NEC

MOS INTEGRATED CIRCUIT

μPD78361A, 78362A

16/8-BIT SINGLE-CHIP MICROCONTROLLER

DESCRIPTION

μPD78362A is provided with a high-speed, high-performance CPU and powerful operation functions. Unlike the existing μPD78328, μPD78362A is also provided with a high-resolution PWM signal output function which substantially contributes to improving the performance of the inverter control.

A PROM model, μPD78P364A, is also available.

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

μPD78362A User's Manual Hardware : U10745E
μPD78356 User's Manual Instruction : U12117E

FEATURES

- Internal 16-bit architecture, external 8-bit data bus
- High-speed processing by pipeline control method and high- speed operating clock
  - Minimum instruction execution time: 125 ns (internal clock: at 16 MHz, external clock: 8MHz)
- Real-time pulse unit for inverter control
- 10-bit resolution A/D converter: 8 channels
- 8-/9-/10-/12-bit resolution variable PWM signal output function: 2 channels
- Powerful serial interface: 2 channels
- Internal memory : ROM 32K bytes (μPD78361A)
  24K bytes (μPD78362A)
  RAM 2K bytes (μPD78361A)
  768 bytes (μPD78362A)

APPLICATION EXAMPLES

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

ORDERING INFORMATION

	Part Number	Package	Internal ROM
★	μPD78361ACW-×××	64-pin plastic shrink DIP (750 mil)	Mask ROM
	μPD78362ACW-×××	64-pin plastic shrink DIP (750 mil)	Mask ROM

Remark ××× indicates a ROM code suffix.

Unless otherwise specified, the μPD78362A is treated as the representative model throughout this document.

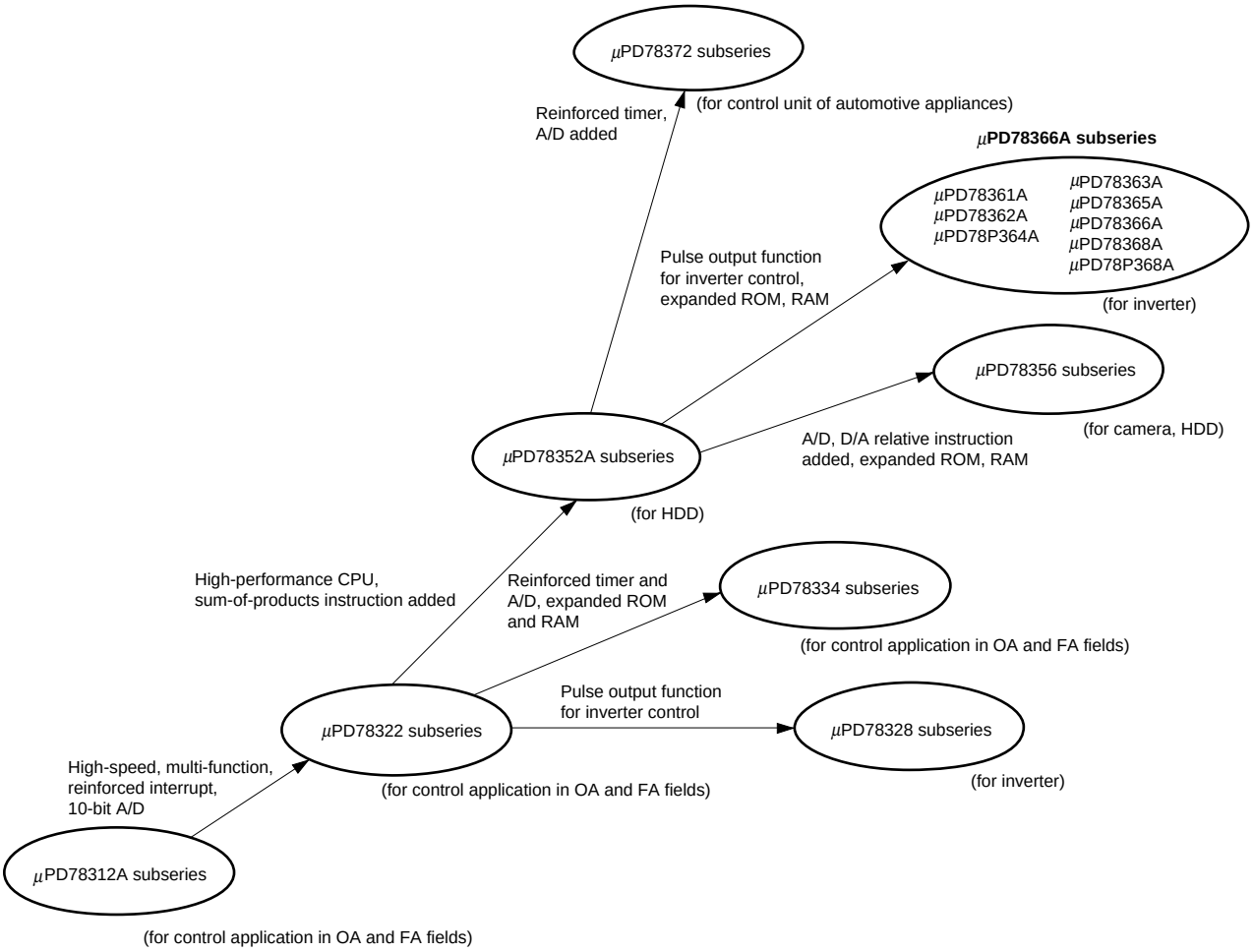
The information in this document is subject to change without notice.

Document No. U10098EJ2V0DS00 (2nd edition)
Date Published August 1997 N
Printed in Japan

The mark ★ shows major revised points.

© NEC Corporation 1996

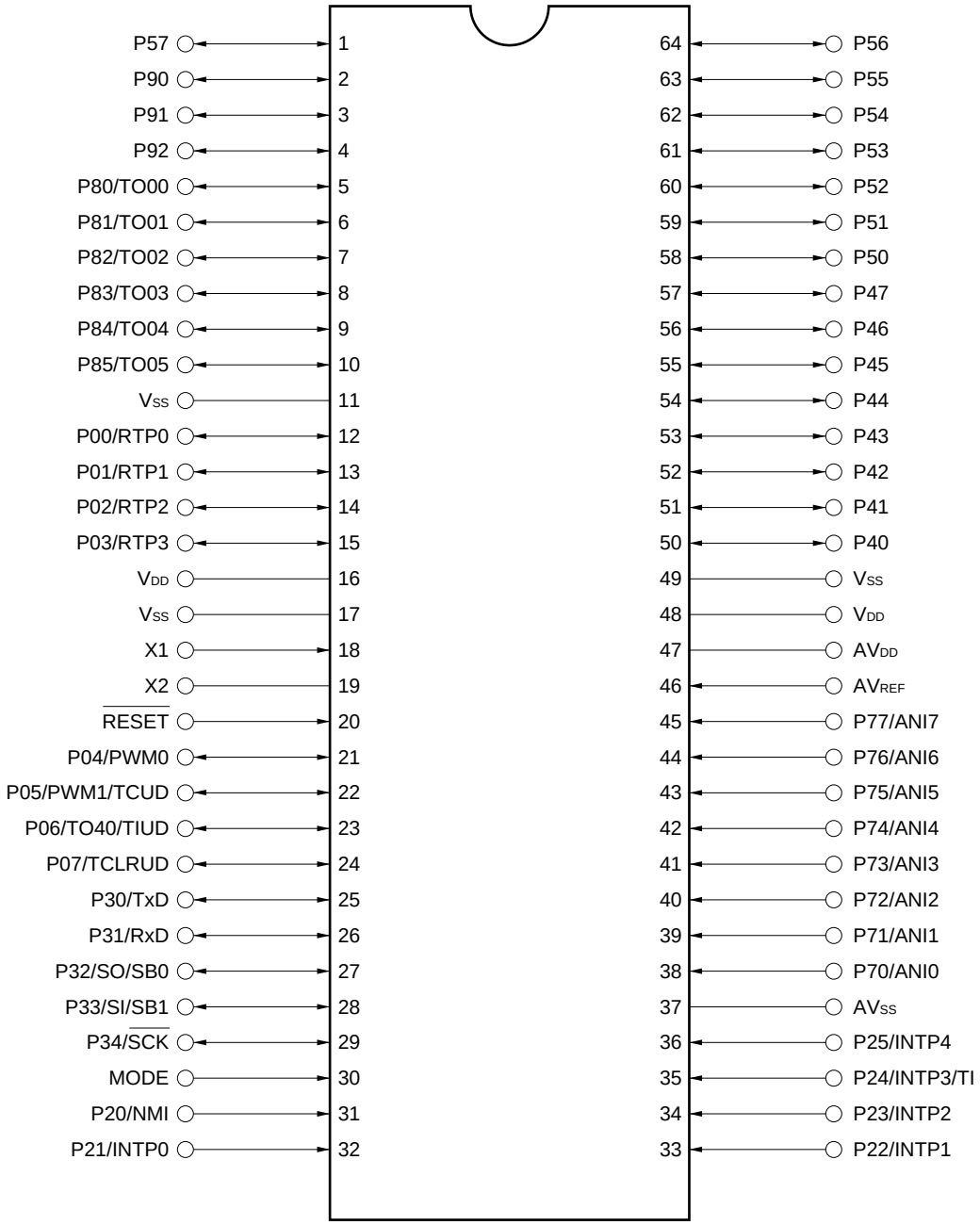
78K/III Series Product Development



PIN CONFIGURATION (TOP VIEW)

- **64-pin plastic shrink DIP (750 mil)**

★ **μPD78361ACW-×××, 78362ACW-×××**



Remark ××× indicates a ROM code suffix.

P00-P07	: Port0	ANI0-ANI7	: Analog Input
P20-P25	: Port2	TxD	: Transmit Data
P30-P34	: Port3	RxD	: Receive Data
P40-P47	: Port4	SI	: Serial Input
P50-P57	: Port5	SO	: Serial Output
P70-P77	: Port7	SB0, SB1	: Serial Bus
P80-P85	: Port8	$\overline{SCK}$	: Serial Clock
P90-P92	: Port9	PWM0, PWM1	: Pulse Width Modulation Output
RTP0-RTP3	: Real-time Port	MODE	: Mode
NMI	: Nonmaskable Interrupt	$\overline{RESET}$	: Reset
INTP0-INTP4	: Interrupt From Peripherals	X1, X2	: Crystal
TO00-TO05, TO40	: Timer Output	AV_{DD}	: Analog V_{DD}
TI	: Timer Input	AV_{SS}	: Analog V_{SS}
TIUD	: Timer Input Up Down Counter	AV_{REF}	: Analog Reference Voltage
TCUD	: Timer Control Up Down Counter	V_{DD}	: Power Supply
TCLRUD	: Timer Clear Up Down Counter	V_{SS}	: Ground

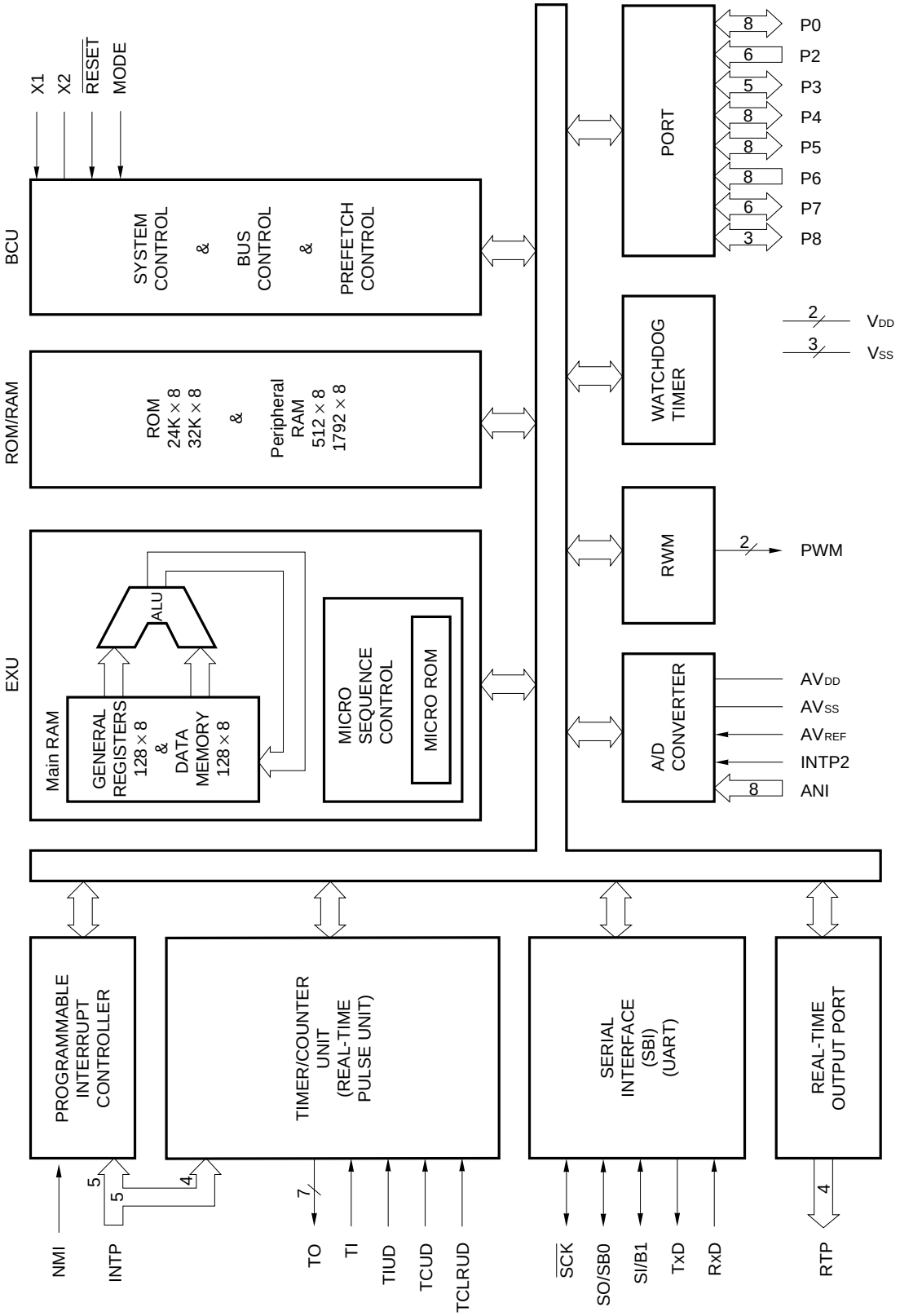
FUNCTIONAL OUTLINE

Item		μPD78361A	μPD78362A
Minimum instruction execution time		125 ns (internal clock: 16 MHz, external clock: 8 MHz)	
Internal memory	ROM	32K bytes	24K bytes
	RAM	2K bytes	768 bytes
Memory space		64K bytes	
General-purpose registers		8 bits × 16 × 8 banks	
Number of basic instructions		115	
Instruction set		• 16-bit transfer/operation • Multiplication/division (16 bits × 16 bits, 32 bits ÷ 16 bits) • Bit manipulation • String • Sum-of-products operation (16 bits × 16 bits + 32 bits) • Relative operation	
I/O lines	Input	14 (of which 8 are shared with analog input)	
	I/O	38	
Real-time pulse unit		• 16-bit timer × 1 10-bit dead time timer × 3 16-bit compare register × 4 2 kinds of output mode can be selected Mode 0, set-reset output: 6 channels Mode 1, buffer output: 6 channels • 16-bit timer × 1 16-bit compare register × 1 • 16-bit timer × 1 16-bit capture register × 1 16-bit capture/compare register × 1 • 16-bit timer × 1 16-bit capture register × 2 16-bit capture/compare register × 1 • 16-bit timer × 1 16-bit compare register × 2 16-bit resolution PWM output: 1 channel	
Real-time output port		Pulse outputs associated with real-time pulse unit: 4 lines	
PWM unit		8-/9-/10-/12-bit resolution variable PWM output: 2 channels	
A/D converter		10-bit resolution, 8 channels	
Serial interface		Dedicated baud rate generator UART: 1 channel Clocked serial interface/SBI: 1 channel	
Interrupt function		• External: 6, internal: 14 (of which 2 are multiplexed with external) • 4 priority levels can be specified through software • 3 types of interrupt service modes selectable (vectored interrupt, macro service, and context switching)	
Package		64-pin plastic shrink DIP (750 mil)	
Others		• Watchdog timer • Standby function (HALT and STOP modes) • PLL control circuit	

DIFFERENCES BETWEEN μPD78362A AND μPD78366A

Item \ Product name		μPD78362A	μPD78366A
Internal ROM	ROM	24K bytes	32K bytes
	RAM	786 bytes	2K bytes
I/O lines	Input	14 (of which 8 are multiplexed with analog input)	
	I/O	38	49
Serial Interface		Dedicated baud rate generator UART: 1 channel Clocked serial interface/SBI: 1 channel	Dedicated baud rate generator UART (with pin selection function): 1 channel Clocked serial interface/SBI: 1 channel
External expansion function		None	Provided
ROM-less mode		None	Provided
MODE setting		Always set as follows: MODE = L	• In ordinary operation mode: MODE0, 1 = LL • In ROM-less mode: MODE0, 1 = HH
Package		64-pin plastic shrink DIP (750 mil)	80-pin plastic QFP (14 × 20 mm)

BLOCK DIAGRAM



Remark Internal ROM and RAM capacities differ depending on the product.

TABLE OF CONTENTS

1. PIN FUNCTIONS 10
- **1.1 PORT PINS 10**
- **1.2 PINS OTHER THAN PORT PINS 11**
- **1.3 PIN I/O CIRCUITS AND PROCESSING OF UNUSED PINS 12**

2. CPU ARCHITECTURE 14
- **2.1 MEMORY SPACE 14**
- **2.2 DATA MEMORY ADDRESSING 16**
- **2.3 PROCESSOR REGISTERS 18**
  - 2.3.1 Control Registers 18
  - 2.3.2 General-Purpose Registers 19
  - 2.3.3 Special Function Registers (SFR) 20

3. FUNCTIONAL BLOCKS 26
- **3.1 EXECUTION UNIT (EXU) 26**
- **3.2 BUS CONTROL UNIT (BCU) 26**
- **3.3 ROM/RAM 26**
- **3.4 PORT FUNCTIONS 26**
- **3.5 CLOCK GENERATOR CIRCUIT 28**
- **3.6 REAL-TIME PULSE UNIT (RPU) 30**
- **3.7 REAL-TIME OUTPUT PORT (RTP) 37**
- **3.8 A/D CONVERTER 38**
- **3.9 SERIAL INTERFACE 39**
- **3.10 PWM UNIT 41**
- **3.11 WATCHDOG TIMER (WDT) 42**

4. INTERRUPT FUNCTIONS 43
- **4.1 OUTLINE 43**
- **4.2 MACRO SERVICE 44**
- **4.3 CONTEXT SWITCHING 47**
  - 4.3.1 Context Switching Function by Interrupt Request 47
  - 4.3.2 Context Switching Function by BRKCS Instruction 48
  - 4.3.3 Restoration from Context Switching 48

5. STANDBY FUNCTIONS 49

6. RESET FUNCTION 50

7. INSTRUCTION SET 51

8. EXAMPLE OF SYSTEM CONFIGURATION 65

9. ELECTRICAL SPECIFICATIONS 66

10. PACKAGE DRAWING 75

11. RECOMMENDED SOLDERING CONDITIONS 76

APPENDIX A. DIFFERENCES BETWEEN μPD78362A AND μPD78328 77

APPENDIX B. TOOLS 78

B.1 DEVELOPMENT TOOLS 78

B.2 EMBEDDED SOFTWARE 83

1. PIN FUNCTIONS

1.1 PORT PINS

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

1.2 PINS OTHER THAN PORT PINS

Pin name	I/O	Function	Shared by:
RTP0-RTP3	Output	Real-time output port that outputs pulses in synchronization with trigger signal from real-time pulse unit.	P00-P03
NMI	Input	Non-maskable interrupt request input.	P20
INTP0		External interrupt request input.	P21
INTP1			P22
INTP2			P23
INTP3			P24/TI
INTP4			P25
TI	Input	External count clock input to timer 1.	P24/INTP3
TCUD		Count operation selection control signal input to up/down counter (timer 4).	P05/PWM1
TIUD		External count clock input to up/down counter (timer 4).	P06/TO40
TCLRUD		Clear signal input to up/down counter (timer 4).	P07
TO00-TO05	Output	Pulse output from real-time pulse unit.	P80-P85
TO40			P06/TIUD
ANI0-ANI7	Input	Analog input to A/D converter.	P70-P77
T_xD	Output	Serial data output of asynchronous serial interface.	P30
R_xD	Input	Serial data input of asynchronous serial interface.	P31
$\overline{SCK}$	I/O	Serial clock input/output of clocked serial interface.	P34
SI	Input	Serial data input of clocked serial interface in 3-line mode.	P33/SB1
SO	Ouput	Serial data output of clocked serial interface in 3-line mode.	P32/SB0
SB0	I/O	Serial data input/output of clocked serial interface in SBI mode.	P32/SO
SB1			P33/SI
PWM0	Output	PWM signal output.	P04
PWM1			P05/TCUD
MODE	Input	Control signal input to set operation mode. Connected to V_{SS}.	–
$\overline{RESET}$	Input	System reset input	–
X1	Input	Crystal oscillator connecting pins for system clock. If a clock is externally supplied, input it to pin X1. Leave pin X2 open.	–
X2	–		
AV_{REF}	Input	A/D converter reference voltage input.	–
AV_{DD}	–	A/D converter analog power supply.	–
AV_{SS}	–	A/D converter GND.	–
V_{DD}	–	Positive power supply	–
V_{SS}	–	GND	–

1.3 PIN I/O CIRCUITS AND PROCESSING OF UNUSED PINS

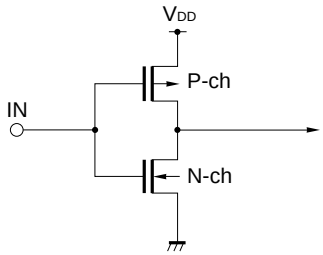
Table 1-1 shows the I/O circuit types of the respective pins, and recommended connections of the unused pins. Figure 1-1 shows the circuits of the respective pins.

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

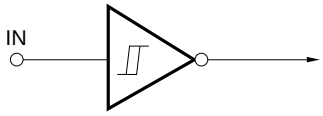
Pin	I/O circuit type	Recommended connections
P00/RTP0-P03/RTP3	5-A	Input : Independently connect to V_{DD} or V_{SS} through resistor Output : Open
P04/PWM0		
P05/TCUD/PWM1		
P06/TIUD/TO40		
P07/TCLRUD		
P20/NMI	2	Connect to V_{SS}
P21/INTP0	2-A	
P22/INTP1		
P23/INTP2		
P24/INTP3/TI		
P25/INTP4		
P30/T_XD	5-A	Input : Independently connect to V_{DD} or V_{SS} through resistor Output : Open
P31/R_XD		
P32/SO/SB0	8-A	
P33/SI/SB1		
P34/$\overline{SCK}$		
P40-P47	5-A	
P50-P57		
P70/ANI0-P77/ANI7	9	Connect to V_{SS}
P80/TO00-P85/TO05	5-A	Input : Independently connect to V_{DD} or V_{SS} through resistor Output : Open
P90-P92		
MODE	1	–
$\overline{RESET}$	2	
AV_{REF}, AV_{SS}	–	Connect to V_{SS}
AV_{DD}		Connect to V_{DD}

Figure 1-1. Pin I/O Circuits

Type 1

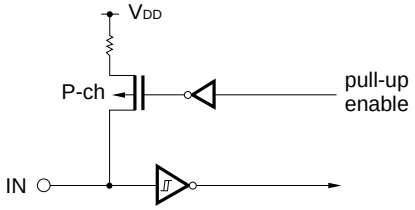


Type 2



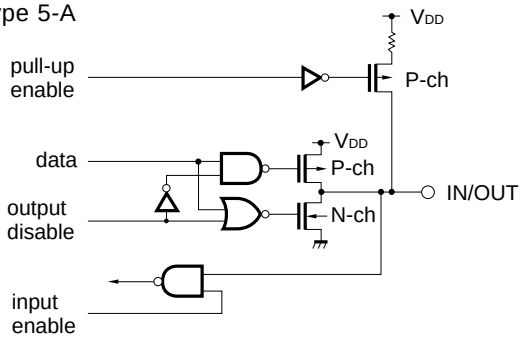
Schmitt trigger input with hysteresis characteristics

Type 2-A

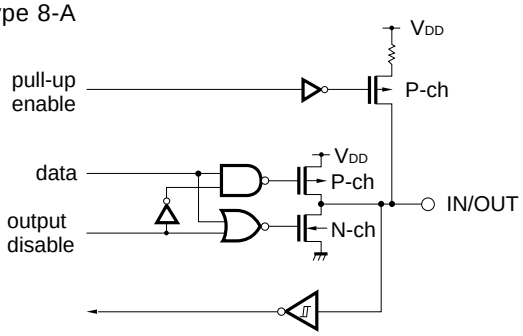


Schmitt trigger input with hysteresis characteristics

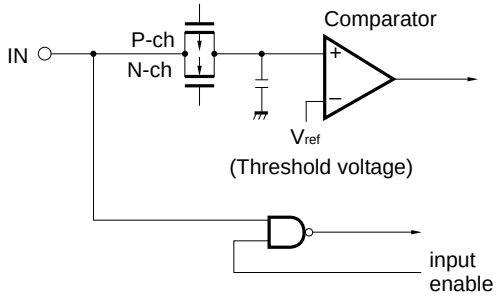
Type 5-A



Type 8-A



Type 9

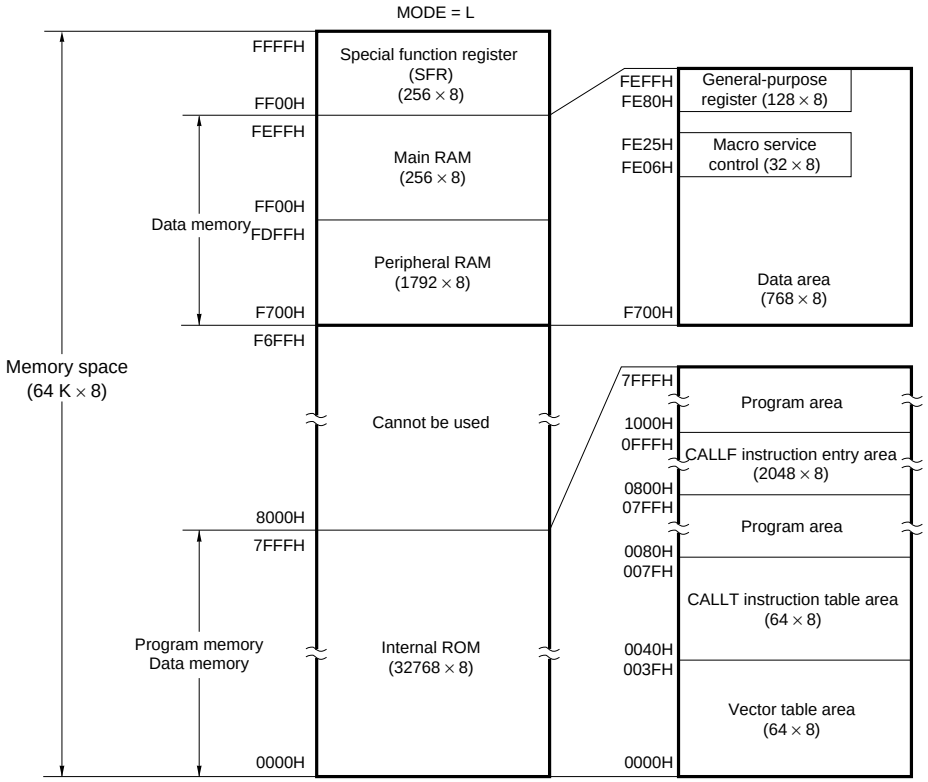


2. CPU ARCHITECTURE

2.1 MEMORY SPACE

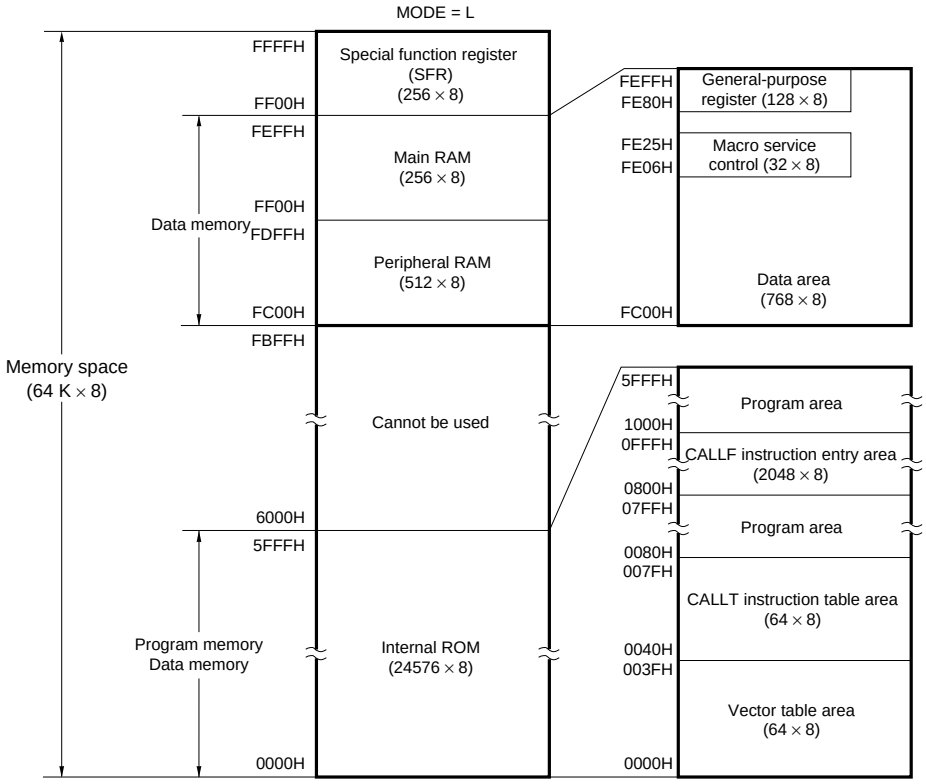
The μPD78362A can access a memory space of 64K bytes. Figure 2-1 and 2-2 show the memory map.

★ **Figure 2-1. Memory Map (μPD78361A)**



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

Figure 2-2. Memory Map (μPD78362A)

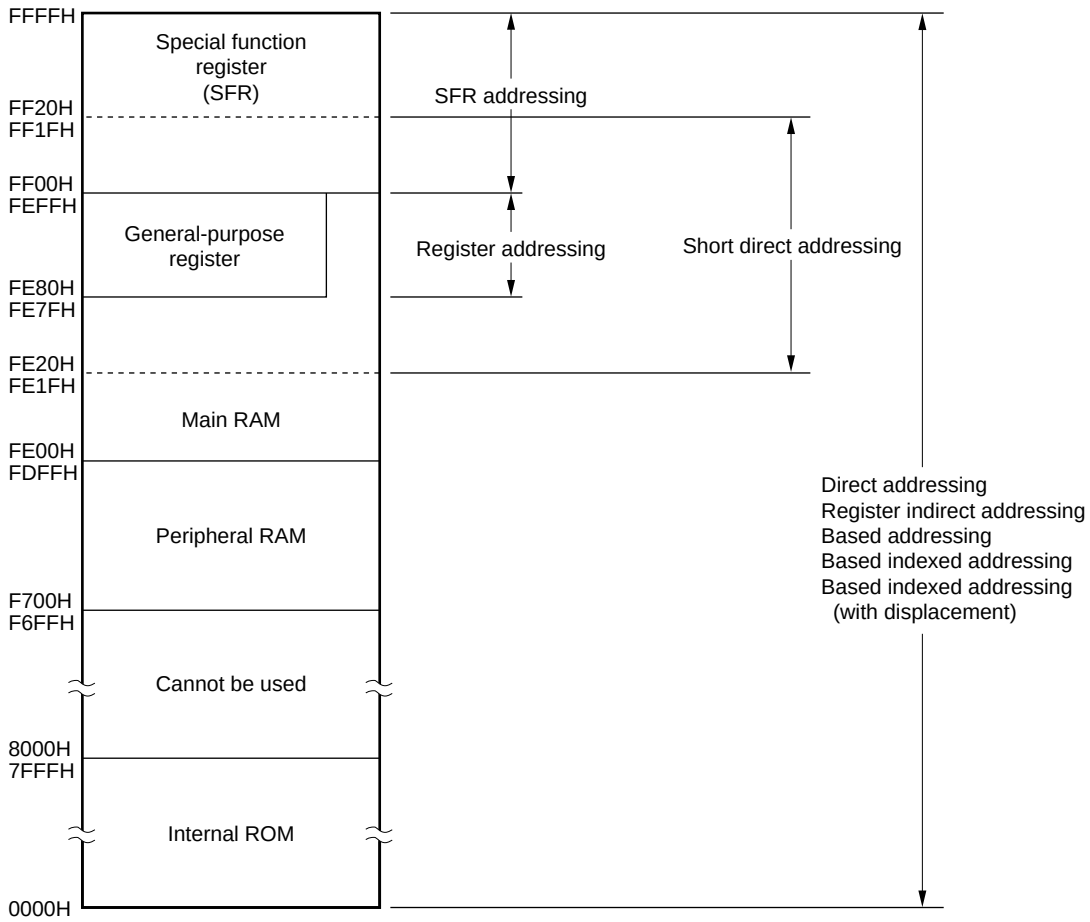


Caution For word access (including stack operations) to the main RAM area (FE00H-FEFFH), the address that specifies the operand must be an even value.

2.2 DATA MEMORY ADDRESSING

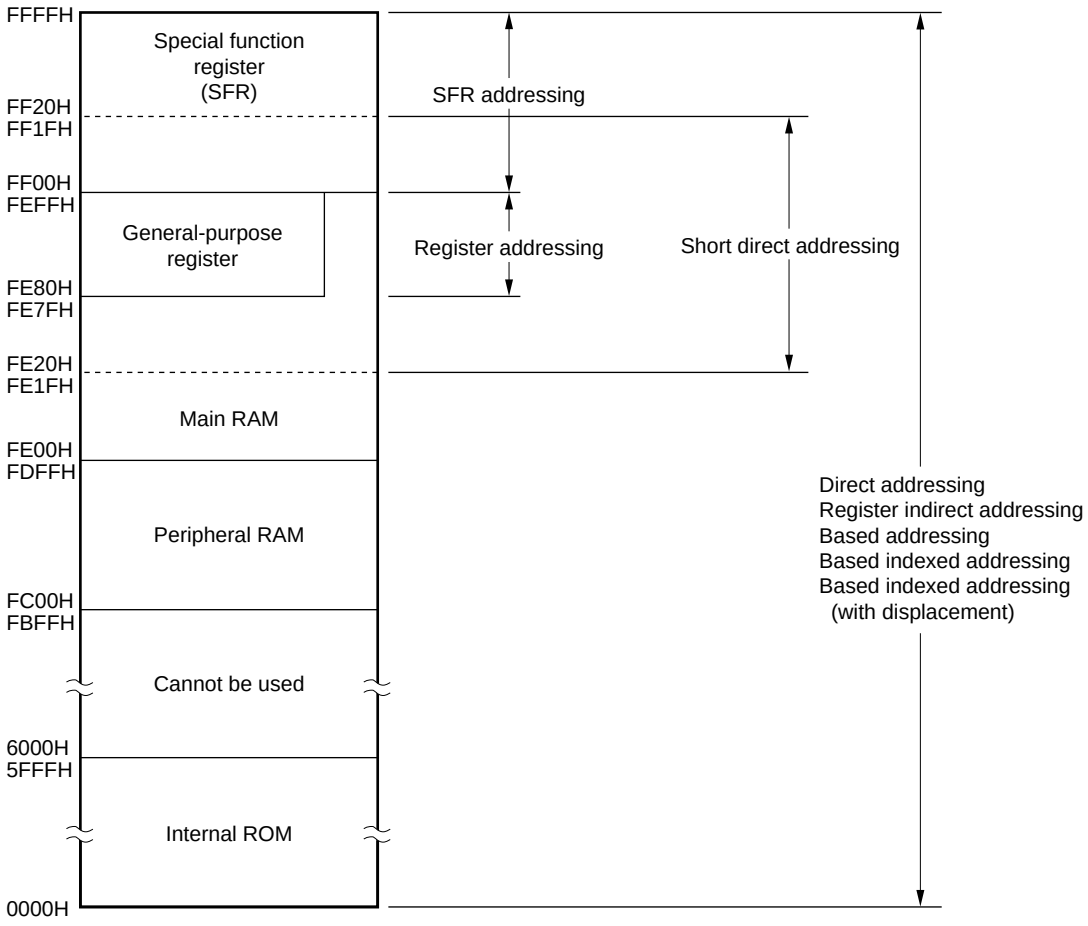
The μPD78362A is provided with many addressing modes that improve the operability of the memory and can be used with high-level languages. Especially, an area of addresses FC00H-FFFFH (F700H-FFFFH in the μPD78361A) to which the data memory is mapped can be addressed in a mode peculiar to the functions provided in this area, including special function registers (SFR) and general-purpose registers.

★ **Figure 2-3. Data Memory Addressing (μPD78361A)**



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

Figure 2-4. Data Memory Addressing (μPD78362A)



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

2.3 PROCESSOR REGISTERS

The μPD78362A is provided with the following three types of processor registers:

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

2.3.1 Control Registers

(1) Program counter (PC)

This is a 16-bit register that holds an address of the instruction to be executed next.

(2) Program status word (PSW)

This 16-bit register indicates the status of the CPU as a result of instruction execution.

(3) Stack pointer (SP)

This 16-bit register indicates the first address of the stack area (LIFO) of the memory.

Figure 2-5. Configuration of Control Registers

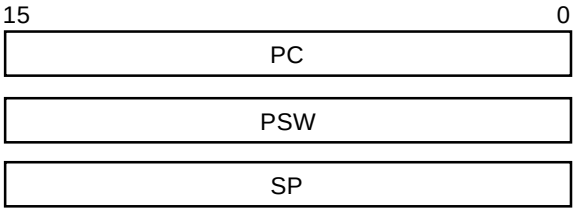
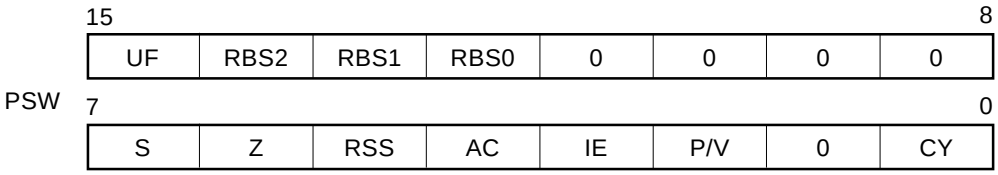


Figure 2-6. Configuration of PSW



UF : User flag
RBS0-RBS2: Register bank select flag
S : Sign flag (MSB of execution result)
Z : Zero flag
RSS : Register set select flag
AC : Auxiliary carry flag
IE : Interrupt request enable flag
P/V : Parity/overflow flag
CY : Carry flag

2.3.2 General-Purpose Registers

The μPD78362A is provided with eight banks of general-purpose registers with one bank consisting of 8 words × 16 bits. Figure 2-7 shows the configuration of the general-purpose register banks. The general-purpose registers are mapped to an area of addresses FE80H-FEFFH. Each of these registers can be used as an 8-bit register. In addition, two registers can be used as one 16-bit register pair (refer to **Figure 2-8**). These general-purpose registers facilitate complicated multitask processing.

Figure 2-7. Configuration of General-Purpose Register Banks

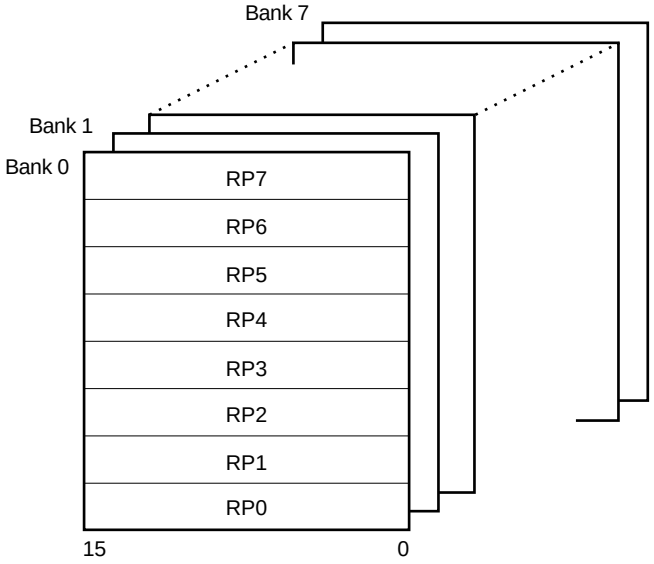
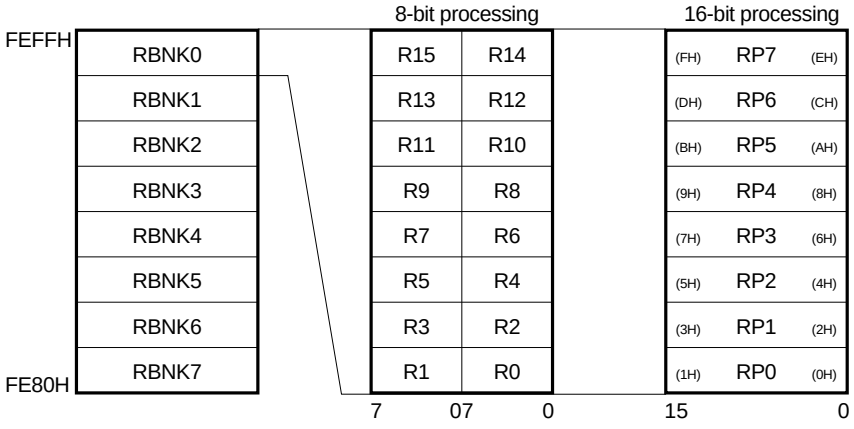


Figure 2-8. Processing Bits of General-Purpose Registers



2.3.3 Special Function Registers (SFR)

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

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

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

Cautions 1. Do not access the addresses in the range FF00H-FFFFH to which no special function register is allocated. If these addresses are accessed, malfunctio ning may occur.

2. Do not write data to the read-only registers. Otherwise, the internal circuit may not operate normally.

3. When using read data as byte data, process undefined bit(s) first.

4. TOUT and TXS are write-only registers. Do no read these registers.

5. Bits 0, 1, and 4 of SBIC are write-only bits. When these bits are read, they are always "0".

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

Address	Special function register (SFR)	Symbol	R/W	Bit units for manipulation			On reset
				1 bit	8 bits	16 bits	
FF00H	Port 0	P0	R/W	○	○	–	Undefined
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		○	○	–	
FF10H FF11H	Compare register 00	CM00		–	–	○	
FF12H FF13H	Compare register 01	CM01		–	–	○	
FF14H FF15H	Compare register 02	CM02		–	–	○	
FF16H FF17H	Compare register 03	CM03		–	–	○	
FF18H FF19H	Buffer register CM00	BFCM00		–	–	○	
FF1AH FF1BH	Buffer register CM01	BFCM01		–	–	○	
FF1CH FF1DH	Buffer register CM02	BFCM02		–	–	○	
FF1EH FF1FH	Timer register 0	TM0	R	–	–	○	0000H
FF20H	Port 0 mode register	PM0	R/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		○	○	–	×××× ×111B
FF2CH FF2DH	Reload register	DTIME	R/W	–	–	○	Undefined
FF2EH	Timer unit mode register 0	TUM0		○	○	–	00H
FF2FH	Timer unit mode register 1	TUM1		○	○	–	
FF30H FF31H	Compare register 10	CM10		–	–	○	Undefined
FF32H FF33H	Timer register 1	TM1	R	–	–	○	0000H

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

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

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

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

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

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

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

Note Can be written when a special instruction is executed.

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

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

★ **Note** The value of the MM register on reset differs depending on the product.
μPD78361A ···· 20H
μPD78362A ···· 60H

3. FUNCTIONAL BLOCKS

3.1 EXECUTION UNIT (EXU)

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

3.2 BUS CONTROL UNIT (BCU)

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

3.3 ROM/RAM

★ Internal ROM and RAM capacities differ depending on the product.

The μPD78361A has a 32K-byte ROM and a 1792-byte peripheral RAM.

The μPD78362A has a 24K-byte ROM and a 512-byte peripheral RAM.

3.4 PORT FUNCTIONS

The μPD78362A is provided with the ports shown in Figure 3-1 for various control operations.

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

Figure 3-1. Port Configuration

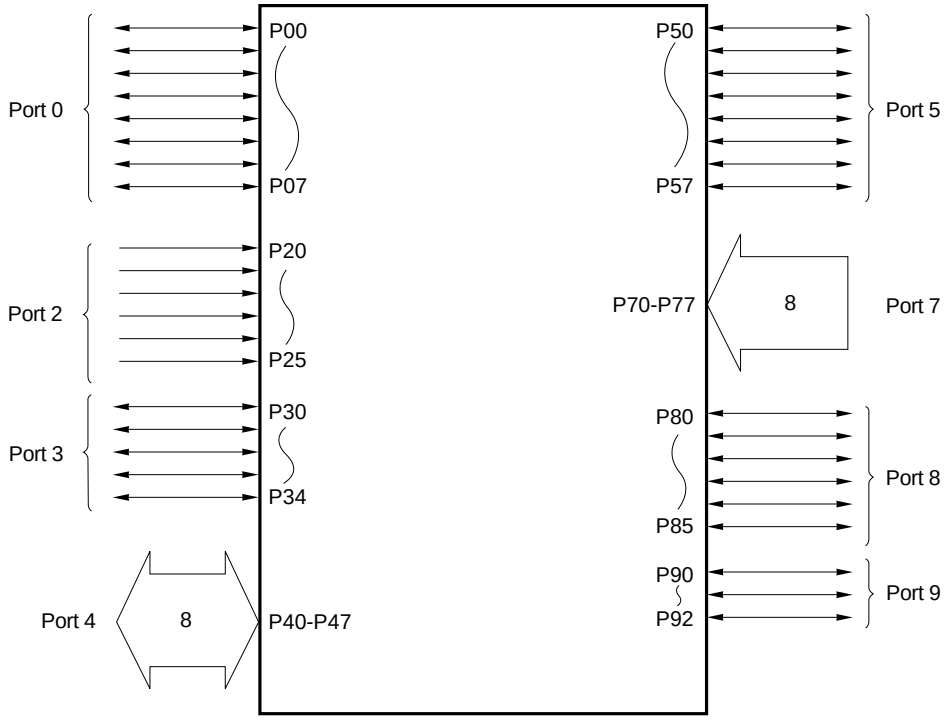


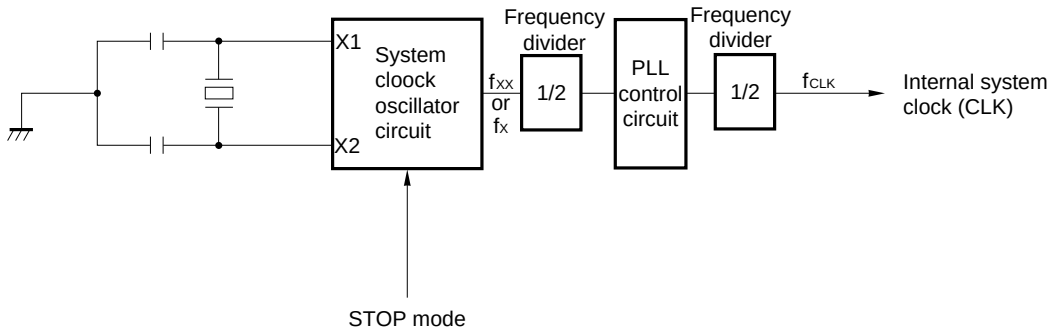
Table 3-1. Functions of Each Port

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

3.5 CLOCK GENERATOR CIRCUIT

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

Figure 3-2. Block Diagram of Clock Generator Circuit



Remarks 1. f_{XX} : crystal oscillation frequency

2. f_X : external clock frequency

3. f_{CLK}: internal system clock frequency

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

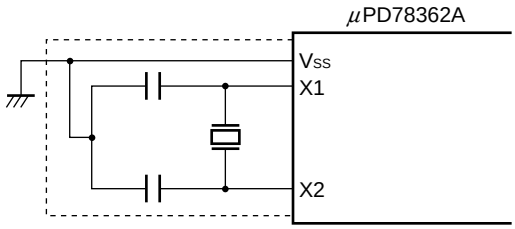
The system clock oscillation circuit oscillates by using the crystal resonator connected across the X1 and X2 pins. It stops oscillation in standby mode.

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

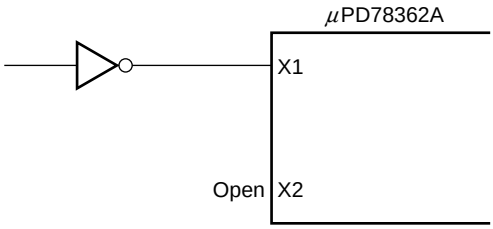
Caution Do not set STOP mode when the external clock is used.

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

(a) crystal oscillator



(b) external clock



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

- **Keep the wiring length as short as possible.**
- **Do not cross the wiring with the other signal line. Make sure that the wiring is not close to lines through which a high alternating current flows.**
- **Always keep the ground point of the capacitor of the oscillation circuit at the same potential as V_{SS}. Do not ground the circuit to a ground pattern through which a high current flows.**
- **Do not extract signals from the oscillator circuit.**

2. To input an external clock, do not connect a load such as wiring capacitance to the X2 pin.

3.6 REAL-TIME PULSE UNIT (RPU)

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

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

Each timer has the following features:

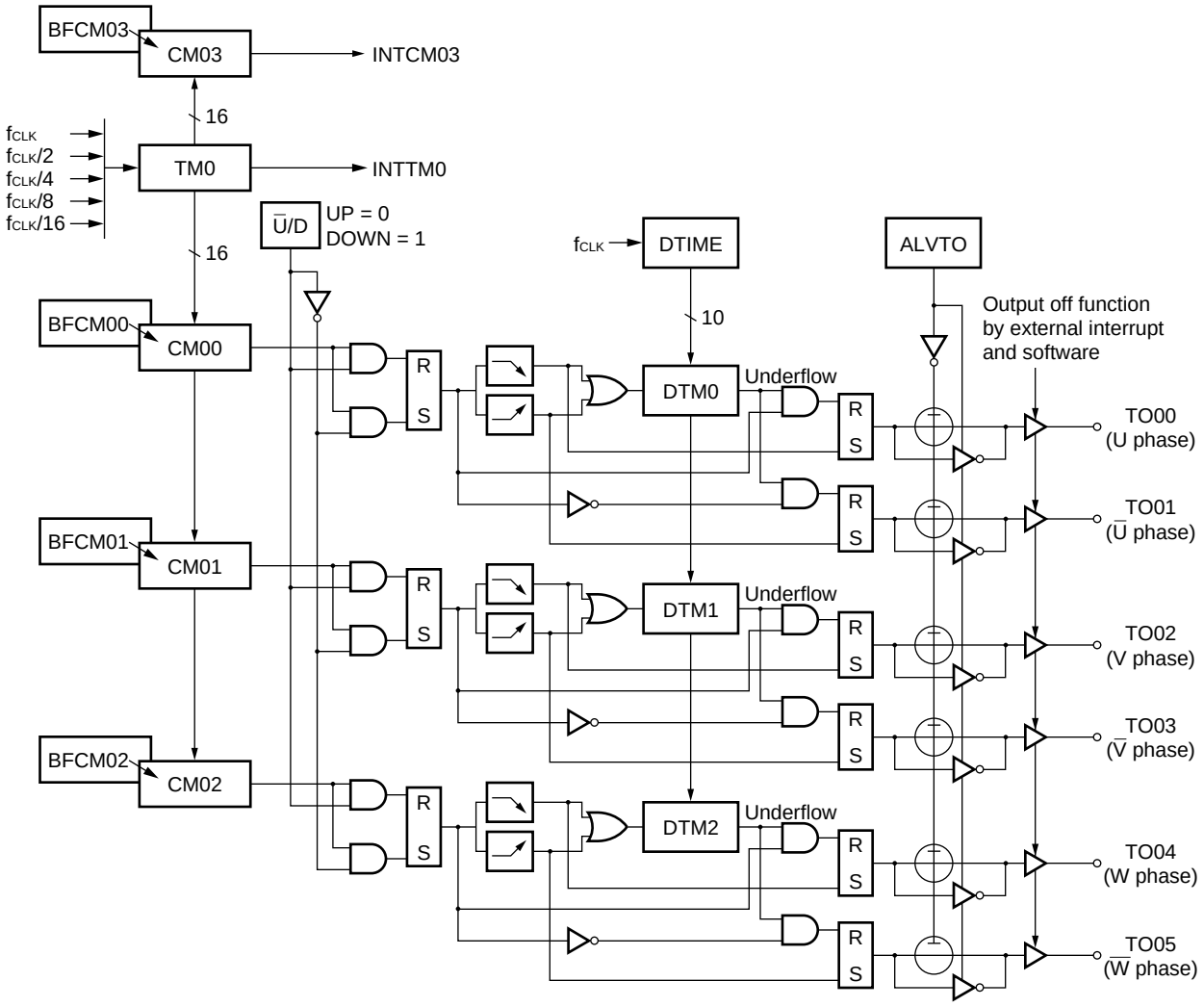
- Timer 0 : Controls the PWM period of the TO00 through TO05 pins. In addition, operates as a general-purpose interval timer. Timer 0 has the following five operation modes:
  - General-purpose interval timer mode
  - PWM mode 0 (symmetrical triangular wave)
  - PWM mode 0 (asymmetrical triangular wave)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

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

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

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

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

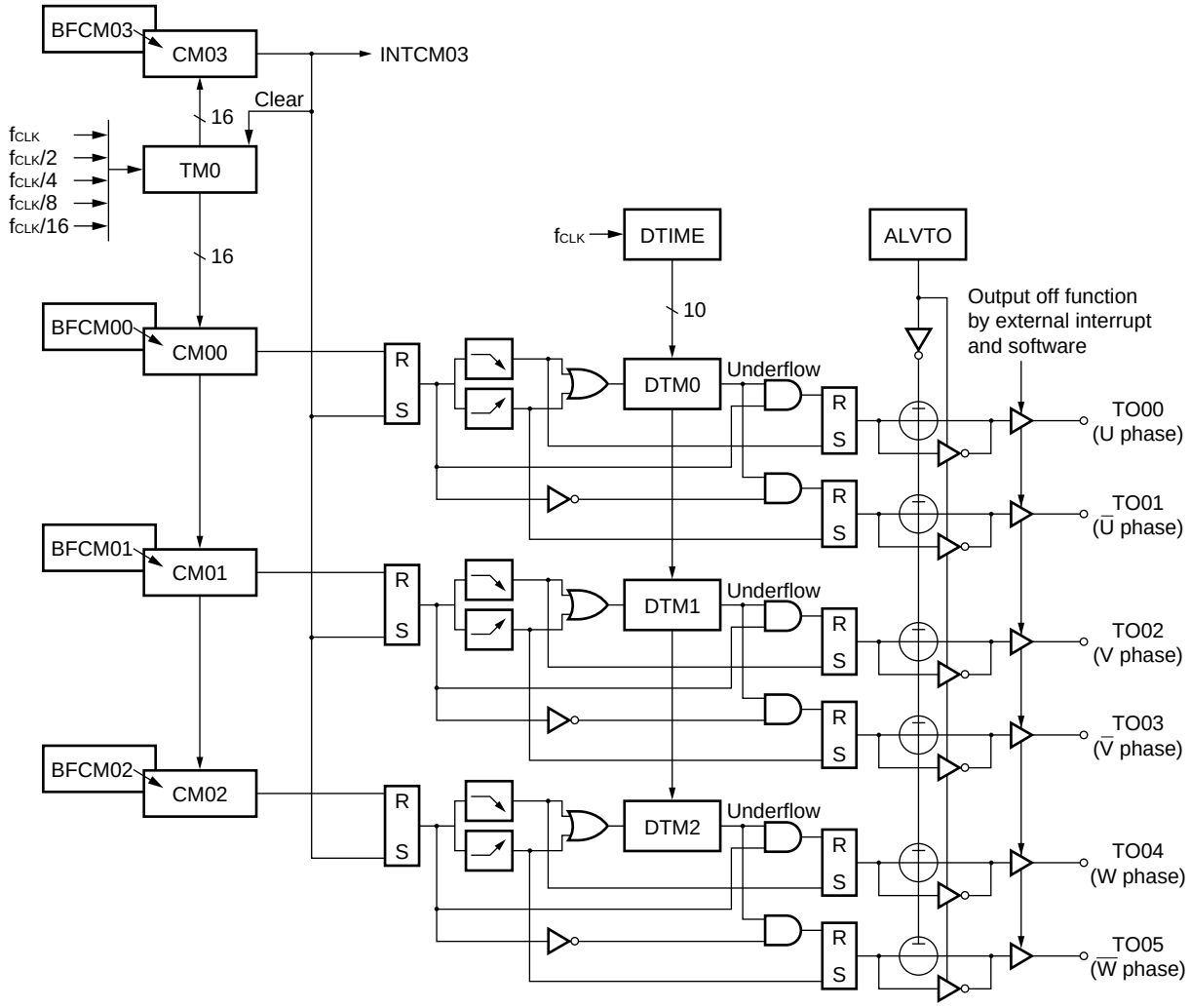


TM0 : Timer register
CM00-CM03 : Compare registers
BFCM00-BFCM03: Buffer registers
DTIME : Reload register
DTM0-DTM2 : Dead time timers

ALVT0 : Bit 2 of TUM0 register
Ū/D : Bit 3 of TMC0 register

Remark f_{CLK}: internal system clock

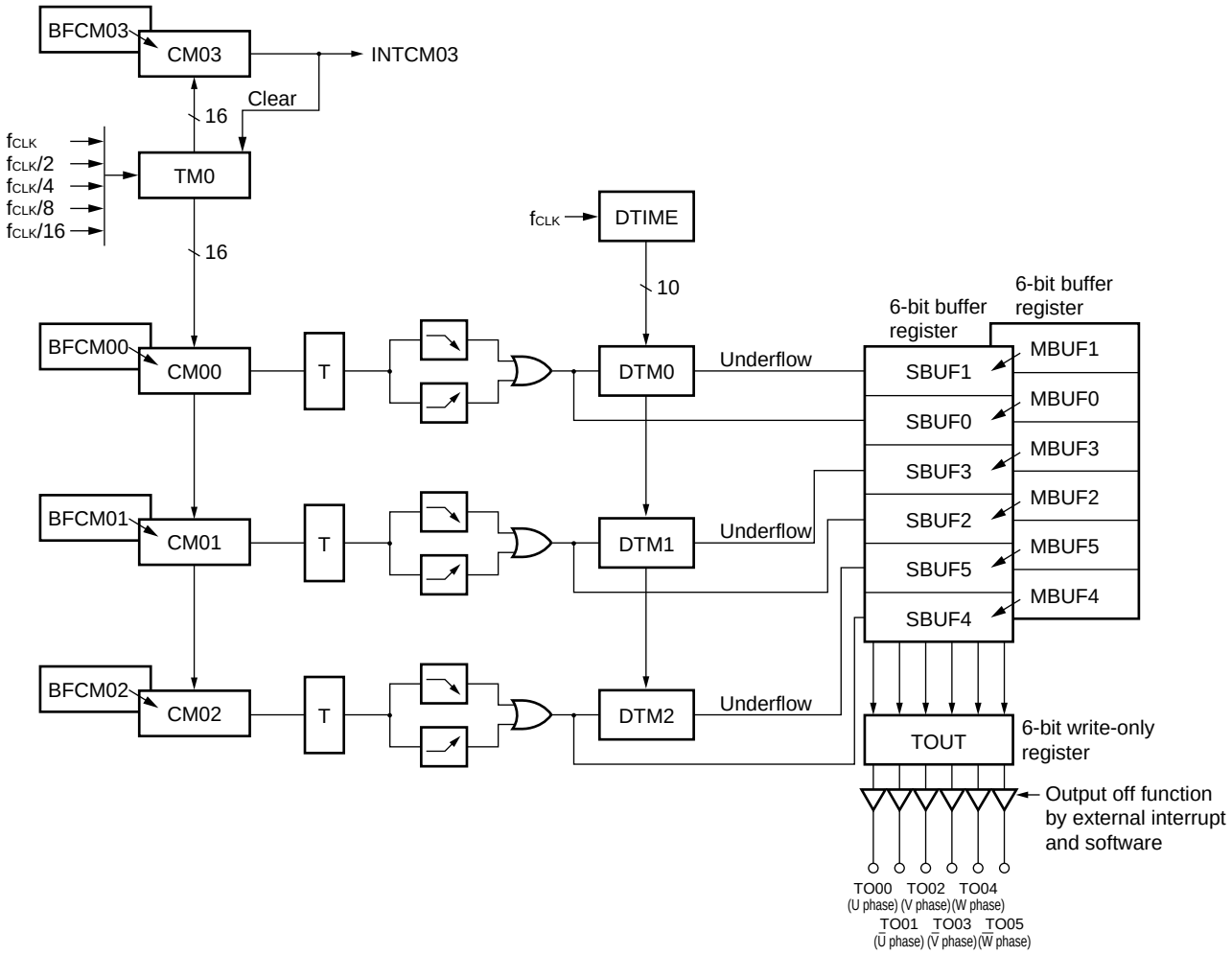
Figure 3-5. Block Diagram of Timer 0 (PWM mode 0 ... saw-tooth wave)



TM0	: Timer register
CM00-CM03	: Compare registers
BFCM00-BFCM03	: Buffer registers
DTIME	: Reload register
DTM0-DTM2	: Dead time timers
ALVT0	: Bit 2 of TUM0 register

Remark f_{CLK}: internal system clock

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



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

Remark f_{CLK}: internal system clock

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

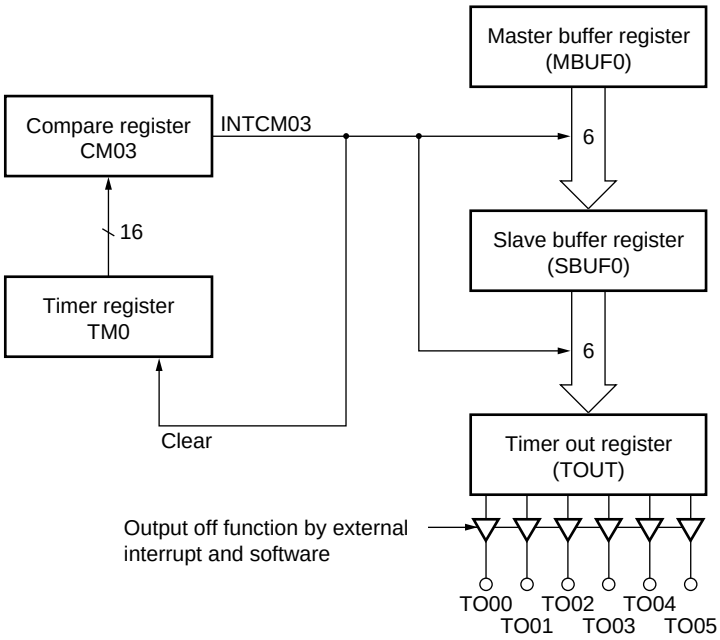
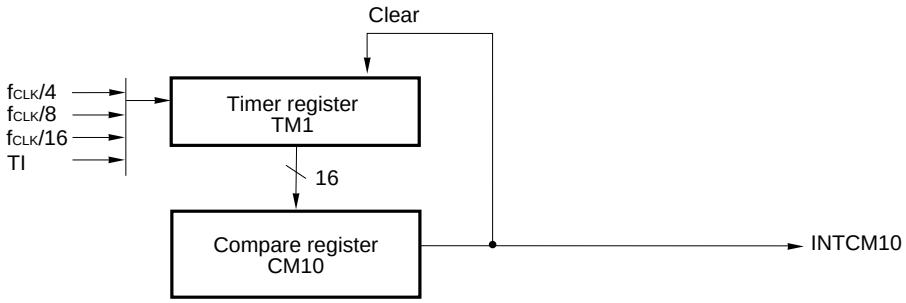
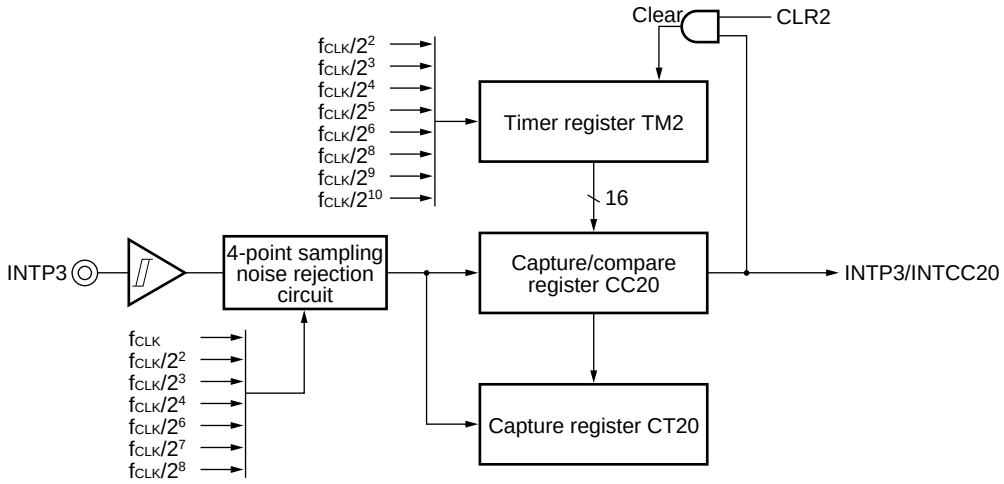


Figure 3-8. Block Diagram of Timer 1



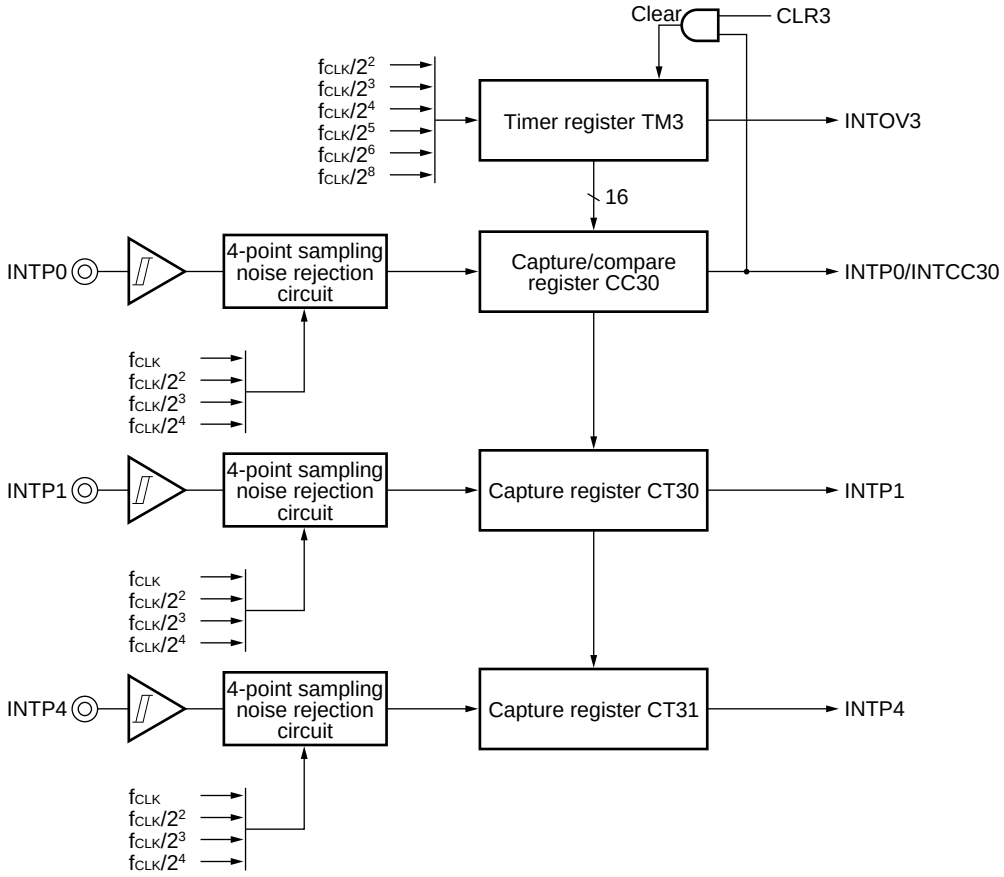
Remark f_{CLK}: internal system clock

Figure 3-9. Block Diagram of Timer 2



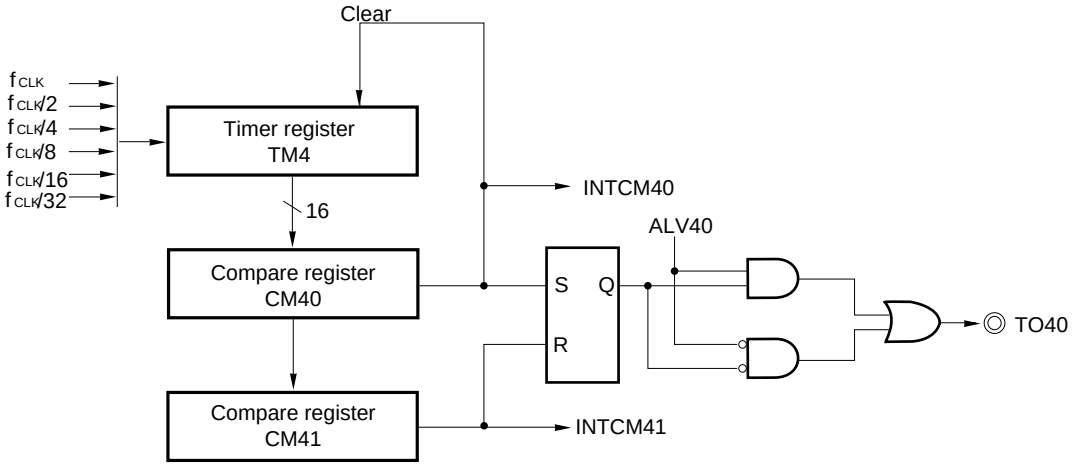
Remark f_{CLK}: internal system clock

Figure 3-10. Block Diagram of Timer 3



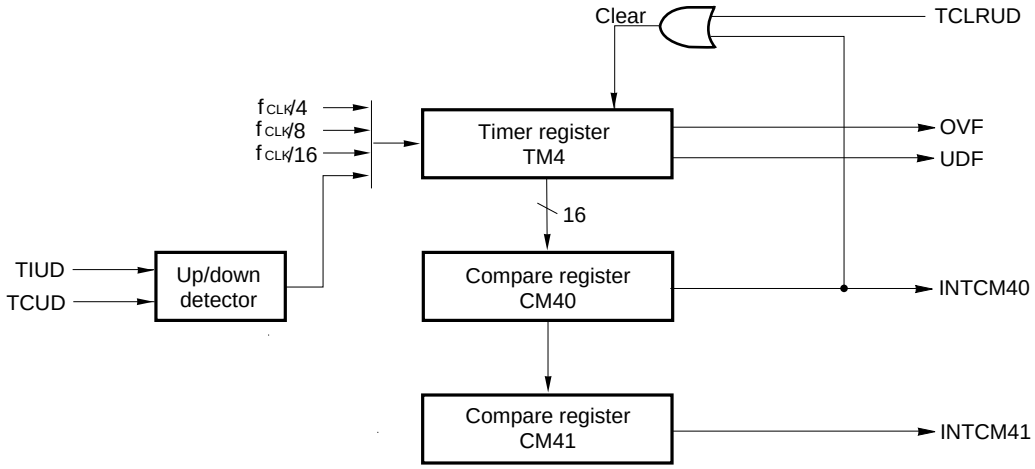
Remark f_{CLK}: internal system clock

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



Remark f_{CLK}: internal system clock

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



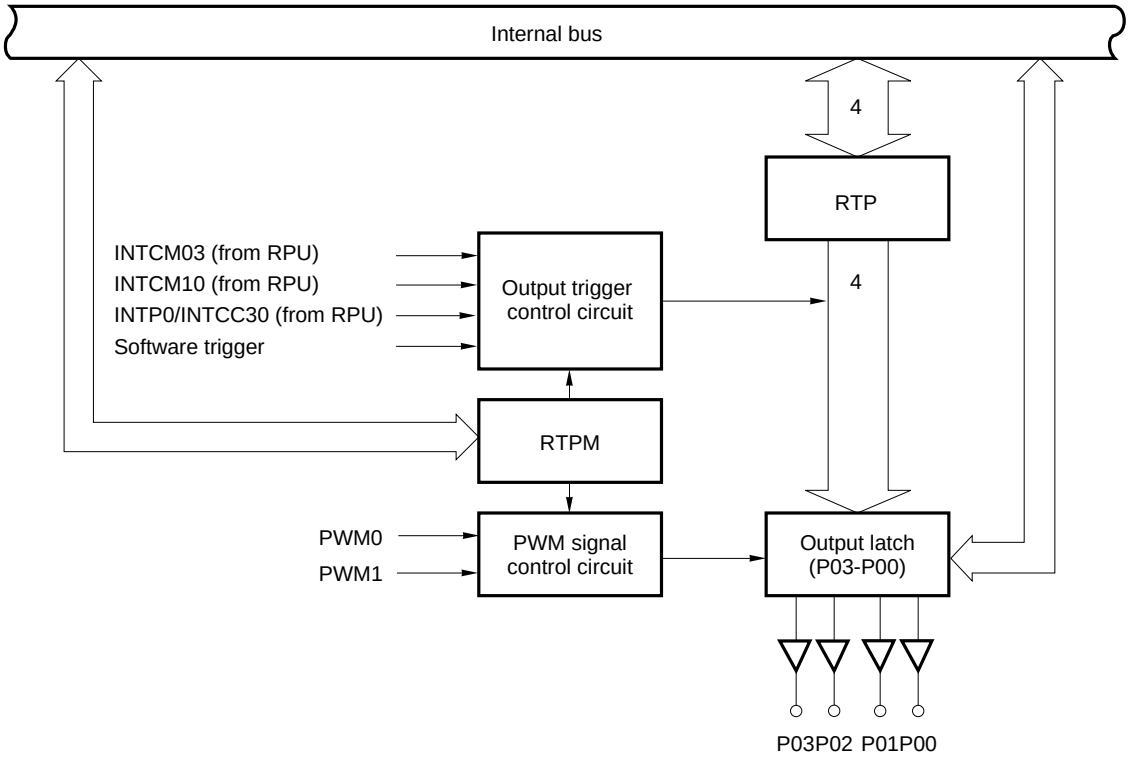
Remark f_{CLK}: internal system clock

3.7 REAL-TIME OUTPUT PORT (RTP)

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

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

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

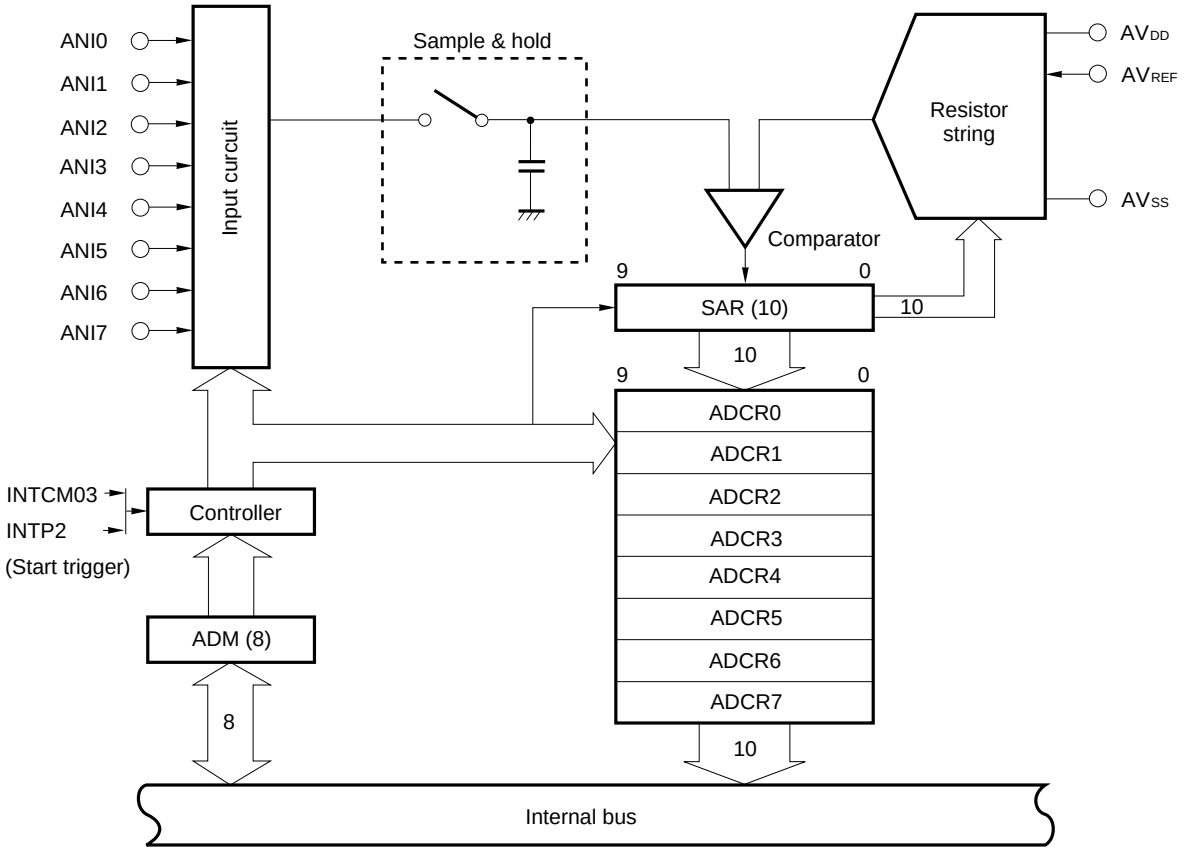


3.8 A/D CONVERTER

The μPD78362A contains a high-speed, high-resolution 10-bit analog-to-digital (A/D) converter (conversion time 12.6μs at an internal clock frequency of 16 MHz). Successive approximation type is adopted. This converter is provided with eight analog input lines (ANI0-ANI7) and can perform various operations as the application requires, in select, scan, and mixed modes.

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

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



3.9 SERIAL INTERFACE

The μPD78362A is provided with the following two independent serial interfaces:

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

Since the μPD78362A contains a baud rate generator (BRG), any serial transfer rate can be set regardless of the operating clock frequency. The baud rate generator is a block to generate the shift clock for the transmit/receive serial interface, and is used commonly with the two channels of the serial interfaces.

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

Figure 3-15. Block Diagram of Asynchronous Serial Interface

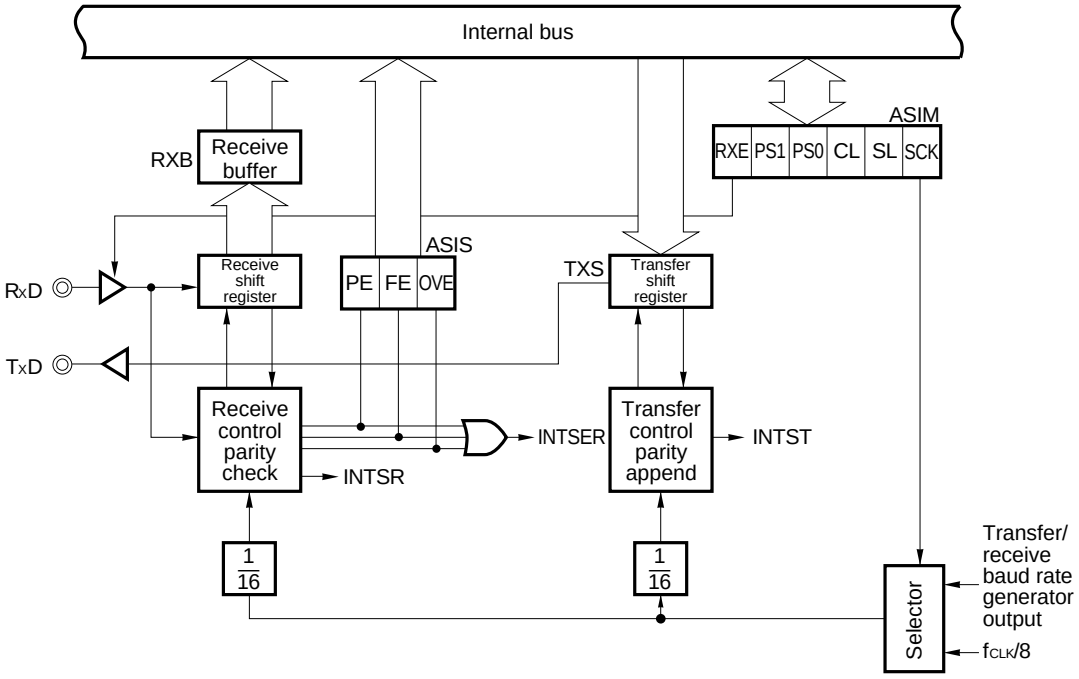


Figure 3-16. Block Diagram of Clocked Serial Interface

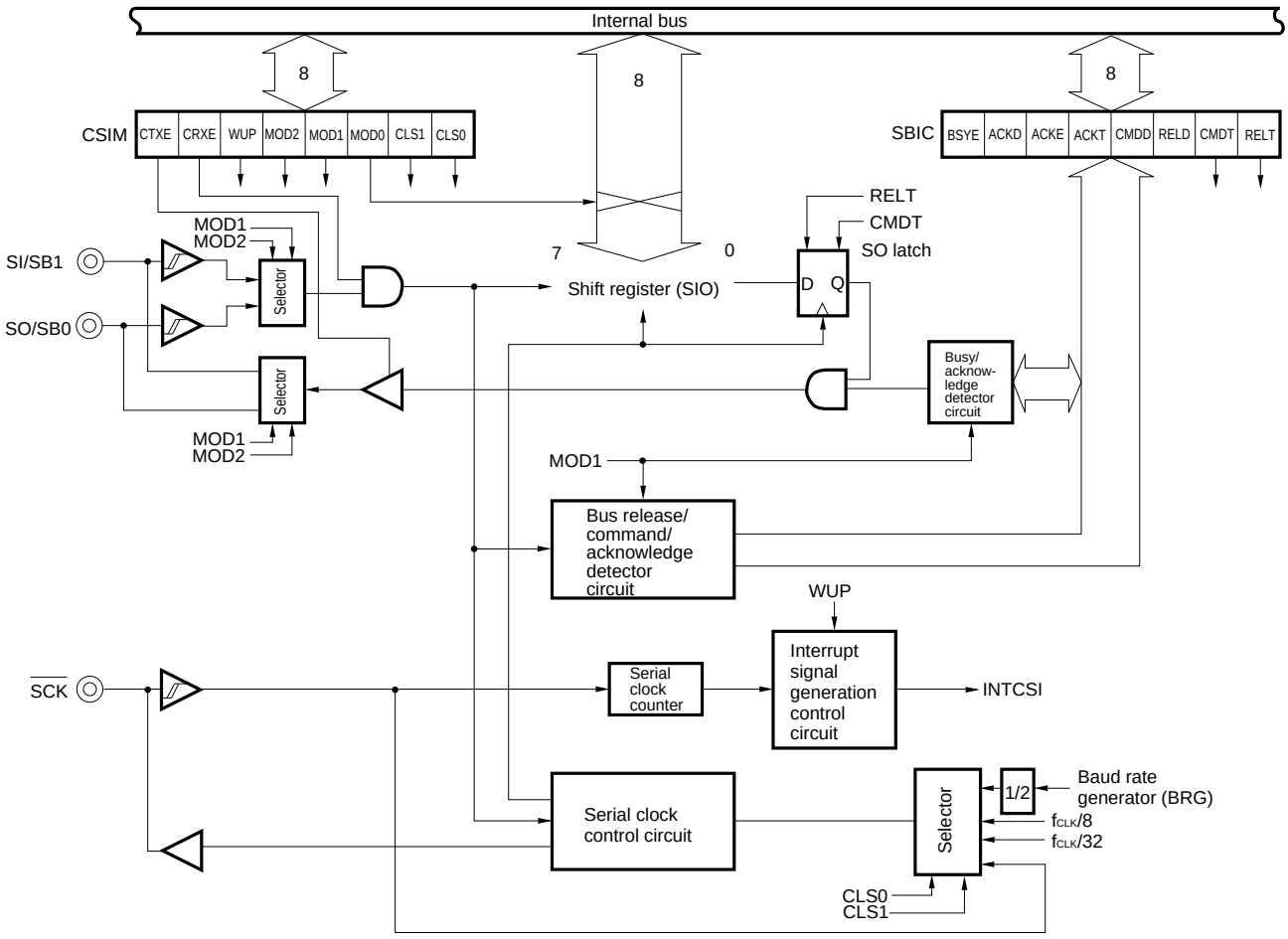
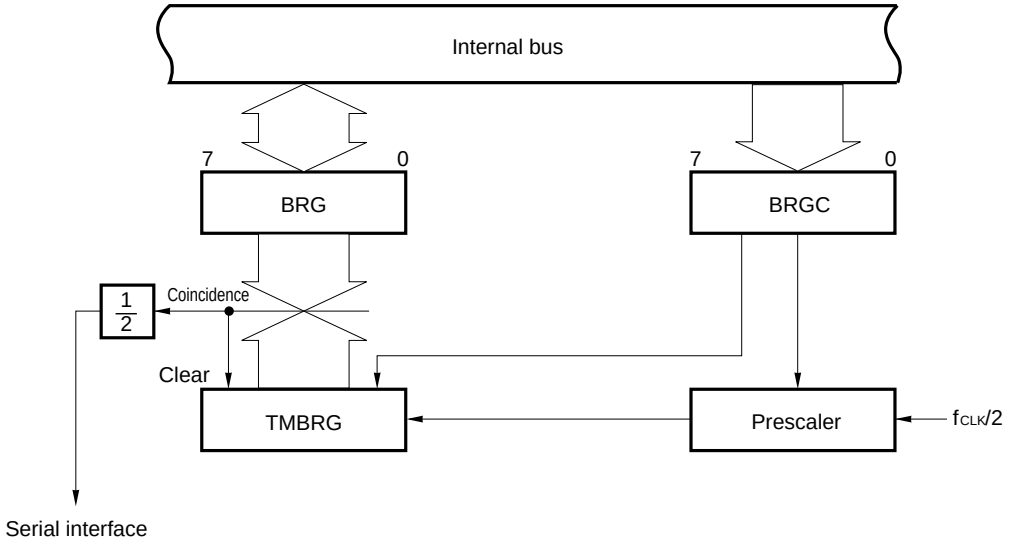


Figure 3-17. Block Diagram of Baud Rate Generator

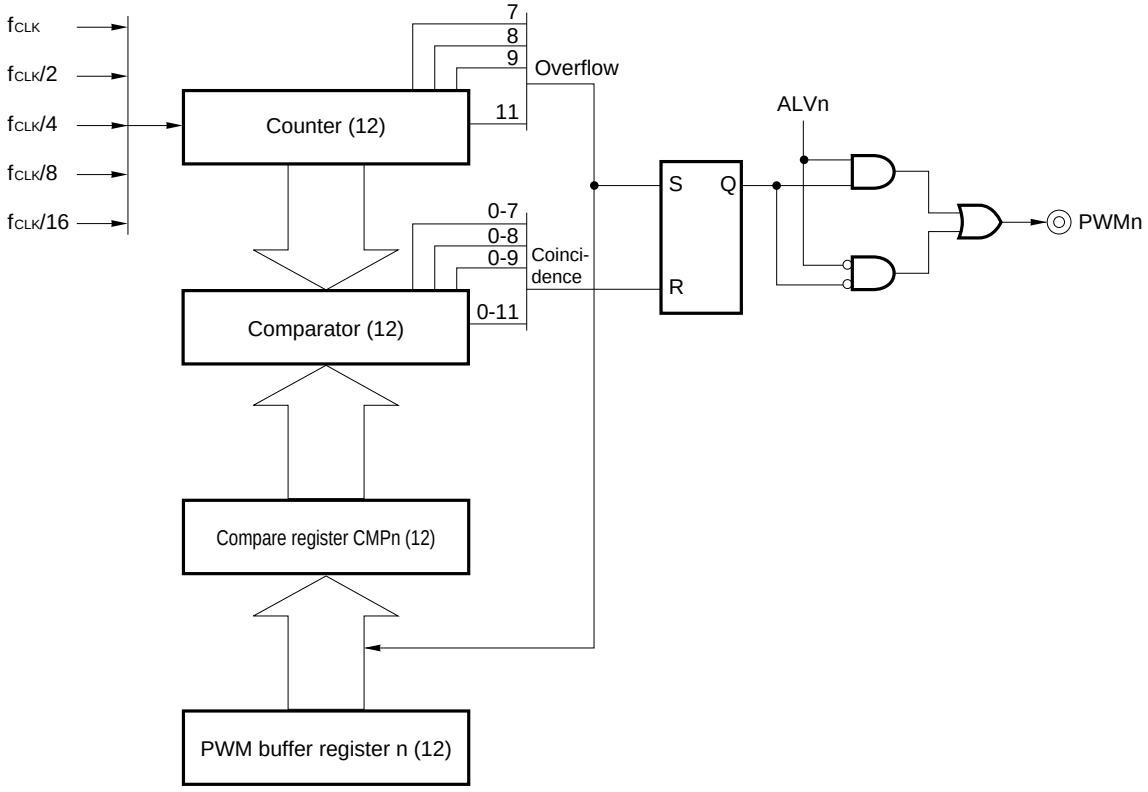


3.10 PWM UNIT

The μPD78362A is provided with two lines that output 8-/9-/10-/12-bit resolution variable PWM signals. The PWM output can be used as a digital-to-analog conversion output by connecting an external lowpass filter, and ideal for controlling actuators such as motors.

An output of between 244 Hz and 62.5 kHz can be obtaind, depending on the combination of the count clock (62.5 ns to 1 μs) and counter bit length (8, 9, 10, or 12) (at an internal clock frequency of 16 MHz).

Figure 3-18. Block Diagram of PWM Unit

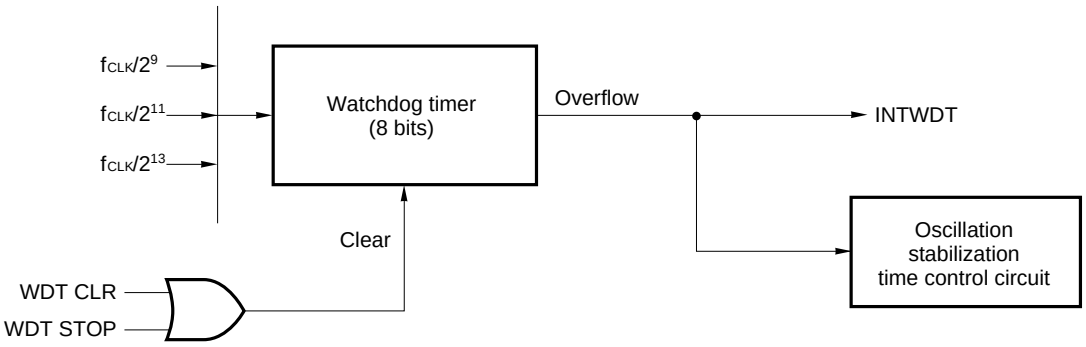


Remark n = 0, 1

3.11 WATCHDOG TIMER (WDT)

The watchdog timer is a free running timer equipped with a non-maskable interrupt function to prevent program hang-up or deadlock. An error of the program can be seen by the occurrence of the overflow interrupt (INTWDT) of the watchdog timer.

Figure 3-19. Block Diagram of Watchdog Timer



4. INTERRUPT FUNCTIONS

4.1 OUTLINE

The μPD78362A is provided with powerful interrupt functions that can service interrupt requests from the internal hardware peripherals and external sources. In addition, the following three interrupt service modes are available. In addition, four levels of interrupt priority can be specified.

- Vectored interrupt service
- Macro service
- Context switching

Table 4-1. Interrupt Sources

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

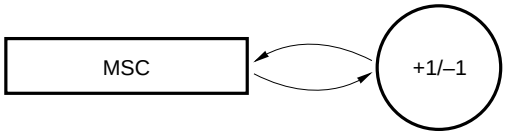
Note Default priority : Priority that takes precedence when two or more maskable interrupts occur at the same time. 0 is the highest priority, and 15 is the lowest.

4.2 MACRO SERVICE

The μPD78362A has a total of five macro services. Each macro service is described below.

(1) Counter mode: EVTCNT

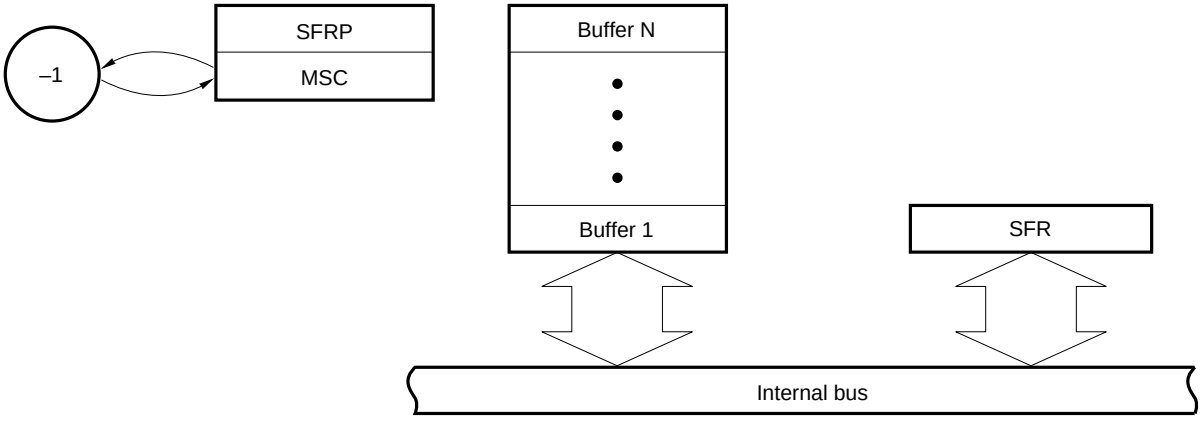
- Operation
  - (a) Increments or decrements an 8-bit macro service counter (MSC).
  - (b) A vectored interrupt request is generated when MSC reaches 0.



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

(2) Block transfer mode: BLKTRS

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



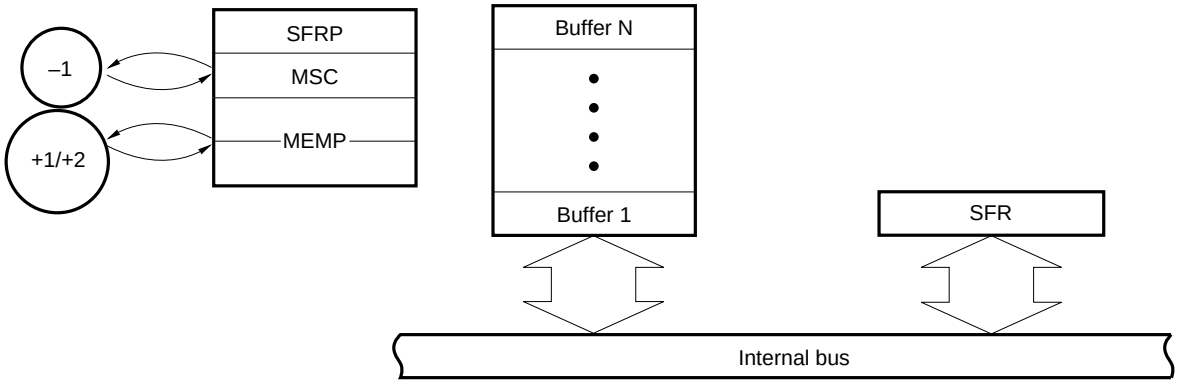
- Application example: To transfer/receive data through serial interface

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

- Operation

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

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

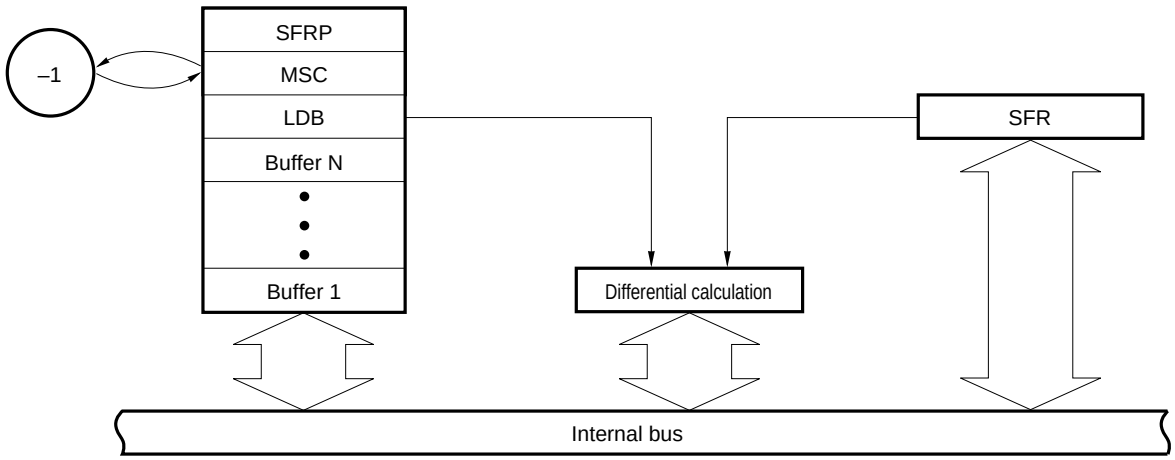


- Application example: Same as **(2)**

(4) Data differential mode: DTADIF

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

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



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

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

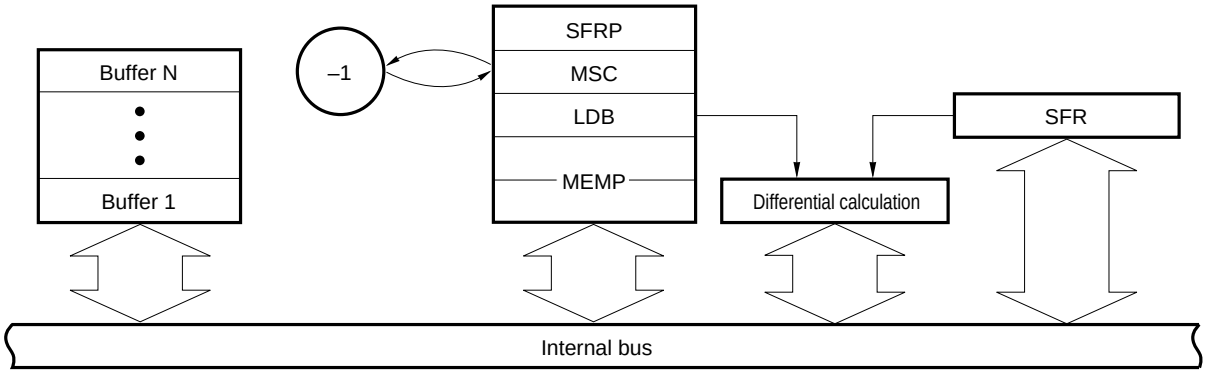
- Operation

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

Remarks 1. The differential calculation can be carried out only with 16-bit SFRs.

2. The buffer is specified by the result of operation by MEMP and MSC[Note]. MEMP is not updated after the data has been transferred.

Note MEMP – (MSC × 2) + 2



- Application example: Same as **(4)**

4.3 CONTEXT SWITCHING

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

4.3.1 Context Switching Function by Interrupt Request

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

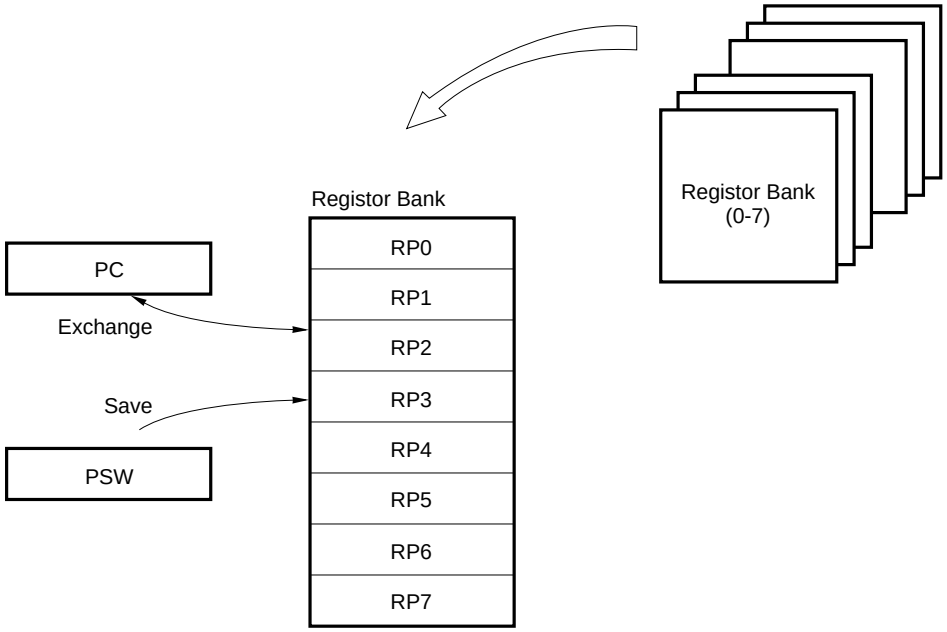
The context switching operation by an interrupt request is performed as follows:

(1) When an interrupt request is generated, a register bank to which the context is to be switched is specified by the contents of the lower 3 bits of the row address (even address) of the corresponding vector table.

(2) A predetermined vector address is transferred to the PC in the register bank to which the context is to be switched, and the contents of the PC and PSW immediately before the switching takes place are saved to the register bank.

(3) Execution branches to an address indicated by the contents of the PC newly set.

Figure 4-1. Operation of Context Switching



4.3.2 Context Switching Function by BRKCS Instruction

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

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

(1) An 8-bit register is specified by the operand of the BRKCS instruction, and the register bank to which the context is to be switched is specified by the contents of this register (only the lower 3 bits of 8 bits are valid).

(2) The vector address predetermined in the register bank to which the context is to be switched is transferred to the PC, and at the same time, the contents of the PC and PSW immediately before the switching takes place are saved to the register bank.

(3) Execution branches to the contents of the PC newly set.

4.3.3 Restoration from Context Switching

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

Table 4-2. Instructions to Restore from Context Switching

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

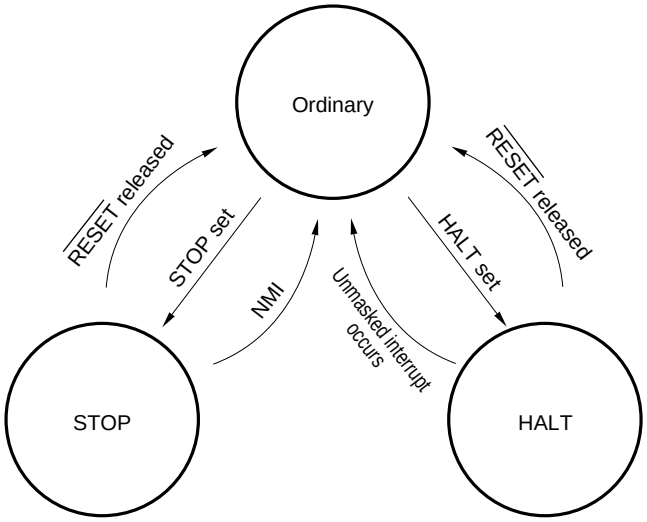
5. STANDBY FUNCTIONS

The μPD78362A is provided with standby functions to reduce the power consumption of the system. The standby functions can be effected in the following two modes:

- HALT mode ..... In this mode, the operating clock of the CPU is stopped. By using this mode in combination with an ordinary operation mode, the μPD78362A operates intermittently to reduce the total power consumption of the system.
- STOP mode .... In this mode, the oscillator is stopped, and therefore the entire system is stopped. Therefore, power consumption can be minimized with only a leakage current flowing.

Each mode is set through software. Figure 5-1 shows the transition of the status in the standby modes (STOP and HALT modes).

Figure 5-1. Transition of Standby Status



6. RESET FUNCTION

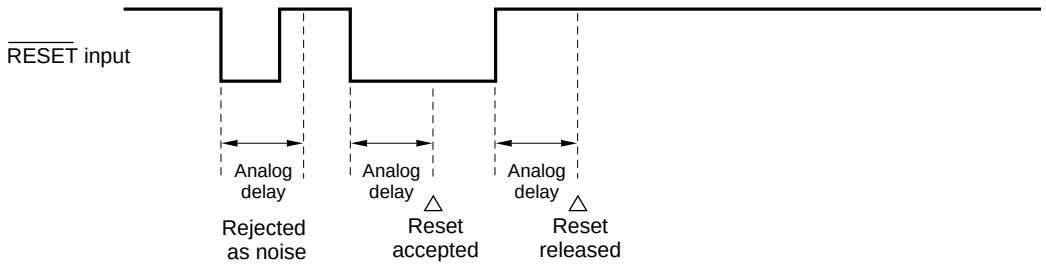
When a low level is input to the $\overline{\text{RESET}}$ pin, the system is reset, and each hardware enters the initial status (reset status). When the $\overline{\text{RESET}}$ pin goes high, the reset status is released, and program execution is started. Initialize the contents of each register through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

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

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

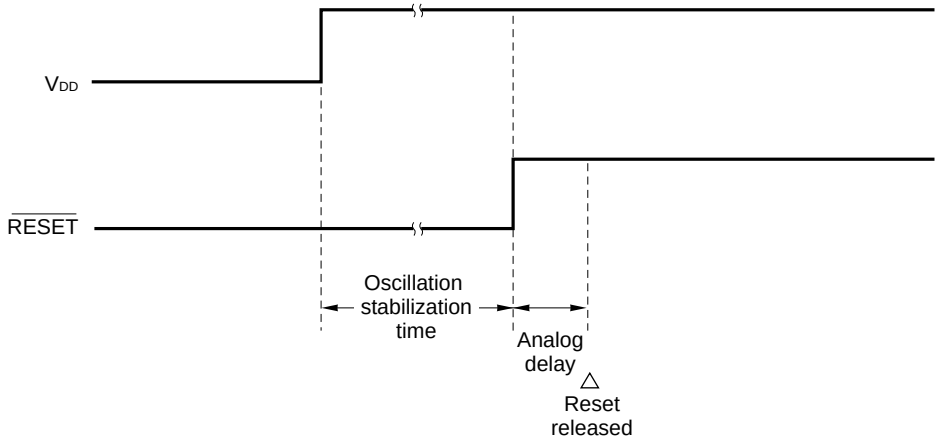
Caution While the $\overline{\text{RESET}}$ pin is active (low level), all the pins go into a high-impedance state (except AV_{REF}, AV_{DD}, AV_{SS}, V_{DD}, V_{SS}, X1, and X2 pins).

Figure 6-1. Accepting Reset Signal



To effect reset on when power is applied, make sure that sufficient time elapses to stabilize the oscillation after the power is applied until the reset signal is accepted, as shown in Figure 6-2.

Figure 6-2. Reset on Power Application



7. INSTRUCTION SET

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

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

Table 7-1. Operand Representation and Description

Representation	Description
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 symbol (Refer to **Table 2-1**.) Special function register symbol (register that can be manipulated in 16-bit units. Refer to **Table 2-1**.)
post	RP0, RP1, RP2, RP3, RP4, RP5/PSW, RP6, RP7 (More than one symbol can be described. However, RP5 can be described only for PUSH and POP instructions, and PSW can be described only for PUSHU and POPU instructions.)
mem	[DE], [HL], [DE +], [HL +], [DE –], [HL –], [VP], [UP] ; register indirect mode [DE + A], [HL + A], [DE + B], [HL + B], [VP + DE], [VP + HL] ; based indexed mode [DE + byte], [HL + byte], [VP + byte], [UP + byte], [SP + byte] ; based mode word[A], word[B], word[DE], word[HL] ; indexed mode
saddr saddrp	FE20H-FF1FH immediate data or label FE20H-FF1EH immediate data (however, bit0 = 0) or label (manipulated in 16-bit units)
$ addr16 ! addr16 addr11 addr5	0000H-FDFFH immediate data or label; relative addressing 0000H-FDFFH immediate data or label; immediate addressing (However, up to FFFFH can be described for MOV instruction. Only FE00H-FEFFH can be described for MOVTBLW instruction.) 800H-FFFH immediate data or label 40H-7EH immediate data (however, 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 (0-7)

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

Remarks 1. rp and rp1 are the same in terms of register name that can be described but are different in code to be generated.

2. r, r1, rp, rp1, and post can be described in absolute name (R0-R15, RP0-RP7) and function name (X, A, C, B, E, D, L, H, AX, BC, DE, HL, VP, and UP).

3. Immediate addressing can address the entire space. Relative addressing can address only a range of –128 to +127 from the first address of the next instruction.

Instructions	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
8-bit data transfer	MOV	r1, #byte	2	r1 ← byte					
		saddr, #byte	3	(saddr) ← byte					
		sfr[Note], #byte	3	sfr ← byte					
		r, r1	2	r ← r1					
		A, r1	1	A ← r1					
		A, saddr	2	A ← (saddr)					
		saddr, A	2	(saddr) ← A					
		saddr, saddr	3	(saddr) ← (saddr)					
		A, sfr	2	A ← sfr					
		sfr, A	2	sfr ← A					
		A, mem	1-4	A ← (mem)					
		mem, A	1-4	(mem) ← A					
		A, [saddrp]	2	A ← ((saddrp))					
		[saddrp], A	2	((saddrp)) ← A					
		A, !addr16	4	A ← (addr16)					
		!addri16, A	4	(addr16) ← A					
		PSWL, #byte	3	PSW_L ← byte	x	x	x	x	x
		PSWH, #byte	3	PSW_H ← byte					
		PSWL, A	2	PSW_L ← A	x	x	x	x	x
		PSWH, A	2	PSW_H ← A					
		A, PSWL	2	A ← PSW_L					
		A, PSWH	2	A ← PSW_H					
	XCH	A, r1	1	A ↔ r1					
		r, r1	2	r ↔ r1					
		A, mem	2-4	A ↔ (mem)					
		A, saddr	2	A ↔ (saddr)					
		A, sfr	3	A ↔ sfr					
		A, [saddrp]	2	A ↔ ((saddrp))					
		saddr, saddr	3	(saddr) ↔ (saddr)					

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

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

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

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

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

Instructions	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
8-bit operation	**OR**	A, #byte	2	A ← A ∨ byte	x	x		P	
		saddr, #byte	3	(saddr) ← (saddr) ∨ byte	x	x		P	
		sfr, #byte	4	sfr ← sfr ∨ byte	x	x		P	
		r, r1	2	r, ← r ∨ r1	x	x		P	
		A, saddr	2	A ← A ∨ (saddr)	x	x		P	
		A, sfr	3	A ← A ∨ sfr	x	x		P	
		saddr, saddr	3	(saddr) ← (saddr) ∨ (saddr)	x	x		P	
		A, mem	2-4	A ← A ∨ (mem)	x	x		P	
		mem, A	2-4	(mem) ← (mem) ∨ A	x	x		P	
	XOR	A, #byte	2	A ← A ⊻ byte	x	x		P	
		saddr, #byte	3	(saddr) ← (saddr) ⊻ byte	x	x		P	
		sfr, #byte	4	sfr ← sfr ⊻ byte	x	x		P	
		r, r1	2	r ← r ⊻ r1	x	x		P	
		A, saddr	2	A ← A ⊻ (saddr)	x	x		P	
		A, sfr	3	A ← A ⊻ sfr	x	x		P	
		saddr, saddr	3	(saddr) ← (saddr) ⊻ (saddr)	x	x		P	
		A, mem	2-4	A ← A ⊻ (mem)	x	x		P	
		mem, A	2-4	(mem) ← (mem) ⊻ A	x	x		P	
	CMP	A, #byte	2	A – byte	x	x	x	V	x
		saddr, #byte	3	(saddr) – byte	x	x	x	V	x
		sfr, #byte	4	sfr – byte	x	x	x	V	x
		r, r1	2	r – r1	x	x	x	V	x
		A, saddr	2	A – (saddr)	x	x	x	V	x
		A, sfr	3	A – sfr	x	x	x	V	x
		saddr, saddr	3	(saddr) – (saddr)	x	x	x	V	x
		A, mem	2-4	A – (mem)	x	x	x	V	x
		mem, A	2-4	(mem) – A	x	x	x	V	x

Instructions	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
16-bit operation	**ADDW**	AX, #word	3	AX, CY ← AX + word	x	x	x	V	x
		saddrp, #word	4	(saddrp), CY ← (saddrp) + word	x	x	x	V	x
		sfrp, #word	5	sfrp, CY ← sfrp + word	x	x	x	V	x
		rp, rp1	2	rp, CY ← rp + rp1	x	x	x	V	x
		AX, saddrp	2	AX, CY ← AX + (saddrp)	x	x	x	V	x
		AX, sfrp	3	AX, CY ← AX + sfrp	x	x	x	V	x
		saddrp, saddrp	3	(saddrp), CY ← (saddrp) + (saddrp)	x	x	x	V	x
	SUBW	AX, #word	3	AX, CY ← AX – word	x	x	x	V	x
		saddrp, #word	4	(saddrp), CY ← (saddrp) – word	x	x	x	V	x
		sfrp, #word	5	sfrp, CY ← sfrp – word	x	x	x	V	x
		rp, rp1	2	rp, CY ← rp – rp1	x	x	x	V	x
		AX, saddrp	2	AX, CY ← AX – (saddrp)	x	x	x	V	x
		AX, sfrp	3	AX, CY ← AX – sfrp	x	x	x	V	x
		saddrp, saddrp	3	(saddrp), CY ← (saddrp) – (saddrp)	x	x	x	V	x
	CMPW	AX, #word	3	AX – word	x	x	x	V	x
		saddrp, #word	4	(saddrp) – word	x	x	x	V	x
		sfrp, #word	5	sfrp – word	x	x	x	V	x
		rp, rp1	2	rp – rp1	x	x	x	V	x
		AX, saddrp	2	AX – (saddrp)	x	x	x	V	x
		AX, sfrp	3	AX – sfrp	x	x	x	V	x
		saddrp, saddrp	3	(saddrp) – (saddrp)	x	x	x	V	x
Multiplication /division	**MULU**	r1	2	AX ← AX × r1					
	DIVUW	r1	2	AX (quotient), r1 (remainder) ← AX ÷ r1					
	MULUW	rp1	2	AX (higher 16 bits), rp1 (lower 16 bits) ← AX × rp1					
	DIVUX	rp1	2	AXDE (quotient), rp1 (remainder) ← AXDE ÷ rp1					
Signed multipli-cation	**MULW**	rp1	2	AX (higher 16 bits), rp1 (lower 16 bits) ← AX × rp1					
Sum-of-products operation	**MACW**	n	3	AXDE ← (B) × (C) + AXDE B ← B + 2, C ← C + 2, n ← n – 1 End if n = 0 or P/V = 1	x	x	x	V	x

Sum-of-products operation with saturation	**MACSW**	n	3	AXDE ← (B) × (C) + AXDE
				B ← B + 2, C ← C + 2, n ← n – 1
				if overflow (P/V = 1) then
				AXDE ← 7FFFFFFFH
				if underflow (P/V = 1) then
				AXDE ← 80000000H
				end if n = 0 or P/V = 1
					x	x	x	V	x

Relative operation	**SACW**	[DE +], [HL +]	4	AX ← AX + | (DE) – (HL) | DE ← DE + 2 HL ← HL + 2 C ← C – 1 end if C = 0 or cy = 1	x	x	x	V	x

Instructions	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
Table shift	**MOVTBLW**	!addr16, n	4	(addr16 + 2) ← (addr16), n ← n-1 addr16 ←addr16-2, End if n = 0					
Increment/decrement	**INC**	r1	1	r1 ← r1 + 1	x	x	x	V	
		saddr	2	(saddr) ← (saddr) + 1	x	x	x	V	
	DEC	r1	1	r1 ← r1 – 1	x	x	x	V	
		saddr	2	(saddr) ← (saddr) – 1	x	x	x	V	
	INCW	rp2	1	rp2 ← rp2 + 1					
		saddrp	3	(saddrp) ← (saddrp) + 1					
	DECW	rp2	1	rp2 ← rp2 – 1					
		saddrp	3	(saddrp) ← (saddrp) – 1					
Shift rotate	**ROR**	r1, n	2	(CY, $r1_7$ ← $r1_0$, $r1_{m-1}$ ← $r1_m$) × n times				P	x
	ROL	r1, n	2	(CY, $r1_0$ ← $r1_7$, $r1_{m+1}$ ← $r1_m$) × n times				P	x
	RORC	r1, n	2	(CY ← $r1_0$, $r1_7$ ← CY, $r1_{m-1}$ ← $r1_m$) × n times				P	x
	ROLC	r1, n	2	(CY ← $r1_7$, $r1_0$ ← CY, $r1_{m+1}$ ← $r1_m$) × n times				P	x
	SHR	r1, n	2	(CY ← $r1_0$, $r1_7$ ← 0, $r1_{m-1}$ ← $r1_m$) × n times	x	x	0	P	x
	SHL	r1, n	2	(CY ← $r1_7$, $r1_0$ ← 0, $r1_{m+1}$ ← $r1_m$) × n times	x	x	0	P	x
	SHRW	rp1, n	2	(CY ← $rp1_0$, $rp1_{15}$ ← 0, $rp1_{m-1}$ ← $rp1_m$) × n times	x	x	0	P	x
	SHLW	rp1, n	2	(CY ← $rp1_{15}$, $rp1_0$ ← 0, $rp1_{m+1}$ ← $rp1_m$) × n times	x	x	0	P	x
	ROR4	[rp1]	2	A_{3-0} ← $(rp1)_{3-0}$, $(rp1)_{7-4}$ ← A_{3-0}, $(rp1)_{3-0}$ ← $(rp1)_{7-4}$					
	ROL4	[rp1]	2	A_{3-0} ← $(rp1)_{7-4}$, $(rp1)_{3-0}$ ← A_{3-0}, $(rp1)_{7-4}$ ← $(rp1)_{3-0}$					
BCD adjustment	**ADJBA**		2	Decimal Adjust Accumelator	x	x	x	P	x
	ADJBS								
Data conversion	**CVTBW**		1	When A_7 = 0, X ← A, A ← 00H When A_7 = 1, X ← A, A ← FFH					

Remarks 1. n of the shift rotate instruction indicates the number of times the shift rotate instruction is executed.

2. The address of the table shift instruction ranges from FE00H to FEFFH.

Instructions	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
Bit manipulation	**MOV1**	CY, saddr.bit	3	CY ← (saddr.bit)					x
		CY, sfr.bit	3	CY ← sfr.bit					x
		CY, A.bit	2	CY ← A.bit					x
		CY, X.bit	2	CY ← X.bit					x
		CY, PSWH.bit	2	CY ← PSW_H.bit					x
		CY, PSWL.bit	2	CY ← PSW_L.bit					x
		saddr.bit, CY	3	(saddr.bit) ← CY					
		sfr.bit, CY	3	sfr.bit ← CY					
		A.bit, CY	2	A.bit ← CY					
		X.bit, CY	2	X.bit ← CY					
		PSWH.bit, CY	2	PSW_H.bit ← CY					
		PSWL.bit, CY	2	PSW_L.bit ← CY	x	x	x	x	
	AND1	CY, saddr.bit	3	CY ← CY ∧ (saddr.bit)					x
		CY, /saddr.bit	3	CY ← CY ∧ $\overline{\text{(saddr.bit)}}$					x
		CY, sfr.bit	3	CY ← CY ∧ sfr.bit					x
		CY, /sfr.bit	3	CY ← CY ∧ $\overline{\text{sfr.bit}}$					x
		CY, A.bit	2	CY ← CY ∧ A.bit					x
		CY, /A.bit	2	CY ← CY ∧ $\overline{\text{A.bit}}$					x
		CY, X.bit	2	CY ← CY ∧ X.bit					x
		CY, /X.bit	2	CY ← CY ∧ $\overline{\text{X.bit}}$					x
		CY, PSWH.bit	2	CY ← CY ∧ PSW_H.bit					x
		CY, /PSWH.bit	2	CY ← CY ∧ $\overline{PSW_H.bit}$					x
		CY, PSWL.bit	2	CY ← CY ∧ PSW_L.bit					x
		CY, /PSWL.bit	2	CY ← CY ∧ $\overline{PSW_L.bit}$					x
	OR1	CY, saddr.bit	3	CY ← CY ∨ (saddr.bit)					x
		CY, /saddr.bit	3	CY ← CY ∨ $\overline{\text{(saddr.bit)}}$					x
		CY, sfr.bit	3	CY ← CY ∨ sfr.bit					x
		CY, /sfr.bit	3	CY ← CY ∨ $\overline{\text{sfr.bit}}$					x
		CY, A.bit	2	CY ← CY ∨ A.bit					x
		CY, /A.bit	2	CY ← CY ∨ $\overline{\text{A.bit}}$					x
		CY, X.bit	2	CY ← CY ∨ X.bit					x
		CY, /X.bit	2	CY ← CY ∨ $\overline{\text{X.bit}}$					x
		CY, PSWH.bit	2	CY ← CY ∨ PSW_H.bit					x
		CY, /PSWH.bit	2	CY ← CY ∨ $\overline{PSW_H.bit}$					x
		CY, PSWL.bit	2	CY ← CY ∨ PSW_L.bit					x
		CY, /PSWL.bit	2	CY ← CY ∨ $\overline{PSW_L.bit}$					x

Instructions	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
Bit manipulation	**XOR1**	CY, saddr.bit	3	CY ← CY ⊻ (saddr.bit)					x
		CY, sfr.bit	3	CY ← CY ⊻ sfr.bit					x
		CY, A.bit	2	CY ← CY ⊻ A.bit					x
		CY, X.bit	2	CY ← CY ⊻ X.bit					x
		CY, PSWH.bit	2	CY ← CY ⊻ PSW_H.bit					x
		CY, PSWL.bit	2	CY ← CY ⊻ PSW_L.bit					x
	SET1	saddr.bit	2	(saddr.bit) ← 1					
		sfr.bit	3	sfr.bit ← 1					
		A.bit	2	A.bit ← 1					
		X.bit	2	X.bit ← 1					
		PSWH.bit	2	PSW_H.bit ← 1					
		PSWL.bit	2	PSW_L.bit ← 1	x	x	x	x	x
	CLR1	saddr.bit	2	(saddr.bit) ← 0					
		sfr.bit	3	sfr.bit ← 0					
		A.bit	2	A.bit ← 0					
		X.bit	2	X.bit ← 0					
		PSWH.bit	2	PSW_H.bit ← 0					
		PSWL.bit	2	PSW_L.bit ← 0	x	x	x	x	x
	NOT1	saddr.bit	3	(saddr.bit) ← $\overline{\text{(saddr.bit)}}$					
		sfr.bit	3	sfr.bit ← $\overline{\text{sfr.bit}}$					
		A.bit	2	A.bit ← $\overline{\text{A.bit}}$					
		X.bit	2	X.bit ← $\overline{\text{X.bit}}$					
		PSWH.bit	2	PSW_H.bit ← $\overline{PSW_H.bit}$					
		PSWL.bit	2	PSW_L.bit ← $\overline{PSW_L.bit}$	x	x	x	x	x
	SET1	CY	1	CY ← 1					1
	CLR1	CY	1	CY ← 0					0
	NOT1	CY	1	CY ← $\overline{CY}$					x

Instructions	Mnemonic	Operand	Byte	Operation	Flag				
					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					
		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					
	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					
		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					
	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					

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

Instructions	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
Special	**CHKL**	sfr	3	(pin level) ⊻ (signal level before output buffer)	x	x		P	
	CHKLA	sfr	3	A ← (pin level) ⊻ (signal level before output buffer)	x	x		P	
Unconditional branch	**BR**	!addr16	3	PC ← addr16					
		rp1	2	PC_H ← $rp1_H$, PC_L ← $rp1_L$					
		[rp1]	2	PC_H ← (rp1 + 1), PC_L ← (rp1)					
		$addr16	2	PC ← PC + 2 + jdisp8					
Conditional branch	**BC**	$addr16	2	PC ← PC + 2 + jdisp8 if CY = 1					
	BL								
	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 PSW_H.bit = 1					
		PSWL.bit, $addr16	3	PC ← PC + 3 + jdisp8 if PSW_L.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 PSW_H.bit = 0					
		PSWL.bit, $addr16	3	PC ← PC + 3 + jdisp8 if PSW_L.bit = 0					

Instructions	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
Conditional branch	**BTCLR**	saddr.bit, $addr16	4	PC ← PC + 4 + jdisp8 if (saddr.bit) = 1 then reset (saddr.bit)					
		sfr.bit, $addr16	4	PC ← PC + 4 + jdisp8 if sfr.bit = 1 then reset sfr.bit					
		A.bit, $addr16	3	PC ← PC + 3 + jdisp8 if A.bit = 1 then reset A.bit					
		X.bit, $addr16	3	PC ← PC + 3 + jdisp8 if X.bit = 1 then reset X.bit					
		PSWH.bit, $addr16	3	PC ← PC + 3 + jdisp8 if PSW_H.bit = 1 then reset PSW_H.bit					
		PSWL.bit, $addr16	3	PC ← PC + 3 + jdisp8 if PSW_L.bit = 1 then reset PSW_L.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 PSW_H.bit = 0 then set PSW_H.bit					
		PSWL.bit, $addr16	3	PC ← PC + 3 + jdisp8 if PSW_L.bit = 0 then set PSW_L.bit	x	x	x	x	x
	DBNZ	r2, $addr16	2	r2 ← r2 – 1, then PC ← PC + 2 + jdisp8 if 2 ≠ 0					
		saddr, $addr16	3	(saddr) ← (saddr) – 1, then PC ← PC + 3 + jdisp8 if (saddr) ≠ 0					
Context switching	**BRKCS**	RBn	2	PC_H ↔ R5, PC_L ↔ R4, R7 ← PSW_H, R6 ← PSW_L, ← RBS2 – 0 ← n, RSS ← 0, IE ← 0					
	RETCS	!addr16	3	PC_H ← R5, PC_L ← R4, R5, R4 ← addr16 PSW_H ← R7, PSW_L ← R6	R	R	R	R	R
	RETCSB	!addr16	4	PC_H ← R5, PC_L ← R4, R5, R4 ← addr16 PSW_H ← R7, PSW_L ← R6	R	R	R	R	R

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

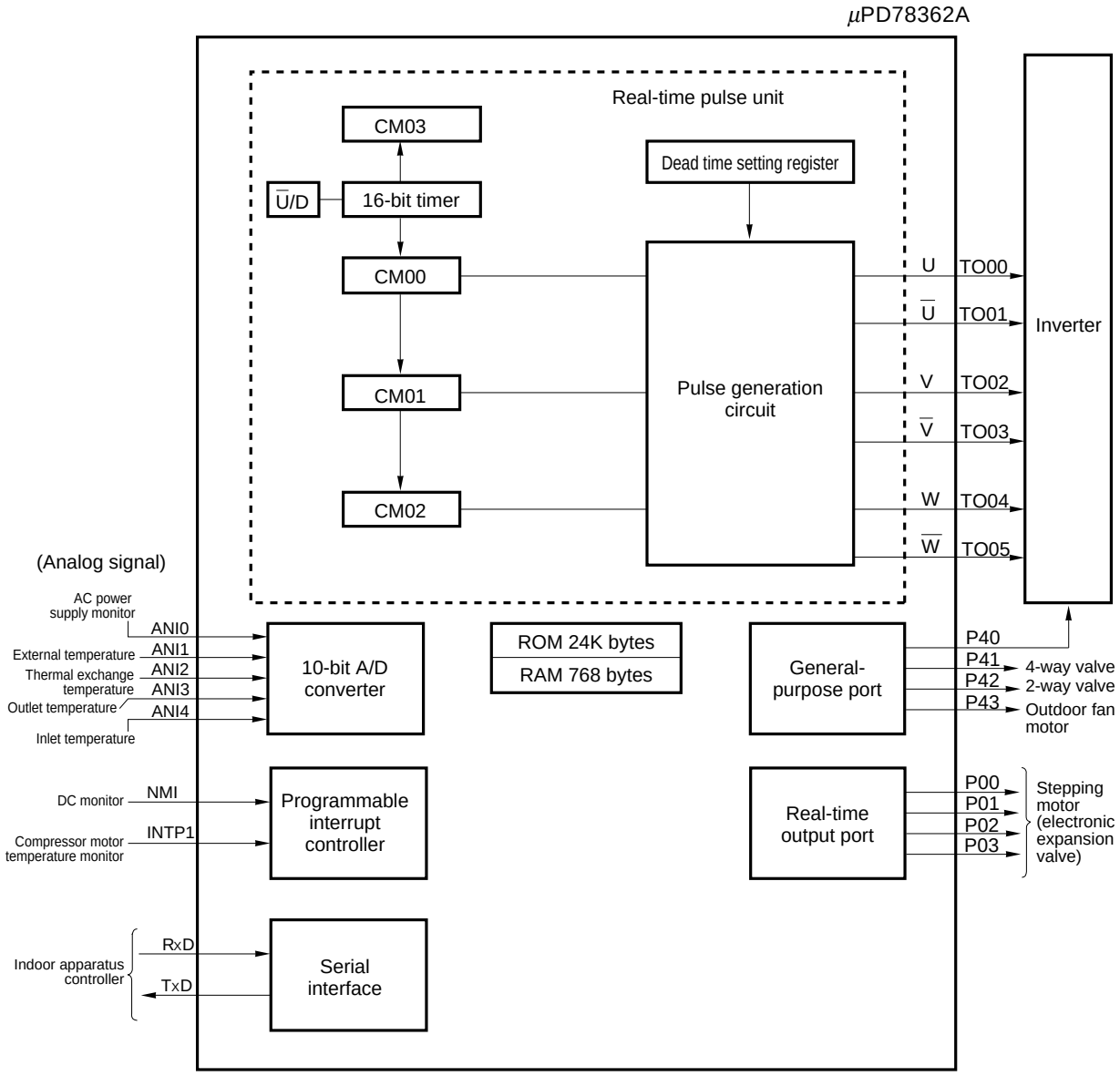
Instructions	Mnemonic	Operand	Byte	Operation	Flag				
					S	Z	AC	P/V	CY
String	**CMPMNC**	[DE +], A	2	(DE +) – A, C ← C – 1 End if C = 0 or CY = 1	x	x	x	V	x
		[DE –], A	2	(DE –) – A, C ← C – 1 End if C = 0 or CY = 1	x	x	x	V	x
	CMPBKNC	[DE +], [HL +]	2	(DE +) – (HL +), C ← C – 1 End if C = 0 or CY = 1	x	x	x	V	x
		[DE –], [HL –]	2	(DE –) – (HL –), C ← C – 1 End if C = 0 or CY = 1	x	x	x	V	x
CPU control	**MOV**	STBC, #byte	4	STBC ← byte**Note**					
		WDM, #byte	4	WDM ← byte**Note**					
	SWRS		1	RSS ← $\overline{\text{RSS}}$					
	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 Interruptt)					
	DI		1	IE ← 0 (Disable Interrupt)					

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

Operation on trap:
$(SP - 1) \leftarrow PSW_H$, $(SP - 2) \leftarrow PSW_L$,
$(SP - 3) \leftarrow (PC - 4)_H$, $(SP - 4) \leftarrow (PC - 4)_L$,
$PC_L \leftarrow (003CH)$, $PC_H \leftarrow (003DH)$,
$SP \leftarrow SP - 4$, $IE \leftarrow 0$

8. EXAMPLE OF SYSTEM CONFIGURATION

Controlling outdoor apparatus of inverter air conditioner



9. ELECTRICAL SPECIFICATIONS

Absolute Maximum Ratings (T_A = 25 °C)

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

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

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

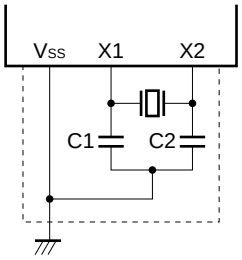
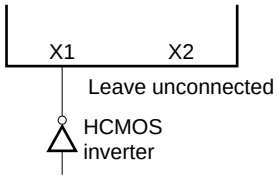
Recommended Operating Conditions

Oscillation frequency	T_A	V_{DD}
3 MHz ≤ f_{XX} ≤ 8 MHz	–40 to +85 °C	+5.0 V ± 10 %

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

Parameter	Symbol	Test conditions	MIN.	TYP.	MAX.	Unit
Input capacitance	C_I	f = 1 MHz			20	pF
Output capacitance	C_O	0 V except measured pins			20	pF
I/O capacitance	C_{IO}				20	pF

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

Resonator	Recommended circuit	Parameter	MIN.	MAX.	Unit
Ceramic resonator or crystal resonator		Oscillation frequency (f_{XX})	3	8	MHz
External clock		X1 input frequency (f_X)	3	8	MHz
		X1 rise/fall time (t_{XR}, t_{XF})	0	30	ns
		X1 input high-/low-level width (t_{WXH}, t_{WXL})	40	170	ns

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

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

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

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

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

2. $\overline{\text{RESET}}$, X1, X2, P20/NMI, P21/INTP0, P22/INTP1, P23/INTP2, P24/INTP3/TI, P25/INTP4, P32/SO/SB0, P33/SI/SB1 and P34/$\overline{\text{SCK}}$ pins.

3. Pins other than those specified in Notes 4 and 5.

4. P80/TO00-P85/TO05 pins (When I_{OL} = 15 mA is in operation, up to three pins can be ON simultaneously.)

5. P00/RTP0-P03/RTP3, P04/PWM0, P05/TCUD/PWM1, P06/TIUD/TO40 and P07/TCLRUD pins (When I_{OL} = 10 mA is in operation, up to four pins can be ON simultaneously.) .

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

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

System Clock Cycle

Parameter	Symbol	Test conditions	MIN.	MAX.	Unit
System clock cycle time	t_{CYK}		62.5	166.7	ns

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

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

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

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

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

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

Parameter	Symbol	Test conditions	MIN.	MAX.	Unit
NMI high-/low-level width	t_{WNIH}, t_{WNIL}		2		μs
$\overline{\text{RESET}}$ high-/low-level width	t_{WRSH}, t_{WRSL}		1.5		μs
INTP0 high-/low-level width	t_{WI0H}, t_{WI0L}	Ts = T	250		ns
		Ts = 4T	1.0		μs
		Ts = 8T	2.0		μs
		Ts = 16T	4.0		μs
INTP1 high-/low-level width	t_{WI1H}, t_{WI1L}	Ts = T	250		ns
		Ts = 4T	1.0		μs
		Ts = 8T	2.0		μs
		Ts = 16T	4.0		μs
INTP2 high-/low-level width	t_{WI2H}, t_{WI2L}	Ts = T	250		ns
		Ts = 4T	1.0		μs
INTP3(TI) high-/low-level width	t_{WI3H}, t_{WI3L}	Ts = T	250		ns
		Ts = 4T	1.0		μs
		Ts = 8T	2.0		μs
		Ts = 16T	4.0		μs
		Ts = 64T	16.0		μs
		Ts = 128T	32.0		μs
		Ts = 256T	64.0		μs
INTP4 high-/low-level width	t_{WI4H}, t_{WI4L}	Ts = T	250		ns
		Ts = 4T	1.0		μs
		Ts = 8T	2.0		μs
		Ts = 16T	4.0		μs

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

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

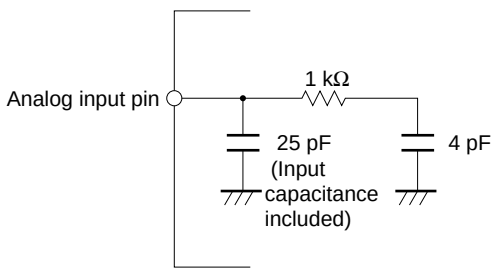
A/D Converter Characteristics (T_A = –40 to +85 °C, V_{DD} = +5 V ± 10 %, V_{SS} = AV_{SS} = 0 V, V_{DD} – 0.5 V ≤ AV_{DD} ≤ V_{DD})

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

Notes 1. The quantization error is excluded.

2. When –0.3 V ≤ V_{IAN} ≤ 0 V, the conversion result becomes 000H.
When 0 V < V_{IAN} < AV_{REF}, the conversion is performed with the 10-bit resolution.
When AV_{REF} ≤ V_{IAN} ≤ +0.3 V, the conversion result becomes 3FFH.

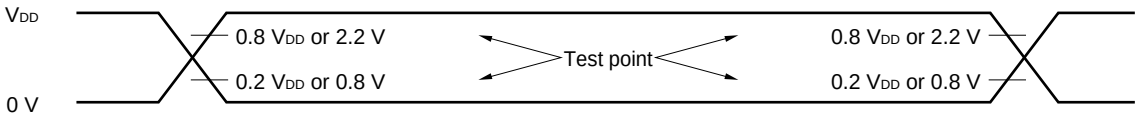
3. The analog input impedance at the time of sampling is the same as the equivalent circuit shown below. (The values in the diagram are TYP. values; they are not guaranteed values)



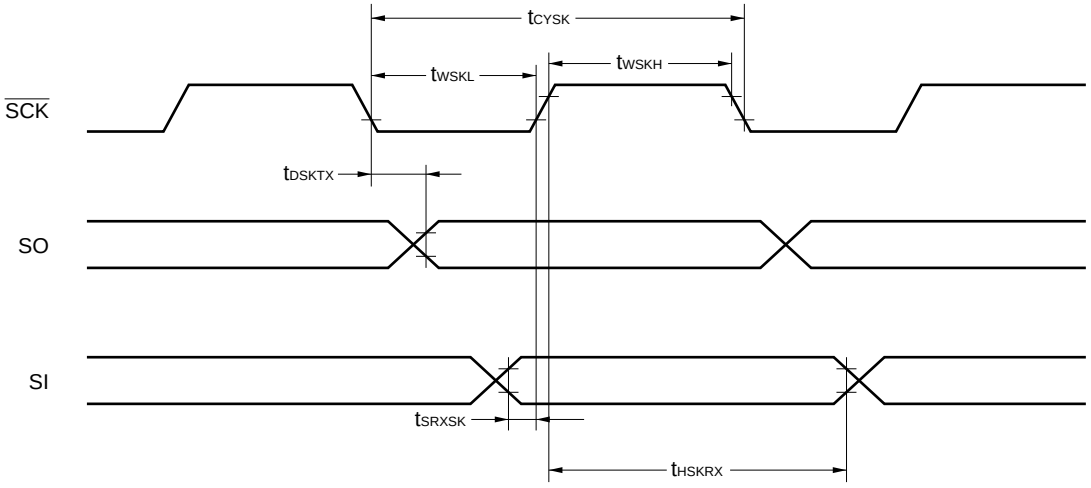
Cautions 1. When using the P70/ANI0-P77/ANI7 pins for both digital and analog inputs, the previously described characteristics are not guaranteed. Therefore, ensure that all of the eight P70/ANI0-P77/ANI7 pins are used either for analog input or digital input.

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

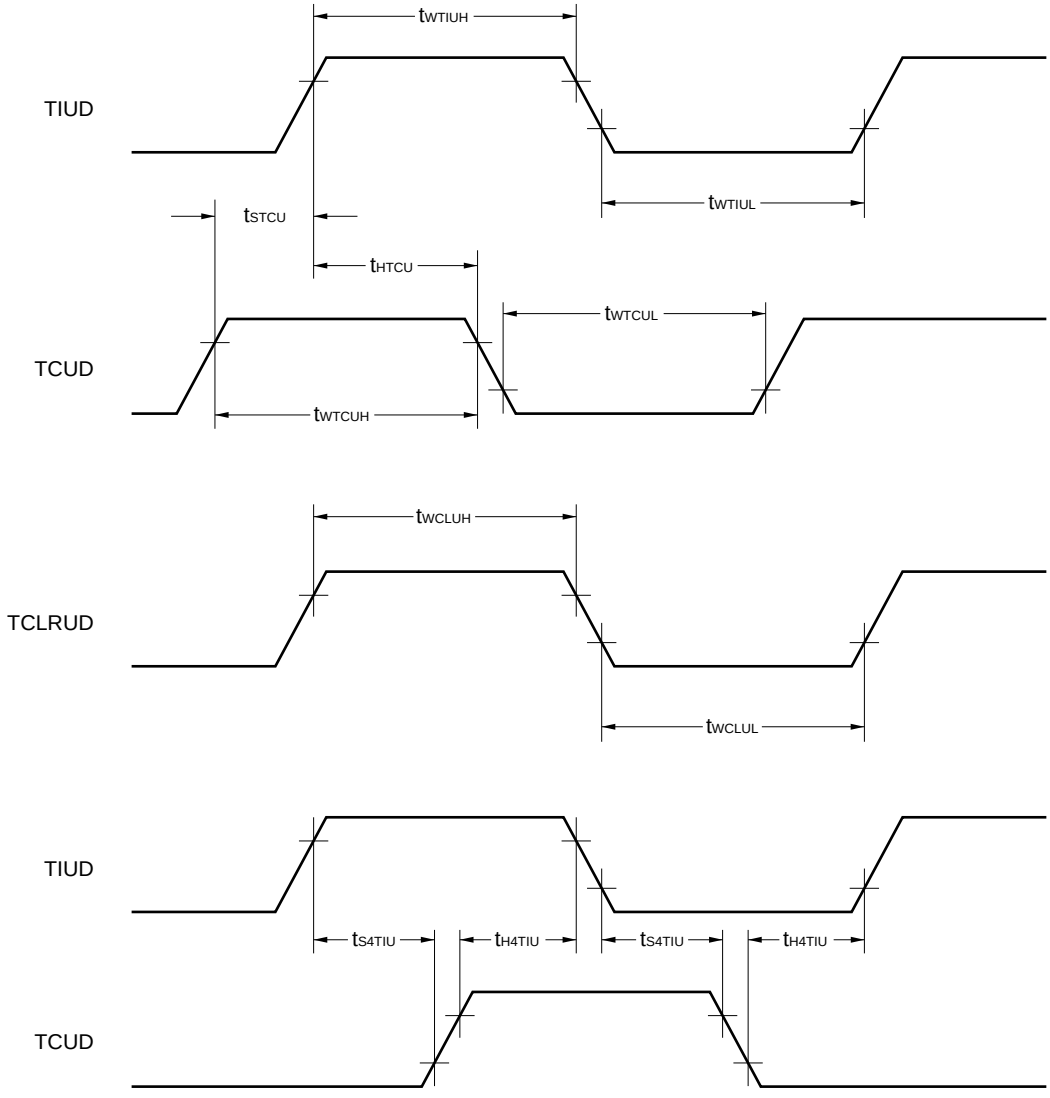
AC Timing Test Point



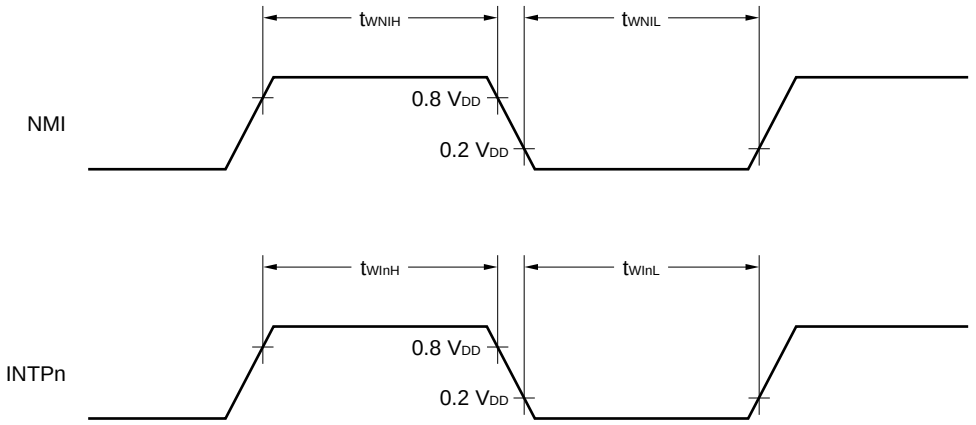
Serial Operation



Up/Down Counter (Timer 4) Input Timing

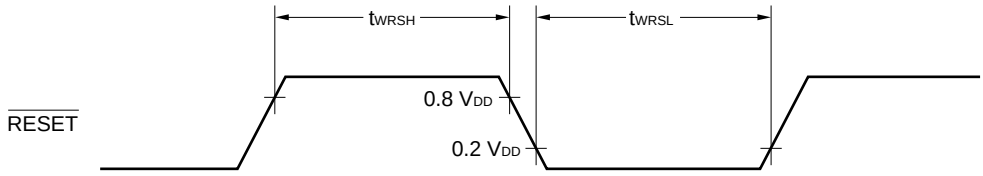


Interrupt Input Timing



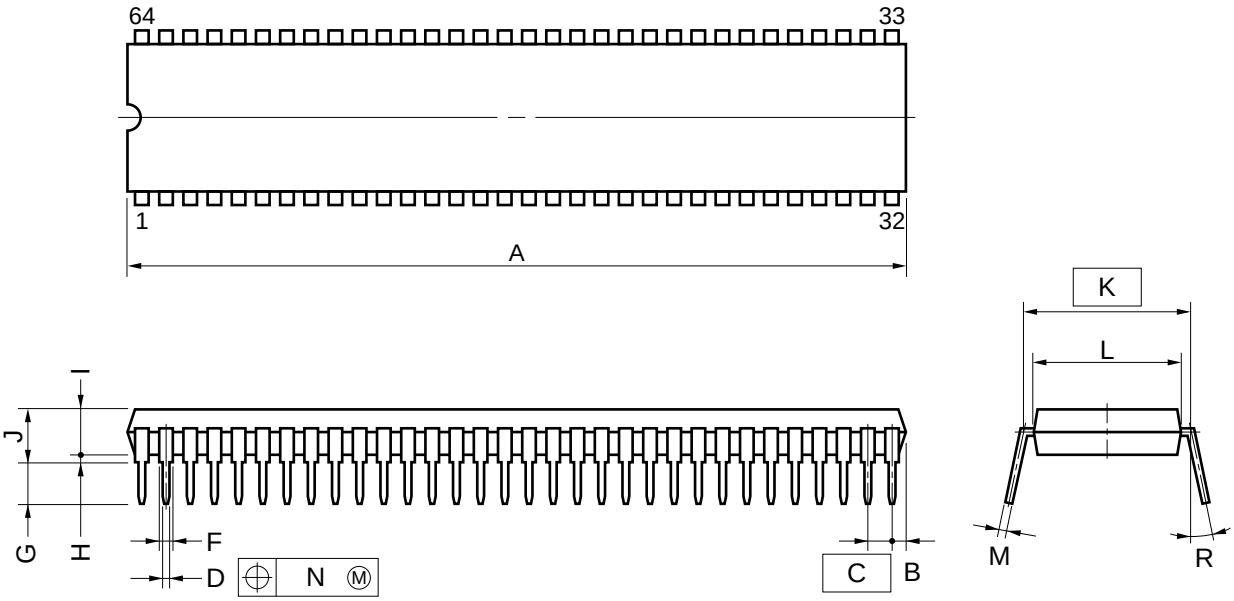
Remark n = 0 – 4

Reset Input Timing



10. PACKAGE DRAWING

64 PIN PLASTIC SHRINK DIP (750 mil)



NOTE

1) Each lead centerline is located within 0.17 mm (0.007 inch) of its true position (T.P.) at maximum material condition.

2) Item "K" to center of leads when formed parallel.

ITEM	MILLIMETERS	INCHES
A	58.68 MAX.	2.311 MAX.
B	1.78 MAX.	0.070 MAX.
C	1.778 (T.P.)	0.070 (T.P.)
D	0.50 ± 0.10	$0.020^{+0.004}_{-0.005}$
F	0.9 MIN.	0.035 MIN.
G	3.2±0.3	0.126±0.012
H	0.51 MIN.	0.020 MIN.
I	4.31 MAX.	0.170 MAX.
J	5.08 MAX.	0.200 MAX.
K	19.05 (T.P.)	0.750 (T.P.)
L	17.0	0.669
M	$0.25^{+0.10}_{-0.05}$	$0.010^{+0.004}_{-0.003}$
N	0.17	0.007
R	0~15°	0~15°

P64C-70-750A,C-1

11. RECOMMENDED SOLDERING CONDITIONS

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

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

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

Table 11-1. Insertion Type Soldering Conditions

★ **μPD78361ACW-×××: 64-pin plastic shrink DIP (750 mil)**
μPD78362ACW-×××: 64-pin plastic shrink-DIP (750 mil)

Soldering method	Soldering conditions
Wave soldering (pin only)	Solder bath temperature: 260 °C or less, Time: 10 sec. max.,
Partial heating	Pin temperature: 300 °C or less, Time: 3 sec. max. (per pin)

Caution Wave soldering is only for the pins in order that jet solder cannot contact with the chip directly.

APPENDIX A. DIFFERENCES BETWEEN μPD78362A AND μPD78328

Item \ Product name		μPD78362A	μPD78328
Minimum instruction execution time		125 ns [internal clock : 16 MHz, external clock : 8 MHz]	250 ns [internal clock : 8 MHz, external clock : 16 MHz]
Internal memory	ROM	24K bytes	16K bytes
	RAM	768 bytes	512 bytes
General-purpose registers		8 bits × 16 × 8 banks	
Number of basic instructions		115	111
Instruction set		• 16-bit transfer/operation • Multiplication/division (16 bits x 16 bits, 32 bits ÷ 16 bits) • Bit manipulation • String	
		• Sum-of-products operation (16 bits × 16 bits + 32 bits) • Relative operation	—
I/O lines	Input	14 (of which 8 are multiplexed with analog input)	11 (of which 8 are multiplexed with analog input)
	I/O	38	41
Real-time pulse unit		• 16-bit timer × 5 • 16-bit compare register × 7 • 16-bit capture register × 3 • 16-bit capture/compare register × 2 • Two output modes selectable Mode 0, set-reset output : 6 channels Mode 1, buffer output : 6 channels • 16-bit resolution PWM output: 1 channel	• 16-bit timer × 3 • 16-bit compare register × 14 • 16-bit capture/compare register × 1 • Two output modes selectable Mode 0, set-reset output : 6 channels toggle output : 1 channel Mode 1, buffer output : 8 channels
Real-time output port		4 (buffer output in 4-bit units)	4/8 (buffer output in 4-/8-bit units)
PWM unit		8-/9-/10-/12-bit resolution variable PWM output: 2 channels	8-bit resolution PWM output: 1 channel
A/D converter		10-bit resolution, 8 channels	
Serial interface		Dedicated baud rate generator UART : 1 channel Clocked serial interface/SBI : 1 channel	
Interrupt function		• External: 6, internal: 14 (2 multiplexed with external) • 4 programmable priority levels	• External: 4, internal: 17 • 3 programmable priority levels
		• Three service selectable (vectored interrupt/macro service/context switching)	
Test source		None	Internal: 1
External expansion function		None	Provided
PLL control circuit		Provided (external 8 MHz → internal: 16 MHz)	None
Package		• 64-pin plastic shrink DIP (750 mil)	• 64-pin plastic shrink DIP • 64-pin plastic QFP (14 × 20 mm)
Others		• Watchdog timer • Standby functions (HALT mode, STOP mode)	

APPENDIX B. TOOLS

B.1 DEVELOPMENT TOOLS

The following development tools are available to support development of μPD78362A program:

LANGUAGE PROCESSOR

<table>
<tr><td>78K/III series
relocatable assembler
(RA78K3)</td><td colspan="4">A relocatable assembler, that can be used commonly for the 78K/III series products. Since this assembler is provided with macro functions, it enhances the developmnt efficency. A structured assembler, that can explicitly describe the program control structure, is also supplied, so that the program productivity and maintainability can be improved.

<table>
<tr><th rowspan="2">Host machine</th><th colspan="2" rowspan="2"></th><th rowspan="2">Order code (product name)</th></tr>
<tr><th>OS</th><th>Supply media</th></tr>
<tr><td rowspan="2">PC-9800 series</td><td rowspan="2">MS-DOS™</td><td>3.5" 2HD</td><td>μS5A13RA78K3</td></tr>
<tr><td>5" 2HD</td><td>μS5A10RA78K3</td></tr>
<tr><td rowspan="2">IBM PC/AT™ and its
compatible model</td><td rowspan="2">PC DOS™</td><td>3.5" 2HC</td><td>μS7B13RA78K3</td></tr>
<tr><td>5" 2HC</td><td>μS7B10RA78K3</td></tr>
<tr><td>HP9000 series 700™</td><td>HP-UX™</td><td>DAT</td><td>μS3P16RA78K3</td></tr>
<tr><td>SPARCstation™</td><td>SunOS™</td><td rowspan="2">Cartridge tape
(QIC-24)</td><td>μS3K15RA78K3</td></tr>
<tr><td>NEWS™</td><td>NEWS-OS™</td><td>μS3R15RA78K3</td></tr>
</table></td></tr>
<tr><td>78K/III series
C compiler
(CC78K3)</td><td colspan="4">This is a C compiler that can be commonly used for 78K/III series.
This program converts the program written in C language to object codes microcontroller can execute. When using this compiler, the 78K3 series relocatable assembler (RA78K/III) is necessary.

<table>
<tr><th rowspan="2">Host machine</th><th colspan="2" rowspan="2"></th><th rowspan="2">Order code (product name)</th></tr>
<tr><th>OS</th><th>Supply media</th></tr>
<tr><td rowspan="2">PC-9800 series</td><td rowspan="2">MS-DOS</td><td>3.5" 2HD</td><td>μS5A13CC78K3</td></tr>
<tr><td>5" 2HD</td><td>μS5A10CC78K3</td></tr>
<tr><td rowspan="2">IBM PC/AT and its
compatible model</td><td rowspan="2">PC DOS</td><td>3.5" 2HC</td><td>μS7B13CC78K3</td></tr>
<tr><td>5" 2HC</td><td>μS7B10CC78K3</td></tr>
<tr><td>HP9000 series 700</td><td>HP-UX</td><td>DAT</td><td>μS3P16CC78K3</td></tr>
<tr><td>SPARC station</td><td>SunOS</td><td rowspan="2">Cartridge tape
(QIC-24)</td><td>μS3K15CC78K3</td></tr>
<tr><td>NEWS</td><td>NEWS-OS</td><td>μS3R15CC78K3</td></tr>
</table></td></tr>
</table>

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

PROM WRITING TOOLS

Hardware	PG-1500	This is a PROM programmer that can program PROM-contained single-chip microcontrollers in standalone mode or under control of a host machine when the accessory board and an optional programmer adapter are connected. It can also program representative PROMs from 256K-bit to 4M-bit models.			
	PA-78P364CW	PROM programmer adapters that writes a program to the μPD78P364A on a general-purpose PROM programmer such as the PG-1500. PA-78P364CW : for μPD78P364ACW			
Software	PG-1500 controller	Connects the PG-1500 and a host machine with a serial intrface and a parallel interface to control the PG-1500 from the host machine.			
		Host machine	OS	Supply media	Order code (part number)
		PC-9800 series	MS-DOS	3.5" 2HD	μS5A13PG1500
				5" 2HD	μS5A10PG1500
		IBM PC/AT and compatible machines	PC DOS	3.5" 2HD	μS7B13PG1500
				5" 2HC	μS7B10PG1500

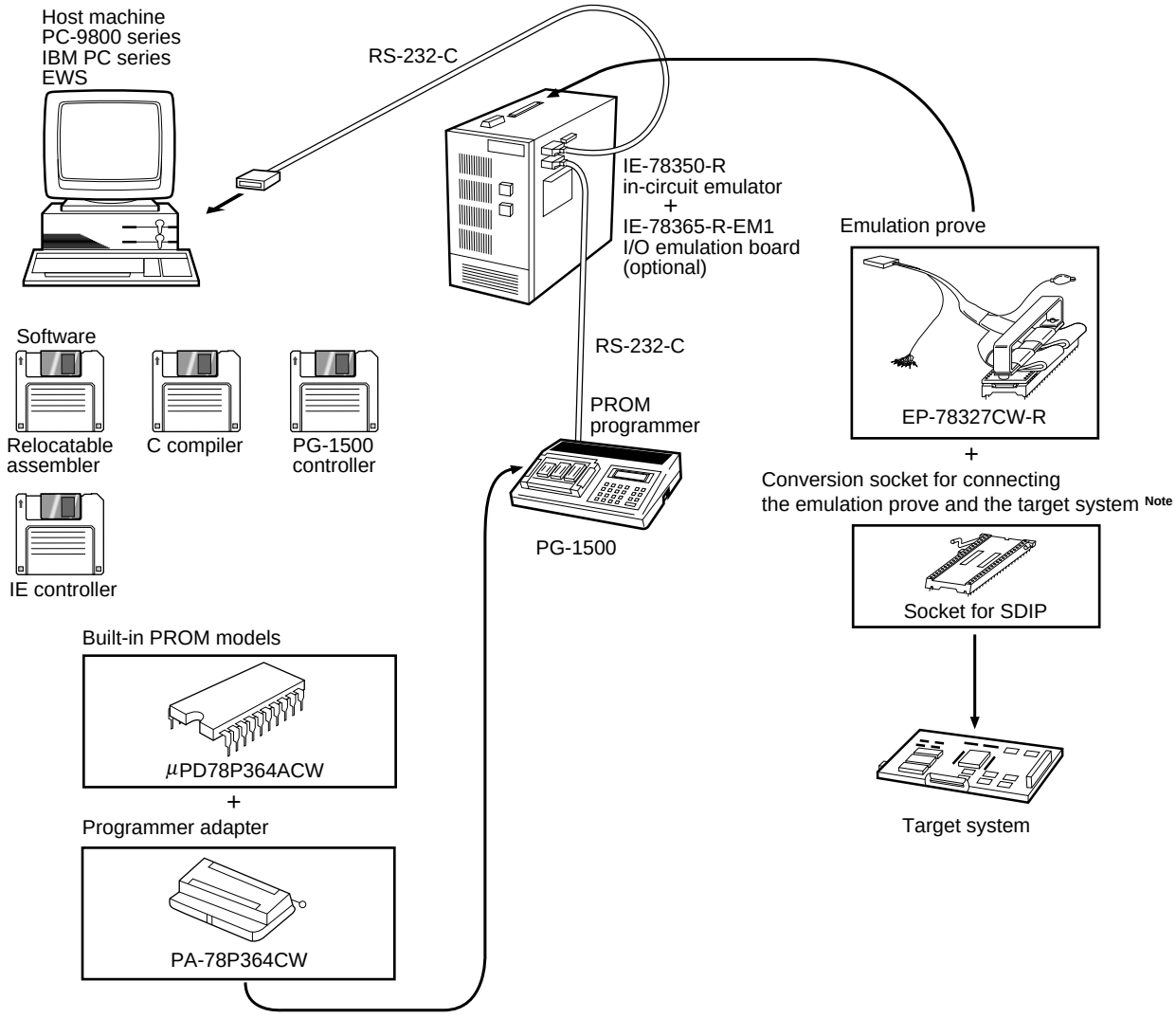
Remark The operation of the PG-1500 controller is guaranteed only on the above host machine and OS.

DEBUGGING TOOLS (WHEN IE CONTROLLER IS USED)

Hardware	IE-78350-R	In-circuit emulator that can be used to develop and debug application systems. Connected to a host machine for debugging.			
	IE-78365-R-EM1	I/O emulation board that emulates the peripheral functions of the target device such as I/O ports.			
	EP-78327CW-R	Emulation prove that connects the IE-78350-R to the target system.			
Software	IE-78350-R control program (IE controller)	Program that controls the IE-78350-R on the host machine. It can automatically execute commands, enhancing debugging efficiency.			
		Host machine	OS	Supply media	Order code (part number)
		PC-9800 series	MS-DOS	3.5" 2HD	μS5A13IE78365A
				5" 2HD	μS5A10IE78365A
		IBM PC/AT and compatible machines	PC DOS	3.5" 2HC	μS7B13IE78365A
				5" 2HC	μS7B10IE78365A

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

DEVELOPMENT TOOL CONFIGURATION (WHEN USING IE CONTROLLER)



Note Use the socket available on the market.

Remarks 1. Host machine and PG-1500 can be directly connected by RS-232-C.

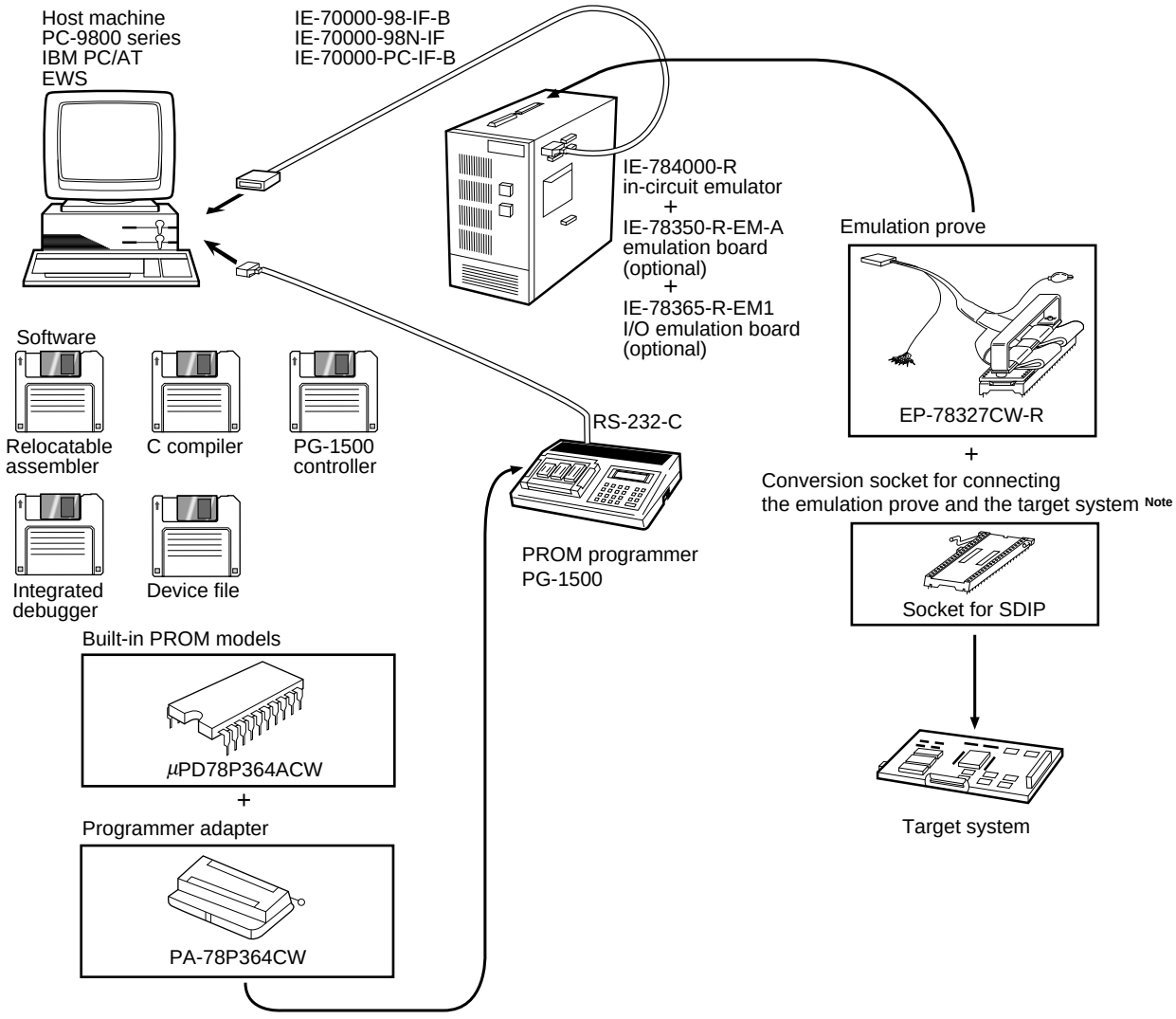
2. 3.5-inch FD represents the supply media of software in this figure.

DEBUGGING TOOLS (WHEN INTEGRATED DEBUGGER IS USED)

Hardware	IE-784000-R	In-circuit emulation that can be used to develop and debug the application system. Connected to a host machine for debugging.			
	IE-78350-R-EM-A	Emulation board that emulates the peripheral functions of the target device such as I/O ports.			
	IE-78365-R-EM1	I/O emulation board that emulates the peripheral functions of the target device such as I/O ports.			
	EP-78327CW-R	Emulation probe connecting the IE-784000-R to the target system.			
	IE-70000-98-IF-B	Interface adapter to connect PC-9800 series (except notebook type personal computer) as the host machine.			
	IE-70000-98N-IF	Interface adapter and cable to connect PC-9800 series notebook type personal computer as the host machine.			
	IE-70000-PC-IF-B	Interface adapter to connect IBM PC as the host machine.			
	IE-78000-R-SV3	Interface adapter and cable to connect EWS as the host machine.			
Software	Integrated debugger (ID78K3)	Program controlling the in-circuit emulator for the 78K/III series. Used in combination with a device file (DF78365). Can debug a program coded in the C language, structured assembly language, or assembly language at source program level. Can also split the screen of the host machine into windows on each of which information is displayed, enhancing debugging efficiency.			
		Host machine			Order code (part number)
			OS	Supply media	
		PC-9800 series	MS-DOS + Windows™	3.5" 2HD	μSAA13ID78K3
				5" 2HD	μSAA10ID78K3
		IBM PC/AT and compatible machines (Japanese Windows)	PC DOS + Windows	3.5" 2HC	μSAB13ID78K3
				5" 2HC	μSAB10ID78K3
		IBM PC/AT and compatible machines (English Windows)		3.5" 2HC	μSBB13ID78K3
				5" 2HC	μSBB10ID78K3
	Device File (DF78365)	File containing information peculiar to device. Use in combination with an assembler (RA78K3), C compiler (CC78K3), and integrated debugger (ID78K3).			
		Host machine			Order code (part number)
			OS	Supply media	
		PC-9800 series	MS-DOS	3.5" 2HD	μS5A13DF78365
				5" 2HD	μS5A10DF78365
		IBM PC/AT and compatible machines	PC DOS	3.5" 2HC	μS7B13DF78365
				5" 2HC	μS7B10DF78365

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

DEVELOPMENT TOOL CONFIGURATION (WHEN USING INTEGRATED DEBUGGER)



Note Use the socket available on the market.

Remarks 1. Desk top-type PC represents host machine in this figure.

2. 3.5-inch FD represents the supply media of software in this figure.

B.2 EMBEDDED SOFTWARE

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

REAL-TIME OS

<table>
<tr><td rowspan="6">Real-time OS
(RX78K/III) Note</td><td colspan="4">RX78K/III is intended to implement a multi-tasking environment for use in the control field where real-time capability is a must. It can allocate the idle time of the CPU to other processing to improve the overall performance of the system.
RX78K/III provides system calls conforming to the μITRON specification.
The RX78K/III package supplies a tool (configurator) to create the nucleus of RX78K/III and multiple information tables.</td></tr>
<tr><td colspan="3">Host machine</td><td>Order code (part number)</td></tr>
<tr><td></td><td>OS</td><td>Supply media</td><td></td></tr>
<tr><td>PC-9800 series</td><td>MS-DOS</td><td>3.5" 2HD</td><td>Pending</td></tr>
<tr><td></td><td></td><td>5" 2HD</td><td>Pending</td></tr>
<tr><td>IBM PC/AT and compatible machines</td><td>PC DOS</td><td>3.5" 2HC</td><td>Pending</td></tr>
<tr><td></td><td></td><td>5" 2HC</td><td>Pending</td></tr>
</table>

Note Under development

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

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

FUZZY INFERENCE DEVELOPMENT SUPPORT SYSTEM

Fuzzy knowledge data creation tool (FE9000, FE9200)	Program that supports input/editing and evaluation (simulation) of fuzzy knowledge (fuzzy rules and membership functions).			
	Host machine			Order code (part number)
		OS	Supply media	
	PC-9800 series	MS-DOS	3.5" 2HD	μS5A13FE9000
			5" 2HD	μS5A10FE9000
	IBM PC/AT and compatible machines	PC DOS + Windows	3.5" 2HC	μS7B13FE9200
			5" 2HC	μS7B10FE9200
Translator (FT78K3) **Note**	Program that converts the fuzzy knowledge data obtained by using the fuzzy knowledge data creation tool into assembler source program for the RA78K3.			
	Host machine			Order code (part number)
		OS	Supply media	
	PC-9800 series	MS-DOS	3.5" 2HD	μS5A13FT78K3
			5" 2HD	μS5A10FT78K3
	IBM PC/AT and compatible machines	PC DOS	3.5" 2HC	μS7B13FT78K3
			5" 2HC	μS7B10FT78K3
Fuzzy inference module (FI78K3) **Note**	Program that executes fuzzy inference when linked with the fuzzy knowledge data converted by the translator.			
	Host machine			Order code (part number)
		OS	Supply media	
	PC-9800 series	MS-DOS	3.5" 2HD	μS5A13FI78K3
			5" 2HD	μS5A10FI78K3
	IBM PC/AT and compatible machines	PC DOS	3.5" 2HC	μS7B13FI78K3
			5" 2HC	μS7B10FI78K3
Fuzzy inference debugger (FD78K3)	Support software that evaluates and adjusts the fuzzy knowledge data at the hardware level by using an in-circuit emulator.			
	Host machine			Order code (part number)
		OS	Supply media	
	PC-9800 series	MS-DOS	3.5" 2HD	μS5A13FD78K3
			5" 2HD	μS5A10FD78K3
	IBM PC/AT and compatible machines	PC DOS	3.5" 2HC	μS7B13FD78K3
			5" 2HC	μS7B10FD78K3

Note Under development

[MEMO]

NOTES FOR CMOS DEVICES

① PRECAUTION AGAINST ESD FOR SEMICONDUCTORS

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

② HANDLING OF UNUSED INPUT PINS FOR CMOS

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

③ STATUS BEFORE INITIALIZATION OF MOS DEVICES

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

Regional Information

Some information contained in this document may vary from country to country. Before using any NEC product in your application, please contact the NEC office in your country to obtain a list of authorized representatives and distributors. They will verify:

- Device availability
- Ordering information
- Product release schedule
- Availability of related technical literature
- Development environment specifications (for example, specifications for third-party tools and components, host computers, power plugs, AC supply voltages, and so forth)
- Network requirements

In addition, trademarks, registered trademarks, export restrictions, and other legal issues may also vary from country to country.

NEC Electronics Inc. (U.S.)
Santa Clara, California
Tel: 800-366-9782
Fax: 800-729-9288

NEC Electronics (Germany) GmbH
Duesseldorf, Germany
Tel: 0211-65 03 02
Fax: 0211-65 03 490

NEC Electronics (UK) Ltd.
Milton Keynes, UK
Tel: 01908-691-133
Fax: 01908-670-290

NEC Electronics Italiana s.r.1.
Milano, Italy
Tel: 02-66 75 41
Fax: 02-66 75 42 99

NEC Electronics (Germany) GmbH
Benelux Office
Eindhoven, The Netherlands
Tel: 040-2445845
Fax: 040-2444580

NEC Electronics (France) S.A.
Velizy-Villacoublay, France
Tel: 01-30-67 58 00
Fax: 01-30-67 58 99

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

NEC Electronics (Germany) GmbH
Scandinavia Office
Taeby, Sweden
Tel: 08-63 80 820
Fax: 08-63 80 388

NEC Electronics Hong Kong Ltd.
Hong Kong
Tel: 2886-9318
Fax: 2886-9022/9044

NEC Electronics Hong Kong Ltd.
Seoul Branch
Seoul, Korea
Tel: 02-528-0303
Fax: 02-528-4411

NEC Electronics Singapore Pte. Ltd.
United Square, Singapore 1130
Tel: 253-8311
Fax: 250-3583

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

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

J96. 8

MS-DOS and windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.
PC/AT and PC DOS are trademarks of IBM Corporation.
HP9000 series 700 and HP-UX are trademarks of Hewlett-Packard Company.
SPARCstation is a trademark of SPARC International, Inc.
SunOS is a trademark of Sun Microsystems, Inc.
NEWS and NEWS-OS are trademark of Sony Corporation.
TRON is an abbreviation of The Realtime Operating system Nucleus.
ITRON is an abbreviation of Industrial TRON.

The export of this product from Japan is regulated by the Japanese government. To export this product may be prohibited without governmental license, the need for which must be judged by the customer. The export or re-export of this product from a country other than Japan may also be prohibited without a license from that country. Please call an NEC sales representative.

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of NEC Corporation. NEC Corporation assumes no responsibility for any errors which may appear in this document.
NEC Corporation does not assume any liability for infringement of patents, copyrights or other intellectual property rights of third parties by or arising from use of a device described herein or any other liability arising from use of such device. No license, either express, implied or otherwise, is granted under any patents, copyrights or other intellectual property rights of NEC Corporation or others.
While NEC Corporation has been making continuous effort to enhance the reliability of its semiconductor devices, the possibility of defects cannot be eliminated entirely. To minimize risks of damage or injury to persons or property arising from a defect in an NEC semiconductor device, customers must incorporate sufficient safety measures in its design, such as redundancy, fire-containment, and anti-failure features.
NEC devices are classified into the following three quality grades:
"Standard", "Special", and "Specific". The Specific quality grade applies only to devices developed based on a customer designated "quality assurance program" for a specific application. The recommended applications of a device depend on its quality grade, as indicated below. Customers must check the quality grade of each device before using it in a particular application.

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

The quality grade of NEC devices is "Standard" unless otherwise specified in NEC's Data Sheets or Data Books. If customers intend to use NEC devices for applications other than those specified for Standard quality grade, they should contact an NEC sales representative in advance.
Anti-radioactive design is not implemented in this product.

M4 96.5