PRELIMINARY PRODUCT INFORMATION

NEC

MOS INTEGRATED CIRCUIT

# μPD789166Y,789167Y,789176Y,789177Y

## 8-BIT SINGLE-CHIP MICROCONTROLLER

**The μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y are μPD789167Y and μPD789177Y sub-series products of the 78K/0S series.**

These microcontrollers feature an 8-bit CPU, I/O ports, timers, a serial interface, A/D converters, and interrupt control circuits.

In addition, a flash memory product (μPD78F9177Y) that can operate within the same voltage range as the masked ROM models, and a range of related development tools are being developed.

**The functions of these microcontrollers are described in the following user's manuals. Refer to these manuals when designing a system based on any of these microcontrollers.**

**μPD789177Y Sub-Series User's Manual : To be created**
**78K/0S Series User's Manual, Instruction: U11047E**

### FEATURES

- ROM and RAM sizes

| Item / Product name | Program memory | Data memory |
|---|---|---|
| | ROM | Internal high-speed RAM |
| μPD789166Y | 16 Kbytes | 512 × 8 bits |
| μPD789167Y | 24 Kbytes | |
| μPD789176Y | 16 Kbytes | |
| μPD789177Y | 24 Kbytes | |

- Variable minimum instruction execution time: From high-speed (0.4 μs: With the main system clock running at 5.0 MHz) to very low-speed (122 μs: With the subsystem clock running at 32.768 kHz)
- Eight A/D converters with 8-bit resolution (for μPD789166Y and μPD789167Y)
- Eight A/D converters with an 10-bit resolution (for μPD789176Y and μPD789177Y)
- 31 I/O ports
- Two serial interface channels:
  - Switchable between three-wire serial I/O and UART modes
  - System management bus (SMB)
- Six timers:
  - 16-bit timer counter
  - Three 8-bit timer/event counters
  - Clock timer
  - Watchdog timer
- 16-bit multiplier
- Power supply voltage: $V_{DD}$ = 1.8 to 5.5 V



Document No. U13216EJ1V0PM00 (1st edition)
Date Published March 1998 J CP(K)
Printed in Japan

NEC

U13216EJ1V0X100

July 1998 NS CP(K)

# Supplement

[Document Name]
μPD789166Y, 789167Y, 789176Y, 789177Y PRELIMINARY PRODUCT INFORMATION

[Document No., Date Published]
U13216EJ1V0PM00 (1st edition), March 1998 J CP(K)

| | before change | after change |
|---|---|---|
| Product Name | μPD789167Y,789177Y Subseries | μPD789167,789177 Subseries |
| | μPD789166Y, 789167Y | μPD789166, 789167 |
| | μPD789176Y, 789177Y | μPD789176, 789177 |
| | μPD78F9177Y | μPD78F9177 |
| Package | 42-pin plastic shrink DIP<br>44-pin plastic QFP<br>48-pin plastic TQFP | 44-pin plastic QFP |
| Part Number | μPD789166YGB-xxx-3BS | μPD789166GB-xxx-3BS-MTX |
| | μPD789167YGB-xxx-3BS | μPD789167GB-xxx-3BS-MTX |
| | μPD789176YGB-xxx-3BS | μPD789176GB-xxx-3BS-MTX |
| | μPD789177YGB-xxx-3BS | μPD789177GB-xxx-3BS-MTX |
| | μPD78F9177YGB-3BS | μPD78F9177GB-3BS-MTX |
| SMB | On chip | Not provided |

## APPLICATIONS

Power windows, keyless entries, battery management units, side air bags, etc.

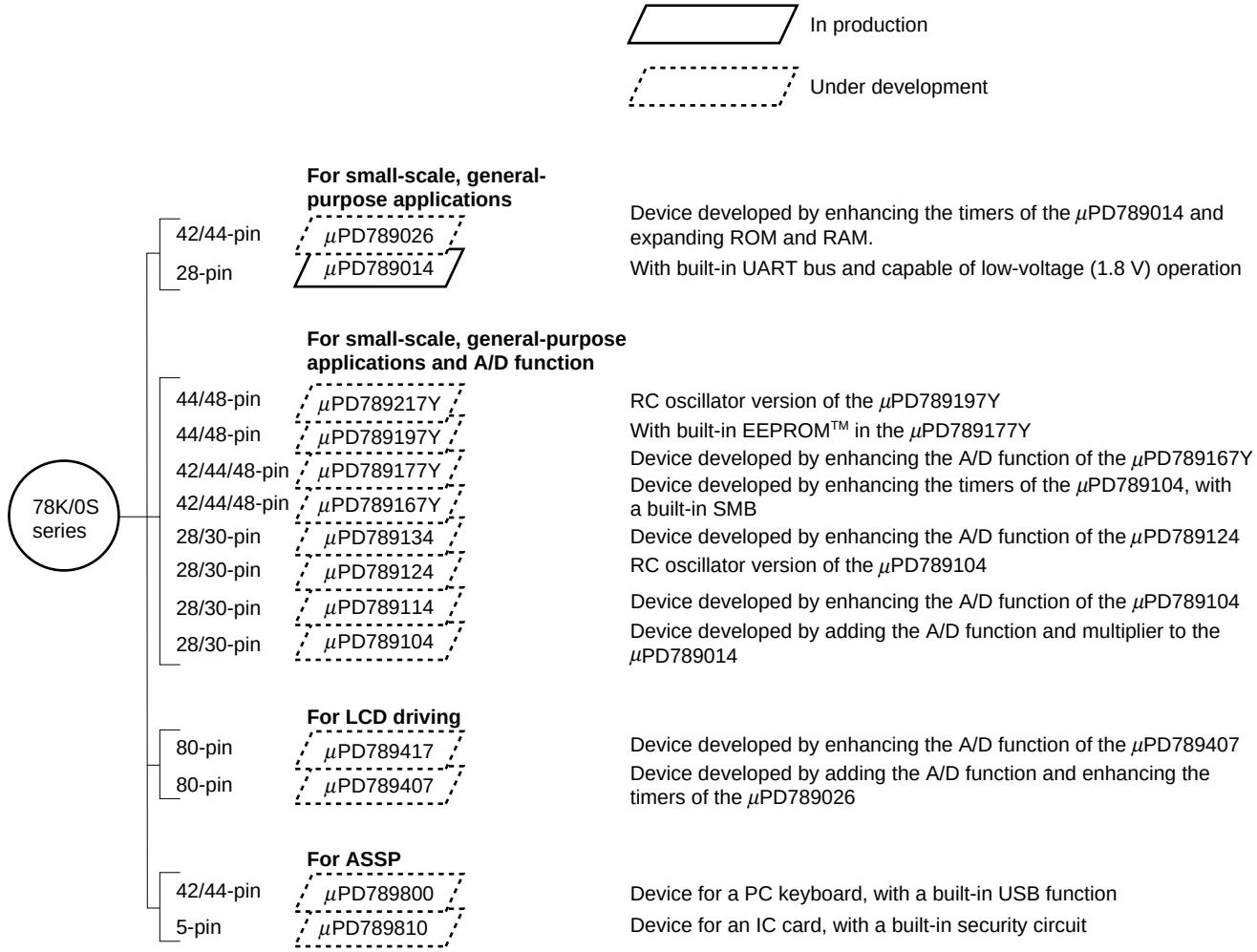
## ORDERING INFORMATION

| Part number | Package |
|---|---|
| μPD789166YCU-×××  | 42-pin plastic shrink DIP (600 mil) |
| μPD789166YGB-×××-3BS | 44-pin plastic QFP (10 × 10 mm) |
| μPD789166YGA-×××-9EU | 48-pin plastic TQFP (fine pitch) (7 × 7 mm) |
| μPD789167YCU-×××  | 42-pin plastic shrink DIP (600 mil) |
| μPD789167YGB-×××-3BS | 44-pin plastic QFP (10 × 10 mm) |
| μPD789167YGA-×××-9EU | 48-pin plastic TQFP (fine pitch) (7 × 7 mm) |
| μPD789176YCU-×××  | 42-pin plastic shrink DIP (600 mil) |
| μPD789176YGB-×××-3BS | 44-pin plastic QFP (10 × 10 mm) |
| μPD789176YGA-×××-9EU | 48-pin plastic TQFP (fine pitch) (7 × 7 mm) |
| μPD789177YCU-×××  | 42-pin plastic shrink DIP (600 mil) |
| μPD789177YGB-×××-3BS | 44-pin plastic QFP (10 × 10 mm) |
| μPD789177YGA-×××-9EU | 48-pin plastic TQFP (fine pitch) (7 × 7 mm) |

**Remark** ××× indicates ROM code suffix.

## 78K/0S SERIES DEVELOPMENT

The 78K/0S series products are shown below. The sub-series names are indicated in frames.



The following table lists the major differences in functions between the sub-series.

| Sub-series \ Function | | ROM size | Timer | | | | 8-bit A/D | 10-bit A/D | Serial interface | I/O | Minimum $V_{DD}$ value | Remarks |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | 8-bit | 16-bit | Clock | WDT | | | | | | |
| Small-scale, general purpose applications | μPD789026 | 4 K-16 K | 1 ch | 1 ch | – | 1 ch | – | – | 1 ch (UART: 1 ch) | 34 pins | 1.8 V | – |
| | μPD789014 | 2 K-4 K | 2 ch | – | | | | | | 22 pins | | |
| Small-scale, general-purpose applications and A/D function | μPD789217Y | 16 K-24 K | 3 ch | 1 ch | 1 ch | 1 ch | – | 8 ch | 2 ch [UART: 1 ch, SMB: 1 ch] | 31 pins | 1.8 V | RC-oscillator version, with built-in EEPROM |
| | μPD789197Y | | | | | | | | | | | With built-in EEPROM |
| | μPD789177Y | | | | | | | | | | | – |
| | μPD789167Y | | | | | | 8 ch | – | | | | |
| | μPD789134 | 2 K-8 K | 1 ch | | – | | – | 4 ch | 1 ch (UART: 1 ch) | 20 pins | | RC-oscillator version |
| | μPD789124 | | | | | | 4 ch | – | | | | |
| | μPD789114 | | | | | | – | 4 ch | | | | – |
| | μPD789104 | | | | | | 4 ch | – | | | | |
| LCD driving | μPD789417 | 12 K-24 K | 3 ch | 1 ch | 1 ch | 1 ch | – | 7 ch | 1 ch (UART: 1 ch) | 43 pins | 1.8 V | – |
| | μPD789407 | | | | | | 7 ch | – | | | | |
| ASSP | μPD789800 | 8 K | 2 ch | – | – | 1 ch | – | – | 2 ch (USB: 1 ch) | 31 pins | 4.0 V | – |
| | μPD789810 | 6 K | – | | | | | | – | 1 pin | 1.8 V | With built-in EEPROM |

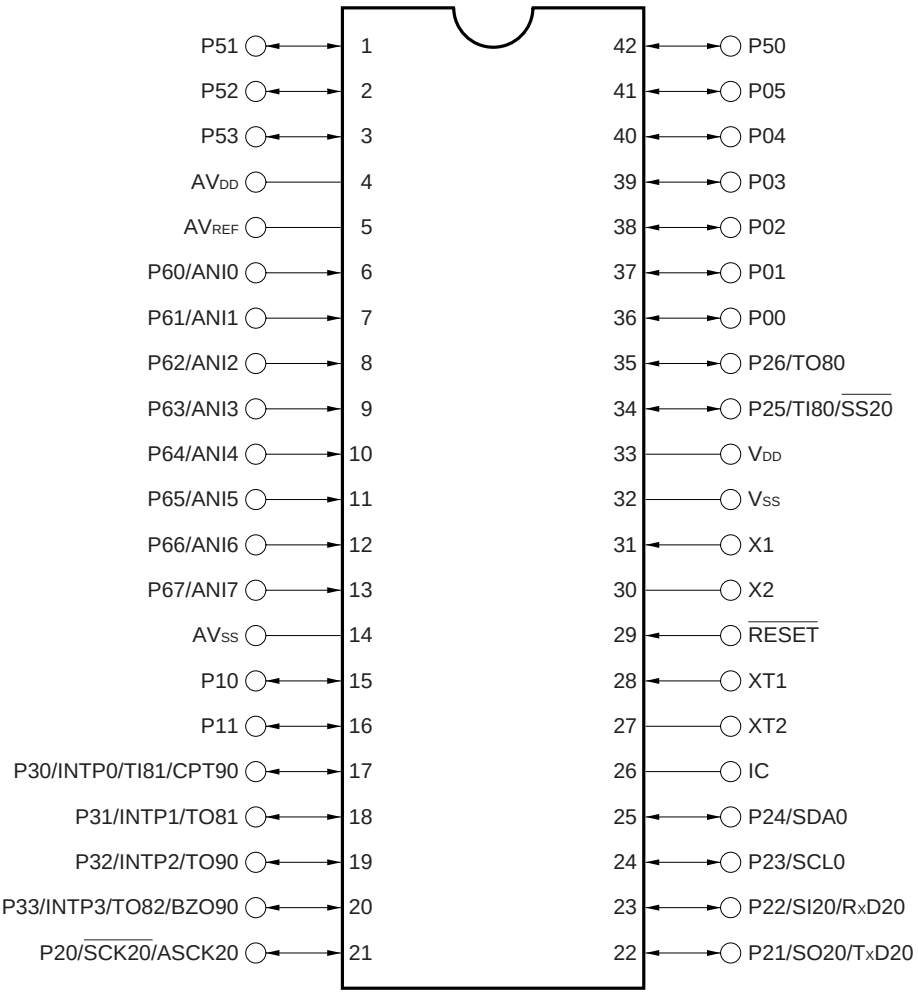
## FUNCTIONS

| Item \ Product | | μPD789166Y | μPD789167Y | μPD789176Y | μPD789177Y |
|---|---|---|---|---|---|
| Internal memory | ROM | 16 Kbytes | 24 Kbytes | 16 Kbytes | 24 Kbytes |
| | High-speed RAM | 512 bytes | | | |
| Minimum instruction execution time | | • 0.4/1.6 μs (operation with main system clock (ceramic/crystal oscillation) running at 5.0 MHz)<br>• 122 μs (operation with subsystem clock running at 32.768 kHz). | | | |
| General-purpose registers | | 8 bits × 8 registers | | | |
| Instruction set | | • 16-bit operations<br>• Bit manipulations (such as set, reset, and test) | | | |
| Multiplier | | 8 bits × 8 bits = 16 bits | | | |
| I/O ports | | Total of 31 port pins<br>• 8 CMOS input pins<br>• 17 CMOS input/output pins<br>• 6 N-channel open-drain pins | | | |
| A/D converters | | Eight channels with 8-bit resolution | | Eight channels with 10-bit resolution | |
| Serial interface | | • Switchable between three-wire serial I/O and UART modes<br>• System management bus (SMB) | | | |
| Timers | | • 16-bit timer counter<br>• Three 8-bit timer/event counters<br>• Clock timer<br>• Watchdog timer | | | |
| Timer output | | Four outputs | | | |
| Buzzer output | | One output | | | |
| Vectored interrupt sources | Maskable | 12 internal and 4 external interrupts | | | |
| | Nonmaskable | Internal interrupt | | | |
| Power supply voltage | | $V_{DD}$ = 1.8 to 5.5 V | | | |
| Operating ambient temperature | | $T_A$ = –40°C to +85°C | | | |
| Package | | 42-pin plastic shrink DIP (600 mil)<br>44-pin plastic QFP (10 × 10 mm)<br>48-pin plastic TQFP (fine pitch) (7 × 7 mm) | | | |

# CONTENTS

## 1. PIN CONFIGURATION (TOP VIEW)

- **42-pin plastic shrink DIP (600 mil)**
  μPD789166YCU-×××
  μPD789167YCU-×××
  μPD789176YCU-×××
  μPD789177YCU-×××



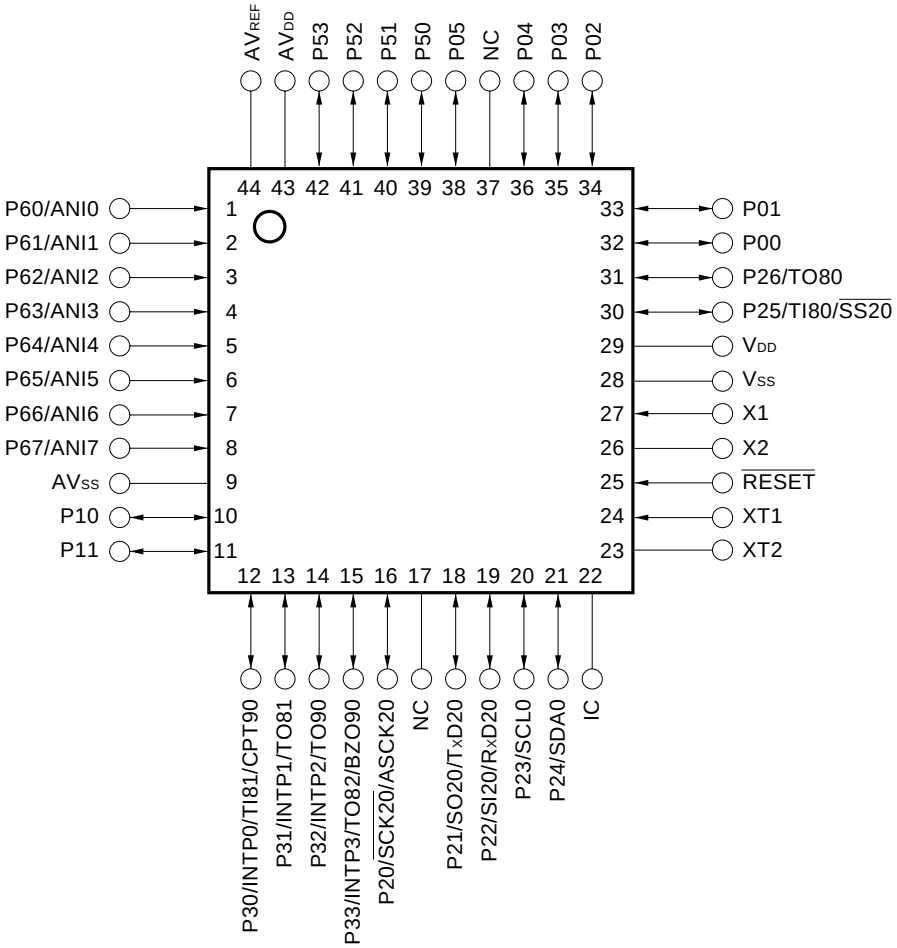

**Cautions 1. Connect the IC (internally connected) pin directly to the $V_{SS}$ pin.**

**2. Connect the $AV_{DD}$ pin to the $V_{DD}$ pin.**

**3. Connect the $AV_{SS}$ pin to the $V_{SS}$ pin.**

- **44-pin plastic QFP (10 × 10 mm)**

  μPD789166YGB-×××-3BS
  μPD789167YGB-×××-3BS
  μPD789176YGB-×××-3BS
  μPD789177YGB-×××-3BS

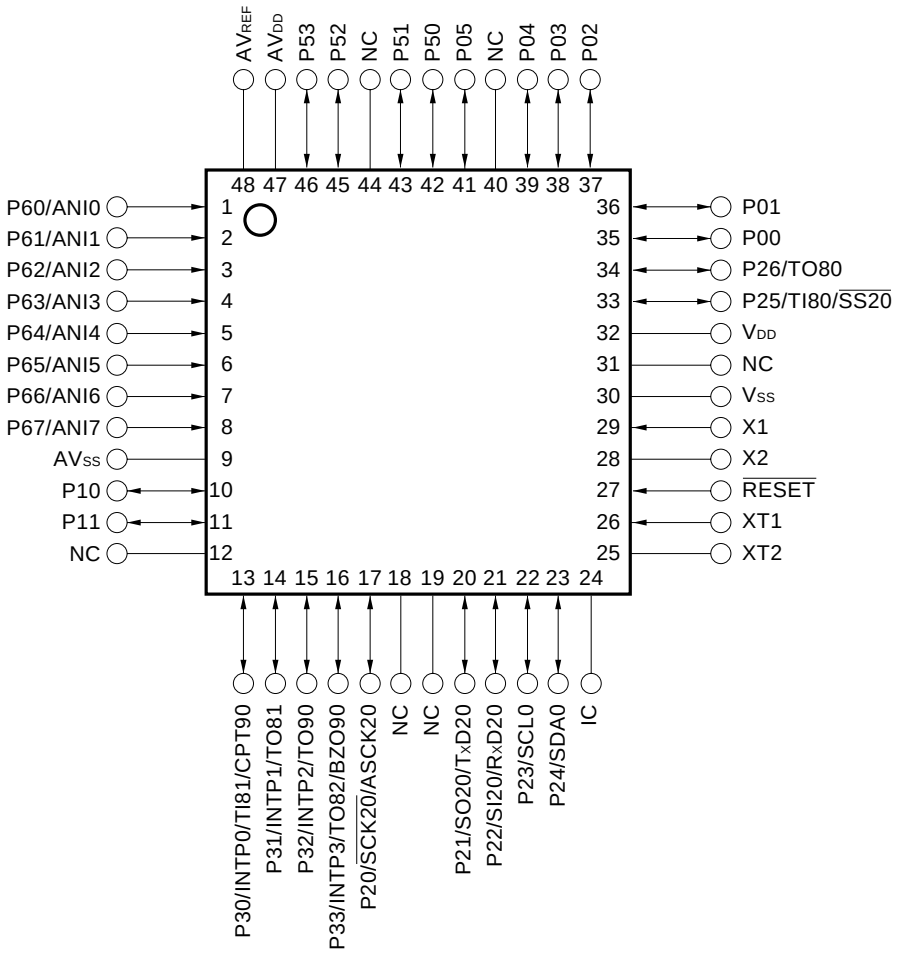



**Cautions 1. Connect the IC (internally connected) pin directly to the $V_{SS}$ pin.**

**2. Connect the $AV_{DD}$ pin to the $V_{DD}$ pin.**

**3. Connect the $AV_{SS}$ pin to the $V_{SS}$ pin.**

- **48-pin plastic TQFP (fine pitch) (7 × 7 mm)**
  μPD789166YGA-×××-9EU
  μPD789167YGA-×××-9EU
  μPD789176YGA-×××-9EU
  μPD789177YGA-×××-9EU




**Cautions 1. Connect the IC (internally connected) pin directly to the $V_{SS}$ pin.**

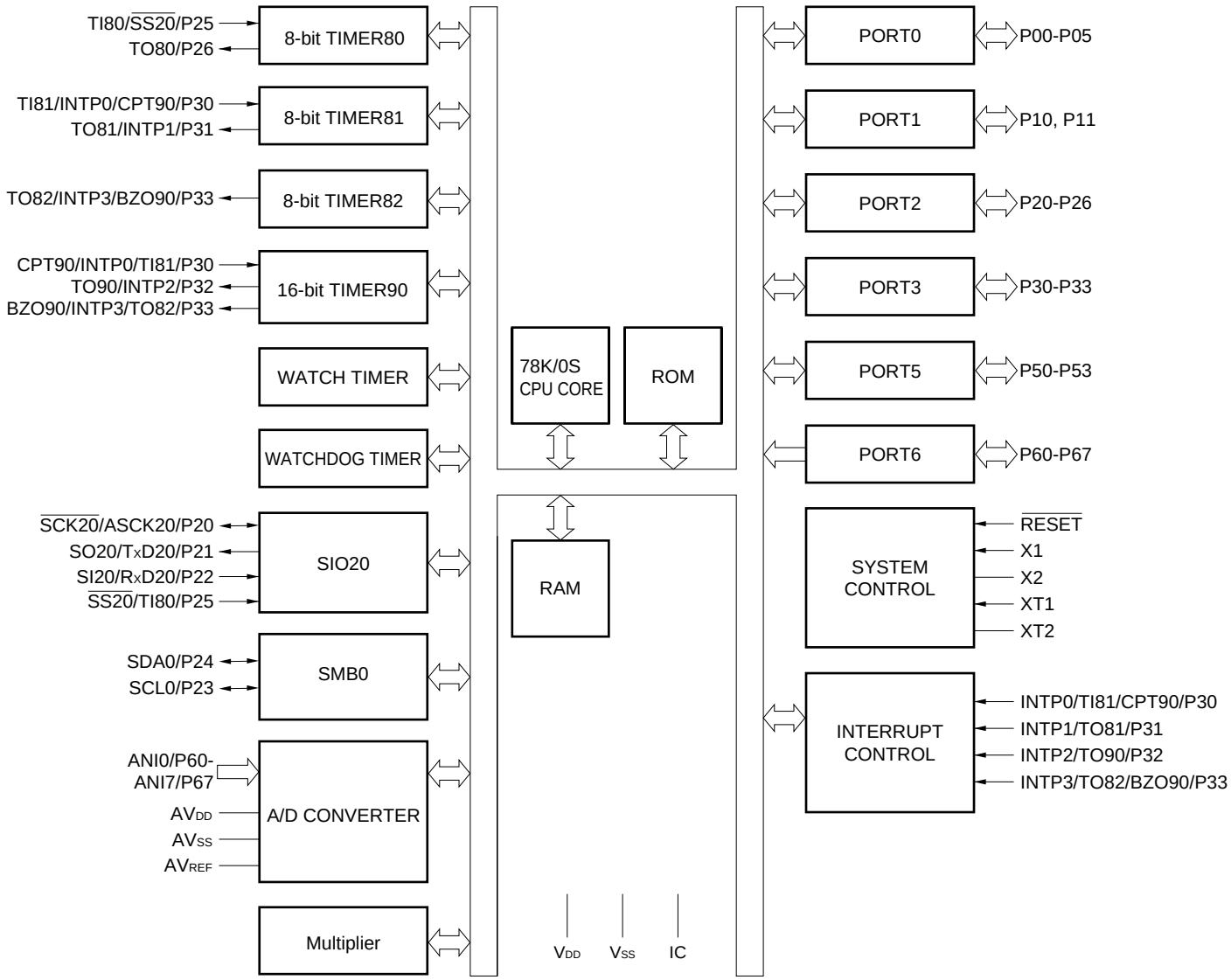
**2. Connect the $AV_{DD}$ pin to the $V_{DD}$ pin.**

**3. Connect the $AV_{SS}$ pin to the $V_{SS}$ pin.**

| | | |
|---|---|---|
| ANI0-ANI7 | : | Analog Input |
| ASCK20 | : | Asynchronous Serial Input |
| $AV_{DD}$ | : | Analog Power Supply |
| $AV_{REF}$ | : | Analog Reference Voltage |
| $AV_{SS}$ | : | Analog Ground |
| BZO90 | : | Buzzer Output |
| CPT90 | : | Capture Trigger Input |
| IC | : | Internally Connected |
| INTP0-INTP3 | : | Interrupt from Peripherals |
| NC | : | Non-connection |
| P00-P05 | : | Port 0 |
| P10, P11 | : | Port 1 |
| P20-P26 | : | Port 2 |
| P30-P33 | : | Port 3 |
| P50-P53 | : | Port 5 |
| P60-P67 | : | Port 6 |

| | | |
|---|---|---|
| $\overline{\text{RESET}}$ | : | Reset |
| RxD20 | : | Receive Data |
| $\overline{\text{SCK20}}$ | : | Serial Clock (for SIO20) |
| SCL0 | : | Serial Clock (for SMB0) |
| SDA0 | : | Serial Data |
| SI20 | : | Serial Input |
| SO20 | : | Serial Output |
| $\overline{\text{SS20}}$ | : | Chip Select Input |
| TI80, TI81 | : | Timer Input |
| TO80-TO82, TO90 | : | Timer Output |
| TxD20 | : | Transmit Data |
| $V_{DD}$ | : | Power Supply |
| $V_{SS}$ | : | Ground |
| X1, X2 | : | Crystal (Main System Clock) |
| XT1, XT2 | : | Crystal (Subsystem Clock) |

## 2. BLOCK DIAGRAM




**Remark** The size of the internal ROM varies depending on the model.

## 3. PIN FUNCTIONS

### 3.1 Port Pins

| Pin name | I/O | Function | When reset | Also used as |
|---|---|---|---|---|
| P00-P05 | I/O | Port 0<br>6-bit input/output port<br>Can be set to either input or output in 1-bit units<br>When used as an input port, whether the on-chip pull-up resistor is to be used can be set by software. | Input | – |
| P10, P11 | I/O | Port 1<br>2-bit input/output port<br>Can be set to either input or output in 1-bit units<br>When used as an input port, whether the on-chip pull-up resistor is to be used can be set by software. | Input | – |
| P20 | I/O | Port 2<br>7-bit input/output port<br>Can be set to either input or output in 1-bit units<br>For P20 to P22, P25, and P26, whether to use the on-chip pull-up resistor can be set by software.<br>Only P23 and P24 can be used as N-channel open-drain input/output port pins. | Input | $\overline{\text{SCK20}}$/ASCK20 |
| P21 | | | | SO20/TxD20 |
| P22 | | | | SI20/RxD20 |
| P23 | | | | SCL0 |
| P24 | | | | SDA0 |
| P25 | | | | TI80/$\overline{\text{SS20}}$ |
| P26 | | | | TO80 |
| P30 | I/O | Port 3<br>4-bit input/output port<br>Can be set to either input or output in 1-bit units<br>Whether to use the on-chip pull-up resistor can be set by software. | Input | INTP0/TI81/CPT90 |
| P31 | | | | INTP1/TO81 |
| P32 | | | | INTP2/TO90 |
| P33 | | | | INTP3/TO82/BZO90 |
| P50-P53 | I/O | Port 5<br>4-bit N-channel open-drain input/output port<br>Can be set to either input or output in 1-bit units<br>Whether to incorporate a pull-up resistor can be set by a mask option. | Input | – |
| P60-P67 | Input | Port 6<br>8-bit input-only port | Input | ANI0-ANI7 |

### 3.2 Non-Port Pins

| Pin name | I/O | Function | When reset | Also used as |
|---|---|---|---|---|
| INTP0 | Input | External interrupt input for which effective edges (rising and/or falling edges) can be set | Input | P30/TI81/CPT90 |
| INTP1 | | | | P31/TO81 |
| INTP2 | | | | P32/TO90 |
| INTP3 | | | | P33/TO82/BZO90 |
| SI20 | Input | Serial data input to serial interface | Input | P22/RxD20 |
| SO20 | Output | Serial data output from serial interface | Input | P21/TxD20 |
| $\overline{SCK20}$ | I/O | Serial clock input/output for serial interface | Input | P20/ASCK20 |
| ASCK20 | Input | Serial clock input to asynchronous serial interface | Input | P20/$\overline{SCK20}$ |
| RxD20 | Input | Serial data input to asynchronous serial interface | Input | P22/SI20 |
| TxD20 | Output | Serial data output from asynchronous serial interface | Input | P21/SO20 |
| SCL0 | I/O | SMB0 clock input/output | Input | P23 |
| SDA0 | I/O | SMB0 data input/output | Input | P24 |
| $\overline{SS20}$ | Input | Chip select input to serial interface | Input | P25/TI80 |
| TI80 | Input | External count clock input to 8-bit timer (TM80) | Input | P25/$\overline{SS20}$ |
| TI81 | Input | External count clock input to 8-bit timer (TM81) | Input | P30/INTP0/CPT90 |
| TO80 | Output | 8-bit timer (TM80) output | Input | P26 |
| TO81 | Output | 8-bit timer (TM81) output | Input | P31/INTP1 |
| TO82 | Output | 8-bit timer (TM82) output | Input | P33/INTP3/BZO90 |
| TO90 | Output | 16-bit timer (TM90) output | Input | P32/INTP2 |
| CPT90 | Input | Capture edge input | Input | P30/INTP0/TI81 |
| BZO90 | Output | Buzzer output | Input | P33/INTP3/TO82 |
| ANI0-ANI7 | Input | A/D converter analog input | Input | P60-P67 |
| $AV_{REF}$ | – | A/D converter reference voltage | – | – |
| $AV_{SS}$ | – | A/D converter ground potential | – | – |
| $AV_{DD}$ | – | A/D converter analog power supply | – | – |
| X1 | Input | Connected to crystal for main system clock oscillation | – | – |
| X2 | – | | – | – |
| XT1 | Input | Connected to crystal for subsystem clock oscillation | – | – |
| XT2 | – | | – | – |
| $\overline{RESET}$ | Input | System reset input | Input | – |
| $V_{DD}$ | – | Positive supply voltage | – | – |
| $V_{SS}$ | – | Ground potential | – | – |
| IC | – | This pin is internally connected. Connect this pin directly to the $V_{SS}$ pin. | – | – |
| NC | – | This pin is not connected internally. Connect this pin to the $V_{SS}$ pin (or leave this pin open). | – | – |

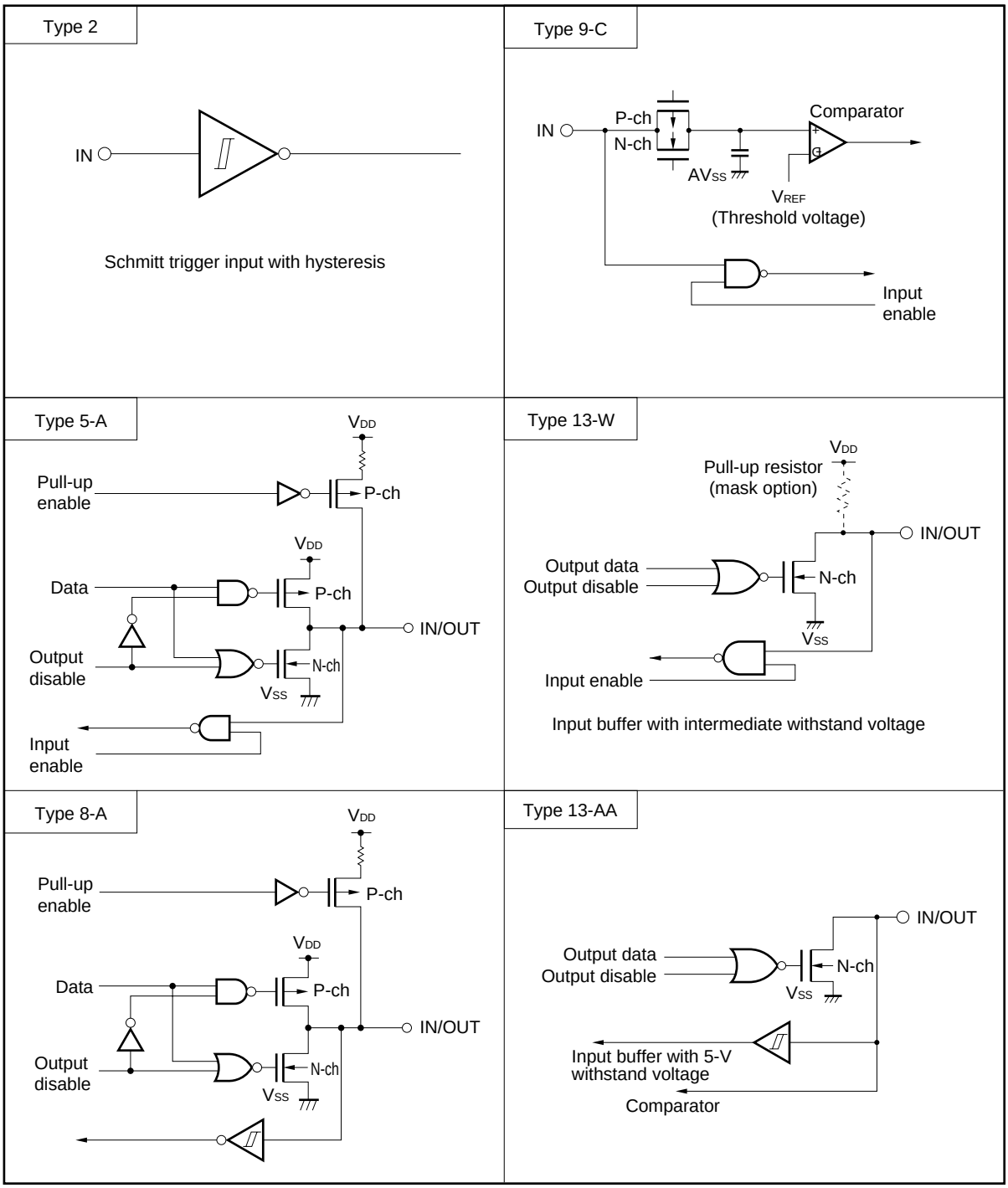
### 3.3 Pin Input/Output Circuits and Handling of Unused Pins

Table 3-1 lists the types of input/output circuits for each pin and explains how unused pins are handled.
Figure 3-1 shows the configuration of each type of input/output circuit.

**Table 3-1. Type of Input/Output Circuit for Each Pin and Handling of Unused Pins**

| Pin name | I/O circuit type | I/O | Recommended connection of unused pins |
|---|---|---|---|
| P00-P05 | 5-A | I/O | Connect these pins to the $V_{DD}$ or $V_{SS}$ pin via respective resistors. |
| P10, P11 | | | |
| P20/$\overline{SCK20}$/ASCK20 | 8-A | | |
| P21/SO20/TxD20 | | | |
| P22/SI20/RxD20 | | | |
| P23/SCL0 | 13-AA | | |
| P24/SDA0 | | | |
| P25/TI80/$\overline{SS20}$ | 8-C | | |
| P26/TO80 | | | |
| P30/INTP0/TI81/CPT90 | | | |
| P31/INTP1/TO81 | | | |
| P32/INTP2/TO90 | | | |
| P33/INTP3/TO82/BZO90 | | | |
| P50-P53 | 13-W | | |
| P60/ANI0-P67/ANI7 | 9-C | Input | |
| XT1 | – | Input | Connect this pin to the $V_{SS}$ pin. |
| XT2 | | – | Leave this pin open. |
| $\overline{RESET}$ | 2 | Input | – |
| IC | – | – | Connect this pin directly to the $V_{SS}$ pin. |
| NC | – | – | |

**Figure 3-1. Pin Input/Output Circuits**

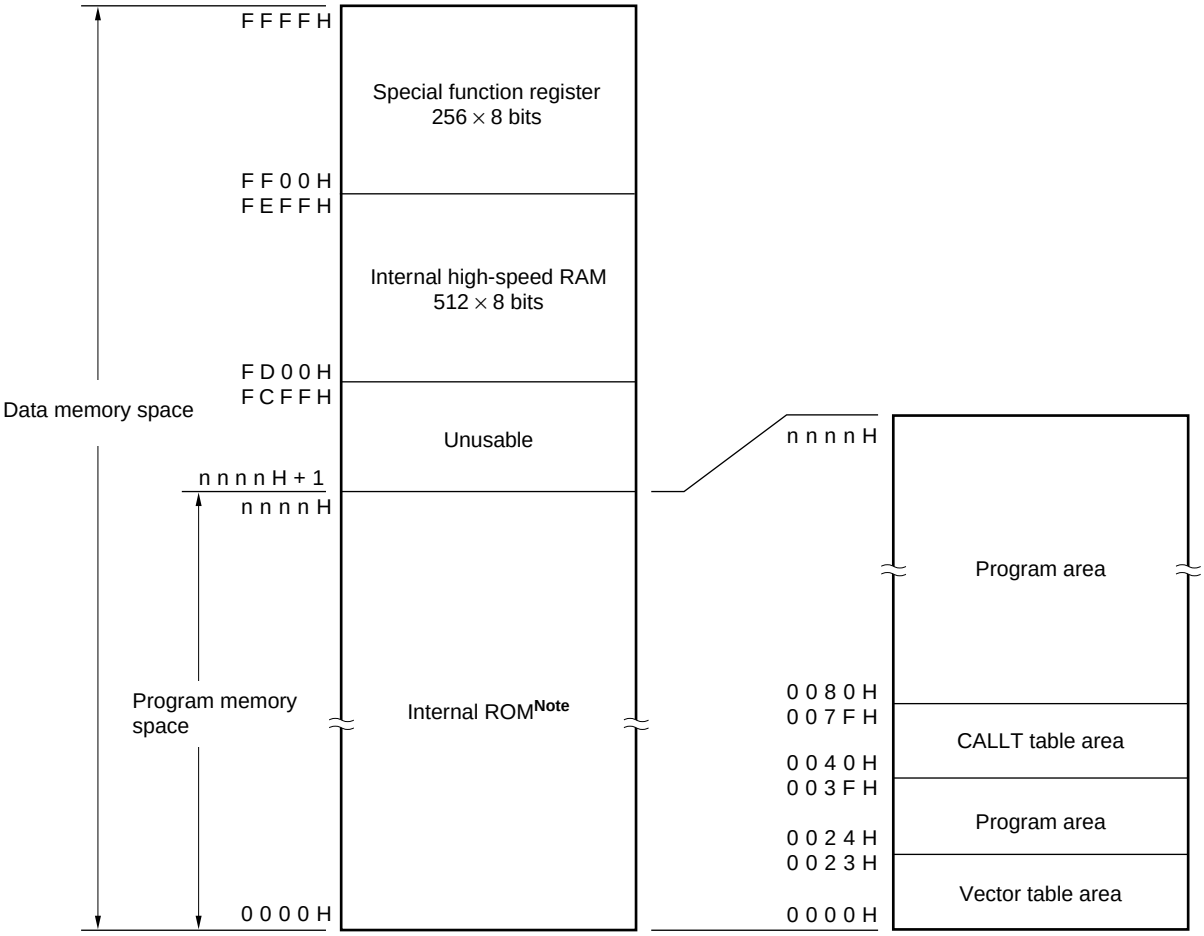


## 4. CPU ARCHITECTURE

### 4.1 Memory Space

The μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y can each access up to 64 Kbytes of memory space. Figure 4-1 shows the memory map.

**Figure 4-1. Memory Map**



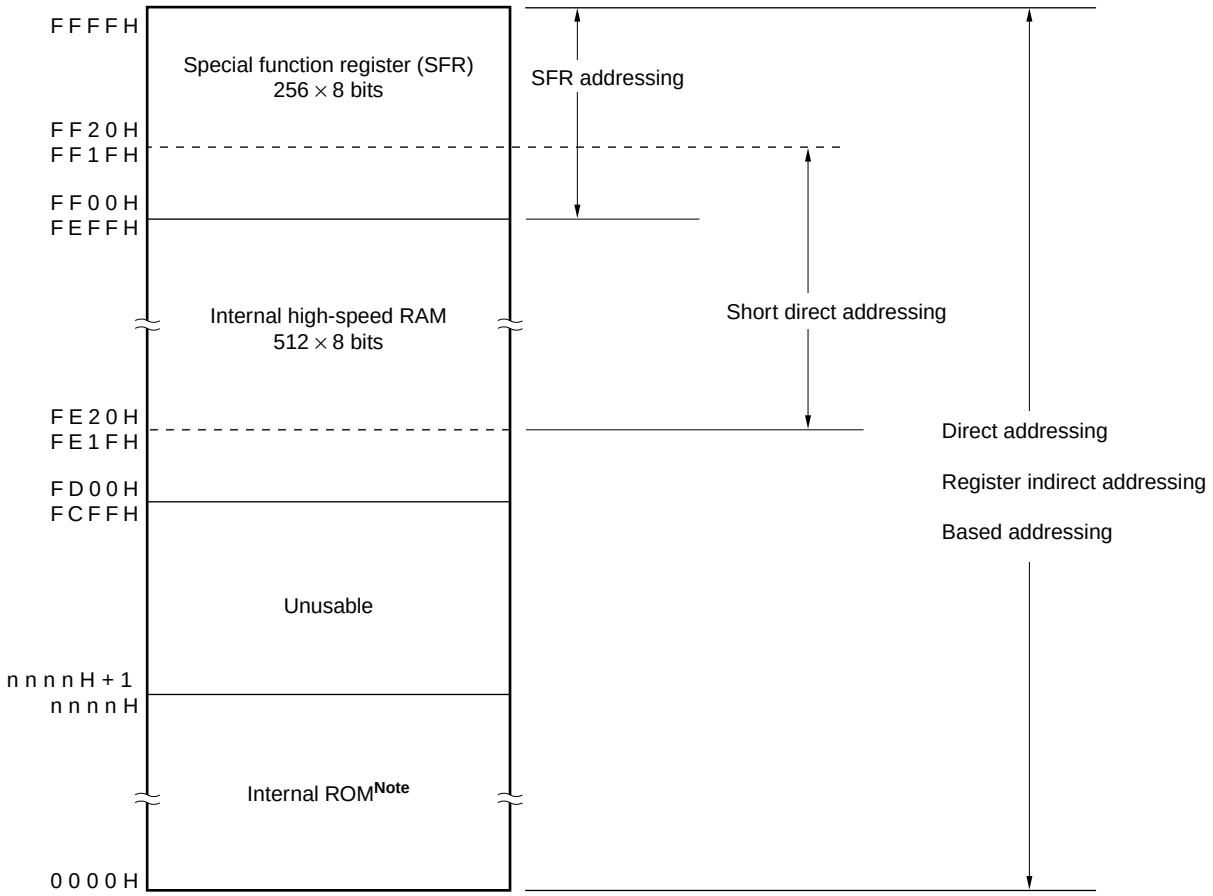

**Note** The size of the internal ROM varies depending on the model (see the following table).

| Product name | Last address of internal ROM nnnnH |
|---|---|
| μPD789166Y and μPD789176Y | 3FFFH |
| μPD789167Y and μPD789177Y | 5FFFH |

## 4.2 Data Memory Addressing

Each of the μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y is provided with a wide range of addressing modes to make memory manipulation as efficient as possible. A data memory area (FD00H to FFFFH) can be accessed using a unique addressing mode according to its use, such as a special function register (SFR). Figure 4-2 illustrates the data memory addressing modes.

**Figure 4-2. Data Memory Addressing Modes**




**Note** The size of internal ROM varies depending on the model (see the following table).

| Product name | Last address of internal ROM nnnnH |
|---|---|
| μPD789166Y and μPD789176Y | 3FFFH |
| μPD789167Y and μPD789177Y | 5FFFH |

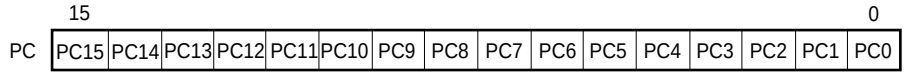
## 4.3 Processor Registers

### 4.3.1 Controller registers

**(1) Program counter (PC)**

The PC is a 16-bit register for holding address information that indicates the next program to be executed.

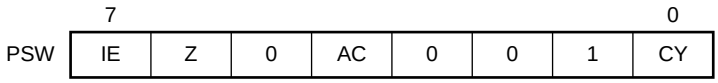
**Figure 4-3. Program Counter Configuration**




**(2) Program status word (PSW)**

The PSW is an 8-bit register for holding the status of the CPU according to the results of instruction execution.

**Figure 4-4. Program Status Word Configuration**




**(a) Interrupt enable flag (IE)**

IE is used to control whether interrupt requests are to be accepted by the CPU.

**(b) Zero flag (Z)**

Z is set (1) if the result of operation is zero. Otherwise, it is reset (0).

**(c) Auxiliary carry flag (AC)**

AC is set (1) if the result of the operation has a carry from bit 3 or a borrow to bit 3. Otherwise, it is reset (0).

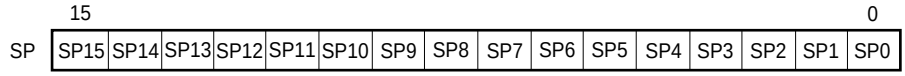
**(d) Carry flag (CY)**

CY is used to indicate whether an overflow or underflow has occurred during the execution of a subtract or add instruction.

**(3) Stack pointer (SP)**

SP is a 16-bit register for holding the start address of a stack area. The stack area can be specified only in an area (FD00H to FEFFH) of internal high-speed RAM.

**Figure 4-5. Stack Pointer Configuration**




**Caution A $\overline{\text{RESET}}$ input makes the SP content undefined. Before executing an instruction, always initialize the SP.**

### 4.3.2 General-purpose registers

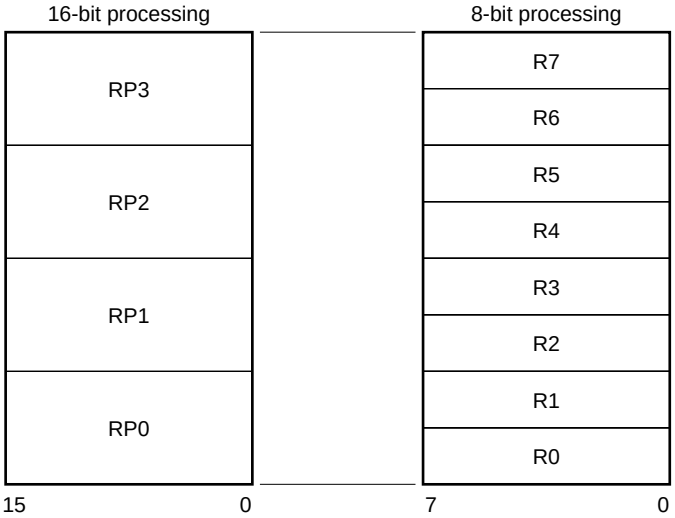
Each device has eight 8-bit general-purpose registers (X, A, C, B, E, D, L, and H).

These registers can be used as 16-bit registers (two 8-bit registers used in pairs like AX, BC, DE, and HL) as well as ordinary 8-bit registers.

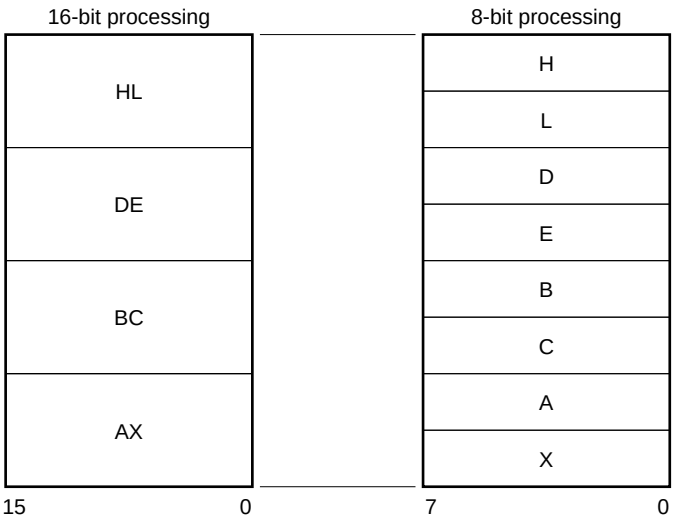
These registers are identified using functional register names (X, A, C, B, E, D, L, H, AX, BC, DE, and HL) and absolute register names (R0 to R7 and RP0 to RP3).

**Figure 4-6. General-Purpose Register Configuration**

**(a) Absolute register names**




**(b) Functional register names**



### 4.3.3 Special function registers (SFRs)

The SFRs are used as peripheral hardware mode registers and control registers. They are mapped in a 256-byte space, from FF00H to FFFFH.

**Table 4-1. Special Function Registers (1/3)**

| Address | Special function register (SFR) name | Symbol | | R/W | Number of bits manipulated simultaneously | | | When reset |
|---|---|---|---|---|---|---|---|---|
| | | | | | 1 bit | 8 bits | 16 bits | |
| FF00H | Port 0 | P0 | | R/W | O | O | – | 00H |
| FF01H | Port 1 | P1 | | | O | O | – | |
| FF02H | Port 2 | P2 | | | O | O | – | |
| FF03H | port 3 | P3 | | | O | O | – | |
| FF05H | Port 5 | P5 | | | O | O | – | |
| FF06H | Port 6 | P6 | | R | O | O | – | |
| FF10H | 16-bit multiplication result storage register 0 | MUL0L | Note 1 MUL0 | R | – | O Note 2 | O Note 3 | Undefined |
| FF11H | | MUL0H | | | | | | |
| FF14H | A/D conversion result register 0 Note 4 | ADCR0 | | R | – | O | O | |
| FF15H | | | | | | | | |
| FF16H | 16-bit compare register 90 | CR90L | Note 1 CR90 | W | – | O Note 2 | O Note 3 | FFFFH |
| FF17H | | CR90H | | | | | | |
| FF18H | 16-bit timer register 90 | TM90L | Note 1 TM90 | R | – | O Note 2 | O Note 3 | 0000H |
| FF19H | | TM90H | | | | | | |
| FF1AH | 16-bit capture register 90 | TCP90L | Note 1 TCP90 | | – | O Note 2 | O Note 3 | Undefined |
| FF1BH | | TCP90H | | | | | | |
| FF20H | Port mode register 0 | PM0 | | R/W | O | O | – | FFH |
| FF21H | Port mode register 1 | PM1 | | | O | O | – | |
| FF22H | Port mode register 2 | PM2 | | | O | O | – | |
| FF23H | Port mode register 3 | PM3 | | | O | O | – | |
| FF25H | Port mode register 5 | PM5 | | | O | O | – | |
| FF32H | Pull-up resistor option register B2 | PUB2 | | | O | O | – | 00H |
| FF33H | Pull-up resistor option register B3 | PUB3 | | | O | O | – | |
| FF42H | Timer clock selection register 2 | TCL2 | | | O | O | – | |

**Notes 1.** MUL0, CR90, TM90, and TCP90 are those SFR symbols used only in 16-bit access mode.

**2.** MUL0, CR90, TM90, and TCP90 are designed only for 16-bit access. With direct addressing, however, they can also be accessed in 8-bit mode.

**3.** 16-bit access is allowed only with short direct addressing.

**4.** When 8-bit A/D converters are used (for the μPD789166Y and μPD789167Y), this register can be accessed only in 8-bit mode. In this case, the address is assumed to be FF15H.

When 10-bit A/D converters are used (for the μPD789176Y and μPD789177Y), this register can be accessed only in 16-bit mode.

When the μPD78F9177Y is used as flash memory for the μPD789166Y or μPD789167Y, 8-bit access is allowed. However, only those object files generated by an assembler used with the μPD789166Y or μPD789167Y are supported for this access.

**Table 4-1. Special Function Registers (2/3)**

| Address | Special function register (SFR) name | Symbol | R/W | Number of bits manipulated simultaneously | | | When reset |
|---|---|---|---|---|---|---|---|
| | | | | 1 bit | 8 bits | 16 bits | |
| FF48H | 16-bit timer mode control register 90 | TMC90 | R/W | O | O | – | 00H |
| FF49H | Buzzer output control register 90 | BZC90 | | O | O | – | |
| FF4AH | Clock timer mode control register | WTM | | O | O | – | |
| FF50H | 8-bit compare register 80 | CR80 | W | – | O | – | Undefined |
| FF51H | 8-bit timer register 80 | TM80 | R | – | O | – | 00H |
| FF53H | 8-bit timer mode control register 80 | TMC80 | R/W | O | O | – | |
| FF54H | 8-bit compare register 81 | CR81 | W | – | O | – | Undefined |
| FF55H | 8-bit timer register 81 | TM81 | R | – | O | – | 00H |
| FF57H | 8-bit timer mode control register 81 | TMC81 | R/W | O | O | – | |
| FF58H | 8-bit compare register 82 | CR82 | W | – | O | – | Undefined |
| FF59H | 8-bit timer register 82 | TM82 | R | – | O | – | 00H |
| FF5BH | 8-bit timer mode control register 82 | TMC82 | R/W | O | O | – | |
| FF70H | Asynchronous serial interface mode register 20 | ASIM20 | | O | O | – | |
| FF71H | Asynchronous serial interface status register 20 | ASIS20 | R | – | O | – | |
| FF72H | Serial operation mode register 20 | CSIM20 | R/W | O | O | – | |
| FF73H | Baud rate generator control register 20 | BRGC20 | | O | O | – | |
| FF74H | Transmission shift register 20 | TXS20 / SIO20 | W | – | O | – | FFH |
| | Reception buffer register 20 | RXB20 / SIO20 | R | – | O | – | Undefined |
| FF78H | SMB control register 0 | SMBC0 | R/W | O | O | – | 00H |
| FF79H | SMB status register 0 | SMBS0 | R | O | O | – | |
| FF7AH | SMB clock selection register 0 | SMBCL0 | R/W | O | O | – | |
| FF7BH | SMB slave address register 0 | SMBSVA0 | | O | O | – | |
| FF7CH | SMB mode register 0 | SMBM0 | | O | O | – | 20H |
| FF7DH | SMB input level setting register 0 | SMBVI0 | | O | O | – | 00H |
| FF7EH | SMB shift register 0 | SMB0 | | O | O | – | |
| FF80H | A/D converter mode register 0 | ADM0 | | O | O | – | |
| FF84H | A/D input selection register 0 | ADS0 | | O | O | – | |
| FFD0H | Multiplication data register A0 | MRA0 | W | O | O | – | Undefined |
| FFD1H | Multiplication data register B0 | MRB0 | | O | O | – | |
| FFD2H | Multiplier control register 0 | MULC0 | R/W | O | O | – | 00H |
| FFE0H | Interrupt request flag register 0 | IF0 | | O | O | – | |
| FFE1H | Interrupt request flag register 1 | IF1 | | O | O | – | |

**Table 4-1. Special Function Registers (3/3)**

| Address | Special function register (SFR) name | Symbol | R/W | Number of bits manipulated simultaneously | | | When reset |
|---|---|---|---|---|---|---|---|
| | | | | 1 bit | 8 bits | 16 bits | |
| FFE4H | Interrupt mask flag register 0 | MK0 | R/W | O | O | – | FFH |
| FFE5H | Interrupt mask flag register 1 | MK1 | | O | O | – | |
| FFECH | External interrupt mode register 0 | INTM0 | | – | O | – | 00H |
| FFEDH | External interrupt mode register 1 | INTM1 | | – | O | – | |
| FFF0H | Suboscillation mode register | SCKM | | O | O | – | |
| FFF2H | Subclock control register | CSS | | O | O | – | |
| FFF7H | Pull-up resistor option register 0 | PU0 | | O | O | – | |
| FFF9H | Watchdog timer mode register | WDTM | | O | O | – | |
| FFFAH | Oscillation settling time selection register | OSTS | | – | O | – | 04H |
| FFFBH | Processor clock control register | PCC | | O | O | – | 02H |

# 5. PERIPHERAL HARDWARE FUNCTIONS

## 5.1 Ports

### 5.1.1 Port functions

The μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y are provided with the ports shown in Figure 5-1. These ports are used to enable several types of control. Table 5-1 lists the functions of each port.

These ports, while originally designed as digital input/output ports, can also be used for other functions, as summarized in **Chapter 3**.

**Figure 5-1. Port Types**

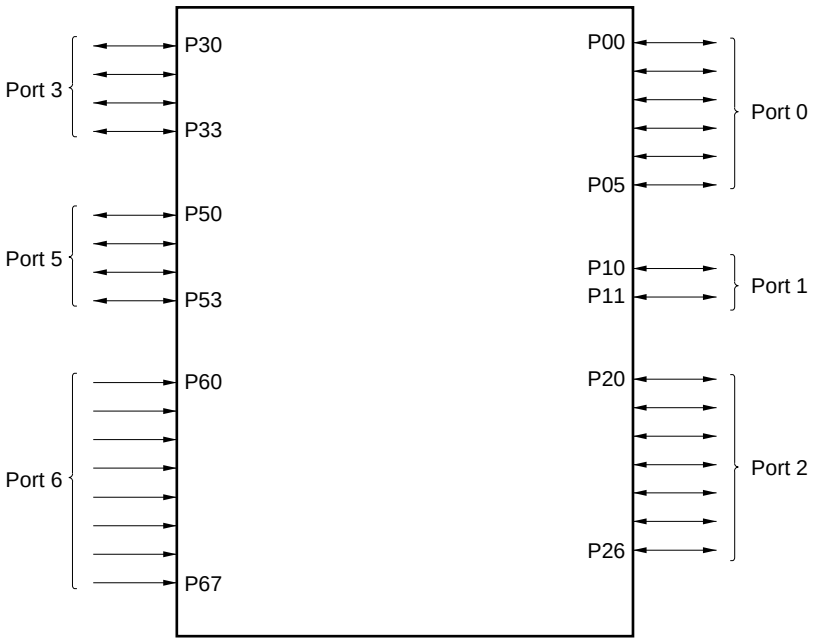


**Table 5-1. Port Functions**

| Port name | Pin name | Description |
|---|---|---|
| Port 0 | P00-P05 | Input/output port. Each bit of the port can be separately specified as being for input or output. A port used for input can be connected to an on-chip pull-up resistor by means of software specification. |
| Port 1 | P10, P11 | Input/output port. Each bit of the port can be separately specified as being for input or output. A port used for input can be connected to an on-chip pull-up resistor by means of software specification. |
| Port 2 | P20-P26 | Input/output port. Each bit of the port can be separately specified as being for input or output. Each of P20 to P22, P25, and P26 can be connected to an on-chip pull-up resistor by means of software specification.<br>(P23 and P24 are used as N-channel open-drain input/output ports (with 5-V withstand voltage).) |
| Port 3 | P30-P33 | Input/output port. Each bit of the port can be separately specified as being for input or output. The port can be connected to an on-chip pull-up resistor by means of software specification. |
| Port 5 | P50-P53 | N-channel open-drain input/output port. Each bit of the port can be separately specified as being for input or output.<br>Whether the port itself is to contain a pull-up resistor is specified with a mask option. |
| Port 6 | P60-P67 | Input-only port |

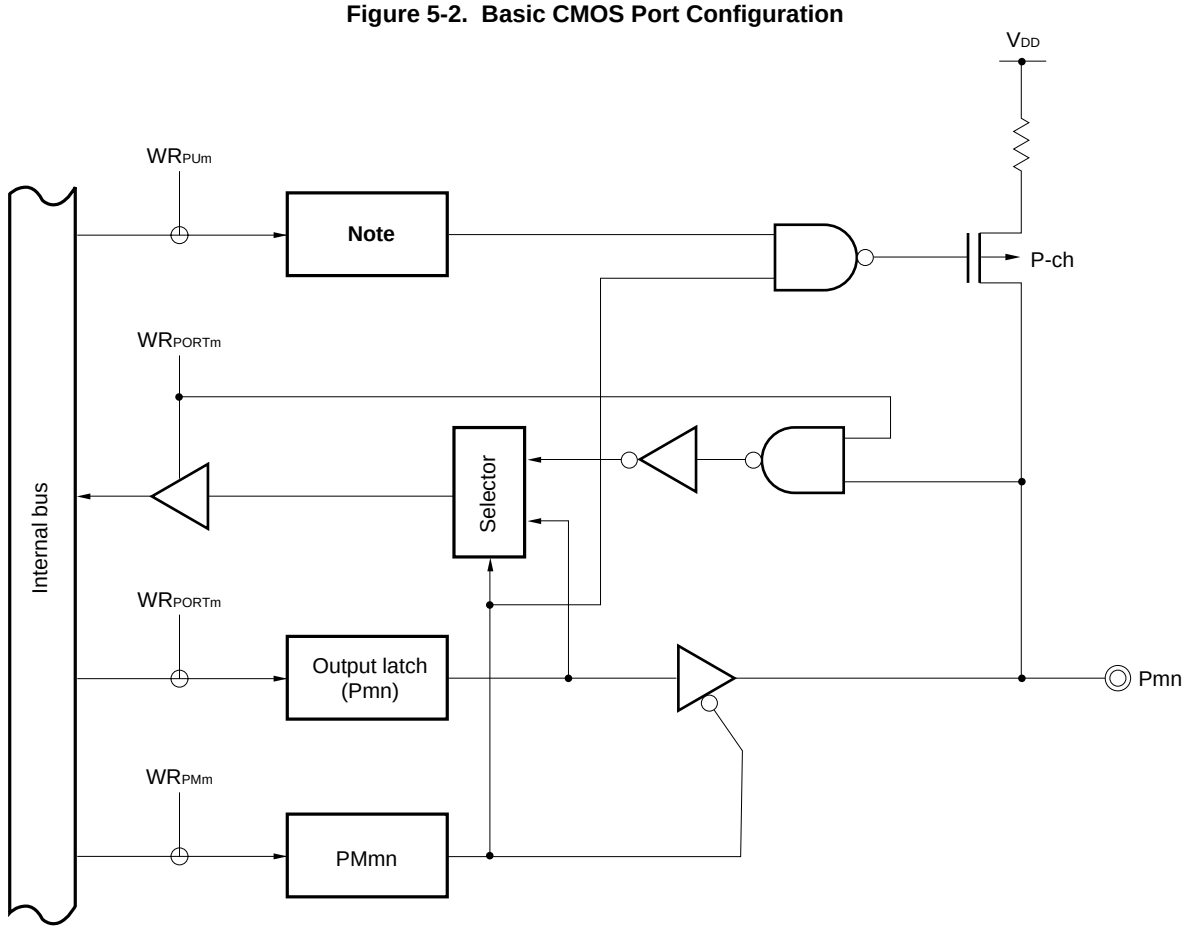
### 5.1.2 Port configuration

The hardware configuration of the ports is as follows.

**Table 5-2. Port Configuration**

| Item | Configuration |
|---|---|
| Control register | Port mode register (PMm, where m = 0, 1, 2, 3, or 5)<br>Pull-up resistor option registers (PU0, PUB2, and PUB3) |
| Port pins | Total: 31 (17 CMOS input/output, 8 CMOS input-only, and 6 N-ch open-drain input/output pins) |
| Pull-up resistors | Total: 21 (on-chip pull-up resistors can be used as specified by software, and whether a port itself is to contain pull-up resistors can be specified with a mask option) |

**Figure 5-2. Basic CMOS Port Configuration**




**Note** Each bit of the pull-up resistor option registers (PU0, PUB2, and PUB3)
PU00 and PU01 for PU0
PUB20 to PUB22, PUB25, and PUB26 for PUB2
PUB30 to PUB33 for PUB3
For details, see **(2)** in **Section 5.1.3**.

**Caution Figure 5-2 shows the basic configuration of the CMOS input/output ports. The configuration differs depending on the functions assigned to the dual-function pins.**
**P23, P24, and port 5 are not included in the basic configuration because they are used as N-channel open-drain input/output ports.**

**Remark** PMmn : Bit n of port mode register m, where m = 0 to 3 and n = 0 to 7
Pmn : Bit n of port m
RD : Port read signal
WR : Port write signal
For details, see **(2)** in **Section 5.1.3**.

### 5.1.3 Port function control registers

The following two types of registers are used to control the ports.

- Port mode registers (PM0 to PM3, and PM5)
- Pull-up resistor option registers (PU0, PUB2, and PUB3)

#### (1) Port mode registers (PM0 to PM3, and PM5)

The port mode registers separately specify each port bit as being for input or output.
Each port mode register is manipulated using a 1-bit or 8-bit memory manipulation instruction.
A $\overline{\text{RESET}}$ input writes FFH into the port mode registers.
When port pins are used for secondary functions, the corresponding port mode register and output latch must be set or reset as described in Table 5-3.

**Caution When port 3 is acting as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as the input for an external interrupt. To use port 3 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.**

**Table 5-3. Port Mode Register and Output Latch Settings for Using Secondary Functions**

| Pin name | Secondary function | | PM×× | P×× |
|---|---|---|---|---|
| | Name | Input/output | | |
| P23 | SCL0 | Input/output | 0 | 0 |
| P24 | SDA0 | Input/output | 0 | 0 |
| P25 | TI80 | Input | 1 | × |
| P26 | TO80 | Output | 0 | 0 |
| P30 | INTP0 | Input | 1 | × |
| | TI81 | Input | 1 | × |
| | CPT90 | Input | 1 | × |
| P31 | INTP1 | Input | 1 | × |
| | TO81 | Output | 0 | 0 |
| P32 | INTP2 | Input | 1 | × |
| | TO90 | Output | 0 | 0 |
| P33 | INTP3 | Input | 1 | × |
| | TO82 | Output | 0 | 0 |
| | BZO90 | Output | 0 | 0 |
| P60-P67 | ANI0-ANI7 | Input | 1 | × |

**Caution When port 2 is being used as a serial interface, it is necessary to specify whether the port is an input or output port, and to set the output latch accordingly. See Table 5-13 for an explanation of how to make this specification.**

**Remark** × : Don't care
PM×× : Port mode register
P×× : Port output latch

**Figure 5-3. Format of Port Mode Register**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PM0 | 1 | 1 | PM05 | PM04 | PM03 | PM02 | PM01 | PM00 | FF20H | FFH | R/W |
| PM1 | 1 | 1 | 1 | 1 | 1 | 1 | PM11 | PM10 | FF21H | FFH | R/W |
| PM2 | 1 | PM26 | PM25 | PM24 | PM23 | PM22 | PM21 | PM20 | FF22H | FFH | R/W |
| PM3 | 1 | 1 | 1 | 1 | PM33 | PM32 | PM31 | PM30 | FF23H | FFH | R/W |
| PM5 | 1 | 1 | 1 | 1 | PM53 | PM52 | PM51 | PM50 | FF25H | FFH | R/W |

| PMmn | Pmn pin input/output mode selection<br>(m = 0 to 3 or 5; n = 0 to 7) |
|---|---|
| 0 | Output mode (output buffer ON) |
| 1 | Input mode (output buffer OFF) |

**(2) Pull-up resistor option registers (PU0, PUB2, PUB3)**

These registers are used to specify pull-up resistor connection on a port-by-port basis and bit-by-bit basis. The method of pull-up resistor connection varies, depending on whether a connection is made on a port-by-port basis or bit-by-bit basis as described below.

**(a) Pull-up resistor option register (PU0)**

This register is used for port-by-port specification. An on-chip pull-up resistor can be used only for those bits set to the input mode of a port for which the use of the on-chip pull-up resistor is specified using PU0. For those bits set to the output mode, on-chip pull-up resistors cannot be used, regardless of the setting of PU0. This also applies to a dual-function pin used as an output pin.

A $\overline{\text{RESET}}$ input clears PU0 to 00H.

**(b) Pull-up resistor option registers (PUB2, PUB3)**

These registers are used for bit-by-bit specification. By setting PUB2 or PUB3, an on-chip pull-up resistor can be used, regardless of the setting of the port mode register.

A $\overline{\text{RESET}}$ input clears PUB2 and PUB3 to 00H.

**Figure 5-4. Format of Pull-Up Resistor Option Register 0**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | ① | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PU0 | 0 | 0 | 0 | 0 | 0 | 0 | PU01 | PU00 | FFF7H | 00H | R/W |

| PU0m | Pm on-chip pull-up resistor selection[Note] (m = 0 or 1) |
|---|---|
| 0 | On-chip pull-up resistor not used |
| 1 | On-chip pull-up resistor used |

**Note** For each port, PU0 selects whether on-chip pull-up resistors are to be used.

**Caution Bits 2 to 7 must be fixed to 0.**

**Figure 5-5. Format of Pull-Up Resistor Option Register B2**

| Symbol | 7 | ⑥ | ⑤ | 4 | 3 | ② | ① | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PUB2 | 0 | PUB26 | PUB25 | 0 | 0 | PUB22 | PUB21 | PUB20 | FF32H | 00H | R/W |

| PUB2m | P2m on-chip pull-up resistor selection[Note] (m = 0 to 2, 5, or 6) |
|---|---|
| 0 | On-chip pull-up resistor not used |
| 1 | On-chip pull-up resistor used |

**Note** PUB2 selects whether on-chip pull-up resistors are to be used in 1-bit units.

**Caution Bits 3, 4, and 7 must be fixed to 0.**

**Figure 5-6. Format of Pull-Up Resistor Option Register B3**

| Symbol | 7 | 6 | 5 | 4 | ③ | ② | ① | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PUB3 | 0 | 0 | 0 | 0 | PUB33 | PUB32 | PUB31 | PUB30 | FF33H | 00H | R/W |

| PUB3m | P3m on-chip pull-up resistor selection[Note] (m = 0 to 3) |
|---|---|
| 0 | On-chip pull-up resistor not used |
| 1 | On-chip pull-up resistor used |

**Note** PUB3 selects whether on-chip pull-up resistors are to be used in 1-bit units.

**Caution Bits 4 to 7 must be fixed to 0.**

## 5.2 Clock Generator

### 5.2.1 Clock generator functions

The clock generator generates the clock pulse to be supplied to the CPU and peripheral hardware. There are two types of system clock oscillators:

- Main system clock oscillator (ceramic or crystal oscillation)
  This circuit generates a frequency of 1.0 to 5.0 MHz. Oscillation can be stopped by executing the STOP instruction or by using the processor clock control register.
- Subsystem clock oscillator
  This circuit generates 32.768 kHz. Oscillation can be stopped by using the suboscillation mode register.

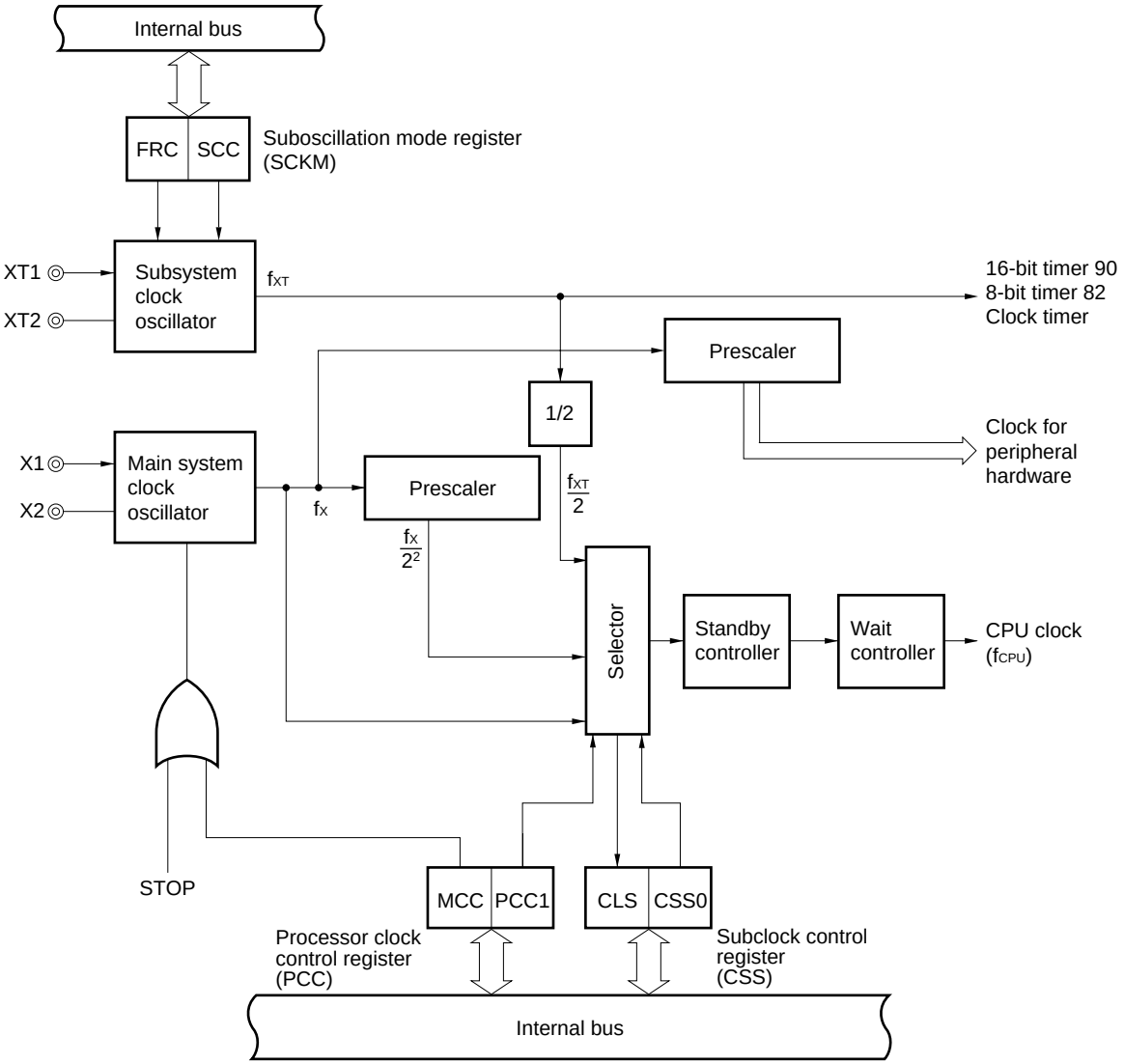
### 5.2.2 Clock generator configuration

The clock generator consists of the following hardware.

**Table 5-4. Clock Generator Configuration**

| Item | Configuration |
|---|---|
| Control register | Processor clock control register (PCC)<br>Suboscillation mode register (SCKM)<br>Subclock control register (CSS) |
| Oscillator | Main system clock oscillator<br>Subsystem clock oscillator |

**Figure 5-7. Block Diagram of Clock Generator**



### 5.2.3 Clock generator control registers

The clock generator is controlled using the following registers.

- Processor clock control register (PCC)
- Suboscillation mode register (SCKM)
- Subclock control register (CSS)

#### (1) Processor clock control register (PCC)

The PCC selects a CPU clock and specifies a corresponding frequency division ratio.

It is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input loads 02H into the PCC.

**Figure 5-8. Format of Processor Clock Control Register**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PCC | MCC | 0 | 0 | 0 | 0 | 0 | PCC1 | 0 | FFFBH | 02H | R/W |

| MCC | Control of main system clock oscillator operation |
|---|---|
| 0 | Operation enabled |
| 1 | Operation disabled |

| CSS0 | PCC1 | CPU clock ($f_{CPU}$) selection[Note] |
|---|---|---|
| 0 | 0 | $f_X$ (0.2 μs) |
| 0 | 1 | $f_X/2^2$ (0.8 μs) |
| 1 | 0 | $f_{XT}/2$ (61 μs) |
| 1 | 1 | |

**Note** A CPU clock is selected by a combination of the PCC1 flag in the processor clock control register (PCC) and the CSS0 flag in the subclock control register (CSS). (See **(3)** in **Section 5.2.3**.)

**Cautions 1. Bit 0 and bits 2 to 6 must be fixed to 0.**

**2. MCC can be set only when the subsystem clock is selected as the CPU clock.**

**3. Never set MCC when an external clock is applied. This is because the X2 pin is pulled up to $V_{DD}$.**

**Remarks 1.** $f_X$ : Main system clock oscillation frequency

**2.** $f_{XT}$: Subsystem clock oscillation frequency

**3.** The parenthesized values apply to operation with $f_X$ = 5.0 MHz or $f_{XT}$ = 32.768 kHz.

**4.** Minimum instruction execution time: 2 $f_{CPU}$

- $f_{CPU}$ = 0.2 μs : 0.4 μs
- $f_{CPU}$ = 0.8 μs : 1.6 μs
- $f_{CPU}$ = 61 μs : 122 μs

**(2) Suboscillation mode register (SCKM)**

The SCKM selects a feedback resistor for the subsystem clock, and controls the oscillation of the clock.

It is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears SCKM to 00H.

**Figure 5-9. Format of Suboscillation Mode Register**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SCKM | 0 | 0 | 0 | 0 | 0 | 0 | FRC | SCC | FFF0H | 00H | R/W |

| FRC | Feedback resistor selection |
|---|---|
| 0 | Internal feedback resistor used |
| 1 | Internal feedback resistor not used |

| SCC | Control of subsystem clock oscillator operation |
|---|---|
| 0 | Operation enabled |
| 1 | Operation disabled |

**Caution Bits 2 to 7 must be fixed to 0.**

**(3) Subclock control register (CSS)**

The CSS specifies whether the main system or subsystem clock oscillator is to be selected. It also specifies how the CPU clock operates.

It is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears CSS to 00H.

**Figure 5-10. Format of Subclock Control Register**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CSS | 0 | 0 | CLS | CSS0 | 0 | 0 | 0 | 0 | FFF2H | 00H | R/W[Note] |

| CLS | CPU clock operation status |
|---|---|
| 0 | Operation based on the output of the divided main system clock |
| 1 | Operation based on the subsystem clock |

| CSS0 | Selection of the main system or subsystem clock oscillator |
|---|---|
| 0 | Divided output from the main system clock oscillator |
| 1 | Output form the subsystem clock oscillator |

**Note** Bit 5 is read-only.

**Caution Bits 0 to 3, 6, and 7 must be fixed to 0.**

## 5.3 16-Bit Timer Counter

### 5.3.1 16-Bit timer counter functions

16-bit timer counter 90 (TM90) has the following functions.

**(1) Timer interrupt**

An interrupt is generated if the TM90 count matches a comparison value.

**(2) Timer output**

The timer output can be controlled if the count matches a comparison value.

**(3) Count capture**

The count in TM90 is captured into the capture register in synchronization with a capture trigger.

**(4) Buzzer output**

The buzzer output can be controlled if the count matches the comparison value.

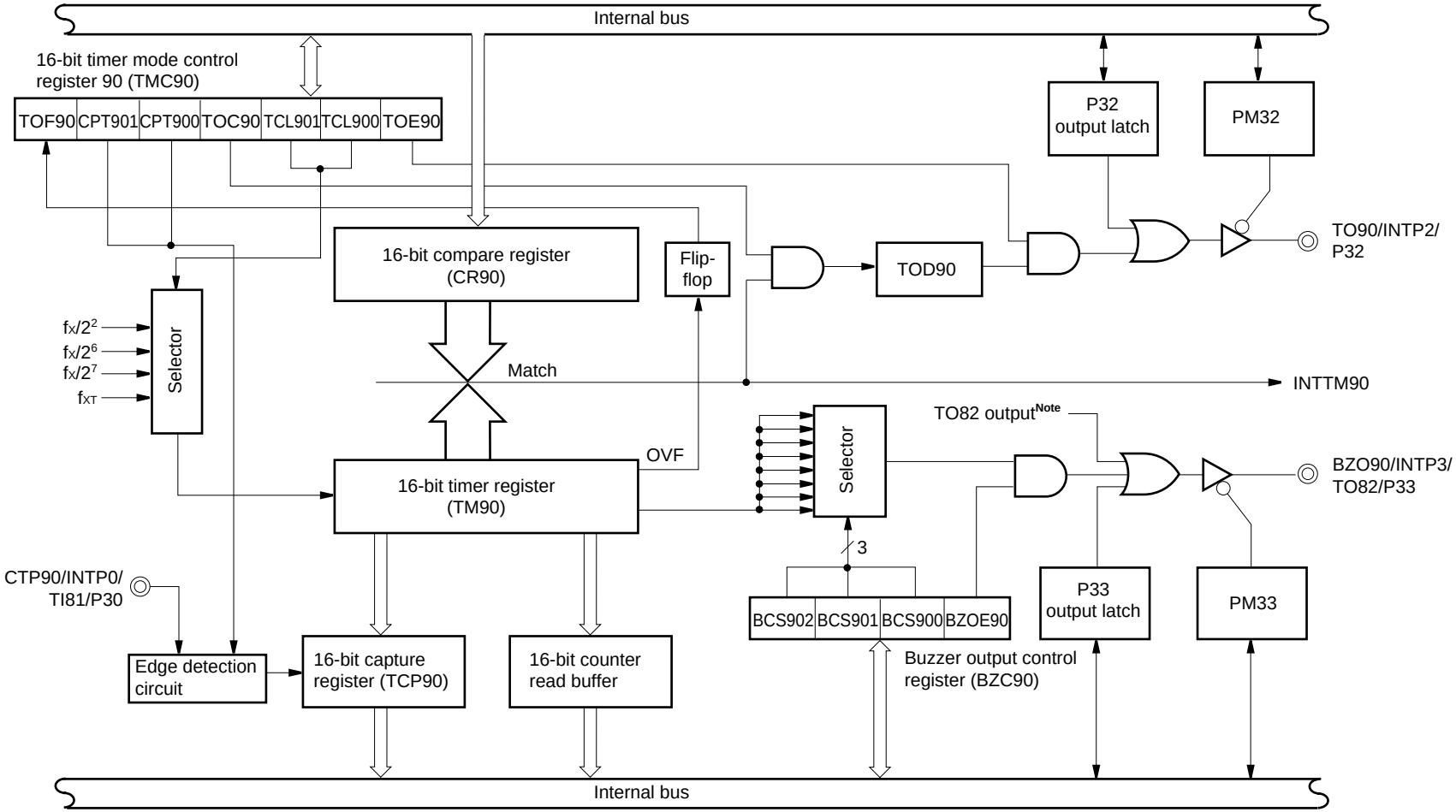
### 5.3.2 16-bit timer counter configuration

16-bit timer counter 90 (TM90) consists of the following hardware.

**Table 5-5. 16-Bit Timer Counter 90 Configuration**

| Item | Configuration |
|---|---|
| Timer register | 16 bits × 1 (TM90) |
| Register | Compare register 90 : 16 bits × 1 (CR90)<br>Capture register 90 : 16 bits × 1 (TCP90) |
| Timer output | 1 (TO90) |
| Control register | 16-bit timer mode control register 90 (TMC90)<br>Buzzer output control register 90 (BZC90)<br>Port mode register 2 (PM2) |

**Figure 5-11. Block Diagram of 16-Bit Timer Counter 90**




**Note** See **Figure 5-17**.

**(1) 16-bit compare register 90 (CR90)**

A value specified in CR90 is compared with the count in 16-bit timer register 90 (TM90). If they match, an interrupt request (INTTM90) is issued.

CR90 is manipulated using an 8-bit or 16-bit memory manipulation instruction. Any value from 0000H to FFFFH can be set.

A $\overline{\text{RESET}}$ input loads FFFFH into CR90.

**Cautions 1. CR90 is designed to be manipulated using a 16-bit memory manipulation instruction. It can also be manipulated using 8-bit memory manipulation instructions, however. When an 8-bit memory manipulation instruction is used to set CR90, it must be in a direct addressing access mode.**

**2. To re-set CR90 during count operation, it is necessary to disable interrupts in advance, using an interrupt mask flag register 1 (MK1). It is also necessary to disable inversion of the timer output data, using 16-bit timer mode control register 90 (TMC90).**

**(2) 16-bit timer register 90 (TM90)**

TM90 is used to count the number of pulses.

The contents of TM90 are read using an 8-bit or 16-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TM90 to 0000H.

**Cautions 1. The count becomes undefined when STOP mode is deselected, because the count operation is performed before oscillation settles.**

**2. TM90 is designed to be manipulated using a 16-bit memory manipulation instruction. It can also be manipulated using 8-bit memory manipulation instructions, however. When an 8-bit memory instruction is used to manipulate TM90, it must be in a direct addressing access mode.**

**3. When an 8-bit memory manipulation instruction is used to manipulate TM90, the lower and upper bytes must be read as a pair, in this order.**

**(3) 16-bit capture register 90 (TCP90)**

TCP90 captures the contents of 16-bit timer 90 (TM90).

It is manipulated using an 8-bit or 16-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input makes TCP90 undefined.

**Caution TCP90 is designed to be manipulated using a 16-bit memory manipulation instruction. It can also be manipulated using 8-bit memory manipulation instructions, however. When an 8-bit memory manipulation instruction is used to manipulate TCP90, it must be in a direct addressing access mode.**

**(4) 16-bit counter read buffer 90**

This buffer is used to latch and hold the count for 16-bit timer 90 (TM90).

### 5.3.3 16-bit timer counter control registers

The following three types of registers are used to control 16-bit timer counter 90 (TM90).

- 16-bit timer mode control register 90 (TMC90)
- Buzzer output control register 90 (BZC90)
- Port mode register 2 (PM2)

**(1) 16-bit timer mode control register 90 (TMC90)**

TMC90 controls the count clock and capture edge settings.

It is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TMC90 to 00H.

**Figure 5-12. Format of 16-Bit Timer Mode Control Register 90**

| Symbol | 7 | ⑥ | 5 | 4 | 3 | 2 | 1 | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TMC90 | TOD90 | TOF90 | CPT901 | CPT900 | TOC90 | TCL901 | TCL900 | TOE90 | FF48H | 00H | R/W[Note] |

| TOD90 | Timer output data |
|---|---|
| 0 | Timer output of 0 |
| 1 | Timer output of 1 |

| TOF90 | Overflow flag control |
|---|---|
| 0 | Reset or cleared by software |
| 1 | Set when the 16-bit timer overflows |

| CPT901 | CPT900 | Capture edge selection |
|---|---|---|
| 0 | 0 | Capture operation disabled |
| 0 | 1 | Captured at the rising edge at the CPT90 pin |
| 1 | 0 | Captured at the falling edge at the CPT90 pin |
| 1 | 1 | Captured at both the rising and falling edges at the CPT90 pin |

| TOC90 | Timer output data inversion control |
|---|---|
| 0 | Inversion disabled |
| 1 | Inversion enabled |

| TCL901 | TCL900 | 16-bit timer counter 90 count clock selection |
|---|---|---|
| 0 | 0 | $f_X/2^2$ (1.25 MHz) |
| 0 | 1 | $f_X/2^6$ (78.125 kHz) |
| 1 | 0 | $f_X/2^7$ (39.063 kHz) |
| 1 | 1 | $f_{XT}$ (32.768 kHz) |

| TOE90 | 16-bit timer counter 90 output control |
|---|---|
| 0 | Output disabled (port mode) |
| 1 | Output enabled |

**Note** Bit 7 is read-only.

**Remarks 1.** $f_X$ : Main system clock oscillation frequency

**2.** $f_{XT}$ : Subsystem clock oscillation frequency

**3.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz or $f_{XT}$ = 32.768 kHz.

**(2) Buzzer output control register (BZC90)**

Based on the count clock (fcl) selected with the count clock selection bits (TCL901 and TCL900), this register sets a buzzer frequency and controls square wave output.

BZC90 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears BZC90 to 00H.

**Figure 5-13. Format of Buzzer Output Control Register**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BZC90 | 0 | 0 | 0 | 0 | BCS902 | BCS901 | BCS900 | BZOE90 | FF49H | 00H | R/W |

| BZOE90 | Buzzer port output control |
|---|---|
| 0 | Disables buzzer port output. |
| 1 | Enables buzzer port output.[Note] |

| BCS902 | BCS901 | BCS900 | Buzzer frequency | | | |
|---|---|---|---|---|---|---|
| | | | fcl = $f_X/2^2$ | fcl = $f_X/2^6$ | fcl = $f_X/2^7$ | fcl = $f_{XT}$ |
| 0 | 0 | 0 | fcl/$2^4$ (78.125 kHz) | fcl/$2^4$ (4,883 Hz) | fcl/$2^4$ (2,441 Hz) | fcl/$2^4$ (2,048 Hz) |
| 0 | 0 | 1 | fcl/$2^5$ (39.063 kHz) | fcl/$2^5$ (2,441 Hz) | fcl/$2^5$ (1,221 Hz) | fcl/$2^5$ (1,024 Hz) |
| 0 | 1 | 0 | fcl/$2^8$ (4,883 Hz) | fcl/$2^8$ (305 Hz) | fcl/$2^8$ (153 Hz) | fcl/$2^8$ (128 Hz) |
| 0 | 1 | 1 | fcl/$2^9$ (2,441 Hz) | fcl/$2^9$ (153 Hz) | fcl/$2^9$ (76 Hz) | fcl/$2^9$ (64 Hz) |
| 1 | 0 | 0 | fcl/$2^{10}$ (1,221 Hz) | fcl/$2^{10}$ (76 Hz) | fcl/$2^{10}$ (38 Hz) | fcl/$2^{10}$ (32 Hz) |
| 1 | 0 | 1 | fcl/$2^{11}$ (610 Hz) | fcl/$2^{11}$ (38 Hz) | fcl/$2^{11}$ (19 Hz) | fcl/$2^{11}$ (16 Hz) |
| 1 | 1 | 0 | fcl/$2^{12}$ (305 Hz) | fcl/$2^{12}$ (19 Hz) | fcl/$2^{12}$ (10 Hz) | fcl/$2^{12}$ (8 Hz) |
| 1 | 1 | 1 | fcl/$2^{13}$ (153 Hz) | fcl/$2^{13}$ (10 Hz) | fcl/$2^{13}$ (5 Hz) | fcl/$2^{13}$ (4 Hz) |

**Note** When setting BZOE90 to 1, TOE82 must be fixed to 0. (See **Figure 5-20**.)

**Cautions 1. Bits 4 to 7 must be fixed to 0.**

**2. If the subclock is selected as the count clock (TCL901 = 1, TCL900 = 1: see Figure 5-12), the subclock is not synchronized when buzzer port output is enabled. In this case, the capture function and TM90 register read function are disabled. In addition, the count value of the TM90 register is undefined.**

**Remarks 1.** $f_X$ : Main system clock oscillation frequency

**2.** $f_{XT}$: Subsystem clock oscillation frequency

**3.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz or $f_{XT}$ = 32.768 kHz.

**(3) Port mode register 3 (PM3)**

PM3 is used to specify port 3 input/output on a bit-by-bit basis.

When the P32/INTP2/TO90 pin is used for timer output, set 0 in the output latch of PM32 and P32. When the P33/INTP3/TO82/BZO90 pin is used for buzzer output[Note], set 0 in the output latch of the PM33 and P33.

PM3 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input loads FFH into PM3.

**Note** Never output the TO82 and BZO90 signals at the same time.

**Figure 5-14. Format of Port Mode Register 3**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PM3 | 1 | 1 | 1 | 1 | PM33 | PM32 | PM31 | PM30 | FF23H | FFH | R/W |

| PM3n | P3n pin I/O mode (n = 2 or 3) |
|---|---|
| 0 | Output mode (output buffer ON) |
| 1 | Input mode (output buffer OFF) |

## 5.4 8-Bit Timer/Event Counter

### 5.4.1 8-bit timer/event counter functions

Devices of the μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y have two timer/event counters (TM80 and TM81) and one timer counter (TM82). Readers who are seeking a description of TM82 should read the term "timer/event counter" as "timer counter."

The 8-bit timer/event counters (TM80, TM81, and TM82) have the following functions.

**(1) 8-bit interval timer (TM80, TM81, and TM82)**

This timer causes interrupts to be issued at specified intervals.

**(2) External event counter (TM80 and TM81)**

This counter is used to count the number of pulses input from an external source.

**(3) Square wave output (TM80, TM81, and TM82)**

A square wave of any frequency can be output.

**(4) PWM output (TM80, TM81, and TM82)**

PWM output with an 8-bit resolution is supported.

**Table 5-6. 8-Bit Timer/Event Counter Types and Functions**

| | | TM80 | TM81 | TM82 |
|---|---|---|---|---|
| Type | Interval timer | One channel | One channel | One channel |
| | External event counter | O | O | × |
| Function | Timer output | O | O | O |
| | Square wave output | O | O | O |
| | PWM output | O | O | O |
| | Interrupt request | O | O | O |

### 5.4.2 8-bit timer/event counter configuration

The 8-bit timer/event counter consists of the following hardware.

**Table 5-7. 8-Bit Timer/Event Counter Configuration**

| Item | Configuration |
|---|---|
| Timer register | 8 bits × 3 (TM80, TM81, TM82) |
| Register | Compare registers: 8 bits × 3 (CR80, CR81, CR82) |
| Timer output | 3 (TO80, TO81, TO82) |
| Control register | 8-bit timer mode control registers 80, 81, 82 (TMC80, TMC81, TMC82)<br>Port mode registers 2, 3 (PM2, PM3) |

**Figure 5-15. Block Diagram of 8-Bit Timer/Event Counter 80**

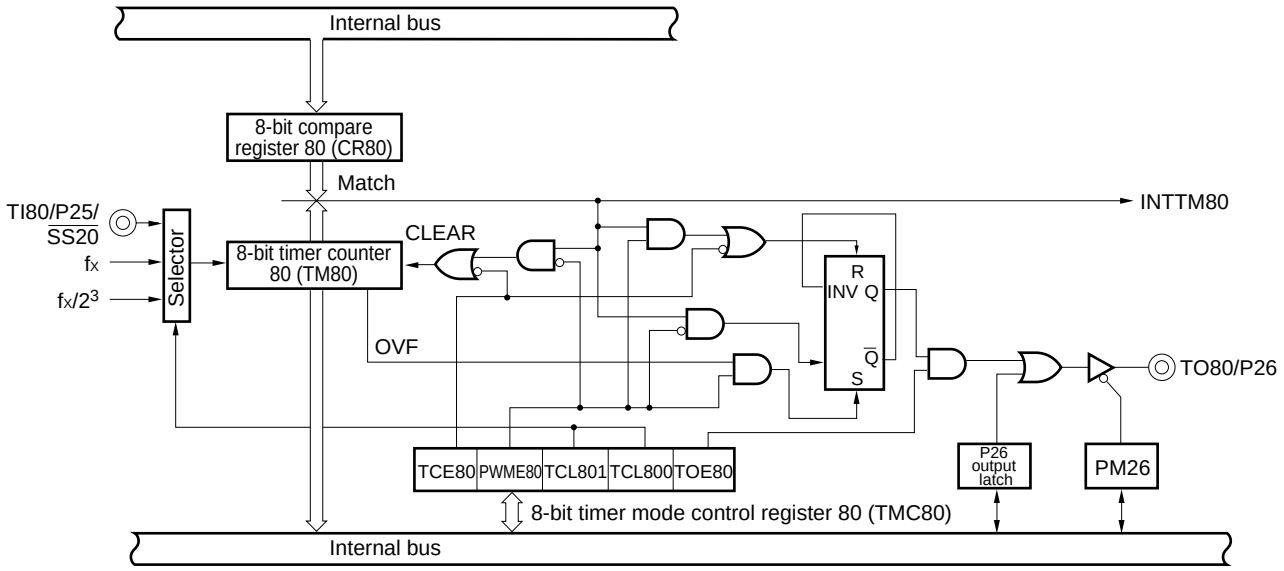



**Figure 5-16. Block Diagram of 8-Bit Timer/Event Counter 81**

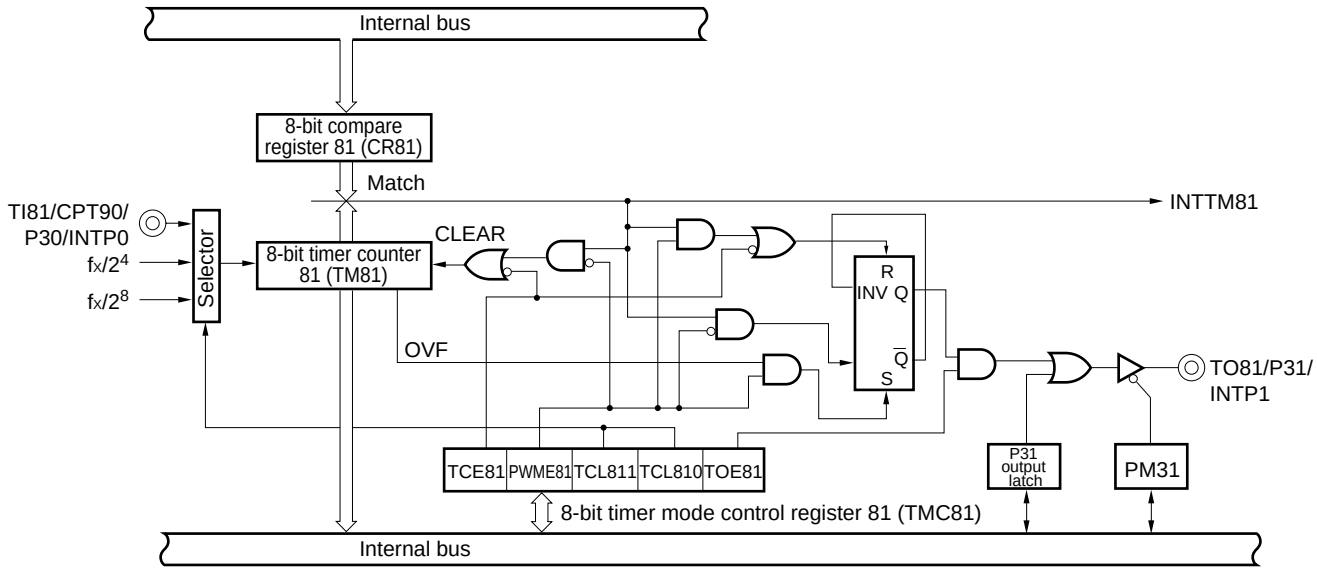
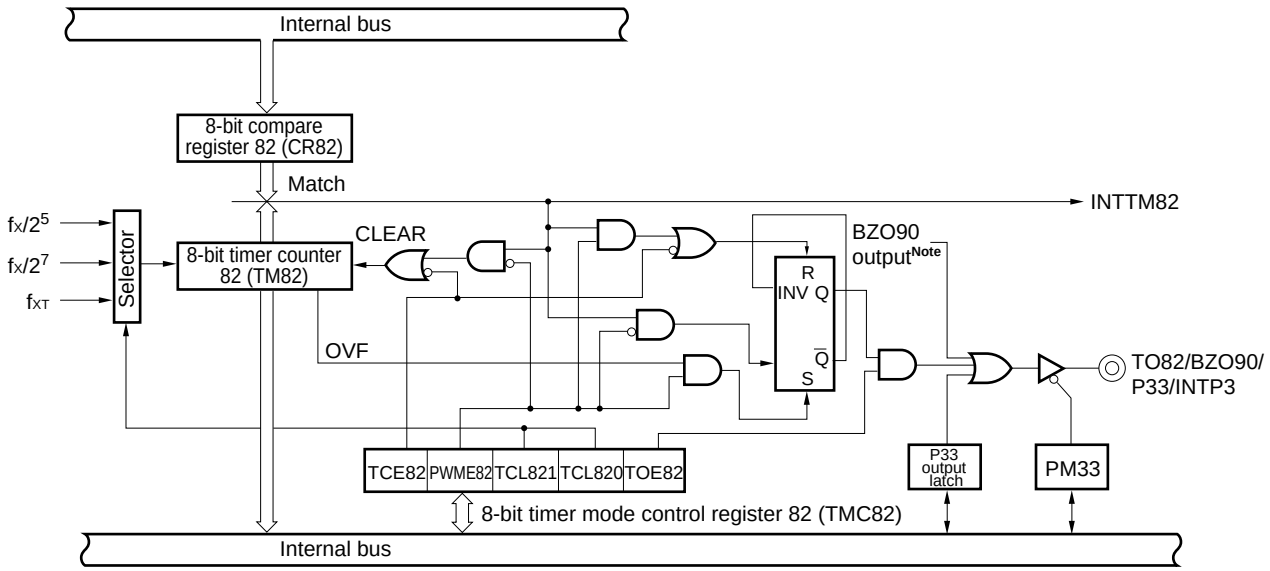


**Figure 5-17. Block Diagram of 8-Bit Timer Counter 82**




**Note** See **Figure 5-11**.

**(1) 8-bit compare register 8n (CR8n)**

A value specified in CR8n is compared with the count in 8-bit timer register 8n (TM8n). If they match, an interrupt request (INTTM8n) is issued.

CR8n is manipulated using an 8-bit memory manipulation instruction. Any value from 00H to FFH can be set.

A $\overline{\text{RESET}}$ input makes CR8n undefined.

**Remark** n = 0 to 2

**(2) 8-bit timer register 8n (TM8n)**

TM8n is used to count the number of pulses.

Its contents are read using an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TM8n to 00H.

**Remark** n = 0 to 2

### 5.4.3 8-bit timer/event counter control registers

The following two types of registers are used to control the 8-bit timer/event counter.

- 8-bit timer mode control registers 80, 81, and 82 (TMC80, TMC81, and TMC82)
- Port mode registers 2 and 3 (PM2 and PM3)

#### (1) 8-bit timer mode control register 80 (TMC80)

TMC80 determines whether to enable or disable 8-bit timer register 80 (TM80) and specifies the count clock for TM80. It also controls the operation of the output control circuit of 8-bit timer counter 80.

TMC80 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TMC80 to 00H.

**Figure 5-18. Format of 8-Bit Timer Mode Control Register 80**

| Symbol | ⑦ | 6 | 5 | 4 | 3 | 2 | 1 | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TMC80 | TCE80 | PWME80 | 0 | 0 | 0 | TCL801 | TCL800 | TOE80 | FF53H | 00H | R/W |

| TCE80 | 8-bit timer register 80 operation control |
|---|---|
| 0 | Operation disabled (TM80 is cleared to 0.) |
| 1 | Operation enabled |

| PWME80 | PWM output selection |
|---|---|
| 0 | Counter operation |
| 1 | PWM output |

| TCL801 | TCL800 | 8-bit timer register 80 count clock selection |
|---|---|---|
| 0 | 0 | $f_X$ (5.0 MHz) |
| 0 | 1 | $f_X/2^3$ (625 kHz) |
| 1 | 0 | Rising edge of TI80 |
| 1 | 1 | Falling edge of TI80 |

| TOE80 | 8-bit timer counter 80 output control |
|---|---|
| 0 | Output disabled (port mode) |
| 1 | Output enabled |

**Cautions 1. Always stop the timer before setting TMC80.**

**2. For PWM mode operation, the interrupt mask flag (TMMK80) must be set.**

**Remarks 1.** $f_X$: Main system clock oscillation frequency

**2.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz.

**(2) 8-bit timer mode control register 81 (TMC81)**

TMC81 determines whether to enable or disable 8-bit timer register 81 (TM81) and specifies the count clock for TM81. It also controls the operation of the output control circuit of 8-bit timer counter 81.

TMC81 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TMC81 to 00H.

**Figure 5-19. Format of 8-Bit Timer Mode Control Register 81**

| Symbol | ⑦ | 6 | 5 | 4 | 3 | 2 | 1 | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TMC81 | TCE81 | PWME81 | 0 | 0 | 0 | TCL811 | TCL810 | TOE81 | FF57H | 00H | R/W |

| TCE81 | 8-bit timer register 81 operation control |
|---|---|
| 0 | Operation disabled (TM81 is cleared to 0.) |
| 1 | Operation enabled |

| PWME81 | PWM output selection |
|---|---|
| 0 | Counter operation |
| 1 | PWM output |

| TCL811 | TCL810 | 8-bit timer register 81 count clock selection |
|---|---|---|
| 0 | 0 | $f_X/2^4$ (312.5 kHz) |
| 0 | 1 | $f_X/2^8$ (19.5 kHz) |
| 1 | 0 | Rising edge of TI81 |
| 1 | 1 | Falling edge of TI81 |

| TOE81 | 8-bit timer counter 81 output control |
|---|---|
| 0 | Output disabled (port mode) |
| 1 | Output enabled |

**Cautions 1. Always stop the timer before setting TMC81.**

**2. For PWM mode operation, the interrupt mask flag (TMMK81) must be set.**

**Remarks 1.** $f_X$: Main system clock oscillation frequency

**2.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz.

**(3) 8-bit timer mode control register 82 (TMC82)**

TMC82 determines whether to enable or disable 8-bit timer register 82 (TM82) and specifies the count clock for TM82. It also controls the operation of the output control circuit of 8-bit timer counter 82.

TMC82 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TMC82 to 00H.

**Figure 5-20. Format of 8-Bit Timer Mode Control Register 82**

| Symbol | ⑦ | 6 | 5 | 4 | 3 | 2 | 1 | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TMC82 | TCE82 | PWME82 | 0 | 0 | 0 | TCL821 | TCL820 | TOE82 | FF5BH | 00H | R/W |

| TCE82 | 8-bit timer register 82 operation control |
|---|---|
| 0 | Operation disabled (TM82 is cleared to 0.) |
| 1 | Operation enabled |

| PWME82 | PWM output selection |
|---|---|
| 0 | Counter operation |
| 1 | PWM output |

| TCL821 | TCL820 | 8-bit timer register 82 count clock selection |
|---|---|---|
| 0 | 0 | $f_X/2^5$ (156.25 kHz) |
| 0 | 1 | $f_X/2^7$ (39.1 kHz) |
| 1 | 0 | $f_{XT}$ (32.768 kHz) |
| 1 | 1 | Not to be set |

| TOE82 | 8-bit timer counter 82 output control |
|---|---|
| 0 | Output disabled (port mode) |
| 1 | Output enabled Note |

**Note** When TOE82 is set to 1, BZOE90 must be set to 0 (see **Figure 5-13**).

**Cautions 1. Always stop the timer before setting TMC82.**

**2. For PWM mode operation, the interrupt mask flag (TMMK82) must be set.**

**Remarks 1.** $f_X$ : Main system clock oscillation frequency

**2.** $f_{XT}$: Subsystem clock oscillation frequency

**3.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz or $f_{XT}$ = 32.768 kHz.

**(4) Port mode registers 2 and 3 (PM2 and PM3)**

PM2 and PM3 specify whether each bit of port 2 and port 3 is used for input or output.

To use the P26/TO80 pin for timer output, the PM26 and P26 output latches must be reset to 0.

To use the P31/TO81/INTP1 pin for timer output, the PM31 and P31 output latches must be reset to 0.

To use the P33/INTP3/TO82/BZO90 pin for timer output, the PM33 and P33 output latches must be reset to 0.

PM2 and PM3 are manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input loads FFH into PM2 and PM3.

**Figure 5-21. Format of Port Mode Register 2**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PM2 | 1 | PM26 | PM25 | PM24 | PM23 | PM22 | PM21 | PM20 | FF22H | FFH | R/W |

| PM26 | P26 pin input/output mode selection |
|---|---|
| 0 | Output mode (output buffer ON) |
| 1 | Input mode (output buffer OFF) |

**Figure 5-22. Format of Port Mode Register 3**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| PM3 | 1 | 1 | 1 | 1 | PM33 | PM32 | PM31 | PM30 | FF23H | FFH | R/W |

| PM31 | P31 pin input/output mode selection |
|---|---|
| 0 | Output mode (output buffer ON) |
| 1 | Input mode (output buffer OFF) |

| PM33 | P33 pin input/output mode selection |
|---|---|
| 0 | Output mode (output buffer ON) |
| 1 | Input mode (output buffer OFF) |

## 5.5 Clock Timer

### 5.5.1 Clock timer functions

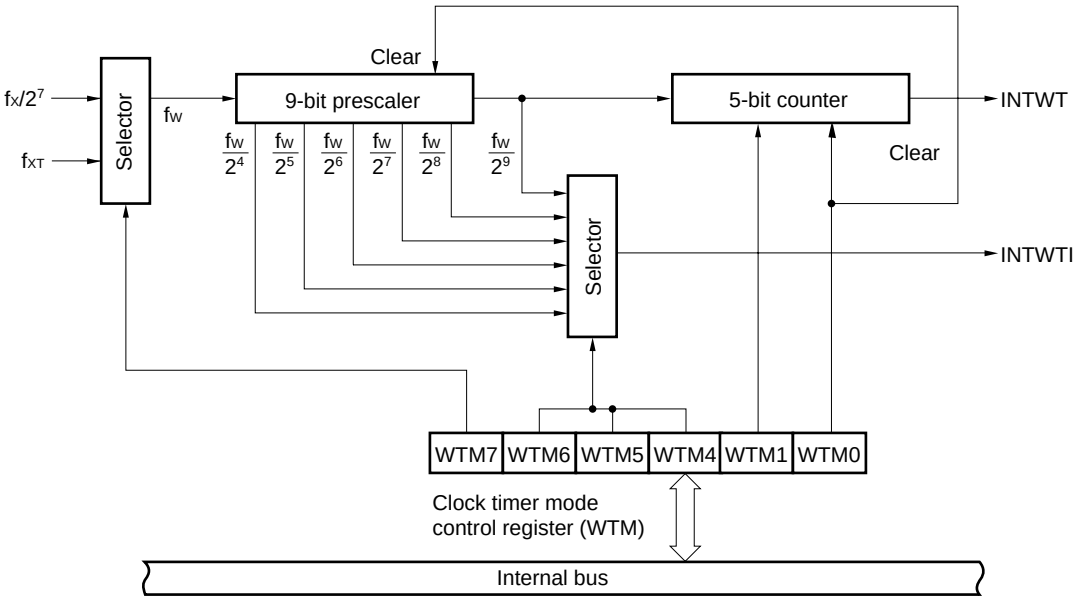
The clock timer has the following functions.

- Clock timer
- Interval timer

The clock and interval timers can be used at the same time.

Figure 5-23 is a block diagram of the clock timer.

**Figure 5-23. Block Diagram of Clock Timer**



**(1) Clock timer**

The 4.19-MHz main system clock or 32.768-kHz subsystem clock is used to issue an interrupt request (INTWT) at 0.5-second intervals.

**Caution When the main system clock is operating at 5.0 MHz, it cannot be used to generate a 0.5-second interval. In this case, the subsystem clock, which operates at 32.768 kHz, should be used instead.**

**(2) Interval timer**

The interval timer is used to generate an interrupt request (INTWTI) at specified intervals.

**Table 5-8. Interval Generated Using the Interval Timer**

| Interval | Operation at $f_X$ = 5.0 MHz | Operation at $f_X$ = 4.19 MHz | Operation at $f_{XT}$ = 32.768 kHz |
|---|---|---|---|
| $2^4 \times 1/f_W$ | 409.6 μs | 489 μs | 488 μs |
| $2^5 \times 1/f_W$ | 819.2 μs | 978 μs | 977 μs |
| $2^6 \times 1/f_W$ | 1.64 ms | 1.96 ms | 1.95 ms |
| $2^7 \times 1/f_W$ | 3.28 ms | 3.91 ms | 3.91 ms |
| $2^8 \times 1/f_W$ | 6.55 ms | 7.82 ms | 7.81 ms |
| $2^9 \times 1/f_W$ | 13.1 ms | 15.6 ms | 15.6 ms |

**Remark** $f_W$ : Clock timer clock frequency ($f_X/2^7$ or $f_{XT}$)
$f_X$ : Main system clock oscillation frequency
$f_{XT}$ : Subsystem clock oscillation frequency

### 5.5.2 Clock timer configuration

The clock timer consists of the following hardware.

**Table 5-9. Clock Timer Configuration**

| Item | Configuration |
|---|---|
| Counter | 5 bits |
| Prescaler | 9 bits |
| Control register | Clock timer mode control register (WTM) |

### 5.5.3 Clock timer control register

The clock timer mode control register (WTM) is used to control the clock timer.

- Clock timer mode control register (WTM)

  The WTM selects a count clock for the clock timer and specifies whether to enable clocking of the timer. It also specifies the prescaler interval and how the 5-bit counter is controlled.

  The WTM is manipulated using a 1-bit or 8-bit memory manipulation instruction.

  A $\overline{\text{RESET}}$ input clears the WTM to 00H.

**Figure 5-24. Format of Clock Timer Mode Control Register**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| WTM | WTM7 | WTM6 | WTM5 | WTM4 | 0 | 0 | WTM1 | WTM0 | FF4AH | 00H | R/W |

| WTM7 | Clock timer count clock selection |
|---|---|
| 0 | $f_X/2^7$ (39.1 kHz) |
| 1 | $f_{XT}$ (32.768 kHz) |

| WTM6 | WTM5 | WTM4 | Prescaler interval selection |
|---|---|---|---|
| 0 | 0 | 0 | $2^4/f_W$ |
| 0 | 0 | 1 | $2^5/f_W$ |
| 0 | 1 | 0 | $2^6/f_W$ |
| 0 | 1 | 1 | $2^7/f_W$ |
| 1 | 0 | 0 | $2^8/f_W$ |
| 1 | 0 | 1 | $2^9/f_W$ |
| Other settings | | | Not to be set |

| WTM1 | 5-bit counter operation control |
|---|---|
| 0 | Cleared after stop |
| 1 | Started |

| WTM0 | Clock timer operation |
|---|---|
| 0 | Operation disabled (both prescaler and timer cleared) |
| 1 | Operation enabled |

**Remarks 1.** $f_W$ : Clock timer clock frequency ($f_X/2^7$ or $f_{XT}$)

**2.** $f_X$ : Main system clock oscillation frequency

**3.** $f_{XT}$: Subsystem clock oscillation frequency

**4.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz or $f_{XT}$ = 32.768 kHz.

## 5.6 Watchdog Timer

### 5.6.1 Watchdog timer functions

The watchdog timer has the following functions.

**(1) Watchdog timer**

The watchdog timer is used to detect unintended program loops. If an unintended program loop is detected, a nonmaskable interrupt or $\overline{\text{RESET}}$ signal is generated.

**(2) Interval timer**

The interval timer is used to generate interrupts at specified intervals.

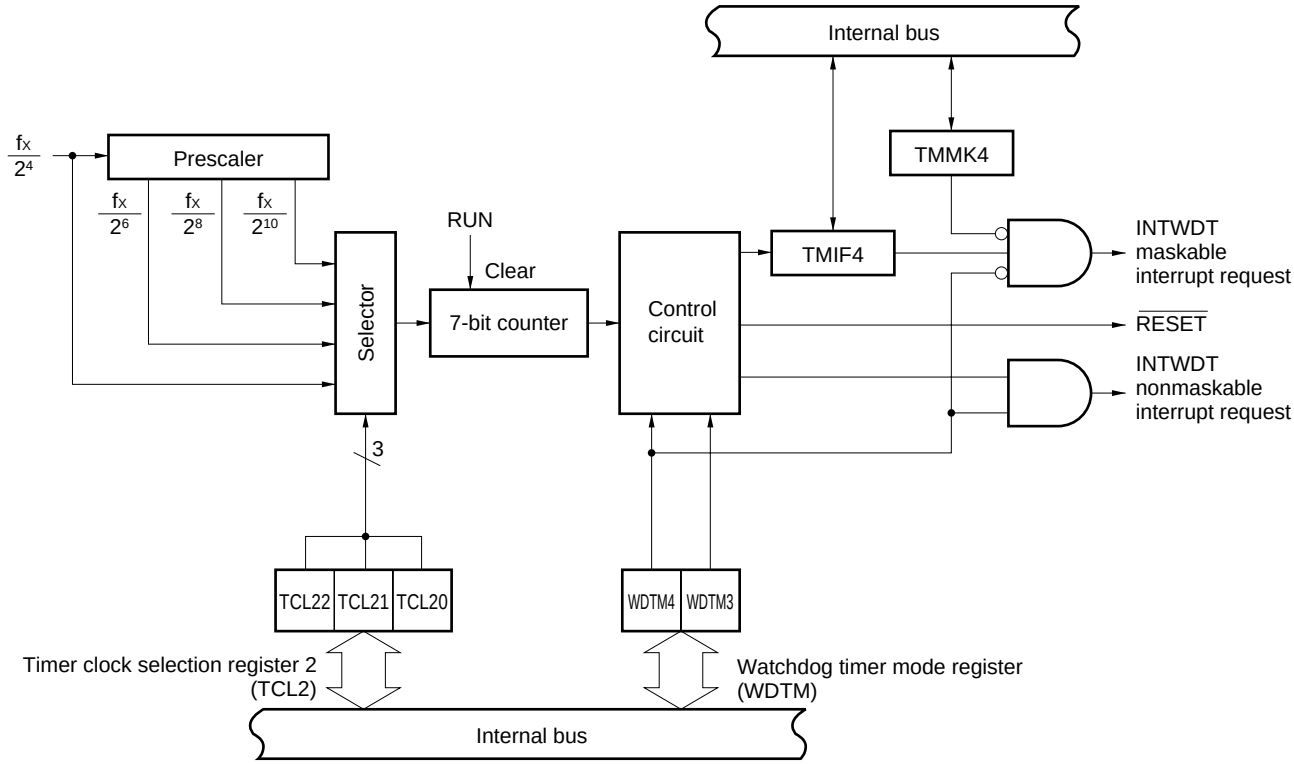
### 5.6.2 Watchdog timer configuration

The watchdog timer consists of the following hardware.

**Table 5-10. Watchdog Timer Configuration**

| Item | Configuration |
|---|---|
| Control register | Timer clock selection register 2 (TCL2)<br>Watchdog timer mode register (WDTM) |

**Figure 5-25. Block Diagram of Watchdog Timer**



### 5.6.3 Watchdog timer control registers

The following two types of registers are used to control the watchdog timer.

- Timer clock selection register 2 (TCL2)
- Watchdog timer mode register (WDTM)

#### (1) Timer clock selection register 2 (TCL2)

TCL2 specifies the count clock for the watchdog timer.

TCL2 is manipulated using an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears TCL2 to 00H.

**Figure 5-26. Format of Timer Clock Selection Register 2**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| TCL2 | 0 | 0 | 0 | 0 | 0 | TCL22 | TCL21 | TCL20 | FF42H | 00H | R/W |

| TCL22 | TCL21 | TCL20 | Watchdog timer count clock selection | Interval time |
|---|---|---|---|---|
| 0 | 0 | 0 | $f_X/2^4$ (312.5 kHz) | $2^{11}/f_X$ (410 μs) |
| 0 | 1 | 0 | $f_X/2^6$ (78.1 kHz) | $2^{13}/f_X$ (1.64 ms) |
| 1 | 0 | 0 | $f_X/2^8$ (19.5 kHz) | $2^{15}/f_X$ (6.55 ms) |
| 1 | 1 | 0 | $f_X/2^{10}$ (4.88 kHz) | $2^{17}/f_X$ (26.2 ms) |
| Other settings | | | Not to be set | |

**Remarks 1.** $f_X$: Main system clock oscillation frequency

**2.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz.

**(2) Watchdog timer mode register (WDTM)**

The WDTM specifies the watchdog timer operation mode and whether to enable or disable counting.

The WDTM is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears the WDTM to 00H.

**Figure 5-27. Format of Watchdog Timer Mode Register**

| Symbol | ⑦ | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| WDTM | RUN | 0 | 0 | WDTM4 | WDTM3 | 0 | 0 | 0 | FFF9H | 00H | R/W |

| RUN | Watchdog timer operation selection[Note 1] |
|---|---|
| 0 | Stops counting. |
| 1 | Clears the counter and causes it to start. |

| WDTM4 | WDTM3 | Watchdog timer operation mode selection[Note 2] |
|---|---|---|
| 0 | 0 | Operation disabled |
| 0 | 1 | Internal timer mode (When an overflow occurs, a maskable interrupt is issued.)[Note 3] |
| 1 | 0 | Watchdog timer mode 1 (When an overflow occurs, a nonmaskable interrupt is issued.) |
| 1 | 1 | Watchdog timer mode 2 (When an overflow occurs, a reset operation is started.) |

**Notes 1.** Once the RUN bit has been set (1), it is impossible to zero-clear it by software. So, once counting begins, it cannot be stopped by any means other than a $\overline{\text{RESET}}$ input.

**2.** Once WDTM3 and WDTM4 have been set (1), it is impossible to zero-clear them by software.

**3.** The interval timer starts operating when the RUN bit is set to 1.

**Cautions 1. If the RUN bit is set to 1, and the watchdog timer is cleared, the actual overflow time becomes 0.8% (maximum) less than the time specified in timer clock selection register 2.**

**2. To use watchdog timer mode 1 or 2, ensure that the interrupt request flag (TMIF4) is set to 0, before setting WDTM4 to 1. If TMIF4 is set to 1, selecting mode 1 or 2 causes a nonmaskable interrupt to be issued at the instant rewriting ends.**

## 5.7 A/D Converter

A/D converters support different conversion resolutions, depending on the microcontroller model, as shown below:

- A/D converters with an 8-bit resolution: for μPD789166Y and μPD789167Y
- A/D converters with a 10-bit resolution: for μPD789176Y and μPD789177Y

### 5.7.1 A/D converter functions

The A/D converter converts input analog voltages to digital signals with an 8-bit or 10-bit resolution. It can control up to eight analog input channels (ANI0 to ANI7).

A/D conversion can be started only by software.

One of analog inputs ANI0 to ANI7 is selected for A/D conversion. A/D conversion is performed repeatedly, with an interrupt request (INTAD0) being issued each time an A/D conversion is completed.

**Caution In standby mode, the A/D converter operation is disabled.**

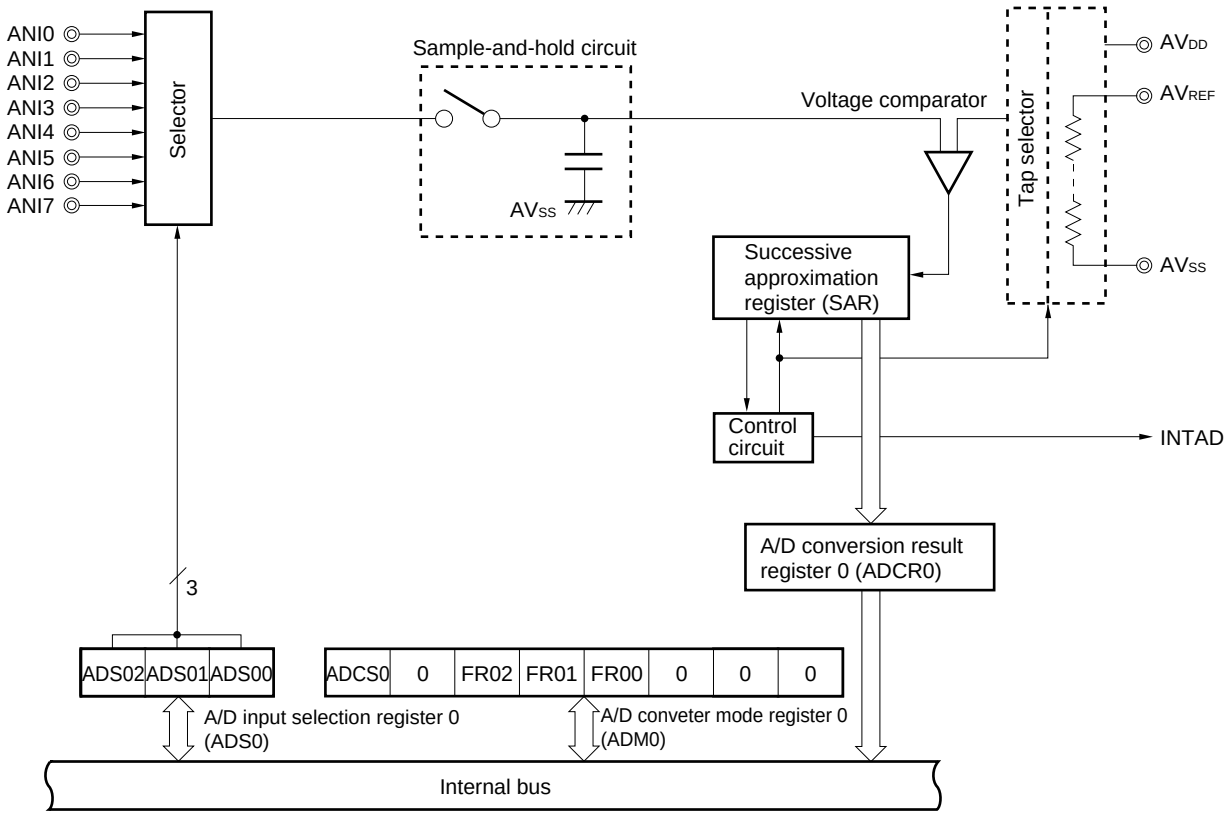
### 5.7.2 A/D converter configuration

The A/D converter consists of the following hardware.

**Table 5-11. A/D Converter Configuration**

| Item | Configuration |
|---|---|
| Analog input | 8 channels (ANI0 to ANI7) |
| Register | Successive approximation register (SAR)<br>A/D conversion result register 0 (ADCR0) |
| Control register | A/D converter mode register 0 (ADM0)<br>A/D input selection register 0 (ADS0) |

**Figure 5-28. Block Diagram of A/D Converter**




**(1) Successive approximation register (SAR)**

The SAR receives the result of comparing an analog input voltage and a voltage at a voltage tap (comparison voltage), received from the serial resistor string, starting from the most significant bit (MSB).

Upon receiving all the bits, down to the least significant bit (LSB), that is, upon the completion of A/D conversion, the SAR sends its contents to the A/D conversion result register.

**(2) A/D conversion result register 0 (ADCR0)**

The ADCR holds the result of A/D conversion. Each time A/D conversion ends, the conversion result received from the successive approximation register is loaded into the ADCR0.

For the μPD789166Y and μPD789167Y (featuring 8-bit A/D converters), the value of ADCR0 is read using an 8-bit memory manipulation instruction.

For the μPD789176Y and μPD789177Y (featuring 10-bit A/D converters), the value of ADCR0 is read using a 16-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input makes ADCR0 undefined.

**Caution When 8-bit A/D converters are used (for the μPD789166Y and μPD789167Y), this register can be accessed only in 8-bit mode. In this case, the address is assumed to be FF15H.**
**When 10-bit A/D converters are used (for the μPD789176Y and μPD789177Y), this register can be accessed only in 16-bit mode.**
**When the μPD78F9177Y is used as flash memory for the μPD789166Y or μPD789167Y, 8-bit access is allowed. However, only those object files generated by an assembler used with the μPD789166Y or μPD789167Y are supported for this access.**

**(3) Sample-and-hold circuit**

The sample-and-hold circuit samples consecutive analog inputs from the input circuit, one by one, and sends them to the voltage comparator. The sampled analog input voltage is held during A/D conversion.

**(4) Voltage comparator**

The voltage comparator compares an analog input with the voltage output by the serial resistor string.

**(5) Serial resistor string**

The serial resistor string is configured between $AV_{REF}$ and $AV_{SS}$. It generates the reference voltages against which analog inputs are compared.

**(6) ANI0 to ANI7 pins**

Pins ANI0 to ANI7 are 8-channel analog input pins for the A/D converter. They are used to receive the analog signals to be subject to A/D conversion.

**Caution Do not supply pins ANI0 to ANI7 with voltages that fall outside the rated range. If a voltage greater than $AV_{REF}$ or less than $AV_{SS}$ (even if within the absolute maximum rating) is supplied to any of these pins, the conversion value for the corresponding channel will be undefined. Furthermore, the conversion values for the other channels may also be affected.**

**(7) $AV_{REF}$ pin**

The $AV_{REF}$ pin is a reference voltage pin for the A/D converter.

Signals received at pins ANI0 to ANI7 are converted to digital signals while referencing the voltage across the $AV_{REF}$ and $AV_{SS}$ pins.

**(8) $AV_{SS}$ pin**

The $AV_{SS}$ pin is a ground potential pin for the A/D converter. This pin must be held at the same potential as the $V_{SS}$ pin, even while the A/D converter is not being used.

**(9) $AV_{DD}$ pin**

The $AV_{DD}$ pin is an analog power supply pin for the A/D converter. This pin must be held at the same potential as the $V_{DD}$ pin, even while the A/D converter is not being used.

### 5.7.3 A/D converter control registers

The following two types of registers are used to control the A/D converter.

- A/D converter mode register 0 (ADM0)
- A/D input selection register 0 (ADS0)

**(1) A/D converter mode register 0 (ADM0)**

ADM0 specifies the conversion time for analog inputs. It also specifies whether to enable conversion.

ADM0 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears ADM0 to 00H.

**Figure 5-29. Format of A/D Converter Mode Register 0**

| Symbol | ⑦ | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ADM0 | ADCS0 | 0 | FR02 | FR01 | FR00 | 0 | 0 | 0 | FF80H | 00H | R/W |

| ADCS0 | A/D conversion control |
|---|---|
| 0 | Conversion disabled |
| 1 | Conversion enabled |

| FR02 | FR01 | FR00 | A/D conversion time selection[Note 1] |
|---|---|---|---|
| 0 | 0 | 0 | 144/fx (28.8 μs) |
| 0 | 0 | 1 | 120/fx (24 μs) |
| 0 | 1 | 0 | 96/fx (19.2 μs) |
| 1 | 0 | 0 | 72/fx (14.4 μs) |
| 1 | 0 | 1 | 60/fx (Not to be set[Note 2]) |
| 1 | 1 | 0 | 48/fx (Not to be set[Note 2]) |
| Other settings | | | Not to be set |

**Notes 1.** The specifications of FR02, FR01, and FR00 must be such that the A/D conversion time is at least 14 μs.

**2.** These bit combinations must not be used, as the A/D conversion time will fall below 14 μs.

**Cautions 1. The result of conversion performed immediately after bit 7 (ADCS0) is set is undefined.**

**2. Bits 0 to 2 and bit 6 must be fixed to 0.**

**Remarks 1.** fx: Main system clock oscillation frequency

**2.** The parenthesized values apply to operation at fx = 5.0 MHz.

**(2) A/D input selection register 0 (ADS0)**

ADS0 specifies the port used to input the analog voltages to be converted to a digital signal. The ADS0 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears ADS0 to 00H.

**Figure 5-30. Format of A/D Input Selection Register 0**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ADS0 | 0 | 0 | 0 | 0 | 0 | ADS02 | ADS01 | ADS00 | FF84H | 00H | R/W |

| ADS02 | ADS01 | ADS00 | Analog input channel specification |
|---|---|---|---|
| 0 | 0 | 0 | ANI0 |
| 0 | 0 | 1 | ANI1 |
| 0 | 1 | 0 | ANI2 |
| 0 | 1 | 1 | ANI3 |
| 1 | 0 | 0 | ANI4 |
| 1 | 0 | 1 | ANI5 |
| 1 | 1 | 0 | ANI6 |
| 1 | 1 | 1 | ANI7 |

**Caution Bits 3 to 7 must be fixed to 0.**

## 5.8 Serial Interface

### 5.8.1 Serial interface 20

**(1) Serial interface 20 functions**

Serial interface 20 has the following three types of modes.

- Operation stopped mode
- Asynchronous serial interface (UART) mode
- Three-wire serial I/O mode

**(a) Operation stopped mode**

This mode is used when serial transfer is not performed. Power consumption is minimized in this mode.

**(b) Asynchronous serial interface (UART) mode**

This mode is used to send and receive the one byte of data that follows a start bit. It supports full-duplex communication.

Serial interface 20 contains a dedicated UART baud rate generator, enabling communication over a wide range of baud rates. It is also possible to define baud rates by dividing the frequency of the input clock pulse at the ASCK20 pin.

**(c) Three-wire serial I/O mode (switchable between MSB-first and LSB-first transmission)**

This mode is used to transmit 8-bit data, using three lines: a serial clock ($\overline{SCK20}$) line and two serial data lines (SI20 and SO20).

As it supports simultaneous transmission and reception, three-wire serial I/O mode requires less processing time for data transmission than asynchronous serial interface mode.

Because, in three-wire serial I/O mode, it is possible to select whether 8-bit data transmission begins with the MSB or LSB, serial interface 20 can be connected to any device regardless of whether that device is designed for MSB-first or LSB-first transmission.

Three-wire serial I/O mode is useful for connecting peripheral I/O circuits and display controllers having conventional clock synchronous serial interfaces, such as those of the 75X/XL, 78K, and 17K series devices.

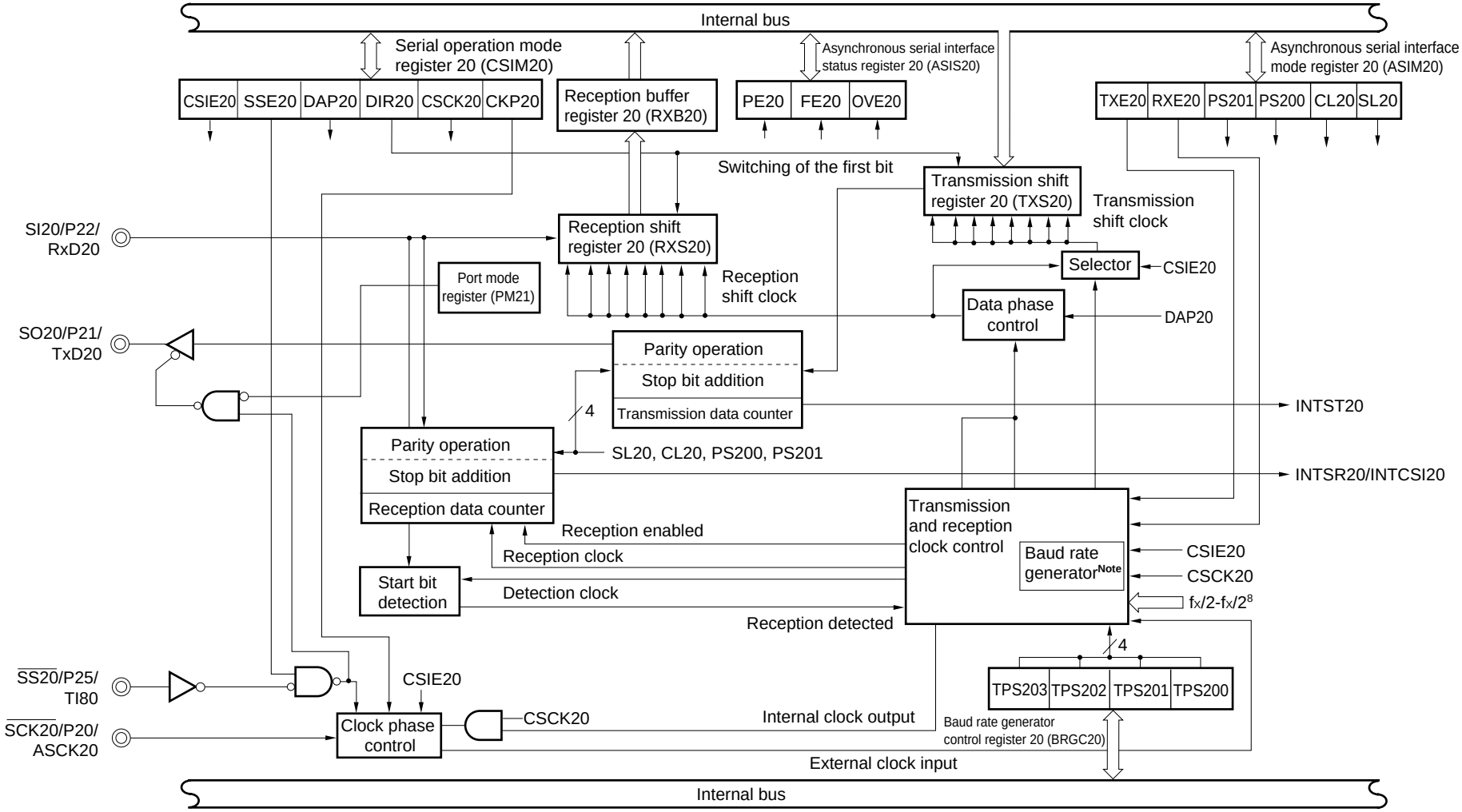
**(2) Serial interface 20 configuration**

Serial interface 20 consists of the following hardware.

**Table 5-12. Serial Interface 20 Configuration**

| Item | Configuration |
|---|---|
| Register | Transmission shift register 20 (TXS20)<br>Reception shift register 20 (RXS20)<br>Reception buffer register 20 (RXB20) |
| Control register | Serial operation mode register 20 (CSIM20)<br>Asynchronous serial interface mode register 20 (ASIM20)<br>Asynchronous serial interface status register 20 (ASIS20)<br>Baud rate generator control register 20 (BRGC20) |

**Figure 5-31. Block Diagram of Serial Interface 20**




**Note** See **Figure 5-32** for the configuration of the baud rate generator.

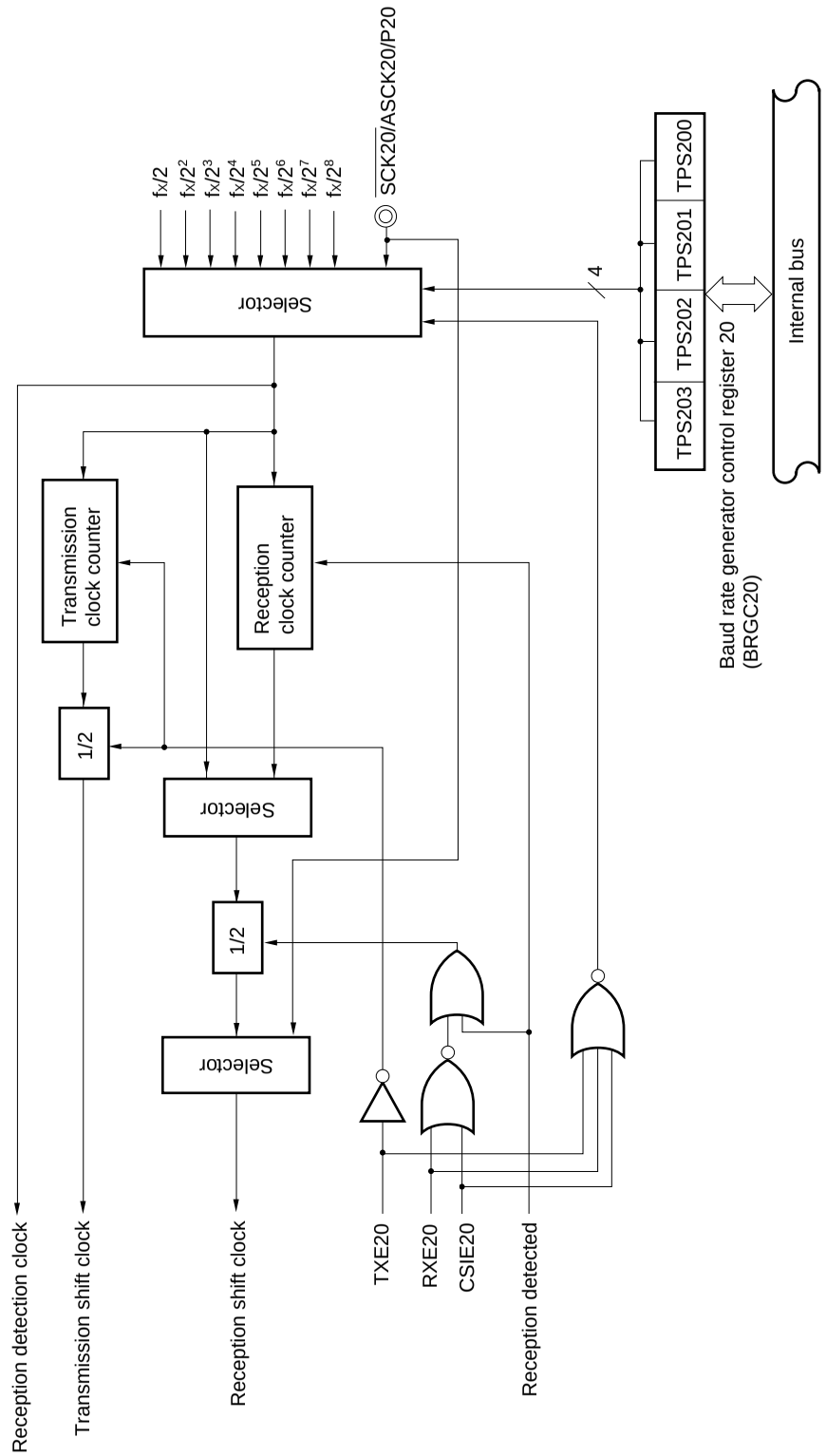



**Figure 5-32. Block Diagram of Baud Rate Generator 20**

**(a) Transmission shift register 20 (TXS20)**

TXS20 is a register in which transmission data is prepared. The transmission data is output from the TXS20 bit-serially.

When the data length is seven bits, bits 0 to 6 of the data in TXS20 will be transmission data. Writing data to TXS20 triggers transmission.

TXS20 can be write-accessed, using an 8-bit memory manipulation instruction, but cannot be read-accessed.

A $\overline{\text{RESET}}$ input loads FFH into TXS20.

**Caution Do not write to TXS20 during transmission.**
**TXS20 and the reception buffer register 20 (RXB20) are mapped at the same address, such that any attempt to read from TXS20 results in a value being read from the RXB.**

**(b) Reception shift register 20 (RXS20)**

RXS20 is a register in which serial data, received at the RxD20 pin, is converted to parallel data. Once one entire byte has been received, RXS20 feeds the reception data to the reception buffer register 20 (RXB20).

RXS20 cannot be manipulated directly by a program.

**(c) Reception buffer register 20 (RXB20)**

RXB20 is used to hold reception data. Once RXS20 has received one entire byte of data, it feeds that data into RXB20.

When the data length is seven bits, the reception data is sent to bits 0 to 6 of RXB20, in which the MSB is fixed to 0.

RXB20 can be read-accessed, using an 8-bit memory manipulation instruction, but cannot be write-accessed.

A $\overline{\text{RESET}}$ input makes RXB20 undefined.

**Caution RXB20 and the transmission shift register 20 (TXS20) are mapped at the same address, such that any attempt to write to RXB20 results in a value being written to TXS20.**

**(d) Transmission control circuit**

The transmission control circuit controls transmission. For example, it adds start, parity, and stop bits to the data in the transmission shift register 20 (TXS20), according to the setting of the asynchronous serial interface mode register 20 (ASIM20).

**(e) Reception control circuit**

The reception control circuit controls reception according to the setting of the asynchronous serial interface mode register 20 (ASIM20). It also checks for errors, such as parity errors, during reception. If an error is detected, the asynchronous serial interface status register 20 (ASIS20) is set according to the status of the error.

### (3) Serial interface 20 control registers

The following four types of registers are used to control serial interface 20.

- Serial operation mode register 20 (CSIM20)
- Asynchronous serial interface mode register 20 (ASIM20)
- Asynchronous serial interface status register 20 (ASIS20)
- Baud rate generator control register 20 (BRGC20)

#### (a) Serial operation mode register 20 (CSIM20)

CSIM20 is used to make the settings related to three-wire serial I/O mode.

CSIM20 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears CSIM20 to 00H.

**Figure 5-33. Format of Serial Operation Mode Register 20**

| Symbol | ⑦ | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| CSIM20 | CSIE20 | SSE20 | 0 | 0 | DAP20 | DIR20 | CSCK20 | CKP20 | FF72H | 00H | R/W |

| CSIE20 | Three-wire serial I/O mode operation control |
|---|---|
| 0 | Operation disabled |
| 1 | Operation enabled |

| SSE20 | $\overline{\text{SS20}}$-pin selection | Functions of the $\overline{\text{SS20}}$/P26 pin | Communication status |
|---|---|---|---|
| 0 | Not used | Port function | Communication enabled |
| 1 | Used | 0 | Communication enabled |
| | | 1 | Communication disabled |

| DAP20 | Three-wire serial I/O mode data phase selection |
|---|---|
| 0 | Outputs at the falling edge of $\overline{\text{SCK20}}$. |
| 1 | Outputs at the rising edge of $\overline{\text{SCK20}}$. |

| DIR20 | First-bit specification |
|---|---|
| 0 | MSB |
| 1 | LSB |

| CSCK20 | Three-wire serial I/O mode clock selection |
|---|---|
| 0 | External clock pulse input to the $\overline{\text{SCK20}}$ pin |
| 1 | Output of the dedicated baud rate generator |

| CKP20 | Three-wire serial I/O mode clock phase selection |
|---|---|
| 0 | Clock is low active; $\overline{\text{SCK20}}$ is high in the idle state |
| 1 | Clock is high active; $\overline{\text{SCK20}}$ is low in the idle state |

**Cautions 1. Bits 4 and 5 must be fixed to 0.**

**2. CSIM20 must be cleared to 00H, if UART mode is selected.**

**(b) Asynchronous serial interface mode register 20 (ASIM20)**

ASIM20 is used to make the settings related to serial interface 20 used in asynchronous serial interface mode.

ASIM20 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears ASIM20 to 00H.

**Figure 5-34. Format of Asynchronous Serial Interface Mode Register 20**

| Symbol | ⑦ | ⑥ | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ASIM20 | TXE20 | RXE20 | PS201 | PS200 | CL20 | SL20 | 0 | 0 | FF70H | 00H | R/W |

| TXE20 | Transmission control |
|---|---|
| 0 | Transmission disabled |
| 1 | Transmission enabled |

| RXE20 | Reception control |
|---|---|
| 0 | Reception disabled |
| 1 | Reception enabled |

| PS201 | PS200 | Parity bit specification |
|---|---|---|
| 0 | 0 | No parity |
| 0 | 1 | At transmission, the parity bit is fixed to 0.<br>At reception, a parity check is not made; no parity error is reported. |
| 1 | 0 | Odd parity |
| 1 | 1 | Even parity |

| CL20 | Transmission data character length specification |
|---|---|
| 0 | 7 bits |
| 1 | 8 bits |

| SL20 | Transmission data stop bit length |
|---|---|
| 0 | 1 bit |
| 1 | 2 bits |

**Cautions 1. Bits 0 and 1 must be fixed to 0.**

**2. If three-wire serial I/O mode is selected, ASIM20 must be cleared to 00H.**

**3. Switch operation mode from one mode to another after stopping both serial transmission and reception.**

**Table 5-13. Serial Interface 20 Operation Mode Settings**

**(i) Operation stopped mode**

| ASIM20 | | CSIM20 | | | PM22 | P22 | PM21 | P21 | PM20 | P20 | First bit | Shift clock | P22/SI20/ RxD20 pin function | P21/SO20/ TxD20 pin function | P20/$\overline{SCK20}$/ ASCK20 pin function |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TXE20 | RXE20 | CSIE20 | DIR20 | CSCK20 | | | | | | | | | | | |
| 0 | 0 | 0 | × | × | × Note 1 | × Note 1 | × Note 1 | × Note 1 | × Note 1 | × Note 1 | - | - | P22 | P21 | P20 |
| Other settings | | | | | | | | | | | Not to be set | | | | |

**(ii) Three-wire serial I/O mode**

| ASIM20 | | CSIM20 | | | PM22 | P22 | PM21 | P21 | PM20 | P20 | First bit | Shift clock | P22/SI20/ RxD20 pin function | P21/SO20/ TxD20 pin function | P20/$\overline{SCK20}$/ ASCK20 pin function |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TXE20 | RXE20 | CSIE20 | DIR20 | CSCK20 | | | | | | | | | | | |
| 0 | 0 | 1 | 0 | 0 | 1 Note 2 | × Note 2 | 0 | 1 | 1 | × | MSB | External clock | SI20 Note2 | SO20 (CMOS output) | $\overline{SCK20}$ input |
| | | | | 1 | | | | | 0 | 1 | | Internal clock | | | $\overline{SCK20}$ output |
| | | 1 | 1 | 0 | | | | | 1 | × | LSB | External clock | | | $\overline{SCK20}$ input |
| | | | | 1 | | | | | 0 | 1 | | Internal clock | | | $\overline{SCK20}$ output |
| Other settings | | | | | | | | | | | Not to be set | | | | |

**(iii) Asynchronous serial interface mode**

| ASIM20 | | CSIM20 | | | PM22 | P22 | PM21 | P21 | PM20 | P20 | First bit | Shift clock | P22/SI20/ RxD20 pin function | P21/SO20/ TxD20 pin function | P20/$\overline{SCK20}$/ ASCK20 pin function |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| TXE20 | RXE20 | CSIE20 | DIR20 | CSCK20 | | | | | | | | | | | |
| 1 | 0 | 0 | 0 | 0 | × Note 1 | × Note 1 | 0 | 1 | 1 | × | LSB | External clock | P22 | TxD20 (CMOS output) | ASCK20 input |
| | | | | | | | | | × Note 1 | × Note 1 | | Internal clock | | | P20 |
| 0 | 1 | 0 | 0 | 0 | 1 | × | × Note 1 | × Note 1 | 1 | × | | External clock | RxD20 | P21 | ASCK20 input |
| | | | | | | | | | × Note 1 | × Note 1 | | Internal clock | | | P20 |
| 1 | 1 | 0 | 0 | 0 | 1 | × | 0 | 1 | 1 | × | | External clock | | TxD20 (CMOS output) | ASCK20 input |
| | | | | | | | | | × Note 1 | × Note 1 | | Internal clock | | | P20 |
| Other settings | | | | | | | | | | | Not to be set | | | | |

**Notes 1.** These pins can be used for port functions.

**2.** When only transmission is used, these pins can be used as P22 (CMOS input/output).

**Remark** ×: Don't care.

**(c) Asynchronous serial interface status register 20 (ASIS20)**

ASIS20 is used to display the type of a reception error, if it occurs while asynchronous serial interface mode is set.

ASIS20 is manipulated using an 8-bit memory manipulation instruction.

The contents of ASIS20 are undefined in three-wire serial I/O mode.

A $\overline{\text{RESET}}$ input clears ASIS20 to 00H.

**Figure 5-35. Format of Asynchronous Serial Interface Status Register 20**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| ASIS20 | 0 | 0 | 0 | 0 | 0 | PE20 | FE20 | OVE20 | FF71H | 00H | R |

| PE20 | Parity error flag |
|---|---|
| 0 | No parity error has occurred. |
| 1 | A parity error has occurred (parity mismatch in transmission data). |

| FE20 | Framing error flag |
|---|---|
| 0 | No framing error has occurred. |
| 1 | A framing error has occurred (no stop bit detected).**Note 1** |

| OVE20 | Overrun error flag |
|---|---|
| 0 | No overrun error has occurred. |
| 1 | An overrun error has occurred**Note 2**<br>(Before data was read from the reception buffer register, the subsequent recepiton sequence was completed.) |

**Notes 1.** Even if 2 is specified for the number of stop bits (using bit 2 (SL20) of ASIM20), only one stop bit is detected at reception.

**2.** After an overrun occurs, read-access the reception buffer register 20 (RXB20). Otherwise, the overrun error will recur each time data is received.

**(d) Baud rate generator control register 20 (BRGC20)**

BRGC20 is used to specify the serial clock for serial interface 20.

BRGC20 is manipulated using an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears BRGC20 to 00H.

**Figure 5-36. Format of Baud Rate Generator Control Register 20**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| BRGC20 | TPS203 | TPS202 | TPS201 | TPS200 | 0 | 0 | 0 | 0 | FF73H | 00H | R/W |

| TPS203 | TPS202 | TPS201 | TPS200 | 3-bit counter source clock selection | n |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | $f_X/2$ (2.5 MHz) | 1 |
| 0 | 0 | 0 | 1 | $f_X/2^2$ (1.25 MHz) | 2 |
| 0 | 0 | 1 | 0 | $f_X/2^3$ (625 kHz) | 3 |
| 0 | 0 | 1 | 1 | $f_X/2^4$ (313 kHz) | 4 |
| 0 | 1 | 0 | 0 | $f_X/2^5$ (156 kHz) | 5 |
| 0 | 1 | 0 | 1 | $f_X/2^6$ (78.1 kHz) | 6 |
| 0 | 1 | 1 | 0 | $f_X/2^7$ (39.1 kHz) | 7 |
| 0 | 1 | 1 | 1 | $f_X/2^8$ (19.5 kHz) | 8 |
| 1 | 0 | 0 | 0 | External clock pulse input at the ASCK20 pin**Note** | - |
| Other settings | | | | Not to be set | |

**Note** An external clock can be used only in UART mode.

**Cautions 1. Any attempt to write to BRGC20 during communication adversely affects the output of the baud rate generator, thus hampering normal operation. Therefore, do not write to BRGC20 during communication.**

**2. Do not select n = 1 during operation at $f_X$ = 5.0 MHz, as n = 1 causes the rated baud rate to be exceeded.**

**3. When the external input clock is selected, set port mode register 2 (PM2) in input mode.**

**Remarks 1.** $f_X$ : Main system clock oscillation frequency

**2.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz.

The transmission and reception clock pulses used to generate the baud rate are obtained by dividing the frequency of the main system clock pulse or a signal input to the ASCK20 pin.

**(i) Generating transmission and reception clock pulses for baud rates based on the main system clock**

The frequency of the main system clock is divided to generate the transmission and reception clock pulses. The baud rate generated based on the main system clock is determined using the following expression.

$$[\text{Baud rate}] = \frac{f_X}{2^{n+1} \times 8} \text{ [Hz]}$$

$f_X$ : Main system clock oscillation frequency

**Table 5-14. Relationships between Main System Clock Frequencies and Baud Rates (Example)**

| Baud rate (bps) | n | BRGC20 setting | Error (%) | |
|---|---|---|---|---|
| | | | $f_X$ = 5.0 MHz | $f_X$ = 4.9152 MHz |
| 1,200 | 8 | 70H | 1.73 | 0 |
| 2,400 | 7 | 60H | | |
| 4,800 | 6 | 50H | | |
| 9,600 | 5 | 40H | | |
| 19,200 | 4 | 30H | | |
| 38,400 | 3 | 20H | | |
| 76,800 | 2 | 10H | | |

**Caution Do not select n = 1 during operation at $f_X$ = 5.0 MHz, as n = 1 causes the rated baud rate to be exceeded.**

**(ii) Generating transmission and reception clock pulses for baud rates based on an external clock pulse received at the ASCK20 pin**

The frequency of an external clock pulse received at the ASCK20 pin is used to generate the transmission and reception clock pulses. The baud rate generated based on the external clock pulse received at the ASCK20 pin is determined using the following expression.

$$[\text{Baud rate}] = \frac{f_{ASCK}}{16} \text{ [Hz]}$$

$f_{ASCK}$: Frequency of clock pulse received at the ASCK20 pin

**Table 5-15. Relationships between ASCK20 Pin Input Frequencies and Baud Rates (When BRGC20 = 80H)**

| Baud rate (bps) | ASCK20 pin input frequency (kHz) |
|---|---|
| 75 | 1.2 |
| 150 | 2.4 |
| 300 | 4.8 |
| 600 | 9.6 |
| 1,200 | 19.2 |
| 2,400 | 38.4 |
| 4,800 | 76.8 |
| 9,600 | 153.6 |
| 19,200 | 307.2 |
| 31,250 | 500.0 |
| 38,400 | 614.4 |

### 5.8.2 SMB0 (system management bus)

**(1) SMB0 functions**

SMB0 has the following two types of modes.

- Operation stopped mode
- SMB mode (supporting multiple masters)

**(a) Operation stopped mode**

This mode is used when serial transfer is not performed. Power consumption is minimized in this mode.

**(b) SMB mode (supporting multiple masters)**

This mode is used for performing 8-bit data transmission to several devices, using a serial clock (SCL0) line and a serial data bus (SDA0) line.

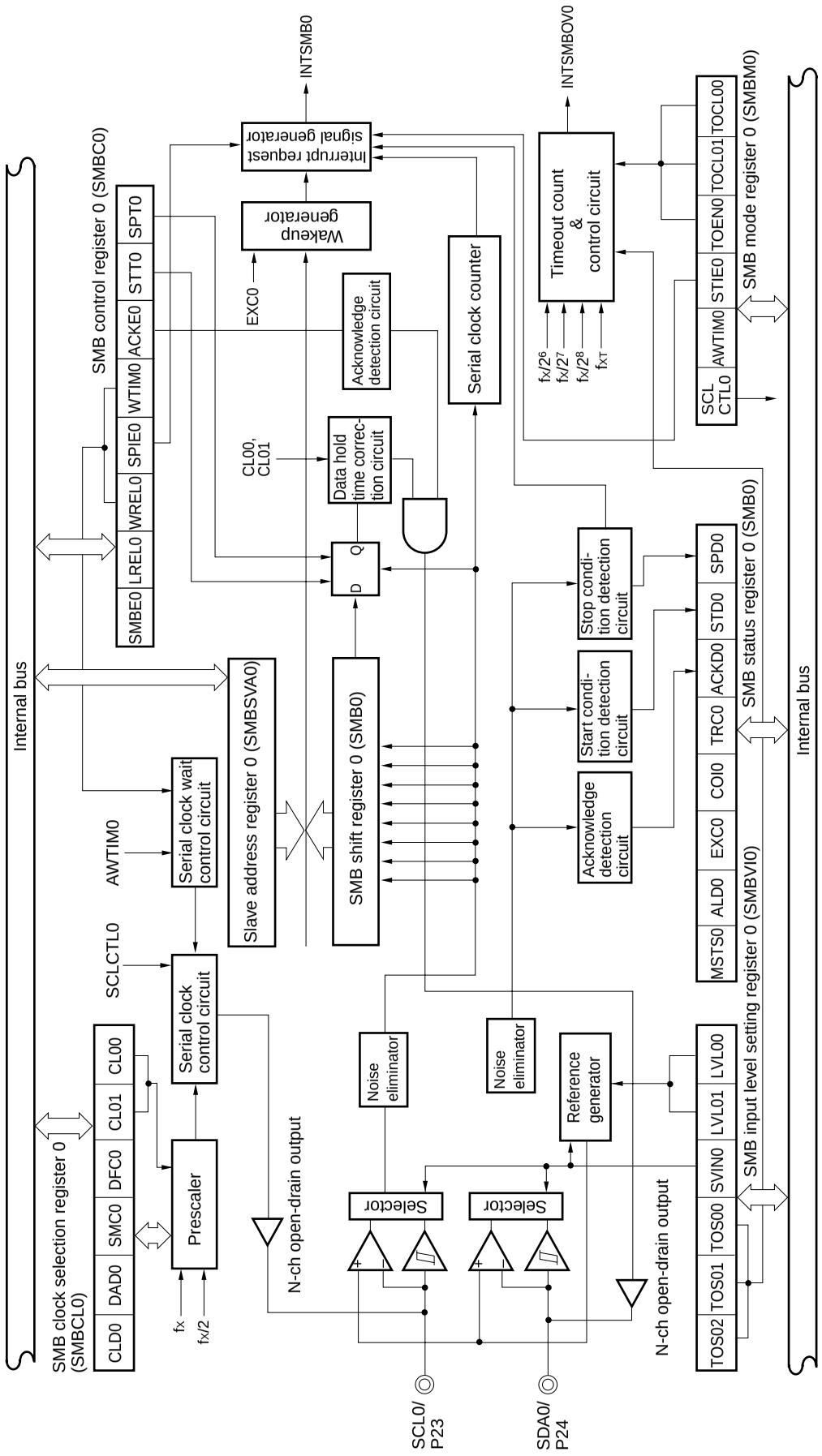
In this mode, which conforms to the SMB format, start conditions, data, stop conditions can be output on the serial data bus during transmission. Moreover, these data can be automatically detected by hardware during reception.

In SMB0, SCL0 and SDA0 are open-drain outputs, and therefore a pull-up resistor is required for the serial clock line and serial data bus line.

$I^2C$ (Inter IC) bus standard mode or high-speed mode can be specified by software in SMB mode.

Figure 5-37 shows the block diagram of SMB0.

**Figure 5-37. Block Diagram of SMB0**



**(2) SMB0 configuration**

SMB0 consists of the following hardware.

**Table 5-16. SMB0 Configuration**

| Item | Configuration |
|---|---|
| Register | SMB shift register 0 (SMB0)<br>Slave address register 0 (SMBSVA0) |
| Control register | SMB control register 0 (SMBC0)<br>SMB status register 0 (SMBS0)<br>SMB clock selection register 0 (SMBCL0)<br>SMB mode register 0 (SMBM0)<br>SMB input level setting register 0 (SMBVI0) |

**(a) SMB shift register 0 (SMB0)**

SMB0 is a register that converts 8-bit serial data to 8-bit parallel data, and vice-versa. SMB0 is used both for sending and receiving data.

Write and read operations for SMB0 control actual send and receive operations.

SMB0 is manipulated with an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears SMB0 to 00H.

**(b) Slave address register 0 (SMBSVA0)**

This register is used to set a local address when used as a slave.

SMBSVA0 is manipulated with an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears SMBSVA0 to 00H.

**(c) SO latch**

The SO latch is a latch that holds the SDA0 pin output level.

**(d) Wakeup control circuit**

This circuit generates an interrupt request when the address value set in slave address register 0 (SMBSVA0) and the received address match, or when an extension code is received.

**(e) Clock selector**

Selects the sampling clock to be used.

**(f) Serial clock counter**

Counts the serial clock output/input during send/receive operations, to check if 8-bit data has been sent or received.

**(g) Interrupt request signal generation circuit**

Controls the generation of interrupt request signals.

SMB interrupts are generated with the following two triggers.

- 8th clock or 9th clock of serial clock (set with WTIM0 bit[Note])
- Generation of interrupt request at detection of stop condition (set with bit SPIE0[Note])

**Note** WTIM0 bit: SMB control register 0 (SMBC0) bit 3
SPIE0 bit: SMB control register 0 (SMBC0) bit 4

**(h) Serial clock control circuit**

In master mode, generates the clock to be output to the SCL0 pin from the sampling clock.

**(i) Serial clock wait control circuit**

Controls the wait timing.

**(j) Acknowledge output circuit, stop condition detection circuit, start condition detection circuit, acknowledge detection circuit**

Perform output and detection of control signals.

**(k) Data hold time correction circuit**

Generates the data hold time from the falling edge of the serial clock.

**(3) SMB0 control registers**

The following five types of registers are used to control SMB0.

- SMB control register 0(SMBC0)
- SMB status register 0 (SMBS0)
- SMB clock selection register 0 (SMBCL0)
- SMB mode register 0 (SMBM0)
- SMB input level setting register 0 (SMBVI0)

The following additional registers are also used.

- SMB shift register 0 (SMB0)
- Slave address register 0 (SMBSVA0)

**(a) SMB control register 0 (SMBC0)**

This register sets SMB operation enable/disable, the wait timing, and other SMB operations.

SMBC0 is manipulated with a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears SMBC0 to 00H.

**Caution In SMB mode, set port mode registers 2 (PM2×) to achieve the following statuses. Also, set each output latch to 0.**

- **P23 (SCL0) is set to output mode (PM23 = 0).**
- **P24 (SDA0) is set to output mode (PM24 = 0).**

**Figure 5-38. Format of SMB Control Register 0 (1/4)**

| Symbol | ⑦ | ⑥ | ⑤ | ④ | ③ | ② | ① | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SMBC0 | SMBE0 | LREL0 | WREL0 | SPIE0 | WTIM0 | ACKE0 | STT0 | SPT0 | FF78H | 00H | R/W |

| SMBE0 | SMB operation |
|---|---|
| 0 | Operation disabled. Presets extension register (SMBS0). Internal operation also disabled. |
| 1 | Operation enabled |

| Clear conditions (SMBE0 = 0) | Set conditions (SMBE0 = 1) |
|---|---|
| • Cleared with instruction<br>• Cleared by $\overline{\text{RESET}}$ input | • Set with instruction |

| LREL0 | Escape from transmission |
|---|---|
| 0 | Normal operation |
| 1 | Escapes from the current transmission and enters the standby status. Automatically cleared after execution.<br>This bit is used when extension codes not relevant to the local station are received.<br>The SCL0 and SDA0 lines enter the high impedance status.<br>The following flags are cleared.<br>• STD0 • STT0 • SPT0 • ACKD0 • TRC0 • COI0 • EXC0 • MSTS0 |

The standby status continues until the following communication participation conditions are met.
- Startup as master after detection of stop condition
- Matching addresses or extension code reception after start condition

| Clear conditions (LREL0 = 0)[Note] | Set conditions (LREL0 = 1) |
|---|---|
| • Automatically cleared after execution<br>• Cleared by $\overline{\text{RESET}}$ input | • Set with instruction |

| WREL0 | Wait cancel |
|---|---|
| 0 | Does not cancel wait. |
| 1 | Cancels wait. Automatically cleared after wait cancellation. |

| Clear conditions (WREL0 = 0)[Note] | Set conditions (WREL0 = 1) |
|---|---|
| • Automatically cleared after execution<br>• Cleared by $\overline{\text{RESET}}$ input | • Set with instruction |

| SPIE0 | Interrupt request generation at stop condition detection |
|---|---|
| 0 | Disabled |
| 1 | Enabled |

| Clear conditions (SPIE0 = 0)[Note] | Set conditions (SPIE0 = 1) |
|---|---|
| • Cleared with instruction<br>• Cleared by $\overline{\text{RESET}}$ input | • Set with instruction |

**Note** This flag's signals are made invalid by setting SMBE0 = 0.

**Figure 5-38. Format of SMB Control Register 0 (2/4)**

| WTIM0 | Wait and interrupt request generation control |
|---|---|
| 0 | Generates interrupt request at falling edge of 8th clock.<br>In case of master: Waits with clock output at low level after 8 clocks have been output.<br>In case of slave: Waits master with clock set to low level after 8 clocks have been input. |
| 1 | Generates interrupt request at falling edge of 9th clock.<br>In case of master: Waits with clock at low level after 9 clocks have been output.<br>In case of slave: Waits master with clock set at low level after 9 clocks have been input. |

The setting of this bit becomes invalid during address transmission, and becomes effective at the end of transmission. During operation as master, a wait is inserted at the falling edge of the 9th clock during address transmission. A slave that receives a local address enters the wait status at the falling edge of the 9th clock after generation of an acknowledge. A slave that receives an extension code enters the wait status at the falling edge of the 8th clock.

| Clear conditions (WTIM0 = 0)[Note] | Set conditions (WTIM0 = 1) |
|---|---|
| • Cleared with instruction<br>• Cleared by $\overline{\text{RESET}}$ input | • Set with instruction |

| ACKE0 | Acknowledge control |
|---|---|
| 0 | Acknowledge disabled |
| 1 | Acknowledge enabled. SDA0 line set to low level during 9 clocks. However, invalid during address transmission, and valid when EXC0 = 1. |

| Clear conditions (ACKE0 = 0)[Note] | Set conditions (ACKE0 = 1) |
|---|---|
| • Cleared with instruction<br>• Cleared by $\overline{\text{RESET}}$ input | • Set with instruction |

**Note** This flag's signals are made invalid by setting SMBE0 = 0.

**Figure 5-38. Format of SMB Control Register 0 (3/4)**

| STT0 | Start condition trigger |
|---|---|
| 0 | Doesn't generate start condition. |
| 1 | When bus is released (stop status):<br>Generates start conditions (activation as master). Changes SDA0 line from high level to low level and generates start condition. Then secures rated time and sets SCL0 to low level.<br>When not participating on bus:<br>Functions as start condition reservation flag. When set, automatically generates start condition after bus is released. |

Cautions regarding set timing
- Master receive operation: Setting during transmission is prohibited.
  Set ACKE0 = 0; Can be set only after end of receive operation has been notified to slave.
- Master send operation: Note that start condition may not be generated normally during $\overline{\text{ACK}}$ period.
- Setting at the same time as SPT0 is prohibited.

| Clear conditions (STT0 = 0)[Note] | Set conditions (STT0 = 1) |
|---|---|
| • Cleared with instruction<br>• Cleared upon defeat in arbitration<br>• Cleared after generation of start condition by master<br>• Cleared by $\overline{\text{RESET}}$ input | • Set with instruction |

**Note** This flag's signals are made invalid by setting SMBE0 = 0.

**Figure 5-38. Format of SMB Control Register 0 (4/4)**

| SPT0 | Stop condition trigger |
|---|---|
| 0 | Does not generate stop condition. |
| 1 | Generates stop condition (end transmission as master).<br>After setting SDA0 line to low level, sets SCL0 line to high level, or maintains SCL0 line at high level.<br>Then, secures rated time, changes SDA0 line from low level to high level, and generates stop condition. |

Cautions regarding set timing

- Master receive operation: Setting during transmission is prohibited.
  Set ACKE0 = 0; Can be set only after end of receive operation has been notified to slave.
- Master send operation: Note that stop condition may not be generated normally during $\overline{\text{ACK}}$ period.
- Setting at the same time as STT0 is prohibited.
- Set SPT0 only during operation as master.[Note 1]
- Note that when WTIM0 = 0, if SPT0 is set during the wait period after 8-clock output, a stop condition is generated during the high-level period of the 9th clock following wait release.
  If it is necessary to output a 9th clock, change the setting of WTIM0 from 0 to 1 during the wait period following 8-clock output, and set SPT0 during the wait period following the 9th clock output.

| Clear conditions (SPT0 = 0)[Note 2] | Set conditions (SPT0 = 1) |
|---|---|
| • Cleared with instruction<br>• Cleared upon defeat in arbitration<br>• Cleared automatically after detection of stop condition<br>• Cleared by $\overline{\text{RESET}}$ input | • Set with instruction |

**Notes 1.** Set STP0 only during operation as master. However, for master operation by the time a stop condition is detected for the first time following operation enable, SPT0 must be set once to generate a stop condition.

**2.** This flag's signals are made invalid by setting SMBE0 = 0.

**Caution While SMB status register 0 (SMBS0) bit 3 (TRC0) = 1, when WREL0 is set at the 9th clock and wait is released, TRC0 is cleared and the SDA0 line is placed in high impedance.**

**Remark** STD0 : SMB status register 0 (SMBS0) bit 1
ACKD0 : SMB status register 0 (SMBS0) bit 2
TRC0 : SMB status register 0 (SMBS0) bit 3
COI0 : SMB status register 0 (SMBS0) bit 4
EXC0 : SMB status register 0 (SMBS0) bit 5
MSTS0 : SMB status register 0 (SMBS0) bit 7

**(b) SMB status register 0 (SMBS0)**

This register indicates the SMB status.

SMBS0 is manipulated with a 1-bit or 8-bit memory manipulation instruction. SMBS0 is a read-only register.

A $\overline{\text{RESET}}$ input clears SMBS0 to 00H.

**Figure 5-39. Format of SMB Status Register 0 (1/3)**

| Symbol | ⑦ | ⑥ | ⑤ | ④ | ③ | ② | ① | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SMBS0 | MSTS0 | ALD0 | EXC0 | COI0 | TRC0 | ACKD0 | STD0 | SPD0 | FF79H | 00H | R |

| MSTS0 | Master status |
|---|---|
| 0 | Slave status or communication wait status |
| 1 | Master transmission status |

| Clear conditions (MSTS0 = 0) | Set conditions (MSTS0 = 1) |
|---|---|
| • Cleared upon detection of stop condition<br>• Cleared when ALD0 = 1<br>• Cleared when LREL0 = 1<br>• Cleared when SMBE0 changes from 1 to 0<br>• Cleared by $\overline{\text{RESET}}$ input | • Set during generation of start condition |

| ALD0 | Arbitration defeat detection |
|---|---|
| 0 | No arbitration, or won in arbitration. |
| 1 | Defeated in arbitration. MSTS0 cleared. |

| Clear conditions (ALD0 = 0) | Set conditions (ALD0 = 1) |
|---|---|
| • Automatically cleared after reading SMBS0[Note]<br>• Cleared when SMBE0 changes from 1 to 0<br>• Cleared by $\overline{\text{RESET}}$ input | • Set upon defeat in arbitration |

| EXC0 | Extension code receive detection |
|---|---|
| 0 | Does not receive extension code. |
| 1 | Receives extension code. |

| Clear conditions (EXC0 = 0) | Set conditions (EXC0 = 1) |
|---|---|
| • Cleared upon detection of start condition<br>• Cleared upon detection of stop condition<br>• Cleared when LREL0 = 1<br>• Cleared when SMBE0 changes from 1 to 0<br>• Cleared by $\overline{\text{RESET}}$ input | • Set when high-order 4 bits of received address are 0000 or 1111 (set at rising edge of 8th clock) |

**Note** The bit is also cleared when a bit manipulation instruction is executed for any of other bit of SMBS0.

**Figure 5-39. Format of SMB Status Register 0 (2/3)**

| COI0 | Matching address detection |
|---|---|
| 0 | Address does not match. |
| 1 | Address matches. |

| Clear conditions (COI0 = 0) | Set conditions (COI0 = 1) |
|---|---|
| • Cleared upon detection of start condition<br>• Cleared upon detection of stop condition<br>• Cleared when LREL0 = 1<br>• Cleared when SMBE0 changes from 1 to 0<br>• Cleared by $\overline{\text{RESET}}$ input | • Set when received address matches local address (SVA0) (set at rising edge of 8th clock) |

| TRC0 | Receive/send status detection |
|---|---|
| 0 | Receive status (when not in send status). Sets SDA0 line to high impedance. |
| 1 | Send status. Sets so that SO latch value can be output to SDA0 line (valid from falling edge of 9th clock of 1st byte). |

| Clear conditions (TRC0 = 0) | Set conditions (TRC0 = 1) |
|---|---|
| • Cleared upon detection of stop condition<br>• Cleared when LREL0 = 1<br>• Cleared SMBE0 changes from 1 to 0<br>• Cleared when WREL0 = 1<br>• Cleared when ALD0 changes from 0 to 1<br>• Cleared by $\overline{\text{RESET}}$ input<br>In case of master:<br>• When "1" is output to 1st byte LSB (transmission direction specification bit)<br>In case of slave:<br>• Upon detection of start condition<br>In case of non-participation in communication | In case of master:<br>• Upon generation of start condition<br>In case of slave:<br>• When "1" is input to 1st byte LSB (transmission direction specification bit) |

| ACKD0 | Acknowledge output |
|---|---|
| 0 | Does not detect acknowledge. |
| 1 | Detects acknowledge. |

| Clear conditions (ACKD0 = 0) | Set conditions (ACKD0 = 1) |
|---|---|
| • Cleared upon detection of stop condition<br>• Cleared at rising edge of 1st clock of following byte<br>• Cleared when LREL0 = 1<br>• Cleared when SMBE0 changes from 1 to 0<br>• Cleared by $\overline{\text{RESET}}$ input | • Set when SDA0 line is low level at rising edge of 9th clock of SCL0 |

**Figure 5-39. Format of SMB Status Register 0 (3/3)**

| STD0 | Start condition detection |
|---|---|
| 0 | Does not detect start condition. |
| 1 | Detects start condition. Indicates that address transmission is in progress. |

| Clear conditions (STD0 = 0) | Set conditions (STD0 = 1) |
|---|---|
| • Cleared upon detection of stop condition<br>• Cleared at rising edge of 1st clock of byte following address transmission<br>• Cleared when LREL0 = 1<br>• Cleared when SMBE0 changes from 1 to 0<br>• Cleared by $\overline{\text{RESET}}$ input | • Set upon detection of start condition |

| SPD0 | Stop condition detection |
|---|---|
| 0 | Does not detect stop condition. |
| 1 | Detects stop condition. Transmission by master is completed and bus is released. |

| Clear conditions (SPD0 = 0) | Set conditions (SPD0 = 1) |
|---|---|
| • Cleared at rising edge of 1st clock of address transfer byte following detection of start condition after this bit has been set<br>• Cleared when SMBE0 changes from 1 to 0<br>• Cleared by $\overline{\text{RESET}}$ input | • Set upon detection of stop condition |

**Remark** LREL0 : SMB control register 0 (SMBC0) bit 6
SMBE0: SMB control register 0 (SMBC0) bit 7

**(c) SMB clock selection register 0 (SMBCL0)**

This register sets the SMB transmission clock.

SMBCL0 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears SMBCL0 to 00H.

**Figure 5-40. Format of SMB Clock Selection Register 0 (1/2)**

| Symbol | 7 | 6 | ⑤ | ④ | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SMBCL0 | 0 | 0 | CLD0 | DAD0 | SMC0 | DFC0 | CL01 | CL00 | FF7AH | 00H | R/W Note 1 |

| CLD0 | SCL0 line level detection (valid only when SMBE0 = 1) |
|---|---|
| 0 | Detects that SCL0 line is low level. |
| 1 | Detects that SCL0 line is high level. |

| Clear conditions (CLD0 = 0) | Set conditions (CLD0 = 1) |
|---|---|
| • Cleared when SCL0 line is low level<br>• Cleared when SMBE0 = 0<br>• Cleared by $\overline{\text{RESET}}$ input | • Set when SCL0 line is high level |

| DAD0 | SDA0 line level detection (valid only when SMBE0 = 1) |
|---|---|
| 0 | Detects that SDA0 line is low level. |
| 1 | Detects that SDA0 line is high level. |

| Clear conditions (DAD0 = 0) | Set conditions (DAD0 = 1) |
|---|---|
| • Cleared when SDA0 line is low level<br>• Cleared when SMBE0 = 0<br>• Cleared by $\overline{\text{RESET}}$ input | • Set when SDA0 line is high level |

| SMC0 | Operating mode switching |
|---|---|
| 0 | IIC standard mode or SMB mode operation |
| 1 | IIC high-speed mode |

| Clear conditions (SMC0 = 0) | Set conditions (SMC0 = 1) |
|---|---|
| • Cleared with instruction<br>• Cleared by $\overline{\text{RESET}}$ input | • Set with instruction |

| DFC0 | Digital filter operation control Note 2 |
|---|---|
| 0 | Digital filter OFF |
| 1 | Digital filter ON |

**Notes 1.** Bits 4 and 5 are read-only.

**2.** The digital filter can be used in the high-speed mode. When used in the high-speed mode, the digital filter provides a slower response.

**Figure 5-40. Format of SMB Clock Selection Register 0 (2/2)**

| Selection clock | | | |
|---|---|---|---|
| CL01 | CL00 | SMB/standard mode | High-speed mode |
| 0 | 0 | $f_X$ Note 1 | $f_X$ Note 2 |
| 0 | 1 | $f_X$ | |
| 1 | 0 | $f_X/2$ | $f_X/2$ Note 2 |
| 1 | 1 | Not to be set | |

**Notes 1.** Available range: 1.0 MHz ≤ $f_X$ ≤ 4.19 MHz

**2.** Available range: $f_X$ ≥ 2.0 MHz

**Caution Bits 6 and 7 must be fixed to 0.**

**Remark** $f_X$: Main system clock oscillation frequency

**(d) SMB mode register 0 (SMBM0)**

SMBM0 is used to specify SCL0 level control and interrupt control.

SMBM0 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input loads 20H into SMBM0.

**Figure 5-41. Format of SMB Mode Register 0 (1/2)**

| Symbol | 7 | 6 | ⑤ | ④ | ③ | ② | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SMBM0 | 0 | 0 | SCLCTL0 | AWTIM0 | STIE0 | TOEN0 | TOCL01 | TOCL00 | FF7CH | 20H | R/W |

| SCLCTL0 | SCL level control[Note 1] |
|---|---|
| 0 | SCL0 is held low.<br>When SCL0 is high, SCL0 is held low after waiting until SCL0 is made low. |
| 1 | Normal operation |

| STIE0 | Start condition interrupt enable |
|---|---|
| 0 | Start condition interrupt generation is disabled. |
| 1 | Normal operation |

| AWTIM0 | Wait and interrupt control when an address match is found[Notes 2, 3] |
|---|---|
| 0 | At the slave, an interrupt request is generated on the falling edge of the 9th clock period when an address match (COI0 = 1) is found during address data reception.<br>The clock is pulled low to cause the master to wait. |
| 1 | At the slave, an interrupt request is generated on the falling edge of the 8th clock period when an address match (COI0 = 1) is found during address data reception.<br>The clock is pulled low to cause the master to wait. |

**Notes 1.** If SCL0 is made low with SCLCTL0, wait state cannot be released with WREL0.

**2.** When an extension code is received (EXC0 = 1), wait state is forcibly set in the 8th clock period.

**3.** During address transfer, the master waits in the 9th clock period.

**Figure 5-41. Format of SMB Mode Register 0 (2/2)**

| TOEN0 | Time-out count enable bit[Note] |
|---|---|
| 0 | The time-out count is cleared to 0, then count operation is disabled. |
| 1 | Time-out count operation is enabled. |

| TOCL01 | TOCL00 | Time-out clock selection bits |
|---|---|---|
| 0 | 0 | $f_X/2^6$ (78.125 kHz) |
| 0 | 1 | $f_X/2^7$ (39.063 kHz) |
| 1 | 0 | $f_X/2^8$ (19.531 kHz) |
| 1 | 1 | $f_{XT}$ (32.768 kHz) |

**Note** An interrupt (INTSMBOV0) is generated when the time-out counter overflows. The hardware does not reset SMB operation. Ensure that SMB operation is reset by software after INTSMBOV0 generation.

**Caution Bits 6 and 7 must be fixed to 0.**

**Remarks 1.** $f_X$ : Main system clock oscillation frequency

**2.** $f_{XT}$: Subsystem clock oscillation frequency

**3.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz or $f_{XT}$ = 32.768 kHz.

**(e) SMB input level setting register 0 (SMBVI0)**

SMBVI0 is manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears SMBVI0 to 00H.

**Figure 5-42. Format of SMB Input Level Setting Register 0**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SMBVI0 | 0 | TOS02 | TOS01 | TOS00 | SVIN0 | 0 | LVL01 | LVL00 | FF7DH | 00H | R/W |

| TOS02 | TOS01 | TOS00 | Time-out time setting bits | | | |
|---|---|---|---|---|---|---|
| | | | $f_{TO} = f_X/2^6$ | $f_{TO} = f_X/2^7$ | $f_{TO} = f_X/2^8$ | $f_{TO} = f_{XT}$ |
| 0 | 0 | 0 | $1,024/f_{TO}$ (13.1 ms) | $1,024/f_{TO}$ (26.2 ms) | $1,024/f_{TO}$ (52.4 ms) | $1,024/f_{TO}$ (31.3 ms) |
| 0 | 0 | 1 | $896/f_{TO}$ (11.5 ms) | $896/f_{TO}$ (22.9 ms) | $896/f_{TO}$ (45.9 ms) | $896/f_{TO}$ (27.3 ms) |
| 0 | 1 | 0 | $768/f_{TO}$ (9.8 ms) | $768/f_{TO}$ (19.7 ms) | $768/f_{TO}$ (39.3 ms) | $768/f_{TO}$ (23.4 ms) |
| 0 | 1 | 1 | $640/f_{TO}$ (8.2 ms) | $640/f_{TO}$ (16.4 ms) | $640/f_{TO}$ (32.8 ms) | $640/f_{TO}$ (19.5 ms) |
| 1 | 0 | 0 | $512/f_{TO}$ (6.6 ms) | $512/f_{TO}$ (13.1 ms) | $512/f_{TO}$ (26.2 ms) | $512/f_{TO}$ (15.6 ms) |
| 1 | 0 | 1 | $384/f_{TO}$ (4.9 ms) | $384/f_{TO}$ (9.8 ms) | $384/f_{TO}$ (19.7 ms) | $384/f_{TO}$ (11.7 ms) |
| 1 | 1 | 0 | $256/f_{TO}$ (3.3 ms) | $256/f_{TO}$ (6.6 ms) | $256/f_{TO}$ (13.1 ms) | $256/f_{TO}$ (7.8 ms) |
| 1 | 1 | 1 | $128/f_{TO}$ (1.6 ms) | $128/f_{TO}$ (3.2 ms) | $128/f_{TO}$ (6.6 ms) | $128/f_{TO}$ (3.9 ms) |

| SVIN0 | Input level selection bit |
|---|---|
| 0 | Same input level as the ordinary hysteresis |
| 1 | The voltage set with LVL01 and LVL00 is used as the SCL0 and SDA0 input level threshold. |

| LVL01 | LVL00 | Input level selection bits[Note] |
|---|---|---|
| 0 | 0 | The input level is $0.1875 \times V_{DD}$. |
| 0 | 1 | The input level is $0.25 \times V_{DD}$. |
| 1 | 0 | The input level is $0.375 \times V_{DD}$. |
| 1 | 1 | The input level is $0.5 \times V_{DD}$. |

**Note** Set an input level from 0.75 to 1.25 V.

**Caution Bits 2 and 7 must be fixed to 0.**

**Remarks 1.** $f_X$ : Main system clock oscillation frequency

**2.** $f_{XT}$ : Subsystem clock oscillation frequency

**3.** $f_{TO}$ : Clock selected using bits 0 and 1 (TOCL00, TOCL01) of SMB mode register 0 (SMBM0)

**4.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz or $f_{XT}$ = 32.768 kHz.

**(f) SMB shift register 0 (SMB0)**

This register is used to perform serial send/receive (shift operation) in synchronization with the serial clock.

Read/write operations can be performed in 8-bit units, but do not write data to the SMB0 during transmission.

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SMB0 | | | | | | | | | FF1BH | 00H | R/W |

**(g) SMB slave address register 0 (SMBSVA0)**

This register stores the SMB slave address.

It can be read/written in 8-bit units, but bit 0 is fixed to 0.

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| SMBSVA0 | | | | | | | | 0 | FF7BH | 00H | R/W |

## 5.9 Multiplier

### 5.9.1 Multiplier function

The multiplier enables a calculation of 8 bits × 8 bits = 16 bits.

### 5.9.2 Multiplier configuration

**(1) Multiplication result storage register 0 (MUL0)**

This register stores 16-bit multiplication results.

This register holds the result of a multiplication after 16 CPU clock periods.

MUL0 is manipulated using a 16-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input makes MUL0 undefined.

**Caution MUL0 is designed to be manipulated using a 16-bit memory manipulation instruction. It can also be manipulated using 8-bit memory manipulation instructions, however. When an 8-bit memory manipulation instruction is used to set MUL0, it must be in a direct addressing access mode.**

**(2) Multiplication data registers A and B (MRA0, MRB0)**

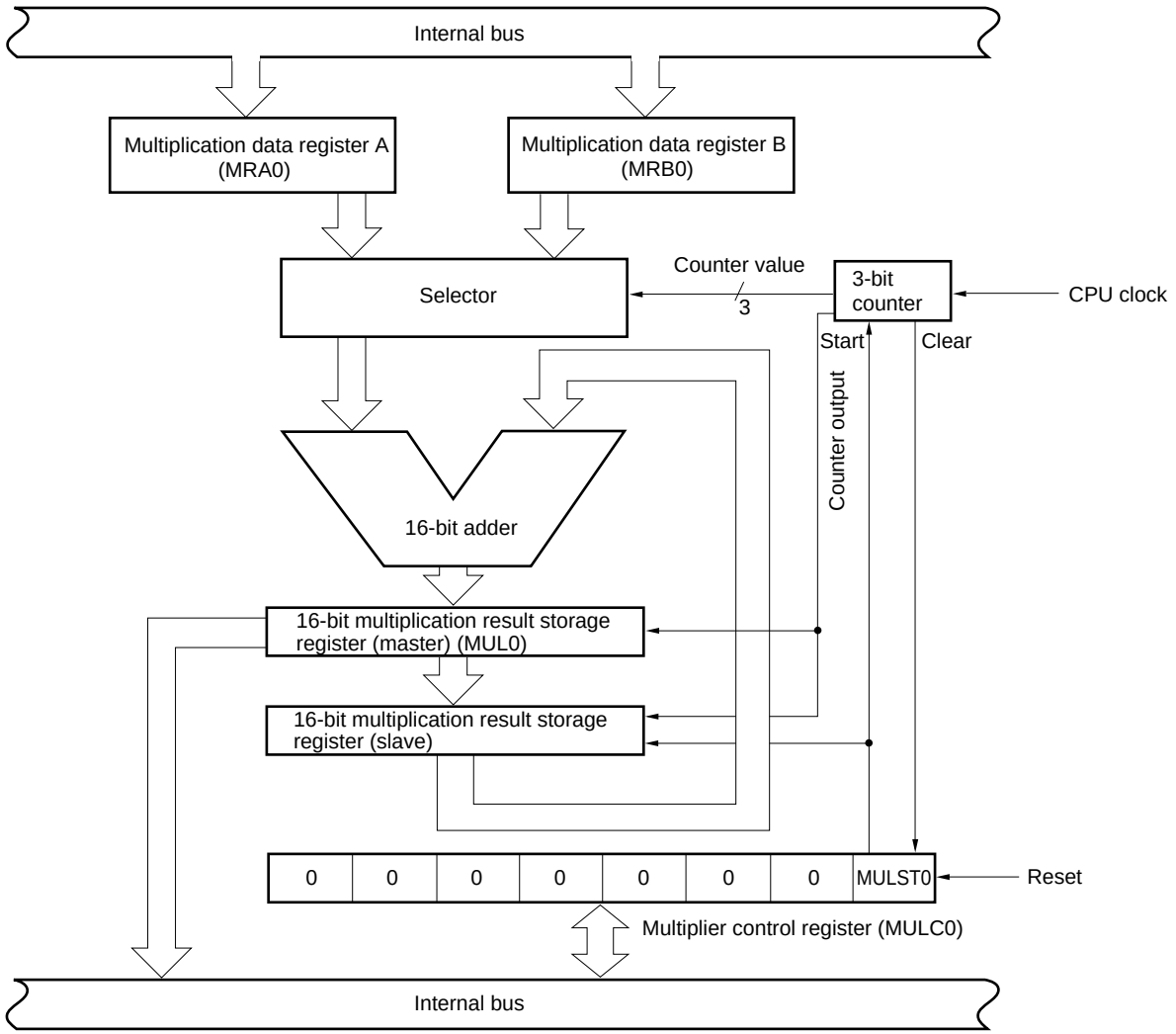
These registers store 8-bit multiplication data. The multiplier multiplies the value of MRA0 by the value of MRB0.

MRA0 and MRB0 are set using an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input makes these registers undefined.

Figure 5-43 shows the block diagram of the multiplier.

**Figure 5-43. Block Diagram of Multiplier**



### 5.9.3 Multiplier control register

The following register is used to control the multiplier:

- Multiplier control register 0 (MULC0)

MULC0 not only controls operations, but also indicates the operation status of the multiplier.
MULC0 is manipulated using a 1-bit or 8-bit memory manipulation instruction.
A $\overline{\text{RESET}}$ input clears MULC0 to 00H.

**Figure 5-44. Format of Multiplier Control Register 0 (MULC0)**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MULC0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | MULST0 | FFD2H | 00H | R/W |

| MULST0 | Multiplier operation start control bit | Multiplier operation status |
|---|---|---|
| 0 | Operation is stopped after the counter is cleared to 0. | Operation is stopped. |
| 1 | Operation is enabled. | Operation is being executed. |

**Caution Bits 1 to 7 must be fixed to 0.**

# 6. INTERRUPT FUNCTIONS

## 6.1 Interrupt Function Types

Two types of interrupt function are supported.

**(1) Nonmaskable interrupt**

A nonmaskable interrupt request is accepted unconditionally, that is, even when interrupts are disabled. A nonmaskable interrupt takes precedence over all other interrupts; it is not subjected to interrupt priority control.

A nonmaskable interrupt causes the standby release signal to be generated.

The μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y support one nonmaskable interrupt source namely, the watchdog timer interrupt.

**(2) Maskable interrupt**

Maskable interrupts are those which are subjected to mask control. If two or more maskable interrupts occur simultaneously, the default priority listed in Table 6-1 applies.

The maskable interrupts cause the standby release signal to be generated.

The maskable interrupts supported by the μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y include 4 external interrupt sources and 12 internal interrupt sources.

## 6.2 Interrupt Sources and Configuration

The μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y each support a total of 17 maskable and nonmaskable interrupt sources. (See **Table 6-1**.)

**Table 6-1. Interrupt Sources**

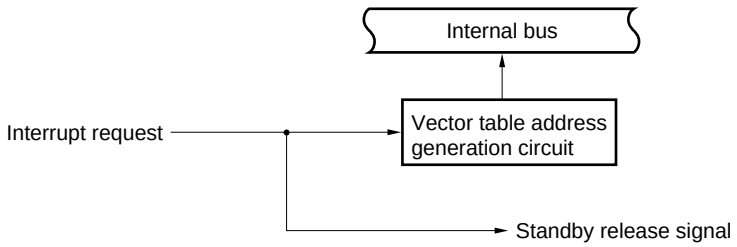
| Interrupt type | Priority[Note 1] | Interrupt source | | Internal/external | Vector table address | Basic configuration type[Note 2] |
|---|---|---|---|---|---|---|
| | | Name | Trigger | | | |
| Nonmaskable interrupt | – | INTWDT | Watchdog timer overflow (when watchdog timer mode 1 is selected) | Internal | 0004H | (A) |
| Maskable interrupt | 0 | INTWDT | Watchdog timer overflow (when the interval timer mode is selected) | | | (B) |
| | 1 | INTP0 | Pin input edge detection | External | 0006H | (C) |
| | 2 | INTP1 | | | 0008H | |
| | 3 | INTP2 | | | 000AH | |
| | 4 | INTP3 | | | 000CH | |
| | 5 | INTSR20 | End of UART reception on serial interface 20 | Internal | 000EH | (B) |
| | | INTCSI20 | End of three-wire SIO transfer reception on serial interface 20 | | | |
| | 6 | INTST20 | End of UART transmission on serial interface 20 | | 0010H | |
| | 7 | INTWT | Clock timer interrupt | | 0012H | |
| | 8 | INTWTI | Interval timer interrupt | | 0014H | |
| | 9 | INTTM80 | Generation of match signal for 8-bit timer/event counter 80 | | 0016H | |
| | 10 | INTTM81 | Generation of match signal for 8-bit timer/event counter 81 | | 0018H | |
| | 11 | INTTM82 | Generation of match signal for 8-bit timer counter 82 | | 001AH | |
| | 12 | INTTM90 | Generation of match signal for 16-bit timer counter 90 | | 001CH | |
| | 13 | INTSMB0 | System management bus interrupt | | 001EH | |
| | 14 | INTSMBOV0 | System management bus time-out interrupt | | 0020H | |
| | 15 | INTAD0 | A/D conversion completion signal | | 0022H | |

**Notes 1.** The priority regulates which maskable interrupt is higher, when two or more maskable interrupts are requested simultaneously. Zero signifies the highest priority, while 15 is the lowest.

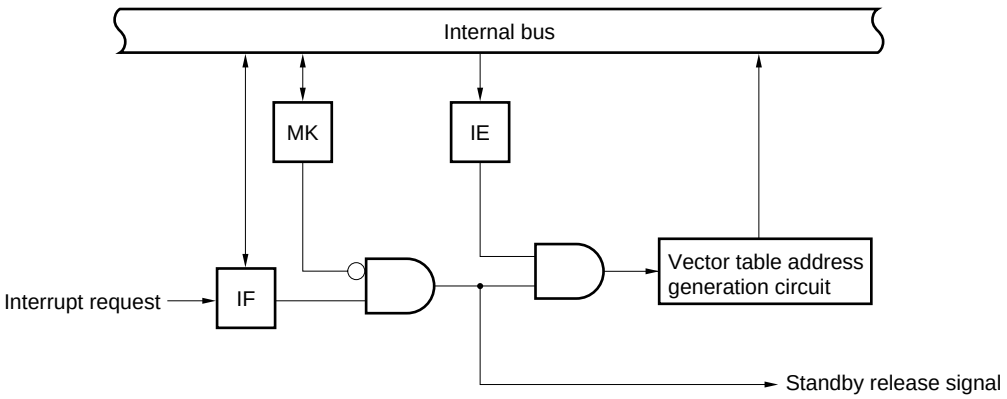
**2.** Basic configuration types (A), (B), and (C) correspond to (A), (B), and (C) in Figure 6-1, respectively.

**Figure 6-1. Basic Configuration of Interrupt Functions**

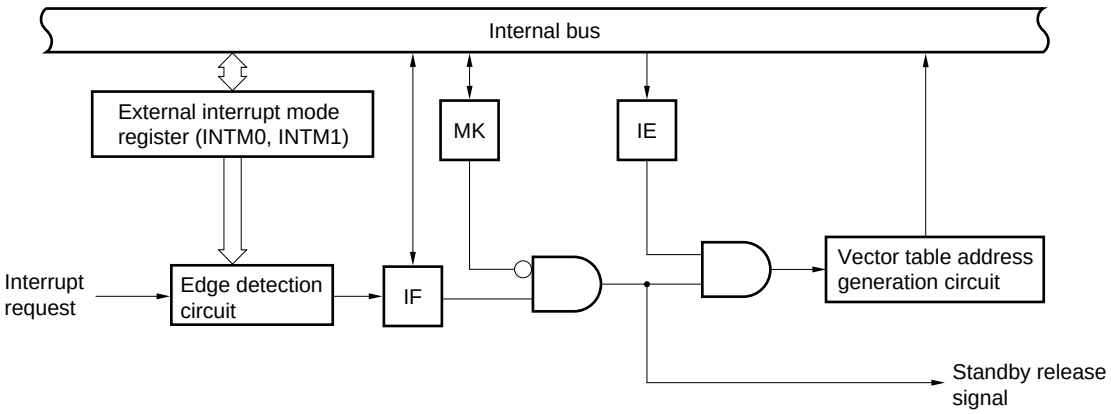
**(A) Internal nonmaskable interrupt**




**(B) Internal maskable interrupt**




**(C) External maskable interrupt**




IF : Interrupt request flag
IE : Interrupt enable flag
MK : Interrupt mask flag

### 6.3 Interrupt Function Control Registers

The interrupt functions are controlled by the following registers.

- Interrupt request flag registers (IF0 and IF1)
- Interrupt mask flag registers (MK0 and MK1)
- External interrupt mode registers (INTM0 and INTM1)
- Program status word (PSW)

Table 6-2 lists interrupt requests, the corresponding interrupt request flags, and interrupt mask flags.

**Table 6-2. Interrupt Request Signals and Corresponding Flags**

| Interrupt request signal | Interrupt request flag | Interrupt mask flag |
|---|---|---|
| INTWDT | TMIF4 | TMMK4 |
| INTP0 | PIF0 | PMK0 |
| INTP1 | PIF1 | PMK1 |
| INTP2 | PIF2 | PMK2 |
| INTP3 | PIF3 | PMK3 |
| INTSR20/INTCSI20 | SRIF20 | SRMK20 |
| INTST20 | STIF20 | STMK20 |
| INTWT | WTIF | WTMK |
| INTWTI | WTIFI | WTMKI |
| INTTM80 | TMIF80 | TMMK80 |
| INTTM81 | TMIF81 | TMMK81 |
| INTTM82 | TMIF82 | TMMK82 |
| INTTM90 | TMIF90 | TMMK90 |
| INTSMB0 | SMBIF0 | SMBMK0 |
| INTSMBOV0 | SMBOVIF0 | SMBOVMK0 |
| INTAD0 | ADIF0 | ADMK0 |

**(1) Interrupt request flag registers (IF0 and IF1)**

An interrupt request flag is set (1), when the corresponding interrupt request is issued, or when the related instruction is executed. It is cleared (0), when the interrupt request is accepted, when a $\overline{\text{RESET}}$ signal is input, or when a related instruction is executed.

IF0 and IF1 are manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears IF0 and IF1 to 00H.

**Figure 6-2. Format of Interrupt Request Flag Register**

| Symbol | ⑦ | ⑥ | ⑤ | ④ | ③ | ② | ① | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| IF0 | WTIF | STIF20 | SRIF20 | PIF3 | PIF2 | PIF1 | PIF0 | TMIF4 | FFE0H | 00H | R/W |
| | ⑦ | ⑥ | ⑤ | ④ | ③ | ② | ① | ⓪ | | | |
| IF1 | ADIF0 | SMBOVIF0 | SMBIF0 | TMIF90 | TMIF82 | TMIF81 | TMIF80 | WTIIF | FFE1H | 00H | R/W |

| XXIFX | Interrupt request flag |
|---|---|
| 0 | No interrupt request signal has been issued. |
| 1 | An interrupt request signal has been issued; an interrupt request has been made. |

**Cautions 1. The TMIF4 flag can be read- and write-accessed only when the watchdog timer is being used as an interval timer. It must be cleared to 0 if the watchdog timer is used in watchdog timer mode 1 or 2.**

**2. When port 3 is being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use port 3 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.**

**(2) Interrupt mask flag registers (MK0 and MK1)**

The interrupt mask flags are used to enable and disable the corresponding maskable interrupts.

MK0 and MK1 are manipulated using a 1-bit or 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input loads FFH into MK0 and MK1.

**Figure 6-3. Format of Interrupt Mask Flag Register**

| Symbol | ⑦ | ⑥ | ⑤ | ④ | ③ | ② | ① | ⓪ | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| MK0 | WTMK | STMK20 | SRMK20 | PMK3 | PMK2 | PMK1 | PMK0 | TMMK4 | FFE4H | FFH | R/W |
| | ⑦ | ⑥ | ⑤ | ④ | ③ | ② | ① | ⓪ | | | |
| MK1 | ADMK0 | SMBOVMK0 | SMBMK0 | TMMK90 | TMMK82 | TMMK81 | TMMK80 | WTIMK | FFE5H | FFH | R/W |

| XXMKX | Interrupt handling control |
|---|---|
| 0 | Enable interrupt handling. |
| 1 | Disable interrupt handling. |

**Cautions 1. When the watchdog timer is being used in watchdog timer mode 1 or 2, any attempt to read TMMK4 flag results in an undefined value being detected.**

**2. When port 3 is being used as an output port, and its output level is changed, an interrupt request flag is set, because this port is also used as an external interrupt input. To use port 3 in output mode, therefore, the interrupt mask flag must be set to 1 in advance.**

**(3) External interrupt mode register 0 (INTM0)**

INTM0 is used to specify an effective edge for INTP0 to INTP2.

INTM0 is manipulated using an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears INTM0 to 00H.

**Figure 6-4. Format of External Interrupt Mode Register 0**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| INTM0 | ES21 | ES20 | ES11 | ES10 | ES01 | ES00 | 0 | 0 | FFECH | 00H | R/W |

| ES21 | ES20 | INTP2 effective edge selection |
|---|---|---|
| 0 | 0 | Falling edge |
| 0 | 1 | Rising edge |
| 1 | 0 | Not to be set |
| 1 | 1 | Both rising and falling edges |

| ES11 | ES10 | INTP1 effective edge selection |
|---|---|---|
| 0 | 0 | Falling edge |
| 0 | 1 | Rising edge |
| 1 | 0 | Not to be set |
| 1 | 1 | Both rising and falling edges |

| ES01 | ES00 | INTP0 effective edge selection |
|---|---|---|
| 0 | 0 | Falling edge |
| 0 | 1 | Rising edge |
| 1 | 0 | Not to be set |
| 1 | 1 | Both rising and falling edges |

**Cautions 1. Bits 0 and 1 must be fixed to 0.**

**2. Before setting INTM0, set the corresponding interrupt mask flag register to 1 to disable interrupts.**

**To enable interrupts, clear (0) the corresponding interrupt request flag, then the corresponding interrupt mask flag register.**

**(4) External interrupt mode register 1 (INTM1)**

INTM1 is used to specify an effective edge for INTP3.

INTM1 is manipulated using an 8-bit memory manipulation instruction.

A $\overline{\text{RESET}}$ input clears INTM1 to 00H.

**Figure 6-5. Format of External Interrupt Mode Register 1**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| INTM1 | 0 | 0 | 0 | 0 | 0 | 0 | ES31 | ES30 | FFEDH | 00H | R/W |

| ES31 | ES30 | INTP3 effective edge selection |
|---|---|---|
| 0 | 0 | Falling edge |
| 0 | 1 | Rising edge |
| 1 | 0 | Not to be set |
| 1 | 1 | Both rising and falling edges |

**Cautions 1. Bits 2 to 7 must be fixed to 0.**

**2. Before setting INTM1, set the corresponding interrupt mask flag register to 1 to disable interrupts.**

**To enable interrupts, clear (0) the corresponding interrupt request flag, then the corresponding interrupt mask flag register.**

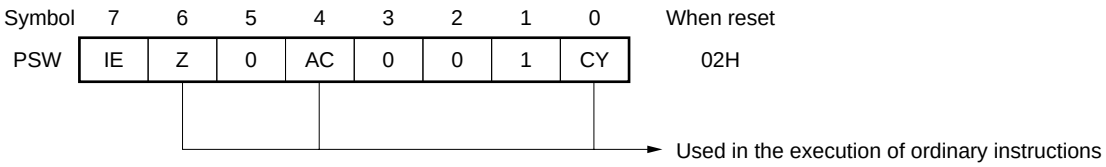
**(5) Program status word (PSW)**

The program status word is used to hold the instruction execution result and the current status of the interrupt requests. The IE flag, used to enable and disable maskable interrupts, is mapped to the PSW.

The PSW can be read- and write-accessed in 8-bit units, as well as in 1-bit units when using bit manipulation instructions and dedicated instructions (EI and DI). When a vector interrupt is accepted, the PSW is automatically saved to a stack, and the IE flag is reset (0).

A $\overline{\text{RESET}}$ input loads 02H into the PSW.

**Figure 6-6. Program Status Word Configuration**




| IE | Whether to enable/disable interrupt acceptance |
|---|---|
| 0 | Disable |
| 1 | Enable |

# 7. STANDBY FUNCTION

## 7.1 Standby Function

The standby function is supported to minimize the system's power consumption. There are two standby modes: HALT and STOP.

HALT and STOP modes are selected using the HALT and STOP instructions, respectively.

**(1) HALT mode**

In HALT mode, the CPU clock is stopped. Interleaving normal mode with HALT mode can reduce the average power consumption.

**(2) STOP mode**

In STOP mode, the main system clock is stopped. As a result, main system clock-based operation is also stopped, thus minimizing power consumption.

**Caution Before shifting to STOP mode, first stop the operation of the hardware, then execute the STOP instruction.**

**Table 7-1. Operation Statuses in HALT Mode**

| Item | HALT mode operation status while the main system clock is running | | HALT mode operation status while the subsystem clock is running | |
|---|---|---|---|---|
| | While the subsystem clock is running | While the subsystem clock is not running | While the main system clock is running | While the main system clock is not running |
| Clock generator | Can operate with the main system clock. | | | Does not run. |
| CPU | Operation disabled | | | |
| Port (output latch) | Remains in the state existing before the selection of HALT mode. | | | |
| 16-bit timer counter (TM90) | Operation enabled | Operation enabled Note 1 | Operation enabled | Operation enabled Note 2 |
| 8-bit timer/event counter (TM80) | Operation enabled | | | Operation enabled Note 3 |
| 8-bit timer/event counter (TM81) | Operation enabled | | | Operation enabled Note 4 |
| 8-bit timer counter (TM82) | Operation enabled | Operation enabled Note 1 | Operation enabled | Operation enabled Note 2 |
| Clock timer | Operation enabled | Operation enabled Note 1 | Operation enabled | Operation enabled Note 2 |
| Watchdog timer | Operation enabled | | | Operation disabled |
| Serial interface | Operation enabled | | | Operation enabled Note 5 |
| SMB | Operation enabled | | | Operation enabled Note 6 |
| A/D converter | Operation disabled | | | |
| Multiplier | Operation disabled | | | |
| External interrupt | Operation enabled Note 7 | | | |

**Notes 1.** Operation is enabled while the main system clock is selected.

**2.** Operation is enabled while the subsystem clock is selected.

**3.** Operation is enabled only when TI80 is selected as the count clock.

**4.** Operation is enabled only when TI81 is selected as the count clock.

**5.** Operation is enabled in both three-wire serial I/O and UART modes while an external clock is being used.

**6.** While in slave mode, an interrupt can be generated when an address match is found.

**7.** Maskable interrupt that is not masked

**Table 7-2. Operation Statuses in STOP Mode**

| Item | STOP mode operation status while the main system clock is running | |
|---|---|---|
| | While the subsystem clock is running | While the subsystem clock is not running |
| Clock generator | Does not operate with the main system clock. | |
| CPU | Operation disabled | |
| Port (output latch) | Remains in the state existing before the selection of STOP mode. | |
| 16-bit timer counter (TM90) | Operation enabled[Note 1] | Operation disabled |
| 8-bit timer/event counter (TM80) | Operation enabled[Note 2] | |
| 8-bit timer/event counter (TM81) | Operation enabled[Note 3] | |
| 8-bit timer counter (TM82) | Operation enabled[Note 1] | Operation disabled |
| Clock timer | Operation enabled[Note 1] | Operation disabled |
| Watchdog timer | Operation disabled | |
| Serial interface | Operation enabled[Note 4] | |
| SMB | Operation enabled[Note 5] | |
| A/D converter | Operation disabled | |
| Multiplier | Operation disabled | |
| External interrupt | Operation enabled[Note 6] | |

**Notes 1.** Operation is enabled while the subsystem clock is selected.

**2.** Operation is enabled only when TI80 is selected as the count clock.

**3.** Operation is enabled only when TI81 is selected as the count clock.

**4.** Operation is enabled in both three-wire serial I/O and UART modes while an external clock is being used.

**5.** While in slave mode, an interrupt can be generated when an address match is found.

**6.** Maskable interrupt that is not masked

## 7.2 Standby Function Control Register

The oscillation settling time selection register (OSTS) is used to control the wait time, from the time STOP mode is deselected by an interrupt request, until oscillation settles.

The OSTS is manipulated using an 8-bit memory manipulation instruction.

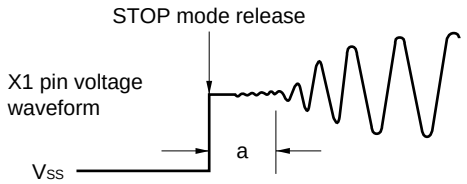
A $\overline{\text{RESET}}$ input loads 04H into the OSTS. If a $\overline{\text{RESET}}$ input is used to deselect STOP mode, the time required for oscillation to settle will be $2^{15}/f_X$, rather than $2^{17}/f_X$.

**Figure 7-1. Format of Oscillation Settling Time Selection Register**

| Symbol | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | Address | When reset | R/W |
|---|---|---|---|---|---|---|---|---|---|---|---|
| OSTS | 0 | 0 | 0 | 0 | 0 | OSTS2 | OSTS1 | OSTS0 | FFFAH | 04H | R/W |

| OSTS2 | OSTS1 | OSTS0 | Oscillation settling time selection |
|---|---|---|---|
| 0 | 0 | 0 | $2^{12}/f_X$ (819 μs) |
| 0 | 1 | 0 | $2^{15}/f_X$ (6.55 ms) |
| 1 | 0 | 0 | $2^{17}/f_X$ (26.2 ms) |
| Other settings | | | Not to be set |

**Caution The wait time required to deselect STOP mode does not include the time ("a" in the following figure) required for the clock oscillation to settle after STOP mode is deselected, regardless of whether STOP mode is deselected by a $\overline{\text{RESET}}$ input or interrupt.**




**Remarks 1.** $f_X$: Main system clock oscillation frequency

**2.** The parenthesized values apply to operation at $f_X$ = 5.0 MHz.

## 8. RESET FUNCTIONS

The μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y can be reset using the following signals.

(1) External reset signal input to the $\overline{\text{RESET}}$ pin
(2) Internal reset signal generated upon the elapse of the period set in the watchdog timer, used for detecting an unintended program loop

The external and internal reset signals are functionally equivalent. When $\overline{\text{RESET}}$ is input, they cause program execution to begin at the addresses indicated at addresses 0000H and 0001H, respectively.

If a low level signal is applied to the $\overline{\text{RESET}}$ pin, or if the watchdog timer overflows, a reset occurs, causing each piece of the hardware to enter the states listed in Table 8-1. While a reset signal is being input, or while the oscillation frequency is settling immediately after the end of a reset sequence, each pin remains in the high-impedance state.

If a high level signal is applied to the $\overline{\text{RESET}}$ pin, a reset sequence is terminated, and program execution begins once the oscillation settling time ($2^{15}/f_X$) elapses. A watchdog timer overflow-based reset sequence is terminated automatically. Similarly, program execution begins upon the elapse of the oscillation settling time ($2^{15}/f_X$).

**Cautions 1. To use an external reset sequence, supply a low level signal to the $\overline{\text{RESET}}$ pin and maintain the signal for at least 10 μs.**

**2. When a reset is used to deselect STOP mode, the information related to STOP mode is held during the reset sequence, that is, while the reset signal is applied. The port pins remain in the high-impedance state, however.**

**Figure 8-1. Reset Function Block Diagram**

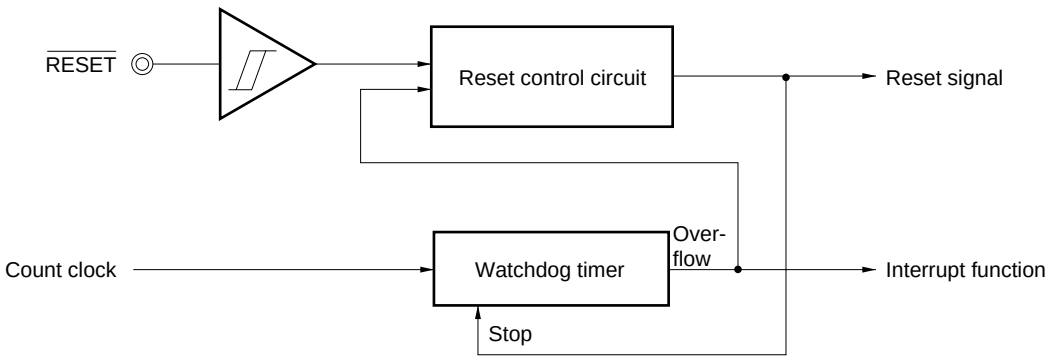


**Table 8-1. State of the Hardware after a Reset (1/2)**

| Hardware | | State after reset |
|---|---|---|
| Program counter (PC)[Note 1] | | Loaded with the contents of the reset vector table (0000H, 0001H) |
| Stack pointer (SP) | | Undefined |
| Program status word (PSW) | | 02H |
| RAM | Data memory | Undefined[Note 2] |
| | General-purpose register | Undefined[Note 2] |
| Ports (P0 to P3, P5, P6) (output latch) | | 00H |
| Port mode registers (PM0 to PM3, PM5) | | FFH |
| Pull-up resistor option registers (PU0, PUB2, PUB3) | | 00H |
| Processor clock control register (PCC) | | 02H |
| Suboscillation mode register (SCKM) | | 00H |
| Subclock control register (CSS) | | 00H |
| Oscillation settling time selection register (OSTS) | | 04H |
| 16-bit timer/counter 90 | Timer register (TM90) | 0000H |
| | Compare register (CR90) | FFFFH |
| | Capture register (TCP90) | Undefined |
| | Mode control register (TMC90) | 00H |
| | Buzzer output control register (BZC90) | 00H |
| 8-bit timer/event counters 80 to 82 | Timer registers (TM80 to TM82) | 00H |
| | Compare registers (CR80 to CR82) | Undefined |
| | Mode control registers (TMC80 to TMC82) | 00H |
| Clock timer | Mode control register (WTM) | 00H |
| Watchdog timer | Timer clock selection register (TCL2) | 00H |
| | Mode register (WDTM) | 00H |
| A/D converter | Mode register (ADM0) | 00H |
| | A/D input selection register (ADS0) | 00H |
| | A/D conversion result register (ADCR0) | Undefined |
| Serial interface 20 | Mode register (CSIM20) | 00H |
| | Asynchronous serial interface mode register (ASIM20) | 00H |
| | Asynchronous serial interface status register (ASIS20) | 00H |
| | Baud rate generator control register (BRGC20) | 00H |
| | Transmission shift register (TXS20) | FFH |
| | Reception buffer register (RXB20) | Undefined |

**Notes 1.** While a reset signal is being input, and during the oscillation settling period, the contents of the PC will be undefined, while the remainder of the hardware will be the same as after the reset.

**2.** In standby mode, the RAM enters the hold state after a reset.

**Table 8-1. State of the Hardware after a Reset (2/2)**

| Hardware | | State after reset |
|---|---|---|
| SMB0 | Control register (SMBC0) | 00H |
| | Status register (SMBS0) | 00H |
| | Clock selection register (SMBCL0) | 00H |
| | Slave address register (SMBSVA0) | 00H |
| | Mode register (SMBM0) | 20H |
| | Input level setting register (SMBVI0) | 00H |
| | Shift register (SMB0) | 00H |
| Multiplier | 16-bit multiplication result storage register (MUL0) | Undefined |
| | Multiplication data register (MRA0, MRB0) | Undefined |
| | Multiplier control register (MULC0) | 00H |
| Interrupts | Request flag registers (IF0, IF1) | 00H |
| | Mask flag registers (MK0, MK1) | FFH |
| | External interrupt mode registers (INTM0, INTM1) | 00H |

## 9. MASK OPTIONS

The μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y have the following mask options.

- Mask option for P50 to P53
  This option is used to specify whether to incorporate a pull-up resistor, as follows:
  **<1>** To indicate whether a pull-up resistor is to be incorporated, an individual bit is specified, independently of the other bits.
  **<2>** The specification of each bit indicates that a pull-up resistor is not to be incorporated.

## 10. INSTRUCTION SET OVERVIEW

The instruction set for the μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y is listed later.

### 10.1 Legend

#### 10.1.1 Operand formats and descriptions

The description made in the operand field of each instruction conforms to the operand format for the instructions listed below (the details conform with the assembly specification). If more than one operand format is listed for an instruction, one is selected. Uppercase letters, #, !, $, and a pair of [ and ] are used to specify keywords, which must be written exactly as they appear. The meanings of these special characters are as follows:

- #: Immediate data specification
- $: Relative address specification
- !: Absolute address specification
- [ and ]: Indirect address specification

Immediate data should be described using appropriate values or labels. The specification of values and labels must be accompanied by #, !, $, or a pair of [ and ].

Operand registers, expressed as r or rp in the formats, can be described using both functional names (X, A, C, etc.) and absolute names (R0, R1, R2, and other names listed in Table 11-1).

**Table 10-1. Operand Formats and Descriptions**

| Format | Description |
|---|---|
| r<br>rp<br>sfr | X (R0), A (R1), C (R2), B (R3), E (R4), D (R5), L (R6), H (R7)<br>AX (RP0), BC (RP1), DE (RP2), HL (RP3)<br>Special function register symbol |
| saddr<br>saddrp | FE20H to FF1FH: Immediate data or label<br>FE20H to FF1FH: Immediate data or label (even addresses only) |
| addr16<br><br>addr5 | 0000H to FFFFH: Immediate data or label<br>(only even address for 16-bit data transfer instructions)<br>0040H to 007FH: Immediate data or label (even addresses only) |
| word<br>byte<br>bit | 16-bit immediate data or label<br>8-bit immediate data or label<br>3-bit immediate data or label |

**Remark** For the special function register symbols, see **Table 4-1**.

### 10.1.2 Descriptions of the operation field

| | | |
|---|---|---|
| A | : | A register (8-bit accumulator) |
| X | : | X register |
| B | : | B register |
| C | : | C register |
| D | : | D register |
| E | : | E register |
| H | : | H register |
| L | : | L register |
| AX | : | AX register pair (16-bit accumulator) |
| BC | : | BC register pair |
| DE | : | DE register pair |
| HL | : | HL register pair |
| PC | : | Program counter |
| SP | : | Stack pointer |
| PSW | : | Program status word |
| CY | : | Carry flag |
| AC | : | Auxiliary carry flag |
| Z | : | Zero flag |
| IE | : | Interrupt request enable flag |
| NMIS | : | Flag to indicate that a nonmaskable interrupt is being handled |
| () | : | Contents of a memory location indicated by a parenthesized address or register name |
| $X_H$, $X_L$ | : | Upper and lower 8 bits of a 16-bit register |
| ^ | : | Logical product (AND) |
| ∨ | : | Logical sum (OR) |
| ∀ | : | Exclusive OR |
| — | : | Inverted data |
| addr16 | : | 16-bit immediate data or label |
| jdisp8 | : | Signed 8-bit data (displacement value) |

### 10.1.3 Description of the flag operation field

| | | |
|---|---|---|
| (blank) | : | No change |
| 0 | : | To be cleared to 0 |
| 1 | : | To be set to 1 |
| × | : | To be set or cleared according to the result |
| R | : | To be restored to the previous value |

## 10.2 Operations

| Mnemonic | Operand | Byte | Clock | Operation | Flag | | |
|---|---|---|---|---|---|---|---|
| | | | | | Z | AC | CY |
| MOV | r, #byte | 3 | 6 | r ← byte | | | |
| | saddr, #byte | 3 | 6 | (saddr) ← byte | | | |
| | sfr, #byte | 3 | 6 | sfr ← byte | | | |
| | A, r Note 1 | 2 | 4 | A ← r | | | |
| | r, A Note 1 | 2 | 4 | r ← A | | | |
| | A, saddr | 2 | 4 | A ← (saddr) | | | |
| | saddr, A | 2 | 4 | (saddr) ← A | | | |
| | A, sfr | 2 | 4 | A ← sfr | | | |
| | sfr, A | 2 | 4 | sfr ← A | | | |
| | A, !addr16 | 3 | 8 | A ← (addr16) | | | |
| | !addr16, A | 3 | 8 | (addr16) ← A | | | |
| | PSW, #byte | 3 | 6 | PSW ← byte | × | × | × |
| | A, PSW | 2 | 4 | A ← PSW | | | |
| | PSW, A | 2 | 4 | PSW ← A | × | × | × |
| | A, [DE] | 1 | 6 | A ← (DE) | | | |
| | [DE], A | 1 | 6 | (DE) ← A | | | |
| | A, [HL] | 1 | 6 | A ← (HL) | | | |
| | [HL], A | 1 | 6 | (HL) ← A | | | |
| | A, [HL + byte] | 2 | 6 | A ← (HL + byte) | | | |
| | [HL + byte], A | 2 | 6 | (HL + byte) ← A | | | |
| XCH | A, X | 1 | 4 | A ↔ X | | | |
| | A, r Note 2 | 2 | 6 | A ↔ r | | | |
| | A, saddr | 2 | 6 | A ↔ (saddr) | | | |
| | A, sfr | 2 | 6 | A ↔ (sfr) | | | |
| | A, [DE] | 1 | 8 | A ↔ (DE) | | | |
| | A, [HL] | 1 | 8 | A ↔ (HL) | | | |
| | A, [HL + byte] | 2 | 8 | A ↔ (HL + byte) | | | |
| MOVW | rp, #word | 3 | 6 | rp ← word | | | |
| | AX, saddrp | 2 | 6 | AX ← (saddrp) | | | |
| | saddrp, AX | 2 | 8 | (saddrp) ← AX | | | |
| | AX, rp Note 3 | 1 | 4 | AX ← rp | | | |
| | rp, AX Note 3 | 1 | 4 | rp ← AX | | | |

**Notes 1.** Except when r = A.

**2.** Except when r = A or X.

**3.** Only when rp = BC, DE, or HL.

**Remark** The instruction clock cycle is based on the CPU clock ($f_{CPU}$), specified in the processor clock controller register (PCC).

| Mnemonic | Operand | Byte | Clock | Operation | Flag | | |
|---|---|---|---|---|---|---|---|
| | | | | | Z | AC | CY |
| XCHW | AX, rp Note | 1 | 8 | AX ↔ rp | | | |
| ADD | A, #byte | 2 | 4 | A, CY ← A + byte | × | × | × |
| | saddr, #byte | 3 | 6 | (saddr), CY ← (saddr) + byte | × | × | × |
| | A, r | 2 | 4 | A, CY ← A + r | × | × | × |
| | A, saddr | 2 | 4 | A, CY ← A + (saddr) | × | × | × |
| | A, !addr16 | 3 | 8 | A, CY ← A + (addr16) | × | × | × |
| | A, [HL] | 1 | 6 | A, CY ← A + (HL) | × | × | × |
| | A, [HL + byte] | 2 | 6 | A, CY ← A + (HL + byte) | × | × | × |
| ADDC | A, #byte | 2 | 4 | A, CY ← A + byte + CY | × | × | × |
| | saddr, #byte | 3 | 6 | (saddr), CY ← (saddr) + byte + CY | × | × | × |
| | A, r | 2 | 4 | A, CY ← A + r + CY | × | × | × |
| | A, saddr | 2 | 4 | A, CY ← A + (saddr) + CY | × | × | × |
| | A, !addr16 | 3 | 8 | A, CY ← A + (addr16) + CY | × | × | × |
| | A, [HL] | 1 | 6 | A, CY ← A + (HL) + CY | × | × | × |
| | A, [HL + byte] | 2 | 6 | A, CY ← A + (HL + byte) + CY | × | × | × |
| SUB | A, #byte | 2 | 4 | A, CY ← A – byte | × | × | × |
| | saddr, #byte | 3 | 6 | (saddr), CY ← (saddr) – byte | × | × | × |
| | A, r | 2 | 4 | A, CY ← A – r | × | × | × |
| | A, saddr | 2 | 4 | A, CY ← A – (saddr) | × | × | × |
| | A, !addr16 | 3 | 8 | A, CY ← A – (addr16) | × | × | × |
| | A, [HL] | 1 | 6 | A, CY ← A – (HL) | × | × | × |
| | A, [HL + byte] | 2 | 6 | A, CY ← A – (HL + byte) | × | × | × |
| SUBC | A, #byte | 2 | 4 | A, CY ← A – byte – CY | × | × | × |
| | saddr, #byte | 3 | 6 | (saddr), CY ← (saddr) – byte – CY | × | × | × |
| | A, r | 2 | 4 | A, CY ← A – r – CY | × | × | × |
| | A, saddr | 2 | 4 | A, CY ← A – (saddr) – CY | × | × | × |
| | A, !addr16 | 3 | 8 | A, CY ← A – (addr16) – CY | × | × | × |
| | A, [HL] | 1 | 6 | A, CY ← A – (HL) – CY | × | × | × |
| | A, [HL + byte] | 2 | 6 | A, CY ← A – (HL + byte) – CY | × | × | × |
| AND | A, #byte | 2 | 4 | A ← A ∧ byte | × | | |
| | saddr, #byte | 3 | 6 | (saddr) ← (saddr) ∧ byte | × | | |
| | A, r | 2 | 4 | A ← A ∧ r | × | | |
| | A, saddr | 2 | 4 | A ← A ∧ (saddr) | × | | |
| | A, !addr16 | 3 | 8 | A ← A ∧ (addr16) | × | | |
| | A, [HL] | 1 | 6 | A ← A ∧ (HL) | × | | |
| | A, [HL + byte] | 2 | 6 | A ← A ∧ (HL + byte) | × | | |

**Note** Only when rp = BC, DE, or HL.

**Remark** The instruction clock cycle is based on the CPU clock ($f_{CPU}$), specified in the processor clock controller register (PCC).

| Mnemonic | Operand | Byte | Clock | Operation | Flag | | |
|---|---|---|---|---|---|---|---|
| | | | | | Z | AC | CY |
| OR | A, #byte | 2 | 4 | A ← A ∨ byte | × | | |
| | saddr, #byte | 3 | 6 | (saddr) ← (saddr) ∨ byte | × | | |
| | A, r | 2 | 4 | A ← A ∨ r | × | | |
| | A, saddr | 2 | 4 | A ← A ∨ (saddr) | × | | |
| | A, !addr16 | 3 | 8 | A ← A ∨ (addr16) | × | | |
| | A, [HL] | 1 | 6 | A ← A ∨ (HL) | × | | |
| | A, [HL + byte] | 2 | 6 | A ← A ∨ (HL + byte) | × | | |
| XOR | A, #byte | 2 | 4 | A ← A ⊻ byte | × | | |
| | saddr, #byte | 3 | 6 | (saddr) ← (saddr) ⊻ byte | × | | |
| | A, r | 2 | 4 | A ← A ⊻ r | × | | |
| | A, saddr | 2 | 4 | A ← A ⊻ (saddr) | × | | |
| | A, !addr16 | 3 | 8 | A ← A ⊻ (addr16) | × | | |
| | A, [HL] | 1 | 6 | A ← A ⊻ (HL) | × | | |
| | A, [HL + byte] | 2 | 6 | A ← A ⊻ (HL + byte) | × | | |
| CMP | A, #byte | 2 | 4 | A – byte | × | × | × |
| | saddr, #byte | 3 | 6 | (saddr) – byte | × | × | × |
| | A, r | 2 | 4 | A – r | × | × | × |
| | A, saddr | 2 | 4 | A – (saddr) | × | × | × |
| | A, !addr16 | 3 | 8 | A – (addr16) | × | × | × |
| | A, [HL] | 1 | 6 | A – (HL) | × | × | × |
| | A, [HL + byte] | 2 | 6 | A – (HL + byte) | × | × | × |
| ADDW | AX, #word | 3 | 6 | AX, CY ← AX + word | × | × | × |
| SUBW | AX, #word | 3 | 6 | AX, CY ← AX – word | × | × | × |
| CMPW | AX, #word | 3 | 6 | AX – word | × | × | × |
| INC | r | 2 | 4 | r ← r + 1 | × | × | |
| | saddr | 2 | 4 | (saddr) ← (saddr) + 1 | × | × | |
| DEC | r | 2 | 4 | r ← r – 1 | × | × | |
| | saddr | 2 | 4 | (saddr) ← (saddr) – 1 | × | × | |
| INCW | rp | 1 | 4 | rp ← rp + 1 | | | |
| DECW | rp | 1 | 4 | rp ← rp – 1 | | | |
| ROR | A, 1 | 1 | 2 | $(CY, A_7 \leftarrow A_0, A_{m-1} \leftarrow A_m) \times 1$ | | | × |
| ROL | A, 1 | 1 | 2 | $(CY, A_0 \leftarrow A_7, A_{m+1} \leftarrow A_m) \times 1$ | | | × |
| RORC | A, 1 | 1 | 2 | $(CY \leftarrow A_0, A_7 \leftarrow CY, A_{m-1} \leftarrow A_m) \times 1$ | | | × |
| ROLC | A, 1 | 1 | 2 | $(CY \leftarrow A_7, A_0 \leftarrow CY, A_{m+1} \leftarrow A_m) \times 1$ | | | × |

**Remark** The instruction clock cycle is based on the CPU clock ($f_{CPU}$), specified in the processor clock controller register (PCC).

| Mnemonic | Operand | Byte | Clock | Operation | Flag | | |
|---|---|---|---|---|---|---|---|
| | | | | | Z | AC | CY |
| SET1 | saddr. bit | 3 | 6 | (saddr. bit) ← 1 | | | |
| | sfr. bit | 3 | 6 | sfr. bit ← 1 | | | |
| | A. bit | 2 | 4 | A. bit ← 1 | | | |
| | PSW. bit | 3 | 6 | PSW. bit ← 1 | × | × | × |
| | [HL]. bit | 2 | 10 | (HL). bit ← 1 | | | |
| CLR1 | saddr. bit | 3 | 6 | (saddr. bit) ← 0 | | | |
| | sfr. bit | 3 | 6 | sfr. bit ← 0 | | | |
| | A. bit | 2 | 4 | A. bit ← 0 | | | |
| | PSW. bit | 3 | 6 | PSW. bit ← 0 | × | × | × |
| | [HL]. bit | 2 | 10 | (HL). bit ← 0 | | | |
| SET1 | CY | 1 | 2 | CY ← 1 | | | 1 |
| CLR1 | CY | 1 | 2 | CY ← 0 | | | 0 |
| NOT1 | CY | 1 | 2 | CY ← $\overline{CY}$ | | | × |
| CALL | !addr16 | 3 | 6 | (SP – 1) ← (PC + 3)$_H$, (SP – 2) ← (PC + 3)$_L$, PC ← addr16, SP ← SP – 2 | | | |
| CALLT | [addr5] | 1 | 8 | (SP – 1) ← (PC + 1)$_H$, (SP – 2) ← (PC + 1)$_L$, PC$_H$ ← (00000000, addr5 + 1), PC$_L$ ← (00000000, addr5), SP ← SP – 2 | | | |
| RET | | 1 | 6 | PC$_H$ ← (SP + 1), PC$_L$ ← (SP), SP ← SP + 2 | | | |
| RETI | | 1 | 8 | PC$_H$ ← (SP + 1), PC$_L$ ← (SP), PSW ← (SP + 2), SP ← SP + 3, NMIS ← 0 | R | R | R |
| PUSH | PSW | 1 | 2 | (SP – 1) ← PSW, SP ← SP – 1 | | | |
| | rp | 1 | 4 | (SP – 1) ← rp$_H$, (SP – 2) ← rp$_L$, SP ← SP – 2 | | | |
| POP | PSW | 1 | 4 | PSW ← (SP), SP ← SP + 1 | R | R | R |
| | rp | 1 | 6 | rp$_H$ ← (SP + 1), rp$_L$ ← (SP), SP ← SP + 2 | | | |
| MOVW | SP, AX | 2 | 8 | SP ← AX | | | |
| | AX, SP | 2 | 6 | AX ← SP | | | |
| BR | !addr16 | 3 | 6 | PC ← addr16 | | | |
| | $addr16 | 2 | 6 | PC ← PC + 2 + jdisp8 | | | |
| | AX | 1 | 6 | PC$_H$ ← A, PC$_L$ ← X | | | |

**Remark** The instruction clock cycle is based on the CPU clock ($f_{CPU}$), specified in the processor clock controller register (PCC).

| Mnemonic | Operand | Byte | Clock | Operation | Flag |
|---|---|---|---|---|---|
| | | | | | Z AC CY |
| BC | $addr16 | 2 | 6 | PC ← PC + 2 + jdisp8 if CY = 1 | |
| BNC | $addr16 | 2 | 6 | PC ← PC + 2 + jdisp8 if CY = 0 | |
| BZ | $addr16 | 2 | 6 | PC ← PC + 2 + jdisp8 if Z = 1 | |
| BNZ | $addr16 | 2 | 6 | PC ← PC + 2 + jdisp8 if Z = 0 | |
| BT | saddr. bit, $addr16 | 4 | 10 | PC ← PC + 4 + jdisp8<br>if (saddr. bit) = 1 | |
| | sfr. bit, $addr16 | 4 | 10 | PC ← PC + 4 + jdisp8 if sfr. bit = 1 | |
| | A. bit, $addr16 | 3 | 8 | PC ← PC + 3 + jdisp8 if A. bit = 1 | |
| | PSW. bit, $addr16 | 4 | 10 | PC ← PC + 4 + jdisp8 if PSW. bit = 1 | |
| BF | saddr. bit, $addr16 | 4 | 10 | PC ← PC + 4 + jdisp8<br>if (saddr. bit) = 0 | |
| | sfr. bit, $addr16 | 4 | 10 | PC ← PC + 4 + jdisp8 if sfr. bit = 0 | |
| | A. bit, $addr16 | 3 | 8 | PC ← PC + 3 + jdisp8 if A. bit = 0 | |
| | PSW. bit, $addr16 | 4 | 10 | PC ← PC + 4 + jdisp8 if PSW. bit = 0 | |
| DBNZ | B, $addr16 | 2 | 6 | B ← B – 1, then<br>PC ← PC + 2 + jdisp8 if B ≠ 0 | |
| | C, $addr16 | 2 | 6 | C ← C – 1, then<br>PC ← PC + 2 + jdisp8 if C ≠ 0 | |
| | saddr, $addr16 | 3 | 8 | (saddr) ← (saddr) – 1, then<br>PC ← PC + 3 + jdisp8 if (saddr) ≠ 0 | |
| NOP | | 1 | 2 | No Operation | |
| EI | | 3 | 6 | IE ← 1 (Enable Interrupt) | |
| DI | | 3 | 6 | IE ← 0 (Disable Interrupt) | |
| HALT | | 1 | 2 | Set HALT Mode | |
| STOP | | 1 | 2 | Set STOP Mode | |

**Remark** The instruction clock cycle is based on the CPU clock ($f_{CPU}$), specified in the processor clock controller register (PCC).

## 11. ELECTRICAL CHARACTERISTICS

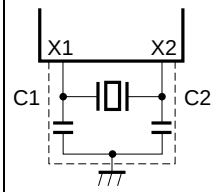
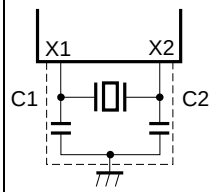
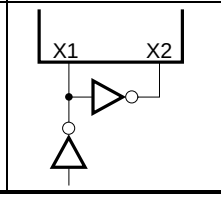
### ABSOLUTE MAXIMUM RATINGS ($T_A$ = 25°C)

| Parameter | Symbol | Conditions | | Rated value | Unit |
|---|---|---|---|---|---|
| Supply voltage | $V_{DD}$ | | | –0.3 to +7.0 | V |
| Input voltage | $V_{I1}$ | Pins other than those for port 5 | | –0.3 to $V_{DD}$ + 0.3 | V |
| | $V_{I2}$ | P50 to P53 | N-ch open drain | –0.3 to +13 | V |
| Output voltage | $V_O$ | | | –0.3 to $V_{DD}$ + 0.3 | V |
| High-level output current | $I_{OH}$ | Each pin | | –10 | mA |
| | | Total for all pins | | –30 | mA |
| Low-level output current | $I_{OL}$ | Each pin | | 30 | mA |
| | | Total for all pins | | 160 | mA |
| Operating ambient temperature | $T_A$ | | | –40 to +85 | °C |
| Storage temperature | $T_{stg}$ | | | –65 to +150 | °C |

**Caution Absolute maximum ratings are rated values beyond which physical damage will be caused to the product; if the rated value of any of the parameters in the above table is exceeded, even momentarily, the quality of the product may deteriorate. Always use the product within its rated values.**

**Remark** The characteristic of a dual-function pin does not differ between the port function and the secondary function, unless otherwise stated.

**CHARACTERISTICS OF THE MAIN SYSTEM CLOCK OSCILLATION CIRCUIT**
**($T_A$ = –40°C to +85°C, $V_{DD}$ = 1.8 to 5.5 V)**

| Resonator | Recommended circuit | Parameter | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Ceramic resonator |  | Oscillator frequency ($f_X$) Note 1 | $V_{DD}$ = oscillation voltage range | 1.0 | | 5.0 | MHz |
| | | Oscillation settling time Notes 2, 3 | Reset by $\overline{\text{RESET}}$ | | $2^{15}/f_X$ | 4 | ms |
| | | | Reset by an interrupt | | Note 4 | | |
| Crystal |  | Oscillator frequency ($f_X$) Note 1 | | 1.0 | | 5.0 | MHz |
| | | Oscillation settling time Note 2 | $V_{DD}$ = 4.5 to 5.5 V | | | 10 | ms |
| | | | | | | 30 | |
| External clock |  | X1 input frequency ($f_X$) Note 1 | | 1.0 | | 5.0 | MHz |
| | | X1 input high/low level width ($t_{XH}$, $t_{XL}$) | | 85 | | 500 | ns |

**Notes 1.** Only the characteristic of the oscillation circuit is indicated. See the description of the AC characteristic for the instruction execution time.

**2.** Time required for oscillation to settle once a reset sequence ends or STOP mode is deselected.

**3.** Time after $V_{DD}$ reaches MIN. of the oscillation voltage range.

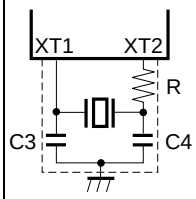
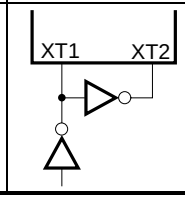
**4.** Bits 0 to 2 (OSTS0 to OSTS2) of the oscillation settling time selection register can be used to select $2^{12}/f_X$, $2^{15}/f_X$, or $2^{17}/f_X$.

**Cautions 1. When using the main system clock oscillation circuit, observe the following conditions for the wiring of that section enclosed in dotted lines in the above diagrams, so as to avoid the influence of the wiring capacitance.**

- **Keep the wiring as short as possible.**
- **Do not allow signal wires to cross one another.**
- **Keep the wiring away from wires that carry a high, non-stable current.**
- **Keep the grounding point of the capacitors at the same level as $V_{SS}$.**
- **Do not connect the grounding point to a grounding wire that carries a high current.**
- **Do not extract a signal from the oscillation circuit.**

**2. Before switching from the subsystem clock back to the main system clock, always allow sufficient time for the oscillation to settle by specifying it in the program.**

**CHARACTERISTICS OF THE SUBSYSTEM CLOCK OSCILLATION CIRCUIT**
**($T_A$ = –40°C to +85°C, $V_{DD}$ = 1.8 to 5.5 V)**

| Resonator | Recommended circuit | Parameter | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Crystal |  | Oscillator frequency ($f_{XT}$) Note 1 | | 32 | 32.768 | 35 | kHz |
| | | Oscillation settling time Note 2 | $V_{DD}$ = 4.5 to 5.5 V | | 1.2 | 2 | s |
| | | | | | | 10 | |
| External clock |  | XT1 input frequency ($f_{XT}$) Note 1 | | 32 | | 35 | kHz |
| | | XT1 input high/low level width ($t_{XTH}$, $t_{XTL}$) | | 14.3 | | 15.6 | μs |

**Notes 1.** Only the characteristic of the oscillation circuit is indicated. See the description of the AC characteristic for the instruction execution time.

**2.** Time required for oscillation to settle after $V_{DD}$ reaches the MIN. value of the oscillation voltage range.

**Cautions 1. When using the subsystem clock oscillation circuit, observe the following conditions for the wiring of that section enclosed in dotted lines in the above diagrams, so as to avoid the influence of the wiring capacitance.**

- **Keep the wiring as short as possible.**
- **Do not allow signal wires to cross one another.**
- **Keep the wiring away from wires that carry a high, non-stable current.**
- **Keep the grounding point of the capacitors at the same level as $V_{SS}$.**
- **Do not connect the grounding point to a grounding wire that carries a high current.**
- **Do not extract a signal from the oscillation circuit.**

**2. The subsystem clock oscillation circuit is designed to have a low amplification degree so as to maintain a low current drain. Therefore, it is more likely to malfunction as a result of noise than the main system clock oscillation circuit. When using the subsystem clock, therefore, pay particularly careful attention to how it is wired.**

**DC CHARACTERISTICS ($T_A$ = –40°C to +85°C, $V_{DD}$ = 1.8 to 5.5 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Low-level output current | $I_{OL}$ | Each pin | | | | Undefined | mA |
| | | All pins | | | | 80 | mA |
| High-level output current | $I_{OH}$ | Each pin | | | | Undefined | mA |
| | | All pins | | | | –15 | mA |
| High-level input voltage | $V_{IH1}$ | P00 to P05, P10, P11, P60 to P67 | $V_{DD}$ = 2.7 to 5.5 V | 0.7$V_{DD}$ | | $V_{DD}$ | V |
| | | | | 0.9$V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH2}$ | P50 to P53 | $V_{DD}$ = 2.7 to 5.5 V | 0.7$V_{DD}$ | | 12 | V |
| | | | | 0.9$V_{DD}$ | | 12 | V |
| | $V_{IH3}$ | $\overline{\text{RESET}}$, P20 to P26, P30 to P33 | $V_{DD}$ = 2.7 to 5.5 V | 0.8$V_{DD}$ | | $V_{DD}$ | V |
| | | | | 0.9$V_{DD}$ | | $V_{DD}$ | V |
| | $V_{IH4}$ | X1, X2 | | $V_{DD}$ – 0.1 | | $V_{DD}$ | V |
| Low-level input voltage | $V_{IL1}$ | P00 to P05, P10, P11, P60 to P67 | $V_{DD}$ = 2.7 to 5.5 V | 0 | | 0.3$V_{DD}$ | V |
| | | | | 0 | | 0.1$V_{DD}$ | V |
| | $V_{IL2}$ | P50 to P53 | $V_{DD}$ = 2.7 to 5.5 V | 0 | | 0.3$V_{DD}$ | V |
| | | | | 0 | | 0.1$V_{DD}$ | V |
| | $V_{IL3}$ | $\overline{\text{RESET}}$, P20 to P26, P30 to P33 | $V_{DD}$ = 2.7 to 5.5 V | 0 | | 0.2$V_{DD}$ | V |
| | | | | 0 | | 0.1$V_{DD}$ | V |
| | $V_{IL4}$ | X1, X2 | | 0 | | 0.1 | V |
| High-level output voltage | $V_{OH}$ | P00 to P05, P10, P11, P20 to P22, P25, P26, P30 to P33 | $V_{DD}$ = 4.5 to 5.5 V, $I_{OH}$ = –1 mA | $V_{DD}$ – 1.0 | | | V |
| | | | $V_{DD}$ = 1.8 to 5.5 V, $I_{OH}$ = –100 μA | $V_{DD}$ – 0.5 | | | V |
| Low-level output voltage | $V_{OL1}$ | Pins other than those for port 5 | $V_{DD}$ = 4.5 to 5.5 V, $I_{OL}$ = 10 mA | | | 1.0 | V |
| | | | $V_{DD}$ = 1.8 to 5.5 V, $I_{OL}$ = 400 μA | | | 0.5 | V |
| | $V_{OL2}$ | P50 to P53 | $V_{DD}$ = 4.5 to 5.5 V, $I_{OL}$ = 10 mA | | | 1.0 | V |
| | | | $V_{DD}$ = 1.8 to 5.5 V, $I_{OL}$ = 1.6 mA | | | 0.4 | V |
| High-level input leakage current | $L_{IH1}$ | $V_{IN}$ = $V_{DD}$ | Pins other than the X1 pin, X2 pin, or those for port 5 | | | 3 | μA |
| | $L_{IH2}$ | | X1, X2 | | | 20 | μA |
| | $L_{IH3}$ | $V_{IN}$ = 12 V | P50 to P53 (N-ch open drain) | | | 20 | μA |
| Low-level input leakage current | $L_{IL1}$ | $V_{IN}$ = 0 V | Pins other than the X1 pin, X2 pin, or those for port 5 | | | –3 | μA |
| | $L_{IL2}$ | | X1, X2 | | | –20 | μA |
| | $L_{IL3}$ | | P50 to P53 (N-ch open drain) During input instruction execution | | | –30 | μA |

**Remark** The characteristic of a dual-function pin does not differ between the port function and the secondary function, unless otherwise stated.

**DC CHARACTERISTICS ($T_A$ = –40°C to +85°C, $V_{DD}$ = 1.8 to 5.5 V)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| High-level output leakage current | $I_{LOH}$ | $V_{OUT} = V_{DD}$ | | | | 3 | μA |
| Low-level output leakage current | $I_{LOL}$ | $V_{OUT}$ = 0 V | | | | –3 | μA |
| Software-specified pull-up resistor | $R_1$ | $V_{IN}$ = 0 V, for pins other than those for P23, P24, and P50 to P53 | | 50 | 100 | 200 | kΩ |
| Mask option-specified pull-up resistor | $R_2$ | $V_{IN}$ = 0 V, P50 to P53 | | 15 | 30 | 60 | kΩ |
| Power supply current Note 1 | $I_{DD1}$ | 5.0-MHz crystal oscillation operating mode | $V_{DD}$ = 5.0 V ± 10% Note 3 | | 5.5 | 16.5 | mA |
| | | | $V_{DD}$ = 3.0 V ± 10% | | 0.7 | 2.1 | mA |
| | | | $V_{DD}$ = 2.0 V ± 10% | | 0.4 | 1.2 | mA |
| | $I_{DD2}$ | 5.0-MHz crystal oscillation HALT mode | $V_{DD}$ = 5.0 V ± 10% | | 1.2 | 3.6 | mA |
| | | | $V_{DD}$ = 3.0 V ± 10% | | 0.5 | 1.5 | mA |
| | | | $V_{DD}$ = 2.0 V ± 10% | | 0.3 | 0.9 | mA |
| | $I_{DD3}$ | 32.768-kHz crystal oscillation operating mode Note 2 | $V_{DD}$ = 5.0 V ± 10% | | 100 | 200 | μA |
| | | | $V_{DD}$ = 3.0 V ± 10% | | 70 | 140 | μA |
| | | | $V_{DD}$ = 2.0 V ± 10% | | 50 | 100 | μA |
| | $I_{DD4}$ | 32.768-kHz crystal oscillation HALT mode Note 2 | $V_{DD}$ = 5.0 V ± 10% | | 25 | 55 | μA |
| | | | $V_{DD}$ = 3.0 V ± 10% | | 5 | 25 | μA |
| | | | $V_{DD}$ = 2.0 V ± 10% | | 2.5 | 12.5 | μA |
| | $I_{DD5}$ | 32.768-kHz crystal stop STOP mode | $V_{DD}$ = 5.0 V ± 10% | | 0.1 | 30 | μA |
| | | | $V_{DD}$ = 3.0 V ± 10% | | 0.05 | 10 | μA |
| | | | $V_{DD}$ = 2.0 V ± 10% | | 0.05 | 10 | μA |
| | $I_{DD6}$ | 5.0-MHz crystal oscillation A/D operating mode | $V_{DD}$ = 5.0 V ± 10% | | 6.1 | 18.3 | mA |
| | | | $V_{DD}$ = 3.0 V ± 10% | | 1.3 | 2.9 | mA |
| | | | $V_{DD}$ = 2.0 V ± 10% | | 1.0 | 3.0 | mA |

**Notes 1.** The power supply current does not include $AV_{REF}$, $AV_{DD}$, or the port current (including the current flowing through the built-in pull-up resistor).

**2.** When the main system clock is not running.

**3.** During high-speed mode operation (when the processor clock control register (PCC) is cleared to 00H.)

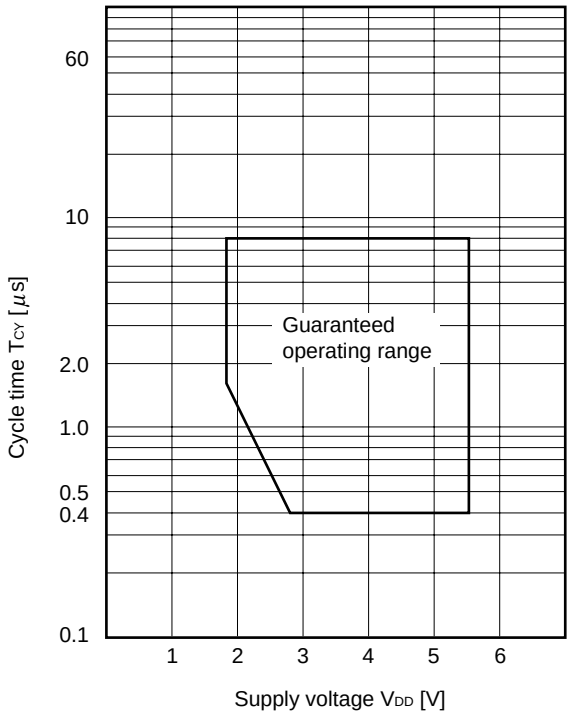
**Remark** The characteristic of a dual-function pin does not differ between the port function and the secondary function, unless otherwise stated.

## AC CHARACTERISTICS

### (1) Basic operations ($T_A$ = –40°C to +85°C, $V_{DD}$ = 1.8 to 5.5 V)

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| Cycle time (minimum instruction execution time) | $T_{CY}$ | Operation based on the main system clock | $V_{DD}$ = 2.7 to 5.5 V | 0.4 | | 8 | μs |
| | | | | 1.6 | | 8 | μs |
| | | Operation based on the subsystem clock | | | 122 | | μs |
| TI80 and TI81 input high/low level width | $t_{TIH}$, $t_{TIL}$ | $V_{DD}$ = 2.7 to 5.5 V | | 0.1 | | | μs |
| | | | | 1.8 | | | μs |
| TI80 and TI81 input frequency | $f_{TI}$ | $V_{DD}$ = 2.7 to 5.5 V | | 0 | | 4 | MHz |
| | | | | 0 | | 275 | kHz |
| Interrupt input high/low level width | $t_{INTH}$, $t_{INTL}$ | INTP0 to INTP3 | $V_{DD}$ = 2.7 to 5.5 V | 10 | | | μs |
| | | | | 20 | | | μs |
| $\overline{\text{RESET}}$ low level width | $t_{RSL}$ | $V_{DD}$ = 2.7 to 5.5 V | | 10 | | | μs |
| | | | | 20 | | | μs |

**$T_{CY}$ vs $V_{DD}$ (main system clock)**



**SERIAL INTERFACE ($T_A$ = –40°C to +85°C, $V_{DD}$ = 1.8 to 5.5 V)**

**(a) Serial interface 20**

**(i) Three-wire serial I/O mode ($\overline{SCK}$...Internal clock output)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| $\overline{SCK}$ cycle time | $t_{KCY1}$ | $V_{DD}$ = 2.7 to 5.5 V | | 800 | | | ns |
| | | | | 3,200 | | | ns |
| $\overline{SCK}$ high/low level width | $t_{KH1}$, $t_{KL1}$ | $V_{DD}$ = 2.7 to 5.5 V | | $t_{KCY1}$/2–50 | | | ns |
| | | | | $t_{KCY1}$/2–150 | | | ns |
| SI setup time (for $\overline{SCK}$ ↑) | $t_{SIK1}$ | $V_{DD}$ = 2.7 to 5.5 V | | 150 | | | ns |
| | | | | 500 | | | ns |
| SI hold time (for $\overline{SCK}$ ↑) | $t_{KSI1}$ | $V_{DD}$ = 2.7 to 5.5 V | | 400 | | | ns |
| | | | | 600 | | | ns |
| Delay from $\overline{SCK}$ ↓ to SO output | $t_{KSO1}$ | R = 1 kΩ, C = 100 pF[Note] | $V_{DD}$ = 2.7 to 5.5 V | 0 | | 250 | ns |
| | | | | 0 | | 1,000 | ns |

**Note** R and C are the resistance and capacitance of the SO output line, respectively.

**(ii) Three-wire serial I/O mode ($\overline{SCK}$...External clock output)**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| $\overline{SCK}$ cycle time | $t_{KCY2}$ | $V_{DD}$ = 2.7 to 5.5 V | | 800 | | | ns |
| | | | | 3,200 | | | ns |
| $\overline{SCK}$ high/low level width | $t_{KH2}$, $t_{KL2}$ | $V_{DD}$ = 2.7 to 5.5 V | | 400 | | | ns |
| | | | | 1,600 | | | ns |
| SI setup time (for $\overline{SCK}$ ↑) | $t_{SIK2}$ | $V_{DD}$ = 2.7 to 5.5 V | | 100 | | | ns |
| | | | | 150 | | | ns |
| SI hold time (for $\overline{SCK}$ ↑) | $t_{KSI2}$ | $V_{DD}$ = 2.7 to 5.5 V | | 400 | | | ns |
| | | | | 600 | | | ns |
| Delay from $\overline{SCK}$ ↓ to SO output | $t_{KSO2}$ | R = 1 kΩ, C = 100 pF[Note] | $V_{DD}$ = 2.7 to 5.5 V | 0 | | 300 | ns |
| | | | | 0 | | 1,000 | ns |

**Note** R and C are the resistance and capacitance of the SO output line, respectively.

**(iii) UART mode (internal clock output)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Transfer rate | | $V_{DD}$ = 2.7 to 5.5 V | | | 78,125 | bps |
| | | | | | 19,531 | bps |

**(iv) UART mode (external clock input)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| ASCK cycle time | $t_{KCY3}$ | $V_{DD}$ = 2.7 to 5.5 V | 800 | | | ns |
| | | | 3,200 | | | ns |
| ASCK high/low level width | $t_{KH3}$, $t_{KL3}$ | $V_{DD}$ = 2.7 to 5.5 V | 400 | | | ns |
| | | | 1,600 | | | ns |
| Transfer rate | | $V_{DD}$ = 2.7 to 5.5 V | | | 39,063 | bps |
| | | | | | 9,766 | bps |
| ASCK rising time, falling time | $t_R$, $t_F$ | | | | 1 | μs |

**(b) SMB0**

**(i) DC characteristics**

| Parameter | Symbol | Conditions | | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|---|
| High-level input voltage | $V_{IH}$ | SCL0, SDA0 | 2.7 V ≤ $V_{DD}$ < 5.5 V | 0.8$V_{DD}$ | | $V_{DD}$ | V |
| | | (during hysteresis) | | 0.9$V_{DD}$ | | $V_{DD}$ | V |
| Low-level input voltage | $V_{IL}$ | SCL0, SDA0 | 2.7 V ≤ $V_{DD}$ < 5.5 V | 0 | | 0.2$V_{DD}$ | V |
| | | (during hysteresis) | | 0 | | 0.1 | V |
| Low-level output voltage | $V_{OL}$ | SCL0, SDA0 | 4.5 V ≤ $V_{DD}$ < 5.5 V<br>$I_{OL}$ = 10 mA | | | 1.0 | V |
| | | | $I_{OL}$ = 400 μA | | | 0.5 | V |
| High-level input leakage current | $I_{LIH}$ | SCL0, SDA0 | $V_{IN}$ = $V_{DD}$ | | | 3 | μA |
| Low-level input leakage current | $I_{LIL}$ | SCL0, SDA0 | $V_{IN}$ = 0 V | | | –3 | μA |

**(ii) DC characteristics (when the comparator is used)**

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Input range | $V_{SDA}$, $V_{SCL}$ | 1.8 V ≤ $V_{DD}$ < 5.5 V | 0 | | 5.5 | V |
| Transfer level | $VI_{SDA}$, $VI_{SCL}$ | | 0.6 | 1.0 | 1.4 | V |

**(iii) AC characteristics**

| Parameter | | Symbol | SMB mode | | I²C standard mode | | I²C high-speed mode | | Unit |
|---|---|---|---|---|---|---|---|---|---|
| | | | MIN. | MAX. | MIN. | MAX. | MIN. | MAX. | |
| SCL0 clock frequency | | $f_{CLK}$ | 10 | 100 | 0 | 100 | 0 | 400 | kHz |
| Bus free time (between stop-start conditions) | | $t_{BUF}$ | 4.7 | – | 4.7 | – | 1.3 | – | μs |
| Hold time Note 1 | | $t_{HD:STA}$ | 4.0 | – | 4.0 | – | 0.6 | – | μs |
| Start/restart condition setup time | | $t_{SU:STA}$ | 4.7 | – | 4.7 | – | 0.6 | – | μs |
| Stop condition setup time | | $t_{SU:STO}$ | 4.0 | – | 4.0 | – | 0.6 | – | μs |
| Data hold time | CBUS-compatible master | $t_{HD:DAT}$ | – | | 500 | – | – | – | μs |
| | SMB/IIC | | 300 | | – | – | 0 Note 2 | 900 Note 3 | ns |
| Data setup time | | $t_{SU:DAT}$ | 250 | – | 250 | – | 100 Note 4 | – | ns |
| SCL0 clock low level width | | $t_{LOW}$ | 4.7 | – | 4.7 | – | 1.3 | – | μs |
| SCL0 clock high level width | | $t_{HIGH}$ | 4.0 | 50 | 4.0 | – | 0.6 | | μs |
| SCL0 and SDA0 signal falling time | | $t_F$ | – | 300 | – | 300 | – | 300 | ns |
| SCL0 and SDA0 signal rising time | | $t_R$ | – | 1,000 | – | 1,000 | – | 300 | ns |
| Pulse width of spikes controlled by the input filter | | $t_{SP}$ | | | | | 0 | 50 | ns |
| Time-out time | | $t_{TIMEOUT}$ | 25 | 35 | – | – | – | – | ms |
| SCL0 clock low level period total extension time (slave) | | $t_{LOW:SEXT}$ | – | 25 | – | – | – | – | ms |
| Cumulative SCL0 clock low level period total extension time (master) | | $t_{LOW:MEXT}$ | – | 10 | – | – | – | – | ms |
| Capacitive load of each bus line | | $C_b$ | – | – | – | 400 | – | 400 | pF |

**Notes 1.** In the start condition, the first clock pulse is generated after this period of time.

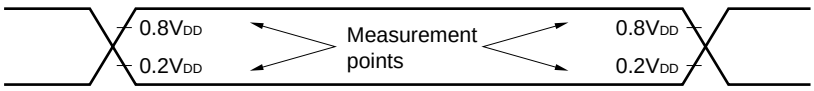
**2.** To fill the undefined area of the SCL0 falling edge (at $V_{IHmin.}$ of the SCL0 signal), the device needs to internally provide a hold time of at least 300 ns for the SDA0 signal.

**3.** If the device does not extend the low hold time ($t_{LOW}$) of the SCL0 signal, the maximum data hold time ($t_{HD:DAT}$) only needs to be satisfied.

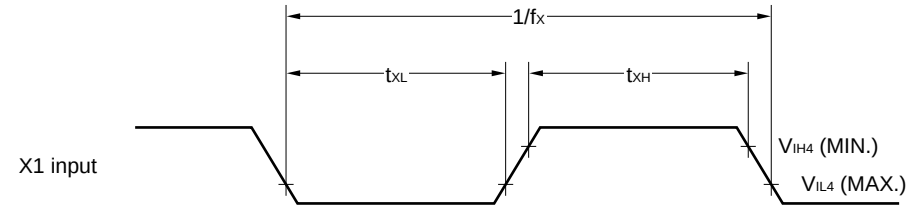
**4.** I²C high-speed mode can be used in SMB mode and I²C standard mode. In this case, the following conditions must be satisfied:

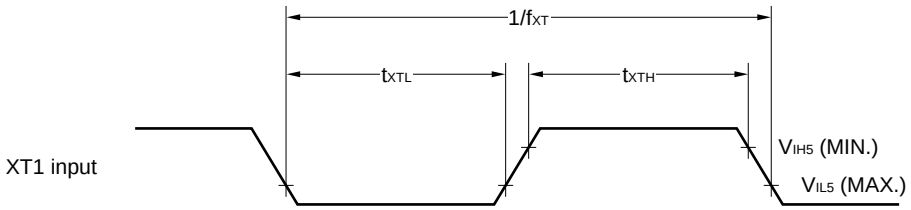
- When the device does not extend the low hold time of the SCL0 signal
  $t_{SU:DAT} \geq 250$ ns
- When the device extends the low hold time of the SCL0 signal
  Before SCL0 is released ($t_{Rmax.} + t_{SU:DAT} = 1{,}000 + 250 = 1{,}250$ ns: in SMB mode or I²C standard mode), the next data bit must be sent onto the SDA0 line.

**AC TIMING MEASUREMENT POINTS (except the X1 and XT1 inputs)**

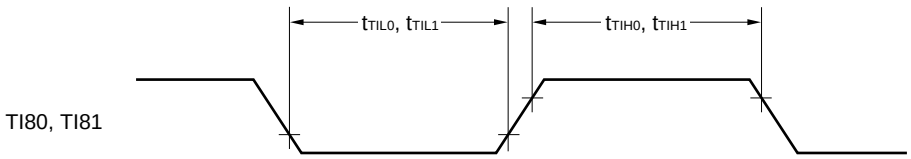



**CLOCK TIMING**

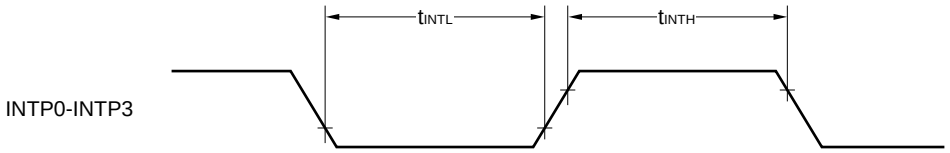


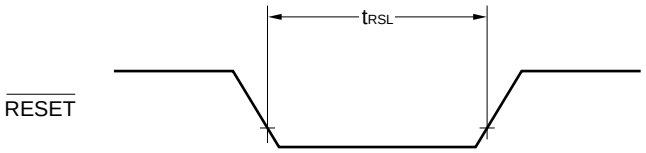

**TI TIMING**



**INTERRUPT INPUT TIMING**

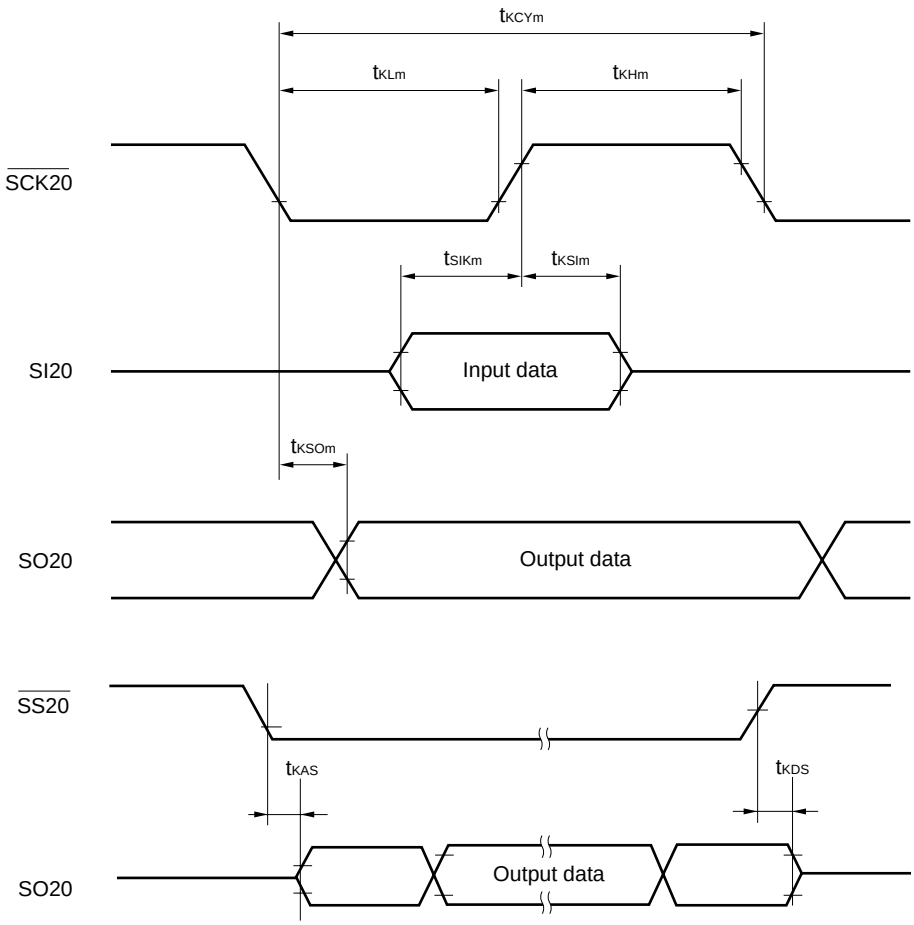



**$\overline{\text{RESET}}$ INPUT TIMING**



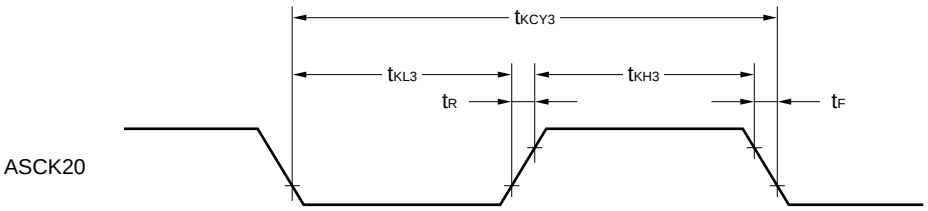
**SERIAL TRANSFER TIMING**

**Three-Wire Serial I/O Mode:**

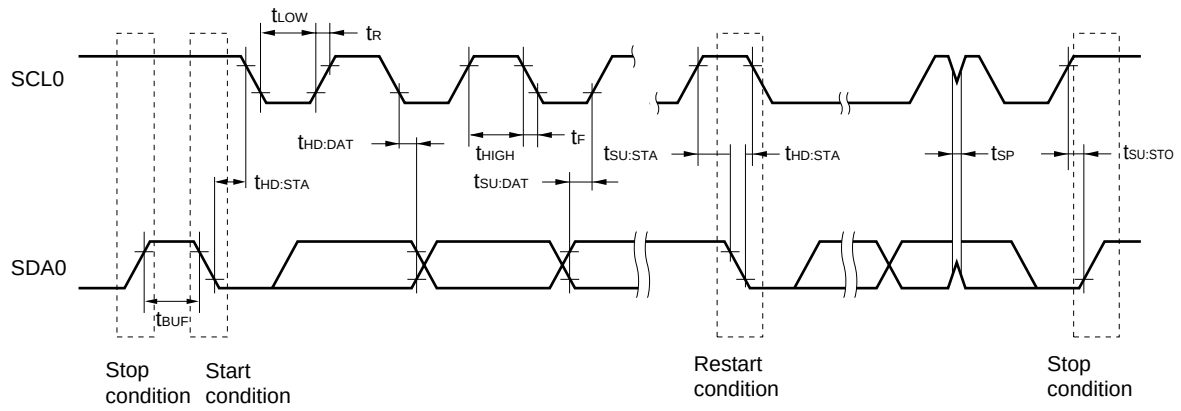



m = 1, 2

**UART Mode (External Clock Input):**



**SMB Mode:**


tLOW
tR
SCL0
tHD:DAT
tHIGH
tF
tSU:STA
tHD:STA
tSP
tSU:STO
tHD:STA
tSU:DAT
SDA0
tBUF
Stop condition
Start condition
Restart condition
Stop condition

**8-BIT A/D CONVERTER CHARACTERISTICS ($T_A$ = –40°C to +85°C, $AV_{DD}$ = $V_{DD}$ = 1.8 to 5.5 V, $AV_{SS}$ = $V_{SS}$ = 0 V)**

(For μPD789166Y and μPD789167Y only)

| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Resolution | | | 8 | 8 | 8 | bit |
| Total error[Note] | | 4.5 V ≤ $AV_{DD}$ ≤ 5.5 V | | 0.5 | 1 | LSB |
| | | 2.7 V ≤ $AV_{DD}$ < 4.5 V | | 1 | 1.75 | LSB |
| | | 1.8 V ≤ $AV_{DD}$ < 2.7 V | | Undefined | Undefined | LSB |
| Conversion time | $t_{CONV}$ | 4.5 V ≤ $AV_{DD}$ ≤ 5.5 V | Undefined | | Undefined | μs |
| | | 2.7 V ≤ $AV_{DD}$ < 4.5 V | Undefined | | Undefined | μs |
| | | 1.8 V ≤ $AV_{DD}$ < 2.7 V | Undefined | | Undefined | μs |
| Analog input voltage | $V_{IAN}$ | | 0 | | $AV_{REF}$ + 0.3 | V |
| Reference voltage | $AV_{REF}$ | | 0 | | $AV_{DD}$ + 0.3 | V |

**Note** No quantization error (±1/2 LSB) is included.

**10-BIT A/D CONVERTER CHARACTERISTICS ($T_A$ = –40°C to +85°C, $AV_{DD}$ = $V_{DD}$ = 1.8 to 5.5 V, $AV_{SS}$ = $V_{SS}$ = 0 V)**

(For μPD789176Y and μPD789177Y only)

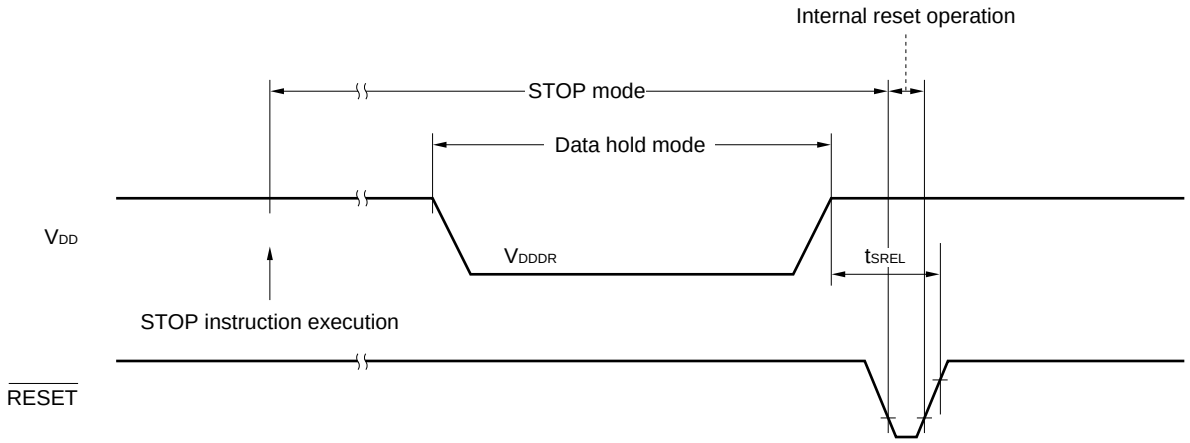
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Resolution | | | 10 | 10 | 10 | bit |
| Total error[Note] | | 4.5 V ≤ $AV_{DD}$ ≤ 5.5 V | | 2 | 4 | LSB |
| | | 2.7 V ≤ $AV_{DD}$ < 4.5 V | | 4 | 7 | LSB |
| | | 1.8 V ≤ $AV_{DD}$ < 2.7 V | | Undefined | Undefined | LSB |
| Conversion time | $t_{CONV}$ | 4.5 V ≤ $AV_{DD}$ ≤ 5.5 V | Undefined | | Undefined | μs |
| | | 2.7 V ≤ $AV_{DD}$ < 4.5 V | Undefined | | Undefined | μs |
| | | 1.8 V ≤ $AV_{DD}$ < 2.7 V | Undefined | | Undefined | μs |
| Analog input voltage | $V_{IAN}$ | | 0 | | $AV_{REF}$ + 0.3 | V |
| Reference voltage | $AV_{REF}$ | | 0 | | $AV_{DD}$ + 0.3 | V |

**Note** No quantization error (±1/2 LSB) is included.

**DATA MEMORY STOP MODE LOW POWER SUPPLY VOLTAGE DATA HOLD CHARACTERISTICS ($T_A$ = –40°C to +85°C)**

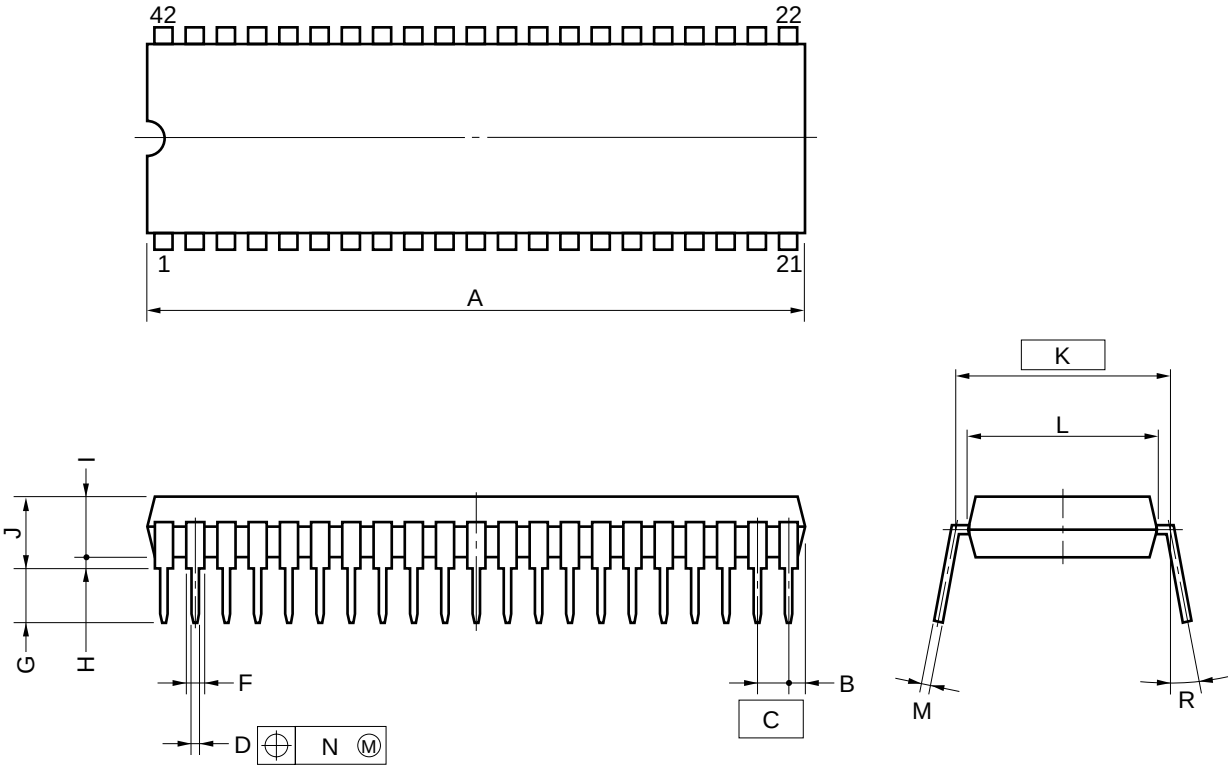
| Parameter | Symbol | Conditions | MIN. | TYP. | MAX. | Unit |
|---|---|---|---|---|---|---|
| Data hold supply voltage | $V_{DDDR}$ | | 1.8 | | 5.5 | V |
| Release signal set time | $t_{SREL}$ | | 0 | | | μs |

**DATA HOLD TIMING (STOP mode release by $\overline{\text{RESET}}$)**



## 12. PACKAGE DRAWINGS

### 42PIN PLASTIC SHRINK DIP (600 mil)




**NOTES**

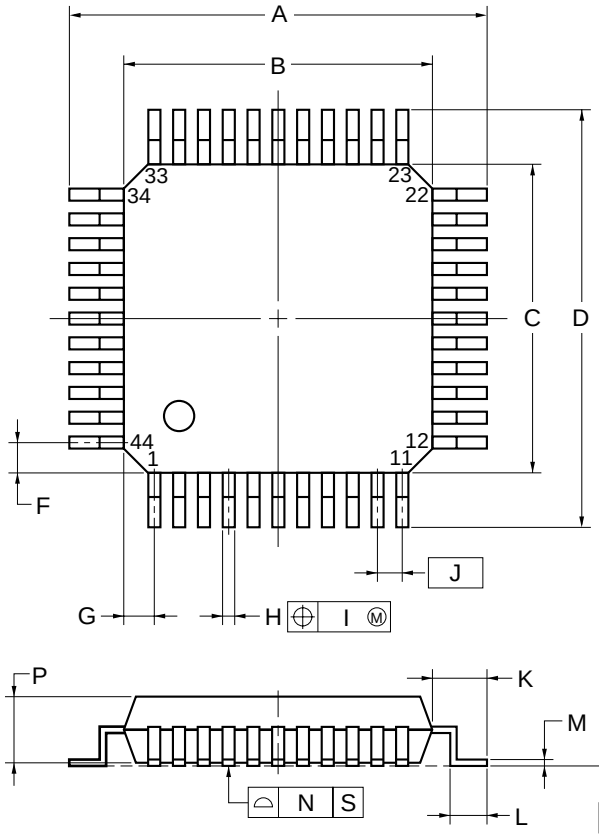
1) Each lead centerline is located within 0.17 mm (0.007 inch) of its true position (T.P.) at maximum material condition.

2) Item "K" to center of leads when formed parallel.

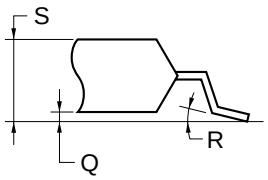
| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 39.13 MAX. | 1.541 MAX. |
| B | 1.78 MAX. | 0.070 MAX. |
| C | 1.778 (T.P.) | 0.070 (T.P.) |
| D | $0.50\pm0.10$ | $0.020^{+0.004}_{-0.005}$ |
| F | 0.9 MIN. | 0.035 MIN. |
| G | $3.2\pm0.3$ | $0.126\pm0.012$ |
| H | 0.51 MIN. | 0.020 MIN. |
| I | 4.31 MAX. | 0.170 MAX. |
| J | 5.08 MAX. | 0.200 MAX. |
| K | 15.24 (T.P.) | 0.600 (T.P.) |
| L | 13.2 | 0.520 |
| M | $0.25^{+0.10}_{-0.05}$ | $0.010^{+0.004}_{-0.003}$ |
| N | 0.17 | 0.007 |
| R | 0~15° | 0~15° |

P42C-70-600A-1

## 44 PIN PLASTIC QFP (□10)




detail of lead end




**NOTE**

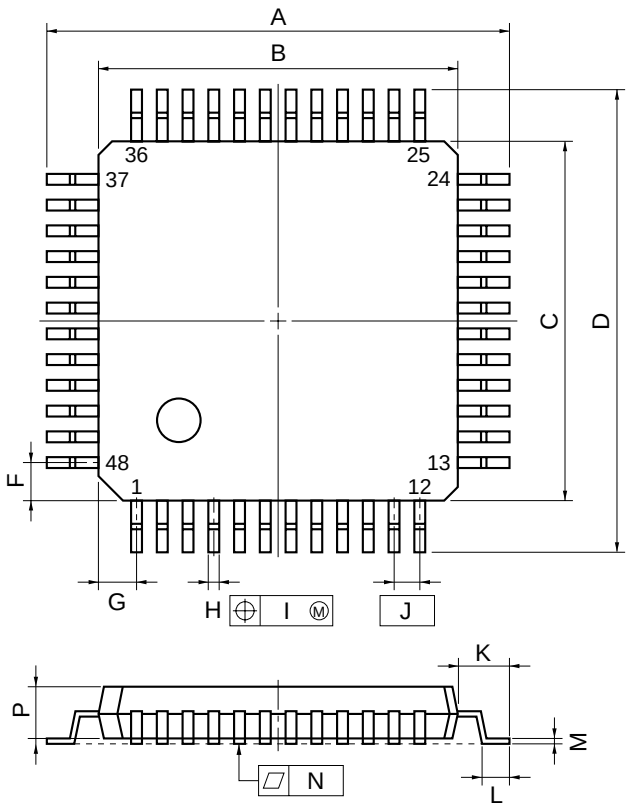
1. Controlling dimension —— millimeter.

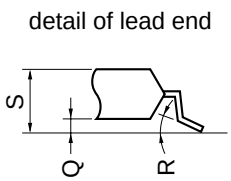
2. Each lead centerline is located within 0.16 mm (0.007 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 13.2±0.2 | $0.520^{+0.008}_{-0.009}$ |
| B | 10.0±0.2 | $0.394^{+0.008}_{-0.009}$ |
| C | 10.0±0.2 | $0.394^{+0.008}_{-0.009}$ |
| D | 13.2±0.2 | $0.520^{+0.008}_{-0.009}$ |
| F | 1.0 | 0.039 |
| G | 1.0 | 0.039 |
| H | $0.37^{+0.08}_{-0.07}$ | $0.015^{+0.003}_{-0.004}$ |
| I | 0.16 | 0.007 |
| J | 0.8 (T.P.) | 0.031 (T.P.) |
| K | 1.6±0.2 | 0.063±0.008 |
| L | 0.8±0.2 | $0.031^{+0.009}_{-0.008}$ |
| M | $0.17^{+0.06}_{-0.05}$ | $0.007^{+0.002}_{-0.003}$ |
| N | 0.10 | 0.004 |
| P | 2.7±0.1 | $0.106^{+0.005}_{-0.004}$ |
| Q | 0.125±0.075 | 0.005±0.003 |
| R | $3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$ | $3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$ |
| S | 3.0 MAX. | 0.119 MAX. |

S44GB-80-3BS-1

## 48 PIN PLASTIC TQFP (FINE PITCH) (□7)




**NOTE**

Each lead centerline is located within 0.10 mm (0.004 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|---|---|---|
| A | 9.0±0.2 | $0.354^{+0.009}_{-0.008}$ |
| B | 7.0±0.2 | $0.276^{+0.008}_{-0.009}$ |
| C | 7.0±0.2 | $0.276^{+0.008}_{-0.009}$ |
| D | 9.0±0.2 | $0.354^{+0.009}_{-0.008}$ |
| F | 0.75 | 0.030 |
| G | 0.75 | 0.030 |
| H | $0.22^{+0.05}_{-0.04}$ | 0.009±0.002 |
| I | 0.10 | 0.004 |
| J | 0.5 (T.P.) | 0.020 (T.P.) |
| K | 1.0±0.2 | $0.039^{+0.009}_{-0.008}$ |
| L | 0.5±0.2 | $0.020^{+0.008}_{-0.009}$ |
| M | $0.145^{+0.055}_{-0.045}$ | 0.006±0.002 |
| N | 0.10 | 0.004 |
| P | 1.0±0.1 | $0.039^{+0.005}_{-0.004}$ |
| Q | 0.1±0.05 | 0.004±0.002 |
| R | $3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$ | $3^{\circ}{}^{+7^{\circ}}_{-3^{\circ}}$ |
| S | 1.27 MAX. | 0.050 MAX. |

**S48GA-50-9EU-1**

## APPENDIX A DEVELOPMENT TOOLS

The following development tools are available for developing systems using the μPD789166Y, μPD789167Y, μPD789176Y, and μPD789177Y.

### LANGUAGE PROCESSING SOFTWARE

| | |
|---|---|
| RA78K0S Notes 1, 2, 3 | Assembler package common to the 78K/0S series |
| CC78K0S Notes 1, 2, 3 | C compiler package common to the 78K/0S series |
| DF789177 Notes 1, 2, 3, 5 | Device file for the μPD789167Y and μPD789177Y sub-series |
| CC78K0S-L Notes 1, 2, 3, 5 | C compiler library source file common to the 78K/0S series |

### FLASH MEMORY WRITE TOOLS

| | |
|---|---|
| Flashpro II Note 4 | Dedicated flash writer (formerly, Flashpro) |
| FA-42CU Note 4 | Flash memory write adapter |
| FA-44GB Note 4 | |
| FA-48GA Notes 4, 5 | |

### DEBUGGING TOOLS

| | |
|---|---|
| IE-78K0S-NS<br>In-circuit emulator | This in-circuit emulator is used to debug hardware or software when application systems which use the 78K/0S series are developed. The IE-78K0S-NS supports the integrated debugger (ID78K0S-NS). The IE-78K0S-NS is used in combination with an interface adapter for connection to an AC adapter, emulation probe, or host machine. |
| IE-70000-MC-PS-B<br>AC adapter | This adapter is used to supply power from a 100-VAC outlet. |
| IE-70000-98-IF-C<br>Interface adapter | This adapter is required when a PC-9800 series computer (other than a notebook type) is used as the host machine for the IE-78K0S-NS. |
| IE-70000-CD-IF<br>PC card/interface | These PC card and interface cable are required when a PC-9800 series computer is used as the host machine for the IE-78K0S-NS. |
| IE-70000-PC-IF-C<br>Interface adapter | This adapter is required when an IBM PC/AT or compatible is used as the host machine for the IE-78K0S-NS. |
| IE-789177-NS-EM1 Note 5<br>Emulation board | This board is used to emulate the peripheral hardware specific to the device. The IE-789198-NS-EM1 is used in combination with the in-circuit emulator. |
| NP-42CU Note 4<br>Emulation probe | This probe is used to connect an in-circuit emulator to the target system. The probe is dedicated to the 42-pin plastic shrink DIP. |
| NP-44GB Note 4<br>Emulation probe | This probe is used to connect an in-circuit emulator to the target system. The probe is dedicated to the 44-pin plastic QFP. |
| NP-48GA Notes 4, 5<br>Emulation probe | This probe is used to connect an in-circuit emulator to the target system. The probe is dedicated to the 48-pin plastic TQFP. |
| SM78K0S Notes 1, 2 | System simulator common to all 78K/0S series units |
| DF789177 Notes 1, 2, 5 | Device file for the μPD789167Y and μPD789177Y sub-series |

**REAL-TIME OS**

| | |
|---|---|
| MX78K0S Notes 1, 2, 5 | OS for the 78K/0S series |

**Notes 1.** Based on the PC-9800 series (MS-DOS™ + Windows™)

**2.** Based on the IBM PC/AT™ and compatibles (PC DOS™/IBM DOS™/MS-DOS + Windows)

**3.** Based on the HP9000 series 700™ (HP-UX™), SPARCstation™ (SunOS™), and NEWS™ (NEWS-OS™)

**4.** Product manufactured by and available from Naito Densei Machida Mfg. Co., Ltd. (044-822-3813).

**5.** Under development

**Remark** The RA78K0S, CC78K0S, and SM78K0S can be used in combination with the DF789177.

## APPENDIX B RELATED DOCUMENTS

### DOCUMENTS RELATED TO DEVICES

| Document name | Document No. | |
|---|---|---|
| | Japanese | English |
| μPD789166Y, 789167Y, 789176Y, 789177Y Preliminary Product Information | U13216J | This manual |
| μPD78F9177Y Preliminary Product Information | U13210J | To be created |
| μPD789177Y Sub-Series User's Manual | To be created | To be created |
| 78K/0 Series User's Manual, Instruction | U11047J | U11047E |
| 78K/0S Series Instruction Summary Sheet | To be created | – |
| 78K/0S Series Instruction Set | To be created | – |

### DOCUMENTS RELATED TO DEVELOPMENT TOOLS (USER'S MANUAL)

| Document name | | Document No. | |
|---|---|---|---|
| | | Japanese | English |
| RA78K0S Assembler Package | Operation | U11622J | U11622E |
| | Language | U11599J | U11599E |
| | Structured Assembly Language | U11623J | U11623E |
| CC78K/0S C Compiler | Operation | U11816J | U11816E |
| | Language | U11817J | U11817E |
| SM78K0S System Simulator Windows Base | Reference | U11489J | U11489E |
| SM78K Series System Simulator | External Parts User Open Interface Specifications | U10092J | U10092E |
| ID78K0S-NS Windows Base | Reference | U12901J | To be created |

### DOCUMENTS RELATED TO SOFTWARE TO BE INCORPORATED INTO THE PRODUCT (USER'S MANUAL)

| Document name | | Document No. | |
|---|---|---|---|
| | | Japanese | English |
| OS for 78K/0S Series MX78K0S | Fundamental | U12938J | U12938E |

**Caution The above documents may be revised without notice. Use the latest versions when you design application systems.**

**OTHER DOCUMENTS**

| Document name | Document No. | |
|---|---|---|
| | Japanese | English |
| IC PACKAGE MANUAL | C10943X | |
| Semiconductor Device Mounting Technology Manual | C10535J | C10535E |
| Quality Grades on NEC Semiconductor Device | C11531J | C11531E |
| NEC Semiconductor Device Reliability/Quality Control System | C10983J | C10983E |
| Guide to Prevent Damage for Semiconductor Devices by Electrostatic Discharge (ESD) | C11892J | C11892E |
| Semiconductor Device Quality Control/Reliability Handbook | C12769J | – |
| Guide for Products Related to Microcomputer: Other Companies | U11416J | – |

**Caution The above documents may be revised without notice. Use the latest versions when you design application systems.**

## NOTES FOR CMOS DEVICES

### ① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

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

### ② HANDLING OF UNUSED INPUT PINS FOR CMOS

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

### ③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

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

**EEPROM is a trademark of NEC Corporation.**

**MS-DOS and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.**

**IBM DOS, PC/AT, and PC DOS are trademarks of IBM Corporation.**

**HP9000 series 700 and HP-UX are trademarks of Hewlett-Packard Company.**

**SPARCstation is a trademark of SPARC International, Inc.**

**SunOS is a trademark of Sun Microsystems, Inc.**

**NEWS and NEWS-OS are trademarks of SONY Corporation.**

# Regional Information

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

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

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

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

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

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

**NEC Electronics Italiana s.r.1.**
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

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

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

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

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

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

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

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

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

**NEC do Brasil S.A.**
Cumbica-Guarulhos-SP, Brasil
Tel: 011-6465-6810
Fax: 011-6465-6829

J98. 2

[MEMO]

The export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

**Caution** This product contains an $I^2C$ bus interface circuit.
When using the $I^2C$ bus interface, notify its use to NEC when ordering custom code. NEC can guarantee the following only when the customer informs NEC of the use of the interface:
Purchase of NEC $I^2C$ components conveys a license under the Philips $I^2C$ Patent Rights to use these components in an $I^2C$ system, provided that the system conforms to the $I^2C$ Standard Specification as defined by Philips.



M4 96.5