HD648180W
MCU (Micro Controller Unit)

■ DESCRIPTION

The HD648180W is an 8-bit CMOS microcontroller in the HD64180 family. Its instruction set is upward-compatible with the HD64180Z, hence with the Z-80. Twelve instructions (seven types) have been added, bringing the total number of instructions to 165.

Compared with the HD64180Z, the HD648180W also has more peripheral functions. In addition to a refresh controller and wait-state controller for memory access support, and a memory management unit (MMU) and DMA controller (DMAC) for processing large quantities of data, the HD64180W has on-chip RAM, EEPROM, and an A/D converter. Timer and I/O-port functions have also been enhanced.

Figure 1 shows the HD64180 family. Table 1 lists the features of the HD64180W.

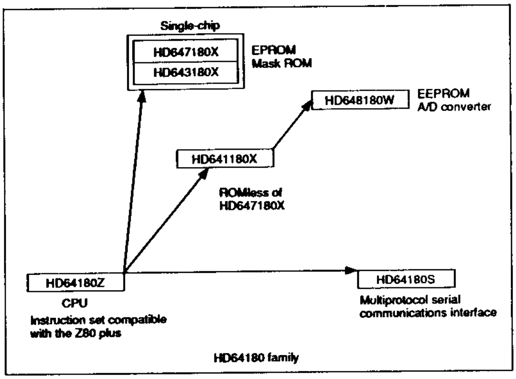
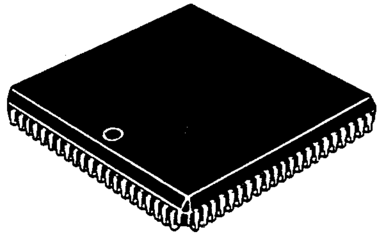
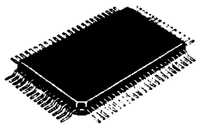


Figure 1 HD64180 Family



(CP-84)



(FP-80A)

■ ORDERING INFORMATION

Type No.	Speed and Package
HD648180W0CP4	4 MHz 84-pin PLCC
HD648180W0CP6	6 MHz 84-pin PLCC
HD648180W0F4	4 MHz 80-pin QFP
HD648180W0F6	6 MHz 80-pin QFP

Table 1 HD648180W Features

Item	Description
CPU	• Instruction set upward-compatible with HD64180Z • Maximum operating speed: 6.144 MHz
MMU	• 1-Mbyte physical address space
Memory	• 256-byte EEPROM on-chip —Byte or page write • 1-kbyte static RAM on-chip —Provides work area, stack area for interrupts, etc.
Timers (4 channels)	• 16-bit free-running timer (1 channel) on-chip —Input capture function for measuring input pulse width —Output compare function for generating waveforms • 16-bit reload timers (3 channels) on-chip —Toggle function for square-wave output with 50% duty cycle
Serial communication interface (2 channels)	• Asynchronous or clocked synchronous mode • Simultaneous transmit and receive (full duplex communication) • On-chip baud rate generator
DMA controller (2 channels)	• Directly addresses 1-Mbyte address space (without using MMU) • Directly addresses 64-kbyte I/O address space • Can transfer 64-kbyte data continuously • Data transfer speed (memory-to-memory): six system clocks/byte
A/D converter (8-bit resolution)	• Four analog input channels • Absolute precision: 2.0 LSB • Conversion time: 17 µs (at 6 MHz)
I/O ports	• 30 input/output lines (including 10 with direct LED driving capability: I_{OL} = 10 mA) • 4 input lines • 1 output line
Wait-state controller	• Wait states can be inserted by external signal input or by software
Refresh controller	• Outputs 8-bit refresh addresses • Programmable refresh interval

Table 1 HD648180W Features (cont)

Item	Description
Interrupts	• Four external interrupt lines: $\overline{NMI}$, $\overline{INT_0}$, $\overline{INT_1}$, $\overline{INT_2}$ • Ten on-chip interrupt sources
Special CPU modes	• Low-power modes (standby modes) • Sleep mode • Halt mode • Bus-release mode
Other features	• On-chip clock oscillator • Interfaces directly with Z-80 peripheral chips

■ Block Diagram

Figure 2 is a block diagram of the HD648180W.

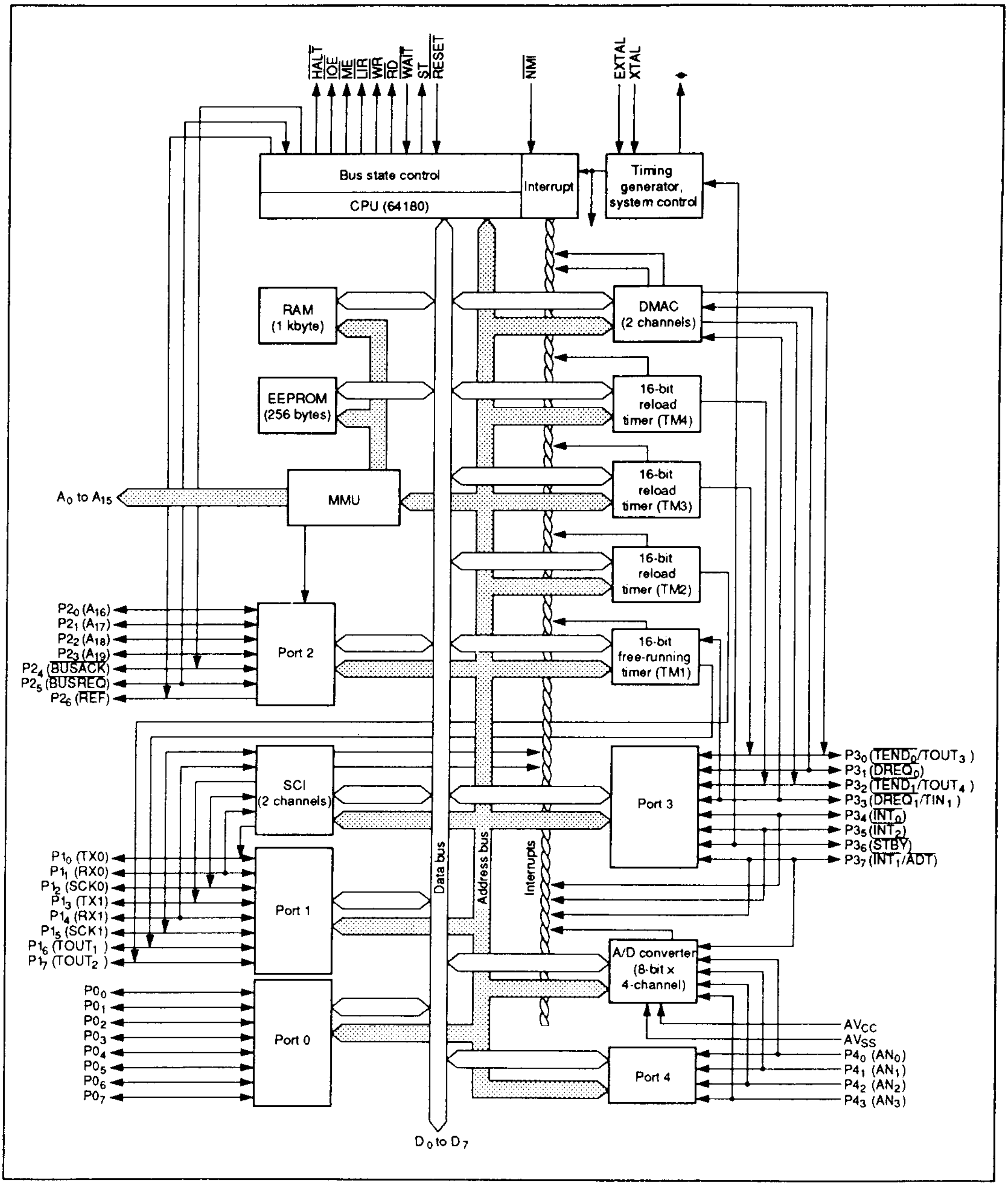


Figure 2 HD648180W Block Diagram

■ Pin Descriptions

- **Pin Arrangement**

Figure 3 shows the pin arrangement of the HD648180W in the FP-80A package. Figure 4 shows the pin arrangement in the CP-84 package.

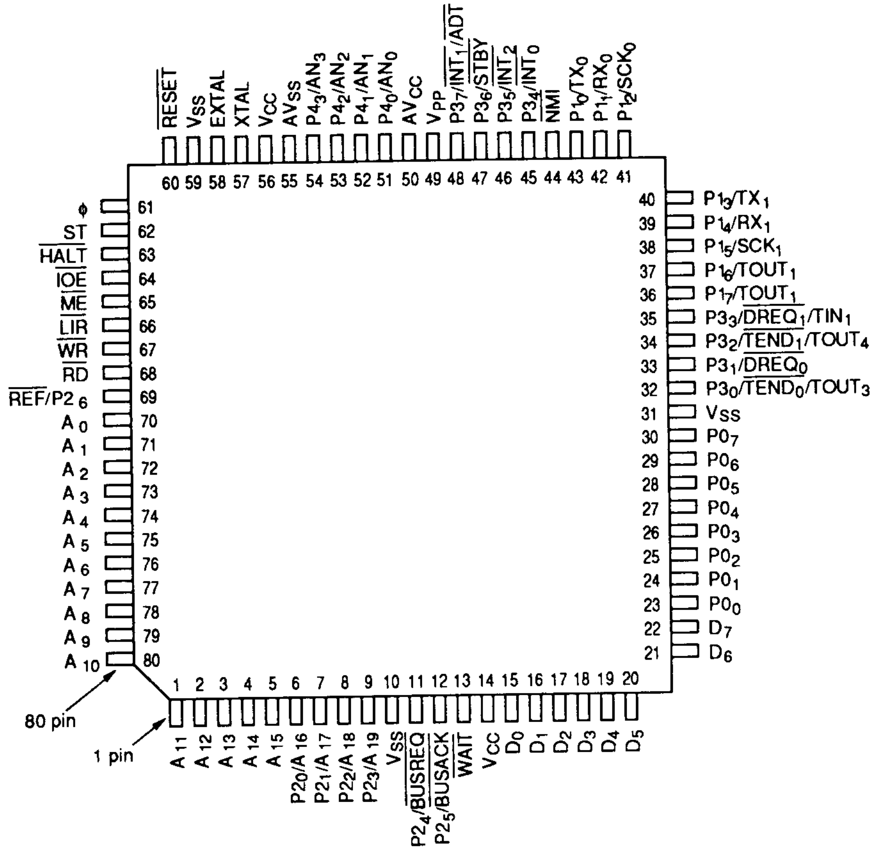
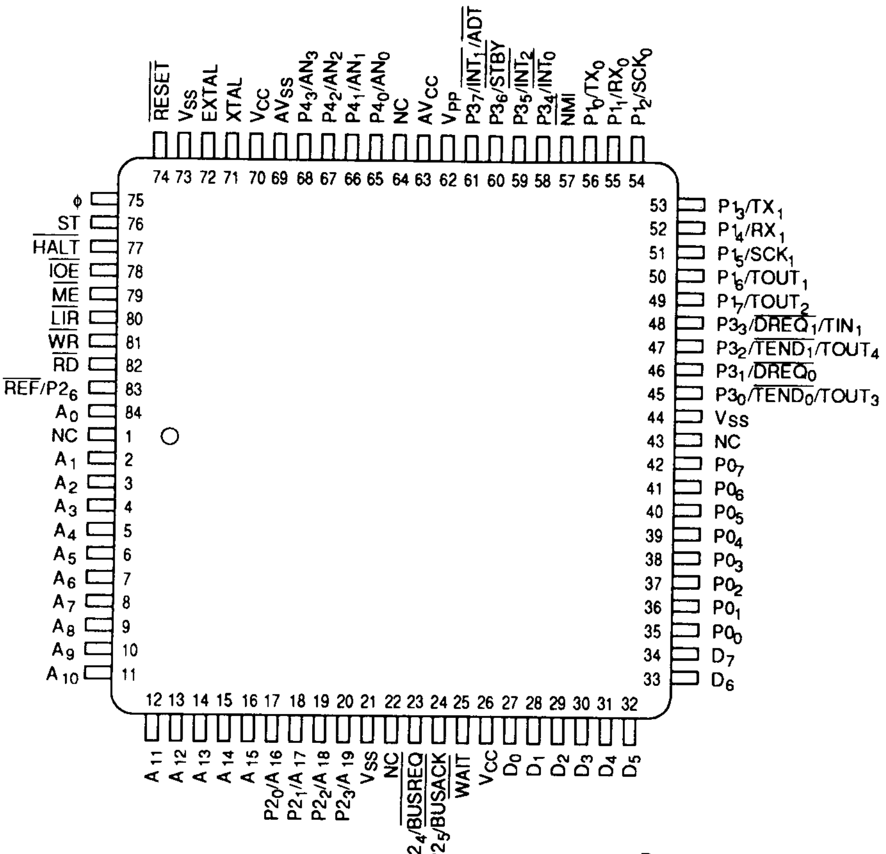


Figure 3 Pin Arrangement (FP-80A)



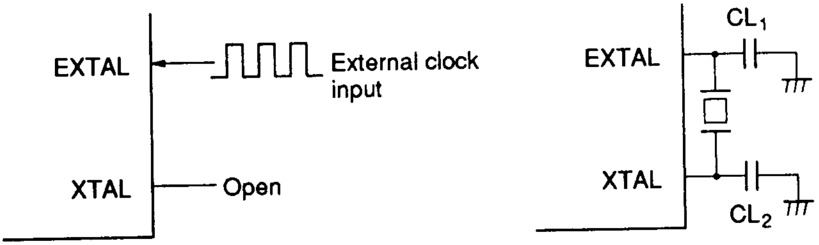
Do not connect anything to NC.

Figure 4 Pin Arrangement (CP-84)

- **Pin Functions**

Table 2 Pin Functions

		Pin Number			
Type	Signal	FP-80A	PC-84	I/O	Name and Pin Function
Power supply GND	V_{CC}	14, 56	26, 70	Input	Power Supply: Connect all V_{CC} pins to system power supply (+5 V).
	V_{SS}	10, 31, 59	21, 44 73	Input	Ground: Connect all V_{SS} pins to system power supply (0 V).
XTAL clock	XTAL	57	71	Input	Connect to crystal oscillator with twice system clock (ϕ) frequency. When inputting external clock at EXTAL, leave XTAL open.
	EXTAL	58	72	Input	Connect to crystal oscillator or external clock. External clock frequency should be 2 times system clock.
	ϕ	61	75	Output	System Clock: Supplies system clock to peripheral devices.
Reset	$\overline{\text{RESET}}$	60	74	Input	Reset: Chip is reset when this pin is dropped low.
Address bus	A_0 to A_{19}	1 to 9, 70 to 80	1 to 20, 84	Output (Tri-state)	Address Bus: For memory access. These lines go to high impedance only in the following cases. a. During reset b. When bus control is granted to another device (when $\overline{\text{BUSACK}}$ drops low because of low input at $\overline{\text{BUSREQ}}$).



Oscillator circuit examples

Table 2 Pin Functions (cont)

		Pin Number			
Type	**Signal**	**FP-80A**	**PC-84**	**I/O**	**Name and Pin Function**
Data bus	D_0 to D_7	15 to 22	27 to 34	Input/ Output	Data Bus: 8-bit, bidirectional data bus
Memory I/O interface signal	$\overline{\text{RD}}$	68	82	Output (Tri-state)	Read: Indicates chip is in read cycle. At this time, data bus is in input mode.
	$\overline{\text{WR}}$	67	81	Output (Tri-state)	Write: Indicates chip is in write cycle. At this time, data bus is in output mode.
	$\overline{\text{ME}}$	65	79	Output (Tri-state)	Memory Enable: Indicates memory read/write is being performed. Goes low in the following cases: a. Instruction fetch and operand read b. Data read/write by memory reference instruction c. Memory access during DMA cycle d. Refresh cycle
	$\overline{\text{IOE}}$	64	78	Output (Tri-state)	I/O Enable: Indicates I/O read/write is being performed. Goes low in the following cases: a. Data read/write by I/O instruction execution b. I/O access during DMA cycle c. $\overline{\text{INT}_0}$ acknowledge cycle
	$\overline{\text{WAIT}}$	13	25	Input	Wait: Used to extend memory and I/O read/write cycles. If low at the falling clock edge in a T_2 or T_W state, a T_W state is inserted next.
System control signals	$\overline{\text{BUSREQ}}$	11	23	Input	Bus Request: Used by another device to issue a bus request. When this pin goes low, CPU terminates instruction execution and sets address bus, data bus, and some memory interface signals ($\overline{\text{RD}}$, $\overline{\text{WR}}$, $\overline{\text{ME}}$, $\overline{\text{IOE}}$) to high impedance.

Table 2 Pin Functions (cont)

Type	Signal	Pin Number FP-80A	 PC-84	I/O	Name and Pin Function
System control signals	$\overline{\text{BUSACK}}$	12	24	Output	Bus Acknowledge: Indicates receipt of $\overline{\text{BUSREQ}}$ signal by CPU and release of bus. Notifies device that output $\overline{\text{BUSREQ}}$ signal that bus is under its control.
	$\overline{\text{HALT}}$	63	77	Output	HALT: Goes low when CPU executes HALT or SLP instruction to indicate that CPU is in halt or sleep mode. Used together with ST and LIR signals (described below) to indicate operational status of internal DMA and CPU mode.
	$\overline{\text{LIR}}$	66	80	Output	Load Instruction Register: Indicates ongoing cycle is op code fetch cycle.
	ST	62	76	Output	Status: Indicates operational status. Do not connect pull-down resistance to this pin.

ST	$\overline{\text{HALT}}$	$\overline{\text{LIR}}$	Operation Mode
0	1	0	CPU operation (1st op-code fetch cycle)
1	1	0	CPU operation (2nd, 3rd op-code fetch cycle)
1	1	1	CPU operation (machine cycles other than op-code fetch cycles)
0	Undefined	1	DMA
0	0	0	Halt mode
1	0	1	Sleep mode

Table 2 Pin Functions (cont)

Type	Signal	Pin Number FP-80A	Pin Number PC-84	I/O	Name and Pin Function
System control signals	$\overline{REF}$	69	83	Output	Refresh: Low level indicates CPU is in DRAM refresh cycle. Refresh address is output on lower 8 bits of address bus (A_0 through A_7). Refresh cycle interval can be programmed to 10, 20, 40, or 80 states.
Interrupt signals	$\overline{NMI}$	44	57	Input	Non-Maskable Interrupt: Non-maskable interrupt request pin.
	$\overline{INT_0}$	45	58	Input	Interrupt 0: Maskable interrupt level 0 request pin. Level 0 has 3 modes. **Mode — Meaning** 0 — Execute instruction on data bus 1 — Execute instructions starting from address 0038H 2 — Vectored
	$\overline{INT_1}$	48	61	Input	Interrupt 1, 2: Maskable interrupt Level 1 and 2 request pins. Vectored.
	$\overline{INT_2}$	46	59	Input	
DMA signals	$\overline{DREQ_0}$	33	46	Input	DMA Request for Channel 0: Asks internal DMAC to perform transfer on channel 0. Enables internal DMAC to synchronize with external I/O device. Internal DMAC channel 0 supports the following transfers. a. Memory ↔ Memory b. Memory ↔ I/O c. Memory ↔ Memory-mapped I/O

Table 2 Pin Functions (cont)

		Pin Number			
Type	Signal	FP-80A	PC-84	I/O	Name and Pin Function
DMA signals	$\overline{TEND_0}$	32	45	Output	Transfer End for Channel 0: Internal DMAC channel 0 transfer end signal. This signal goes low at write cycle of final data transfer.
	$\overline{DREQ_1}$	35	48	Input	DMA Request for Channel 1: Asks internal DMAC to perform transfer on channel 1. Channel 1 supports Memory ↔ I/O transfers only.
	$\overline{TEND_1}$	34	47	Output	Transfer End for Channel 1: Internal DMAC channel 1 transfer end signal. This signal goes low at write cycle of final data transfer.
Serial communications	TX_0	43	56	Output	Transmit Data for SCI Channel 0: SCI channel 0 transmit data pin.
	RX_0	42	55	Input	Receive Data for SCI Channel 0: SCI channel 0 receive data pin.
	SCK_0	41	54	Input/ Output	Serial Clock for SCI Channel 0: SCI channel 0 clock input/output pin.
	TX_1	40	53	Output	Transmit Data for SCI Channel 1: SCI channel 1 transmit data pin.
	RX_1	39	52	Input	Receive Data for SCI Channel 1: SCI channel 1 receive data pin.
	SCK_1	38	51	Input/ Output	Serial Clock for SCI Channel 1: SCI channel 1 clock input/output pin.

Table 2 Pin Functions (cont)

		Pin Number			
Type	**Signal**	**FP-80A**	**PC-84**	**I/O**	**Name and Pin Function**
Timers	TIN_1	35	48	Input	Timer Input for Channel 1: Input signal pin for timer 1 input capture. Signal transitions at this pin transfer free-running counter value to input capture register.
	$TOUT_1$	37	50	Output	Timer Output for Channel 1: Timer 1 waveform output pin. When output compare register and free-running counter match, value of OLVL bit of timer control/status register 1 is output from this pin.
	$TOUT_2$	36	49	Output	Timer Output for Channel 2: Timer 2 waveform output pin. When timer 2 time constant register and timer 2 up-counter match, value selected by bit 2 and bit 3 of timer control/status register 2 is output from this pin.
	$TOUT_3$	32	45	Output	Timer Output for Channel 3: Timer 3 waveform output pin. When timer 3 time constant register and timer 3 up-counter match, value selected by bit 2 and bit 3 of timer control/status register 2 is output from this pin.
	$TOUT_4$	34	47	Output	Timer Output for Channel 4: Timer 4 waveform output pin. When timer 4 time constant register and timer 4 up-counter match, value selected by bit 2 and bit 3 of timer control/status register 4 is output from this pin.

Table 2 Pin Functions (cont)

Type	Signal	Pin Number FP-80A	Pin Number PC-84	I/O	Name and Pin Function
A/D converter	AV_{SS}	55	69	Input	Analog Ground: A/D converter power supply (ground)
	AV_{CC}	50	63	Input	Analog Power Supply: A/D converter power supply (+5 V). Connect to system power supply (V_{CC}).
	AN_0 to AN_3	51 to 54	65 to 68	Input	Analog Input: A/D converter input signal pins
	$\overline{ADT}$	48	61	Input	A/D Trigger: External trigger input pin to start A/D converter
EEPROM	V_{PP}	49	62	Output	Test pin. Do not connect.
Parallel I/O	$P0_0$ to $P0_7$	23 to 30	35 to 42	Input/ Output	Port 0: 8-bit parallel I/O port. Switched between input and output by data direction register 0 (DDR0).
	$P1_0$ to $P1_7$	36 to 43	49 to 56	Input/ Output	Port 1: 8-bit parallel I/O port. Switched between input and output by data direction register 1 (DDR1).
	$P2_0$ to $P2_6$	6 to 9 11, 12, 69	17 to 20 23, 24, 83	Input/ Output	Port 2: 7-bit parallel I/O port. Switched between input and output by data direction register 2 (DDR2). P26 is output only.
	$P3_0$ to $P3_7$	32 to 35 45 to 48	45 to 48 58 to 61	Input/ Output	Port 3: 8-bit parallel I/O port. Switched between input and output by data direction register 3 (DDR3).
	$P4_0$ to $P4_3$	51 to 54	65 to 68	Input	Port 4: 4-bit input-only port.

Basic CPU Architecture

The HD648180W has an advanced, high-speed, eight-bit CPU. Its instruction set is upward-compatible with the HD64180Z.

• Features

The HD648180W CPU has the following features.

- CPU architecture based on the Z-80
- Expanded instruction set with instructions for:
  - Input from on-chip I/O
  - Output to on-chip I/O
  - Block transfer between memory and on-chip I/O
  - 8-bit × 8-bit multiplication
  - AND operations on on-chip I/O
  - AND operations on accumulator A
  - Transition to sleep mode
- Eight addressing modes
  - Implied
  - Register direct
  - Register indirect
  - Indexed
  - Extended
  - Immediate
  - Relative
  - I/O
- Maximum operating speed: 6.144 MHz
- Special CPU operating modes
  - Low-power modes (standby modes)
    Software standby mode
    Hardware standby mode
  - Sleep mode
  - Halt mode
  - Bus-release mode

• Address Space

The HD648180W CPU has a 64-kbyte logical address space and 64-kbyte I/O address space.

The 64-kbyte logical address space is mapped into a 1-Mbyte physical address space by the memory management unit (MMU). For details, see Memory Management Unit (MMU).

The 64-kbyte I/O address space is assigned to on-chip and off-chip I/O. For details, see On-Chip I/O Address Space Map.

• Register Configuration

The CPU includes two general register sets (GR and GR'), and a single special-purpose register set. Register sets GR and GR' each include an accumulator, a flag register, and six general-purpose registers. The special-purpose register set consists of an interrupt vector register, an R counter, two 16-bit index registers, a stack pointer, and a program counter.

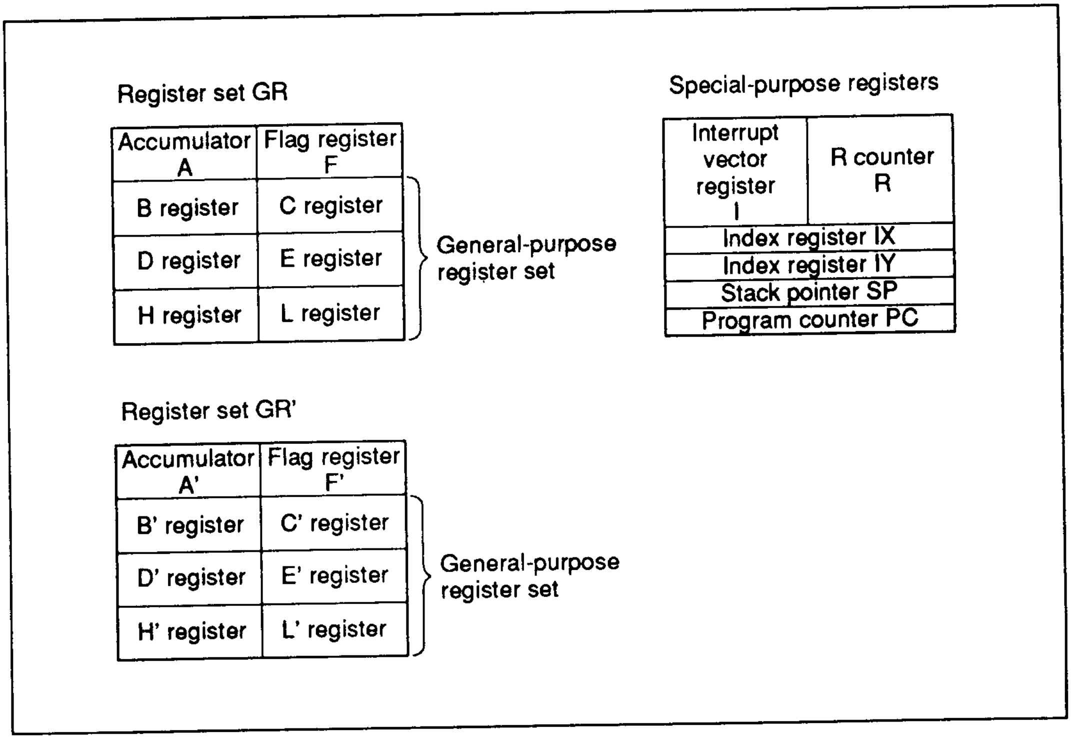


Figure 5 CPU Register Configuration

• Register Descriptions

The following table describes the function of each register. Unless otherwise specified, register content is undefined following a reset.

Register Name	Symbol	Function
Accumulators	A, A'	Accumulators are registers in which 8-bit arithmetic operations, logical operations, and shifts are performed. Accumulator A' is used in place of accumulator A following execution of the instruction EX AF, AF'.
Flag registers	F, F'	Flag registers store the status of operations. Flag register F' is used in place of flag register F following execution of the instruction EX AF, AF'.
General-purpose registers	B, C D, E H, L	The registers in register set GR can be used as 8-bit registers (B, C, D, E, H, L) or 16-bit registers (BC, DE, HL). They can be used to execute operations or store addresses.
	B', C' D', E' H', L'	The registers in register set GR' supplement the GR registers. The two register sets can be swapped using the EXX instruction.
Interrupt vector register	I	This 8-bit register stores the upper 8 bits of the 16-bit vector produced by an interrupt. Vectors are used for $\overline{INT_0}$ mode 2, $\overline{INT_1}$, $\overline{INT_2}$, and internal interrupts (7 sources). A reset initializes this register to 00H.
R counter	R	This 8-bit register counts the number of op-code fetch cycles performed. The content of this counter is unrelated to the refresh address, which is generated by another on-chip counter that cannot be accessed by the user. A reset initializes this register to 00H.
Index registers	IX, IY	These 16-bit registers are used for indexed addressing and 16-bit operations. In indexed addressing, the base address is stored in the index registers. A signed 8-bit displacement is added to the base address to generate the operand's effective address. In 16-bit operations that involve the index registers, a general-purpose register (except HL), the index registers, or the stack pointer can be used for the other operand.
Stack pointer	SP	This 16-bit register stores the top address of the stack area. A reset initializes this register to 00H.
Program counter	PC	This 16-bit register stores the logical address of the next instruction to be executed. This register is normally incremented by 1 each time a 1-byte instruction code is accessed, but execution of a jump instruction causes the address of the jump destination to replace the current content of the program counter. A reset initializes this register to 0000H.

- Flag Register

The flag register (F) contains individual flags that are set and reset to represent the status of the result of an 8-bit or 16-bit operation. This register is referenced by extended arithmetic instructions and conditional jump instructions.

	7	6	5	4	3	2	1	0
Flag register F	S	Z	—	H	—	P/V	N	C

Flag Name	Symbol	Function
Sign	S	Bit 7 is set when an operation produces a negative result (MSB = 1) and reset by a positive result (MSB = 0).
Zero	Z	Bit 6 is set when an instruction execution produces a zero result, and reset by any other result.
Half-carry	H	Bit 5 is set when an operation results in a carry or borrow from the 4th bit. The half-carry flag is reset when neither a carry nor borrow is generated. This flag is referenced for adjustment of decimal operations (DAA instruction).
Parity/overflow	P/V	Bit 2 can be used as either a parity or an overflow flag. The parity flag indicates the parity of the data stored in the accumulator following execution of a logical operation. An even number of 1s in the accumulator value sets the parity flag, while an odd number of 1s resets it. The overflow flag is set when the result of a signed arithmetic operation exceeds the range of +127 through –128 for an 8-bit operation, or the range of +32767 through –32768 for a 16-bit operation. Results within these ranges reset the overflow flag.
Negate	N	Bit 1 is set by execution of a subtraction instruction (examples: SUB, DEC, CP) and reset by an addition instruction (examples: ADD, INC).
Carry	C	Bit 0 is set when an operation results in a carry or borrow from the most significant bit. Any other result resets this flag. The following lists the various types of carries and borrows. • Carry produced by an addition result • Borrow produced by a subtraction result • Carry produced by a shift or rotate

Addressing Modes

The CPU instruction set has eight addressing modes.

1. Implied Addressing (IMP)

   This addressing mode is based on data included within op codes. It is used by instructions that manipulate bit positions specified by the accumulator (A), index registers (IX, IY), stack pointer (SP), HL general-purpose register, or op codes.

2. Register Direct Addressing (REG)

   In this addressing mode, 8-bit and 16-bit registers are specified by op code fields g, g', ww, xx, yy, and zz. The following tables show the relationships between field codes and registers.

8-Bit Register Specification

g or g' Field	Specified Register
000	B
001	C
010	D
011	E
100	H
101	L
110	—
111	A

16-bit Register Specification

ww Field	Specified Register
00	BC
01	DE
10	HL
11	SP

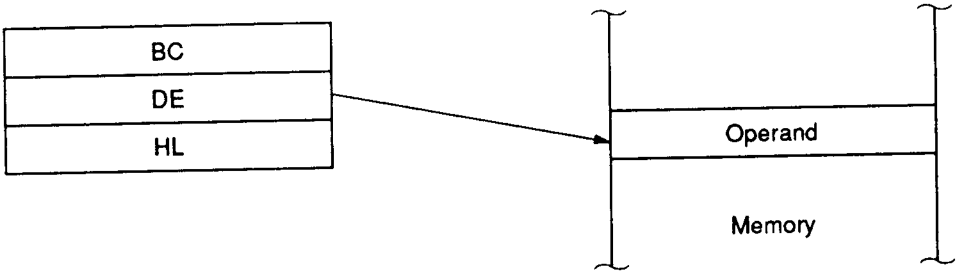
xx Field	Specified Register
00	BC
01	DE
10	IX
11	SP

yy Field	Specified Register
00	BC
01	DE
10	IY
11	SP

zz Field	Specified Register
00	BC
01	DE
10	HL
11	AF

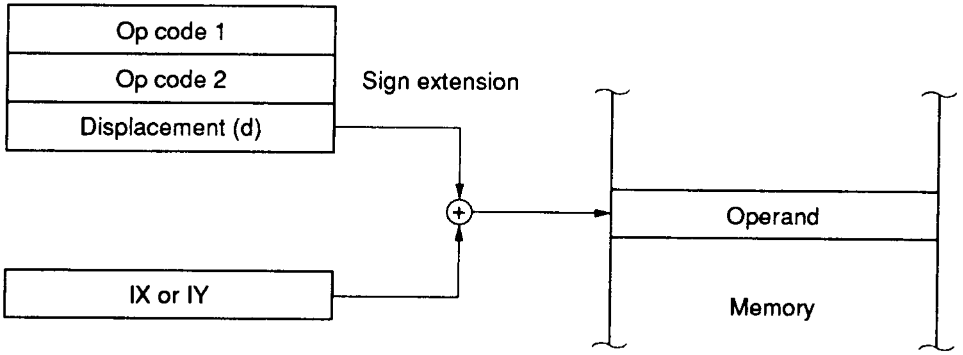
3. Register Indirect Addressing (REGI)

In this addressing mode, general-purpose registers are used as 16-bit address registers for specification of operands in memory.



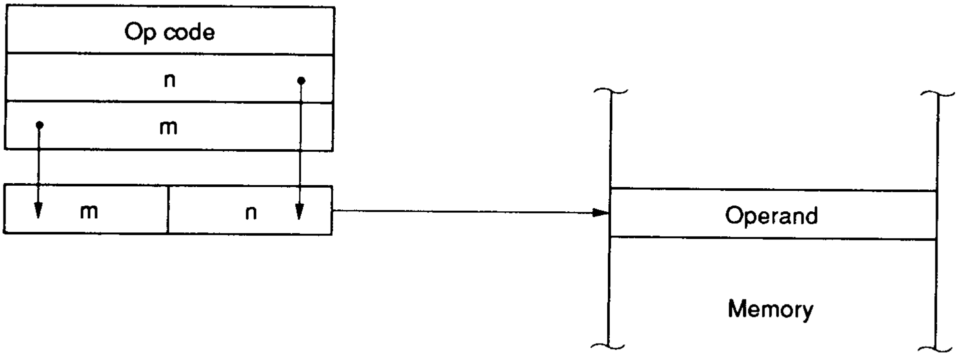
4. Indexed Addressing (INDX)

In this addressing mode, effective addresses are generated by adding a displacement (d: signed 8-bit value) to the index registers (IX, IY).



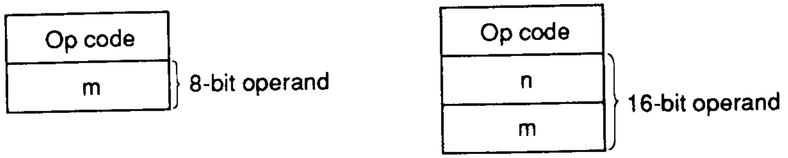
5. Extended Addressing (EXT)

With this addressing mode, 2 bytes (m, n) following the op code are used as a 16-bit address for direct specification of an operand in memory.



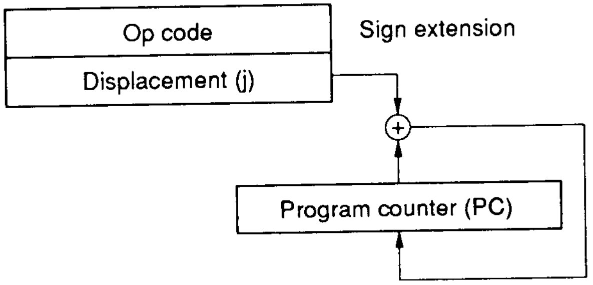
6. Immediate Addressing (IMMED)

With this addressing mode, 1 byte (m) or 2 bytes (m, n) following the op code are used as a direct operand.



7. Relative Addressing (REL)

This addressing mode can be used with branching instructions only. A displacement (j: signed 8-bit value) is added to the program counter (PC) to generate a branch address. In the case of a conditional branch, the branch is taken only when a condition is satisfied.



8. I/O Addressing

This addressing mode can be used with I/O instructions only. The specified address is an I/O address ($\overline{IOE}$ = 0). One of the following addresses is output by this addressing mode.

- An address formed by the content of the operand, which is output to A_0 through A_7, and the content of accumulator A, which is output to A_8 through A_{15}.
- An address formed by the content of register C, which is output to A_0 through A_7, and the content of register B, which is output to A_8 through A_{15}.
- An address formed by the content of the operand, which is output to A_0 through A_7, and 00H, which is output to A_8 through A_{15}. This is convenient for accessing on-chip I/O registers.
- An address formed by the content of register C, which is output to A_0 through A_7, and 00H, which is output to A_8 through A_{15}. This is convenient for accessing on-chip I/O registers.

Instruction Set Overview

The CPU instruction set for this chip can be broken down as follows.

- Operation instructions
- Load instructions
- Program control instructions
- I/O instructions
- Special control instructions

These instructions consist of 1 to 4 bytes. Typical examples are shown below.

	7	6	5	4	3	2	1	0	
1-byte instruction	0	1	g			g'			LD g, g'

	7	6	5	4	3	2	1	0	
2-byte instruction	0	0	g			1	1	0	LD g, m
	m								Immediate data

	7	6	5	4	3	2	1	0	
3-byte instruction	1	1	0	1	1	1	0	1	LD g, (IX + d)
	0	1	g			1	1	0	
	d								Displacement

	7	6	5	4	3	2	1	0	
4-byte instruction	1	1	0	1	1	1	0	1	LD (IX + d), m
	0	0	1	1	0	1	1	0	
	d								Displacement
	m								Immediate data

See Instruction Set Lists for further details.

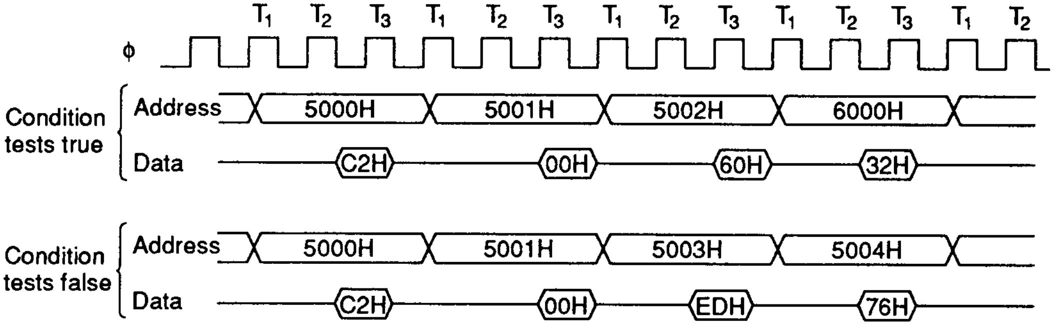
Conditional Jump and Conditional Call Instruction Precautions

The following illustrates operation when the conditional jump instruction JP f, mn is executed. Note the difference in operation when the condition tests true and when it tests false.

Example

When the instruction JP NZ, 6000H is at address 5000H (MMU sets base register to 00H).

5000H	JP NZ, (C2H)
5001H	00H
5002H	60H
5003H	SLP (EDH)
	76H
6000H	LD (32H)
6001H	00
	70



Note that when the condition tests false, execution proceeds to the next instruction without reading the 2nd operand of JP f, mm. This is also true in the case of the conditional call: CALL f, mn.

■ Absolute Maximum Ratings

Item	Symbol	Rating	Unit
Analog supply voltage	AV_{CC}	–0.3 to +7.0	V*1
Input voltage (port 4)	$V_{in}1$	–0.3 to AV_{CC} + 0.3	V
Analog input voltage	A_{in}	–0.3 to AV_{CC} + 0.3	V
Total current inflow	ΣI_{OL}	80	mA*2
Total current outflow	$\Sigma \lvert I_{OH} \rvert$	30	mA*3

Notes: 1. AV_{CC} must equal V_{CC}.

2. Total current inflow is the total current sunk simultaneously from all input/output lines via output buffers to the HD648180W's V_{SS}.

3. Total current outflow is the total current sourced simultaneously from the HD648180W's V_{CC} via output buffers to all input/output pins.

■ Electrical Characteristics

DC Characteristics (V_{CC} = 5 V ±10%, V_{SS} = 0 V, T_a = –20 to +75°C unless otherwise noted)

Item	Symbol	Test Conditions	Min	Typ	Max	Unit
Input high voltage level for $\overline{\text{RESET}}$, EXTAL, $\overline{\text{NMI}}$, $\overline{\text{STBY}}$	V_{IH1}		V_{CC} – 0.6	—	V_{CC} + 0.3	V
Input high voltage level for other pins	V_{IH2}		2.0	—	V_{CC} + 0.3	V
Input low voltage level for $\overline{\text{RESET}}$, EXTAL, $\overline{\text{NMI}}$, $\overline{\text{STBY}}$	V_{IL1}		–0.3	—	0.6	V
Input low voltage level for other pins	V_{IL2}		–0.3	—	0.8	V
Output high voltage level for all output pins	V_{OH}	I_{OH} = –200 µA	2.4	—	—	V
		I_{OH} = –20 µA	V_{CC} – 1.2	—	—	
Output low voltage level for all output pins	V_{OL}	I_{OL} = 2.2 mA	—	—	0.45	V
Input leakage current for all input pins except XTAL and EXTAL	I_{IL}	V_{in} = 0.5 to V_{CC} – 0.5	—	—	1.0	µA
Tri-state leakage current	I_{TL}	V_{in} = 0.5 to V_{CC} – 0.5	—	—	1.0	µA
Current consumption (normal operation)$^{\text{Note}}$	I_{CC}	f = 4 MHz	—	15	30	mA
		f = 6 MHz	—	20	40	mA
Current consumption (standby mode)$^{\text{Note}}$	I_{CCSTBY}	—	—	10	20	µA
Pin capacitance	C_P	V_{in} = 0 V, f = 1 MHz T_a = 25°C	—	—	15	pF

Note: Signal levels:
$\overline{\text{RESET}}$, EXTAL, $\overline{\text{NMI}}$: V_{IH}min = V_{CC} – 0.6 V, V_{IL}max = 0.6 V
Other pins: V_{IH}min = V_{CC} – 1.0 V, V_{IL}max = 0.8 V
All output pins are unloaded.

- **DC Characteristics (I/O ports) (V_{cc}=5V ± 10%, V_{SS}=0 V, T_a=–20 to +75°C unless otherwise noted)**

Item		Symbol	Test Conditions	Min	Typ	Max	Unit
Input high voltage	Ports 0 to 4	V_{IHP}		2.0	—	V_{CC} + 0.3	V
Input low voltage	Ports 0 to 4	V_{ILP}		–0.3	—	0.8	V
Output high voltage	Ports 0 to 3	V_{OHP}	I_{OH} = –200 µA	2.4	—	—	V
			I_{OH} = –20 µA	V_{CC} – 1.2	—	—	
Output low voltage	Ports 0 to 3	V_{OLP}	I_{OL} = 2.2 mA	—	—	0.45	V
	Port 0		I_{OL} = 10 mA	—	—	1.0	
Input leakage current		$\lvert I_{ILP}\rvert$	V_{in} = 0.5 to V_{CC} – 0.5	—	—	1.0	µA

- **A/D Conversion Characteristics (AV_{CC}=V_{CC}, V_{SS}=0 V, T_a=–20 to +75°C unless otherwise noted)**

Item	Symbol	Test Conditions	Min	Typ	Max	Unit
Analog input voltage	AV_{in}		AV_{SS}	—	AV_{CC}	V
Analog input leakage current	$\lvert I_{AL}\rvert$		—	—	1.0	µA
Resolution			—	8	—	Bit
Input hold time (conversion time)		A/D conversion clock = ϕ/3 = 2.048 MHz when ϕ = 6.144 MHz	16.6	—	—	µs
Number of inputs			0	—	4	Channel
Absolute precision		T_a = 25°C, V_{CC} = 5.0 V	—	—	2.0	LSB

Note: AV_{CC} must equal V_{CC}.

AC Characteristics (V_{CC} = 5 V ±10%, V_{SS} = 0 V, T_a = –20 to +75°C unless otherwise noted)

			HD648180W-4			HD648180W-6			
Item	Symbol	Test Conditions	Min	Typ	Max	Min	Typ	Max	Unit
Clock cycle time	t_{cyc}	Figure 6	250	—	333	162	—	333	ns
Clock high pulse duration	t_{CHW}		110	—	—	65	—	—	ns
Clock low pulse duration	t_{CLW}		110	—	—	65	—	—	ns
Clock fall time	t_{cf}		—	—	15	—	—	15	ns
Clock rise time	t_{cr}		—	—	15	—	—	15	ns
External clock cycle time	t_{ECYC}	Figure 18	125	—	—	81	—	—	ns
External clock high pulse duration	t_{EXHW}		50	—	—	32	—	—	ns
External clock low pulse duration	t_{EXLW}		50	—	—	32	—	—	ns
External clock rise time	t_{EXr}*1	Figure 18	—	—	25	—	—	25	ns
External clock fall time	t_{EXf}*1		—	—	25	—	—	25	ns
Address delay time	t_{AD}	Figure 6	—	—	110	—	—	90	ns
Address setup time (measured from $\overline{ME}$ or $\overline{IOE}$ falling edge)	t_{AS}		50	—	—	30	—	—	ns
$\overline{ME}$ delay time 1	t_{MED1}	Figure 6, 7,	—	—	85	—	—	60	ns
$\overline{RD}$ delay time 1 (when $\overline{IOC}$ = 1)	t_{RDD1}	Figure 6	—	—	85	—	—	60	ns
(when $\overline{IOC}$ = 0)		Figure 8	—	—	85	—	—	65	ns
$\overline{LIR}$ delay time 1	t_{LD1}	Figure 6	—	—	100	—	—	80*2	ns
Address hold time (measured from $\overline{ME}$, $\overline{IOE}$, $\overline{RD}$, or $\overline{WR}$ rising edge)	t_{AH}		80	—	—	35	—	—	ns
$\overline{ME}$ delay time 2	t_{MED2}	Figure 6, 7	—	—	85	—	—	60	ns
$\overline{RD}$ delay time 2	t_{RDD2}	Figure 6	—	—	85	—	—	60	ns
$\overline{LIR}$ delay time 2	t_{LD2}		—	—	100	—	—	80*2	ns

Notes: 1. If external clock pulse duration specifications (t_{EXHW}, t_{EXLW}) are not satisfied, adjust t_{EXr} and t_{EXf}.

2. t_{LD1} (t_{LD2}) = 60 ns when bus timing test load capacitance C = 40 pF.

AC Characteristics (V_{CC} = 5 V ±10%, V_{SS} = 0 V, T_a = –20 to +75°C unless otherwise noted) (cont)

			HD648180W-4			HD648180W-6			
Item	**Symbol**	**Test Conditions**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Data read setup time	t_{DRS}	Figure 6, 7	50	—	—	40	—	—	ns
Data read hold time	t_{DRH}		0	—	—	0	—	—	ns
ST delay time 1	t_{STD1}	Figure 6 Figure 10	—	—	110	—	—	90	ns
ST delay time 2	t_{STD2}		—	—	110	—	—	90	ns
$\overline{\text{WAIT}}$ setup time	t_{WS}	Figure 6	80	—	—	40	—	—	ns
$\overline{\text{WAIT}}$ hold time	t_{WH}		70	—	—	40	—	—	ns
Write data floating delay time	t_{WDZ}	Figure 6	—	—	100	—	—	95	ns
$\overline{\text{WR}}$ delay time 1	t_{WRD1}		—	—	90	—	—	65	ns
Write data delay time	t_{WDD}		—	—	110	—	—	90	ns
Write data setup time (measured from WR falling edge)	t_{WDS}		60	—	—	40	—	—	ns
$\overline{\text{WR}}$ delay time 2	t_{WRD2}		—	—	90	—	—	80	ns
$\overline{\text{WR}}$ pulse duration	t_{WRP}		280	—	—	170	—	—	ns
Write data hold time (measured from $\overline{\text{WR}}$ rising edge)	t_{WDH}		60	—	—	40	—	—	ns
$\overline{\text{IOE}}$ delay time 1 (when $\overline{\text{IOC}}$ = 1)	t_{IOD1}	Figure 6	—	—	85	—	—	60	ns
(when $\overline{\text{IOC}}$ = 0)		Figure 8	—	—	85	—	—	65	ns
$\overline{\text{IOE}}$ delay time 2	t_{IOD2}	Figure 6	—	—	85	—	—	60	ns
$\overline{\text{IOE}}$ delay time 3 (measured from $\overline{\text{LIR}}$ falling edge)	t_{IOD3}	Figure 7	540	—	—	340	—	—	ns

AC Characteristics (V_{CC} = 5 V ±10%, V_{SS} = 0 V, T_a = –20 to +75°C unless otherwise noted) (cont)

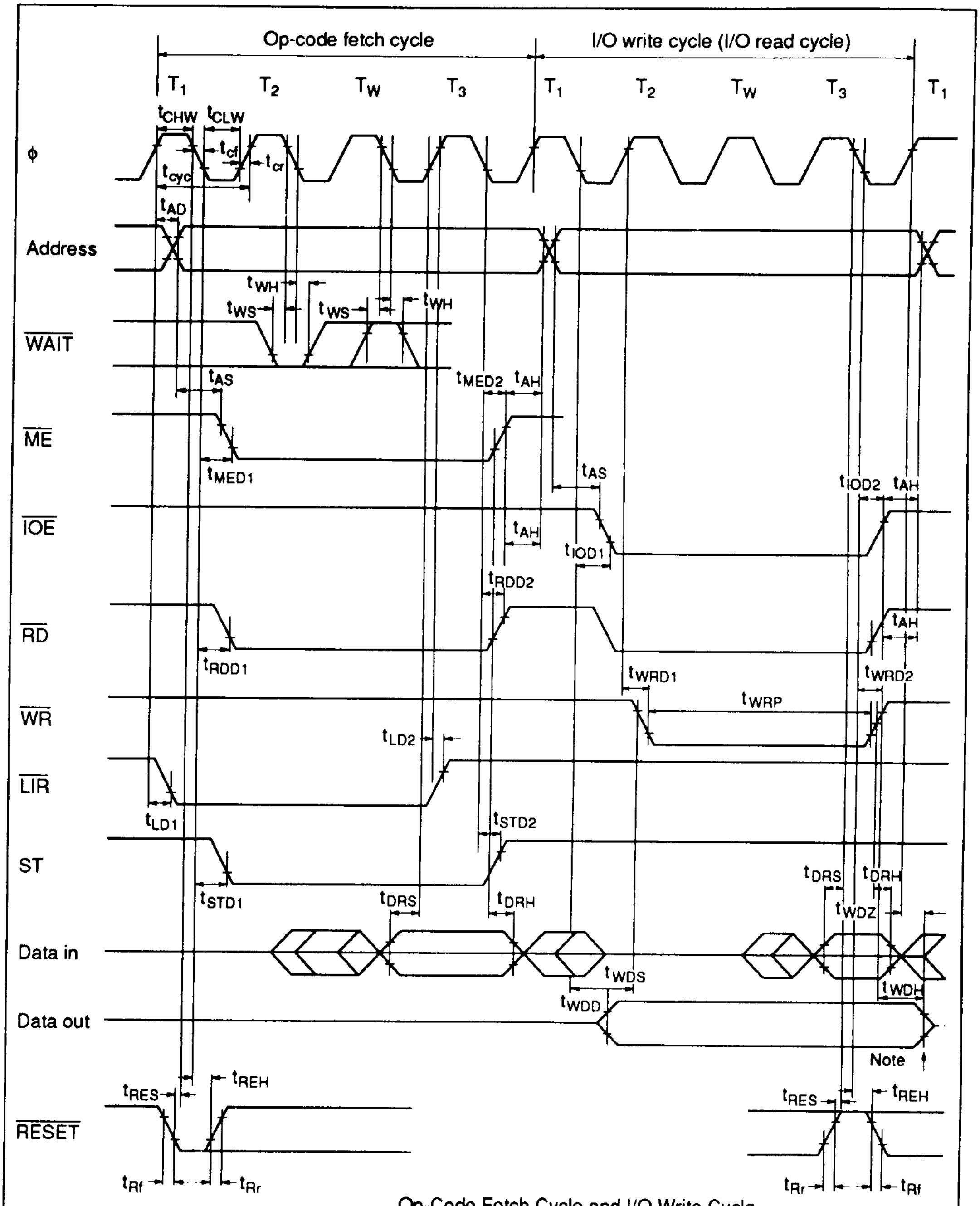
			HD648180W-4			HD648180W-6			
Item	**Symbol**	**Test Conditions**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
$\overline{\text{INT}}$ setup time (measured from ϕ falling edge)	t_{INTS}	Figure 7 Figure 13	80	—	—	40	—	—	ns
$\overline{\text{INT}}$ hold time (measured from ϕ falling edge)	t_{INTH}		70	—	—	40	—	—	ns
$\overline{\text{NMI}}$ pulse duration	t_{NMIW}		120	—	—	120	—	—	ns
$\overline{\text{BUSREQ}}$ setup time (measured from ϕ falling edge)	t_{BRS}	Figure 7	80	—	—	40	—	—	ns
$\overline{\text{BUSREQ}}$ hold time (measured from ϕ falling edge)	t_{BRH}		70	—	—	40	—	—	ns
$\overline{\text{BUSACK}}$ delay time 1	t_{BAD1}		—	—	100	—	—	95	ns
$\overline{\text{BUSACK}}$ delay time 2	t_{BAD2}		—	—	100	—	—	95	ns
Bus floating delay time	t_{BZD}		—	—	130	—	—	125	ns
$\overline{\text{ME}}$ pulse duration (high)	t_{MEWH}		200	—	—	110	—	—	ns
$\overline{\text{ME}}$ pulse duration (low)	t_{MEWL}		210	—	—	125	—	—	ns
Port data output delay time	t_{PWD}	Figure 13	—	—	110	—	—	90	ns
Port data input setup time	t_{PDS}	Figure 13	180	—	—	150	—	—	ns
Port data input hold time	t_{PDH}	Figure 13	60	—	—	40	—	—	ns
$\overline{\text{REF}}$ delay time 1	t_{RFD1}	Figure 7	—	—	110	—	—	90	ns
$\overline{\text{REF}}$ delay time 2	t_{RFD2}		—	—	110	—	—	90	ns
$\overline{\text{HALT}}$ delay time 1	t_{HAD1}	Figure 7	—	—	110	—	—	90	ns
$\overline{\text{HALT}}$ delay time 2	t_{HAD2}	Figure 12	—	—	110	—	—	90	ns
$\overline{\text{DREQi}}$ setup time 1	t_{DRQS}	Figure 10	80	—	—	40	—	—	ns
$\overline{\text{DREQi}}$ hold time 1	t_{DRQH}		70	—	—	40	—	—	ns
$\overline{\text{TENDi}}$ delay time 1	t_{TED1}		—	—	85	—	—	70	ns
$\overline{\text{TENDi}}$ delay time 2	t_{TED2}		—	—	85	—	—	70	ns

AC Characteristics (V_{CC} = 5 V ±10%, V_{SS} = 0 V, T_a = –20 to +75°C unless otherwise noted) (cont)

			HD648180W-4			HD648180W-6			
Item	**Symbol**	**Test Conditions**	**Min**	**Typ**	**Max**	**Min**	**Typ**	**Max**	**Unit**
Timer output delay time	t_{TOD}	Figure 11	—	—	300	—	—	300	ns
SCI input clock cycle (clocked synchronous mode)	t_{scyc}	Figure 17	40	—	—	40	—	—	t_{cyc}
SCI transmit data delay time (clocked synchronous mode)	t_{TXD}		—	—	200	—	—	200	ns
SCI receive data setup time (clocked synchronous mode)	t_{SRX}	Figure 17	260	—	—	260	—	—	ns
SCI receive data hold time (clocked synchronous mode)	t_{HRX}		100	—	—	100	—	—	ns
SCI input clock pulse duration	t_{PWSCXH} t_{PWSCXL}	Figure 14	0.4*1	—	0.6*1	0.4*1	—	0.6*1	t_{scyc}
Timer 1 input clock pulse duration	t_{PWTH} t_{PWTL}	Figure 15	8	—	—	8	—	—	t_{cyc}
$\overline{\text{RESET}}$ setup time	t_{RES}	Figure 15	120	—	—	120	—	—	ns
$\overline{\text{RESET}}$ hold time	t_{REH}		80	—	—	80	—	—	ns
$\overline{\text{RESET}}$ rise time	t_{Rr}		—	—	50*2	—	—	50*2	ms
$\overline{\text{RESET}}$ fall time	t_{Rf}		—	—	50*2	—	—	50*2	ms
Input pin rise time (other than $\overline{\text{RESET}}$)	t_{Ir}	Figure 19	—	—	100*2	—	—	100*2	ns
Input pin fall time (other than $\overline{\text{RESET}}$)	t_{If}		—	—	100*2	—	—	100*2	ns
$\overline{\text{ADT}}$ input pulse duration	t_{PWADH} t_{PWADL}	Figure 15	8	—	—	8	—	—	t_{cyc}
$\overline{\text{STBY}}$ input delay duration	t_{STBYD}	Figure 6	0	—	—	0	—	—	ns
Oscillation stabilization time	t_{OSC}	Figure 9	—	—	20	—	—	20	ms

Notes: 1. Set so: (t_{PWSCXH}) + (t_{PWSCXL}) + (t_{IR}) + (t_{IF}) = t_{SCYC}.

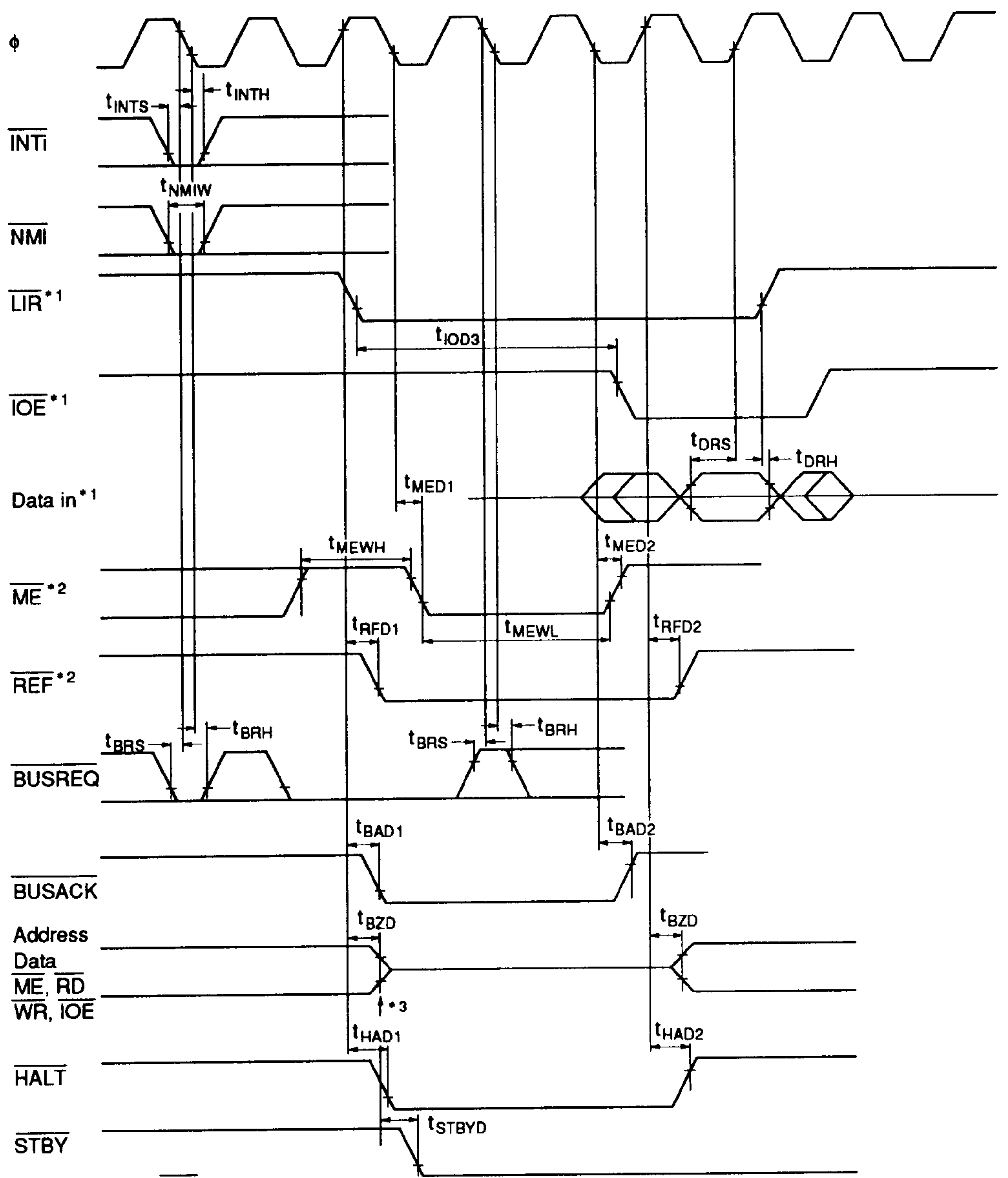
2. If these rise and fall times are satisfied, but other specifications are not satisfied, adjust these rise and fall times to satisfy the other specifications.



Op-Code Fetch Cycle and I/O Write Cycle
(I/O Read Cycle) when $\overline{IOC}$ = 1

Note: Output buffer off point.

Figure 6 CPU Timing (1)



Notes: 1. For $\overline{\text{INT}}_0$ acknowledge cycle
2. For refresh cycle
3. Output buffer off point

$\overline{\text{INT}}_0$ Acknowledge Cycle, Refresh Cycle, Bus Release Mode, Halt Mode, sleep Mode, and System Stop Mode

Figure 7 CPU Timing (2)

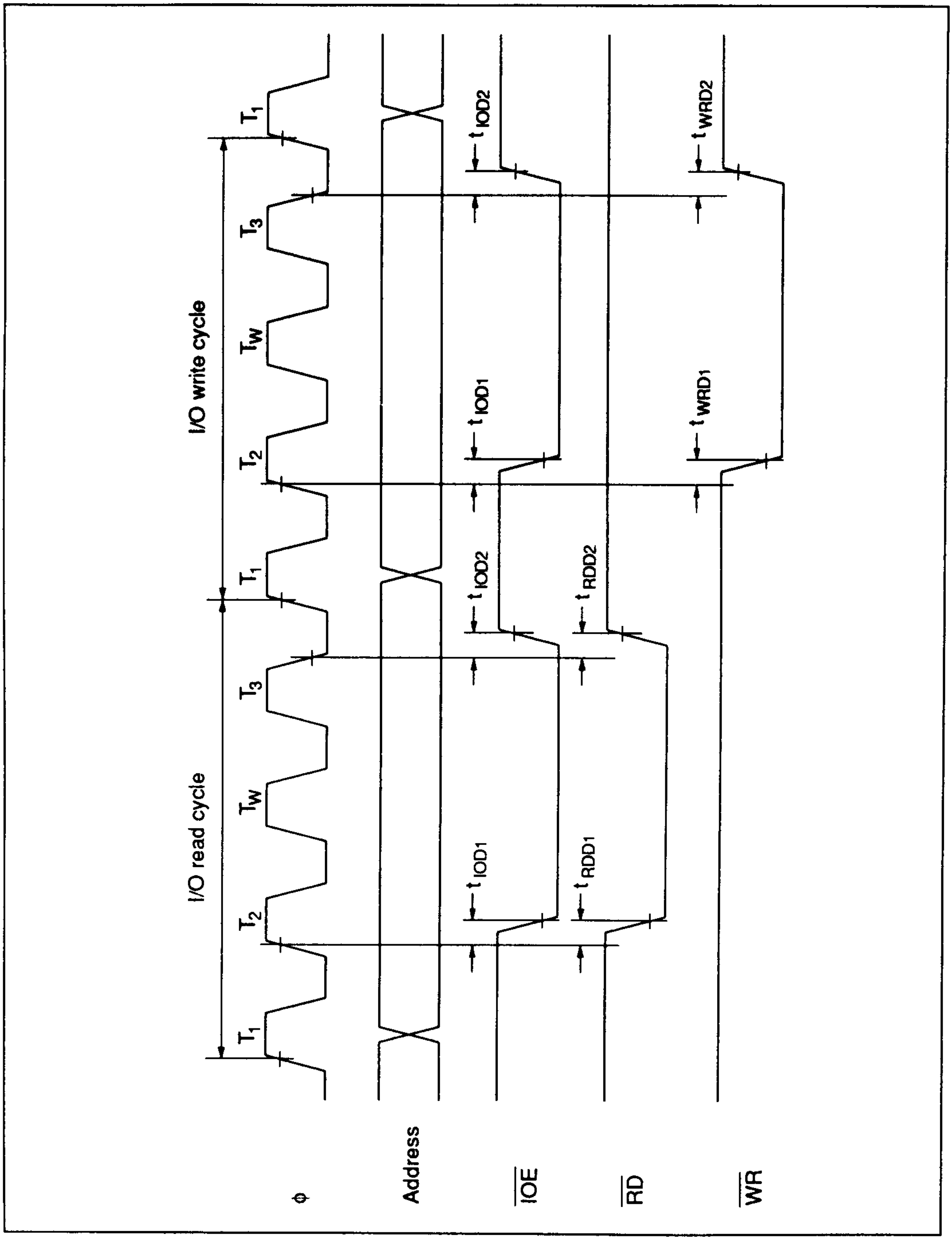


Figure 8 CPU Timing (3)

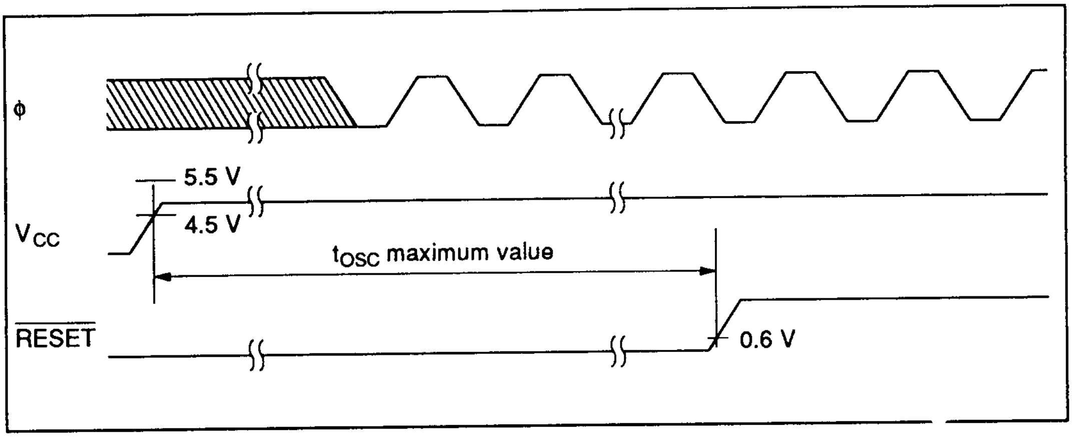
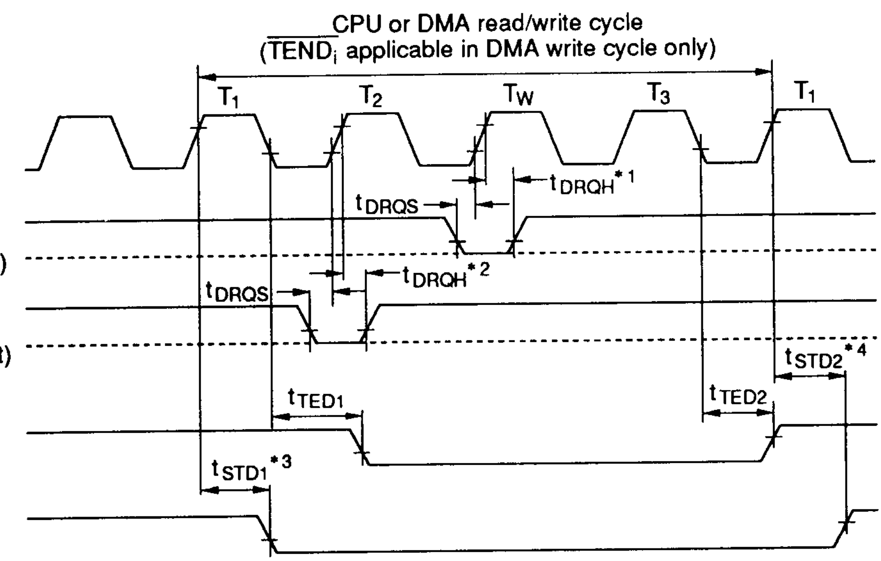


Figure 9 CPU Timing (4)



Notes: 1. Prescribed for clock rising edge immediately before T_3
2. Prescribed for each clock rising edge
3. T_1 of DMA cycle start
4. T_1 of CPU cycle start

Figure 10 DMA Control Signals

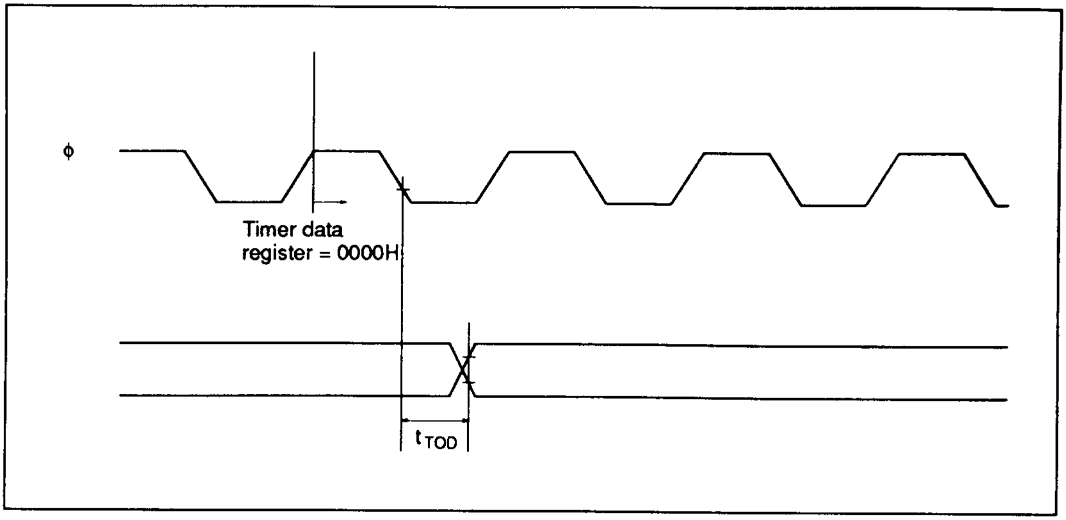


Figure 11 Timer Output

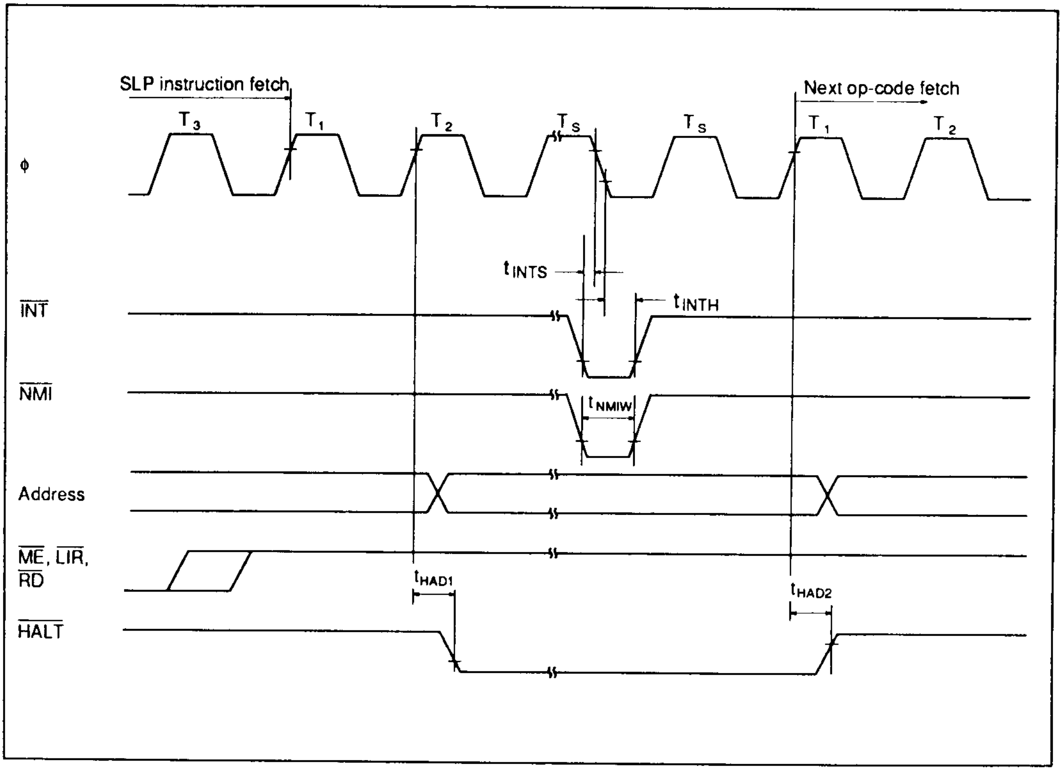


Figure 12 SLP Execution Cycle

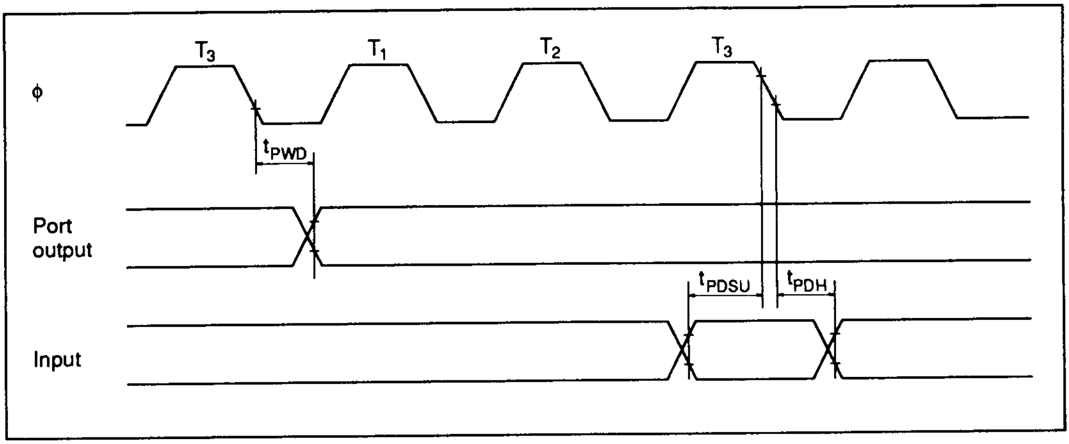


Figure 13 Port I/O Timing

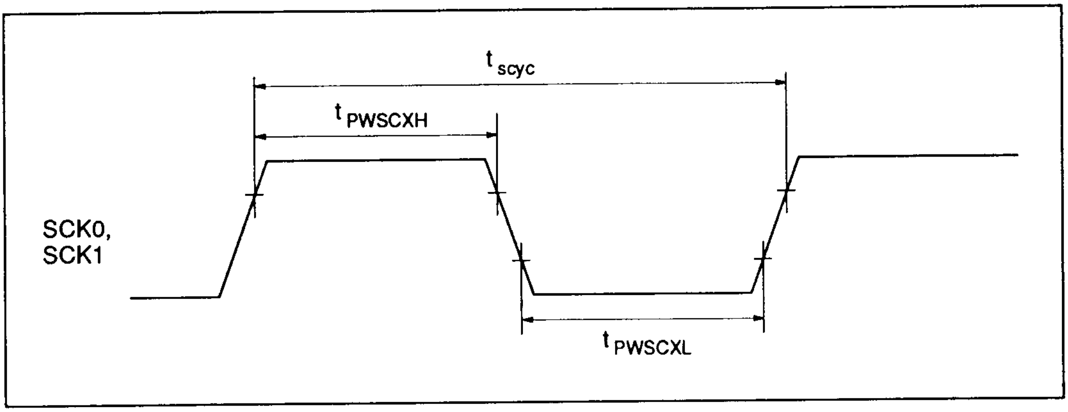


Figure 14 SCI Clock Input Timing

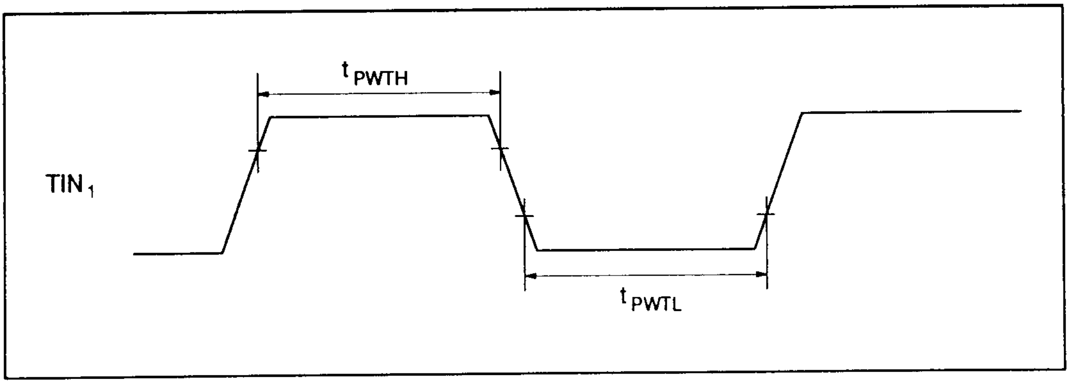


Figure 15 Timer 1 Input Pulse Duration

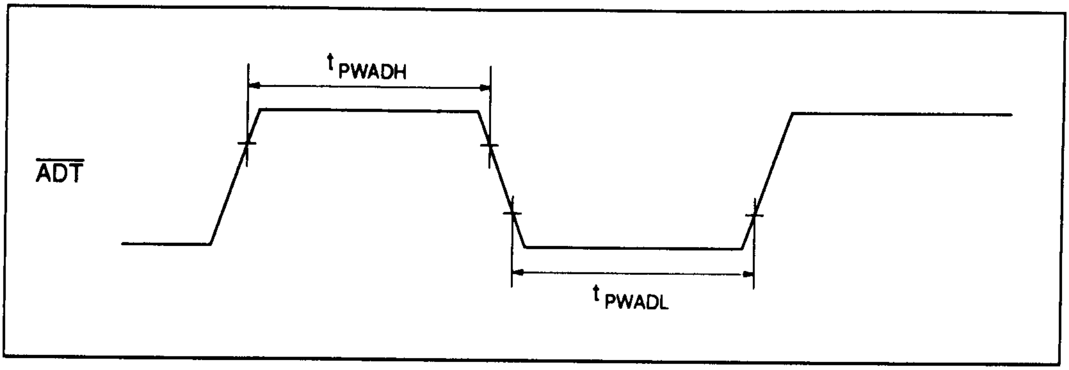


Figure 16 $\overline{\text{ADT}}$ Input Pulse Duration

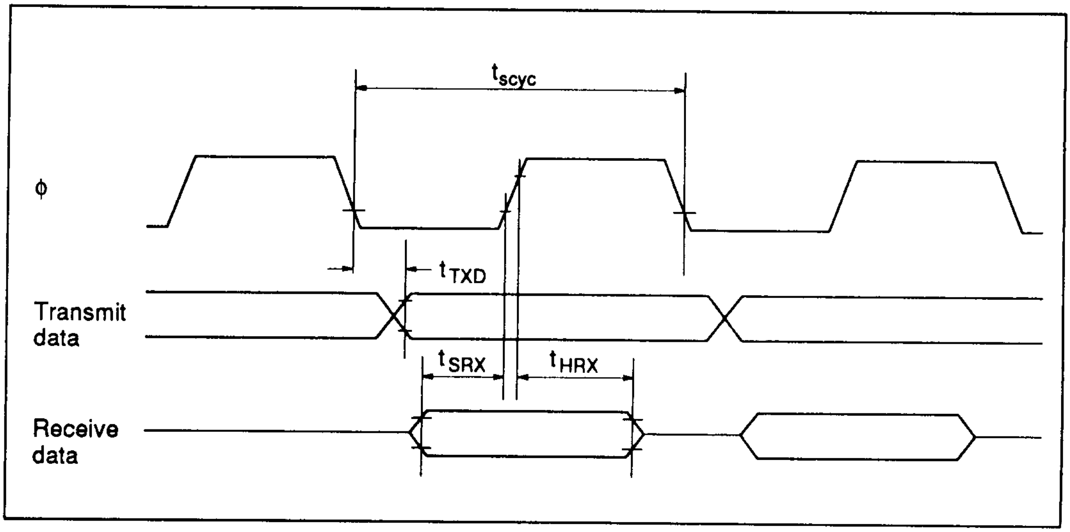


Figure 17 SCI Clocked Synchronous Timing (direct phase)

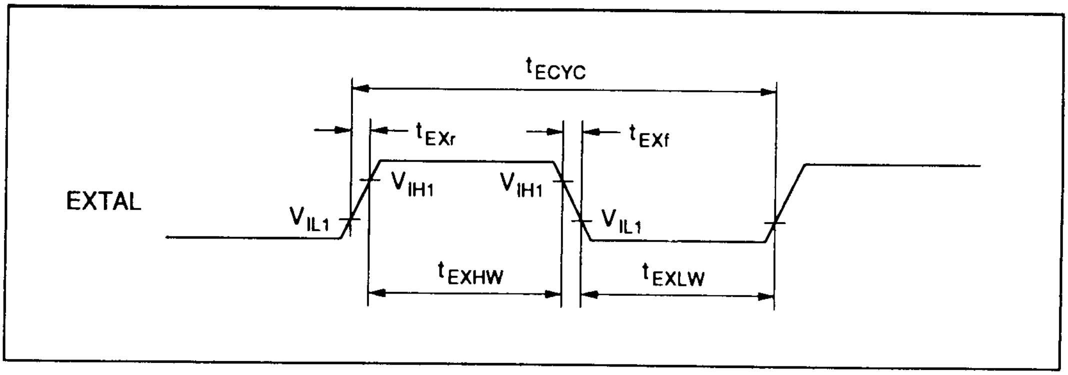


Figure 18 Rise and Fall Times for External Clock Input Signal

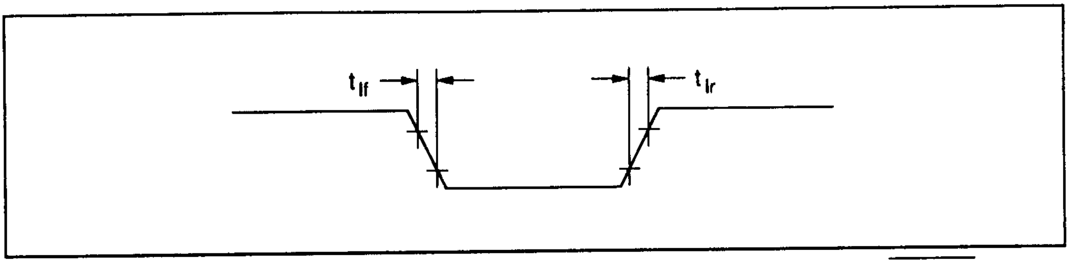


Figure 19 Rise and Fall Times for Input Signals (except EXTAL AND $\overline{\text{RESET}}$)

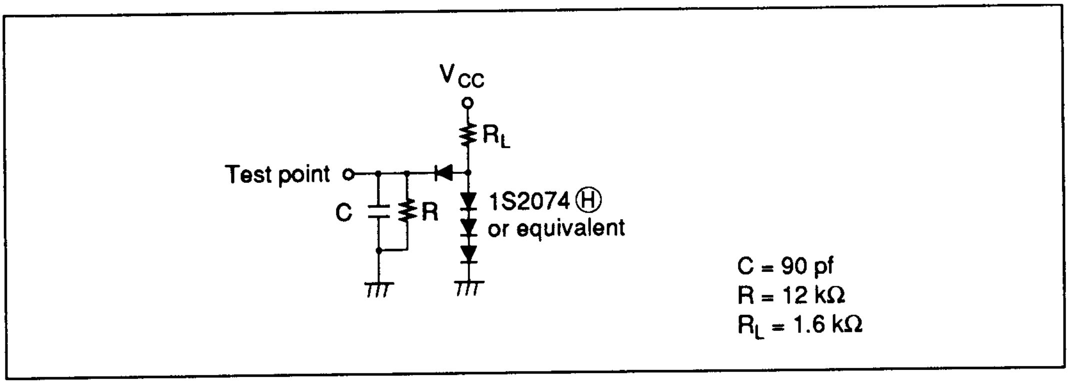


Figure 20 Bus Timing Test Load (TTL load)

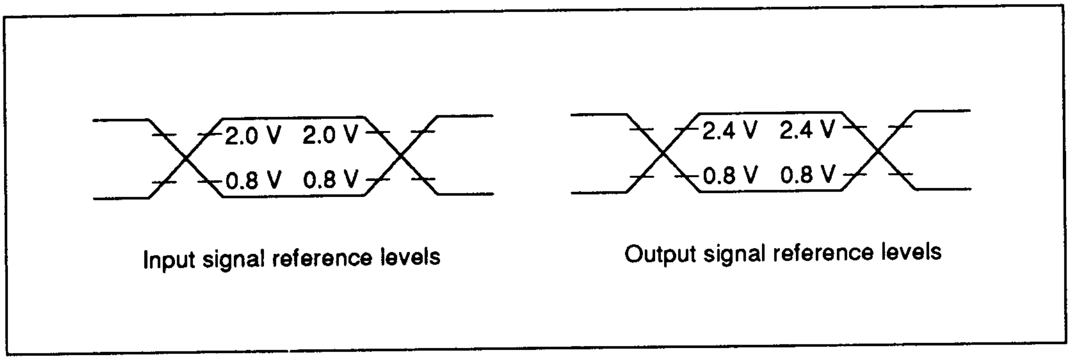


Figure 21 Reference Levels

■ Instruction Set Lists

The following explains the symbols used throughout the instruction set lists contained in this appendix.

1. Register Specification

The register specification symbols are: g, g', ww, xx, yy, and zz. The symbols g and g' represent 8-bit registers, while ww, xx, yy, and zz represent 16-bit registers, as shown below.

g, g'	Reg.	ww	Reg.	xx	Reg.	yy	Reg.	zz	Reg.
0 0 0	B	0 0	BC	0 0	BC	0 0	BC	0 0	BC
0 0 1	C	0 1	DE	0 1	DE	0 1	DE	0 1	DE
0 1 0	D	1 0	HL	1 0	IX	1 0	IY	1 0	HL
0 1 1	E	1 1	SP	1 1	SP	1 1	SP	1 1	AF
1 0 0	H								
1 0 1	L								
1 1 1	A								

Note: The letters "H" (high) and "L" (low) are appended to the 16-bit register symbols to indicate the upper 8 bits and lower 8 bits. For example: wwH, IXL.

2. Bit Specification

The symbol b in the bit operand of a bit manipulation instruction, specifies the bit location as shown below.

b	Bit
0 0 0	0
0 0 1	1
0 1 0	2
0 1 1	3
1 0 0	4
1 0 1	5
1 1 0	6
1 1 1	7

3. Condition Specification

The symbol f specifies the condition against which the result of an operation is compared to determine the next instruction to be executed, as shown below.

f	Condition	
0 0 0	NZ	non zero
0 0 1	Z	zero
0 1 0	NC	non carry
0 1 1	C	carry
1 0 0	PO	parity odd
1 0 1	PE	parity even
1 1 0	P	sign plus
1 1 1	M	sign minus

4. Restart Address

The symbol v specifies the restart address for a restart instruction, as shown below.

v	Address
0 0 0	00H
0 0 1	08H
0 1 0	10H
0 1 1	18H
1 0 0	20H
1 0 1	28H
1 1 0	30H
1 1 1	38H

5. Flags

The following symbols are used to denote changes in flags.

–: Flag not affected
×: Flag change undefined
↕: Flag affected according to operation result
S: Flag set to 1
R: Flag reset to 0
P: Flag changes as a parity flag
V: Flag changes as an overflow flag

6. Other Symbols

$(\ )_M$: Parentheses contain memory address
$(\ )_I$: Parentheses contain I/O address
m or n: 8-bit value
mn: 16-bit value
r: r suffix indicates 8-bit register
R: R suffix indicates 16-bit register
b • $(\ )_M$: b-th bit of memory address in parentheses
b • gr: b-th bit of general register gr
d or j: Signed 8-bit displacement
S: Source addressing mode
D: Destination addressing mode
•: AND operation
+: OR operation
⊕: XOR operation

• Data Manipulation Instructions

1. Arithmetic and Logical Instructions (8-bit)

			Addressing										Flag					
													7	6	4	2	1	0
Operation Name	Mnemonics	Op Code	IMMED	EXT	INDX	REG	REGI	IMP	REL	Bytes	States	Operation	S	Z	H	P/V	N	C
ADD	ADD A, g	10 000 g				S		D		1	4	Ar + gr → Ar	↕	↕	↕	V	R	↕
	ADD A, (HL)	10 000 110					S	D		1	6	Ar + $(HL)_M$ → Ar	↕	↕	↕	V	R	↕
	ADD A, m	11 000 110 <m>	S					D		2	6	Ar + m → Ar						
	ADD A, (IX + d)	11 011 101 10 000 110 <d>			S			D		3	14	Ar + $(IX + d)_M$ → Ar	↕	↕	↕	V	R	↕
	ADD A, (IY + d)	11 111 101 10 000 110 <d>			S			D		3	14	Ar + $(IY + d)_M$ → Ar	↕	↕	↕	V	R	↕
ADC	ADC A, g	10 001 g				S		D		1	4	Ar + gr + c → Ar	↕	↕	↕	V	R	↕
	ADC A, (HL)	10 001 110					S	D		1	6	Ar + $(HL)_M$ + c → Ar	↕	↕	↕	V	R	↕
	ADC A, m	11 001 110 <m>	S					D		2	6	Ar + m + c → Ar	↕	↕	↕	V	R	↕
	ADC A, (IX + d)	11 011 101 10 001 110 <d>			S			D		3	14	Ar + $(IX + d)_M$ + c → Ar	↕	↕	↕	V	R	↕
	ADC A, (IY + d)	11 111 101 10 001 110 <d>			S			D		3	14	Ar + $(IY + d)_M$ + c → Ar	↕	↕	↕	V	R	↕
AND	AND g	10 100 g				S		D		1	4	Ar·gr → Ar	↕	↕	S	P	R	R
	AND (HL)	10 100 110					S	D		1	6	Ar·$(HL)_M$ → Ar	↕	↕	S	P	R	R
	AND m	11 100 110 <m>	S					D		2	6	Ar·m → Ar	↕	↕	S	P	R	R
	AND (IX + d)	11 011 101 10 100 110 <d>			S			D		3	14	Ar·$(IX + d)_M$ → Ar	↕	↕	S	P	R	R
	AND (IY + d)	11 111 101 10 100 110 <d>			S			D		3	14	Ar·$(IY + d)_M$ → Ar	↕	↕	S	P	R	R
Compare	CP g	10 111 g				S		D		1	4	Ar − gr	↕	↕	↕	V	S	↕
	CP (HL)	10 111 110					S	D		1	6	Ar − $(HL)_M$	↕	↕	↕	V	S	↕
	CP m	11 111 110 <m>	S					D		2	6	Ar − m	↕	↕	↕	V	S	↕

3

HITACHI

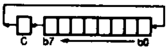
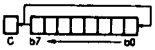
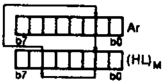
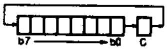
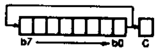
1. Arithmetic and Logical Instructions (8-bit) (cont)

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	INDX	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
Compare	CP (IX + d)	11 011 101 10 111 110 <d>			S			D		3	14	Ar − $(IX + d)_M$	↕	↕	↕	V	S	↕
	CP (IY + d)	11 111 101 10 111 110 <d>			S			D		3	14	Ar − $(IY + d)_M$	↕	↕	↕	V	S	↕
Comple-ment	CPL	00 101 111						S/D		1	3	$\overline{Ar}$ → Ar	—	—	S	—	S	—
DEC	DEC g	00 g 101				S/D				1	4	gr − 1 → gr	↕	↕	↕	V	S	—
	DEC (HL)	00 110 101					S/D			1	10	$(HL)_M$ − 1 → $(HL)_M$	↕	↕	↕	V	S	—
	DEC (IX + d)	11 011 101 00 110 101 <d>			S/D					3	18	$(IX + d)_M$ − 1 → $(IX + d)_M$	↕	↕	↕	V	S	—
	DEC (IY + d)	11 111 101 00 110 101 <d>			S/D					3	18	$(IY + d)_M$ − 1 → $(IY + d)_M$	↕	↕	↕	V	S	↕
INC	INC g	00 g 100				S/D				1	4	gr + 1 → gr	↕	↕	↕	V	R	—
	INC (HL)	00 110 100					S/D			1	10	$(HL)_M$ + 1 → $(HL)_M$	↕	↕	↕	V	R	—
	INC (IX + d)	11 011 101 00 110 100 <d>			S/D					3	18	$(IX + d)_M$ + 1 → $(IX + d)_M$	↕	↕	↕	V	R	—
	INC (IY + d)	11 111 101 00 110 100 <d>			S/D					3	18	$(IY + d)_M$ + 1 → $(IY + d)_M$	↕	↕	↕	V	R	—
MULT	MLT ww	11 101 101 01 ww1 100				S/D				2	17	wwHr × wwLr → ww	—	—	—	—	—	—
NEGATE	NEG	11 101 101 01 000 100						S/D		2	6	0 − Ar → Ar	↕	↕	↕	V	S	↕
OR	OR g	10 110 g				S		D		1	4	Ar + gr → Ar	↕	↕	R	P	R	R
	OR (HL)	10 110 110					S	D		1	6	Ar + $(HL)_M$ → Ar	↕	↕	R	P	R	R
	OR m	11 110 110 <m>	S					D		2	6	Ar + m → Ar	↕	↕	R	P	R	R
	OR (IX + d)	11 011 101 10 110 110 <d>			S			D		3	14	Ar + $(IX + d)_M$ → Ar	↕	↕	R	P	R	R
	OR (IY + d)	11 111 101 10 110 110 <d>			S			D		3	14	Ar + $(IY + d)_M$ → Ar	↕	↕	R	P	R	R

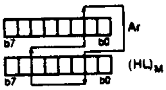
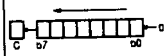
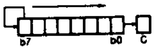
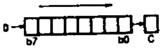
1. Arithmetic and Logical Instructions (8-bit) (cont)

			Addressing										Flag					
													7	6	4	2	1	0
Operation Name	Mnemonics	Op Code	IMMED	EXT	INDX	REG	REGI	IMP	REL	Bytes	States	Operation	S	Z	H	P/V	N	C
SUB	SUB g	10 010 g				S		D		1	4	$Ar - gr \rightarrow Ar$	↕	↕	↕	V	S	↕
	SUB (HL)	10 010 110					S	D		1	6	$Ar - (HL)_M \rightarrow Ar$	↕	↕	↕	V	S	↕
	SUB m	11 010 110 <m>	S					D		2	6	$Ar - m \rightarrow Ar$	↕	↕	↕	V	S	↕
	SUB (IX + d)	11 011 101 10 010 110 <d>			S			D		3	14	$Ar - (IX + d)_M \rightarrow Ar$	↕	↕	↕	V	S	↕
	SUB (IY + d)	11 111 101 10 010 110 <d>			S			D		3	14	$Ar - (IY + d)_M \rightarrow Ar$	↕	↕	↕	V	S	↕
SUBC	SBC A, g	10 011 g				S		D		1	4	$Ar - gr - c \rightarrow Ar$	↕	↕	↕	V	S	↕
	SBC A, (HL)	10 011 110					S	D		1	6	$Ar - (HL)_M - c \rightarrow Ar$	↕	↕	↕	V	S	↕
	SBC A, m	11 011 110 <m>	S					D		2	6	$Ar - m - c \rightarrow Ar$	↕	↕	↕	V	S	↕
	SBC A, (IX + d)	11 011 101 10 011 110 <d>			S			D		3	14	$Ar - (IX + d)_M - c \rightarrow Ar$	↕	↕	↕	V	S	↕
	SBC A, (IY + d)	11 111 101 10 011 110 <d>			S			D		3	14	$Ar - (IY + d)_M - c \rightarrow Ar$	↕	↕	↕	V	S	↕
TEST	TST g	11 101 101 00 g 100				S				2	7	$Ar \cdot gr$	↕	↕	S	P	R	R
	TST (HL)	11 101 101 00 110 100					S			2	10	$Ar \cdot (HL)_M$	↕	↕	S	P	R	R
	TST m	11 101 101 01 100 100 <m>	S							3	9	$Ar \cdot m$	↕	↕	S	P	R	R
XOR	XOR g	10 101 g				S		D		1	4	$Ar \oplus gr \rightarrow Ar$	↕	↕	R	P	R	R
	XOR (HL)	10 101 110					S	D		1	6	$Ar \oplus (HL)_M \rightarrow Ar$	↕	↕	R	P	R	R
	XOR m	11 101 110 <m>	S					D		2	6	$Ar \oplus m \rightarrow Ar$	↕	↕	R	P	R	R
	XOR (IX + d)	11 011 101 10 101 110 <d>			S			D		3	14	$Ar \oplus (IX + d)_M \rightarrow Ar$	↕	↕	R	P	R	R
	XOR (IY + d)	11 111 101 10 101 110 <d>			S			D		3	14	$Ar \oplus (IY + d)_M \rightarrow Ar$	↕	↕	R	P	R	R

2. Rotate and Shift Instructions

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	INDX	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
Rotate and shift data	RLA	00 010 111						S/D		1	3		—	—	R	—	R	↕
	RL g	11 001 011 00 010 g				S/D				2	7		↕	↕	R	P	R	↕
	RL (HL)	11 001 011 00 010 110					S/D			2	13		↕	↕	R	P	R	↕
	RL (IX + d)	11 011 101 11 001 011 <d> 00 010 110			S/D					4	19		↕	↕	R	P	R	↕
	RL (IY + d)	11 111 101 11 001 011 <d> 00 010 110			S/D					4	19		↕	↕	R	P	R	↕
	RLCA	00 000 111						S/D		1	3		—	—	R	—	R	↕
	RLC g	11 001 011 00 000 g				S/D				2	7		↕	↕	R	P	R	↕
	RLC (HL)	11 001 011 00 000 110					S/D			2	13		↕	↕	R	P	R	↕
	RLC (IX + d)	11 011 101 11 001 011 <d> 00 000 110			S/D					4	19		↕	↕	R	P	R	↕
	RLC (IY + d)	11 111 101 11 001 011 <d> 00 000 110			S/D					4	19		↕	↕	R	P	R	↕
	RLD	11 101 101 01 101 111						S/D		2	16		↕	↕	R	P	R	—
	RRA	00 011 111						S/D		1	3		—	—	R	—	R	↕
	RR g	11 001 011 00 011 g				S/D				2	7		↕	↕	R	P	R	↕
	RR (HL)	11 001 011 00 011 110					S/D			2	13		↕	↕	R	P	R	↕
	RR (IX + d)	11 011 101 11 001 011 <d> 00 011 110			S/D					4	19		↕	↕	R	P	R	↕
	RR (IY + d)	11 111 101 11 001 011 <d> 00 011 110			S/D					4	19		↕	↕	R	P	R	↕
	RRCA	00 001 111						S/D		1	3		—	—	R	—	R	↕
	RRC g	11 001 011 00 001 g				S/D				2	7		↕	↕	R	P	R	↕

2. Rotate and Shift Instructions (cont)

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	INDX	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
Rotate and shift data	RRC (HL)	11 001 011 00 001 110					S/D			2	13		↕	↕	R	P	R	↕
	RRC (IX + d)	11 011 101 11 001 011 <d> 00 001 110			S/D					4	19		↕	↕	R	P	R	↕
	RRC (IY + d)	11 111 101 11 001 011 <d> 00 001 110			S/D					4	19		↕	↕	R	P	R	↕
	RRD	11 101 101 01 100 111						S/D		2	16		↕	↕	R	P	R	—
	SLA g	11 001 011 00 100 g				S/D				2	7		↕	↕	R	P	R	↕
	SLA (HL)	11 001 011 00 100 110					S/D			2	13		↕	↕	R	P	R	↕
	SLA (IX + d)	11 011 101 11 001 011 <d> 00 100 110			S/D					4	19		↕	↕	R	P	R	↕
	SLA (IY + d)	11 111 101 11 001 011 <d> 00 100 110			S/D					4	19		↕	↕	R	P	R	↕
	SRA g	11 001 011 001 101 g				S/D				2	7		↕	↕	R	P	R	↕
	SRA (HL)	11 001 011 00 101 110					S/D			2	13		↕	↕	R	P	R	↕
	SRA (IX + d)	11 011 101 11 001 011 <d> 00 101 110			S/D					4	19		↕	↕	R	P	R	↕
	SRA (IY + d)	11 111 101 11 001 011 <d> 00 101 110			S/D					4	19		↕	↕	R	P	R	↕
	SRL g	11 001 011 00 111 g				S/D				2	7		↕	↕	R	P	R	↕
	SRL (HL)	11 001 011 00 111 110					S/D			2	3		↕	↕	R	P	R	↕
	SRL (IX + d)	11 011 101 11 001 011 <d> 00 111 110			S/D					4	19		↕	↕	R	P	R	↕
	SRL (IY + d)	11 111 101 11 001 011 <d> 00 111 110			S/D					4	19		↕	↕	R	P	R	↕

3. Bit Manipulation Instructions

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
													7	6	4	2	1	0
			IMMED	EXT	INDX	REG	REGI	IMP	REL				S	Z	H	P/V	N	C
Bit set	SET b, g	11 001 011 11 b g				S/D				2	7	1 → b·gr	—	—	—	—	—	—
	SET b, (HL)	11 001 011 11 b 110					S/D			2	13	1 → b·$(HL)_M$	—	—	—	—	—	—
	SET b, (IX + d)	11 011 101 11 001 011 <d> 11 b 110			S/D					4	19	1 → b·$(IX + d)_M$	—	—	—	—	—	—
	SET b, (IY + d)	11 111 101 11 001 011 <d> 11 b 110			S/D					4	19	1 → b·$(IY + d)_M$	—	—	—	—	—	—
Bit reset	RES b, g	11 001 011 10 b g				S/D				2	7	0 → b·gr	—	—	—	—	—	—
	RES b, (HL)	11 001 011 10 b 110					S/D			2	13	0 → b·$(HL)_M$	—	—	—	—	—	—
	RES b, (IX + d)	11 011 101 11 001 011 <d> 10 b 110			S/D					4	19	0 → b·$(IX + d)_M$	—	—	—	—	—	—
	RES b, (IY + d)	11 111 101 11 001 011 <d> 10 b 110			S/D					4	19	0 → b·$(IY + d)_M$	—	—	—	—	—	—
Bit test	BIT b, g	11 001 011 01 b g				S				2	6	$\overline{b \cdot gr} \rightarrow z$	X	↕	S	X	R	—
	BIT b, (HL)	11 001 011 01 b 110					S			2	9	$\overline{b \cdot (HL)_M} \rightarrow z$	X	↕	S	X	R	—
	BIT b, (IX + d)	11 011 101 11 001 011 <d> 01 b 110			S					4	15	$\overline{b \cdot (IX + d)_M} \rightarrow z$	X	↕	S	X	R	—
	BIT b, (IY + d)	11 111 101 11 001 011 <d> 01 b 110			S					4	15	$\overline{b \cdot (IY + d)_M} \rightarrow z$	X	↕	S	X	R	—

4. Arithmetic Instructions (16-bit)

			Addressing										Flag					
													7	6	4	2	1	0
Operation Name	Mnemonics	Op Code	IMMED	EXT	INDX	REG	REGI	IMP	REL	Bytes	States	Operation	S	Z	H	P/V	N	C
ADD	ADD HL, ww	00 ww1 001				S		D		1	7	$HL_R + ww_R \rightarrow HL_R$	—	—	X	—	R	↕
	ADD IX, xx	11 011 101 00 xx1 001				S		D		2	10	$IX_R + xx_R \rightarrow IX_R$	—	—	X	—	R	↕
	ADD IY, yy	11 111 101 00 yy1 001				S		D		2	10	$IY_R + yy_R \rightarrow IY_R$	—	—	X	—	R	↕
ADC	ADC HL, ww	11 101 101 01 ww1 010				S		D		2	10	$HL_R + ww_R + c \rightarrow HL_R$	↕	↕	X	V	R	↕
DEC	DEC ww	00 ww1 011				S/D				1	4	$ww_R - 1 \rightarrow ww_R$	—	—	—	—	—	—
	DEC IX	11 011 101 00 101 011						S/D		2	7	$IX_R - 1 \rightarrow IX_R$	—	—	—	—	—	—
	DEC IY	11 111 101 00 101 011						S/D		2	7	$IY_R - 1 \rightarrow IY_R$	—	—	—	—	—	—
INC	INC ww	00 ww0 011				S/D				1	4	$ww_R + 1 \rightarrow ww_R$	—	—	—	—	—	—
	INC IX	11 011 101 00 100 011						S/D		2	7	$IX_R + 1 \rightarrow IX_R$	—	—	—	—	—	—
	INC IY	11 111 101 00 100 011						S/D		2	7	$IY_R + 1 \rightarrow IY_R$	—	—	—	—	—	—
SBC	SBC HL, ww	11 101 101 01 ww0 010				S		D		2	10	$HL_R - ww_R - c \rightarrow HL_R$	↕	↕	X	V	S	↕

3

• Data Transfer Instructions

1. 8-Bit Load

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	INDX	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
Load 8-bit data	LD A, I	11 101 101 01 010 111						S/D		2	6	Ir → Ar	↕	↕	R	IEF_2	R	—
	LD A, R	11 101 101 01 011 111						S/D		2	6	Rr → Ar	↕	↕	R	IEF_2	R	—
	LD A, (BC)	00 001 010					S	D		1	6	$(BC)_M$ → Ar(Note)	—	—	—	—	—	—
	LD A, (DE)	00 011 010					S	D		1	6	$(DE)_M$ → Ar	—	—	—	—	—	—
	LD A, (mn)	00 111 010 <n> <m>		S				D		3	12	$(mn)_M$ → Ar	—	—	—	—	—	—
	LD I, A	11 101 101 01 000 111						S/D		2	6	Ar → Ir	—	—	—	—	—	—
	LD R, A	11 101 101 01 001 111						S/D		2	6	Ar → Rr	—	—	—	—	—	—
	LD (BC), A	00 000 010					D	S		1	7	Ar → $(BC)_M$	—	—	—	—	—	—
	LD (DE), A	00 010 010					D	S		1	7	Ar → $(DE)_M$	—	—	—	—	—	—
	LD (mn), A	00 110 010 <n> <m>		D				S		3	13	Ar → $(mn)_M$	—	—	—	—	—	—
	LD g, g'	01 g g'				S/D				1	4	gr' → gr	—	—	—	—	—	—
	LD g, (HL)	01 g 110				D	S			1	6	$(HL)_M$ → gr	—	—	—	—	—	—
	LD g, m	00 g 110 <m>	S			D				2	6	m → gr	—	—	—	—	—	—
	LD g, (IX + d)	11 011 101 01 g 110 <d>			S	D				3	14	$(IX + d)_M$ → gr	—	—	—	—	—	—
	LD g, (IY + d)	11 111 101 01 g 110 <d>			S	D				3	14	$(IY + d)_M$ → gr	—	—	—	—	—	—
	LD (HL), m	00 110 110 <m>	S				D			2	9	m → $(HL)_M$	—	—	—	—	—	—
	LD (IX + d), m	11 011 101 00 110 110 <d> <m>	S		D					4	15	m → $(IX + d)_M$	—	—	—	—	—	—
	LD (IY + d), m	11 111 101 00 110 110 <d> <m>	S		D					4	15	m → $(IY + d)_M$	—	—	—	—	—	—

Note: Interrupts are not detected at the end of the LD A, I and LD A, R instructions.

1. 8-Bit Load (cont)

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	INDX	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
Load 8-bit data	LD (HL), g	01 110 g				S	D			1	7	$gr \rightarrow (HL)_M$	—	—	—	—	—	—
	LD (IX + d), g	11 011 101 01 110 g <d>			D	S				3	15	$gr \rightarrow (IX + d)_M$	—	—	—	—	—	—
	LD (IY + d), g	11 111 101 01 110 g <d>			D	S				3	15	$gr \rightarrow (IY + d)_M$	—	—	—	—	—	—

2. 16-Bit Load

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	INDX	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
Load 16-bit data	LD ww, mn	00 ww0 001 <n> <m>	S			D				3	9	$mn \rightarrow ww_R$	—	—	—	—	—	—
	LD IX, mn	11 011 101 00 100 001 <n> <m>	S					D		4	12	$mn \rightarrow IX_R$	—	—	—	—	—	—
	LD IY, mn	11 111 101 00 100 001 <n> <m>	S					D		4	12	$mn \rightarrow IY_R$	—	—	—	—	—	—
	LD SP, HL	11 111 001						S/D		1	4	$HL_R \rightarrow SP_R$	—	—	—	—	—	—
	LD SP, IX	11 011 101 11 111 001						S/D		2	7	$IX_R \rightarrow SP_R$	—	—	—	—	—	—
	LD SP, IY	11 111 101 11 111 001						S/D		2	7	$IY_R \rightarrow SP_R$	—	—	—	—	—	—
	LD ww, (mn)	11 101 101 01 ww1 011 <n> <m>		S		D				4	18	$(mn + 1)_M \rightarrow wwHr$	—	—	—	—	—	—
	LD HL, (mn)	00 101 010 <n> <m>		S				D		3	15	$(mn + 1)_M \rightarrow Hr$ $(mn)_M \rightarrow Lr$	—	—	—	—	—	—

2. 16-Bit Load (cont)

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	INDX	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
Load 16-bit data	LD IX, (mn)	11 011 101 00 101 010 <n> <m>		S				D		4	18	$(mn+1)_M$ → IXHr $(mn)_M$ → IXLr	—	—	—	—	—	—
	LD IY, (mn)	11 111 101 00 101 010 <n> <m>		S				D		4	18	$(mn+1)_M$ → IYHr $(mn)_M$ → IYLr	—	—	—	—	—	—
	LD (mn), ww	11 101 101 01 ww0 011 <n> <m>		D		S				4	19	wwHr → $(mn+1)_M$ wwLr → $(mn)_M$	—	—	—	—	—	—
	LD (mn), HL	00 100 010 <n> <m>		D				S		3	16	Hr → $(mn+1)_M$ Lr → $(mn)_M$	—	—	—	—	—	—
	LD (mn), IX	11 011 101 00 100 010 <n> <m>		D				S		4	19	IXHr → $(mn+1)_M$ IXLr → $(mn)_M$	—	—	—	—	—	—
	LD (mn), IY	11 111 101 00 100 010 <n> <m>		D				S		4	19	IYHr → $(mn+1)_M$ IYLr → $(mn)_M$	—	—	—	—	—	—

3. Block Transfer

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
													7	6	4	2	1	0
			IMMED	EXT	INDX	REG	REGI	IMP	REL				S	Z	H	P/V	N	C
														*2		*1		
Block transfer search data	CPD	11 101 101 10 101 001					S	S		2	12	$Ar - (HL)_M$ $BC_R - 1 \rightarrow BC_R$ $HL_R - 1 \rightarrow HL_R$	↕	↕	↕	↕	S	—
														*2		*1		
	CPDR	11 101 101 10 111 001					S	S		2	14 12	$BC_R \neq 0$ $Ar \neq (HL)_M$ $BC_R = 0$ or $Ar = (HL)_M$	↕	↕	↕	↕	S	—
												Q { $Ar - (HL)_R$ $BC_R - 1 \rightarrow BC_R$ $HL_R - 1 \rightarrow HL_R$						
														*2		*1		
												Repeat Q until $Ar = (HL)_M$ or $BC_R = 0$	↕	↕	↕	↕	S	—
	CPI	11 101 101 10 100 001					S	S		2	12	$Ar - (HL)_M$ $BC_R - 1 \rightarrow BC_R$ $HL_R + 1 \rightarrow HL_R$	↕	↕	↕	↕	S	—
														*2		*1		
	CPIR	11 101 101 10 110 001					S	S		2	14 12	$BC_R \neq 0$ $Ar \neq (HL)_M$ $BC_R = 0$ or $Ar = (HL)_M$ Q { $Ar - (HL)_M$ $BC_R - 1 \rightarrow BC_R$ $HL_R + 1 \rightarrow HL_R$ Repeat Q until $Ar = (HL)_M$ or $BC_R = 0$	↕	↕	↕	↕	S	—
																*1		
	LDD	11 101 101 10 101 000					S/D			2	12	$(HL)_M \rightarrow (DE)_M$ $BC_R - 1 \rightarrow BC_R$ $DE_R - 1 \rightarrow DE_R$ $HL_R - 1 \rightarrow HL_R$	—	—	R	↕	R	—
	LDDR	11 101 101 10 111 000					S/D			2	14 ($BC_R \neq 0$) 12 ($BC_R = 0$)	Q { $(HL)_M \rightarrow (DE)_M$ $BC_R - 1 \rightarrow BC_R$ $DE_R - 1 \rightarrow DE_R$ $HL_R - 1 \rightarrow HL_R$ Repeat Q until $BC_R = 0$	—	—	R	R	R	—
																*1		
	LDI	11 101 101 10 100 000					S/D			2	12	$(HL)_M \rightarrow (DE)_M$ $BC_R - 1 \rightarrow BC_R$ $DE_R + 1 \rightarrow DE_R$ $HL_R + 1 \rightarrow HL_R$	—	—	R	↕	R	—
	LDIR	11 101 101 10 110 000					S/D			2	14 ($BC_R \neq 0$) 12 ($BC_R = 0$)	Q { $(HL)_M \rightarrow (DE)_M$ $BC_R - 1 \rightarrow BC_R$ $DE_R + 1 \rightarrow DE_R$ $HL_R + 1 \rightarrow HL_R$ Repeat Q until $BC_R = 0$	—	—	R	R	R	—

Notes: 1. P/V = 0: $BC_R - 1 = 0$
P/V = 1: $BC_R - 1 \neq 0$
2. Z = 1: $Ar = (HL)_M$
Z = 0: $Ar \neq (HL)_M$

4. Stack and Exchange

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	INDX	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
PUSH	PUSH zz	11 zz0 101				S		D		1	11	$zzLr \rightarrow (SP-2)_M$ $zzHr \rightarrow (SP-1)_M$ $SP_R - 2 \rightarrow SP_R$	—	—	—	—	—	—
	PUSH IX	11 011 101 11 100 101						S/D		2	14	$IXLr \rightarrow (SP-2)_M$ $IXHr \rightarrow (SP-1)_M$ $SP_R - 2 \rightarrow SP_R$	—	—	—	—	—	—
	PUSH IY	11 111 101 11 100 101						S/D		2	14	$IYLr \rightarrow (SP-2)_M$ $IYHr \rightarrow (SP-1)_M$ $SP_R - 2 \rightarrow SP_R$	—	—	—	—	—	—
POP	POP zz	11 zz0 001				D		S		1	9	$(SP+1)_M \rightarrow zzHr$(Note) $(SP)_M \rightarrow zzLr$ $SP_R + 2 \rightarrow SP_R$	—	—	—	—	—	—
	POP IX	11 011 101 11 100 001						S/D		2	12	$(SP+1)_M \rightarrow IXHr$ $(SP)_M \rightarrow IXr$ $SP_R + 2 \rightarrow SP_R$	—	—	—	—	—	—
	POP IY	11 111 101 11 100 001						S/D		2	12	$(SP+1)_M \rightarrow IYHr$ $(SP)_M \rightarrow IYLr$ $SP_R + 2 \rightarrow SP_R$	—	—	—	—	—	—
Exchange	EX AF, AF'	00 001 000						S/D		1	4	$AF_R \rightarrow AF_R'$	—	—	—	—	—	—
	EX DE, HL	11 101 011						S/D		1	3	$DE_R - HL_R$	—	—	—	—	—	—
	EXX	11 011 001						S/D		1	3	$BC_R - BC_R'$ $DE_R - DE_R'$ $HL_R - HL_R'$	— —	— —	— —	— —	— —	— —
	EX (SP), HL	11 100 011						S/D		1	16	$Hr - (SP+1)_M$ $Lr - (SP)_M$	—	—	—	—	—	—
	EX (SP), IX	11 011 101 11 100 011						S/D		2	19	$IXHr - (SP+1)_M$ $IXLr - (SP)_M$	—	—	—	—	—	—
	EX (SP), IY	11 111 101 11 100 011						S/D		2	19	$IYHr - (SP+1)_M$ $IYLr - (SP)_M$	—	—	—	—	—	—

Note: POP AF writes stack contents to flag.

• Program Control Instructions

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
			IMMED	EXT	INDX	REG	REGI	IMP	REL				7	6	4	2	1	0
													S	Z	H	P/V	N	C
Call	CALL mn	11 001 101 <n> <m>		D·						3	16	$PCHr \rightarrow (SP-1)_M$ $PCLr \rightarrow (SP-2)_M$ $mn \rightarrow PC_R$ $SP_R - 2 \rightarrow SP_R$	—	—	—	—	—	—
	CALL f, mn	11 f 100 <n> <m>		D						3	6 (f: false) 16 (f: true)	continue: f is false CALL mn: f is true	—	—	—	—	—	—
Jump	DJNZ j	00 010 000 <j – 2>							D	2 2	9 (Br ≠ 0) 7 (Br = 0)	Br – 1 → Br continue: Br = 0 $PC_R + j \rightarrow PC_R$: Br ≠ 0	—	—	—	—	—	—
	JP f, mn	11 f 010 <n> <m>		D						3 3	6 (f: false) 9 (f: true)	$mn \rightarrow PC_R$: f is true continue: f is false	—	—	—	—	—	—
	JP mn	11 000 011 <n> <m>		D						3	9	$mn \rightarrow PC_R$	—	—	—	—	—	—
	JP (HL)	11 101 001					D			1	3	$HL_R \rightarrow PC_R$	—	—	—	—	—	—
	JP (IX)	11 011 101 11 101 001					D			2	6	$IX_R \rightarrow PC_R$	—	—	—	—	—	—
	JP (IY)	11 111 101 11 101 001					D			2	6	$IY_R \rightarrow PC_R$	—	—	—	—	—	—
	JR j	00 011 000 <j– 2>							D	2	8	$PC_R + j \rightarrow PCv$	—	—	—	—	—	—
	JR C, j	00 111 000 <j– 2>							D	2 2	6 8	continue: C = 0 $PC_R + j \rightarrow PC_R$: C = 1	—	—	—	—	—	—
	JR NC, j	00 110 000 <j-2>							D	2 2	6 8	continue: C = 1 $PC_R + j \rightarrow PC_R$: C = 0	—	—	—	—	—	—
	JR Z, j	00 101 000 <j-2>							D	2 2	6 8	continue: Z = 0 $PC_R + j \rightarrow PC_R$: Z = 1	—	—	—	—	—	—
	JR NZ, j	00 100 000 <j-2>							D	2 2	6 8	continue: Z = 1 $PC_R + j \rightarrow PC_R$: Z = 0	—	—	—	—	—	—
Return	RET	11 001 001						D		1	9	$(SP)_M \rightarrow PCLr$ $(SP+1)_M \rightarrow PCHr$ $SP_R + 2 \rightarrow SP_R$	—	—	—	—	—	—
	RET f	11 f 000							D	1 1	5 (f: false) 10 (f: true)	continue: f is false RET: f is true	—	—	—	—	—	—
	RETI	11 101 101 01 001 101						D		2	22 (Z) 12 (R1)	$(SP)_M \rightarrow PCLr$ $(SP+1)_M \rightarrow PCHr$ $SP_R + 2 \rightarrow SP_R$	—	—	—	—	—	—

• Program Control Instructions (cont)

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
													7	6	4	2	1	0
			IMMED	EXT	INDX	REG	REGI	IMP	REL				S	Z	H	P/V	N	C
Return	RETN	11 101 101 01 000 101						D		2	12	$(SP)_M \rightarrow PCLr$ $(SP+1)_M \rightarrow PCHr$ $SP_R+2 \rightarrow SP_R$ IEF2 → IEF1	—	—	—	—	—	—
Restart	RST v	11 v 111						D		1	11	$PCHr \rightarrow (SP-1)_M$ $PCLr \rightarrow (SP-2)_M$ 0 → PCHr v → PCLr $SP_R-2 \rightarrow SP_R$	—	—	—	—	—	—

• I/O Instructions

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
													7	6	4	2	1	0
			IMMED	EXT	INDX	REG	REGI	IMP	REL				S	Z	H	P/V	N	C
Input	IN A, (m)	11 011 011 <m>						D	S	2	9	$(Am)_I \rightarrow Ar$ $m \rightarrow A_0$ to A_7 $Ar \rightarrow A_8$ to A_{15}	—	—	—	—	—	—
	IN g, (C)	11 101 101 01 g 000				D			S	2	9	$(BC)_I \rightarrow gr$ g = 110: only the flags will change. $Cr \rightarrow A_0$ to A_7 $Br \rightarrow A_8$ to A_{15}	↕	↕	R	P	R	—
	IN0 g, (m)	11 101 101 00 g 000 <m>				D			S	3	12	$(00m)_I \rightarrow gr$ g = 110: only the flags will change. $m \rightarrow A_0$ to A_7 $00 \rightarrow A_8$ to A_{15}	↕	↕	R	P	R	—
	IND	11 101 101 10 101 010					D		S	2	12	$(BC)_I \rightarrow (HL)M$ $HL_R-1 \rightarrow HL_R$ Br − 1 → Br $Cr \rightarrow A_0$ to A_7 $Br \rightarrow A_8$ to A_{15}	X	*1 ↕	X	X	*2 ↕ *2	X

Notes: 1. Z = 1: Br − 1 = 0
Z = 0: Br − 1 ≠ 0
2. N = 1: MSB of data = 1
N = 0: MSB of data = 0

• I/O Instructions (cont)

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
													7	6	4	2	1	0
			IMMED	EXT	INDX	REG	REGI	IMP	REL				S	Z	H	P/V	N	C
Input	INDR	11 101 101 10 111 010					D		S	2	14 (Br ≠ 0) 12 (Br = 0)	Q { $(BC)_I \rightarrow (HL)_M$ $HL_R - 1 \rightarrow HL_R$ $Br - 1 \rightarrow Br$ } Repeat Q until Br = 0 $Cr \rightarrow A_0$ to A_7 $Br \rightarrow A_6$ to A_{15}	X	S *1	X	X	↕ *2	X
	INI	11 101 101 10 100 010					D		S	2	12	$(BC)_I \rightarrow (HL)_M$ $HL_R + 1 \rightarrow HL_R$ $Br - 1 \rightarrow Br$ $Cr \rightarrow A_0$ to A_7 $Br \rightarrow A_6$ to A_{15}	X	↕	X	X	↕ *2	X
	INIR	11 101 101 10 110 010					D		S	2	14 (Br ≠ 0) 12 (Br = 0)	Q { $(BC)_I \rightarrow (HL)_M$ $HL_R + 1 \rightarrow HL_R$ $Br - 1 \rightarrow Br$ } Repeat Q until Br = 0 $Cr \rightarrow A_0$ to A_7 $Br \rightarrow A_6$ to A_{15}	X	S	X	X	↕	X
Output	OUT (m), A	11 010 011 <m>						S	D	2	10	$Ar \rightarrow (Am)_I$ $m \rightarrow A_0$ to A_7 $Ar \rightarrow A_6$ to A_{15}	—	—	—	—	—	—
	OUT (C), g	11 101 101 01 g 001				S			D	2	10	$gr \rightarrow (BC)_I$ $Cr \rightarrow A_0$ to A_7 $Br \rightarrow A_6$ to A_{15}	—	—	—	—	—	—
	OUT0 (m), g	11 101 101 00 g 001 <m>				S			D	3	13	$gr \rightarrow (00m)_I$ $m \rightarrow A_0$ to A_7 $00 \rightarrow A_6$ to A_{15}	—	— *1	—	—	— *2	—
	OTDM	11 101 101 10 001 011					S		D	2	14	$(HL)_M \rightarrow (00C)_I$ $HL_R - 1 \rightarrow HL_R$ $Cr - 1 \rightarrow Cr$ $Br - 1 \rightarrow Br$ $Cr \rightarrow A_0$ to A_7 $00 \rightarrow A_6$ to A_{15}	↕	↕	↕	P	↕ *2	↕
	OTDMR	11 101 101 10 011 011					S		D	2	16 (Br ≠ 0) 14 (Br = 0)	Q { $(HL)_M \rightarrow (00C)_I$ $HL_R - 1 \rightarrow HL_R$ $Cr - 1 \rightarrow Cr$ $Br - 1 \rightarrow Br$ } Repeat Q until Br = 0 $Cr \rightarrow A_0$ to A_7 $00 \rightarrow A_6$ to A_{15}	R	S	R	S	↕ *2	R

Notes: 1. Z = 1: Br − 1 = 0
Z = 0: Br − 1 ≠ 0
2. N = 1: MSB of data = 1
N = 0: MSB of data = 0

• I/O Instructions (cont)

			Addressing										Flag					
													7	6	4	2	1	0
Operation Name	Mnemonics	Op Code	IMMED	EXT	INDX	REG	REGI	IMP	REL	Bytes	States	Operation	S	Z	H	P/V	N	C
Output	OTDR	11 101 101 10 111 011					S		D	2	14 (Br ≠ 0) 12 (Br = 0)	$Q\left\{\begin{array}{l}(HL)_M \rightarrow (BC)_I\\ HL_R - 1 \rightarrow HL_R\\ Br - 1 \rightarrow Br\end{array}\right.$ Repeat Q until Br = 0 $Cr \rightarrow A_0$ to A_7 $Br \rightarrow A_8$ to A_{15}	X	S *1	X	X	↕ *2	X
	OUTI	11 101 101 10 100 011					S		D	2	12	$(HL)_M \rightarrow (BC)_I$ $HL_R + 1 \rightarrow HL_R$ $Br - 1 \rightarrow Br$ $Cr \rightarrow A_0$ to A_7 $Br \rightarrow A_8$ to A_{15}	X	↕	X	X	↕ *2	X
	OTIR	11 101 101 10 110 011					S		D	2	14 (Br ≠ 0) 12 (Br = 0)	$Q\left\{\begin{array}{l}(HL)_M \rightarrow (BC)_I\\ HL_R + 1 \rightarrow HL_R\\ Br - 1 \rightarrow Br\end{array}\right.$ Repeat Q until Br = 0 $Cr \rightarrow A_0$ to A_7 $Br \rightarrow A_8$ to A_{15}	X	S	X	X	↕	X
	TSTIO m	11 101 101 01 110 100 <m>	S						S	3	12	$(00C)_I \cdot m$ $Cr \rightarrow A_0$ to A_7 $00 \rightarrow A_8$ to A_{15}	↕	↕ *1	S	P	R *2	R
	OTIM	11 101 101 10 000 011					S		D	2	14	$(HL)_M \rightarrow (00C)_I$ $HL_R + 1 \rightarrow HL_R$ $Cr + 1 \rightarrow Cr$ $Br - 1 \rightarrow Br$ $Cr \rightarrow A_0$ to A_7 $00 \rightarrow A_8$ to A_{15}	↕	↕	↕	P	↕ *2	↕
	OTIMR	11 101 101 10 010 011					S		D	2	16 (Br ≠ 0) 14 (Br = 0)	$Q\left\{\begin{array}{l}(HL)_M \rightarrow (00C)_I\\ HL_R + 1 \rightarrow HL_R\\ Cr + 1 \rightarrow Cr\\ Br - 1 \rightarrow Br\end{array}\right.$ Repeat Q until Br = 0 $Cr \rightarrow A_0$ to A_7 $00 \rightarrow A_8$ to A_{15}	R	S *1	R	S	↕ *2	R
	OUTD	11 101 101 10 101 011					S		D	2	12	$(HL)_M \rightarrow (BC)_I$ $HL_R - 1 \rightarrow HL_R$ $Br - 1 \rightarrow Br$ $Cr \rightarrow A_0$ to A_7 $Br \rightarrow A_8$ to A_{15}	X	↕	X	X	↕	X

Notes: 1. Z = 1: Br − 1 = 0
Z = 0: Br − 1 ≠ 0
2. N = 1: MSB of data = 1
N = 0: MSB of data = 0

• Special Control Instructions

Operation Name	Mnemonics	Op Code	Addressing							Bytes	States	Operation	Flag					
													7	6	4	2	1	0
			MMED	EXT	NDX	REG	REGI	MP	REL				S	Z	H	P/V	N	C
Special function	DAA	00 100 111						S/D		1	4	Decimal adjust accumulator	↕	↕	↕	P	—	↕
Carry control	CCF	00 111 111								1	3	$\bar{C} \rightarrow C$	—	—	R	—	R	↕
	SCF	00 110 111								1	3	$1 \rightarrow C$	—	—	R	—	R	S
CPU control	DI	11 110 011								1	3	$0 \rightarrow IEF_1, 0 \rightarrow IEF_2$ (Note)	—	—	—	—	—	—
	EI	11 111 011								1	3	$1 \rightarrow IEF_1, 1 \rightarrow IEF_2$ (Note)	—	—	—	—	—	—
	HALT	01 110 110								1	3	CPU halted	—	—	—	—	—	—
	IM 0	11 101 101 01 000 110								2	6	Interrupt mode 0	—	—	—	—	—	—
	IM 1	11 101 101 01 010 110								2	6	Interrupt mode 1	—	—	—	—	—	—
	IM 2	11 101 101 01 011 110								2	6	Interrupt mode 2	—	—	—	—	—	—
	NOP	00 000 000								1	3	No operation	—	—	—	—	—	—
	SLP	11 101 101 01 110 110								2	8	Sleep	—	—	—	—	—	—

Note: Interrupts are not detected at the end of the DI or EI instruction.

■ Alphabetical Instruction List

Mnemonics	Bytes	Machine Cycles	States
ADC A, m	2	2	6
ADC A, g	1	2	4
ADC A, (HL)	1	2	6
ADC A, (IX + d)	3	6	14
ADC A, (IY + d)	3	6	14
ADC HL, ww	2	6	10
ADD A, m	2	2	6
ADD A, g	1	2	4
ADD A, (HL)	1	2	6
ADD A, (IX + d)	3	6	14
ADD A, (IY + d)	3	6	14
ADD HL, ww	1	5	7
ADD IX, xx	2	6	10
ADD IY, yy	2	6	10
AND m	2	2	6
AND g	1	2	4
AND (HL)	1	2	6
AND (IX + d)	3	6	14
AND (IY + d)	3	6	14
BIT b, (HL)	2	3	9
BIT b, (IX + d)	4	5	15
BIT b, (IY + d)	4	5	15
BIT b, g	2	2	6
CALL f, mn	3	2	6 (If condition is false)
	3	6	16 (If condition is true)
CALL mn	3	6	16
CCF	1	1	3
CPD	2	6	12
CPDR	2	8	14 (If $BC_R \neq 0$ and $Ar \neq (HL)_M$)
	2	6	12 (If $BC_R = 0$ or $Ar = (HL)_M$)

Mnemonics	Bytes	Machine Cycles	States
CP (HL)	1	2	6
CPI	2	6	12
CPIR	2 2	8 6	14 (If $BC_R \neq 0$ and $Ar \neq (HL)_M$) 12 (If $BC_R = 0$ or $Ar = (HL)_M$)
CP (IX + d)	3	6	14
CP (IY + d)	3	6	14
CPL	1	1	3
CP m	2	2	6
CP g	1	2	4
DAA	1	2	4
DEC (HL)	1	4	10
DEC IX	2	3	7
DEC IY	2	3	7
DEC (IX + d)	3	8	18
DEC (IY + d)	3	8	18
DEC g	1	2	4
DEC ww	1	2	4
DI	1	1	3
DJNZ j	2 2	5 3	9 (If $Br \neq 0$) 7 (If $Br = 0$)
EI	1	1	3
EX AF, AF'	1	2	4
EX DE, HL	1	1	3
EX (SP), HL	1	6	16
EX (SP), IX	2	7	19
EX (SP), IY	2	7	19
EXX	1	1	3
HALT	1	1	3
IM 0	2	2	6
IM 1	2	2	6
IM 2	2	2	6

Mnemonics	Bytes	Machine Cycles	States
INC g	1	2	4
INC (HL)	1	4	10
INC (IX + d)	3	8	18
INC (IY + d)	3	8	18
INC ww	1	2	4
INC IX	2	3	7
INC IY	2	3	7
IN A, (m)	2	3	9
IN g, (C)	2	3	9
INI	2	4	12
INIR	2	6	14 (If Br ≠ 0)
INIR	2	4	12 (If Br = 0)
IND	2	4	12
INDR	2	6	14 (If Br ≠ 0)
	2	4	12 (If Br = 0)
IN0 g, (m)	3	4	12
JP f, mn	3	2	6 (If f is false)
	3	3	9 (If f is true)
JP (HL)	1	1	3
JP (IX)	2	2	6
JP (IY)	2	2	6
JP mm	3	3	9
JR j	2	4	8
JR C, j	2	2	6 (If condition is false)
	2	4	8 (If condition is true)
JR NC, j	2	2	6 (If condition is false)
	2	4	8 (If condition is true)
JR Z, j	2	2	6 (If condition is false)
	2	4	8 (If condition is true)
JR NZ, j	2	2	6 (If condition is false)
	2	4	8 (If condition is true)
LD A, (BC)	1	2	6
LD A, (DE)	1	2	6

Mnemonics	Bytes	Machine Cycles	States
LD A, I	2	2	6
LD A, (mm)	3	4	12
LD A, R	2	2	6
LD (BC), A	1	3	7
LDD	2	4	12
LD (DE), A	1	3	7
LD ww, mn	3	3	9
LD ww, (mn)	4	6	18
LDDR	2 2	6 4	14 (If $BC_R \neq 0$) 12 (If $BC_R = 0$)
LD (HL), m	2	3	9
LD HL, (mn)	3	5	15
LD (HL), g	1	3	7
LDI	2	4	12
LDI, A	2	2	6
LDIR	2 2	6 4	14 (If $BC_R \neq 0$) 12 (If $BC_R = 0$)
LD IX, mn	4	4	12
LD IX, (mn)	4	6	18
LD (IX + d), m	4	5	15
LD (IX + d), g	3	7	15
LD IY, mn	4	4	12
LD IY, (mn)	4	6	18
LD (IY + d), m	4	5	15
LD (IY + d), g	3	7	15
LD (mn), A	3	5	13
LD (mn), ww	4	7	19
LD (mn), HL	3	6	16
LD (mn), IX	4	7	19
LD (mn), IY	4	7	19
LD R, A	2	2	6
LD g, (HL)	1	2	6

Mnemonics	Bytes	Machine Cycles	States
LD g, (IX + d)	3	6	14
LD g, (IY + d)	3	6	14
LD g, m	2	2	6
LD g, g'	1	2	4
LD SP, HL	1	2	4
LD SP, IX	2	3	7
LD SP, IY	2	3	7
MLT ww	2	13	17
NEG	2	2	6
NOP	1	1	3
OR (HL)	1	2	6
OR (IX + d)	3	6	14
OR (IY + d)	3	6	14
OR m	2	2	6
OR g	1	2	4
OTDM	2	6	14
OTDMR	2	8	16 (If Br ≠ 0)
	2	6	14 (If Br = 0)
OTDR	2	6	14 (If Br ≠ 0)
	2	4	12 (If Br = 0)
OTIM	2	6	14
OTIMR	2	8	16 (If Br ≠ 0)
	2	6	14 (If Br = 0)
OTIR	2	6	14 (If Br ≠ 0)
	2	4	12 (If Br = 0)
OUTD	2	4	12
OUTI	2	4	12
OUT (m), A	2	4	10
OUT (C), g	2	4	10
OUT0 (m), g	3	5	13
POP IX	2	4	12
POP IY	2	4	12

Mnemonics	Bytes	Machine Cycles	States
POP zz	1	3	9
PUSH IX	2	6	14
PUSH IY	2	6	14
PUSH zz	1	5	11
RES b, (HL)	2	5	13
RES b, (IX + d)	4	7	19
RES b, (IY + d)	4	7	19
RES b, g	2	3	7
RET	1	3	9
RET f	1 1	3 4	5 (If condition is false) 10 (If condition is true)
RETI	2	10 (Z) 4 (R1)	22 (Z) 12 (R1)
RETN	2	4	12
RLA	1	1	3
RLCA	1	1	3
RLC (HL)	2	5	13
RLC (IX + d)	4	7	19
RLC (IY + d)	4	7	19
RLC g	2	3	7
RLD	2	8	16
RL (HL)	2	5	13
RL (IX + d)	4	7	19
RL (IY + d)	4	7	19
RL g	2	3	7
RRA	1	1	3
RRCA	1	1	3
RRC (HL)	2	5	13
RRC (IX + d)	4	7	19
RRC (IY + d)	4	7	19
RRC g	2	3	7

Mnemonics	Bytes	Machine Cycles	States
RRD	2	8	16
RR (HL)	2	5	13
RR (IX + d)	4	7	19
RR (IY + d)	4	7	19
RR g	2	3	7
RST v	1	5	11
SBC A, (HL)	1	2	6
SBC A, (IX + d)	3	6	14
SBC A, (IY + d)	3	6	14
SBC A, m	2	2	6
SBC A, g	1	2	4
SBC HL, ww	2	6	10
SCF	1	1	3
SET b, (HL)	2	5	13
SET b, (IX + d)	4	7	19
SET b, (IY + d)	4	7	19
SET b, g	2	3	7
SLA (HL)	2	5	13
SLA (IX + d)	4	7	19
SLA (IY + d)	4	7	19
SLA g	2	3	7
SLP	2	2	8
SRA (HL)	2	5	13
SRA (IX + d)	4	7	19
SRA (IY + d)	4	7	19
SRA g	2	3	7
SRL (HL)	2	5	13
SRL (IX + d)	4	7	19
SRL (IY + d)	4	7	19
SRL g	2	3	7
SUB (HL)	1	2	6

Mnemonics	Bytes	Machine Cycles	States
SUB (IX + d)	3	6	14
SUB (IY + d)	3	6	14
SUB m	2	2	6
SUB g	1	2	4
TSTIO m	3	4	12
TST g	2	3	7
TST m	3	3	9
TST (HL)	2	4	10
XOR (HL)	1	2	6
XOR (IX + d)	3	6	14
XOR (IY + d)	3	6	14
XOR m	2	2	6
XOR g	1	2	4

■ Op Code Map

Table 3 Op Code Map (1)

First op code
Instruction format: XX

			ww (LO = ALL)												LO = 0 to 7				
			BC	DE	HL	SP									BC	DE	HL	AF	zz
			g (LO = 0 to 7)												NZ	NC	PO	P	f
			B	D	H	(HL)	B	D	H	(HL)					00H	10H	20H	30H	v
		HI	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
s (HI = ALL)	LO		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
B	0000	0	NOP	DJNZ, j	JR NZ, j	JR NC, j	LD g, s			* 1	ADD A , s	SUB s	AND s	OR s	RET f				0
C	0001	1	LD ww, mn												POP zz				1
D	0010	2	LD (ww), A		LD (mn) , HL	LD (mn) , A									JP f, mn				2
E	0011	3	INC ww												JP mn	OUT (m) ,A	EX (SP) , HL	DI	3
H	0100	4	INC g			* 1									CALL f, mn				4
L	0101	5	DEC g			* 1									PUSH zz				5
(HL)	0110	6	LD g, m			* 1	Note 2			HALT					ADD A, m	SUB m	AND m	OR m	6
A	0111	7	RLCA	RLA	DAA	SCF					* 2	* 2	* 2	* 2	RST v				7
B	1000	8	EXAF, AF	JR j	JR Z, j	JR C, j	LD g, s				ADC A , s	SBC A , s	XOR s	CP s	RET f				8
C	1001	9	ADD HL, ww												RET	EXX	JP (HL)	LD SP, HL	9
D	1010	A	LD A, (ww)		LD HL, (mn)	LD A, (mn)									JP f, mn				A
E	1011	B	DEC ww												Table 2	IN A, (m)	EXDE, HL	EI	B
H	1100	C	INC g												CALL f, mn				C
L	1101	D	DEC g												CALL mn	* 3	Table 3	* 3	D
(HL)	1110	E	LD g, m				* 2				* 2	* 2	* 2	* 2	ADC A, m	SBC A, m	XOR m	CP m	E
A	1111	F	RRCA	RRA	CPL	CCF									RST v				F
			0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
			C	E	L	A	C	E	L	A					Z	C	PE	M	f
			g (LO = 8 to F)												08H	18H	28H	38H	v
															LO = 8 to F				

Notes: 1. g is replaced by (HL).

2. s is replaced by (HL).

3. Appending DD to the beginning of an op code (DD XX) in an instruction that has HL or (HL) in its operand produces the same operation as the original format with the following replacements:

HL replaced by IX

(HL) replaced by (IX + d)

Example:

22H ; LD (mn), HL → DDH 22H ; LD (mn), IX

Similarly, appending FD to the beginning of an op code (FD XX) in an instruction that has HL or (HL) in its operand produces the same operation as the original format with the following replacements:

HL replaced by IY

(HL) replaced by (IY + d)

Example:

34H ; INC (HL) → FDH 34H ; INC (IY + d)

An exception is the JP (HL) instruction (E9H). Appending DDH or FDH to the beginning replaces (HL) with (IX) or (IY).

If DDH or FDH is appended to the EX DE, HL instruction (EBH), an undefined instruction results, without replacement of HL.

Table 4 Op Code Map (2)

Second op code
Instruction format: CB XX

									b (LO = 0 to 7)											
									0	2	4	6	0	2	4	6	0	2	4	6	
		LO	HI	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
				0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
g (HI = ALL)	B	0000	0	RLC g	RL g	SLA g		BIT b, g				RES b, g				SET b, g				0
	C	0001	1																	1
	D	0010	2																	2
	E	0011	3																	3
	H	0100	4																	4
	L	0101	5																	5
	(HL)	0110	6	* 1	* 1	* 1		* 1				* 1				* 1				6
	A	0111	7																	7
	B	1000	8	RRC g	RR g	SRA g	SRL g	BIT b, g				RES b, g				SET b, g				8
	C	1001	9																	9
	D	1010	A																	A
	E	1011	B																	B
	H	1100	C																	C
	L	1101	D																	D
	(HL)	1110	E	Note	Note	Note	Note	Note				Note				Note				E
	A	1111	F																	F
				0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
								1	3	5	7	1	3	5	7	1	3	5	7	
								b (LO = 8 to F)												

Note: If DDH is appended to the beginning of the op code, the instruction is executed by the op code DD CB d XX, replacing (HL) with (IX + d). Similarly, if FDH is appended to the beginning of the op code, the instruction is executed by the op code FD CB d XX, replacing (HL) with (IY + d).

Table 5 Op Code Map (3)

Second op code
Instruction format: ED XX

						ww (LO = ALL)												
						BC	DE	HL	SP									
		g (LO = 0 to 7)																
		B	D	H		B	D	H										
	HI	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
LO		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0000	0	IN0 g, (m)				IN g, (C)						LDI	LDIR					0
0001	1	OUT0 (m), g				OUT (C), g						CPI	CPIR					1
0010	2					SBC HL, ww						INI	INIR					2
0011	3					LD (mn), ww				OTIM	OTIMR	OUTI	OTIR					3
0100	4	TST g			TST (HL)	NEG		TST M	TSTIO m									4
0101	5					RETN												5
0110	6					IM 0	IM 1		SLP									6
0111	7					LD I, A	LD A, I	RRD										7
1000	8	IN0 g, (m)				IN g, (C)						LDD	LDDR					8
1001	9	OUT0 (m), g				OUT (C), g						CPD	CPDR					9
1010	A					ADC HL, ww						IND	INDR					A
1011	B					LD ww, (mn)				OTDM	OTDMR	OUTD	OTDR					B
1100	C	TST g				MLT ww												C
1101	D					RETI												D
1110	E						IM 2											E
1111	F					LD R, A	LD A, R	RLD										F
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
		C	E	L	A	C	E	L	A									
		g (LO = 8 to F)																

Bus Cycle Conditions

A hyphen in the Address column indicates that address output is undefined. A "Z" in the Data column indicates that the data pin is at high impedance.

Instruction	Machine Cycle	States	Address	Data	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{ME}}$	$\overline{\text{IOE}}$	$\overline{\text{LIR}}$	$\overline{\text{HALT}}$	ST
ADD HL, ww	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2 to MC_3	TiTiTiTi	—	Z	1	1	1	1	1	1	1
ADD IX, ww ADD IY, yy	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3 to MC_6	TiTiTiTi	—	Z	1	1	1	1	1	1	1
ADC HL, ww SBC HL, ww	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3 to MC_6	TiTiTiTi	—	Z	1	1	1	1	1	1	1
ADD A, g ADC A, g SUB g SBC A, g AND g OR g XOR g CP g	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti	—	Z	1	1	1	1	1	1	1
ADD A, m ADC A, m SUB m SBC A, m AND m OR m XOR m CP m	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd operand address	2nd op-code	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{ME}}$	$\overline{\text{IOE}}$	$\overline{\text{LIR}}$	$\overline{\text{HALT}}$	ST
ADD A, (HL) ADC A, (HL) SUB (HL) SBC A, (HL) AND (HL) OR (HL) XOR (HL) CP (HL)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
ADD A, (IX + d) ADD A, (IY + d) ADC A, (IX + d) ADC A, (IY + d) SUB (IX + d) SUB (IY + d) SBC A, (IX + d) SBC A, (IY + d) AND (IX + d) AND (IY + d) OR (IX + d) OR (IY + d) XOR (IX + d) XOR (IY + d) CP (IX + d) CP (IY + d)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	d	0	1	0	1	1	1	1
	MC_4 to MC_5	TiTi	—	Z	1	1	1	1	1	1	1
	MC_6	$T_1T_2T_3$	IX + d IY + d	Data	0	1	0	1	1	1	1
BIT b, g	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
BIT b, (HL)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
BIT b, (IX + d) BIT b, (IY + d)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	d	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	3rd op-code address	3rd op-code	0	1	0	1	0	1	1
	MC_5	$T_1T_2T_3$	IX + d IY + d	Data	0	1	0	1	1	1	1
CALL mn	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
	MC_4	Ti	—	Z	1	1	1	1	1	1	1
	MC_5	$T_1T_2T_3$	SP – 1	PCH	1	0	0	1	1	1	1
	MC_6	$T_1T_2T_3$	SP – 2	PCL	1	0	0	1	1	1	1
CALL f, mn (If condition is false)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
CALL f, mn (If condition is false)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
	MC_4	Ti	—	Z	1	1	1	1	1	1	1
	MC_5	$T_1T_2T_3$	SP – 1	PCH	1	0	0	1	1	1	1
	MC_6	$T_1T_2T_3$	SP – 2	PCL	1	0	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{ME}}$	$\overline{\text{IOE}}$	$\overline{\text{LIR}}$	$\overline{\text{HALT}}$	ST
CCF	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
CPI CPD	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4 to MC_6	TiTiTi	—	Z	1	1	1	1	1	1	1
CPIR CPDR (If BCR ≠ 0 and Ar ≠ $(HL)_M$)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4 to MC_8	TiTiTiTiTi	—	Z	1	1	1	1	1	1	1
CPIR CPDR (If BCR = 0 or Ar = $(HL)_M$)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4 to MC_6	TiTiTiTiTi	—	Z	1	1	1	1	1	1	1
CPL	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
DAA	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti	—	Z	1	1	1	1	1	1	1
DI$^{\text{Note}}$	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0

Note: Interrupts are not detected at the end of the DI instruction.

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
DJNZ j (If Br ≠ 0)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti*1	—	Z	1	1	1	1	1	1	1
	MC_3	$T_1T_2T_3$	1st operand address	j – 2	0	1	0	1	1	1	1
	MC_4 to MC_5	TiTi	—	Z	1	1	1	1	1	1	1
DJNZ j (If Br = 0)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti*1	—	Z	1	1	1	1	1	1	1
	MC_3	$T_1T_2T_3$	1st operand address	j – 2	0	1	0	1	1	1	1
EI*2	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
EX DE, HL EXX	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
EX AF, AF'	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti	—	Z	1	1	1	1	1	1	1
EX (SP), HL	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	SP	Data	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	SP + 1	Data	0	1	0	1	1	1	1
	MC_4	Ti	—	Z	1	1	1	1	1	1	1
	MC_5	$T_1T_2T_3$	SP + 1	H	1	0	0	1	1	1	1
	MC_6	$T_1T_2T_3$	SP	L	1	0	0	1	1	1	1

Notes: 1. Immediately after this state, DMA, refresh, and bus release cannot be executed. Any such requests are ignored.

2. Interrupts are not detected at the end of the EI instruction.

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
EX (SP), IX EX (SP), IY	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	SP	Data	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	SP + 1	Data	0	1	0	1	1	1	1
	MC_5	Ti	—	Z	1	1	1	1	1	1	1
	MC_6	$T_1T_2T_3$	SP + 1	IXH IYH	1	0	0	1	1	1	1
	MC_7	$T_1T_2T_3$	SP	IXL IYL	1	0	0	1	1	1	1
HALT	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	—	—	Next op-code address	Next op-code	0	1	0	1	0	0	0
IM 0 IM 1 IM 2	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
INC g DEC g	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti	—	Z	1	1	1	1	1	1	1
INC (HL) DEC (HL)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	HL	Data	1	1	1	1	1	1	1
INC (IX + d) INC (IY + d) DEC (IX + d) DEC (IY + d)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	1st operand address	d	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{ME}}$	$\overline{\text{IOE}}$	$\overline{\text{LIR}}$	$\overline{\text{HALT}}$	ST
INC (IX + d) INC (IY + d) DEC (IX + d) DEC (IY + d)	MC_4 to MC_5	TiTi	—	Z	1	1	1	1	1	1	1
	MC_6	$T_1T_2T_3$	IX + d IY + d	Data	0	1	0	1	1	1	1
	MC_7	Ti	—	Z	1	1	1	1	1	1	1
	MC_8	$T_1T_2T_3$	IX + d IY + d	Data	1	0	0	1	1	1	1
INC ww DEC ww	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti	—	Z	1	1	1	1	1	1	1
INC IX INC IY DEC IX INC IY	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
IN A, (m)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	m	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	m to A_0–A_7 A to A_8–A_{15}	Data	0	1	1	0	1	1	1
IN g, (C)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	BC	Data	0	1	1	0	1	1	1
IN0 g, (m)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	m	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	m to A_0–A_7 00H to A_8–A_{15}	Data	0	1	1	0	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{ME}}$	$\overline{\text{IOE}}$	$\overline{\text{LIR}}$	$\overline{\text{HALT}}$	ST
INI IND	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	BC	Data	0	1	1	0	1	1	1
	MC_4	$T_1T_2T_3$	HL	Data	1	0	0	1	1	1	1
INIR INDR (If Br ≠ 0)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	BC	Data	0	1	1	0	1	1	1
	MC_4	$T_1T_2T_3$	HL	Data	1	0	0	1	1	1	1
	MC_5 to MC_6	TiTi	—	Z	1	1	1	1	1	1	1
INIR INDR (If Br = 0)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	BC	Data	0	1	1	0	1	1	1
	MC_4	$T_1T_2T_3$	HL	Data	1	0	0	1	1	1	1
JP mn	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
JP f, mn (If f is false)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
JP f, mn (If is true)	MC$_1$	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC$_3$	$T_1T_2T_3$	2nd operand address	m	0	1	1	1	1	1	1
JP (HL)	MC$_1$	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
JP (IX) JP (IY)	MC$_1$	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
JR j	MC$_1$	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	$T_1T_2T_3$	1st operand address	j – 2	0	1	0	1	1	1	1
	MC$_3$ to MC$_4$	TiTi	—	Z	1	1	1	1	1	1	1
JR C, j JR NC, j JR Z, j JR NZ, j (If condition is false)	MC$_1$	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	$T_1T_2T_3$	1st operand address	j – 2	0	1	0	1	1	1	1
JR C, j JR NC, j JR Z, j JR NZ, j (If condition is true)	MC$_1$	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	$T_1T_2T_3$	1st operand address	j – 2	0	1	0	1	1	1	1
	MC$_3$ to MC$_4$	TiTi	—	Z	1	1	1	1	1	1	1
LD g, g'	MC$_1$	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	Ti	—	Z	1	1	1	1	1	1	1
LD g, m	MC$_1$	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	$T_1T_2T_3$	1st operand address	m	0	1	0	1	1	1	1
LD g, (HL)	MC$_1$	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC$_2$	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
LD g, (IX + d) LD g, (IY + d)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	d	0	1	0	1	1	1	1
	MC_4 to MC_5	TiTi	—	Z	1	1	1	1	1	1	1
	MC_6	$T_1T_2T_3$	IX + d IY + d	Data	0	1	0	1	1	1	1
LD (HL), g	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti	—	Z	1	1	1	1	1	1	1
	MC_3	$T_1T_2T_3$	HL	g	1	0	0	1	1	1	1
LD (IX + d), g LD (IY + d), g	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	d	0	1	0	1	1	1	1
	MC_4 to MC_6	TiTiTi	—	Z	1	1	1	1	1	1	1
	MC_7	$T_1T_2T_3$	IX + d IY + d	g	1	0	0	1	1	1	1
LD (HL), m	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	m	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	HL	Data	1	0	0	1	1	1	1
LD (IX + d), m LD (IY + d), m	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	d	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
	MC_5	$T_1T_2T_3$	IX + d IY + d	Data	1	0	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
LD A, (BC) LD A, (DE)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	BC DE	Data	0	1	0	1	1	1	1
LD A, (mn)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	mn	Data	1	1	1	1	1	1	1
LD (BC), A LD (DE), A	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti	—	Z	1	1	1	1	1	1	1
	MC_3	$T_1T_2T_3$	BC DE	A	1	0	0	1	1	1	1
LD (mn), A	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	2nd op-code address	m	0	1	0	1	1	1	1
	MC_4	Ti	—	Z	1	1	1	1	1	1	1
	MC_5	$T_1T_2T_3$	mn	A	1	0	0	1	1	1	1
LD A, I[Note] LD A, R[Note] LD I, A LD R, A	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
LD ww, mn	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	2nd op-code address	m	0	1	0	1	1	1	1

Note: Interrutps are not detected at the end of the LD A, I or LD A, R instruction.

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
LD IX, mn LD IY, mn	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
LD HL, (mn)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	mn	Data	0	1	0	1	1	1	1
	MC_5	$T_1T_2T_3$	mn + 1	Data	0	1	0	1	1	1	1
LD ww, (mn)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
	MC_5	$T_1T_2T_3$	mn	Data	0	1	0	1	1	1	1
	MC_6	$T_1T_2T_3$	mn + 1	Data	0	1	0	1	1	1	1
LD IX, (mn) LD IY, (mn)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
	MC_5	$T_1T_2T_3$	mn	Data	0	1	0	1	1	1	1
	MC_6	$T_1T_2T_3$	mn + 1	Data	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
LD (mn), HL	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
	MC_4	Ti	—	Z	1	1	1	1	1	1	1
	MC_5	$T_1T_2T_3$	mn	L	1	0	0	1	1	1	1
	MC_6	$T_1T_2T_3$	mn + 1	H	1	0	0	1	1	1	1
LD (mn), ww	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
	MC_5	Ti	—	Z	1	1	1	1	1	1	1
	MC_6	$T_1T_2T_3$	mn	wwL	1	0	0	1	1	1	1
	MC_7	$T_1T_2T_3$	mn + 1	wwH	1	0	0	1	1	1	1
LD (mn), IX LD (mn), IY	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	n	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	2nd operand address	m	0	1	0	1	1	1	1
	MC_5	Ti	—	Z	1	1	1	1	1	1	1
	MC_6	$T_1T_2T_3$	mn	IXL IYL	1	0	0	1	1	1	1
	MC_7	$T_1T_2T_3$	mn + 1	IXH IYH	1	0	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
LD SP, HL	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti	—	Z	1	1	1	1	0	1	1
LD SP, IX LD SP, IY	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
LDI LDD	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	DE	Data	1	0	0	1	1	1	1
LDIR LDDR (If BCR ≠ 0)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	DE	Data	1	0	0	1	1	1	1
	MC_5 to MC_6	TiTi	—	Z	1	1	1	1	1	1	1
LDIR LDDR (If BCR = 0)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	DE	Data	1	0	0	1	1	1	1
MLT ww	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3 to MC_{13}	TiTiTiTi TiTiTiTi TiTiTi	—	Z	1	1	1	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
NEG	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
NOP	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
OUT (m), A	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	1st operand address	m	0	1	0	1	1	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	m to A_0–A_7 A to A_8–A_{15}	A	1	0	1	0	1	1	1
OUT (C), g	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	BC	g	1	0	1	0	1	1	1
OUT0 (m), g	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	m	0	1	0	1	1	1	1
	MC_4	Ti	—	Z	1	1	1	1	1	1	1
	MC_5	$T_1T_2T_3$	m to A_0–A_7 00H to A_8–A_{15}	g	1	0	1	0	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
OTIM OTDM	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_5	$T_1T_2T_3$	C to A_0–A_7 00H to A_8–A_{15}	Data	1	0	1	0	1	1	1
	MC_6	Ti	—	Z	1	1	1	1	1	1	1
OTIMR OTDMR (If Br ≠ 0)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_5	$T_1T_2T_3$	C to A_0–A_7 00H to A_8–A_{15}	Data	1	0	1	0	1	1	1
	MC_6 to MC_8	TiTiTi	—	Z	1	1	1	1	1	1	1
OTIMR OTDMR (If Br = 0)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_5	$T_1T_2T_3$	C to A_0–A_7 00H to A_8–A_{15}	Data	1	0	1	0	1	1	1
	MC_6 to	Ti	—	Z	1	1	1	1	1	1	1

	Machine										
Instruction	Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
OUTI OUTD	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	BC	Data	1	0	1	0	1	1	1
OTIR OTDR (If Br ≠ 0)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	BC	Data	1	0	1	0	1	1	1
	MC_5 to MC_6	TiTi	—	Z	1	1	1	1	1	1	1
OTIR OTDR (If Br = 0)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	BC	Data	1	0	1	0	1	1	1
POP zz	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	SP	Data	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	SP + 1	Data	0	1	0	1	1	1	1
POP IX POP IY	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	SP	Data	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	SP + 1	Data	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
PUSH zz	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2 to MC_3	TiTi	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	SP – 1	zzH	1	0	0	1	1	1	1
	MC_5	$T_1T_2T_3$	SP – 1	zzL	1	0	0	1	1	1	1
PUSH IX PUSH IY	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3 to MC_4	TiTi	—	Z	1	1	1	1	1	1	1
	MC_5	$T_1T_2T_3$	SP – 1	IXH IYH	1	0	0	1	1	1	1
	MC_6	$T_1T_2T_3$	SP – 2	IXL IYL	1	0	0	1	1	1	1
RET	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	SP	Data	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	SP + 2	Data	0	1	0	1	1	1	1
RET f (If condition is false)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2 to MC_3	TiTi	—	Z	1	1	1	1	1	1	1
RET f (If condition is true)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	Ti	—	Z	1	1	1	1	1	1	1
	MC_3	$T_1T_2T_3$	SP	Data	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	SP + 1	Data	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
RETI (R1) RETN	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	SP	Data	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	SP + 1	Data	0	1	0	1	1	1	1
RETI (Z)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0[Note] 1	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0[Note] 1	1	1
	MC_3 to MC_5	TiTiTi	—	Z	1	1	1	1	1[Note] 1	1	1
	MC_6	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0[Note] 0	1	1
	MC_7	Ti	—	Z	1	1	1	1	1[Note] 1	1	1
	MC_8	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0[Note] 0	1	1
	MC_9	$T_1T_2T_3$	SP	Data	0	1	0	1	1[Note] 1	1	1
	MC_{10}	$T_1T_2T_3$	SP + 1	Data	0	1	0	1	1[Note] 1	1	1
RLCA RLA RRCA RRA	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
RLC g RL g RRC g RR g SLA g SRA g SRL g	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1

Note: Upper value is $\overline{LIR}$ status when LIRE = 1. Lower value is $\overline{LIR}$ status when LIRE = 0.

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
RLC (HL) RL (HL) RRC (HL) RR (HL) SLA (HL) SRA (HL) SRL (HL)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4	Ti	—	Z	1	1	1	1	1	1	1
	MC_5	$T_1T_2T_3$	HL	Data	1	0	0	1	1	1	1
RLC (IX + d) RLC (IY + d) RL (IX + d) RL (IY + d) RRC (IX + d) RRC (IY + d) RR (IX + d) RR (IY + d) SLA (IX + d) SLA (IY + d) SRA (IX + d) SRA (IY + d) SRL (IX + d) SRL (IY + d)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	d	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	3rd op-code address	3rd op-code	0	1	0	1	0	1	1
	MC_5	$T_1T_2T_3$	IX + d IY + d	Data	0	1	0	1	1	1	1
	MC_6	Ti	—	Z	1	1	1	1	1	1	1
	MC_7	$T_1T_2T_3$	IX + d IY + d	Data	1	0	0	1	1	1	1
RLD RRD	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4 to MC_7	TiTiTiTi	—	Z	1	1	1	1	1	1	1
	MC_8	$T_1T_2T_3$	HL	Data	1	0	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{ME}}$	$\overline{\text{IOE}}$	$\overline{\text{LIR}}$	$\overline{\text{HALT}}$	ST
RST v	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2 to MC_3	TiTi	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	SP – 1	PCH	1	0	0	1	1	1	1
	MC_5	$T_1T_2T_3$	SP – 2	PCL	1	0	0	1	1	1	1
SCF	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
SET b, g RES b, g	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
SET b, (HL) RES b, (HL)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1
	MC_4	Ti	—	Z	1	1	1	1	1	1	1
	MC_5	$T_1T_2T_3$	HL	Data	1	0	0	1	1	1	1

SET b, (IX + d)
SET b, (IY + d)
RES b, (IX + d)
RES b, (IY + d)
	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0

	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	d	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	3rd op-code address	3rd op-code	0	1	0	1	0	1	1
	MC_5	$T_1T_2T_3$	IX + d IY + d	Data	0	1	0	1	1	1	1
	MC_6	Ti	—	Z	1	1	1	1	1	1	1
	MC_7	$T_1T_2T_3$	IX + d IY + d	Data	1	0	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{ME}}$	$\overline{\text{IOE}}$	$\overline{\text{LIR}}$	$\overline{\text{HALT}}$	ST
SLP	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	—	—	FFFFFH	Z	1	1	1	1	1	0	1
TSTIO m	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	m	0	1	0	1	1	1	1
	MC_4	$T_1T_2T_3$	C to A_0–A_7 00H to A_8–A_{15}	Data	0	1	1	1	1	1	1
TST g	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
TST m	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	$T_1T_2T_3$	1st operand address	m	1	1	1	1	1	1	1
TST (HL)	MC_1	$T_1T_2T_3$	1st op-code address	1st op-code	0	1	0	1	0	1	0
	MC_2	$T_1T_2T_3$	2nd op-code address	2nd op-code	0	1	0	1	0	1	1
	MC_3	Ti	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	HL	Data	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{ME}}$	$\overline{\text{IOE}}$	$\overline{\text{LIR}}$	$\overline{\text{HALT}}$	ST
$\overline{\text{NMI}}$	MC_1	$T_1T_2T_3$	Next op-code address (PC)		0	1	0	1	0	1	0
	MC_2 to MC_3	TiTi	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	SP – 1	PCH	1	0	0	1	1	1	1
	MC_5	$T_1T_2T_3$	SP – 2	PCL	1	0	0	1	1	1	1
$\overline{\text{INT}}_0$ mode 0 (RST inserted)	MC_1	$T_1T_2T_W$ T_WT_3	Next op-code address (PC)	1st op-code	1	1	1	0	0	1	0
	MC_2 to MC_3	TiTi	—	Z	1	1	1	1	1	1	1
	MC_4	$T_1T_2T_3$	SP – 1	PCH	1	0	0	1	1	1	1
	MC_5	$T_1T_2T_3$	SP – 2	PCL	1	0	0	1	1	1	1
$\overline{\text{INT}}_0$ mode 0 (CALL inserted)	MC_1	$T_1T_2T_W$ T_WT_3	Next op-code address (PC)	1st op-code	1	1	1	0	0	1	0
	MC_2	$T_1T_2T_3$	PC	n	0	1	0	1	1	1	1
	MC_3	$T_1T_2T_3$	PC + 1	m	0	1	0	1	1	1	1
	MC_4	Ti	—	Z	1	1	1	1	1	1	1
	MC_5	$T_1T_2T_3$	SP – 1	PC + 2 (H)	1	0	0	1	1	1	1
	MC_6	$T_1T_2T_3$	SP – 2	PC + 2 (L)	1	0	0	1	1	1	1
$\overline{\text{INT}}_0$ mode 1	MC_1	$T_1T_2T_W$ T_WT_3	Next op-code address (PC)		1	1	1	0	0	1	0
	MC_2	$T_1T_2T_3$	SP – 1	PCH	1	0	0	1	1	1	1
	MC_3	$T_1T_2T_3$	SP + 2	PCL	1	0	0	1	1	1	1
$\overline{\text{INT}}_0$ mode 2	MC_1	$T_1T_2T_W$ T_WT_3	Next op-code address (PC)	vector	1	1	1	0	0	1	0
	MC_2	Ti	—	Z	1	1	1	1	1	1	1
	MC_3	$T_1T_2T_3$	SP – 1	PCH	1	0	0	1	1	1	1
	MC_4	$T_1T_2T_3$	SP – 2	PCL	1	0	0	1	1	1	1
	MC_5	$T_1T_2T_3$	I, vector	Data	0	1	0	1	1	1	1
	MC_6	$T_1T_2T_3$	I, vector + 1	Data	0	1	0	1	1	1	1

Instruction	Machine Cycle	States	Address	Data	$\overline{RD}$	$\overline{WR}$	$\overline{ME}$	$\overline{IOE}$	$\overline{LIR}$	$\overline{HALT}$	ST
$\overline{INT_1}$, $\overline{INT_2}$, internal interrupts	MC_1	$T_1T_2T_W$ T_WT_3	Next op-code address (PC)		1	1	1	1	1	1	0
	MC_2	Ti	—	Z	1	1	1	1	1	1	1
	MC_3	$T_1T_2T_3$	SP – 1	PCH	1	0	0	1	1	1	1
	MC_4	$T_1T_2T_3$	SP – 2	PCL	1	0	0	1	1	1	1
	MC_5	$T_1T_2T_3$	I, vector	Data	0	1	0	1	1	1	1
	MC_6	$T_1T_2T_3$	I, vector + 1	Data	0	1	0	1	1	1	1

Acceptable Requests in Each Mode

	Current Status							
Action Request	**Normal Mode (CPU Mode) (IOSTOP Mode)**	**Wait State**	**Refresh Cycle**	**Interrupt Acknowledge Cycle**	**DMA Cycle**	**Bus Release Mode**	**Sleep Mode**	**Standby Mode**
$\overline{\text{WAIT}}$	Accepted	Accepted	Not accepted	Accepted	Accepted	Not accepted	Not accepted	Not accepted
Refresh request (request for insertion of refresh cycle by on-chip refresh circuit)	Refresh cycle inserted at machine cycle break	Not accepted	Not accepted	Refresh cycle inserted at machine cycle break	Refresh cycle inserted at machine cycle break	Not accepted	Not accepted	Not accepted
$\overline{\text{DREQ}_0}$ $\overline{\text{DREQ}_1}$	DMA cycle inserted at machine cycle break	Accepted, but DMA cycle not inserted until machine cycle break	Accepted * Following completion of refresh cycle, 1 machine cycle is executed and then DMA cycle is inserted.	DMA cycle inserted at machine cycle break	Accepted (Consult this manual for full details.)	Accepted * Following completion of bus release cycle, 1 machine cycle is executed and then DMA cycle is inserted.	Not accepted	Not accepted
$\overline{\text{BUSREQ}}$	Bus release mode entered at machine cycle break	Not accepted	Not accepted	Bus release mode entered at machine cycle break	Bus release mode entered at machine cycle break	Bus release mode continues	Accepted	Accepted

Note: * Not accepted when $\overline{\text{DREQ}_0}$ and $\overline{\text{DREQ}_1}$ are set for level detection.

		Current Status							
	Action Request	**Normal Mode (CPU Mode) (IOSTOP Mode)**	**Wait State**	**Refresh Cycle**	**Interrupt Acknowledge Cycle**	**DMA Cycle**	**Bus Release Mode**	**Sleep Mode**	**Standby Mode**
Interrupts	$\overline{INT_0}$, $\overline{INT_1}$, $\overline{INT_2}$	Accepted in final machine cycle of instruction	Accepted in final machine cycle of instruction	Not accepted	Not accepted	Not accepted	Not accepted	Accepted, sleep mode exited and normal status recovered	Not accepted
	Internal I/O interrupt request	Accepted in final machine cycle of instruction	Accepted in final machine cycle of instruction	Not accepted	Not accepted	Not accepted	Not accepted	Accepted, sleep mode exited and normal status recovered	Not accepted
	$\overline{NMI}$	Accepted in final machine cycle of instruction	Accepted in final machine cycle of instruction	Not accepted	Not accepted **Accepted in final machine cycle of instruction following completion of acknowledge cycle.	Accepted, DMA suspended	Not accepted	Accepted, sleep mode exited and normal status recovered	Accepted, standby mode exited and normal status recovered. Note that IOSTOP bit remains set to 1.

Note: ** Accepted when $\overline{INT_0}$ is being used in mode 0. The NMI acknowledge cycle begins following execution of the instruction placed on the bus.

■ Request Priority Sequence

Three types of requests can be input to the CPU.

1. Requests that can be accepted and executed at the end of a state (WAIT)

2. Requests that can be accepted and executed at the end of a machine cycle (refresh requests, DMA requests, $\overline{\text{BUSREQ}}$)

3. Requests that can be accepted and executed at the end of an instruction (all interrupts)

   Basically, the priority sequence gives 1 the highest priority, and 3 the lowest.

   Within group 2 above, the descending priority sequence is:

   $\overline{\text{BUSREQ}}$
   Refresh request
   DMA request

   If a $\overline{\text{BUSREQ}}$ and refresh request are input simultaneously, the $\overline{\text{BUSREQ}}$ is given priority.

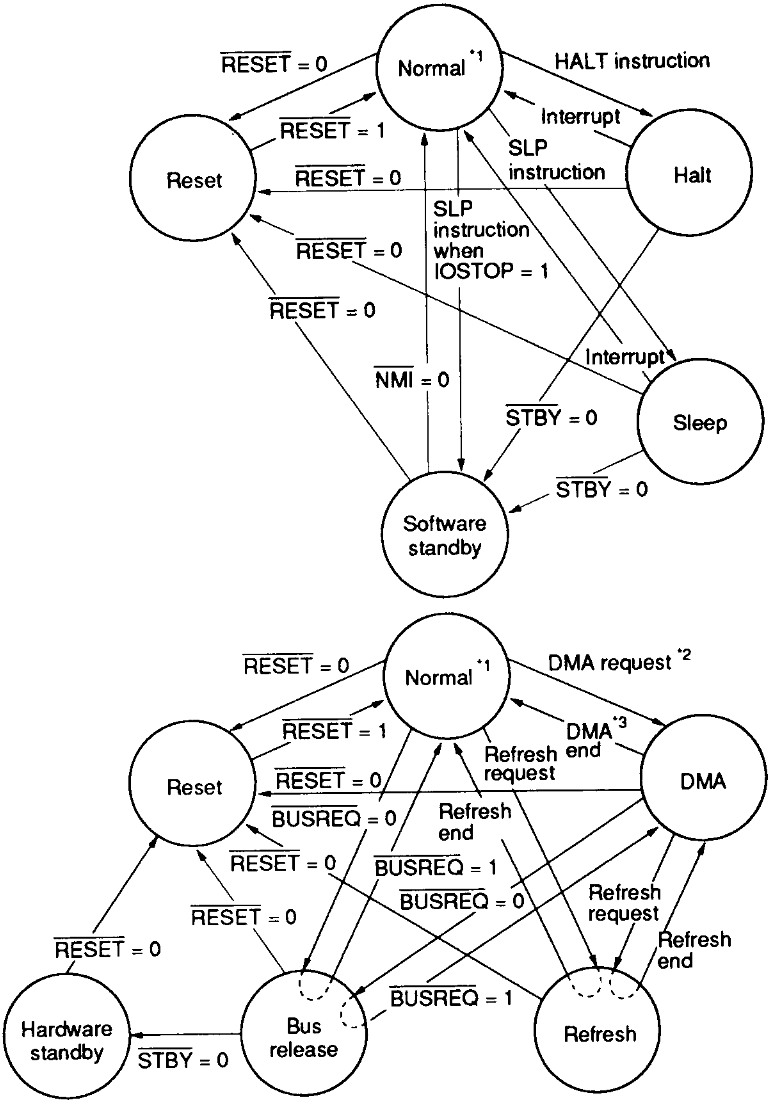
   For the priority sequence of the interrupts in group 3 above, see the section of this manual that details interrupt operations.

4. Among requests that are accepted and executed in the final machine cycle of an instruction, the descending priority sequence is:

   Bus requests ($\overline{\text{BUSREQ}}$, refresh request, DMA request)

   Interrupt

■ Mode Transitions



Notes: 1. Normal: Normal CPU instruction execution mode

2. DMA request: $\overline{\text{DREQ}}_0$ or $\overline{\text{DREQ}}_1$ = 0 (for memory ↔ (memory-mapped) I/O transfers)
DE0 = 1 (for Memory ↔ Memory transfers)

3. DMA end: $\overline{\text{DREQ}}_0$ or $\overline{\text{DREQ}}_1$ = 1 (for memory ↔ (memory-mapped) I/O transfers)
BCR0 and BCR1 = 0000H (for all transfer modes)
$\overline{\text{NMI}}$ = 0 (for all transfer modes)

In addition to the above, the following mode transitions are also possible.

1. From halt to DMA, refresh, or bus release, and vice versa.
2. From sleep to bus release, and vice versa.

■ Status Signal List

The following list shows the status signal output for each mode.

Mode		$\overline{\text{LIR}}$	$\overline{\text{ME}}$	$\overline{\text{IOE}}$	$\overline{\text{RD}}$	$\overline{\text{WR}}$	$\overline{\text{REF}}$	$\overline{\text{HALT}}$	$\overline{\text{BUSACK}}$	ST	Address Bus	Data Bus
CPU operation	1st op code fetch	0	0	1	0	1	1	1	1	0	A	In
	Other op code fetch	0	0	1	0	1	1	1	1	1	A	In
	Memory read	1	0	1	0	1	1	1	1	1	A	In
	Memory write	1	0	1	1	0	1	1	1	1	A	Out
	I/O read	1	1	0	0	1	1	1	1	1	A	In
	I/O write	1	1	0	1	0	1	1	1	1	A	Out
	Internal operation	1	1	1	1	1	1	1	1	1	A	In
Refresh		1	0	1	1	1	0	1	1	*	A	In
Interrupt acknowledge (1st machine cycle)	$\overline{\text{NMI}}$	0	0	1	0	1	1	1	1	0	A	In
	$\overline{\text{INT}_0}$	0	1	0	1	1	1	1	1	0	A	In
	$\overline{\text{INT}_1}$, $\overline{\text{INT}_2}$, and internal interrupts	1	1	1	1	1	1	1	1	0	A	In
Bus release		1	Z	Z	Z	Z	1	1	0	*	Z	In
Halt		0	0	1	0	1	1	0	1	0	A	In
Sleep		1	1	1	1	1	1	0	1	1	1	In
Internal DMA	Memory read	1	0	1	0	1	1	*	1	0	A	In
	Memory write	1	0	1	1	0	1	*	1	0	A	Out
	I/O read	1	1	0	0	1	1	*	1	0	A	In
	I/O write	1	1	0	1	0	1	1	1	0	A	Out
Reset	1	1	1	1	1	1	1	1	1	1	Z	In

1 = high level; 0 = low level; * = undefined; A = any value; Z = high impedance; IN = input; OUT = output

Internal I/O Register Reference

The upper 8 bits of the I/O register address are all 0. The most significant bit of the lower 8 bits can be set using the IOA7 bit of the IO control register. The table below shows the addresses when IOA7 = 0.

<table>
<tr><th>Register</th><th>Address</th><th>Remarks</th></tr>
<tr><td>DMA source address register channel 0 L: SAR0L</td><td>20</td><td></td></tr>
<tr><td>DMA source address register channel 0 H: SAR0H</td><td>21</td><td></td></tr>
<tr><td>DMA source address register channel 0 B: SAR0B</td><td>22</td><td>Only bits 0, 1, 2, and 3 are used
<table>
<tr><th>A_{19},</th><th>A_{18},</th><th>A_{17},</th><th>A_{16}</th><th>DMA Transfer Request</th></tr>
<tr><td>×</td><td>×</td><td>0</td><td>0</td><td>$\overline{\text{DREQ}}_0$ (external)</td></tr>
<tr><td>×</td><td>×</td><td>0</td><td>1</td><td>Not used</td></tr>
<tr><td>×</td><td>×</td><td>1</td><td>0</td><td>Not used</td></tr>
<tr><td>×</td><td>×</td><td>1</td><td>1</td><td>Not used</td></tr>
</table></td></tr>
<tr><td>DMA destination address register channel 0 L: DAR0L</td><td>23</td><td></td></tr>
<tr><td>DMA destination address register channel 0 H: DAR0H</td><td>24</td><td></td></tr>
<tr><td>DMA destination address register channel 0 B: DAR0B</td><td>25</td><td>Only bits 0, 1, 2, and 3 are used
<table>
<tr><th>A_{19},</th><th>A_{18},</th><th>A_{17},</th><th>A_{16}</th><th>DMA Transfer Request</th></tr>
<tr><td>×</td><td>×</td><td>0</td><td>0</td><td>$\overline{\text{DREQ}}_0$ (external)</td></tr>
<tr><td>×</td><td>×</td><td>0</td><td>1</td><td>Not used</td></tr>
<tr><td>×</td><td>×</td><td>1</td><td>0</td><td>Not used</td></tr>
<tr><td>×</td><td>×</td><td>1</td><td>1</td><td>Not used</td></tr>
</table></td></tr>
<tr><td>DMA byte count register channel 0 L: BCR0L</td><td>26</td><td></td></tr>
<tr><td>DMA byte count register channel 0 H: BCR0H</td><td>27</td><td></td></tr>
</table>

<table>
<tr><th>Register</th><th>Address</th><th>Remarks</th></tr>
<tr><td>DMA memory address register channel 1 L: MAR1L</td><td>28</td><td></td></tr>
<tr><td>DMA memory address register channel 1 H: MAR1H</td><td>29</td><td></td></tr>
<tr><td>DMA memory address register channel 1 B: MAR1B</td><td>2A</td><td>Only bits 0, 1, 2, and 3 are used</td></tr>
<tr><td>DMA I/O address register channel 1 L: IAR1L</td><td>2B</td><td></td></tr>
<tr><td>DMA I/O address register channel 1 H: IAR1H</td><td>2C</td><td></td></tr>
<tr><td>DMA byte count register channel 1 L: BCR1L</td><td>2E</td><td></td></tr>
<tr><td>DMA byte count register channel 1 H: BCR1H</td><td>2F</td><td></td></tr>
<tr><td>DMA status register: DSTAT</td><td>30</td><td>
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>DE1</td><td>DE0</td><td>$\overline{DWE1}$</td><td>$\overline{DWE0}$</td><td>DIE1</td><td>DIE0</td><td>—</td><td>DME</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>0</td><td>1</td><td>0</td></tr>
<tr><td>Read/Write</td><td>R/W</td><td>R/W</td><td>W</td><td>W</td><td>R/W</td><td>R/W</td><td>—</td><td>R</td></tr>
</table>
</td></tr>
<tr><td>DMA mode register: DMODE</td><td>31</td><td>
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>—</td><td>—</td><td>DM1</td><td>DM0</td><td>SM1</td><td>SM0</td><td>MMOD</td><td>—</td></tr>
<tr><td>Initial value</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td><td>0</td><td>1</td></tr>
<tr><td>Read/Write</td><td>—</td><td>—</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>—</td></tr>
</table>
</td></tr>
<tr><td>DMA/WAIT control register: DCNTL</td><td>32</td><td>
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>MWI1</td><td>MWI0</td><td>IWI1</td><td>IWI0</td><td>DMS1</td><td>DMS0</td><td>DIM1</td><td>DIM0</td></tr>
<tr><td>Initial value</td><td>1</td><td>1</td><td>1</td><td>1</td><td>0</td><td>0</td><td>0</td><td>0</td></tr>
<tr><td>Read/Write</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td><td>R/W</td></tr>
</table>
</td></tr>
<tr><td>Interrupt vector low register: IL</td><td>33</td><td>
<table>
<tr><td>Bit</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr>
<tr><td></td><td>IL7</td><td>IL6</td><td>IL5</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td>Initial value</td><td>0</td><td>0</td><td>0</td><td>1</td><td>1</td><td>1</td><td>1</td><td>1</td></tr>
<tr><td>Read/Write</td><td>R/W</td><td>R/W</td><td>R/W</td><td>—</td><td>—</td><td>—</td><td>—</td><td>—</td></tr>
<tr><td></td><td colspan="3">Arbitrary values</td><td colspan="5">Fixed code</td></tr>
</table>
</td></tr>
</table>

Register	Address	Remarks							
INT/TRAP control register: ITC	34	Bit	7	6	5	4	3	2	1	0
			TRAP	UFO	—	—	—	ITE2	ITE1	ITE0
		Initial value	0	0	1	1	1	0	0	1
		Read/Write	R/W	R	—	—	—	R/W	R/W	R/W
Refresh control register: RCR	36	Bit	7	6	5	4	3	2	1	0
			REFE	REFW	—	—	—	—	CYC1	CYC0
		Initial value	1	1	1	1	1	1	0	0
		Read/Write	R/W	R/W	—	—	—	—	R/W	R/W
MMU common base register: CBR	38	Bit	7	6	5	4	3	2	1	0
			CB7	CB6	CB5	CB4	CB3	CB2	CB1	CB0
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MMU bank base register: BBR	39	Bit	7	6	5	4	3	2	1	0
			BB7	BB6	BB5	BB4	BB3	BB2	BB1	BB0
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
MMU common/bank area register: CBAR	3A	Bit	7	6	5	4	3	2	1	0
			CA3	CA2	CA1	CA0	BA3	BA2	BA1	BA0
		Initial value	1	1	1	1	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Operating mode control register: OMCR	3E	Bit	7	6	5	4	3	2	1	0
			LIRE	$\overline{\text{LIRTE}}$	$\overline{\text{IOC}}$	—	—	—	—	—
		Initial value	1	1	1	1	1	1	1	1
		Read/Write	R/W	W	R/W	—	—	—	—	—
I/O control register: IOCR	3F	Bit	7	6	5	4	3	2	1	0
			IOAR	—	IOSTOP	—	—	—	—	—
		Initial value	0	1	0	1	1	1	1	1
		Read/Write	R/W	—	R/W	—	—	—	—	—

Register	Address	Remarks							
Free-running counter H: FRCH	40	Bit	7	6	5	4	3	2	1	0
			FRC15	FRC14	FRC13	FRC12	FRC11	FRC10	FRC9	FRC8
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Free-running counter L: FRCL	41	Bit	7	6	5	4	3	2	1	0
			FRC7	FRC6	FRC5	FRC4	FRC3	FRC2	FRC1	FRC0
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Timer control/status register 1: TCSR1	42	Bit	7	6	5	4	3	2	1	0
			ICF	OCF	TOF	EICI	EOCI	ETOI	IEDG	OLVL
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Output compare register 1 H: OCR1H	43	Bit	7	6	5	4	3	2	1	0
			OCR15	OCR14	OCR13	OCR12	OCR11	OCR10	OCR9	OCR8
		Initial value	1	1	1	1	1	1	1	1
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Output compare register 1 L: OCR1L	44	Bit	7	6	5	4	3	2	1	0
			OCR7	OCR6	OCR5	OCR4	OCR3	OCR2	OCR1	OCR0
		Initial value	1	1	1	1	1	1	1	1
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Input capture register H: ICRH[Note]	45	Bit	7	6	5	4	3	2	1	0
			ICR15	ICR14	ICR13	ICR12	ICR11	ICR10	ICR9	ICR8
		Initial value	*	*	*	*	*	*	*	*
		Read/Write	R	R	R	R	R	R	R	R
										* Undefined
Input capture register L: ICRL[Note]	46	Bit	7	6	5	4	3	2	1	0
			ICR7	ICR6	ICR5	ICR4	ICR3	ICR2	ICR1	ICR0
		Initial value	*	*	*	*	*	*	*	*
		Read/Write	R	R	R	R	R	R	R	R
										* Undefined

Note: ICRH and ICRL are not initialized by reset.

Register	Address	Remarks							
Transmit/receive control/status register A 0: TRCSRA0	47	Bit	7	6	5	4	3	2	1	0
			RDRF0	ORFE0	TDRE0	RIE0	RE0	TIE0	TE0	WU0
		Initial value	0	0	1	0	0	0	0	0
		Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Transmit/receive control/status register B 0: TRCSRB0	48	Bit	7	6	5	4	3	2	1	0
			RDRF0	ORFE0	TDRE0	PER0	—	PEN0	EOP0	SBL0
		Initial value	0	0	1	0	1	0	0	0
		Read/Write	R	R	R	R	—	R/W	R/W	R/W
Rate/mode control register 0: RMCR0	49	Bit	7	6	5	4	3	2	1	0
			—	—	SS02	CC02	CC01	CC00	SS01	SS00
		Initial value	1	1	0	0	0	0	0	0
		Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W
Receive data register 0: RDR0	4A									
Transmit data register 0: TDR0	4B									
Serial port control register: SCIPCR	4C	Bit	7	6	5	4	3	2	1	0
			SCKHL0	SOUTM0	SINM0	SCKM0	SCKHL1	SOUTM1	SINM1	SCKM1
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
A/D control register: ADCR	4D	Bit	7	6	5	4	3	2	1	0
			—	—	—	—	—	—	ANE	$\overline{\text{AVREF}}$
		Initial value	1	1	1	1	1	1	0	0
		Read/Write	—	—	—	—	—	—	R/W	R/W
A/D control/status register: ADCSR	4E	Bit	7	6	5	4	3	2	1	0
			ADEF	ADS	EADEI	CH2	CH1	CH0	ACS1	ACS0
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register	Address	Remarks								
A/D result register: ADRR	4F	Bit	7	6	5	4	3	2	1	0
			ADRR7	ADRR6	ADRR5	ADRR4	ADRR3	ADRR2	ADRR1	ADRR0
		Initial value	•	•	•	•	•	•	•	•
		Read/Write	R	R	R	R	R	R	R	R
										• Undefined
Transmit/receive control/status register A 1: TRCSRA1	50	Bit	7	6	5	4	3	2	1	0
			RDRF1	ORFE1	TDRE1	RIE1	RE1	TIE1	TE1	WU1
		Initial value	0	0	1	0	0	0	0	0
		Read/Write	R	R	R	R/W	R/W	R/W	R/W	R/W
Transmit/receive control/status register B 1: TRCSRB1	51	Bit	7	6	5	4	3	2	1	0
			RDRF1	ORFE1	TDRE1	PER1	—	PEN1	EOP1	SBL1
		Initial value	0	0	1	0	1	0	0	0
		Read/Write	R	R	R	R	—	R/W	R/W	R/W
Rate/mode control register 1: RMCR1	52	Bit	7	6	5	4	3	2	1	0
			—	—	SS12	CC12	CC11	CC10	SS11	SS10
		Initial value	1	1	0	0	0	0	0	0
		Read/Write	—	—	R/W	R/W	R/W	R/W	R/W	R/W
Receive data register 1: RDR1	53									
Transmit data register 1: TDR1	54									
Timer 2 up-counter H: T2CNTH	55	Bit	7	6	5	4	3	2	1	0
			T2C15	T2C14	T2C13	T2C12	T2C11	T2C10	T2C9	T2C8
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Timer 2 up-counter L: T2CNTL	56	Bit	7	6	5	4	3	2	1	0
			T2C7	T2C6	T2C5	T2C4	T2C3	T2C2	T2C1	T2C0
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register	Address	Remarks								
Timer 2 time constant register H: T2CONRH	57	Bit	7	6	5	4	3	2	1	0
			T2CN15	T2CN14	T2CN13	T2CN12	T2CN11	T2CN10	T2CN9	T2CN8
		Initial value	1	1	1	1	1	1	1	1
		Read/Write	W	W	W	W	W	W	W	W
Timer 2 time constant register L: T2CONRL	58	Bit	7	6	5	4	3	2	1	0
			T2CN7	T2CN6	T2CN5	T2CN4	T2CN3	T2CN2	T2CN1	T2CN0
		Initial value	1	1	1	1	1	1	1	1
		Read/Write	W	W	W	W	W	W	W	W
Timer control/status register 2: TCSR2	59	Bit	7	6	5	4	3	2	1	0
			CMF2	ECM2I	—	—	T2OS1	T2OS0	CK2S1	CK2S0
		Initial value	0	0	1	1	0	0	0	0
		Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W
Timer 3 up-counter H: T3CNTH	5A	Bit	7	6	5	4	3	2	1	0
			T3C15	T3C14	T3C13	T3C12	T3C11	T3C10	T3C9	T3C8
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Timer 3 up-counter L: T3CNTL	5B	Bit	7	6	5	4	3	2	1	0
			T3C7	T3C6	T3C5	T3C4	T3C3	T3C2	T3C1	T3C0
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Timer 3 time constant register H: T3CONRH	5C	Bit	7	6	5	4	3	2	1	0
			T3CN15	T3CN14	T3CN13	T3CN12	T3CN11	T3CN10	T3CN9	T3CN8
		Initial value	1	1	1	1	1	1	1	1
		Read/Write	W	W	W	W	W	W	W	W
Timer 3 time constant register L: T3CONRL	5D	Bit	7	6	5	4	3	2	1	0
			T3CN7	T3CN6	T3CN5	T3CN4	T3CN3	T3CN2	T3CN1	T3CN0
		Initial value	1	1	1	1	1	1	1	1
		Read/Write	W	W	W	W	W	W	W	W
Timer control/status register 3: TCSR3	5E	Bit	7	6	5	4	3	2	1	0
			CMF3	ECM3I	—	—	T3OS1	T3OS0	—	—
		Initial value	0	0	1	1	0	0	1	1
		Read/Write	R/W	R/W	—	—	R/W	R/W	—	—

Register	Address	Remarks								
Timer 4 up-counter H: T4CNTH	5F	Bit	7	6	5	4	3	2	1	0
			T4C15	T4C14	T4C13	T4C12	T4C11	T4C10	T4C9	T4C8
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Timer 4 up-counter L: T4CNTL	60	Bit	7	6	5	4	3	2	1	0
			T4C7	T4C6	T4C5	T4C4	T4C3	T4C2	T4C1	T4C0
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
Timer 4 time constant register H: T4CONRH	61	Bit	7	6	5	4	3	2	1	0
			T4CN15	T4CN14	T4CN13	T4CN12	T4CN11	T4CN10	T4CN9	T4CN8
		Initial value	1	1	1	1	1	1	1	1
		Read/Write	W	W	W	W	W	W	W	W
Timer 4 time constant register L: T4CONRL	62	Bit	7	6	5	4	3	2	1	0
			T4CN7	T4CN6	T4CN5	T4CN4	T4CN3	T4CN2	T4CN1	T4CN0
		Initial value	1	1	1	1	1	1	1	1
		Read/Write	W	W	W	W	W	W	W	W
Timer control/status register 4: TCSR4	63	Bit	7	6	5	4	3	2	1	0
			CMF4	ECM4I	—	—	T4OS1	T4OS0	CK4S1	CK4S0
		Initial value	0	0	1	1	0	0	0	0
		Read/Write	R/W	R/W	—	—	R/W	R/W	R/W	R/W
Output data register 0: ODR0	64	Bit	7	6	5	4	3	2	1	0
			$ODR0_7$	$ODR0_6$	$ODR0_5$	$ODR0_4$	$ODR0_3$	$ODR0_2$	$ODR0_1$	$ODR0_0$
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Note: Reading obtains input port values.								
Output data register 1: ODR1	65	Bit	7	6	5	4	3	2	1	0
			$ODR1_7$	$ODR1_6$	$ODR1_5$	$ODR1_4$	$ODR1_3$	$ODR1_2$	$ODR1_1$	$ODR1_0$
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Note: Reading obtains input port values.								
Output data register 2: ODR2	66	Bit	7	6	5	4	3	2	1	0
			—	$ODR2_6$	$ODR2_5$	$ODR2_4$	$ODR2_3$	$ODR2_2$	$ODR2_1$	$ODR2_0$
		Initial value	1	0	0	0	0	0	0	0
		Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W
		Note: Reading obtains input port values.								

Register	Address	Remarks							
Output data register 3: ODR3	67	Bit	7	6	5	4	3	2	1	0
			ODR3$_7$	ODR3$_6$	ODR3$_5$	ODR3$_4$	ODR3$_3$	ODR3$_2$	ODR3$_1$	ODR3$_0$
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
							Note: Reading obtains input port values.			
Port 4: PORT4	68									
Port 0 data direction register: DDR0	69	Bit	7	6	5	4	3	2	1	0
			DDR0$_7$	DDR0$_6$	DDR0$_5$	DDR0$_4$	DDR0$_3$	DDR0$_2$	DDR0$_1$	DDR0$_0$
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	W	W	W	W	W	W	W	W
Port 1 data direction register: DDR1	6A	Bit	7	6	5	4	3	2	1	0
			DDR1$_7$	DDR1$_6$	DDR1$_5$	DDR1$_4$	DDR1$_3$	DDR1$_2$	DDR1$_1$	DDR1$_0$
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	W	W	W	W	W	W	W	W
Port 2 data direction register: DDR2	6B	Bit	7	6	5	4	3	2	1	0
			—	—	DDR2$_5$	DDR2$_4$	DDR2$_3$	DDR2$_2$	DDR2$_1$	DDR2$_0$
		Initial value	—	—	0	0	0	0	0	0
		Read/Write	—	—	W	W	W	W	W	W
Port 3 data direction register: DDR3	6C	Bit	7	6	5	4	3	2	1	0
			DDR3$_7$	DDR3$_6$	DDR3$_5$	DDR3$_4$	DDR3$_3$	DDR3$_2$	DDR3$_1$	DDR3$_0$
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	W	W	W	W	W	W	W	W
I/O port control register 1: IOPCR1	6D	Bit	7	6	5	4	3	2	1	0
			DREQ$_0$E	TEND$_0$E	P2$_6$E	BUSE	AOUTE	TOUT$_1$E	SCK$_1$E	SCK$_0$E
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
I/O port control register 2: IOPCR2	6E	Bit	7	6	5	4	3	2	1	0
			—	P3$_7$C1	P3$_7$C0	INT$_2$E	INT$_0$E	P3$_3$C1	P3$_3$C0	TEND$_1$E
		Initial value	1	0	0	0	0	0	0	0
		Read/Write	—	R/W	R/W	R/W	R/W	R/W	R/W	R/W

Register	Address	Remarks								
EEPROM control register 1: EEC1	70	Bit	7	6	5	4	3	2	1	0
			BUSY	BYTE/ PAGE	PERM	—	—	—	—	—
		Initial value	0	0	0	1	1	1	1	1
		Read/Write	R	R/W	R/W	—	—	—	—	—
EEPROM control register 2: EEC2	71	Bit	7	6	5	4	3	2	1	0
			PW	—	—	—	—	—	—	—
		Initial value	0	1	1	1	1	1	1	1
		Read/Write	W	—	—	—	—	—	—	—
Memory relocate register: MRR	72	Bit	7	6	5	4	3	2	1	0
			RMR3	RMR2	RMR1	RMR0	EPR3	EPR2	EPR1	EPR0
		Initial value	0	0	0	0	0	0	0	0
		Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W
System control register: SYSCR	7F	Bit	7	6	5	4	3	2	1	0
			RAME	—	—	—	STBYE	CKC2	CKC1	CKC0
		Initial value	0	1	1	1	0	0	0	0
		Read/Write	R/W	—	—	—	R/W	R/W	R/W	R/W