

16/8-BIT SINGLE-CHIP MICROCONTROLLERS

DESCRIPTION

μ PD78366A is provided with a high-speed, high-performance CPU and powerful operation functions. Unlike the existing μ PD78328, μ PD78366A is also provided with a high-resolution PWM signal output function which substantially contributes to improving the performance of the inverter control.

A PROM model, μ PD78P368A, is also available.

Detailed functions, etc. are described in the following user's manual. Be sure to read the manual to design systems.

μ PD78366A User's Manual Hardware: U10205E

μ PD78356 User's Manual : U12117E

FEATURES

- Internal 16-bit architecture, external 8-bit data bus
- High-speed processing by pipeline control method and high-speed operating clock
 - Minimum instruction execution time: 125 ns (internal clock: at 16 MHz, external clock: 8 MHz)
- Real-time pulse unit for inverter control
- 10-bit resolution A/D converter: 8 channels
- 8-/9-/10-/12-bit resolution variable PWM signal output function: 2 channels
- Powerful serial interface: 2 channels
- Internal memory:

ROM: none (μ PD78365A)

24K bytes (μ PD78363A)

32K bytes (μ PD78366A)

48K bytes (μ PD78368A)

RAM: 768 bytes (μ PD78363A)

2K bytes (μ PD78365A, 78366A, 78368A)

APPLICATION EXAMPLES

- Inverter air conditioner
- Factory automation fields, such as industrial robots and machine tools.

ORDERING INFORMATION

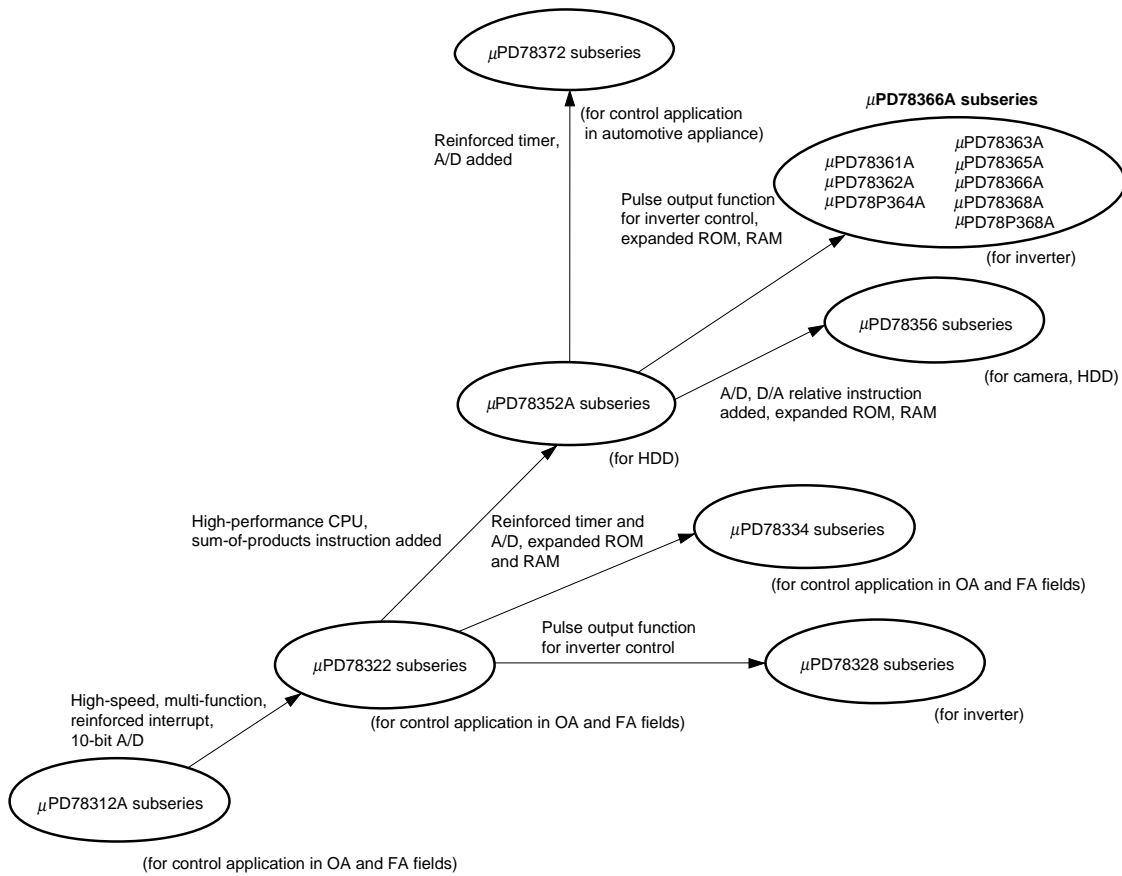
| Part Number | Package | Internal ROM |
|----------------------------|---------------------------------|--------------|
| μ PD78363AGF-xxx-3B9 | 80-pin plastic QFP (14 × 20 mm) | Mask ROM |
| μ PD78365AGF-3B9 | 80-pin plastic QFP (14 × 20 mm) | None |
| μ PD78366AGF-xxx-3B9 | 80-pin plastic QFP (14 × 20 mm) | Mask ROM |
| ★ μ PD78368AGF-xxx-3B9 | 80-pin plastic QFP (14 × 20 mm) | Mask ROM |

Remark xxx indicates a ROM code suffix.

Unless otherwise specified, the functions and performances of the μ PD78366 are described throughout this document.

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

78K/III Series Product Development

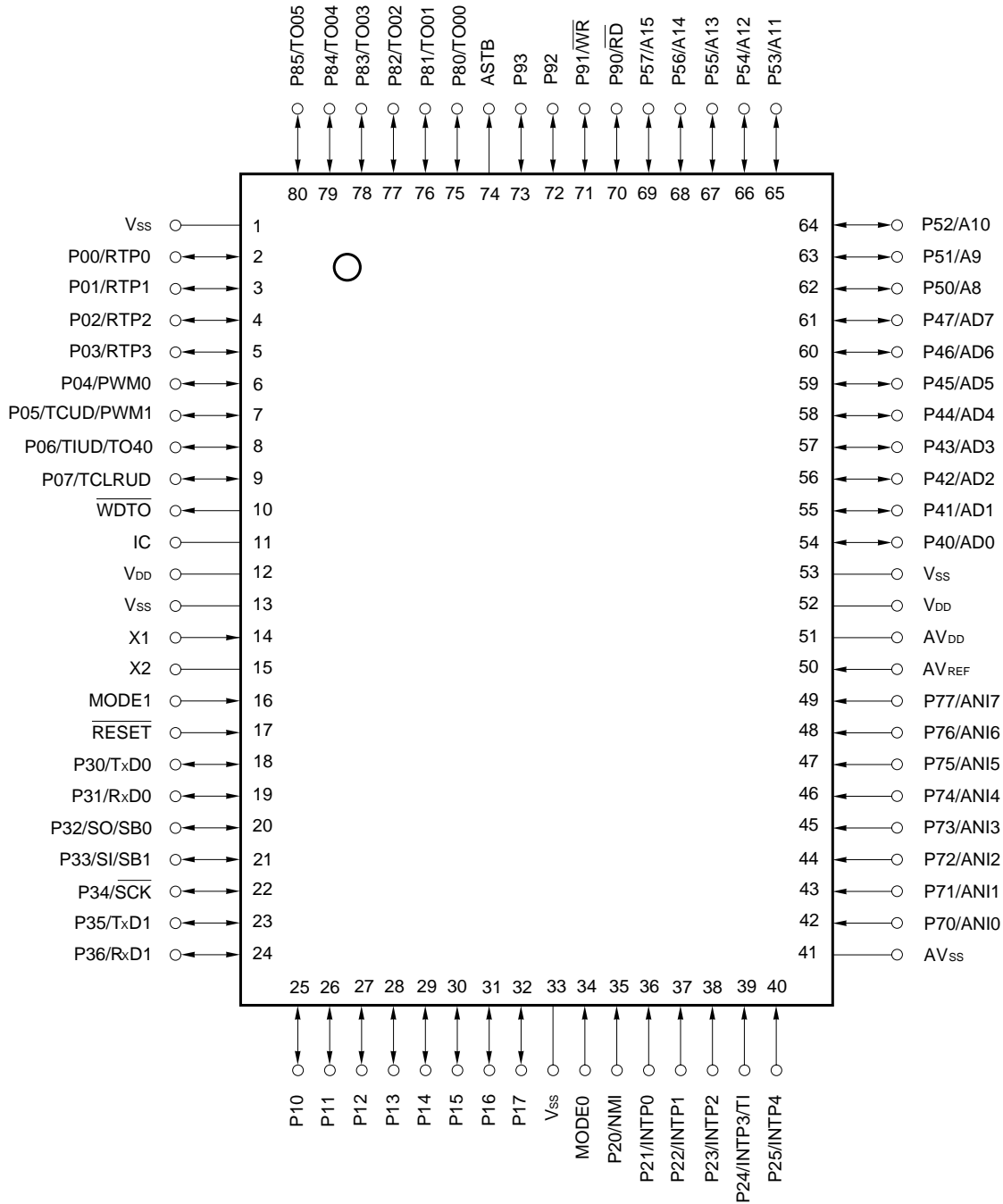


PIN CONFIGURATION (TOP VIEW)

- 80-pin plastic QFP (14 × 20 mm)



μPD78363AGF-xxx-3B9, 78365AGF-3B9, 78366AGF-xxx-3B9, 78368AGF-xxx-3B9



Caution Connect the IC pin directly to Vss.

Remark xxx indicates a ROM code suffix

| | |
|---------------------------|---------------------------------|
| P00-P07 | : Port0 |
| P10-P17 | : Port1 |
| P20-P25 | : Port2 |
| P30-P36 | : Port3 |
| P40-P47 | : Port4 |
| P50-P57 | : Port5 |
| P70-P77 | : Port7 |
| P80-P85 | : Port8 |
| P90-P93 | : Port9 |
| RTP0-RTP3 | : Real-time Port |
| NMI | : Nonmaskable Interrupt |
| INTP0-INTP4 | : Interrupt From Peripherals |
| TO00-TO05, TO04 | : Timer Output |
| TI | : Timer Input |
| TIUD | : Timer Input Up Down Counter |
| TCUD | : Timer Control Up Down Counter |
| TCLRUD | : Timer Clear Up Down Counter |
| ANI0-ANI7 | : Analog Input |
| TxD0, TxD1 | : Transmit Data |
| RxD0, RxD1 | : Receive Data |
| SI | : Serial Input |
| SO | : Serial Output |
| SB0, SB1 | : Serial Bus |
| $\overline{\text{SCK}}$ | : Serial Clock |
| PWM0, PWM1 | : Pulse Width Modulation Output |
| WDTO | : Watchdog Timer Output |
| MODE0, MODE1 | : Mode |
| AD0-AD7 | : Address/Data Bus |
| A8-A15 | : Address Bus |
| ASTB | : Address Strobe |
| $\overline{\text{RD}}$ | : Read Strobe |
| $\overline{\text{WR}}$ | : Write Strobe |
| $\overline{\text{RESET}}$ | : Reset |
| X1, X2 | : Crystal |
| AV _{DD} | : Analog V _{DD} |
| AV _{SS} | : Analog V _{SS} |
| AV _{REF} | : Analog Reference Voltage |
| V _{DD} | : Power Supply |
| V _{SS} | : Ground |
| IC | : Internally Connected |

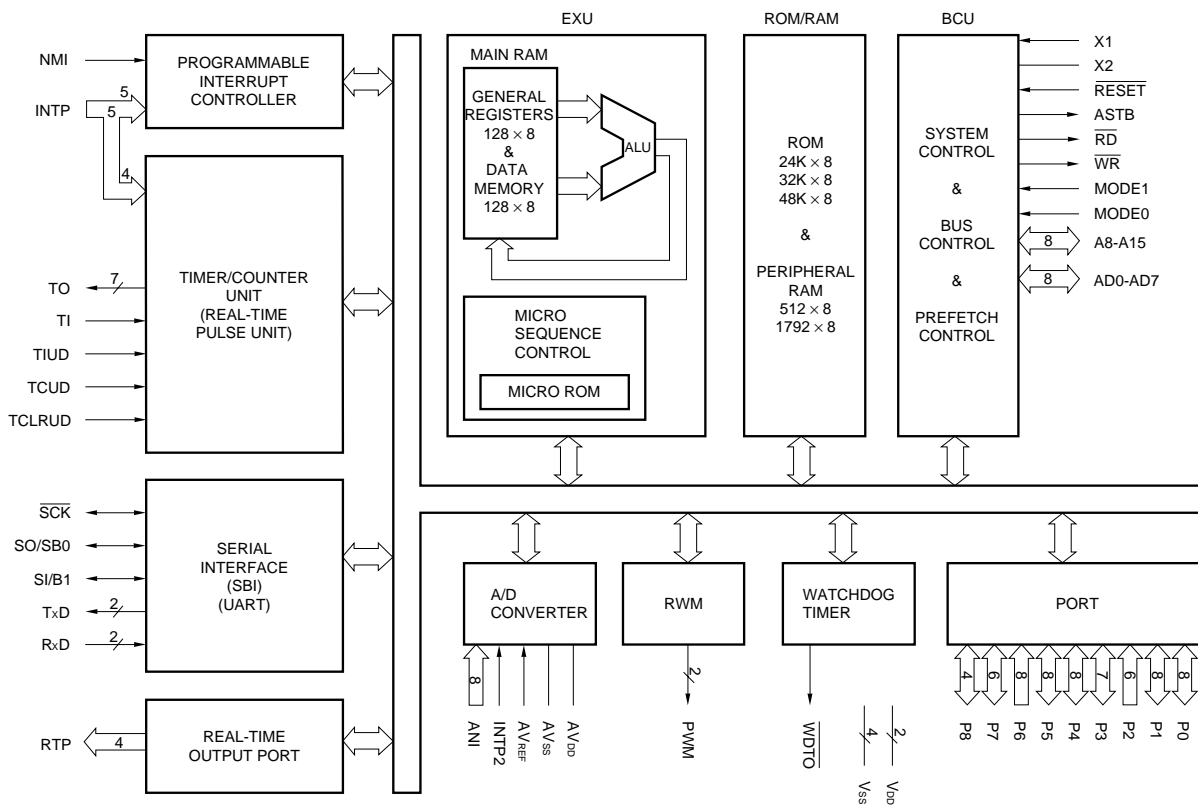
FUNCTIONAL OUTLINE

| Item | | Product name | | | |
|------------------------------------|-------|---|-----------|-----------|-----------|
| | | μPD78363A | μPD78365A | μPD78366A | μPD78368A |
| Minimum instruction execution time | | 125 ns (internal clock: 16 MHz, external clock: 8 MHz) | | | |
| Internal memory | ROM | 24K bytes | None | 32K bytes | 48K bytes |
| | RAM | 768 bytes | 2K bytes | | |
| Memory space | | 64K bytes (externally expandable) | | | |
| General-purpose registers | | 8 bits × 16 × 8 banks | | | |
| Number of basic instructions | | 115 | | | |
| Instruction set | | <ul style="list-style-type: none"> • 16-bit transfer/operation • Multiplication/division (16 bits × 16 bits, 32 bits ÷ 16 bits) • Bit manipulation • String • Sum-of-products operation (16 bits × 16 bits + 32 bits) • Relative operation | | | |
| I/O lines | Input | 14 (of which 8 are shared with analog input) | | | |
| | I/O | 49 | 31 | 49 | |
| Real-time pulse unit | | <ul style="list-style-type: none"> • 16-bit timer × 1 <ul style="list-style-type: none"> 10-bit dead time timer × 3 16-bit compare register × 4 2 kinds of output mode can be selected <ul style="list-style-type: none"> Mode 0, set-reset output: 6 channels Mode 1, buffer output: 6 channels • 16-bit timer × 1 <ul style="list-style-type: none"> 16-bit compare register × 1 • 16-bit timer × 1 <ul style="list-style-type: none"> 16-bit capture register × 1 16-bit capture/compare register × 1 • 16-bit timer × 1 <ul style="list-style-type: none"> 16-bit capture register × 2 16-bit capture/compare register × 1 • 16-bit timer × 1 <ul style="list-style-type: none"> 16-bit compare register × 2 16-bit resolution PWM output: 1 channel | | | |
| Real-time output port | | Pulse outputs associated with real-time pulse unit: 4 lines | | | |
| PWM unit | | 8-/9-/10-/12-bit resolution variable PWM output: 2 channels | | | |
| A/D converter | | 10-bit resolution, 8 channels | | | |
| Serial interface | | Dedicated baud rate generator UART (w/pin selection function): 1 channel Clocked serial interface/SBI: 1 channel | | | |
| Interrupt function | | <ul style="list-style-type: none"> • External: 6, internal: 14 (of which 2 are multiplexed with external) • 4 priority levels can be specified through software • 3 types of interrupt processing modes selectable (vectored interrupt, macro service, and context switching) | | | |
| Package | | 80-pin plastic QFP (14 × 20 mm) | | | |
| Others | | <ul style="list-style-type: none"> • Watchdog timer • Standby function (HALT and STOP modes) | | | |

DIFFERENCES BETWEEN μPD78363A, 78365A, 78366A, AND 78368A

| Product name | | μPD78363A | μPD78366A | μPD78368A | μPD78365A |
|-------------------------------------|---|---|-----------|---|-----------|
| Internal ROM | ROM | 24K bytes | 32K bytes | 48K bytes | None |
| | RAM | 786 bytes | 2K bytes | | |
| I/O lines | Input | 14 (of which 8 are multiplexed with analog input) | | | |
| | I/O | 49 | | | 31 |
| Port 4 (P40-P47) | Can be set in input or output mode in units of 8 bits. In external memory expansion mode, this port functions as multiplexed address/data bus (AD0-AD7). | | | Always functions as multiplexed address/data bus (AD0-AD7). | |
| Port 5 (P50-P57) | Can be set in input or output mode in 1-bit units. In external memory expansion mode, this port functions as address bus (A8-A15). | | | Always functions as address bus (A8-A15) | |
| Port 9 (P90-P93) | Can be set in input or output mode in 1-bit units. In external memory expansion mode, P90 outputs \overline{RD} strobe signal, and P91 outputs \overline{WR} strobe signal. | | | P90 always functions as \overline{RD} strobe signal output pin, and P91 always functions as \overline{WR} strobe signal output pin. P92 and P93 function as I/O port lines. | |
| Memory expansion mode register (MM) | Sets port 4 in input or output mode in units of 8 bits. In external memory expansion mode, sets memory expansion width of ports 4 and 5. | | | Always fixed to external memory expansion mode. | |
| Port 5 mode register (PM5) | Sets port 5 in input or output mode in 1-bit units. | | | None | |
| Setting of MODE0, MODE1 | <ul style="list-style-type: none"> In ordinary operation mode: MODE0, 1 = LL In ROM-less mode: MODE0, 1 = HH | | | <ul style="list-style-type: none"> Always set as follows: MODE0, 1 = HH | |

BLOCK DIAGRAM



Remark The internal ROM and RAM capacities differ depending on the product.

CONTENTS

| | |
|---|-----------|
| 1. PIN FUNCTIONS | 10 |
| 1.1 PORT PINS | 10 |
| 1.2 PINS OTHER THAN PORT PINS | 11 |
| 1.3 PIN I/O CIRCUITS AND PROCESSING OF UNUSED PINS | 13 |
| 2. CPU ARCHITECTURE | 15 |
| 2.1 MEMORY SPACE | 15 |
| 2.2 DATA MEMORY ADDRESSING | 18 |
| 2.3 PROCESSOR REGISTERS | 20 |
| 2.3.1 Control Registers | 21 |
| 2.3.2 General-Purpose Registers | 22 |
| 2.3.3 Special Function Registers (SFR) | 23 |
| 3. FUNCTIONAL BLOCKS | 29 |
| 3.1 EXECUTION UNIT (EXU) | 29 |
| 3.2 BUS CONTROL UNIT (BCU) | 29 |
| 3.3 ROM/RAM | 29 |
| 3.4 PORT FUNCTIONS | 30 |
| 3.5 CLOCK GENERATOR CIRCUIT | 32 |
| 3.6 REAL-TIME PULSE UNIT (RPU) | 34 |
| 3.7 REAL-TIME OUTPUT PORT (RTP) | 42 |
| 3.8 A/D CONVERTER | 43 |
| 3.9 SERIAL INTERFACE | 44 |
| 3.10 PWM UNIT | 46 |
| 3.11 WATCHDOG TIMER (WDT) | 47 |
| 4. INTERRUPT FUNCTIONS | 48 |
| 4.1 OUTLINE | 48 |
| 4.2 MACRO SERVICE | 49 |
| 4.3 CONTEXT SWITCHING | 52 |
| 4.3.1 Context Switching Function by Interrupt Request | 52 |
| 4.3.2 Context Switching Function by BRKCS Instruction | 53 |
| 4.3.3 Restoration from Context Switching | 53 |
| 5. EXTERNAL DEVICE EXPANSION FUNCTION | 54 |
| 6. STANDBY FUNCTIONS | 55 |
| 7. RESET FUNCTION | 56 |
| 8. INSTRUCTION SET | 57 |
| 9. EXAMPLE OF SYSTEM CONFIGURATION | 71 |
| 10. ELECTRICAL SPECIFICATIONS | 72 |

- 11. PACKAGE DRAWING 83
- 12. RECOMMENDED SOLDERING CONDITIONS 84
- APPENDIX A. DIFFERENCES BETWEEN μPD78366A AND μPD78328 85
- APPENDIX B. TOOLS 86
 - B.1 DEVELOPMENT TOOLS 86
 - B.2 EMBEDDED SOFTWARE 91

1. PIN FUNCTIONS

1.1 PORT PINS

| Pin name | I/O | Function | Shared by: |
|----------|-------|--|-----------------|
| P00-P03 | I/O | Port 0. 8-bit I/O port. Can be set in input or output mode in 1-bit units. | RTP0-RTP3 |
| P04 | | | PWM0 |
| P05 | | | TCUD/PWM1 |
| P06 | | | TIUD/TO40 |
| P07 | | | TCLRUD |
| P10-P17 | I/O | Port 1. 8-bit I/O port. Can be set in input or output mode in 1-bit units. | - |
| P20 | Input | Port 2. 6-bit input port. | NMI |
| P21 | | | INTP0 |
| P22 | | | INTP1 |
| P23 | | | INTP2 |
| P24 | | | INTP3/TI |
| P25 | | | INTP4 |
| P30 | I/O | Port 3. 7-bit I/O port. Can be set in input or output mode in 1-bit units. | TxD0 |
| P31 | | | RxD0 |
| P32 | | | SO/SB0 |
| P33 | | | SI/SB1 |
| P34 | | | SCK |
| P35 | | | TxD1 |
| P36 | | | RxD1 |
| P40-P47 | I/O | Port 4. 8-bit I/O Port. Can be set in input or output mode in 8-bit units. | AD0-AD7 |
| P50-P57 | I/O | Port 5. 8-bit I/O port. Can be set in input or output mode in 1-bit units. | A8-A15 |
| P70-P77 | Input | Port 7. 8-bit input port | ANI0-ANI7 |
| P80-P85 | I/O | Port 8. 6-bit I/O port. Can be set in input or output mode in 1-bit units. | TO00-TO05 |
| P90 | I/O | Port 9. 4-bit I/O port. Can be set in input or output mode in 1-bit units. | \overline{RD} |
| P91 | | | \overline{WR} |
| P92 | | | - |
| P93 | | | - |

1.2 PINS OTHER THAN PORT PINS (1/2)

| Pin name | I/O | Function | Shared by: |
|-----------|--------|--|------------|
| RTP0-RTP3 | Output | Real-time output port that outputs pulses in synchronization with trigger signal from real-time pulse unit. | P00-P03 |
| NMI | Input | Non-maskable interrupt request input. | P20 |
| INTP0 | | External interrupt request input. | P21 |
| INTP1 | | P22 | |
| INTP2 | | P23 | |
| INTP3 | | P24/TI | |
| INTP4 | | P25 | |
| TI | Input | External count clock input to timer 1. | P24/INTP3 |
| TCUD | | Count operation selection control signal input to up/down counter (timer 4). | P05/PWM1 |
| TIUD | | External count clock input to up/down counter (timer 4). | P06/TO40 |
| TCLRUD | | Clear signal input to up/down counter (timer 4). | P07 |
| TO00-TO05 | Output | Pulse output from real-time pulse unit. | P80-P85 |
| TO40 | | | P06/TIUD |
| ANI0-ANI7 | Input | Analog input to A/D converter. | P70-P77 |
| TxD0 | Output | Serial data output of asynchronous serial interface. | P30 |
| TxD1 | | | P35 |
| RxD0 | Input | Serial data input of asynchronous serial interface. | P31 |
| RxD1 | | | P36 |
| SCK | I/O | Serial clock input/output of clocked serial interface. | P34 |
| SI | Input | Serial data input of clocked serial interface in 3-line mode. | P33/SB1 |
| SO | Output | Serial data output of clocked serial interface in 3-line mode. | P32/SB0 |
| SB0 | I/O | Serial data input/output of clocked serial interface in SBI mode. | P32/SO |
| SB1 | | | P33/SI |
| PWM0 | Output | PWM signal output. | P04 |
| PWM1 | | | P05/TCUD |
| WDT0 | Output | Signal output indicating overflow of watchdog timer (generates non-maskable interrupt). | - |
| AD0-AD7 | I/O | Multiplexed address/data bus when memory is externally expanded. | P40-P47 |
| A8-A15 | | Address bus when memory is externally expanded. | P50-P57 |
| ASTB | Output | Outputs timing signal at which address information output from AD0-AD7 and A8-A15 pins to access external memory is to be latched. | - |
| RD | | Read strobe signal output to external memory. | P90 |
| WR | | Write strobe signal output to external memory. | P91 |

1.2 PINS OTHER THAN PORT PINS (2/2)

| Pin name | I/O | Function | Shared by: |
|--------------------|-------|---|------------|
| MODE0 | Input | Control signal input to set operation mode. With μ PD78363A, 78366A, and 78368A MODE0 and MODE1 are usually connected to V_{SS} . With μ PD78365A, MODE0 and MODE1 are always connected to V_{DD} . | - |
| MODE1 | | | |
| \overline{RESET} | Input | System reset input | - |
| X1 | Input | Crystal oscillator connecting pins for system clock. If a clock is externally supplied, input it to pin X1. Leave pin X2 open. | - |
| X2 | - | | |
| AV_{REF} | Input | A/D converter reference voltage input. | - |
| AV_{DD} | - | A/D converter analog power supply. | - |
| AV_{SS} | - | A/D converter GND. | - |
| V_{DD} | - | Positive power supply | - |
| V_{SS} | - | GND | - |
| IC | - | Internally connected. Connect this pin to V_{SS} . | - |

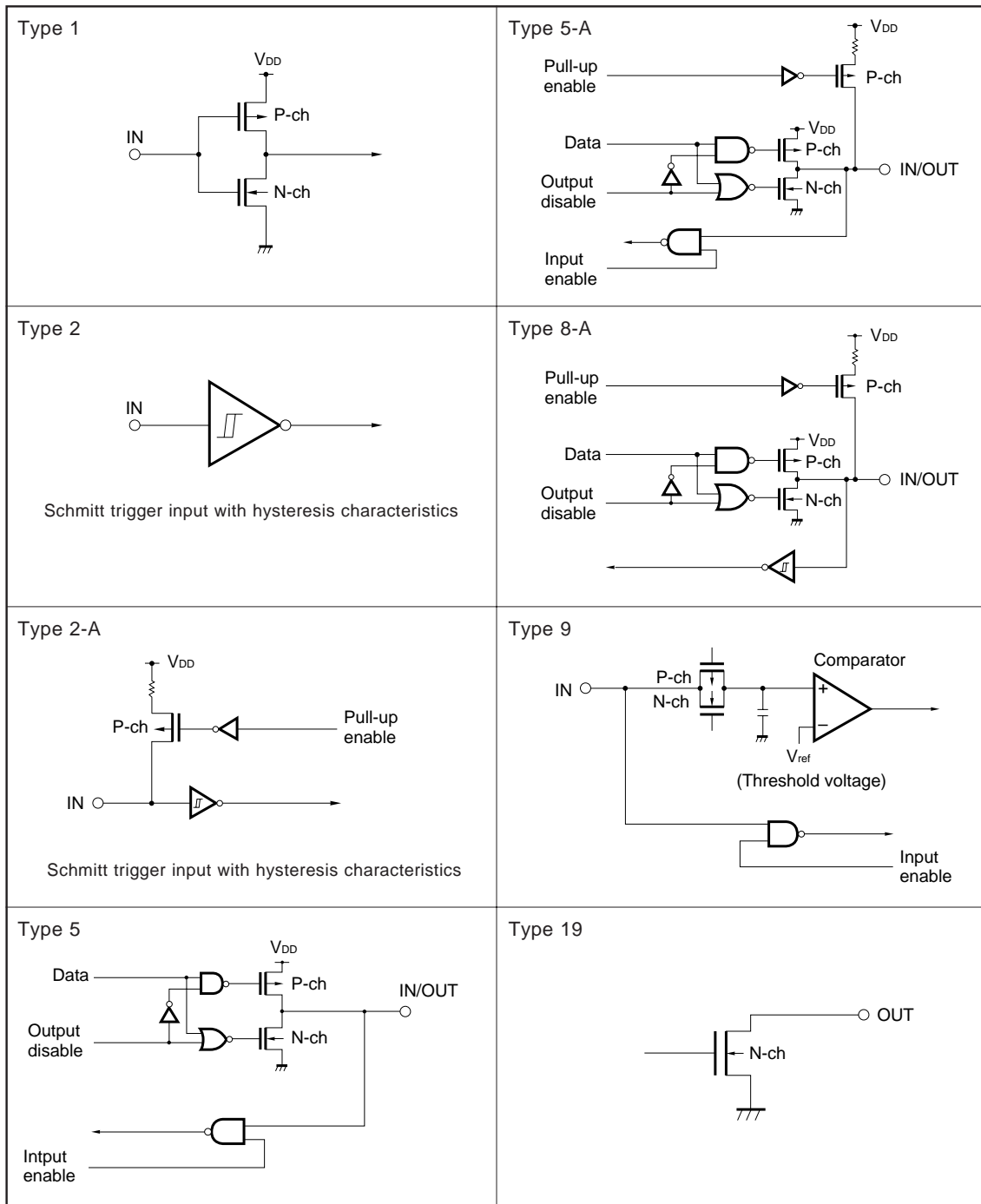
1.3 PIN I/O CIRCUITS AND PROCESSING OF UNUSED PINS

Table 1-1 shows the I/O circuit types of the respective pins, and recommended connections of the unused pins. Figure 1-1 shows the circuits of the respective pins.

Table 1-1. Pin I/O Circuit Type and Recommended Connections of Unused Pins

| Pin | I/O circuit type | Recommended connections | |
|--------------------------------------|------------------|--|----------------------------|
| P00/RTP0-P03/RTP3 | 5-A | Input : Independently connect to V _{DD} or V _{SS} through resistor Output : Leave unconnected | |
| P04/PWM0 | | | |
| P05/TCUD/PWM1 | | | |
| P06/TIUD/TO40 | | | |
| P07/TCLRUD | | | |
| P10-P17 | | | |
| P20/NMI | 2 | Connect to V _{SS} | |
| P21/INTP0 | 2-A | | |
| P22/INTP1 | | | |
| P23/INTP2 | | | |
| P24/INTP3/TI | | | |
| P25/INTP4 | | | |
| P30/TxD0 | 5-A | Input : Independently connect to V _{DD} or V _{SS} through resistor Output : Leave unconnected | |
| P31/RxD0 | | | |
| P32/SO/SB0 | 8-A | | |
| P33/SI/SB1 | | | |
| P34/SCK | | | |
| P35/TxD1 | 5-A | | |
| P36/RxD1 | | | |
| P40/AD0-P47/AD7 | | | |
| P50/A8-P57/A15 | | | |
| P70/ANI0-P77/ANI7 | | | 9 |
| P80/TO00-P85/TO05 | 5-A | Input : Independently connect to V _{DD} or V _{SS} through resistor Output : Leave unconnected | |
| P90/RD | | | |
| P91/W _R | | | |
| P92, P93 | | | |
| ASTB | 5 | - | |
| WDTO | 19 | | |
| MODE0, MODE1 | 1 | | |
| RESET | 2 | - | |
| AV _{REF} , AV _{SS} | - | | Connect to V _{SS} |
| AV _{DD} | | | Connect to V _{DD} |
| IC | | | Connect to V _{SS} |

Figure 1-1. Pin I/O Circuits



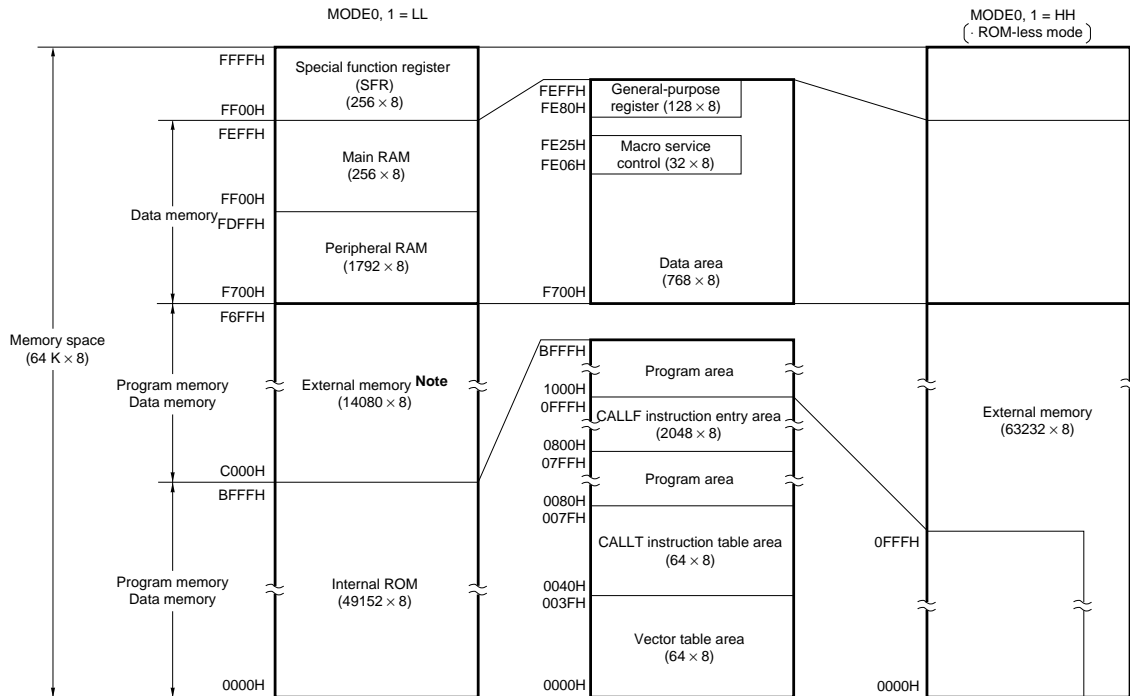
2. CPU ARCHITECTURE

2.1 MEMORY SPACE

The μPD78366A can access a memory space of 64K bytes. Figures 2-1 through 2-3 show the memory map.

★

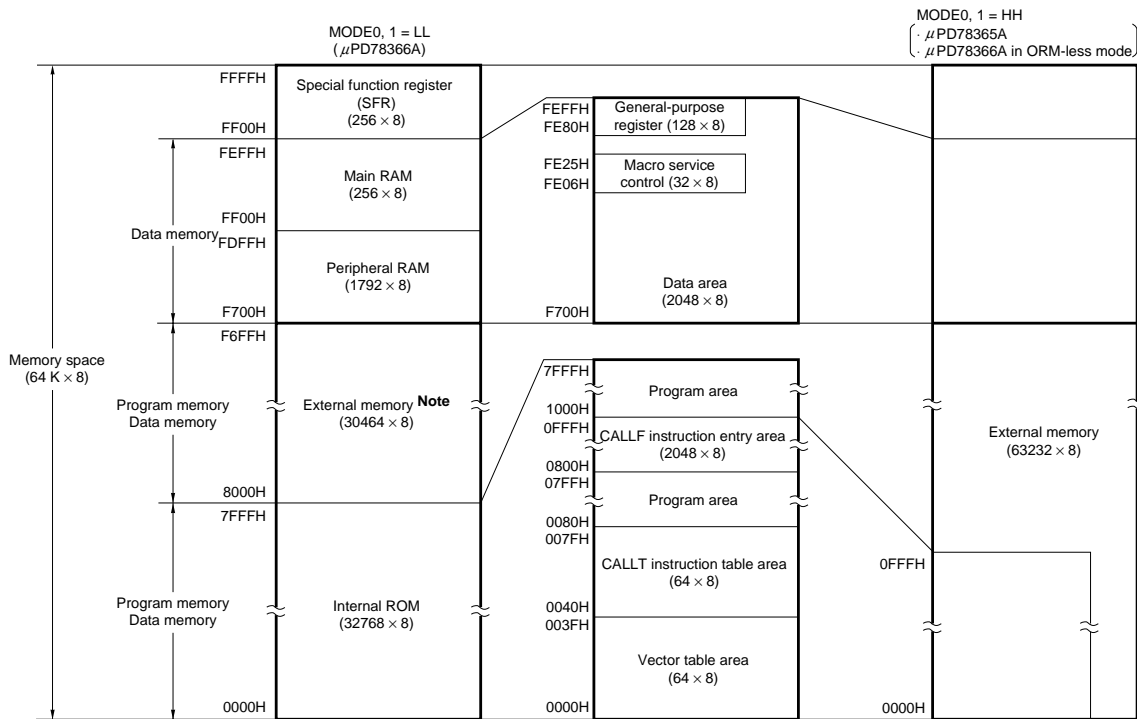
Figure 2-1. Memory Map (μPD78368A)



Note Accessed in external memory expansion mode.

Caution For word access (including stack operations) to the main RAM area (FE00H-FEFFH), the address that specifies the operand must be an even value.

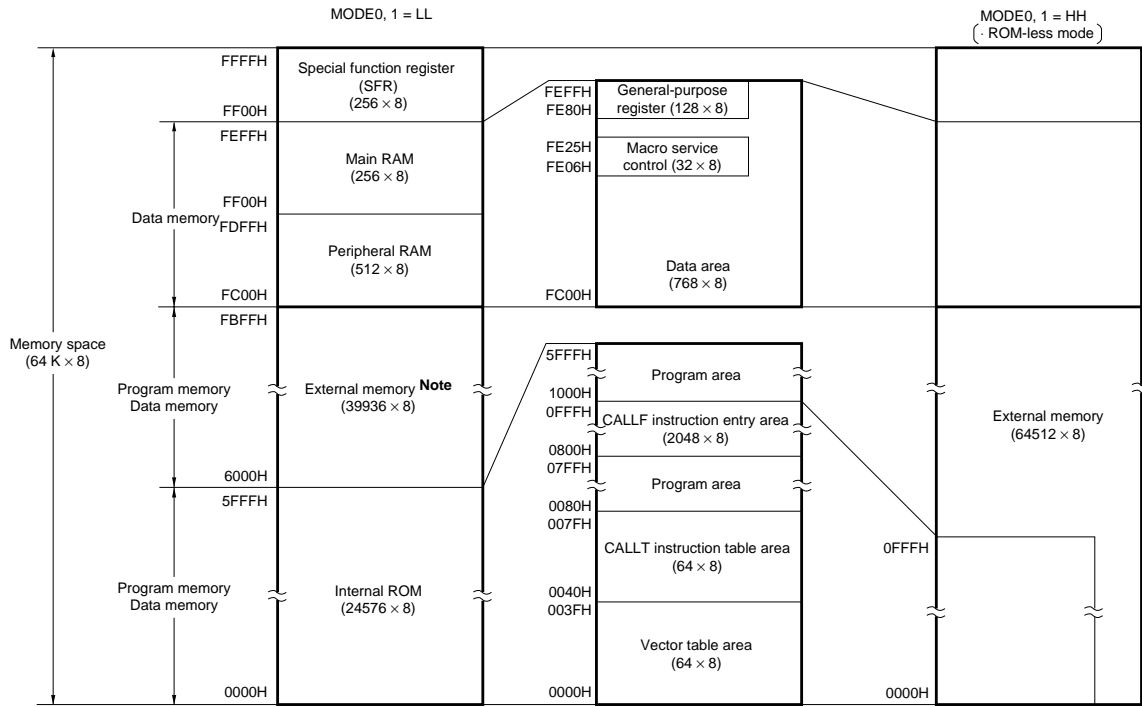
Figure 2-2. Memory Map (μPD78365A, 78366A)



Note Accessed in external memory expansion mode.

Caution For word access (including stack operations) to the main RAM area (FE00H-FEFFFH), the address that specifies the operand must be an even value.

Figure 2-3. Memory Map (μ PD78363A)



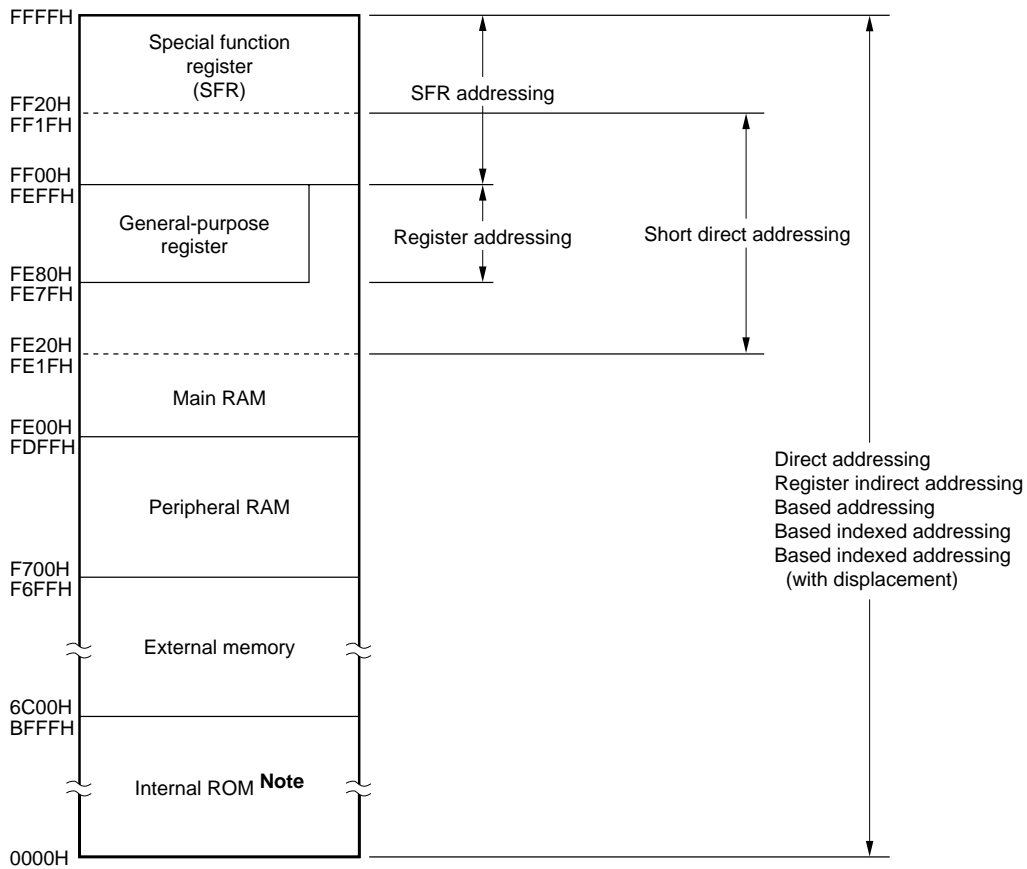
Note Accessed in external memory expansion mode.

Caution For word access (including stack operations) to the main RAM area (FE00H-FEFFFH), the address that specifies the operand must be an even value.

2.2 DATA MEMORY ADDRESSING

The μPD78366A is provided with many addressing modes that improve the operability of the memory and can be used with high-level languages. Especially, an area of addresses F700H-FFFFH (In the μPD78363A, FC00H-FFFFH) to which the data memory is mapped can be addressed in a mode peculiar to the functions provided in this area, including special function registers (SFR) and general-purpose registers.

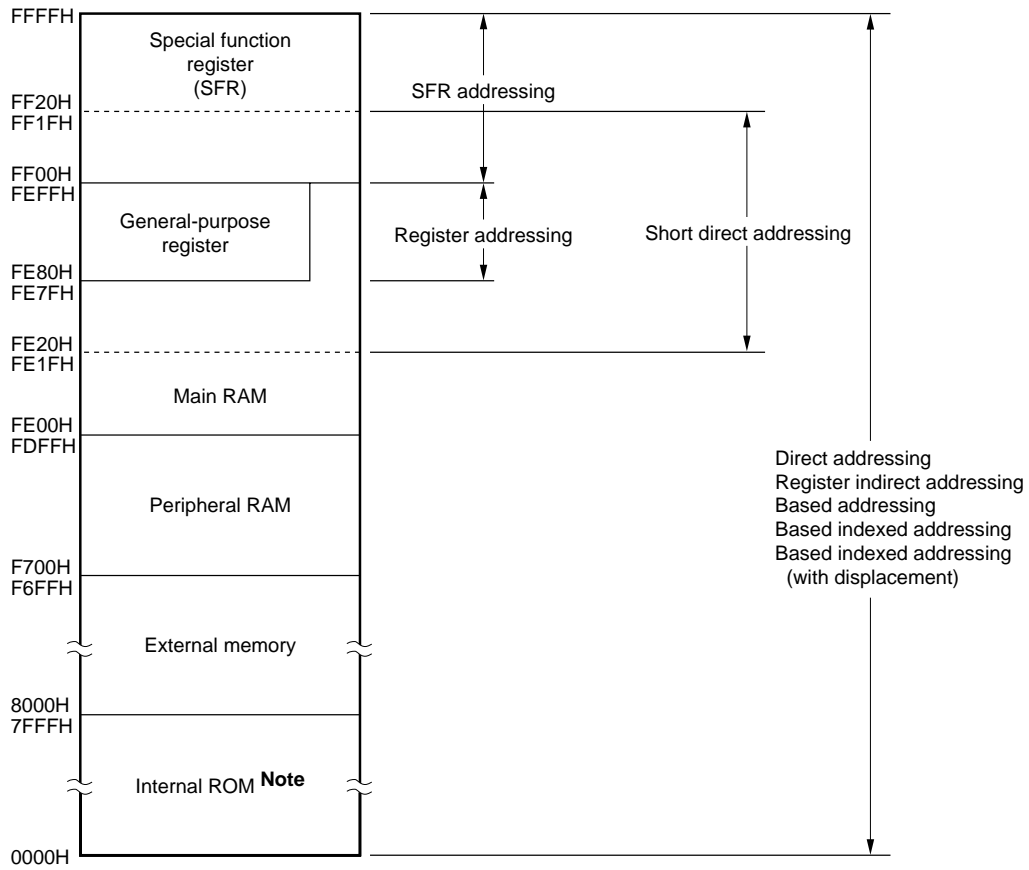
Figure 2-4. Data Memory Addressing (μPD78368A)



Note Is external memory in the ROMless mode.

Caution For word access (including stack operations) to the main RAM area (FE00H-FEFFFH), the address that specifies the operand must be an even value.

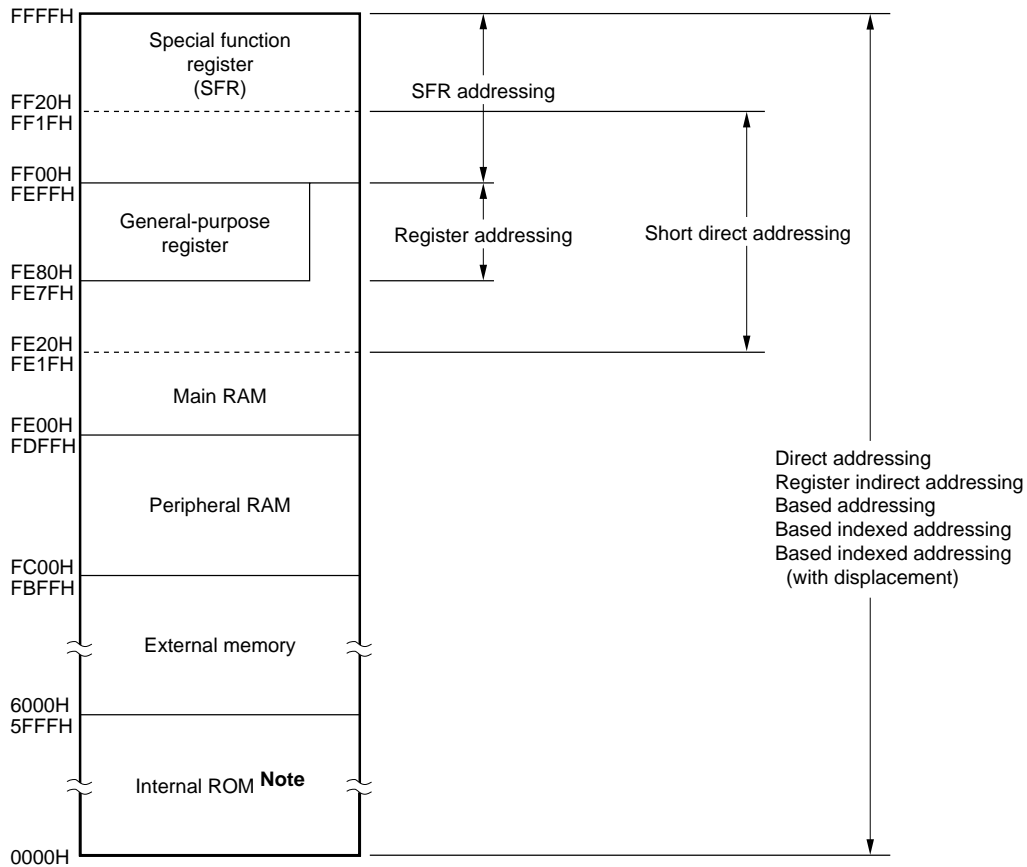
Figure 2-5. Data Memory Addressing (μPD78365A, 78366A)



Note Is external memory in the ROMless mode of the μPD78365A or μPD78366A.

Caution For word access (including stack operations) to the main RAM area (FE00H-FEFFFH), the address that specifies the operand must be an even value.

Figure 2-6. Data Memory Addressing (μPD78363A)



Note Is external memory in the ROMless mode.

Caution For word access (including stack operations) to the main RAM area (FE00H-FE7FH), the address that specifies the operand must be an even value.

2.3 PROCESSOR REGISTERS

The μPD78366A is provided with the following three types of processor registers:

- Control registers
- General-purpose registers
- Special function registers (SFRs)

2.3.2 General-Purpose Registers

The μPD78366A is provided with eight banks of general-purpose registers with one bank consisting of 8 words × 16 bits. Figure 2-10 shows the configuration of the general-purpose register banks. The general-purpose registers are mapped to an area of addresses FE80H-FEFFFH. Each of these registers can be used as an 8-bit register. In addition, two registers can be used as one 16-bit register pair (refer to Figure 2-11). These general-purpose registers facilitate complicated multitask processing.

Figure 2-10. Configuration of General-Purpose Register Banks

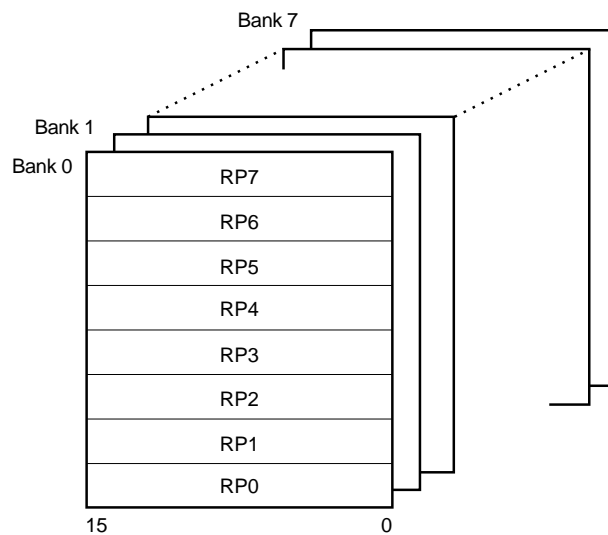
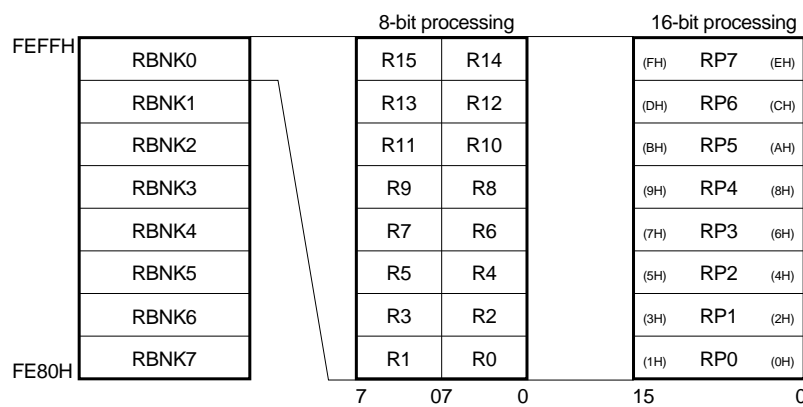


Figure 2-11. Processing Bits of General-Purpose Registers



2.3.3 Special Function Registers (SFR)

Special function registers (SFRs) are registers assigned special functions such as mode registers and control registers for internal peripheral hardware, and are mapped to a 256-byte address space at FF00H through FFFFH.

Table 2-1 lists the SFRs. The meanings of the symbols in this table are as follows:

- Symbol Indicates the mnemonic symbol for an SFR.
This mnemonic can be coded in the operand field of an instruction.
- R/W Indicates whether the SFR can be read or written.
R/W : Read/write
R : Read only
W : Write only
- Bit units for manipulation Indicates bit units in which the SFR can be manipulated. The SFRs that can be manipulated in 16-bit units can be coded as an sfrp operand. Specify an even address for these SFRs.
The SFRs that can be manipulated in 1-bit units can be coded as the operand of bit manipulation instructions.
- On reset Indicates the status of the register at $\overline{\text{RESET}}$ input.

- Cautions**
1. Do not access the addresses in the range FF00H through FFFFH to which no special function register is allocated. If these addresses are accessed, malfunctioning may occur.
 2. Do not write data to the read-only registers. Otherwise, the internal circuit may not operate normally.
 3. When using read data as byte data, process undefined bit(s) first.
 4. TOUT and TXS are write-only registers. Do no read these registers.
 5. Bits 0, 1, and 4 of SBIC are write-only bits. When these bits are read, they are always "0".

Table 2-1. List of Special Function Registers (1/5)

| Address | Special function register (SFR) | Symbol | R/W | Bit units for manipulation | | | On reset | |
|---------|---------------------------------|---------------------|-----|----------------------------|--------|---------|------------|-------|
| | | | | 1 bit | 8 bits | 16 bits | | |
| FF00H | Port 0 | P0 | R/W | ○ | ○ | – | Undefined | |
| FF01H | Port 1 | P1 | | ○ | ○ | – | | |
| FF02H | Port 2 | P2 | R | ○ | ○ | – | | |
| FF03H | Port 3 | P3 | R/W | ○ | ○ | – | | |
| FF04H | Port 4 | P4 ^{Note} | | ○ | ○ | – | | |
| FF05H | Port 5 | P5 ^{Note} | | ○ | ○ | – | | |
| FF07H | Port 7 | P7 | R | ○ | ○ | – | | |
| FF08H | Port 8 | P8 | R/W | ○ | ○ | – | | |
| FF09H | Port 9 | P9 | | ○ | ○ | – | | |
| FF10H | Compare register 00 | CM00 | | – | – | ○ | | |
| FF11H | | | | | | | | |
| FF12H | Compare register 01 | CM01 | | – | – | ○ | | |
| FF13H | | | | | | | | |
| FF14H | Compare register 02 | CM02 | | – | – | ○ | | |
| FF15H | | | | | | | | |
| FF16H | Compare register 03 | CM03 | | – | – | ○ | | |
| FF17H | | | | | | | | |
| FF18H | Buffer register CM00 | BFCM00 | – | – | ○ | | | |
| FF19H | | | | | | | | |
| FF1AH | Buffer register CM01 | BFCM01 | – | – | ○ | | | |
| FF1BH | | | | | | | | |
| FF1CH | Buffer register CM02 | BFCM02 | – | – | ○ | | | |
| FF1DH | | | | | | | | |
| FF1EH | Timer register 0 | TM0 | R | – | – | ○ | 0000H | |
| FF1FH | | | | | | | | |
| FF20H | Port 0 mode register | PM0 | R/W | ○ | ○ | – | FFH | |
| FF21H | Port 1 mode register | PM1 | | ○ | ○ | – | | |
| FF23H | Port 3 mode register | PM3 | | ○ | ○ | – | x111 1111B | |
| FF25H | Port 5 mode register | PM5 ^{Note} | | ○ | ○ | – | FFH | |
| FF28H | Port 8 mode register | PM8 | | ○ | ○ | – | xx11 1111B | |
| FF29H | Port 9 mode register | PM9 | | ○ | ○ | – | xxxx 1111B | |
| FF2CH | Reload register | DTIME | R/W | – | – | ○ | Undefined | |
| FF2DH | | | | | | | | |
| FF2EH | Timer unit mode register 0 | TUM0 | | ○ | ○ | – | 00H | |
| FF2FH | Timer unit mode register 1 | TUM1 | | ○ | ○ | – | | |
| FF30H | Compare register 10 | CM10 | | – | – | ○ | Undefined | |
| FF31H | | | | | | | | |
| FF32H | Timer register 1 | TM1 | | R | – | – | ○ | 0000H |
| FF33H | | | | | | | | |

Note Not provided for the μPD78365A.

Table 2-1. List of Special Function Registers (2/5)

| Address | Special function register (SFR) | Symbol | R/W | Bit units for manipulation | | | On reset |
|---------|-------------------------------------|--------|-----|----------------------------|--------|---------|------------|
| | | | | 1 bit | 8 bits | 16 bits | |
| FF34H | Capture/compare register 20 | CC20 | R/W | - | - | ○ | Undefined |
| FF35H | | | | | | | |
| FF36H | Capture register 20 | CT20 | R | - | - | ○ | 0000H |
| FF37H | | | | | | | |
| FF38H | Timer register 2 | TM2 | R | - | - | ○ | 0000H |
| FF39H | | | | | | | |
| FF3AH | Buffer register CM03 | BFCM03 | R/W | - | - | ○ | Underfined |
| FF3BH | | | | | | | |
| FF3CH | External interrupt mode register 0 | INTM0 | R/W | ○ | ○ | - | 00H |
| FF3DH | External interrupt mode register 1 | INTM1 | | ○ | ○ | - | |
| FF40H | Port 0 mode control register | PMC0 | R/W | ○ | ○ | - | ×000 0000B |
| FF43H | Port 3 mode control register | PMC3 | | ○ | ○ | - | |
| FF44H | Pull-up resistor option register L | PUOL | R/W | ○ | ○ | - | 00H |
| FF45H | Pull-up resistor option register H | PUOH | | ○ | ○ | - | |
| FF48H | Port 8 mode control register | PMC8 | R/W | ○ | ○ | - | ××00 0000B |
| FF4EH | Sampling control register 0 | SMPC0 | | ○ | ○ | - | |
| FF4FH | Sampling control register 1 | SMPC1 | R/W | ○ | ○ | - | 00H |
| FF50H | Capture/compare register 30 | CC30 | | - | - | ○ | |
| FF51H | | | | | | | |
| FF52H | Capture register 30 | CT30 | R | - | - | ○ | Undefined |
| FF53H | | | | | | | |
| FF54H | Capture register 31 | CT31 | R | - | - | ○ | 0000H |
| FF55H | | | | | | | |
| FF56H | Timer register 3 | TM3 | R | - | - | ○ | 0000H |
| FF57H | | | | | | | |
| FF58H | Compare register 40 | CM40 | R/W | - | - | ○ | Undefined |
| FF59H | | | | | | | |
| FF5AH | Compare register 41 | CM41 | R/W | - | - | ○ | 0000H |
| FF5BH | | | | | | | |
| FF5CH | Timer register 4 | TM4 | R | - | - | ○ | 0000H |
| FF5DH | | | | | | | |
| FF5EH | Timer control register 4 | TMC4 | R/W | - | ○ | - | 00H |
| FF5FH | Timer out register | TOUT | W | - | ○ | - | ××01 0101B |
| FF60H | Real-time output port register | RTP | R/W | ○ | ○ | - | Undefined |
| FF61H | Real-time output port mode register | RTPM | | ○ | ○ | - | |
| FF62H | Port read control register | PRDC | R/W | ○ | ○ | - | 00H |
| FF68H | A/D converter mode register | ADM | | ○ | ○ | - | |

Table 2-1. List of Special Function Registers (3/5)

| Address | Special function register (SFR) | Symbol | R/W | Bit units for manipulation | | | On reset | |
|---------|---|--------|----------|----------------------------|--------|---------|-----------|-----------|
| | | | | 1 bit | 8 bits | 16 bits | | |
| FF70H | Slave buffer register 0 | SBUF0 | R/W | ○ | ○ | – | Undefined | |
| FF71H | Slave buffer register 1 | SBUF1 | | ○ | ○ | – | | |
| FF72H | Slave buffer register 2 | SBUF2 | | ○ | ○ | – | | |
| FF73H | Slave buffer register 3 | SBUF3 | | ○ | ○ | – | | |
| FF74H | Slave buffer register 4 | SBUF4 | | ○ | ○ | – | | |
| FF75H | Slave buffer register 5 | SBUF5 | | ○ | ○ | – | | |
| FF76H | Master buffer register 0 | MBUF0 | | ○ | ○ | – | | |
| FF77H | Master buffer register 1 | MBUF1 | | ○ | ○ | – | | |
| FF78H | Master buffer register 2 | MBUF2 | | ○ | ○ | – | | |
| FF79H | Master buffer register 3 | MBUF3 | | ○ | ○ | – | | |
| FF7AH | Master buffer register 4 | MBUF4 | | ○ | ○ | – | | |
| FF7BH | Master buffer register 5 | MBUF5 | | ○ | ○ | – | | |
| FF7CH | Timer control register 0 | TMC0 | | ○ | ○ | – | | 00H |
| FF7DH | Timer control register 1 | TMC1 | | ○ | ○ | – | | |
| FF7EH | Timer control register 2 | TMC2 | | ○ | ○ | – | | |
| FF7FH | Timer control register 3 | TMC3 | ○ | ○ | – | | | |
| FF80H | Clocked serial interface mode register | CSIM | ○ | ○ | – | | | |
| FF82H | Serial bus interface control register | SBIC | R/W>Note | ○ | ○ | – | Undefined | |
| FF84H | Baud rate generator control register | BRGC | R/W | ○ | ○ | – | | |
| FF85H | Baud rate generator compare register | BRG | | – | ○ | – | | |
| FF86H | Serial I/O shift register | SIO | | ○ | ○ | – | | |
| FF88H | Asynchronous serial interface mode register | ASIM | R | ○ | ○ | – | 80H | |
| FF8AH | Asynchronous serial interface status register | ASIS | | ○ | ○ | – | 00H | |
| FF8CH | Serial receive buffer: UART | RXB | | – | ○ | – | Undefined | |
| FF8EH | Serial transfer shift register: UART | TXS | W | – | ○ | – | 00H | |
| FFA0H | PWM control register 0 | PWMC0 | R/W | ○ | ○ | – | | |
| FFA1H | PWM control register 1 | PWMC1 | | ○ | ○ | – | | |
| FFA2H | PWM register 0L | PWM0L | | ○ | ○ | – | | |
| FFA2H | PWM register 0 | PWM0 | | – | – | ○ | | Undefined |
| FFA3H | | | | | | | | |

Note Bits 7 and 5 : read/write
 Bits 6, 3, and 2 : read-only
 Bits 4, 1, and 0 : write-only

Table 2-1. List of Special Function Registers (4/5)

| Address | Special function register (SFR) | Symbol | R/W | Bit units for manipulation | | | On reset | | |
|---------|-----------------------------------|----------------------|-----|----------------------------|--------|---------|-----------|---|------------|
| | | | | 1 bit | 8 bits | 16 bits | | | |
| FFA4H | PWM register 1L | PWM1L | R/W | ○ | ○ | – | Undefined | | |
| FFA4H | PWM register 1 | PWM1 | | – | – | ○ | | | |
| FFA5H | | | | | | | | | |
| FFA8H | In-service priority register | ISPR | R | ○ | ○ | – | 00H | | |
| FFAAH | Interrupt mode control register | IMC | R/W | ○ | ○ | – | 80H | | |
| FFACH | Interrupt mask register 0L | MK0L | | ○ | ○ | – | FFH | | |
| FFACH | Interrupt mask register 0 | MK0 | | – | – | ○ | FFFFH | | |
| FFADH | | | | | | | | | |
| FFADH | Interrupt mask register 0H | MK0H | | ○ | ○ | – | FFH | | |
| FFB0H | A/D conversion result register 0 | ADCR0 | R | – | – | ○ | Undefined | | |
| FFB1H | | | | | | – | | ○ | – |
| FFB1H | A/D conversion result register 0H | ADCR0H | | – | ○ | – | | | |
| FFB2H | A/D conversion result register 1 | ADCR1 | | – | – | ○ | | | |
| FFB3H | | | | | | – | | ○ | – |
| FFB3H | A/D conversion result register 1H | ADCR1H | | – | ○ | – | | | |
| FFB4H | A/D conversion result register 2 | ADCR2 | | – | – | ○ | | | |
| FFB5H | | | | | | – | | ○ | – |
| FFB5H | A/D conversion result register 2H | ADCR2H | | – | ○ | – | | | |
| FFB6H | A/D conversion result register 3 | ADCR3 | | – | – | ○ | | | |
| FFB7H | | | | | | – | | ○ | – |
| FFB7H | A/D conversion result register 3H | ADCR3H | | – | ○ | – | | | |
| FFB8H | A/D conversion result register 4 | ADCR4 | | – | – | ○ | | | |
| FFB9H | | | | | | – | | ○ | – |
| FFB9H | A/D conversion result register 4H | ADCR4H | | – | ○ | – | | | |
| FFBAH | A/D conversion result register 5 | ADCR5 | | – | – | ○ | | | |
| FFBBH | | | | | | – | | ○ | – |
| FFBBH | A/D conversion result register 5H | ADCR5H | | – | ○ | – | | | |
| FFBCH | A/D conversion result register 6 | ADCR6 | | – | – | ○ | | | |
| FFBDH | | | | | | – | | ○ | – |
| FFBDH | A/D conversion result register 6H | ADCR6H | | – | ○ | – | | | |
| FFBEH | A/D conversion result register 7 | ADCR7 | | – | – | ○ | | | |
| FFBFH | | | | | | – | | ○ | – |
| FFBFH | A/D conversion result register 7H | ADCR7H | | – | ○ | – | | | |
| FFC0H | Standby control register | STBC ^{Note} | | R/W | – | ○ | | – | 0000 ×000B |
| FFC1H | CPU control word | CCW | | | ○ | ○ | | – | 00H |
| FFC2H | Watchdog timer mode register | WDM ^{Note} | | | – | ○ | | – | |

Note Can be written when a special instruction is executed.

Table 2-1. List of Special Function Registers (5/5)

| Address | Special function register (SFR) | Symbol | R/W | Bit units for manipulation | | | On reset |
|---------------------|--|--------|-----|----------------------------|--------|---------|-------------|
| | | | | 1 bit | 8 bits | 16 bits | |
| FFC4H | Memory expansion mode register | MM | R/W | ○ | ○ | – | Note |
| FFC6H | Programmable wait control register | PWC | | – | – | ○ | C0AAH |
| FFC7H | | | | | | | |
| FFD0H FFDFH | External SFR area | – | | ○ | ○ | – | Undefined |
| FFE0H | Interrupt control register (INTOV3) | OVIC3 | | ○ | ○ | – | 43H |
| FFE1H | Interrupt control register (INTP0/INTCC30) | PIC0 | | ○ | ○ | – | |
| FFE2H | Interrupt control register (INTP1) | PIC1 | | ○ | ○ | – | |
| FFE3H | Interrupt control register (INTP2) | PIC2 | | ○ | ○ | – | |
| FFE4H | Interrupt control register (INTP3/INTCC20) | PIC3 | | ○ | ○ | – | |
| FFE5H | Interrupt control register (INTP4) | PIC4 | | ○ | ○ | – | |
| FFE6H | Interrupt control register (INTTM0) | TMIC0 | | ○ | ○ | – | |
| FFE7H | Interrupt control register (INTCM03) | CMIC03 | | ○ | ○ | – | |
| FFE8H | Interrupt control register (INTCM10) | CMIC10 | | ○ | ○ | – | |
| FFE9H | Interrupt control register (INTCM40) | CMIC40 | | ○ | ○ | – | |
| FFEAH | Interrupt control register (INTCM41) | CMIC41 | | ○ | ○ | – | |
| FFEBH | Interrupt control register (INTSER) | SERIC | | ○ | ○ | – | |
| FFECH | Interrupt control register (INTSR) | SRIC | | ○ | ○ | – | |
| FFEDH | Interrupt control register (INTST) | STIC | | ○ | ○ | – | |
| FFEEH | Interrupt control register (INTCSI) | CSIIC | | ○ | ○ | – | |
| FFEFH | Interrupt control register (INTAD) | ADIC | | ○ | ○ | – | |

Note The value of the MW register at reset time differs depending on the product.

μPD78363A : 60H

μPD78365A, 78366A : 20H

★ μPD78368A : 00H

3. FUNCTIONAL BLOCKS

3.1 EXECUTION UNIT (EXU)

EXU controls address computation, arithmetic and logical operations, and data transfer through microprogram. EXU has an internal main RAM. This RAM can be accessed by instructions faster than the peripheral RAM.

3.2 BUS CONTROL UNIT (BCU)

BCU starts necessary bus cycles according to the physical address obtained by the execution unit (EXU). If EXU does not request start of the bus cycle, an address is generated to prefetch an instruction. The prefetched op code is stored in an instruction queue.

3.3 ROM/RAM

The internal ROM and RAM capacities differ depending on the product.

The μ PD78363A has a 24-KB ROM and a 512-B peripheral RAM. The μ PD78366A has a 32-KB ROM and a 1792-B peripheral RAM. The μ PD78368A has a 48-KB ROM and a 1792-B peripheral RAM. The μ PD78365A does not have a ROM and only has a 1792-B peripheral RAM.

Access to the ROM can be disabled by using the MODE0 and MODE1 pins, in which case an external memory of 64 KB can be accessed.

3.4 PORT FUNCTIONS

The μPD78366A is provided with the ports shown in Figure 3-1 for various control operations.

The functions of each port are listed in Table 3-1. These ports function not only as digital ports but also as input/output lines of the internal hardware.

Figure 3-1. Port Configuration

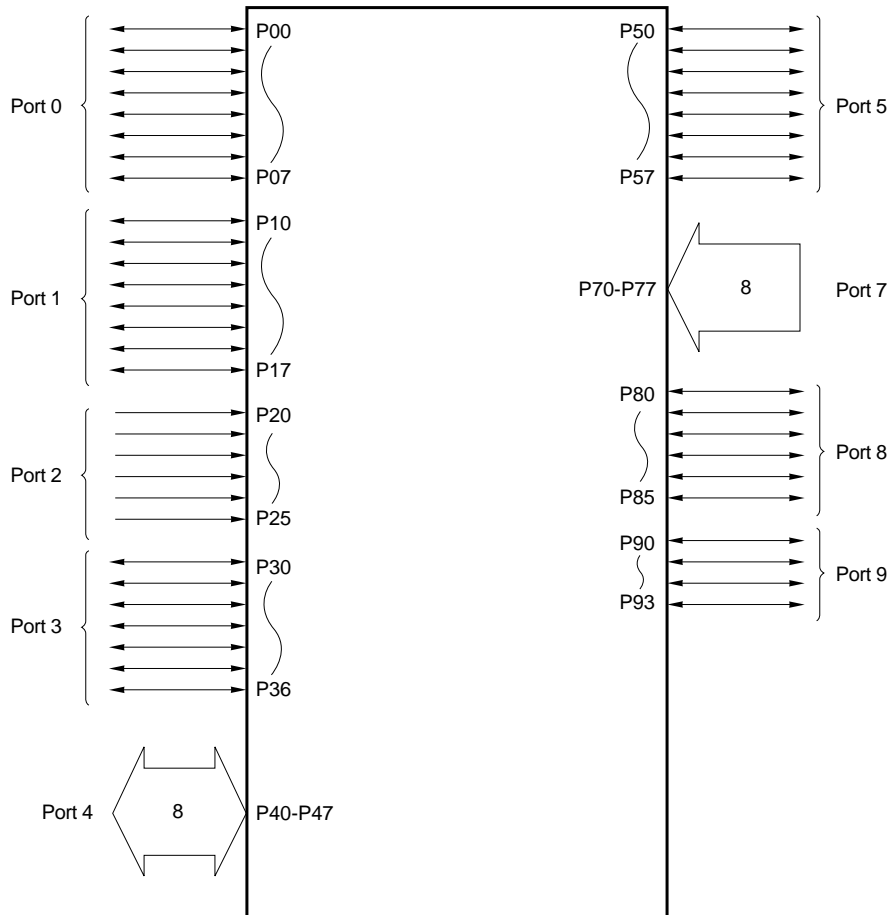


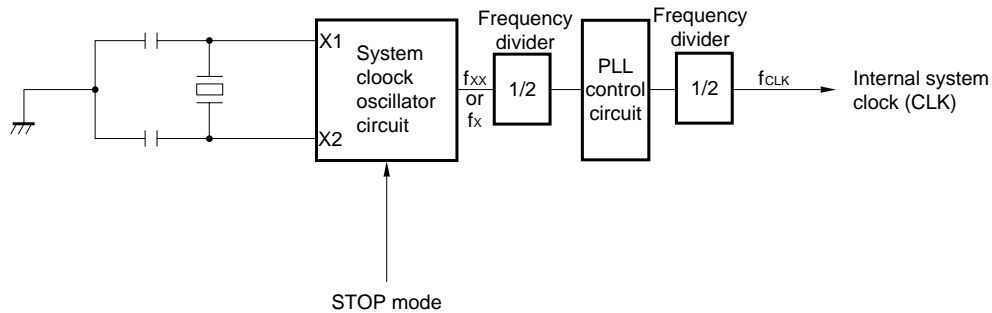
Table 3-1. Functions of Each Port

| Port | Port function | Multiplexed function |
|--------|--|--|
| Port 0 | 8-bit I/O port. Can be set in input or output mode in 1-bit units. | In control mode, serves as real-time output port (RTP), or input operation control signal of real-time pulse unit (RPU) and output PWM signal. |
| Port 1 | 8-bit I/O port. Can be set in input or output mode in 1-bit units. | — |
| Port 2 | 6-bit input port. | Inputs external interrupt and count pulse of real-time pulse unit (RPU) (fixed to the control mode). |
| Port 3 | 7-bit I/O port. Can be set in input or output in 1-bit units. | In control mode, inputs/outputs signals of serial interfaces (UART, CSI). |
| Port 4 | 8-bit I/O port. Can be set in input or output mode in 8-bit units. | Address data bus (AD0-AD7) when memory is externally expanded. |
| Port 5 | 8-bit I/O port. Can be set in input or output mode in 1-bit units. | Address bus (A8-A15) when memory is externally expanded. |
| Port 7 | 8-bit input port. | Input analog signals to A/D converter (fixed to the control mode). |
| Port 8 | 6-bit I/O port. Can be set in input or output mode in 1-bit units. | In control mode, outputs timer of real-time pulse unit (RPU). |
| Port 9 | 4-bit I/O port. Can be set in input or output mode in 1-bit units. | Outputs control signal when memory is externally expanded. |

3.5 CLOCK GENERATOR CIRCUIT

The clock generator circuit generates and controls the internal system clock (CLK) that is supplied to the CPU.

Figure 3-2. Block Diagram of Clock Generator Circuit



- Remarks 1.** f_{xx} : crystal oscillation frequency
2. f_x : external clock frequency
3. f_{CLK} : internal system clock frequency

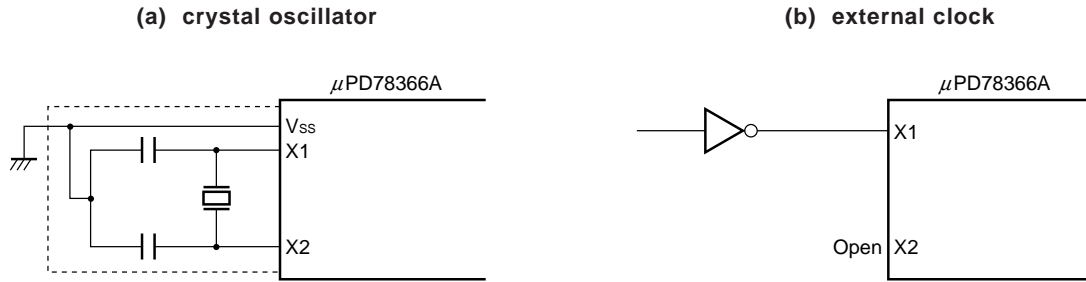
By connecting an 8-MHz crystal resonator across the X1 and X2 pins, an internal system clock of up to 16 MHz (f_{CLK}) can be generated.

The system clock oscillation circuit oscillates by using the crystal resonator connected across the X1 and X2 pins. It stops oscillation in standby mode.

An external clock can also be input. To do so, input the clock signal to the X1 pin and leave the X2 pin open.

Caution Do not set STOP mode when the external clock is used.

Figure 3-3. External Circuit of System Clock Oscillator Circuit



Cautions 1. Wire the portion enclosed by dotted line in Figure 3-3 as follows to avoid adverse influences due to wiring capacity when using the system clock oscillation circuit.

- Keep the wiring length as short as possible.
 - Do not cross the wiring with the other signal line. Make sure that the wiring is not close to lines through which a high alternating current flows.
 - Always keep the ground point of the capacitor of the oscillation circuit at the same potential as V_{ss}. Do not ground the circuit to a ground pattern through which a high current flows.
 - Do not extract signals from the oscillator circuit.
- 2.** To input an external clock, do not connect a load such as wiring capacitance to the X2 pin.

3.6 REAL-TIME PULSE UNIT (RPU)

The real-time pulse unit (RPU) can measure pulse intervals and frequencies, and output programmable pulses (six channels of PWM control signals).

The RPU consists of five 16-bit timers (timers 0 through 4), of which one is provided with a 10-bit dead time timer, which is ideal for inverter control. In addition, a function to turn off the output by the software or an external interrupt is also provided.

Each timer has the following features:

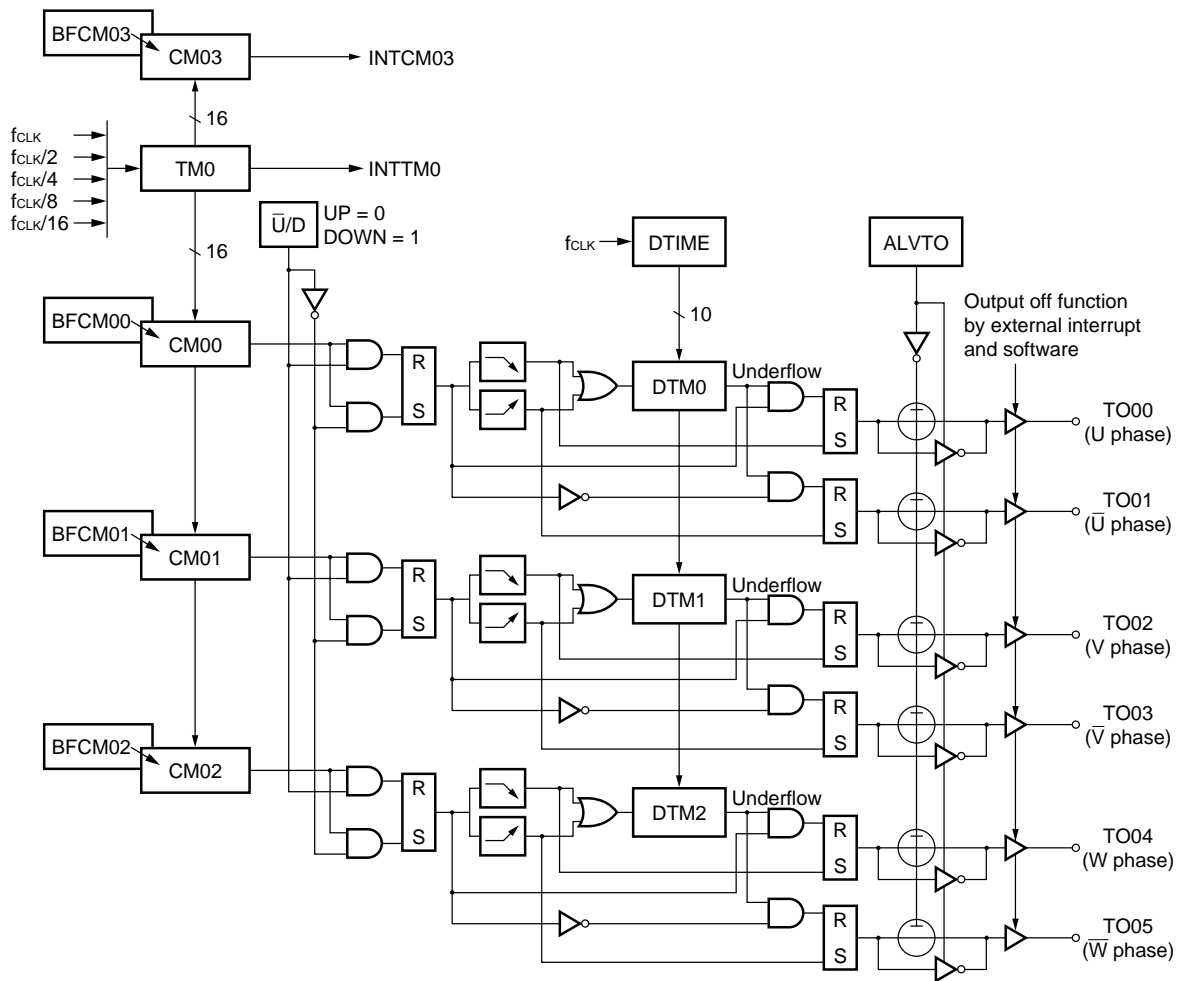
- Timer 0 : Controls the PWM period of the TO00 through TO05 pins. In addition, operates as a general-purpose interval timer. Timer 0 has the following five operation modes:
 - General-purpose interval timer mode
 - PWM mode 0 (symmetrical triangular wave)
 - PWM mode 0 (asymmetrical triangular wave)
 - PWM mode 0 (saw-tooth wave)
 - PWM mode 1
- Timer 1 : Operates as a general-purpose interval timer.
- Timers 2, 3 : Has a programmable input sampling circuit that rejects the noise of an input signal, and a capture function.
- Timer 4 : Operates as a general-purpose timer or an up-down counter. When operating as a general-purpose timer, controls the PWM cycle of the TO40 output pin. Timer 4 has the following two operation modes:
 - General-purpose timer mode
 - Up/down counter mode (UDC mode)

The RPU consists of the hardware shown in Table 3-2. Figures 3-4 through 3-12 show the block diagrams of the respective timers.

Table 3-2. Configuration of Real-Time Pulse Unit (RPU)

| | Timer register | Register | Compare register coincidence interrupt | Capture trigger | Timer output | Timer clear |
|---------|--------------------|--|--|-------------------------|--------------|-------------------|
| Timer 0 | 16-bit timer (TM0) | 16-bit compare register (CM00) | – | – | 6 | INTCM03 |
| | | 16-bit compare register (CM01) | – | | | |
| | | 16-bit compare register (CM02) | – | | | |
| | | 16-bit compare register (CM03) | INTCM03 | | | |
| Timer 1 | 16-bit timer (TM1) | 16-bit compare register (CM10) | INTCM10 | – | – | INTCM10 |
| Timer 2 | 16-bit timer (TM2) | 16-bit capture/compare register (CC20) | INTCC20 | INTP3 | – | INTCC20 |
| | | 16-bit capture register (CT20) | – | | | |
| Timer 3 | 16-bit timer (TM3) | 16-bit capture/compare register (CC30) | INTCC30 | INTP0 INTP1 INTP4 | – | INTCC30 |
| | | 16-bit capture register (CT30) | – | | | |
| | | 16-bit capture register (CT31) | – | | | |
| Timer 4 | 16-bit timer (TM4) | 16-bit compare register (CM40) | INTCM40 | – | 1 | TCLRUD INTCM40 |
| | | 16-bit compare register (CM41) | INTCM41 | | | |

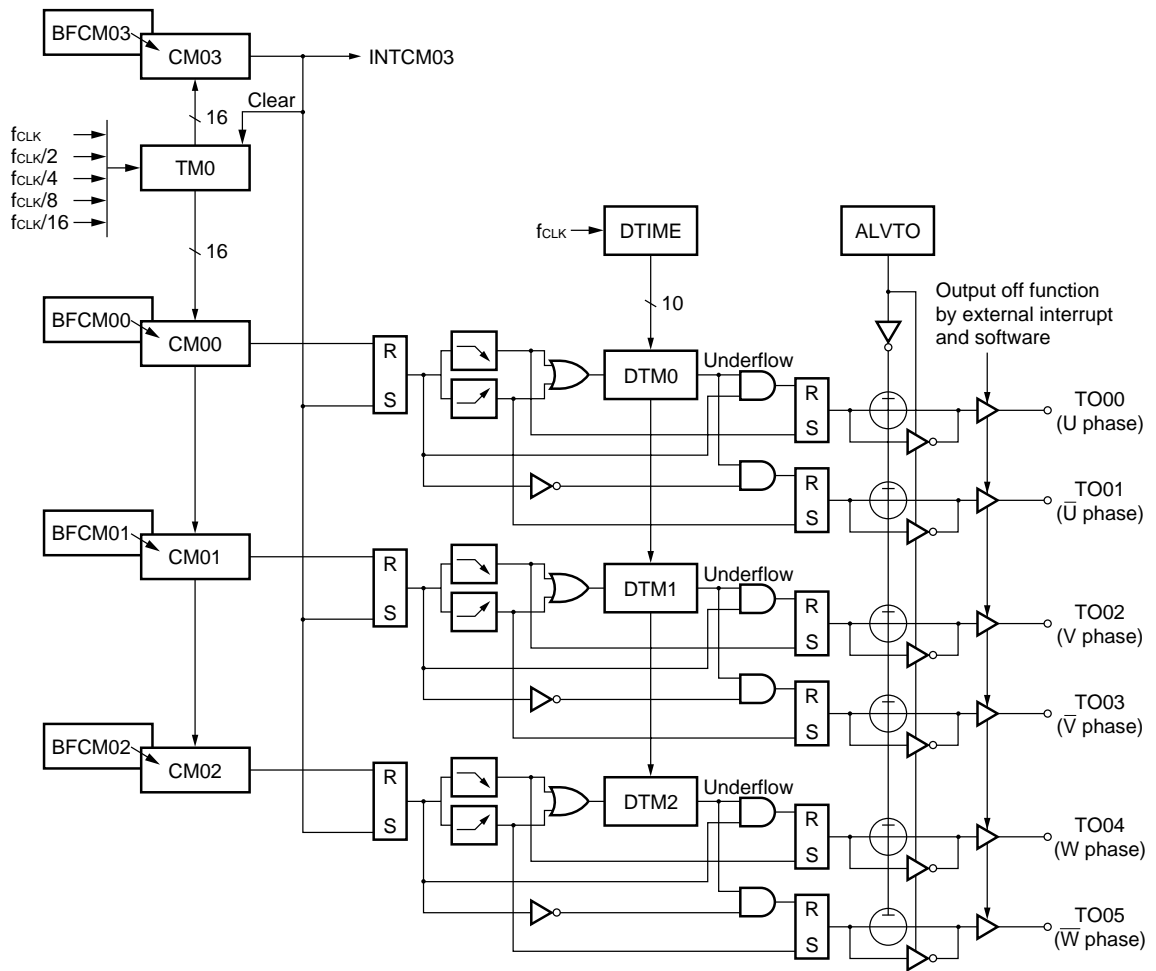
Figure 3-4. Block Diagram of Timer 0 (PWM mode 0 ... symmetrical triangular wave, asymmetrical triangular wave)



TM0 : Timer register
 CM00-CM03 : Compare registers
 BFCM00-BFCM03: Buffer registers
 DTIME : Reload register
 DTM0-DTM2 : Dead time timers
 ALVTO: Bit 2 of TUM0 register
 \bar{U}/D : Bit 3 of TMC0 register

Remark f_{CLK}: internal system clock

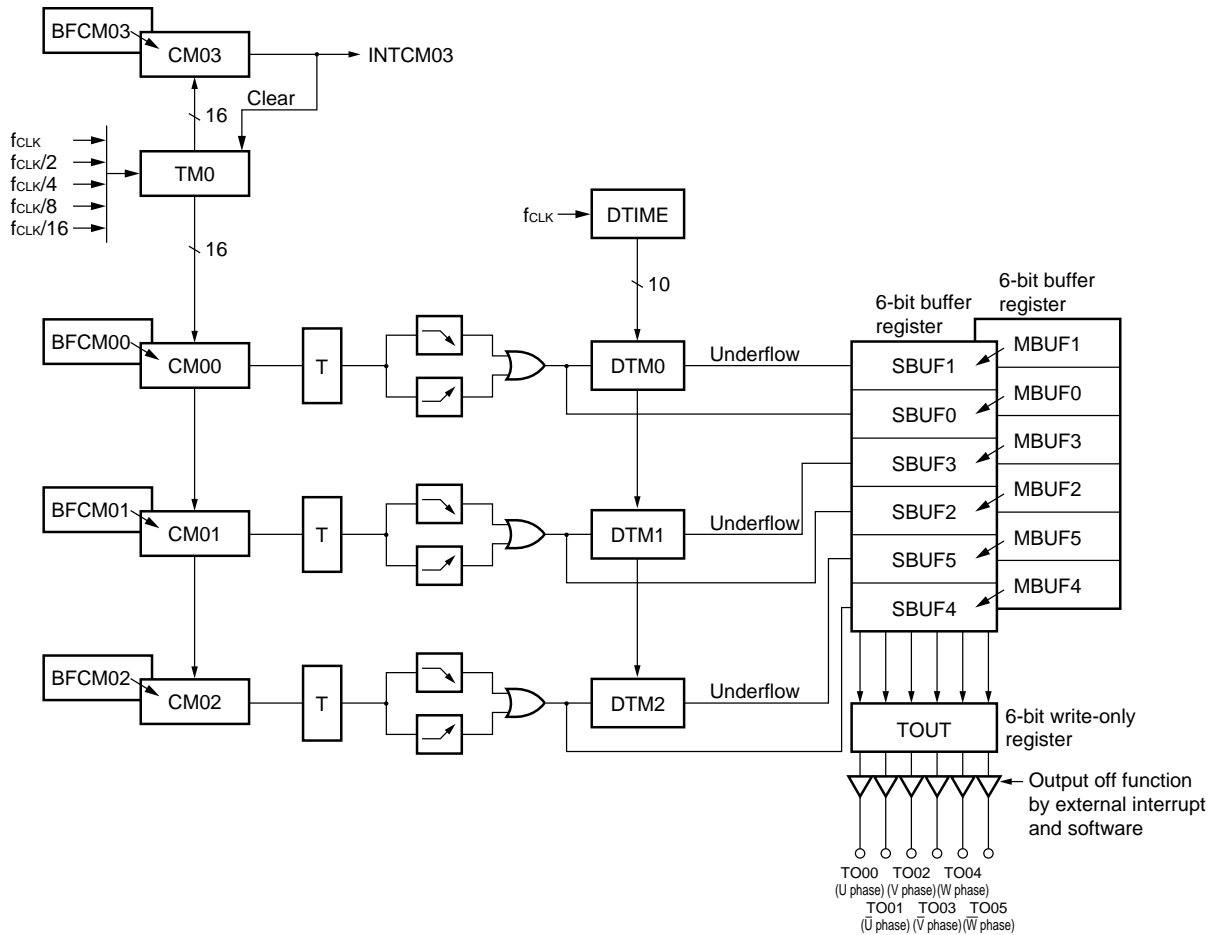
Figure 3-5. Block Diagram of Timer 0 (PWM mode 0 ... saw-tooth wave)



- TM0 : Timer register
- CM00-CM03 : Compare registers
- BFCM00-BFCM03: Buffer registers
- DTIME : Reload register
- DTM0-DTM2 : Dead time timers
- ALVTO : Bit 2 of TUM0 register

Remark f_{CLK}: internal system clock

Figure 3-6. Block Diagram of Timer 0 (PWM mode 1)



- | | | | |
|---------------|---------------------|-------------|---------------------------|
| TM0 | : Timer register | MBUF0-MBUF5 | : Master buffer registers |
| CM00-CM03 | : Compare registers | SBUF0-SBUF5 | : Slave buffer registers |
| BFCM00-BFCM03 | : Buffer registers | TOUT | : Timer out register |
| DTIME | : Reload register | | |
| DTM0-DTM2 | : Dead time timers | | |

Remark f_{CLK}: internal system clock

Figure 3-7. Block Diagram of Timer 0 (general-purpose interval timer mode)

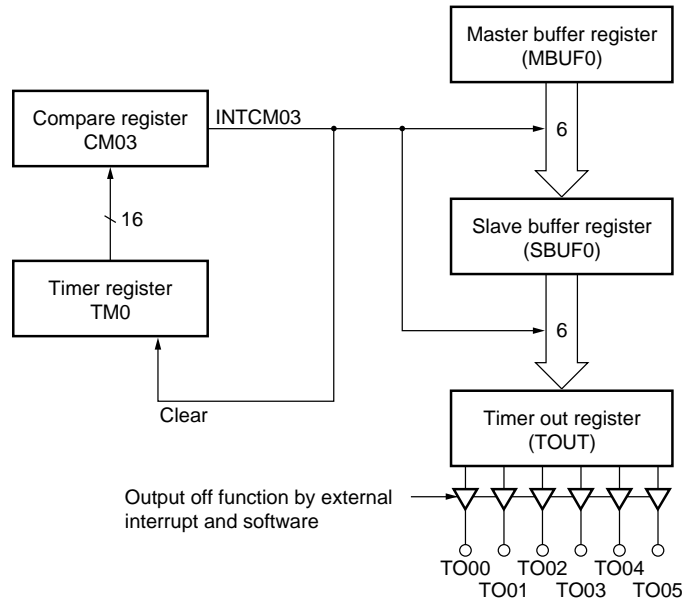
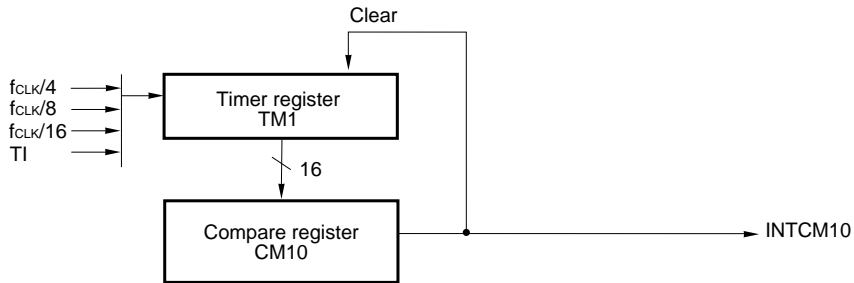
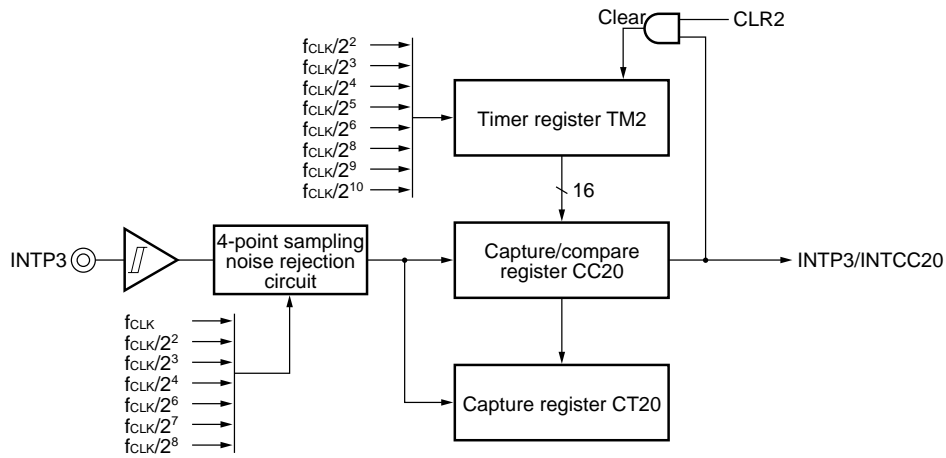


Figure 3-8. Block Diagram of Timer 1



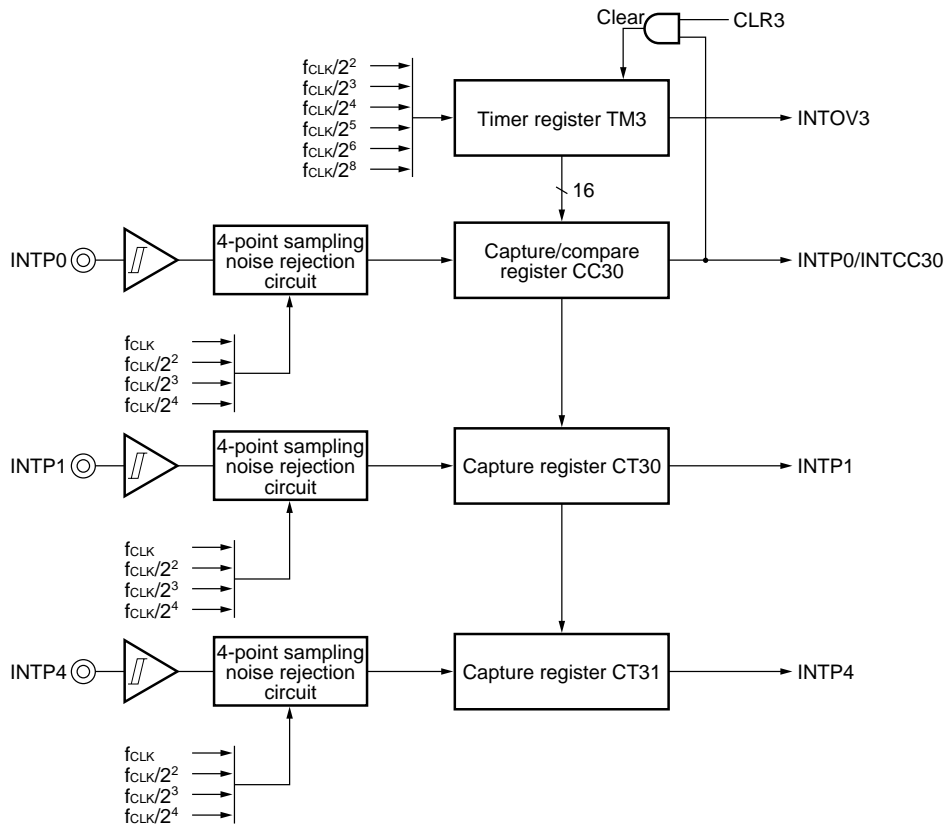
Remark f_{CLK}: internal system clock

Figure 3-9. Block Diagram of Timer 2



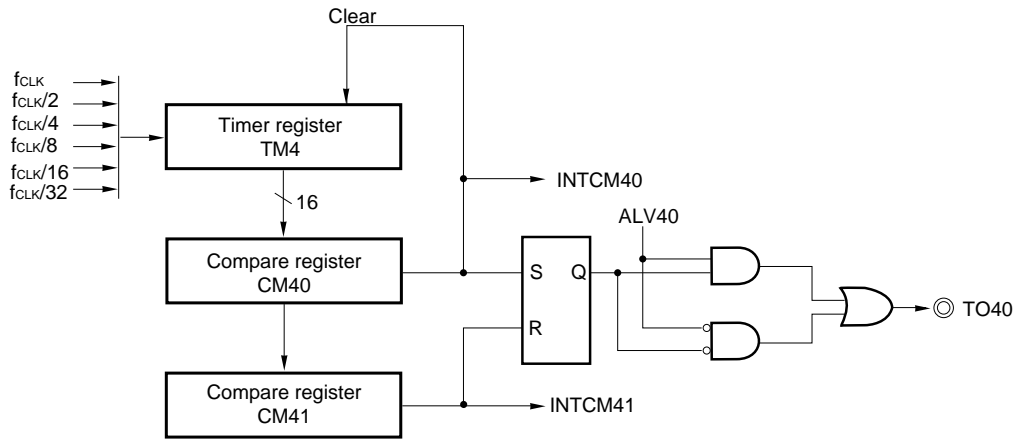
Remark f_{CLK}: internal system clock

Figure 3-10. Block Diagram of Timer 3



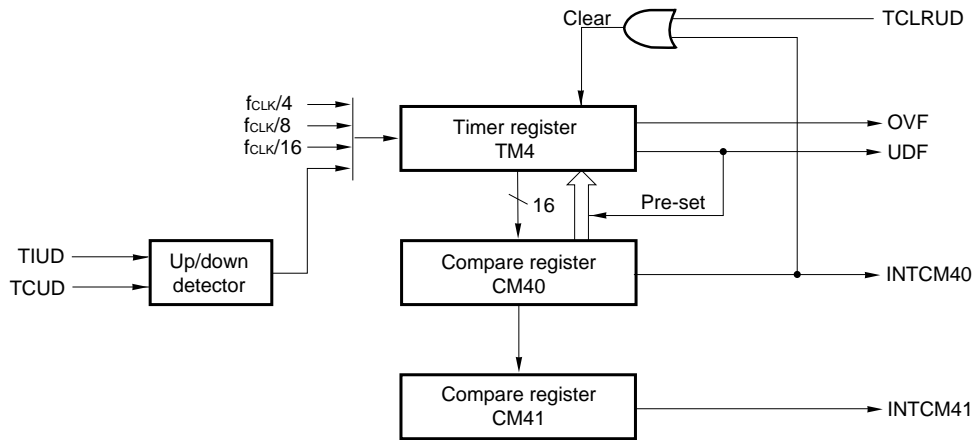
Remark f_{CLK}: internal system clock

Figure 3-11. Block Diagram of Timer 4 (General-Purpose Timer Mode)



Remark fCLK: internal system clock

Figure 3-12. Block Diagram of Timer 4 (UDC Mode)



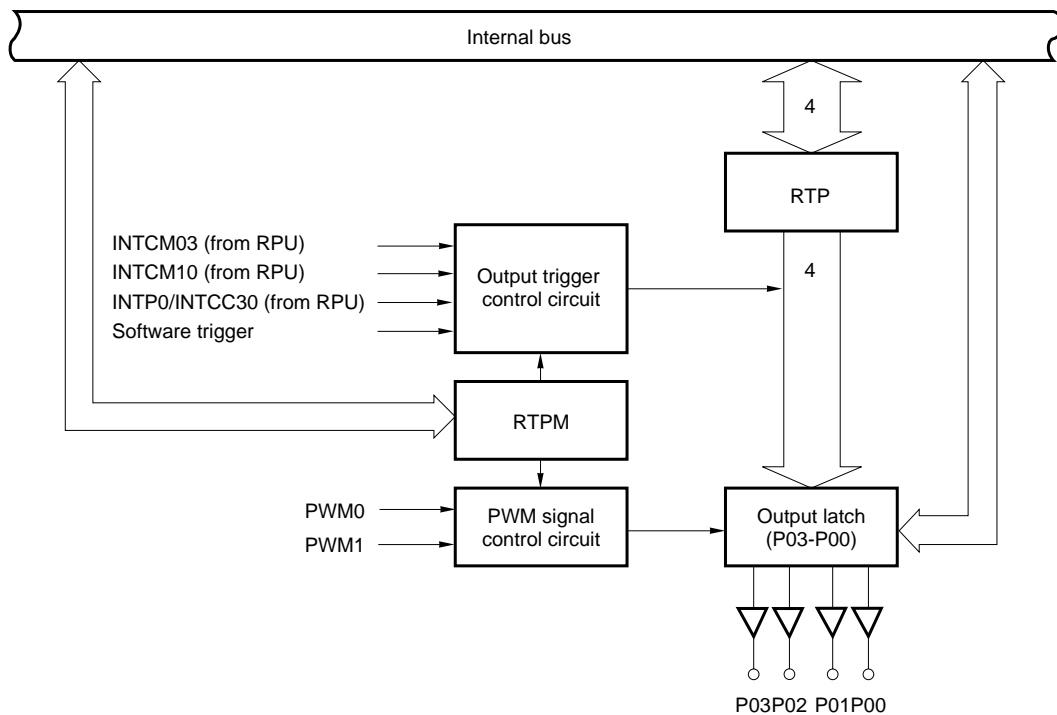
Remark fCLK: internal system clock

3.7 REAL-TIME OUTPUT PORT (RTP)

The real-time output port is a 4-bit port that can output the contents of the real-time output port register (RTP) in synchronization with the trigger signal from the real-time pulse unit (RPU). It can output synchronization pulses of multiple channels.

Also, PWM modulation can be applied to P00-P03.

Figure 3-13. Block Diagram of Real-Time Output Port

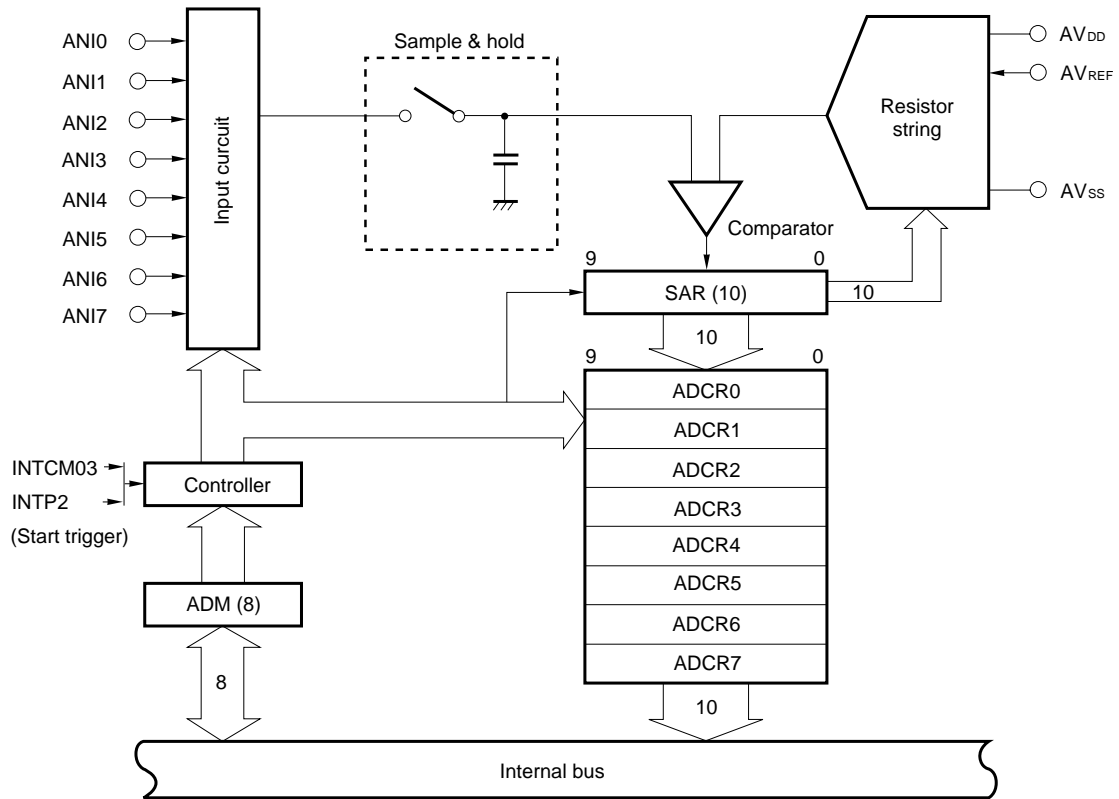


3.8 A/D CONVERTER

The μ PD78366A contains a high-speed, high-resolution 10-bit analog-to-digital (A/D) converter (conversion time 12.6 μ s at an internal clock frequency of 16 MHz). Successive approximation type is adopted. This converter is provided with eight analog input lines (ANI0-ANI7) and can perform various operations as the application requires, in select, scan, and mixed modes.

When A/D conversion ends, an internal interrupt (INTAD) occurs. This interrupt can start a macro service that executes automatic data transfer through hardware.

Figure 3-14. Block Diagram of A/D Converter



3.9 SERIAL INTERFACE

The μPD78366A is provided with the following two independent serial interfaces:

- Asynchronous serial interface (UART) (with pin selection function)
- Clocked serial interface
 - 3-line serial I/O mode
 - Serial bus interface mode (SBI mode)

Since the μPD78366A contains a baud rate generator (BRG), any serial transfer rate can be set regardless of the operating clock frequency. The baud rate generator is a block to generate the shift clock for the transmit/receive serial interface, and is used commonly with the two channels of the serial interfaces.

The serial transfer rate can be selected in a range of 110 bps to 38.4 Kbps by the mode register.

Figure 3-15. Block Diagram of Asynchronous Serial Interface

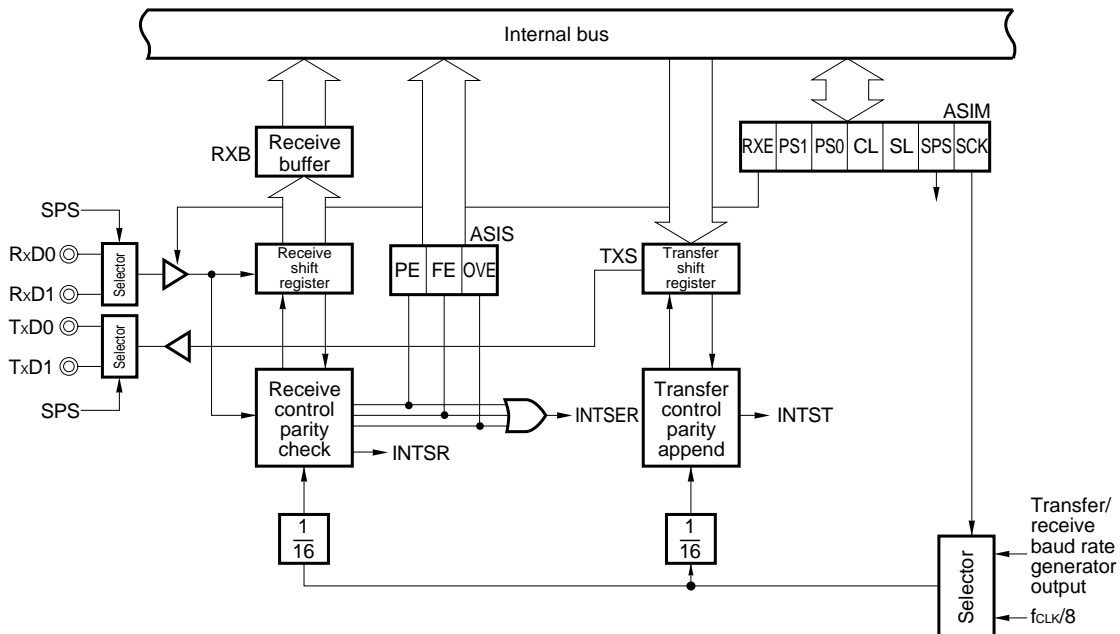


Figure 3-16. Block Diagram of Clocked Serial Interface

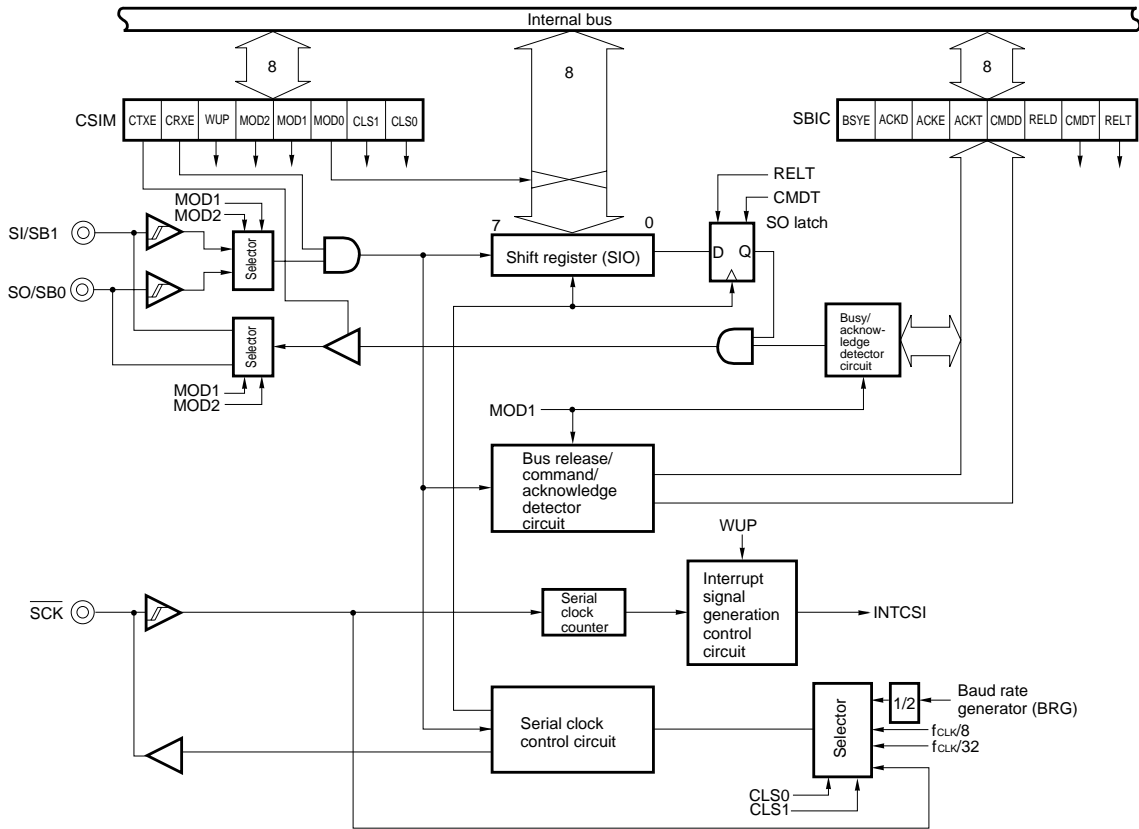
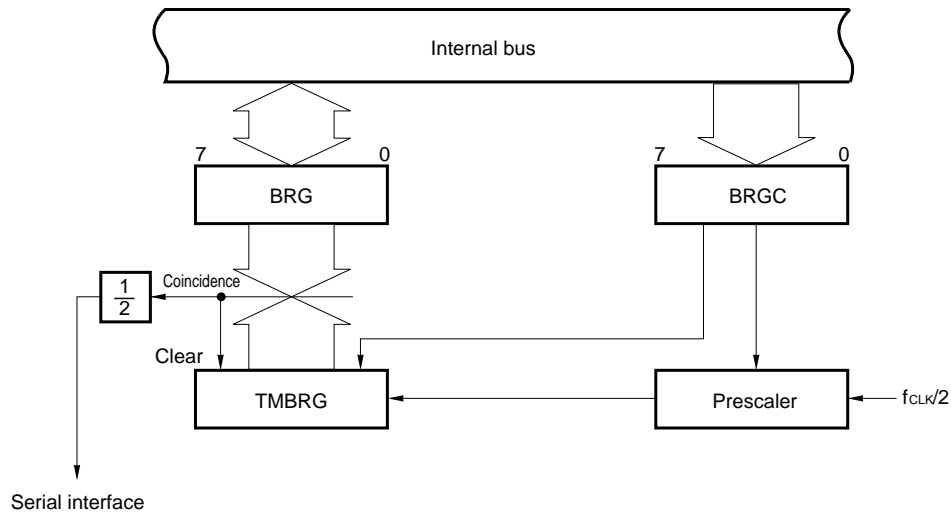


Figure 3-17. Block Diagram of Baud Rate Generator

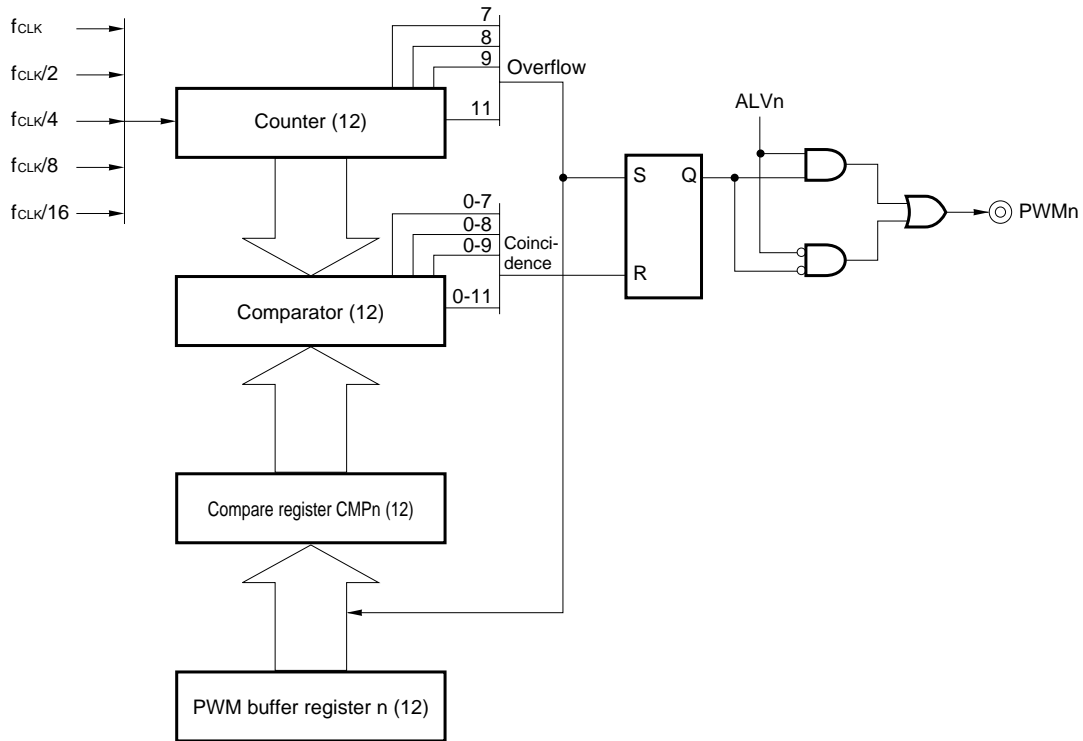


3.10 PWM UNIT

The μ PD78366A is provided with two lines that output 8-/9-/10-/12-bit resolution variable PWM signals. The PWM output can be used as a digital-to-analog conversion output by connecting an external lowpass filter, and ideal for controlling actuators such as motors.

An output of between 244 Hz and 62.5 kHz can be obtained, depending on the combination of the count clock (62.5 ns to $1\ \mu\text{s}$) and counter bit length (8, 9, 10, or 12) (at an internal clock frequency of 16 MHz).

Figure 3-18. Block Diagram of PWM Unit

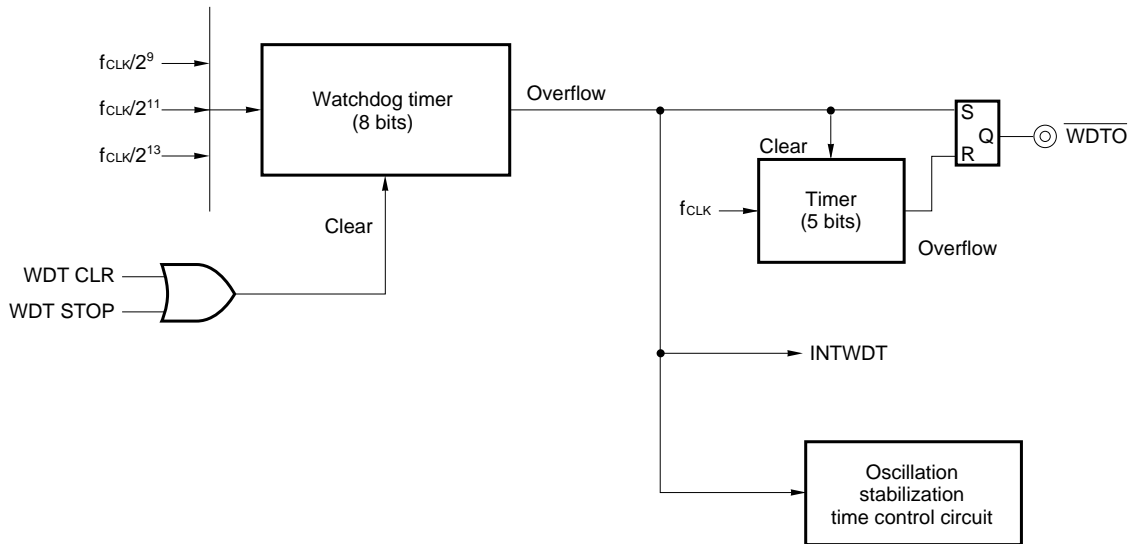


Remark $n = 0, 1$

3.11 WATCHDOG TIMER (WDT)

The watchdog timer is a free running timer equipped with a non-maskable interrupt function to prevent program hang-up or deadlock. When an error of the program is detected, the overflow interrupt (INTWDT) of the watchdog timer occurs and the watchdog timer output pin ($\overline{\text{WDTO}}$) goes low. By connecting this output pin to the $\overline{\text{RESET}}$ pin, any malfunctioning of the application system due to program error can be prevented.

Figure 3-19. Block Diagram of Watchdog Timer



4. INTERRUPT FUNCTIONS

4.1 OUTLINE

The μPD78366A is provided with powerful interrupt functions that can process interrupt requests from the internal hardware peripherals and external sources. In addition, the following three interrupt processing modes are available. In addition, four levels of interrupt priority can be specified.

- Vectored interrupt processing
- Macro service
- Context switching

Table 4-1. Interrupt Sources

| Type | Note | Interrupt source | | Source unit | Vector table address | Macro service | Context switching |
|--------------|-------|-----------------------|---|--------------|----------------------|---------------|-------------------|
| | | Name | Trigger | | | | |
| Non-maskable | – | NMI | NMI pin input | External | 0002H | None | None |
| | – | INTWDT | Watchdog timer | WDT | 0004H | | |
| Maskable | 0 | INTOV3 | Overflow of timer 3 | RPU | 0006H | Provided | Provided |
| | 1 | INTP0/INTCC30 | INTP0 pin input/CC30 coincidence signal | External/RPU | 0008H | | |
| | 2 | INTP1 | INTP1 pin input | External | 000AH | | |
| | 3 | INTP2 | INTP2 pin input | | 000CH | | |
| | 4 | INTP3/INTCC20 | INTP3 pin input/CC20 coincidence signal | External/RPU | 000EH | | |
| | 5 | INTP4 | INTP4 pin input | External | 0010H | | |
| | 6 | INTTM0 | Underflow of timer 0 | RPU | 0012H | | |
| | 7 | INTCM03 | CM03 coincidence signal | | 0014H | | |
| | 8 | INTCM10 | CM10 coincidence signal | | 0016H | | |
| | 9 | INTCM40 | CM40 coincidence signal | | 0018H | | |
| | 10 | INTCM41 | CM41 coincidence signal | | 001AH | | |
| | 11 | INTSER | Receive error of UART | UART | 001CH | | |
| | 12 | INTSR | End of UART reception | | 001EH | | |
| | 13 | INTST | End of UART transfer | | 0020H | | |
| | 14 | INTCSI | End of CSI transfer/reception | CSI | 0022H | | |
| 15 | INTAD | End of A/D conversion | A/D | 0024H | | | |
| Software | – | BRK | BRK instruction | – | 003EH | None | None |
| | – | BRKCS | BRKCS instruction | – | – | | Provided |
| Exception | – | TRAP | Illegal op code trap | – | 003CH | | None |
| Reset | – | RESET | Reset input | – | 0000H | None | |

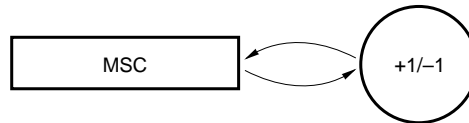
Note Default priority : Priority that takes precedence when two or more maskable interrupts occur at the same time. 0 is the highest priority, and 15 is the lowest.

4.2 MACRO SERVICE

The μPD78366A has a total of five macro services. Each macro service is described below.

(1) Counter mode: EVTCNT

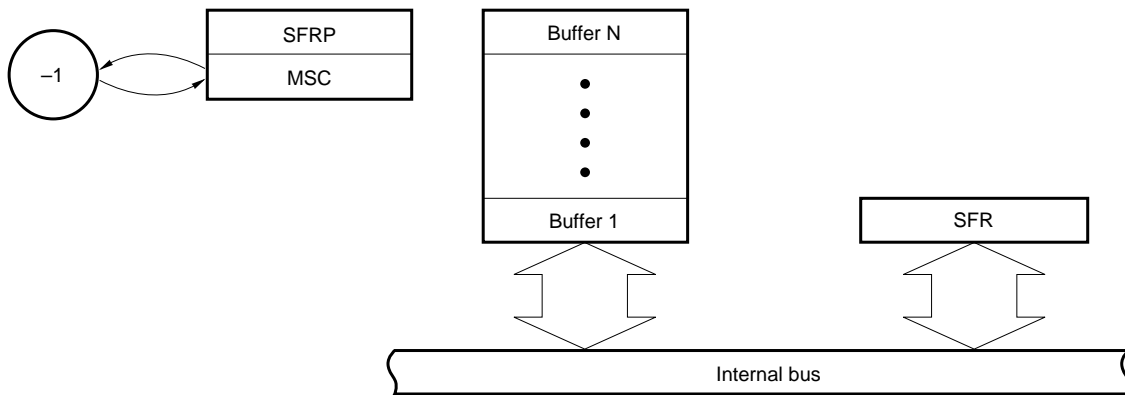
- Operation
 - (a) Increments or decrements an 8-bit macro service counter (MSC).
 - (b) A vector interrupt request is generated when MSC reaches 0.



- Application example: As event counter, or to measure number of times a value is captured

(2) Block transfer mode: BLKTRS

- Operation
 - (a) Transfers data block between a buffer and a SFR specified by SFR pointer (SFRP).
 - (b) The transfer source and destination can be in SFR or buffer area. The length of the transfer data can be specified to be byte or word.
 - (c) The number of times the data is to be transferred (block size) is specified by MSC.
 - (d) MSC is auto decremented by one each time the macro service has been executed.
 - (e) When MSC reaches 0, a vector interrupt request is generated.

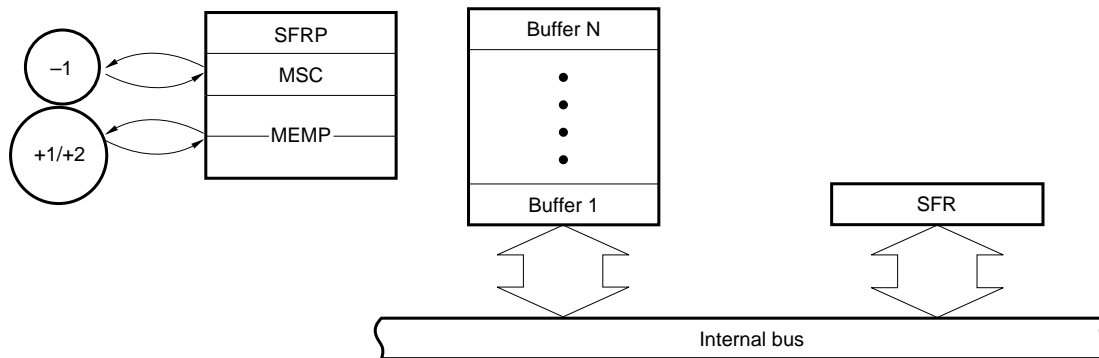


- Application example: To transfer/receive data through serial interface

(3) Block transfer mode (with memory pointer): BLKTRS-P

- Operation
This is the block transfer mode in (2) above with a memory pointer (MEMP). The appended buffer area of MEMP can be freely set on the memory space.

Remark Each time the macro service is executed, MEMP is auto incremented (by one for byte data transfer and by two for word data transfer).

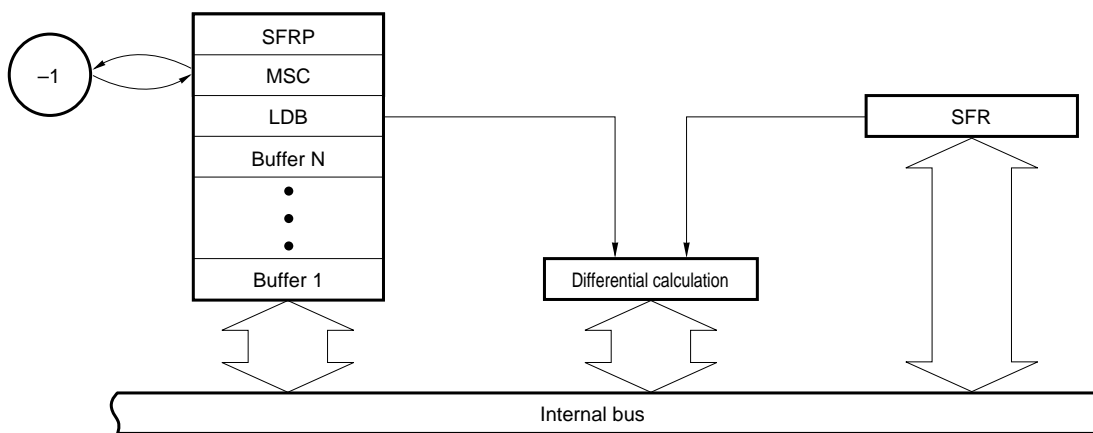


- Application example: Same as (2)

(4) Data differential mode: DTADIF

- Operation
 - Calculates the difference between the contents of SFR (current value) specified by SFRP and the contents of SFR saved to the last data buffer (LDB).
 - Stores the result of the calculation in a predetermined buffer area.
 - Stores the contents of the current value of the SFR in LDB.
 - The number of times the data is to be transferred (block size) is specified by MSC. Each time the macro service is executed, MSC is auto decremented by one.
 - When MSC reaches 0, a vector interrupt request is generated.

Remark The differential calculation can be carried out only with 16-bit SFRs.



- Application example : To measure cycle and pulse width by the capture register of the real-time pulse unit (RPU)

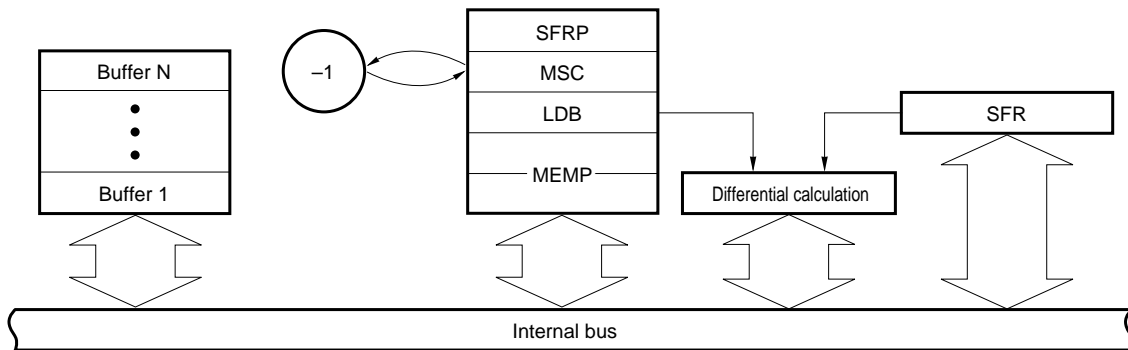
(5) Data differential mode (with memory pointer): DTADIF-P

- Operation

This is the data differential mode in (4) above with memory pointer (MEMP). By appending MEMP, the buffer area in which the differential data is to be stored can be set freely on the memory space.

- Remarks**
1. The differential calculation can be carried out only with 16-bit SFRs.
 2. The buffer is specified by the result of operation by MEMP and MSC^{Note}. MEMP is not updated after the data has been transferred.

Note $MEMP - (MSC \times 2) + 2$



- Application example: Same as (4)

4.3 CONTEXT SWITCHING

This function is to select a specific register bank through the hardware, and to branch execution to a vector address predetermined in the register bank. At the same time, it saves the present contents of the PC and PSW to the register bank when an interrupt occurs, or when the BRKCS instruction is executed.

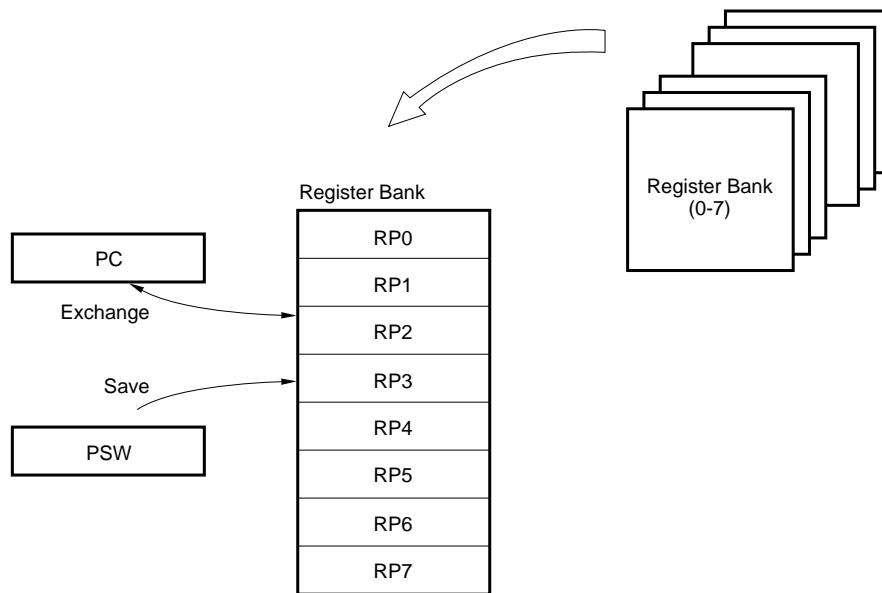
4.3.1 Context Switching Function by Interrupt Request

When a context switching enable flag corresponding to each maskable interrupt request is set to 1 in the EI (interrupt enable) status, the context switching function can be started.

The context switching operation by an interrupt request is performed as follows:

- (1) When an interrupt request is generated, a register bank to which the context is to be switched is specified by the contents of the low-order 3 bits of the row address (even address) of the corresponding vector table.
- (2) A predetermined vector address is transferred to the PC in the register bank to which the context is to be switched, and the contents of the PC and PSW immediately before the switching takes place are saved to the register bank.
- (3) Execution branches to an address indicated by the contents of the PC newly set.

Figure 4-1. Operation of Context Switching



4.3.2 Context Switching Function by BRKCS Instruction

The context switching function can be started by the BRKCS instruction.
 The operation of context switching by an interrupt request is as follows:

- (1) An 8-bit register is specified by the operand of the BRKCS instruction, and the register bank to which the context is to be switched is specified by the contents of this register (only the low-order 3 bits of 8 bits are valid).
- (2) The vector address predetermined in the register bank to which the context is to be switched is transferred to the PC, and at the same time, the contents of the PC and PSW immediately before the switching takes place are saved to the register bank.
- (3) Execution branches to the contents of the PC newly set.

4.3.3 Restoration from Context Switching

To restore from the switched context, one of the following two instructions are used. Which instruction is to be executed is determined by the source that has started the context switching.

Table 4-2. Instructions to Restore from Context Switching

| Restore instruction | Context switching starting source |
|---------------------|-----------------------------------|
| RETCS | Occurrence of interrupt |
| RETCSB | Execution of BRKCS instruction |

5. EXTERNAL DEVICE EXPANSION FUNCTION

The μ PD78366A can connect external devices (data memory, program memory, and peripheral devices) in addition to the internal ROM and RAM areas. To connect an external device, the address/data bus and read/write strobe signals are controlled by using ports 4, 5, and 9.

Table 5-1. Pin Function with External Device Connected

| Pin | Pin function with external device connected | |
|---------|---|-----------------|
| | Function | Name |
| P40-P47 | Multiplexed address/data bus | AD0-AD7 |
| P50-P57 | Address bus | A8-A15 |
| P90 | Read strobe | \overline{RD} |
| P91 | Write strobe | \overline{WR} |
| ASTB | Address strobe | ASTB |

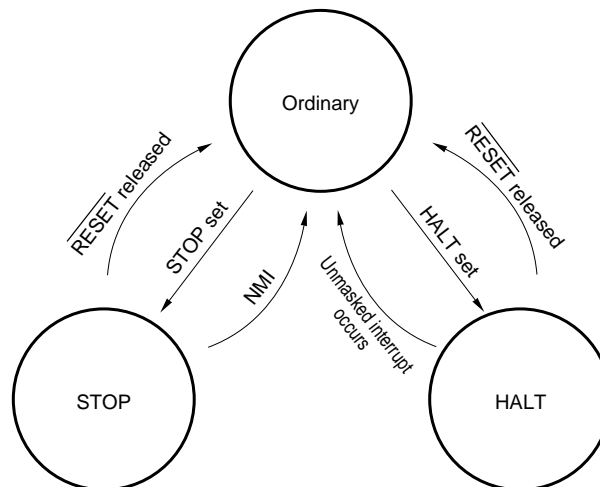
6. STANDBY FUNCTIONS

The μPD78366A is provided with standby functions to reduce the power consumption of the system. The standby functions can be effected in the following two modes:

- HALT mode In this mode, the operating clock of the CPU is stopped. By using this mode in combination with an ordinary operation mode, the μPD78366A operates intermittently to reduce the total power consumption of the system.
- STOP mode In this mode, the oscillator is stopped, and therefore the entire system is stopped. Therefore, power consumption can be minimized with only a leakage current flowing.

Each mode is set through software. Figure 6-1 shows the transition of the status in the standby modes (STOP and HALT modes).

Figure 6-1. Transition of Standby Status



7. RESET FUNCTION

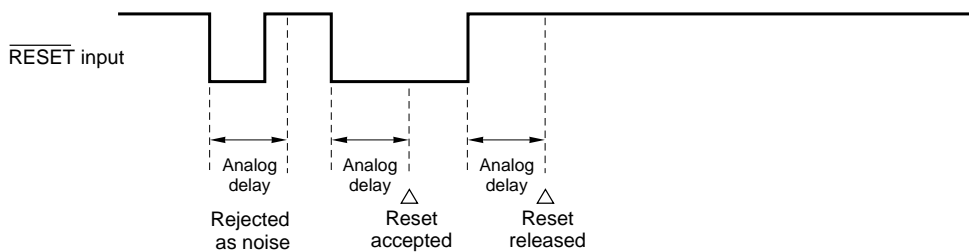
When a low level is input to the $\overline{\text{RESET}}$ pin, the system is reset, and each hardware enters the initial status (reset status). When the $\overline{\text{RESET}}$ pin goes high, the reset status is released, and program execution is started. Initialize the contents of each register through program as necessary.

Especially, change the number of cycles of the programmable wait control register as necessary.

The $\overline{\text{RESET}}$ pin is equipped with a noise rejecter circuit of analog delay to prevent malfunctioning due to noise.

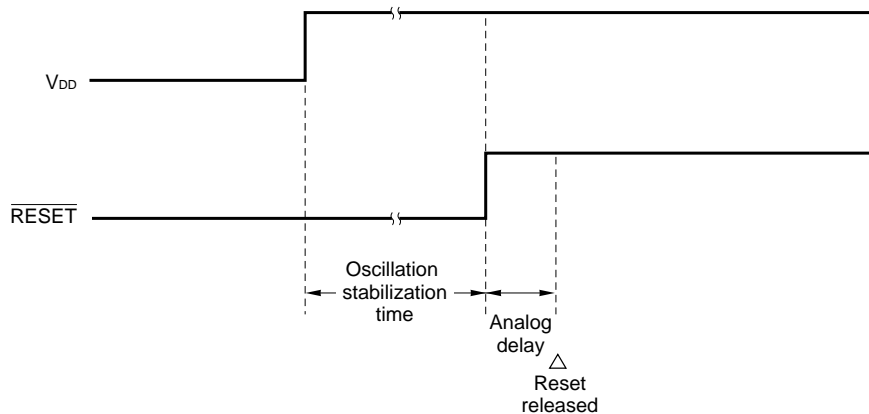
- Cautions**
1. While the $\overline{\text{RESET}}$ pin is active (low level), all the pins go into a high-impedance state (except $\overline{\text{WDTO}}$, AV_{REF} , AV_{DD} , AV_{SS} , V_{DD} , V_{SS} , X1, and X2 pins).
 2. When an external RAM is connected, do not connect a pull-up resistor to the P90/ $\overline{\text{RD}}$ and P91/ $\overline{\text{WR}}$ pins, because the P90/ $\overline{\text{RD}}$ and P91/ $\overline{\text{WR}}$ pins may go into a high-impedance state, resulting in destruction of the contents of the external RAM. In addition, signal contention occurs on the address/data bus, resulting in damage to the input/output circuit.

Figure 7-1. Accepting Reset Signal



To effect reset on when power is applied, make sure that sufficient time elapses to stabilize the oscillation after the power is applied until the reset signal is accepted, as shown in Figure 7-2.

Figure 7-2. Reset on Power Application



8. INSTRUCTION SET

Write an operand in the operand field of each instruction according to the description of the instruction (for details, refer to the Assembler Specifications). Some instructions have two or more operands. Select one of them. Uppercase characters, +, -, #, \$, !, [, and] are keywords and must be written as is.

Write an appropriate numeric value or label as immediate data. To write a label, be sure to write #, \$, !, [, or].

Table 8-1. Operand Representation and Description

| Representation | Description |
|----------------|---|
| r | R0, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15 |
| r1 | R0, R1, R2, R3, R4, R5, R6, R7 |
| r2 | C, B |
| rp | RP0, RP1, RP2, RP3, RP4, RP5, RP6, RP7 |
| rp1 | RP0, RP1, RP2, RP3, RP4, RP5, RP6, RP7 |
| rp2 | DE, HL, VP, UP |
| sfr | Special function register symbol (Refer to Table 2-1.) |
| sfrp | Special function register symbol (register that can be manipulated in 16-bit units. Refer to Table 2-1.) |
| post | RP0, RP1, RP2, RP3, RP4, RP5/PSW, RP6, RP7 (More than one symbol can be written. However, RP5 can be written only for PUSH and POP instructions, and PSW can be written only for PUSHU and POPU instructions.) |
| mem | [DE], [HL], [DE+], [HL+], [DE-], [HL-], [VP], [UP] ; register indirect mode [DE + A], [HL + A], [DE + B], [HL + B], [VP + DE], [VP + HL] ; based indexed mode [DE + byte], [HL + byte], [VP + byte], [UP + byte], [SP + byte] ; based mode word[A], word[B], word[DE], word[HL] ; indexed mode |
| saddr | FE20H-FF1FH immediate data or label |
| saddrp | FE20H-FF1EH immediate data (however, bit0 = 0) or label (manipulated in 16-bit units) |
| \$ addr16 | 0000H-FDFFH immediate data or label; relative addressing |
| ! addr16 | 0000H-FDFFH immediate data or label; immediate addressing (However, up to FFFFH can be written for MOV instruction. Only FE00H-FEFFFH can be written for MOVTBLW instruction.) |
| addr11 | 800H-FFFH immediate data or label |
| addr5 | 40H-7EH immediate data (however, bit0 = 0) ^{Note} or label |
| word | 16-bit immediate data or label |
| byte | 8-bit immediate data or label |
| bit | 3-bit immediate data or label |
| n | 3-bit immediate data (0-7) |

Note Do not access bit0 = 1 (odd address) in word units.

- Remarks**
1. rp and rp1 are the same in terms of register name that can be written but are different in code to be generated.
 2. r, r1, rp, rp1, and post can be written in absolute name (R0-R15, RP0-RP7) and function name (X, A, C, B, E, D, L, H, AX, BC, DE, HL, VP, and UP).
 3. Immediate addressing can address the entire space. Relative addressing can address only a range of -128 to +127 from the first address of the next instruction.

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | | |
|---------------------|----------|-----------------------------|----------------------|-------------------------|------|---|----|-----|----|---|
| | | | | | S | Z | AC | P/V | CY | |
| 8-bit data transfer | MOV | r1, #byte | 2 | r1 ← byte | | | | | | |
| | | saddr, #byte | 3 | (saddr) ← byte | | | | | | |
| | | sfr ^{Note} , #byte | 3 | sfr ← byte | | | | | | |
| | | r, r1 | 2 | r ← r1 | | | | | | |
| | | A, r1 | 1 | A ← r1 | | | | | | |
| | | A, saddr | 2 | A ← (saddr) | | | | | | |
| | | saddr, A | 2 | (saddr) ← A | | | | | | |
| | | saddr, saddr | 3 | (saddr) ← (saddr) | | | | | | |
| | | A, sfr | 2 | A ← sfr | | | | | | |
| | | sfr, A | 2 | sfr ← A | | | | | | |
| | | A, mem | 1-4 | A ← (mem) | | | | | | |
| | | mem, A | 1-4 | (mem) ← A | | | | | | |
| | | A, [saddrp] | 2 | A ← ((saddrp)) | | | | | | |
| | | [saddrp], A | 2 | ((saddrp)) ← A | | | | | | |
| | | A, !addr16 | 4 | A ← (addr16) | | | | | | |
| | | !addr16, A | 4 | (addr16) ← A | | | | | | |
| | | PSWL, #byte | 3 | PSW _L ← byte | | | x | x | x | x |
| | | PSWH, #byte | 3 | PSW _H ← byte | | | | | | |
| | | PSWL, A | 2 | PSW _L ← A | | | x | x | x | x |
| | | PSWH, A | 2 | PSW _H ← A | | | | | | |
| | A, PSWL | 2 | A ← PSW _L | | | | | | | |
| | A, PSWH | 2 | A ← PSW _H | | | | | | | |
| | XCH | A, r1 | 1 | A ↔ r1 | | | | | | |
| | | r, r1 | 2 | r ↔ r1 | | | | | | |
| | | A, mem | 2-4 | A ↔ (mem) | | | | | | |
| | | A, saddr | 2 | A ↔ (saddr) | | | | | | |
| | | A, sfr | 3 | A ↔ sfr | | | | | | |
| | | A, [saddrp] | 2 | A ↔ ((saddrp)) | | | | | | |
| saddr, saddr | | 3 | (saddr) ↔ (saddr) | | | | | | | |

Note When STBC or WDM is written as sfr, this instruction is treated as a dedicated instruction whose number of bytes is different from that of this instruction.

Remark For symbols in flag, refer to the table below.

| Symbol | Remarks |
|---------|------------------------------------|
| (Blank) | No change |
| 0 | Cleared to 0 |
| 1 | Set to 1 |
| × | Set/cleared according to result |
| P | P/V flag functions as parity flag |
| V | P/V flag operates as overflow flag |
| R | Value previously saved is restored |

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | |
|----------------------|-----------------|----------------|----------------------------|--------------------------------------|------------------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| 16-bit data transfer | MOVW | rp1, #word | 3 | rp1 ← word | | | | | |
| | | saddrp, #word | 4 | (saddrp) ← word | | | | | |
| | | sfrp, #word | 4 | sfrp ← word | | | | | |
| | | rp, rp1 | 2 | rp ← rp1 | | | | | |
| | | AX, saddrp | 2 | AX ← (saddrp) | | | | | |
| | | saddrp, AX | 2 | (saddrp) ← AX | | | | | |
| | | saddrp, saddrp | 3 | (saddrp) ← (saddrp) | | | | | |
| | | AX, sfrp | 2 | AX ← sfrp | | | | | |
| | | sfrp, AX | 2 | sfrp ← AX | | | | | |
| | | rp1, !addr16 | 4 | rp1 ← (addr16) | | | | | |
| | | !addr16, rp1 | 4 | (addr16) ← rp1 | | | | | |
| | | AX, mem | 2-4 | AX ← (mem) | | | | | |
| | | mem, AX | 2-4 | (mem) ← AX | | | | | |
| | XCHW | AX, saddrp | 2 | AX ↔ (saddrp) | | | | | |
| | | AX, sfrp | 3 | AX ↔ sfrp | | | | | |
| | | saddrp, saddrp | 3 | (saddrp) ↔ (saddrp) | | | | | |
| | | rp, rp1 | 2 | rp ↔ rp1 | | | | | |
| | | AX, mem | 2-4 | AX ↔ (mem) | | | | | |
| | 8-bit operation | ADD | A, #byte | 2 | A, CY ← A + byte | x | x | x | V |
| saddr, #byte | | | 3 | (saddr), CY ← (saddr) + byte | x | x | x | V | x |
| sfr, #byte | | | 4 | sfr, CY ← sfr + byte | x | x | x | V | x |
| r, r1 | | | 2 | r, CY ← r + r1 | x | x | x | V | x |
| A, saddr | | | 2 | A, CY ← A + (saddr) | x | x | x | V | x |
| A, sfr | | | 3 | A, CY ← A + sfr | x | x | x | V | x |
| saddr, saddr | | | 3 | (saddr), CY ← (saddr) + (saddr) | x | x | x | V | x |
| A, mem | | | 2-4 | A, CY ← A + (mem) | x | x | x | V | x |
| mem, A | | 2-4 | (mem), CY ← (mem) + A | x | x | x | V | x | |
| ADDC | | A, #byte | 2 | A, CY ← A + byte + CY | x | x | x | V | x |
| | | saddr, #byte | 3 | (saddr), CY ← (saddr) + byte + CY | x | x | x | V | x |
| | | sfr, #byte | 4 | sfr, CY ← sfr + byte + CY | x | x | x | V | x |
| | | r, r1 | 2 | r, CY ← r + r1 + CY | x | x | x | V | x |
| | | A, saddr | 2 | A, CY ← A + (saddr) + CY | x | x | x | V | x |
| | | A, sfr | 3 | A, CY ← A + sfr + CY | x | x | x | V | x |
| | | saddr, saddr | 3 | (saddr), CY ← (saddr) + (saddr) + CY | x | x | x | V | x |
| | | A, mem | 2-4 | A, CY ← A + (mem) + CY | x | x | x | V | x |
| | mem, A | 2-4 | (mem), CY ← (mem) + A + CY | x | x | x | V | x | |

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | |
|-----------------|-------------|--------------|------|--------------------------------------|------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| 8-bit operation | SUB | A, #byte | 2 | A, CY ← A – byte | × | × | × | V | × |
| | | saddr, #byte | 3 | (saddr), CY ← (saddr) – byte | × | × | × | V | × |
| | | sfr, #byte | 4 | sfr, CY ← sfr – byte | × | × | × | V | × |
| | | r, r1 | 2 | r, CY ← r – r1 | × | × | × | V | × |
| | | A, saddr | 2 | A, CY ← A – (saddr) | × | × | × | V | × |
| | | A, sfr | 3 | A, CY ← A – sfr | × | × | × | V | × |
| | | saddr, saddr | 3 | (saddr), CY ← (saddr) – (saddr) | × | × | × | V | × |
| | | A, mem | 2-4 | A, CY ← A – (mem) | × | × | × | V | × |
| | | mem, A | 2-4 | (mem), CY ← (mem) – A | × | × | × | V | × |
| | SUBC | A, #byte | 2 | A, CY ← A – byte – CY | × | × | × | V | × |
| | | saddr, #byte | 3 | (saddr), CY ← (saddr) – byte – CY | × | × | × | V | × |
| | | sfr, #byte | 4 | sfr, CY ← sfr – byte – CY | × | × | × | V | × |
| | | r, r1 | 2 | r, CY ← r – r1 – CY | × | × | × | V | × |
| | | A, saddr | 2 | A, CY ← A – (saddr) – CY | × | × | × | V | × |
| | | A, sfr | 3 | A, CY ← A – sfr – CY | × | × | × | V | × |
| | | saddr, saddr | 3 | (saddr), CY ← (saddr) – (saddr) – CY | × | × | × | V | × |
| | | A, mem | 2-4 | A, CY ← A – (mem) – CY | × | × | × | V | × |
| | | mem, A | 2-4 | (mem), CY ← (mem) – A – CY | × | × | × | V | × |
| | AND | A, #byte | 2 | A ← A ∧ byte | × | × | | P | |
| | | saddr, #byte | 3 | (saddr) ← (saddr) ∧ byte | × | × | | P | |
| | | sfr, #byte | 4 | sfr ← sfr ∧ byte | × | × | | P | |
| | | r, r1 | 2 | r ← r ∧ r1 | × | × | | P | |
| | | A, saddr | 2 | A ← A ∧ (saddr) | × | × | | P | |
| | | A, sfr | 3 | A ← A ∧ sfr | × | × | | P | |
| | | saddr, saddr | 3 | (saddr) ← (saddr) ∧ (saddr) | × | × | | P | |
| | | A, mem | 2-4 | A ← A ∧ (mem) | × | × | | P | |
| | | mem, A | 2-4 | (mem) ← (mem) ∧ A | × | × | | P | |

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | |
|-----------------|------------|--------------|------|--|------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| 8-bit operation | OR | A, #byte | 2 | $A \leftarrow A \vee \text{byte}$ | x | x | | P | |
| | | saddr, #byte | 3 | $(\text{saddr}) \leftarrow (\text{saddr}) \vee \text{byte}$ | x | x | | P | |
| | | sfr, #byte | 4 | $\text{sfr} \leftarrow \text{sfr} \vee \text{byte}$ | x | x | | P | |
| | | r, r1 | 2 | $r \leftarrow r \vee r1$ | x | x | | P | |
| | | A, saddr | 2 | $A \leftarrow A \vee (\text{saddr})$ | x | x | | P | |
| | | A, sfr | 3 | $A \leftarrow A \vee \text{sfr}$ | x | x | | P | |
| | | saddr, saddr | 3 | $(\text{saddr}) \leftarrow (\text{saddr}) \vee (\text{saddr})$ | x | x | | P | |
| | | A, mem | 2-4 | $A \leftarrow A \vee (\text{mem})$ | x | x | | P | |
| | | mem, A | 2-4 | $(\text{mem}) \leftarrow (\text{mem}) / A$ | x | x | | P | |
| | XOR | A, #byte | 2 | $A \leftarrow A \nabla \text{byte}$ | x | x | | P | |
| | | saddr, #byte | 3 | $(\text{saddr}) \leftarrow (\text{saddr}) \nabla \text{byte}$ | x | x | | P | |
| | | sfr, #byte | 4 | $\text{sfr} \leftarrow \text{sfr} \nabla \text{byte}$ | x | x | | P | |
| | | r, r1 | 2 | $r \leftarrow r \nabla r1$ | x | x | | P | |
| | | A, saddr | 2 | $A \leftarrow A \nabla (\text{saddr})$ | x | x | | P | |
| | | A, sfr | 3 | $A \leftarrow A \nabla \text{sfr}$ | x | x | | P | |
| | | saddr, saddr | 3 | $(\text{saddr}) \leftarrow (\text{saddr}) \nabla (\text{saddr})$ | x | x | | P | |
| | | A, mem | 2-4 | $A \leftarrow A \nabla (\text{mem})$ | x | x | | P | |
| | | mem, A | 2-4 | $(\text{mem}) \leftarrow (\text{mem}) \nabla A$ | x | x | | P | |
| | CMP | A, #byte | 2 | $A - \text{byte}$ | x | x | x | V | x |
| | | saddr, #byte | 3 | $(\text{saddr}) - \text{byte}$ | x | x | x | V | x |
| | | sfr, #byte | 4 | $\text{sfr} - \text{byte}$ | x | x | x | V | x |
| | | r, r1 | 2 | $r - r1$ | x | x | x | V | x |
| | | A, saddr | 2 | $A - (\text{saddr})$ | x | x | x | V | x |
| | | A, sfr | 3 | $A - \text{sfr}$ | x | x | x | V | x |
| | | saddr, saddr | 3 | $(\text{saddr}) - (\text{saddr})$ | x | x | x | V | x |
| | | A, mem | 2-4 | $A - (\text{mem})$ | x | x | x | V | x |
| | | mem, A | 2-4 | $(\text{mem}) - A$ | x | x | x | V | x |

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | |
|---|--------------|------------------|------|--|------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| 16-bit operation | ADDW | AX, #word | 3 | AX, CY ← AX + word | x | x | x | V | x |
| | | saddrp, #word | 4 | (saddrp), CY ← (saddrp) + word | x | x | x | V | x |
| | | sfrp, #word | 5 | sfrp, CY ← sfrp + word | x | x | x | V | x |
| | | rp, rp1 | 2 | rp, CY ← rp + rp1 | x | x | x | V | x |
| | | AX, saddrp | 2 | AX, CY ← AX + (saddrp) | x | x | x | V | x |
| | | AX, sfrp | 3 | AX, CY ← AX + sfrp | x | x | x | V | x |
| | | saddrp, saddrp | 3 | (saddrp), CY ← (saddrp) + (saddrp) | x | x | x | V | x |
| | SUBW | AX, #word | 3 | AX, CY ← AX – word | x | x | x | V | x |
| | | saddrp, #word | 4 | (saddrp), CY ← (saddrp) – word | x | x | x | V | x |
| | | sfrp, #word | 5 | sfrp, CY ← sfrp – word | x | x | x | V | x |
| | | rp, rp1 | 2 | rp, CY ← rp – rp1 | x | x | x | V | x |
| | | AX, saddrp | 2 | AX, CY ← AX – (saddrp) | x | x | x | V | x |
| | | AX, sfrp | 3 | AX, CY ← AX – sfrp | x | x | x | V | x |
| | | saddrp, saddrp | 3 | (saddrp), CY ← (saddrp) – (saddrp) | x | x | x | V | x |
| | CMPW | AX, #word | 3 | AX – word | x | x | x | V | x |
| | | saddrp, #word | 4 | (saddrp) – word | x | x | x | V | x |
| | | sfrp, #word | 5 | sfrp – word | x | x | x | V | x |
| | | rp, rp1 | 2 | rp – rp1 | x | x | x | V | x |
| | | AX, saddrp | 2 | AX – (saddrp) | x | x | x | V | x |
| | | AX, sfrp | 3 | AX – sfrp | x | x | x | V | x |
| | | saddrp, saddrp | 3 | (saddrp) – (saddrp) | x | x | x | V | x |
| Multiplication /division | MULU | r1 | 2 | AX ← AX × r1 | | | | | |
| | DIVUW | r1 | 2 | AX (quotient), r1 (remainder) ← AX ÷ r1 | | | | | |
| | MULUW | rp1 | 2 | AX (high-order 16 bits), rp1 (low-order 16 bits) ← AX × rp1 | | | | | |
| | DIVUX | rp1 | 2 | AXDE (quotient), rp1 (remainder) ← AXDE ÷ rp1 | | | | | |
| Signed multiplication | MULW | rp1 | 2 | AX (high-order 16 bits), rp1 (low-order 16 bits) ← AX × rp1 | | | | | |
| Sum-of-products operation | MACW | n | 3 | AXDE ← (B) × (C) + AXDE B ← B + 2, C ← C + 2, n ← n – 1 End if n = 0 or P/V = 1 | x | x | x | V | x |
| Sum-of-products operation with saturation | MACSW | n | 3 | AXDE ← (B) × (C) + AXDE B ← B + 2, C ← C + 2, n ← n – 1 if overflow (P/V = 1) then AXDE ← 7FFFFFFFH if underflow (P/V = 1) then AXDE ← 80000000H end if n = 0 or P/V = 1 | x | x | x | V | x |
| Relative operation | SACW | [DE +], [HL +] | 4 | AX ← AX + (DE) – (HL) DE ← DE + 2 HL ← HL + 2 C ← C – 1 end if C = 0 or cy = 1 | x | x | x | V | x |

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | | |
|---------------------|----------------|------------|---|---|------|---|----|-----|----|--|
| | | | | | S | Z | AC | P/V | CY | |
| Table shift | MOVTBLW | !addr16, n | 4 | (addr16 + 2) ← (addr16), n ← n - 1 addr16 ← addr16 - 2, End if n = 0 | | | | | | |
| Increment/decrement | INC | r1 | 1 | r1 ← r1 + 1 | × | × | × | V | | |
| | | saddr | 2 | (saddr) ← (saddr) + 1 | × | × | × | V | | |
| | DEC | r1 | 1 | r1 ← r1 - 1 | × | × | × | V | | |
| | | saddr | 2 | (saddr) ← (saddr) - 1 | × | × | × | V | | |
| | INCW | rp2 | 1 | rp2 ← rp2 + 1 | | | | | | |
| | | saddrp | 3 | (saddrp) ← (saddrp) + 1 | | | | | | |
| DECW | rp2 | 1 | rp2 ← rp2 - 1 | | | | | | | |
| | saddrp | 3 | (saddrp) ← (saddrp) - 1 | | | | | | | |
| Shift rotate | ROR | r1, n | 2 | (CY, r17 ← r10, r1 _{m-1} ← r1 _m) × n times | | | | P | × | |
| | ROL | r1, n | 2 | (CY, r10 ← r17, r1 _{m+1} ← r1 _m) × n times | | | | P | × | |
| | RORC | r1, n | 2 | (CY ← r10, r17 ← CY, r1 _{m-1} ← r1 _m) × n times | | | | P | × | |
| | ROLC | r1, n | 2 | (CY ← r17, r10 ← CY, r1 _{m+1} ← r1 _m) × n times | | | | P | × | |
| | SHR | r1, n | 2 | (CY ← r10, r17 ← 0, r1 _{m-1} ← r1 _m) × n times | × | × | 0 | P | × | |
| | SHL | r1, n | 2 | (CY ← r17, r10 ← 0, r1 _{m+1} ← r1 _m) × n times | × | × | 0 | P | × | |
| | SHRW | rp1, n | 2 | (CY ← rp1 ₀ , rp1 ₁₅ ← 0, rp1 _{m-1} ← rp1 _m) × n times | × | × | 0 | P | × | |
| | SHLW | rp1, n | 2 | (CY ← rp1 ₁₅ , rp1 ₀ ← 0, rp1 _{m+1} ← rp1 _m) × n times | × | × | 0 | P | × | |
| | ROR4 | [rp1] | 2 | A ₃₋₀ ← (rp1) ₃₋₀ , (rp1) ₇₋₄ ← A ₃₋₀ , (rp1) ₃₋₀ ← (rp1) ₇₋₄ | | | | | | |
| ROL4 | [rp1] | 2 | A ₃₋₀ ← (rp1) ₇₋₄ , (rp1) ₃₋₀ ← A ₃₋₀ , (rp1) ₇₋₄ ← (rp1) ₃₋₀ | | | | | | | |
| BCD adjustment | ADJBA | | 2 | Decimal Adjust Accumelator | × | × | 0 | P | × | |
| | ADJBS | | | | | | | | | |
| Data conversion | CVTBW | | 1 | When A ₇ = 0, X ← A, A ← 00H When A ₇ = 1, X ← A, A ← FFH | | | | | | |

- Remarks**
- n of the shift rotate instruction indicates the number of times the shift rotate instruction is executed.
 - The address of the table shift instruction ranges from FE00H to FEFFH.

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | | | |
|------------------|----------|----------------|------|---|------|---|----|-----|----|---|--|
| | | | | | S | Z | AC | P/V | CY | | |
| Bit manipulation | MOV1 | CY, saddr.bit | 3 | $CY \leftarrow (\text{saddr.bit})$ | | | | | | x | |
| | | CY, sfr.bit | 3 | $CY \leftarrow \text{sfr.bit}$ | | | | | | x | |
| | | CY, A.bit | 2 | $CY \leftarrow \text{A.bit}$ | | | | | | x | |
| | | CY, X.bit | 2 | $CY \leftarrow \text{X.bit}$ | | | | | | x | |
| | | CY, PSWH.bit | 2 | $CY \leftarrow \text{PSWH.bit}$ | | | | | | x | |
| | | CY, PSWL.bit | 2 | $CY \leftarrow \text{PSWL.bit}$ | | | | | | x | |
| | | saddr.bit, CY | 3 | $(\text{saddr.bit}) \leftarrow CY$ | | | | | | | |
| | | sfr.bit, CY | 3 | $\text{sfr.bit} \leftarrow CY$ | | | | | | | |
| | | A.bit, CY | 2 | $\text{A.bit} \leftarrow CY$ | | | | | | | |
| | | X.bit, CY | 2 | $\text{X.bit} \leftarrow CY$ | | | | | | | |
| | | PSWH.bit, CY | 2 | $\text{PSWH.bit} \leftarrow CY$ | | | | | | | |
| | | PSWL.bit, CY | 2 | $\text{PSWL.bit} \leftarrow CY$ | | | x | x | x | x | |
| | AND1 | CY, saddr.bit | 3 | $CY \leftarrow CY \wedge (\text{saddr.bit})$ | | | | | | x | |
| | | CY, /saddr.bit | 3 | $CY \leftarrow CY \wedge \overline{(\text{saddr.bit})}$ | | | | | | x | |
| | | CY, sfr.bit | 3 | $CY \leftarrow CY \wedge \text{sfr.bit}$ | | | | | | x | |
| | | CY, /sfr.bit | 3 | $CY \leftarrow CY \wedge \overline{\text{sfr.bit}}$ | | | | | | x | |
| | | CY, A.bit | 2 | $CY \leftarrow CY \wedge \text{A.bit}$ | | | | | | x | |
| | | CY, /A.bit | 2 | $CY \leftarrow CY \wedge \overline{\text{A.bit}}$ | | | | | | x | |
| | | CY, X.bit | 2 | $CY \leftarrow CY \wedge \text{X.bit}$ | | | | | | x | |
| | | CY, /X.bit | 2 | $CY \leftarrow CY \wedge \overline{\text{X.bit}}$ | | | | | | x | |
| | | CY, PSWH.bit | 2 | $CY \leftarrow CY \wedge \text{PSWH.bit}$ | | | | | | x | |
| | | CY, /PSWH.bit | 2 | $CY \leftarrow CY \wedge \overline{\text{PSWH.bit}}$ | | | | | | x | |
| | | CY, PSWL.bit | 2 | $CY \leftarrow CY \wedge \text{PSWL.bit}$ | | | | | | x | |
| | | CY, /PSWL.bit | 2 | $CY \leftarrow CY \wedge \overline{\text{PSWL.bit}}$ | | | | | | x | |
| | OR1 | CY, saddr.bit | 3 | $CY \leftarrow CY \vee (\text{saddr.bit})$ | | | | | | x | |
| | | CY, /saddr.bit | 3 | $CY \leftarrow CY \vee \overline{(\text{saddr.bit})}$ | | | | | | x | |
| | | CY, sfr.bit | 3 | $CY \leftarrow CY \vee \text{sfr.bit}$ | | | | | | x | |
| | | CY, /sfr.bit | 3 | $CY \leftarrow CY \vee \overline{\text{sfr.bit}}$ | | | | | | x | |
| | | CY, A.bit | 2 | $CY \leftarrow CY \vee \text{A.bit}$ | | | | | | x | |
| | | CY, /A.bit | 2 | $CY \leftarrow CY \vee \overline{\text{A.bit}}$ | | | | | | x | |
| | | CY, X.bit | 2 | $CY \leftarrow CY \vee \text{X.bit}$ | | | | | | x | |
| | | CY, /X.bit | 2 | $CY \leftarrow CY \vee \overline{\text{X.bit}}$ | | | | | | x | |
| | | CY, PSWH.bit | 2 | $CY \leftarrow CY \vee \text{PSWH.bit}$ | | | | | | x | |
| | | CY, /PSWH.bit | 2 | $CY \leftarrow CY \vee \overline{\text{PSWH.bit}}$ | | | | | | x | |
| | | CY, PSWL.bit | 2 | $CY \leftarrow CY \vee \text{PSWL.bit}$ | | | | | | x | |
| | | CY, /PSWL.bit | 2 | $CY \leftarrow CY \vee \overline{\text{PSWL.bit}}$ | | | | | | x | |

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | | |
|------------------|-------------|---------------|------|---|------|---|----|-----|----|---|
| | | | | | S | Z | AC | P/V | CY | |
| Bit manipulation | XOR1 | CY, saddr.bit | 3 | $CY \leftarrow CY \oplus (\text{saddr.bit})$ | | | | | | × |
| | | CY, sfr.bit | 3 | $CY \leftarrow CY \oplus \text{sfr.bit}$ | | | | | | × |
| | | CY, A.bit | 2 | $CY \leftarrow CY \oplus \text{A.bit}$ | | | | | | × |
| | | CY, X.bit | 2 | $CY \leftarrow CY \oplus \text{X.bit}$ | | | | | | × |
| | | CY, PSWH.bit | 2 | $CY \leftarrow CY \oplus \text{PSWH.bit}$ | | | | | | × |
| | | CY, PSWL.bit | 2 | $CY \leftarrow CY \oplus \text{PSWL.bit}$ | | | | | | × |
| | SET1 | saddr.bit | 2 | $(\text{saddr.bit}) \leftarrow 1$ | | | | | | |
| | | sfr.bit | 3 | $\text{sfr.bit} \leftarrow 1$ | | | | | | |
| | | A.bit | 2 | $\text{A.bit} \leftarrow 1$ | | | | | | |
| | | X.bit | 2 | $\text{X.bit} \leftarrow 1$ | | | | | | |
| | | PSWH.bit | 2 | $\text{PSWH.bit} \leftarrow 1$ | | | | | | |
| | | PSWL.bit | 2 | $\text{PSWL.bit} \leftarrow 1$ | | × | × | × | × | × |
| | CLR1 | saddr.bit | 2 | $(\text{saddr.bit}) \leftarrow 0$ | | | | | | |
| | | sfr.bit | 3 | $\text{sfr.bit} \leftarrow 0$ | | | | | | |
| | | A.bit | 2 | $\text{A.bit} \leftarrow 0$ | | | | | | |
| | | X.bit | 2 | $\text{X.bit} \leftarrow 0$ | | | | | | |
| | | PSWH.bit | 2 | $\text{PSWH.bit} \leftarrow 0$ | | | | | | |
| | | PSWL.bit | 2 | $\text{PSWL.bit} \leftarrow 0$ | | × | × | × | × | × |
| | NOT1 | saddr.bit | 3 | $(\text{saddr.bit}) \leftarrow \overline{(\text{saddr.bit})}$ | | | | | | |
| | | sfr.bit | 3 | $\text{sfr.bit} \leftarrow \overline{\text{sfr.bit}}$ | | | | | | |
| | | A.bit | 2 | $\text{A.bit} \leftarrow \overline{\text{A.bit}}$ | | | | | | |
| | | X.bit | 2 | $\text{X.bit} \leftarrow \overline{\text{X.bit}}$ | | | | | | |
| | | PSWH.bit | 2 | $\text{PSWH.bit} \leftarrow \overline{\text{PSWH.bit}}$ | | | | | | |
| | | PSWL.bit | 2 | $\text{PSWL.bit} \leftarrow \overline{\text{PSWL.bit}}$ | | × | × | × | × | × |
| | SET1 | CY | 1 | $CY \leftarrow 1$ | | | | | | 1 |
| | CLR1 | CY | 1 | $CY \leftarrow 0$ | | | | | | 0 |
| | NOT1 | CY | 1 | $CY \leftarrow \overline{CY}$ | | | | | | × |

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | |
|--------------------|--------------|-----------|--|---|------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| Call/return | CALL | laddr16 | 3 | $(SP - 1) \leftarrow (PC + 3)_H, (SP - 2) \leftarrow (PC + 3)_L,$ $PC \leftarrow \text{addr16}, SP \leftarrow SP - 2$ | | | | | |
| | CALLF | laddr11 | 2 | $(SP - 1) \leftarrow (PC + 2)_H, (SP - 2) \leftarrow (PC + 2)_L,$ $PC_{15-11} \leftarrow 00001, PC_{10-0} \leftarrow \text{addr11}, SP \leftarrow SP - 2$ | | | | | |
| | CALLT | [addr5] | 1 | $(SP - 1) \leftarrow (PC + 1)_H, (SP - 2) \leftarrow (PC + 1)_L,$ $PC_H \leftarrow (TPF, 00000000, \text{addr5} + 1),$ $PC_L \leftarrow (TPF, 00000000, \text{addr5}), SP \leftarrow SP - 2$ | | | | | |
| | CALL | rp1 | 2 | $(SP - 1) \leftarrow (PC + 2)_H, (SP - 2) \leftarrow (PC + 2)_L,$ $PC_H \leftarrow rp1_H, PC_L \leftarrow rp1_L, SP \leftarrow SP - 2$ | | | | | |
| | | [rp1] | 2 | $(SP - 1) \leftarrow (PC + 2)_H, (SP - 2) \leftarrow (PC + 2)_L,$ $PC_H \leftarrow (rp1 + 1), PC_L \leftarrow (rp1), SP \leftarrow SP - 2$ | | | | | |
| | BRK | | 1 | $(SP - 1) \leftarrow PSW_H, (SP - 2) \leftarrow PSW_L$ $(SP - 3) \leftarrow (PC + 1)_H, (SP - 4) \leftarrow (PC + 1)_L,$ $PC_L \leftarrow (003EH), PC_H \leftarrow (003FH),$ $SP \leftarrow SP - 4, IE \leftarrow 0$ | | | | | |
| | RET | | 1 | $PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1), SP \leftarrow SP + 2$ | | | | | |
| | RETB | | 1 | $PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1)$ $PSW_L \leftarrow (SP + 2), PSW_H \leftarrow (SP + 3)$ $SP \leftarrow SP + 4$ | R | R | R | R | R |
| RETI | | 1 | $PC_L \leftarrow (SP), PC_H \leftarrow (SP + 1)$ $PSW_L \leftarrow (SP + 2), PSW_H \leftarrow (SP + 3)$ $SP \leftarrow SP + 4$ | R | R | R | R | R | |
| Stack manipulation | PUSH | sfrp | 3 | $(SP - 1) \leftarrow sfr_H$ $(SP - 2) \leftarrow sfr_L$ $SP \leftarrow SP - 2$ | | | | | |
| | | post | 2 | $\{(SP - 1) \leftarrow \text{post}_H, (SP - 2) \leftarrow \text{post}_L, SP \leftarrow SP - 2\} \times n \text{ times}$ | | | | | |
| | | PSW | 1 | $(SP - 1) \leftarrow PSW_H, (SP - 2) \leftarrow PSW_L, SP \leftarrow SP - 2$ | | | | | |
| | PUSHU | post | 2 | $\{(UP - 1) \leftarrow \text{post}_H, (UP - 2) \leftarrow \text{post}_L, UP \leftarrow UP - 2\} \times n \text{ times}$ | | | | | |
| | POP | sfrp | 3 | $sfr_L \leftarrow (SP)$ $sfr_H \leftarrow (SP + 1)$ $SP \leftarrow SP + 2$ | | | | | |
| | | post | 2 | $\{\text{post}_L \leftarrow (SP), \text{post}_H \leftarrow (SP + 1), SP \leftarrow SP + 2\} \times n \text{ times}$ | | | | | |
| | | PSW | 1 | $PSW_L \leftarrow (SP), PSW_H \leftarrow (SP + 1), SP \leftarrow SP + 2$ | R | R | R | R | R |
| | POPU | post | 2 | $\{\text{post}_L \leftarrow (UP), \text{post}_H \leftarrow (UP + 1), UP \leftarrow UP + 2\} \times n \text{ times}$ | | | | | |
| | MOVW | SP, #word | 4 | $SP \leftarrow \text{word}$ | | | | | |
| | | SP, AX | 2 | $SP \leftarrow AX$ | | | | | |
| | | AX, SP | 2 | $AX \leftarrow SP$ | | | | | |
| INCW | SP | 2 | $SP \leftarrow SP + 1$ | | | | | | |
| DECW | SP | 2 | $SP \leftarrow SP - 1$ | | | | | | |

Remark n of the stack manipulation instruction is the number of registers written as post.

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | |
|----------------------|--------------|---------------------|--------------------------------------|---|------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| Special | CHKL | sfr | 3 | (pin level) ⇄ (signal level before output buffer) | × | × | | P | |
| | CHKLA | sfr | 3 | A ← (pin level) ⇄ (signal level before output buffer) | × | × | | P | |
| Unconditional branch | BR | !addr16 | 3 | PC ← addr16 | | | | | |
| | | rp1 | 2 | PC _H ← rp1 _H , PC _L ← rp1 _L | | | | | |
| | | [rp1] | 2 | PC _H ← (rp1 + 1), PC _L ← (rp1) | | | | | |
| | | \$addr16 | 2 | PC ← PC + 2 + jdisp8 | | | | | |
| Conditional branch | BC | \$addr16 | 2 | PC ← PC + 2 + jdisp8 if CY = 1 | | | | | |
| | BL | | 2 | PC ← PC + 2 + jdisp8 if CY = 0 | | | | | |
| | BNC | \$addr16 | 2 | PC ← PC + 2 + jdisp8 if CY = 0 | | | | | |
| | BNL | | 2 | PC ← PC + 2 + jdisp8 if CY = 1 | | | | | |
| | BZ | \$addr16 | 2 | PC ← PC + 2 + jdisp8 if Z = 1 | | | | | |
| | BE | | 2 | PC ← PC + 2 + jdisp8 if Z = 0 | | | | | |
| | BNZ | \$addr16 | 2 | PC ← PC + 2 + jdisp8 if Z = 0 | | | | | |
| | BNE | | 2 | PC ← PC + 2 + jdisp8 if Z = 1 | | | | | |
| | BV | \$addr16 | 2 | PC ← PC + 2 + jdisp8 if P/V = 1 | | | | | |
| | BPE | | 2 | PC ← PC + 2 + jdisp8 if P/V = 0 | | | | | |
| | BNV | \$addr16 | 2 | PC ← PC + 2 + jdisp8 if P/V = 0 | | | | | |
| | BPO | | 2 | PC ← PC + 2 + jdisp8 if P/V = 1 | | | | | |
| | BN | \$addr16 | 2 | PC ← PC + 2 + jdisp8 if S = 1 | | | | | |
| | BP | \$addr16 | 2 | PC ← PC + 2 + jdisp8 if S = 0 | | | | | |
| | BGT | \$addr16 | 3 | PC ← PC + 3 + jdisp8 if (P/V ⇄ S) / Z = 0 | | | | | |
| | BGE | \$addr16 | 3 | PC ← PC + 3 + jdisp8 if P/V ⇄ S = 0 | | | | | |
| | BLT | \$addr16 | 3 | PC ← PC + 3 + jdisp8 if P/V ⇄ S = 1 | | | | | |
| | BLE | \$addr16 | 3 | PC ← PC + 3 + jdisp8 if (P/V ⇄ S) / Z = 1 | | | | | |
| | BH | \$addr16 | 3 | PC ← PC + 3 + jdisp8 if Z ∨ CY = 0 | | | | | |
| | BNH | \$addr16 | 3 | PC ← PC + 3 + jdisp8 if Z ∨ CY = 1 | | | | | |
| | BT | saddr.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if (saddr.bit) = 1 | | | | | |
| | | sfr.bit, \$addr16 | 4 | PC ← PC + 4 + jdisp8 if sfr.bit = 1 | | | | | |
| | | A.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if A.bit = 1 | | | | | |
| | | X.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if X.bit = 1 | | | | | |
| | | PSWH.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if PSWH.bit = 1 | | | | | |
| | | PSWL.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if PSWL.bit = 1 | | | | | |
| | BF | saddr.bit, \$addr16 | 4 | PC ← PC + 4 + jdisp8 if (saddr.bit) = 0 | | | | | |
| | | sfr.bit, \$addr16 | 4 | PC ← PC + 4 + jdisp8 if sfr.bit = 0 | | | | | |
| A.bit, \$addr16 | | 3 | PC ← PC + 3 + jdisp8 if A.bit = 0 | | | | | | |
| X.bit, \$addr16 | | 3 | PC ← PC + 3 + jdisp8 if X.bit = 0 | | | | | | |
| PSWH.bit, \$addr16 | | 3 | PC ← PC + 3 + jdisp8 if PSWH.bit = 0 | | | | | | |
| PSWL.bit, \$addr16 | | 3 | PC ← PC + 3 + jdisp8 if PSWL.bit = 0 | | | | | | |

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | |
|--------------------|-----------------|---------------------|--|--|------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| Conditional branch | BTCLR | saddr.bit, \$addr16 | 4 | PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit) | | | | | |
| | | sfr.bit, \$addr16 | 4 | PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit | | | | | |
| | | A.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit | | | | | |
| | | X.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if X.bit = 1 then reset X.bit | | | | | |
| | | PSWH.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if PSWH.bit = 1 then reset PSWH.bit | | | | | |
| | | PSWL.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if PSWL.bit = 1 then reset PSWL.bit | x | x | x | x | x |
| | BFSET | saddr.bit, \$addr16 | 4 | PC ← PC + 4 + jdisp8 if (saddr.bit) = 0 then set (saddr.bit) | | | | | |
| | | sfr.bit, \$addr16 | 4 | PC ← PC + 4 + jdisp8 if sfr.bit = 0 then set sfr.bit | | | | | |
| | | A.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if A.bit = 0 then set A.bit | | | | | |
| | | X.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if X.bit = 0 then set X.bit | | | | | |
| | | PSWH.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if PSWH.bit = 0 then set PSWH.bit | | | | | |
| | | PSWL.bit, \$addr16 | 3 | PC ← PC + 3 + jdisp8 if PSWL.bit = 0 then set PSWL.bit | x | x | x | x | x |
| DBNZ | r2, \$addr16 | 2 | r2 ← r2 - 1, then PC ← PC + 2 + jdisp8 if 2 ≠ 0 | | | | | | |
| | saddr, \$addr16 | 3 | (saddr) ← (saddr) - 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0 | | | | | | |
| Context switching | BRKCS | R _{Bn} | 2 | PC _H ↔ R5, PC _L ↔ R4, R7 ← PSWH, R6 ← PSWL, ← RBS2 - 0 ← n, RSS ← 0, IE ← 0 | | | | | |
| | RETCS | !addr16 | 3 | PC _H ← R5, PC _L ← R4, R5, R4 ← addr16 PSWH ← R7, PSWL ← R6 | R | R | R | R | R |
| | RETCSB | !addr16 | 4 | PC _H ← R5, PC _L ← R4, R5, R4 ← addr16 PSWH ← R7, PSWL ← R6 | R | R | R | R | R |

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | |
|--------------|----------------|--------------|------|--|------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| String | MOVM | [DE+], A | 2 | (DE+) ← A, C ← C - 1 End if C = 0 | | | | | |
| | | [DE-], A | 2 | (DE-) ← A, C ← C - 1 End if C = 0 | | | | | |
| | MOVBK | [DE+], [HL+] | 2 | (DE+) ← (HL+), C ← C - 1 End if C = 0 | | | | | |
| | | [DE-], [HL-] | 2 | (DE-) ← (HL-), C ← C - 1 End if C = 0 | | | | | |
| | XCHM | [DE+], A | 2 | (DE+) ↔ A, C ← C - 1 End if C = 0 | | | | | |
| | | [DE-], A | 2 | (DE-) ↔ A, C ← C - 1 End if C = 0 | | | | | |
| | XCHBK | [DE+], [HL+] | 2 | (DE+) ↔ (HL+), C ← C - 1 End if C = 0 | | | | | |
| | | [DE-], [HL-] | 2 | (DE-) ↔ (HL-), C ← C - 1 End if C = 0 | | | | | |
| | CMPME | [DE+], A | 2 | (DE+) - A, C ← C - 1 End if C = 0 or Z = 0 | × | × | × | V | × |
| | | [DE-], A | 2 | (DE-) - A, C ← C - 1 End if C = 0 or Z = 0 | × | × | × | V | × |
| | CMPBKE | [DE+], [HL+] | 2 | (DE+) - (HL+), C ← C - 1 End if C = 0 or Z = 0 | × | × | × | V | × |
| | | [DE-], [HL-] | 2 | (DE-) - (HL-), C ← C - 1 End if C = 0 or Z = 0 | × | × | × | V | × |
| | CMPMNE | [DE+], A | 2 | (DE+) - A, C ← C - 1 End if C = 0 or Z = 1 | × | × | × | V | × |
| | | [DE-], A | 2 | (DE-) - A, C ← C - 1 End if C = 0 or Z = 1 | × | × | × | V | × |
| | CMPBKNE | [DE+], [HL+] | 2 | (DE+) - (HL+), C ← C - 1 End if C = 0 or Z = 1 | × | × | × | V | × |
| | | [DE-], [HL-] | 2 | (DE-) - (HL-), C ← C - 1 End if C = 0 or Z = 1 | × | × | × | V | × |
| | CMPMC | [DE+], A | 2 | (DE+) - A, C ← C - 1 End if C = 0 or CY = 0 | × | × | × | V | × |
| | | [DE-], A | 2 | (DE-) - A, C ← C - 1 End if C = 0 or CY = 0 | × | × | × | V | × |
| | CMPBKC | [DE+], [HL+] | 2 | (DE+) - (HL+), C ← C - 1 End if C = 0 or CY = 0 | × | × | × | V | × |
| | | [DE-], [HL-] | 2 | (DE-) - (HL-), C ← C - 1 End if C = 0 or CY = 0 | × | × | × | V | × |

| Instructions | Mnemonic | Operand | Byte | Operation | Flag | | | | |
|--------------|----------|--------------|----------------------------|--|------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| String | CMPMNC | [DE+], A | 2 | (DE+) - A, C ← C - 1 End if C = 0 or CY = 1 | x | x | x | V | x |
| | | [DE-], A | 2 | (DE-) - A, C ← C - 1 End if C = 0 or CY = 1 | x | x | x | V | x |
| | CMPBKNC | [DE+], [HL+] | 2 | (DE+) - (HL+), C ← C - 1 End if C = 0 or CY = 1 | x | x | x | V | x |
| | | [DE-], [HL-] | 2 | (DE-) - (HL-), C ← C - 1 End if C = 0 or CY = 1 | x | x | x | V | x |
| CPU control | MOV | STBC, #byte | 4 | STBC ← byte ^{Note} | | | | | |
| | | WDM, #byte | 4 | WDM ← byte ^{Note} | | | | | |
| | SWRS | | 1 | RSS ← \overline{RSS} | | | | | |
| | SEL | RBn | 2 | RBS2 - 0 ← n, RSS ← 0 | | | | | |
| | | RBn, ALT | 2 | RBS2 - 0 ← n, RSS ← 1 | | | | | |
| | NOP | | 1 | No Operation | | | | | |
| | EI | | 1 | IE ← 1 (Enable Interrupt) | | | | | |
| DI | | 1 | IE ← 0 (Disable Interrupt) | | | | | | |

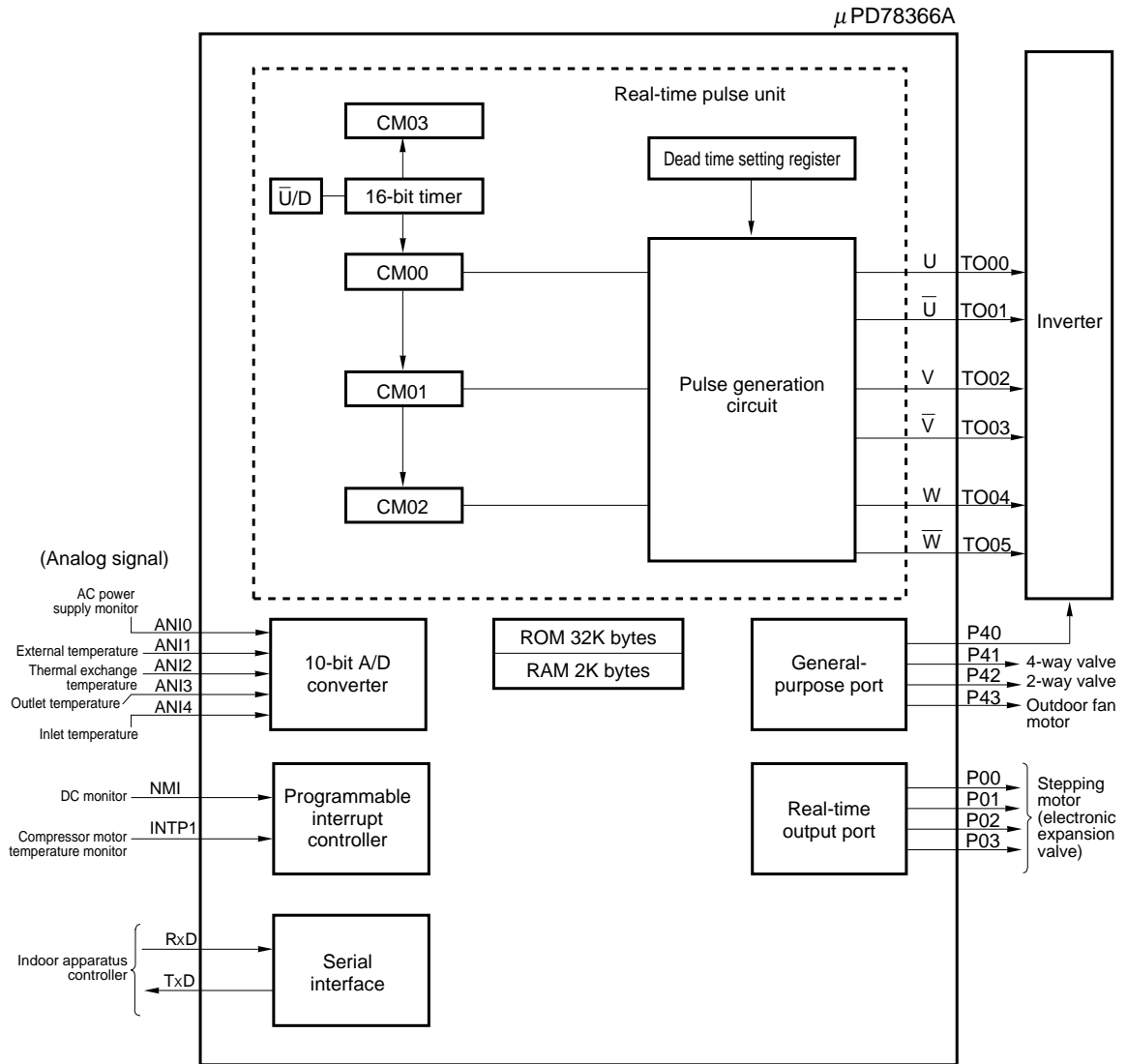
Note If the op code of the STBC register and WDM register manipulation instructions is wrong, an op code trap interrupt occurs.

Operation on trap:

(SP - 1) ← PSW_H, (SP - 2) ← PSW_L,
 (SP - 3) ← (PC - 4)_H, (SP - 4) ← (PC - 4)_L,
 PC_L ← (003CH), PC_H ← (003DH),
 SP ← SP - 4, IE ← 0

9. EXAMPLE OF SYSTEM CONFIGURATION

Controlling outdoor apparatus of inverter air conditioner



10. ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings (T_A = 25 °C)

| Parameter | Symbol | Test conditions | Rating | Unit |
|---------------------------------------|-------------------|--|--|------|
| Power supply voltage | V _{DD} | | -0.5 to +7.0 | V |
| | AV _{DD} | | -0.5 to V _{DD} + 0.5 | V |
| | AV _{SS} | | -0.5 to +0.5 | V |
| Input voltage | V _i | Pins other than P70/ANI0-P77/ANI7 | -0.5 to V _{DD} + 0.5 | V |
| Output voltage | V _o | | -0.5 to V _{DD} + 0.5 | V |
| Low-level output current | I _{OL} | Note | 20 | mA |
| | | Output pins other than those in the note | 4.0 | mA |
| | | Total of all output pins | 200 | mA |
| High-level output current | I _{OH} | All output pins | -3.0 | mA |
| | | Total of all output pins | -25 | mA |
| Analog input voltage | V _{IAN} | P70/ANI0-P77/ANI7 pins | AV _{SS} - 0.5 to AV _{DD} + 0.5 | V |
| A/D converter reference input voltage | AV _{REF} | | AV _{SS} - 0.5 to AV _{DD} + 0.5 | V |
| Operating ambient temperature | T _A | | -40 to +85 | °C |
| Storage temperature | T _{stg} | | -60 to +150 | °C |

Note P00/RTP0-P03/RTP3, P04/PWM0, P05/TCUD/PWM1, P06/TIUD/TO40, P07/TCLRUD, P10-P17, and P80/TO00-P85/TO05 pins.

Caution Product quality may suffer if the absolute rating is exceeded for any parameter, even momentarily. In other words, an absolute maximum rating is a value at which the possibility of physical damage to the product cannot be ruled out. Care must therefore be taken to ensure that the these ratings are not exceeded during use of the product.

Recommended Operating Conditions

| Oscillation frequency | T _A | V _{DD} |
|---------------------------------|----------------|-----------------|
| 3 MHz ≤ f _{xx} ≤ 8 MHz | -40 to +85 °C | +5.0 V ± 10 % |

Capacitance (T_A = 25 °C, V_{SS} = V_{DD} = 0 V)

| Parameter | Symbol | Test conditions | MIN. | TYP. | MAX. | Unit |
|--------------------|-----------------|---------------------------------------|------|------|------|------|
| Input capacitance | C _i | f = 1 MHz 0 V except measured pins | | | 20 | pF |
| Output capacitance | C _o | | | | 20 | pF |
| I/O capacitance | C _{IO} | | | | 20 | pF |

Oscillator Characteristics (T_A = -40 to +85 °C, V_{DD} = +5 V ± 10 %, V_{SS} = 0 V)

| Resonator | Recommended circuit | Parameter | MIN. | MAX. | Unit |
|--|---------------------|---|------|------|------|
| Ceramic resonator or crystal resonator | | Oscillation frequency (f _{xx}) | 3 | 8 | MHz |
| External clock | | X1 input frequency (f _x) | 3 | 8 | MHz |
| | | X1 rise/fall time (t _{xR} , t _{xF}) | 0 | 30 | ns |
| | | X1 input high-/low-level width (t _{wxH} , t _{wxL}) | 40 | 170 | ns |

Caution When using system clock oscillation circuits, to reduce the effect of the wiring capacitance, etc, wire the area indicated by dotted-line as follows:

- Make the wiring as short as possible.
- Do not allow the wiring to intersect other signal lines. Keep it away from other lines in which varying high currents flow.
- Make sure that the ground point of the oscillation circuit capacitor is always at the same electric potential as V_{SS}. Do not allow the wiring to be grounded to a ground pattern in which very high currents are flowing.
- Do not extract signals from the oscillation circuit.

DC Characteristics (T_A = -40 to +85 °C, V_{DD} = +5 V ± 10 %, V_{SS} = 0 V)

| Parameter | Symbol | Test conditions | | MIN. | TYP. | MAX. | Unit |
|--------------------------------|-------------------|---|----------------------------------|-----------------------|------|--------------------|------|
| Low-level input voltage | V _{IL1} | Note 1 | | 0 | | 0.8 | V |
| | V _{IL2} | Note 2 | | 0 | | 0.2V _{DD} | V |
| High-level input voltage | V _{IH1} | Note 1 | | 2.2 | | | V |
| | V _{IH2} | Note 2 | | 0.8V _{DD} | | | V |
| Low-level output voltage | V _{OL1} | Note 3 | I _{OL} = 2.0 mA | | | 0.45 | V |
| | V _{OL2} | Note 4 | I _{OL} = 15 mA | | | 1.5 | V |
| | V _{OL3} | Note 5 | I _{OL} = 10 mA | | | 1.5 | V |
| High-level output voltage | V _{OH} | I _{OH} = -400 μA | | V _{DD} - 1.0 | | | V |
| Input leakage current | I _{LI} | 0 V ≤ V _I ≤ V _{DD} , AV _{DD} = V _{DD} | | | | ±10 | μA |
| Output leakage current | I _{LO} | 0 V ≤ V _O ≤ V _{DD} , AV _{DD} = V _{DD} | | | | ±10 | μA |
| V _{DD} supply current | I _{DD1} | Operating mode | | | 70 | 120 | mA |
| | I _{DD2} | HALT mode | | | 45 | 70 | mA |
| Data retention voltage | V _{DDDR} | STOP mode | | 2.5 | | | V |
| Data retention current | I _{DDDR} | STOP mode | V _{DDDR} = 2.5 V | | 2 | 10 | μA |
| | | | V _{DDDR} = 5.0 V ± 10 % | | 10 | 50 | μA |
| Pull-up resistance | R _L | V _I = 0 V | | 15 | 60 | 150 | kΩ |

Notes 1. Pins other than those specified in **Note 2**.

2. $\overline{\text{RESET}}$, X1, X2, P20/NMI, P21/INTP0, P22/INTP1, P23/INTP2, P24/INTP3/TI, P25/INTP4, P32/SO/SB0, P33/SI/SB1 and P34/ $\overline{\text{SCK}}$ pins.
3. Pins other than those specified in Notes 4 and 5.
4. P80/TO00-P85/TO05 pins (When I_{OL} = 15 mA is in operation, up to three pins can be ON simultaneously.)
5. P00/RTP0-P03/RTP3, P04/PWM0, P05/TCUD/PWM1, P06/TIUD/TO40 and P07/TCLRUD pins (When I_{OL} = 10 mA is in operation, up to four pins can be ON simultaneously.) as well as P10-P17 pins (When I_{OL} = 10 mA is in operation, up to four pins can be ON simultaneously.).

Caution When the P80-P85, P00-P07, and P10-P17 pins are not used under the conditions specified in Notes 4 and 5, they have the same characteristics as in Note 3.

AC Characteristics (T_A = -40 to +85 °C, V_{DD} = +5 V ± 10 %, V_{SS} = 0 V, C_L = 100 pF, f_{XX} = 8 MHz)

Read/Write Operation (when general-purpose memory is connected)

| Parameter | Symbol | Test conditions | MIN. | MAX. | Unit |
|---|-------------------|-----------------|------|-------|------|
| System clock cycle time | t _{CYK} | | 62.5 | 166.7 | ns |
| Address setup time (vs. ASTB ↓) | t _{SAST} | | 7 | | ns |
| Address hold time (vs. ASTB ↓) | t _{HSTA} | | 11 | | ns |
| \overline{RD} ↓ → address float time | t _{FRA} | | | 24 | ns |
| Address → data input time | t _{DAID} | | | 100 | ns |
| \overline{RD} ↓ → data input time | t _{DRID} | | | 49 | ns |
| ASTB ↓ → \overline{RD} ↓ delay time | t _{DSTR} | | 15 | | ns |
| Data hold time (vs. \overline{RD} ↑) | t _{HRID} | | 0 | | ns |
| \overline{RD} ↑ → address active time | t _{DRA} | | 17 | | ns |
| \overline{RD} low-level width | t _{WRL} | | 63 | | ns |
| ASTB high-level width | t _{WSTH} | | 14 | | ns |
| \overline{WR} ↓ → data output time | t _{DWOD} | | | 21 | ns |
| ASTB ↓ → \overline{WR} ↓ delay time | t _{DSTW} | | 15 | | ns |
| \overline{WR} ↑ → ASTB ↑ delay time | t _{DWST} | | 78 | | ns |
| Data setup time (vs. \overline{WR} ↑) | t _{SODW} | | 57 | | ns |
| Data hold time (vs. \overline{WR} ↑) | t _{HWOD} | | 8 | | ns |
| \overline{WR} low-level width | t _{WWL} | | 63 | | ns |

t_{CYK}-dependent Bus Timing Definition

| Parameter | Arithmetic expression | MIN./MAX. | Unit |
|-------------------|-----------------------|-----------|------|
| t _{SAST} | (0.5 + a) T - 24 | MIN. | ns |
| t _{HSTA} | 0.5T - 20 | MIN. | ns |
| t _{WSTH} | (0.5 + a) T - 17 | MIN. | ns |
| t _{DSTR} | 0.5T - 16 | MIN. | ns |
| t _{WRL} | (1.5 + n) T - 30 | MIN. | ns |
| t _{DAID} | (2.5 + a + n) T - 56 | MAX. | ns |
| t _{DRID} | (1.5 + n) T - 44 | MAX. | ns |
| t _{DRA} | 0.5T - 14 | MIN. | ns |
| t _{DSTW} | 0.5T - 16 | MIN. | ns |
| t _{DWST} | 1.5T - 15 | MIN. | ns |
| t _{WWL} | (1.5 + n) T - 30 | MIN. | ns |
| t _{DWOD} | 0.5T - 10 | MAX. | ns |
| t _{SODW} | (1 + n) T - 5 | MIN. | ns |

Remarks 1. T = t_{CYK} = 1/f_{CLK} (f_{CLK} refers to the internal system clock frequency.)

2. a becomes 1 when the address wait is inserted. Otherwise, it becomes 0.
3. n refers to the number of wait cycles that is inserted by specifying the PWC register.
4. Only the bus timings indicated in this table depend on t_{CYK}.

Serial Operation (T_A = -40 to +85 °C, V_{DD} = +5 V ± 10 %, V_{SS} = 0 V)

| Parameter | Symbol | Test conditions | | MIN. | MAX. | Unit |
|---|--------------------|--------------------------------|---------------------|------|------|------|
| Serial clock cycle time | t _{CYSK} | $\overline{\text{SCK}}$ output | Internal 8 dividing | 500 | | ns |
| | | $\overline{\text{SCK}}$ input | External clock | 500 | | ns |
| Serial clock low-level width | t _{WSKL} | $\overline{\text{SCK}}$ output | Internal 8 dividing | 210 | | ns |
| | | $\overline{\text{SCK}}$ input | External clock | 210 | | ns |
| Serial clock high-level width | t _{WSKH} | $\overline{\text{SCK}}$ output | Internal 8 dividing | 210 | | ns |
| | | $\overline{\text{SCK}}$ input | External clock | 210 | | ns |
| SI setup time (vs. $\overline{\text{SCK}}$ ↑) | t _{SRXSK} | | | 80 | | ns |
| SI hold time (vs. $\overline{\text{SCK}}$ ↑) | t _{HSKRX} | | | 80 | | ns |
| $\overline{\text{SCK}}$ ↓ → SO delay time | t _{DSKTX} | R = 1 kΩ, C = 100 pF | | | 210 | ns |

Up/Down Counter Operation (T_A = -40 to +85 °C, V_{DD} = +5 V ± 10 %, V_{SS} = 0 V)

| Parameter | Symbol | Test conditions | MIN. | MAX. | Unit |
|------------------------------|---|-------------------|------|------|------|
| TIUD high-/low-level width | t _{WTIUH} , t _{WTIUL} | Other than mode 4 | 2T | | ns |
| | | Mode 4 | 4T | | ns |
| TCUD high-/low-level width | t _{WTCUH} , t _{WTCUL} | Other than mode 4 | 2T | | ns |
| | | Mode 4 | 4T | | ns |
| TCLRUD high-/low-level width | t _{WCLUH} , t _{WCLUL} | | 2T | | ns |
| TCUD setup time (vs. TIUD ↑) | t _{STCU} | Mode 3 | T | | ns |
| TCUD hold time (vs. TIUD ↑) | t _{HTCU} | Mode 3 | T | | ns |
| TIUD setup time (vs. TCUD) | t _{S4TIU} | Mode 4 | 2T | | ns |
| TIUD hold time (vs. TCUD) | t _{H4TIU} | Mode 4 | 2T | | ns |
| TIUD & TCUD cycle time | t _{CYC} | Other than mode 4 | | 4 | MHz |
| | t _{CYC4} | Mode 4 | | 2 | MHz |

Remark T = t_{CYK} = 1/f_{CLK} (f_{CLK} refers to the internal system clock frequency.)

Other Operations (T_A = -40 to +85 °C, V_{DD} = +5 V ± 10 %, V_{SS} = 0 V)

| Parameter | Symbol | Test conditions | MIN. | MAX. | Unit |
|---------------------------------|---------------------------------------|-----------------|------|------|------|
| NMI high-/low-level width | t _{WNH} , t _{WNL} | | 2 | | μs |
| RESET high-/low-level width | t _{WRSH} , t _{WRSL} | | 1.5 | | μs |
| INTP0 high-/low-level width | t _{WI0H} , t _{WI0L} | Ts = T | 250 | | ns |
| | | Ts = 4T | 1.0 | | μs |
| | | Ts = 8T | 2.0 | | μs |
| | | Ts = 16T | 4.0 | | μs |
| INTP1 high-/low-level width | t _{WI1H} , t _{WI1L} | Ts = T | 250 | | ns |
| | | Ts = 4T | 1.0 | | μs |
| | | Ts = 8T | 2.0 | | μs |
| | | Ts = 16T | 4.0 | | μs |
| INTP2 high-/low-level width | t _{WI2H} , t _{WI2L} | Ts = T | 250 | | ns |
| | | Ts = 4T | 1.0 | | μs |
| INTP3(TI) high-/low-level width | t _{WI3H} , t _{WI3L} | Ts = T | 250 | | ns |
| | | Ts = 4T | 1.0 | | μs |
| | | Ts = 8T | 2.0 | | μs |
| | | Ts = 16T | 4.0 | | μs |
| | | Ts = 64T | 16.0 | | μs |
| | | Ts = 128T | 32.0 | | μs |
| INTP4 high-/low-level width | t _{WI4H} , t _{WI4L} | Ts = T | 250 | | ns |
| | | Ts = 4T | 1.0 | | μs |
| | | Ts = 8T | 2.0 | | μs |
| | | Ts = 16T | 4.0 | | μs |

Remarks 1. T = t_{CYK} = 1/f_{CLK} (f_{CLK} refers to the internal system clock frequency.)

2. Ts refers to the input sampling frequency. INTP0-INTP4 can be selected to programmable.

A/D Converter Characteristics (T_A = -40 to +85 °C, V_{DD} = +5 V ± 10 %, V_{SS} = AV_{SS} = 0 V, V_{DD} - 0.5 V ≤ AV_{DD} ≤ V_{DD})

| Parameter | Symbol | Test conditions | MIN. | TYP. | MAX. | Unit | |
|--|--------------------|--|---------------------------------|---------------|-------------------------|------------------|----|
| Resolution | | | 10 | | | bit | |
| Total error ^{Note 1} | | 4.5 V ≤ AV _{REF} ≤ AV _{DD} | | | ±0.4 | %FSR | |
| | | 3.4 V ≤ AV _{REF} ≤ AV _{DD} | | | ±0.7 | %FSR | |
| Quantization error | | | | | ±1/2 | LSB | |
| ★ Conversion time | t _{CONV} | 62.5 ns ≤ t _{CYK} < 80 ns | 208 | | | t _{CYK} | |
| | | 80 ns ≤ t _{CYK} ≤ 166.6 ns | 169 | | | t _{CYK} | |
| ★ Sampling time | t _{SAMP} | 62.5 ns ≤ t _{CYK} < 80 ns | 24 | | | t _{CYK} | |
| | | 80 ns ≤ t _{CYK} ≤ 166.6 ns | 20 | | | t _{CYK} | |
| Zero-scale error ^{Note 1} | | 4.5 V ≤ AV _{REF} ≤ AV _{DD} | | ±1.5 | ±2.5 | LSB | |
| | | 3.4 V ≤ AV _{REF} ≤ AV _{DD} | | ±1.5 | ±4.5 | LSB | |
| Full-scale error ^{Note 1} | | 4.5 V ≤ AV _{REF} ≤ AV _{DD} | | ±1.5 | ±2.5 | LSB | |
| | | 3.4 V ≤ AV _{REF} ≤ AV _{DD} | | ±1.5 | ±4.5 | LSB | |
| Nonlinearity error ^{Note 1} | | 4.5 V ≤ AV _{REF} ≤ AV _{DD} | | ±1.5 | ±2.5 | LSB | |
| | | 3.4 V ≤ AV _{REF} ≤ AV _{DD} | | ±1.5 | ±4.5 | LSB | |
| Analog input voltage ^{Note 2} | V _{IAN} | | -0.3 | | AV _{REF} + 0.3 | V | |
| Analog input impedance | R _{AN} | When not sampling | | 10 | | MΩ | |
| | | When sampling | | Note 3 | | | |
| Reference voltage | AV _{REF} | | 3.4 | | AV _{DD} | V | |
| AV _{REF1} current | AI _{REF} | | | 1.0 | 3.0 | mA | |
| AV _{DD} supply current | AI _{DD} | Operating mode | | 2.0 | 6.0 | mA | |
| A/D converter data retention current | AI _{DDDR} | STOP mode | AV _{DDDR} = 2.5 V | | 2 | 10 | μA |
| | | | AV _{DDDR} = 5 V ± 10 % | | 10 | 50 | μA |

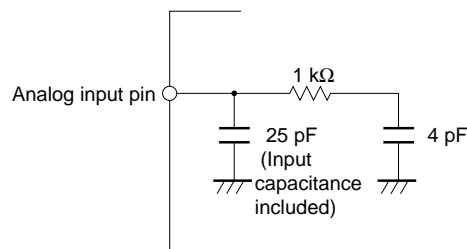
Notes 1. The quantization error is excluded.

2. When -0.3 V ≤ V_{IAN} ≤ 0 V, the conversion result becomes 000H.

When 0 V < V_{IAN} < AV_{REF}, the conversion is performed with the 10-bit resolution.

When AV_{REF} ≤ V_{IAN} ≤ +0.3 V, the conversion result becomes 3FFH.

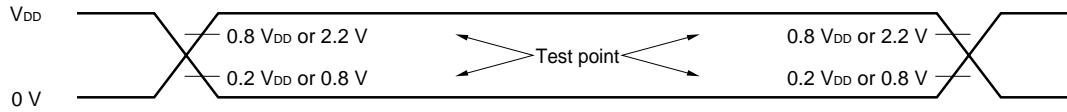
3. The analog input impedance at the time of sampling is the same as the equivalent circuit shown below. (The values in the diagram are TYP. values; they are not guaranteed values)



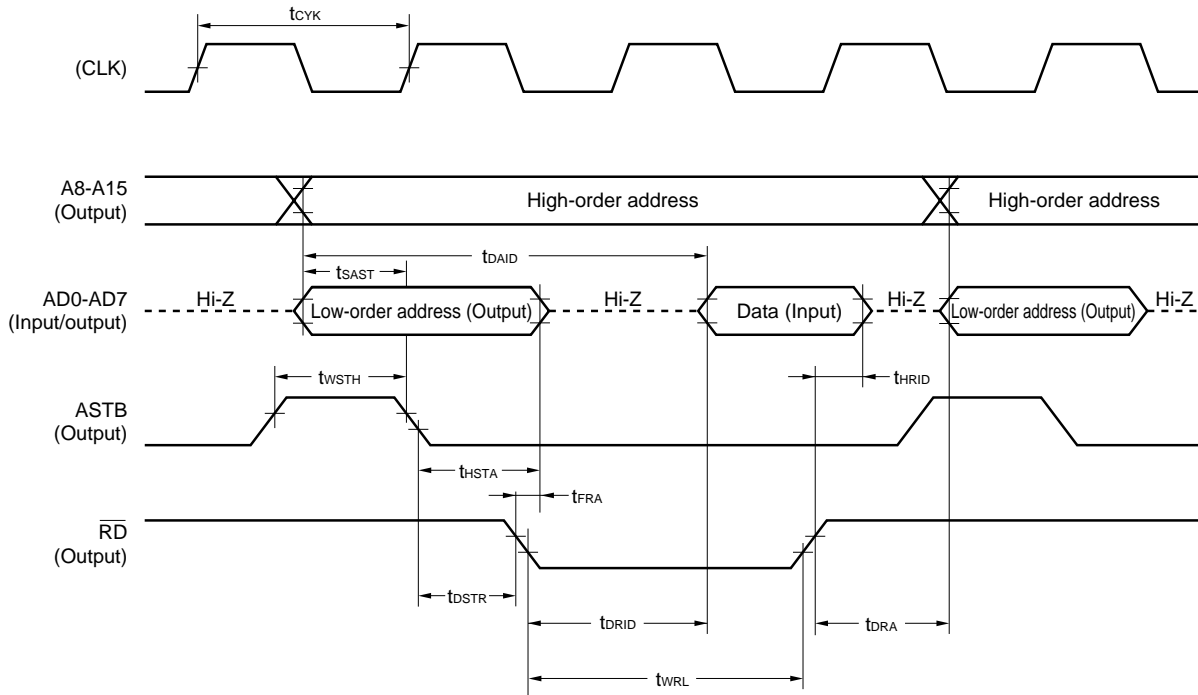
Cautions1. When using the P70/ANI0-P77/ANI7 pins for both digital and analog inputs, the previously described characteristics are not guaranteed. Therefore, ensure that all of the eight P70/ANI0-P77/ANI7 pins are used either for analog input or digital input.

2. When using the P70/ANI0-P77/ANI7 pins as digital input, make sure to set that $AV_{DD} = V_{DD}$, and $AV_{SS} = V_{SS}$.

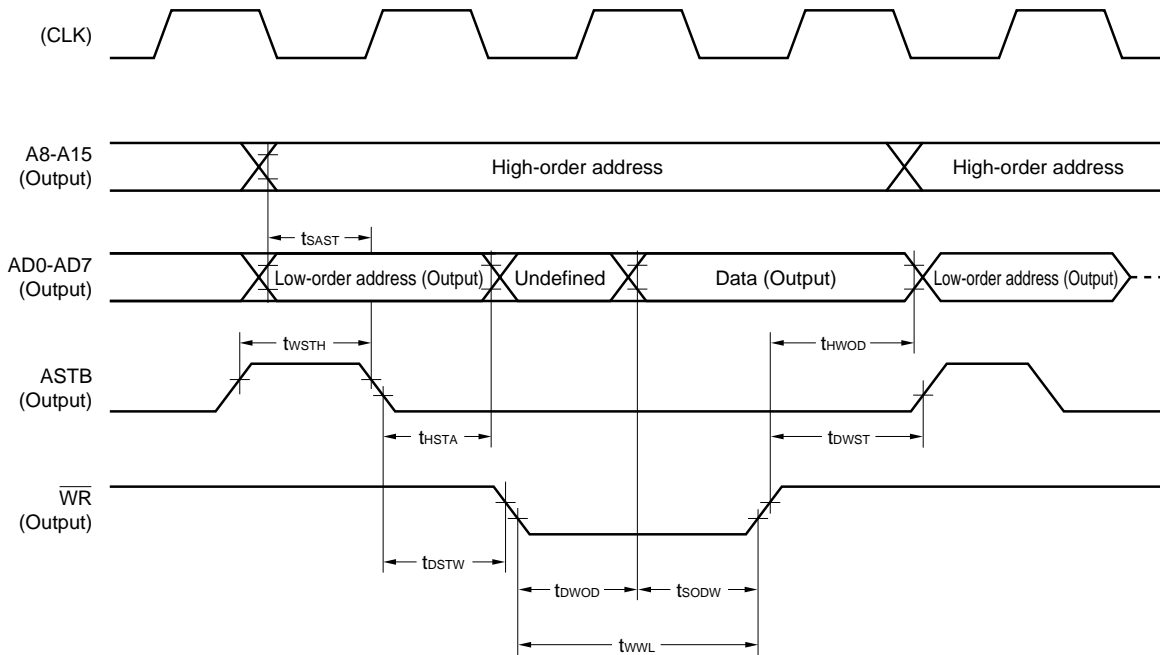
AC Timing Test Point



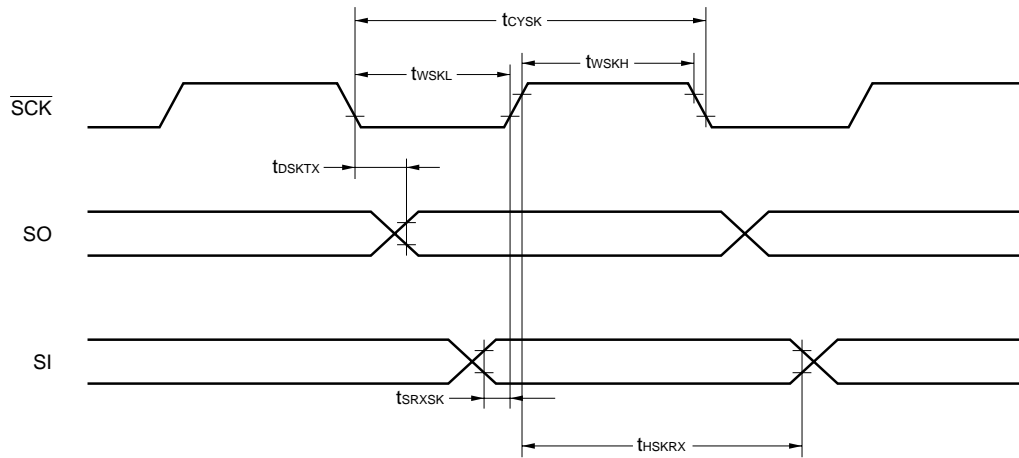
Read Operation



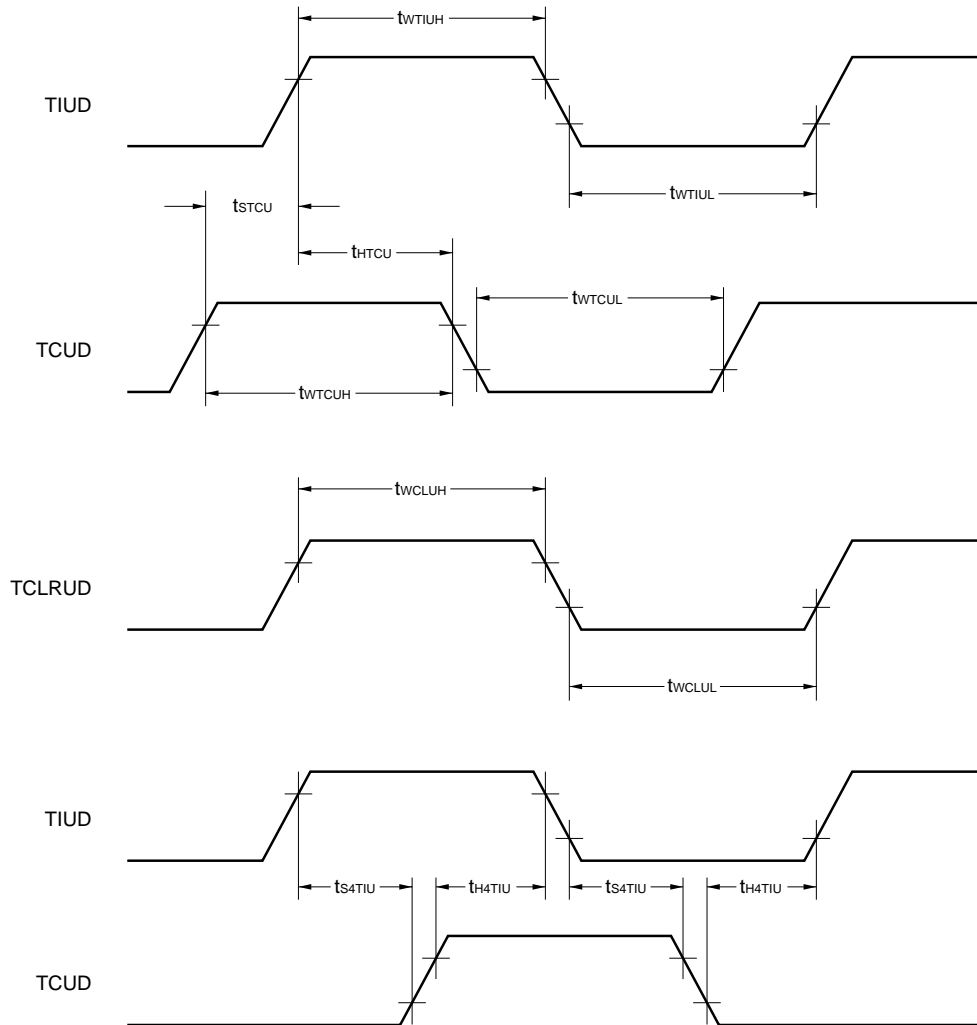
Write Operation



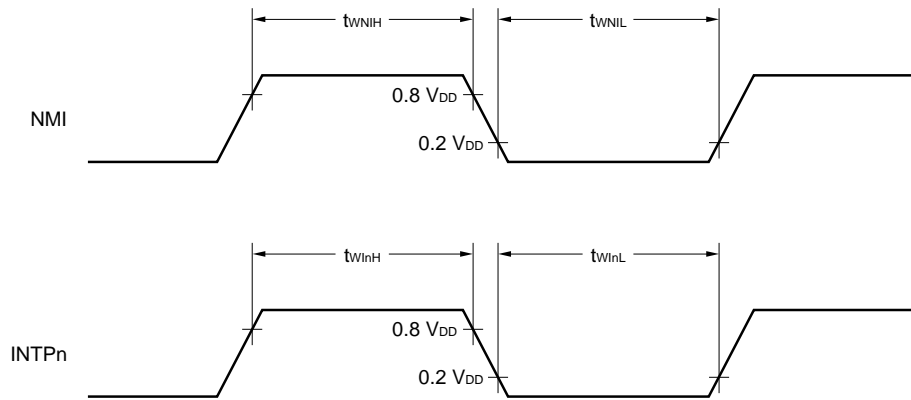
Serial Operation



Up/Down Counter (Timer 4) Input Timing

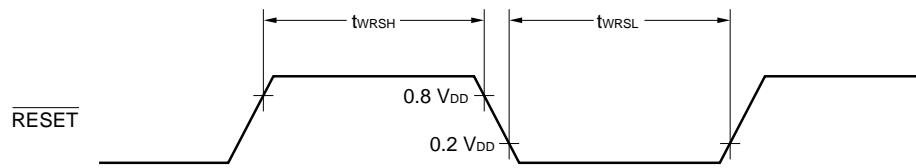


Interrupt Input Timing



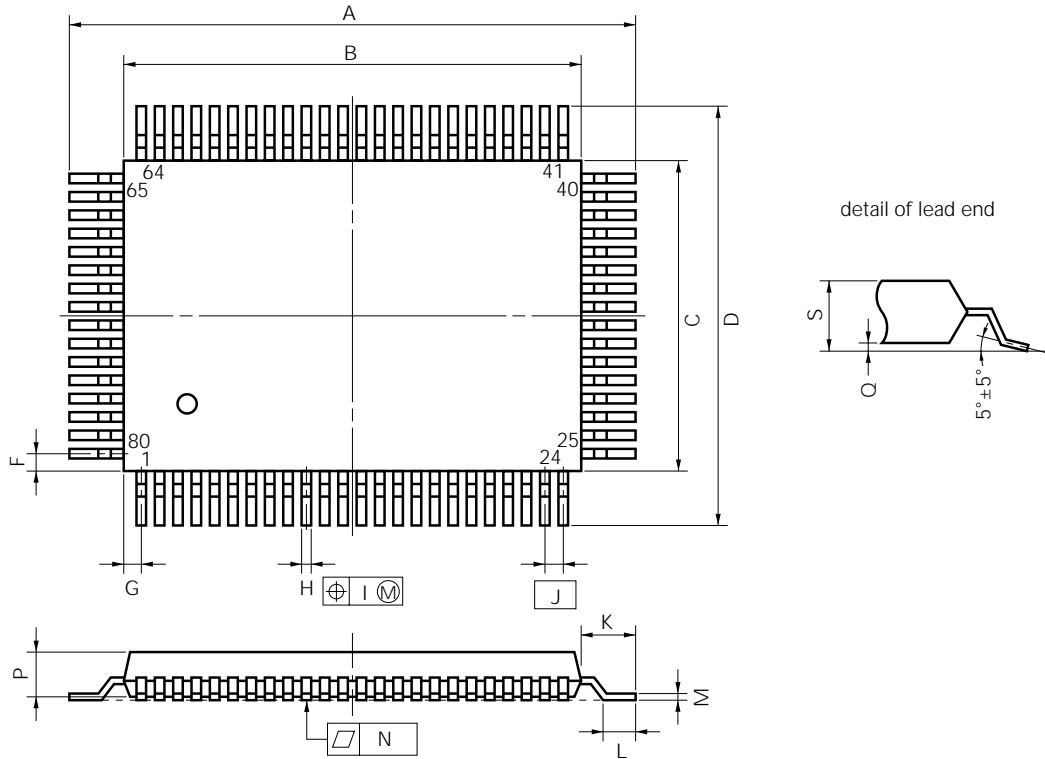
Remark n = 0 to 4

Reset Input Timing



11. PACKAGE DRAWING

80 PIN PLASTIC QFP (14×20)



P80GF-80-3B9-2

NOTE

Each lead centerline is located within 0.15 mm (0.006 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|------|--|---|
| A | 23.6±0.4 | 0.929±0.016 |
| B | 20.0±0.2 | 0.795 ^{+0.009} _{-0.008} |
| C | 14.0±0.2 | 0.551 ^{+0.009} _{-0.008} |
| D | 17.6±0.4 | 0.693±0.016 |
| F | 1.0 | 0.039 |
| G | 0.8 | 0.031 |
| H | 0.35±0.10 | 0.014 ^{+0.004} _{-0.005} |
| I | 0.15 | 0.006 |
| J | 0.8 (T.P.) | 0.031 (T.P.) |
| K | 1.8±0.2 | 0.071 ^{+0.008} _{-0.009} |
| L | 0.8±0.2 | 0.031 ^{+0.009} _{-0.008} |
| M | 0.15 ^{+0.10} _{-0.05} | 0.006 ^{+0.004} _{-0.003} |
| N | 0.15 | 0.006 |
| P | 2.7 | 0.106 |
| Q | 0.1±0.1 | 0.004±0.004 |
| S | 3.0 MAX. | 0.119 MAX. |

12. RECOMMENDED SOLDERING CONDITIONS

These products should be soldered and mounted under the conditions recommended below.

For details of recommended soldering conditions, refer to the information document **Semiconductor Device Mounting Technology Manual (C10535E)**.

For soldering methods and conditions other than those recommended, please contact your NEC sales representative.

★ **Table 12-1. Surface Mount Type Soldering Conditions**

μPD78363AGF-xxx-3B9: 80-Pin Plastic QFP (14 × 20 mm)

μPD78365AGF-3B9 : 80-Pin Plastic QFP (14 × 20 mm)

μPD78366AGF-xxx-3B9: 80-Pin Plastic QFP (14 × 20 mm)

μPD78368AGF-xxx-3B9: 80-Pin Plastic QFP (14 × 20 mm)

| Soldering method | Soldering conditions | Recommended condition symbol |
|------------------|---|------------------------------|
| Infrared reflow | Package peak temperature: 235 °C, Duration: 30 sec. max. (210 °C or above) Number of times: 3 max. | IR35-00-3 |
| VPS | Package peak temperature: 215 °C, Duration: 40 sec. max. (200 °C or above) Number of times: 3 max. | VP15-00-3 |
| Wave soldering | Solder bath temperature: 260 °C or less, Time: 10 sec. max., Number of times: 1, Pre-heating temperature: 120 °C max. (Package surface temperature) | WS60-00-1 |
| Partial heating | Pin temperature: 300 °C or less Duration: 3 sec. max. (per side of device) | — |

Caution Use of more than one soldering method should be avoided (except in the case of partial heating).

APPENDIX A. DIFFERENCES BETWEEN μPD78366A AND μPD78328

| Product name | | μPD78366A | μPD78328 |
|------------------------------------|-------|--|--|
| Item | | | |
| Minimum instruction execution time | | 125 ns [internal clock : 16 MHz external clock : 8 MHz] | 250 ns [internal clock : 8 MHz, external clock : 16 MHz] |
| Internal memory | ROM | 32K bytes | 16K bytes |
| | RAM | 2K bytes | 512 bytes |
| Memory space | | 64K bytes (can be externally expanded) | |
| General-purpose registers | | 8 bits × 16 × 8 banks | |
| Number of basic instructions | | 115 | 111 |
| Instruction set | | <ul style="list-style-type: none"> • 16-bit transfer/operation • Multiplication/division (16 bits × 16 bits, 32 bits ÷ 16 bits) • Bit manipulation • String | |
| | | <ul style="list-style-type: none"> • Sum-of-products operation (16 bits × 16 bits + 32 bits) • Relative operation | — |
| I/O lines | Input | 14 (of which 8 are multiplexed with analog input) | 11 (of which 8 are multiplexed with analog input) |
| | I/O | 49 | 41 |
| Real-time pulse unit | | <ul style="list-style-type: none"> • 16-bit timer × 5 • 16-bit compare register × 7 • 16-bit capture register × 3 • 16-bit capture/compare register × 2 • Two output modes selectable <ul style="list-style-type: none"> Mode 0, set-reset output : 6 channels Mode 1, buffer output : 6 channels • 16-bit resolution PWM output: 1 channel | <ul style="list-style-type: none"> • 16-bit timer × 3 • 16-bit compare register × 14 • 16-bit capture/compare register × 1 • Two output modes selectable <ul style="list-style-type: none"> Mode 0, set-reset output : 6 channels toggle output : 1 channel Mode 1, buffer output : 8 channels |
| Real-time output port | | 4 (buffer output in 4-bit units) | 4/8 (buffer output in 4-/8-bit units) |
| PWM unit | | 8-/9-/10-/12-bit resolution variable PWM output: 2 channels | 8-bit resolution PWM output: 1 channel |
| A/D converter | | 10-bit resolution, 8 channels | |
| Serial interface | | Dedicated baud rate generator UART (with pin selection function) : 1 channel Clocked serial interface/SBI : 1 channel | Dedicated baud rate generator UART : 1 channel Clocked serial interface/SBI : 1 channel |
| Interrupt function | | <ul style="list-style-type: none"> • External: 6, internal: 14 (2 multiplexed with external) • 4 programmable priority levels | |
| | | <ul style="list-style-type: none"> • External: 4, internal: 17 • 3 programmable priority levels | |
| Test source | | None | Internal: 1 |
| PLL control circuit | | Provided (external 8 MHz → internal: 16 MHz) | None |
| Package | | • 80-pin plastic QFP (14 × 20 mm) | <ul style="list-style-type: none"> • 64-pin plastic shrink DIP • 64-pin plastic QFP (14 × 20 mm) |
| Others | | <ul style="list-style-type: none"> • Watchdog timer • Standby functions (HALT mode, STOP mode) | |

APPENDIX B. TOOLS

B.1 DEVELOPMENT TOOLS

The following development tools are available to support the system development using μPD78366A :

Language Processor

| | | | | |
|---|---|--------------|----------------------------|---------------------------|
| 78K/III series relocatable assembler (RA78K3) | A relocatable assembler, that can be used commonly for the 78K/III series products. Since this assembler is provided with macro functions, it enhances the developmnt efficiency. A structured assembler, that can explicitly describe the program control structure, is also supplied, so that the program productivity and maintainability can be improved. | | | |
| | Host machine | OS | Supply media | Order code (product name) |
| | PC-9800 series | MS-DOS™ | 3.5" 2HD | μS5A13RA78K3 |
| | | | 5" 2HD | μS5A10RA78K3 |
| | IBM PC/AT™ and its compatible model | PC DOS™ | 3.5" 2HC | μS7B13RA78K3 |
| | | | 5" 2HC | μS7B10RA78K3 |
| | HP9000 series 700™ | HP-UX™ | DAT | μS3P16RA78K3 |
| | SPARC station™ | SunOS™ | Cartridge tape (QIC-24) | μS3K15RA78K3 |
| NEWS™ | NEWS-OS™ | μS3R15RA78K3 | | |
| 78K/III series C compiler (CC78K3) | This is a C compiler that can be commonly used for 78K/III series. This program converts the program written in C language to object codes microcomputer can execute. When using this compiler, the 78K/III series relocatable assembler (RA78K3) is necessary. | | | |
| | Host machine | OS | Supply media | Order code (product name) |
| | PC-9800 series | MS-DOS | 3.5" 2HD | μS5A13CC78K3 |
| | | | 5" 2HD | μS5A10CC78K3 |
| | IBM PC/AT and its compatible model | PC DOS | 3.5" 2HC | μS7B13CC78K3 |
| | | | 5" 2HC | μS7B10CC78K3 |
| | HP9000 series 700 | HP-UX | DAT | μS3P16CC78K3 |
| | SPARC station | SunOS | Cartridge tape (QIC-24) | μS3K15CC78K3 |
| NEWS | NEWS-OS | μS3R15CC78K3 | | |

Remark The operations of the relocatable assembler and C compiler are guaranteed only on the specified host machine and OS described above.

PROM Writing Tools

| | | | | | |
|----------|----------------------------|--|--------|--------------|--------------------------|
| Hardware | PG-1500 | This is a PROM programmer that can program PROM-contained single-chip microcontrollers in standalone mode or under control of a host machine when the accessory board and an optional programmer adapter are connected. It can also program representative PROMs from 256K-bit to 4M-bit models. | | | |
| | PA-78P368GF PA-78P368KL | PROM programmer adapters that writes a program to the μPD78P368A on a general-purpose PROM programmer such as the PG-1500. PA-78P368GF: for μPD78P368AGF PA-78P368KL: for μPD78P368AKL | | | |
| Software | PG-1500 controller | Connects the PG-1500 and a host machine with a serial interface and a parallel interface to control the PG-1500 from the host machine. | | | |
| | | Host machine | OS | Supply media | Order code (part number) |
| | | PC-9800 series | MS-DOS | 3.5" 2HD | μS5A13PG1500 |
| | | | | 5" 2HD | μS5A10PG1500 |
| | | IBM PC/AT and compatible machines | PC DOS | 3.5" 2HC | μS7B13PG1500 |
| 3.5" 2HC | μS7B10PG1500 | | | | |

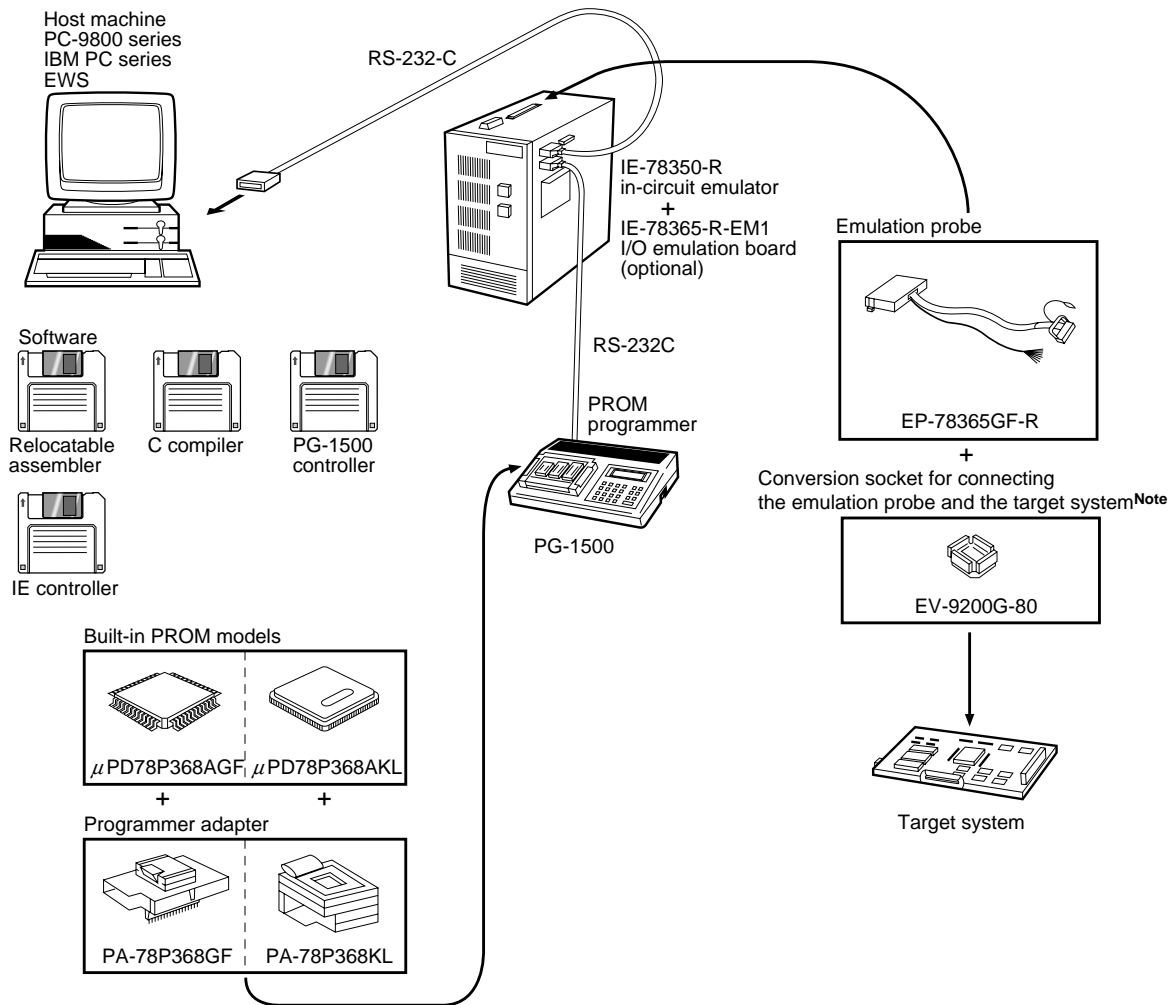
Remark The operation of the PG-1500 controller is guaranteed only on the above host machine and OS.

Debugging Tools (When IE Controller Is Used)

| | | | | | |
|----------|--|---|--------|--------------|--------------------------|
| Hardware | IE-78350-R | In-circuit emulator that can be used to develop and debug application systems. Connected to a host machine for debugging. | | | |
| | IE-78365-R-EM1 | I/O emulation board that emulates the peripheral functions of the target device such as I/O ports. | | | |
| | EP-78365GF-R EV-9200G-80 | Emulation probe that connects the IE-78350-R to the target system. One conversion socket, EV-9200G-80, used to connect the target system is supplied as an accessory. | | | |
| Software | IE-78350-R control program (IE controller) | Program that controls the IE-78350-R on the host machine. It can automatically execute commands, enhancing debugging efficiency. | | | |
| | | Host machine | OS | Supply media | Order code (part number) |
| | | PC-9800 series | MS-DOS | 3.5" 2HD | μS5A13IE78365A |
| | | | | 5" 2HD | μS5A10IE78365A |
| | | IBM PC/AT and compatible machines | PC DOS | 3.5" 2HC | μS7B13IE78365A |
| 3.5" 2HC | μS7B10IE78365A | | | | |

Remark The operation of the IE controller is guaranteed only on the above host machine and OS.

Development Tool Configuration (When Using IE Controller)



Note A socket is provided with the emulation probe.

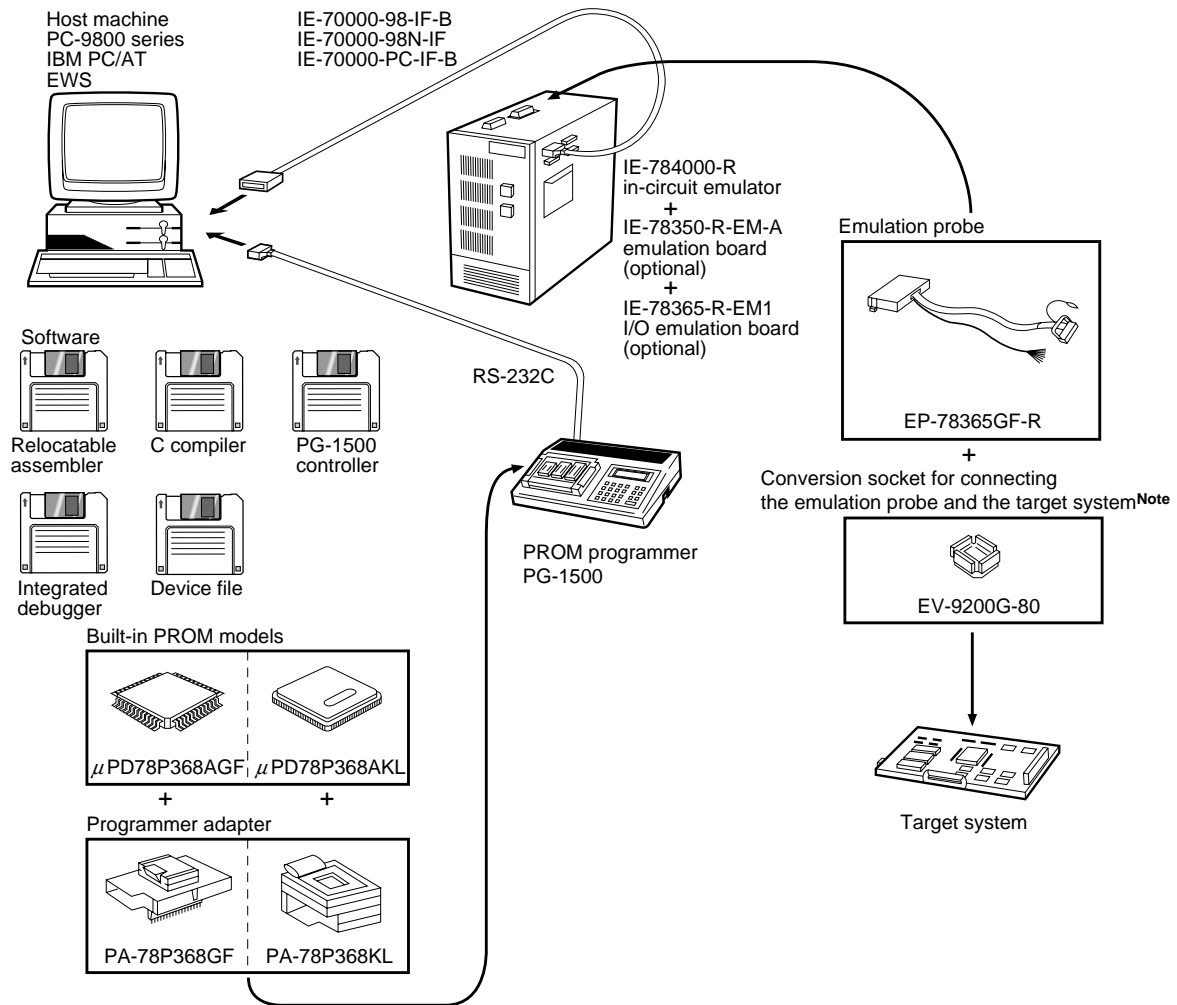
- Remarks 1.** Host machine and PG-1500 can be directly connected by RS-232-C.
2. 3.5-inch FD represents the supply media of software in this figure.

Debugging Tools (When Integrated Debugger Is Used)

| | | | | | |
|-----------------------------------|---|--|---|--------------------------|--------------|
| Hardware | IE-784000-R | In-circuit emulation that can be used to develop and debug the application system. Connected to a host machine for debugging. | | | |
| | IE-78350-R-EM-A | Emulation board that emulates the peripheral functions of the target device such as I/O ports. | | | |
| | IE-78365-R-EM1 | I/O emulation board that emulates the peripheral functions of the target device such as I/O ports. | | | |
| | EP-78365GF-R | Emulation probe connecting the IE-784000-R to the target system. One conversion socket, EV-9200G-80, used to connect the target system is supplied as an accessory. | | | |
| | EV-9200G-80 | | | | |
| | IE-70000-98-IF-B | Interface adapter to connect PC-9800 series (except notebook type personal computer) as the host machine. | | | |
| | IE-70000-98N-IF | Interface adapter and cable to connect PC-9800 series notebook type personal computer as the host machine. | | | |
| | IE-70000-PC-IF-B | Interface adapter and cable to connect IBM PC as the host machine. | | | |
| IE-78000-R-SV3 | Interface board to connect EWS as the host machine. | | | | |
| Software | Integrated debugger (ID78K3) | Program controlling the in-circuit emulator for the 78K/III series. Used in combination with a device file (DF78365). Can debug a program coded in the C language, structured assembly language, or assembly language at source program level. Can also split the screen of the host machine into windows on each of which information is displayed, enhancing debugging efficiency. | | | |
| | | Host machine | | Order code (part number) | |
| | | | OS | Supply media | |
| | | PC-9800 series | MS-DOS + Windows™ | 3.5" 2HD | μSAA13ID78K3 |
| | | | | 5" 2HD | μSAA10ID78K3 |
| | | IBM PC/AT and compatible machines (Japanese Windows) | PC DOS + Windows | 3.5" 2HC | μSAB13ID78K3 |
| | | | | 5" 2HC | μSAB10ID78K3 |
| | | IBM PC/AT and compatible machines (English Windows) | | 3.5" 2HC | μSBB13ID78K3 |
| | | | | 5" 2HC | μSBB10ID78K3 |
| | | Device File (DF78365) | File containing information peculiar to device. Use in combination with an assembler (RA78K3), C compiler (CC78K3), and integrated debugger (ID78K3). | | |
| | Host machine | | Order code (part number) | | |
| | | | OS | Supply media | |
| PC-9800 series | MS-DOS | | 3.5" 2HD | μS5A13DF78365 | |
| | | | 5" 2HD | μS5A10DF78365 | |
| IBM PC/AT and compatible machines | PC DOS | | 3.5" 2HC | μS7B13DF78365 | |
| | | 5" 2HC | μS7B10DF78365 | | |

Remark The operation of the integrated debugger and device file is guaranteed only on the above host machine and OS.

Development Tool Configuration (When Using Integrated Debugger)



Note A socket is provided with the emulation probe.

- Remarks 1.** Desk top-type PC represents host machine in this figure.
2. 3.5-inch FD represents the supply media of software in this figure.

B.2 EMBEDDED SOFTWARE

The following embedded software is available for enhancing the efficiency of program development and maintenance.

REAL-TIME OS

| | | | | |
|---|---|--------|--------------|--------------------------|
| Real-time OS (RX78K/III) ^{Note} | RX78K/III is intended to implement a multi-tasking environment for use in the control field where real-time capability is a must. It can allocate the idle time of the CPU to other processing to improve the overall performance of the system. RX78K/III provides system calls conforming to the μITRON specification. The RX78K/III package supplies a tool (configurator) to create the nucleus of RX78K/III and multiple information tables. | | | |
| | Host machine | OS | Supply media | Order code (part number) |
| | PC-9800 series | MS-DOS | 3.5" 2HD | Pending |
| | | | 5" 2HD | Pending |
| | IBM PC/AT and compatible machines | PC DOS | 3.5" 2HC | Pending |
| 5" 2HC | | | Pending | |

Note Under development

Caution Before purchasing this product, you are requested to conclude a contract licensing use by filling out a specified form.

Remark When using the RX78K/III real-time OS, the RA78K3 assembler package (optional) is necessary.

Fuzzy Inference Development Support System

| | | | | |
|---|--|----------|--------------------------|--------------|
| Fuzzy knowledge data creation tool (FE9000, FE9200) | Program that supports input/editing and evaluation (simulation) of fuzzy knowledge (fuzzy rules and membership functions). | | | |
| | Host machine | | Order code (part number) | |
| | | OS | Supply media | |
| | PC-9800 series | MS-DOS | 3.5" 2HD | μS5A13FE9000 |
| | | | 5" 2HD | μS5A10FE9000 |
| IBM PC/AT and compatible machines | PC DOS + Windows | 3.5" 2HC | μS7B13FE9200 | |
| | | 5" 2HC | μS7B10FE9200 | |
| Translator (FT78K3) ^{Note} | Program that converts the fuzzy knowledge data obtained by using the fuzzy knowledge data creation tool into assembler source program for the RA78K/III. | | | |
| | Host machine | | Order code (part number) | |
| | | OS | Supply media | |
| | PC-9800 series | MS-DOS | 3.5" 2HD | μS5A13FT78K3 |
| | | | 5" 2HD | μS5A10FT78K3 |
| IBM PC/AT and compatible machines | PC DOS | 3.5" 2HC | μS7B13FT78K3 | |
| | | 5" 2HC | μS7B10FT78K3 | |
| Fuzzy inference module (FI78K/III) ^{Note} | Program that executes fuzzy inference when linked with the fuzzy knowledge data converted by the translator. | | | |
| | Host machine | | Order code (part number) | |
| | | OS | Supply media | |
| | PC-9800 series | MS-DOS | 3.5" 2HD | μS5A13FI78K3 |
| | | | 5" 2HD | μS5A10FI78K3 |
| IBM PC/AT and compatible machines | PC DOS | 3.5" 2HC | μS7B13FI78K3 | |
| | | 5" 2HC | μS7B10FI78K3 | |
| Fuzzy inference debugger (FD78K/III) | Support software that evaluates and adjusts the fuzzy knowledge data at the hardware level by using an in-circuit emulator. | | | |
| | Host machine | | Order code (part number) | |
| | | OS | Supply media | |
| | PC-9800 series | MS-DOS | 3.5" 2HD | μS5A13FD78K3 |
| | | | 5" 2HD | μS5A10FD78K3 |
| IBM PC/AT and compatible machines | PC DOS | 3.5" 2HC | μS7B13FD78K3 | |
| | | 5" 2HC | μS7B10FD78K3 | |

Note Under development

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

Note: No connection for CMOS device inputs can be cause of malfunction. If no connection is provided to the input pins, it is possible that an internal input level may be generated due to noise, etc., hence causing malfunction. CMOS device 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.

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: 800-366-9782
Fax: 800-729-9288

NEC Electronics (Germany) GmbH

Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.

Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.l.

Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH

Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.

Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

NEC Electronics (France) S.A.

Spain Office
Madrid, Spain
Tel: 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: 253-8311
Fax: 250-3583

NEC Electronics Taiwan Ltd.

Taipei, Taiwan
Tel: 02-719-2377
Fax: 02-719-5951

NEC do Brasil S.A.

Sao Paulo-SP, Brasil
Tel: 011-889-1680
Fax: 011-889-1689

MS-DOS and windows are either registered trademarks or trademarks of Microsoft Corporation in the United states and/or other countries.

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.

TRON is an abbreviation of The Realtime Operating system Nucleus.

ITRON is an abbreviation of Industrial TRON.

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

Lisence not needed : μ PD78365A

The customer must judge the need for license : μ PD78363A, 78366A, 78368A

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

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

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

NEC devices are classified into the following three quality grades:

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

Standard: Computers, office equipment, communications equipment, test and measurement equipment, audio and visual equipment, home electronic appliances, machine tools, personal electronic equipment and industrial robots

Special: Transportation equipment (automobiles, trains, ships, etc.), traffic control systems, anti-disaster systems, anti-crime systems, safety equipment and medical equipment (not specifically designed for life support)

Specific: Aircrafts, aerospace equipment, submersible repeaters, nuclear reactor control systems, life support systems or medical equipment for life support, etc.

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

Anti-radioactive design is not implemented in this product.