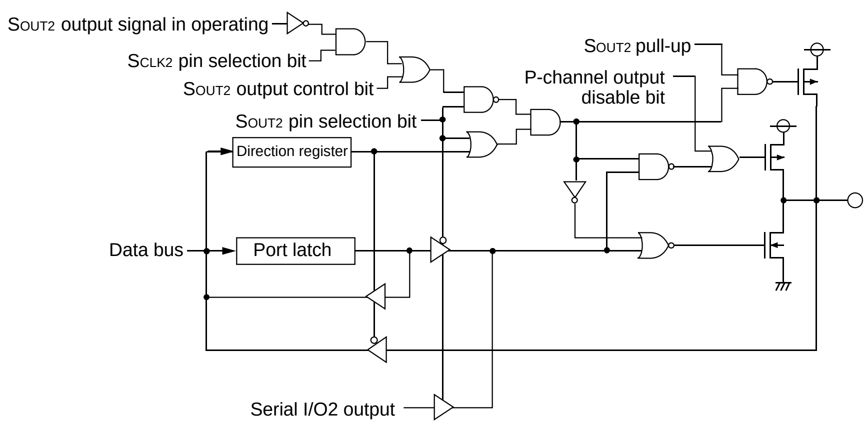
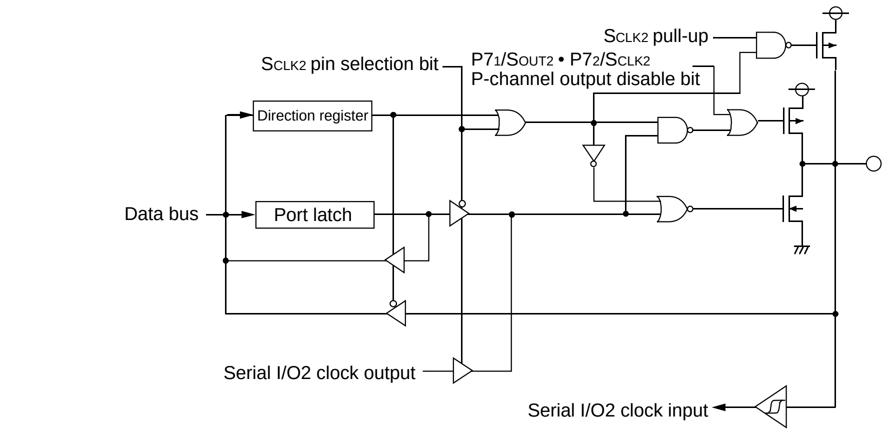
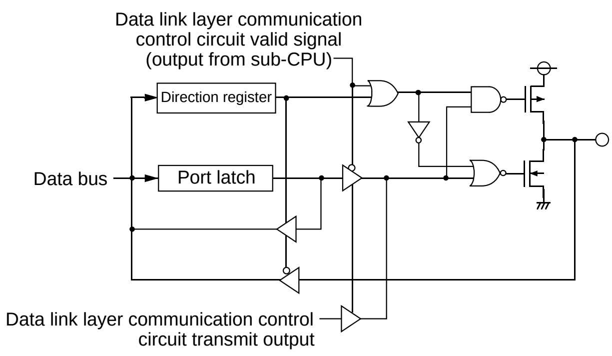



Fig. 14 Port block diagram (3)

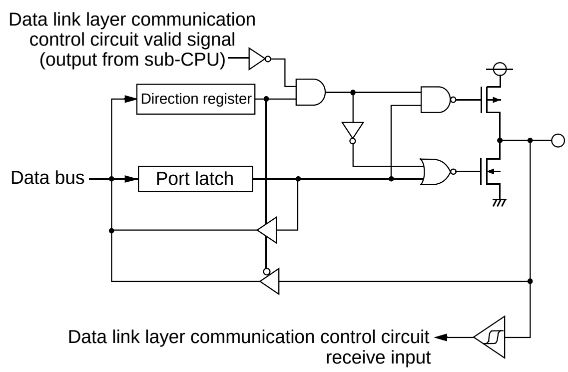
(17) Port $P7_2$



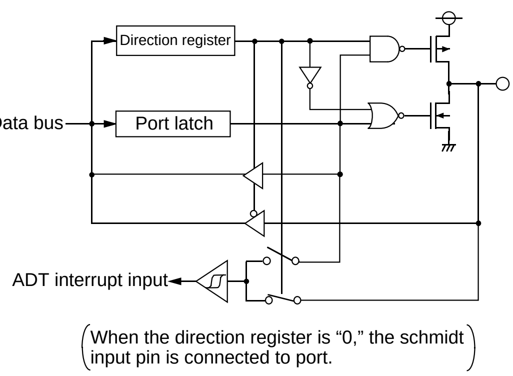

(18) Port $P7_5$




(19) Port $P7_6$




(20) Port $P7_7$




(21) Port $P8_0$

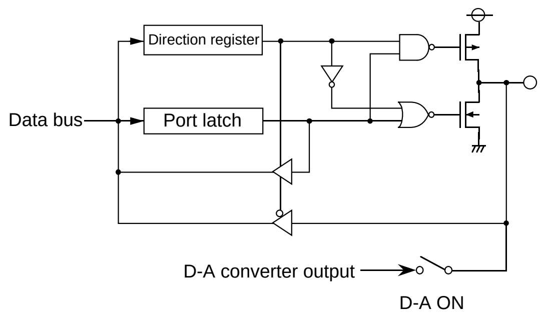



Fig. 15 Port block diagram (4)

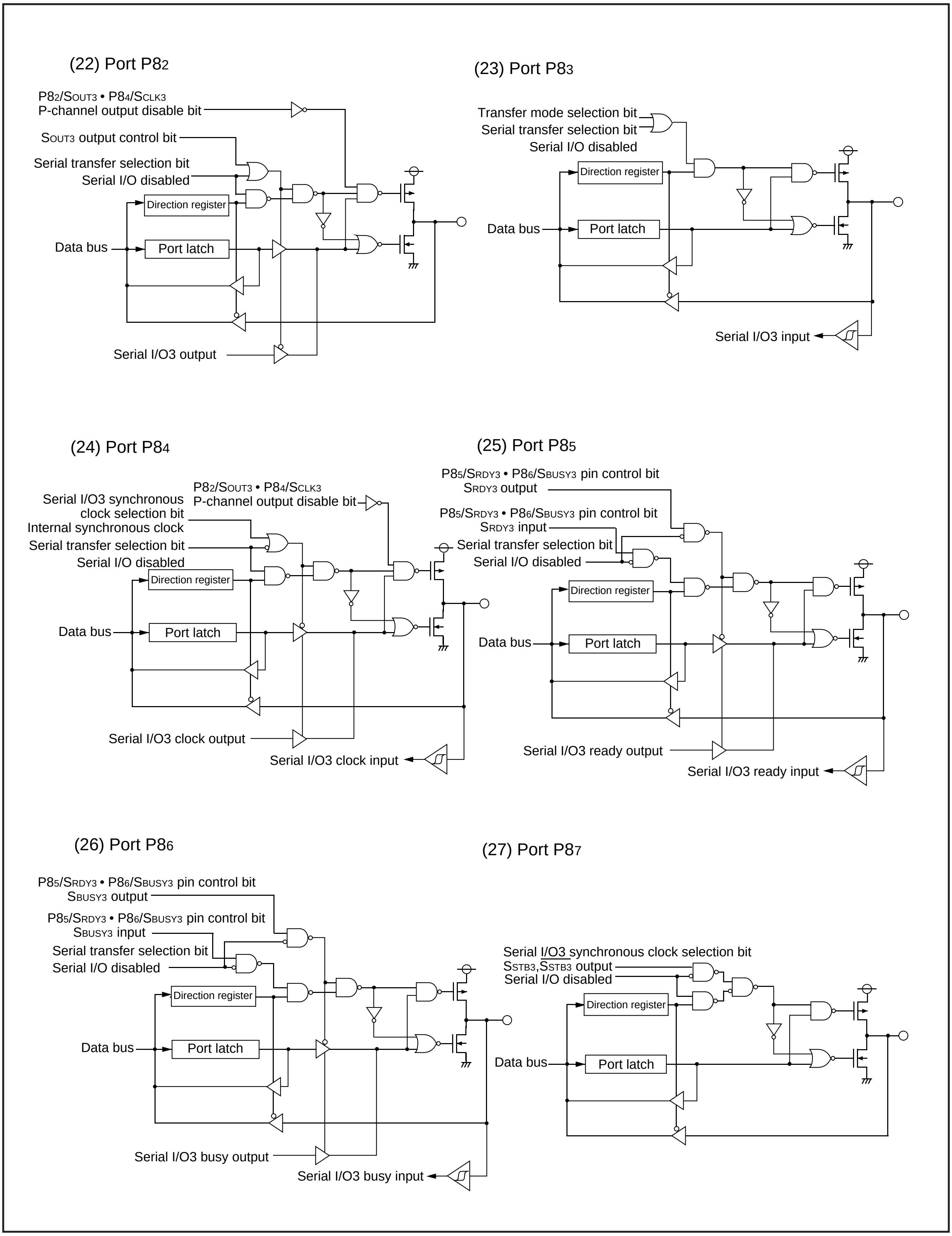



Fig. 16 Port block diagram (5)

(28) Port $P9_7$

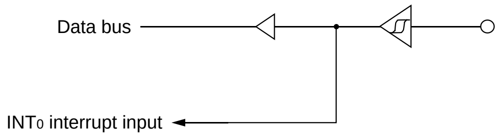



Fig. 17 Port block diagram (6)

## INTERRUPTS

Interrupts occur by 27 sources: 10 external, 16 internal, and 1 software.

### Interrupt Control

Each interrupt is controlled by an interrupt request bit, an interrupt enable bit, and the interrupt disable flag except for the software interrupt set by the BRK instruction. An interrupt occurs if the corresponding interrupt request and enable bits are “1” and the interrupt disable flag is “0”.
Interrupt enable bits can be set or cleared by software.
Interrupt request bits can be cleared by software, but cannot be set by software.
The BRK instruction cannot be disabled with any flag or bit. The I (interrupt disable) flag disables all interrupts except the BRK instruction interrupt.
The interrupt control circuit consists of two types of interrupts: “one factor/one vector interrupt” and “multiple factors/one vector interrupt”. The configuration is shown in Figure 18.

### Interrupt Operation

When an interrupt occurs, the following operations are automatically performed:

1. The contents of the program counter and the processor status register are pushed onto the stack.
2. The interrupt disable flag is set and the corresponding interrupt request bit for each vector is cleared. (The corresponding interrupt request bit for each interrupt factor is not cleared.)
3. The interrupt jump destination address of interrupt which has the highest priority is loaded to the program counter.

### Interrupt Factor Determination

The interrupt request bit for each vector of “multiple factors/one vector interrupt” is set to “1” when the interrupt disable flag (I) is “0” and one of the factor interrupt enable bits is “1” and the corresponding factor interrupt request bit changes from “0” to “1”. At this time, if the vector interrupt enable bit is “1”, the interrupt occurs. (Note that the interrupt request bit for each vector and the factor interrupt request bit are both edge sense.)
When 2 or more interrupt requests of interrupt factors assigned to one interrupt vector are generated at the same time, confirm the interrupt request bits for each interrupt factor assigned to the vector, and process according to the priority.
If the interrupt request bit for the interrupt factor is “1” and the interrupt enable bits for interrupt factor and each vector are both “1”; for example, when an interrupt of another interrupt factor assigned to the same vector occurs while an interrupt processing routine is executed, the interrupt occurs again after returning. Clear the interrupt request bits which are not necessary or which have been already processed before executing the interrupt flag clear (CLI) or interrupt processing routine return (RTI) instruction.
The interrupt request bits for each interrupt factor are not cleared by hardware after an interrupt vector address branching. Clear these bits by software in the interrupt processing routine. Use the LDM, STA, etc. instructions to do it. Do not use the read- modify-write instruction; for example, the CLB.

### ■ Notes

When the active edge of an external interrupt ($INT_0$–$INT_5$, $CNTR_0$, $CNTR_1$) is set, the corresponding interrupt request bit may also be set. Therefore, take following sequence:

(1) Disable the external interrupt which is selected.
(2) Change the active edge in interrupt edge selection register (in case of $CNTR_0$: Timer X mode register; in case of $CNTR_1$: Timer Y mode register).
(3) Clear the set interrupt request bit to “0”.
(4) Enable the external interrupt which is selected.

**Table 7 Interrupt vector addresses and priority**

| Interrupt Sources | Priority | Vector Addresses **(Note 1)** | | Interrupt Request Generating Conditions | Remarks |
|---|---|---|---|---|---|
| | | High | Low | | |
| Reset **(Note 2)** | 1 | $FFFD_{16}$ | $FFFC_{16}$ | At reset | Non-maskable |
| $INT_0$ | 2 | $FFFB_{16}$ | $FFFA_{16}$ | At detection of either rising or falling edge of $INT_0$ input | External interrupt (active edge selectable) |
| $INT_1$ | 3 | $FFF9_{16}$ | $FFF8_{16}$ | At detection of either rising or falling edge of $INT_1$ input | External interrupt (active edge selectable) |
| Receive bus interrupt source 1 | 4 | $FFF7_{16}$ | $FFF6_{16}$ | When receive bus interrupt source 1 request bit becomes “1” from “0” | The condition which the receive bus interrupt factor request bit becomes “1” is defined according to each communication protocol specification confirmation. |
| Receive bus interrupt source 2 | | | | When receive bus interrupt source 2 request bit becomes “1” from “0” | |
| Receive bus interrupt source 3 | | | | When receive bus interrupt source 3 request bit becomes “1” from “0” | |
| Transmit bus interrupt source 1 | 5 | $FFF5_{16}$ | $FFF4_{16}$ | When transmit bus interrupt source 1 request bit becomes “1” from “0” | The condition which the transmit bus interrupt factor request bit becomes “1” is defined according to each communication protocol specification confirmation. |
| Transmit bus interrupt source 2 | | | | When transmit bus interrupt source 2 request bit becomes “1” from “0” | |
| Transmit bus interrupt source 3 | | | | When transmit bus interrupt source 3 request bit becomes “1” from “0” | |
| Timer X | 6 | $FFF3_{16}$ | $FFF2_{16}$ | At timer X underflow | |
| Timer Y | 7 | $FFF1_{16}$ | $FFF0_{16}$ | At timer Y underflow | |
| Timer 2 | 8 | $FFEF_{16}$ | $FFEE_{16}$ | At timer 2 underflow | |
| Timer 3 | 9 | $FFED_{16}$ | $FFEC_{16}$ | At timer 3 underflow | |
| $INT_2$ | 10 | $FFEB_{16}$ | $FFEA_{16}$ | At detection of either rising or falling edge of $INT_2$ input | External interrupt (active edge selectable) |
| Serial I/O3 interrupt | 11 | $FFE9_{16}$ | $FFE8_{16}$ | At completion of serial I/O3 data transmission/reception | Valid only when serial I/O3 is selected |
| $CNTR_0$ | | | | At detection of either rising or falling edge of $CNTR_0$ input | External interrupt (active edge selectable) |
| $CNTR_1$ | 12 | $FFE7_{16}$ | $FFE6_{16}$ | At detection of either rising or falling edge of $CNTR_1$ input | External interrupt (active edge selectable) |
| Timer 1 | 13 | $FFE5_{16}$ | $FFE4_{16}$ | At timer 1 underflow | |
| $INT_3$ | 14 | $FFE3_{16}$ | $FFE2_{16}$ | At detection of either rising or falling edge of $INT_3$ input | External interrupt (active edge selectable) |
| $INT_4$ | | | | At detection of either rising or falling edge of $INT_4$ input | External interrupt (active edge selectable) |
| $INT_5$ | | | | At detection of either rising or falling edge of $INT_5$ input | External interrupt (active edge selectable) |
| ADT | 15 | $FFE1_{16}$ | $FFE0_{16}$ | At falling of ADT pin input | Valid only when ADT interrupt is selected External interrupt (falling valid) |
| A-D converter | | | | At completion of A-D converter | Valid only when A-D converter interrupt is selected |
| Serial I/O2 interrupt | | | | At completion of serial I/O2 data transmission/reception | Valid only when serial I/O2 is selected |
| Key input (key-on wake-up) | 16 | $FFDF_{16}$ | $FFDE_{16}$ | At falling of port $P2_0$ to $P2_5$ (at input) input logical level AND | External interrupt (falling valid) |
| Serial I/O1 receive | | | | At completion of serial I/O1 data reception | Valid only when serial I/O1 is selected |
| Serial I/O1 transmit | | | | At completion of serial I/O1 transmission shift or when transmission buffer is empty | Valid only when serial I/O1 is selected |
| BRK instruction | 17 | $FFDD_{16}$ | $FFDC_{16}$ | At BRK instruction execution | Non-maskable software interrupt |

**Notes 1:** Vector addresses contain interrupt jump destination addresses.
**2:** Reset function in the same way as an interrupt with the highest priority.
**3:** Either ADT interrupt or A-D converter interrupt can be used. Both ADT interrupt and A-D converter interrupt cannot be used.

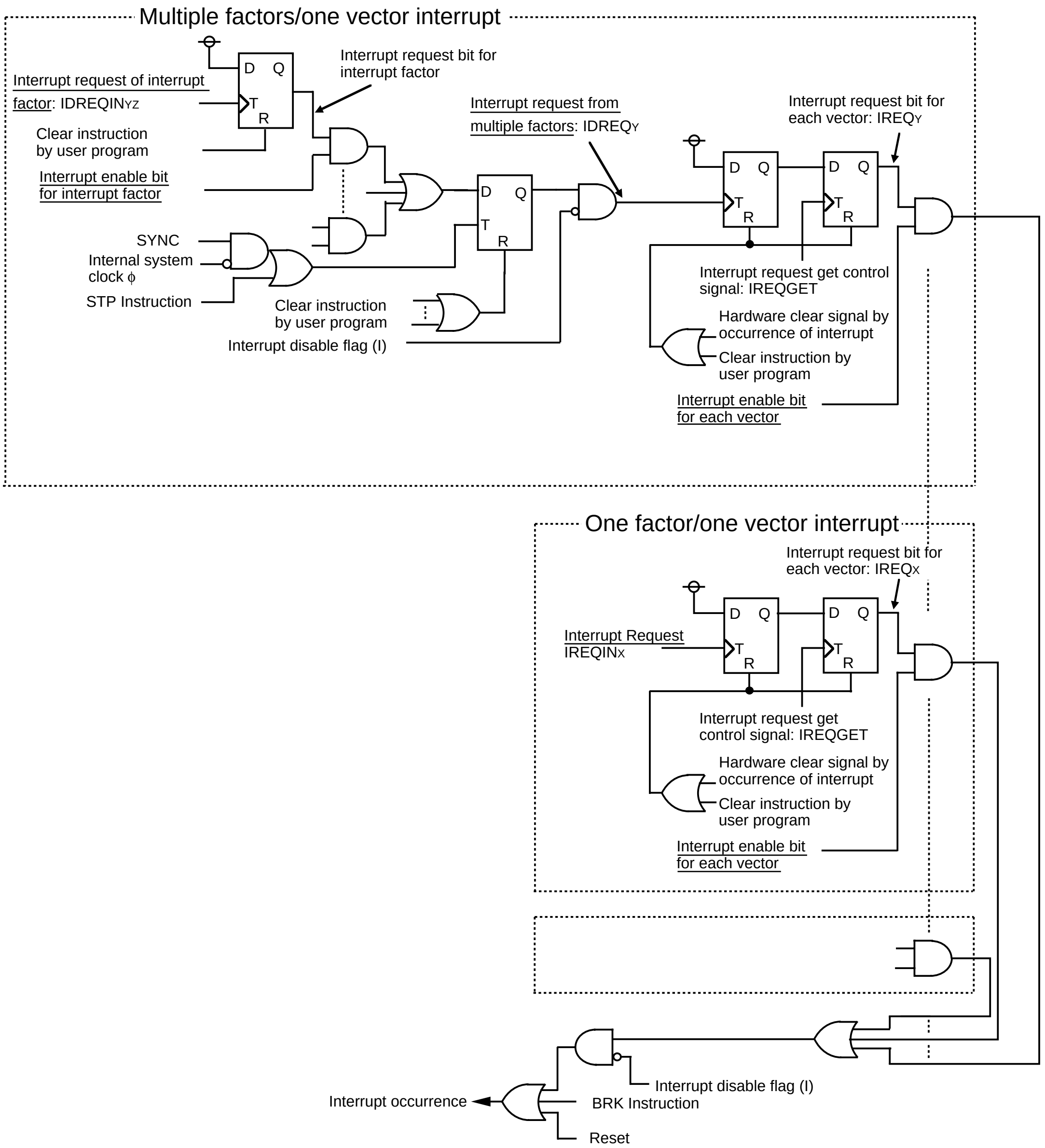



Fig.18 Interrupt control diagram

## Timing to Interrupt Request Acceptance

The cycle number of internal system clock required from occurrence to acceptance of an interrupt request depends on the type of interrupt: "multiple factors/one vector" or "one factor/one vector".

For "one factor/one vector interrupt", the CPU starts processing the management after interrupt acceptance at the next instruction execution timing (rising edge of SYNC signal) immediately after the interrupt request is generated. For "multiple factors/one vector interrupt", the CPU starts processing the management after interrupt acceptance at the second instruction execution timing (rising edge of SYNC signal) after the interrupt request for interrupt factor determination is generated. In other words, "multiple factors/one vector interrupt" required one instruction execution cycle number (2 to 16 cycles of internal system clock) more than that of "one factor/one vector interrupt" to begin the interrupt sequence.

Figure 18 shows the interrupt control diagram and Figure 19 shows the timing from occurrence to acceptance of interrupt request.

For "one factor/one vector interrupt", the interrupt request is generated at Timing (A) and the processing after acceptance begins at Timing (B). For "multiple factors/one vector interrupt", the interrupt factor determination request is generated at Timing (C), the interrupt request is generated at Timing (D), and the processing after acceptance begins at Timing (E).

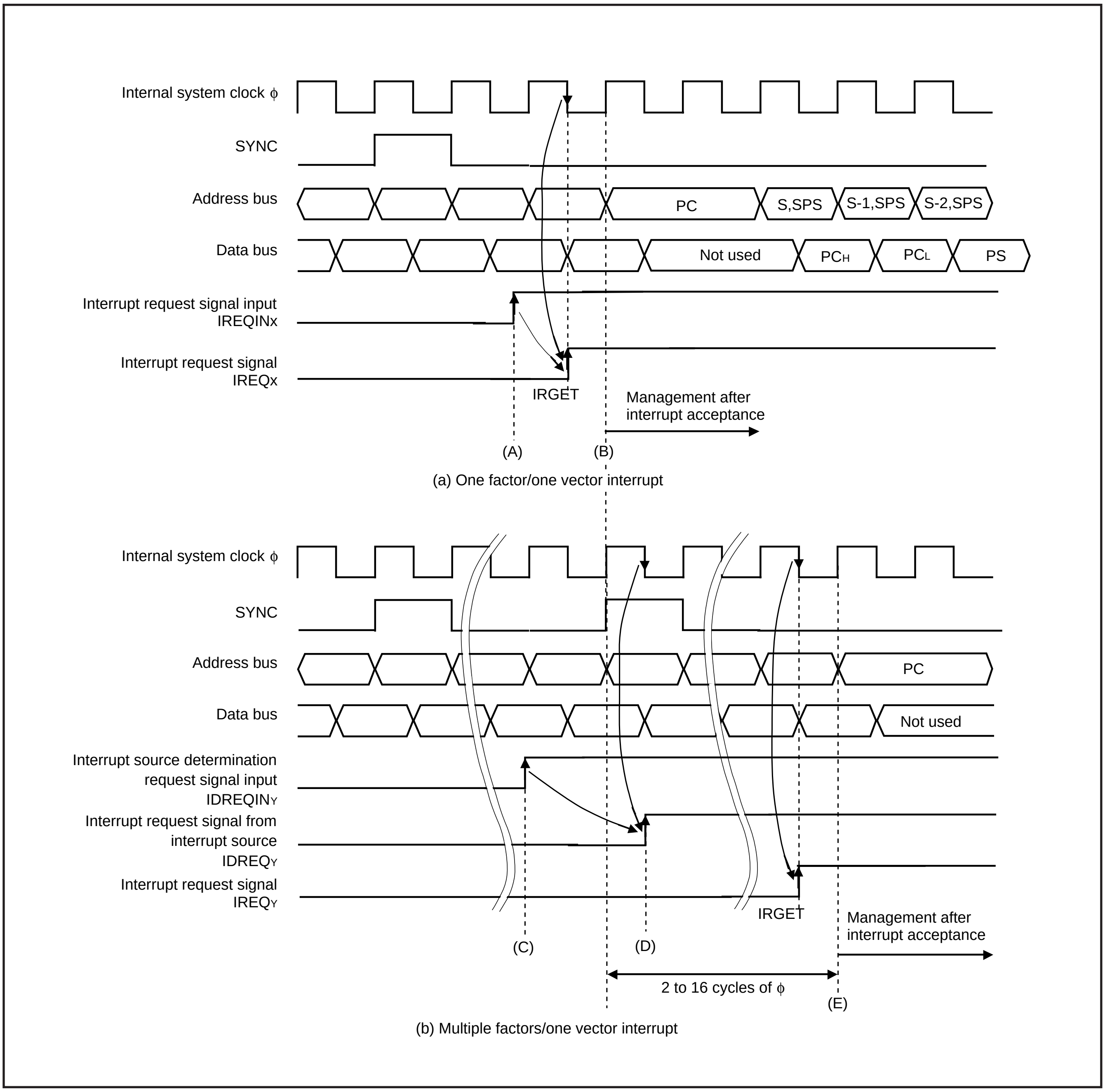



Fig.19 Timing from occurrence to acceptance of interrupt

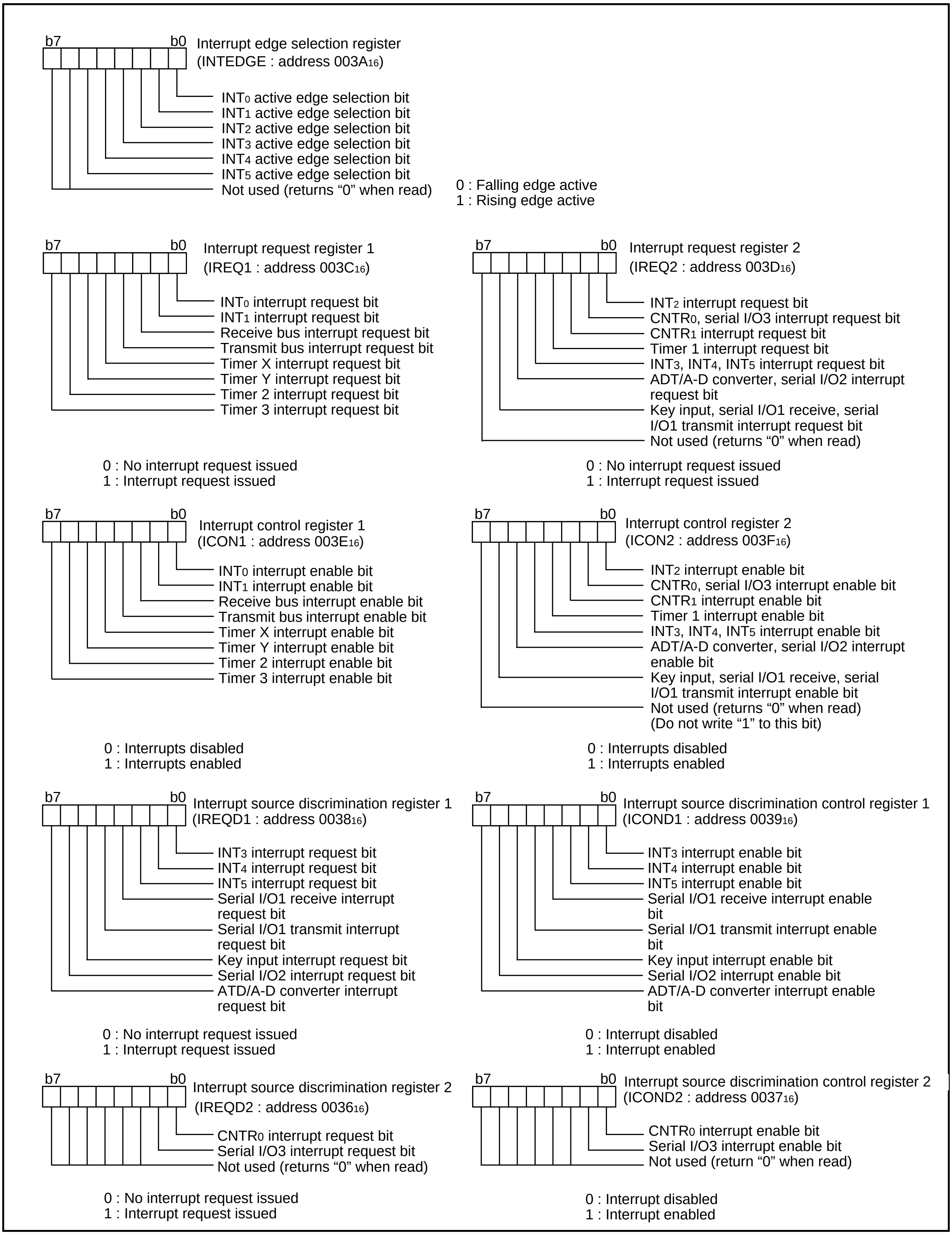



Fig. 20 Structure of interrupt-related registers

## Key Input Interrupt

A Key input interrupt request is generated by applying “L” level to any pin of port P2 that have been set to input mode. In other words, it is generated when AND of input level goes from “1” to “0”.

An example of using a key input interrupt is shown in Figure 21, where an interrupt request is generated by pressing one of the keys consisted as an active-low key matrix which inputs to ports $P2_0$–$P2_4$.

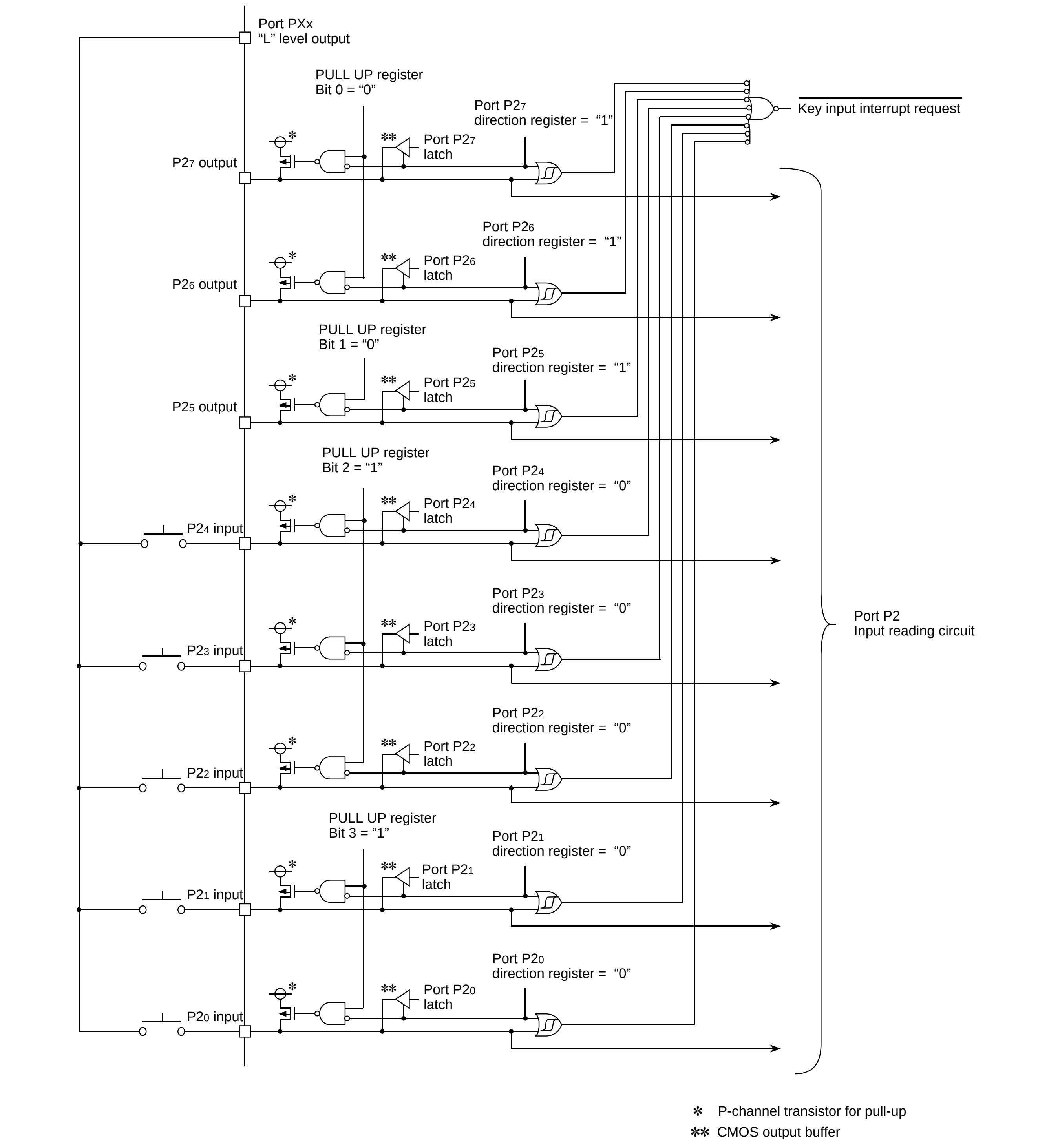



Fig. 21 Connection example when using key input interrupt and port P2 block diagram

## TIMERS

The 3874 group has five timers: timer X, timer Y, timer 1, timer 2, and timer 3. Timer X and timer Y are 16-bit timers, and timer 1, timer 2, and timer 3 are 8-bit timers.

All timers are down count timers. When the timer reaches "$00_{16}$" or "$0000_{16}$", an underflow occurs at the next count pulse and the corresponding timer latch is reloaded into the timer and the count is continued. When a timer underflows, the interrupt request bit corresponding to that timer is set to "1".

Read and write operation on 16-bit timer must be performed for both high and low-order bytes. When reading a 16-bit timer, read the high-order byte first. When writing to a 16-bit timer, write the low-order byte first. The 16-bit timer cannot perform the correct operation when reading during the write operation, or when writing during the read operation.

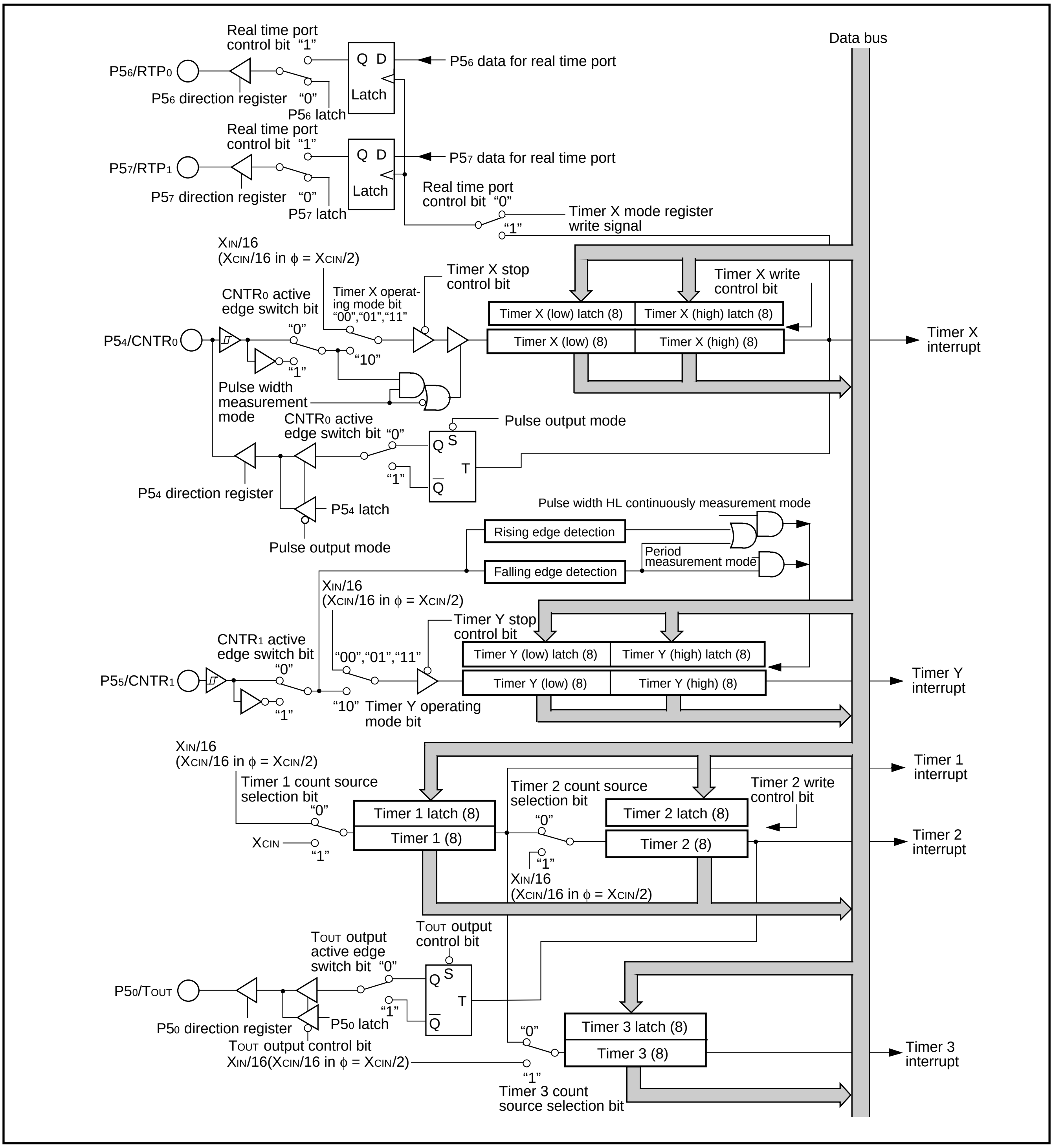



Fig. 22 Timer block diagram

## Timer X

Timer X is a 16-bit timer that can be selected in one of four modes and can be controlled the timer X write and the real time port by setting the timer X mode register.

### (1) Timer Mode

The timer counts $f(X_{IN})/16$ (or $f(X_{CIN})/16$ in system clock $\phi = X_{CIN}/2$).

### (2) Pulse Output Mode

Each time the timer underflows, a signal output from the $CNTR_0$ pin is inverted. Except for this, the operation in pulse output mode is the same as in timer mode. When using a timer in this mode, set the direction register of corresponding port to output mode.

### (3) Event Counter Mode

The timer counts signals input through the $CNTR_0$ pin.
Except for this, the operation in event counter mode is the same as in timer mode.

### (4) Pulse Width Measurement Mode

The count source is $f(X_{IN})/16$ (or $f(X_{CIN})/16$ in system clock $\phi = X_{CIN}/2$. If $CNTR_0$ active edge switch bit is “0”, the timer counts while the input signal of $CNTR_0$ pin is at “H”. If it is “1”, the timer counts while the input signal of $CNTR_0$ pin is at “L”.

## ■ Notes

**● Timer X write control**

If the timer X write control bit is “1”, when the value is written in the address of timer X, the value is loaded only in the latch. The value in the latch is loaded in timer X after timer X underflows.
If the timer X write control bit is “0”, when the value is written in the address of timer X, the value is loaded in the timer X and the latch at the same time.
When the value is to be written in latch only, if the value is written to the latch at timer X underflows, the value is consequently loaded in the timer X and the latch at the same time. Unexpected value may be set in the high-order counter when the writing in high-order latch and the underflow of timer X are performed at the same timing.

**● $CNTR_0$ interrupt active edge selection**

$CNTR_0$ interrupt active edge depends on the $CNTR_0$ active edge switch bit.

**● Real time port control**

Data for the real time port are output from ports $P5_6$ and $P5_7$ each time the timer X underflows. (However, if the real time port control bit is changed from “0” to “1”, data are output independent of the timer X operation.) When the data for the real time port is changed while the real time port function is valid, the changed data are output at the next underflow of timer X.
Before using this function, set the corresponding port direction registers to output mode.

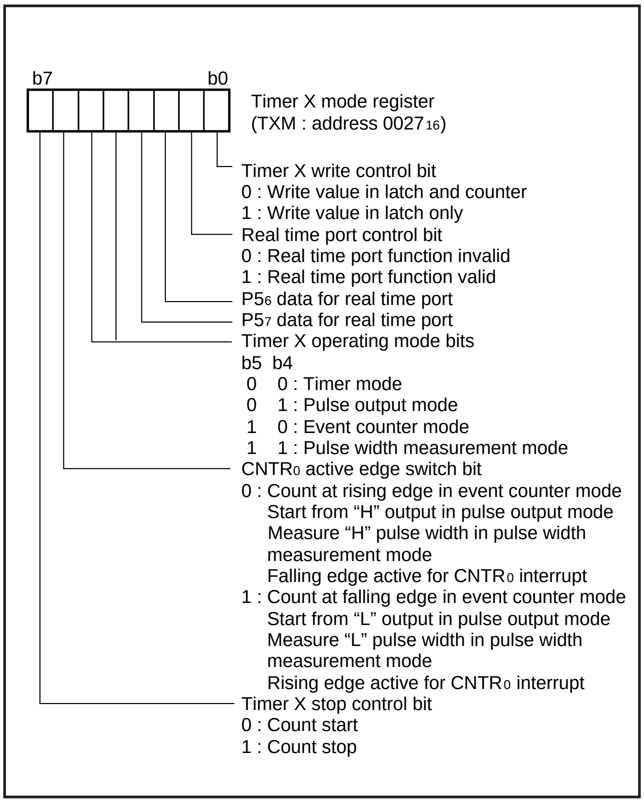



Fig. 23 Structure of timer X mode register

## Timer Y

Timer Y is a 16-bit timer that can be selected in one of four modes.

### (1) Timer Mode

The timer counts f($X_{IN}$)/16 (or f($X_{CIN}$)/16 in system clock $\phi = X_{CIN}/2$).

### (2) Period Measurement Mode

$CNTR_1$ interrupt request is generated at rising/falling edge of $CNTR_1$ pin input signal. Simultaneously, the value in timer Y latch is reloaded in timer Y and timer Y continues counting down. Except for the above-mentioned, the operation in period measurement mode is the same as in timer mode.
The timer value just before the reloading at rising/falling of $CNTR_1$ pin input signal is retained until the timer Y is read once after the reload.
The rising/falling timing of $CNTR_1$ pin input signal is found by $CNTR_1$ interrupt.

### (3) Event Counter Mode

The timer counts signals input through the $CNTR_1$ pin.
Except for this, the operation in event counter mode is the same as in timer mode.

### (4) Pulse Width HL Continuously Measurement Mode

$CNTR_1$ interrupt request is generated at both rising and falling edges of $CNTR_1$ pin input signal. Except for this, the operation in pulse width HL continuously measurement mode is the same as in period measurement mode.

## ■ Note

**● $CNTR_1$ interrupt active edge selection**

$CNTR_1$ interrupt active edge depends on the $CNTR_1$ active edge switch bit. However, in pulse width HL continuously measurement mode, $CNTR_1$ interrupt request is generated at both rising and falling edges of $CNTR_1$ pin input signal regardless of the setting of $CNTR_1$ active edge switch bit.

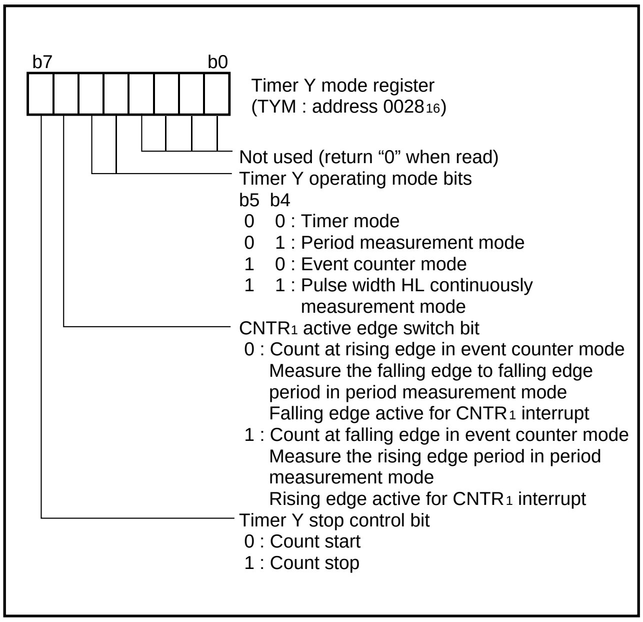



Fig. 24 Structure of timer Y mode register

## Timer 1, Timer 2, Timer 3

Timer 1, timer 2, and timer 3 are 8-bit timers. The count source for each timer can be selected by timer 123 mode register.

**● Timer 2 write control**

When the timer 2 write control bit is "1", and the value is written in the address of timer 2, the value is loaded only in the latch. The value in the latch is loaded in timer 2 after timer 2 underflows.
When the timer 2 write control bit is "0", and the value is written in the address of timer 2, the value is loaded in the timer 2 and the latch at the same time.

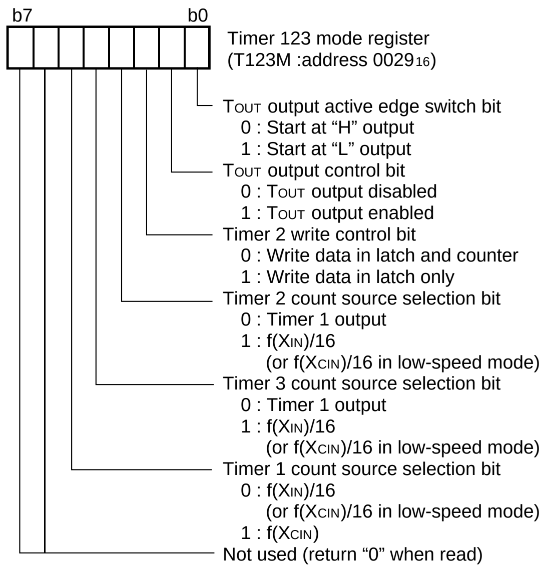
**● Timer 2 output control**

An inversion signal from TOUT pin is output each time timer 2 underflows.
In this case, set the port $P5_0$ direction register to the output mode.

## ■ Note

**● Timer 1 to timer 3**

When the count source of timer 1 to 3 is changed, the timer counting value may be changed large because a thin pulse is generated in count input of timer. If timer 1 output is selected as the count source of timer 2 or timer 3, when timer 1 is written, the counting value of timer 2 or timer 3 may be changed large because a thin pulse is generated in timer 1 output.
Therefore, set the value of timer in the order of timer 1, timer 2 and timer 3 after the count source selection of timer 1 to 3.




**Note :** Internal clock φ is $f(X_{CIN})/2$ in the low-speed mode.

Fig. 25 Structure of timer 123 mode register

## SERIAL I/O
### Serial I/O1

Serial I/O can be used as either clock synchronous or asynchronous (UART) serial I/O. A dedicated timer (baud rate generator) is also provided for baud rate generation.

### (1) Clock Synchronous Serial I/O1 Mode

Clock synchronous serial I/O1 mode can be selected by setting the serial I/O1 mode selection bit (b6) of the serial I/O1 control register to “1”.

For clock synchronous serial I/O1, the transmitter and the receiver must use the same clock. If an internal clock is used, transfer is started by a write signal to the transmit/receive buffer register (address $0018_{16}$).

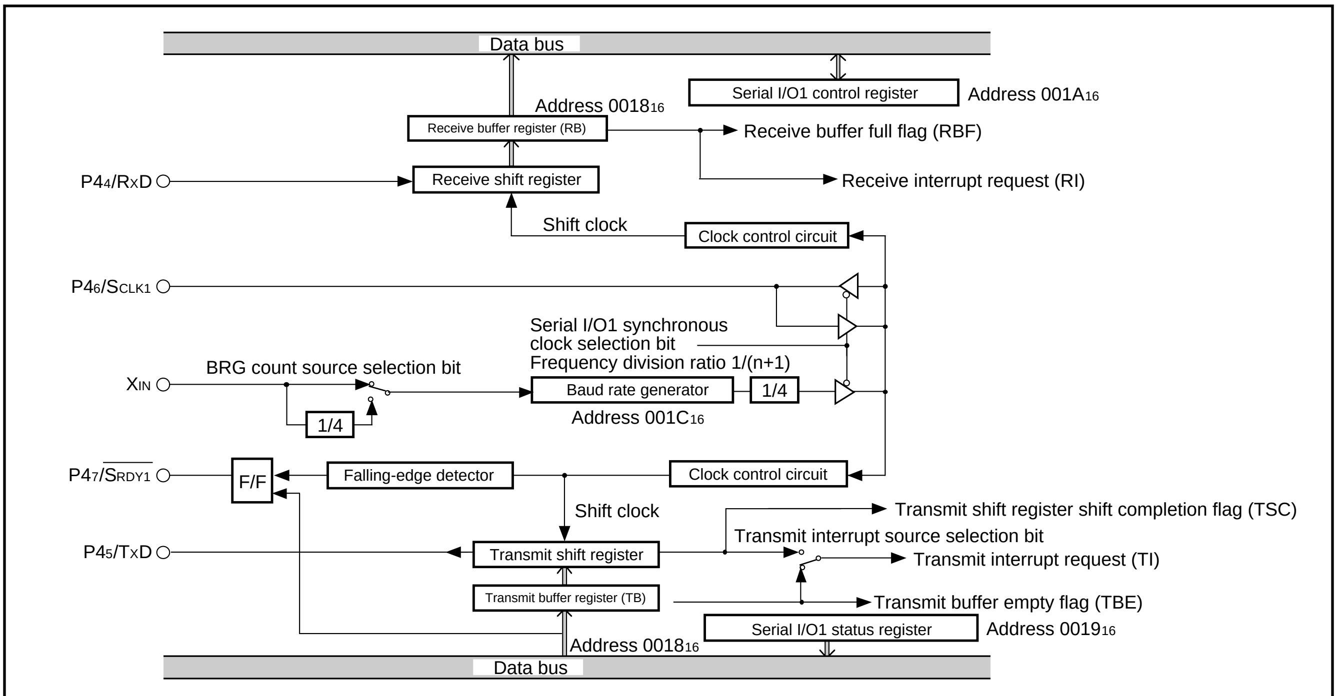
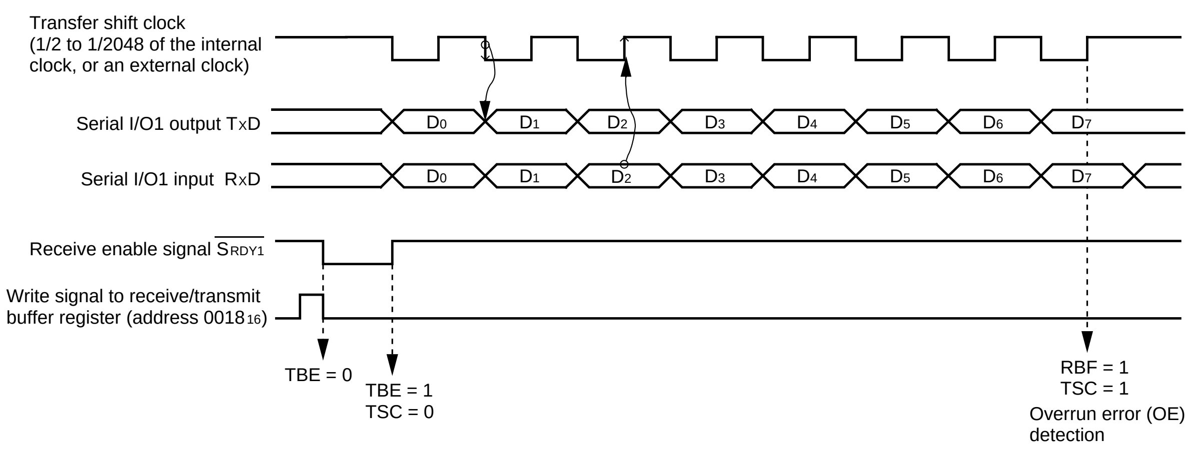



Fig. 26 Block diagram of clock synchronous serial I/O




**Notes 1 :** The transmit interrupt (TI) can be selected to occur either when the transmit buffer register has emptied (TBE=1) or after the transmit shift operation has ended (TSC=1), by setting the transmit interrupt source selection bit (TIC) of the serial I/O1 control register.
**2 :** If data is written to the transmit buffer register when TSC=0, the transmit clock is generated continuously and serial data is output continuously from the TXD pin.
**3 :** The receive interrupt (RI) is set when the receive buffer full flag (RBF) becomes “1” .

Fig. 27 Operation of clock synchronous serial I/O1 function

## (2) Asynchronous Serial I/O (UART) Mode

Clock asynchronous serial I/O mode (UART) can be selected by clearing the serial I/O1 mode selection bit (b6) of the serial I/O1 control register to “0”.

Eight serial data transfer formats can be selected, and the transfer formats used by a transmitter and receiver must be identical.

The transmit and receive shift registers each have a buffer register, but the two buffers have the same address in memory. Since the shift register cannot be written to or read from directly, transmit data is written to the transmit buffer, and receive data is read from the receive buffer.

The transmit buffer can also hold the next data to be transmitted, and the receive buffer register can hold a character while the next character is being received.

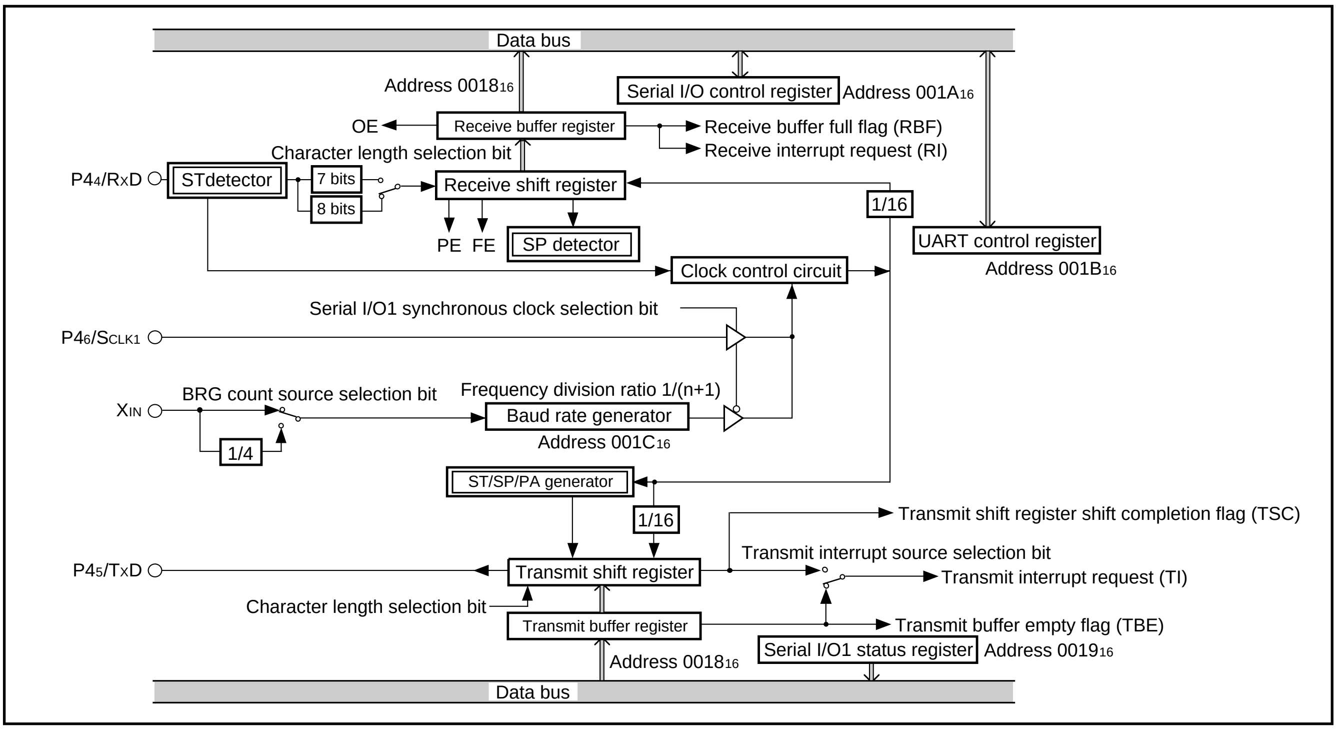
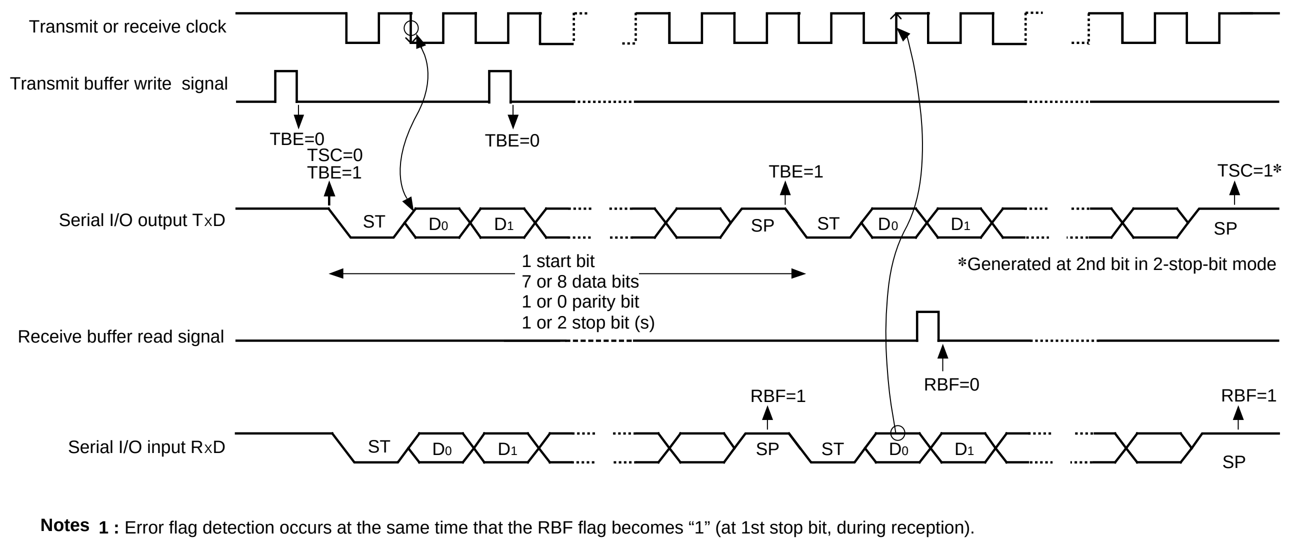



Fig. 28 Block diagram of UART serial I/O1




**Notes**
1 : Error flag detection occurs at the same time that the RBF flag becomes “1” (at 1st stop bit, during reception).
2 : The transmit interrupt (TI) can be selected to occur when either the TBE or TSC flag becomes “1”, depending on the setting of the transmit interrupt source selection bit (TIC) of the serial I/O1 control register.
3 : The receive interrupt (RI) is set when the RBF flag becomes “1”.
4 : After data is written to the transmit buffer register when TSC=1, 0.5 to 1.5 cycles of the data shift cycle is necessary until changing to TSC=0.

Fig. 29 Operation of UART serial I/O function

### [Transmit Buffer/Receive Buffer Register (TB/RB)] $0018_{16}$

The transmit buffer register and the receive buffer register are located at the same address. The transmit buffer register is write-only and the receive buffer register is read-only. If a character bit length is 7 bits, the MSB of data stored in the receive buffer register is “0”.

### [Serial I/O1 Status Register (SIO1STS)] $0019_{16}$

The read-only serial I/O1 status register consists of seven flags (bits 0 to 6) which indicate the operating status of the serial I/O function and various errors.
Three of the flags (bits 4 to 6) are valid only in UART mode.
The receive buffer full flag (bit 1) is cleared to “0” when the receive buffer is read.
If there is an error, it is detected at the same time that data is transferred from the receive shift register to the receive buffer register, and the receive buffer full flag is set. A write to the serial I/O1 status register clears all the error flags OE, PE, FE, and SE (bit 3 to bit 6, respectively). Writing “0” to the serial I/O1 enable bit (bit 7) of the Serial I/O1 control register also clears all the status flags, including the error flags.
All bits of the serial I/O1 status register are initialized to “0” at reset, but if the transmit enable bit (bit 4) of the serial I/O1 control register has been set to “1”, the transmit shift register shift completion flag (bit 2) and the transmit buffer empty flag (bit 0) become “1”.

### [Serial I/O1 Control Register (SIO1CON)] $001A_{16}$

The serial I/O1 control register contains eight control bits for the serial I/O1 function.

### [UART Control Register (UARTCON)] $001B_{16}$

The UART control register consists of four control bits (bits 0 to 3) which are valid when asynchronous serial I/O is selected and set the data format of a data transfer. One bit in this register (bit 4) is always valid and sets the output structure of the $P4_5/T_XD$ pin and $P4_6/S_{CLK1}$ pin.

### [Baud Rate Generator (BRG)] $001C_{16}$

The baud rate generator determines the baud rate for serial transfer.
The baud rate generator divides the frequency of the count source by 1/(n + 1), where n is the value written to the baud rate generator.

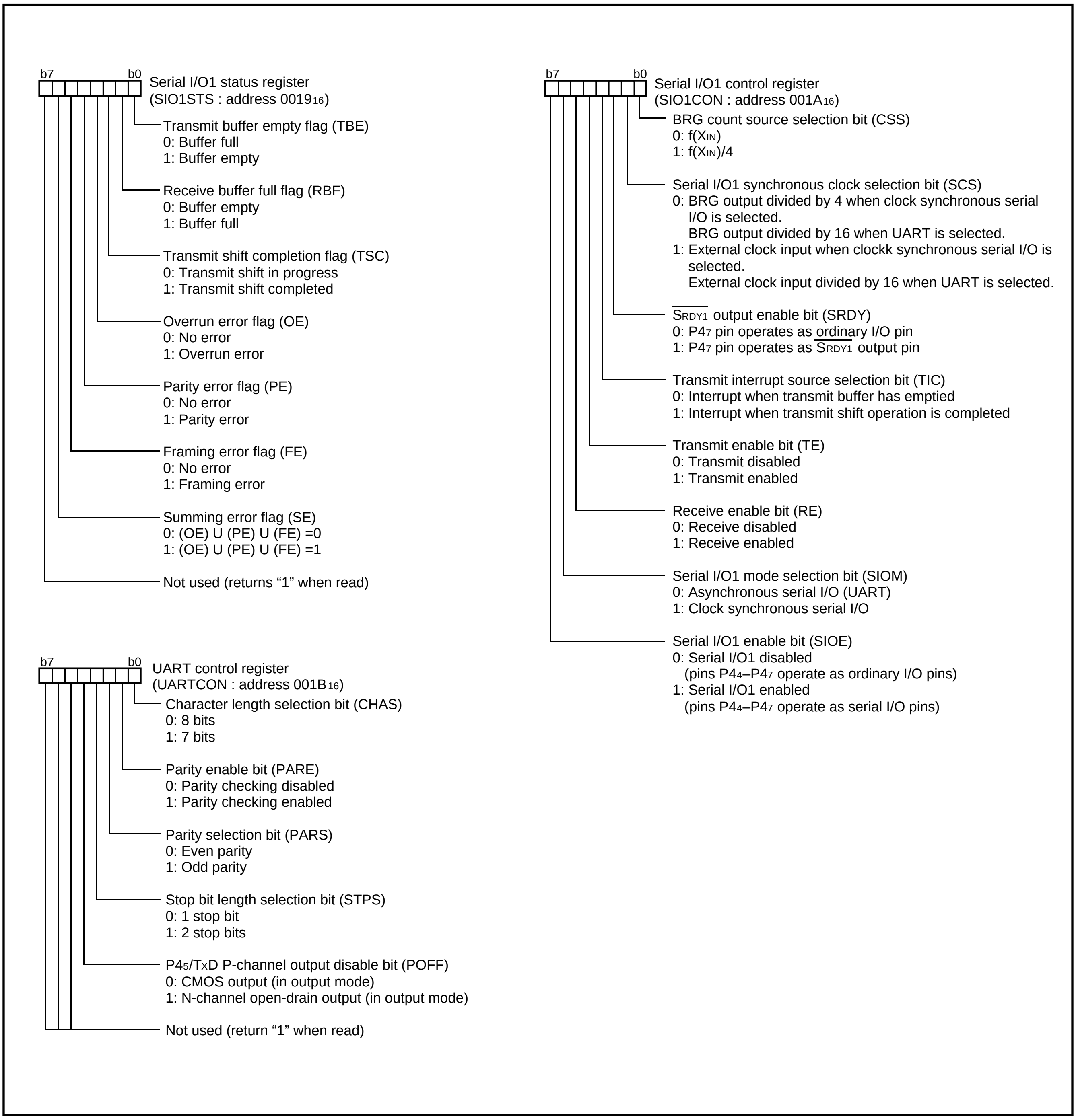



Fig. 30 Structure of serial I/O1 control register

## Serial I/O2

The Serial I/O2 function can be used only for clock synchronous serial I/O.

For clock synchronous serial I/O2, the transmitter and the receiver must use the same clock. When the internal clock is used, transfer is started by a write signal to the serial I/O2 register.

### [Serial I/O2 Control Register (SIO2CON)] 001D16

The serial I/O2 control register contains 8 bits which control various serial I/O functions.

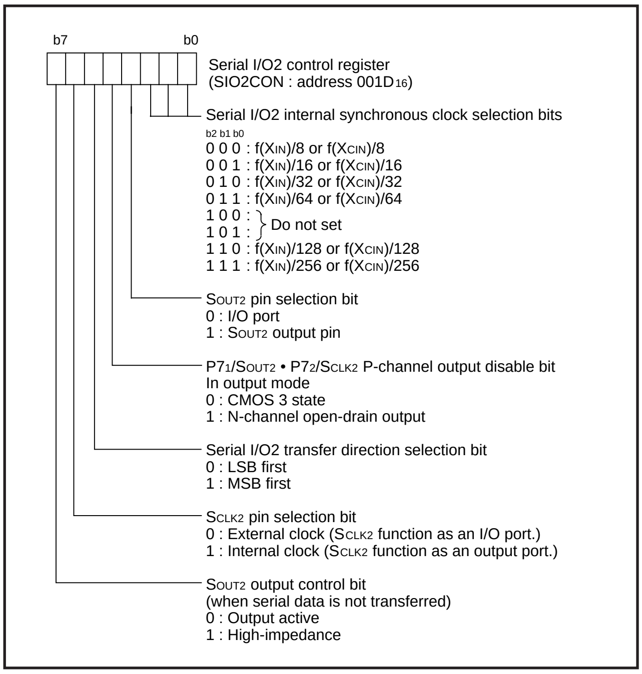



Fig. 31 Structure of serial I/O2 control register

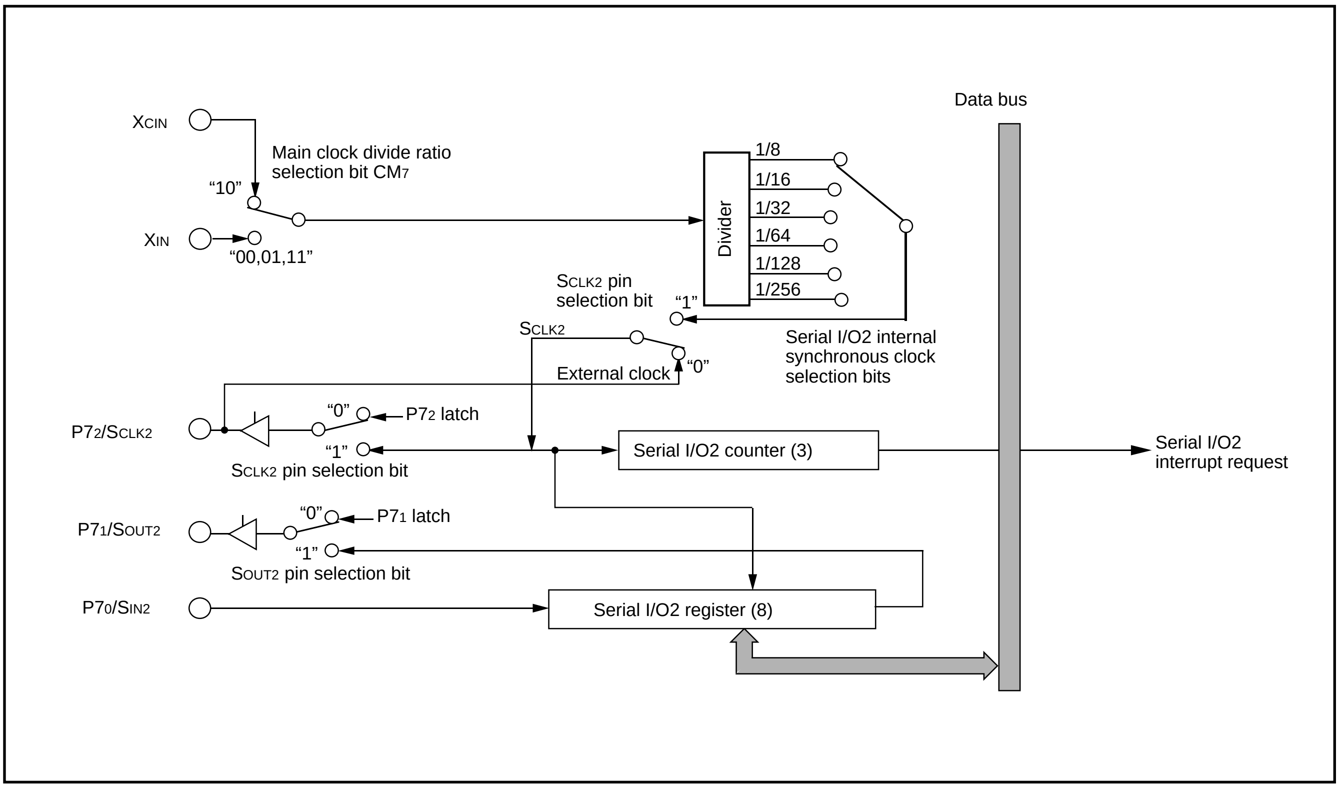



Fig. 32 Block diagram of serial I/O2

**●Serial I/O2 Operation**

When writing to the serial I/O2 register (001F16), the serial I/O2 counter is set to “7”.

After the write is completed, data is output from the SOUT2 pin each time the transfer clock goes from “H” to “L”. In addition, each time the transfer clock goes from “L” to “H”, the contents of the serial I/O2 register are shifted by 1 bit data is simultaneously received from the SIN2 pin.

When selecting an internal clock as the transfer clock source, the serial I/O2 counter goes to “0” by counting the transfer clock 8 times, and the transfer clock stops at “H”, and the interrupt request bit is set to “1”. In addition, the SOUT2 pin becomes the high-impedance state after the completion of data transfer. (Bit 7 of the serial I/O2 control register does not go to “1” and only the SOUT2 pin becomes the high-impedance state.)

When selecting an external clock as the transfer clock source, the interrupt request bit is set when counting the transfer clock 8 times. However, the transfer clock does not stop, so that control the clock externally. The SOUT2 pin does not become the high-impedance state after completion of data transmit.

In order to set the SOUT2 pin to the high-impedance state when selecting an external clock, set “1” to bit 7 of the serial I/O2 control register after completion of data transmit. Also, make sure that SCLK2 is at “H” for this process. When the next data is transmitted (falling of transfer clock), bit 7 of the serial I/O2 control register goes to “0” and the SOUT2 pin goes to an active state.

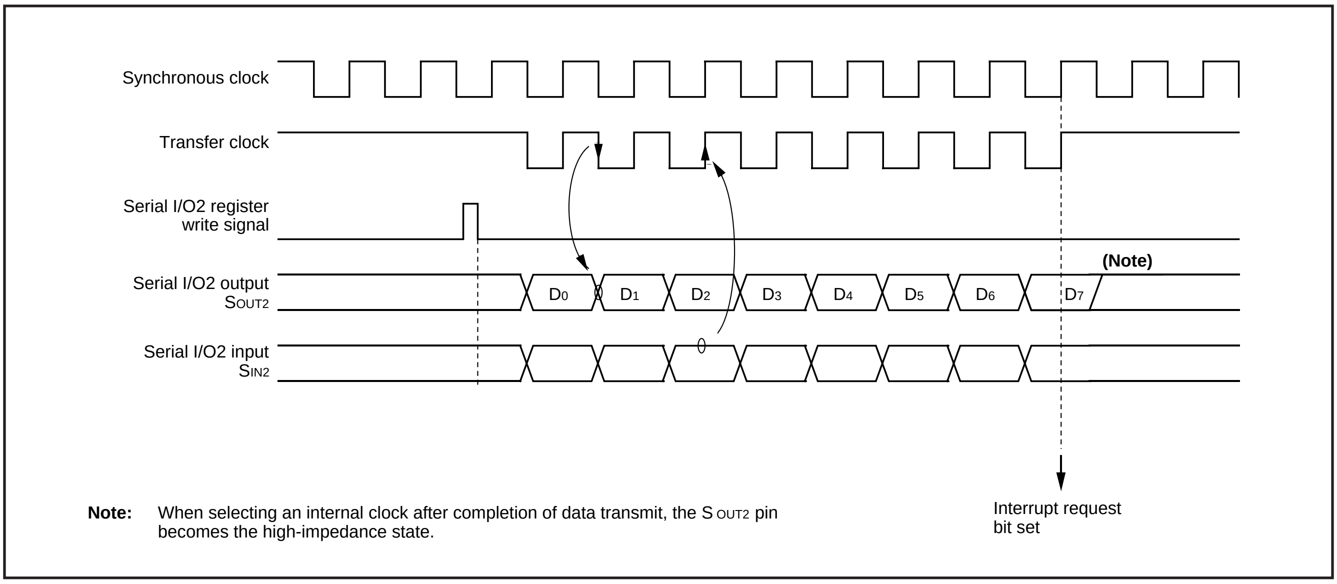



Fig. 33 Serial I/O2 timing (LSB first)

## Serial I/O3

Serial I/O3 has the following modes: 8-bit serial I/O, arbitrary bits from 1 to 256 serial I/O, up to 256-byte auto-transfer serial I/O. The 8-bit serial I/O transfers through serial I/O3 register (address $0013_{16}$). The arbitrary bits and auto-transfer serial I/O modes transfer through the 256-byte serial I/O3 auto-transfer RAM (addresses $0200_{16}$ to $02FF_{16}$).

The $P8_5$/SRDY3, $P8_6$/SBUSY3, and $P8_7$/SSTB3 pins all have the handshake input/output signal function and can perform active logic high/low selection.

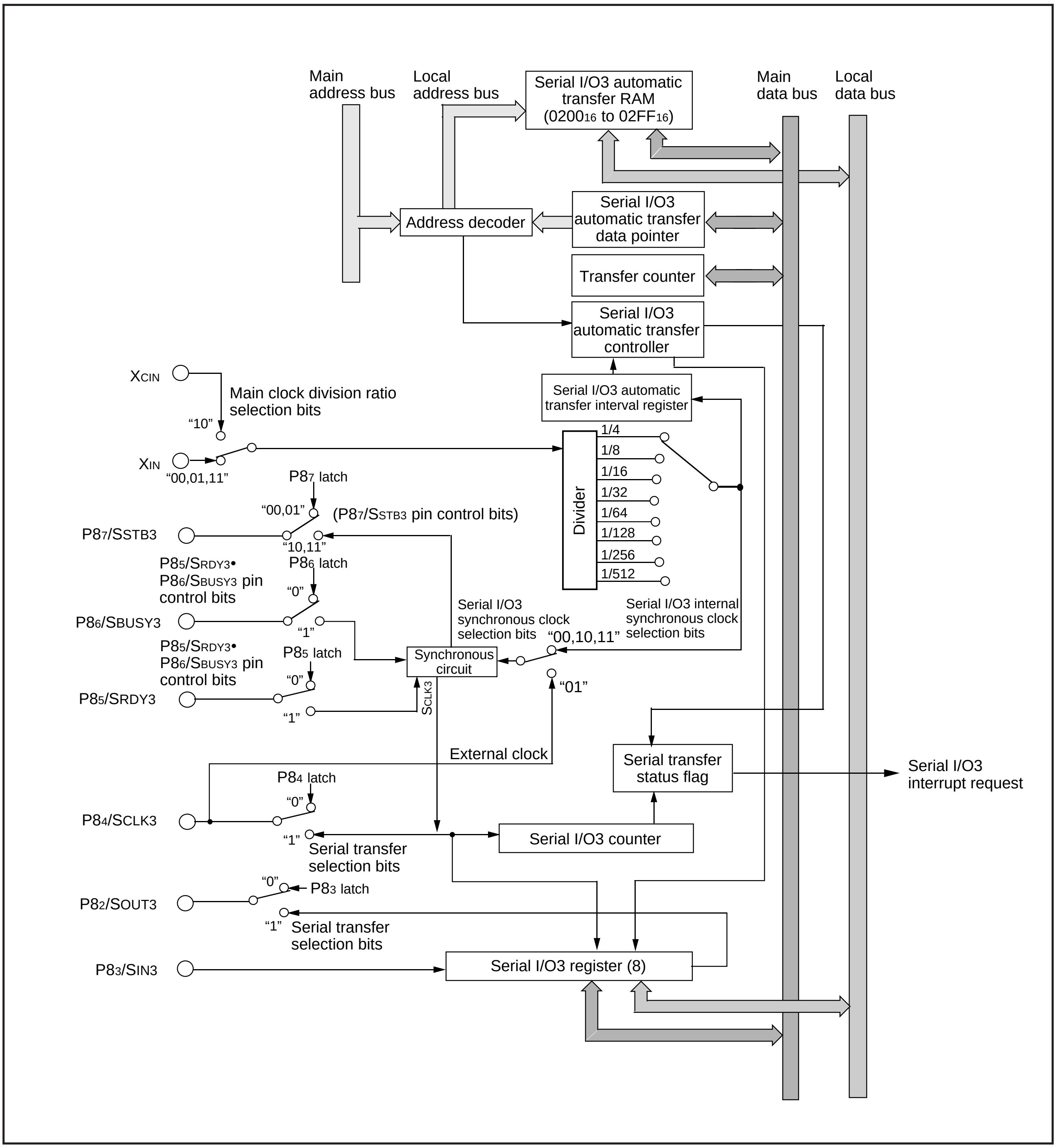



Fig. 34 Block diagram of serial I/O3

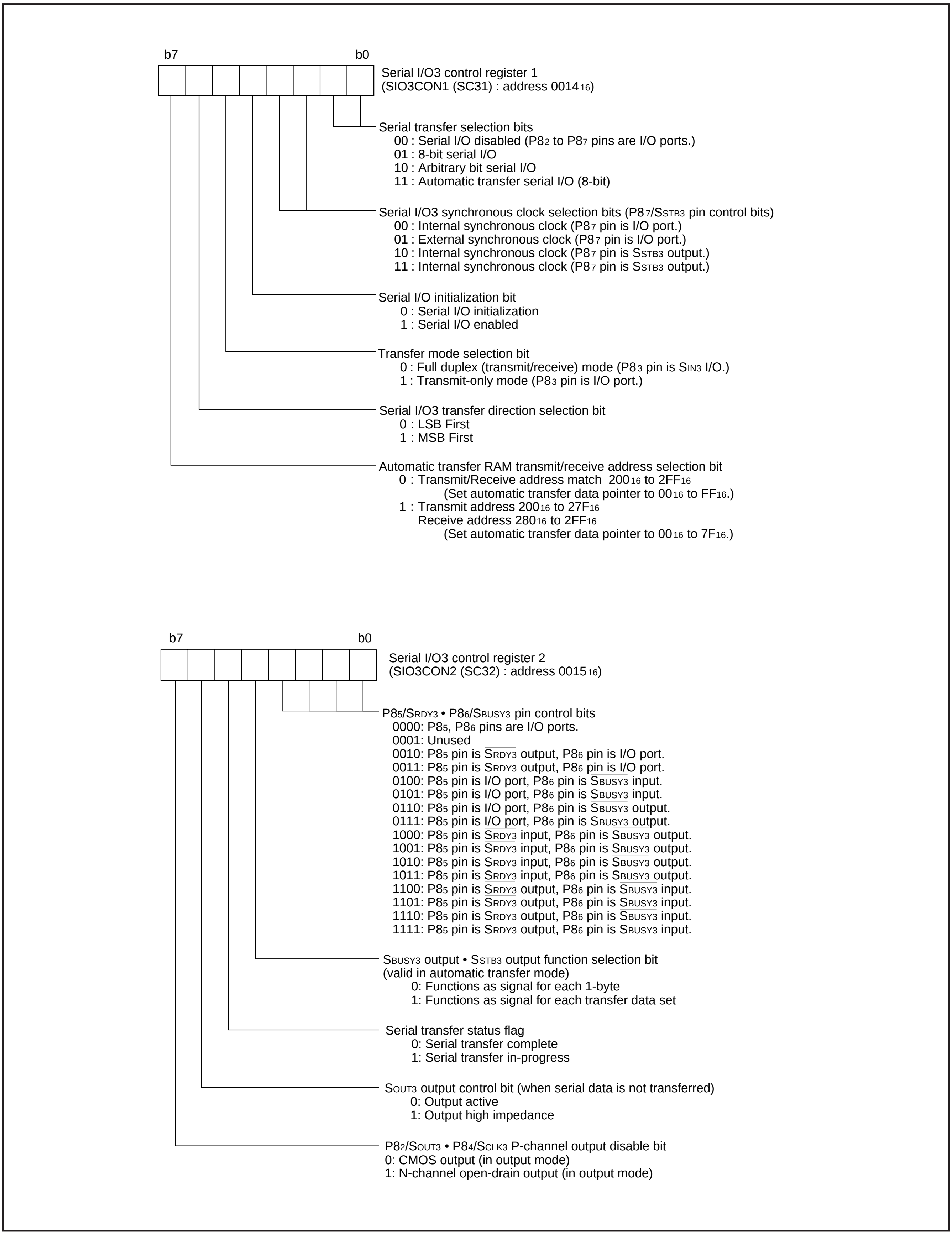



Fig. 35 Structure of serial I/O3 control registers 1 and 2

**●Serial I/O3 Operation**

An internal or external synchronous clock can be selected as the serial transfer synchronous clock by the serial I/O3 synchronous clock selection bits of the serial I/O3 control register 1.

Since the internal synchronous clock has its own built-in divider, 8 types of clocks can be selected by the serial I/O3 internal synchronous clock selection bits of the serial I/O3 control register 3.

Either I/O port or handshake I/O signal function can be selected for the P8$_5$/SRDY3, P8$_6$/SBUSY3, and P8$_7$/SSTB3 pins by the serial I/O3 synchronous clock selection bits (P8$_7$/SSTB3 pin control bits) of the serial I/O3 control register 1 or the P8$_5$/SRDY3•P8$_6$/SBUSY3 pin control bits of the serial I/O3 control register 2.

CMOS output or N-channel open-drain output can be selected for the SCLK3 and SOUT3 output pins by the P8$_2$/SOUT3 • P8$_4$/SCLK3 P-channel output disable bit of the serial I/O3 control register 2.

The SOUT3 output control bit of the serial I/O3 control register 2 can be used to select the status of the SOUT3 pin when serial data is not transferred; either output active or high-impedance. However, when selecting an external synchronous clock, the SOUT3 pin can go to the high-impedance status by setting the SOUT3 output control bit to “1” when SCLK3 input is at “H” after transfer completion. When the next serial transfer begins and SCLK3 goes to “L”, the SOUT3 output control bit is automatically reset to “0” and goes to an output active status.

Regardless of selecting an internal or external synchronous clock, the serial transfer has both a full duplex mode as well as a transmit-only mode. These modes are set by the transfer mode selection bit of serial I/O3 control register 1.

LSB first or MSB first can be selected for the input/output order of the serial transfer bit string by the serial I/O3 transfer direction selection bit of serial I/O3 control register 1.

In order to use serial I/O3, the following process must be followed after all of the above set have been completed: First, select any one of 8-bit serial I/O, arbitrary bit serial I/O, or auto-transfer serial I/O by setting the serial transfer selection bits of the serial I/O3 control register 1. Then, enable the serial I/O by setting the serial I/O initialization bit of the serial I/O3 control register 1 to “1”.

Whether using an internal or external synchronous clock, set the serial I/O initialization bit to “0” when terminating a serial transfer during the transmission.

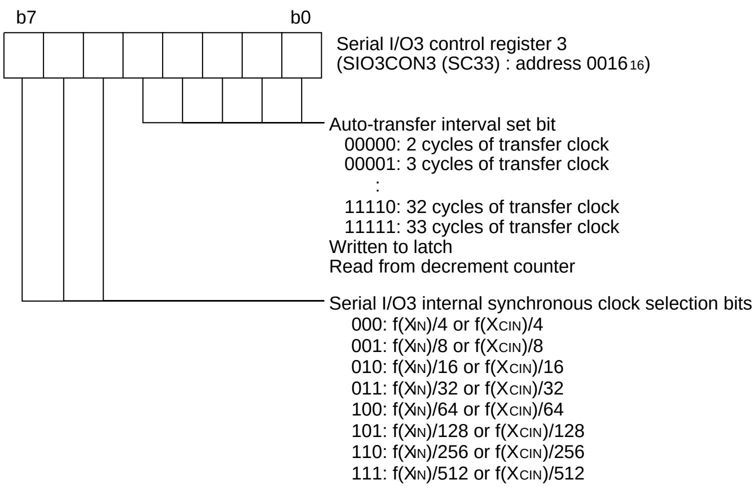



Fig. 36 Structure of serial I/O3 control register 3

## (1) 8-bit serial I/O mode

Address $0013_{16}$ is the serial I/O3 register. When selecting an internal synchronous clock, serial transfer of the 8-bit serial I/O starts by the write signal to the serial I/O3 register (address $0013_{16}$).
The serial transfer status flag of the serial I/O3 control register 2 indicates the serial I/O3 register status. The flag is set to "1" by a serial I/O3 register write, which triggers a transfer start. After the 8-bit transfer is completed, the flag is reset to "0" and a serial I/O3 interrupt request occurs simultaneously.
When an external synchronous clock is selected, the contents of the serial I/O3 register are continually shifted while the transfer clock inputs to SCLK3. In this case, control the clock externally.

## (2) Auto-transfer serial I/O mode

Since read and write to the serial I/O3 register are controlled by the serial I/O3 automatic transfer controller, address $0013_{16}$ functions as the transfer counter (in byte units).
In order to make a serial transfer through the serial I/O3 automatic transfer RAM (addresses $0200_{16}$ to $02FF_{16}$), it is necessary to set the serial I/O3 automatic transfer data pointer before transferring data. The automatic transfer data pointer set bits indicate the low-order 8 bits of the start data stored address. The automatic transfer RAM transmit/receive address select bit can divide the 256-byte serial I/O3 automatic transfer RAM into two areas: 128-byte transmit data area and 128-byte receive data area.
When an internal synchronous clock is selected and any of the following conditions apply, the transfer interval between each 1-byte data can be set by the automatic transfer interval set bits of the serial I/O3 control register 3:

1. The handshake signal is not used.
2. The handshake signal's SRDY3 output, SBUSY3 output, and SSTB3 output are used independently.
3. The handshake signal's output is used in groups: SRDY3/SSTB3 output or SBUSY3/SSTB3.

There are 32 values among 2 and 33 cycles of the transfer clock.
When the automatic transfer interval setting is valid and SBUSY3 output is used, and the SBUSY3 and SSTB3 output function as signal for each transfer data set by the SBUSY3 output•SSTB3 output function selection bit, there is the transfer interval before the first data is transmitted/received, as well as after the last data is transmitted/received. When using SSTB3 output, regardless of the contents of the SBUSY3 output • SSTB3 output function selection bit, this transfer interval become 2 cycles longer than the value set for each 1-byte data. In addition, when using the combined output of SBUSY3 and SSTB3 as the signal for each transfer data set, the transfer interval after completion of transmission/receipt of the last data become 2 cycles longer than the set value.
When selecting an external synchronous clock, the automatic transfer interval cannot be set.
After all of the above bit settings have been completed, and an internal synchronous clock has been selected, serial automatic transfer starts when the value of the number of transfer bytes, decremented by 1, is written to the transfer counter (address $0013_{16}$). When an external synchronous clock is selected, write the value of the transfer bytes, decremented by 1, to the transfer counter, and input the transfer clock to SCLK3 after 5 or more cycles of internal clock $\phi$.
Set the transfer interval of each 1-byte data transmission to 5 or more cycles of the internal clock $\phi$ after the rising edge of the last bit of a 1-byte data.
Regardless of internal or external synchronous clock, the automatic transfer data pointer and transfer counter are both decremented after receipt of each 1-byte data is completed and it is written to the automatic transfer RAM. The serial transfer status flag is set to "1" by writing to the transfer counter which triggers the start of transmission. After the last data is written to the automatic transfer RAM, the serial transfer status flag is set to "0" and a serial I/O3 interrupt request occurs simultaneously.
The write values of the automatic transfer data pointer set bits and the automatic transfer interval set bits are kept in the latch. As a transfer counter write occurs, each value is transferred to its corresponding decrement counter.

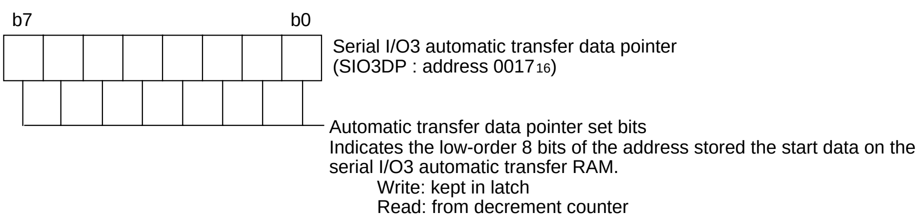



Fig. 37 Structure of serial I/O3 automatic transfer data pointer

## (3) Arbitrary bit serial I/O mode

Since read and write of the serial I/O3 register are controlled by the serial I/O3 automatic transfer controller, address $0013_{16}$ functions as the transfer counter (in byte units).
After the serial I/O3 automatic transfer data pointer and automatic transfer interval set bits have been set, and an internal synchronous clock selected, serial automatic transfer starts when the value of the number of transfer bits decremented by 1 is written to the transfer counter (address $0013_{16}$), just as in the automatic transfer serial I/O. When selecting an external synchronous clock, write the value of the transfer bits decremented by 1 to the transfer counter, then input the transfer clock to $S_{CLK3}$ after 5 or more cycles of internal clock $\phi$. The transfer interval after each 8-bit data transfer must be 5 or more cycles of internal clock $\phi$ after the rising edge of the last bit of the 8-bit data.
When selecting an internal synchronous clock, the automatic transfer interval can be specified regardless of the contents of the selected handshake signal.
In this case, when the automatic transfer interval setting is valid and $S_{BUSY3}$ output is used there are the transfer interval before the first data is transmitted/received, as well as after the last data is transmitted/received just as in the automatic transfer serial I/O mode. When using $S_{STB3}$ output, this transfer interval become 2 cycles longer than the value set for each 8-bit data. In addition, when using the combined output of $S_{BUSY3}$ and $S_{STB3}$, the transfer interval after completion of transmission/receipt of the last data become 2 cycles longer than the set value.
When selecting an external synchronous clock, the automatic transfer interval cannot be specified.
Regardless of internal or external synchronous clock, the automatic transfer data pointer is decremented after each 8-bit data is received and then written to the auto-transfer RAM. The transfer counter is decremented with the transfer clock. The serial transfer status flag is set to "1" by writing to the transfer counter which triggers the start of transmission. After the last data is written to the automatic transfer RAM, the serial transfer status flag is set to "0" and a serial I/O3 interrupt request occurs simultaneously.
The write values of the automatic transfer data pointer set bits and the automatic transfer interval set bits are kept in the latch. As a transfer counter write occurs, each value is transferred to its corresponding decrement counter.
If the last data does not fill 8 bits, the receive data stored in the serial I/O3 automatic transfer RAM become the closest MSB odd bit if the transfer direction select bit is set to LSB first, or the closest LSB odd bit if the transfer direction select bit is set to MSB first.

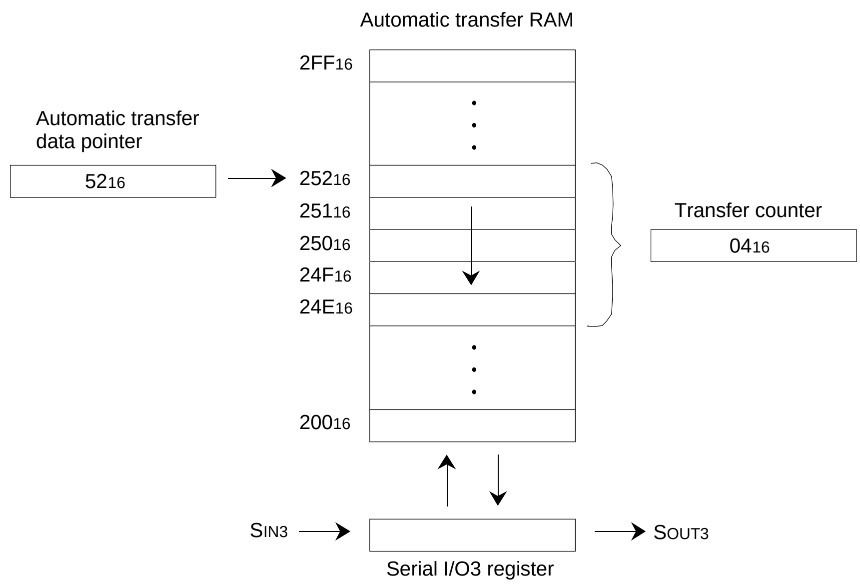



Fig. 38 Automatic transfer serial I/O operation

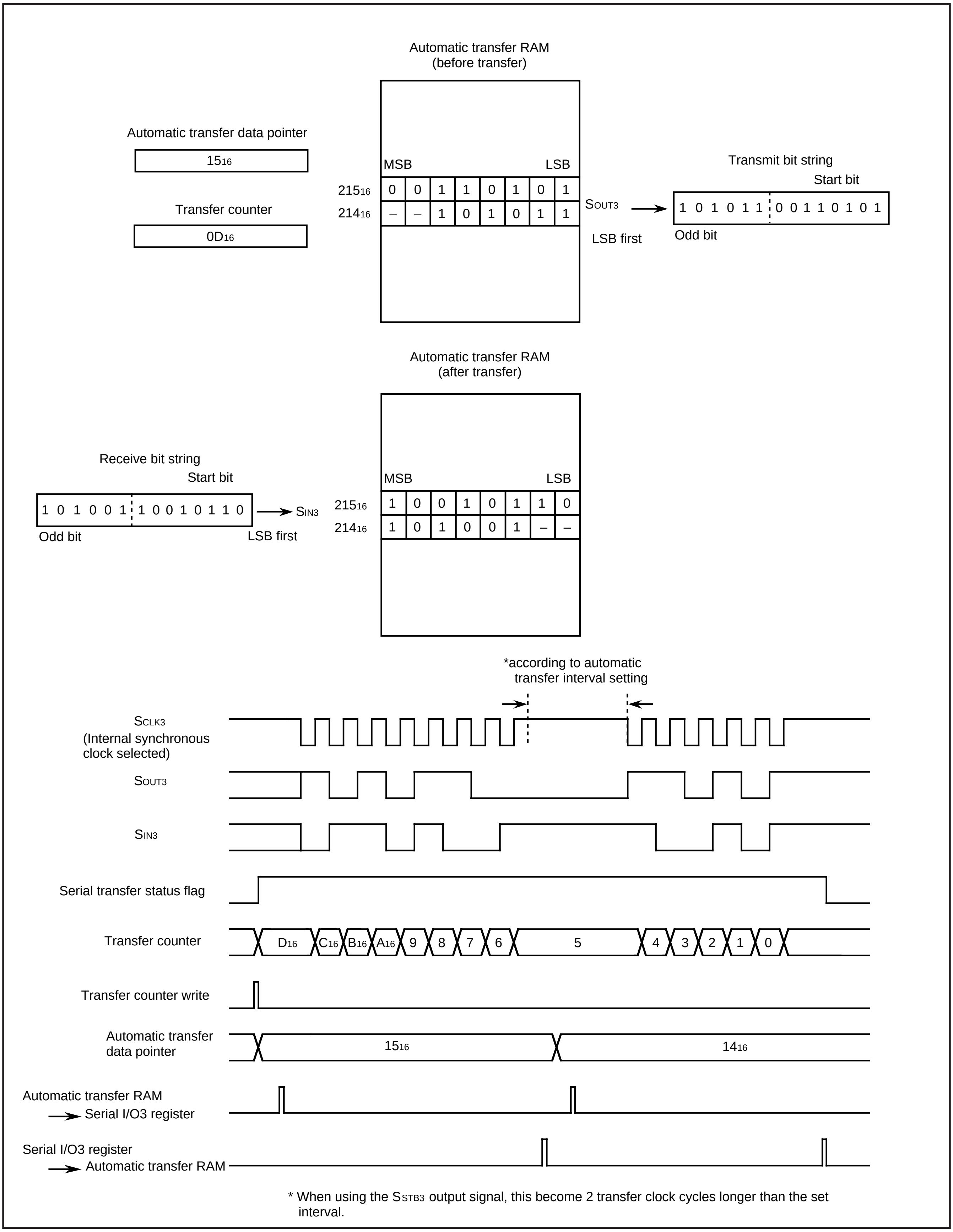



Fig. 39 Arbitrary bit serial I/O operation

## Handshake Signal

● **SSTB3 output signal**

The SSTB3 output is a signal to inform an end of transmission/reception to the serial transfer destination . The SSTB3 output signal can be used only when the internal synchronous clock is selected.
In the initial status, that is, in the status in which the serial I/O initialization bit (b4) is reset to “0”, the SSTB3 output goes to “L”, and the $\overline{\text{SSTB3}}$ output goes to “H”.
At the end of transmit/receive operation, when the data of the serial I/O3 register is all output from SOUT3, pulses which are the SSTB3 output of “H” and the $\overline{\text{SSTB3}}$ output of “L” are output in the period of 1 cycle of the transfer clock. After that, each pulse is returned to the initial status in which SSTB3 output goes to “L” and the $\overline{\text{SSTB3}}$ output goes to “H”.
Furthermore, after 1 cycle, the serial transfer status flag (b5) is reset to “0”.
In the automatic transfer serial I/O mode, whether making the SSTB3 output active at an end of each 1-byte data or after completion of transfer of all data can be selected by the SBUSY3 output • SSTB3 output function selection bit (b4 of address $0015_{16}$) of serial I/O3 control register 2.

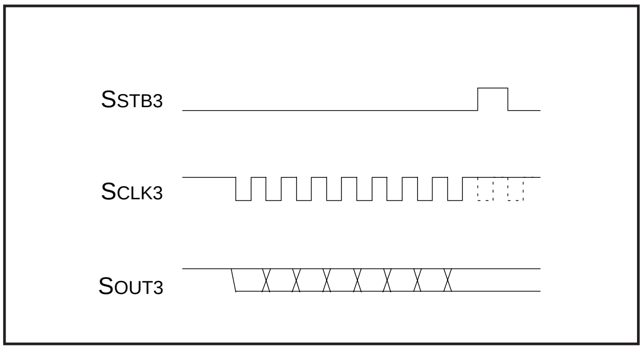



Fig. 40 SSTB3 output operation

● **SBUSY3 input signal**

The SBUSY3 input is a signal which receives a request for a stop of transmission/reception from the serial transfer destination.
When the internal synchronous clock is selected, input an “H” level signal into the SBUSY3 input and an “L” level signal into the $\overline{\text{SBUSY3}}$ input in the initial status in which transfer is stopped.
When starting a transmit/receive operation, input an “L” level signal into the SBUSY3 input and an “H” level signal into the $\overline{\text{SBUSY3}}$ input in the period of 1.5 cycles or more of the transfer clock. Then, transfer clocks are output from the SCLK3 output.
When an “H” level signal is input into the SBUSY3 input and an “L” level signal into the $\overline{\text{SBUSY3}}$ input after a transmit/receive operation is started, this transmit/receive operation are not stopped immediately and the transfer clocks from the SCLK3 output are not stopped until the specified number of bits is transmitted and received.
The handshake unit of the 8-bit serial I/O is 8 bits and that of the arbitrary bit serial I/O is the bit number adding “1” to the set value to the transfer counter, and that of the automatic transfer serial I/O is 8 bits.

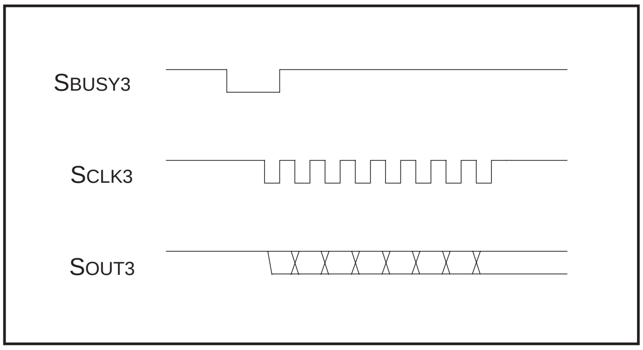



Fig. 41 SBUSY3 input operation (internal synchronous clock)

When the external synchronous clock is selected, input an “H” level signal into the SBUSY3 input and an “L” level signal into the $\overline{\text{SBUSY3}}$ input in the initial status in which transfer is stopped. At this time, the transfer clocks to be input in SCLK3 become invalid.
During serial transfer, the transfer clocks to be input in SCLK3 become valid, enabling a transmit/receive operation, while an “L” level signal is input into the SBUSY3 input and an “H” level signal is input into the $\overline{\text{SBUSY3}}$ input.
When changing the input values in to the SBUSY3 input and the $\overline{\text{SBUSY3}}$ input in these operations, change them while the SCLK3 input is in a high state.
When the high impedance of the SOUT3 output is selected by the SOUT3 output control bit (b6), the SOUT3 output becomes active, enabling serial transfer by inputting a transfer clock to SCLK3, while an “L” level signal is input into the SBUSY3 input and an “H” level signal is input into the $\overline{\text{SBUSY3}}$ input.

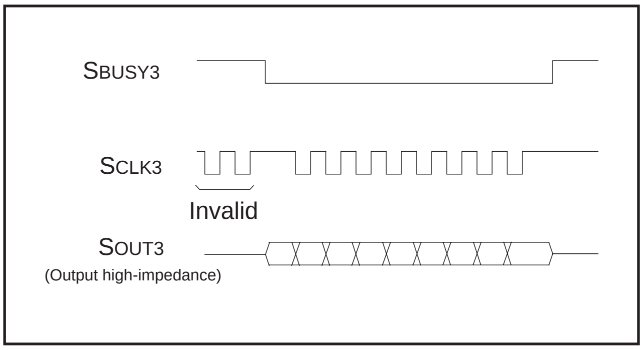



Fig. 42 SBUSY3 input operation (external synchronous clock)

● **SBUSY3 output signal**

The SBUSY3 output is a signal which requests a stop of transmission/reception to the serial transfer destination. In the automatic transfer serial I/O mode, regardless of the internal or external synchronous clock, whether making the SBUSY3 output active at transfer of each 1-byte data or during transfer of all data can be selected by the SBUSY3 output • SSTB3 output function selection bit (b4).
In the initial status, that is, the status in which the serial I/O initialization bit (b4) is reset to “0”, the SBUSY3 output goes to “H” and the $\overline{\text{SBUSY3}}$ output goes to “L”.

When the internal synchronous clock is selected, in the 8-bit serial I/O mode and the automatic transfer serial I/O mode (SBUSY3 output function outputs in 1-byte units), the SBUSY3 output goes to “L” and the $\overline{\text{SBUSY3}}$ output goes to “H” before 0.5 cycle (transfer clock) of the timing at which the transfer clock from the SCLK3 output goes to “L” at a start of transmit/receive operation.

In the automatic transfer serial I/O mode (the SBUSY3 output function outputs all transfer data), the SBUSY3 output goes to “L” and the $\overline{\text{SBUSY3}}$ output goes to “H” when the first transmit data is written into the serial I/O3 register (address $0013_{16}$).

When the external synchronous clock is selected, the SBUSY3 output goes to “L” and the $\overline{\text{SBUSY3}}$ output goes to “H” when transmit data is written into the serial I/O3 register to start a transmit operation, regardless of the serial I/O transfer mode.

At termination of transmit/receive operation, the SBUSY3 output returns to “H” and the $\overline{\text{SBUSY3}}$ output returns to “L”, the initial status, when the serial transfer status flag is set to “0”, regardless of selecting the internal or external synchronous clock.

Furthermore, in the automatic transfer serial I/O mode (SBUSY3 output function outputs in 1-byte units), the SBUSY3 output goes to “H” and the $\overline{\text{SBUSY3}}$ output goes to “L” each time 1-byte of receive data is written into the automatic transfer RAM.

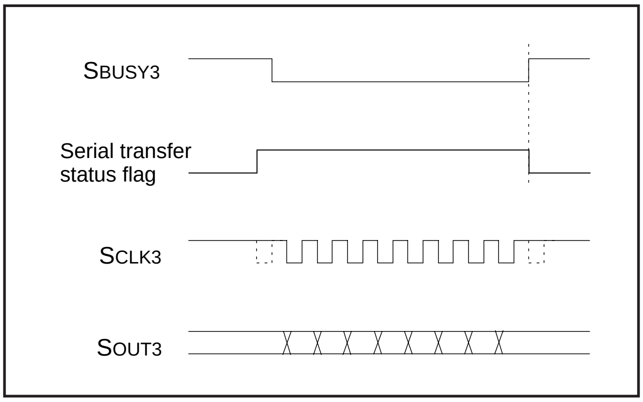



Fig. 43 SBUSY3 output operation
(internal synchronous clock, 8-bits serial I/O)

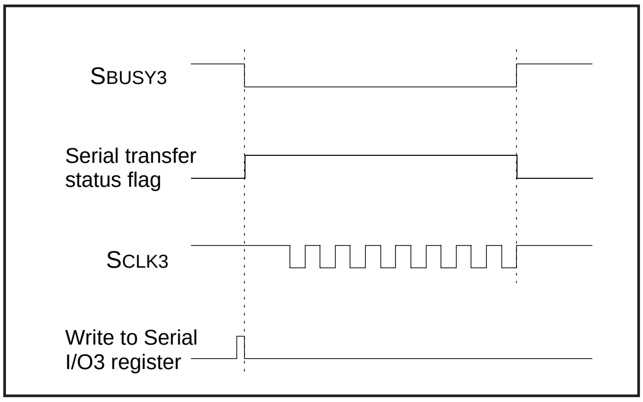



Fig. 44 SBUSY3 output operation
(external synchronous clock, 8-bits serial I/O)

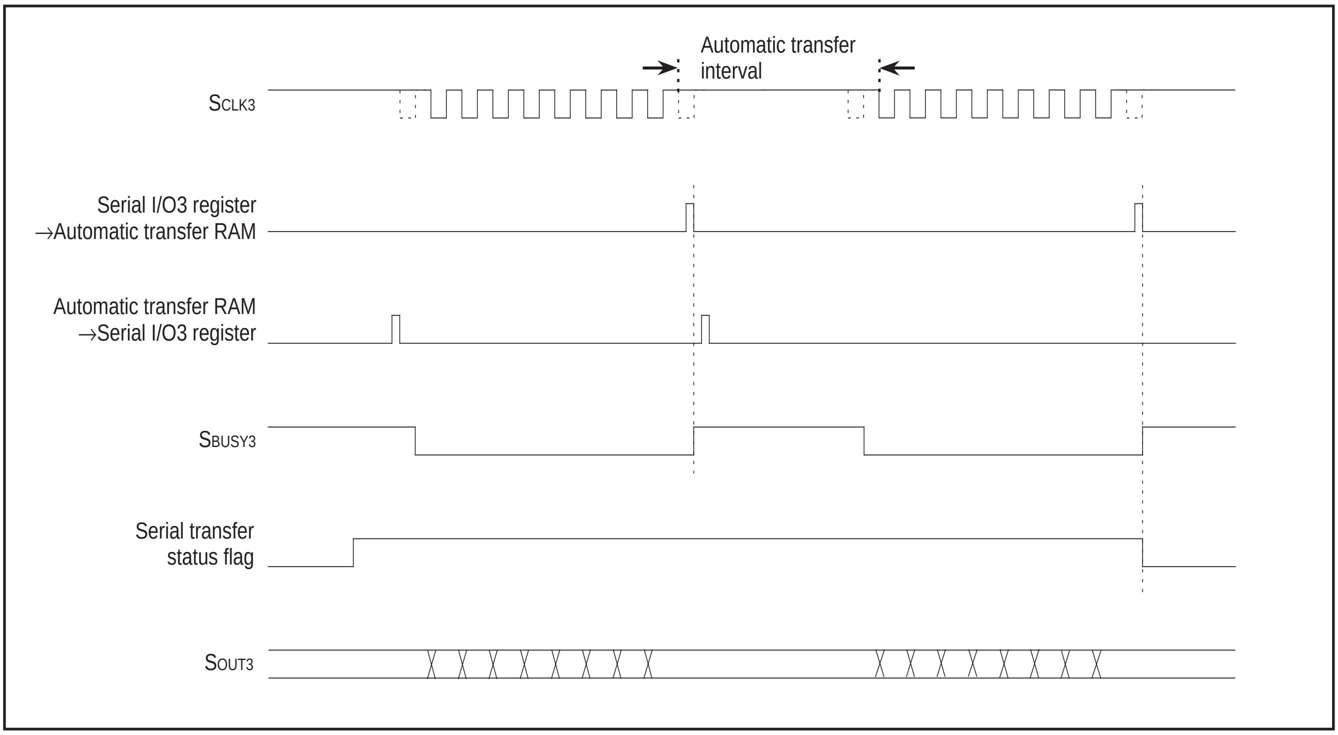



Fig. 45 SBUSY3 output operation in automatic transfer serial I/O mode
(internal synchronous clock, SBUSY3 output function outputs each 1-byte)

**● SRDY3 output signal**

The SRDY3 output is a transmit/receive enable signal which informs the serial transfer destination that transmit/receive is ready. In the initial status, that is, when the serial I/O initialization bit (b4) is reset to “0”, the SRDY3 output goes to “L” and the $\overline{\text{SRDY3}}$ output goes to “H”. After transmitted data is stored in the serial I/O3 register (address $0013_{16}$) and a transmit/receive operation becomes ready, the SRDY3 output goes to “H” and the $\overline{\text{SRDY3}}$ output goes to “L”. When a transmit/receive operation is started and the transfer clock goes to “L”, the SRDY3 output goes to “L” and the $\overline{\text{SRDY3}}$ output goes to “H”.

**● SRDY3 input signal**

The SRDY3 input signal becomes valid only when the SRDY3 input and the SBUSY3 output are used. The SRDY3 input is a signal for receiving a transmit/receive ready completion signal from the serial transfer destination.

When the internal synchronous clock is selected, input a low level signal into the SRDY3 input and a high level signal into the $\overline{\text{SRDY3}}$ input in the initial status in which the transfer is stopped.

When an “H” level signal is input into the SRDY3 input and an “L” level signal is input into the $\overline{\text{SRDY3}}$ input for a period of 1.5 cycles or more of transfer clock, transfer clocks are output from the SCLK3 output and a transmit/receive operation is started.

After the transmit/receive operation is started and an “L” level signal is input into the SRDY3 input and an “H” level signal into the $\overline{\text{SRDY3}}$ input, this operation cannot be immediately stopped.

After the specified number of bits are transmitted and received, the transfer clocks from the SCLK3 output is stopped. The handshake unit of the 8-bit serial I/O and that of the automatic transfer serial I/O are of 8 bits. That of the arbitrary bit serial I/O is the bit number adding “1” to the set value to the transfer counter.

When the external synchronous clock is selected, the SRDY3 input becomes one of the triggers to output the SBUSY3 signal.

To start a transmit/receive operation (SBUSY3 output to “L”, $\overline{\text{SBUSY3}}$ output to “H”), input an “H” level signal into the SRDY3 input and an “L” level signal into the $\overline{\text{SRDY3}}$ input, and also write transmit data into the serial I/O3 register.

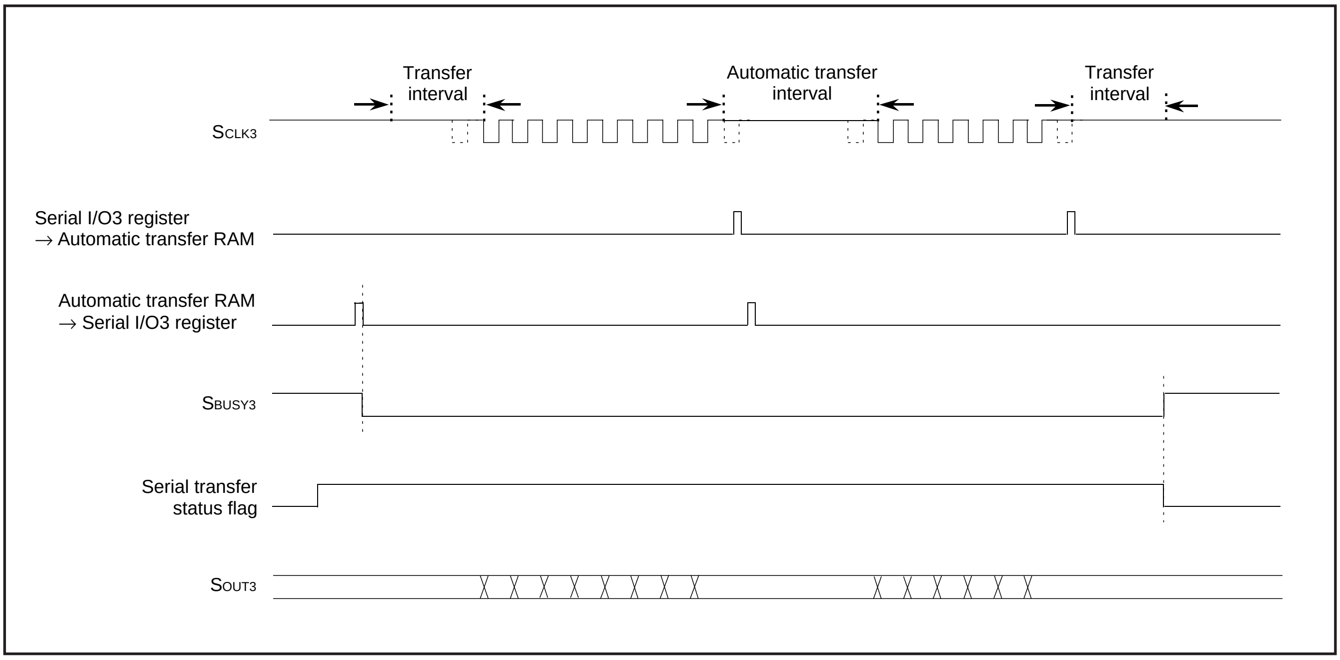



Fig. 46 SBUSY3 output operation in arbitrary bit serial I/O mode (internal synchronous clock)

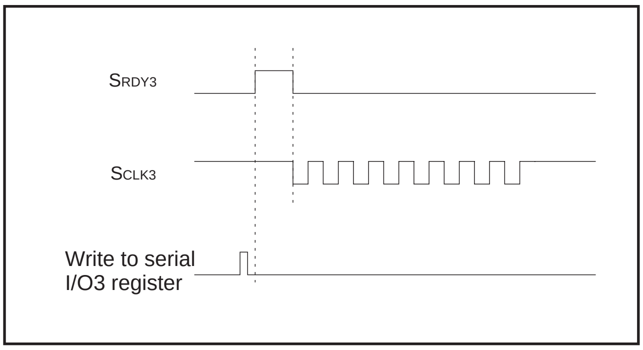



Fig. 47 SRDY3 Output Operation

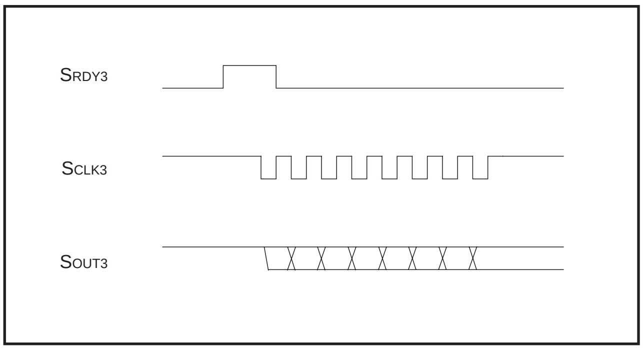



Fig. 48 SRDY3 Input Operation (internal synchronous clock)

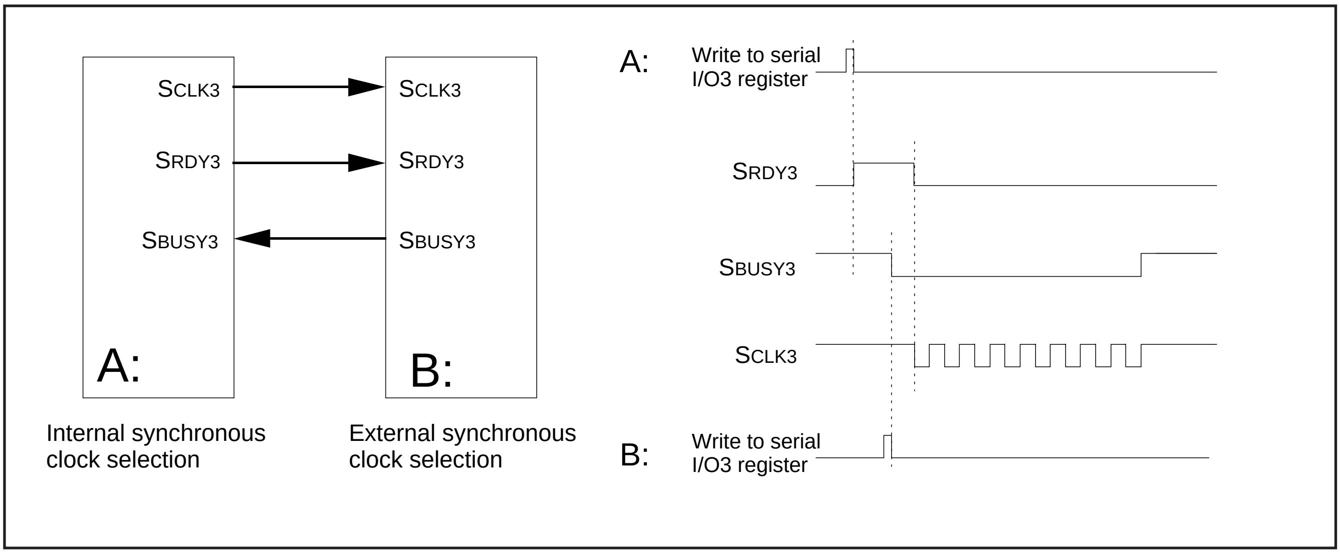



Fig. 49 Handshake operation at serial I/O3 mutual connecting (1)

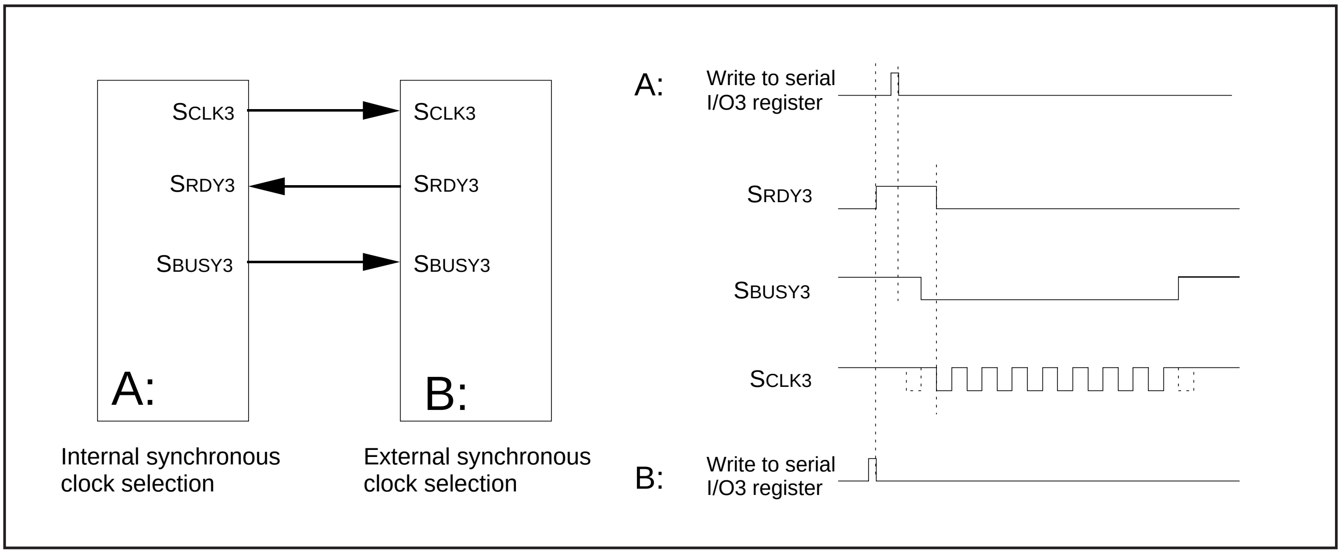



Fig. 50 Handshake operation at serial I/O3 mutual connecting (2)

## DATA LINK LAYER COMMUNICATION CONTROL CIRCUIT

The 3874 Group has a built-in data link layer communication control circuit.
This data link layer communication control circuit is applicable for multi-master serial bus communication control used only with data lines through an external driver/receiver.
The data link layer communication control circuit consists of following.

- Communication mode register (address $002A_{16}$)
- Transmit control register (address $002B_{16}$)
- Transmit status register (address $002C_{16}$)
- Receive control register (address $002D_{16}$)
- Receive status register (address $002E_{16}$)
- Bus interrupt factor determination control register (address $002F_{16}$)
- Control field select register (address $0030_{16}$)
- Control field data register (address $0031_{16}$)
- Transmit/Receive FIFO (address $0032_{16}$)

This function is realized by hardware and firmware so that communication protocol can be partially modified according to the user's specification.
The following are the standard communication rate and functions which the data link layer communication control circuit can perform.

- Communication rate: Approx. 40 kbps
  The communication rate depends on frame or bit protocol.
- Synchronous method: Half-duplex asynchronous
- Modification method: PWM method, NRZ, etc.
- Communication functions:
  ①Bus arbitration
  (CSMA/CD method, etc.)
  ②Error detection
  (parity, acknowledge, CRC, etc.)
  ③Frame, data retry

The transmission signal is output from the $\overline{BUS_{OUT}}$ pin and input to the $\overline{BUS_{IN}}$ pin.
Detailed specifications for communication protocol, bit assignment, function, etc. of each register are defined according to each communication protocol specification confirmation.

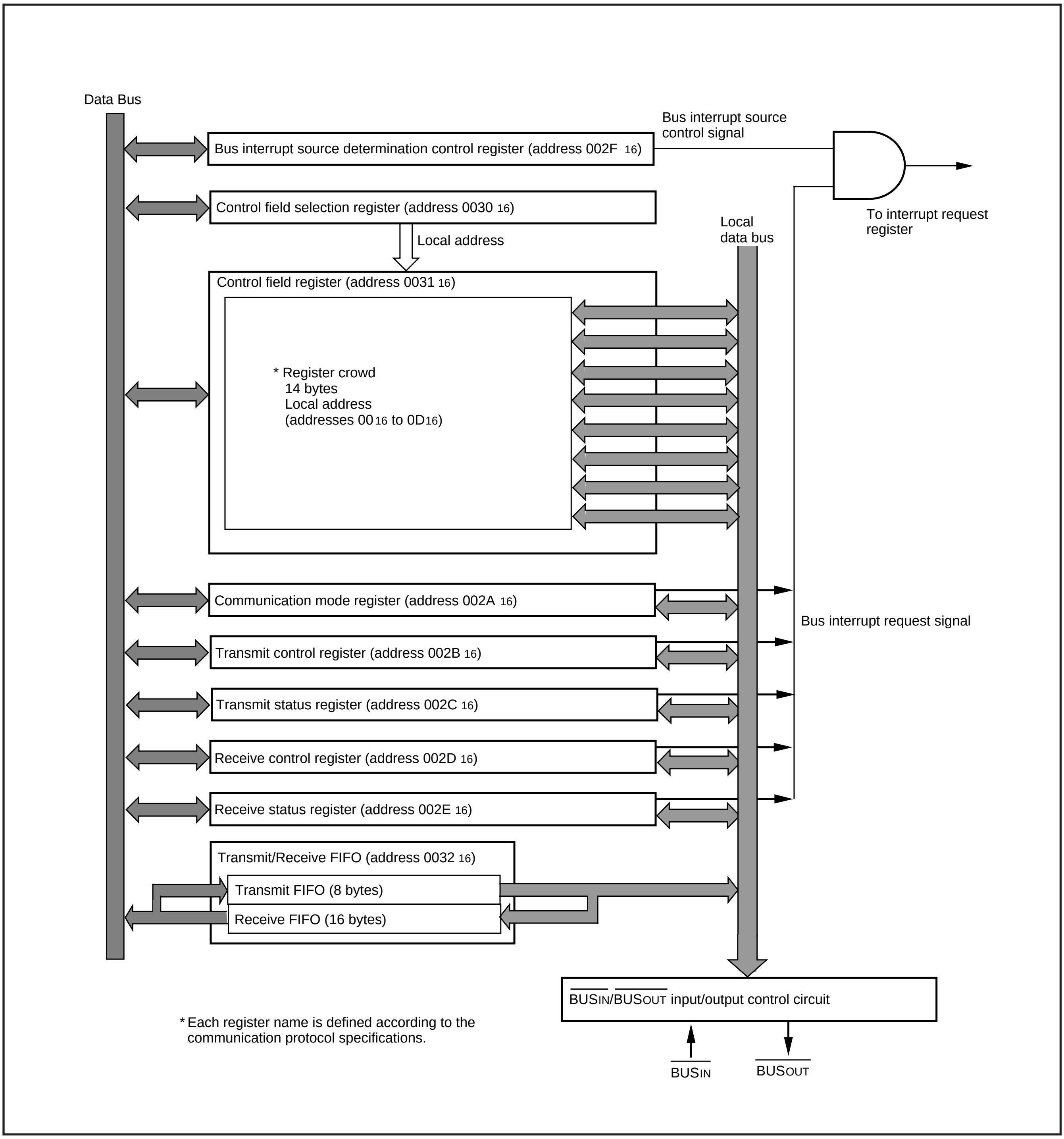



Fig. 51 Data link layer communication control circuit block example

## [Communication Mode Register (BUSM)] 002A16

The communication mode register (address $002A_{16}$) has 6 bits and consists of all the control bits for the communication mode.

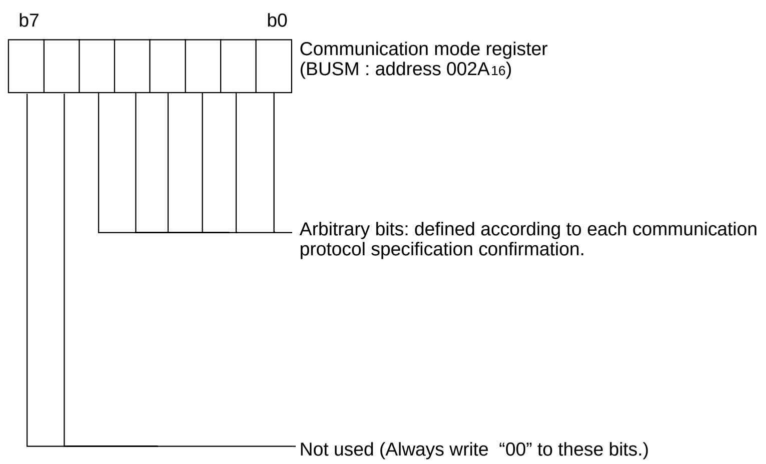



Fig. 52 Structure of communication mode register

## [Transmit Control Register (TXDCON)] 002B16

The transmit control register (address 002B16) has 7 bits and consists of the transmit control and transmit status flags.

## [Transmit Status Register (TXDSTS)] 002C16

The transmit status register (address 002C16) has 8 bits and consists of the transmit error flag and transmit interrupt request flag.

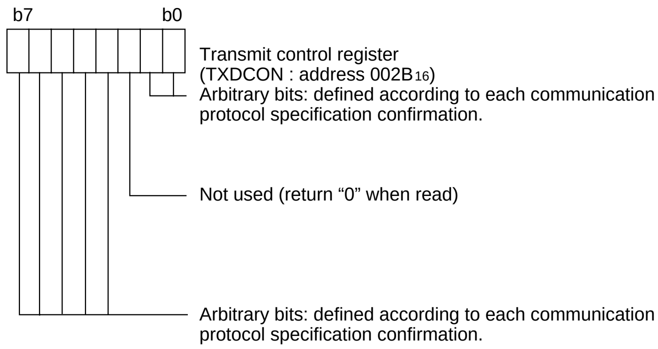
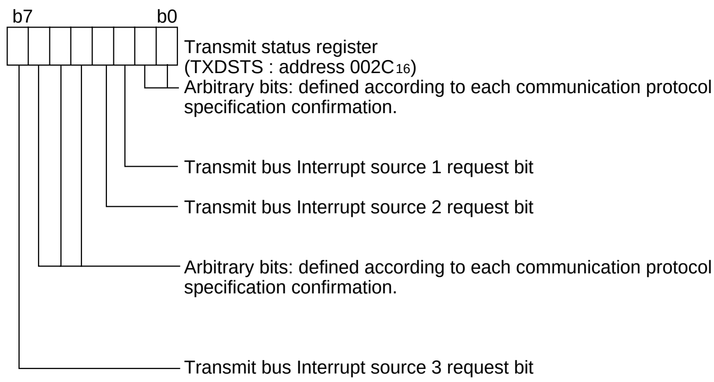



Fig. 53 Structure of transmit control register




**Note:** Bits 0 to 3, bit 5, and bit 7 can be cleared only by software.

When a transmit bus interrupt source request bit is "1," an interrupt request occurs. The name and function of each transmit bus interrupt source is defined according to the communication protocol specification confirmation.

Fig. 54 Structure of transmit status register

## [Receive control register (RXDCON)] $002D_{16}$

The receive control register has 7 bits and consists of the receive control and receive status flags.

## [Receive status register (RXDSTS)] $002E_{16}$

The receive status register has 8 bits and consists of the receive error flag and receive interrupt request flags.

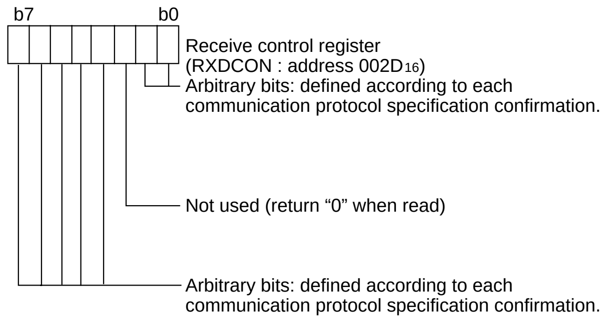



Fig. 55 Structure of receive control register

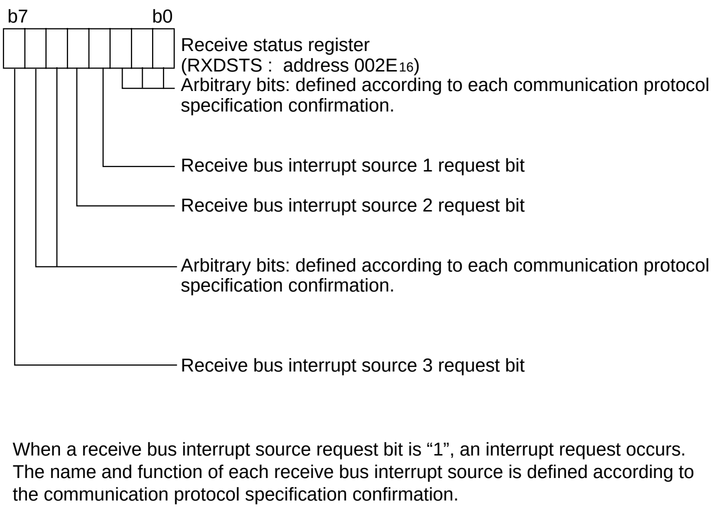



Fig. 56 Structure of receive status register

### [Control field selection register (CFSEL)] $0030_{16}$
### [Control field register (CF)] $0031_{16}$

The control field data select the control field selection register (address $0030_{16}$) value as the pointer. The data can be confirmed and changed by a read/write of the control field register (address $0030_{16}$).

For example, when reading/writing the local address "$00_{16}$," the control field selection register is set to "$00_{16}$" and the control field register is read/written.

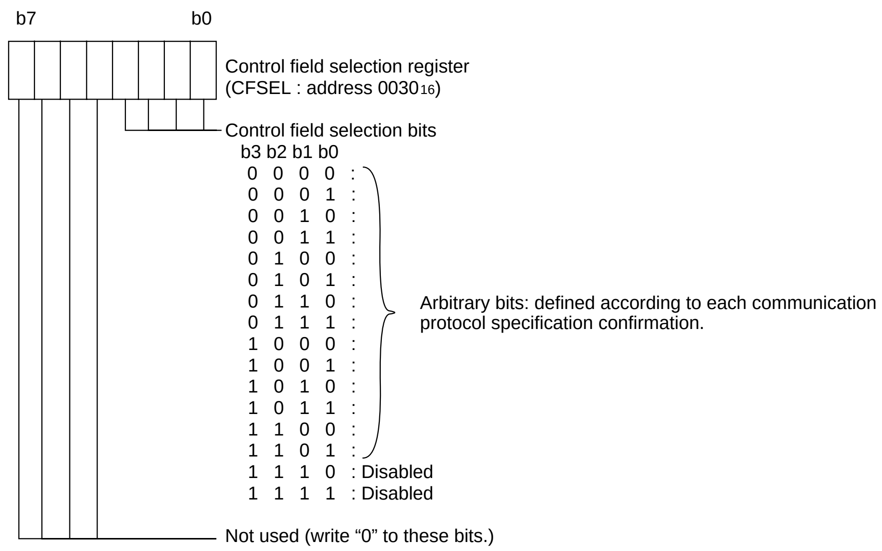
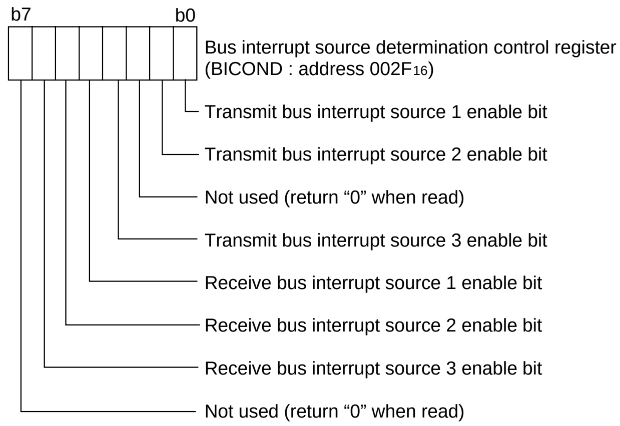



Fig. 57 Structure of control field selection register

### [Bus interrupt source determination control register (BICOND)] $002F_{16}$

The bus interrupt source determination control register (address $002F_{16}$) has 6 bits and controls bus-related interrupts. Refer to the section concerning interrupts for details about priority and vector addresses.




0: Interrupt disabled
1: Interrupt enabled

The name and function of each transmit/receive bus interrupt source is defined according to the communication protocol specification confirmation.

Fig. 58 Structure of bus interrupt source determination control register

## A-D CONVERTER
## [A-D/D-A Conversion Register (AD)] $0035_{16}$

The A-D/D-A conversion register is a register (at reading) that contains the result of an A-D conversion. When reading this register during an A-D conversion, the previous conversion result is read.

## [A-D Control Register (ADCON)] $0034_{16}$

The A-D control register controls the A-D/D-A conversion process. Bits 0 to 2 of this register select specific analog input pins. Bit 3 signals the completion of an A-D conversion. The value of this bit remains at “0” during an A-D conversion, then changes to “1” when the A-D conversion is completed. Writing “0” to this bit starts the A-D conversion. When bit 5, which is the AD external trigger valid bit, is set to “1”, this bit enables A-D conversion even by a falling edge of an ADT input. Set “0” (input port) to the direction register corresponding the ADT pin. Bit 6 is the interrupt source selection bit. Writing “0” to this bit, A-D converter interrupt request occurs at completion of A-D conversion. Writing “1” to this bit the interrupt request occurs at falling edge of an ADT input.

## Comparison Voltage Generator

The comparison voltage generator divides the voltage between $AV_{SS}$ and $V_{REF}$ by 256, and outputs the divided voltages.

## Channel Selector

The channel selector selects one of the input ports $P6_7/AN_7$ to $P6_0/AN_0$ and inputs it to the comparator.

## Comparator and Control Circuit

The comparator and control circuit compares an analog input voltage with the comparison voltage and stores the result in the A-D/D-A conversion register. When an A-D conversion is completed, the control circuit sets the AD conversion completion bit and the AD conversion interrupt request bit to “1”.

Note that the comparator is constructed linked to a capacitor, so set $f(X_{IN})$ to at least 500 kHz during A-D conversion. Use a CPU system clock dividing the main clock $X_{IN}$.

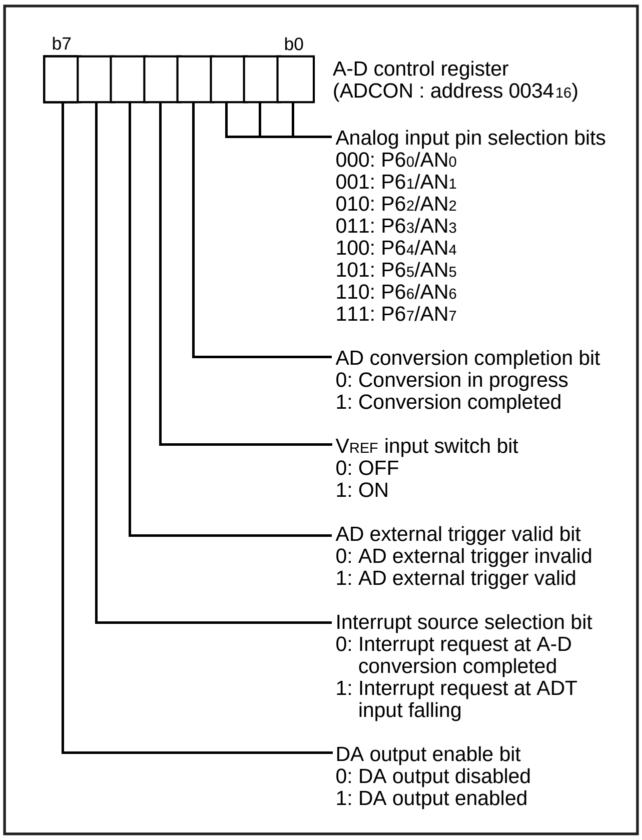



Fig. 59 Structure of A-D control register

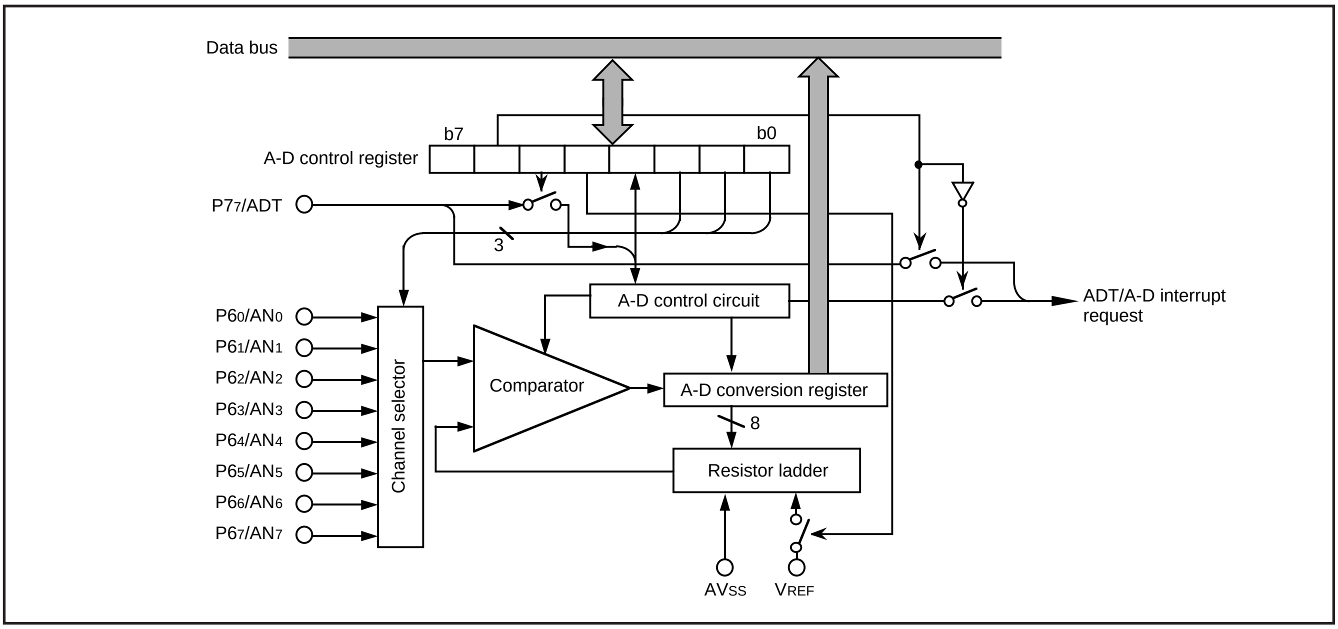



Fig. 60 Block diagram of A-D converter

## D-A CONVERTER

The 3874 group has an on-chip D-A converter with 8-bit resolution and 1 channel. The D-A conversion is performed by setting the value in the A-D/D-A conversion register. The result of D-A converter is output from DA pin by setting the DA output enable bits to "1". When using the D-A converter, the corresponding port direction register bit ($P8_0$/DA) should be set to "0" (input status).
The output analog voltage V is determined by the value n (base 10) in the A-D/D-A conversion register as follows:

$V = V_{REF} \times n/256$ (n=0 to 255)
Where $V_{REF}$ is the reference voltage.

At reset, the D-A conversion registers are cleared to "$00_{16}$", the DA output enable bits are cleared to "0", and $P8_0$/DA pin becomes high impedance. The DA output is not buffered, so connect an external buffer when driving a low-impedance load. When using D-A converter, set 4.0 V or more to $V_{CC}$.

## ■ Note

When reading the A-D/D-A conversion register, the A-D conversion result is read, and the set value for D-A conversion is not read.

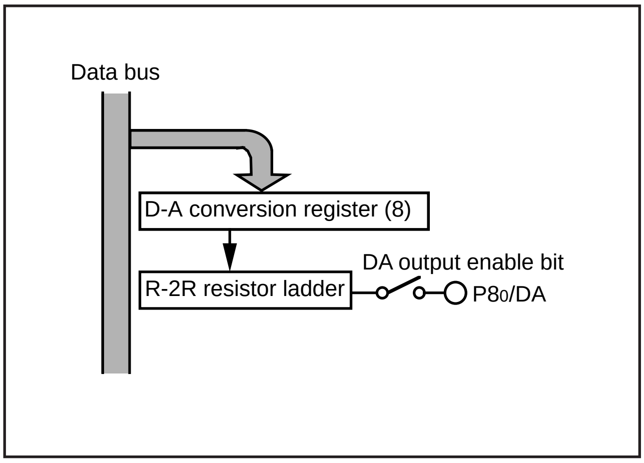


Fig. 61 Block diagram of D-A converter

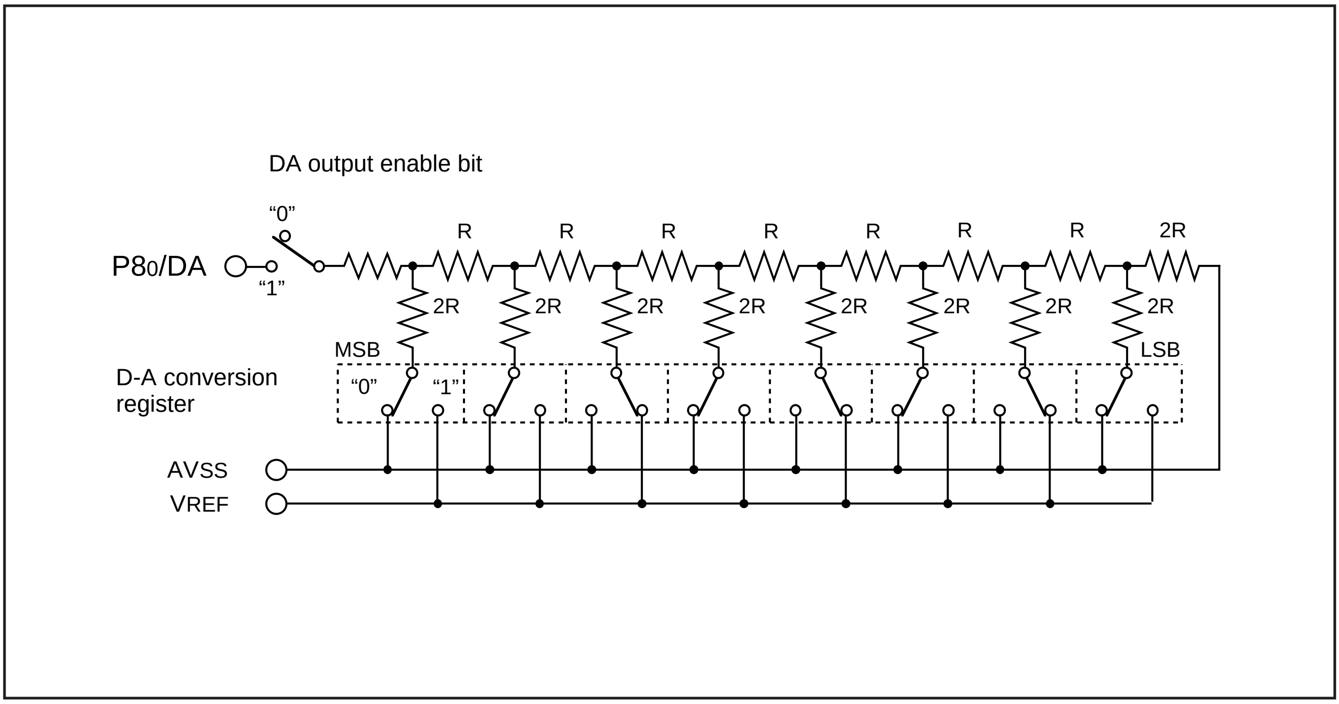


Fig. 62 Equivalent connection circuit of D-A converter

## WATCHDOG TIMER

The watchdog timer gives a mean of returning to the reset status when a program cannot run on a normal loop (for example, because of a software run-away). The watchdog timer consists of an 8-bit watchdog timer L and a 12-bit watchdog timer H.

### Watchdog Timer Initial Value

Watchdog timer L is set to “$FF_{16}$” and watchdog timer H is set to “$FFF_{16}$” by writing to the watchdog timer control register or at a reset. Any write instruction that causes a write signal can be used, such as the STA, LDM, CLB, etc. Data can only be written to bits 6 and 7 of the watchdog control register. Regardless of the value written to bits 0 to 5, the above-mentioned value will be set to each timer.

### Watchdog Timer Operations

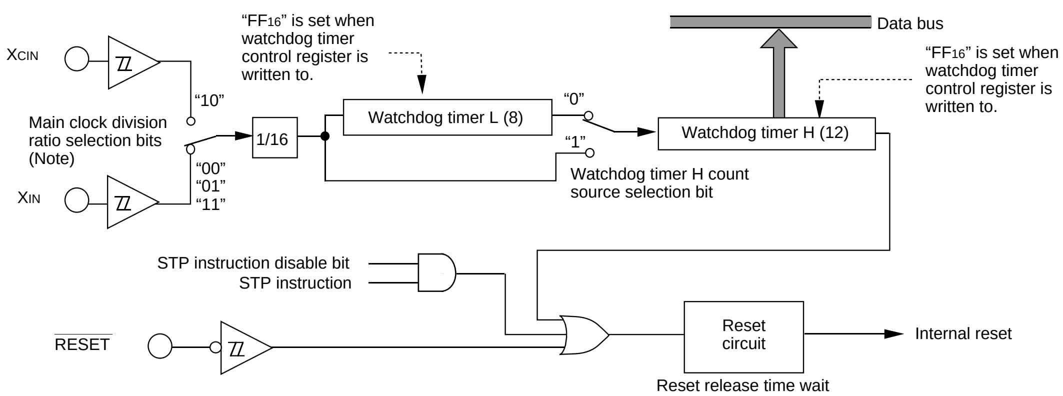
The watchdog timer stops at reset and a countdown is started by the writing to the watchdog timer control register. An internal reset occurs when watchdog timer H underflows. The reset is released after its release time. After the release, the program is restarted from the reset vector address. Usually, write to the watchdog timer control register by software before an underflow of the watchdog timer H. The watchdog timer does not function if the watchdog timer control register is not written to at least once.

When bit 6 of the watchdog timer control register is kept at “0”, the STP instruction is enabled. When that is executed, both the clock and the watchdog timer stop. Count re-starts at the same time as the release of stop mode **(Note)**. The watchdog timer does not stop while a WIT instruction is executed. In addition, the STP instruction is disabled by writing “1” to this bit again. When the STP instruction is executed at this time, it is processed as an undefined instruction, and an internal reset occurs. Once a “1” is written to this bit, it cannot be programmed to “0” again.
The following shows the period between the write execution to the watchdog timer control register and the underflow of watchdog timer H.

Bit 7 of the watchdog timer control register is “0”:
 when $X_{CIN}$ = 32 kHz; 524 s
 when $X_{IN}$ = 6.4 MHz; 2.6 s

Bit 7 of the watchdog timer control register is “1”:
 when $X_{CIN}$ = 32 kHz; 2 s
 when $X_{IN}$ = 6.4 MHz; 10 ms

**Note:** The watchdog timer continues to count even while waiting for a stop release. Therefore, make sure that watchdog timer H does not underflow during this period.




**Note:** Either double-speed, high-speed, middle-speed, or low-speed mode is selected by bits 7 and 6 of the CPU mode register.

Fig. 63 Block diagram of Watchdog timer

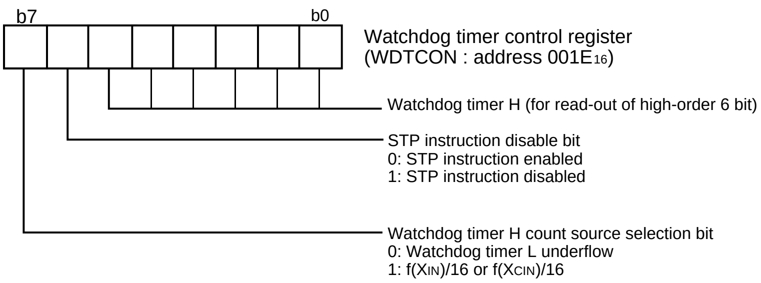



Fig. 64 Structure of Watchdog timer control register

## RESET CIRCUIT

To reset the microcomputer, $\overline{\text{RESET}}$ pin should be held at an “L” level for 2 µs or more. Then the $\overline{\text{RESET}}$ pin is returned to an “H” level (the power source voltage should be between 3.0 V and 5.5 V, and the oscillation should be stable), reset is released. After the reset is completed, the program starts from the address contained in address $FFFD_{16}$ (high-order byte) and address $FFFC_{16}$ (low-order byte). Make sure that the reset input voltage is 0.6 V or less for $V_{CC}$ of 3.0 V.

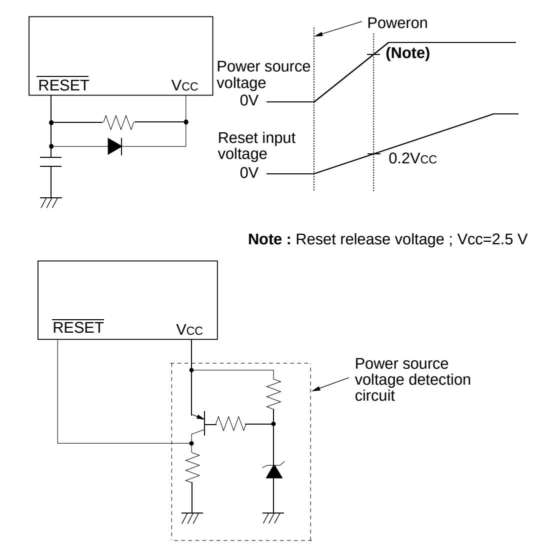
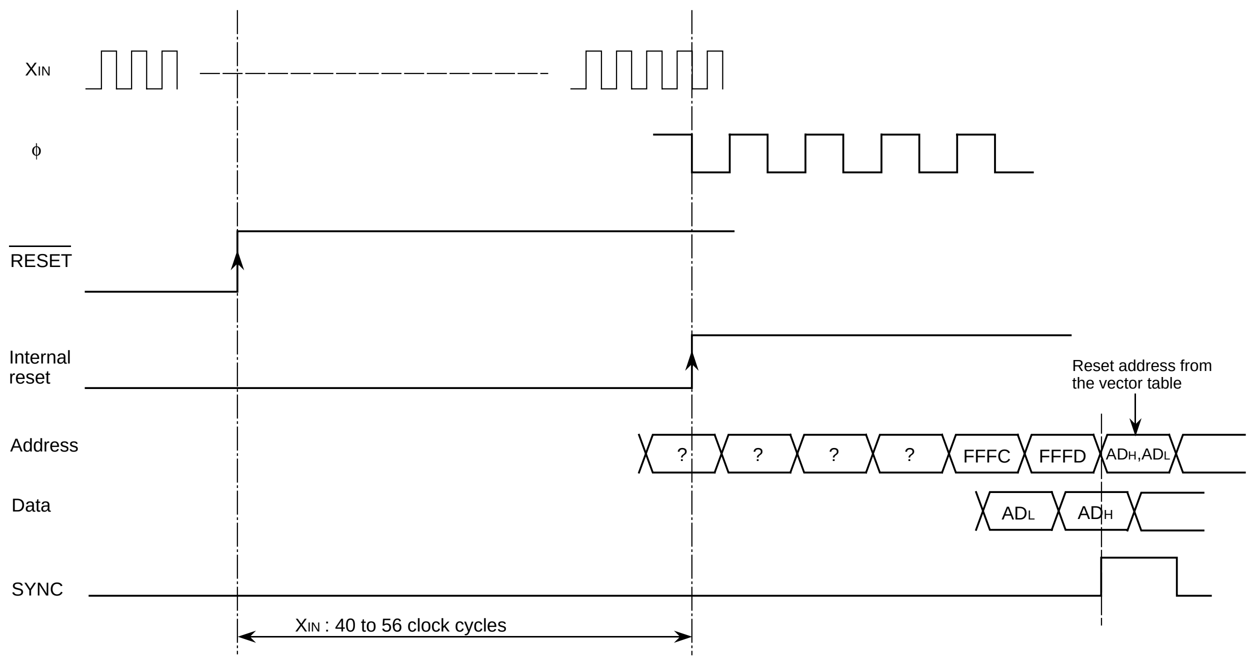



Fig. 65 Reset circuit example




**Notes 1:** The frequency relation of f($X_{IN}$) and f(φ) is f($X_{IN}$)=8 • f(φ).
**2:** The question marks (?) indicate an undefined state that depends on the previous state.

Fig. 66 Reset sequence

| No. | Register | Address | Register contents |
|---|---|---|---|
| (1) | Port P0 | $0000_{16}$ | $00_{16}$ |
| (2) | Port P0 direction register | $0001_{16}$ | $00_{16}$ |
| (3) | Port P1 | $0002_{16}$ | $00_{16}$ |
| (4) | Port P1 direction register | $0003_{16}$ | $00_{16}$ |
| (5) | Port P2 | $0004_{16}$ | $00_{16}$ |
| (6) | Port P2 direction register | $0005_{16}$ | $00_{16}$ |
| (7) | Port P3 | $0006_{16}$ | $00_{16}$ |
| (8) | Port P3 direction register | $0007_{16}$ | $00_{16}$ |
| (9) | Port P4 | $0008_{16}$ | $00_{16}$ |
| (10) | Port P4 direction register | $0009_{16}$ | $00_{16}$ |
| (11) | Port P5 | $000A_{16}$ | $00_{16}$ |
| (12) | Port P5 direction register | $000B_{16}$ | $00_{16}$ |
| (13) | Port P6 | $000C_{16}$ | $00_{16}$ |
| (14) | Port P6 direction register | $000D_{16}$ | $00_{16}$ |
| (15) | Port P7 | $000E_{16}$ | $00_{16}$ |
| (16) | Port P7 direction register | $000F_{16}$ | $00_{16}$ |
| (17) | Port P8 | $0010_{16}$ | $00_{16}$ |
| (18) | Port P8 direction register | $0011_{16}$ | $00_{16}$ |
| (19) | Port P9 | $0012_{16}$ | × 0 0 0 0 0 0 0 |
| (20) | Serial I/O3 control register 1 | $0014_{16}$ | $00_{16}$ |
| (21) | Serial I/O3 control register 2 | $0015_{16}$ | $00_{16}$ |
| (22) | Serial I/O3 control register 3 | $0016_{16}$ | $00_{16}$ |
| (23) | Serial I/O3 automatic transfer data pointer | $0017_{16}$ | $00_{16}$ |
| (24) | Serial I/O1 status register | $0019_{16}$ | $80_{16}$ |
| (25) | Serial I/O1 control register | $001A_{16}$ | $00_{16}$ |
| (26) | UART control register | $001B_{16}$ | $E0_{16}$ |
| (27) | Serial I/O2 control register | $001D_{16}$ | $00_{16}$ |
| (28) | Watchdog timer control register | $001E_{16}$ | $3F_{16}$ |
| (29) | Timer X (low-order) | $0020_{16}$ | $FF_{16}$ |
| (30) | Timer X (high-order) | $0021_{16}$ | $FF_{16}$ |
| (31) | Timer Y (low-order) | $0022_{16}$ | $FF_{16}$ |
| (32) | Timer Y (high-order) | $0023_{16}$ | $FF_{16}$ |
| (33) | Timer 1 | $0024_{16}$ | $FF_{16}$ |
| (34) | Timer 2 | $0025_{16}$ | $01_{16}$ |
| (35) | Timer 3 | $0026_{16}$ | $FF_{16}$ |
| (36) | Timer X mode register | $0027_{16}$ | $00_{16}$ |
| (37) | Timer Y mode register | $0028_{16}$ | $00_{16}$ |
| (38) | Timer 123 mode register | $0029_{16}$ | $00_{16}$ |
| (39) | Communication mode register | $002A_{16}$ | 0 0 0 0 × × × 0 |
| (40) | Transmit control register | $002B_{16}$ | $20_{16}$ |
| (41) | Transmit status register | $002C_{16}$ | $00_{16}$ |
| (42) | Receive control register | $002D_{16}$ | $10_{16}$ |
| (43) | Receive status register | $002E_{16}$ | $01_{16}$ |
| (44) | Bus interrupt source discrimination control register | $002F_{16}$ | $00_{16}$ |
| (45) | Control field selection register | $0030_{16}$ | $00_{16}$ |
| (46) | PULL UP register | $0033_{16}$ | $00_{16}$ |
| (47) | A-D control register | $0034_{16}$ | $08_{16}$ |
| (48) | Interrupt source discrimination register 2 | $0036_{16}$ | $00_{16}$ |
| (49) | Interrupt source discrimination control register 2 | $0037_{16}$ | $00_{16}$ |
| (50) | Interrupt source discrimination register 1 | $0038_{16}$ | $00_{16}$ |
| (51) | Interrupt source discrimination control register 1 | $0039_{16}$ | $00_{16}$ |
| (52) | Interrupt edge selection register | $003A_{16}$ | $00_{16}$ |
| (53) | CPU mode register | $003B_{16}$ | $48_{16}$ |
| (54) | Interrupt request register 1 | $003C_{16}$ | $00_{16}$ |
| (55) | Interrupt request register 2 | $003D_{16}$ | $00_{16}$ |
| (56) | Interrupt control register 1 | $003E_{16}$ | $00_{16}$ |
| (57) | Interrupt control register 2 | $003F_{16}$ | $00_{16}$ |
| (58) | Processor status register | (PS) | × × × × × 1 × × |
| (59) | Program counter | (PCH) | $FFFD_{16}$ contents |
| | | (PCL) | $FFFC_{16}$ contents |

**Notes:** × : Not fixed
Since the initial values for other than above-mentioned registers and RAM contents are indefinite at reset, they must be set.

Fig. 67 Internal status at reset

## CLOCK GENERATING CIRCUIT

The 3874 group has two built-in oscillation circuits. An oscillation circuit can be formed by connecting a resonator between $X_{IN}$ and $X_{OUT}$ ($X_{CIN}$ and $X_{COUT}$). Use the circuit constants in accordance with the resonator manufacturer's recommended values. No external resistor is needed between $X_{IN}$ and $X_{OUT}$ since a feed-back resistor exists on-chip. However, an external feed-back resistor is needed between $X_{CIN}$ and $X_{COUT}$.
Immediately after power on, only the $X_{IN}$ oscillation circuit starts oscillating, and $X_{CIN}$ and $X_{COUT}$ pins function as I/O ports.
When using the $X_{CIN}$ oscillation circuit, $X_{CIN}$ and $X_{COUT}$ pins' pull-up resistors need to be invarid.

### Frequency Control
### (1) Middle-speed mode

The internal clock $\phi$ is the frequency of $X_{IN}$ divided by 8. After reset, this mode is selected.

### (2) Double-speed mode

The internal clock $\phi$ is the frequency of $X_{IN}$.

### (3) High-speed mode

The internal clock $\phi$ is half the frequency of $X_{IN}$.

### (4) Low-speed mode

The internal clock $\phi$ is half the frequency of $X_{CIN}$.

### ■ Note

When switching the mode between double/middle/high-speed and low-speed, stabilize both $X_{IN}$ and $X_{CIN}$ oscillations. Sufficient time is required for the sub clock to stabilize, especially immediately after power on and at returning from stop mode. When switching the mode between double/middle/high-speed and low-speed, set the frequency on condition that $f(X_{IN}) > 3f(X_{CIN})$.
It takes the cycle number mentioned below to switch between each mode (machine cycle = cycle of internal clock $\phi$).

Double-speed mode→Except double-speed mode
  1 to 8 machine cycles
High-speed mode→Except high-speed mode
  1 to 4 machine cycles
Middle-speed mode→Except middle-speed mode
  1 machine cycle
Low-speed mode→Except low-speed mode
  1 to 4 machine cycles

The 3874 group operates in the previous mode while the mode is switched.

### (5) Low power dissipation mode

The low power consumption operation can be realized by stopping the main clock $X_{IN}$ in low-speed mode. To stop the main clock, set bit 5 of the CPU mode register to "1". When the main clock $X_{IN}$ is restarted (by setting the main clock stop bit to "0"), set sufficient time for oscillation to stabilize.
By clearing furthermore the $X_{COUT}$ drivability selection bit (b3) of the CPU mode register to "0", low power consumption operation can be realized by reducing the drivability between $X_{CIN}$ and $X_{COUT}$. At reset or during STP instruction execution this bit is set to "1" and a reduced drivability that has an easy oscillation start is set. The sub-clock $X_{CIN}$-$X_{COUT}$ oscillating circuit can no directly input clocks that are generated externally. Accordingly, make sure to cause an external resonator to oscillate.

### Oscillation Control
### (1) Stop mode

When the STP instruction is executed, the internal clock $\phi$ stops at an "H" level, and $X_{IN}$ and $X_{CIN}$ oscillators stop. The value set to the timer 1 latch and the timer 2 latch is set to timer 1 and timer 2. Either $X_{IN}$ or $X_{CIN}$ divided by 16 is input to timer 1 as count source, and the output of timer 1 is connected to timer 2. The bits of the timer 123 mode register except the timer 3 count source selection bit (b4) are cleared to "0". Set the interrupt enable bits of timer 1 and timer 2 to the disabled state ("0") before executing the STP instruction.
Oscillator restarts at reset or when an external interrupt is received, but the internal clock $\phi$ is not supplied to the CPU until timer 2 underflows. This allows time for the clock circuit oscillation to stabilize. Timer 1 latch and timer 2 latch should be set to proper values for stabilizing oscillation before executing the STP instruction.

### (2) Wait mode

If the WIT instruction is executed, the internal clock $\phi$ stops at an "H" level. The states of $X_{IN}$ and $X_{CIN}$ are the same as the state before executing the WIT instruction. The internal clock $\phi$ restarts at reset or when an interrupt is received. Since the oscillator does not stop, normal operation can be started immediately after the clock is restarted.

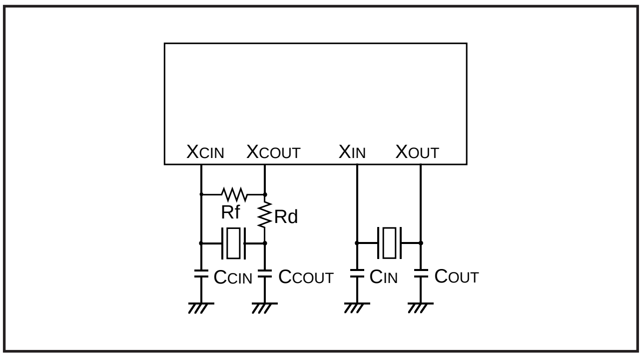



Fig. 68 Ceramic resonator circuit

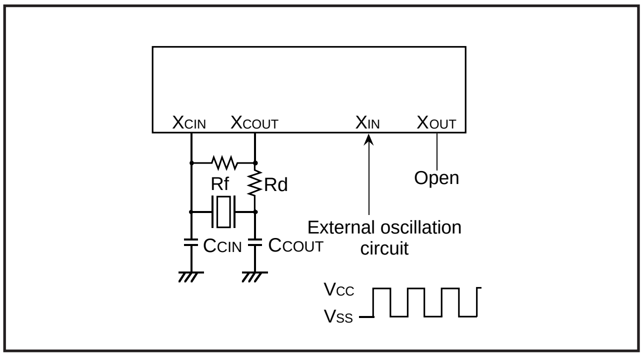



Fig. 69 External clock input circuit

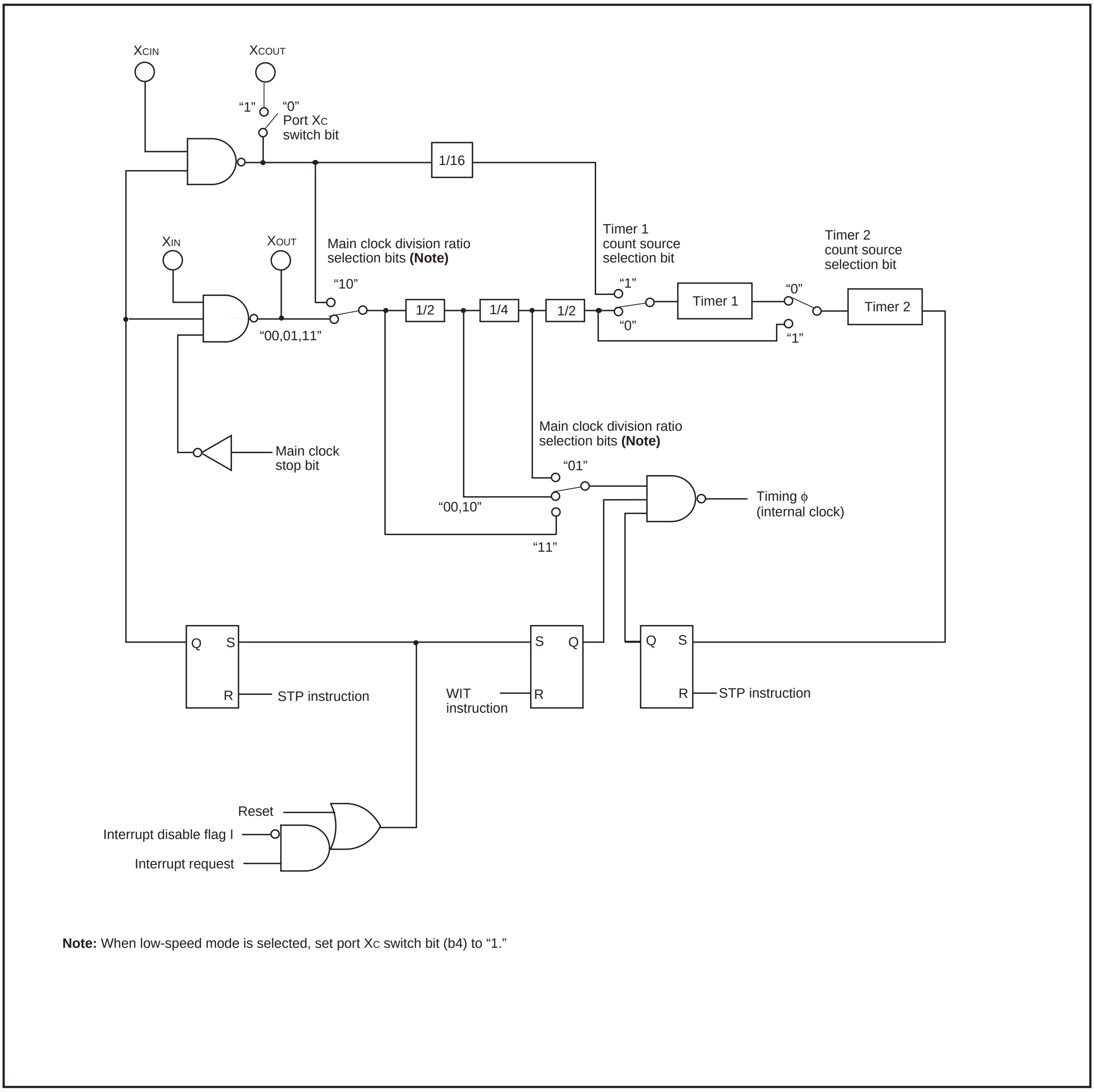
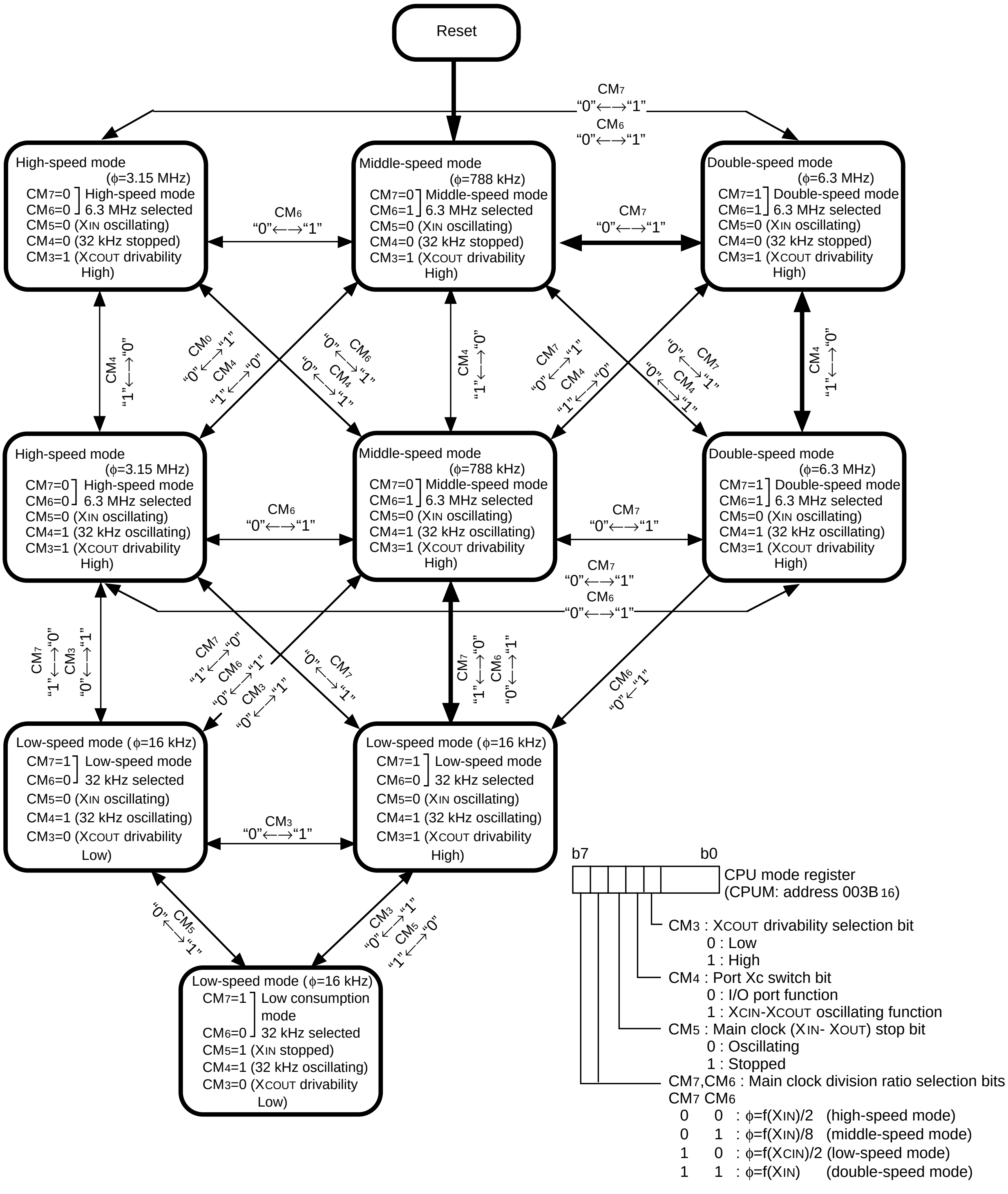



Fig. 70 System clock generating circuit block diagram




**Notes 1:** Switch the mode by the arrows shown between the mode blocks. (Do not switch between the modes directly without an arrow.)
**2:** All modes can be switched to the stop mode or the wait mode and return to the source mode when the stop mode or the wait mode is ended.
**3:** Timer operates in the wait mode.
**4:** When the stop mode is ended, wait time is generated automatically by connecting timer 1 and timer 2.
**5:** The example assumes that 6.3 MHz is being applied to the $X_{IN}$ pin and 32 kHz to the $X_{CIN}$ pin. φ indicates the internal clock.
**6:** We recommend that $X_{COUT}$ drivability selection bit is set to "1" (high) because reliance of oscillation stability is improved.

Fig. 71 State transitions of system clock

## NOTES ON PROGRAMMING

### Processor Status Register

The contents of the processor status register (PS) after a reset are undefined, except for the interrupt disable flag (I) which is “1”. After a reset, initialize flags which affect program execution. In particular, it is essential to initialize the index X mode (T) and the decimal mode (D) flags because of their effect on calculations.

### Interrupts

The contents of the interrupt request bits do not change immediately after they have been written. After writing to an interrupt request register, execute at least one instruction before performing a BBC or BBS instruction.

### Interrupt Source Determination

- Use LDM, STA, etc., instructions to clear interrupt request bits assigned to the interrupt source determination register 1, the interrupt source determination register 2, the transmit status register, or the receive status register. (Do not use read-modify-write instructions such as CLB, SEB, etc. Use the LDM or STA instruction to clear these bits.)
- Request bits of interrupt source determination registers are not automatically cleared when an interrupt occurs. After an interrupt source has been determined, and before execution of the RTI or CLI instruction, the user must clear the bit by program. (Use the LDM or STA instruction to clear.)
- The interrupt assigned to the interrupt source determination registers occur 1 instruction execution later than a normal interrupt. The maximum timing is 16 machine cycles in the MUL, DIV instructions.

### Decimal Calculations

- To calculate in decimal notation, set the decimal mode flag (D) to “1”, then execute an ADC or SBC instruction. After executing an ADC or SBC instruction, execute at least one instruction before executing a SEC, CLC, or CLD instruction.
- In decimal mode, the values of the negative (N), overflow (V), and zero (Z) flags are invalid.

### Timers

If a value n (between 0 and 255) is written to a timer latch, the frequency division ratio is 1/(n+1).

### Multiplication and Division Instructions

- The index X mode (T) and the decimal mode (D) flags do not affect the MUL and DIV instruction.
- The execution of these instructions does not change the contents of the processor status register.

### Ports

The contents of the port direction registers cannot be read. The following cannot be used:

- The data transfer instruction (LDA, etc.)
- The operation instruction when the index X mode flag (T) is “1”
- The addressing mode which uses the value of a direction register as an index
- The bit-test instruction (BBC or BBS, etc.) to a direction register
- The read-modify-write instructions (ROR, CLB, or SEB, etc.) to a direction register.

Use instructions such as LDM and STA, etc., to set the port direction registers.

### Serial I/O1

- In clock synchronous serial I/O, if the receive side is using an external clock and it is to output the $\overline{\text{SRDY1}}$ signal, set the transmit enable bit, the receive enable bit, and the $\overline{\text{SRDY}}$ output enable bit to “1”.
  Serial I/O1 continues to output the final bit from the TXD pin after transmission is completed.
- In order to stop a transmit, set the transmit enable bit to “0” (transmit disable).
  Do not set only the serial I/O1 enable bit to “0”.
- A receive operation can be stopped by either setting the receive enable bit to “0” or the serial I/O1 enable bit to “0”.
- To stop a transmit when transferring in clock synchronous serial I/O mode, set both the transmit enable bit and the receive enable bit to “0” at the same time.
- To set the serial I/O1 control register again, first set the transmit enable/receive enable bits to “0”. Next, reset the transmit/receive circuits, and, finally, reset the serial I/O1 control register.
- Note when confirming the transmit shift register completion flag and controlling the data transmit after writing a transmit data to the transmit buffer. There is a delay of 0.5 to 1.5 shift clock cycles while the transmit shift register completion flag goes from “1” to “0”.

## Serial I/O3

- When writing “1” to the serial I/O initialization bit of the serial I/O3 control register 1, serial I/O3 is enabled, but each register is not initialized. Set the value of each register by program.
- A serial I/O3 interrupt request occurs when “0” is written to the serial I/O initialization bit during an operation in automatic transfer serial I/O mode. Disable the interrupt enable bit as necessary by program.

## A-D Converter/D-A Converter

- The A-D/D-A conversion register functions as an A-D conversion register during a read and a D-A conversion during a write. Accordingly, the D-A conversion register set value cannot be read out.
- The comparator for A-D converter uses capacitive coupling amplifier whose charge will be lost if the clock frequency is too low. Therefore, make sure that f($X_{IN}$) is at least on 500 kHz during an A-D conversion.
  Do not execute the STP or WIT instruction during an A-D conversion.

## Instruction Execution Time

The instruction execution time is obtained by multiplying the frequency of the internal clock $\phi$ by the number of cycles needed to execute an instruction. The number of cycles required to execute an instruction is shown in the list of machine instructions.
The frequency of the internal clock $\phi$ is half of the $X_{IN}$ frequency.

## Data Link Layer Communication Control

- The data link layer communication control circuit stops after a reset. To restart or change modes, write “$00XXXXX1_2$” to the communication mode register. Note that bits 4 and 5 are read-only bits.
- The $P7_5/\overline{BUS_{OUT}}$ pin operates as a general-purpose pin after release from reset. As a general-purpose port, its input/output can be switched by the direction register.

## Clock Changes

- Use the LDM, STA, etc. instructions to modify the division ratio of internal system clock $\phi$. (Do not use read-modify-write instructions such as CLB, SEB, etc.)
- Do not modify the division ratio of the internal system clock until the mode has been changed. For details concerning the number of cycles necessary to change modes, refer to the clock section in the explanation of about function blocks.
- Use the LDM, STA, etc., instructions to clear interrupt request bits assigned to the interrupt source determination register 1, the interrupt source determination register 2, the transmit status register, or the receive status register. (Do not use read-modify-write instructions such as CLB, SEB, etc.)
- Before executing the CLI or RTI instruction during an interrupt processing routine, use the LDM or STA instruction to clear the interrupt request bits of interrupt source determination registers which have completed the interrupt processing.
- If switching the mode between low-speed and double-speed, switch the mode to middle/high-speed first, and then switch the mode to double-speed by program. Do not switch the mode from low-speed to double-speed directly. 1 to 4 machine cycles are required for switching from low-speed mode to other mode. Insert “clock switch timing wait” for switching the mode to middle/high-speed, and then switch the mode to double-speed.
  Table 8 lists the recommended transition process for system clock switch.
  Figure 72 shows the program example.

**Table 8 Clock switch combination**

| Recommended transition process | |
|---|---|
| Low-speed→High-speed | Middle-speed→High-speed |
| Low-speed→Middle-speed | Middle-speed→Middle-speed |
| Double-speed→High-speed | Middle-speed→Low-speed |
| Double-speed→Middle-speed | High-speed→Double-speed |
| Double-speed→Low-speed | High-speed→MIddle-speed |
| | High-speed→Low-speed |

```
Low-speed mode → Middle/High-speed mode → Double-speed mode switch
  LDM  xx, CPUM  •••Low-speed mode → Middle/High-speed mode switch
  NOP  )            Clock switch timing wait
  NOP  )            (1 to 4 machine cycles are required for switching mode.)
  LDM  yy, CPUM  •••Switch mode to double-speed

      Note: CPUM = CPU mode register (address 003B16)
```

**Fig. 72 Program example**


RENESAS
Renesas Technology Corp.

## DATA REQUIRED FOR MASK ORDERS

The following are necessary when ordering a mask ROM production:

1. Mask ROM Order Confirmation Form
2. Mark Specification Form
3. Data to be written to ROM, in EPROM form (three identical copies)

## DATA REQUIRED FOR ROM WRITING ORDERS

The following are necessary when ordering a ROM writing:

1. ROM Writing Confirmation Form
2. Mark Specification Form
3. Data to be written to ROM, in EPROM form (three identical copies)

## ROM PROGRAMMING METHOD

The PROM of the blank One Time PROM version is not tested or screened in the assembly process and following processes. To ensure proper operation after programming, the procedure shown in Figure 73 is recommended to verify programming.

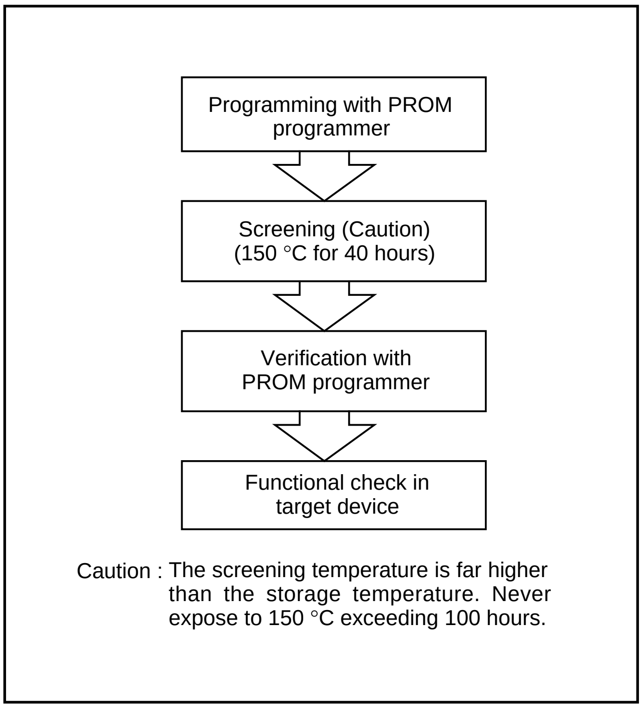



Fig. 73 Programming and testing of One Time PROM version

## ELECTRICAL CHARACTERISTICS

**Table 9 Absolute maximum ratings** (extended operating temperature version and automotive version)

| Symbol | Parameter | Conditions | Ratings | Unit |
|---|---|---|---|---|
| $V_{CC}$ | Power source voltage | All voltages are based on Vss. Output transistors are cut off. | –0.3 to 7.0 | V |
| $V_I$ | Input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $V_{REF}$ | | –0.3 to Vcc +0.3 | V |
| $V_I$ | Input voltage $\overline{RESET}$, $X_{IN}$ | | –0.3 to Vcc +0.3 | V |
| $V_I$ | Input voltage $P9_7$ | | –0.3 to Vcc +0.3 | V |
| $V_O$ | Output voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $X_{OUT}$ | | –0.3 to Vcc +0.3 | V |
| $P_d$ | Power dissipation | Ta = 25°C | 500 | mW |
| $T_{opr}$ | Operating temperature | | –40 to 85 | °C |
| $T_{stg}$ | Storage temperature | | –60 to 150 | °C |

**Table 10 Recommended operating conditions**
(extended operating temperature version and automotive version, Vcc = 3.0 to 5.5 V, Ta = –40 to 85°C, unless otherwise noted)

| Symbol | Parameter | | Power source voltage Min. | Typ. | Max. | Unit |
|---|---|---|---|---|---|---|
| $V_{CC}$ | Power source voltage | At operating data link layer communication control circuit | 4.0 | 5.0 | 5.5 | V |
| | | Double-speed mode | 4.0 | 5.0 | 5.5 | V |
| | | High-speed mode | 4.0 | 5.0 | 5.5 | V |
| | | Middle-speed mode | 3.0 | 5.0 | 5.5 | V |
| | | Low-speed mode | 3.0 | 5.0 | 5.5 | V |
| $V_{SS}$ | Power source voltage | | | 0 | | V |
| $V_{REF}$ | Analog reference voltage (when A-D converter is used) | | 2.0 | | $V_{CC}$ | V |
| | Analog reference voltage (when D-A converter is used) | | 3.0 | | $V_{CC}$ | V |
| $AV_{SS}$ | Analog power source voltage | | | 0 | | V |
| $V_{IA}$ | Analog input voltage $AN_0$ to $AN_7$ | | $AV_{SS}$ | | $V_{CC}$ | V |
| $V_{IH}$ | "H" input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_7$ | | $0.8V_{CC}$ | | $V_{CC}$ | V |
| $V_{IH}$ | "H" input voltage $\overline{RESET}$, $X_{IN}$ | | $0.8V_{CC}$ | | $V_{CC}$ | V |
| $V_{IL}$ | "L" input voltage $P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$, $P9_7$ | | 0 | | $0.2V_{CC}$ | V |
| $V_{IL}$ | "L" input voltage $\overline{RESET}$ | | 0 | | $0.2V_{CC}$ | V |
| $V_{IL}$ | "L" input voltage $X_{IN}$ | | 0 | | $0.16V_{CC}$ | V |

**Table 11 Recommended operating conditions (1)**

(extended operating temperature version and automotive version, Vcc = 3.0 to 5.5 V, Ta = –40 to 85°C, unless otherwise noted)

| Symbol | Parameter | Limits | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| $\Sigma I_{OH(peak)}$ | "H" total peak output current **(Note 1)**<br>$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P8_0$–$P8_7$ | | | –80 | mA |
| $\Sigma I_{OH(peak)}$ | "H" total peak output current<br>$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$ | | | –80 | mA |
| $\Sigma I_{OL(peak)}$ | "L" total peak output current<br>$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P8_0$–$P8_7$ | | | 80 | mA |
| $\Sigma I_{OL(peak)}$ | "L" total peak output current<br>$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$ | | | 80 | mA |
| $\Sigma I_{OH(avg)}$ | "H" total average output current **(Note 1)**<br>$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P8_0$–$P8_7$ | | | –40 | mA |
| $\Sigma I_{OH(avg)}$ | "H" total average output current<br>$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$ | | | –40 | mA |
| $\Sigma I_{OL(avg)}$ | "L" total average output current<br>$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P8_0$–$P8_7$ | | | 40 | mA |
| $\Sigma I_{OL(avg)}$ | "L" total average output current<br>$P4_0$–$P4_7$, $P5_0$–$P5_7$, $P6_0$–$P6_7$, $P7_0$–$P7_7$ | | | 40 | mA |
| $I_{OH(peak)}$ | "H" peak output current **(Note 2)**<br>$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$,<br>$P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$ | | | –10 | mA |
| $I_{OL(peak)}$ | "L" peak output current<br>$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$,<br>$P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$ | | | 10 | mA |
| $I_{OH(avg)}$ | "H" average output current **(Note 3)**<br>$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$,<br>$P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$ | | | –5.0 | mA |
| $I_{OL(avg)}$ | "L" average output current<br>$P0_0$–$P0_7$, $P1_0$–$P1_7$, $P2_0$–$P2_7$, $P3_0$–$P3_7$, $P4_0$–$P4_7$, $P5_0$–$P5_7$,<br>$P6_0$–$P6_7$, $P7_0$–$P7_7$, $P8_0$–$P8_7$ | | | 5.0 | mA |
| $f(CNTR_0)$<br>$f(CNTR_1)$ | Timer X, timer Y input oscillation frequency<br>(at duty cycle of 50%) | | | 2.5 | MHz |
| $f(X_{IN})$ | Main clock input oscillation frequency **(Note 4)** | | | 6.4 | MHz |
| $f(X_{CIN})$ | Sub-clock input oscillation frequency **(Notes 4, 5)** | | 32.768 | 50 | kHz |

**Notes 1:** The total output current is the sum of all the currents flowing through all the applicable ports. The total average current is an average value measured over 100 ms. The total peak current is the peak value of all the currents.

**2:** The peak output current is the peak current flowing in each port.

**3:** The average output current $I_{OL(avg)}$, $I_{OH(avg)}$ in an average value measured over 100 ms.

**4:** Choose an external oscillator which ensures no warps in the oscillation waveform as well as sufficient amplitude for the main clock oscillation circuit. Use according to the manufacturer's recommended conditions.

**Table 12 Recommended operating conditions (2) (when ROM/PROM size is 60 Kbytes)**

(Vcc = 3.0 to 5.5 V, Ta = –40 to 85°C, unless otherwise noted)

| Symbol | Parameter | | Limits | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $f(X_{IN})$ | Main clock input oscillation frequency | High-speed mode/Middle-speed mode | | | 6.4 | MHz |
| | | Double-speed mode ($4.0 \leq V_{CC} < 4.5V$) | | | $2.8V_{CC}-6.2$ | MHz |
| | | Double-speed mode ($4.5 \leq V_{CC} \leq 5.5V$) | | | 6.4 | MHz |

**Note 5:** When using the microcomputer in the low-speed mode, set the sub-clock input oscillation frequency on condition that $f(X_{CIN}) < f(X_{IN})/3$.

**Table 13 Electrical characteristics**
(extended operating temperature version and automotive version, Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –40 to 85°C, unless otherwise noted)

| Symbol | Parameter | Test conditions | Limits | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| $V_{OH}$ | "H" output voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P80–P87 **(Note)** | $I_{OH}$ = –10 mA $V_{CC}$ = 4.0–5.5 V | $V_{CC}$–2.0 | | | V |
| | | $I_{OH}$ = –1 mA $V_{CC}$ = 3.0–5.5 V | $V_{CC}$–1.0 | | | V |
| $V_{OL}$ | "L" output voltage P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70–P77, P80–P87 | $I_{OL}$ = 10 mA $V_{CC}$ = 4.0–5.5 V | | | 2.0 | V |
| | | $I_{OL}$ = 1.0 mA $V_{CC}$ = 3.0–5.5 V | | | 1.0 | V |
| $V_{T+}$–$V_{T-}$ | Hysteresis INT0–INT5, ADT, CNTR0, CNTR1 | | | 0.5 | | V |
| $V_{T+}$–$V_{T-}$ | Hysteresis RxD, SCLK1, SIN2, SCLK2, P20–P27 | Valid hysteresis only when these pins is used as the function | | 0.5 | | V |
| $V_{T+}$–$V_{T-}$ | Hysteresis $\overline{RESET}$ | | | 0.5 | | V |
| $I_{IH}$ | "H" input current P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70–P77, P80–P87 | $V_I = V_{CC}$ | | | 5.0 | µA |
| $I_{IH}$ | "H" input current P97 | $V_I = V_{CC}$ | | | 5.0 | µA |
| $I_{IH}$ | "H" input current $\overline{RESET}$ | $V_I = V_{CC}$ | | | 5.0 | µA |
| $I_{IH}$ | "H" input current XIN | $V_I = V_{CC}$ | | 4.0 | | µA |
| $I_{IL}$ | "L" input current P00–P07, P10–P17, P20–P27, P30–P37, P40–P47, P50–P57, P60–P67, P70–P77, P80–P87 | $V_I = V_{SS}$ | | | –5.0 | µA |
| $I_{IL}$ | "L" input current P97 | $V_I = V_{SS}$ | | | –5.0 | µA |
| $I_{IL}$ | "L" input current $\overline{RESET}$ | $V_I = V_{SS}$ | | | –5.0 | µA |
| $I_{IL}$ | "L" input current XIN | $V_I = V_{SS}$ | | –4.0 | | µA |
| $V_{RAM}$ | RAM hold voltage | When clock stopped | 2.0 | | 5.5 | V |

**Note:** When P45/TxD, P71/SOUT2, and P72/SCLK2 are CMOS output states (when not P-channel output disable states)

**Table 14 Electrical characteristics**
(extended operating temperature version and automotive version, Vcc = 4.0 to 5.5 V, Vss = 0 V, Ta = –40 to 85°C, unless otherwise noted)

| Symbol | Parameter | Test conditions | | Limits Min. | Limits Typ. | Limits Max. | Unit |
|---|---|---|---|---|---|---|---|
| $I_{CC}$ | Power source current | Double-speed mode, at operating data link layer communication control circuit<br>$f(X_{IN})$ = 6.29 MHz<br>$f(X_{CIN})$ = 32 kHz<br>Output transistors “off”<br>During A-D conversion | | | 18.0 | 24.0 | mA |
| | | Double-speed mode, at stopping data link layer communication control circuit<br>$f(X_{IN})$ = 6.29 MHz<br>$f(X_{CIN})$ = 32 kHz<br>Output transistors “off”<br>During A-D conversion | | | 12.0 | 18.0 | mA |
| | | Double-speed mode, at stopping data link layer communication control circuit<br>$f(X_{IN})$ = 6.29 MHz (in WIT state)<br>$f(X_{CIN})$ = 32 kHz<br>Output transistors “off”<br>During A-D conversion | | | 2.0 | 3.5 | mA |
| | | High-speed mode, at operating data link layer communication control circuit<br>$f(X_{IN})$ = 6.29 MHz<br>$f(X_{CIN})$ = 32 kHz<br>Output transistors “off”<br>During A-D conversion | | | 12.0 | 19.0 | mA |
| | | High-speed mode, at stopping data link layer communication control circuit<br>$f(X_{IN})$ = 6.29 MHz<br>$f(X_{CIN})$ = 32 kHz<br>Output transistors “off”<br>During A-D conversion | | | 8.0 | 12.0 | mA |
| | | High-speed mode, at stopping data link layer communication control circuit<br>$f(X_{IN})$ = 6.29 MHz (in WIT state)<br>$f(X_{CIN})$ = 32 kHz<br>Output transistors “off”<br>During A-D conversion | | | 2.0 | 3.5 | mA |
| | | Low-speed mode ($V_{CC}$ = 3.0 V)<br>$f(X_{IN})$ = stopped<br>$f(X_{CIN})$ = 32 kHz<br>Low power dissipation mode ($CM_5$ = 0)<br>Output transistors “off” | | | 60 | 200 | µA |
| | | Low-speed mode ($V_{CC}$ = 3.0 V)<br>$f(X_{IN})$ = stopped<br>$f(X_{CIN})$ = 32 kHz (in WIT state)<br>Low power dissipation mode ($CM_5$ = 0)<br>Output transistors “off” | | | 20 | 40 | µA |
| | | All oscillation stopped (in STP state)<br>Output transistors “off” | Ta = 25°C **(Note)** | | 0.1 | 1.0 | µA |
| | | | Ta = 85°C **(Note)** | | | 10 | µA |

**Note:** The A-D conversion is inactive. (The A–D conversion complete.) $V_{REF}$ current is not included.

**Table 15 A-D converter characteristics**

(extended operating temperature version and automotive version, $V_{CC}$ = 4.0 to 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0 V, $V_{REF}$ = 2.0 V to $V_{CC}$, Ta = –40 to 85°C, unless otherwise noted)

| Symbol | Parameter | Test conditions | Limits | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| – | Resolution | | | | 8 | Bits |
| – | Absolute accuracy (excluding quantization error) | | | ±1 | ±2.5 | LSB |
| $t_{CONV}$ | Conversion time | | | | 50 | tc(ϕ) |
| $R_{LADDER}$ | Ladder resistor | | 12 | 35 | 100 | kΩ |
| $I_{VREF}$ | Reference power source input current | $V_{REF}$ = 5.0 V | 50 | 150 | 200 | µA |
| $I_{I(AD)}$ | Analog port input current | | | 0.5 | 5.0 | µA |

**Table 16 D-A converter characteristics**

(extended operating temperature version and automotive version, $V_{CC}$ = 4.0 to 5.5 V, $V_{SS}$ = $AV_{SS}$ = 0 V, $V_{REF}$ = 2.0 V to $V_{CC}$, Ta = –40 to 85°C, unless otherwise noted)

| Symbol | Parameter | Test conditions | Limits | | | Unit |
|---|---|---|---|---|---|---|
| | | | Min. | Typ. | Max. | |
| – | Resolution | | | | 8 | Bits |
| – | Absolute accuracy | | | | 1.0 | % |
| tsu | Setting time | | | | 3.0 | µs |
| RO | Output resistor | | 1 | 2.5 | 4.0 | kΩ |
| $I_{VREF}$ | Reference power source input current | | | | 3.2 | mA |

## TIMING REQUIREMENTS

**Table 17 Timing requirements**

(extended operating temperature version and automotive version, $V_{CC}$ = 4.0 to 5.5 V, $V_{SS}$ = 0 V, Ta = –40 to 85°C, unless otherwise noted)

| Symbol | Parameter | Limits | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| $t_{W}(\overline{RESET})$ | Reset input “L” pulse width | 2 | | | µs |
| $t_{C}(X_{IN})$ | External clock input cycle time | 159 | | | ns |
| $t_{WH}(X_{IN})$ | External clock input “H” pulse width | 63 | | | ns |
| $t_{WL}(X_{IN})$ | External clock input “L” pulse width | 63 | | | ns |
| $t_{C}(CNTR)$ | $CNTR_0$, $CNTR_1$ input cycle time | 200 | | | ns |
| $t_{WH}(CNTR)$ | $CNTR_0$, $CNTR_1$ input “H” pulse width | 80 | | | ns |
| $t_{WH}(INT)$ | $INT_0$ to $INT_5$ input “H” pulse width | 80 | | | ns |
| $t_{WL}(CNTR)$ | $CNTR_0$, $CNTR_1$ input “L” pulse width | 80 | | | ns |
| $t_{WL}(INT)$ | $INT_0$ to $INT_5$ input “L” pulse width | 80 | | | ns |
| $t_{C}(S_{CLK1})$ | Serial I/O1 clock input cycle time **(Note)** | 800 | | | ns |
| $t_{C}(S_{CLK2})$ | Serial I/O2 clock input cycle time | 1000 | | | ns |
| $t_{C}(S_{CLK3})$ | Serial I/O3 clock input cycle time | 1000 | | | ns |
| $t_{WH}(S_{CLK1})$ | Serial I/O1 clock input “H” pulse width **(Note)** | 370 | | | ns |
| $t_{WH}(S_{CLK2})$ | Serial I/O2 clock input “H” pulse width | 400 | | | ns |
| $t_{WH}(S_{CLK3})$ | Serial I/O3 clock input “H” pulse width | 400 | | | ns |
| $t_{WL}(S_{CLK1})$ | Serial I/O1 clock input “L” pulse width **(Note)** | 370 | | | ns |
| $t_{WL}(S_{CLK2})$ | Serial I/O2 clock input “L” pulse width | 400 | | | ns |
| $t_{WL}(S_{CLK3})$ | Serial I/O3 clock input “L” pulse width | 400 | | | ns |
| $t_{su}(R_xD\text{-}S_{CLK1})$ | Serial I/O1 input setup time | 220 | | | ns |
| $t_{su}(S_{IN2}\text{-}S_{CLK2})$ | Serial I/O2 input setup time | 200 | | | ns |
| $t_{su}(R_{IN3}\text{-}S_{CLK3})$ | Serial I/O3 input setup time | 200 | | | ns |
| $t_{h}(S_{CLK1}\text{-}R_xD)$ | Serial I/O1 input hold time | 100 | | | ns |
| $t_{h}(S_{CLK2}\text{-}S_{IN2})$ | Serial I/O2 input hold time | 200 | | | ns |
| $t_{h}(S_{CLK3}\text{-}S_{IN3})$ | Serial I/O3 input hold time | 200 | | | ns |

**Note :** When bit 6 of address $001A_{16}$ is “1” (clock synchronous).
Divide this value by four when bit 6 of address $001A_{16}$ is “0” (UART).

**Table 18 Switching characteristics**
(extended operating temperature version and automotive version, $V_{CC}$ = 4.0 to 5.5 V, $V_{SS}$ = 0 V, Ta = –40 to 85°C, unless otherwise noted)

| Symbol | Parameter | Limits | | | Unit |
|---|---|---|---|---|---|
| | | Min. | Typ. | Max. | |
| $t_{WH}$ ($S_{CLK1}$) | Serial I/O1 clock output “H” pulse width | $t_C(S_{CLK1})/2–30$ | | | ns |
| $t_{WH}$ ($S_{CLK2}$) | Serial I/O2 clock output “H” pulse width **(Note 1)** | $t_C(S_{CLK2})/2–30$ | | | ns |
| $t_{WH}$ ($S_{CLK3}$) | Serial I/O3 clock output “H” pulse width **(Note 5)** | $t_C(S_{CLK3})/2–30$ | | | ns |
| $t_{WL}$ ($S_{CLK1}$) | Serial I/O1 clock output “L” pulse width | $t_C(S_{CLK1})/2–30$ | | | ns |
| $t_{WL}$ ($S_{CLK2}$) | Serial I/O2 clock output “L” pulse width **(Note 1)** | $t_C(S_{CLK2})/2–30$ | | | ns |
| $t_{WL}$ ($S_{CLK3}$) | Serial I/O3 clock output “L” pulse width **(Note 5)** | $t_C(S_{CLK3})/2–30$ | | | ns |
| $t_d$ ($S_{CLK1}$-$T_XD$) | Serial I/O1 output delay time **(Note 3)** | | | 140 | ns |
| $t_d$ ($S_{CLK2}$-$S_{OUT2}$) | Serial I/O2 output delay time **(Notes 1, 2)** | | | 140 | ns |
| $t_d$ ($S_{CLK3}$-$S_{OUT3}$) | Serial I/O3 output delay time **(Notes 5, 6)** | | | 140 | ns |
| $t_V$ ($S_{CLK1}$-$T_XD$) | Serial I/O1 output valid time **(Note 3)** | –30 | | | ns |
| $t_V$ ($S_{CLK2}$-$S_{OUT2}$) | Serial I/O2 output valid time **(Notes 1, 2)** | 0 | | | ns |
| $t_V$ ($S_{CLK3}$-$S_{OUT3}$) | Serial I/O3 output valid time **(Notes 5, 6)** | 0 | | | ns |
| $t_r$ ($S_{CLK1}$) | Serial I/O1 clock output rising time | | 10 | 30 | ns |
| $t_f$ ($S_{CLK1}$) | Serial I/O1 clock output falling time | | 10 | 30 | ns |
| $t_r$ ($S_{CLK2}$) | Serial I/O2 clock output rising time **(Note 1)** | | 10 | 30 | ns |
| $t_f$ ($S_{CLK2}$) | Serial I/O2 clock output falling time **(Note 1)** | | 10 | 30 | ns |
| $t_r$ ($S_{CLK3}$) | Serial I/O3 clock output rising time **(Note 5)** | | 10 | 30 | ns |
| $t_f$ ($S_{CLK3}$) | Serial I/O3 clock output falling time **(Note 5)** | | 10 | 30 | ns |
| $t_r$ (CMOS) | CMOS output rising time **(Note 4)** | | 10 | 30 | ns |
| $t_f$ (CMOS) | CMOS output falling time **(Note 4)** | | 10 | 30 | ns |

**Notes** **1:** When $P7_2/S_{CLK2}$ is CMOS output.
**2:** When $P7_1/S_{OUT2}$ is CMOS output.
**3:** When $P4_5/T_XD$ is CMOS output.
**4:** The $X_{OUT}$ pin is excluded.
**5:** When $P8_4/S_{CLK3}$ is CMOS output.
**6:** When $P8_2/S_{OUT3}$ is CMOS output.

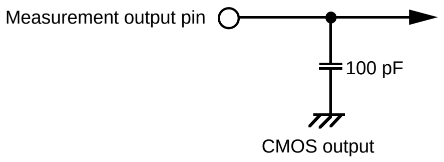



Fig. 74 Circuit for measuring output switching characteristics

## Timing diagram

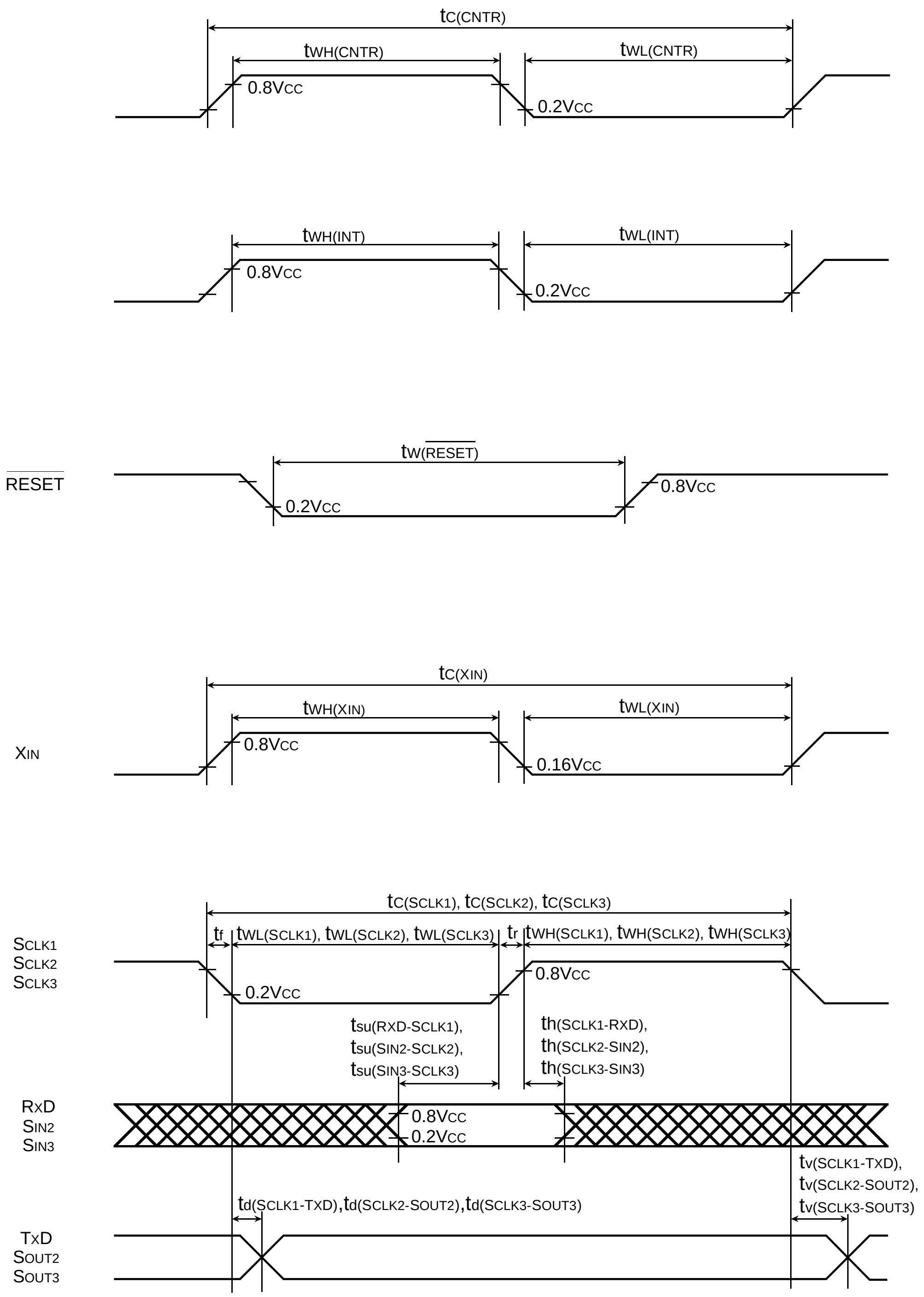



Fig. 75 Timing diagram (in single-chip mode)

GZZ-SH52-76B<84A0>

| Mask ROM number | |
|---|---|

# 740 FAMILY MASK ROM CONFIRMATION FORM
# SINGLE-CHIP MICROCOMPUTER M38747M4T-XXXGP
# MITSUBISHI ELECTRIC

| Receipt | Date: | |
|---|---|---|
| | Section head signature | Supervisor signature |
| | | |

Note : Please fill in all items marked ❋.

| ❋ Customer | Company name | TEL ( ) | Issuance signature | Submitted by | Supervisor |
|---|---|---|---|---|---|
| | Date issued | Date: | | | |

❋ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

| Checksum code for entire EPROM | | | | | (hexadecimal notation) |
|---|---|---|---|---|---|

| Sub ROM number of data link layer communication control circuit | | | | | | | | |
|---|---|---|---|---|---|---|---|---|

EPROM type (indicate the type used)

| ☐ 27512 EPROM address | | ☐ 27101 EPROM address | |
|---|---|---|---|
| $0000_{16}$ – $000F_{16}$ | Product name ASCII code : 'M38747M4T–' | $0000_{16}$ – $000F_{16}$ | Product name ASCII code : 'M38747M4T–' |
| $0010_{16}$ – $001F_{16}$ | Sub ROM number ASCII code | $0010_{16}$ – $001F_{16}$ | Sub ROM number ASCII code |
| $0020_{16}$ – $C07F_{16}$ | (shaded) | $0020_{16}$ – $C07F_{16}$ | (shaded) |
| $C080_{16}$ – $FFFD_{16}$ | Data ROM 16K-130 bytes | $C080_{16}$ – $FFFD_{16}$ | Data ROM 16K-130 bytes |
| $FFFE_{16}$ – $FFFF_{16}$ | (shaded) | $FFFE_{16}$ – $1FFFF_{16}$ | (shaded) |

In the address space of the microcomputer, the internal ROM area is from address $C080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "$FF_{16}$".
(2) The ASCII codes of the product name "M38747M4T–" must be entered in addresses $0000_{16}$ to $0009_{16}$. And set the data "$FF_{16}$" in addresses $000A_{16}$ to $000F_{16}$. ASCII codes and addresses are listed to the next page.
(3) Addresses $0010_{16}$ to $001F_{16}$ are ASCII codes reserved area of Sub ROM number for the data link layer communication control circuit. Write ASCII codes of Sub ROM number for the data link layer communication control circuit, which has been used at developing the submitted ROM, to addresses $0010_{16}$ to $001F_{16}$ of EPROM certainly. Refer to ASCII codes of the next page at writing.

(1/3)

GZZ-SH52-76B<84A0>

| Mask ROM number | |
|---|---|

# 740 FAMILY MASK ROM CONFIRMATION FORM
# SINGLE-CHIP MICROCOMPUTER M38747M4T-XXXGP
# MITSUBISHI ELECTRIC

| Address | | Address | |
|---|---|---|---|
| $0000_{16}$ | ‘M’ = $4D_{16}$ | $0008_{16}$ | ‘ T ’ =$54_{16}$ |
| $0001_{16}$ | ‘3’ = $33_{16}$ | $0009_{16}$ | ‘ - ’ =$2D_{16}$ |
| $0002_{16}$ | ‘8’ = $38_{16}$ | $000A_{16}$ | $FF_{16}$ |
| $0003_{16}$ | ‘7’ = $37_{16}$ | $000B_{16}$ | $FF_{16}$ |
| $0004_{16}$ | ‘4’ = $34_{16}$ | $000C_{16}$ | $FF_{16}$ |
| $0005_{16}$ | ‘7’ = $37_{16}$ | $000D_{16}$ | $FF_{16}$ |
| $0006_{16}$ | ‘M’ = $4D_{16}$ | $000E_{16}$ | $FF_{16}$ |
| $0007_{16}$ | ‘4’ = $34_{16}$ | $000F_{16}$ | $FF_{16}$ |

ASCII codes

| | | | | |
|---|---|---|---|---|
| ‘ 0 ’ =$30_{16}$ | ‘ 8 ’ =$38_{16}$ | ‘ G ’ =$38_{16}$ | ‘ R ’ =$52_{16}$ | ‘ Z ’ =$5A_{16}$ |
| ‘ 1 ’ =$31_{16}$ | ‘ 9 ’ =$39_{16}$ | ‘ H ’ =$39_{16}$ | ‘ S ’ =$53_{16}$ | |
| ‘ 2 ’ =$32_{16}$ | ‘ A ’ =$41_{16}$ | ‘ K ’ =$4B_{16}$ | ‘ T ’ =$54_{16}$ | |
| ‘ 3 ’ =$33_{16}$ | ‘ B ’ =$42_{16}$ | ‘ L ’ =$4C_{16}$ | ‘ U ’ =$55_{16}$ | |
| ‘ 4 ’ =$34_{16}$ | ‘ C ’ =$43_{16}$ | ‘ M’ =$4D_{16}$ | ‘ V ’ =$56_{16}$ | |
| ‘ 5 ’ =$35_{16}$ | ‘ D ’ =$44_{16}$ | ‘ N ’ =$4E_{16}$ | ‘ W ’=$57_{16}$ | |
| ‘ 6 ’ =$36_{16}$ | ‘ E ’ =$45_{16}$ | ‘ P ’ =$50_{16}$ | ‘ X ’ =$58_{16}$ | |
| ‘ 7 ’ =$37_{16}$ | ‘ F ’ =$46_{16}$ | ‘ Q ’ =$51_{16}$ | ‘ Y ’ =$59_{16}$ | |

We recommend the use of the following pseudo-command to set the start address of the assembier source program because ASCII codes of the product name are written to addresses $0000_{16}$ to $0009_{16}$ of EPROM. ASCII codes of sub ROM number are written to addresses $0010_{16}$ to $0017_{16}$ by using the pseudo-command in the same way.

| EPROM type | 27512 | 27101 |
|---|---|---|
| The pseudo-command | *=Δ$0000<br>.BYTEΔ‘M38747M4T–’ | *=Δ$0000<br>.BYTEΔ‘M38747M4T–’ |

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

❅ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (80P6S) and attach it to the mask ROM confirmation form.

❅ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the $X_{IN}$-$X_{OUT}$ oscillator?

☐ Ceramic resonator ☐ Quartz crystal
☐ External clock input ☐ Other ( )

At what frequency? $f(X_{IN})$ = ☐ MHz

(2) How will you use the $X_{CIN}$-$X_{COUT}$ oscillator?

☐ Ceramic resonator ☐ Quartz crystal
☐ External clock input ☐ Other ( )
☐ Not use (Use for $P4_0$,$P4_1$)

At what frequency? $f(X_{CIN})$ = ☐ MHz

(3) Which clock division ratio will you use? (possible to select plural)

☐ $\phi = X_{IN}$ (Double-speed mode) ☐ $\phi = X_{IN}/2$ (High-speed mode)
☐ $\phi = X_{IN}/8$ (Middle-speed mode) ☐ $\phi = X_{CIN}/2$ (Low-speed mode)

(2/3)



RENESAS
Renesas Technology Corp.

GZZ-SH52-76B<84A0>

| Mask ROM number | |
|---|---|

# 740 FAMILY MASK ROM CONFIRMATION FORM
# SINGLE-CHIP MICROCOMPUTER M38747M4T-XXXGP
# MITSUBISHI ELECTRIC

(4) Will you use the data link layer communication control circuit?

☐ Yes
☐ No

❋ 4. Comments

(3/3)

GZZ-SH52-77B<84A0>

| Mask ROM number | |
|---|---|

# 740 FAMILY MASK ROM CONFIRMATION FORM
# SINGLE-CHIP MICROCOMPUTER M38747M6T-XXXGP
# MITSUBISHI ELECTRIC

| Receipt | Date: | |
|---|---|---|
| | Section head signature | Supervisor signature |
| | | |

Note : Please fill in all items marked ❈.

| | | | | Issuance signature | Submitted by | Supervisor |
|---|---|---|---|---|---|---|
| ❈ | Customer | Company name | TEL ( ) | | | |
| | | Date issued | Date: | | | |

❈ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Checksum code for entire EPROM | | | | | (hexadecimal notation)

Sub ROM number of data link layer communication control circuit | | | | | | | | |

EPROM type (indicate the type used)

| ☐ 27512 | | ☐ 27101 | |
|---|---|---|---|
| EPROM address | | EPROM address | |
| $0000_{16}$ – $000F_{16}$ | Product name ASCII code : 'M38747M6T–' | $0000_{16}$ – $000F_{16}$ | Product name ASCII code : 'M38747M6T–' |
| $0010_{16}$ – $001F_{16}$ | Sub ROM number ASCII code | $0010_{16}$ – $001F_{16}$ | Sub ROM number ASCII code |
| $0020_{16}$ – $A07F_{16}$ | (shaded) | $0020_{16}$ – $A07F_{16}$ | (shaded) |
| $A080_{16}$ – $FFFD_{16}$ | Data ROM 24K-130 bytes | $A080_{16}$ – $FFFD_{16}$ | Data ROM 24K-130 bytes |
| $FFFE_{16}$ – $FFFF_{16}$ | (shaded) | $FFFE_{16}$ – $1FFFF_{16}$ | (shaded) |

In the address space of the microcomputer, the internal ROM area is from address $A080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "$FF_{16}$".
(2) The ASCII codes of the product name "M38747M6T–" must be entered in addresses $0000_{16}$ to $0009_{16}$. And set the data "$FF_{16}$" in addresses $000A_{16}$ to $000F_{16}$. ASCII codes and addresses are listed to the next page.
(3) Addresses $0010_{16}$ to $001F_{16}$ are ASCII codes reserved area of Sub ROM number for the data link layer communication control circuit. Write ASCII codes of Sub ROM number for the data link layer communication control circuit, which has been used at developing the submitted ROM, to addresses $0010_{16}$ to $001F_{16}$ of EPROM certainly. Refer to ASCII codes of the next page at writing.

(1/3)


RENESAS
Renesas Technology Corp.

GZZ-SH52-77B<84A0>

| Mask ROM number | |
|---|---|

## 740 FAMILY MASK ROM CONFIRMATION FORM
## SINGLE-CHIP MICROCOMPUTER M38747M6T-XXXGP
## MITSUBISHI ELECTRIC

| Address | | Address | |
|---|---|---|---|
| 0000$_{16}$ | ‘M’ = 4D$_{16}$ | 0008$_{16}$ | ‘ T ’ =54$_{16}$ |
| 0001$_{16}$ | ‘3’ = 33$_{16}$ | 0009$_{16}$ | ‘ - ’ =2D$_{16}$ |
| 0002$_{16}$ | ‘8’ = 38$_{16}$ | 000A$_{16}$ | FF$_{16}$ |
| 0003$_{16}$ | ‘7’ = 37$_{16}$ | 000B$_{16}$ | FF$_{16}$ |
| 0004$_{16}$ | ‘4’ = 34$_{16}$ | 000C$_{16}$ | FF$_{16}$ |
| 0005$_{16}$ | ‘7’ = 37$_{16}$ | 000D$_{16}$ | FF$_{16}$ |
| 0006$_{16}$ | ‘M’ = 4D$_{16}$ | 000E$_{16}$ | FF$_{16}$ |
| 0007$_{16}$ | ‘6’ = 36$_{16}$ | 000F$_{16}$ | FF$_{16}$ |

ASCII codes

| | | | | |
|---|---|---|---|---|
| ‘ 0 ’ =30$_{16}$ | ‘ 8 ’ =38$_{16}$ | ‘ G ’ =38$_{16}$ | ‘ R ’ =52$_{16}$ | ‘ Z ’ =5A$_{16}$ |
| ‘ 1 ’ =31$_{16}$ | ‘ 9 ’ =39$_{16}$ | ‘ H ’ =39$_{16}$ | ‘ S ’ =53$_{16}$ | |
| ‘ 2 ’ =32$_{16}$ | ‘ A ’ =41$_{16}$ | ‘ K ’ =4B$_{16}$ | ‘ T ’ =54$_{16}$ | |
| ‘ 3 ’ =33$_{16}$ | ‘ B ’ =42$_{16}$ | ‘ L ’ =4C$_{16}$ | ‘ U ’ =55$_{16}$ | |
| ‘ 4 ’ =34$_{16}$ | ‘ C ’ =43$_{16}$ | ‘ M’ =4D$_{16}$ | ‘ V ’ =56$_{16}$ | |
| ‘ 5 ’ =35$_{16}$ | ‘ D ’ =44$_{16}$ | ‘ N ’ =4E$_{16}$ | ‘ W ’=57$_{16}$ | |
| ‘ 6 ’ =36$_{16}$ | ‘ E ’ =45$_{16}$ | ‘ P ’ =50$_{16}$ | ‘ X ’ =58$_{16}$ | |
| ‘ 7 ’ =37$_{16}$ | ‘ F ’ =46$_{16}$ | ‘ Q ’ =51$_{16}$ | ‘ Y ’ =59$_{16}$ | |

We recommend the use of the following pseudo-command to set the start address of the assembier source program because ASCII codes of the product name are written to addresses 0000$_{16}$ to 0009$_{16}$ of EPROM. ASCII codes of sub ROM number are written to addresses 0010$_{16}$ to 0017$_{16}$ by using the pseudo-command in the same way.

| EPROM type | 27512 | 27101 |
|---|---|---|
| The pseudo-command | *=Δ$0000<br>.BYTEΔ‘M38747M6T–’ | *=Δ$0000<br>.BYTEΔ‘M38747M6T–’ |

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

❋ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (80P6S) and attach it to the mask ROM confirmation form.

❋ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the $X_{IN}$-$X_{OUT}$ oscillator?

☐ Ceramic resonator ☐ Quartz crystal
☐ External clock input ☐ Other ( )

At what frequency? f($X_{IN}$) = ☐ MHz

(2) How will you use the $X_{CIN}$-$X_{COUT}$ oscillator?

☐ Ceramic resonator ☐ Quartz crystal
☐ External clock input ☐ Other ( )
☐ Not use (Use for $P4_0$,$P4_1$)

At what frequency? f($X_{CIN}$) = ☐ MHz

(3) Which clock division ratio will you use? (possible to select plural)

☐ $\phi = X_{IN}$ (Double-speed mode) ☐ $\phi = X_{IN}/2$ (High-speed mode)
☐ $\phi = X_{IN}/8$ (Middle-speed mode) ☐ $\phi = X_{CIN}/2$ (Low-speed mode)

(2/3)

GZZ-SH52-77B<84A0>

| Mask ROM number | |
|---|---|

## 740 FAMILY MASK ROM CONFIRMATION FORM
## SINGLE-CHIP MICROCOMPUTER M38747M6T-XXXGP
## MITSUBISHI ELECTRIC

(4) Will you use the data link layer communication control circuit?

☐ Yes
☐ No

❋ 4. Comments

(3/3)

GZZ-SH52-75B<84A0>

| Mask ROM number | |
|---|---|

# 740 FAMILY MASK ROM CONFIRMATION FORM
# SINGLE-CHIP MICROCOMPUTER M38747MCT-XXXGP
# MITSUBISHI ELECTRIC

| Receipt | Date: | |
|---|---|---|
| | Section head signature | Supervisor signature |
| | | |

Note : Please fill in all items marked ❋.

| ❋ Customer | Company name | TEL ( ) | Issuance signature | Submitted by | Supervisor |
|---|---|---|---|---|---|
| | Date issued | Date: | | | |

❋ 1. Confirmation

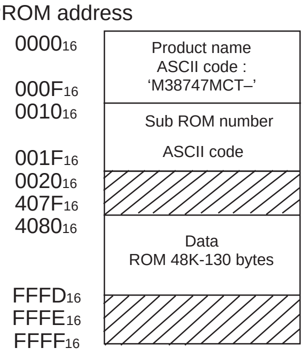
Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce masks based on this data. We shall assume the responsibility for errors only if the mask ROM data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

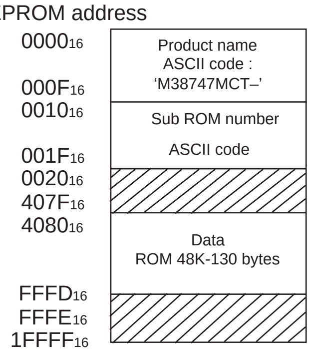
Checksum code for entire EPROM ☐☐☐☐ (hexadecimal notation)

Sub ROM number of data link layer communication control circuit ☐☐☐☐☐☐☐☐

EPROM type (indicate the type used)

| ☐ 27512 | ☐ 27101 |
|---|---|
| EPROM address | EPROM address |




In the address space of the microcomputer, the internal ROM area is from address $4080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "$FF_{16}$".
(2) The ASCII codes of the product name "M38747MCT–" must be entered in addresses $0000_{16}$ to $0009_{16}$. And set the data "$FF_{16}$" in addresses $000A_{16}$ to $000F_{16}$. ASCII codes and addresses are listed to the next page.
(3) Addresses $0010_{16}$ to $001F_{16}$ are ASCII codes reserved area of Sub ROM number for the data link layer communication control circuit. Write ASCII codes of Sub ROM number for the data link layer communication control circuit, which has been used at developing the submitted ROM, to addresses $0010_{16}$ to $001F_{16}$ of EPROM certainly. Refer to ASCII codes of the next page at writing.

(1/3)

GZZ-SH52-75B<84A0>

| Mask ROM number | |
|---|---|

## 740 FAMILY MASK ROM CONFIRMATION FORM
## SINGLE-CHIP MICROCOMPUTER M38747MCT-XXXGP
## MITSUBISHI ELECTRIC

| Address | | Address | |
|---|---|---|---|
| $0000_{16}$ | ‘M’ = $4D_{16}$ | $0008_{16}$ | ‘ T ’ = $54_{16}$ |
| $0001_{16}$ | ‘3’ = $33_{16}$ | $0009_{16}$ | ‘ - ’ = $2D_{16}$ |
| $0002_{16}$ | ‘8’ = $38_{16}$ | $000A_{16}$ | $FF_{16}$ |
| $0003_{16}$ | ‘7’ = $37_{16}$ | $000B_{16}$ | $FF_{16}$ |
| $0004_{16}$ | ‘4’ = $34_{16}$ | $000C_{16}$ | $FF_{16}$ |
| $0005_{16}$ | ‘7’ = $37_{16}$ | $000D_{16}$ | $FF_{16}$ |
| $0006_{16}$ | ‘M’ = $4D_{16}$ | $000E_{16}$ | $FF_{16}$ |
| $0007_{16}$ | ‘C’ = $43_{16}$ | $000F_{16}$ | $FF_{16}$ |

ASCII codes

| | | | | |
|---|---|---|---|---|
| ‘ 0 ’ = $30_{16}$ | ‘ 8 ’ = $38_{16}$ | ‘ G ’ = $38_{16}$ | ‘ R ’ = $52_{16}$ | ‘ Z ’ = $5A_{16}$ |
| ‘ 1 ’ = $31_{16}$ | ‘ 9 ’ = $39_{16}$ | ‘ H ’ = $39_{16}$ | ‘ S ’ = $53_{16}$ | |
| ‘ 2 ’ = $32_{16}$ | ‘ A ’ = $41_{16}$ | ‘ K ’ = $4B_{16}$ | ‘ T ’ = $54_{16}$ | |
| ‘ 3 ’ = $33_{16}$ | ‘ B ’ = $42_{16}$ | ‘ L ’ = $4C_{16}$ | ‘ U ’ = $55_{16}$ | |
| ‘ 4 ’ = $34_{16}$ | ‘ C ’ = $43_{16}$ | ‘ M’ = $4D_{16}$ | ‘ V ’ = $56_{16}$ | |
| ‘ 5 ’ = $35_{16}$ | ‘ D ’ = $44_{16}$ | ‘ N ’ = $4E_{16}$ | ‘ W ’ = $57_{16}$ | |
| ‘ 6 ’ = $36_{16}$ | ‘ E ’ = $45_{16}$ | ‘ P ’ = $50_{16}$ | ‘ X ’ = $58_{16}$ | |
| ‘ 7 ’ = $37_{16}$ | ‘ F ’ = $46_{16}$ | ‘ Q ’ = $51_{16}$ | ‘ Y ’ = $59_{16}$ | |

We recommend the use of the following pseudo-command to set the start address of the assembier source program because ASCII codes of the product name are written to addresses $0000_{16}$ to $0009_{16}$ of EPROM. ASCII codes of sub ROM number are written to addresses $0010_{16}$ to $0017_{16}$ by using the pseudo-command in the same way.

| EPROM type | 27512 | 27101 |
|---|---|---|
| The pseudo-command | *=Δ$0000<br>.BYTEΔ‘M38747MCT–’ | *=Δ$0000<br>.BYTEΔ‘M38747MCT–’ |

Note : If the name of the product written to the EPROMs does not match the name of the mask confirmation form, the ROM will not be processed.

❋ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (80P6S) and attach it to the mask ROM confirmation form.

❋ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the $X_{IN}$-$X_{OUT}$ oscillator?

- ☐ Ceramic resonator ☐ Quartz crystal
- ☐ External clock input ☐ Other ( )

At what frequency? $f(X_{IN})$ = ☐ MHz

(2) How will you use the $X_{CIN}$-$X_{COUT}$ oscillator?

- ☐ Ceramic resonator ☐ Quartz crystal
- ☐ External clock input ☐ Other ( )
- ☐ Not use (Use for $P4_0$,$P4_1$)

At what frequency? $f(X_{CIN})$ = ☐ MHz

(3) Which clock division ratio will you use? (possible to select plural)

- ☐ $\phi = X_{IN}$ (Double-speed mode) ☐ $\phi = X_{IN}/2$ (High-speed mode)
- ☐ $\phi = X_{IN}/8$ (Middle-speed mode) ☐ $\phi = X_{CIN}/2$ (Low-speed mode)

(2/3)

GZZ-SH52-75B<84A0>

| Mask ROM number | |
|---|---|

# 740 FAMILY MASK ROM CONFIRMATION FORM
# SINGLE-CHIP MICROCOMPUTER M38747MCT-XXXGP
# MITSUBISHI ELECTRIC

(4) Will you use the data link layer communication control circuit?

☐ Yes
☐ No

❋ 4. Comments

(3/3)

GZZ-SH52-78B<84A0>

| ROM number | |
|---|---|

# 740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
# SINGLE-CHIP MICROCOMPUTER M38749EFT-XXXGP
# MITSUBISHI ELECTRIC

| Receipt | Date: | |
|---|---|---|
| | Section head signature | Supervisor signature |
| | | |

Note : Please fill in all items marked ❈.

| ❈ Customer | Company name | TEL ( ) | Issuance signature | Submitted by | Supervisor |
|---|---|---|---|---|---|
| | Date issued | Date: | | | |

❈ 1. Confirmation

Specify the name of the product being ordered and the type of EPROMs submitted.
Three EPROMs are required for each pattern.
If at least two of the three sets of EPROMs submitted contain identical data, we will produce ROM programming based on this data. We shall assume the responsibility for errors only if the ROM programming data on the products we produce differs from this data. Thus, extreme care must be taken to verify the data in the submitted EPROMs.

Checksum code for entire EPROM [ ][ ][ ][ ] (hexadecimal notation)

Sub ROM number of data link layer communication control circuit [ ][ ][ ][ ][ ][ ][ ][ ]

EPROM type (indicate the type used)

| ☐ 27512 | | ☐ 27101 | |
|---|---|---|---|
| EPROM address | | EPROM address | |
| $0000_{16}$ – $000F_{16}$ | Product name ASCII code : 'M38749EFT–' | $0000_{16}$ – $000F_{16}$ | Product name ASCII code : 'M38749EFT–' |
| $0010_{16}$ – $001F_{16}$ | Sub ROM number ASCII code | $0010_{16}$ – $001F_{16}$ | Sub ROM number ASCII code |
| $0020_{16}$ – $107F_{16}$ | (shaded) | $0020_{16}$ – $107F_{16}$ | (shaded) |
| $1080_{16}$ – $FFFD_{16}$ | Data ROM 60K-130 bytes | $1080_{16}$ – $FFFD_{16}$ | Data ROM 60K-130 bytes |
| $FFFE_{16}$ – $FFFF_{16}$ | (shaded) | $FFFE_{16}$ – $1FFFF_{16}$ | (shaded) |

In the address space of the microcomputer, the internal ROM area is from address $1080_{16}$ to $FFFD_{16}$. The reset vector is stored in addresses $FFFC_{16}$ and $FFFD_{16}$.

(1) Set the data in the unused area (the shaded area of the diagram) to "$FF_{16}$".

(2) The ASCII codes of the product name "M38749EFT–" must be entered in addresses $0000_{16}$ to $0009_{16}$. And set the data "$FF_{16}$" in addresses $000A_{16}$ to $000F_{16}$. The ASCII codes and addresses are listed to the next page.

(3) Addresses $0010_{16}$ to $001F_{16}$ are ASCII codes reserved area of Sub ROM number for the data link layer communication control circuit. Write ASCII codes of Sub ROM number for the data link layer communication control circuit, which has been used at developing the submitted ROM, to addresses $0010_{16}$ to $001F_{16}$ of EPROM certainly. Refer to ASCII codes of the next page at writing.

(1/3)

GZZ-SH52-78B<84A0>

| ROM number | |
|---|---|

# 740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
# SINGLE-CHIP MICROCOMPUTER M38749EFT-XXXGP
# MITSUBISHI ELECTRIC

| Address | | Address | |
|---|---|---|---|
| $0000_{16}$ | ‘M’ = $4D_{16}$ | $0008_{16}$ | ‘ T ’ = $54_{16}$ |
| $0001_{16}$ | ‘3’ = $33_{16}$ | $0009_{16}$ | ‘ - ’ = $2D_{16}$ |
| $0002_{16}$ | ‘8’ = $38_{16}$ | $000A_{16}$ | $FF_{16}$ |
| $0003_{16}$ | ‘7’ = $37_{16}$ | $000B_{16}$ | $FF_{16}$ |
| $0004_{16}$ | ‘4’ = $34_{16}$ | $000C_{16}$ | $FF_{16}$ |
| $0005_{16}$ | ‘9’ = $39_{16}$ | $000D_{16}$ | $FF_{16}$ |
| $0006_{16}$ | ‘E’ = $45_{16}$ | $000E_{16}$ | $FF_{16}$ |
| $0007_{16}$ | ‘F’ = $46_{16}$ | $000F_{16}$ | $FF_{16}$ |

ASCII codes

| | | | | |
|---|---|---|---|---|
| ‘ 0 ’ = $30_{16}$ | ‘ 8 ’ = $38_{16}$ | ‘ G ’ = $38_{16}$ | ‘ R ’ = $52_{16}$ | ‘ Z ’ = $5A_{16}$ |
| ‘ 1 ’ = $31_{16}$ | ‘ 9 ’ = $39_{16}$ | ‘ H ’ = $39_{16}$ | ‘ S ’ = $53_{16}$ | |
| ‘ 2 ’ = $32_{16}$ | ‘ A ’ = $41_{16}$ | ‘ K ’ = $4B_{16}$ | ‘ T ’ = $54_{16}$ | |
| ‘ 3 ’ = $33_{16}$ | ‘ B ’ = $42_{16}$ | ‘ L ’ = $4C_{16}$ | ‘ U ’ = $55_{16}$ | |
| ‘ 4 ’ = $34_{16}$ | ‘ C ’ = $43_{16}$ | ‘ M’ = $4D_{16}$ | ‘ V ’ = $56_{16}$ | |
| ‘ 5 ’ = $35_{16}$ | ‘ D ’ = $44_{16}$ | ‘ N ’ = $4E_{16}$ | ‘ W ’ = $57_{16}$ | |
| ‘ 6 ’ = $36_{16}$ | ‘ E ’ = $45_{16}$ | ‘ P ’ = $50_{16}$ | ‘ X ’ = $58_{16}$ | |
| ‘ 7 ’ = $37_{16}$ | ‘ F ’ = $46_{16}$ | ‘ Q ’ = $51_{16}$ | ‘ Y ’ = $59_{16}$ | |

We recommend the use of the following pseudo-command to set the start address of the assembier source program because ASCII codes of the product name are written to addresses $0000_{16}$ to $0009_{16}$ of EPROM. ASCII codes of sub ROM number are written to addresses $0010_{16}$ to $0017_{16}$ by using the pseudo-command in the same way.

| EPROM type | 27512 | 27101 |
|---|---|---|
| The pseudo-command | *=Δ$0000<br>.BYTEΔ‘M38749EFT–’ | *=Δ$0000<br>.BYTEΔ‘M38749EFT–’ |

Note : If the name of the product written to the EPROMs does not match the name of the ROM programming confirmation form, the ROM will not be processed.

❅ 2. Mark specification

Mark specification must be submitted using the correct form for the package being ordered. Fill out the appropriate mark specification form (80P6S) and attach it to the ROM programming confirmation form.

❅ 3. Usage conditions

Please answer the following questions about usage for use in our product inspection :

(1) How will you use the $X_{IN}$-$X_{OUT}$ oscillator?

☐ Ceramic resonator　☐ Quartz crystal

☐ External clock input　☐ Other (　　　　)

At what frequency?　f($X_{IN}$) = [　　　　] MHz

(2) How will you use the $X_{CIN}$-$X_{COUT}$ oscillator?

☐ Ceramic resonator　☐ Quartz crystal

☐ External clock input　☐ Other (　　　　)

☐ Not use (Use for $P4_0$,$P4_1$)

At what frequency?　f($X_{CIN}$) = [　　　　] MHz

(3) Which clock division ratio will you use? (possible to select plural)

☐ $\phi$ = $X_{IN}$ (Double-speed mode)　☐ $\phi$ = $X_{IN}$ /2 (High-speed mode)

☐ $\phi$ = $X_{IN}$ /8 (Middle-speed mode)　☐ $\phi$ = $X_{CIN}$ /2 (Low-speed mode)

(2/3)

GZZ-SH52-78B<84A0>

| ROM number | |
|---|---|

# 740 FAMILY ROM PROGRAMMING CONFIRMATION FORM
# SINGLE-CHIP MICROCOMPUTER M38749EFT-XXXGP
# MITSUBISHI ELECTRIC

(4) Will you use the data link layer communication control circuit?

☐ Yes
☐ No

❋ 4. Comments

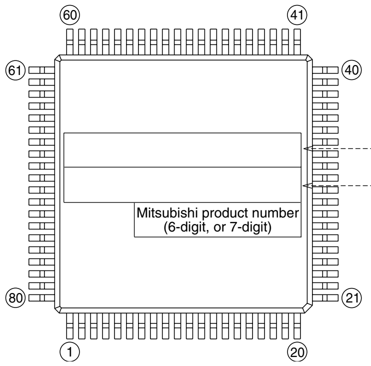
(3/3)

## 80P6S (80-PIN QFP) MARK SPECIFICATION FORM
## 80P6D, 80P6Q (80-PIN Fine-pitch QFP)

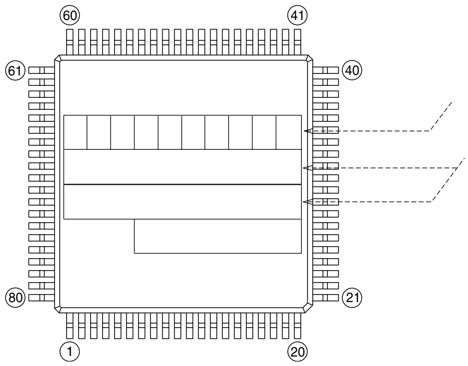
Mitsubishi IC catalog name

Please choose one of the marking types below (A, B, C), and enter the Mitsubishi IC catalog name and the special mark (if needed).

A. Standard Mitsubishi Mark



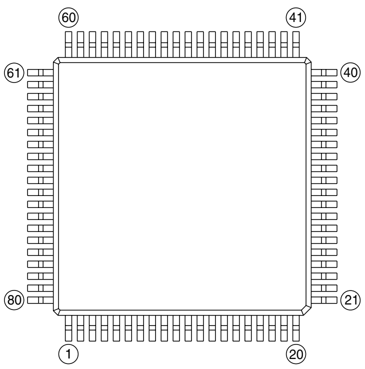

B. Customer's Parts Number + Mitsubishi IC Catalog Name




Customer's Parts Number
Note : The fonts and size of characters are standard Mitsubishi type.
Mitsubishi IC catalog name

Notes 1 : The mark field should be written right aligned.
2 : The fonts and size of characters are standard Mitsubishi type.
3 : Customer's parts number can be up to 10 alphanumeric characters for capital letters, hyphens, commas, periods and so on.

C. Special Mark Required



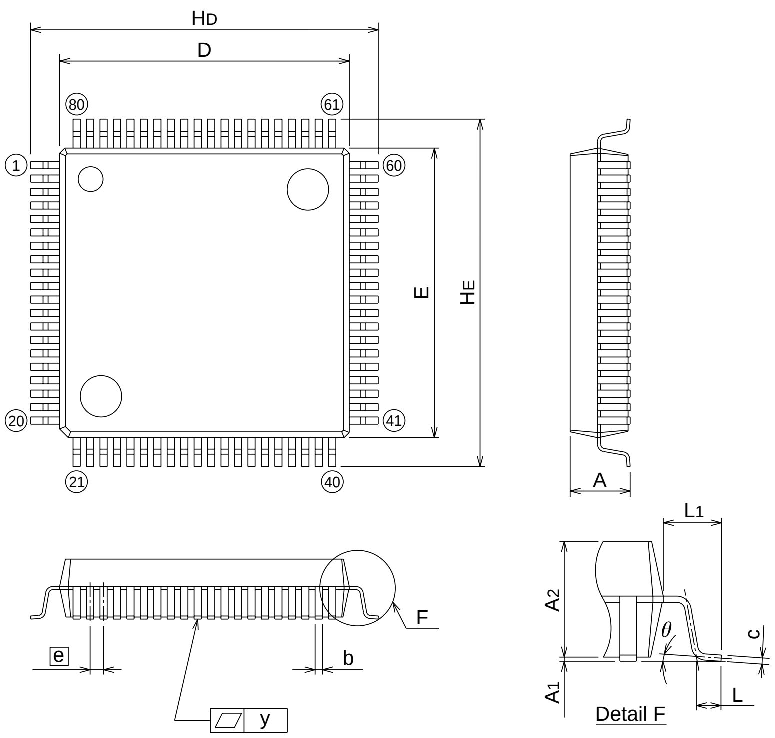

Notes 1 : If Special mark is to be printed, indicate the desired layout of the mark in the left figure. The layout will be duplicated technically as close as possible. Mitsubishi product number (6-digit, or 7-digit) and Mask ROM number (3-digit) are always marked for sorting the products.
2 : If special character fonts (e.g., customer's trade mark logo) must be used in Special Mark, check the box below.
For the new special character fonts, a clean font original (ideally logo drawing) must be submitted.

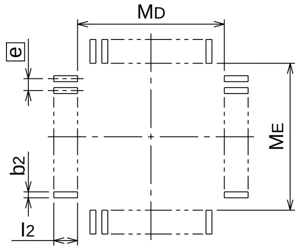
Special character fonts required

## 80P6S-A

**Plastic 80pin 14×14mm body QFP**

| EIAJ Package Code | JEDEC Code | Weight(g) | Lead Material |
|---|---|---|---|
| QFP80-P-1414-0.65 | | 1.11 | Alloy 42 |



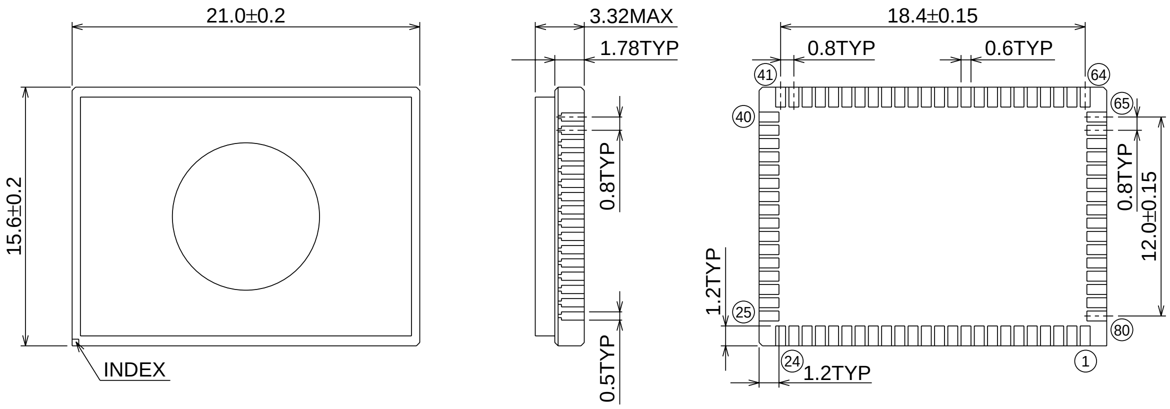

Recommended Mount Pad

| Symbol | Dimension in Millimeters | | |
|---|---|---|---|
| | Min | Nom | Max |
| A | – | – | 3.05 |
| $A_1$ | 0 | 0.1 | 0.2 |
| $A_2$ | – | 2.8 | – |
| b | 0.25 | 0.3 | 0.4 |
| c | 0.13 | 0.15 | 0.2 |
| D | 13.8 | 14.0 | 14.2 |
| E | 13.8 | 14.0 | 14.2 |
| e | – | 0.65 | – |
| $H_D$ | 16.5 | 16.8 | 17.1 |
| $H_E$ | 16.5 | 16.8 | 17.1 |
| L | 0.4 | 0.6 | 0.8 |
| $L_1$ | – | 1.4 | – |
| y | – | – | 0.1 |
| $\theta$ | 0° | – | 10° |
| $b_2$ | – | 0.35 | – |
| $I_2$ | 1.3 | – | – |
| $M_D$ | – | 14.6 | – |
| $M_E$ | – | 14.6 | – |

## 80D0

**Glass seal 80pin QFN**

| EIAJ Package Code | JEDEC Code | Weight(g) |
|---|---|---|
| – | – | |



Renesas Technology Corp.
Nippon Bldg.,6-2,Otemachi 2-chome,Chiyoda-ku,Tokyo,100-0004 Japan



RENESAS
Renesas Technology Corp.

| REVISION DESCRIPTION LIST | 3874 GROUP DATA SHEET |
|---|---|

| Rev. No. | Revision Description | Rev. date |
|---|---|---|
| 1.0 | First Edition | 980602 |