

16/8-BIT SINGLE-CHIP MICROCONTROLLER

DESCRIPTION

The μ PD78324 is a 16/8-bit single-chip microcontroller that incorporates a high-performance 16-bit CPU. The μ PD78324 is one of 78K/III series. The internal capacity is significantly increased compared with the conventional μ PD78322.

A realtime pulse unit for realtime pulse control required in motor control, an A/D converter, a ROM, and a RAM have been integrated into one chip.

The μ PD78324 incorporates 32K-byte mask ROM and 1024-byte RAM.

The μ PD78323 is a ROM-less version of the μ PD78324. Also, It is provided the μ PD78P324 as an on-chip PROM product.

Detailed information about product features and specifications can be found in the following document.

μ PD78322 User's Manual : IEU-1248

FEATURES

- Internal 16-bit architecture and external 8-bit data bus
- High-speed processing by pipeline control and instruction prefetch
 - Minimum instruction execution time: 250 ns (with 16 MHz external clock in operation)
- Instruction set suitable for control operations (μ PD78312 upward compatible)
 - Multiply/divide instructions (16 bits \times 16 bits, 32 bits \div 16 bits)
 - Bit manipulation instruction
 - String instruction, etc.
- On-chip high-function interrupt controller
 - 3-level priority specifiable
 - 3-type interrupt processing mode selectable
(Vectored interrupt function, context switching function, and macro service function)
- Variety of peripheral hardware
 - Realtime pulse unit
 - 8-channel, 10-bit A/D converter
 - Watchdog timer
- Powerful serial interface (with an on-chip dedicated baud rate generator)
 - UART 1 channel
 - SBI (NEC Standard Serial Bus Interface) } 1 channel
 - 3-wire serial I/O

APPLICATIONS

- Motor control devices

Unless there are any particular differences, the μ PD78324 is described as the representative model in this document.

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

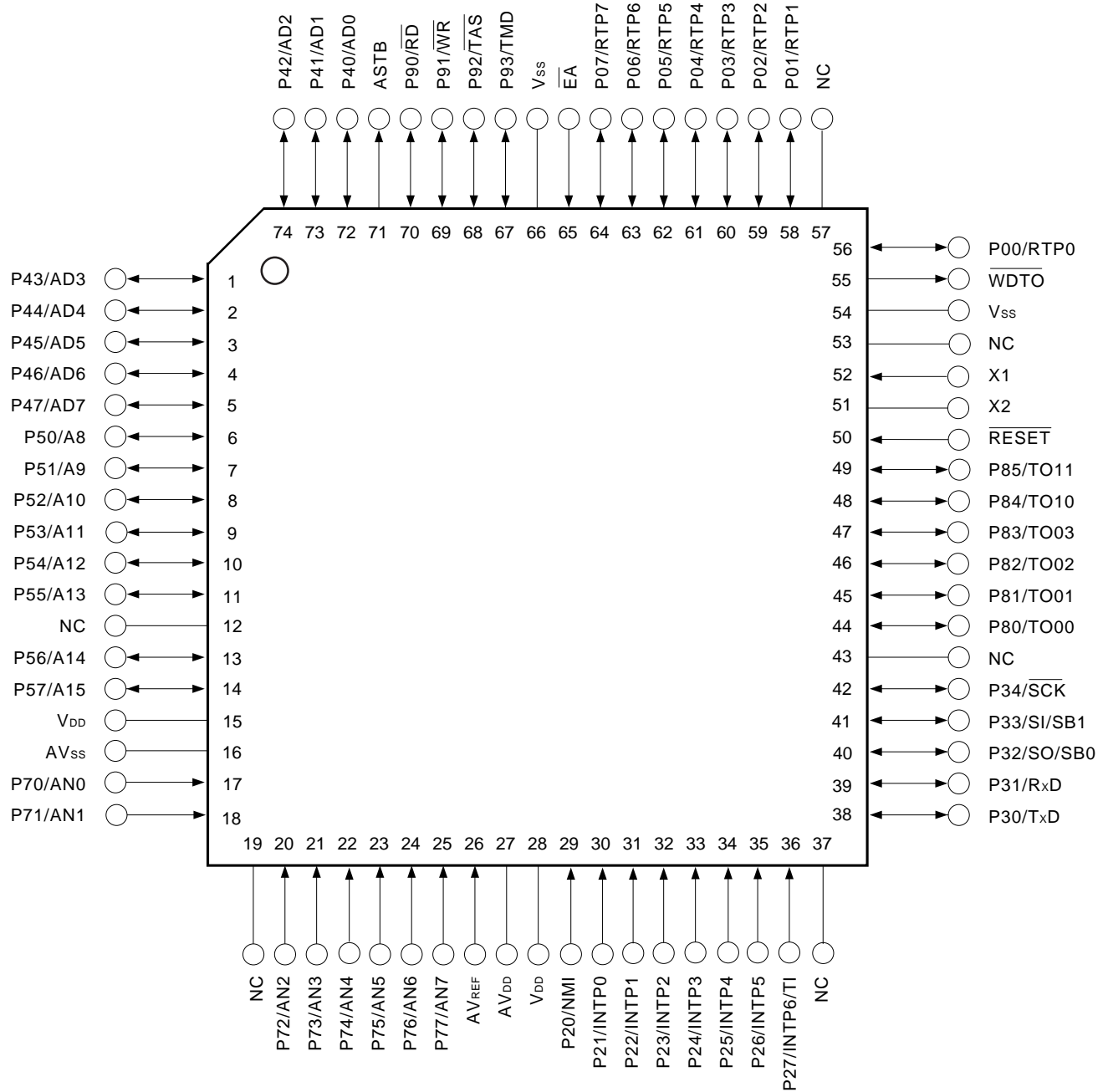
ORDERING INFORMATION

| Part Number | Package | On-chip ROM |
|--------------------|---------------------------------|-------------|
| μPD78323GJ-5BJ | 74-pin plastic QFP (20 × 20 mm) | None |
| μPD78323LP | 68-pin plastic QFJ (□950 mil) | None |
| μPD78324GJ-×××-5BJ | 74-pin plastic QFP (20 × 20 mm) | Mask ROM |
| μPD78324LP-××× | 68-pin plastic QFJ (□950 mil) | Mask ROM |

Remark ××× Indicates ROM code number.

PIN CONFIGURATION

- 74-pin plastic QFP (20 × 20 mm)
 μPD78323GJ-5BJ
 μPD78324GJ-xxx-5BJ

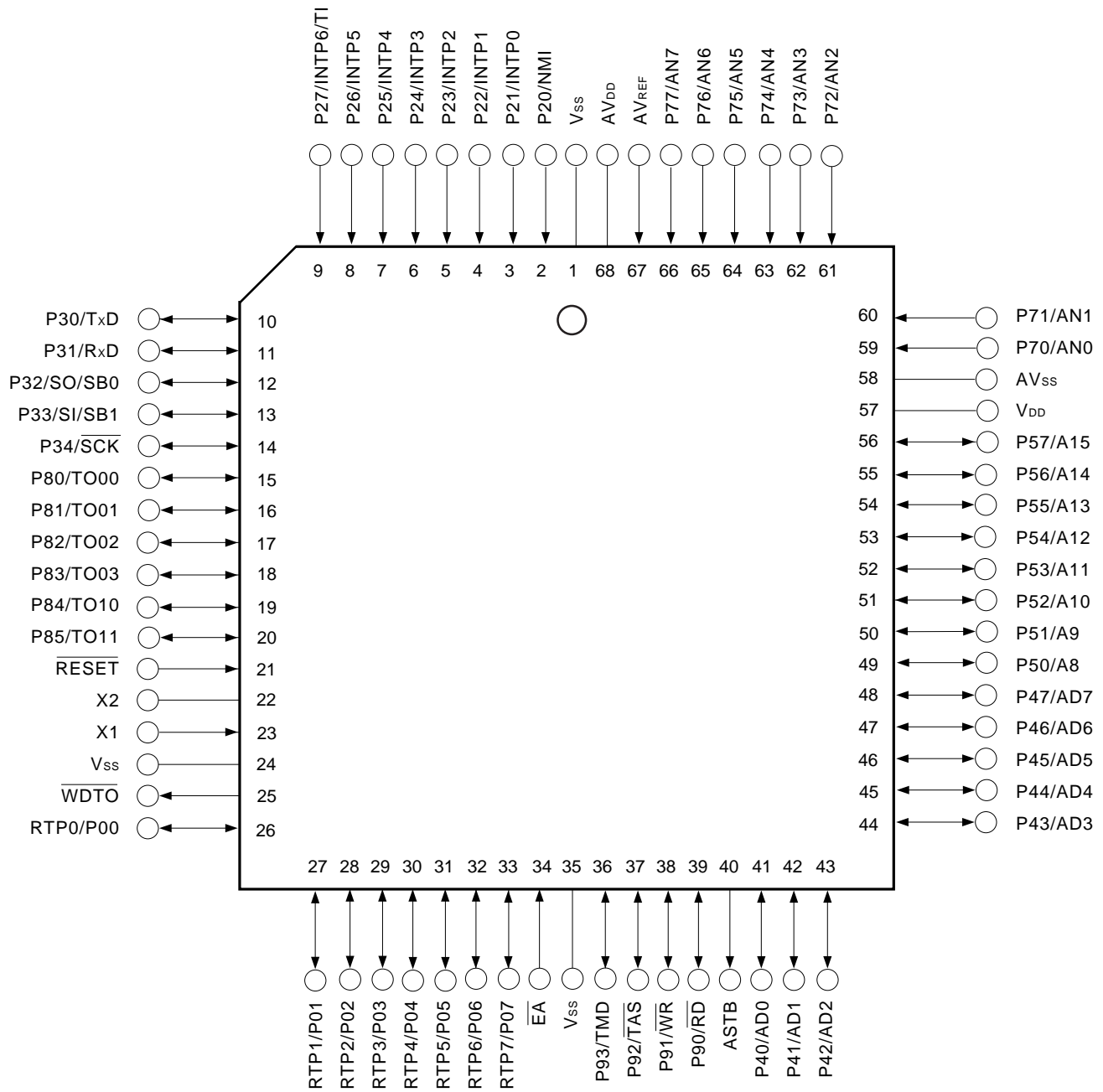


Caution The NC pin should be connected to VSS for noise control (can also be left open).

• 68-pin plastic QFJ (□950 mil)

μPD78323LP

μPD78324LP-xxx



| | | | |
|----------------|------------------------------|------------|----------------------------|
| P00 to P07 | : Port0 | RESET | : Reset |
| P20 to P27 | : Port2 | X1, X2 | : Crystal |
| P30 to P34 | : Port3 | WDTO | : Watchdog Timer Output |
| P40 to P47 | : Port4 | EA | : External Access |
| P50 to P57 | : Port5 | TMD | : Turbo Mode |
| P70 to P77 | : Port7 | TAS | : Turbo Access Strobe |
| P80 to P85 | : Port8 | WR | : Write Strobe |
| P90 to P93 | : Port9 | RD | : Read Strobe |
| NMI | : Nonmaskable Interrupt | ASTB | : Address Strobe |
| INTP0 to INTP6 | : Interrupt From Peripherals | AD0 to AD7 | : Address/Data Bus |
| RTP0 to RTP7 | : Realtime Port | A8 to A15 | : Address Bus |
| TI | : Timer Input | AN0 to AN7 | : Analog Input |
| TxD | : Transmit Data | AVREF | : Analog Reference Voltage |
| RxD | : Receive Data | AVSS | : Analog Vss |
| SB0/SO | : Serial Bus/Serial Output | AVDD | : Analog VDD |
| SB1/SI | : Serial Bus/Serial Input | VDD | : Power Supply |
| SCK | : Serial Clock | VSS | : Ground |
| TO00 to TO03 | : } Timer Output | NC | : Non-connection |
| TO10 to TO11 | | | |

GENERAL DESCRIPTION OF FUNCTIONS

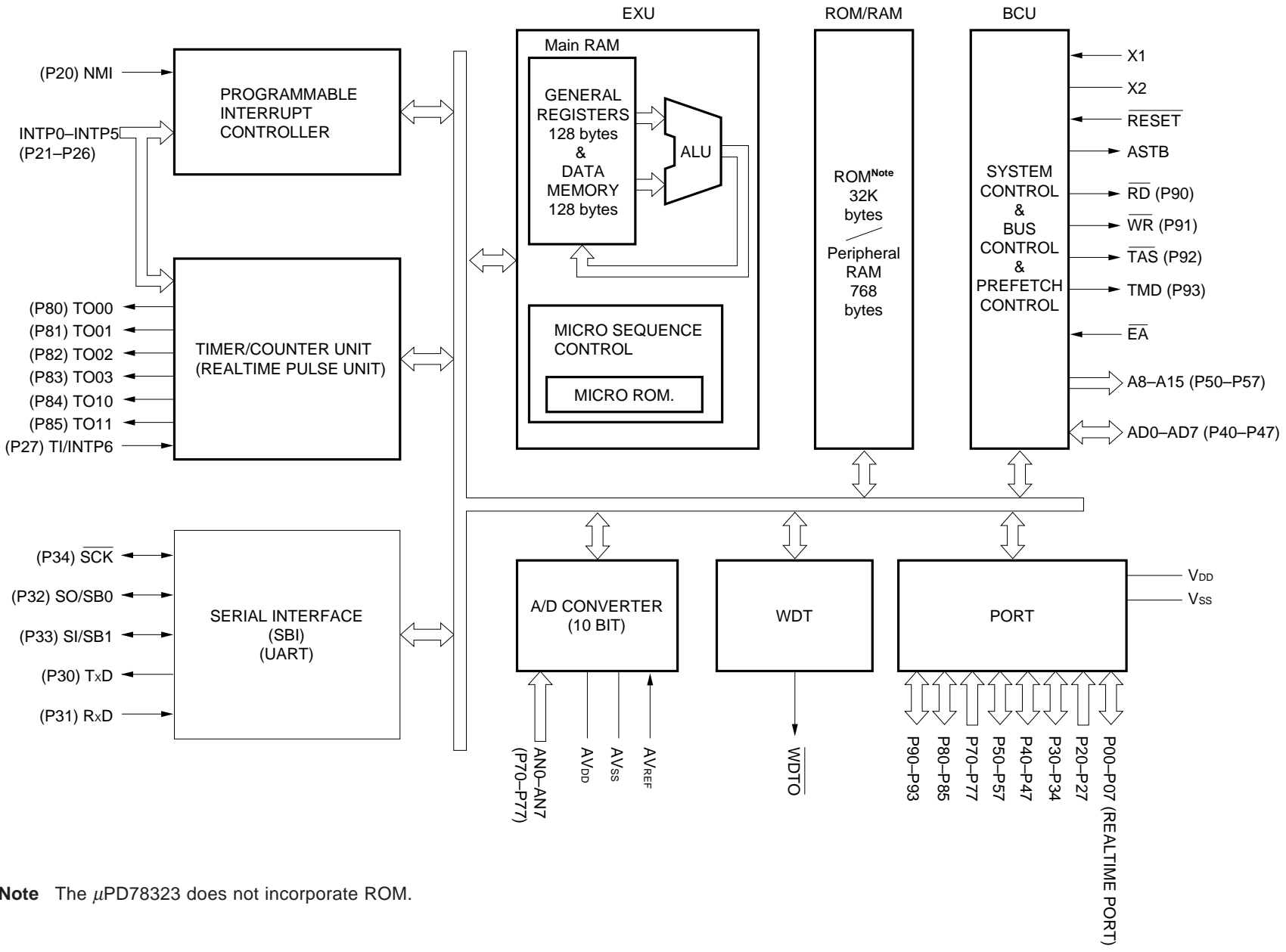
| | |
|------------------------------------|---|
| Basic instructions | 111 |
| Minimum instruction execution time | 250 ns (with 16 MHz external clock in operation) |
| Internal memory | <ul style="list-style-type: none"> • ROM : 32K bytes (μPD78324) None (μPD78323) • RAM : 1K bytes |
| Memory space | 64K bytes |
| General registers | 8 bits × 16 × 8 banks (memory mapping) |
| I/O line | <ul style="list-style-type: none"> • Input port : 16 (dual-function as analog input: 8) • Input/output port : 39 (μPD78324) 21 (μPD78323) |
| Real-time pulse unit | <ul style="list-style-type: none"> • 18/16-bit free running timer × 1 • 16-bit timer/event counter × 1 • 16-bit compare register × 6 • 18-bit capture register × 4 • 18-bit capture/compare register × 2 • Realtime output port × 8 |
| Serial communication interface | Serial interface with a dedicated baud rate generator <ul style="list-style-type: none"> • UART : 1 channel • SBI (NEC Serial Bus Interface) : 1 channel |
| A/D converter | 10-bit resolution (8 analog inputs) |
| Interrupt | <ul style="list-style-type: none"> • External : 8, internal : 14 (dual-function as external : 2) • 3 processing modes (vectored interrupt function, context switching function, and macro service function) |
| Test factor | Internal : 1 |
| Standby | STOP mode/HALT mode |
| Instruction set | 16-bit transfer/operation instruction, multiplication/division instruction (16 × 16, 32 ÷ 16), bit manipulation instruction, string instruction, etc. |
| Others | On-chip watchdog timer |
| Package | <ul style="list-style-type: none"> • 68-pin plastic QFJ (□950 mil) • 74-pin plastic QFP (20 × 20 mm) |

DIFFERENCES BETWEEN μPD78324 AND 78323

| Product Name | | μPD78324 | μPD78323 |
|-------------------------------------|--|---------------------------------------|---|
| Item | | μPD78324 | μPD78323 |
| Internal ROM | | 32K bytes | None |
| I/O line | Input | 16 (dual-function as analog input: 8) | |
| | Input /output | 39 | 21 |
| Port 4 (P40 to P47) | Specifiable as I/O as an 8-bit unit. Functions as multiplexed address/data buses (AD0 to AD7) in the external memory expansion mode. | | Functions always as multiplexed address/data buses. |
| Port 5 (P50 to P57) | Specifiable as I/O bit-wise. Functions as address bus (A8 to A15) in the external memory expansion mode. | | Functions always as address bus. |
| Port 9 (P90 to P93) | Specifiable as I/O bit-wise. In the external memory expansion mode, P90 and P91 function as RD strobe signal output and WR strobe signal output, respectively. In the external memory high-speed fetch mode, P92 P93 function as TAS output and TMD output respectively. | | Always P90 and P91 function as RD strobe and WR strobe signal output, respectively. |
| Memory expansion mode register (MM) | Port 4 I/O mode is set as an 8-bit unit . | | In the μPD78324 emulation mode, turbo access manager (μPD71P301) Note PA and PB pins are controlled as port 4 and port 5 emulation pins. |
| Port 5 mode register (PM5) | Port 5 I/O mode is set bit-wise. | | |

★

Note Maintenance product



Note The $\mu\text{PD78323}$ does not incorporate ROM.

CONTENTS

1. LIST OF PIN FUNCTIONS 11

 1.1 PORT PINS 11

 1.2 PINS OTHER THAN PORTS 12

 1.3 PIN INPUT/OUTPUT CIRCUITS AND RECOMMENDED CONNECTION OF UNUSED PINS 14

2. CPU ARCHITECTURE 16

 2.1 MEMORY SPACE 16

 2.2 PROCESSOR REGISTERS 19

 2.2.1 Control Registers 20

 2.2.2 General Registers 22

 2.2.3 Special Function Registers (SFR) 24

 2.3 DATA MEMORY ADDRESSING 29

 2.3.1 General Register Addressing 29

 2.3.2 Short Direct Addressing 29

 2.3.3 Special Function Register (SFR) Addressing 29

3. BLOCK FUNCTIONS 30

 3.1 BUS CONTROL UNIT (BCU) 30

 3.2 EXECUTION UNIT (EXU) 30

 3.3 ROM/RAM 30

 3.4 INTERRUPT CONTROLLER 30

 3.5 PORT FUNCTIONS 31

 3.6 CLOCK GENERATOR 32

 3.7 REALTIME PULSE UNIT (RPU) 34

 3.7.1 Configuration 34

 3.7.2 Realtime Output Function 36

 3.8 A/D CONVERTER 37

 3.9 SERIAL INTERFACE 37

 3.10 WATCHDOG TIMER 40

4. INTERRUPT FUNCTIONS 41

 4.1 OVERVIEW 41

 4.2 MACRO SERVICE 42

 4.3 CONTEXT SWITCHING FUNCTION 44

 4.3.1 Context Switching Function at Interrupt Request 44

 4.3.2 Context Switching Function by BRKCS Instruction 45

5. STANDBY FUNCTIONS 46

6. EXTERNAL DEVICE EXPANSION FUNCTION 47

7. OPERATION AFTER RESET 48

8. INSTRUCTION SET 49

9. ELECTRICAL SPECIFICATIONS 63

10. PACKAGE DRAWINGS 74

11. RECOMMENDED SOLDERING CONDITIONS 76

APPENDIX A. LIST OF 78K/III SERIES PRODUCTS 77

APPENDIX B. TOOLS 79

 B.1 DEVELOPMENT TOOLS 79

 B.2 EVALUATION TOOLS 83

★ B.3 EMBEDDED SOFTWARE 83

1. LIST OF PIN FUNCTIONS

1.1 PORT PINS

| Pin Name | I/O | Function | Dual-Function Pin |
|------------|--------------|---|-------------------|
| P00 to P07 | Input/output | Port 0 8-bit input/output port Input/output can be specified bit-wise Also serves as a realtime output port. | RTP0 to RTP7 |
| P20 | Input | Port 2 Dedicated port for 8-bit input | NMI |
| P21 | | | INTP0 |
| P22 | | | INTP1 |
| P23 | | | INTP2 |
| P24 | | | INTP3 |
| P25 | | | INTP4 |
| P26 | | | INTP5 |
| P27 | | | INTP6/TI |
| P30 | Input/output | Port 3 5-bit input/output port Input/output can be specified bit-wise | TxD |
| P31 | | | RxD |
| P32 | | | SO/SB0 |
| P33 | | | SI/SB1 |
| P34 | | | SCK |
| P40 to P47 | Input/output | Port 4 8-bit input/output port Input/output can be specified in 8-bit unit. | AD0 to AD7 |
| P50 to P57 | Input/output | Port 5 8-bit input/output port Input/output can be specified bit-wise | A8 to A15 |
| P70 to P77 | Input | Port 7 Dedicated port for 8-bit input | AN0 to AN7 |
| P80 | Input/output | Port 8 6-bit input/output port Input/output can be specified bit-wise | TO00 |
| P81 | | | TO01 |
| P82 | | | TO02 |
| P83 | | | TO03 |
| P84 | | | TO10 |
| P85 | | | TO11 |
| P90 | Input/output | Port 9 4-bit input/output port Input/output can be specified bit-wise | \overline{RD} |
| P91 | | | \overline{WR} |
| P92 | | | TAS |
| P93 | | | TMD |

1.2 PINS OTHER THAN PORTS (1/2)

| Pin Name | I/O | Function | Dual-Function Pin |
|--------------------------|---------------|--|-------------------|
| RTP0 to RTP7 | Output | Realtime output port which generates pulses in synchronization with the trigger signal transmitted from the realtime pulse unit (RPU). | P00 to P07 |
| NMI | Input | Nonmaskable interrupt request input capable of specifying the effective at the rising or falling edge by a mode register. | P20 |
| INTP0 | Input | External interrupt request input capable of specifying the effective edge by a mode register. | P21 |
| INTP1 | | | P22 |
| INTP2 | | | P23 |
| INTP3 | | | P24 |
| INTP4 | | | P25 |
| INTP5 | | | P26 |
| INTP6 | | | P27/TI |
| TI | Input | External count clock input to timer 1 (TM1) | P27/INTP6 |
| TxD | Output | Serial data output of asynchronous serial interface (UART) | P30 |
| RxD | Input | Serial data input of asynchronous serial interface (UART) | P31 |
| SO | Output | Serial data output of clock synchronous serial interface in 3-wire mode | P32/SB0 |
| SI | Input | Serial data input of clock synchronous serial interface in 3-wire mode | P33/SB1 |
| SB0 | Input /output | Serial data output of clock synchronous serial interface in SBI mode | P32/SO |
| SB1 | | | P33/SI |
| $\overline{\text{SCK}}$ | Input /output | Serial clock input/output of clock synchronous serial interface | P34 |
| AD0 to AD7 | Input /output | Multiplexed address/data bus for external memory expansion | P40 to P47 |
| A8 to A15 | Output | Address bus for external memory expansion | P50 to P57 |
| TO00 | Output | Pulse output from the realtime pulse unit | P80 |
| TO01 | | | P81 |
| TO02 | | | P82 |
| TO03 | | | P83 |
| TO10 | | | P84 |
| TO11 | | | P85 |
| $\overline{\text{RD}}$ | Output | Strobe signal output generated for external memory read operation | P90 |
| $\overline{\text{WR}}$ | | Strobe signal output generated for external memory write operation | P91 |
| $\overline{\text{TAS}}$ | | Control signal output generated for access to turbo access manager μPD71P301 ^{Note} | P92 |
| TMD | | | P93 |
| $\overline{\text{WDTO}}$ | Output | Signal output indicating that the watchdog timer has generated a nonmaskable interrupt. | — |
| ASTB | Output | Timing signal output generated for externally latching the address information output from pins AD0 to AD7 in order to access the external memory. | — |

★

Note Maintenance product

1.2 PINS OTHER THAN PORTS (2/2)

| Pin Name | I/O | Function | Dual-Function Pin |
|--------------------|-------|--|-------------------|
| \overline{EA} | Input | In the μ PD78324, \overline{EA} pin is normally connected to V_{DD} . Connecting \overline{EA} pin to V_{SS} sets the ROM-less mode and accesses the external memory. In the μ PD78323, this pin should be fixed to "0" (low level). The \overline{EA} pin level cannot be changed during operation. | — |
| AN0 to AN7 | Input | A/D converter analog input. | — |
| AV_{REF} | Input | A/D converter reference voltage input. | — |
| AV_{DD} | — | A/D converter analog power supply | — |
| AV_{SS} | — | A/D converter GND | — |
| \overline{RESET} | Input | System reset input | — |
| X1 | Input | Crystal connect pin for sysem clock oscillation. When an external clock is supplied, the clock is input to X1 and the inverted clock is input to X2. (X2 can also be left open.) | — |
| X2 | — | | — |
| V_{DD} | — | Positive power supply | — |
| V_{SS} | — | GND pin | — |
| NC | — | Not internally connected. Connected to V_{SS} (GND) (can also be left open). | — |

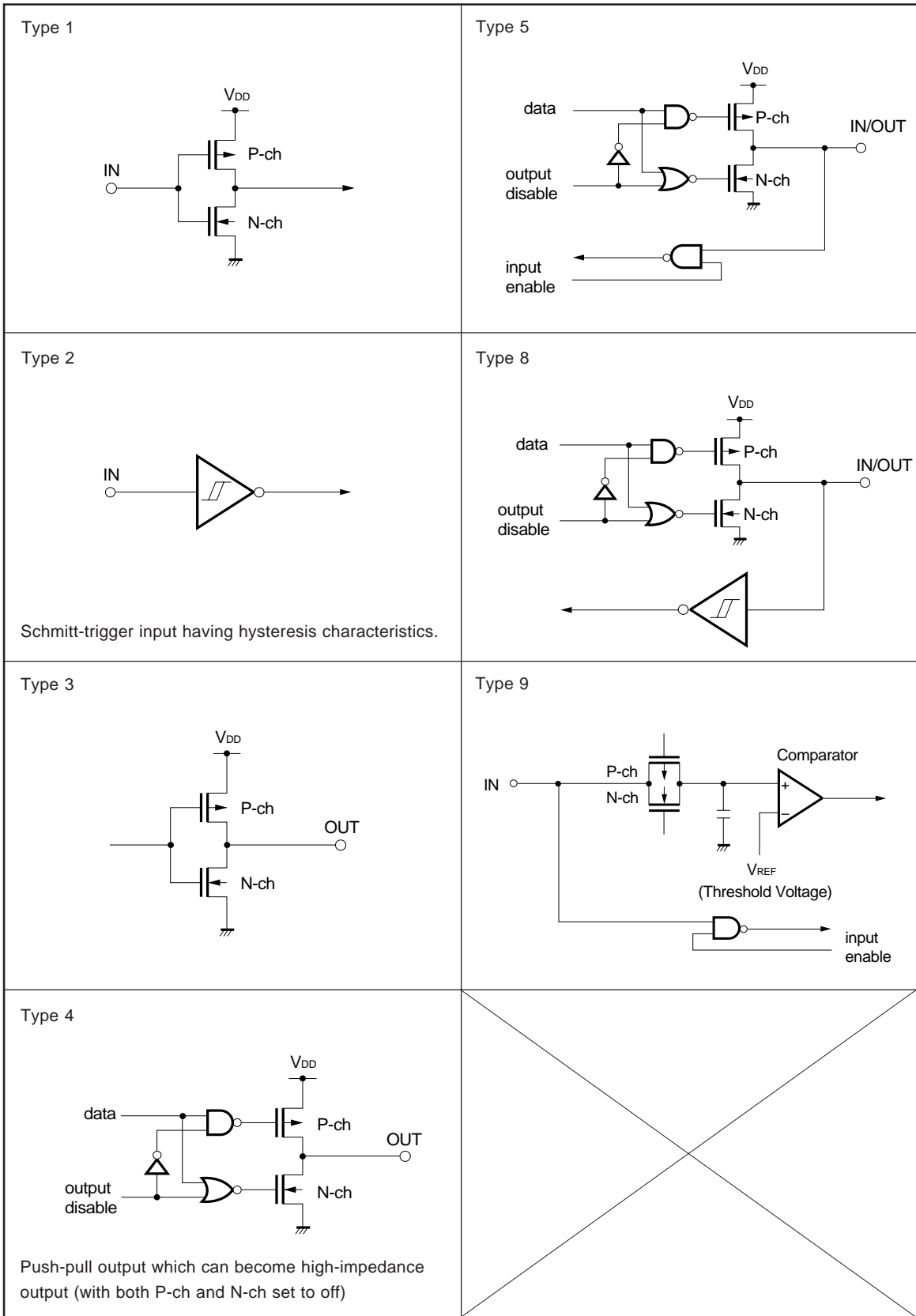
1.3 PIN INPUT/OUTPUT CIRCUITS AND RECOMMENDED CONNECTION OF UNUSED PINS

The pin input/output circuits, partly simplified, are shown in Table 1-1 and Figure 1-1.

Table 1-1. I/O Circuit Types of Pins and Their Recommended Connection Methods when Unused

| Pin | Input/Output Circuit Type | Recommended Connection Method |
|---|---------------------------|---|
| P00/RTP0 to P07/RTP7 | 5 | Input mode : Individually connected to V_{DD} or V_{SS} via resistor Output mode: Leave open |
| P20/NMI P21/INTP0 to P26/INTP5 P27/INTP6/TI | 2 | Connected to V_{SS} |
| P30/TxD P31/RxD | 5 | Input mode : Individually connected to V_{DD} or V_{SS} via resistor Output mode: Leave open |
| P32/SO/SB0 P33/SI/SB1 P34/SCK | 8 | |
| P40/AD0 to P47/AD7 P50/A8 to P57/A15 | 5 | |
| P70/AN0 to P77/AN7 | 9 | Connected to V_{SS} |
| P80/TO00 to P83/TO03 P84/TO10, P85/TO11 | 5 | Input mode : Individually connected to V_{DD} or V_{SS} via resistor Output mode: Leave open |
| P90/RD P91/WR P92/TAS P93/TMD | 5 | |
| WDTO | 3 | Leav open |
| ASTB | 4 | |
| EA | 1 | — |
| RESET | 2 | — |
| AV_{REF} , AV_{SS} | — | Connected to V_{SS} |
| AV_{DD} | — | Connected to V_{DD} |
| NC | — | Connected to V_{SS} (can also be left open) |

Figure 1-1. Pin Input/Output Circuits



2. CPU ARCHITECTURE

2.1 MEMORY SPACE

In the μPD78324 a maximum of 64K bytes of memory can be addressed (see **Figure 2-1**).

Program fetches can be performed within the area from 0000H to FDFFH. However, when external memory expansion is implemented in the area from FE00H to FFFFH (main RAM and special function register area), program fetches can also be performed on this area. In this case, a program fetch is performed on the external memory, not on the main RAM or special function registers.

(1) Vector table area

Interrupt request from the peripheral hardware, reset input, external interrupt request and interrupt branch address by break instruction are stored in the 0000H to 003FH 64-byte area. Generation of an interrupt request sets the even address content of each table in the lower 8 bits of the program counter (PC) and the odd address content in the higher 8 bits.

| Interrupt Source | | Vector Table Address |
|---------------------|---|----------------------|
| RESET | (RESET pin input) | 0000H |
| NMI | (NMI pin input) | 0002H |
| WDT | (Watchdog timer) | 0004H |
| TMF0 | (Realtime pulse unit) | 0006H |
| EXF0 | (INTP0 pin input) | 0008H |
| EXF1 | (INTP1 pin input) | 000AH |
| EXF2 | (INTP2 pin input) | 000CH |
| EXF3 | (INTP3 pin input) | 000EH |
| EXF4/CCFX0 | (INTP4 pin input/realtime pulse unit) | 0010H |
| EXF5/CCFX1 | (INTP5 pin input/realtime pulse unit) | 0012H |
| EXF6/TI | (INTP6/TI pin input) | 0014H |
| CMF00 | (Realtime pulse unit) | 0016H |
| CMF01 | (Realtime pulse unit) | 0018H |
| CMF02 | (Realtime pulse unit) | 001AH |
| CMF03 | (Realtime pulse unit) | 001CH |
| CMF10 | (Realtime pulse unit) | 001EH |
| CMF11 | (Realtime pulse unit) | 0020H |
| SRF | (Serial receive complete) | 0024H |
| STF | (Serial send complete) | 0026H |
| CSIIIF | (Clock synchronous serial interface) | 0028H |
| ADF | (A/D converter) | 002AH |
| Operation code trap | | 003CH |
| BRK | (Break instruction) | 003EH |

If bit 1 (TPF) of CPU control word (CCW) is set to 1, the 8002H to 803FH external memory area is used as an interrupt vector table in place of 0002H to 003FH.

(2) CALLT table area

32 tables of call addresses of 1-byte call instruction (CALLT) can be stored in the 0040H to 007FH 64-byte area.

If bit 1 (TPF) of CPU control word (CCW) is set to 1, the 8040H to 807FH external memory area is used as a CALLT instruction table in place of 0040H to 007FH.

(3) CALLF entry area

The 0800H to 0FFFFH area can be directly subroutine-called by 2-byte call instruction (CALLF).

(4) On-chip RAM area

A 1024-byte RAM is built in FB00H to FEFFH. This area is composed of the following 2 RAMs.

- Peripheral RAM : FB00H to FDFFH (768 bytes)
- Main RAM : FE00H to FEFFH (256 bytes)

The main RAM can be accessed at high speed.

In the main RAM area, the macro service control word and general register group composed of 8 register banks are mapped onto the 36 bytes from FE06H to FE2BH and the 128 bytes from FE80H to FEFFH, respectively.

(5) Special function register (SFR) area

Registers having specially assigned functions, such as on-chip peripheral hardware mode registers and control registers, are mapped in the FF00H to FFFFH area. Addresses without mapped registers cannot be accessed.

(6) External memory area

The μ PD78324 can add external memories (ROM, RAM) to the 32K-byte (8000H to FFFFH) area.

The μ PD78323 can connect external memories (ROM, RAM) to the 64K-byte (0000H to FFFFH) area.

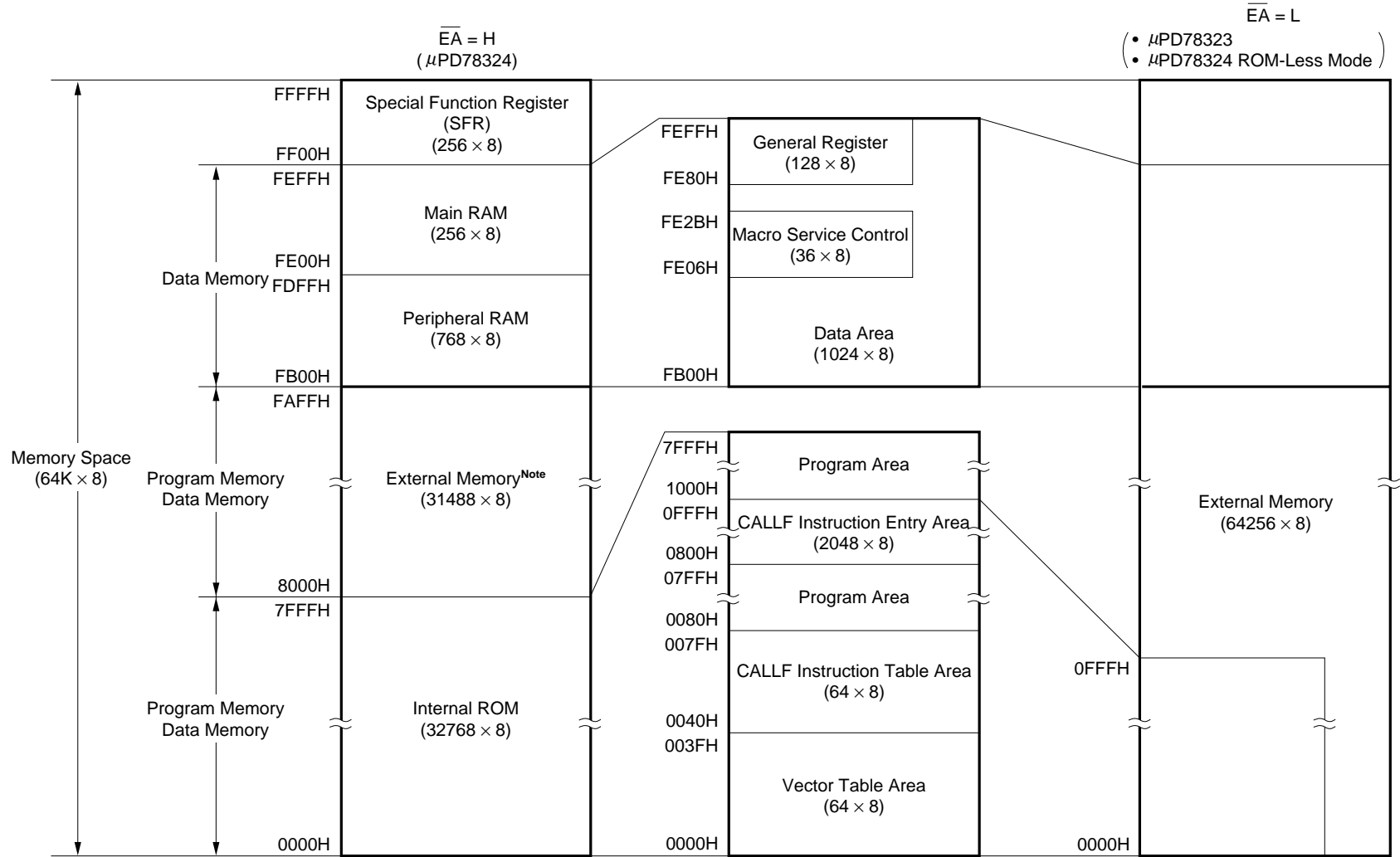
Each external memory can be accessed using P40/AD0 to P47/AD7 (multiplexed address/data bus), P50/A8 to P57/A15 (address bus) and \overline{RD} , \overline{WR} and ASTB signals.

The external access area is mapped in the FFD0H to FFDFH 16-byte area of the special function register (SFR). In this way, the external memory can be accessed by SFR addressing.

Dedicated pins (\overline{TAS} and TMD pins) are provided to connect turbo access manager (μ PD71P301)^{Note}. If the μ PD71P301 is used, the program processing speed equal to that of the on-chip ROM can be obtained. ★

Note Maintenance product

Figure 2-1. Memory Map



Note Accessed in external memory expansion mode.

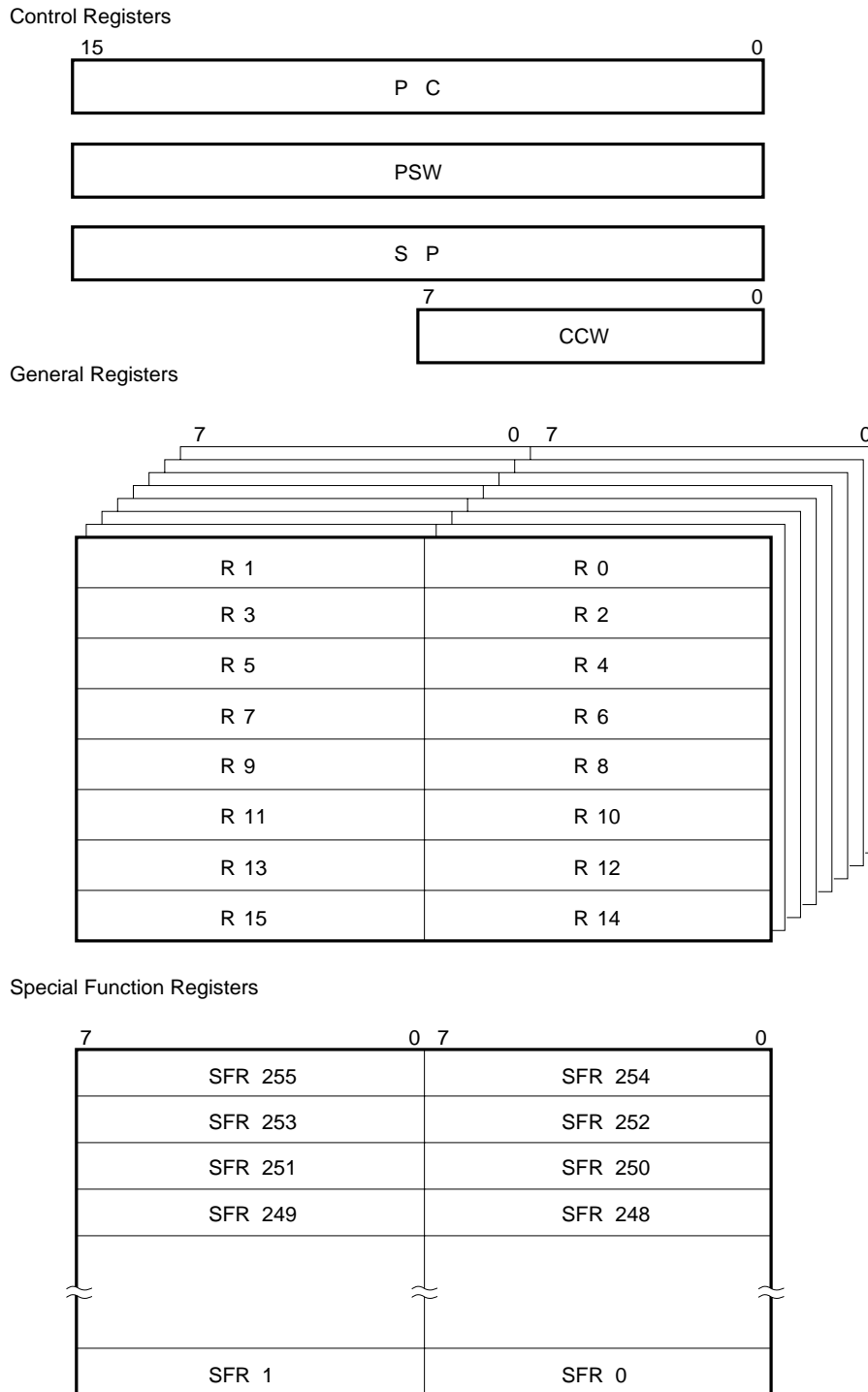
★

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 PROCESSOR REGISTERS

The processor registers consist mainly of three groups. They are general registers consisting of 8 banks of sixteen 8-bit registers, control registers consisting of one 8-bit register and three 16-bit registers, and special function registers such as peripheral hardware I/O mode registers.

Figure 2-2. Register Configuration



Remark The CCWs of the control registers are mapped in the special function register (SFR) area.

2.2.1 Control Register

The control registers carry out dedicated functions such as control of the program sequence, status and stack memory, and modification of operand addressing. They consist of three 16-bit registers and one 8-bit register.

(1) Program counter (PC)

This is a 16-bit register which holds the address information of the next program to be executed. It is normally incremented according to the number of bytes of the instruction to be fetched. If an instruction with data branch is executed, immediate data and the register content are set. RESET input sets and branches the data of 0000H and 0001H reset vector tables in the PC.

(2) Program status word (PSW)

This is a 16-bit register consisting of various flags which are set or reset by the result of instruction execution. Read/write access is carried out in units of the higher 8 bits (PSWH) or lower 8 bits (PSWL). Each flag can be operated using the bit operation instruction. If an interrupt request is made or BRK instruction is executed, data is automatically saved in the stack and is recovered by RETI or RETB instruction.

All bits are reset to 0 by RESET input.

Figure 2-3. PSW Format

(a) Interrupt priority level transition flag (LT)

| | | | | | | | | |
|------|----|------|------|------|----|-----|----|----|
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PSWH | UF | RBS2 | RBS1 | RBS0 | 0 | 0 | 0 | 0 |
| | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
| PSWL | S | Z | RSS | AC | IE | P/V | LT | CY |

This flag is used to control the interrupt priority. For normal operation of the interrupt control circuit, this bit must not be operated by a program.

(b) Carry flag (CY)

If a carry is generated out of bit 7 or 15 as a result of the execution of an operation instruction or a borrow is generated into bit 7 or 15, this flag is set to 1. In all other cases, this flag is reset to 0. This flag can be tested by the conditional branch instruction.

When a bit control instruction is executed, this flag functions as a bit accumulator.

(c) Zero flag (Z)

When the operation result is zero, this flag is set to 1. In all other cases, this flag is reset to 0. This flag can be tested by the conditional branch instruction.

(d) Sign flag (S)

When MSB of the operation result is "1", this flag is set to 1. When the MSB is "0", this flag is reset to 0. This flag can be tested by the conditional branch instruction.

(e) Parity/overflow flag (P/V)

Only when an overflow or underflow occurs as two's complement during execution of an arithmetic operation instruction, this flag is set to 1. In all other cases, it is reset to 0 (overflow flag operation).

If the bit number of the operation result set to 1 is even during execution of an logic operation instruction, this flag is set to 1. If the bit number is odd, this flag is reset to 0 (parity flag operation).

This flag can be tested by the conditional branch instruction.

(f) Auxiliary carry flag (AC)

If a carry is generated out of bit 3 as a result of operation or a borrow is generated into bit 3, this flag is set to 1. In all other cases, this flag is reset to 0. This flag can be tested by the conditional branch instruction.

(g) Register set select flag (RSS)

This flag is used to specify general registers X, A, C and B. As shown in Table 2-1, the RSS value determines the relationship between the functional register and the absolute register.

Thus, another register set (X, A, C, B) can be used by switching the RSS flag.

(h) Interrupt request enable flag (IE)

This flag is used to indicate interrupt request enable/disable. This flag is set to 1 by execution of EI instruction and is reset to 0 by execution of DI instruction or acceptance of an interrupt.

(i) Register bank select flag (RBS0 to RBS2)

This is a 3-bit flag to select one of eight register banks (RBANK0 to RBANK7).

(j) User flag (UF)

This flag is set or reset in the user program and can be used for program control.

(3) Stack pointer (SP)

This is a 16-bit register which holds the first address of the stack area (LIFO format) of the memory.

It is operated by a dedicated instruction.

SP is decremented before write (save) operation into the stack memory and is incremented after read (return) operation from the stack memory.

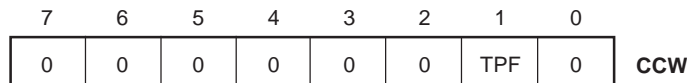
Since SP becomes indeterminate by $\overline{\text{RESET}}$ input, it must be set before subroutine call.

(4) CPU control word (CCW)

This is an 8-bit register consisting of CPU control related flags. It is mapped in the special function register area and can be controlled by the software.

All bits are reset to 0 by $\overline{\text{RESET}}$ input.

Figure 2-4. CCW Format



• Table position flag (TPF)

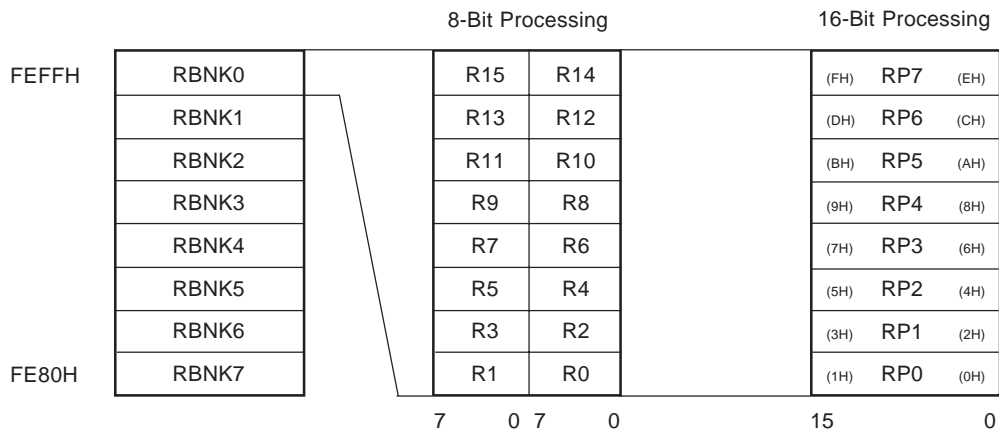
This flag is used to specify the interrupt vector table area and the memory area used as CALLT instruction table area.

As TPF has been reset to 0 after application of $\overline{\text{RESET}}$ input, the 0000H to 007FH address is used as each table area. The 8002H to 807FH address of the external memory area in place of 0002H to 007FH address can be used as each table area by setting TPF to 1 using the software. The vector tables of the BRK instruction, operation code trap interrupt and reset input are fixed to 003EH, 003CH and 0000H, respectively, and they are not affected by TPF.

2.2.2 General Registers

These are 128-byte registers mapped in the special area (FE80H to FEFFH) of the internal RAM space. They consist of eight register banks. The general register in the bank consists of sixteen 8-bit registers.

Figure 2-5. General Register Memory Location



The sixteen 8-bit registers can function as eight 16-bit register pairs (RP0 to RP7) as well.

As shown in Table 2-1, the sixteen 8-bit registers are characterized by functional names. The X register functions as the lower half of the 16-bit accumulator, the A register functions as the upper half of the 8-bit or 16-bit accumulator, the B and C registers function as a counter, and DE, HL, VP and UP function as address register pairs. In particular the VP register is function as a base register and the UP register is as a user stack pointer.

The unique function register charges as shown in Table 2-1 according to the value of the register set select flag (RSS) in the PSW. Thus, if the program is described by the functional name, another register set of X, A, C and B can be used by means of the RSS flag.

The μPD78324 can carry out processed data addressing operations, implied addressing by functional names with importance attached to the unique function of each register and register addressing by absolute names with a view to fast processing with a small number of data transfers or creating highly descriptive programs.

Table 2-1. General Register Configuration

| Absolute Name | Functional Name | | Absolute Name | Functional Name | |
|---------------|-----------------|-----------------|---------------|-----------------|---------|
| | RSS = 0 | RSS = 1 | | RSS = 0 | RSS = 1 |
| R0 | X | | RP0 | AX | |
| R1 | A | | RP1 | BC | |
| R2 | C | | RP2 | | AX |
| R3 | B | | RP3 | | BC |
| R4 | | X | RP4 | VP | VP |
| R5 | | A | RP5 | UP | UP |
| R6 | | C | RP6 | DE | DE |
| R7 | | B | RP7 | HL | HL |
| R8 | VP _L | VP _L | | | |
| R9 | VP _H | VP _H | | | |
| R10 | UP _L | UP _L | | | |
| R11 | UP _H | UP _H | | | |
| R12 | E | E | | | |
| R13 | D | D | | | |
| R14 | L | L | | | |
| R15 | H | H | | | |

2.2.3 Special Function Registers (SFR)

These registers are provided with special functions. They include various peripheral hardware mode registers and control registers (CCW).

The special function registers are assigned in the FF00H to FFFFH 256-byte space. Short direct memory addressing is applied to the FF00H to FF1FH 32-byte area for processing with a short word length.

The bit manipulation, arithmetic and transfer instructions can be executed in all areas. The FFD0H to FFDFH 16-byte area is externally accessible by SFR addressing. Thus, the external memory can be accessed and the external device bit manipulation can be carried out by an instruction having a short word length.

Table 2-2 lists the special function registers (SFR). The items in the table have the following meanings.

- Symbol..... Indicates the address of the built-in special function register.
Can be described in the instruction operand column.
- R/W.....Indicates if the corresponding special function register can read or write.
R/W : Read/write enable
R : Read only enable (register bit test enable)
W : Write only enable
- Manipulable bit unit
..... Indicates the applicable operation bit unit for the corresponding special function register.
16-bit manipulable SFR can be described in operand sfrp. When specified by an address, an even address is described.
1-bit manipulable SFR can be described by the bit operation instruction.
- On resetIndicates the state of each register when $\overline{\text{RESET}}$ is input.

- Cautions**
1. **Addresses for which no special function registers have been assigned cannot be accessed in the FF00H to FFFFH area.**
 2. **Do not write to the read only register. If data is written, the internal circuit may malfunction.**

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

| Address | Special Function Register (SFR) Name | Symbol | R/W | Manipulable Bit Unit | | | On Reset |
|---------|--------------------------------------|--------|-----|----------------------|--------|---------|------------|
| | | | | 1 bit | 8 bits | 16 bits | |
| FF00H | Port 0 | P0 | R/W | ○ | ○ | — | Undefined |
| FF02H | Port 2 | P2 | R | — | ○ | — | |
| FF03H | Port 3 | P3 | R/W | ○ | ○ | — | |
| FF04H | Port 4 | P4 | | ○ | ○ | — | |
| FF05H | Port 5 | P5 | ○ | ○ | — | | |
| FF07H | Port 7 | P7 | R | — | ○ | — | |
| FF08H | Port 8 | P8 | R/W | ○ | ○ | — | |
| FF09H | Port 9 | P9 | | ○ | ○ | — | |
| FF0AH | Free running counter | TM0LW | R | — | — | ○ | 0000H |
| FF0BH | (lower 16 bits) Note | | | — | — | ○ | Undefined |
| FF10H | Capture register X0 | CTX0LW | R | — | — | ○ | |
| FF11H | (lower 16 bits) Note | | | — | — | ○ | |
| FF12H | Capture register 01 | CT01LW | R | — | — | ○ | |
| FF13H | (lower 16 bits) Note | | | — | — | ○ | |
| FF14H | Capture register 02 | CT02LW | R | — | — | ○ | |
| FF15H | (lower 16 bits) Note | | | — | — | ○ | |
| FF16H | Capture register 03 | CT03LW | R | — | — | ○ | |
| FF17H | (lower 16 bits) Note | | | — | — | ○ | |
| FF18H | Capture/compoare register X0 | CCX0LW | R/W | — | — | ○ | Undefined |
| FF19H | (lower 16 bits) Note | | | — | — | ○ | |
| FF1AH | Capture/compoare register 01 | CC01LW | R/W | — | — | ○ | |
| FF1BH | (lower 16 bits) Note | | | — | — | ○ | |
| FF20H | Port 0 mode register | PM0 | W | — | ○ | — | FFH |
| FF23H | Port 3 mode register | PM3 | | — | ○ | — | ×××1 1111B |
| FF25H | Port 5 mode register | PM5 | | — | ○ | — | FFH |
| FF28H | Port 8 mode register | PM8 | | — | ○ | — | ××11 1111B |
| FF29H | Port 9 mode register | PM9 | | — | ○ | — | ×××× 1111B |
| FF2AH | Free running counter | TM0UW | R | — | — | ○ | 0000H |
| FF2BH | (higher 16 bits) Note | | | — | — | ○ | |
| FF2CH | Timer register 1 | TM1 | R | — | — | ○ | Undefined |
| FF2DH | | | | — | — | ○ | |
| FF30H | Capture register X0 | CTX0UW | R | — | — | ○ | |
| FF31H | (higher 16 bits) Note | | | — | — | ○ | |
| FF32H | Capture register 01 | CT01UW | R | — | — | ○ | |
| FF33H | (higher 16 bits) Note | | | — | — | ○ | |
| FF34H | Capture register 02 | CT02UW | R | — | — | ○ | |
| FF35H | (higher 16 bits) Note | | | — | — | ○ | |

Note Upper or lower half of 18-bit register.

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

| Address | Special Function Register (SFR) Name | Symbol | R/W | Manipulable Bit Unit | | | On Reset | |
|----------------|--|--------|-----|----------------------|--------|---------|------------|------------|
| | | | | 1 bit | 8 bits | 16 bits | | |
| FF36H FF37H | Capture register 03 (higher 16 bits) Note | CT03UW | R | — | — | ○ | Undefined | |
| FF38H FF39H | Capture/compoare register X0 (higher 16 bits) Note | CCX0UW | R/W | — | — | ○ | | |
| FF3AH FF3BH | Capture/compoare register 01 (higher 16 bits) Note | CC01UW | | — | — | ○ | | |
| FF40H | Port 0 mode control register | PMC0 | | W | — | ○ | | — |
| FF41H | Realtime output port reset register | RTPS | | R/W | ○ | ○ | | — |
| FF43H | Port 3 mode control register | PMC3 | W | — | ○ | — | | ×××0 0000B |
| FF48H | Port 8 mode control register | PMC8 | | — | ○ | — | ××00 0000B | |
| FF4CH FF4DH | Baud rate generator | BRG | R/W | — | — | ○ | Undefined | |
| FF60H | Realtime output port register | RTP | | ○ | ○ | — | | |
| FF61H | Realtime output port reset register | RTPR | | ○ | ○ | — | 00H | |
| FF62H | Port read control register | PRDC | | ○ | ○ | — | | |
| FF68H | A/D converter mode register | ADM | | ○ | ○ | — | | |
| FF6AH | A/D conversion result register (for 16-bit access) | ADCR | R | — | — | ○ | Undefined | |
| FF6BH | A/D conversion result register (for upper 8-bit access) | ADCRH | | — | ○ | — | | |
| FF70H FF71H | Compare register 00 | CM00 | R/W | — | — | ○ | Undefined | |
| FF72H FF73H | Compare register 01 | CM01 | R/W | — | — | ○ | | |
| FF74H FF75H | Compare register 02 | CM02 | | — | — | ○ | | |
| FF76H FF77H | Compare register 03 | CM03 | | — | — | ○ | | |
| FF7CH FF7DH | Compare register 10 | CM10 | | — | — | ○ | | |
| FF7EH FF7FH | Compare register 11 | CM11 | | — | — | ○ | | |
| FF80H | Clock synchronous serial interface mode register | CSIM | R/W | ○ | ○ | — | | 00H |
| FF82H | Serial bus interface control register | SBIC | | ○ | ○ | — | | |
| FF86H | Serial I/O shift register | SIO | | ○ | ○ | — | | Undefined |

Note Upper or lower half of 18-bit register.

Table 2-2. List of Special Function Registers (3/4)

| Address | Special Function Register (SFR) Name | Symbol | R/W | Manipulable Bit Unit | | | On Reset | |
|---------------|---|--------|---------------------|----------------------|--------|---------|-------------|-----|
| | | | | 1 bit | 8 bits | 16 bits | | |
| FF88H | Asynchronous serial interface mode register | ASIM | R/W | ○ | ○ | — | 80H | |
| FF8AH | Asynchronous serial interface status register | ASIS | R | ○ | ○ | — | 00H | |
| FF8CH | Serial receive buffer :UART | RXB | | — | ○ | — | Undefined | |
| FF8EH | Serial send shift register :UART | TXS | W | — | ○ | — | | |
| FFB0H | Timer control register | TMC | R/W | ○ | ○ | — | 00H | |
| FFB1H | Baud rate generator mode register | BRGM | | ○ | ○ | — | | |
| FFB2H | Prescaler mode register | PRM | | ○ | ○ | — | | |
| FFB8H | Timer output control register 0 | TOC0 | | ○ | ○ | — | | |
| FFB9H | Timer output control register 1 | TOC1 | | ○ | ○ | — | | |
| FFBFH | RPU mode register | RPUM | | ○ | ○ | — | | |
| FFC0H | Standby control register | STBC | R/W ^{Note} | ○ | ○ | — | 0000 × 000B | |
| FFC1H | CPU control word | CCW | R/W | ○ | ○ | — | 00H | |
| FFC2H | Watchdog timer mode register | WDM | R/W ^{Note} | ○ | ○ | — | | |
| FFC4H | Memory expansion mode register | MM | R/W | ○ | ○ | — | 22H | |
| FFC6H | Programmable weight control register | PWC | | ○ | ○ | — | 00H | |
| FFC9H | Fetch cycle control register | FCC | | ○ | ○ | — | Undefined | |
| FFD0H to FFDH | External acces area | | | ○ | ○ | — | 00H | |
| FFE0H | Interrupt request flag rgister 0L | IF0L | | IF0 | ○ | ○ | | ○ |
| FFE1H | Interrupt request flag rgister 0H | IF0H | | | ○ | ○ | ○ | |
| FFE2H | Interrupt request flag rgister 1L | IF1L | | IF1 | ○ | ○ | ○ | — |
| FFE3H | — | — | | | — | — | — | |
| FFE4H | Interrupt mask flag rgister 0L | MK0L | | MK0 | ○ | ○ | ○ | FFH |
| FFE5H | Interrupt mask flag rgister 0H | MK0H | | | ○ | ○ | ○ | |
| FFE6H | Interrupt mask flag rgister 1L | MK1L | MK1 | ○ | ○ | ○ | ×××× × 111B | |
| FFE7H | — | — | | — | — | — | | |
| FFE8H | Priority specify bufer register 0L | PB0L | PB0 | ○ | ○ | ○ | 00H | |
| FFE9H | Priority specify bufer register 0H | PB0H | | ○ | ○ | ○ | | |
| FFEAH | Priority specify bufer register 1L | PB1L | PB1 | ○ | ○ | ○ | — | |
| FFEBH | — | — | | — | — | — | | |
| FFECH | Interrupt processing mode specify register 0L | ISM0L | ISM0 | ○ | ○ | ○ | 00H | |
| FFEDH | Interrupt processing mode specify register 0H | ISM0H | | ○ | ○ | ○ | | |
| FFEEH | Interrupt processing mode specify register 1L | ISM1L | | ○ | ○ | ○ | | |
| FFEFH | — | — | ISM1 | — | — | ○ | — | |

Note Write enable in case of special instructions.

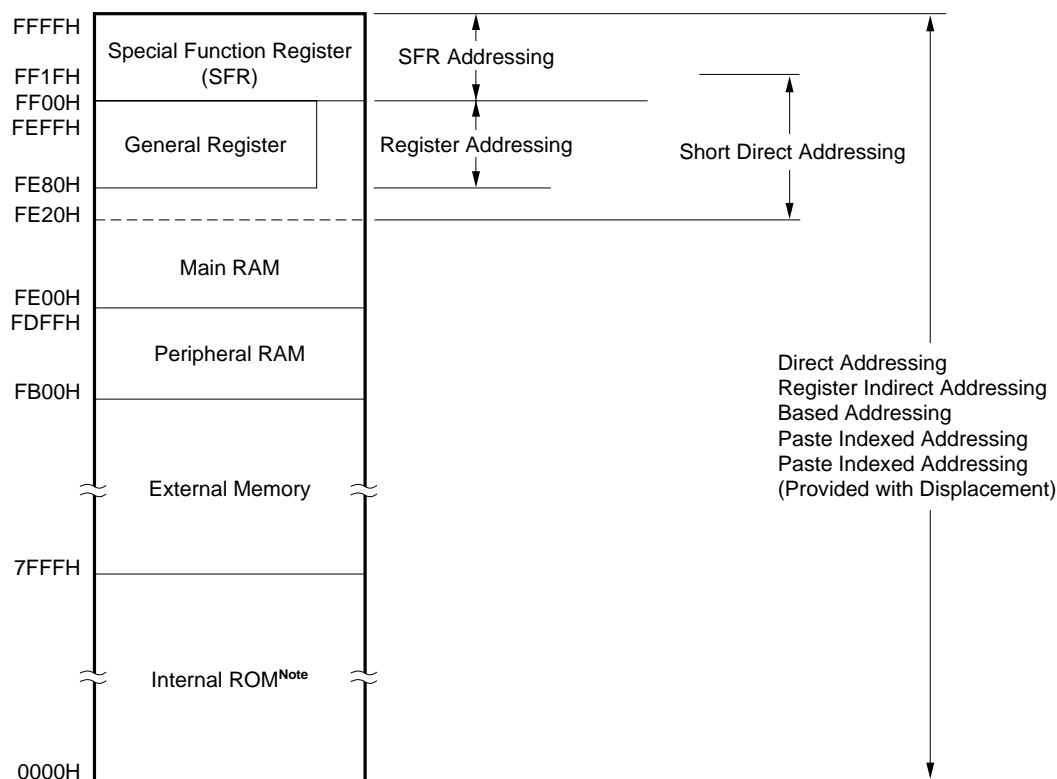
Table 2-2. List of Special Function Registers (4/4)

| Address | Special Function Register (SFR) Name | Symbol | R/W | Manipulable Bit Unit | | | On Reset |
|---------|--------------------------------------|--------|-----|----------------------|--------|---------|----------|
| | | | | 1 bit | 8 bits | 16 bits | |
| FFF0H | Context switching enable register 0L | CSE0L | R/W | ○ | ○ | ○ | 00H |
| FFF1H | Context switching enable register 0H | CSE0H | | ○ | ○ | | |
| FFF2H | Context switching enable register 1L | CSE1L | | ○ | ○ | ○ | |
| FFF3H | — | — | | — | — | | |
| FFF4H | External interrupt mode register 0 | INTM0 | | ○ | ○ | — | 00H |
| FFF5H | External interrupt mode register 1 | INTM1 | | ○ | ○ | — | |
| FFF8H | In-service priority register | ISPR | R | — | ○ | — | |
| FFF9H | Priority specify register | PRSL | R/W | ○ | ○ | — | |

2.3 DATA MEMORY ADDRESSING

In the μPD78324, the internal RAM space (FB00H to FFFFH) and the special function register area (FF00H to FFFFH) are mapped in the FB00H to FFFFH area. In the FE20H to FF1FH space of the data memory, short direct addressing enables direct addressing by 1-byte data in an instruction word.

Figure 2-6. Data Memory Addressing Space



Note When $\overline{EA} = L$, and with the μPD78323, this is external memory.

★ **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.

2.3.1 General Register Addressing

The general registers consist of eight register banks, each consisting of sixteen 8-bit registers or eight 16-bit registers.

General register addressing is carried out using the register specify field of 3 or 4 bits supplied from an instruction word, the register bank select flag (RBS0 to RBS2) and the register set select flag (RSS) in the PSW.

2.3.2 Short Direct Addressing

Short direct addressing which enables direct address specification by 1-byte data in an instruction word is applied to the FE20H to FF1FH space. The short direct memory is accessed as 8-bit or 16-bit data. When accessing the memory as 16-bit data, specification of even data for 1-byte address specify data will cause 2-byte data specified by continuous addresses of even and odd addresses to be accessed. (Do not specify odd number for address specify data.)

2.3.3 Special Function Register (SFR) Addressing

This addressing is applied to operations for the special function register (SFR) mapped in the SFR area of FF00H to FFFFH. Addressing is performed by 1-byte data in the instruction word corresponding to the lower 8 bits of the special function register address. For 16-bit access of 16-bit operational SFR, 2-byte data specified by continuous even and odd addresses is accessed as is the case with short direct addressing.

3. BLOCK FUNCTIONS

3.1 BUS CONTROL UNIT (BCU)

In the BCU, the necessary bus cycle is started according to the physical address obtained by the execution unit (EXU). If no bus cycle startup request is made from the EXU, a prefetch address is generated and instruction prefetch is carried out. The prefetched instruction code is fetched into the instruction queue.

3.2 EXECUTION UNIT (EXU)

In the EXU, address calculation, arithmetic logical operation and data transfer are controlled by microprograms. A 256-byte RAM is built in the EXU.

The 256-byte RAM in the EXU is accessible by the relevant instruction faster than peripheral RAM (768 bytes).

3.3 ROM/RAM

This block consists of a 32K-byte ROM and a 768-byte RAM. However, the μ PD78323 does not incorporate ROM. ROM access can be disabled by \overline{EA} pin.

3.4 INTERRUPT CONTROLLER

Various interrupt requests (NMI, INTP0 to INTP6) generated either externally or from the peripheral hardware are processed by the context switch, vectored interrupt or macro service function.

The 3-level interrupt priority is also specified.

3.5 PORT FUNCTIONS

Table 3-1 lists the digital input/output ports.

Each port can carry out many control operations including 8 and other bit data input/output operations.

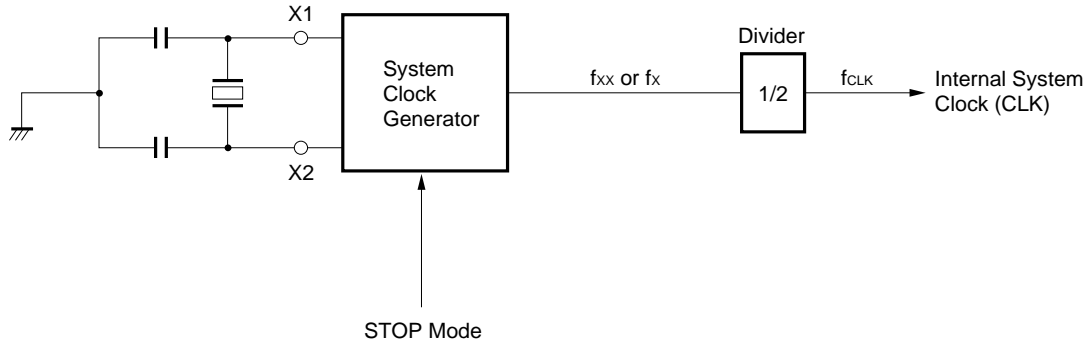
Table 3-1. Port Functions and Features

| Port Name | Function | Feature | Remarks |
|-----------|--------------------|--|---|
| Port 0 | 8-bit input/output | Specifiable bit-wise for input/output. Also specifiable for realtime output port. | Serves as RTP0 to RTP7 and pins. |
| Port 2 | 8-bit input | Input port pin. Functions as an external interrupt input. | Serves as NMI, INTP0 to INTP5, INTP6/TI and pins. |
| Port 3 | 5-bit input/output | Specifiable bit-wise for port pins or control pins. | Serves as TxD, RxD, SO/SB0, SI/SB1, SCK and pins. |
| Port 4 | 8-bit input/output | Specifiable in 8-bit units for input or output. Functions as the multiplexed address/data bus (AD0 to AD7) in the external memory expansion mode. | _____ |
| Port 5 | 8-bit input/output | Specifiable bit-wise for input or output. Functions as the address bus (A8 to A15) in the external memory expansion mode. Pins which are not used as the address bus can be used as a port. | _____ |
| Port 7 | 8-bit input | Input port pin. Also functions as analog input to the A/D converter. | Serves as AN0 to AN7 and pins. |
| Port 8 | 6-bit input/output | Specifiable bit-wise for the port pin or control pin. | Functions as TO00 to TO03, TO10 to TO11 and pins. |
| Port 9 | 4-bit input/output | Specifiable bit-wise for input/output. P90 and P91 function as \overline{RD} output and \overline{WR} output, respectively, in the external memory expansion mode. P92 and P93 function as \overline{TAS} output and TMD output, respectively, in the high-speed fetch mode. | _____ |

3.6 CLOCK GENERATOR

The clock generator generates and controls internal system clocks (CLK) supplied to the CPU. It is configured as shown in Figure 3-1.

Figure 3-1. Block Diagram of Clock Generator



- Remarks**
1. f_{xx} : Crystal oscillator frequency
 2. f_x : External clock frequency
 3. f_{CLK} : Internal system clock frequency

The system clock oscillator oscillates by a crystal resonator connected to X1 and X2 pins. It stops oscillating when set to the standby mode (STOP).

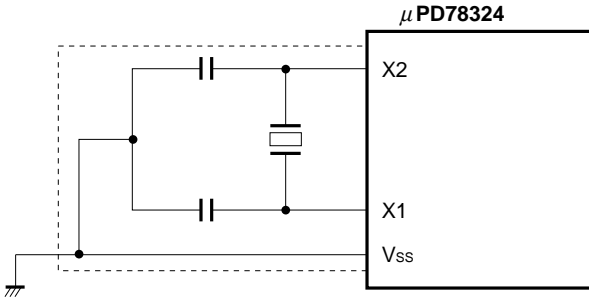
External clocks can be input to the system clock oscillator. In such cases, input a clock signal to the X1 pin and input the reverse phase of the clock signal to the X2 pin. The X2 pin can also be left open.

Caution When using external clocks, do not set the STBC STP bit.

The divider generates internal system clocks (f_{CLK}) by dividing a system clock oscillator output (f_{xx} for crystal oscillation and f_x for external clocks) into two parts.

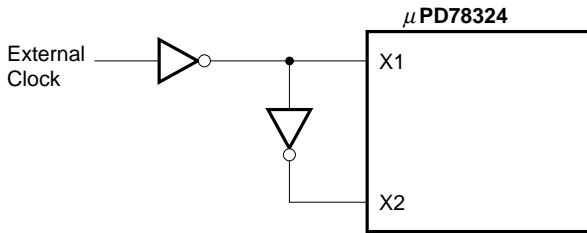
Figure 3-2. Externally-Mounted System Clock Generator

(a) Crystal oscillator

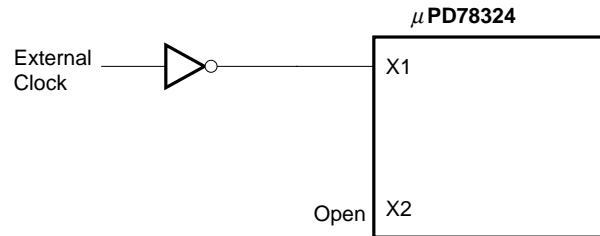


(b) External clock

(i) When the inverted phase of an external clock to be input to the X1 pin is input to the X2 pin



(ii) When X2 pin is left open



Cautions 1. When the system clock oscillator is used, the following points should be noted concerning wiring within broken lines shown in Figure 3-2, in order to prevent the effects of wiring capacitance, etc.

- Keep the wiring as short as possible.
 - Do not cross any other signal lines, and keep clear of lines in which a high fluctuating current flows.
 - Ensure that oscillator capacitor connection points are always at the same potential as Vss. Do not ground in a ground pattern in which a high current flows.
 - Do not take a signal from the oscillator.
2. When an external clock is input to the X1 pin and the X2 pin is left open, ensure that no loads such as wiring capacitance are connected to the X2 pin.

3.7 REALTIME PULSE UNIT (RPU)

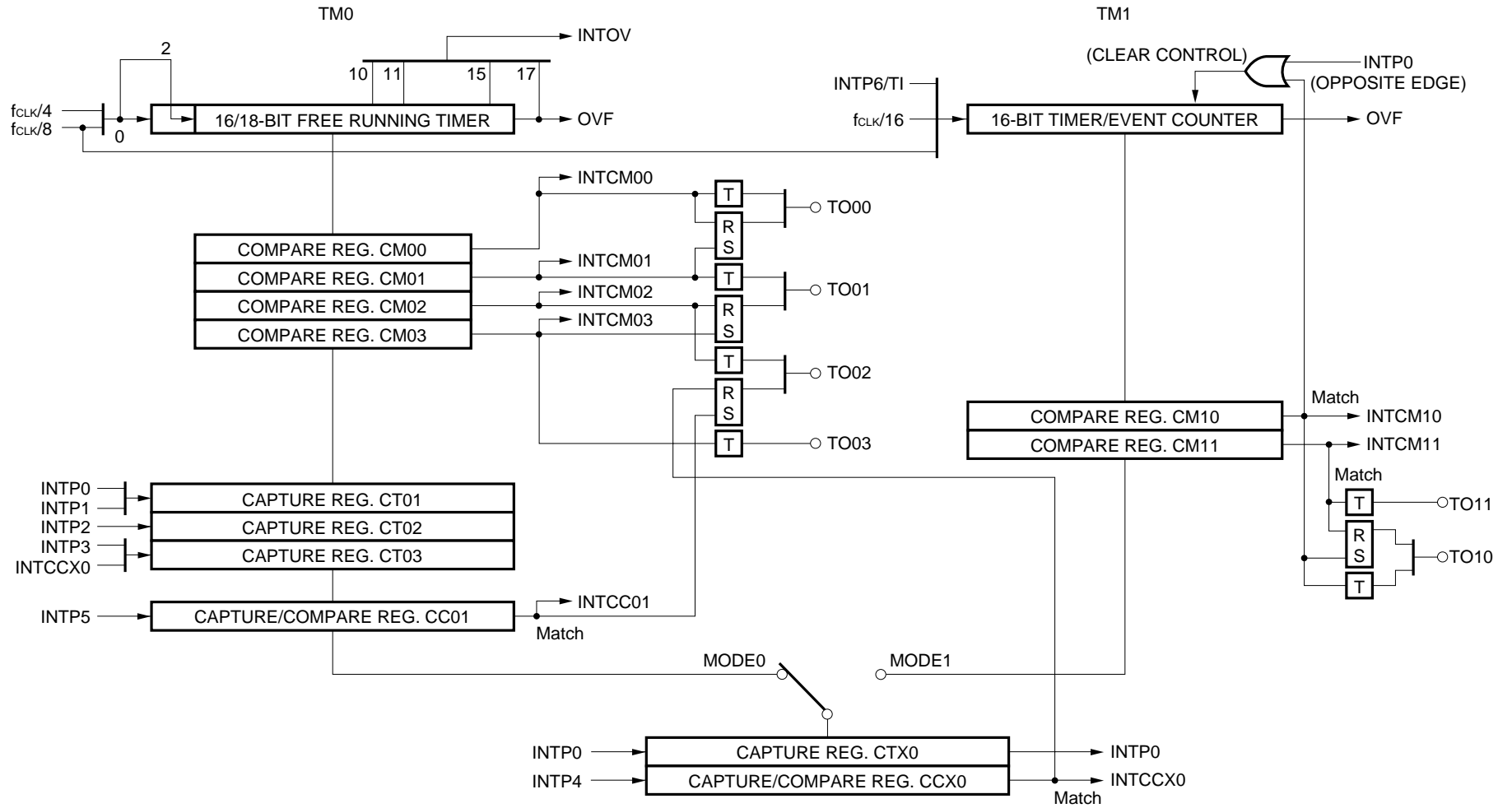
This unit can measure pulse intervals and frequencies, and generate programmable pulse outputs.

It consists mainly of two timers. To flexibly cope with many applications, the configuration of registers connected to the timers can be changed using programs. To meet various applications, toggle output (6 max.) or set/ reset output (4 max.) can be selected as timer output.

3.7.1 Configuration

The realtime pulse unit is configured mainly of timer 0 (TM0) which functions as a 16-bit or 18-bit free running timer and timer 1 (TM1) which functions as a 16-bit timer/event counter shown in Figure 3-3.

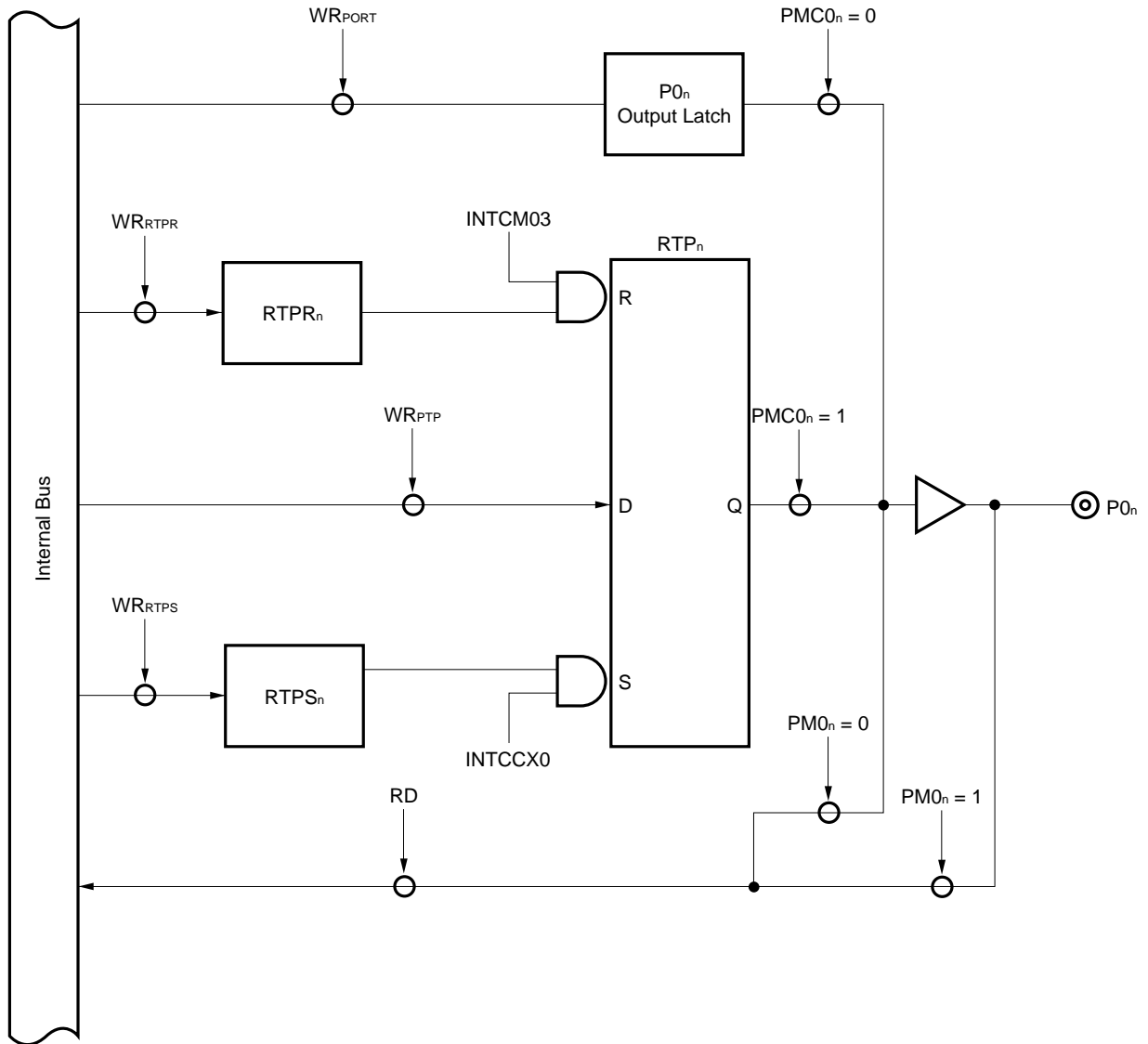
Figure 3-3. Realtime Pulse Unit Configuration



3.7.2 Realtime Output Function

The realtime output port can set/reset port outputs bit-wise in synchronization with the trigger signal transmitted from the RPU (Realtime Pulse Unit). It enables to generate multi-channel synchronous pulses easily.

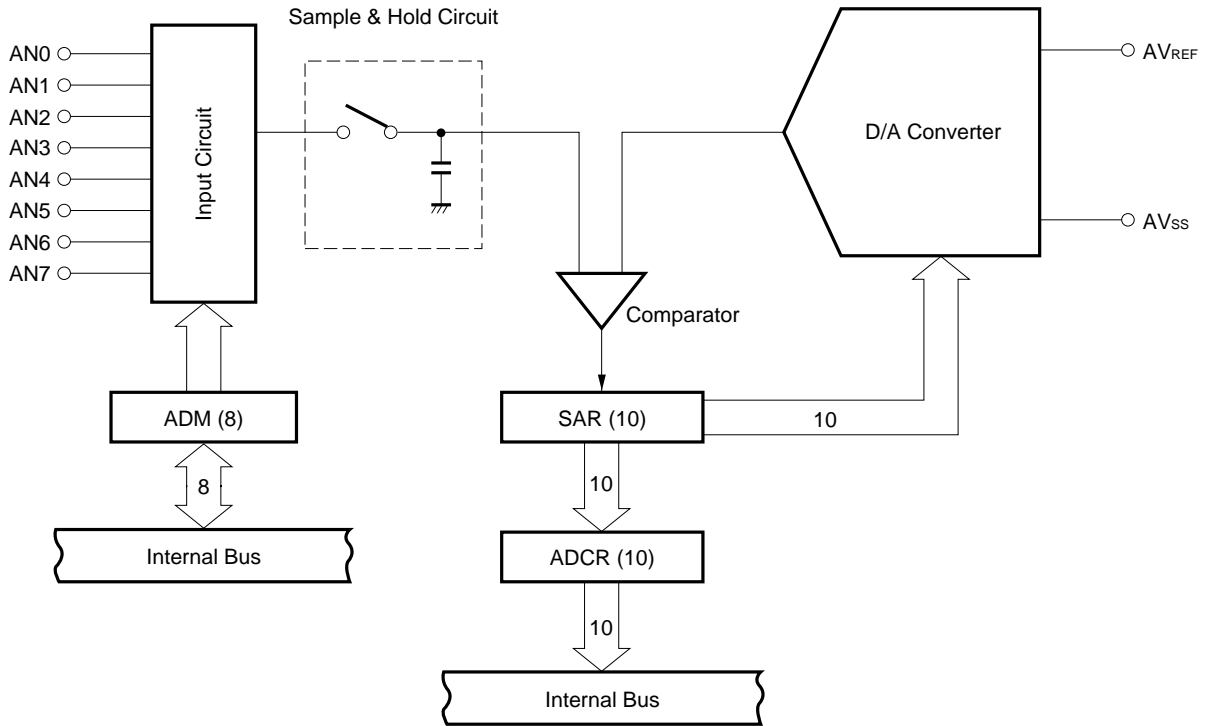
Figure 3-4. Realtime Output Port



3.8 A/D CONVERTER

The μPD78324 incorporates a high-speed, high-resolution 10-bit analog/digital (A/D) converter. This A/D converter is equipped with eight analog inputs (AN0 to AN7) and A/D conversion result register (ADCR) which holds the conversion results. Upon termination of conversion, the interrupt which can start the macro service is generated.

Figure 3-5. A/D Converter Block Diagram



3.9 SERIAL INTERFACE

The μPD78324 is equipped with the following two independent channels for the serial interface function.

- Asynchronous serial interface
- Clock synchronous serial interface
 - 3-wire serial I/O mode
 - Serial bus interface mode (SBI mode)

Since the μPD78324 incorporates a baud rate generator, it can set any serial transfer rate irrespective of the operating frequency. The baud rate generator functions for the 2-channel serial interface.

The serial transfer rate can be selected from 75 bps to 19.2 Kbps by setting the mode register.

Figure 3-6. Asynchronous Serial Interface Block Diagram

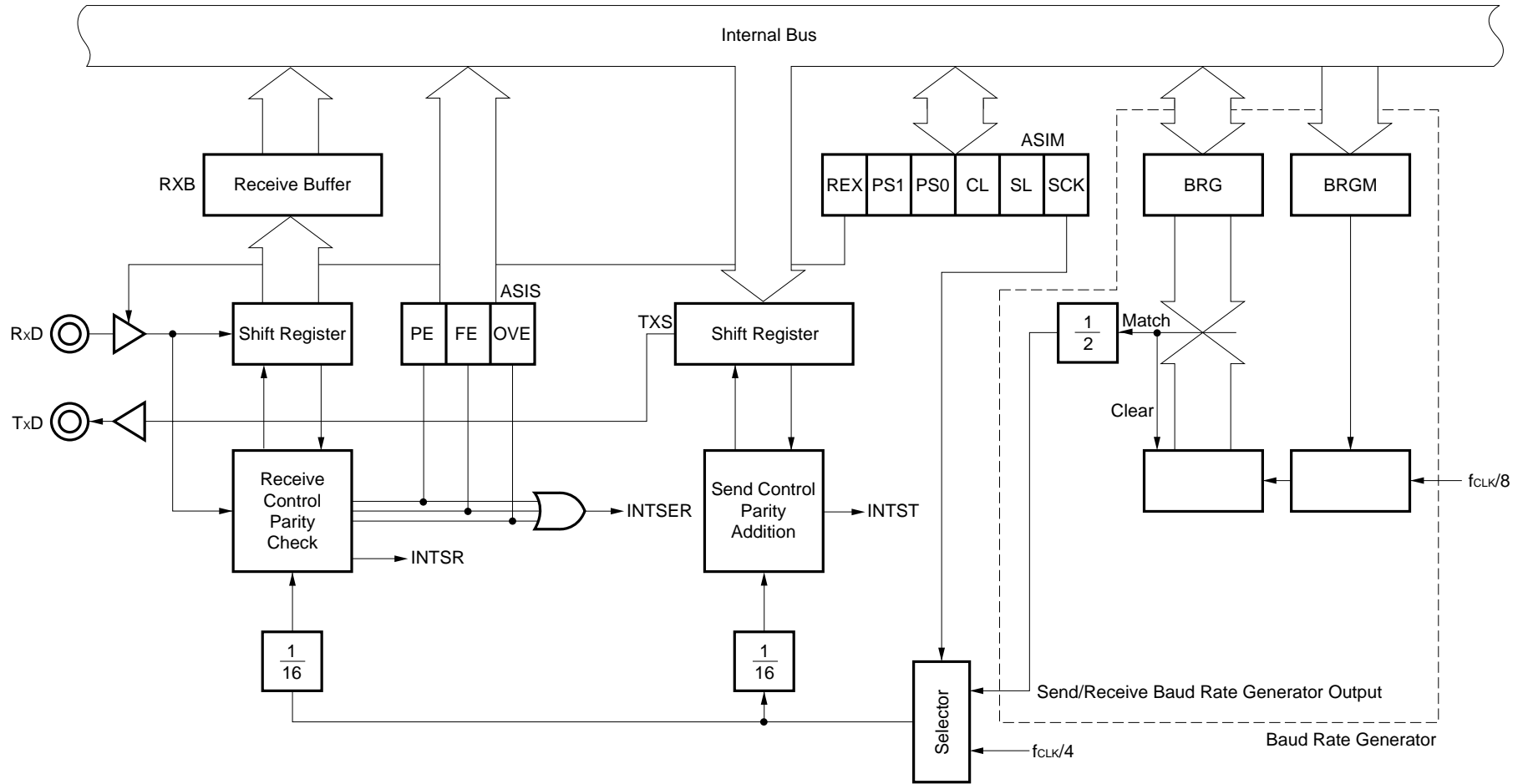
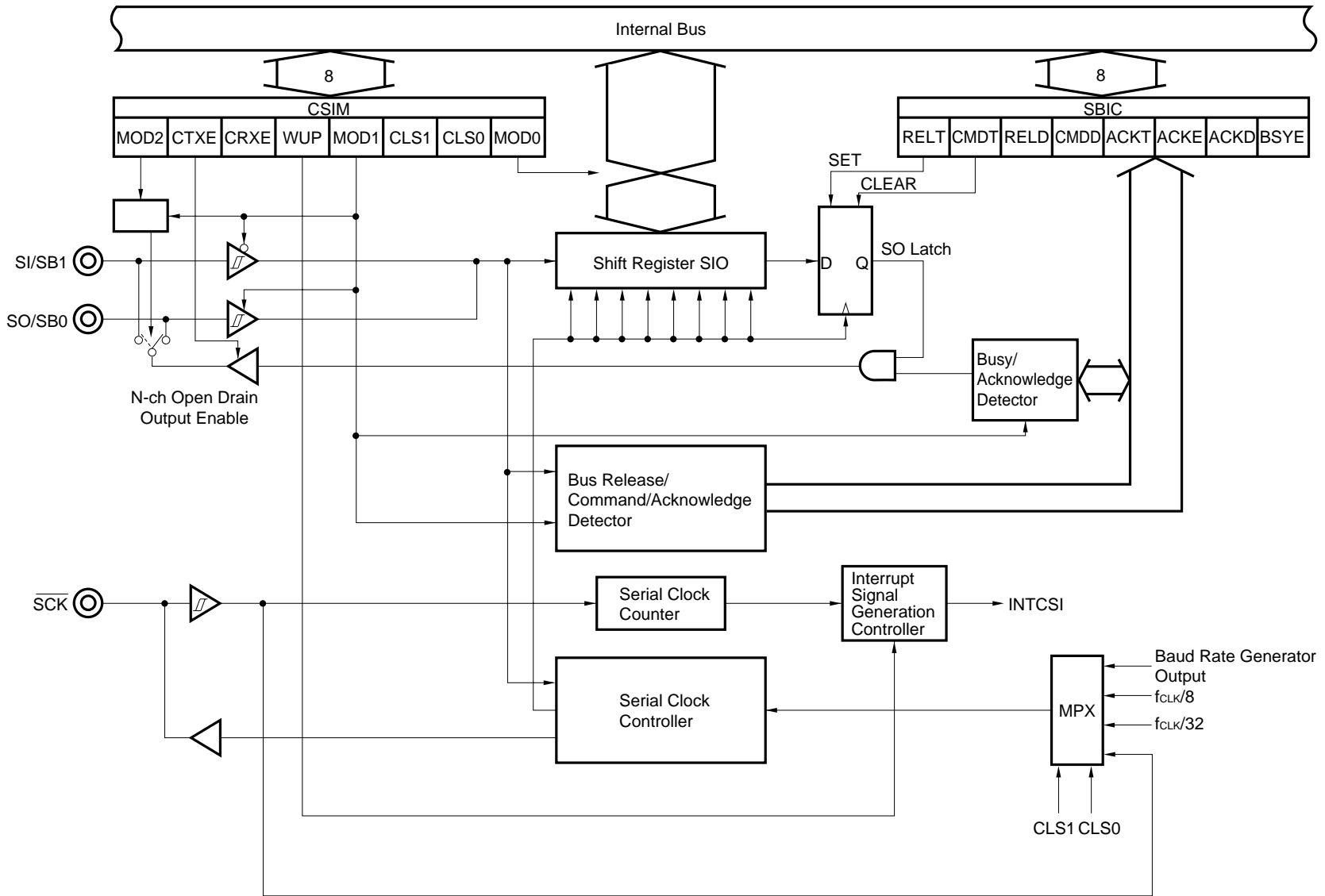


Figure 3-7. Block Diagram of Clock Synchronous Serial Interface



3.10 WATCHDOG TIMER

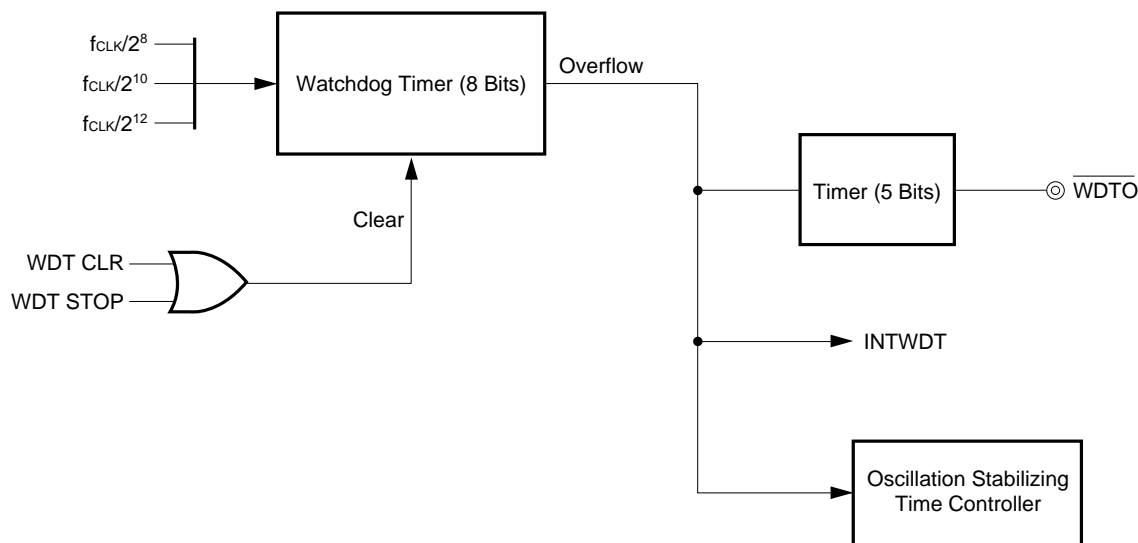
The watchdog timer is used to prevent program overrun and deadlock. Normal operation of the program or system can be confirmed by checking that no watchdog timer interrupt has been generated. Thus, an instruction to clear the watchdog timer (timer start) is set into each program module.

If the watchdog timer clear instruction is not cleared within the time period set into the watchdog timer and the watchdog timer overflows, a watchdog timer interrupt is generated, and a low level is generated to $\overline{\text{WDTO}}$ pin, thereby notifying of an error in the program.

The watchdog timer can also be used to maintain the oscillation stabilizing time of the oscillator after the stop mode has been released.

Figure 3-8 shows the watchdog timer configuration.

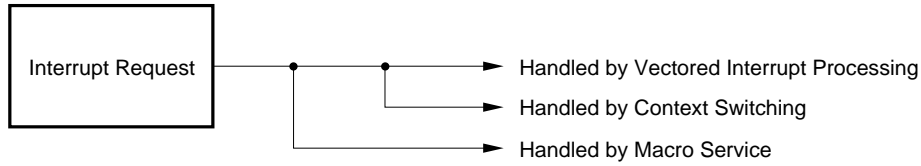
Figure 3-8. Watchdog Timer Configuration



4. INTERRUPT FUNCTIONS

4.1 OVERVIEW

In the μPD78324, various interrupt requests generated externally or from the on-chip peripheral hardware are handled in the following three processing modes.



Interrupt requests are classified into the following three groups.

- Nonmaskable interrupt requests
- Maskable interrupt requests
- Interrupt requests by software

Figure 4-1 shows the maskable interrupt request processing modes. Table 4-1 gives a listing of interrupt factors which can be processed.

Figure 4-1. Interrupt Request Processing Modes

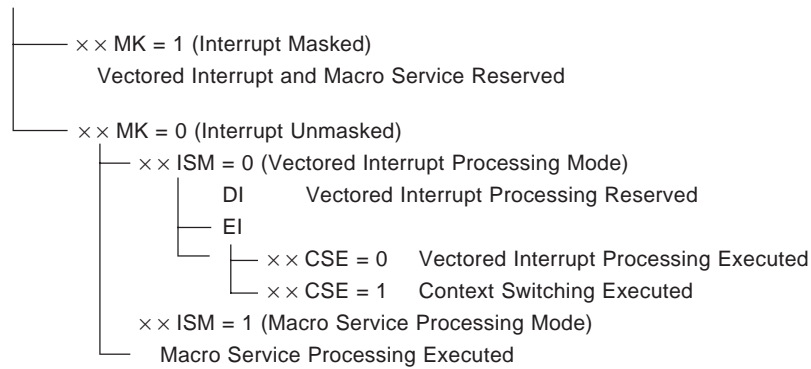


Table 4-1. List of Interrupt Factors

| Interrupt Request Type | Default Priority | Interrupt Factor | | Generator Unit | Macro Service | Vector Table Address |
|------------------------|------------------|------------------------------------|------------------------------------|----------------------|---------------|----------------------|
| | | Request Signal | Function | | | |
| Software | — | — | BRK instruction | — | — | 003EH |
| | — | — | Operation code trap | — | — | 003CH |
| Non-maskable | — | NMI | NMI pin input | (External interrupt) | — | 0002H |
| | — | INTWDT | Watchdog timer | (WDT) | — | 0004H |
| Maskable | 0 | INTOV | Timer 0 overflow | (RPU) | Available | 0006H |
| | 1 | INTP0 | INTP0 pin input | (External) | | 0008H |
| | 2 | INTP1 | INTP1 pin input | (External) | | 000AH |
| | 3 | INTP2 | INTP2 pin input | (External) | | 000CH |
| | 4 | INTP3 | INTP3 pin input | (RPU/exteranl) | | 000EH |
| | 5 | INTP4/INTCCX0 | INTP4 pin input/CCX0 match signal | (RPU/exteranl) | | 0010H |
| | 6 | INTP5/INTCC01 | INTP5 pin input/CC01 match signal | (RPU/exteranl) | | 0012H |
| | 7 | INTP6/TI | INTP6 pin input/TI input | (Exteranl) | | 0014H |
| | 8 | INTCM00 | CM00 match signal | (RPU) | | 0016H |
| | 9 | INTCM01 | CM01 match signal | (RPU) | | 0018H |
| | 10 | INTCM02 | CM02 match signal | (RPU) | | 001AH |
| | 11 | INTCM03 | CM03 match signal | (RPU) | | 001CH |
| | 12 | INTCM10 | CM10 match signal | (RPU) | | 001EH |
| | 13 | INTCM11 | CM11 match signal | (RPU) | | 0020H |
| | 14 | INTSR | Serial receive terminate interrupt | (UART) | | 0024H |
| | 15 | INTST | Serial send terminate interrupt | (UART) | | 0026H |
| | 16 | INTCSI | Serial send/receive interrupt | (CSI) | | 0028H |
| 17 | INTAD | A/D conversion terminate interrupt | (A/D) | 002AH | | |
| — | — | INTSER ^{Note} | Serial receive error signal | (UART) | — | — ^{Note} |
| Reset | — | RESET | Reset input | — | — | 0000H |

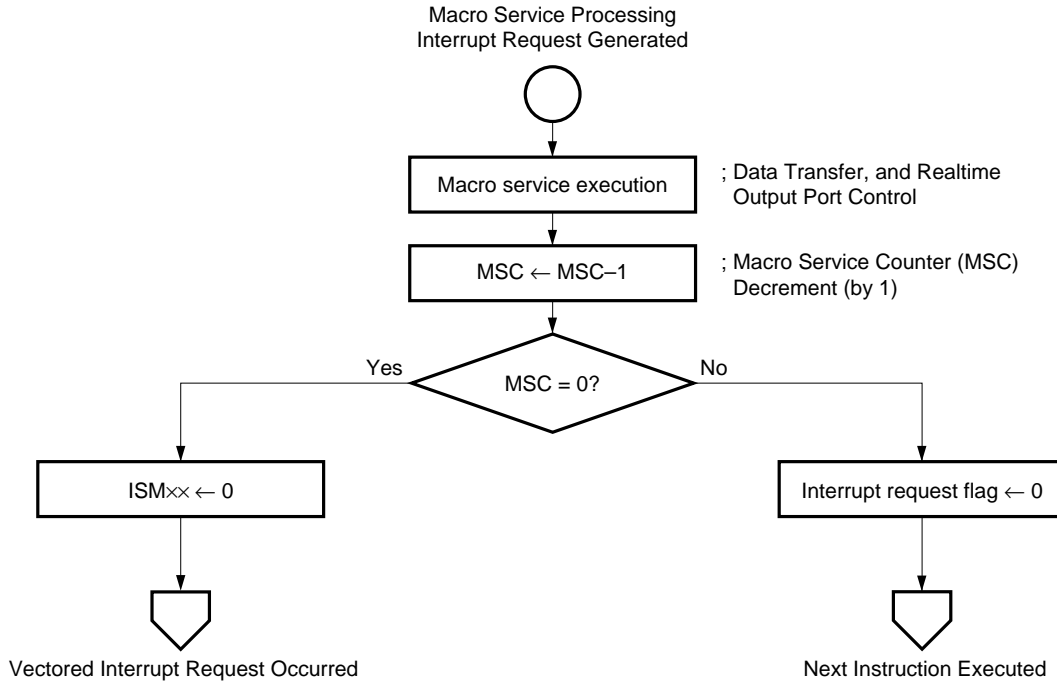
Note This is a test factor. A vectored interrupt is not generated.

4.2 MACRO SERVICE

The macro service function is executed at the interrupt request to carry out data operation and data transfer in hardware terms between the special function register area and the memory space.

Upon startup of the macro service, the CPU stops program execution temporarily. 1-byte/2-byte data operation and transfer are automatically carried out between the special function register (SFR) and the memory. Upon termination of the macro service, the interrupt request flag is reset to 0 and the CPU restarts program execution. When the CPU carries out the macro service operations as many as set into the macro service counter (MSC), a vectored interrupt request is generated.

Figure 4-2. Macro Service Processing Sequence Example



4.3 CONTEXT SWITCHING FUNCTION

This is the function to first select the specified register bank in hardware terms by generating an interrupt request or executing BRKCS instruction, to branch the selected register bank to the vector address prestored in the register bank, and also to stack the current PC and PSW contents into the register bank.

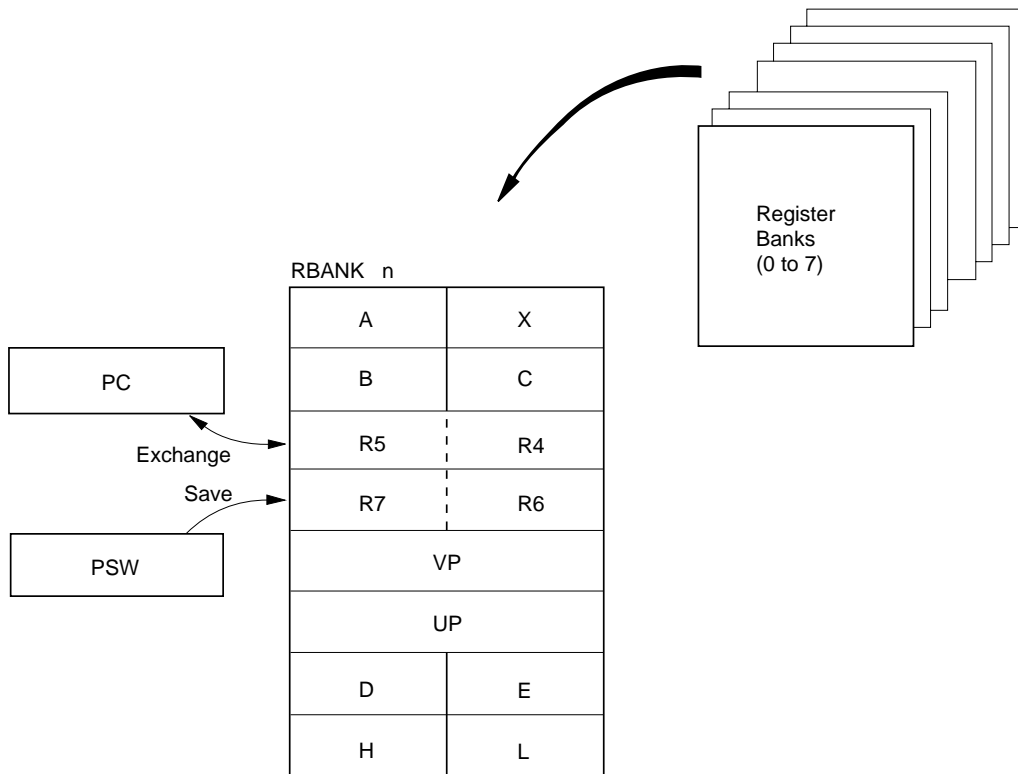
4.3.1 Context Switching Function at Interrupt Request

The context switching function start is enabled by setting the ××CSE bit preset at each interrupt request to 1.

If an unmasked interrupt request for which the context switching function has been enabled is generated in the EI state, the register bank which is specified by the lower 3 bits of the lower address (even address) of the corresponding interrupt vector table address is selected. The vector address prestored in the selected register bank is transferred to the PC, the PC and PSW contents are saved into the register bank, and the operation is branched to the interrupt processing routine.

Return is by means of executing the RETCS instruction.

Figure 4-3. Context Switching at Interrupt Request



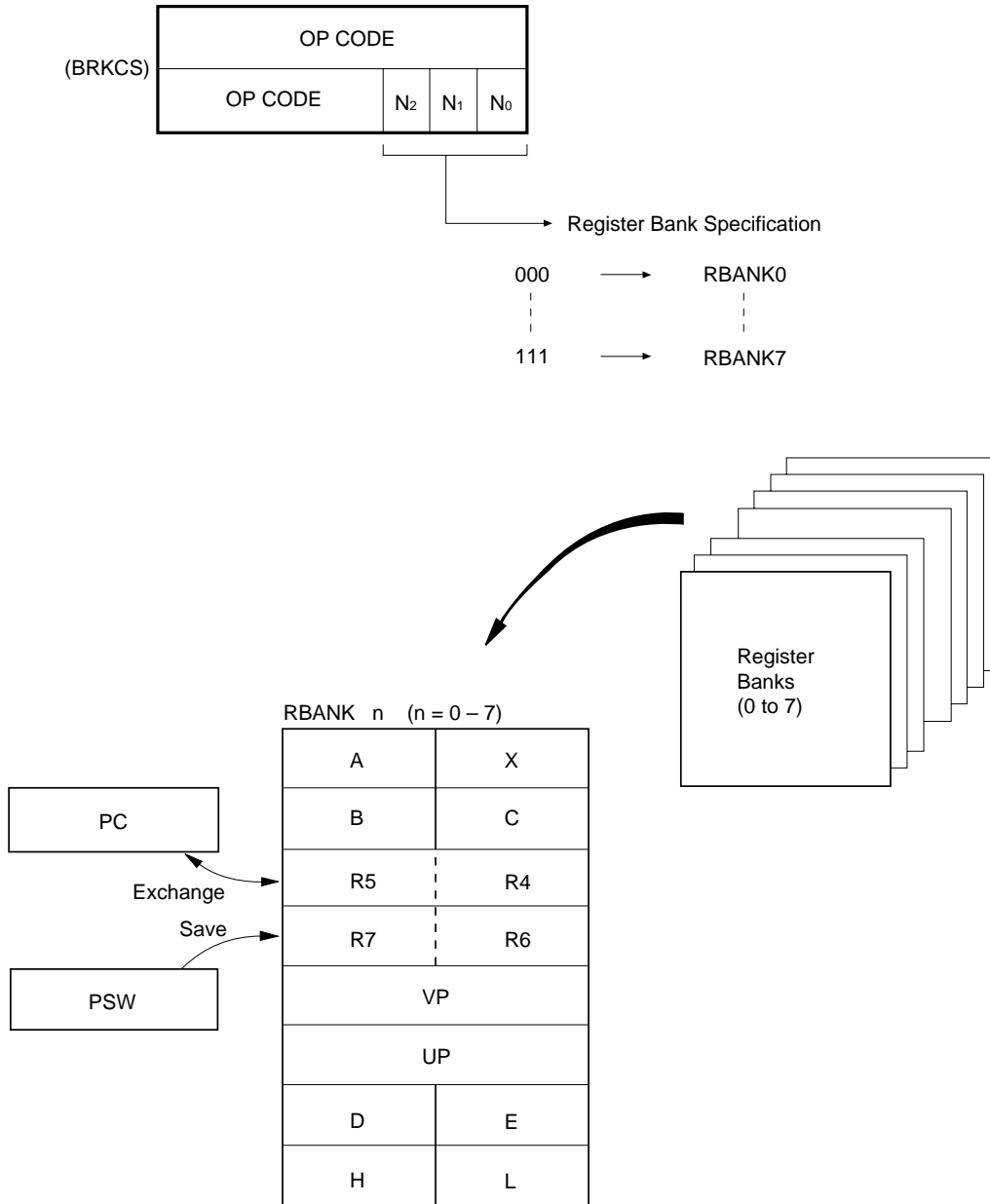
4.3.2 Context Switching Function by BRKCS Instruction

The context switching function can be started by executing BRKCS instruction.

The context switched register bank is specified by the lower 3-bit immediate data of the 2nd operation code of BRKCS instruction. When BRKCS instruction is executed, the register bank specified by the 3-bit immediate data is selected, the vector address prestored in the register bank is set and branched to the PC, and the PC and PSW contents are saved into the register bank.

Return is by means of executing the RETCSB instruction.

Figure 4-4. Context Switching by Execution of BRKCS Instruction



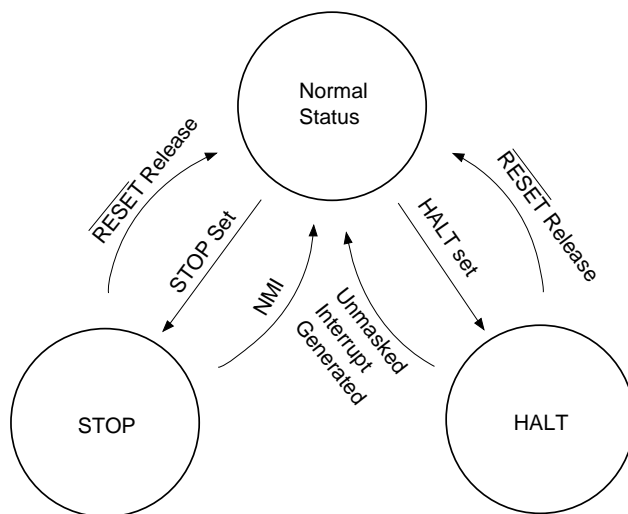
5. STANDBY FUNCTIONS

The μPD78324 has the standby function to decrease the power consumption of the system. The following two modes are available for execution of the standby function.

- HALT mode..... Mode for halting the CPU operation clock. The total power consumption of the system can be decreased by intermittent operation in combination with the normal operating mode.
- STOP mode..... Mode for stopping the whole system by stopping the oscillator. Considerably low power consumption with leak current only can be set.

Each mode is set by the software. Figure 5-1 shows standby mode (STOP/HALT mode) transition.

Figure 5-1. Standby Status Transition



6. EXTERNAL DEVICE EXPANSION FUNCTION

The μPD78324 can expand external devices (data memory, program memory peripheral device) for areas (8000H to FAFFH) except the internal ROM and RAM areas. Table 6-1 and 6-2 show the pin used for external device access and the pin function setting procedure.

Table 6-1. Pin Function Setting (μPD78324)

| EA Pin | Memory Expansion Mode Register | | Fetch Cycle Control Register | Pin Function | | | | | | Remarks |
|--------|--------------------------------|-----|------------------------------|--------------|---------------------------|-----|-----|--------------|-----|---------------------------------|
| | MM0 to MM2 | MM7 | | P40 to P47 | P50 to P57 | P90 | P91 | P92 | P93 | |
| 1 | Port mode | 0 | 00H | General port | | | | | | — |
| | | 1 | Setting prohibited | | | | | | | |
| | Expansion mode | 0 | 00H | AD0 to AD7 | Set to A8 to A15 in steps | RD | WR | General port | | External device connection mode |
| | | 1 | Except 00H | | | | | TAS | TMD | μPD71P301 connection mode |

P50 to P57 pins according to the externally expanded memory size. The memory can be expanded in steps from 256 bytes to about 32K bytes. The pins which are not used as the address bus can be used as the general-purpose input/output port.

Table 6-2. Port and Address Setting for Port 5 (μPD78324)

| | | | | | | | | |
|------|------|------|------|------|------|------|------|-------------------------|
| P57 | P56 | P55 | P54 | P53 | P52 | P51 | P50 | External Address Space |
| Port | Port | Port | Port | Port | Port | Port | Port | 256 bytes or less |
| Port | Port | Port | Port | A11 | A10 | A9 | A8 | 4K bytes or less |
| Port | Port | A13 | A12 | A11 | A10 | A9 | A8 | 16K bytes or less |
| A15 | A14 | A13 | A12 | A11 | A10 | A9 | A8 | About 32K bytes or less |

Table 6-3. Pin Function Setting (μPD78323)

| EA Pin | Memory Expansion Mode Register | Fetch Cycle Control Register | Pin Function | | | | | | Remarks |
|--------|--------------------------------|------------------------------|--------------|-----------|----|----|--------------|-----|---------------------------------|
| | MM7 | | AD0 to AD7 | A8 to A15 | RD | WR | P92 | P93 | |
| ASTB | — | — | AD0 to AD7 | A8 to A15 | RD | WR | TAS | TMD | μPD78324 emulation mode |
| 1 | 0 | 00H | | | | | General port | | External device connection mode |
| | 1 | Except 00H | | | | | TAS | TMD | μPD71P301 connection mode |

7. OPERATION AFTER RESET

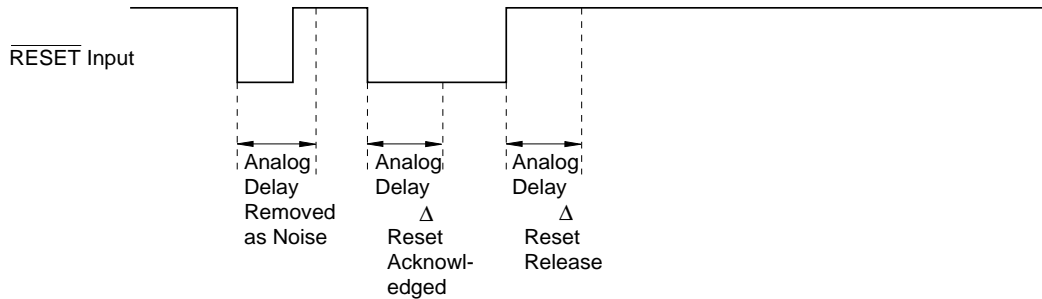
If the $\overline{\text{RESET}}$ input pin is set to the low level, the system reset is applied and each hardware becomes as initialized status (reset status). If $\overline{\text{RESET}}$ input becomes high level, program execution is started. Initialize the contents of various registers in the program as required.

Change the number of cycles for the programmable wait register and the fetch cycle control register in particular.

The $\overline{\text{RESET}}$ input pin is equipped with an analog delay noise suppressor to prevent malfunctioning due to noise.

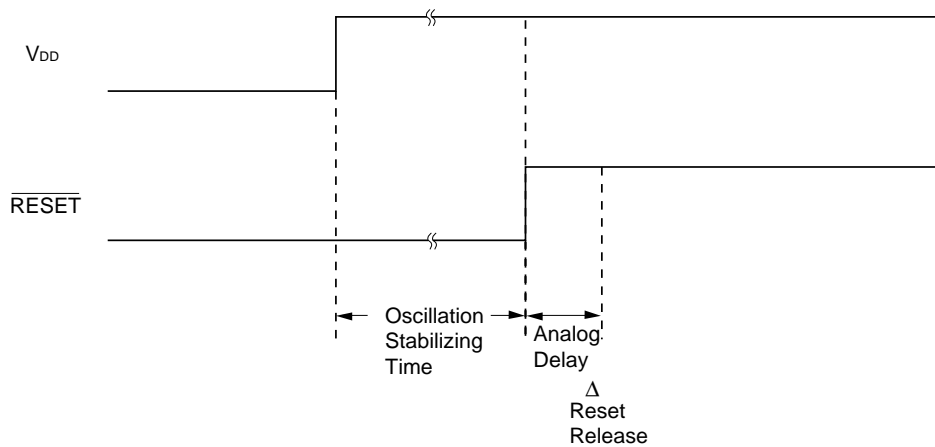
- Cautions**
1. While $\overline{\text{RESET}}$ is active (low level), all pins remain high impedance (except $\overline{\text{WDTO}}$, AV_{REF} , AV_{DD} , AV_{SS} , V_{DD} , V_{SS} , X1 and X2).
 2. If RAM has been expanded externally, mount a pull-up resistor to the $\text{P90}/\overline{\text{RD}}$ and $\text{P91}/\overline{\text{WR}}$ pins. It is possible that the $\text{P90}/\overline{\text{RD}}$ and $\text{P91}/\overline{\text{WR}}$ pins become high impedance resulting in an external RAM contents corruption or input unit damage. In addition, signals may collide on the address/data bus, resulting in the destruction of the input/output circuit.

Figure 7-1. Reset Signal Acknowledge



For reset operation upon power-up, secure the oscillation stabilizing time of about 40 msec from power-up to reset acknowledge as shown in Figure 7-2.

Figure 7-2. Reset Upon Power-Up



8. INSTRUCTION SET

This chapter covers instruction operations.

For the operation codes and the number of instruction execution clock cycles, see *μPD78322 User's Manual (IEU-1248)*.

(1) Operand representation format and description method

In each instruction operand column, enter the operand using the description method for the instruction operand representation format (refer to the assembler specification for details). If two or more factors are included in the description method column, select one factor. The capital alphabetic letters and +, -, #, \$, ! and [] symbols are keywords and should be described as they are.

In case of immediate data, describe appropriate numeric values or labels. When describing labels, make sure to describe #, \$, ! and [] symbols.

Table 8-1. Operand Representation and Description Method

| Representation Format | Description Method |
|--|--|
| r r1 r2 | R0, R1, R2, R3, R4, R5, R6, R7, R8, R9, R10, R11, R12, R13, R14, R15 R0, R1, R2, R3, R4, R5, R6, R7 C, B |
| rp rp1 rp2 | RP0, RP1, RP2, RP3, RP4, RP5, RP6, RP7 RP0, RP1, RP2, RP3, RP4, RP5, RP6, RP7 DE, HL, VP, UP |
| sfr sfrp | Special function register code (see Table 2-2) Special function register code (16-bit operation enable register; see Table 2-2) |
| post | RP0, RP1, RP2, RP3, RP4, RP5/PSW, RP6, RP7 (Two or more instructions can be described. Only PUSH and POP instructions can be described for RP5 and only PUSHU and POPU instructions can be described for PSW.) |
| 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] ; Index mode |
| saddr saddrp | FE20H to FF1FH Immediate data or label FE20H to FF1EH Immediate data (bit0 = 0) or label (for 16-bit operation) |
| \$addr16 !addr16 addr11 addr5 | 0000H to FDFFH Immediate data or label; relative addressing 0000H to FDFFH Immediate data or label; immediate addressing (Up to FFFFH describable by MOV instruction) 800H to FFFH Immediate data or label 40H to 7EH Immediate data (bit0 = 0) ^{Note} or label |
| word byte bit n | 16-bit immediate data or label 8-bit immediate data or label 3-bit immediate data or label 3-bit immediate data (0 to 7) |

Note Do not make work access to bit0 = 1 (odd address).

- Remarks**
1. Although rp and rp1 have the same describable register names, they generate different codes.
 2. r, r1, rp, rp1 and post can be described with absolute names (R0 to R15, RP0 to RP7) as well as functional names (X, A, C, B, E, D, L, H, AX, BC, DE, HL, VP, UP (refer to Table 2-1 for details of the relationships between the absolute and functional names).
 3. Immediate addressing is enabled for all spaces. Relative addressing is only enabled from the first address of the subsequent instruction to the range of -128 to +127.

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | | |
|---------------------|----------|-----------------------------|----------------------|-------------------------|-------|---|----|-----|----|---|
| | | | | | 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 | | | × | × | × | × |
| | | PSWH, #byte | 3 | PSW _H ← byte | | | | | | |
| | | PSWL, A | 2 | PSW _L ← A | | | × | × | × | × |
| | | 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 If STBC and WDM are described for sft, a different dedicated instruction having a different number of bytes is used.

Remark For the symbols in the Flags column, refer to the table below.

| Symbol | Description |
|---------|--|
| (Blank) | No change |
| 0 | Clear to 0. |
| 1 | Set to 1. |
| × | Set/clear according to the result. |
| P | P/V flag operates as a parity flag |
| V | P/V flag operates as an overflow flag. |
| R | The previously stored value is restored. |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | | |
|----------------------|----------------|----------------|--------------|--------------------------------------|------------------------------|---|----|-----|----|---|
| | | | | | 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 opration | ADD | 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 | × | |
| ADDC | | 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 | × | |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|-------------------|-------------|--------------|-------|--|-------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| 8-bit operation | SUB | A, #byte | 2 | $A, CY \leftarrow A - \text{byte}$ | x | x | x | V | x |
| | | saddr, #byte | 3 | $(\text{saddr}), CY \leftarrow (\text{saddr}) - \text{byte}$ | x | x | x | V | x |
| | | sfr, #byte | 4 | $\text{sfr}, CY \leftarrow \text{sfr} - \text{byte}$ | x | x | x | V | x |
| | | r, r1 | 2 | $r, CY \leftarrow r - r1$ | x | x | x | V | x |
| | | A, saddr | 2 | $A, CY \leftarrow A - (\text{saddr})$ | x | x | x | V | x |
| | | A, sfr | 3 | $A, CY \leftarrow A - \text{sfr}$ | x | x | x | V | x |
| | | saddr, saddr | 3 | $(\text{saddr}), CY \leftarrow (\text{saddr}) - (\text{saddr})$ | x | x | x | V | x |
| | | A, mem | 2-4 | $A, CY \leftarrow A - (\text{mem})$ | x | x | x | V | x |
| | | mem, A | 2-4 | $(\text{mem}), CY \leftarrow (\text{mem}) - A$ | x | x | x | V | x |
| | SUBC | A, #byte | 2 | $A, CY \leftarrow A - \text{byte} - CY$ | x | x | x | V | x |
| | | saddr, #byte | 3 | $(\text{saddr}), CY \leftarrow (\text{saddr}) - \text{byte} - CY$ | x | x | x | V | x |
| | | sfr, #byte | 4 | $\text{sfr}, CY \leftarrow \text{sfr} - \text{byte} - CY$ | x | x | x | V | x |
| | | r, r1 | 2 | $r, CY \leftarrow r - r1 - CY$ | x | x | x | V | x |
| | | A, saddr | 2 | $A, CY \leftarrow A - (\text{saddr}) - CY$ | x | x | x | V | x |
| | | A, sfr | 3 | $A, CY \leftarrow A - \text{sfr} - CY$ | x | x | x | V | x |
| | | saddr, saddr | 3 | $(\text{saddr}), CY \leftarrow (\text{saddr}) - (\text{saddr}) - CY$ | x | x | x | V | x |
| | | A, mem | 2-4 | $A, CY \leftarrow A - (\text{mem}) - CY$ | x | x | x | V | x |
| | | mem, A | 2-4 | $(\text{mem}), CY \leftarrow (\text{mem}) - A - CY$ | x | x | x | V | x |
| | AND | A, #byte | 2 | $A \leftarrow A \wedge \text{byte}$ | x | x | | P | |
| | | saddr, #byte | 3 | $(\text{saddr}) \leftarrow (\text{saddr}) \wedge \text{byte}$ | x | x | | P | |
| | | sfr, #byte | 4 | $\text{sfr} \leftarrow \text{sfr} \wedge \text{byte}$ | x | x | | P | |
| | | r, r1 | 2 | $r \leftarrow r \wedge r1$ | x | x | | P | |
| | | A, saddr | 2 | $A \leftarrow A \wedge (\text{saddr})$ | x | x | | P | |
| | | A, sfr | 3 | $A \leftarrow A \wedge \text{sfr}$ | x | x | | P | |
| | | saddr, saddr | 3 | $(\text{saddr}) \leftarrow (\text{saddr}) \wedge (\text{saddr})$ | x | x | | P | |
| | | A, mem | 2-4 | $A \leftarrow A \wedge (\text{mem})$ | x | x | | P | |
| | | mem, A | 2-4 | $(\text{mem}) \leftarrow (\text{mem}) \wedge A$ | x | x | | P | |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|-------------------|------------|--------------------|---|--|-------|---|----|-----|----|
| | | | | | 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 | $(\text{mem}) \leftarrow (\text{mem}) \vee A$ | 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 | | |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|-------------------------|--------------|----------------|-------|--|-------|---|----|-----|----|
| | | | | | 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 ← A × r1 | | | | | |
| | DIVUM | r1 | 2 | AX(quotient), r1(remainder) ← AX ÷ r1 | | | | | |
| | MULUW | rp1 | 2 | AX(higher 16 bits), rp1(lower 16 bits) ← AX × rp1 | | | | | |
| | DIVUX | rp1 | 2 | AXDE(quotient), rp1(remainder) ← AXDE ÷ rp1 | | | | | |
| Signed multiplication | MULW | rp1 | 2 | AX(higher 16 bits), rp1(lower 16 bits) ← AX × rp1 | | | | | |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|-------------------|----------|---------|------------------------------------|--|-------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| Increase/decrease | INC | r1 | 1 | $r1 \leftarrow r1 + 1$ | x | x | x | V | |
| | | saddr | 2 | $(saddr) \leftarrow (saddr) + 1$ | x | x | x | V | |
| | DEC | r1 | 1 | $r1 \leftarrow r1 - 1$ | x | x | x | V | |
| | | saddr | 2 | $(saddr) \leftarrow (saddr) - 1$ | x | x | x | V | |
| | INCW | rp2 | 1 | $rp2 \leftarrow rp2 + 1$ | | | | | |
| | | saddrp | 3 | $(saddrp) \leftarrow (saddrp) + 1$ | | | | | |
| DECW | rp2 | 1 | $rp2 \leftarrow rp2 - 1$ | | | | | | |
| | saddrp | 3 | $(saddrp) \leftarrow (saddrp) - 1$ | | | | | | |
| Shift-rotate | ROR | r1, n | 2 | $(CY, r17 \leftarrow r10, r1_{m-1} \leftarrow r1_m) \times n \text{ times}$ | | | | P | x |
| | ROL | r1, n | 2 | $(CY, r1_0 \leftarrow r17, r1_{m+1} \leftarrow r1_m) \times n \text{ times}$ | | | | P | x |
| | RORC | r1, n | 2 | $(CY \leftarrow r1_0, r17 \leftarrow CY, r1_{m-1} \leftarrow r1_m) \times n \text{ times}$ | | | | P | x |
| | ROLC | r1, n | 2 | $(CY \leftarrow r17, r1_0 \leftarrow CY, r1_{m+1} \leftarrow r1_m) \times n \text{ times}$ | | | | P | x |
| | SHR | r1, n | 2 | $(CY \leftarrow r1_0, r17 \leftarrow 0, r1_{m-1} \leftarrow r1_m) \times n \text{ times}$ | x | x | 0 | P | x |
| | SHL | r1, n | 2 | $(CY \leftarrow r17, r1_0 \leftarrow 0, r1_{m+1} \leftarrow r1_m) \times n \text{ times}$ | x | x | 0 | P | x |
| | SHRW | rp1, n | 2 | $(CY \leftarrow rp1_0, rp1_{15} \leftarrow 0, rp1_{m-1} \leftarrow rp1_m) \times n \text{ times}$ | x | x | 0 | P | x |
| | SHLW | rp1, n | 2 | $(CY \leftarrow rp1_{15}, rp1_0 \leftarrow 0, rp1_{m+1} \leftarrow rp1_m) \times n \text{ times}$ | x | x | 0 | P | x |
| | ROR4 | [rp1] | 2 | $A_{3-0} \leftarrow (rp1)_{3-0},$ $(rp1)_{7-4} \leftarrow A_{3-0},$ $(rp1)_{3-0} \leftarrow (rp1)_{7-4}$ | | | | | |
| | ROL4 | [rp1] | 2 | $A_{3-0} \leftarrow (rp1)_{7-4},$ $(rp1)_{3-0} \leftarrow A_{3-0},$ $(rp1)_{7-4} \leftarrow (rp1)_{3-0}$ | | | | | |
| BCD calibration | ADJBA | | 2 | Decimal Adjust Accumulator | x | x | x | P | x |
| | ADJBS | | | | | | | | |
| Data conversion | CVTBW | | 1 | When $A_7 = 0, X \leftarrow A, A \leftarrow 00H$ When $A_7 = 1, X \leftarrow A, A \leftarrow FFH$ | | | | | |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|-------------------|----------|-----------------|-------|--|-------|---|----|-----|----|
| | | | | | 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 | $CY \leftarrow \text{PSW}_H.\text{bit}$ | | | | | x |
| | | CY, PSWL. bit | 2 | $CY \leftarrow \text{PSW}_L.\text{bit}$ | | | | | 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 | $\text{PSW}_H.\text{bit} \leftarrow CY$ | | | | | |
| | | PSWL. bit, CY | 2 | $\text{PSW}_L.\text{bit} \leftarrow CY$ | | | | | |
| | 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 | $CY \leftarrow CY \wedge \text{PSW}_H.\text{bit}$ | | | | | x |
| | | CY, /PSWH. bit | 2 | $CY \leftarrow CY \wedge \overline{\text{PSW}_H.\text{bit}}$ | | | | | x |
| | | CY, PSWL. bit | 2 | $CY \leftarrow CY \wedge \text{PSW}_L.\text{bit}$ | | | | | x |
| | | CY, /PSWL. bit | 2 | $CY \leftarrow CY \wedge \overline{\text{PSW}_L.\text{bit}}$ | | | | | 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 | $CY \leftarrow CY \vee \text{PSW}_H.\text{bit}$ | | | | | x |
| | | CY, /PSWH. bit | 2 | $CY \leftarrow CY \vee \overline{\text{PSW}_H.\text{bit}}$ | | | | | x |
| | | CY, PSWL. bit | 2 | $CY \leftarrow CY \vee \text{PSW}_L.\text{bit}$ | | | | | x |
| | | CY, /PSWL. bit | 2 | $CY \leftarrow CY \vee \overline{\text{PSW}_L.\text{bit}}$ | | | | | x |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|-------------------|-------------|----------------|-------|---|-------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| Bit manipulation | XOR1 | CY, saddr. bit | 3 | $CY \leftarrow CY \nabla (\text{saddr.bit})$ | | | | | x |
| | | CY, sfr. bit | 3 | $CY \leftarrow CY \nabla \text{sfr.bit}$ | | | | | x |
| | | CY, A. bit | 2 | $CY \leftarrow CY \nabla A.\text{bit}$ | | | | | x |
| | | CY, X. bit | 2 | $CY \leftarrow CY \nabla X.\text{bit}$ | | | | | x |
| | | CY, PSWH. bit | 2 | $CY \leftarrow CY \nabla \text{PSWH.bit}$ | | | | | x |
| | | CY, PSWL. bit | 2 | $CY \leftarrow CY \nabla \text{PSWL.bit}$ | | | | | x |
| | SET1 | saddr. bit | 2 | $(\text{saddr.bit}) \leftarrow 1$ | | | | | |
| | | sfr. bit | 3 | $\text{sfr.bit} \leftarrow 1$ | | | | | |
| | | A. bit | 2 | $A.\text{bit} \leftarrow 1$ | | | | | |
| | | X. bit | 2 | $X.\text{bit} \leftarrow 1$ | | | | | |
| | | PSWH. bit | 2 | $\text{PSWH.bit} \leftarrow 1$ | | | | | |
| | | PSWL. bit | 2 | $\text{PSWL.bit} \leftarrow 1$ | | x | x | x | x |
| | CLR1 | saddr. bit | 2 | $(\text{saddr.bit}) \leftarrow 0$ | | | | | |
| | | sfr. bit | 3 | $\text{sfr.bit} \leftarrow 0$ | | | | | |
| | | A. bit | 2 | $A.\text{bit} \leftarrow 0$ | | | | | |
| | | X. bit | 2 | $X.\text{bit} \leftarrow 0$ | | | | | |
| | | PSWH. bit | 2 | $\text{PSWH.bit} \leftarrow 0$ | | | | | |
| | | PSWL. bit | 2 | $\text{PSWL.bit} \leftarrow 0$ | | x | x | x | x |
| | 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 | $A.\text{bit} \leftarrow \overline{A.\text{bit}}$ | | | | | |
| | | X. bit | 2 | $X.\text{bit} \leftarrow \overline{X.\text{bit}}$ | | | | | |
| | | PSWH. bit | 2 | $\text{PSWH.bit} \leftarrow \overline{\text{PSWH.bit}}$ | | | | | |
| | | PSWL. bit | 2 | $\text{PSWL.bit} \leftarrow \overline{\text{PSWL.bit}}$ | | x | x | x | x |
| | SET1 | CY | 1 | $CY \leftarrow 1$ | | | | | 1 |
| | CLR1 | CY | 1 | $CY \leftarrow 0$ | | | | | 0 |
| | NOT1 | CY | 1 | $CY \leftarrow \overline{CY}$ | | | | | x |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|--------------------|--------------|-----------|---|---|-------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| Call-return | CALL | !addr16 | 3 | (SP-1) ← (PC+3) _H , (SP-2) ← (PC+3) _L , PC ← addr16, SP ← SP-2 | | | | | |
| | CALLF | !addr11 | 2 | (SP-1) ← (PC+2) _H , (SP-2) ← (PC+2) _L , PC ₁₅₋₁₁ ← 00001, PC ₁₀₋₀ ← addr11, SP ← SP-2 | | | | | |
| | CALLT | [addr5] | 1 | (SP-1) ← (PC+1) _H , (SP-2) ← (PC+1) _L , PC _H ← (TPF, 00000000, addr5+1), PC _L ← (TPF, 00000000, addr5), SP ← SP-2 | | | | | |
| | CALL | rp1 | 2 | (SP-1) ← (PC+2) _H , (SP-2) ← (PC+2) _L , PC _H ← rp1 _H , PC _L ← rp1 _L , SP ← SP-2 | | | | | |
| | | [rp1] | 2 | (SP-1) ← (PC+2) _H , (SP-2) ← (PC+2) _L , PC _H ← (rp1+1), PC _L ← (rp1), SP ← SP-2 | | | | | |
| | BRK | | 1 | (SP-1) ← PSW _H , (SP-2) ← PSW _L (SP-3) ← (PC+1) _H , (SP-4) ← (PC+1) _L , PC _L ← (003EH), PC _H ← (003FH), SP ← SP-4 IE ← 0 | | | | | |
| | RET | | 1 | PC _L ← (SP), PC _H ← (SP+1), SP ← SP+2 | | | | | |
| | RETB | | 1 | PC _L ← (SP), PC _H ← (SP+1) PSW _L ← (SP+2), PSW _H ← (SP+3) SP ← SP+4 | R | R | R | R | R |
| RETI | | 1 | PC _L ← (SP), PC _H ← (SP+1) PSW _L ← (SP+2), PSW _H ← (SP+3) SP ← SP+4 | R | R | R | R | R | |
| Stack manipulation | PUSH | sfrp | 3 | (SP-1) ← sfr _H (SP-2) ← sfr _L SP ← SP-2 | | | | | |
| | | post | 2 | {(SP-1) ← post _H , (SP-2) ← post _L , SP ← SP-2} × n times ^{Note} | | | | | |
| | | PSW | 1 | (SP-1) ← PSW _H , (SP-2) ← PSW _L , SP ← SP-2 | | | | | |
| | PUSHU | post | 2 | {(UP-1) ← post _H , (UP-2) ← post _L , UP ← UP-2} × n times ^{Note} | | | | | |
| | POP | sfrp | 3 | sfr _L ← (SP) sfr _H ← (SP+1) SP ← SP+2 | | | | | |
| | | post | 2 | {post _L ← (SP), post _H ← (SP+1), SP ← SP+2} × n times ^{Note} | | | | | |
| | | PSW | 1 | PSW _L ← (SP), PSW _H ← (SP+1), SP ← SP+2 | R | R | R | R | R |
| | POPU | post | 2 | {post _L ← (UP), post _H ← (UP+1), UP ← UP+2} × n times ^{Note} | | | | | |
| | MOVW | SP, #word | 4 | SP ← word | | | | | |
| | | SP, AX | 2 | SP ← AX | | | | | |
| | | AX, SP | 2 | AX ← SP | | | | | |
| INCW | SP | 2 | SP ← SP+1 | | | | | | |
| DECW | SP | 2 | SP ← SP-1 | | | | | | |
| Special | CHKL | sfr | 3 | (pin level) ∨ (signal level before output buffer) | × | × | | P | |
| | CHKLA | sfr | 3 | A ← (pin level) ∨ (signal level before output buffer) | × | × | | P | |

Note n indicates the number of registers described as post.

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|----------------------|-----------------------|-----------------------|-----------------------------------|---|-------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| 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 | | | | | | | | |
| | BNC | \$ addr16 | 2 | PC ← PC+2+jdisp8 if CY=0 | | | | | |
| | BNL | | | | | | | | |
| | BZ | \$ addr16 | 2 | PC ← PC+2+jdisp8 if Z=1 | | | | | |
| | BE | | | | | | | | |
| | BNZ | \$ addr16 | 2 | PC ← PC+2+jdisp8 if Z=0 | | | | | |
| | BNE | | | | | | | | |
| | BV | \$ addr16 | 2 | PC ← PC+2+jdisp8 if P/V=1 | | | | | |
| | BPE | | | | | | | | |
| | BNV | \$ addr16 | 2 | PC ← PC+2+jdisp8 if P/V=0 | | | | | |
| | BPO | | | | | | | | |
| | 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 | | | | | | |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|--------------------|------------------|----------------------|---|--|-------|---|----|-----|----|
| | | | | | 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 r2≠0 | | | | | | |
| | saddr, \$ addr16 | 3 | (saddr) ← (saddr)-1, then PC ← PC+3+jdisp8 if (saddr) ≠0 | | | | | | |
| Context switching | BRKCS | RBn | 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 |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|-------------------|----------------|------------------|-------|---|-------|---|----|-----|----|
| | | | | | 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 | × |

| Instruction Group | Mnemonic | Operand | Bytes | Operation | Flags | | | | |
|-------------------|----------------|------------------|----------------------------|--|-------|---|----|-----|----|
| | | | | | S | Z | AC | P/V | CY |
| String | 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 | × |
| | CMPMNC | [DE +], A | 2 | (DE +) – A, C ← C–1 End if C=0 or CY=1 | × | × | × | V | × |
| | | [DE –], A | 2 | (DE –) – A, C ← C–1 End if C=0 or CY=1 | × | × | × | V | × |
| | CMPBKNC | [DE +], [HL +] | 2 | (DE +) – (HL +), C ← C–1 End if C=0 or CY=1 | × | × | × | V | × |
| | | [DE –], [HL –] | 2 | (DE –) – (HL –), C ← C–1 End if C=0 or CY=1 | × | × | × | V | × |
| CPU control | MOV | STBC, #byte | 4 | STBC ← byte ^{Note} | | | | | |
| | | WDM, #byte | 4 | WDM ← byte ^{Note} | | | | | |
| | SWRS | | 1 | RSS ← $\overline{\text{RSS}}$ | | | | | |
| | 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 operation code of STBC register and WDM register operation instructions is abnormal, an operation code trap interrupt is generated.

Operation in the event of 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. ELECTRICAL SPECIFICATIONS

ABSOLUTE MAXIMUM RATINGS (T_A = 25 °C)

| Parameter | Symbol | Test Conditions | Rating | Unit |
|---------------------------------------|-------------------|--|--------------------------------|------|
| Supply voltage 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 | Note 1 | -0.5 to V _{DD} + 0.5 | V |
| Output voltage | V _O | | -0.5 to V _{DD} + 0.5 | V |
| Output current low | I _{OL} | All output pins | 4.0 | mA |
| | | All output pins total | 90 | mA |
| Output current high | I _{OH} | All output pins | -1.0 | mA |
| | | All output pins total | -20 | mA |
| Analog input voltage | V _{IAN} | Note 2 AV _{DD} > V _{DD} | -0.5 to V _{DD} + 0.5 | V |
| | | V _{DD} ≥ AV _{DD} | -0.5 to AV _{DD} + 0.5 | |
| A/D converter reference input voltage | AV _{REF} | AV _{DD} > V _{DD} | -0.5 to V _{DD} + 0.3 | V |
| | | V _{DD} ≥ AV _{DD} | -0.5 to AV _{DD} + 0.3 | |
| Operating ambient temperature | T _A | | -10 to + 70 | °C |
| Storage temperature | T _{stg} | | -65 to + 150 | °C |

- Notes** 1. Except the pin described in **Note 2**.
 2. P70/ANI0 to P77/ANI7 pins

Caution If the absolute maximum rating of any one of the parameters is exceeded even momentarily, the quality of the product may be degraded. In other words, the product may be physically damaged if any of the absolute maximum ratings is exceeded. Be sure to use the product without exceeding these ratings. ★

RECOMMENDED OPERATING CONDITION

| Oscillation frequency | T _A | V _{DD} |
|----------------------------------|----------------|-----------------|
| 8 MHz ≤ f _{xx} ≤ 16 MHz | -10 to +70 °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 Unmeasured pins returned to 0 V. | | | 10 | pF |
| Output capacitance | C _O | | | | 20 | pF |
| I/O capacitance | C _{IO} | | | | 20 | pF |

OSCILLATOR CHARACTERISTICS (T_A = -10 to +70 °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}) | 8 | 16 | MHz |
| External clock | | X1 input frequency (f _x) | 8 | 16 | MHz |
| | | X1 input rise/fall time (t _{xR} , t _{xF}) | 0 | 20 | ns |
| | | X1 input high/low level width (t _{wxH} , t _{wxL}) | 25 | 80 | ns |

★ **Caution** When using the system clock oscillation circuit, wire the part encircled in the dotted line in the following manner to avoid the influence of the wiring capacity, etc.

- Make the wiring as short as possible.
- Avoid intersecting other signal conductors. Avoid approaching lines in which very high fluctuating currents run.
- Make sure that the grounding point of the oscillation circuit capacitor always has the same electrical potential as V_{SS}. Avoid grounding with a grand pattern in which very high currents run.
- Do not fetch signals from the oscillation circuit.

RECOMMENDED OSCILLATOR CONSTANT

CERAMIC RESONATOR

| Manufacturer | Product Name | Frequency [MHz] | Recommended Constant | |
|-----------------------|----------------|-----------------|----------------------|---------|
| | | | C1 [pF] | C2 [pF] |
| Murata Mfg. Co., Ltd. | CSA8.00MT | 8.0 | 30 | 30 |
| | CSA12.0MT | 12.0 | | |
| | CSA14.74MXZ040 | 14.74 | 15 | 15 |
| | CSA16.00MXZ040 | 16.0 | | |
| | CST8.00MTW | 8.0 | On-chip | On-chip |
| | CST12.0MTW | 12.0 | | |
| CST14.74MXW0C3 | 17.74 | | | |
| CST16.00MXW0C3 | 16.0 | | | |

CRYSTAL RESONATOR

| Manufacturer | Product Name | Frequency [MHz] | Recommended Constant | |
|-------------------|--------------|-----------------|----------------------|---------|
| | | | C1 [pF] | C2 [pF] |
| Kinseki Co., Ltd. | HC49/U-S | 8 to 16 | 10 | 10 |
| | HC49/U | | | |

DC CHARACTERISTICS (T_A = -10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V)

| Parameter | Symbol | Test Conditions | MIN. | TYP. | MAX. | Unit | |
|--------------------------------|-------------------|--|---------------------------------|------|------|------|----|
| Input voltage low | V _{IL} | | 0 | | 0.8 | V | |
| Input voltage high | V _{IH1} | Note 1 | 2.2 | | | V | |
| | V _{IH2} | Note 2 | 0.8V _{DD} | | | | |
| Output voltage low | V _{OL} | I _{OL} = 2.0 mA | | | 0.45 | V | |
| Output voltage high | V _{OH} | I _{OH} = -400 μA | V _{DD} - 1.0 | | | V | |
| Input leakage current | I _{LI} | 0 V ≤ V _I ≤ V _{DD} | | | ±10 | μA | |
| Output leakage current | I _{LO} | 0 V ≤ V _O ≤ V _{DD} | | | ±10 | μA | |
| V _{DD} supply current | I _{DD1} | Operating mode | | 40 | 75 | mA | |
| | I _{DD2} | HALT mode | | 20 | 45 | 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 |

- Notes**
1. Except the pin described in **Note 2**.
 2. $\overline{\text{RESET}}$, X1, X2, P20/NMI, P21/INTP0, P22/INTP1, P23/INTP2, P24/INTP3, P25/INTP4, P26/INTP5, P27/INTP6/TI, P32/SO/SB0, P33/SI/SB1, P34/SCK pins.

AC CHARACTERISTICS (T_A = -10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V)

★

Non-consecutive read/write operation (with general-purpose memory connected)

| Parameter | Symbol | Test Conditions | MIN. | MAX. | Unit |
|--|--------------------|-----------------|------|------|------|
| System clock cycle time | t _{CYK} | | 125 | 250 | ns |
| Address setup time (vs. ASTB↓) | t _{SAST} | | 32 | | ns |
| Address hold time (vs. ASTB↓) | t _{HSTA} | | 32 | | ns |
| \overline{RD} ↓ delay time from address | t _{DAR} | | 85 | | ns |
| Address float time from \overline{RD} ↓ | t _{FRA} | | | 10 | ns |
| Data input time from address | t _{DAID} | | | 222 | ns |
| Data input time from \overline{RD} ↓ | t _{DRID} | | | 112 | ns |
| \overline{RD} ↓ delay time from ASTB↓ | t _{DSTR} | | 42 | | ns |
| Data hold time (vs. \overline{RD} ↑) | t _{HRID} | | 0 | | ns |
| Address active time from \overline{RD} ↑ | t _{DRA} | | 50 | | ns |
| \overline{RD} low-level width | t _{WRL} | | 147 | | ns |
| ASTB high-level width | t _{WSTH} | | 37 | | ns |
| \overline{WR} ↓ delay time from address | t _{DAW} | | 85 | | ns |
| Data output time from ASTB↓ | t _{DSTOD} | | | 102 | ns |
| Data output time from \overline{WR} ↓ | t _{DWOD} | | | 40 | ns |
| \overline{WR} ↓ delay time from ASTB↓ | t _{DSTW} | | 42 | | ns |
| Data setup time (vs. \overline{WR} ↑) | t _{SODW} | | 147 | | ns |
| Data hold time (vs. \overline{WR} ↑) | t _{HWOD} | | 32 | | ns |
| ASTB↑ delay time from \overline{WR} ↑ | t _{DWST} | | 42 | | ns |
| \overline{WR} low-level width | t _{WWL} | | 147 | | ns |

t_{CYK} Dependent Bus Timing Definition

| Parameter | Expression | MIN./MAX. | Unit |
|--------------------|------------------|-----------|------|
| t _{SAST} | 0.5T – 30 | MIN. | ns |
| t _{HSTA} | 0.5T – 30 | MIN. | ns |
| t _{DAR} | T – 40 | MIN. | ns |
| t _{DAID} | (2.5 + n) T – 90 | MAX. | ns |
| t _{DRID} | (1.5 + n) T – 75 | MAX. | ns |
| t _{DSTR} | 0.5T – 20 | MIN. | ns |
| t _{DRA} | 0.5T – 12 | MIN. | ns |
| t _{WRL} | (1.5 + n) T – 40 | MIN. | ns |
| t _{WSTH} | 0.5T – 25 | MIN. | ns |
| t _{DAW} | T – 40 | MIN. | ns |
| t _{DSTOD} | 0.5T + 40 | MAX. | ns |
| t _{DSTW} | 0.5T – 20 | MIN. | ns |
| t _{SODW} | 1.5T – 40 | MIN. | ns |
| t _{HWOD} | 0.5T – 30 | MIN. | ns |
| t _{DWST} | 0.5T – 20 | MIN. | ns |
| t _{WWL} | (1.5 + n) T – 40 | MIN. | ns |

- Remarks**
1. T = t_{CYK} = 1/f_{CLK} (f_{CLK} is internal system clock frequency)
 2. n indicates the number of wait cycles defined by user software.
 3. Depends on t_{CYK} for the bus timing shown in this table only.

SERIAL OPERATION (T_A = -10 to +70 °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 division by 8 | 1 | | μs |
| | | $\overline{\text{SCK}}$ input | External clock | 1 | | μs |
| Serial clock low-level width | t _{WSKL} | $\overline{\text{SCK}}$ output | Internal division by 8 | 420 | | ns |
| | | $\overline{\text{SCK}}$ input | External clock | 420 | | ns |
| Serial clock high-level width | t _{WSKH} | $\overline{\text{SCK}}$ output | Internal division by 8 | 420 | | ns |
| | | $\overline{\text{SCK}}$ input | External clock | 420 | | ns |
| SI setup time (to $\overline{\text{SCK}}\uparrow$) | t _{SRXSK} | | | 80 | | ns |
| SI hold time (from $\overline{\text{SCK}}\uparrow$) | t _{HSKRX} | | | 80 | | ns |
| SO delay time from $\overline{\text{SCK}}\downarrow$ | t _{DSKTX} | R = 1 kΩ, C = 100 pF | | | 210 | ns |

OTHER OPERATION (T_A = -10 to +70 °C, V_{DD} = +5 V ±10 %, V_{SS} = 0 V)

| Parameter | Symbol | Test Conditions | MIN. | MAX. | Unit |
|--|---------------------------------------|---------------------------|------|------|------------------|
| NMI high/low-level width | t _{WNIH} , t _{WNIL} | | 5 | | μs |
| INTP0 high/low-level width | t _{WI0H} , t _{WI0L} | | 8T | | t _{CYK} |
| INTP1 high/low-level width | t _{WI1H} , t _{WI1L} | | 8T | | t _{CYK} |
| INTP2 high/low-level width | t _{WI2H} , t _{WI2L} | | 8T | | t _{CYK} |
| NTP3 high/low-level width | t _{WI3H} , t _{WI3L} | | 8T | | t _{CYK} |
| NTP4 high/low-level width | t _{WI4H} , t _{WI4L} | | 8T | | t _{CYK} |
| INTP5 high/low-level width | t _{WI5H} , t _{WI5L} | | 8T | | t _{CYK} |
| INTP6 high/low-level width | t _{WI6H} , t _{WI6L} | | 8T | | t _{CYK} |
| $\overline{\text{RESET}}$ high/low-level width | t _{WRSH} , t _{WRSL} | | 5 | | μs |
| TI high/low-level width | t _{WTIH} , t _{WTIL} | In TM1 event counter mode | 8T | | t _{CYK} |

A/D CONVERTER CHARACTERISTICS($T_A = -10$ to $+70$ °C, $V_{DD} = +5$ V ± 10 %, $V_{SS} = AV_{SS} = 0$ V,
 $V_{DD} - 0.5$ V $\leq AV_{DD} \leq V_{DD}$)

| Parameter | Symbol | Test Conditions | | MIN. | TYP. | MAX. | Unit |
|--------------------------------------|------------|--------------------------------------|-----------------------------|------|-----------|-------------|-----------|
| Resolution | | | | 10 | | | bit |
| Total error Note 1 | | 4.5 V $\leq AV_{REF} \leq AV_{DD}$ | | | | ± 0.4 % | %FSR |
| | | 3.4 V $\leq AV_{REF} \leq AV_{DD}$ | | | | ± 0.7 | %FSR |
| Quantization error | | | | | | $\pm 1/2$ | LSB |
| Conversion time | t_{CONV} | | | 144 | | | t_{CYK} |
| Sampling time | t_{SAMP} | | | 24 | | | t_{CYK} |
| Zero scale error Note 1 | | 4.5 V $\leq AV_{REF} \leq AV_{DD}$ | | | ± 1.5 | ± 2.5 | LSB |
| | | 3.4 V $\leq AV_{REF} \leq AV_{DD}$ | | | ± 1.5 | ± 4.5 | LSB |
| Full scale error Note 1 | | 4.5 V $\leq AV_{REF} \leq AV_{DD}$ | | | ± 1.5 | ± 2.5 | LSB |
| | | 3.4 V $\leq AV_{REF} \leq AV_{DD}$ | | | ± 1.5 | ± 4.5 | LSB |
| Non-linear error Note 1 | | 4.5 V $\leq AV_{REF} \leq AV_{DD}$ | | | ± 1.5 | ± 2.5 | LSB |
| | | 3.4 V $\leq AV_{REF} \leq AV_{DD}$ | | | ± 1.5 | ± 4.5 | LSB |
| ★ Analog input voltage Note 2 | V_{IAN} | | | -0.3 | | AV_{DD} | V |
| Reference voltage | AV_{REF} | | | 3.4 | | AV_{DD} | V |
| AV_{REF} current | AI_{REF} | | | | 1.0 | 3.0 | mA |
| AV_{DD} supply current | AI_{DD} | | | | 2.0 | 6.0 | mA |
| A/D converter data retention current | AI_{DDR} | STOP mode | $AV_{DDR} = 2.5$ V | | 2.0 | 10 | μA |
| | | | $AV_{DDR} = 5$ V ± 10 % | | 10 | 50 | μA |

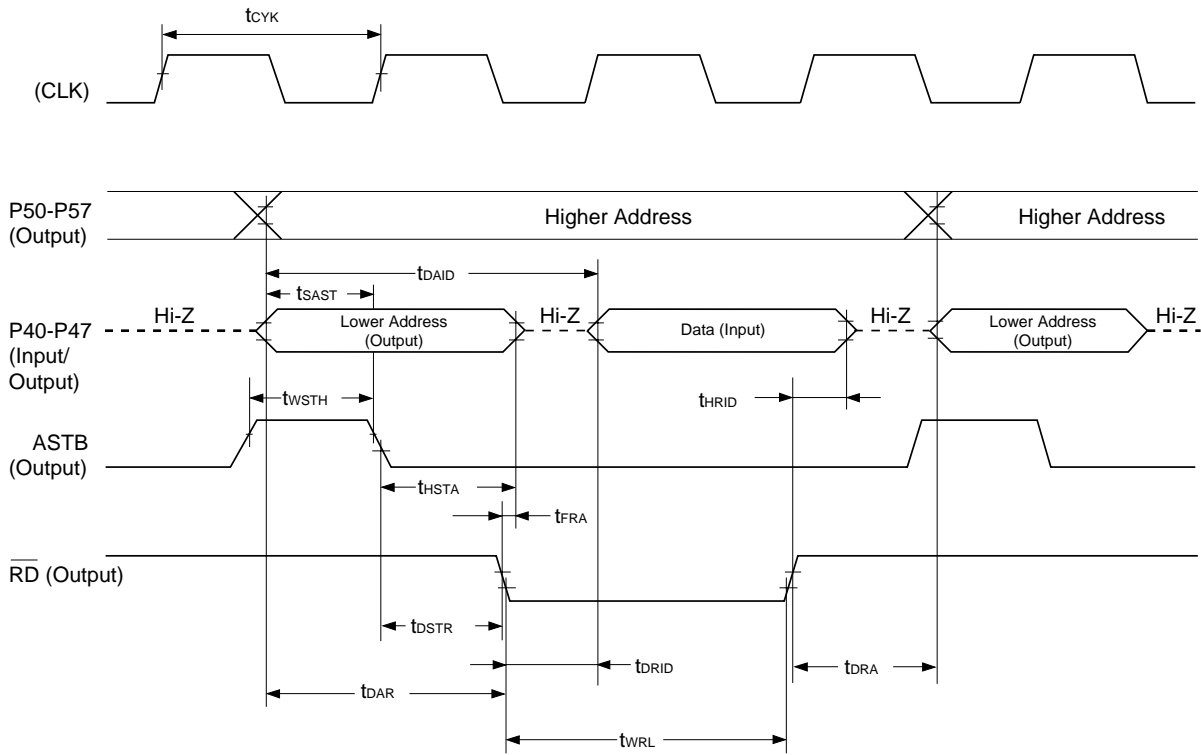
Notes 1. Quantization error excluded.

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

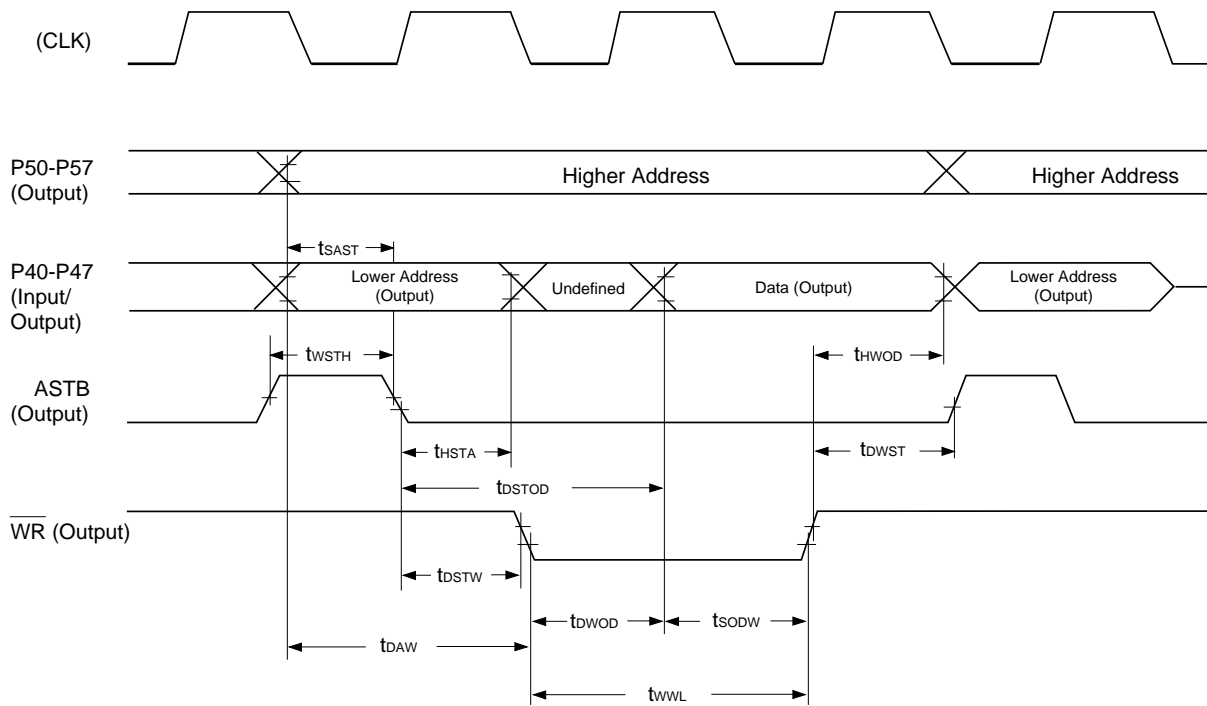
When 0 V $< V_{IAN} < AV_{REF}$, the conversion is performed at a resolution of 10 bits.

When $AV_{REF} \leq V_{IAN} \leq AV_{DD}$, the conversion result is 3FFH.

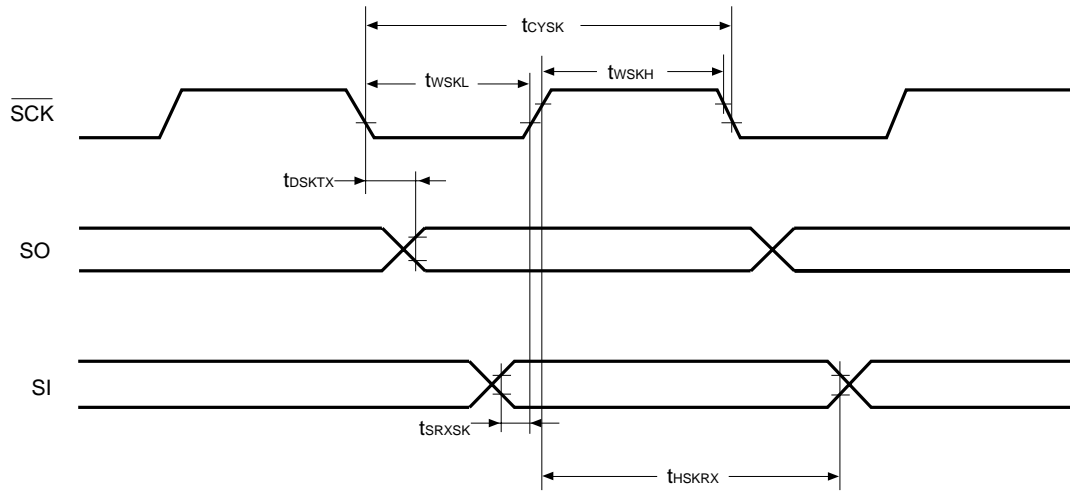
Non-Consecutive Read Operation



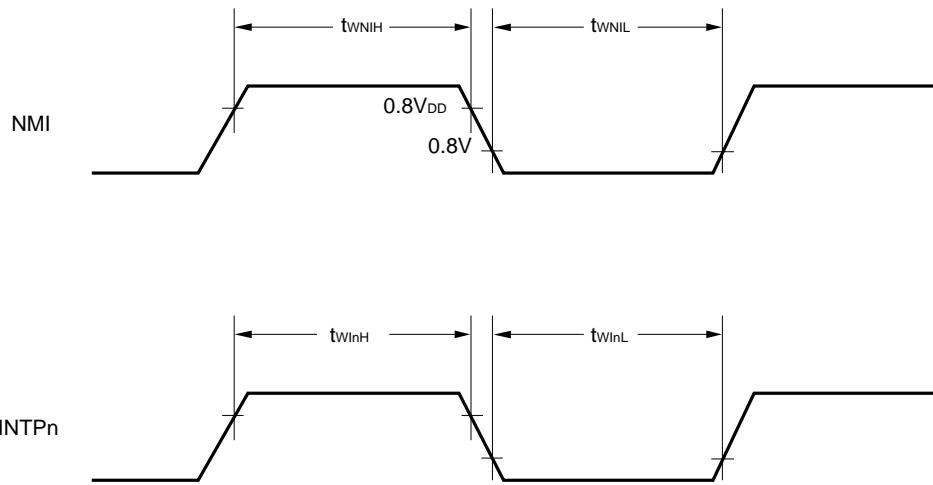
Non-Consecutive Write Operation



Serial Operation

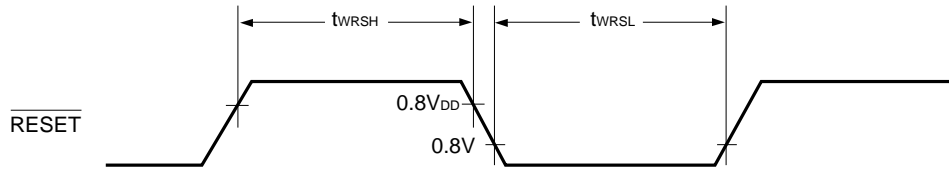


Interrupt Input Timing

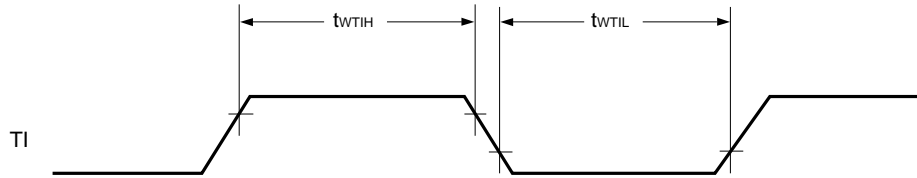


Remarks n = 0 to 6

Reset Input Timing

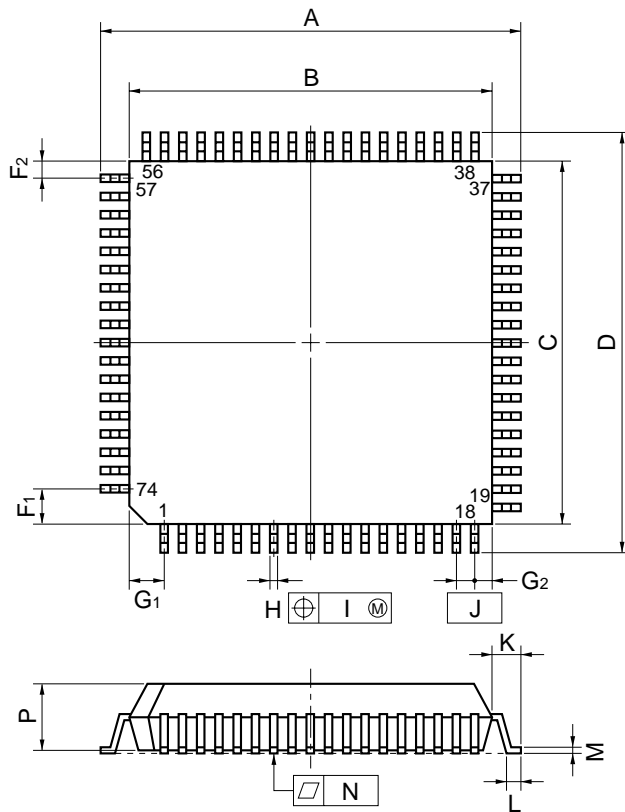


TI Pin Input Timing

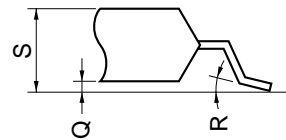


10. PACKAGE DRAWINGS

74 PIN PLASTIC QFP (□20)



detail of lead end



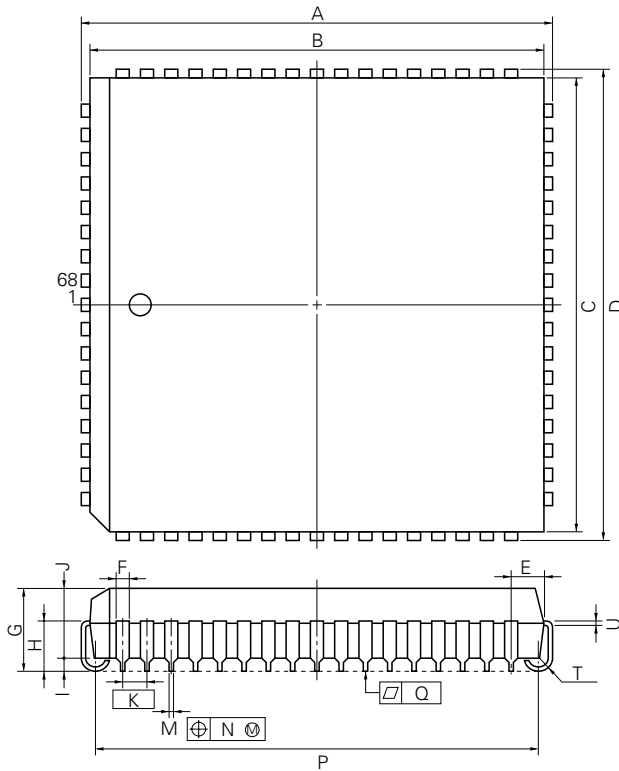
NOTE

Each lead centerline is located within 0.20 mm (0.008 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|------|--|---|
| A | 23.2±0.4 | 0.913 ^{+0.017} _{-0.016} |
| B | 20.0±0.2 | 0.787 ^{+0.009} _{-0.008} |
| C | 20.0±0.2 | 0.787 ^{+0.009} _{-0.008} |
| D | 23.2±0.4 | 0.913 ^{+0.017} _{-0.016} |
| F1 | 2.0 | 0.079 |
| F2 | 1.0 | 0.039 |
| G1 | 2.0 | 0.079 |
| G2 | 1.0 | 0.039 |
| H | 0.40±0.10 | 0.016 ^{+0.004} _{-0.005} |
| I | 0.20 | 0.008 |
| J | 1.0 (T.P.) | 0.039 (T.P.) |
| K | 1.6±0.2 | 0.063±0.008 |
| 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 | 0.10 | 0.004 |
| P | 3.7 | 0.146 |
| Q | 0.1±0.1 | 0.004±0.004 |
| R | 5°±5° | 5°±5° |
| S | 4.0 MAX. | 0.158 MAX. |

S74GJ-100-5BJ-3

68 PIN PLASTIC QFJ (□950 mil)



P68L-50A1-2

NOTE

Each lead centerline is located within 0.12 mm (0.005 inch) of its true position (T.P.) at maximum material condition.

| ITEM | MILLIMETERS | INCHES |
|------|--|---|
| A | 25.2±0.2 | 0.992±0.008 |
| B | 24.20 | 0.953 |
| C | 24.20 | 0.953 |
| D | 25.2±0.2 | 0.992±0.008 |
| E | 1.94±0.15 | 0.076 ^{+0.007} _{-0.006} |
| F | 0.6 | 0.024 |
| G | 4.4±0.2 | 0.173 ^{+0.009} _{-0.008} |
| H | 2.8±0.2 | 0.110 ^{+0.009} _{-0.008} |
| I | 0.9 MIN. | 0.035 MIN. |
| J | 3.4 | 0.134 |
| K | 1.27 (T.P.) | 0.050 (T.P.) |
| M | 0.40±1.0 | 0.016 ^{+0.004} _{-0.005} |
| N | 0.12 | 0.005 |
| P | 23.12±0.20 | 0.910 ^{+0.009} _{-0.008} |
| Q | 0.15 | 0.006 |
| T | R 0.8 | R 0.031 |
| U | 0.20 ^{+0.10} _{-0.05} | 0.008 ^{+0.004} _{-0.002} |

11. RECOMMENDED SOLDERING CONDITIONS

The μPD78323 and 78324 should be soldered and mounted under the conditions recommended in the table below.

For detail of recommended soldering conditions, refer to the information document **Semiconductor Device Mounting Technology Manual (IE1-1207)**.

For soldering methods and conditions other than those recommended below, contact our salesman.

Table 11-1. Soldering Conditions for Surface Mount Type

μPD78323GJ-5BJ : 74-pin plastic QFP (20 × 20 mm)

μPD78324GJ-xxx-5BJ : 74-pin plastic QFP (20 × 20 mm)

| Soldering Method | Soldering Conditions | Recommended Condition Symbol |
|------------------|---|------------------------------|
| Infrared reflow | Package peak temperature: 230 °C, Time: 30 sec. max. (at 210 °C or above) Number of times: Once, Time limit: 7 days ^{Note} (thereafter 10 hours prebaking required at 125 °C) | IR30-107-1 |
| VPS | Package peak temperature: 215 °C, Time: 40 sec. max. (at 200 °C or above) Number of times: Once, Time limit: 7 days ^{Note} (thereafter 10 hours prebaking required at 125 °C) | VP15-107-1 |
| Pin part heating | Pin temperature: 300 °C max, Time: 3 sec. max. (Per side of the device) | ———— |

★ μPD78323LP : 68-pin plastic QFJ (□950 mil)

★ μPD78324LP-xxx : 68-pin plastic QFJ (□950 mil)

| Soldering Method | Soldering Conditions | Recommended Condition Symbol |
|------------------|--|------------------------------|
| Infrared reflow | Package peak temperature: 235 °C, Time: 30 sec. max. (at 210 °C or above) Number of times: twice or less, Time limit: 7 days ^{Note} (thereafter 36 hours prebaking required at 125 °C) <Caution> (1) The second reflow should be started after the temperature of the device which would have been changed by the first reflow has returned to normal. (2) Please avoid flux water washing after the first reflow. | IR30-367-2 |
| VPS | Package peak temperature: 215 °C, Time: 40 sec. max. (at 200 °C or above), Number of times: twice or less, Time limit: 7 days ^{Note} (thereafter 36 hours prebaking required at 125 °C) <Caution> (1) The second reflow should be started after the temperature of the device which would have been changed by the first reflow has returned to normal. (2) Please avoid flux water washing after the first reflow. | VP15-367-2 |
| Pin part heating | Pin temperature: 300 °C max., Time: 3 sec. max. (Per side of the device)) | ———— |

Note For the storage period after dry-pack decompression, storage conditions are max. 25 °C, 65 % RH.

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

APPENDIX A. LIST OF 78K/III SERIES PRODUCTS (1/2)

| | | μPD78324 | μPD78323 | μPD78322 | μPD78320 | μPD78312A | μPD78310A |
|------------------------------------|--------|--|----------|----------------|----------|---|-----------|
| Basic instruction | | 111 | | | | 96 | |
| Minimum instruction execution time | | 250 ns (at 16 MHz operation) | | | | 500 ns (at 12 MHz operation) | |
| Internal memory | ROM | 32768 × 8 bits | — | 16384 × 8 bits | — | 8192 × 8 bits | — |
| | RAM | 1024 × 8 bits | | 640 × 8 bits | | 256 × 8 bits | |
| Memory space | | 64K bytes | | | | | |
| I/O lines | Input | 16 (including 8 analog inputs) | | | | 12 (including 4 analog inputs) | |
| | Output | — | | | | 1 | |
| | I/O | 39 | 21 | 39 | 21 | 40 | 24 |
| Pulse unit | | <ul style="list-style-type: none"> Real-time pulse unit • 18/16-bit free running timer × 1 • 16-bit timer/event counter × 1 • 16-bit compare register × 6 • 18-bit capture register × 4 • 18-bit capture/compare register × 2 • Real-time output port × 8 | | | | <ul style="list-style-type: none"> Multi-function pulse I/O unit • 16-bit presetable up-/down-counter × 2 • 16-bit free running counter capture function × 2 • 16-bit interval timer × 2 • High-precision PWM output × 2 • Real-time output port : 4 bits × 2 | |
| | | mode) function available | | | | Count unit mode 4 (4-multiplication) | |
| | | | | | | Counter start function by interval timer external trigger available | |
| Serial communication interface | | <ul style="list-style-type: none"> • Dedicated on-chip baud rate generator • UART ...1 channel • SBI } ...1 channel • 3-wire serial I/O } | | | | <ul style="list-style-type: none"> • 8 bits (full-duplex transmission/reception) • Dedicated on-chip baud rate generator • 2 transfer modes (asynchronous mode, I/O interface mode) | |
| A/D converter | | Eight 10-bit resolution inputs | | | | Four 8-bit resolution inputs | |
| Interrupt | | <ul style="list-style-type: none"> • 8 external, 14 internal (shared with external 2) • 3-level programmable priority order | | | | <ul style="list-style-type: none"> • 4 external, 13 internal • 8-level programmable priority order | |
| | | • 3 processing methods (vectored interrupt, context switching and macro service functions) | | | | | |

LIST OF 78K/III SERIES PRODUCTS (2/2)

| | μ PD78324 | μ PD78323 | μ PD78322 | μ PD78320 | μ PD78312A | μ PD78310A |
|-----------------|--|---------------|---|---------------|---|----------------|
| Test source | Internal : 1 | | | | — | |
| Instruction set | Instructions for μ PD78312 and 78310 significantly increased. | | | | Following instructions added for μ PD78312 and 78310 <ul style="list-style-type: none"> • MOVW rp1, !addr16 instruction • MOVW !addr16, rp1 instruction | |
| Others | <ul style="list-style-type: none"> • On-chip watchdog timer • Standby function (STOP/HALT) | | | | | |
| | — | | | | <ul style="list-style-type: none"> • 20-bit time base counter • Pseudo static RAM refresh function | |
| Package | <ul style="list-style-type: none"> • 68-pin plastic QFJ (\square 950 mil) • 74-pin plastic QFP (20 \times 20 mm) | | <ul style="list-style-type: none"> • 68-pin plastic QFJ (\square 950 mil) • 74-pin plastic QFP (20 \times 20 mm) • 80-pin plastic QFP (14 \times 20 mm) | | <ul style="list-style-type: none"> • 64-pin plastic shurink DIP (750 mil) • 64-pin plastic QFP (14 \times 20 mm) • 64-pin plastic QUIP • 68-pin plastic QFJ (\square 950 mil) | |

APPENDIX B. TOOLS

B.1 DEVELOPMENT TOOLS

The following development tools are available for system development using the μPD78324.

Language Processor

| | | | | |
|--|--|--------------|-------------------------|--------------|
| 78K/III series relocatable assembler (RA78K/III) | Refers to the relocatable assembler which can be used commonly for the 78K/III series. Equipped with the macro function, the relocatable assembler is aimed at improved development efficiency. The assembler is also accompanied by the structured assembler which can describe the program control structure explicitly, thus making it possible to improve the productivity and the maintainability of the program. | | | |
| | Host machine | OS | Supply medium | |
| | PC-9800 series | MS-DOS™ | 3.5-inch 2HD | μS5A13RA78K3 |
| | | | 5-inch 2HD | μS5A10RA78K3 |
| | IBM PC/AT™ and its compatible machine | PC DOS™ | 3.5-inch 2HC | μS7B13RA78K3 |
| | | | 5-inch 2HC | μS7B10RA78K3 |
| | HP9000 series 700™ | HP-UX™ | DAT | μS3P16RA78K3 |
| | SPARCstation™ | SunOS™ | Cartridge tape (QIC-24) | μS3K15RA78K3 |
| NEWS™ | NEWS-OS™ | μS3R15RA78K3 | | |
| 78K/III series C compiler (CC78K/III) | Refers to the C compiler which can be commonly used in the 78K/III series. This compiler is a program converting the programs written in the C language to those object codes which are executable by microcontrollers. When using this compiler, the 78K/III series relocatable assembler (RA78K/III) is required. | | | |
| | Host machine | OS | Supply medium | |
| | PC-9800 series | MS-DOS | 3.5-inch 2HD | μS5A13CC78K3 |
| | | | 5-inch 2HD | μS5A10CC78K3 |
| | IBM PC/AT and its compatible machine | PC DOS | 3.5-inch 2HC | μS7B13CC78K3 |
| | | | 5-inch 2HC | μS7B10CC78K3 |
| | HP9000 series 700 | HP-UX | DAT | μS3P16CC78K3 |
| | SPARCstation | SunOS | Cartridge tape (QIC-24) | μS3K15CC78K3 |
| NEWS | NEWS-OS | μS3R15CC78K3 | | |

Remark Relocatable assembler and C compiler operations are assured only on the host machine and the OS above.

PROM Writing Tools

| | | | | | |
|------------|--|--|--------|---------------|------------------------------|
| Hardware | PG-1500 | This PROM programmer allows programming, in standalone mode or via operation from a host computer, of a singlechip microcontroller with on-chip PROM by connection of the board provided and a separately available programmer adapter. It can program typical 256K-bit to 4M-bit PROMs. | | | |
| | UNISITE 2900 | PROM programmer made by Data I/O Japan Corporation. | | | |
| | PA-78P324GJ PA-78P324KC PA-78P324KD PA-78P324LP | PROM programmer adapters for writing programs to the μPD78P324 with a general PROM programmer such as the PG-1500. PA-78P324GJ ... For μPD78P324GJ PA-78P324KC ... For μPD78P324KC PA-78P324KD ... For μPD78P324KD PA-78P324LP ... For μPD78P324LP | | | |
| Software | PG-1500 controller | Connects PG-1500 and host machine via a serial and parallel interface, and controls the PG-1500 on the host machine. | | | |
| | | Host Machine | OS | Supply Medium | Ordering Code (Product Name) |
| | | PC-9800 series | MS-DOS | 3.5-inch 2HD | μS5A13PG1500 |
| | | | | 5-inch 2HD | μS5A10PG1500 |
| | | IBM PC/AT and its compatible machine | PC DOS | 3.5-inch 2HC | μS7B13PG1500 |
| 5-inch 2HC | μS7B10PG1500 | | | | |

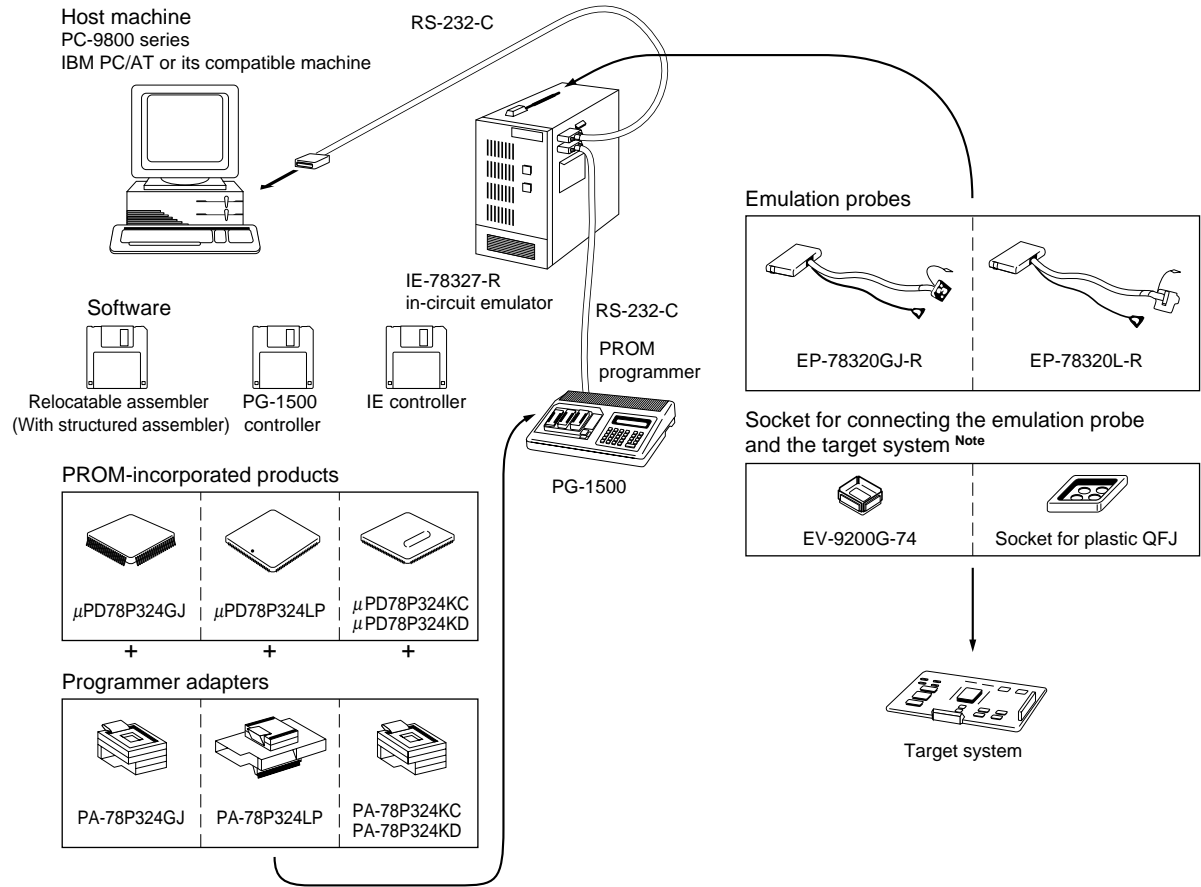
Remark Operation of the PG-1500 controller is guaranteed only on the host machines and operating systems quoted above.

Debugging Tools

| | | | | | |
|------------|--|--|--------|---------------|---------------|
| Hardware | IE-78327-R IE-78320-R ^{Note} | These are the in-circuit emulators which can be used for the development and debugging of application systems. Debugging is performed by connecting them to a host machine. The IE-78327-R can be used commonly for both the μPD78322 subseries and the μPD78328 subseries. The IE-78320-R can be used for the μPD78322 subseries. | | | |
| | EP-78320GJ-R EP-78320L-R | These are the emulation probes for connecting the IE-78327-R or IE-78320-R to a target system. EP-78320GJ-R: for 74-pin plastic QFP EP-78320L-R: for 68-pin plastic QFJ | | | |
| | IE-78327-R control program (IE controller) | This program is for controlling the IE-78327-R from a host machine. It can execute commands automatically, thus enabling more efficient debugging. | | | |
| | | Host machine | OS | Supply medium | Part number |
| | | PC-9800 series | MS-DOS | 3.5-inch 2HD | μS5A13IE78327 |
| | | | | 5-inch 2HD | μS5A10IE78327 |
| | | IBM PC/AT and its compatible machine | PC DOS | 3.5-inch 2HC | μS7B13IE78327 |
| 5-inch 2HC | μS7B10IE78327 | | | | |
| Software | IE-78320-R control program ^{Note} (IE controller) | This program is for controlling the IE-78320-R from a host machine. It can execute commands automatically, thus enabling more efficient debugging. | | | |
| | | Host machine | OS | Supply medium | Part number |
| | | PC-9800 series | MS-DOS | 3.5-inch 2HD | μS5A13IE78320 |
| | | | | 5-inch 2HD | μS5A10IE78320 |
| | | IBM PC/AT and its compatible machine | PC DOS | 5-inch 2HC | μS7B10IE78320 |

- Remarks**
- The operation of each software is assured only on the host machine and the OS above.
 - μPD78322 subseries: μPD78320, 78322, 78P322, 78323, 78324, 78P324, 78320(A), 78320(A1), 78320(A2), 78322(A), 78322(A1), 78322(A2), 78323(A), 78323(A1), 78323(A2), 78324(A), 78324(A1), 78324(A2), 78P324(A), 78P324(A1), 78P324(A2)
μPD78328 subseries: μPD78327, 78328, 78P328, 78327(A), 78328(A)

Note The existing product IE-78320-R is a maintenance product. If you are going to newly purchase an in-circuit emulator, please use the alternative product IE-78327-R.



Note The socket is supplied with the emulation probe.

- Remarks**
1. It is also possible to use the host machine and the PG-1500 by connecting them directly by the RS-232-C.
 2. In the diagram above, representative software supply media and 3.5-inch FDs.

B.2 EVALUATION TOOLS

To evaluate the functions of the μPD78324, the following tools are made available.

| Part Number | Host Machine | Function |
|-------------|-------------------------------------|--|
| EB-78320-98 | PC-9800 series | By connecting to a host machine, it is possible to evaluate the functions equipped by the μPD78324 in a simple manner. The command system of this product basically conforms to that of IE-78327-R and IE-78320-R. Therefore, it is easy to move to the development work of application systems by IE-78327-R or IE-78320-R. In addition a turbo access manager (μPD71P301) ^{Note} can be mounted on the board. |
| EB-78320-PC | IBM PC/AT or its compatible machine | |

Note The turbo access manager (μPD71P301) is a maintenance product.

- Cautions**
1. This product is not a development tool of μPD78324 application systems.
 2. This product is not equipped with the emulation function for executing the ROM incorporated in the μPD78324.

B.3 EMBEDDED SOFTWARE

The following embedded software programs are available to perform program development and maintenance more efficiently.

Real-time OS

| | | | | |
|--------------------------|---|--------|---------------|---------------|
| Real-time OS (RX78K/III) | The RX78K/III is designed to provide a multi-task environment in the field of control application where real-time operation is required. By using this real-time OS, the performance of the whole system can be improved by allocating CPU's idle time to other processings. The RX78K/III provides the system call based on the μITRON specifications. The RX78K/III package provides tools (configurators) for creating RX78K/III's nucleus and multiple information table. | | | |
| | Host machine | OS | Supply medium | Part number |
| | PC-9800 series | MS-DOS | 3.5-inch 2HD | μS5A13RX78320 |
| | | | 5-inch 2HD | μS5A10RX78320 |
| | IBM PC/AT and its compatible machine | PC DOS | 3.5-inch 2HC | μS7B13RX78320 |
| 5-inch 2HC | | | μS7B10RX78320 | |

Caution To purchase the operating system above, you need to fill in a purchase application form beforehand and sign a contract allowing you to use the software.

Remark When using the real-time OS RX78K/III, you need the assembler package RA78K/III (optional) as well.

Fuzzy Inference Development Support System

| | | | | | |
|--|--|--------|--------------|---------------|--------------|
| Fuzzy knowledge data creation tools (FE9000, FE9200) | This program supports inputting/editing/evaluating (through simulation) of the fuzzy knowledge data (fuzzy rules and membership functions). | | | | |
| | Host machine | | OS | Supply medium | Part number |
| | PC-9800 series | MS-DOS | 3.5-inch 2HD | μS5A13FE9000 | |
| | | | 5-inch 2HD | μS5A10FE9000 | |
| | IBM PC/AT and its compatible machine | PC DOS | Windows™ | 3.5-inch 2HC | μS7B13FE9200 |
| 5-inch 2HC | | | | μS7B10FE9200 | |
| Translator (FT78K3) ^{Note} | This program converts the fuzzy knowledge data obtained with fuzzy knowledge data creation tools to an assembler source program for RA78K/III. | | | | |
| | Host machine | | OS | Supply medium | Part number |
| | PC-9800 series | MS-DOS | 3.5-inch 2HD | μS5A13FT78K3 | |
| | | | 5-inch 2HD | μS5A10FT78K3 | |
| | IBM PC/AT and its compatible machine | PC DOS | 3.5-inch 2HC | μS7B13FT78K3 | |
| 5-inch 2HC | | | μS7B10FT78K3 | | |
| Fuzzy inference module (FI78K/III) ^{Note} | This program executes fuzzy inference. Fuzzy inference is executed by being linked to the fuzzy knowledge data converted by the translator. | | | | |
| | Host machine | | OS | Supply medium | Part number |
| | PC-9800 series | MS-DOS | 3.5-inch 2HD | μS5A13FI78K3 | |
| | | | 5-inch 2HD | μS5A10FI78K3 | |
| | IBM PC/AT and its compatible machine | PC DOS | 3.5-inch 2HC | μS7B13FI78K3 | |
| 5-inch 2HC | | | μS7B10FI78K3 | | |
| Fuzzy inference debugger (FD78K/III) | This is a support software program for evaluating and adjusting the fuzzy knowledge data at a hardware level by using the in-circuit emulator. | | | | |
| | Host machine | | OS | Supply medium | Part number |
| | PC-9800 series | MS-DOS | 3.5-inch 2HD | μS5A13FD78K3 | |
| | | | 5-inch 2HD | μS5A10FD78K3 | |
| | IBM PC/AT and its compatible machine | PC DOS | 3.5-inch 2HC | μS7B13FD78K3 | |
| 5-inch 2HC | | | μS7B10FD78K3 | | |

Note Under development

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

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

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

License not needed : μPD78323

The customer must judge the need for license : μPD78324

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, customer 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 in “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 NEC Sales Representative in advance.

Anti-radioactive design is not implemented in this product.

M4 94.11

MS-DOS and Windows are trademarks of Microsoft Corporation.

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.