μPD78310/312CW/G
8-BIT CMOS
MICROCOMPUTER

PRODUCT DESCRIPTION

µPD78310/312CW/G

8-BIT CMOS MICROCOMPUTER

PRODUCT DESCRIPTION

Table of contents

CHAPTER 4 INTERRUPT REQUEST SERVICE FUNCTIONS

CHAPTER 5 STANDBY FUNCTIONS

CHAPTER 6 LOCAL BUS INTERFACE FUNCTIONS

CHAPTER 7 OPERATIONS AFTER RESET

CHAPTER 8 INSTRUCTION SET

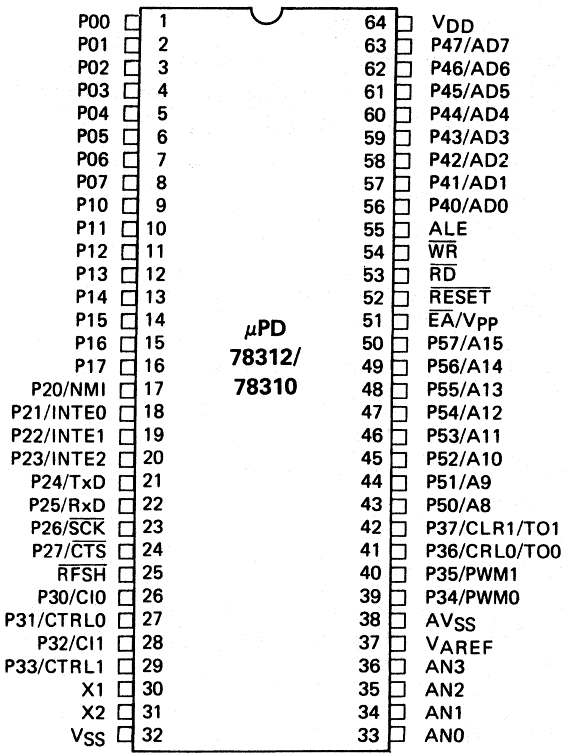
INTRODUCTION

The μPD78312 is a CMOS single-chip 8-bit microcomputer, incorporating a 16-bit CPU, ROM, RAM, A/D converter, general-purpose serial communication interface, and multipurpose pulse input/output units, which perform real-time digital signal control. In addition, the μPD78312 can directly control external memory in either ROM or RAM form. The μPD78310 is a version of the μPD78312, from which the ROM is eliminated, and it can directly access external memory up to 64K bytes.

Features

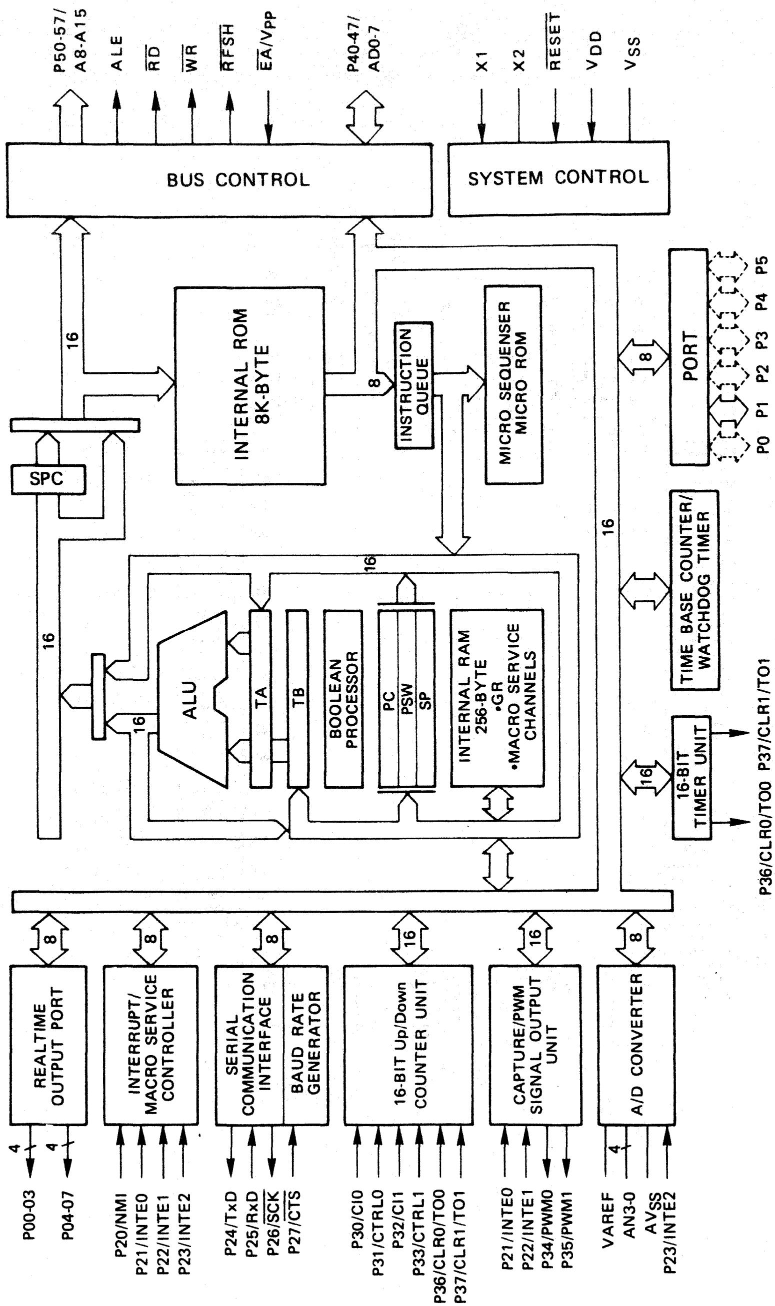
- Single-chip microcomputer (μCOM-78K)
- 96 instructions with abundant addressing modes
  - 16-bit operation instructions, bit manipulation instructions, multiplication/division instructions, string instructions, user stack manipulation instructions
- Instruction cycle: 500ns/12MHz
- Internal ROM: 8,192W x 8 bits (μPD78312)
- Internal RAM: 256 x 8 bits
- Capable of directly addressing external (ROM/RAM) up to 64K bytes
- Memory mapping of on-chip peripheral hardware (special-purpose register)
- Multipurpose pulse input/output unit
  - Two 16-bit presettable up/down counters
  - Two 16-bit interval timers
  - Two 16-bit capture registers
  - Two high-accuracy PWM outputs
  - Two real-time output port channels
- Four channel high-precision 8-bit A/D converter
- I/O ports
  - 8 input lines
  - 40 I/O lines (μPD78312)
  - 28 I/O lines (μPD78310) (for 4K-byte access)

- General-purpose serial communication interface (with dedicated baud rate generator)
  - Asynchronous mode, I/O interface mode
- Interrupt request controller
  - Context switching function levels
  - Macro service function
- Pseudo static memory refresh pulse output function
- Watchdog timer, time base counter
- Standby function (STOP/HALT)
- CMOS
- Single power supply
- 64-pin plastic shrink DIP (750mil) (μPD78312CW/μPD78310CW)
  64-pin plastic QUIP/flat package (μPD78312G/μPD78310G)

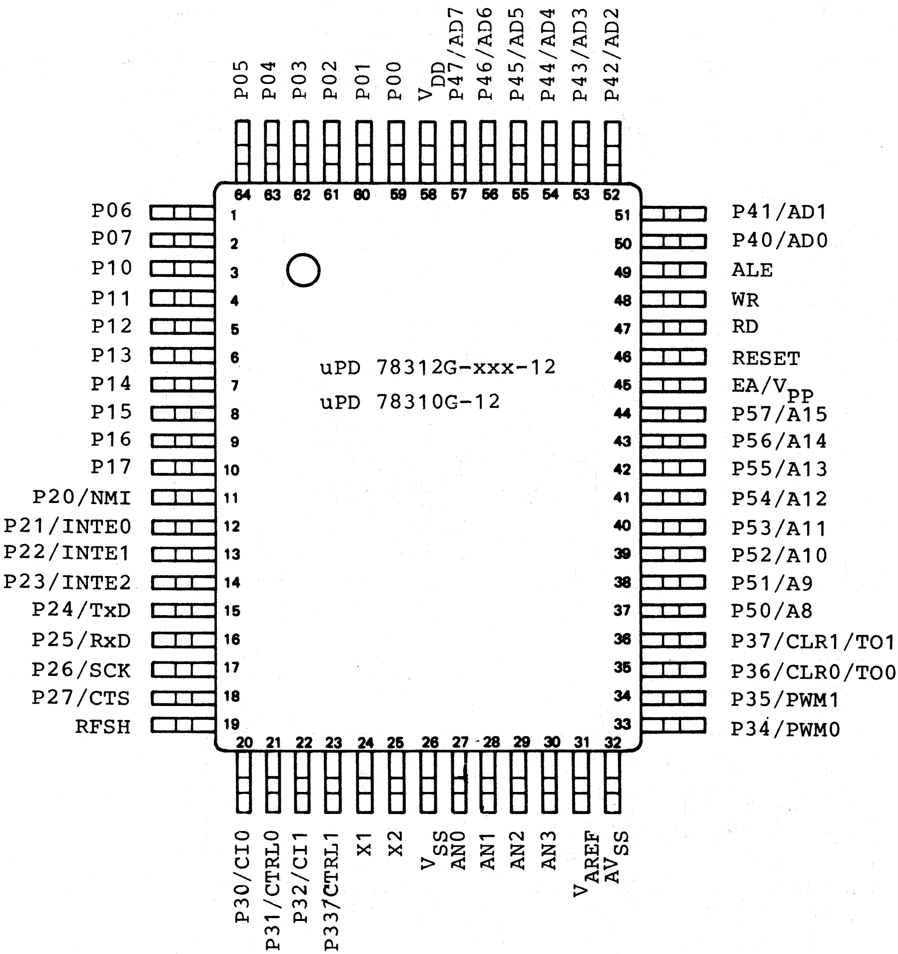


Pin Connection (Top View)

µPD78312 BLOCK DIAGRAM



PINNING OUTLINE (FLAT PACKAGE)



CHAPTER 1 PIN FUNCTIONS

1.1 Port Pins

Pin name	Input/Output	Function	
P00-07	Input/output/ real-time output	(port 0) These pins function as an 8-bit port that can be set for input or output per bit.	This port functions as a real-time output port which outputs the contents of the buffer register with a programmed interval.
P10-17	Input/output	(port 1) These pins function as an 8-bit general-purpose I/O port that can be set for input or output per bit.	
P20/ NMI	Input/Input	(port 2) These pins function as an 8-bit port that can be set for input or output per bit (pins 20 to 23 are only for input).	These is a nonmaskable interrupt request input pin of which either the rising or falling edge can be specified as the effective edge by the mode register.
P21/ INTE0	Input/Input		These are external interrupt request pins whose effective edge can be specified by the mode register
P22/ INTE1	Input/Input		
P23/ INTE2	Input/Input		
P24/ TxD	Input/output/ Output		This is a serial data output pin.
P25/ RxD	Input/output/ Input		This is a serial data input pin.
P26/ $\overline{SCK}$	Input/output/ Output		This is a serial clock output pin.

Pin name	Input/Output	Function	
P27/ $\overline{\text{CTS}}$	Input/output/ Input/output		In the asynchronous mode, this pin becomes an input pin of transmit enable. In the I/O interface mode, this pin becomes a serial clock I/O pin.
P30/ CI0	Input/Input	(port 3) These pins function as an 8-bit port that can be set for input and output per bit (pins 39 to 33 can be specified only for input).	This is an external count clock input pin to the count unit.
P31/ CTRL0	Input/Input		This is a count function up/down selection or control input pin to be specified per bit count unit.
P32/ CI1	Input/Input		This is an external count clock input pin to the count unit.
P33/ CTRL1	Input/input		This is a count function (up/down switching) or control input pin to the count unit.
P34/ PWM0	Input/output/ Output		These are PWM output pins.
P35/ PWM1	Input/output/ Output		
P36/ TO1/ CLR1	Input/output/ Output/Input		These are pulse output pins from the timer unit. These are up/down counter clear input pins.
P37/ TO1/ CLR0	Input/output/ output/input		

Pin name	Input/Output	Function	
P40-47/ AD0-7	Input/output/ Input/output	(port 4) These pins function as an 8-bit I/O port that can be set per byte.	This port functions as a multiplexed address/data bus when the external memory is connected.
P50-57/ A8-15	Input/output/ Input/output	(port 5) These pins function as an 8-bit I/O port that can be set per bit.	This port functions as an address bus when the external memory is connected.

1.2 Other Pins

Pin name	Input/Output	Function
$\overline{WR}$	Output	This pin outputs the strobe signal for external memory write operation.
$\overline{RD}$	Output	This pin outputs the strobe signal for external memory read operation.
ALE	Output	This pin outputs the timing signal for latching the address data output to port 4 when accessing the external memory.
$\overline{EA}$	Input	For the μPD78312, the EA pin is normally set to 1 (high level). The external memory can be accessed in the ROM-less mode by setting the EA pin to 0 (low level).
AN0-3	Input	These pins are analog input pins to the A/D converter.
AV_{REF}	Input	This pin is the A/D converter reference voltage input pin.
AV_{SS}		This pin is the A/D converter ground pin.
X1, X2		A crystal for the system clock is connected across these pins. When the external clock is used, the X1 pin inputs the external clock signal.

Pin name	Input/Output	Function
$\overline{RFSH}$	Output	When an external pseudo static memory is connected, this pin outputs a refresh pulse to the external pseudo static memory.
$\overline{RESET}$	Input	This pin inputs a low level active system reset signal.
V_{DD}		Positive power supply pin.
V_{SS}		GND

CHAPTER 2 CPU ARCHITECTURE

2.1 Memory Space

The μPD78312/78310 can address memory address up to 64K bytes (refer to Fig. 2.1). Of the 64K bytes, a space from 0000 to FEFFH can be fetched by program. A 256-byte area (FF00 to FFFFH) is reserved as a special register area.

(1) Vector table area

The interrupt request from peripheral hardware, reset input, external interrupt request, and interrupt branch address generated by the break instruction are stored in the area of 64K bytes (0000 to 003FH).

$\overline{\text{RESET}}$	(reset input)	0000H
NMI	(NMI pin input)	0002H
EXIF0	(INTE0 pin input)	0004H
EXIF1	(INTE1 pin input)	0006H
EXIF2	(INTE2 pin input)	0008H
WDT	(Watchdog timer)	000AH
TBF	(Time base counter)	000CH
TMF0	(Timer unit)	000EH
TMF1	(Timer unit)	0010H
TMF2	(Timer unit)	0012H
CRF00	(Count unit)	001AH
CRF01	(Count unit)	001CH
CRF10	(Count unit)	001EH
CRF11	(Count unit)	0020H
SEF	(Serial receive error)	0022H
SRF	(Serial receive complete) ...	0024H
STF	(Serial transmit complete) ..	0026H
ADF	(A/D converter)	0028H
BRK	(Break instruction)	003EH

(2) CALLT table area

The call address for 1-byte call instruction (CALLT) can be stored in the 64K-byte area of addresses 0040 to 007FH.

(3) CALLF entry area

The area 0800 to 0FFFH can be directly addressed by 2-byte call instruction (CALLF).

(4) Internal RAM area

A 256-byte RAM is provided in addresses FE00 to FEFFH. Eight general-purpose register banks are mapped to the area of 128 bytes (FE80 to FEFFH) of the internal RAM area.

(5) Special register area

Special function registers, such as the on-chip peripheral hardware mode register and control register, are mapped to the area FF00 to FFFFH.

(6) External memory area

For the μPD78312, the external memory (ROM or RAM) can be connected to the 56K-byte area (2000 to FDFFH). For the μPD78310, the external memory (ROM or RAM) can be connected to the 64K-byte area (0000 to FDFFH). The external memory can be accessed by using P47-40 (multiplexed address/data bus), $\overline{\text{RD}}$, $\overline{\text{WR}}$, and ALE signals. The pseudo static memory refresh pulse output port ($\overline{\text{RFSH}}$) is provided; therefore, a pseudo static memory equivalent to μPD4168 can be connected.

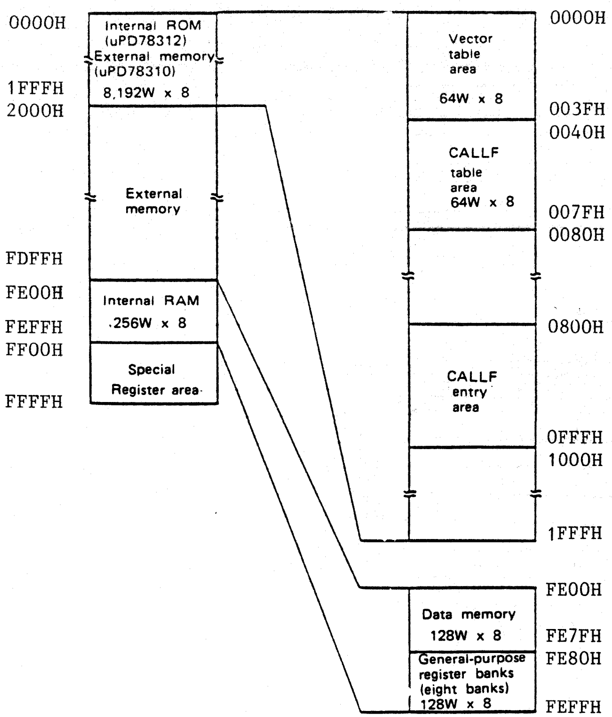
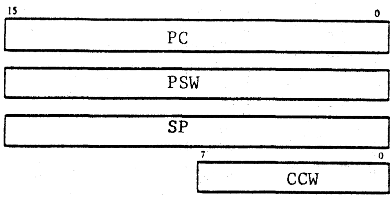


Fig. 2.1 Memory Map

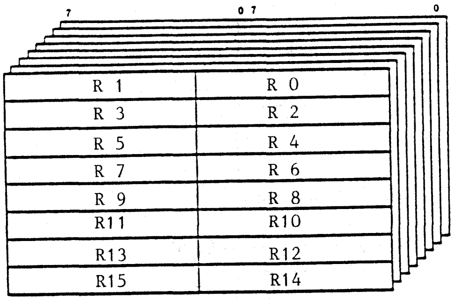
2.2 Processor Register

The processor register consists of three groups of registers. These three groups are: the general-purpose register group of eight banks of which each consists of sixteen 8-bit registers, the control register group which consists of an 8-bit register and three 16-bit registers, and the special register group which consists of registers having special functions such as the peripheral hardware I/O mode register.

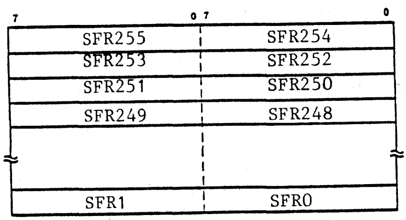
Control Register



General-Purpose Register



Special Register



Note: The PSW, SP, and CCW of the control register group are mapped to the special register area (SFR).

Fig. 2.2 Register Configuration

2.2.1 Control register

The control register group consists of three 16-bit registers and one 8-bit register, and has dedicated functions such as controlling program sequence, status, and stack memory, or qualifying operand addressing.

(1) Program counter (PC)

The PC is a 16-bit register that holds the address of the next instruction to be executed; the counter is automatically incremented by the number of bytes of the instruction to be fetched. Either immediate data or the contents of a specified register are loaded if a branching instruction is executed. When a $\overline{\text{RESET}}$ is input, the reset vector data of 00H and 01H are loaded to the PC and branched.

(2) Program status word (PSW)

This is a 16-bit register which consists of various flags that are set or reset according to instruction execution results. The contents of the PSW are automatically saved to the stack when an interrupt request is generated or the BRK instruction is executed, and restored by executing the RETI instruction. All bits are reset to 0 by $\overline{\text{RESET}}$ input.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	RBS 2	RBS 1	RBS 0			IE		S	Z	RSS	AC	UF	P/V	SUB	CY

Fig. 2.3 PSW Configuration

(a) Subtraction flag (SUB)

This flag is set to 1 when the arithmetic logical unit (ALU) has performed a subtraction operation; otherwise, it is reset to 0. This flag is effective for executing a decimal adjust instruction after a BCD subtraction execution.

(b) Carry flag (CY)

This bit is set to 1 when a carry is made after the operation from either bit 7 or 15 or a borrow is made to one of them; otherwise, it is reset to 0. This bit can be tested by a conditional branch instruction. When a bit manipulation instruction is executed, this bit functions as a bit accumulator.

(c) Zero flag (Z)
This bit is set to 1 if the operation results in 0; otherwise, it is reset to 0. This bit can be tested by a conditional branch instruction.

(d) Sign flag (S)
This bit is set to 1 if the MSB of the operation result is 1, and reset if the MSB of the result is 0. This bit can be tested by a conditional branch instruction.

(e) Parity/overflow flag (P/V)
When an arithmetic operation instruction is executed, this bit is set only when an overflow or underflow has occurred as 2's complement; otherwise, it is reset to 0 (overflow flag operation).
When a logical operation instruction is executed, this bit is set to 1 if the number of bits set to 1 as the operation result is even, or reset to 0 if odd (parity flag operation).
This bit can be tested by a conditional branch instruction.

(f) Auxiliary carry flag (AC)
This bit is set to 1 when a carry is made after the operation from bit 3 or a borrow is made to it; otherwise, this bit is reset to 0. This bit can be tested by a conditional branch instruction.

(g) Register set selection flag (RSS)
This bit selects the general-purpose registers that function as the X, A, C, and B registers. When the RSS is set to 0, each functional register and absolute register corresponds in this way; X(R0), A(R1), C(R2), and B(R3). When the RSS is set to 1, they correspond in this way; X(R4), A(R5), C(R6), and B(R7). Refer to Fig. 2.6, General-Purpose Register Configuration.

(h) Interrupt request enable flag (IE)
This bit indicates whether or not the interrupt is enabled. This flag is set to 1 by executing the EI instruction, and reset to 0 by executing the DI instruction, or when an interrupt is accepted.

(i) Register bank selection flag (RBS0-2)
This 3-bit flag selects one of eight register banks (RBANK0 to RBANK7).

(j) User flag (UF)
This flag can be set or reset by user program for program control.

(3) CPU control word (CCW)
This is a 2-bit register related to the CPU control. It is mapped to the special register area and can be controlled by software. All bits are reset to 0 by $\overline{\text{RESET}}$ input.

7	6	5	4	3	2	1	0	
0	0	0	—	0	0	TPF	EOS	CCW

Fig. 2.4 CCW Configuration

(a) Table position flag (TPF)
This flag specifies the location of the vector table referenced for the CALLT instruction or interrupt request. When the TPF is at 0 (reset), the vector table location is 0000 to 007FH. When the TPF is at 1 (set), the vector table location is specified to 8000 to 807FH.

(b) End of software interrupt flag (EOS)
This flag disables resetting (to 0) of the in-service priority register (ISPR) when the processing returns by executing the RETI or RETCS instruction from the interrupt service routine started by executing the BRK or BRKCS instruc-

tion. Resetting of the ISPR is disabled when the EOS is set to 1. Return from the software interrupt can be done by setting the EOS flag to 1. The EOS flag is reset to 0 after the RETI or RETCS instruction is executed.

Note: ISPR is an 8-bit register that retains the priority order level of the interrupt request source being accepted. (See Section 4.1.3, Interrupt request by software.)

2.2.2 General-purpose register

The general-purpose registers consist of eight register banks, and each bank consists of sixteen 8-bit registers. 128 bytes of register banks are mapped in the specified area (FE80 to FEFFH) of the internal RAM space.

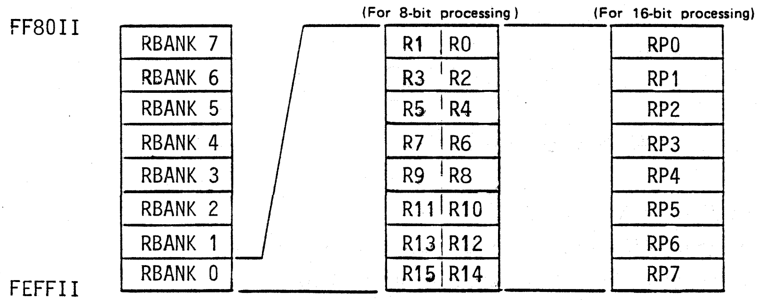


Fig. 2.5 General-Purpose Register Memory Location

Two 8-bit registers can be paired to form a 16-bit pair register (RP0-RP7). Each of the sixteen 8-bit registers is characterized as shown in Fig. 2.6 according to its functional name. The X register functions as the lower side of the 16-bit accumulator. The A register functions as the 8-bit accumulator or the upper side of the 16-bit accumulator. The B and C registers function as the counter. The DE, HL, VP, and UP registers can be paired to form an address register. Especially, the VP and UP registers have functions of base register and user stack pointer, respectively.

Absolute name	Functional name		Absolute name	Functional name	
	RSS=0	RSS=1		RSS=0	RSS=1
R 0	X		R 8	VPL	VPL
R 1	A		R 9	VPH	VPH
R 2	C		R10	UPL	UPL
R 3	B		R11	UPH	UPH
R 4		X	R12	E	E
R 5		A	R13	D	D
R 6		C	R14	L	L
R 7		B		II	II

Fig. 2.6 General-Purpose Register Configuration

Depending on the value of the register set selection flag (RSS) of PSW, the registers having inherent functions will change. For process data address addressing, the μPD78312/78310 is capable of implied addressing by functional name in which the inherent function of each register is respected and register addressing by absolute name is done for the purpose of creating programs that allow high-speed processing to reduce the number of transfers thus being highly descriptive.

2.2.3 Special register (SFR)

This is a register group to which special functions, such as mode registers for various peripheral hardware and control registers (PSW, SP, and CCW), are assigned. The special register group is mapped to a space of 256 bytes (FF00 to FFFFH). In the area of 32 bytes (FF00 to FF1FH), short direct memory addressing is used so that processing can be done with short word length. Bit manipulation, operation, and transfer instructions can be executed in the entire area. The area of 16 bytes (FFB0 to FFBFH) can be externally accessed by SFR addressing.

Table 2.1 is a special register (SFR) table. The meaning of each item in the table is as follows:

- Abbreviation
  The symbol that indicates the address of the internal special register. This can be written in the operand field of the instruction.
- R/W
  Indicates whether the corresponding special register can be read or written.
  R/W: Read/write enabled
  R : Read only
  W : Write only
- 16-bit manipulation
  The register that can be manipulated in 16 bits. This can be written in sfrp of the operand field. For 16-bit manipulation, address should be specified in even address.
- When RESET
  Indicates the status of each register when $\overline{\text{RESET}}$ is input.

Table 2.1 Special Register (SFR) Table

ADDRESS	DESCRIPTION	FORMAT 15 8	FORMAT 7 0	R/W	RESET DATA
FF00	PORT 0 (USE AS G.P. PORT)	–	P0	R/W	X
FF01	PORT 1	–	P1	R/W	X
FF02	PORT 2	–	P2	R/W	X
FF03	PORT 3	–	P3	R *	X
FF04	PORT 4	–	P4	R/W	X
FF05	PORT 5	–	P5	R/W	X
FF08	UP/DOWN COUNTER UNIT 0, CAPTURE REGISTER 0, LOW BYTE	–	CR00L	R/W	X
FF09	UP/DOWN COUNTER UNIT 0, CAPTURE REGISTER 0, HIGH BYTE	CR00H	–	R/W	X
FF0A	UP/DOWN COUNTER UNIT 0, CAPTURE REGISTER 1, LOW BYTE	–	CR01L	R/W	X
FF0B	UP/DOWN COUNTER UNIT 0, CAPTURE REGISTER 1, HIGH BYTE	CR01H	–	R/W	X
FF0C	UP/DOWN COUNTER UNIT 1, CAPTURE REGISTER 0, LOW BYTE	–	CR10L	R/W	X
FF0D	UP/DOWN COUNTER UNIT 1, CAPTURE REGISTER 0, HIGH BYTE	CR10H	–	R/W	X
FF0E	UP/DOWN COUNTER UNIT 1, CAPTURE REGISTER 1, LOW BYTE	–	CR11L	R/W	X
FF0F	UP/DOWN COUNTER UNIT 1, CAPTURE REGISTER 1, HIGH BYTE	CR11H	–	R/W	X
FF10	CAPTURE/PWM UNIT, CAPTURE LATCH 0, LOW BYTE	–	CPT0L	R/W	X
FF11	CAPTURE/PWM UNIT, CAPTURE LATCH 0, HIGH BYTE	CPT0H	–	R/W	X
FF12	CAPTURE/PWM UNIT, CAPTURE LATCH 1, LOW BYTE	–	CPT1L	R/W	X
FF13	CAPTURE/PWM UNIT, CAPTURE LATCH 1, HIGH BYTE	CPT1H	–	R/W	X
FF14	CAPTURE/PWM UNIT 0, PWM REGISTER, LOW BYTE	–	PWM0L	R/W	X
FF15	CAPTURE/PWM UNIT 0, PWM REGISTER, HIGH BYTE	PWM0H	–	R/W	X
FF16	CAPTURE/PWM UNIT 1, PWM REGISTER, LOW BYTE	–	PWM1L	R/W	X
FF17	CAPTURE/PWM UNIT 1, PWM REGISTER, HIGH BYTE	PWM1H	–	R/W	X
FF1C	UP/DOWN COUNTER UNIT 0, UP/DOWN COUNTER REG, LOW BYTE	–	UDC0L	R/W	X
FF1D	UP/DOWN COUNTER UNIT 0, UP/DOWN COUNTER REG, HIGH BYTE	UDC0H	–	R/W	X
FF1E	UP/DOWN COUNTER UNIT 1, UP/DOWN COUNTER REG, LOW BYTE	–	UDC1L	R/W	X
FF1F	UP/DOWN COUNTER UNIT 1, UP/DOWN COUNTER REG, HIGH BYTE	UDC1H	–	R/W	X
FF20	PORT 0, PORT MODE REGISTER	–	PM0	R/W	FFH
FF21	PORT 1, PORT MODE REGISTER	–	PM1	R/W	FFH
FF22	PORT 2, PORT MODE REGISTER	–	PM2	R/W	FFH
FF23	PORT 3, PORT MODE REGISTER	–	PM3	R/W	FFH
FF25	PORT 5, PORT MODE REGISTER	–	PM5	R/W	FFH
FF32	PORT 2, PORT MODE CONTROL REGISTER	–	PMC2	R/W	0FH
FF33	PORT 3, PORT MODE CONTROL REGISTER	–	PMC3	R/W	0FH
FF38	REAL TIME OUTPUT PORT CONTROL REGISTER	–	RTPC	R/W	08H
FF3A	REAL TIME OUTPUT PORT LOWER NIBBLE (PORT 0, BUFFER REG.)	–	P0L	R/W	X
FF3B	REAL TIME OUTPUT PORT HIGHER NIBBLE (PORT 0, BUFFER REG.)	–	P0H	R/W	X

*: Bits 0 to 3 of P2 and those of P3 are read only.

**: P0H and P0L are 4-bit buffers, and they can manipulate the higher 4 bits and lower 4 bits of 8-bit data. When they are paired, they can manipulate 8-bit data.

ADDRESS	DESCRIPTION	FORMAT 15 8	FORMAT 7 0	R/W	RESET DATA
FF40	MEMORY MAPPING REGISTER (MEMORY EXPANSION)	–	MM	R/W	30H
FF41	REFRESH MODE REGISTER	–	RFM	R/W	10H
FF42	WATCH DOG TIMER, WATCH DOG MODE REGISTER	–	WDM	R/W	00H
FF44	STAND BY MODES, STAND BY CONTROL REGISTER	–	STBC	R/W	2XH *
FF46	TIME BASE COUNTER, TIME BASE MODE REGISTER	–	TBM	R/W	00H
FF48	INTERRUPT MODE REGISTER (EXTERNAL INTERRUPTS)	–	INTM	R/W	00H
FF4A	INT SERVICE PRIORITY REGISTER	–	ISPR	R/W	00H
FF4E	CPU CONTROL WORD	–	CCW	R/W	00H
FF50	SERIAL INTERFACE, SERIAL COMMUNICATION MODE REGISTER	–	SCM	R/W	00H
FF52	SERIAL INTERFACE, SERIAL COMMUNICATION CONTROL REGISTER	–	SCC	R/W	00H
FF53	SERIAL INTERFACE, BAUD RATE GENERATOR REGISTER	–	BRG	R/W	00H
FF56	SERIAL INTERFACE, RECEIVE BYTE REGISTER	–	RxB	R/O	X
FF57	SERIAL INTERFACE, TRANSMIT BYTE REGISTER	–	TxB	W/O	X
FF60	CAPTURE UNIT, FREE RUNNING COUNTER CONTROL REGISTER	–	FRCC	R/W	00H
FF64	CAPTURE/PWM UNIT, CAPTURE MODE REGISTER	–	CPTM	R/W	00H
FF66	CAPTURE/PWM UNIT, PWM MODE REGISTER	–	PWMM	R/W	00H
FF68	A/D CONVERTER, A/D CONVERTER MODE REGISTER	–	ADM	R/W	00H
FF6A	A/D CONVERTER, CONVERSION RESULT REG	–	ADCR	R/O	X
FF70	UP/DOWN COUNTER, COUNTER UNIT INPUT MODE REGISTER	–	CUIM	R/W	00H
FF72	UP/DOWN COUNTER 0, CONTROL REGISTER	–	UDCC0	R/W	00H
FF74	UP/DOWN COUNTER, CAPTURE/COMPARE CONTROL REG	–	CRC	R/W	00H
FF7A	UP/DOWN COUNTER 1, CONTROL REGISTER	–	UDCC1	R/W	00H
FF80	TIMER UNIT TIMER 0, TIMER CONTROL REGISTER	–	TMC0	R/W	00H
FF82	TIMER UNIT TIMER 1, TIMER CONTROL REGISTER	–	TMC1	R/W	00H
FF88	TIMER UNIT TIMER 0, TIMER REGISTER LOW BYTE	–	TM0L	R/W	X
FF89	TIMER UNIT TIMER 0, TIMER REGISTER HIGH BYTE	TM0H	–	R/W	X
FF8A	TIMER UNIT TIMER 0, MODULO REGISTER LOW BYTE	–	MD0L	R/W	X
FF8B	TIMER UNIT TIMER 0, MODULO REGISTER HIGH BYTE	MD0H	–	R/W	X
FF8C	TIMER UNIT TIMER 1, TIMER REGISTER LOW BYTE	–	TM1L	R/W	X
FF8D	TIMER UNIT TIMER 1, TIMER REGISTER HIGH BYTE	TM1H	–	R/W	X
FF8E	TIMER UNIT TIMER 1, MODULO REGISTER LOW BYTE	–	MD1L	R/W	X
FF8F	TIMER UNIT TIMER 1, MODULO REGISTER HIGH BYTE	MD1H	–	R/W	X

*: Bit 3 is not affected by $\overline{\text{RESET}}$ input; therefore, the value of the lower 4 bits is either 0 or 8.

ADDRESS	DESCRIPTION	FORMAT 15 8	7 0	R/W	RESET DATA
FFC0	INTERRUPT CONTROL REGISTER UP/DOWN COUNTER 0 CR00	–	CRIC00	R/W	47H
FFC1	MACRO SERVICE CONTROL REGISTER UP/DOWN COUNTER 0 CR00	–	CRMS00	R/W	X
FFC2	INTERRUPT CONTROL REGISTER UP/DOWN COUNTER 0 CR01	–	CRIC01	R/W	47H
FFC4	INTERRUPT CONTROL REGISTER UP/DOWN COUNTER 1 CR10	–	CRIC10	R/W	47H
FFC5	MACRO SERVICE CONTROL REGISTER UP/DOWN COUNTER 1 CR10	–	CRMS10	R/W	X
FFC6	INTERRUPT CONTROL REGISTER UP/DOWN COUNTER 1 CR11	–	CRIC11	R/W	47H
FFC8	INTERRUPT CONTROL REGISTER EXT. INTERRUPT 0	–	EIC0	R/W	47H
FFC9	MACRO SERVICE CONTROL REGISTER EXT. INTERRUPT 0	–	EMS0	R/W	X
FFCA	INTERRUPT CONTROL REGISTER EXT. INTERRUPT 1	–	EIC1	R/W	47H
FFCB	MACRO SERVICE CONTROL REGISTER EXT. INTERRUPT 1	–	EMS1	R/W	X
FFCC	INTERRUPT CONTROL REGISTER EXT. INTERRUPT 2	–	EIC2	R/W	47H
FFCD	MACRO SERVICE CONTROL REGISTER EXT. INTERRUPT 2	–	EMS2	R/W	X
FFCE	INTERRUPT CONTROL REGISTER TIMER INTERRUPT 0	–	TMIC0	R/W	47H
FFCF	MACRO SERVICE CONTROL REGISTER TIMER INTERRUPT 0	–	TMMS0	R/W	X
FFD0	INTERRUPT CONTROL REGISTER TIMER INTERRUPT 1	–	TMIC1	R/W	47H
FFD1	MACRO SERVICE CONTROL REGISTER TIMER INTERRUPT 1	–	TMMS1	R/W	X
FFD2	INTERRUPT CONTROL REGISTER TIMER INTERRUPT 2	–	TMIC2	R/W	47H
FFD3	MACRO SERVICE CONTROL REGISTER TIMER INTERRUPT 2	–	TMMS2	R/W	X
FFDA	INTERRUPT CONTROL REGISTER SERIAL ERROR INTERRUPT	–	SEIC	R/W	47H
FFDC	INTERRUPT CONTROL REGISTER SERIAL RECEIVE INTERRUPT	–	SRIC	R/W	47H
FFDD	MACRO SERVICE CONTROL REGISTER SERIAL RECEIVE INTERRUPT	–	SRMS	R/W	X
FFDE	INTERRUPT CONTROL REGISTER SERIAL TRANSMIT INTERRUPT	–	STIC	R/W	47H
FFDF	MACRO SERVICE CONTROL REGISTER SERIAL TRANSMIT INTERRUPT	–	STMS	R/W	X
FFE0	INTERRUPT CONTROL REGISTER A/D CONVERTER INTERRUPT	–	ADIC	R/W	47H
FFE1	MACRO SERVICE CONTROL REGISTER A/D CONVERTER INTERRUPT	–	ADMS	R/W	X
FFE2	INTERRUPT CONTROL REGISTER TIME BASE COUNTER	–	TBIC	R/W	47H
FFFC	STACK POINTER, LOW BYTE	–	SPL	R/W *	X
FFFD	STACK POINTER, HIGH BYTE	SPH	–	R/W *	X
FFFE	PROGRAM STATUS WORD, LOW BYTE	–	PSWL	R/W *	00H
FFFF	PROGRAM STATUS WORD, HIGH BYTE	PSWH	–	R/W *	00H

*: This space uses SFR addressing and can access the external memory.

**: SP and PSW are not manipulated by SFR addressing, but by a dedicated instruction.

2.3 Data Memory Addressing

In the μPD78312/78310, the internal RAM space (FE00 to FEFFH) and special register area (FF00 to FFFFH) are mapped to the area FE00 to FFFFH. Short direct memory addressing is used in some part (FE20 to FF1FH) of the data memory space; therefore, direct addressing using 1-byte data of instruction is possible.

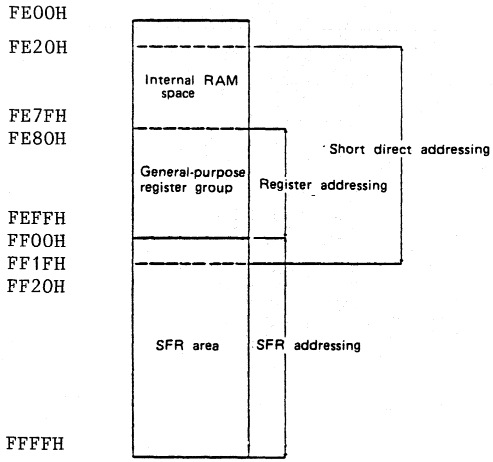


Fig. 2.7 Data Memory Addressing Space

2.3.1 General-purpose register addressing

The general-purpose register consists of eight register banks, each of which consists of sixteen 8-bit registers or eight 16-bit registers. General-purpose register addressing is done by the 3- or 4-bit register specification field provided from the instruction word and the register bank selection flag (RBS0-2) in PSW.

2.3.2 Short direct addressing

Short direct addressing which can directly specify the address by 1-byte data in the instruction word can be used in the area FE20 to FF1FH. The short direct memory is accessed as 8-bit data or 16-bit data. When accessed as 16-bit data, 2-byte data specified by continuous addresses of even and odd is accessed regardless of whether the 1-byte address specification data is odd or even.

2.3.3 Special register (SFR) addressing

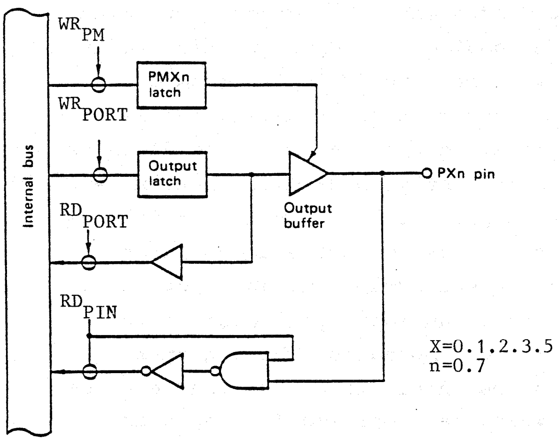
This addressing is used to manipulate the special registers (SFR) mapped in the SFR area which is from FF00 to FFFFH. Addressing is performed by the 1-byte data of the instruction word corresponding to the lower 8 bits of the special register address.

CHAPTER 3 PERIPHERAL HARDWARE FUNCTIONS

3.1 Port Function

3.1.1 Hardware configuration

Basically, each port of the μPD78312/78310 is configured as a three-state bidirectional port as shown in Fig. 3.1. When $\overline{\text{RESET}}$ is input, all bits of the port mode register are set to 1 and the corresponding port becomes an input port. All pins of the port enter the high impedance state. The contents of the output latch are not affected by $\overline{\text{RESET}}$ input.



Note: PMXn latch: Bit n of the PMX port mode register.

Fig. 3.1 Port Configuration

(1) When specified as output port (PMXn=0)
When a port is designated as an output port, the output latch becomes valid and data can be exchanged between the output latch and the accumulator using the transfer instruction. The contents of the output latch can be set by the logical operation instruction if necessary. The data, once written to the output latch, can be held until an instruction is executed to operate the port.

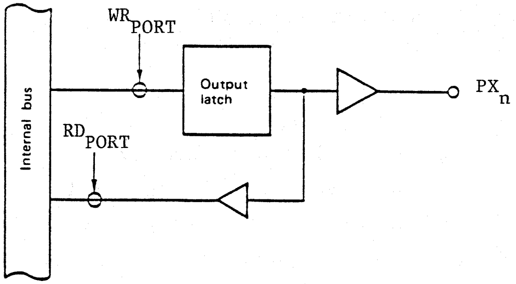


Fig. 3.2 Port Specified as Output Port

(2) When specified as input port (PMXn=1)

When a port is specified as an input port, the level of the port can be loaded into the accumulator. Even when the port is designated for input, writing to its output latch can still be done. This is because data transferred from the accumulator by a transfer instruction will be sent to all output latches regardless of whether the port is currently input or output. However, the contents of an output latch of a bit specified as an input port cannot be loaded to the accumulator. Neither can the contents be output to an external pin (which is functioning as an input pin) because the output buffer is at high impedance. The contents of the output latch can be output to a port pin when the corresponding bit which is specified for input is respecified as an output port. Refer to Fig. 3.3.

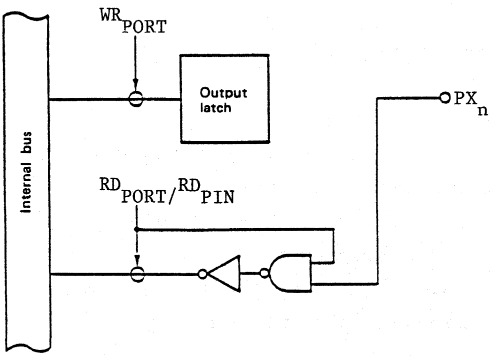


Fig. 3.3 Port Specified as Input Port

(3) When specified as a control signal port (PMC2n, PMC3n=1)
Port 2 or 3 can input or output the control signal in a bit unit by setting the bit of the corresponding port mode control register (PMC2, PMC3) to 1 regardless of the setting of the port mode register (PM2, PM3). When all pins of the port are used as control signal pins, the status of the control signal can be checked by executing a port access instruction.

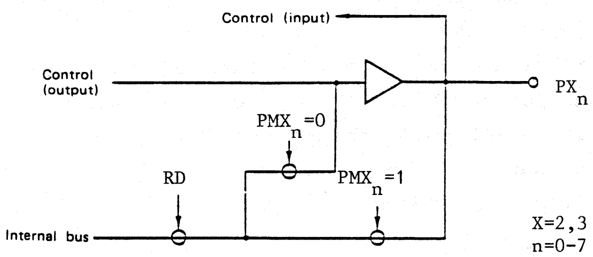


Fig. 3.4 Port Specified as Control Signal Port

(a) When port is control signal output
If the port read instruction is executed when the port mode register (PM2n, PM3n) is set to 1, the status of the control signal pin can be read by executing the port read instruction.
If the port read instruction is executed when the port mode register (PM2n, PM3n) is reset to 0, the status of the internal control signal can be read.

(b) When port is control signal input
The status of the control signal pin can be read by executing the port read instruction only when the port mode register is set to 1.

3.1.2 Port function

(1) P00-07 (port 0) ... Three-state Input/Output
Port 0 is an 8-bit bidirectional port, for which input or output may be specified per bit using the

Port 0 mode register (PM0). This port also functions as a real-time output port which outputs the contents of the Port 0 buffer register (P0L, P0H) in programmable intervals by setting the real-time output port control register (RTPC) (refer to 4.4, Real-Time Output Port Function).

When $\overline{\text{RESET}}$ is input, all bits of the Port 0 mode register (PM0) are set to 1, and port 0 becomes an input port (output high impedance).

(a) Port 0 mode register (PM0)

This is an 8-bit register which specifies port 0 for input or output per bit. If a bit of the Port 0 mode register is set to 1, the corresponding pin becomes an input. If the bit is reset to 0, the corresponding pin becomes an output.

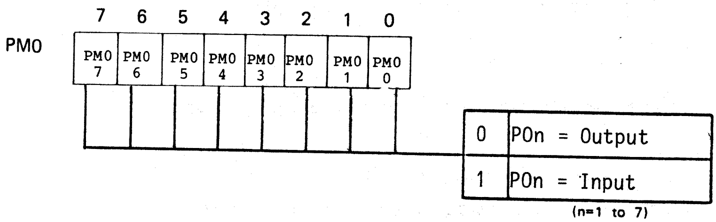


Fig. 3.5 Port 0 Mode Register Format

(2) P10-17 (port 1) ... Three-state Input/Output

This is a general-purpose 8-bit bidirectional port, for which input or output may be specified per bit using the Port 1 mode register.

(a) Port 1 mode register (PM1)

In the same way as the Port 0 mode register (PM0), this 8-bit register specifies Port 1 for input or output per bit. When $\overline{\text{RESET}}$ is input, all bits of this register are set to 1, and Port 1 enter the input port mode (output high impedance).

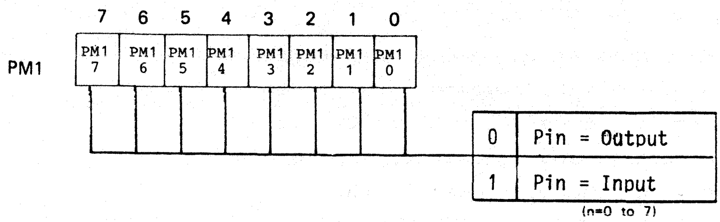


Fig. 3.6 Port 1 Mode Register Format

(3) P20-27 (port 2) ... Three-state Input/Output

This is an 8-bit special I/O port. In the same way as port 0, this port functions as a general-purpose I/O port (pins 20 to 23 are input only), for which input or output may be specified per bit. Additionally, the port functions as control port in various modes. These modes may be specified per bit using the Port 2 mode register (PM2) and Port 2 mode control register (PMC2).

Table 3.1 Port 2 Function

	PMC2n=1		PMC2n=0	
	PM2n=1	PM2n=0	PM2n=0	PM2n=1
P20	NMI input			
P21	INTE0 input			
P22	INTE1 input			
P23	INTE2 input			
P24	TxD input		Output port	Input port
P25	RxD input		Output port	Input port
P26	$\overline{\text{SCK}}$ output		Output port	Input port
P27	$\overline{\text{CTS}}$ input		Output port	Input port

(n = 0 to 7)

Levels of pins P20 to 23 can be read by performing Port 2 (P2) direct read access (refer to Table 3.2).

(a) Port 2 mode control register (PMC2)

This is an 8-bit register which specifies port/control signal input or output for Port 2. If the corresponding bit of the PMC2 register is set to 1, a bit of Port 2 enters the control.

If the corresponding bit is reset to 0, a bit of Port 2 enters the input/output mode. When $\overline{\text{RESET}}$ is input, all bits of the PMC2 register are set to 1 and enter the control mode. Pins 20 to 23 will be fixed in the control mode; however, the levels of these pins may be read by accessing the Port 2 register (P2).

7	6	5	4	3	2	1	0
PMC2 7	PMC2 6	PMC2 5	PMC2 4	1	1	1	1

0	P2n= port mode
1	P2n= control mode

Fig 3.7 Port 2 Mode Control Register Format

(b) Port 2 mode register (PM2)

In the same way as the PM0 register of Port 0, this 8-bit register specifies input or output for Port 2 per bit. All bits of this register are set to 1 by inputting a $\overline{\text{RESET}}$.

7	6	5	4	3	2	1	0
PM2 7	PM2 6	PM2 5	PM2 4	1	1	1	1

0	P2n=Output
1	P2n=Input

Fig. 3.8 Port 2 Mode Register Format

Table 3.2 Port 2 (P2) Read Access

	PM2	PMC2	To read
P20/NMI P21/INTE0 P22/INTE1 P23/INTE2	1	1	Pin level
P24/TxD	0	0	Content of output latch
	0	1	Internal control signal level
	1	x	Pin level
P25/RxD	0	0	Content of output latch
	0	1	Content of output latch
	1	x	Pin level
P26/SCK	0	0	Content of output latch
	0	1	Internal control signal level
	1	x	Pin level
P27/$\overline{CTS}$	0	0	Content of output latch
	0	1	Content of output latch
	1	x	Pin level

x: don't care

(4) P30-37 (port 3) ... Three-state Input/Output
In the same way as Port 2, this 8-bit special I/O port functions as a general-purpose I/O port, for which input or output may be specified per bit (pins 30 to 33 are input only). Additionally, the port functions as control pins in various modes. These modes may be specified per bit using Port 3 mode register (PM3) and Port 3 mode control register (PMC3).

Table 3.3 Port 3 Function

	PMC3n=1		PMC3n=0	
	PM3n=1	PM3n=0	PM3n=0	PM3n=1
P30	CI0 input			
P31	CTRL0 input			
P32	CI1 input			
P33	CTRL1 input			
P34	PMW0 output		Output	Input
P35	PMW1 output		Output	Input
P36	TO0 output/CLR0 input		Output	Input
P37	TO1 output/CLR1 input		Output	Input

Levels of pins P30 to 33 can be read by performing port 3 (P3) direct read access (refer to Table 3.4).

(a) Port 3 mode control register (PMC3)

This is an 8-bit register which specifies port/ control signal input or output for port 3. If the corresponding bit of the PMC3 register is set to 1, a bit of Port 3 enters the control mode.
If the corresponding bit is reset to 0, a bit of port 3 is input\output port mode. When $\overline{RESET}$ is input, all bits of the PMC3 register are set to 1 and enters the control mode. Pins 30 to 33 will be fixed in the control mode; however, the levels of these pins may be read by accessing the Port 3 register (P3).

7	6	5	4	3	2	1	0
PMC3 7	PMC3 6	PMC3 5	PMC3 4	1	1	1	1

0	P3n= port mode
1	P3n=control mode

(n= 0 to 7)

Fig 3.9 Port 3 Mode Control Register Format

(b) Port 3 mode register (PM3)

In the same way as the PM0 register of Port 0, this 8-bit register specifies input or output for Port 3 per bit. All bits of this register are set to 1 by inputting $\overline{RESET}$.

7	6	5	4	3	2	1	0
PM3 7	PM3 6	PM3 5	PM3 4	1	1	1	1

0	P3n= Output
1	P3n= Input

(n= 0 to 7)

Fig. 3.10 Port 3 Mode Register Format

Table 3.4 Port 3 (P3) Read Access

	PM3	PMC3	To read
P30/CI0	1	1	Pin level
P31/CTRL0			
P32/CI1			
P33/CTRL1			
P34/PWM0	0	0	Content of output latch
	0	1	Internal control signal level
	1	x	Pin level
P35/PWM1	0	0	Content of output latch
	0	1	Internal control signal level
	1	x	Pin level
P36/TO0 /CLR0	0	0	Content of output latch
	0	1	Internal control signal level
	1	x	Pin level
P37/TO1 /CLR1	0	0	Content of output latch
	0	1	Internal signal level
	1	x	Pin level

x: don't care

(5) P40-47 (port 4) ... Three-state Input/Output
This port functions as a general-purpose 8-bit bidirectional port, for which input or output may be specified per bit using the memory extension mode register (MM).
Additionally, when the external memory or I/O is extended, this port functions as an address/data bus regardless of the port 4 register (P4) setting.

(6) P50-57 (port 5) ... Three-state Input/Output
This port functions as a general-purpose 8-bit bidirectional port, for which input or output may be specified per bit using the memory extension mode register (MM).
Additionally, when the external memory or I/O is extended, this port functions as an address/data bus regardless of the Port 5 register (P5) setting.
Bits specified for extension mode by setting the memory extension mode register (MM) become an address

bus regardless of the setting of the Port 5 register (P5) and the Port 5 mode register (PM5). Bits specified for port mode can be specified for input or output using the Port 5 mode register (PM5).

Table 3.5 Functions of Ports 4 and 5

Mode specification by MM register		Port 4	Port 5							
			P57	P56	P55	P54	P53	P52	P51	P50
Port mode	Single chip	Input port	Port mode							
		Output port	Port mode							
Expansion mode	256-byte expansion	Multiplexed address/data bus	Port mode							
	4K-byte expansion	Multiplexed address/data bus	Port mode				A 11	A 10	A 9	A 8
	16K-byte expansion	Multiplexed address/data bus	port mode		A 13	A 12	A 11	A 10	A 9	A 8
	56K-byte expansion	Multiplexed address/data bus	A 15	A 14	A 13	A 12	A 11	A 10	A 9	A 8

An: Address bus

(a) Port 5 mode register (PM5)

In the same way as Port 0, this register specifies Port 5 for input or output per bit. When $\overline{\text{RESET}}$ is input, all bits of this register are set to 1, and the port becomes an input (output high impedance) port.

7	6	5	4	3	2	1	0	
PM5 7	PM5 6	PM5 5	PM5 4	PM5 3	PM5 2	PM5 1	PM5 0	PM5

0	P5n=Output port
1	P5n=Input port

(n= 0 to 7)

Fig. 3.11 Port 5 Mode Register Format

(b) Memory extension register (MM)

When the external memory or I/O is connected, this 8-bit register controls control signals such as address bus, address/data bus, $\overline{\text{RD}}$, and $\overline{\text{WR}}$ signals. A bit field consisting of bits MM0 to 2 of the memory extension register (MM) specifies

the size of the external memory which becomes effective when the $\overline{EA}$ pin is at high (1) level. Bits PW0 to 1 specify the number of wait cycles to be inserted to the external access cycle regardless of the $\overline{EA}$ pin level.
When $\overline{RESET}$ is input, Memory extension register is set to 30H, and the number of wait cycles to be inserted to the external access cycle is set to 3.

7	6	5	4	3	2	1	0
0	0	PW1	PW0	0	MM2	MM1	MM0

Operations of P40-47, and P50-57

0	0	0	Port mode	Single-chip mode	P40-47: Input port P50-57: Port mode
0	0	1			P40-47: Output port P50-57: Port mode
0	1	1	Ex-pan-sion mode	256 bytes	P40-47: Expansion mode P50-57: Port mode
1	0	0		4K bytes	P40-47: Expansion mode P50-53: Expansion mode P54-57: Port mode
1	0	1		16K bytes	P40-47: Expansion mode P50-55: Expansion mode P56-57: Port mode
1	1	1		56K bytes	P40-47: Expansion mode P50-57: Expansion mode

Wait insertion for external access

0	0	No wait cycle is inserted to the external access cycle.
0	1	1 wait cycle is inserted to the external access cycle.
1	0	2 wait cycles are inserted to the external access cycle.
1	1	3 wait cycles are inserted to the external access cycle.

Fig. 3.12 MM Register Format

3.2 Pulse Input/Output Unit

The pulse input/output unit consists of four blocks, a count unit, capture/PWM unit, timer unit, and real-time output port.

3.2.1 Count unit

3.2.1.1 Count unit configuration

The count unit consists of 16-bit presettable up/down count registers (UDC0, UDC1), 16-bit capture compare registers (CR00, CR01, CR10, CR11), up/down count control registers (UDCC0, UDCC1), capture compare register control register (CRC), count unit input mode register (CUIM), and interrupt request control registers (CRIC00/01, CRIC10/11). The presettable up/down count registers (UDC0, UDC1) count the internal* or external clock. Up count or down count operation can be controlled by software or through the external pins (CTRL0, CTRL1). Each of the UDC0 and UDC1 registers has a clear function and a data preset function.
Count clock specification, count register clear function, and a data preset function controls are performed by the up/down count control registers (UDCC0, UDCC1).

*: SCLK/3; SCLK is the internal system clock.

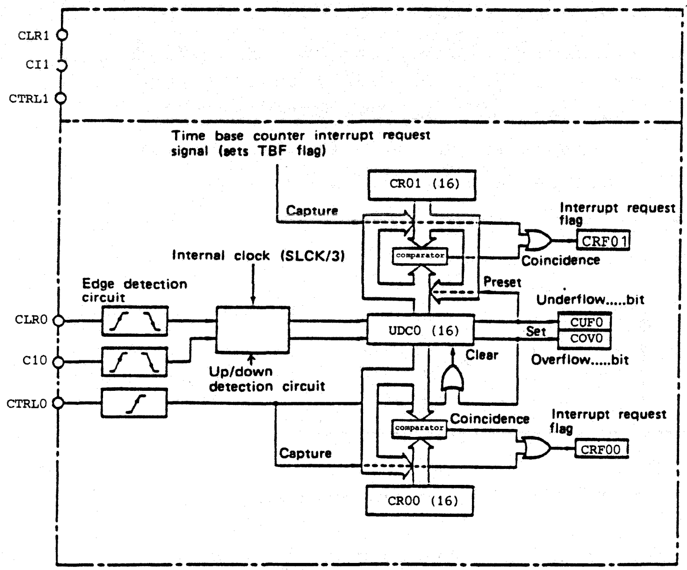


Fig. 3.11 Count Unit Block Diagram

(1) Up/down counter control register (UDCC0, UDCC1)

UDCC0 and UDCC1 registers are 8-bit registers which control the operation mode and count operation of the UDC0 and UDC1 presettable up/down counters.

(a) MOD bit

Bit 0 (MOD) of each UDCC0 and UDCC1 specifies the count operation mode for UDC0 and UDC1. An up/down counter performs the operation specified by the MOD bit regardless of the specification of the capture compare register control register (CRC).

° Mode 0 .... Normal mode (MOD=0)

The UDC0 (UDC1) register functions as a 16-bit binary counter which counts up or down the count clock selected by the $\overline{\text{I}}$/E bit according to the specification made by the $\overline{\text{U}}$/D bit.

° Mode 1 .... Up/down modulo mode (MOD=1)
In this mode, the UDC0 (UDC1) register is cleared when the contents of the UDC0 (UDC1) register and CR01 (CR11) register coincide. The value of the CR01 (CR11) is preset to the UDC0 (UDC1) register when the UDC0 counts down to 0000H. Refer to Fig. 3.12.

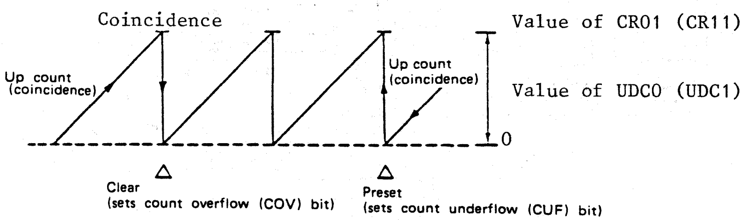


Fig. 3.12 Operation of Up/Down Modulo Mode Counter

An up/down counter register is cleared when the values of the capture compare register and presettable up/down counter coincide in the up count condition, and preset when the value of the presettable up/down counter becomes 0000H in the down count condition.

(b) ENCLR bit
Bit 2 of the UDCC0 (UDCC1) register, according to the CLR0 (CLR1) pin input, controls the clearing operation in which the UDC0 (UDC1) is cleared to 0000H. The UDCC0 can be cleared at the rising edge of the CLR0 (CLR1) pin signal if the ENCLR bit is set to 1.

(c) $\overline{\text{I}}$/E bit
Bit 3 ($\overline{\text{I}}$/E) of the UDCC0 (UDCC1) register specifies the count clock for the UDC0 (UDC1). The internal clock (SCLK/3) or external clock (clock input to CI0 or CI1 pin) is selected as the count clock.

(d) $\overline{\text{U}}$/D bit

Bit 4 ($\overline{\text{U}}$/D) of the UDCC0 (UDCC1) register specifies the initial condition (up count operation or down count operation) of count operation for the UDC0 (UDC1). The count operation specified by the $\overline{\text{U}}$/D bit continues unless the edge input (CTRL0, CTRL1 pin input) specified by the count unit input mode register is generated or the $\overline{\text{U}}$/D bit is inverted by software. The $\overline{\text{U}}$/D bit is reset to 0 when the counter is counting up, and set to 1 when the counter is counting down. Whether the counter is counting up or down can be determined by checking whether the $\overline{\text{U}}$/D bit is set or reset by means of software.

(e) CUF0/CUF1 bit

Bit 5 of the UDCC0 (UDCC1) indicates count underflow. This bit is set to 1 when the UDC0 (UDC1) counts down from FFFHH to 0000H or when the value of the CR01 (CR11) register is preset to the UDC0 (UDC1).

(f) COV0/COV1 bit

Bit 6 of the UDCC0 (UDCC1) indicates count overflow. This bit is set to 1 when the UDC0 (UDC1) counts up from FFFF to 0000H or when the values of the CR01 (CR11) register and UDC0 (UDC1) coincide.

(g) CS0/CS1 bit

Bit 7 of the UDCC0 (UDCC1) controls the count operation of the UDC0 (UDC1). The UDC0 (UDC1) starts count operation when this bit is set to 1, and stops count operation when this bit is reset to 0.

All bits of the UDCC0 (UDCC1) register are cleared to 0 when $\overline{\text{RESET}}$ is input.

UDCC0

7	6	5	4	3	2	1	0
CS0	COV 0	CUF 0	$\overline{U}$/D	$\overline{I}$/E	EN CLR	0	MOD

UDCC1

7	6	5	4	3	2	1	0
CS1	COV 1	CUF 1	$\overline{U}$/D	$\overline{I}$/E	EN CLR	0	MOD

Clear/preset function control (MOD)

0	The UDC0 (UDC1) is not cleared or preset.
1	The UDC0 (UDC1) is cleared when the values of the CR01 (CR11) reg. and UDC0 (UDC1) coincide (upcounting). The value of the CR01 (CR11) reg. is preset to the UDC0 (UDC1) when counter underflow of the UDC0 (UDC1) register occurs (downcounting)

CLR0/CLR1 pin function (EN CLR)

0	The UDC0 (UDC1) is not cleared.
1	The UDC0 (UDC1) is cleared at the rising edge of input.

Count clock specification ($\overline{I}$/E)

0	Internal clock (SCLK/3)
1	External clock (CI0/CI1 pin input)

Up/down operation flag ($\overline{U}$/D)

0	Up count operation
1	Down count operation

CUF: UDC0/UDC1 register count underflow flag

COV: UDC0/UDC1 register count overflow flag

Up/down count operation (CS)

0	Halted
1	Performed

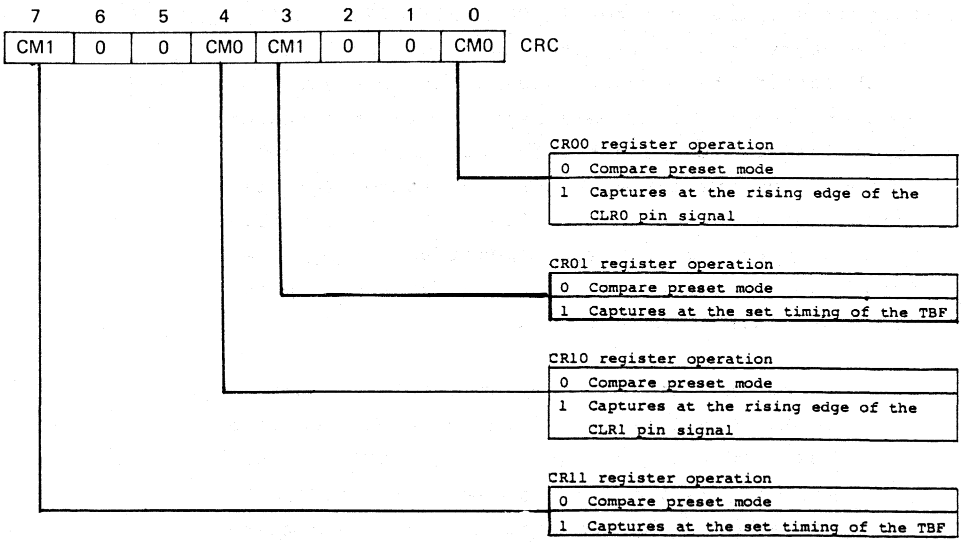
SCLK: Internal system clock

Fig. 3.13 Up/Down Counter Control Register Format

(2) Capture compare register (CR00, CR01, CR10, CR11)
The capture compare register functions as a register which retains the preset value of the presettable up/down counter, a compare register which compares its value with the count value of the presettable up/down count register, and as a capture register which clocks in the count value of the presettable up/down count register with specified timing. These functions can be specified by the capture compare register control register (CRC).

Note: Either the rising edge input to the external pin (CLR0, CLR1) or the timing by which the time base counter Int request flag (TBF) to 1 can be selected as a timinα trigger for the UDC0/1 value to be clocked into the capture compare register.

(3) Capture compare register control register (CRC)
The CRC register is an 8-bit register which controls the operation of the capture compare register (CR00, CR01, CR10, CR11). This register is cleared to 00H when $\overline{\text{RESET}}$ is input.



TBF: Time base counter Int request flag

Fig. 3.14 Format of Capture compare Register Control Register

3.2.1.2 Count unit operation

(1) When compare/preset mode is specified

The compare/preset mode can be obtained by resetting the CM bit of the capture/compare register control register (CRC) to 0.

(a) When the operation mode of the up/down counter UDC0 (UDC1) is set to mode 0 (MOD=0) by setting the up/down control register UDCC0 (UDCC1), the capture compare registers CR00 (CR10) and CR01 (CR11) compare the values retained in them and the count value of the up/down counter. Int request CRF00 (CRF10) and CRF01 (CRF11) are generated when they coincide.

(b) When the operation mode of the up/down counter UDC0 (UDC1) is set to mode 1 (MOD=1) by setting the up/down counter control register UDCC0 (UDCC1), the capture compare register CR00 (CR10) compares the value retained in it and the count value of the up/down counter. An Int request CRF00 (CRF10) is generated when they coincide.
Upon detecting a coincidence with the contents of the up/down counter, the capture compare register sets the count overflow flag (COV). When the count overflow flag is set to 1, the up/down counter UDC0 (UDC1) is cleared. When count underflow of the up/down counter UDC0 (UDC1) occurs, the capture compare register functions as a preset register which presets the value retained in it to the up/down counter.

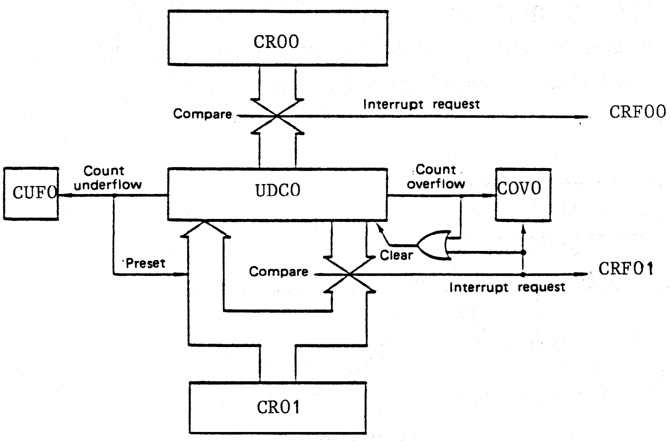


Fig. 3.15 When Compare/Preset Mode Is Specified

(2) When capture mode is specified
The capture/compare register is set to capture mode by setting the CM bit of the capture compare register control register (CRC) to 1.

The capture compare register performs capturing operations in which the count value of the up/down counter is clocked in and retained with specified timing. In the capture mode, an interrupt is generated with the timing at which the capturing is done.
The capture trigger for each capture compare register is as follows:

CR00 (CR10) ... Rising edge input to the CLR0 (CLR1) pin.
CR01 (CR11) ... Interrupt request generation from time base counter (TBF flag set timing).

In capturing by the CLR0 (CLR1) pin input, the function of this pin is not affected by the Port 3 mode register (PM3) and Port 3 mode control register (PMC3) settings. If the ENCLR bit of the up/down counter control register UDCC0 (UDCC1) is set to 1 to select the clear function of the up/down counter, the up/down counter is cleared after capturing is done.

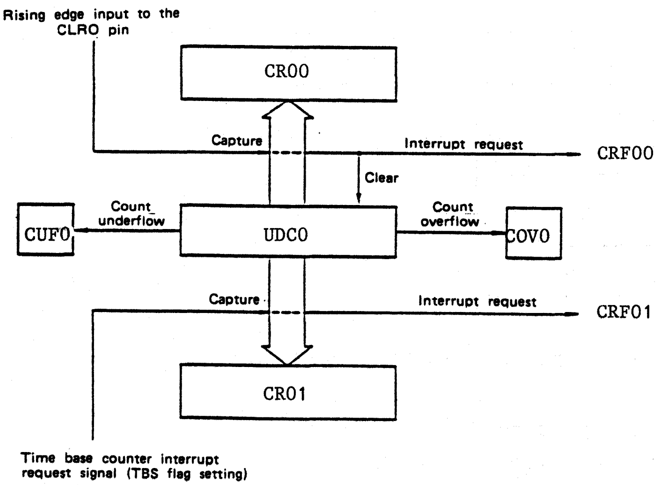


Fig. 3.16 When Capture Mode Is Specified

3.2.1.3 Count unit input mode register (CUIM)

This register controls the function of the count unit input pins (refer to Fig. 3.19).

(1) ES bit

Bits 4 and 0 of the CUIM register specify the effective edge of the CI pin inputs. Resetting the ES bit to 0 specifies the rising edge as an effective edge. When the ES bit is set to 1, both the rising and falling edges are specified as effective edges. Bit 0 is effective for the CI0 pin input, and bit 4 is effective for the CI1 pin input.

(2) CTRLM0 and CTRLM1 bits

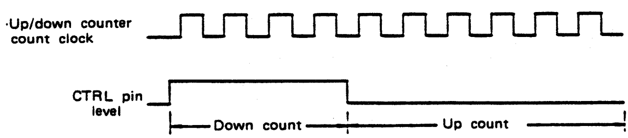
A bit field consisting of bits 7 and 6 and bits 3 and 2 specifies the CTRL pin function. Up/down switching of the corresponding register during count operation can be set as follows:

(a) Mode 0 (CTRLM1=0, CTRLM0=0)

In mode 0, the CTRL pin does not function as a CTRL pin but can be used as a general input port.

(b) Mode 1 (CTRLM1=0, CTRLM0=1)

In mode 1, the up/down counter counts down the count clock while the CTRL pin is at high level, and counts up while the CTRL pin is at low level.



(c) Mode 2 (CTRLM1=1, CTRLM0=0)

In mode 2, the operation of the up/down counter differs whether the internal or external clock is specified as the count clock.

° When internal clock is specified

When the effective edge (ES bit specification) is input to the CI0 (CI1) pin, the up/down counter is specified for an up counting operation, and counts up the internal clock. When the rising edge is input to the CTRL0 (CTRL1) pin, the up/down counter is specified for a down counting operation, and counts down the internal clock.

When the up/down counter is set to mode 2 by setting the count unit input mode register (CUIM) and the CS bit of the up/down counter control register to 1, the up/down counter counts up or down in the mode specified by the up/down counter control register (UDCC0, UDCC1). When the next effective edge is input to the CI0 (C11) and CTRL0 (CTRL1) pins, the count operation is automatically switched to up or down counting.

Example: Up/down switching operation when the rising edge is specified as an effective edge of C10 by resetting the ES bit of the count unit input mode register (CUIM) to 0 is shown in Fig. 3.17.

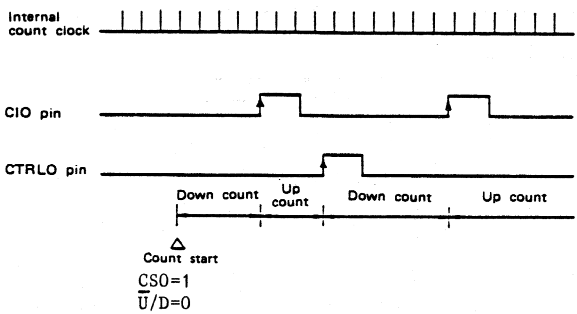


Fig. 3.17 Mode 2 Operation When Internal Clock Count Is Specified

° When external clock is specified
Effective edge inputs (ES bit specification) to the CI0 (CI1) pin are counted up, and rising edge inputs to the CTRL0 (CTRL1) pin are counted down.

This mode can be selected by the count input mode register (CUIM). When the CS bit of the up/down counter control register is set to 1, the up/down counter starts counting in the mode specified by its control register. The CI0 (CI1) and CTRL0 (CTRL1) pin inputs are then counted up and down, respectively. When the CI0 (CI1) and CTRL0 (CTRL1) pins input the count clock simultaneously, count operation will not be performed but the previous count value is retained.

Example: Up/down counter operation when the rising edge of CI0 pin is specified as an effective edge by resetting the ES bit of the count unit input mode register (CUIM) to 0 is shown in Fig. 3.18.

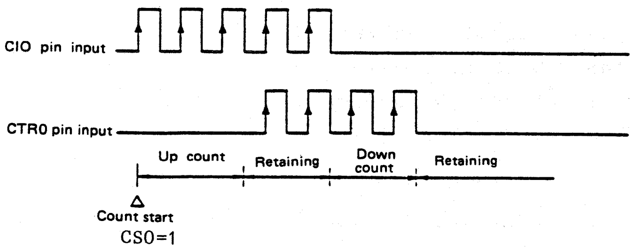


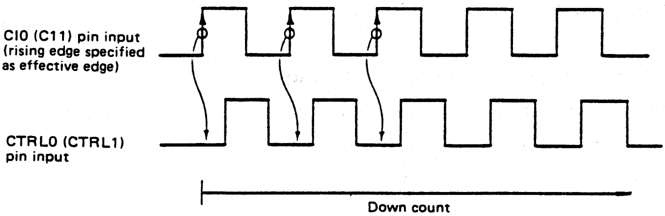
Fig. 3.18 Mode 2 Operation When External Clock Is Specified As Count Clock

(d) Mode 3 (CTRLM1=1, CTRLM0=1)
Mode 3 is most effective when a 2-phase signal of which the phase angle is 90 degrees such as servo motor shaft output, is input to the CI0 (CI1) and CTRL0 (CTRL1) pins as the count clock. The count

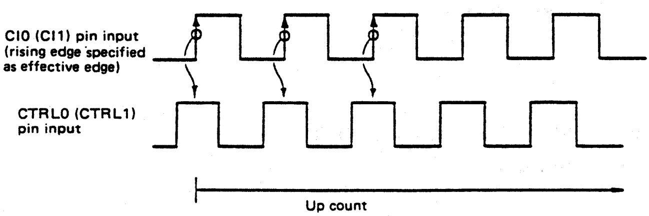
unit automatically switches the up/down count operation of the up/ down counter by detecting the relative phase lead or delay of this 2-phase signal.

When a 2-phase signal having 90 degrees phase differential is input to the CI0 (CI1) and CTRL0 (CTRL1) pins, the CTRL0 (CTRL1) pin level is sampled at the effective edge (specified by the count unit input mode register) of the CI0 (CI1) pin input.

If the CTRL0(CTRL1) pin level sampled at the effective edge input to the CI0 (CI1) pin is low, the up/down counter counts down the effective edge input to the CI0 (CI1) pin.



If the CTRL0 (CTRL1) pin level sampled at the effective edge input to the CI0 (CI1) pin is high, the up/down counter counts up the effective edge input to the CI0 (CI1) pin.



7	6	5	4	3	2	1	0	
CTRLM 1	CTRLM 0	0	ES	CTRLM 1	CTRLM 0	0	ES	CUIM

Effective edge specification for CI0 pin input

0	Rising edge
1	Both rising and falling edge

CTRLM 1	CTRLM 0	Specifies function of CTRL0 pin input
0	0	Does not function as CTRL0 pin
0	1	When CTRL0=1: Down count When CTRL0=0: Up count
1	0	Counts up at effective edge input to CI0 pin Counts down at rising edge input to CTRL0 pin
1	1	CTRL0 pin level is sampled at effective edge input to CI0 pin to automtically determine up/down counting

Effective edge specification for CI1 pin input

0	Rising edge
1	Both rising and falling edges

CTRLM 1	CTRLM 0	Specifies function of CTRL1 pin input
0	0	Does not function as CTRL1 pin
0	1	When CTRL1=1: Down count When CTRL1=0: Up count
1	0	Counts up at effective edge input to CI1 pin Counts down at rising edge input to CTRL1 pin
1	1	CTRL0 pin level is sampled at effective edge input to CI1 pin to automatically determine up/down counting

Fig. 3.19 Format of Count Unit Input Mode Register

3.2.1.4 Count unit interrupt request control register (CRIC00, CRIC01, CRIC10, CRIC11)

This is an 8-bit register which controls four interrupt requests (CRF00, CRF01, CRF10, CRF11) generated from the count unit. These four interrupt requests form a group, and priority order of these interrupt requests can be specified by program. Within a group, priorities of these interrupt requests are fixed by means of hardware as follows:

CRF00>CRF01>CRF10>CRF11

The functions of bits 0 to 6 of the interrupt request control register are explained in Chapter 4, Interrupt Request Function. Bit 7 functions as the interrupt request flag. The set condition of each interrupt request flag is indicated in Table 3.5. An interrupt request flag is reset to 0 when an interrupt request is accepted or by means of software.

7	6	5	4	3	2	1	0	
CRF 00	CRMK 00	MS/$\overline{\text{INT}}$	EN CS	0	PR2	PR1	PR0	CRIC00
CRF 01	CRMK 01	0	EN CS	0	-	-	-	CRIC01
CRF 10	CRMK 10	MS/$\overline{\text{INT}}$	EN CS	0	-	-	-	CRIC10
CRF 11	CRMK 11	0	EN	0	-	-	-	CRIC11

Fig. 3.20 Count Unit Interrupt Request Control Register Format

Table 3.5 Count Unit Interrupt Request Flag Set Condition

Interrupt request flag	Operation mode	Operation mode of up/down counter	Count operation	Interrupt request flag set condition
CRF00	Capture	--	--	When the value of the UDC0 is captured to the CR00 at the rising edge of the CLR0 pin.
	Compare/ preset	--	--	When the contents of the CR00 and UDC0 coincide.
CRF01	Capture	--	--	When the value of the UDC0 is captured to the CR01 by the time base Int request generation (TBF setting).
	Compare/ preset	Mode 0 (normal mode)	--	When the contents of the CR01 and UDC0 coincide.
		Mode 1 (up/down modulo mode)	Down	When the contents of the CR01 is preset to the UDC0.
			Up	When the contents of the CR01 and UDC0 coincide, and the value of the UDC0 is cleared.
CRF10	Capture	--	--	When the value of the UDC1 is captured to the CR10 at the rising edge input at the CLR1 pin.
	Compare/ preset	--	--	When the contents of CR10 and UDC1 coincide.
CRF11	Capture	--	--	When the value of the UDC1 is captured to the CR11 by the time base Int request generation (TBF setting).
	Compare/ preset	Mode 0 (normal mode)	--	When the contents of CR11 and UDC1 coincide.
		Mode 1 (up/down modulo mode)	Down	When the contents of the CR11 is preset to the UDC1.
			Up	When the contents of the CR11 and UDC1 coincide, and the value of the UDC1 is cleared.

3.2.1.5 Count unit macro service control register (CRMS00, CRMS10)

Of the four interrupt requests generated from the count unit, CRF00 and CRF10 can generate macro service requests. CRMS00 and CRMS10 are 8-bit registers which control the macro service requests generated by setting the CRF00 and CRF10 to 1, respectively. The function of each bit is explained in the section of macro service function.

	7	6	5	4	3	2	1	0
CRMS00	MSM2	MSM1	MSM0	DIR	0	CH2	CH1	CH0

	7	6	5	4	3	2	1	0
CRMS10	MSM2	MSM1	MSM0	DIR	0	CH2	CH1	CH0

Fig. 3.21 Macro Service Control Register Format

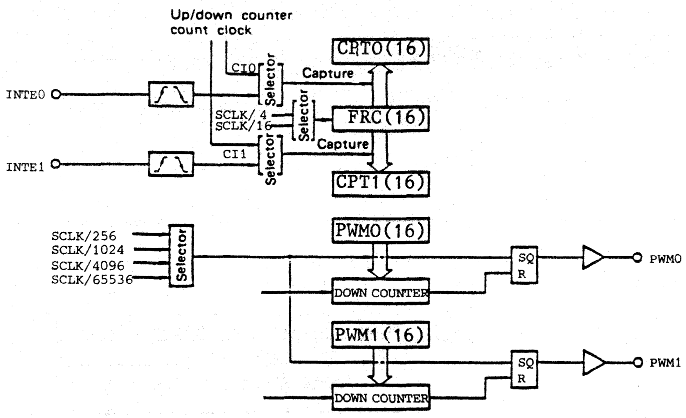
3.2.2 Capture/PWM unit

3.2.2.1 Configuration of capture/PWM unit

The capture/PWM unit has free running counter capture and PWM output functions. This unit consists of a 16-bit free running counter (FRC), 16-bit capture registers (CPT0, CPT1), 16-bit PWM registers(PWM0, PWM1), control registers (CPTM, PWMM, FRCC), and down counters.

The free running counter is also used as a 20-bit time base counter which counts the internal system clock (SCLK). The free running counter cannot be cleared to 00H other than by inputting $\overline{\text{RESET}}$.

The capture registers (CPT0, CPT1) function as capture registers for the FRC. The count value of the FRC is clocked into the CPT0 (CPT1) register using the count clock input of the count unit or effective edge input to the INTE0 (INTE1) pin as a trigger. This is not affected by $\overline{\text{RESET}}$ input. The PWM register (PWM0, PWM1) controls pulse width (duty) when in the PWM output mode. This is not affected by $\overline{\text{RESET}}$ input.



Note: SCLK: Internal system clock

Fig. 3.22 Capture/PWM Unit Block Diagram

(1) Capture mode register (CPTM), PWM mode register (PWMM)

These registers specify the operation of the capture registers(CPT0, CPT1) and PWM registers(PWM0, PWM1). These registers are cleared to 00H when RESET is input.

(a) MOD0 and MOD1 bits

Each of these bits specifies the operation mode of the corresponding register. Capture mode/PWM mode can be specified by MOD0 and MOD1 bits. However, the CPT register cannot be specified for the PWM mode; neither can the PWM register be specified for the capture mode.

(b) CT0 and CT1 bits

The functions of the CT0 and CT1 bits differ depending on the operation mode of the corresponding register. For the CPT register, these bits specify the capture trigger. For the PWM register, these bits specify the number of effective bits (resolution).

7	6	5	4	3	2	1	0
CT1	0	0	MOD0	CT1	0	0	MOD0

MOD0	Operation of CPT0 register
0	No operation
1	Operates as capture register

CT1	CPT0 register capture trigger specification
0	Set timing of external interrupt flag (EXIF0)
1	UDC0 count clock (CI0)

MOD0	Operation of CPT1 register
0	No operation
1	Operates as capture register

CT1	CPT1 register capture trigger specification
0	Set timing of external interrupt flag (EXIF1)
1	UDC1 count clock (CI1)

Fig. 3-23 Formats of Capture Mode Register and PMW Mode Register

7	6	5	4	3	2	1	0	
CT1	CT0	MOD1	0	CT1	CT0	MOD1	0	PWMM

MOD1	Operation of PWM0 register
0	No operation
1	Operates as PWM register

CT1	CT0	Specifies the accuracy of PWM output signal of PWM1 register
0	0	8-bit
0	1	10-bit
1	0	12-bit
1	1	16-bit

MOD1	Operation of PWM1 register
0	No operation
1	Operates as PWM register

CT1	CT0	Specifies the accuracy of PWM output signal of PWM1 register
0	0	8-bit
0	1	10-bit
1	0	12-bit
1	1	16-bit

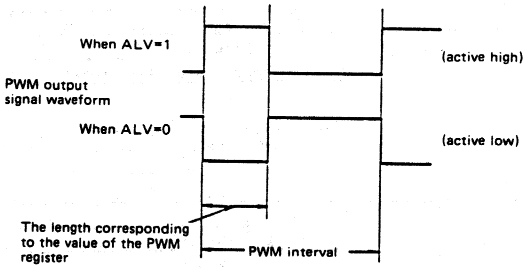
Fig. 3.24

(2) Free running counter control register (FRCC)

The FRCC register is an 8-bit register which controls the FRC (free running counter) and PWM output. The FRCC register is cleared to 00H when $\overline{\text{RESET}}$ is input.

(a) ALV bit (bits 2 and 0)

The ALV bit specifies the effective level of PWM0/PWM1 output to either active low or active high.



(b) ENPWM0 bit, ENPWM1 bit (bits 1 and 3)

These bits enable PWM output. The port is at the inactive level ($\overline{\text{ALV}}$) in PWM output disabled state.

(c) CCLK bit (bit 4)

Of the 20 bits of the time base counter (TBC), 16 bits for capture operation are specified as the free running counter (FRC) by the CCLK bit. By setting (to 1) and resetting (to 0), the capture register (CPT0, CPT1) captures 16-bit data as shown in Fig. 3-24. On software, it looks like the count clock of the free running counter (FRC) is switched. This is not affecting the PWM operation.

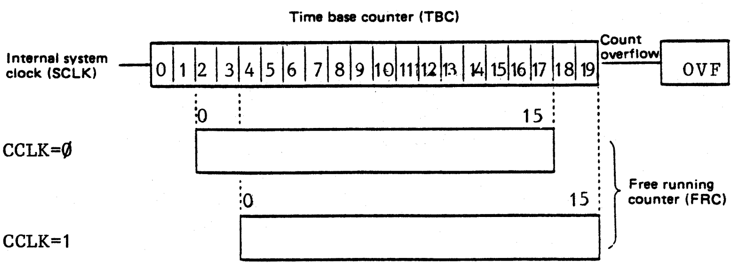


Fig. 3-24 Free Running Counter Count Clock Specification

(d) OVF bit

This bit indicates FRC overflow. The OVF flag is set to 1 when the 16 bits specified by the CCLK bit overflow. The OVF flag, if once set to 1, cannot be reset to 0 by hardware even if the CCLK bit is switched. The OVF bit can be reset to 0 only by software.

7	6	5	4	3	2	1	0	
OVF	0	0	CCLK	EN PWM1	ALV	EN PWM0	ALV	FRCC

ALV	PWM0 output active level specification
0	Low level
1	High level

EN PWM0	PWM0 output control
0	PWM output disabled
1	PWM output enabled

ALV	PWM1 output active level specification
0	Low level
1	High level

EN PWM1	PWM1 output control
0	PWM output disabled
1	PWM output enabled

CCLK	Free running counter (FRC) count clock specification
0	Internal system clock (SCLK) x 1/4
1	Internal system clock (SCLK) x 1/16

Free running counter (FRC) overflow flag

Fig. 3-25 Format of Free Running Counter Control Register

3.2.2.2 Operation of capture/PWM unit

(1) Capture operation

The CPT0/CPT1 register clocks in and retains the 16-bit data specified by the CCLK bit of the free running counter register (FRCC) when the capture trigger specified by the corresponding CT1 bit occurs.

When the external interrupt flag (EXIF0, EXIF1) is specified as the capture trigger, a capture operation is performed each time the effective edge is input to the external interrupt pin (INTE0, INTE1).

When the count clock C10/C11 is specified as the capture trigger, the count clock is used as the capture trigger regardless of the count clock source specification (internal or external) and count operation (up count or down count).

(2) PWM operation

The PWM0/PWM1 register, according to the effective bit specification by bits CT0 to 1, sets the PWM output to active level at a timimg at which the time base counter tap output (TBC8, TBC10, TBC12, TBC16) becomes high level, and presets the PWM0 (PWM1) register value to the down counter, then starts down counting. When a count underflow occurs in down counting, the down counter stops down counting and sets the PWM output to the inactive state. In this way, the pulse width (duty) of the PWM output is controlled.

Example: PWM output when time base counter output
tap is specified to TBC8

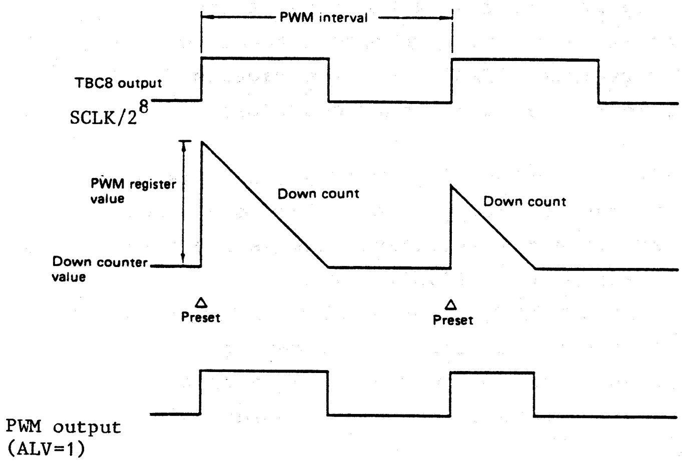


Fig. 3.26 PWM Output Operation Timing

Table 3.6 PWM Interval Setting

CT1	CT0	PWM accuracy	Frequency	PWM interval
0	0	8-bit PWM output	23.4KHz	42μs
0	1	10-bit PWM output	5.9KHz	169μs
1	0	12-bit PWM output	1.5KHz	714μs
1	1	16-bit PWM output	91.6Hz	10ms

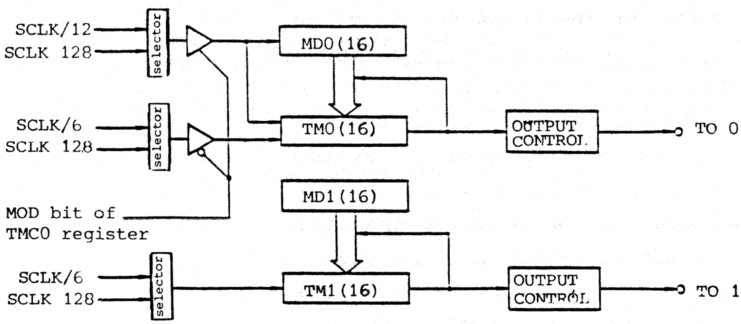
SLCK=6MHz

3.2.3 Timer unit

3.2.3.1 Timer unit configuration

The timer unit can be used as an interval timer, one-shot timer, and a timer that controls the output of the square-wave output and real-time output ports.

As shown in Fig. 3-27, the unit consists of two 16-bit timer registers, two 16-bit modulo/timer registers, and an 8-bit timer control register.



Note: SCLK: Internal system clock

Fig. 3-27 Timer Unit Block Diagram

(1) Timer control register (TMC0, TMC1)

The TMC0 and TMC1 registers are 8-bit registers. The TMC0 register controls the operation of the TM0 and MD0 registers. The TMC1 register controls the operation of the TM1 and MD1 registers. The TMC0 and TMC1 registers can be read and written. These registers are cleared to 00H when $\overline{\text{RESET}}$ is input. The formats of TMC0 and TMC1 registers are different.

(a) The MOD bit of the TMC register specifies the operation mode of the TM0 and MD0 registers. When this bit is reset to 0, interval timer operation mode is assumed; the TM0 functions as a timer register which counts down the set value, and the MD0 register functions as a modulo register which retains the interval set value. When the MOD bit is set to 1, one-shot timer operation mode is assumed; both the TM0 and MD0 function as timer registers which count down the set value.
The TRG bit determines whether or not the timer can be started by the effective edge input to the INTE2 pin. When the TRG bit is reset to 0, the timer cannot be started by the effective edge input to the INTE2 pin. When this bit is set to 1, the timer can be started by the effective edge input to the INTE2 pin.

The TCLK0 bit specifies the count clock for the TM0 register. The TM0 register uses a different count clock depending on whether the TM0 is set to the one-shot mode or interval timer mode. Reference values when the 6MHz internal system clock is used are shown in Table 3-7.
The TS0 bit controls the TM0 register operation. If the TS0 bit is set to 1 when the TM0 register is set to the interval timer mode (MOD=0), the MD0 register value is set to the TM0 register, the TM0 register then counts down the value with the count clock specified by the TCLK0 bit. If the external trigger is specified as a timer start, the TS0 bit is set to 1 by means of hardware when the effective edge is input to the INTE2 pin. The TM0 register then starts counting down. When the TM0 register is set to the one-shot timer mode (MOD=1), the TS0 bit is set to 1 by means of hardware when data is written to the TM0 register. The TM0 register then starts counting down. When the count value reaches 0, the TS0 bit is reset to 0, and counting stops.

Note: If one attempts to set the TSØ bit to 1 again when the TS0 bit is set to 1, the MD0 register value is again set to the TM0 register, and the timer is again started.

The MCLK bit specifies the count clock for the MD0 register; however, this bit is effective only when the MD0 register is set to the one-shot timer mode. Reference values when the 6MHz internal system clock is used are shown in Table 3-8. When the MD0 register is set to the interval timer mode, the MCLK bit has no effect on count operation.

The MS0 bit is effective only when the MD0 register is set to the one-shot timer mode, and controls the count operation of the MD0 register. The MS0 bit is set to 1 by means of hardware when data is written to the MD0 register. Count operation then starts. When the MD0 register value reaches 0, the MS0 bit is automatically reset to 0 to stop count operation. The MS0 bit has no effect on count operation in the interval timer mode.
The ENTO0 bit enables square-wave output to the TO0 pin. When the ENTO0 bit is reset to 0, the TO0 pin becomes inactive. When the ENTO0 bit is set to 1, the TO0 pin level is inverted at the timing at which the timer flag TMF0 is set. The ALV bit specifies the active level of the TO0 pin output; low level active when the ALV bit is reset to 0, and high level active when it is set to 1.

Table 3-7 Timer Register 0 (TM0) Count Time

SCLK=6MHz

TCLK0	Interval timer mode			One-shot timer mode		
	Count clock	Resolution	Full count	Count clock	Resolution	Full count
0	SCLK/6	1.0μs	65.5ms	SCLK/12	2.0μs	131.3ms
1	SCLK/128	21.3μs	1.4s	SCLK/128	21.3μs	1.4s

Table 3-8 Modulo/Timer Register 0 (MD0) Count Time in One-Shot Timer Mode

SCLK=6MHz

MCLK0	Count clock	Resolution	Full count
0	SCLK/12	2.0μs	131.1ms
1	SCLK/128	21.3μs	1.4s

(c) The TCLK bit of the TMC1 register specifies the count clock of the TM1 register. Reference values when the 6MHz internal system clock is used are shown in Table 3-9. The TS1 bit controls timer operation.

When the TS1 bit is set to 1, the MD1 register value is set to the TM1 register, and the TM1 register value is counted down with the count clock specified by the TCLK bit. When the TS1 bit is reset to 0, countdown stops, and the TM1 and MD1 register values are retained.

The ENTO1 bit enables squarewave output to the TO1 pin. When the ENTO1 bit is reset to 0, the TO1 pin level is fixed to the inactive level. If the ENTO1 bit is set to 1, the TO1 pin level is inverted at the timing at which the TMF2 is set.

The ALV bit specifies the active level of the TO1 pin output.

Table 3–9 Timer Register 1 (TM1) Count Time (interval timer mode)

SCLK=6MHz

TCLK1	Count clock	Resolution	Full count
0	SCLK/6	1.0µs	65.5ms
1	SCLK/128	21.3µs	1.4s

7	6	5	4	3	2	1	0	
TS0	TCLK0	MS0	MCLK0	EN TO0	ALV	TRG	MOD	TMC0

Operation mode specification (MOD)

0	Interval timer mode
1	One-shot timer mode

Timer operation control (TRG)

0	Timer start by INTE2 pin input disabled
1	Timer start by effective edge input to INTE2 pin

TO0 pin effective level specification (ALV)

0	Sets TO0 pin output active level to low level
1	Sets TO0 pin output active level to high level

TO0 pin operation specification (EN TO0)

0	Fixes TO0 pin level to inactive (ALV)
1	Inverts TO0 pin level at TMF0 set timing

MD0 register clock source specification (in one-shot time mode) (MCLK0)

0	SCLK/12 is specified as count clock of MD0 register
1	SCLK/128 is specified as count clock of MOD register

MD0 register operation control (in one-shot timer mode) (MS0)

This bit is set when data is written to the MD0 and reset at the timing at which the MD0 reaches 00H. Counting starts when this bit is set to 1, and stops when this bit is reset to 0.

TM0 register clock source specification (TCLK0)

In interval timer mode	0	SCLKK/6 is specified as TM0 count clock
	1	SCLK/128 is specified as TM0 count clock
In one-shot timer mode	0	SCLK/12 is specified as TM0 count clock
	1	SCLK/128 is specified as TM0 count clock

TM0 register operation control (TS0)

0	Stops counting
1	Starts counting

Fig. 3-28 Timer Control Register (TMC0, TMC1) Format

7	6	5	4	3	2	1	0	
TS1	TCLK1	0	0	EN TO1	ALV	0	0	TMC1

ALV	TO1 pin active level specification
0	Low level
1	High level

EN TO1	TO1 pin operation specification
0	Fixes TO1 pin output level to inactive (ALV)
1	Inverts TO1 pin output level at TMF2 set timing

TCLK1	TM1 register clock source specification
0	Specifies SCLK/6 as count clock
1	Specifies SCLK/128 as count clock

TS1	TM1 register control
0	Stops counting
1	Starts counting

Fig. 3-28 Timer Control Register (TMC0, TMC1) Format (2/2)

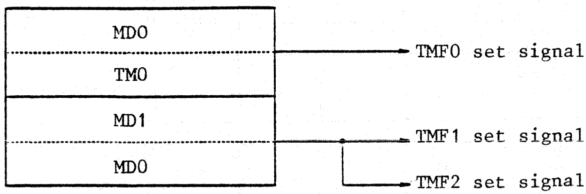
3.2.3.2 Timer unit interrupt request

Three interrupt requests (TMF0-2) are generated from the timer unit. The condition in which interrupts are generated from the timer unit differs depending on the timer operation mode specification.

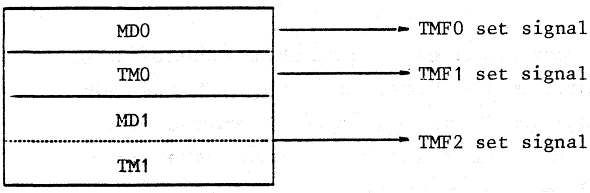
When the TM0 and MD0 are set to interval timer mode, the TMF0 is set to 1 when an underflow occurs by the TM0 register countdown, and the TMF1 and TMF2 are set to 1 when an underflow occurs by the TM1 register count down. Refer to Fig. 3-29a.

In one-shot timer mode, the TMF0 is set to 1 when an underflow occurs by the MD0 register countdown, and the TMF1 is set to 1 when an underflow occurs by the TM0 register countdown. In this case, the TMF2 is set to 1 when an underflow occurs in the countdown by the TM1 register functioning as an interval timer.

a. When TM0 and MD0 are set to internal timer mode



b. When TM0 and MD0 are set to one-shot timer mode



TMF0-2: Time unit interrupt request flags 0 to 2

Fig. 3-29 Interrupt Request from Timer Unit

(1) Timer unit interrupt request control register (TMIC0, TMIC1, TIMIC2)

The TMIC0 to TMIC2 registers control three interrupt requests generated from the timer unit. These three interrupt requests form a group as the timer unit interrupt request to which a priority is specified by program. Within the group, priority order is fixed by means of hardware as follows:

TMF0>TMF1>TMF2

Each bit of the TMIC0-2 registers is explained in Chapter 4, Interrupt Request Functions.

	7	6	5	4	3	2	1	0
TMIC0	TMF0	TM MK0	MS /$\overline{\text{INT}}$	EN CS	0	PR2	PR1	PR0
TMIC1	TMF1	TM MK1	MS /$\overline{\text{INT}}$	EN CS	0	-	-	-
TMIC2	TMF2	TM MK2	MS /$\overline{\text{INT}}$	EN CS	0	-	-	-

Fig. 3-30 Timer Unit Interrupt Request Control Register Format

(2) Timer unit macro service control register (TMMS0 to 2)
This register controls the macro service activated by the three types of interrupt requests generated from the timer unit. The TMMS0 register controls the macro service activated by the TMF0 flag. The TMMS1 and TMMS2 registers control macro services activated by the TMF1 and TMF2 flags, respectively.

7	6	5	4	3	2	1	0
MSM2	MSM1	MSM0	DIR	0	CH2	CH1	CH0

Fig. 3-31 Macro Service Control Register Format

Each bit of the TMMS0-2 registers is explained in section 4.2, Macro Service Function.

3.2.4 Real-time output port function

Real-time output port function is built into Port 0 and can output the contents of the Port 0 buffer register with programmable intervals in 4- or 8-bit width.
As shown in Fig. 3-32, the real-time output port is configured in a master-slave configuration and consists of a buffer register, output latch, and the RTPC register that controls port outputs.

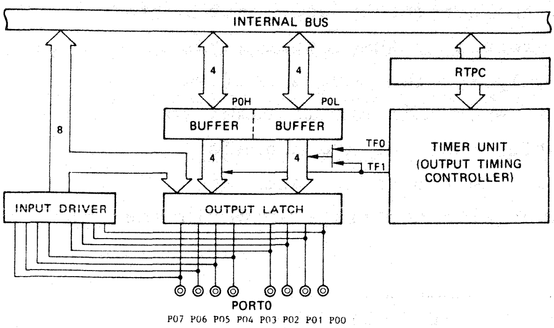


Fig. 3-32 Real Time Output Port Block Diagram

(1) Port 0 buffer register

When Port 0 is set to the real-time output port mode, the buffer register retains the data to be output next. The contents of the buffer register are transferred to the output latch when the corresponding interrupt request flag (TMF0-1) of the timer unit is set to 1. The contents of the buffer register are not affected by $\overline{\text{RESET}}$ input.

(2) Output latch

The output latch retains the port output data.

(3) Real-time output port control register (RTPC)

This is an 8-bit register which specifies the Port 0 operation mode. When $\overline{\text{RESET}}$ is input, the real-time output port control register (RTPC) is set to 08H and Port 0 is set to the input/output port mode.

The P0LM and P0HM bits specify the operation mode of the lower 4 bits and higher 4 bits of port 0 respectively. When the P0LM and P0HM bits are reset to 0, Port 0 becomes a normal three-state input/output port. When they are set to 1, Port 0 functions as a real-time output port.

The BYTE bit is set when the higher 4 bits and lower 4 bits of Port 0 are not used as independent 4-bit real-time output ports but used as 1-byte real-time output ports.

The TRG bit* specifies the trigger by which the buffer register value is transferred to the output latch. This bit is effective only for the port specified as a real-time output port. When this bit is reset to 0, the contents of the buffer register are transferred to the output latch with the timing at which each flag is set in accordance with the flags shown in the table below:

TRG bit	Real-time output port	BYTE=0	BYTE=1
0	POL	TMF0	TMF1
	POH	TMF1	

* When the TRG bit is set to 1, the value of the register specified as the real-time output port is transferred to the output latch at its set timing. The TRG bit retains 1 after it is set.
  This can be regarded as a software-trigger for this realtime output.

7	6	5	4	3	2	1	0	
0	0	0	0	BYTE	TRG	POHM	POLM	RTPC

POHM	POLM	Port 0 function specification
0	0	P00-P07: Three-state input/ output port
0	1	P00-P03: Real-time output port P04-P07: Three-state input/ output port
1	0	P00-P03: Three-state input/ output port P04-P07: Real-time output port
1	1	P00-P07: Real-time output port

TRG	Real-time output port transfer timing specification
0	The contents of the buffer register are transferred at the TFMn flag set timing (POL is by TMF0. POH is by TMF1. When the POL and POH are paired as an 8-bit register, TMF1 set timing is used).
1	Transfers when 1 is written to this bit.

BYTE	Real-time output port operation mode
0	4-bit independent real-time out-put port
1	8-bit real-time output port

Fig. 3-33 Real-Time Output Port Control Register Format

3.3 A/D Converter Function

The µPD78312/78310 has a built-in 8-bit high-speed high-accuracy analog-to-digital (A/D) converter having four multiplexed analog inputs (AN3-0). The A/D converter uses the consecutive comparison method, and is provided with an A/D conversion result register (ADCR) that retains the result of the conversion. Refer to Fig. 3-34.

The A/D converter can be operated in either the scan or select mode; this can be selected by software.

In the select mode, one analog input is selected by the A/D channel mode register (ADM) to perform A/D conversion. The result of the conversion is stored to the ADCR register and the ADF flag* is then set to 1.

In the scan mode, analog inputs are converted in the order specified by the ADM register. If bit 2 (ANI1) and bit 1 (ANI0) are set to 0 and 1, respectively, analog inputs are selected in this order; AN0 → AN1 → AN0. After each input is selected, the result of the conversion is stored to the ADCR register, and the ADF flag is then set to 1.

The contents of the ADCR register are not affected by $\overline{\text{RESET}}$ input.

* The ADF flag is bit 7 of the ADCR register which is set after the conversion operation; however, this flag has no effect on conversion operation.

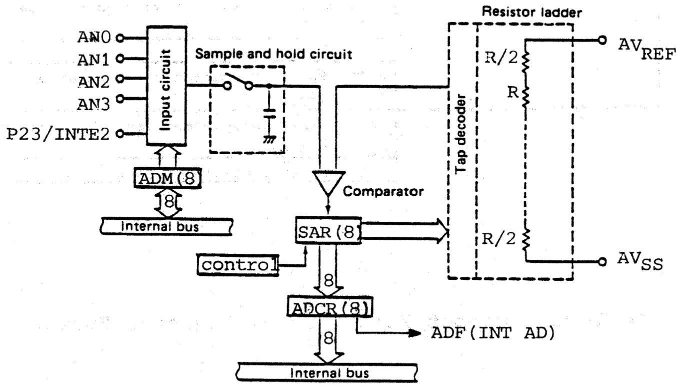


Fig. 3-34 A/D Converter Block Diagram

(1) A/D channel mode register (ADM)

The A/D channel mode register is an 8-bit mode register which controls the operation of the A/D converter. The MS bit specifies the operation mode of the A/D converter. Resetting the MS bit to 0 sets the A/D converter to the scan mode, and setting this bit to 1 sets the A/D converter to the select mode.

In the scan mode, the ANI0 and ANI1 bits specify the analog inputs to be scanned. In the select mode, these bits specify the analog input to be A/D converted.

The FR bit is a control bit which prevents the A/D conversion time from varying considerably, even if the oscillation frequency is changed.

The TRG bit enables external synchronization in an A/D conversion operation. If the TRG bit is reset to 0, conversion operations are consecutively performed. However, if this bit is set to 1, A/D conversion starts when the effective edge is input to the external trigger pin (INTE2). Conversion sequence is initialized each time the effective edge is input to the INTE2 pin.

The CS bit controls the A/D conversion operation. Conversion starts when this bit is set to 1, and conversion stops when this bit is reset to 0, even if a conversion is being performed. In this case, the contents of the ADF register are not modified. Neither can the ADF flag be set.

The ADM register is cleared to 00H when $\overline{\text{RESET}}$ is input.

7	6	5	4	3	2	1	0	
CS	TRG	0	FR	0	ANI1	ANI0	MS	ADM

AN1	AN0	MS	A/D conversion mode specification	
0	0	0	Scan mode	Scans AN0 input
0	1	0		Scans AN0 and AN1 inputs
1	0	0		Scans AN0, AN1, and AN2 inputs
1	1	0		Scans AN0, AN1, AN2, and AN3 inputs
0	0	1	Select mode	Selects AN0 input
0	1	1		Selects AN1 input
1	0	1		Selects AN2 input
1	1	1		Selects AN3 input

FR	Conversion time switching
0	180 states
1	120 states

TRG	External pin trigger control
0	External trigger disabled
1	External trigger enabled

CS	A/D conversion operation control
0	Stops A/D conversion
1	Starts A/D conversion

Fig. 3-35 A/D Conversion Mode Register Format

(2) A/D converter interrupt request control register (ADIC)
This is an 8-bit register which controls the interrupt request generated from the A/D converter. The interrupt request from the A/D converter and one from the interrupt request control register (TBIC) of the time base counter (TBC)* form a group of interrupt requests. The priority of the group can be specified by software. Within the group, the priority is fixed by means of hardware as follows:

ADF (A/D conversion interrupt request) > TBF (time base counter interrupt request)

Each bit of the interrupt request control register is explained in Chapter 4, Interrupt Request Process Functions.

* Refer to 3.6, Time Base Counter/Watchdog Timer Function.

7	6	5	4	3	2	1	0	
ADF	AD MK	MS/$\overline{INT}$	EN CS	0	PR 2	PR 1	PR 0	ADIC
TBF	TB MK	0	EN CS	0	-	-	-	TBIC

Fig. 3-36 Interrupt Request Control Register Format

(3) A/D converter macro service control register (ADMS)
This register controls the macro service activated by an interrupt request generated from the A/D converter. Each bit of the ADMS register is explained in Section 4.2, Macro Service Function.

7	6	5	4	3	2	1	0
MSM2	MSM1	MSM0	DIR	0	CH2	CH1	CH0

Fig. 3-37 Macro Service Control Register Format

3.4 Serial Communication Interface

3.4.1 Serial communication interface configuration

The µPD78312/78310 is provided with a serial communication interface that has a built-in dedicated baud rate generator.

Start/stop bit transmit/receive method is used in this serial communication interface. This interface can operate in two modes: asynchronous mode, in which bit synchronization and character synchronization of bit are obtained by the start bit; and I/O interface mode, in which data is transferred in synchronization with the serial clock controlled in the same manner as the serial data transfer of such as the µCOM-87 family.

The serial communication interface block consists of a serial data input (RxD), a serial data output (TxD), a serial clock output ($\overline{SCK}$), transmit enable control input ($\overline{CTS}$) pins, and a transfer control circuit, an 8-bit serial register for transmit, an 8-bit serial register for receive, a transmit buffer, a receive buffer, and a baud rate generator. Serial registers and buffers are separately provided for transmit and receive; therefore, transmit and receive can be independently performed. In the I/O interface mode, the $\overline{CTS}$ pin functions as a receive clock input/output pin; therefore, full-duplex communications are possible even in the I/O interface mode.

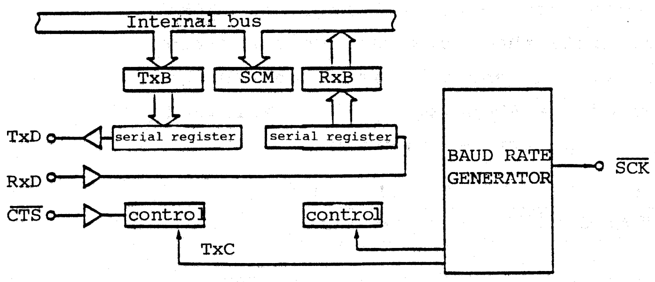


Fig. 3-38 Serial Communication Interface Block Diagram

(1) Serial communication mode register (SCM)

This is an 8-bit register which specifies the transfer mode of the serial interface. The functions of bits 2 to 7 are determined by bits 1 (MD1) and 0 (MD0).

	7	6	5	4	3	2	1	0
SCM							MD1	MD0

MD1, MD0 = 0, 1 (asynchronous mode)

TxRDY	RxE	PEN	EP	CL	SL	0	1

MD1, MD0 = 0, 0 (I/O interface mode)

TxE	RxE	0	0	TSK	RSCK	0	0

The bit field consisting of the MD1 and MD0 bits specifies the transfer mode of the serial interface. When the MD1 and MD0 are set to 0 and 1, the serial communication interface is set to the asynchronous mode. When the MD1 and MD0 are set to 0 and 0, the serial communication interface is set to the I/O interface mode. When $\overline{RESET}$ is input, this register is cleared to 00H, and the serial communication interface is set to the I/O interface mode.

(a) When set to asynchronous mode

The RxE bit determines whether or not the receive operation is enabled in the asynchronous and I/O interface modes. If this bit is set to receive disable during a receive operation, the receive operation is interrupted at that time, and the receive complete interrupt request will not be generated. The SL bit specifies the length of the stop bit. When this bit is reset to 0, the stop bit length is set to 1 bit. When this bit is set to 1, the stop bit length is set to 2 bits. The CL bit specifies the character length. When this bit is reset to 0, the character length is set to 7 bits. When this bit is set to 1, the character length is set to 8 bits.

The PEN bit specifies parity enable. When this bit is reset to 0, no parity bit is attached. When this bit is set to 1, the parity bit is attached. When the parity bit is attached, odd or even parity is specified by the EP bit. When the EP bit is reset to 0, odd parity is specified. When the EP bit is set to 1, even parity is specified. The EP bit is effective only when the PEN bit is set to 1.

The TxRDY bit controls transmit enable. Transmit is enabled when the $\overline{\text{CTS}}$ pin is at low level and the TxRDY bit is set to 1.

Tx RDY	RxE	PEN	EP	CL	SL	0	1

SCM

Stop bit length specification

0	1 stop bit
1	2 stop bits

Character length specification

0	7 bits
1	8 bits

Odd/even parity specification

0	Odd parity
1	Even parity

Parity bit control

0	No parity
1	Parity attached

0	Receive disabled
1	Receive enabled

0	Transmit disabled
1	Transmit enabled when $\overline{CTS}$ pin is at low level

Fig. 3-39 Serial Communication Mode Register (in asynchronous mode)

(b) When set to I/O interface mode

The RSCK bit specifies the serial receive clock source. When this bit is reset to 0, the external receive clock is used for the receive operation. When this bit is set to 1, the internal receive clock is used. Input/output of the receive clock is done through the $\overline{\text{CTS}}$ pin.

The TSK bit functions as a receive clock output trigger bit. This bit is effective only when the RSCK bit is set to 1. Eight shift clocks are output through the $\overline{\text{CTS}}$ pin when 1 is written to the TSK bit.

The RxE bit determines whether or not receive operation is enabled. When the RxE bit is set to 1, receive operation is enabled. When this bit is reset to 0, receive operation is disabled. If receive operation is disabled during a receive operation, the receive operation is terminated at that time. In this case, no receive complete interrupt request will be generated.

The TxE bit determines whether or not transmit operation is enabled. When the TxE bit is set to 1, transmit operation is enabled. When this bit is reset to 0, transmit operation is disabled.

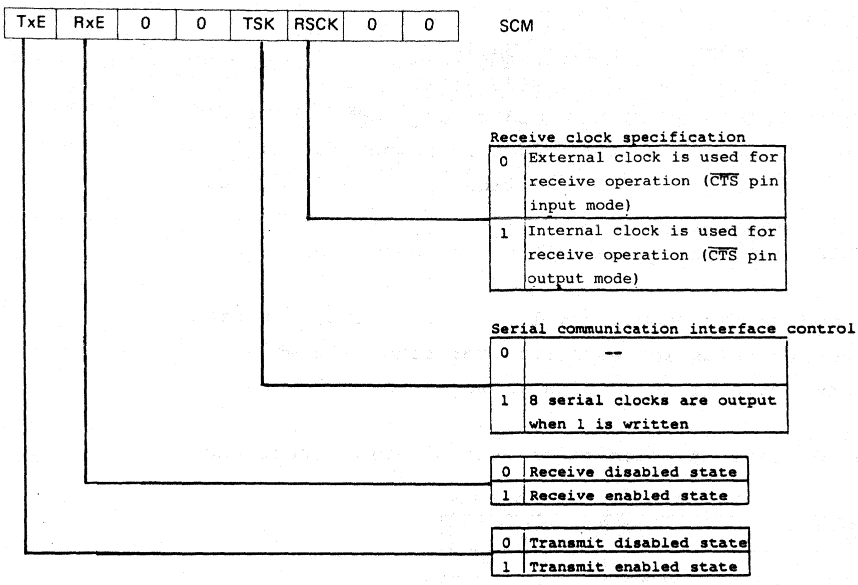


Fig. 3-40 Serial Communication Mode Register Format (I/O interface mode)

3.4.2 Baud rate generator

The baud rate generator is an 8-bit timer used exclusively for serial communication interface, and generates shift clocks for transmit/receive operation. Both the transmit and receive baud rate generators are provided.

The input clock to the baud rate generator can be specified by selecting the time base counter output tap using the PRS2-0 bits of the serial communication control register (SCC). One-half of the baud rate generator signal is used as the serial communi-cation interface shift clock. Each parameter is determined in a manner that satisfies the following formula for setting the baud rate generator for a certain baud rate:

$$B \times G = 10^6 \times SCLK/2^{n+1}$$

where;

B: Transfer baud rate (bps)
B=110, 150, ..., 9600, 19200. ...

G: Value set to the baud rate generator ($1 \leq G \leq 255$)

n: Input clock specification number for the baud rate generator specified by the SCC register ($0 \leq n \leq 7$)

SCLK: Internal system clock frequency (MHz)

Based on the above formula, the baud rate generator preset value for each transfer baud rate when a 12MHz crystal is used is shown in Table 3-10.

Table 3-10 Baud Rate Generator Setting Value (reference)

SCLK= 6MHz

Transfer baud rate	n	Baud rate generator setting value (G)	Tolerance (%)
110	7	213	0.03
150	7	156	0.16
300	6	156	0.16
600	5	156	0.16
1200	4	156	0.16
2400	3	156	0.16
4800	2	156	0.16
9600	1	156	0.16
19200	0	156	0.16
38400	0	78	0.16
1.5M	0	2	0

n: Input clock specification number for baud rate generator.

(1) Serial communication control register (SCC)

The serial communication control register (SCC) controls the transfer rate of the serial communication interface.

This register is initialized to 00H when $\overline{\text{RESET}}$ is input. The bit field consisting of the PRS2-0 bits specifies the output tap of the time base counter input to the baud rate generator.

7	6	5	4	3	2	1	0	
0	0	0	0	0	PRS2	PRS1	PRS0	SCC

PRS 2	PRS 1	PRS 0	n	Input clock for baud rate generator
0	0	0	0	SCLK
0	0	1	1	SCLK/2
0	1	0	2	SCLK/4
0	1	1	3	SCLK/8
1	0	0	4	SCLK/16
1	0	1	5	SCLK/32
1	1	0	6	SCLK/64
1	1	1	7	SCLK/128

SCLK: Internal system clock
n: Input clock specification number

Fig. 3-41 Serial Communication Control Register Format

3.4.3 Asynchronous mode

In the asynchronous mode, the character length, number of stop bits, parity enable, and odd/even parity specifications are controlled by the serial communication mode register (SCM).

(1) Transmit

Transmit is enabled* when bit 7 (TxRDY) of the serial communication mode register (SCM) is set to 1 and the $\overline{\text{CTS}}$ pin is at the active level (0).

Transmission can be started in the following three ways:

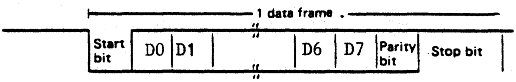
(a) First, generate a transmit complete interrupt request by setting the TxRDY bit to transmit enable when the transmit buffer (TxB) is empty. Second, write the transmit data to the transmit buffer during the interrupt process.

(b) Send the transmit data to the transmit buffer during the transmit enabled state. This transmit data will be transmitted after the previous transmission is complete.

(c) First, write the transmit data to the transmit buffer during the transmit disabled state; the data retained in the transmit buffer will be transmitted when the transmit is enabled.

* There is no specific constraint on the transmit enabling order; the TxRDY bit can be set to 1 either before or after setting the $\overline{CTS}$ pin to active.

As shown in the figure below, a data frame of the transmit data format consists of the start bit, character bit, and stop bit. Transmit data is output from the TxD pin with the least significant bit (LSB) first. The TxD pin enters the mark state (1) when transmit is disabled or it has no data to transmit to the serial register.



Start bit ....1 bit
Character bit .....7/8 bits
Parity bit ....Odd/even/none
Stop bit ....1/2 bit(s)

The transmit complete interrupt request is generated when the transmit buffer (TxB) becomes empty. The transmit buffer (TxB) becomes empty when $\overline{RESET}$ is input. If transmit is enabled this time, the transmit complete interrupt request is generated. Additionally, when the transmit data in the transmit buffer is transferred to the shift register because of the transmission operation, the transmit buffer becomes empty and the transmit complete interrupt request is generated.

By writing transmit data to the transmit buffer each time the transmit complete interrupt request is generated, data can be transmitted continuously without having mark state (1) inserted.
If transmit is disabled during the transmit operation, one frame of the data being transmitted will be transmitted completely. However, if new transmit data is already written to the transmit buffer, data transfer from the transmit buffer to the shift register is disabled, and the contents of the transmit buffer are retained as they are. The contents of the transmit buffer are transferred to the shift register when transmit is enabled again. When the transmission starts, the transmit complete interrupt request is simultaneously generated.

(2) Receive

Receive operation is enabled when bit 6 (RxE) of the serial communication mode register (SCM) is set to 1. In receive disabled state, receive hardware is at standby in the initialized state. Receive operation is started when the falling edge of the RxD pin input is detected in sampling using the input clock to the baud rate generator. The baud rate generator for the receive operation then starts countdown. If a low level of the RxD pin is detected by the first timing signal from the baud rate generator, this is recognized as a start bit. The receive operation will then continue. If a high level is detected by the first timing signal, this is not recognized as a start bit; the baud rate generator is then initialized to stop the receive operation.
The sampling of receive data will be synchronized with the rising edge of the shift clock after a start bit is detected.
The receive complete interrupt request is generated when the receive data in the shift register is transferred to the receive buffer (RxB) after the data whose character length is specified by bit 3 (CL) of the serial communication mode register is received.

Odd/even parity check (when PEN bit*= 1) is performed during the receive operation. If no coincidence (parity error) occurs, the stop bit is at a low level (framing error), or the next data is transferred to the receive buffer which is already full (overrun error), the receive error flag is set and the receive error interrupt request is generated.

* The PEN bit is bit 5 of the serial communication mode register.

3.4.4 I/O interface mode

The I/O interface mode of the μPD78312/78310 is the same as the serial interface mode of the μCOM-87. This is effective for I/O expansion and connecting I/O controllers(such as A/D converters and LCD controllers).

In the I/O interface mode, the character length is fixed to 8 bits, and data is transferred from the most significant bit (MSB) first without a parity bit.

(1) Transmit

Transmit operation is enabled when bit 7 (TxE) of the serial communication mode register is set to 1. In the I/O interface mode, the SCK pin functions as the transmit clock output pin. In the same way as the asynchronous mode, transmission can be started in the following three ways:

(a) First, generate a transmit complete interrupt request by setting the TxRDY bit to transmit enable when the transmit buffer (TxB) is empty. Second, write the transmit data to the transmit buffer during the interrupt process.

(b) Send transmit data to the transmit buffer during the transmit enabled state. This transmit data will be transmitted after the previous transmission is complete.

(c) First, write the transmit data to the transmit buffer during transmit disabled states, the data retained in the transmit buffer will be transmitted when the transmit is enabled.

The transmit complete interrupt request is generated when the transmit buffer (TxB) becomes empty. The transmit buffer (TxB) becomes empty when $\overline{\text{RESET}}$ is input. If transmit is enabled at this time, the transmit complete interrupt request is generated. Additionally, when the transmit data in the transmit buffer is transferred to the shift register because of the transmission operation, the transmit buffer becomes empty and the transmit complete interrupt request is generated.

(2) Receive operation

Receive operation is enabled when bit 6 (RxE) of the serial communication mode register (SCM) is set to 1. Receive data is input to the serial register at the rising edge of the receive clock. When the serial register receives 8-bit data, this 8-bit data is transferred to the receive buffer (RxB). The receive complete interrupt request is then generated.

In the I/O interface mode, the receive clock can be selected from either the external or internal receive clock by the specification of bit 2 (RSCK) of the serial communication mode register (SCM).

The $\overline{\text{CTS}}$ pin functions as the receive clock input/output pin in the I/O interface mode. During a receive operation, if the next data is transferred to the receive buffer (RxB) which is full (overrun error), the receive error flag is set and the receive error interrupt request is generated.

3.4.5 Serial communication interface interrupt request

Three interrupt requests (transmit complete interrupt, receive complete interrupt, and receive error interrupt) are generated from the serial communication interface.

(1) Serial communication interrupt request control register (SEIC, SRIC, STIC)

These registers control the three interrupt requests (transmit complete interrupt request (STF), receive complete interrupt request (SRF), and receive error interrupt request (SEF)).

These three interrupt request control registers form a group. The group can be given a priority order as the serial communication interface interrupt request. Within the group, priority is fixed as follows:

SEF > SRF > STF

7	6	5	4	3	2	1	0	
SEF	SEMK	0	ENCS	0	PR2	PR1	PR0	SEIC
SRF	SRMK	MS/$\overline{INT}$	ENCS	0	-	-	-	SRIC
STF	SEMK	MS/$\overline{INT}$	ENCS	0	-	-	-	STIC

Fig. 3-42 Interrupt Request Control Register Format

The SEF, SRF, and STF bits are interrupt request flags, and are set to 1 by receive error, receive complete, and transmit complete, and are reset by interrupt request acceptance or software.

Other bit fields are explained in Chapter 4, Interrupt Request Process Function.

(2) Serial communication macro service control register (SRMS, STMS)

The SRMS is an 8-bit register which specifies the macro service process and mode channel required upon the completion of the serial communication receive operation. The STMS is an 8-bit register which specifies the macro service process and mode channel required upon the completion of the serial communication transmit operation. Each bit of the macro service control register is explained in Section 4.2, Macro Service Function.

7	6	5	4	3	2	1	0	
MSM2	MSM1	MSM0	DIR	0	CH2	CH1	CH0	SRMS

7	6	5	4	3	2	1	0	
MSM2	MSM1	MSM0	DIR	0	CH2	CH1	CH0	STMS

Fig. 3-43 Macro Service Control Register Format

3.5 Time Base Counter/Watchdog Timer Function

The time base counter/watchdog timer function consists of a 20-bit time base counter and 8-bit watchdog timer which use the outputs of the time base counter as clock source.

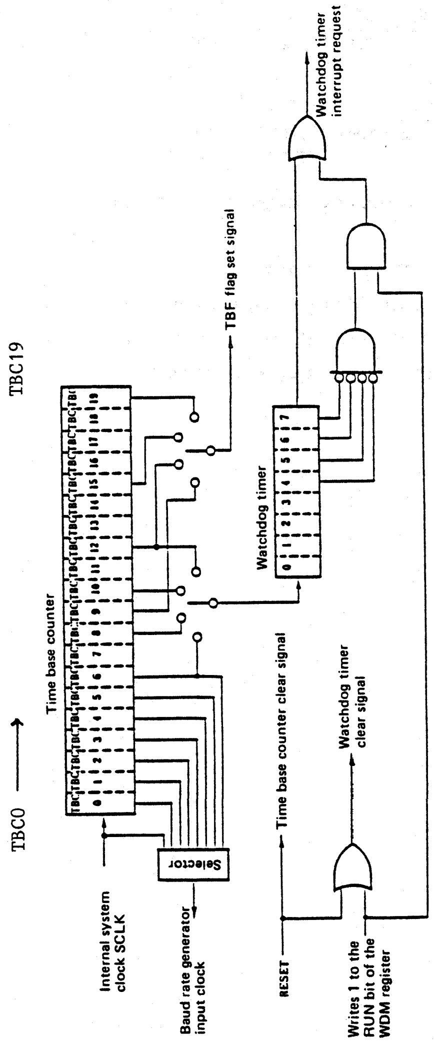


Fig. 3-44 Time Base Counter/Watchdog Timer Configuration

3.5.1 Time base counter

In system controller application, various processes may be performed based on time base. In the μPD78312/78310, a 20-bit time base counter which divides the internal system clock (SCLK) is provided to obtain time base output for reference.

The interval time of the interrupt request generated from the time base counter can be selected from the four types in Table 3-11 by using the time base count mode register (TBM). The interrupt request is generated when the interrupt request flag (TBF) is set to 1 at the falling edge of the time base counter output tap specified by the time base counter mode register (TBM) after it was set to high level.

The time base counter is cleared to 00H only by $\overline{\text{RESET}}$ input. After it is cleared, it is continuously incremented.

Note: The time base counter is also used as a free-running counter of the capture/PWM unit.

Table 3-11 Time Base Counter Interval Time

Mode	Output tap	Interval time
0	TBC9	2^{10}/SCLK (170μs)
1	TBC12	2^{13}/SCLK (1.36ms)
2	TBC15	2^{16}/SCLK (10.9ms)
3	TBC19	2^{20}/SCLK (175ms)

Note: The interval time is undefined for the period between mode register setting and the first interrupt request generation.

7	6	5	4	3	2	1	0	
0	0	0	0	0	0	TB1	TB0	TBM

TB1	TB0	
0	0	TBC 9 of the time base counter outputs interval
0	1	TBC 12 of the time base counter outputs interval
1	0	TBC 15 of the time base counter outputs interval
1	1	TBC 19 of the time base counter outputs interval

Fig. 3-45 Time Base Mode Register Format

3.5.2 Watchdog timer

When 1 is written to bit 7 (RUN) of the watchdog timer mode register (WDM), the watchdog timer is cleared to 00H and starts counting down the time base counter tap output specified by the watchdog timer mode register.

A nonmaskable interrupt request is generated if the watchdog timer is cleared to 00H before bit 4 of the watchdog timer is first set, or the watchdog timer is not cleared before it overflows.

Priority order of the nonmaskable interrupt request from the watchdog timer can be specified against the non-maskable interrupt request from the NMI pin. The priority order can be specified by bit 4 (PRC) of the watchdog timer mode register as follows:

PRC = 0 .... NMI > WDT
PRC = 1 .... WDT > NMI

The watchdog timer overflow time and the time from count start until bit 4 is first set to 1 (6.25% of overflow time) when the 6MHz internal system clock is used are shown in Table 3-12.

Table 3-12 Watchdog Timer Count Clock and Overflow Time

Watchdog timer count clock	Overflow time	6.25% of overflow time
SCLK/2^7 (TBC6 tap output)	5.5ms	343µs
SCLK/2^9 (TBC8 tap output)	21.8ms	1.36ms
SCLK/2^{11} (TBC10 tap output)	87.4ms	5.46ms
SCLK/2^{13} (TBC12 tap output)	349.5ms	22ms

(1) Watchdog timer mode register (WDM)

The watchdog timer mode register (WDM) is an 8-bit register which controls the operation of the watchdog timer. The WDM register can be written only by a special instruction, so that the contents of the WDM register are not easily modified in the event of a program runaway. The WDM register is cleared to 00H when $\overline{\text{RESET}}$ is input.

The RUN bit controls the start of the watchdog timer count operation. The watchdog timer stops count operation when the RUN bit is reset to 0. When the RUN bit is set to 1, the watchdog timer is cleared to 00H and starts count operation. WDI0-3 is the bit field which selects the count clock for the watchdog timer; that is, the output tap from the time base counter is specified by these bits.

The PRC bit specifies the priority order of the non-maskable interrupt request from the watchdog timer and the interrupt request from the NMI pin input.

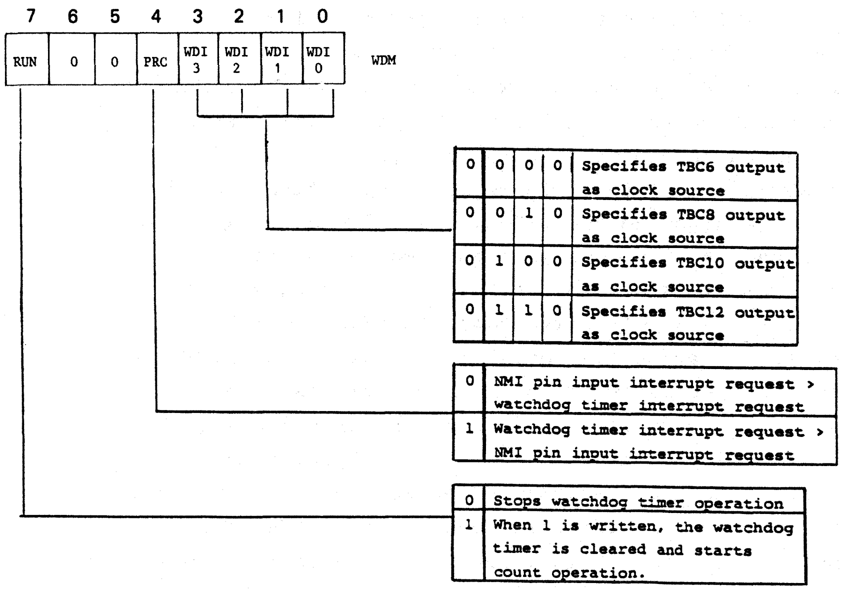


Fig. 3-46 Format of Watchdog Timer Mode Register

(2) Time base counter interrupt request control register (TBIC)

This is an 8-bit register which controls the interrupt request generated from the time base counter. Together with the A/D converter interrupt request (ADIC), this interrupt forms a group of interrupt requests. For the TBIC register format, refer to A/D converter interrupt request control register.

Bit 7 (TBF) of the TBIC register is the flag for the interrupt request from the time base counter. This flag is set to 1 at the falling edge of the tap output selected by the time base mode register (TBM) after it is set to 1. The TBF flag is reset to 0 when an interrupt is accepted or reset by software.

3.6 External Interrupt Request Function

The external interrupt request is generated when the effective edge specified by the external interrupt mode register (INTM) is input to the INTE0-2 or NMI pin.

(1) External interrupt mode register (INTM)

This is an 8-bit register which specifies the effective edge input to the INTE0-2 and NMI pins. This register is cleared to 00H when $\overline{\text{RESET}}$ is input.

The ESNMI bit of the INTM register specifies the effective edge input to the NMI pin. When this bit is reset to 0, the falling edge is specified as an effective edge. When this bit is set to 1, the rising edge is specified as an effective edge. The ES0 and ES1 bits specify the effective edges of the INTE0 and INTE1 pin input, respectively. When the ES0/1 bit is reset to 0, the falling edge is specified as an effective edge. When the ES0/1 bit is set to 1, both the rising and falling edge are specified as effective edges. The ES2 bit specifies the effective edge of the INTE2 pin input. When this bit is reset to 0, the falling edge is specified as an effective edge. When this bit is set to 1, the rising edge is specified as an effective edge.

7	6	5	4	3	2	1	0	
0	ES2	0	ES1	0	ES0	0	ES NMI	INTM

NMI pin effective edge specification

0	Specifies the falling edge of the NMI pin input as effective edge
1	Specifies the rising edge of the NMI pin input as effective edge

INTE0 pin effective edges specification

0	Specifies the falling edge of the INTE0 pin input as effective edge
1	Specifies both the rising and falling edges of the INTE0 pin input as effective edges

INTE1 pin effective edge specification

0	Specifies the falling edge of the INTE1 pin input as effective edge
1	Specifies both the rising and falling edges of the INTE1 pin input as effective edges

INTE2 pin effective edge specification

0	Specifies the falling edge of the INTE2 pin input as effective edge
1	Specifies the rising edge of the INTE2 pin input as effective edge

Fig. 3-47 External Interrupt Mode Register Format

(2) External interrupt pin interrupt request control register (EXIC0 t0 2)

These registers control interrupt requests generated from the external interrupt pin (INTE0 to 2). Three interrupt requests form a group of interrupt requests. The priority order of this group can be specified by program. Within the group, the priority order is fixed by means of hardware as follows:

INTE0 > INTE1 > INTE2

	7	6	5	4	3	2	1	0	
	EXIF0	EX1 MK0	MS/$\overline{INT}$	EN CS	0	PR2	PR1	PR0	EXIC0
	EXIF1	EX1 MK1	MS/$\overline{INT}$	EN CS	0	-	-	-	EXIC1
	EXIF2	EX1 MK2	MS/$\overline{INT}$	EN CS	0	-	-	-	EXIC2

Fig. 3-48 Interrupt Request Control Register Format

The EXIF0, EXIF1, and EXIF2 bits are interrupt request flags. These flags are set by effective edge input to the INTE0, INTE1, and INTE2 pins, respectively, and are reset to 0 when an interrupt is accepted or by software. Each bit is explained in Chapter 4, Interrupt Request Process Function.

(3) External interrupt pin macro service control register (EXMS0-2)

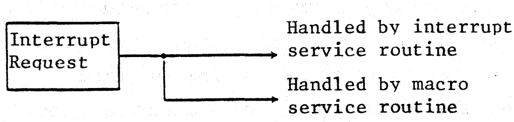
These are 8-bit registers which specify the macro service operation mode.

	7	6	5	4	3	2	1	0
EXMS0	MSM2	MSM1	MSM0	DIR	0	CH2	CH1	CH0
EXMS1	MSM2	MSM1	MSM0	DIR	0	CH2	CH1	CH0
EXMS2	MSM2	MSM1	MSM0	DIR	0	CH2	CH1	CH0

Fig. 3-49 Macro Service Control Register Format

CHAPTER 4 INTERRUPT REQUEST SERVICE FUNCTIONS

The μPD78312/78310 handles interrupt requests generated from the on-chip peripheral hardware and the external devices by software-oriented interrupt service routine or by macro service function (refer to 4.2, Macro Service Functions) which simply transfers data by hardware.



4.1 Interrupt Request

The interrupt requests are divided into the following three types:

- Nonmaskable interrupt requests
- Maskable interrupt requests
- Software interrupt requests

4.1.1 Nonmaskable interrupt requests

Nonmaskable interrupt requests are always acknowledged, even in the DI state. A nonmaskable interrupt request can be generated by either a valid input to the NMI pin or the watchdog timer.

Unlike the other maskable interrupt requests, the non-maskable interrupt requests are not subject to programmable priority control.

The priority between the interrupt requests generated by an input to the NMI pin and the watchdog timer can be specified by the watchdog timer mode register (refer to 3.6, Time Base Counter/Watchdog Time).

4.1.2 Maskable interrupt requests

The priority of the maskable interrupts can be controlled group-wise by program. Within a group, the default priority is determined by the hardware. (Refer to Table 4-1.)

When two or more interrupt requests are simultaneously generated to the processor, an interrupt request of the highest priority is selected by the priority determination circuit and is processed by the interrupt service or by macro service. When an interrupt request is acknowledged, the processor enters the DI state, thus disabling subsequent interrupts. When the EI instruction is executed within the interrupt service, the processor enters the EI state and interrupt requests are enabled for all priority levels higher than that being currently acknowledged as an interrupt request. The maskable interrupt requests are further divided into two types: one generates macro service requests and the other does not generate them.

(1) Maskable interrupt requests which do not generate macro service requests

These are maskable interrupt requests which generate only interrupt services. Each source of the interrupt requests is provided with an interrupt control register.

(2) Interrupt requests which generate macro service requests

These are maskable interrupt requests which can generate interrupt and macro service requests. Each source of the interrupt requests is provided with an interrupt control and a macro service control register.

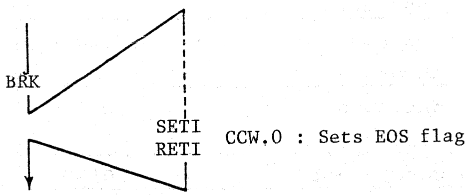
4.1.3 Software interrupt request

The software interrupt request is generated by executing the BRK instruction and acknowledged even in the DI state. The software interrupt is not subject to priority control.

- BRK (Break) instruction

  The BRK instruction is a nonmaskable interrupt request and is also acknowledged in the DI state. The BRK instruction may be nested in its own routine by executing a BRK instruction within a BRK service routine. Return from a BRK instruction service routine does not reset the in-service priority register* (ISPR) to "0"; therefore, the EOS flag in the CPU control word (CCW) must be set to "1" before executing a RETI instruction.

* In-service priority register (ISPR)



The in-service priority register is an 8-bit register which contains the priority specified to the currently acknowledged interrupt request source. When an interrupt request is acknowledged, the bit corresponding to the priority level of the interrupt being acknowledged is set to "1" and retained during the service processing. Execution of the RETI or RETCS instruction resets the bit corresponding to the highest priority interrupt request being acknowledged to "0".

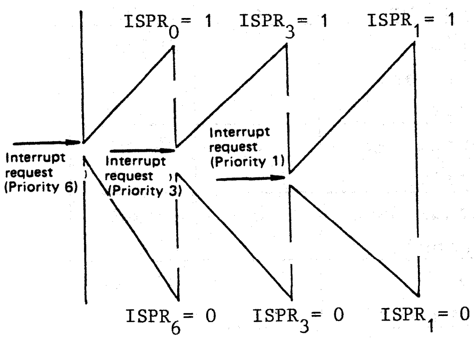


Table 4-1 Sources of Interrupt Requests

Type of interrupt request	Default priority	Source of interrupt request		Macro service	Vector table address
Software	-	BRK	Break instruction	-	003EH
Nonmaskable	-	NMI	NMI pin input	-	0002H
	-	WDT	Watchdog timer	-	000AH
Maskable	0	CRF00	Count unit	o	001AH
	1	CRF01	Count unit	-	001CH
	2	CRF10	Count unit	o	001EH
	3	CRF11	Count unit	-	0020H
Maskable	4	EXIF0	INTE0 pin input	o	0004H
	5	EXIF1	INTE1 pin input	o	0006H
	6	EXIF2	INTE2 pin input	o	0008H
Maskable	7	TMF0	Timer unit	o	000EH
	8	TMF1	Timer unit	o	0010H
	9	TMF2	Timer unit	o	0012H
Maskable	10	SEF	Serial interface error	-	0022H
	11	SRF	Serial data receive	o	0024H
	12	STF	Serial data transmit	o	0026H
Maskable	13	ADF	A/D converter	o	0028H
	14	TBF	Time base counter	-	000CH
Reset	-	RESET	Reset input	-	0000H

Note: Default priority: Priority fixed by hardware

- DI (Interrupt disabled): All interrupts other than maskable interrupts are pending.
- EI (Interrupt enabled)
  - Interrupt request flag is reset to "0"
  - Interrupt request flag is set to "1"
    - MS/INT=1 Macro service handling
    - MS/INT=0
      - MK=1 (masked) pending
      - MK=0 (unmasked)
        - ENCS=1 ⟶ (Context switching interrupt)
        - ENCS=0 ⟶ Normal interrupt

Fig. 4-1 Interrupt Request Servicing Format

4.1.4 Interrupt request control registers

An interrupt request control register is provided for each source of an interrupt requests, such as serial interrupt pin. This register is used to control the programmable priority and mask of the corresponding interrupt request. Bits PR0 to 2 of the interrupt request control register comprise a bit field which specifies, group-wise, the priority of the interrupt request services. Priority level 0 is the highest priority of the maskable interrupt requests that can specify their priority. The MS/INT bit specifies whether the interrupt request service is to be processed by the macro service or normal interrupt service. This bit is fixed to "0" for the interrupt requests which do not request macro service (normal interrupt request).

The IMK bit is a mask bit of the corresponding interrupt request flag. When this bit is set to "1", the corresponding interrupt request is masked. The IF bit is an interrupt request flag generated by each peripheral hardware block. The ENCS bit is used to enable register bank switching with the context switching function. When the ENCS bit is set to "1", the register bank is switched by the hardware to the register bank whose number is the same as the value

Given as priority for the group to which the given interrupt belongs (refer to 4.3, Context Switching).
Context switching is disabled when the ENCS bit is reset to "0". Each interrupt control register is set to 47H by $\overline{\text{RESET}}$ input.

7	6	5	4	3	2	1	0
xxFa	xxAIKa	MS/$\overline{\text{INT}}$	EN CS	0	PR2	PR1	PR0

Priority for the interrupt requests of each group (PR2, PR1, PR0)

PR2	PR1	PR0		
0	0	0	Priority level 0	High
:	:	:	:	↑
1	1	1	Priority level 7	Low

Context switching function control (ENCS)

0	Context switching disabled
1	Context switching enabled

Macro service function control (MS/$\overline{\text{INT}}$)

0	Vectored interrupt processing
1	Macro service processing

Mask control of the interrupt request (xxAIKa)

0	Interrupt request enabled
1	Interrupt request disabled

Interrupt request flag (xxFa)

0	No interrupt request
1	Interrupt is being requested

Fig. 4-2 Int Request Control Register Format

4.2 Macro Service Functions

The macro service functions are intended to reduce the number of software-oriented interrupts to occur in order to minimize overhead caused by the interrupts such as interrupt service processing, register saving, register recovering, and return from the interrupt service routine, thus improving the processor's service time.
When a macro service request is generated, the CPU halts program execution and performs a special hardware transfer

instruction, and then returns to the program execution upon completion of a one-word data transfer. Therefore, the macro service processing is completely transparent to the user program. As shown in a flow chart in Fig. 4-3, macro service is achieved by activating one or two bytes of the data transfer operation.
Macro service is completed after executing the number of macro services preset to the macro service counter. After having transferred the specified number of data, the macro service enable flag is reset to "0" and, at the same times, a macro service completion interrupt is generated. The macro service completion interrupt is performed by branching to the appropriate vector address of the interrupt request source which generated the macro service request. In the macro service completion interrupt processing, such operations as resetting the macro service pointer, the SFR pointer, and the macro service counter are performed.

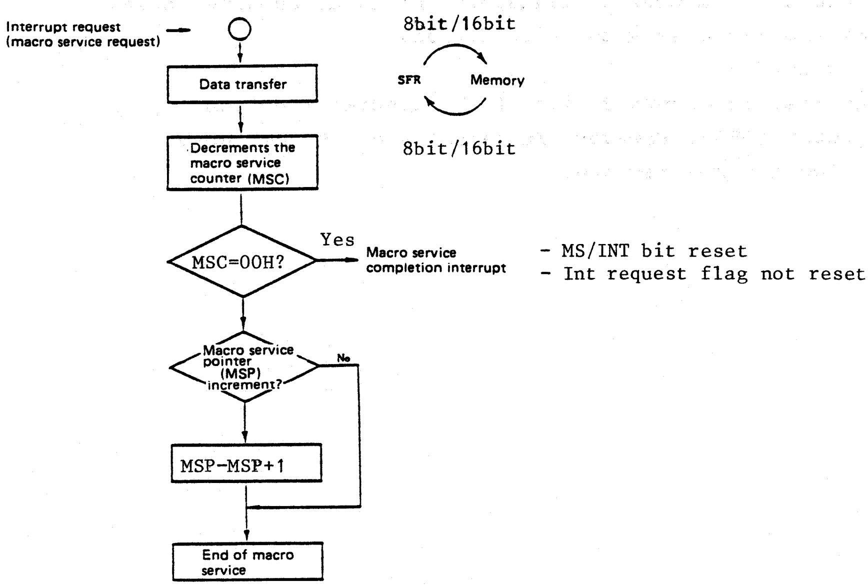


Fig. 4-3 Macro Service Transfer Flow

(1) Macro service channels

The macro service channel is composed of two pointers and a counter to control macro service transfer processing. Eight channels are provided at locations FEE0 to FEFFH of the internal RAM (which also serve as register banks 0 and bank 1).

15 8	7 0
Macro service pointer (MSP)	
SFR pointer' (SFRP)	Macro service counter (MSC)

Fig. 4-4 Configuration of Macro Service Channel

- Macro service pointer (MSP):
  Specifies source or destination memory address.
- Macro service counter (MSC):
  Controls the number of transfers and is decremented each time the macro service is performed.
- SFR pointer:
  Specifies the lower 8 bits of the special function register (SFR). However, it cannot specify the macro service control register.

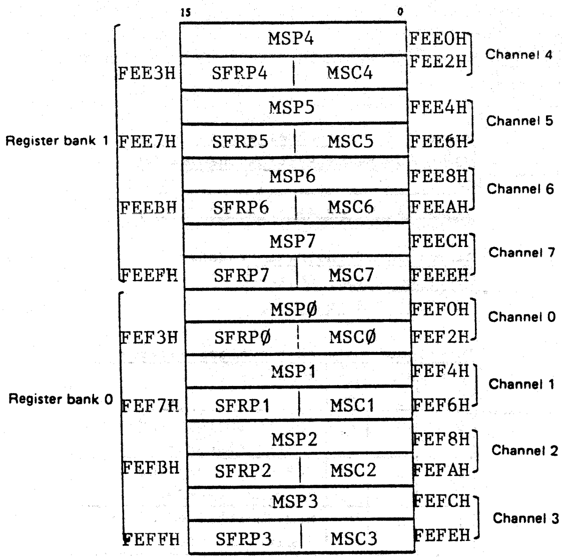


Fig. 4-5 Mapping of Macro Service Channel

(2) Macro service control registers

The Macro service control register is used to control the selection of the transfer mode of the macro service and the macro service channels. It is provided in each source of the maskable interrupt requests which generate macro service requests (refer to Fig. 4-6).

Bits CH0 to 2 of the macro service control register comprise a bit field which selects one of the eight macro service channels in the internal RAM. The DIR bit is used to control a direction of the data transfer. When the DIR bit is reset to "0", the data are transferred from memory to SFR, whereas when set to "1", the data are transferred from SFR to memory. Bits MSM0 to 2 comprise a bit field which specifies the transfer mode of the macro service, that is, whether it is an 8-bit transfer or a 16-bit transfer, and whether the macro service pointer is incremented or retained after the data transfer.

7	6	5	4	3	2	1	0
MSM2	MSM1	MSM0	DIR	0	CH2	CH1	CH0

Macro service channel select

CH2	CH1	CH0	
0	0	0	Macro service channel 0
:	:	:	:
1	1	1	Macro service channel 7

Direction of the data transfer

DIR	
0	From memory to SFR
1	From SFR to memory

Functional mode of the macro service

MSM 2	MSM 1	MSM 0	Transfer mode
0	0	0	MSP is incremented after 8 bits transfer
0	0	1	MSP is incremented after 16 bits transfer
1	0	0	MSP is retained after 8 bits transfer
1	0	1	MSP is retained after 16 bits transfer

Fig. 4-6 Macro Service Control Register Format

4.3 Context Switching Function

This is a function which selects an appropriate register bank by hardware as a result of an interrupt request or execution of the BRKCS instruction and then branches to the vector address prestored in the register bank, while stacking the current contents of the PC and PSW into the register bank.

4.3.1 Context switching function activated by interrupt request

The activation of the context switching function is enabled by setting the ENCS bit of the interrupt request control register defined for each source of interrupt requests. When an unmasked interrupt request is generated, the appropriate register bank equivalent to the priority level of the group to which the corresponding interrupt request belongs is selected. The prestored vector address in the selected bank is loaded into the PC, and at the same time, the current contents of the PC and PSW are saved in the register bank.

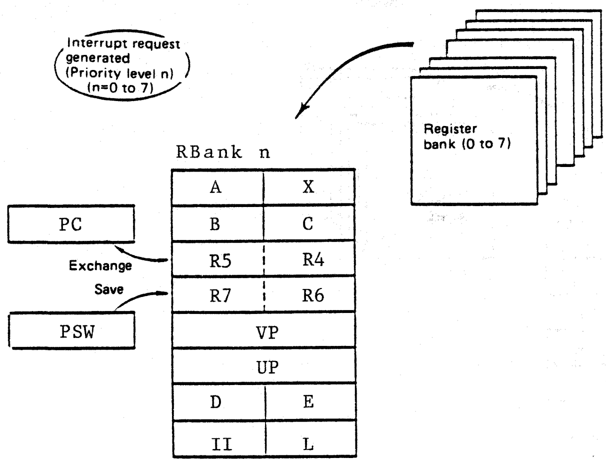
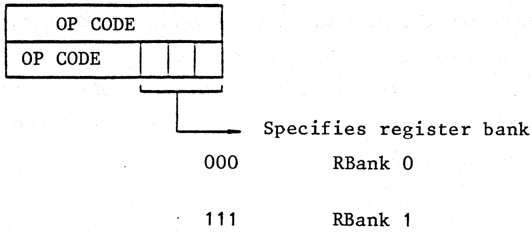


Fig. 4-7 Context Switching Activated by Interrupt Request

4.3.2 Context switching function activated by BRKCS instruction

The context switching function can be activated as a result of execution of the BRKCS instruction. The register bank to be selected is specified by the lower 3 bits of the immediate data in the second OP code of the BRKCS instruction. When the BRKCS instruction is executed, the register bank specified by the 3-bit immediate data is selected, and branch to the vector address prestored in the register bank occurs. At the same time, the current contents of the PC and PSW are saved in the register bank.



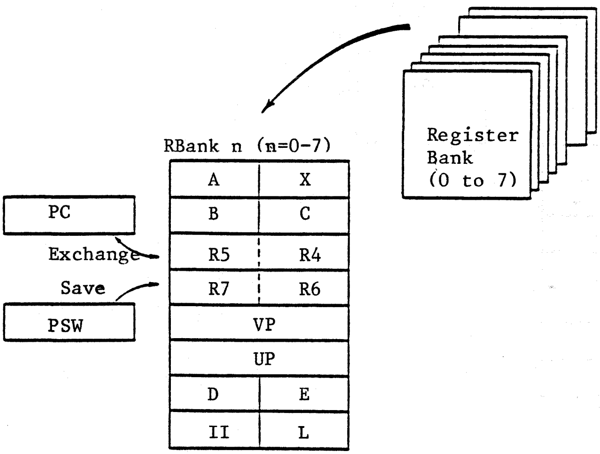


Fig. 4-8 Context Switching by BRKCS Instruction Execution

4.3.3 Return from the address branched by context switching function

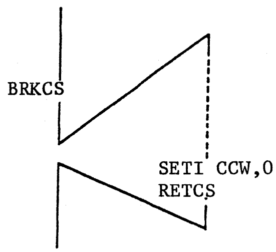
Return from the branch operation is performed by the RETCS instruction. By executing the RETCS instruction, the contents of the R4 and R5 registers within the register bank selected at the time of execution are loaded into the PC and the contents of the R6 and R7 registers are loaded into the PSW. At the same time, the immediate data specified by the second and third byte of the OP code are stored into the R4 and R5 registers in the register bank. When the same register bank is selected again by a switching function, the 16-bit immediate data specified by the second and third byte of the OP code of the RETCS instruction becomes the branch address.

RETCS Instruction format

First OP code	0	0	1	0	1	0	0	1
Second OP code	Lower 8-bit immediate data							
Third OP code	Upper 8-bit immediate data							

RETCS Instruction Format

When returning from the branch operation, the corresponding bit of the highest priority of those set in the in-service priority register (ISPR) is reset to "0". When the context switching is activated by executing the BRKCS instruction, resetting the ISPR register to "0" causes the interrupt nesting control to be destroyed. To prevent this, bit 0 (EOS) of the CPU control word (CCW) must be set to "1" immediately preceding execution of the RETCS instruction.



CHAPTER 5 STANDBY FUNCTIONS

The μPD78312/78310 provide the following three modes which control the operating clock as standby functions:

- Clock alterable mode
  In this mode, operations over a wide range of power supply voltages are permitted by altering the dividing ratio of the internal system clock (SCLK).

- HALT mode
  In this mode, the CPU clock is halted. The total power consumption of a system can be reduced by the intermittent operation resulting from the combination with the normal operation mode.

- STOP mode
  In this mode, the internal oscillator is turned off. Data can be retained with low power consumption.

Each mode is set by the standby control register (STBC). Writing to the standby control register (STBC) is permitted only by special instructions like those for the watchdog timer mode register (WDM) to prevent accidental entry to the STOP mode due to program overrun and other system failure.

- Standby control register (STBC)
  The standby control register controls the modes of the standby functions. The HLT bit is used to place the processor into the HALT mode. The HLT bit is reset to "0" during the normal operating state and set to "1" to enter the HALT mode. The STP bit is used to place the processor in the STOP mode. The STP bit is reset to "0" during the normal operating mode and set to "1" to enter the STOP mode.
  The CK0 and CK1 comprise a bit field which selects a dividing ratio of the internal clock. The oscillator output frequency is internally divided by the value specified by the CK0 and CK1 to produce the internal system clock.

The SBF bit is a standby flag which can be used to test the conditions when returning from the STOP mode. The standby flag (SBF) is reset to "0" only at the rising edge of the power supply (V_{DD}) and not affected by an active low input to the $\overline{\text{RESET}}$ pin. The standby control register (STBC) is set to 0010X000B by the $\overline{\text{RESET}}$ input to enter the low-speed operation mode.

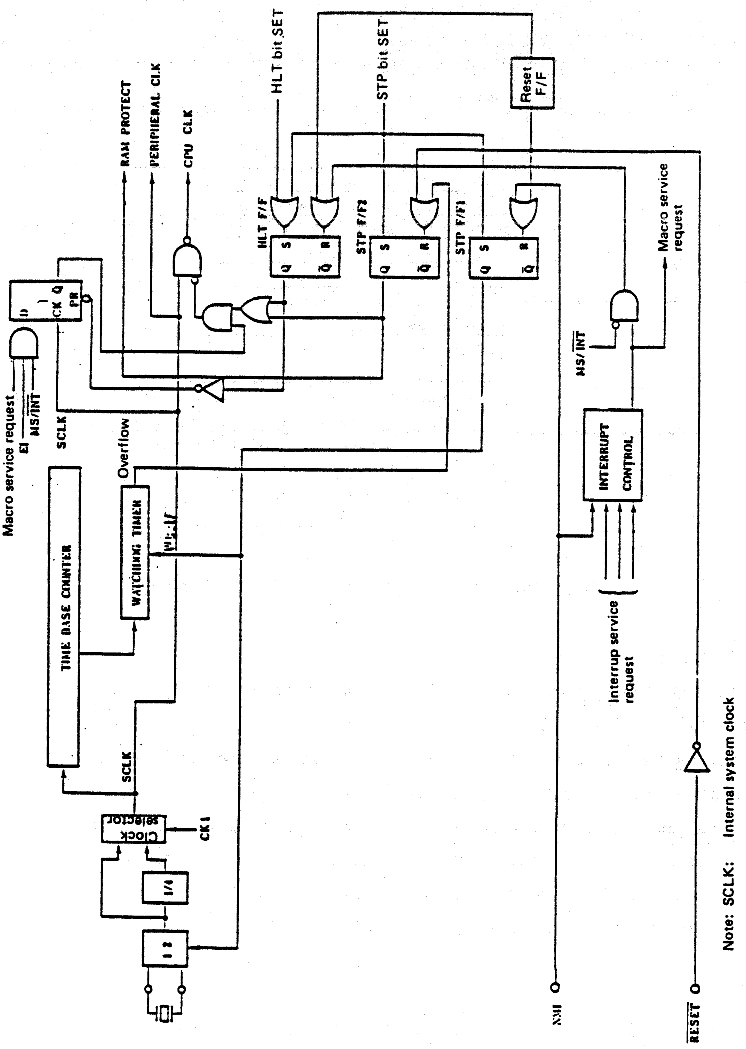


Fig. 5-1 Standby Function Block Diagram

7	6	5	4	3	2	1	0	
0	0	CK1	CK0	SBF	0	STP	HLT	STBC

STP	HLT	CPU operating mode control
0	1	Normal operation mode
0	1	HALT mode
1	-	STOP mode

Reset to "0" at the rising edge of the power supply

CK1	CK0	Selection of dividing ratio of the internal clock (SCLK)
0	0	SCLK=Oscillator frequency (fosc) x 1/2
1	0	SCLK=Oscillator frequency (fosc) x 1/8

Fig. 5-2 Standby Control Register (STBC) Format

5.1 Clock Alterable Mode

In the clock alterable mode, the internal clocks of the µPD78312/78310 are further divided by four. A low CPU operating speed allows a long-time stable operation of the battery-driven system, even if the supply voltage drops to a certain extent. In this mode, the oscillator frequency of the internal system clock is selected to either 1/2 or 1/8 by specifying bits 5 and 6 (CK1 to 0) of the standby control register (STBC).

5.2 HALT Mode

In the HALT mode, the CPU clock is turned off. The power consumption of the entire system can be reduced by setting the HALT mode during idle time of the CPU operation. The HALT mode is in effect by setting bit 0 (HLT) of the standby control register (STBC). When the CPU enters the HALT mode, the internal processor clock is stopped. This halts the program execution but the previous contents of all registers and the on-chip RAM are retained. The effects of the HALT mode on each of the CPU system hardware are described in Table 5-2.

5.2.1 Clearing HALT mode

The HALT mode is cleared by a nonmaskable interrupt request, an unmasked maskable interrupt, a macro service request, or $\overline{\text{RESET}}$ input.

(1) Clearing by interrupt request

(a) When the HALT mode is set in an interrupt service routine

In this case, the HALT mode is cleared by an unmasked maskable interrupt request with higher priority than that of the currently serviced interrupt, or by generating a nonmaskable interrupt request.

(b) Other than (a)

In this case, the HALT mode is cleared by a non-maskable interrupt request or an unmasked maskable interrupt request regardless of the priority.

(2) Clearing by $\overline{\text{RESET}}$ input.

Same as the normal reset operation.

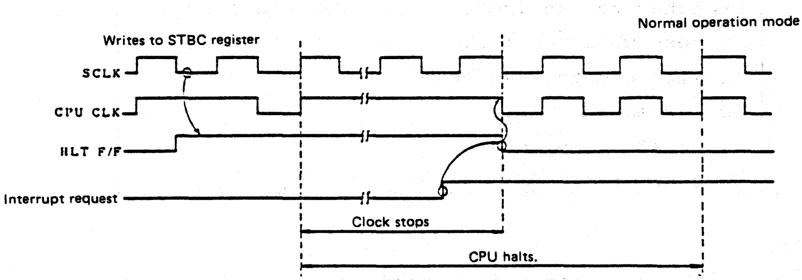


Fig. 5-3 Clearing HALT Mode by Interrupt Request

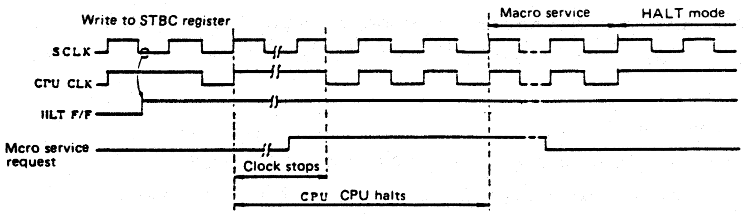


Fig. 5-4 Activation of Macro Service During HALT Mode

Table 5-1 Operation after Clearing HALT Mode by Interrupt Request

Source causing clearing	EI state	DI state
Nonmaskable interrupt request	Branches to the vector address on clearing	Branches to the vector address on clearing
Maskable interrupt request	Branches to the vector address on clearing	Executes the next instruction on clearing
Macro service request	Macro service routine is activated and branches to the vector address when the macro service counter reaches 0H; re-enters HALT state if the counter does not reach 0H.	Does not activate macro service routine and executes the next instruction

5.3 STOP Mode

In this mode, the oscillator is turned off. The STOP mode allows operation of extremely low-power consumption when activity of the entire application system is stopped. The STOP mode becomes effective by setting bit 1 (STO) of the standby control register to "1". In the STOP mode, all clocks are stopped. Program execution is halted but the current contents of all registers and the on-chip RAM are retained. The effects of the STOP mode on each of the CPU system hardware are described in Table 5-2.

5.3.1 Clearing STOP mode

The STOP mode is cleared by an interrupt request on the NMI pin or by $\overline{\text{RESET}}$ input.

(1) Clearing by input to the NMI pin

When a valid transition of the input signal is detected at the NMI pin, the oscillator is restarted. The time base counter and watchdog timer start operating. The watchdog timer starts supplying the CPU clock after the time count specified by the watchdog timer mode register (WDM) has elapsed. The processor first enters a state similar to that of the HALT mode and performs the same clearing operation.

(2) Clearing by $\overline{\text{RESET}}$ input

Same as the normal resetting operation.

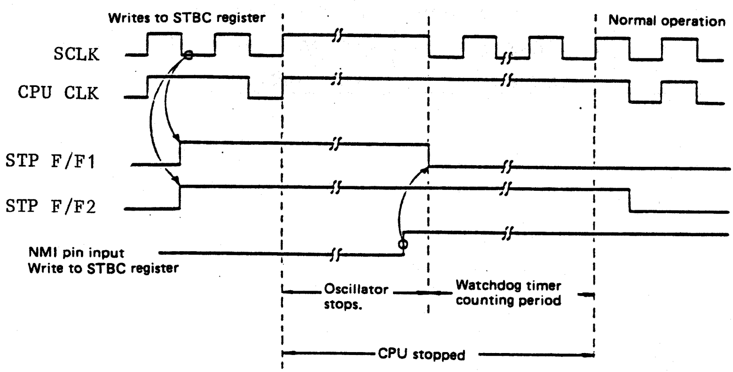


Fig. 5-5 Clearing STOP Mode by NMI Pin Input

Table 5-2 HALT Mode/STOP Mode

Item		HALT mode	STOP mode
Oscillator		Active	Stopped
Internal system clock		Stopped	
Pulse I/O unit Time base counter Watchdog timer Serial interface Interrupt request controller A/D converter		Active	
I/O line		Retained	Retained
Bus lines	A8 to 5	Retained	Retained
	AD0 to 7	High impedance	High impedance
$\overline{RD}$, $\overline{WR}$ output		High level	High level
ALE output		Low level	Low level
Data retention		All internal data such as contents of the CPU status and RAM are retained.	All internal data such as contents of the CPU status and RAM are retained.

CHAPTER 6 LOCAL BUS INTERFACE FUNCTIONS

In addition to the on-chip RAM, external memory (ROM and RAM) and I/O devices can be interfaced to the μPD78312/78310. The processors are also provided with a programmable wait function that automatically inserts waits (up to 3 waits) into the external access cycle when interfaced to the external slow-speed memory. The size of the external access space and the number of waits to be inserted into the external access cycle are specified by the memory expansion mode register (MM).

The μPD78312/78310 contains a pseudostatic RAM refresh function which permits direct interface to the μPD4168 pseudostatic RAM (or equivalents). By setting the refresh mode register, the output interval of the refresh pulses and the external access cycle are switched to the refresh cycle which is compatible with the bus cycle of the μPD4168.

6.1 Pseudostatic RAM Refresh Function

The μPD78312/78310 have the following capabilities to support pulse refresh and power down self-refresh operations.

(1) Pulse refresh operation

The μPD78312/78310 output refresh pulses from the $\overline{\text{RFSH}}$ pin in synchronization with the bus cycle to support the refresh cycle of the μPD4168.

(a) When accessing to the internal memory

The refresh bus cycle is output at the interval specified by the refresh mode register (RFM) to retain the data stored in the μPD4168 during internal memory access time in which no access is made to the external μPD4168.

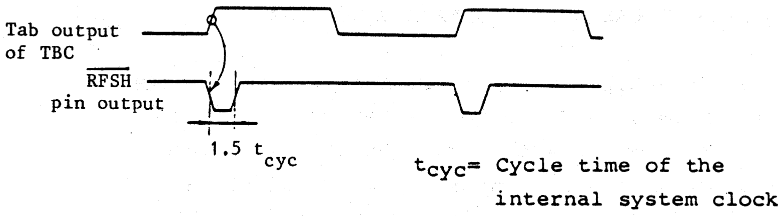


Fig. 6-1 Pulse Refresh Timing for Internal RAM Access

(b) When accessing to the external memory
Since overlapping of the access timing and refresh timing may cause the µPD4168 to malfunction, the µPD78312/78310 generate refresh pulses in synchronization with the bus cycle. During the refresh bus cycle, the 3-wait insertion mode is automatically specified. The refresh bus cycle occurs at the interval specified by the refresh mode register (RFM).

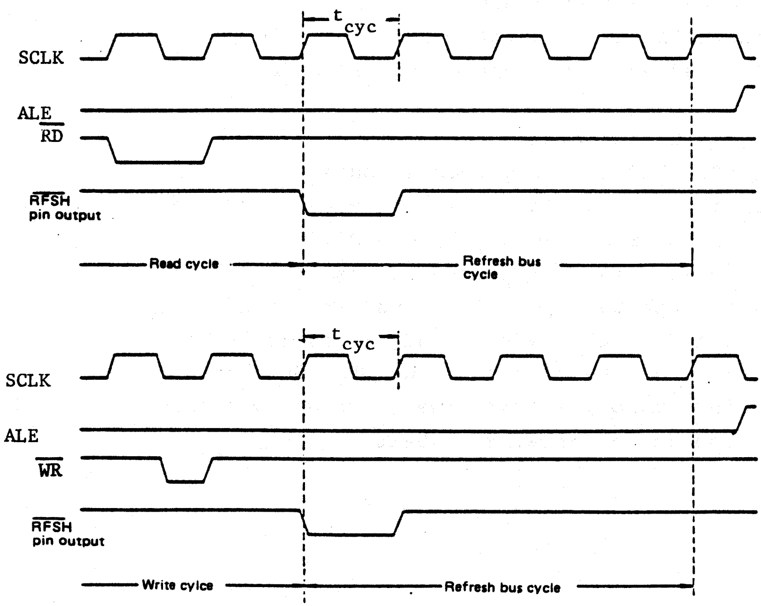


Fig. 6-2 Pulse Refresh Timing for External Memory Access

(2) Power down self-refresh operation

The power down self-refresh operation mode becomes effective as a result of a low level output at the $\overline{RFSH}$ pin by manipulating the refresh mode register (RFM) with software or by setting the standby mode. Return from the power down self-refresh operation is achieved by setting the $\overline{RFSH}$ pin to a high level with the software.

(a) Return from power down self-refresh operation

Refresh pulses to the μPD4168 are disabled for 2us following the rising edge of the RFSH line, the RFSH pin will go high in synchronization with the time base counter.

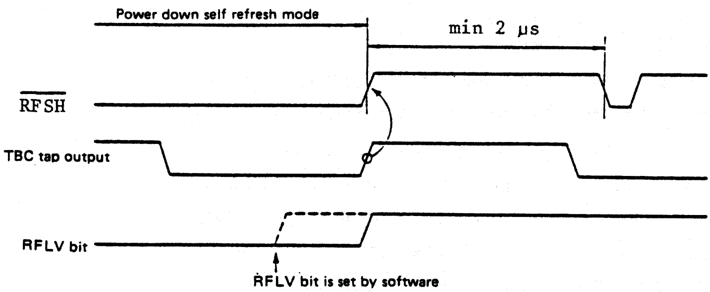
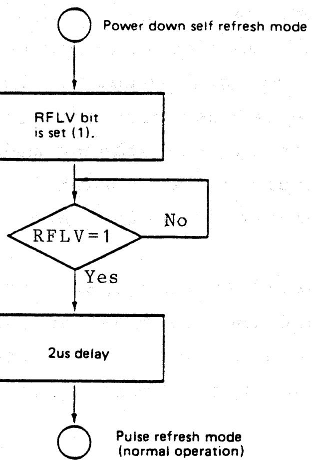


Fig. 6-3 Return from Power Down Self-Refresh Operation

Note:

The following software operation is required when returning from a power down self-refresh operation, since setting of the RFLV bit to "1" by software does not cause immediate effects, and is internally set to "1" in synchronization with a tap output of the time base counter.



(b) Refresh Mode Register (RFM)

This is an 8-bit register used to control switching between the refresh cycle for the pseudostatic RAM and the power down refresh cycle. RFT0 to 1 (bit 0, 1) comprise a bit field used to select the refresh pulse cycle. One of four tap outputs of the time base counter (TBC) can be specified. The RFEN (bit 4) controls the refresh pulse output. When the RFEN bit is reset to "0", the refresh pulses are not output and the $\overline{\text{RFSH}}$ pin is configured as 1 bit output port. When the RFEN bit is set to "1", the $\overline{\text{RFSH}}$ pin goes high level; when reset to "0", it goes low level. When the RFEN bit is set to "1", this bit will act as a bit to select the power down self-refresh mode. When the RFLV bit is set to "1", the pulse-refresh mode becomes effective; when reset to "0", the power down self-refresh mode becomes effective.

The RFM is set to 10H by $\overline{\text{RESET}}$ input to select the power down self-refresh mode.

7	6	5	4	3	2	1	0	
RFLV	0	0	RFEN	0	0	RFT1	RFT0	RFM

RFT1	RFT0	Specification of refresh pulse output period
0	0	16/SLCK (TBC3 tap output of TBC)
0	1	32/SLCK (TBC4 tap output of TBC)
1	0	64/SLCK (TBC5 tap output of TBC)
1	1	128/SLCK (TBC6 tap output of TBC)

RFLV	RFEN	RFSH pin output control
0		Low-level output
1	0	High-level output
0		Power down self-refresh operation $\overline{\text{RFSH}}$ pin becomes low level.
1	1	Refresh pulse output enabled

Fig. 6-4 Refresh Mode Register (RFM) Format

CHAPTER 7 OPERATIONS AFTER RESET

An active low level on the $\overline{\text{RESET}}$ input pin causes a system reset to be performed and each of the CPU system hardware will be brought to the states shown in Table 7-1. The reset condition is released when $\overline{\text{RESET}}$ input returns to high, and then program execution will be resumed.
The contents of each register should be initialized as needed within a program.

Table 7-1 Status of Hardware after Reset

Hardware		States after reset
Program counter (PC)		00H
Stack pointer (SP)		Undefined
Program status word (PSW)		00H
CPU control word (CCW)		00H
Internal RAM	Data memory	Undefined
	General-purpose register (R0 to 15)	
Port	Port register (P0 to 5)	Undefined
	Mode register (PM0 to 3, 5)	FFH (input mode)
	Mode control register (PM2, 3)	0FH
Count unit	Capture compare register (CR00, CR01, CR10, CR11)	Undefined
	Up-down count register (UDC0, 1)	Undefined
	Input mode register (CUIM)	00H
	UDC control register (UDCC0, 1)	00H
	Capture compare register control register (CRC)	00H Undefined

Table 7-1 Status of Hardware after Reset (Cont.)

Hardware		Status after reset
Capture PWM unit	PWM registers (PWM0,1)	Undefined
	FRC control register (FRCC)	00H
	Capture mode register (CPTM)	00H
	PWM mode register (PWMM)	00H
Real-time output port	Control register (RTPC)	08H
	Port 0L register (P0L)	Undefined
	Port 0H register (P0H)	Undefined
Timer unit	Timer registers (TM0, 1)	Undefined
	Modulo/timer registers (MD0, 1)	Undefined
	Timer control registers (TMC0, 1)	00H
A/D converter	Mode register (ADM)	00H
	Conversion result register (ADCR)	Undefined

Peripheral hardware		Status after reset
Serial communication interface	Serial mode register (SCM)	00H
	Serial control register (SCC)	00H
	Baud rate generator settings (BRG)	00H
	Receive buffer register (RxB)	Undefined
	Transmit buffer register (TxB)	Undefined
Time base counter		00H
Time base mode register (TBM)		00H
Standby control register (STBC)		2xH*
Watchdog timer mode register (WDM)		00H

Peripheral hardware		Status after reset
Memory expansion mode (MM)		30H
Refresh mode register (RFM)		10H
Interrupt request	External interrupt mode register (INTM)	00H
	In-service priority register (ISPR)	00H 00H
	Interrupt request control register	47H
	Macro service control register	Undefined

Note: Bit 3 of the STBC is not affected by RESET input, therefore, the lower 4 bits are set to 0 or 8.

CHAPTER 8 INSTRUCTION SET

8.1 Instruction Set and Its Operations

(1) Notations and descriptions of operands

The operands are described in the operand column of each instruction according to the descriptive method for the operand of the appropriate instructions. (Details should be followed by the assembler specification.) One of the elements should be selected from the descriptions in which alternatives exist. Capital letters and +, -, #, $, !, and [ ] are keywords; therefore they should be described as they are. For immediate data, the appropriate numerical values or labels should be described.

Expression	Description
r	R0. R1. R2. R3. R4. R5. R6. R7. R8. R9. R10. R11. R12. R13. R14. R15.
r1	R0. R1. R2. R3. R4. R5. R6. R7
r2	C. B
rp	RP0. RP1. RP2. RP3. RP4. RP5. RP6. RP7.
rp1	RP0. RP1. RP2. RP3. RP4. RP5. RP6. RP7.
rp2	DE. HL. VP. UP
sfr	Special register name (refer to Table 2-1)
sfrp	Special register name (16-bit register: refer to Table 2-1)
post	RP0. RP1. RP2. RP3. RP4. RP5/PSW. RP6. RP7 More than one description are allowed. Note that, however, RP5 is only for PUSH and POP instructions, and PSW is only for PUSHU and POPU instructions. These can also be described in the functional names (such as AX, BC, DE, HL, VP, and UP). Refer to Fig. 2-6 for each correspondent.

Expression	Description
mem	Register indirect mode: [DE], [HL], [DE+], [HL+], [DE-], [HL-], [VP], [UP] Base indexed mode [DE+A], [HL+A], [DE+B], [HL+B], [VP+DE], [VP+HL] Base mode [DE+byte], [HL+byte], [VP+byte], [UP+byte], [SP+byte] Index mode word[A], word[B], word[DE], word[HL]
saddr	FE20H-FF1FH Immediate data or label
saddrp	FE20H-FF1EH Immediate data (bit 0 = 0) or label (16-bit operation)
addr16	0000H-FEFFH Immediate data or label (note that up to FFFFH can be described in the MOV instruction)
addr11	800H-FFFH Immediate data or label
addr5	40H-7EH Immediate data (bit 0 = 0) or label
word	16-bit immediate data
byte	8-bit immediate data
bit	3-bit immediate data or label
n	3-bit immediate data (0-7)

Note:

1. For rp and rpl, the same registers can be described; however, unidentical codes will be generated (see page ____).
2. In this version, instructions having sfrsfrp, -sfrp, and [sfrp] in their operands have been deleted.

(2) Legend used for the operation descriptions:

A	:	A register; 8-bit accumulator
X	:	X register
B	:	B register
C	:	C register
D	:	D register
E	:	E register
H	:	H register
L	:	L register
RO-R15	:	REgister0-register 15 (absolute name)
AX	:	Register pair (AX); 16-bit accumulator
BC	:	Register pair (BC)
DE	:	Register pair (DE)
HL	:	Register pair (HL)
RPO-RP7	:	Register pair 0-register pair 7 (absolute name)
PC	:	Program counter
SP	:	Stack pointer
UP	:	User stack pointer
PSW	:	Program status word
CY	:	Carry flag
AC	:	Additional carry flag
Z	:	Zero flag
P/V	:	Parity/overflow flag
S	:	Sign flag
SUB	:	Subtract flag
TPF	:	Table position flag
RBS	:	Register bank select flag
RES	:	Register set flag
IE	:	Interrupt enable flag
EOS	:	End-of-software interrupt flag
STBC	:	Standby control register
WDM	:	Watchdog timer mode register
( )	:	Contents of the memory shown in the parentheses. (+) and (-) indicate that the contents shown in the parenthesis are incremented and decremented respectively after execution.
(( ))	:	Contents of the memory indicated by the contents of the memory shown in the parentheses.

XXH : Hexadecimal
XH, XL : Higher 8 bits and lower 8 bits
[DE], [HL],

(3) Instruction execution times and symbols in the state column

The numbers shown in the state column indicate the number of states required to execute the instruction. One state is about 167ns when 6MHZ of the internal clock is used. The minimum execution time in this case would be 500ns for 3-state instructions. When n is described in the state column, the value for n is determined as shown in the following:

- Stack operation instructions:
  Number of registers to be stored/loaded
- Shift and locate instructions:
  Number of bits to be shifted
- String instructions
  Number of instruction execution times before exiting loop upon satisfying the conditions.

The numbers in parentheses in the state column on the conditional branch instructions indicate state numbers when respective branching does not occur.

(4) Numbers of states for the instructions containing mem in their operands

The number of states differ depending on the mode described in mem as shown in the following table:

<table>
<tr><th colspan="2">mem / Instruction</th><th>Register indirect mode</th><th>Vector index mode</th><th>Base mode</th><th>Index mode</th></tr>
<tr><td rowspan="2">MOV</td><td>A, mem</td><td rowspan="2">5</td><td rowspan="2">6</td><td rowspan="2">6</td><td rowspan="2">6</td></tr>
<tr><td>mem, A</td></tr>
<tr><td rowspan="2">XCH</td><td>A, mem</td><td rowspan="2">7</td><td rowspan="2">8</td><td rowspan="2">8</td><td rowspan="2">8</td></tr>
<tr><td>mem, A</td></tr>
<tr><td rowspan="2">ADD, ADDC,
SUB, SUBC,
AND, OR,
XOR</td><td>A, mem</td><td>6</td><td>7</td><td>7</td><td>7</td></tr>
<tr><td>mem, A</td><td>7</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td rowspan="2">CMP</td><td>A, mem</td><td rowspan="2">6</td><td rowspan="2">7</td><td rowspan="2">7</td><td rowspan="2">7</td></tr>
<tr><td>mem, A</td></tr>
</table>

) Symbol descriptions for the flag column

Symbol	Description
(Blank)	Not affected
0	Cleared to "0"
1	Set to "1"
X	Set or cleared according to the results
P	P/V flag acts as a parity flag
V	P/V flag acts as a overflow flag
U	Underfined
R	Previously saved values are restored

8-bit registers, cross reference list of special register names and functional names

special name	functional name		special name	functional name	
	RSS = 0	RSS = 1		RSS = 0	RSS = 1
R 0	X		R 8	VP_L	VP_L
R 1	A		R 9	VP_H	VP_H
R 2	C		R 10	UP_L	UP_L
R 3	B		R 11	UP_H	UP_H
R 4		X	R 12	E	E
R 5		A	R 13	D	D
R 6		C	R 14	L	L
R 7		B	R 15	H	H

special name	functional name	
	RSS = 0	RSS = 1
RP0	AX	
RP1	BC	
RP2		AX
RP3		BC
RP4	VP	VP
RP5	UP	UP
RP6	DE	DE
RP7	HL	HL

INSTR GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	S	Z	AC	P/V	SUB	CY
8-bit data transfer instructions	MOV	r1, #byte	2	3	r1 ← byte						
		saddr, #byte	3	3	(saddr) ← byte						
		sfr*, #byte	3	3	sfr ← byte						
		r, r1	2	3	r ← r1						
		A, r1	1	3	A ← r1						
		A, saddr	2	3	A ← (saddr)						
		saddr, A	2	3	(saddr) ← A						
		saddr, saddr	3	4	(saddr) ← (saddr)						
		A, sfr	2	3	A ← sfr						
		sfr, A	2	3	sfr ← A						
		A, mem	1–4	5–6	A ← (mem)						
		mem, A	1–4	5–6	(mem) ← A						
		A, [saddrp]	2	5	A ← ((saddrp))						
		[saddrp], A	2	4	((saddrp)) ← A						
		A, addr16	4	4	A ← (addr16)						
		addr16, A	4	3	(addr16) ← A						
		PSWL, #byte	3	3	PSW_L ← byte	×	×	×	×	×	×
		PSWH, #byte	3	3	PSW_H ← byte						
		PSWL, A	2	3	PSW_L ← A	×	×	×	×	×	×
		PSWH, A	2	3	PSW_H ← A						
		A, PSWL	2	3	A ← PSW_L						
		A, PSWH	2	3	A ← PSW_H						
	XCH	A, r1	1	4	A ↔ r1						
		A, r1	2	4	r ↔ r1						
		A, mem	2–4	7–8	A ↔ (mem)						
		A, saddr	2	4	A ↔ (saddr)						
		A, sfr	3	7	A ↔ sfr						
		A, [saddrp]	2	6	A ↔ ((saddrp))						
		saddr, saddr	3	8	(saddr) ↔ (saddr)						

*: for all sfr except STBC, WDM

INSTR. GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	S	Z	AC	P/V	SUB	CY
16-bit Data Transfer Instruction	MOVW	rpl, #word	3	3	rpl←word						
		saddrp, #word	4	4	(saddrp)←word						
		sfrp, #word	4	3	sfrp←word						
		rp, rpl	2	3	rp←rpl						
		AX, saddrp	2	3	AX← (saddrp)						
		saddrp, AX	2	3	(saddrp)←AX						
		saddrp, saddrp	3	4	(saddrp)←(saddrp)						
		AX, strp	2	3	AX←sfrp						
		sfrp, AX	2	3	sfrp←AX						
	XCHW	AX, saddrp	2	4	AX↔(saddrp)						
		AX, sfrp	3	7	AX↔sfrp						
		saddrp, saddrp	3	8	(saddrp)↔(saddrp)						
		rp, rpl	2	5	rp↔rpl						
8-bit Arithmetic Instruction	ADD	A, #byte	2	3	A, CY←A+byte	×	×	×	V	0	×
		saddr, #byte	3	4	(saddr), CY←(saddr)+byte	×	×	×	V	0	×
		sfr, #byte	4	7	sfr, CY←sfr + byte	×	×	×	V	0	×
		r, r1	2	3	r, CY←r + r1	×	×	×	V	0	×
		A, saddr	2	3	A, CY←A+(saddr)	×	×	×	V	0	×
		A, sfr	3	6	A, CY←A+sfr	×	×	×	V	0	×
		saddr, saddr	3	6	(saddr), CY←(saddr)+(saddr)	×	×	×	V	0	×
		A, mem	2–4	6–7	A, CY←A+(mem)	×	×	×	V	0	×
		mem, A	2–4	7–8	(mem), CY←(mem)+A	×	×	×	V	0	×
	ADDC	A, #byte	2	3	A, CY←A+byte+CY	×	×	×	V	0	×
		saddr, #byte	3	4	(saddr), CY←(saddr)+byte+CY	×	×	×	V	0	×
		sfr, #byte	4	7	sfr, CY←sfr+byte+CY	×	×	×	V	0	×
		r, r1	2	3	r, CY←r+r1+CY	×	×	×	V	0	×
		A, saddr	2	3	A, CY←A+(saddr)+CY	×	×	×	V	0	×
		A, sfr	3	6	A, CY←A+sfr+CY	×	×	×	V	0	×
		saddr, saddr	3	6	(saddr), CY←(saddr)+(saddr)+CY	×	×	×	V	0	×
		A, mem	2–4	6–7	A, CY←A+(mem)+CY	×	×	×	V	0	×
		mem, A	2–4	7–8	(mem), CY←(mem)+A+CY	×	×	×	V	0	×

INSTRUCTION GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	FLAG S	Z	AC	P/V	SUB	CY
8-bit Arithmetic Instruction	SUB	A, #byte	2	3	A, CY←A－byte	×	×	×	V	1	×
		saddr, #byte	3	4	(saddr), CY←(saddr)－byte	×	×	×	V	1	×
		sfr, #byte	4	7	sfr, CY←sfr－byte	×	×	×	V	1	×
		r, r1	2	3	r, CY←r－r1	×	×	×	V	1	×
		A, saddr	2	3	A, CY←A－(saddr)	×	×	×	V	1	×
		A, sfr	3	6	A, CY←A－sfr	×	×	×	V	1	×
		saddr, saddr	3	6	(saddr), CY←(saddr)－(saddr)	×	×	×	V	1	×
		A, mem	2–4	6–7	A, CY←A－(mem)	×	×	×	V	1	×
		mem, A	2–4	7–8	(mem), CY←(mem)－A	×	×	×	V	1	×
	SUBC	A, #byte	2	3	A, CY←A－byte－CY	×	×	×	V	1	×
		saddr, #byte	3	4	(saddr), CY←(saddr)－byte－CY	×	×	×	V	1	×
		sfr, #byte	4	7	sfr, CY←sfr－byte－CY	×	×	×	V	1	×
		r, r1	2	3	r, CY←r－r1－CY	×	×	×	V	1	×
		A, saddr	2	3	A, CY←A－(saddr)－CY	×	×	×	V	1	×
		A, sfr	3	6	A, CY←A－sfr－CY	×	×	×	V	1	×
		saddr, saddr	3	6	(saddr), CY←(saddr)－(saddr)－CY	×	×	×	V	1	×
		A, mem	2–4	6–7	A, CY←A－(mem)－CY	×	×	×	V	1	×
		mem, A	2–4	7–8	(mem), CY←(mem)－A－CY	×	×	×	V	1	×
	AND	A, #byte	2	3	A←A∧byte	×	×		P	0	
		saddr, #byte	3	4	(saddr)←(saddr)∧byte	×	×		P	0	
		sfr, #byte	4	7	sfr←sfr∧byte	×	×		P	0	
		r, r1	2	3	r←r∧r1	×	×		P	0	
		A, saddr	2	3	A←A∧(saddr)	×	×		P	0	
		A, sfr	3	6	A←A∧sfr	×	×		P	0	
		saddr, saddr	3	6	(saddr)←(saddr)∧(saddr)	×	×		P	0	
		A, mem	2–4	6–7	A←A∧(mem)	×	×		P	0	
		mem, A	2–4	7–8	(mem)←(mem)∧A	×	×		P	0	
	OR	A, #byte	2	3	A←A∨byte	×	×		P	0	
		saddr, #byte	3	4	(saddr)←(saddr)∨byte	×	×		P	0	
		sfr, #dyte	4	7	sfr←sfr∨byte	×	×		P	0	
		r, r1	2	3	r←r∨r1	×	×		P	0	
		A, saddr	2	3	A←A∨(saddr)	×	×		P	0	
		A, sfr	3	6	A←A∨sfr	×	×		P	0	
		saddr, saddr	3	6	(saddr)←(saddr)∨(saddr)	×	×		P	0	
		A, mem	2–4	6–7	A←A∨(mem)	×	×		P	0	
		mem, A	2–4	7–8	(mem)←(mem)∨A	×	×		P	0	

INSTR. GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	FLAG S	Z	AC	P/V	SUB	CY
8-bit Arithmetic Instruction	XOR	A, #byte	2	3	A←A∀byte	×	×		P	0	
		saddr, #byte	3	4	(saddr)←(saddr)∀byte	×	×		P	0	
		sfr, #byte	4	7	sfr←sfr∀byte	×	×		P	0	
		r, r1	2	3	r←r∀r1	×	×		P	0	
		A, saddr	2	3	A←A∀(saddr)	×	×		P	0	
		A, sfr	3	6	A←A∀sfr	×	×		P	0	
		saddr, saddr	3	6	(saddr)←(saddr)∀(saddr)	×	×		P	0	
		A, mem	2–4	6–7	A←A∀(mem)	×	×		P	0	
		mem, A	2–4	7–8	(mem)←(mem)∀A	×	×		P	0	
	CMP	A, #byte	2	3	A−byte	×	×	×	V	1	×
		saddr, #byte	3	4	(saddr)−byte	×	×	×	V	1	×
		sfr, #byte	4	7	sfr−byte	×	×	×	V	1	×
		r, r1	2	3	r−r1	×	×	×	V	1	×
		A, saddr	2	3	A−(saddr)	×	×	×	V	1	×
		A, sfr	3	6	A−sfr	×	×	×	V	1	×
		saddr, saddr	3	6	(saddr)−(saddr)	×	×	×	V	1	×
		A, mem	2–4	6–7	A−(mem)	×	×	×	V	1	×
		mem, A	2–4	6–7	(mem)−A	×	×	×	V	1	×

INSTR. GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	FLAG S	Z	AC	P/V	SUB	CY
16-bit Arithmetic Instruction	ADDW	AX, #word	3	4	AX, CY←AX+word	×	×		V	0	×
		saddrp, #word	4	5	(saddrp), CY←(saddrp)+word	×	×		V	0	×
		sfrp, #word	5	8	sfrp, CY←sfrp+word	×	×		V	0	×
		rp, rp1	2	4	rp, CY←rp+rp1	×	×		V	0	×
		AX, saddrp	2	4	AX, CY←AX+(saddrp)	×	×		V	0	×
		AX, sfrp	3	7	AX, CY←AX+sfrp	×	×		V	0	×
		saddrp, saddrp	3	6	(saddrp), CY←(saddrp)+(saddrp)	×	×		V	0	×
	SUBW	AX, #word	3	4	AX, CY←AX−word	×	×		V	1	×
		saddrp, #word	4	5	(saddrp), CY←(saddrp)−word	×	×		V	1	×
		sfrp, #word	5	8	sfrp, CY←sfrp−word	×	×		V	1	×
		rp, rp1	2	4	rp, CY←rp−rp1	×	×		V	1	×
		AX, saddrp	2	4	AX, CY←AX−(saddrp)	×	×		V	1	×
		AX, sfrp	3	7	AX, CY←AX−sfrp	×	×		V	1	×
		saddrp, saddrp	3	6	(saddrp), CY←(saddrp)−(saddrp)	×	×		V	1	×
	CMPW	AX, #word	3	4	AX−word	×	×		V	1	×
		saddrp, #word	4	5	(saddrp)−word	×	×		V	1	×
		sfrp, #word	5	8	strp−word	×	×		V	1	×
		rp, rp1	2	4	rp−rp1	×	×		V	1	×
		AX, saddrp	2	4	AX−(saddrp)	×	×		V	1	×
		AX, sfrp	3	7	AX−sfrp	×	×		V	1	×
		saddrp, saddrp	3	6	(saddrp)−(saddrp)	×	×		V	1	×
Multiply/divide Instructions	MULU	r1	2	18	AX←A×r1						
	DIVU	r1	2	18	AX(商), r1(余り)←AX÷r1						
	MULUW	rp1	2	27	AX(上位16ビット), rp1(下位16ビット) ←AX×rp1						
	DIVUX	rp1	2	50	AXDE(商), rp1(余り)←AXDE÷rp1						

NEC

INSTRUCTION GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	S	Z	AC	P/V	SUB	CY
Increment/Decrement Instruction	INC	r1	1	3	r1 ← r1 + 1	×	×	×	V	0	
		saddr	2	4	(saddr) ← (saddr) + 1	×	×	×	V	0	
	DEC	r1	1	3	r1 ← r1 − 1	×	×	×	V	1	
		saddr	2	4	(saddr) ← (saddr) − 1	×	×	×	V	1	
	INCW	rp2	1	3	rp2 ← rp2 + 1						
		saddrp	3	6	(saddrp) ← (saddrp) + 1						
	DECW	rp2	1	3	rp2 ← rp2 − 1						
		saddrp	3	6	(saddrp) ← (saddrp) − 1						
Shift / Rotate Instructions	ROR	r1, n	2	4+3n	(CY, $r1_7$ ← $r1_0$, $r1_{m-1}$ ← $r1_m$) × n 回				P	0	×
	ROL	r1, n	2	4+3n	(CY, $r1_0$ ← $r1_7$, $r1_{m+1}$ ← $r1_m$) × n 回				P	0	×
	RORC	r1, n	2	4+3n	(CY ← $r1_0$, $r1_7$ ← CY, $r1_{m-1}$ ← $r1_m$) × n 回				P	0	×
	ROLC	r1, n	2	4+3n	(CY ← $r1_7$, $r1_0$ ← CY, $r1_{m+1}$ ← $r1_m$) × n 回				P	0	×
	SHR	r1, n	2	4+3n	(CY ← $r1_0$, $r1_7$ ← 0, $r1_{m-1}$ ← $r1_m$) × n 回	×	×	0	P	0	×
	SHL	r1, n	2	4+3n	(CY ← $r1_7$, $r1_0$ ← 0, $r1_{m+1}$ ← $r1_m$) × n 回	×	×	0	P	0	×
	SHRW	rp, n	2	4+3n	(CY ← rp_0, rp_{15} ← 0, rp_{m-1} ← rp_m) × n 回	×	×	0	P	0	×
	SHLW	rp, n	2	4+3n	(CY ← rp_{15}, rp_0 ← 0, rp_{m+1} ← rp_m) × n 回	×	×	0	P	0	×
	ROR4	[rp1]	2	8	A_{3-0} ← $(rp1)_{3-0}$, $(rp1)_{7-4}$ ← A_{3-0}, $(rp1)_{3-0}$ ← $(rp1)_{7-4}$						
	ROL4	[rp1]	2	8	A_{3-0} ← $(rp1)_{7-4}$, $(rp1)_{3-0}$ ← A_{3-0}, $(rp1)_{7-4}$ ← $(rp1)_{3-0}$						
BCD Adjustment	ADJ4		1	3	Decimal Adjust Accumulator	×	×	×	P		×

INSTR. GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	S	Z	AC	P/V	SUB	CY
Bit Operation Instructions	MOV1	CY, saddr. dit	3	6	CY←(saddr. bit)						×
		CY, sfr. bit	3	6	CY← sfr. bit						×
		CY, A. bit	2	6	CY←A. bit						×
		CY, X. bit	2	6	CY←X. bit						×
		CY, PSWH. bit	2	6	CY←PSW_H. bit						×
		CY, PSWL. bit	2	6	CY←PSW_L . bit						×
		saddr. bit, CY	3	7	(saddr. bit)←CY						
		sfr. bit, CY	3	7	sfr. bit←CY						
		A. bit, CY	2	8	A. bit←CY						
		X. bit, CY	2	8	X. bit←CY						
		PSWH. bit, CY	2	8	PSW_H. bit←CY						
		PSWL. bit, CY	2	8	PSW_L. bit←CY						
	AND1	CY, saddr. bit	3	6	CY←CY∧(saddr. bit)						×
		CY, /saddr. bit	3	6	CY←CY∧$\overline{\text{(saddr. bit)}}$						×
		CY, sfr. bit	3	6	CY←CY∧sfr. bit						×
		CY, /sfr. bit	3	6	CY←CY∧$\overline{\text{sfr. bit}}$						×
		CY, A. bit	2	6	CY←CY∧A. bit						×
		CY, /A. bit	2	6	CY←CY∧$\overline{\text{A. bit}}$						×
		CY, X. bit	2	6	CY←CY∧X. bit						×
		CY, /X. bit	2	6	CY←CY∧$\overline{\text{X. bit}}$						×
		CY, PSWH. bit	2	6	CY←CY∧PSW_H. bit						×
		CY, /PSWH. bit	2	6	CY←CY∧$\overline{PSW_H\text{. bit}}$						×
		CY, PSWL. bit	2	6	CY←CY∧PSW_L. bit						×
		CY, /PSWL. bit	2	6	CY←CY∧$\overline{PSW_L\text{. bit}}$						×
	OR1	CY, saddr. bit	3	6	CY←CY∨(saddr. bit)						×
		CY, /saddr. bit	3	6	CY←CY∨$\overline{\text{(saddr. bit)}}$						×
		CY, sfr. bit	3	6	CY←CY∨sfr. bit						×
		CY, /sfr. bit	3	6	CY←CY∨$\overline{\text{sfr. bit}}$						×
		CY, A. bit	2	6	CY←CY∨A. bit						×
		CY, /A. bit	2	6	CY←CY∨$\overline{\text{A. bit}}$						×
		CY, X. bit	2	6	CY←CY∨X. bit						×
		CY, /X. bit	2	6	CY←CY∨$\overline{\text{X. bit}}$						×
		CY, PSWH. bit	2	6	CY←CY∨PSW_H. bit						×
		CY, /PSWH. bit	2	6	CY←CY∨$\overline{PSW_H\text{. bit}}$						×
		CY, PSWL. bit	2	6	CY←CY∨PSW_L. bit						×
		CY, /PSWL. bit	2	6	CY←CY∨$\overline{PSW_L\text{. bit}}$						×

INSTR. GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	S	Z	AC	P/V	SUB	CY
Bit Operation Instructions	XOR1	CY, saddr. bit	3	6	CY←CY∀(saddr. bit)						×
		CY, sfr. bit	3	6	CY←CY∀sfr. bit						×
		CY, A. bit	2	6	CY←CY∀A. bit						×
		CY, X. bit	2	6	CY←CY∀X. bit						×
		CY, PSWH. bit	2	6	CY←CY∀PSW_H. bit						×
		CY, PSWL. bit	2	6	CY←CY∀PSW_L. bit						×
	SET1	saddr. bit	2	5	(saddr. bit)←1						
		sfr. bit	3	6	sfr. bit←1						
		A. bit	2	7	A. bit←1						
		X. bit	2	7	X. bit←1						
		PSWH. bit	2	7	PSW_H. bit←1						
		PSWL bit	2	7	PSW_L. bit←1						
	CLR1	saddr. bit	2	5	(saddr. bit)←0						
		sfr. bit	3	6	sfr. bit←0						
		A. bit	2	7	A. bit←0						
		X. bit	2	7	X. bit←0						
		PSWH. bit	2	7	PSW_H. bit←0						
		PSWL. bit	2	7	PSW_L. bit←0						
	NOT1	saddr. bit	3	6	(saddr. bit)←$\overline{\text{(saddr. bit)}}$						
		sfr. bit	3	6	sfr. bit←$\overline{\text{sfr. bit}}$						
		A. bit	2	7	A. bit←$\overline{\text{A. bit}}$						
		X. bit	2	7	X. bit←$\overline{\text{X. bit}}$						
		PSWH. bit	2	7	PSW_H. bit←$\overline{PSW_H\text{. bit}}$						
		PSWL. bit	2	7	PSW_L. bit←$\overline{PSW_L\text{. bit}}$						
	SET1	CY	1	3	CY←1						1
	CLR1	CY	1	3	CY←0						0
	NOT1	CY	1	3	CY←$\overline{\text{CY}}$						×

INSTR. GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	S	Z	AC	P/V	SUB	CY
CALL/RETURN Instructions	CALL	!addr16	3	8	(SP−1)←$(PC+3)_H$, (SP−2)←$(PC+3)_L$, PC←addr16, SP←SP−2						
	CALLF	!addr11	2	8	(SP−1)←$(PC+2)_H$, (SP−2)←$(PC+2)_L$, PC_{15-11}←00001, PC_{10-0}←addr11, SP←SP−2						
	CALLT	[addr5]	1	10	PC_H←(TPF, 00000000, addr5+1), PC_L←(TPF, 00000000, addr5), SP←SP−2						
	CALL	rp1	2	13	(SP−1)←$(PC+2)_H$, (SP−2)←$(PC+2)_L$, PC_H←$rp1_H$, PC_L←$rp1_L$, SP←SP−2						
		[rp1]	2	11	(SP−1)←$(PC+2)_H$, (SP−2)←$(PC+2)_L$, PC_H←$(rp1)_H$, PC_L←$(rp1)_L$, SP←SP−2						
	BRK		1	16	(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						
	RET		1	8	PC_L←(SP), PC_H←(SP+1), SP←SP+2						
	RETI		1	14	PC_L←(SP), PC_H←(SP+1), PSW_L←(SP+2), PSW_H←(SP+3), SP←SP+4, EOS←0	R	R	R	R	R	R
STack Operation Instructions	PUSH	post	2	7+8n	{(SP−1)←$post_H$, (SP−2)←$post_L$, SP←SP−2}×n *						
		PSW	1	5	(SP−1)←PSW_H, (SP−2)←PSW_L, SP←SP−2						
	PUSHU	post	2	8+8n	{(UP−1)←$post_H$, (UP−2)←$post_L$, UP←UP−2}×n *						
	POP	post	2	7+8n	{$post_L$←(SP), $post_H$←(SP+1), SP←SP+2}×n *						
		PSW	1	5	PSW_L←(SP), PSW_H←(SP+1), SP←SP+2	R	R	R	R	R	R
	POPU	post	2	8+8n	{$post_L$←(UP), $post_H$←(UP+1), UP←UP+2}×n *						
	MOVW	SP, #word	4	3	SP←word						
		SP, AX	2	3	SP←AX						
		AX, SP	2	3	AX←SP						
	INCW	SP	2	6	SP←SP+1						
	DECW	SP	2	6	SP←SP−1						

*: n = number of transfers depending on value of post

NEC

INSTR. GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	FLAG S Z AC P/V SUB CY
uncond. branch instructions	BR	!addr16	3	4	PC←addr16	
		rp1	2	6	PC_H←$rp1_H$, PC_L←$rp1_L$	
		(rp1)	2	9	PC_H←$(rp1)_H$, PC_L←$(rp1)_L$	
		$addr16	2	7	PC←addr16	
conditional branch instructions	BC BL	$addr16	2	7 (3)	PC←addr16 if CY=1	
	BNC BNL	$addr16	2	7 (3)	PC←addr16 if CY=0	
	BZ BE	$addr16	2	7 (3)	PC←addr16 if Z=1	
	BNZ BNE	$addr16	2	7 (3)	PC←addr16 if Z=0	
	BV BPE	$addr16	2	7 (3)	PC←addr16 if P/V=1	
	BNV BPO	$addr16	2	7 (3)	PC←addr16 if P/V=0	
	BN	$addr16	2	7 (3)	PC←addr16 if S=1	
	BP	$addr16	2	7 (3)	PC←addr16 if S=0	
	BGT	$addr16	3	9 (5)	PC←addr16 if $(P/V \veebar S) \vee Z=0$	
	BGE	$addr16	3	9 (5)	PC←addr16 if $P/V \veebar S=0$	
	BLT	$addr16	3	9 (5)	PC←addr16 if $P/V \veebar S=1$	
	BLE	$addr16	3	9 (5)	PC←addr16 if $(P/V \veebar S) \vee Z=1$	
	BH	$addr16	3	9 (5)	PC←addr16 if $Z \vee CY=0$	
	BNH	$addr16	3	9 (5)	PC←addr16 if $Z \vee CY=1$	
	BT	saddr.bit, $addr16	3	9 (7)	PC←addr16 if (saddr. bit)=1	
		sfr.bit, $addr16	4	10(7)	PC←addr16 if sfr.bit=1	
		A.bit, $addr 16	3	10(7)	PC←addr16 if A. bit=1	
		X. bit, $addr 16	3	10(7)	PC←addr16 if X. bit=1	
		PSWH.bit, $addr16	3	10(7)	PC←addr16 if PSW_H. bit=1	
		PSWL.bit, $addr16	3	10(7)	PC←addr16 if PSW_L. bit=1	
	BF	saddr.bit, $addr16	4	10(7)	PC←addr16 if (saddr. bit)=0	
		sfr.bit, $addr16	4	10(7)	PC←addr16 if sfr.bit=0	
		A. bit, $addr16	3	10(7)	PC←addr16 if A. bit=0	
		X. bit, $addr16	3	10(7)	PC←addr16 if X. bit=0	
		PSWH. bit, $addr16	3	10(7)	PC←addr16 if PSW_H. bit=0	
		PSWL. bit, $addr16	3	10(7)	PC←addr16 if PSW_L. bit=0	

INSTRUCTION GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	FLAG S Z AC P/V SUB CY
Conditional Branch Instructions	BTCLR	saddr. bit, \$addr16	4	12 (7)	PC←addr16 if(saddr. bit)=1 then reset(saddr. bit)	
		sfr. bit, \$addr16	4	12 (7)	PC←addr16 if sfr. bit=1 then reset sfr. bit	
		A. bit, \$addr16	3	11 (7)	PC←addr16 if A. bit=1 then reset A. bit	
		X. bit, \$addr16	3	11 (7)	PC←addr16 if X. bit=1 then reset X. bit	
		PSWH. bit, \$addr16	3	12 (7)	PC←addr16 if PSW_H. bit=1 then reset PSW_H. bit	
		PSWL. bit, \$addr16	3	12 (7)	PC←addr16 if PSW_L. bit=1 then reset PSW_L. bit	
	BFSET	saddr. bit, \$addr16	4	12 (7)	PC←addr16 if(saddr. bit)=0 then set(saddr. bit)	
		sfr. bit, \$addr16	4	12 (7)	PC←addr16 if sfr. bit=0 then set sfr. bit	
		A. bit, \$addr16	3	11 (7)	PC←addr16 if A. bit=0 then set A. bit	
		X. bit, \$addr16	3	11 (7)	PC←addr16 if X. bit=0 then set X. bit	
		PSWH. bit, \$addr16	3	12 (7)	PC←addr16 if PSW_H. bit=0 then set PSW_H. bit	
		PSWL. bit, \$addr16	3	12 (7)	PC←addr16 if PSW_L. bit=0 then set PSW_L. bit	
	DBNZ	r2, \$addr16	2	8 (5)	r2←r2−1, then PC←addr16 if r2≠0	
		saddr, \$addr16	3	7 (6)	(saddr)←(saddr)−1, then PC←addr16 if(saddr)≠0	
Context Switching Instructions	BRKCS	RBn	2	13	PC_H↔R5, PC_L↔R4, R7←PSW_H R6←PSW_L, RBS2−0←n, RSS←0, IE←0	
	RETCS	!addr16	3	6	PC_H←R5, PC_L←R4, R4, R5←(addr16), PSW_H←R7, PSW_L←R6, EOS←0	

INSTRUCTION GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	S	Z	AC	P/V	SUB	CY
String Operation Instructions	MOVM	[DE+], A	2	2+7n	(DE+)←A, C←C−1 End if C=0						
		[DE−], A	2	2+7n	(DE−)←A, C←C−1 End if C=0						
	MOVBK	[DE+], [HL+]	2	2+10n	(DE+)←(HL+), C←C−1 End if C=0						
		[DE−], [HL−]	2	2+10n	(DE−)←(HL−), C←C−1 End if C=0						
	XCHM	[DE+], A	2	2+12n	(DE+)↔A, C←C−1 End if C=0						
		[DE−], A	2	2+12n	(DE−)↔A, C←C−1 End if C=0						
	XCHBK	[DE+], [HL+]	2	2+15n	(DE+)↔(HL+), C←C−1 End if C=0						
		[DE−], [HL−]	2	2+15n	(DE−)↔(HL−), C←C−1 End if C=0						
	CMPME	[DE+], A	2	2+8n	(DE+)−A, C←C−1 End if C=0 or Z=0	×	×	×	V	1	×
		[DE−], A	2	2+8n	(DE−)−A, C←C−1 End if C=0 or Z=0	×	×	×	V	1	×
	CMPBKE	[DE+], [HL+]	2	2+11n	(DE+)−(HL+), C←C−1 End if C=0 or Z=0	×	×	×	V	1	×
		[DE−], [HL−]	2	2+11n	(DE−)−(HL−), C←C−1 End if C=0 or Z=0	×	×	×	V	1	×
	CMPMNE	[DE+], A	2	2+8n	(DE+)−A, C←C−1 End if C=0 or Z=1	×	×	×	V	1	×
		[DE−], A	2	2+8n	(DE−)−A, C←C−1 End if C=0 or Z=1	×	×	×	V	1	×
	CMPBKNE	[DE+], [HL+]	2	2+11n	(DE+)−(HL+), C←C−1 End if C=0 or Z=1	×	×	×	V	1	×
		[DE−], [HL−]	2	2+11n	(DE−)−(HL−), C←C−1 End if C=0 or Z=1	×	×	×	V	1	×
	CMPMC	[DE+], A	2	2+8n	(DE+)−A, C←C−1 End if C=0 or CY=0	×	×	×	V	1	×
		[DE−], A	2	2+8n	(DE−)−A, C←C−1 End if C=0 or CY=0	×	×	×	V	1	×

INSTRUCTION GROUP	MNEMONIC	OPERAND	BYTE	STATES	OPERATION	S	Z	AC	P/V	SUB	CY
String Operation Instructions	CMPBKC	[DE+], [HL+]	2	2+11n	(DE+)−(HL+), C←C−1 End if C=0 or CY=0	×	×	×	V	1	×
		[DE−], [HL−]	2	2+11n	(DE−)−(HL−), C←C−1 End if C=0 or CY=0	×	×	×	V	1	×
	CMPMNC	[DE+], A	2	2+8n	(DE+)−A, C←C−1 End if C=0 or CY=1	×	×	×	V	1	×
		[DE−], A	2	2+8n	(DE−)−A, C←C−1 End if C=0 or CY=1	×	×	×	V	1	×
	CMPBKNC	[DE+], [HL+]	2	2+11n	(DE+)−(HL+), C←C−1 End if C=0 or CY=1	×	×	×	V	1	×
		[DE−], [HL−]	2	2+11n	(DE−)−(HL−), C←C−1 End if C=0 or CY=1	×	×	×	V	1	×
CPU control instruction	MOV	STBC, #byte	4	5	STBC←byte						
		WDM, #byte	4	5	WDM←byte						
	SWRS		1	3	RSS←$\overline{\text{RSS}}$						
	SEL	RBn	2	3	RSS←0, RBS2−0←n						
		RBn, ALT	2	3	RSS←1, RBS2−0←n						
	NOP		1	3	No Operation						
	EI		1	3	IE←1 (Enable Interrupt)						
	DI		1	3	IE←0 (Disable Interrupt)						

9.2 Instruction Code for each Instruction

(1) Description of Instruction codes

r, r1

R_3	R_2	R_1	R_0	reg
0	0	0	0	R0
0	0	0	1	R1
0	0	1	0	R2
0	0	1	1	R3
0	1	0	0	R4
0	1	0	1	R5
0	1	1	0	R6
0	1	1	1	R7
1	0	0	0	R8
1	0	0	1	R9
1	0	1	0	R10
1	0	1	1	R11
1	1	0	0	R12
1	1	0	1	R13
1	1	1	0	R14
1	1	1	1	R15

r1

r

r2

C	reg
0	C
1	B

rp

P_2	P_1	P_0	reg-pair
0	0	0	RP0
0	0	1	RP1
0	1	0	RP2
0	1	1	RP3
1	0	0	RP4
1	0	1	RP5
1	1	0	RP6
1	1	1	RP7

rp1

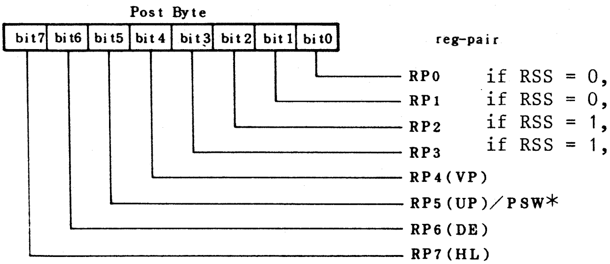
Q_2	Q_1	Q_0	reg-pair
0	0	0	RP0
0	0	1	RP4
0	1	0	RP1
0	1	1	RP5
1	0	0	RP2
1	0	1	RP6
1	1	0	RP3
1	1	1	RP7

rp2

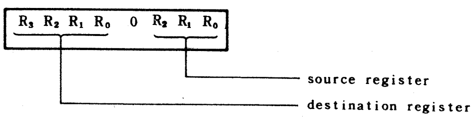
S_1	S_0	reg-pair
0	0	VP
0	1	UP
1	0	DE
1	1	HL

Bn	:	immediate data corresponding to bit
Nn	:	immediate data corresponding to n
Data	:	8-bit immediate data corresponding to byte
Low/High Byte	:	16-bit immediate data corresponding to word
Saddr-offset	:	Lower 8-bit offset data of 16-bit address corresponding to Saddr.

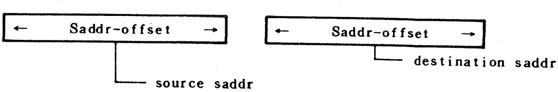
Sfr-offset	:	Lower 8-bit of 16 bit address in sfr area
Low/High offset	:	8-/16-bit offset data for memory addressing with Base addressing / Base Index addressing
Low/High Addr.	:	16-bit immediate data corresponding to addr16
jdisp	:	immediate value, taken as two's complement to define distance between 8-bit address of next instruction and branch address
fa	:	Lower 11-bit of immediate data corresponding to addr.11
ta	:	Lower 5-bit of immediate data corresponding to (addr.5x1/2)
Post Byte	:	8-bit data defining reg-pair for stacking operation each bit defines a reg-pair with its value



MOV r, r1



ADD saddr, saddr.



mem \ mod	1 0110 Register Indirect Mode	1 0111 Base Mode	0 0110 Base Mode	0 1010 Index Mode
0 0 0	[DE+] *	[DE+A]	[DE+byte]	word[DE]
0 0 1	[HL+] *	[HL+A]	[SP+byte]	word[A]
0 1 0	[DE−] *	[DE+B]	[HL+byte]	word[HL]
0 1 1	[HL−] *	[HL+B]	[UP+byte]	word[B]
1 0 0	[DE] *	[VP+DE]	[VP+byte]	—
1 0 1	[HL] *	[VP+HL]	—	—
1 1 0	[VP]	—	—	—
1 1 1	[UP]	—	—	—

INSTRUCTION GROUP	MNEMONIC	OPERAND	Instruction Code B1 / B4	B2 / B5	B3
8-bit Data Transfer Instructions	MOV	r1, #byte	1 0 1 1 1 $R_2R_1R_0$	← Data →	
		saddr, #byte	0 0 1 1 1 0 1 0	← Saddr-offset →	← Data →
		sfr, #byte	0 0 1 0 1 0 1 1	← Sfr-offset →	← Data →
		r, r1	0 0 1 0 0 1 0 0	$R_3R_2R_1R_0$ 0 $R_2R_1R_0$	
		A, r1	1 1 0 1 0 $R_2R_1R_0$		
		A, saddr	0 0 1 0 0 0 0 0	← Saddr-offset →	
		saddr, A	0 0 1 0 0 0 1 0	← Sfr-offset →	
		saddr, saddr	0 0 1 1 1 0 0 0	← Saddr-offset →	← Saddr-offset →
		A, sfr	0 0 0 1 0 0 0 0	← Sfr-offset →	
		sfr, A	0 0 0 1 0 0 1 0	← Sfr-offset →	
		A, mem	0 1 0 1 1 mem		
			0 0 0 mod	0 mem 0 0 0 0	← Low Offset →
			← High Offset →		
		mem, A	0 1 0 1 0 mem		
			0 0 0 mod	1 mem 0 0 0 0	← Low Offset →
			← High Offset →		
		A, [saddrp]	0 0 0 1 1 0 0 0	← Saddr-offset →	
		[saddrp], A	0 0 0 1 1 0 0 1	← Saddr-offset →	
		A, addr16	0 0 0 0 1 0 0 1	1 1 1 1 0 0 0 0	← Low Addr. →
			← High Addr. →		
		addr16, A	0 0 0 0 1 0 0 1	1 1 1 1 0 0 0 1	← Low Addr. →
			← High Addr. →		
		PSWL, #byte	0 0 1 0 1 0 1 1	1 1 1 1 1 1 1 0	← Data →
		PSWH, #byte	0 0 1 0 1 0 1 1	1 1 1 1 1 1 1 1	← Data →
		PSWL, A	0 0 0 1 0 0 1 0	1 1 1 1 1 1 1 0	
		PSWH, A	0 0 0 1 0 0 1 0	1 1 1 1 1 1 1 1	
		A, PSWL	0 0 0 1 0 0 0 0	1 1 1 1 1 1 1 0	
		A, PSWH	0 0 0 1 0 0 0 0	1 1 1 1 1 1 1 1	

INSTR. GROUP	MNEMONIC	OPERAND	Instruction Code B1 / B4	B2 / B5	B3
8-bit Data Transfer Instructions	XCH	A, rl	1 1 0 1 1 $R_2R_1R_0$		
		r, rl	0 0 1 0 0 1 0 1	$R_3R_2R_1R_0$ 0 $R_2R_1R_0$	
		A, mem	0 0 0 mod	0 mem 0 1 0 0	← Low Offset →
			← High Offset →		
		A, saddr	0 0 1 0 0 0 0 1	← Saddr-offset →	
		A, sfr	0 0 0 0 0 0 0 1	0 0 1 0 0 0 0 1	← Sfr-offset →
		A, [saddrp]	0 0 1 0 0 0 1 1	← Saddr-offset →	
		saddr, saddr	0 0 1 1 1 0 0 1	← Saddr-offset →	← Saddr-offset →
16-bit Data Transfer Instructions	MOVW	rpl, #word	0 1 1 0 0 $Q_2Q_1Q_0$	← Low Byte →	← Hight Byte →
		saddrp, #word	0 0 0 0 1 1 0 0	← Saddr-offset →	← Low Byte →
			← High Byte →		
		sfrp, #word	0 0 0 0 1 0 1 1	← Sfr-offset →	← Low Byte →
			← High Byte →		
		rp, rpl	0 0 1 0 0 1 0 0	$P_2P_1P_0$ 0 1 $Q_2Q_1Q_0$	
		AX, saddrp	0 0 0 1 1 1 0 0	← Saddr-offset →	
		saddrp, AX	0 0 0 1 1 0 1 0	← Saddr-offset →	
		saddrp, saddrp	0 0 1 1 1 1 0 0	← Saddr-offset →	← Saddr-offset →
		AX, sfrp	0 0 0 1 0 0 0 1	← Sfr-offset →	
		sfrp, AX	0 0 0 1 0 0 1 1	← Sfr-offset →	
	XCHW	AX, saddrp	0 0 0 1 1 0 1 1	← Saddr-offset →	
		AX, sfrp	0 0 0 0 0 0 0 1	0 0 0 1 1 0 1 1	← Sfr-offset →
		saddrp, saddrp	0 0 1 0 1 0 1 0	← Saddr-offset →	← Saddr-offset →
		rp, rpl	0 0 1 0 0 1 0 1	$P_2P_1P_0$ 0 1 $Q_2Q_1Q_0$	
8-bit Arithmetic Instruction	ADD	A, #byte	1 0 1 0 1 0 0 0	← Data →	
		saddr, #byte	0 1 1 0 1 0 0 0	← Saddr-offset →	← Data →
		sfr, #byte	0 0 0 0 0 0 0 1	0 1 1 0 1 0 0 0	← Sfr-offset →
			← Data →		
		r, rl	1 0 0 0 1 0 0 0	$R_3R_2R_1R_0$ 0 $R_2R_1R_0$	
		A, saddr	1 0 0 1 1 0 0 0	← Saddr-offset →	
		A, sfr	0 0 0 0 0 0 0 1	1 0 0 1 1 0 0 0	← Sfr-offset →
		saddr, saddr	0 1 1 1 1 0 0 0	← Saddr-offset →	← Saddr-offset →
		A, mem	0 0 0 mod	0 mem 1 0 0 0	← Low Offset →
			← High Offset →		
		mem, A	0 0 0 mod	1 mem 1 0 0 0	← Low Offset →
			← High Offset →		

INSTR. GROUP	MNEMONIC	OPERAND	Instruction Code B1 / B4	B2 / B5	B3
8-bit Arithmetic Instructions	ADDC	A, #byte	1010 1001	← Data →	
		saddr, #byte	0110 1001	← Saddr-offset →	← Data →
		sfr, #byte	0000 0001	0110 1001	← Sfr-offset →
			← Data →		
		r, r1	1000 1001	$R_3R_2R_1R_0$ 0 $R_2R_1R_0$	
		A, saddr	1001 1001	← Saddr-offset →	
		A, sfr	0000 0001	1001 1001	← Sfr-offset →
		saddr, saddr	0111 1001	← Saddr-offset →	← Saddr-offset →
		A, mem	000 mod	0 mem 1001	← Low Offset →
			← High Offset →		
		mem, A	000 mod	1 mem 1001	← Low Offset →
			← High Offset →		
	SUB	A, #byte	1010 1010	← Data →	
		saddr, #byte	0110 1010	← Saddr-offset →	← Data →
		sfr, #byte	0000 0001	0110 1010	← Sfr-offset →
			← Data →		
		r, r1	1000 1010	$R_3R_2R_1R_0$ 0 $R_2R_1R_0$	
		A, saddr	1001 1010	← Saddr-offset →	
		A, sfr	0000 0001	1001 1010	← Sfr-offset →
		saddr, saddr	0111 1010	← Saddr-offset →	← Saddr-offset →
		A, mem	000 mod	0 mem 1010	← Low Offset ←
			← High Offset →		
		mem, A	000 mod	1 mem 1010	← Low Offset →
			← High Offset →		

INSTR. GROUP	MNEMONIC	OPERAND	Instruction Code B1 / B4	B2 / B5	B3
8-bit Arithmetic Instructions	SUBC	A, #byte	1010 1011	← Data →	
		saddr, #byte	0110 1011	← Saddr-offset →	← Data →
		sfr, #byte	0000 0001	0110 1011	← Sfr-offset →
			← Data →		
		r, rl	1000 1011	$R_3R_2R_1R_0$ 0 $R_2R_1R_0$	
		A, saddr	1001 1011	← Saddr-offset →	
		A, sfr	0000 0001	1001 1011	← Sfr-offset →
		saddr, saddr	0111 1011	← Saddr-offset →	← Saddr-offset →
		A, mem	000 mod	0 mem 1011	← Low Offset →
			← High Offset →		
		mem, A	000 mod	1 mem 1011	← Low Offset →
			← High Offset →		
	AND	A, #byte	1010 1100	← Data →	
		saddr, #byte	0110 1100	← Saddr-offset →	← Data →
		sfr, #byte	0000 0001	0110 1100	← Sfr-offset →
			← Data →		
		r, rl	1000 1100	$R_3R_2R_1R_0$ 0 $R_2R_1R_0$	
		A, saddr	1001 1100	← Saddr-offset →	
		A, sfr	0000 0001	1001 1100	← Sfr-offset →
		saddr, saddr	0111 1100	← Saddr-offset →	← Saddr-offset →
		A, mem	000 mod	0 mem 1100	← Low Offset →
			← High Offset →		
		mem, A	000 mod	1 mem 1100	← Low Offset →
			← High Offset →		

INSTR. GROUP	MNEMONIC	OPERAND	Instruction Code		
			B 1	B 2	B 3
			B 4	B 5	
8-bit Arithmetic Instructions	OR	A, #byte	1 0 1 0 1 1 1 0	← Data →	
		saddr, #byte	0 1 1 0 1 1 1 0	← Saddr-offset →	← Data →
		sfr, #byte	0 0 0 0 0 0 0 1	0 1 1 0 1 1 1 0	← Sfr-offset →
			← Data →		
		r, r1	1 0 0 0 1 1 1 0	$R_3R_2R_1R_0$ 0 $R_2R_1R_0$	
		A, saddr	1 0 0 1 1 1 1 0	← Saddr-offset →	
		A, sfr	0 0 0 0 0 0 0 1	1 0 0 1 1 1 1 0	← Sfr-offset →
		saddr, saddr	0 1 1 1 1 1 1 0	← Saddr-offset →	← Saddr-offset →
		A, mem	0 0 0 mod	0 mem 1 1 1 0	← Low Offset →
			← High Offset →		
		mem, A	0 0 0 mod	1 mem 1 1 1 0	← Low Offset →
			← High Offset →		
	XOR	A, #byte	1 0 1 0 1 1 0 1	← Data →	
		saddr, #byte	0 1 1 0 1 1 0 1	← Saddr-offset →	← Data →
		sfr, #byte	0 0 0 0 0 0 0 1	0 1 1 0 1 1 0 1	← Sfr-offset →
			← Data →		
		r, r1	1 0 0 0 1 1 0 1	$R_3R_2R_1R_0$ 0 $R_2R_1R_0$	
		A, saddr	1 0 0 1 1 1 0 1	← Saddr-offset →	
		A, sfr	0 0 0 0 0 0 0 1	1 0 0 1 1 1 0 1	← Sfr-offset →
		saddr, saddr	0 1 1 1 1 1 0 1	← Saddr-offset →	← Saddr-offset →
		A, mem	0 0 0 mod	0 mem 1 1 0 1	← Low Offset →
			← High Offset →		
		mem, A	0 0 0 mod	1 mem 1 1 0 1	← Low Offset →
			← High Offset →		

INSTR. GROUP	MNEMONIC	OPERAND	Instruction Code B1 / B4	Instruction Code B2 / B5	Instruction Code B3
8-bit Arithmetic Instructions	CMP	A, #byte	1010 1111	← Data →	
		saddr, #byte	0110 1111	← Saddr-offset →	← Data →
		sfr, #byte	0000 0001	0110 1111	← Sfr-offset →
			← Data →		
		r, rl	1000 1111	$R_3R_2R_1R_0$ 0 $R_2R_1R_0$	
		A, saddr	1001 1111	← Saddr-offset →	
		A, sfr	0000 0001	1001 1111	← Sfr-offset →
		saddr, saddr	0111 1111	← Saddr-offset →	← Saddr-offset →
		A, mem	000 mod	0 mem 1111	← Low Offset →
			← High Offset →		
		mem, A	000 mod	1 mem 1111	← Low Offset →
			← High Offset →		
16-bit Arithmetic Instructions	ADDW	AX, #word	0010 1101	← Low Byte →	← High Byte →
		saddrp, #word	0000 1101	← Saddr-ofset →	← Low Byte →
			← High Byte →		
		sfrp, #word	0000 0001	0000 1101	← Sfr-offset →
			← Low Byte →	← High Byte →	
		rp, rpl	1000 1000	$P_2P_1P_0$ 0 1 $Q_2Q_1Q_0$	
		AX, saddrp	0001 1101	← Saddr-offset →	
		AX, sfrp	0000 0001	0001 1101	← Sfr-offset →
		saddrp, saddrp	0011 1101	← Saddr-offset →	← Saddr-offset →
	SUBW	AX, #word	0010 1110	← Low Byte →	← High Byte →
		saddrp, #word	0000 1110	← Saddr-offset →	← Low Byte →
			← High Byte →		
		sfrp, #word	0000 0001	0000 1110	← Sfr-offset →
			← Low Byte →	← High Byte →	
		rp, rpl	1000 1010	$P_2P_1P_0$ 0 1 $Q_2Q_1Q_0$	
		AX, saddrp	0001 1110	← Saddr-offset →	
		AX, sfrp	0000 0001	0001 1110	← Sfr-offset →
		saddrp, saddrp	0011 1110	← Saddr-offset →	← Saddr-offset →

INSTR. GROUP	MNEMONIC	OPERAND	Instruction Code		
			B 1	B 2	B 3
			B 4	B 5	
16-bit Arithmetic Instruction	CMPW	AX, #word	0 0 1 0 1 1 1 1	← Low Byte →	← High Byte →
		saddrp, #word	0 0 0 0 1 1 1 1	← Saddr-offset →	← Low Byte →
			← High Byte →		
		sfrp, #word	0 0 0 0 0 0 0 1	0 0 0 0 1 1 1 1	← Sfr-offset →
			← Low Byte →	← High Byte →	
		rp, rp1	1 0 0 0 1 1 1 1	$P_2P_1P_0$ 0 1 $Q_2Q_1Q_0$	
		AX, saddrp	0 0 0 1 1 1 1 1	← Saddr-offset →	
		AX, sfrp	0 0 0 0 0 0 0 1	0 0 0 1 1 1 1 1	← Sfr-offset →
		saddrp, saddrp	0 0 1 1 1 1 1 1	← Saddr-offset →	← Saddr-offset →
Multiply/divide Instructions	MULU	r1	0 0 0 0 0 1 0 1	0 0 0 0 1 $R_2R_1R_0$	
	DIVU	r1		0 0 0 1 1 $R_2R_1R_0$	
	MULUW	rp1		0 0 1 0 1 $Q_2Q_1Q_0$	
	DIVUX	rp1		1 1 1 0 1 $Q_2Q_1Q_0$	
Increment/Decrement Instruction	INC	r1	1 1 0 0 0 $R_2R_1R_0$		
		saddr	0 0 1 0 0 1 1 0	← Saddr-offset →	
	DEC	r1	1 1 0 0 1 $R_2R_1R_0$		
		saddr	0 0 1 0 0 1 1 1	← Saddr-offset →	
	INCW	rp2	0 1 0 0 0 1 S_1S_0		
		saddrp	0 0 0 0 0 1 1 1	1 1 1 0 1 0 0 0	← Saddr-offset →
	DECW	rp2	0 1 0 0 1 1 S_1S_0		
		saddrp	0 0 0 0 0 1 1 1	1 1 1 0 1 0 0 1	← Saddr-offset →
Shift / Rotate Instructions	ROR	r1, n	0 0 1 1 0 0 0 0	0 1 N_2N_1 $N_0R_2R_1R_0$	
	ROL	r1, n	0 0 0 1	0 1 N_2N_1 $N_0R_2R_1R_0$	
	RORC	r1, n	0 0 0 0	0 0 N_2N_1 $N_0R_2R_1R_0$	
	ROLC	r1, n	0 0 0 1	0 0 N_2N_1 $N_0R_2R_1R_0$	
	SHR	r1, n	0 0 0 0	1 0 N_2N_1 $N_0R_2R_1R_0$	
	SHL	r1, n	0 0 0 1	1 0 N_2N_1 $N_0R_2R_1R_0$	
	SHRW	rp, n	0 0 0 0	1 1 N_2N_1 $N_0P_2P_1P_0$	
	SHLW	rp, n	0 0 0 1	1 1 N_2N_1 $N_0P_2P_1P_0$	
	ROR4	[rp1]	0 0 0 0 0 1 0 1	1 0 0 0 1 $Q_2Q_1Q_0$	
	ROL4	[rp1]	0 0 0 0 0 1 0 1	1 0 0 1 1 $Q_2Q_1Q_0$	
BCD Adj.I	ADJ 4		0 0 0 0 0 1 0 0		

INSTR. GROUP	MNEMONIC	OPERAND	Instruction Code B1 / B4	B2 / B5	B3
Bit Operation Instructions	MOV1	CY, saddr. bit	0000 1000	0000 0 $B_2B_1B_0$	← Saddr-offset →
		CY, sfr. bit	1000	1 $B_2B_1B_0$	← Sfr-offset →
		CY, A. bit	0011	1 $B_2B_1B_0$	
		CY, X. bit	0011	0 $B_2B_1B_0$	
		CY, PSWH. bit	0010	1 $B_2B_1B_0$	
		CY, PSWL. bit	0010	0 $B_2B_1B_0$	
		saddr. bit, CY	1000	0001 0 $B_2B_1B_0$	← Saddr-offset →
		sfr. bit, CY	1000	1 $B_2B_1B_0$	← Sfr-offset →
		A. bit, CY	0011	1 $B_2B_1B_0$	
		X. bit, CY	0011	0 $B_2B_1B_0$	
		PSWH. bit, CY	0010	1 $B_2B_1B_0$	
		PSWL. bit, CY	0010	0 $B_2B_1B_0$	
	AND1	CY, saddr. bit	0000 1000	0010 0 $B_2B_1B_0$	← Saddr-offset →
		CY, /saddr. bit		0011 0 $B_2B_1B_0$	← Saddr-offset →
		CY. sfr. bit		0010 1 $B_2B_1B_0$	← Sfr-offset →
		CY, /sfr. bit		0011 1 $B_2B_1B_0$	← Sfr-offset →
		CY, A. bit	0011	0010 1 $B_2B_1B_0$	
		CY, /A. bit		0011 1 $B_2B_1B_0$	
		CY, X. bit		0010 0 $B_2B_1B_0$	
		CY, /X. bit		0011 0 $B_2B_1B_0$	
		CY, PSWH. bit	0010	0010 1 $B_2B_1B_0$	
		CY, /PSWH. bit		0011 1 $B_2B_1B_0$	
		CY, PSWL. bit		0010 0 $B_2B_1B_0$	
		CY, /PSWL. bit		0011 0 $B_2B_1B_0$	

INSTR GROUP	MNEMONIC	OPERAND	Instruction Code B1 (B4)		Instruction Code B2 (B5)		Instruction Code B3
Bit Operation Instructions	OR1	CY, saddr. bit	0000	1000	0100	0 $B_2B_1B_0$	← Saddr-offset →
		CY,/saddr. bit			0101	0 $B_2B_1B_0$	← Saddr-offset →
		CY, sfr. bit			0100	1 $B_2B_1B_0$	← Sfr-offset →
		CY,/sfr. bit			0101	1 $B_2B_1B_0$	← Sfr-offset →
		CY, A.bit		0011	0100	1 $B_2B_1B_0$	
		CY,/A.bit			0101	1 $B_2B_1B_0$	
		CY, X.bit			0100	0 $B_2B_1B_0$	
		CY,/X.bit			0101	0 $B_2B_1B_0$	
		CY, PSWH.bit		0010	0100	1 $B_2B_1B_0$	
		CY,/PSWH.bit			0101	1 $B_2B_1B_0$	
		CY, PSWL.bit			0100	0 $B_2B_1B_0$	
		CY,/PSWL.bit			0101	0 $B_2B_1B_0$	
	XOR1	CY, saddr. bit	0000	1000	0110	0 $B_2B_1B_0$	← Saddr-offset →
		CY, sfr. bit		1000		1 $B_2B_1B_0$	← Sfr-offset →
		CY, A.bit		0011		1 $B_2B_1B_0$	
		CY, X.bit		0011		0 $B_2B_1B_0$	
		CY, PSWH.bit		0010		1 $B_2B_1B_0$	
		CY, PSWL.bit		0010		0 $B_2B_1B_0$	
	SET1	saddr. bit	1011	0 $B_2B_1B_0$	← Saddr-offset →		
		sfr. bit	0000	1000	1000	1 $B_2B_1B_0$	← Sfr-offset →
		A.bit		0011		1 $B_2B_1B_0$	
		X.bit		0011		0 $B_2B_1B_0$	
		PSWH.bit		0010		1 $B_2B_1B_0$	
		PSWL.bit		0010		0 $B_2B_1B_0$	
	CLR1	saddr. bit	1010	0 $B_2B_1B_0$	← Saddr-offset →		
		sfr. bit	0000	1000	1001	1 $B_2B_1B_0$	← Sfr-offset →
		A.bit		0011		1 $B_2B_1B_0$	
		X.bit		0011		0 $B_2B_1B_0$	
		PSWH.bit		0010		1 $B_2B_1B_0$	
		PSWL.bit		0010		0 $B_2B_1B_0$	

INSTR. GROUP	MNEMONIC	OPERAND	Instruction Code B1 / B4	B2 / B5	B3
Bit Operation Instruction	NOT1	saddr. bit	0000 1000	0111 0 $B_2B_1B_0$	← Saddr-offset →
		sfr. bit	↓ 1000	↓ 1 $B_2B_1B_0$	← Sfr-offset →
		A. bit	↓ 0011	↓ 1 $B_2B_1B_0$	
		X. bit	↓ 0011	↓ 0 $B_2B_1B_0$	
		PSWH. bit	↓ 0010	↓ 1 $B_2B_1B_0$	
		PSWL. bit	↓ 0010	↓ 0 $B_2B_1B_0$	
	SET1	CY	0100 0001		
	CLR1	CY	↓ 0000		
	NOT1	CY	↓ 0010		
CALL/RETURN Instruct.	CALL	! addr16	0010 1000	← Low Addr. →	← High Addr. →
	CALLF	! addr11	1001 0 ←	fa →	
	CALLT	[addr5]	111 ← ta →		
	CALL	rp1	0000 0101	0101 1 $Q_2Q_1Q_0$	
		[rp1]	0000 0101	0111 1 $Q_2Q_1Q_0$	
	BRK		0101 1110		
	RET		0101 0110		
	RETI		0101 0111		
Stack Operation Instruction	PUSH	post	0011 0101	← Post Byte →	
		PSW	0100 1001		
	PUSHU	post	0011 0111	← Post Byte →	
	POP	post	0011 0100	← Post Byte ←	
		PSW	0100 1000		
	POPU	post	0011 0110	← Post Byte →	
	MOVW	SP, #word	0000 1011	1111 1100	← Low Byte →
			← High Byte →		
		SP, AX	0001 0011	1111 1100	
		AX, SP	0001 0001	1111 1100	
	INCW	SP	0000 0101	1100 1000	
	DECW	SP	0000 0101	1100 1001	
unconditional branch instr.	BR	! addr16	0010 1100	← Low Addr. →	← High Addr. →
		rp1	0000 0101	0100 1 $Q_2Q_1Q_0$	
		[rp1]	0000 0101	0110 1 $Q_2Q_1Q_0$	
		$ addr16	0001 0100	← jdisp →	

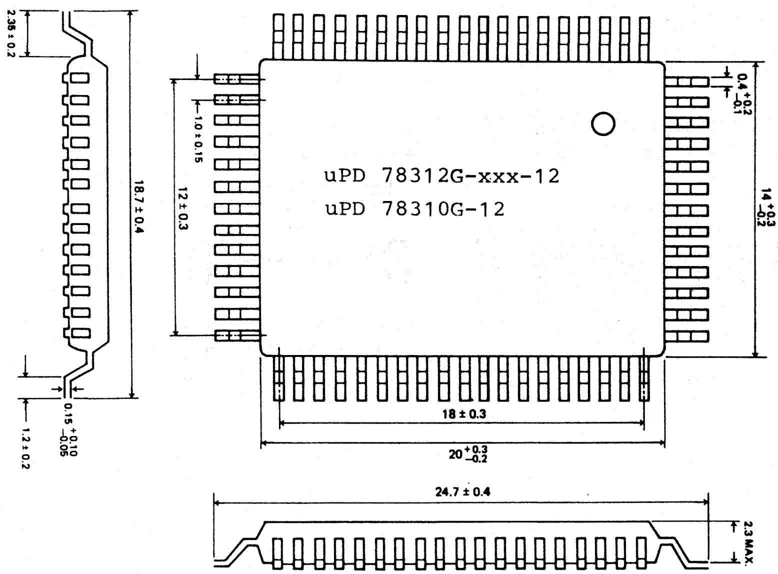
INSTR GROUP	MNEMONIC	OPERAND	Instruction Code B1 / B4	B2 / B5	B3
Conditional Branch Instruction	BC BL	$ addr16	1000 0011	← jdisp →	
	BNC BNL	$ addr16	0010	← jdisp →	
	BZ BE	$ addr16	0001	← jdisp →	
	BNZ BNE	$ addr16	0000	← jdisp →	
	BV BPE	$ addr16	0101	← jdisp →	
	BNV BPO	$ addr16	0100	← jdisp →	
	BN	$ addr16	0111	← jdisp →	
	BP	$ addr16	0110	← jdisp →	
	BGT	$ addr16	0000 0111	1111 1011	← jdisp →
	BGE	$ addr16		1001	← jdisp →
	BLT	$ addr16		1000	← jdisp →
	BLE	$ addr16		1010	← jdisp →
	BH	$ addr16		1101	← jdisp →
	BNH	$ addr16		1100	← jdisp →
	BT	saddr.bit, $ addr16	0111 0 $B_2B_1B_0$	← Saddr-offset →	← jdisp →
		sfr.bit, $ addr16	0000 1000	1011 1 $B_2B_1B_0$	← Sfr-offset →
			← jdisp →		
		A.bit, $ addr16	0000 0011	1011 1 $B_2B_1B_0$	← jdisp →
		X.bit, $ addr16	0011	0 $B_2B_1B_0$	← jdisp →
		PSWH.bit, $ addr16	0010	1 $B_2B_1B_0$	← jdisp →
		PSWL.bit, $ addr16	0010	0 $B_2B_1B_0$	← jdisp →
	BF	saddr.bit, $ addr16	0000 1000	1010 0 $B_2B_1B_0$	← Saddr-offset →
			← jdisp →		
		sfr.bit, $ addr16	0000 1000	1010 1 $B_2B_1B_0$	← Sfr-offset →
			← jdisp →		
		A.bit, $ addr16	0000 0011	1010 1 $B_2B_1B_0$	← jdisp →
		X.bit, $ addr16	0011	0 $B_2B_1B_0$	← jdisp →
		PSWH.bit, $ addr16	0010	1 $B_2B_1B_0$	← jdisp →
		PSWL.bit, $ addr16	0010	0 $B_2B_1B_0$	← jdisp →

I G N R S O T U R. P	MNEMONIC	OPERAND	Instruction Code B1 / B4	B2 / B5	B3
Conditional Branch Instructions	BTCLR	saddr. bit, $ addr16	0000 1000	1101 0 $B_2B_1B_0$	← Saddr-offset →
			← jdisp →		
		sfr. bit, $ addr16	0000 1000	1101 1 $B_2B_1B_0$	← Sfr-offset →
			← jdisp →		
		A.bit, $ addr16	0000 0011	1101 1 $B_2B_1B_0$	← jdisp →
		X.bit, $ addr16	↓ 0011	↓ 0 $B_2B_1B_0$	← jdisp →
		PSWH.bit, $ addr16	↓ 0010	↓ 1 $B_2B_1B_0$	← jdisp →
		PSWL.bit, $ addr16	↓ 0010	↓ 0 $B_2B_1B_0$	← jdisp →
	BFSET	saddr. bit, $ addr16	0000 1000	1100 0 $B_2B_1B_0$	← Saddr-offset →
			← jdisp →		
		sfr.bit, $ addr16	0000 1000	1100 1 $B_2B_1B_0$	← Sfr-offset →
			← jdisp →		
		A. bit, $ addr16	0000 0011	1100 1 $B_2B_1B_0$	← jdisp →
		X. bit, $ addr16	↓ 0011	↓ 0 $B_2B_1B_0$	← jdisp →
		PSWH. bit, $ addr16	↓ 0010	↓ 1 $B_2B_1B_0$	← jdisp →
		PSWL. bit, $ addr16	↓ 0010	↓ 0 $B_2B_1B_0$	← jdisp →
	DBNZ	r2, $ addr16	0011 001C	← jdisp →	
		saddr, $ addr16	0011 1011	← Saddr-offset →	← jdisp →
Context Switch.Ins.	BRKCS	RB n	0000 0101	1101 1 $N_2N_1N_0$	
	RETCS	! addr16	0010 1001	← Low Addr. →	← High Addr. →
String Operation Instructions	MOVM	[DE+], A	0001 0101	0000 0000	
		[DE−], A	↓ ↓	0001 ↓	
	MOVBK	[DE+],[HL+]	↓ ↓	0010 ↓	
		[DE−],[HL−]	↓ ↓	0011 ↓	
	XCHM	[DE+], A	↓ ↓	0000 0001	
		[DE−], A	↓ ↓	0001 ↓	
	XCHBK	[DE+],[HL+]	↓ ↓	0010 ↓	
		[DE−],[HL−]	↓ ↓	0011 ↓	
	CMPME	[DE+], A	↓ ↓	0000 0100	
		[DE−], A	↓ ↓	0001 ↓	
	CMPBKE	[DE+],[HL+]	↓ ↓	0010 ↓	
		[DE−],[HL−]	↓ ↓	0011 ↓	
	CMPMNE	[DE+], A	↓ ↓	0000 0101	
		[DE−], A	↓ ↓	0001 ↓	

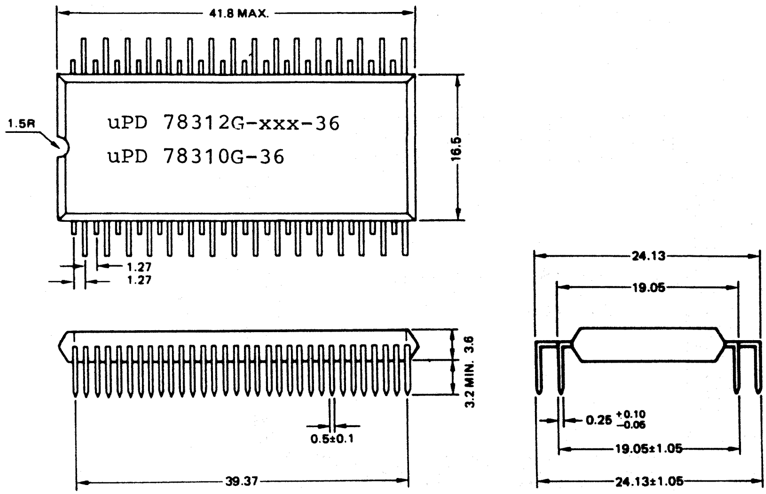
INSTR. GROUP	MNEMONIC	OPERAND	Instruction Code B1 / B4	B2 / B5	B3
String Operation Instr.	CMPBKNE	[DE+],[HL+]	0 0 0 1 0 1 0 1	0 0 1 0 0 1 0 1	
		[DE−],[HL−]		0 0 1 1	
	CMPMC	[DE+], A		0 0 0 0 0 1 1 1	
		[DE−], A		0 0 0 1	
	CMPBKC	[DE+],[HL+]		0 0 1 0	
		[DE−],[HL−]		0 0 1 1	
	CMPMNC	[DE+], A		0 0 0 0 0 1 1 0	
		[DE−], A		0 0 0 1	
	CMPBKNC	[DE+],[HL+]		0 0 1 0	
		[DE−],[HL−]		0 0 1 1	
CPU Control Instruction	MOV	STBC, #byte	0 0 0 0 1 0 0 1	0 1 0 0 0 1 0 0	← $\overline{\text{Data}}$ →
			← Data →		
		WDM, #byte	0 0 0 0 1 0 0 1	0 1 0 0 0 0 1 0	← $\overline{\text{Data}}$ →
			← Data →		
	SWRS		0 1 0 0 0 0 1 1		
	SEL	RBn	0 0 0 0 0 1 0 1	1 0 1 0 1 $N_2N_1N_0$	
		RBn, ALT	0 0 0 0 0 1 0 1	1 0 1 1 1 $N_2N_1N_0$	
	NOP		0 0 0 0 0 0 0 0		
	EI		0 1 0 0 1 0 1 1		
	DI		0 1 0 0 1 0 1 0		

CHAPTER 9 Package Outline

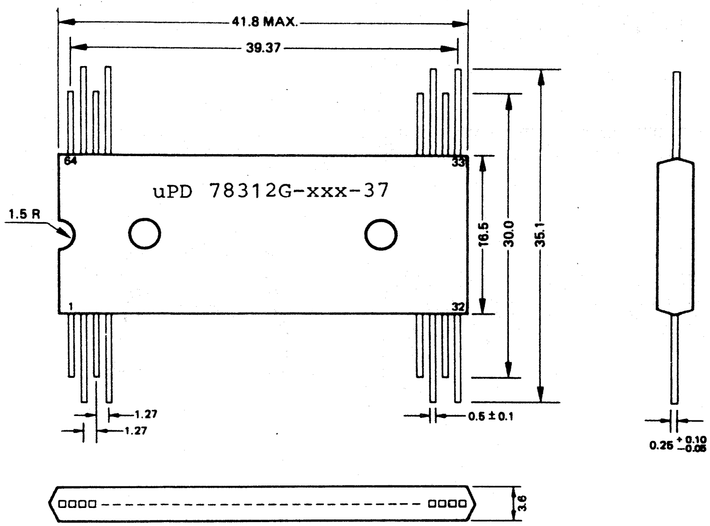
PACKAGE DIMENSIONS ,64-pin FLAT PACK



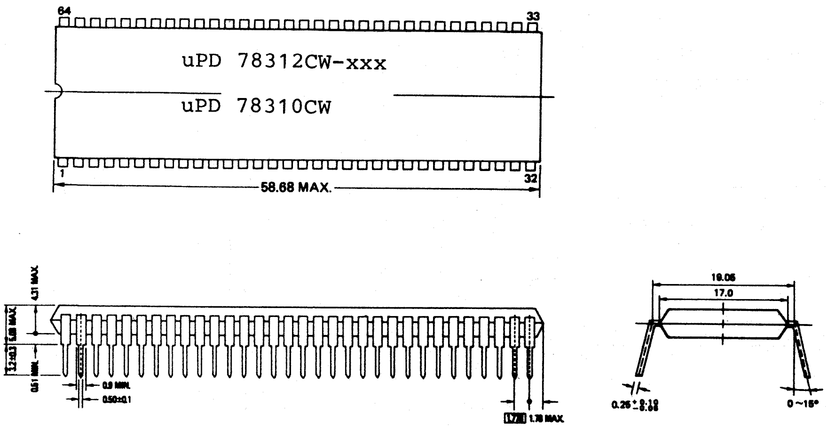
PACKAGE DIMENSIONS ,64-pin QUIL,bent leads



PACKAGE DIMENSIONS ,64-pin QUIL,flat leads



PACKAGE DIMENSIONS ,64-pin SHRINK DIP



NEC Electronics (Europe) GmbH

EUROPEAN DISTRIBUTORS

AUSTRIA
A & D
ABRAHAMCZIK & DEMEL
GES. MBH. & CO KG
EICHENSTRASSE 58-64/1
1120 WIEN
TEL.: (222) 857661
TLX.: 134273

BELGIUM
CN ROOD
DE JAMBLINNE DE MEUXPLEIN 37
1040 BRUSSEL
TEL.: (02) 7352135
TLX.: 22846

MALCHUS ELECTRONICS PVBA
PLANTIN EN MORETUSLEI 172
2000 ANTWERPEN
TEL.: (032) 353256
TLX.: 33637

DENMARK
MER-EL A/S
VED KLAEDEBO 18
2970 HOERSHOLM
TEL.: (2) 571000
TLX.: 37360

FINLAND
OY FERRADO A/B
P.O. BOX 54
VALIMONTIE 1
00380 HELSINKI 38
TEL.: (0) 550002
TLX.: 122214

FRANCE
TEKELEC
RUE CARLE VERNET
CITE DES BRUYERES
92310 SEVRES
TEL.: (1) 45 34 75 35

EALING
BATIMENT AUVIDULIS
AVENUE D'OCEANIE
Z.A. D'ORSAY COURTABŒUF
BP 90
91943 LES ULIS CEDEX
TEL.: (1) 69 28 01 31

CELTI
Z.I. DE COURTABŒUF
9, AVENUE DU QUEBEC
91940 LES ULIS
TEL.: (1) 64 46 09 09

ASAP
RUE DE TROIS PEUPLES
78190 MONTIGNY LE BRETONNEUX
TEL.: (1) 30 43 82 33
TLX.: 698887

ASAP
MONSIEUR LEGRIS
42, RUE HENRI MATISSE
59930 LA CHAPELLE D'ARMENTIERES
TEL.: 20 35 11 10

CCI
5, RUE MARCELIN BERTHELOT
BP 92
92164 ANTONY
TEL.: (1) 46 66 21 82
TLX.: 203881

CCI
5, RUE BATAILLE
69008 LYON
TEL.: 78 74 44 56

DIM INTER
65 – 67, RUE DES CITES
93300 AUBERVILLIERS
TEL.: (1) 48 34 93 70
TLX.: 230524

DIM INTER (VILLEURBANNE)
101, RUE DEDIEU
69100 VILLEURBANNE
TEL.: 78 68 32 29

DIM INTER (COLMAR)
27, RUE KLEBER
68000 COLMAR
TEL.: 89 41 15 43

GEDIS
352, AVENUE G. CLEMENCEAU
92000 NANTERRE
TEL.: (1) 42 04 04 04

GEDIS (ALPES)
21, RUE DES GLAISONS
38400 ST. MARTIN D'HERES
TEL.: 75 51 23 32

GEDIS (AIX)
MERCURE C
Z.I. D'AIX EN PROVENCE
13763 LES MILLES CEDEX
TEL.: 42 60 01 77

CEDIS (TOURS)
1, RUE DU DANEMARK
37100 TOURS
TEL.: 47 41 76 46

SERTRONIQUE (MANS)
60, RUE SAGEBIEN
CEDEX 43
72040 LE MANS
TEL.: 43 84 24 60
TLX.: 720019

SERTRONIQUE (LILLE)
20, RUE CABANIS
BP 35
59007 LILLE CEDEX
TEL.: 20 47 70 70

GERMANY
GLYN GMBH
SCHÖNE AUSSICHT 30
6272 NIEDERNHAUSEN
TEL.: (0 61 27) 80 77
TLX.: 4 186 911

MICROSCAN GMBH
ÜBERSEERING 31
2000 HAMBURG 60
TEL.: (0 40) 6 32 00 30
TLX.: 2 13 288

REIN ELEKTRONIK GMBH
LÖTSCHERWEG 66
4054 NETTETAL 1
TEL.: (0 21 53) 73 31 11
TLX.: 8 54 251

ULTRATRONIK GMBH
MÜNCHENER STRASSE 6
8031 SEEFELD
TEL.: (0 81 52) 70 90
TLX.: 5 26 459

H3W ELEKTRONIK VERTRIEB GMBH
NEUMARKTER STRASSE 75
8000 MÜNCHEN 80
TEL.: (0 89) 4 31 32 60
TLX.: 5 214 514

NIPTRON GMBH
LUITPOLTSTRASSE 52
8300 LANDSHUT
TEL.: (08 71) 6 90 48
TLX.: 5 8 443

SYSTEM ELEKTRONIK VERTRIEB GMBH
HEESFELD 4
3300 BRAUNSCHWEIG
TEL.: (05 31) 31 40 95
TLX.: 9 52 351

GLEICHMANN + CO ELECTRONICS GMBH
INDUSTRIESTRASSE 16
7513 STUTENSEE 3
TEL.: (0 72 49) 70 01
TLX.: 7 825 602

ITALY
MELCHIONI S.P.A.
VIA COLETTA, 37
20135 MILANO
TEL.: (02) 57941

CLAITRON S.P.A.
VIA GALLARATE, 211
20151 MILANO
TEL.: (02) 3010091

ADELSY S.R.L.
VIA DEL FONDITORE, 5
LOCALITA ROVERI
40127 BOLOGNA
TEL.: (051) 532119

PANTRONIC S.R.L.
VIA MATTIA BATTISTINI, 212/a
00167 ROMA
TEL.: (06) 6273909

NETHERLANDS
INNOCIRCUIT
MALCHUS ELECTRONICA
ADVIESGROEP
MALCHUS B.V.
FOKKERSTRAAT 511-513
3125 BD SCHIEDAM
TEL.: (010) 373777
TLX.: 21598

CN ROOD
CORT V.D. LINDENSTRAAT 11-13
2288 EV RIJSWIJK
TEL.: (070) 996360
TLX.: 31238

NORWAY
JAKOB HATTELAND ELECTRONIC A/S
P.B. 25
5578 NEDRE VATS
TEL.: (47) 63 111
TLX.: 428 50

PORTUGAL
AMPEREL S.A.
AV. FONTES PEREIRA DE MELO 47, 4D
1000 LISBOA
TEL.: (1) 53 26 98
TLX.: 18588

SPAIN
LOBER S.A.
MONTE ESQUINZA 28
MADRID 4
TEL.: (1) 442 11 00
TLX.: 49533

COMELTA S.A.
EMILIO MUNOZ 41, NAVE 1-1-2
MADRID 17
TEL.: (1) 754 30 01
TLX.: 42007

AMITRON S.A.
AVENIDA DE VALLADOLID 47 A
28008 MADRID
TEL.: (1) 247 93 13
TLX.: 45550

SWEDEN
TH'S ELEKTRONIK
BOX 3027
16303 SPAANGA
TEL.: (0) 8362970
TLX.: 11145

NORDQVIST & BERG
BOX 9145
AARSTAAENGS VAEGEN 19
10272 STOCKHOLM
TEL.: (0) 8690400
TLX.: 10407

SWITZERLAND
MEMOTEC AG
GASWERKSTRASSE 32
4901 LANGENTHAL
TEL.: (63) 28 11 22
TLX.: 9 82 550

TURKEY
BURC ELEKTRONIK
VE MAKINA
SANAYI VE TICARET A.S.
BANKATCHI-SOKAK 15/2
KÜCÜKESAT
ANKARA
TEL.: (0090) 41250300
TLX.: 43430

UNITED KINGDOM
ANZAC COMPONENTS LTD
BURNHAM LANE
SLOUGH SL1 6LN
ENGLAND
TEL.: (06286) 4701

FARNEHL ELECTRONIC
COMPONENTS LTD
CANAL ROAD
LEEDS LS12 2TU
ENGLAND
TEL.: (0532) 636311

STC MULTI COMPONENTS
EDINBURGH WAY
HARLOW
CM20 2DF
ENGLAND
TEL.: (0279) 442971

DIALOGUE DISTRIBUTION LTD
WATCHMOOR ROAD
CAMBERLER
SURREY GU15 3AQ
ENGLAND
TEL.: (0276) 688001

IMPULSE ELECTRONICS LTD
HAMMOND HOUSE
CATERHAM
SURREY CR3 6XG
TEL.: (0883) 46433

VSI ELECTRONICS LTD
ROYDOMBURY INDUSTRIAL PARK
HORSECROFT ROAD 9
HARLOW, 5
ESSEX CM19 5BYQM
TEL.: (0279) 29666

NEC OFFICES

NEC Electronics (Europe) GmbH, Oberrather Str. 4, 4000 Düsseldorf 30, W. Germany,
Tel. (0211) 650301, Telex 858996-0

NEC Electronics (Germany) GmbH, Oberrather Str. 4, 4000 Düsseldorf 30,
Tel. (0211) 650302, Telex 858996-0
- Hindenburgstr. 28/29, 3000 Hannover 1, Tel. (0511) 881013–16, Telex 9230109
- Arabellastr. 17, 8000 München 2, Tel. (089) 4160020, Telex 522971
- Heilbronner Str. 314, 7000 Stuttgart 30, Tel. (0711) 890910, Telex 7252220

NEC Electronics (BNL) – Boschdijk 187a, NL-5612 HB Eindhoven, Tel. (040) 445845,
Telex 51923

NEC Electronics (Scandinavia) – Box 4039, S-18304 Täby, Tel. (08) 7328200,
Telex 13839

NEC Electronics (France) S.A., 9, rue Paul Dautier, B. P. 187,
F-78142 Velizy Villacoublay Cedex, Tél. (1) 39469617, Télex 699499

NEC Electronics (France) S.A., Representacion en Espana, Edificio «La Caixa»,
Paseo de la Castellana 51, E-28046 Madrid, Tél. (1) 4194150, Télex 41316

NEC Electronics Italiana S.R.L., Via Fabio Filzi, 25A, I-20124 Milano, Tel. (02) 6709108,
Telex 315355

NEC Electronics (UK) Ltd., Block 3 Carfin Industrial Estate, Motherwell M L1 4UL,
Scotland, Tel. (0698) 732221, Telex 777565
- Birmingham Office, 9th Floor, Swan Office Centre, 1508 Coventry Road, Yardley, Birmingham B 25 8 VL, Tel. (021) 7081500, Telex 333014
- Reading Office, Reading Central Building, 30 Garrad Street, Reading Berks, RG1 1NR, Tel. (0734) 596551, Telex 847998
- Dublin Office, 34/35 South William Street, Dublin 2, Ireland, Tel. (0001) 710200