8-BIT N-CHANNEL MICROPROCESSOR COMPLETELY Z80™ COMPATIBLE

DESCRIPTION

The μPD780 and μPD780-1 processors are single-chip microprocessors developed from third-generation technology. Their increased computational power produces higher system through-put and more efficient memory utilization, surpassing that of any second-generation microprocessor. The single voltage requirement of the μPD780 and μPD780-1 processors makes it easy to implement them into a system. All output signals are fully decoded and timed to either standard memory or peripheral circuits. An N-channel, ion-implanted, silicon gate MOS process is utilized in implementing the circuit.

The block diagram shows the functions of the processor and details the internal register structure. The structure contains 26 bytes of Read/Write (R/W) memory available to the programmer. Included in the registers are two sets of six general purpose registers, which may be used individually as 8-bit registers, or as 6-bit register pairs. Also included are two sets of accumulator and flag registers.

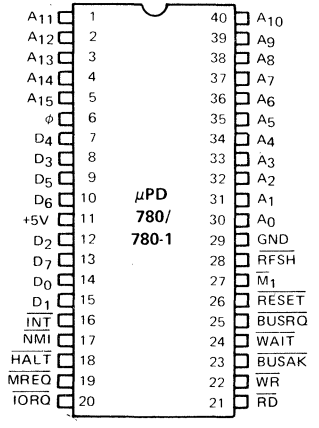
Through a group of exchange instructions the programmer has access to either set of main or alternate registers. The alternate register permits foreground/background mode of operation, or may be used for fast interrupt response. A 16-bit stack pointer is also included in each processor, simplifying implementation of multiple level interrupts, permitting unlimited subroutine nesting, and simplifying many types of data handling.

The two 16-bit index registers simplify implementation of relocatable code and manipulation of tabular data. The refresh register automatically refreshes external dynamic memories. A powerful interrupt response mode uses the I register to form the upper 8 bits of a pointer to an interrupt service address table, while the interrupting apparatus supplies the lower 8 bits of the pointer. An indirect call will then be made to service this address.

FEATURES

- Single Chip, N-Channel Silicon Gate Processor
- 158 Instructions – Including all 78 of the 8080A Instructions, Permitting Total Software Compatibility
- New 4-, 8-, and 16-Bit Operations Featuring Useful Addressing Modes such as Indexed, Bit and Relative
- 17 Internal Registers
- Three Modes of Rapid Interrupt Response, and One Non-Maskable Interrupt
- Directly Connects Standard Speed Dynamic or Static Memories, with Minimum Support Circuitry
- Single-Phase +5 Volt Clock and 5 VDC Supply
- TTL Compatibility
- Automatic Dynamic RAM Refresh Circuitry
- Available in Plastic Package

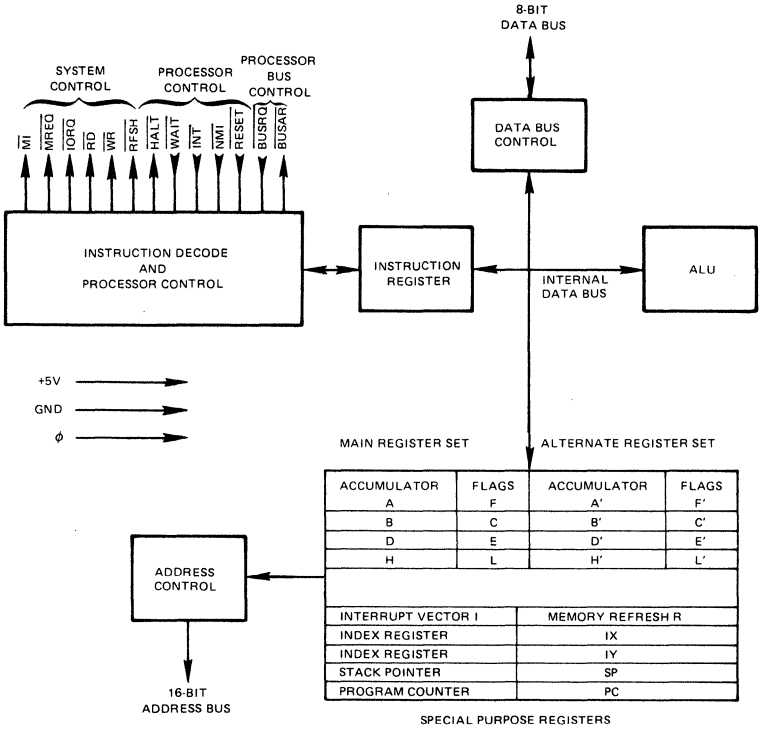
PIN CONFIGURATION



8

TM: Z80 is a registered trademark of Zilog, Inc.

BLOCK DIAGRAM



PIN IDENTIFICATION

PIN NO.	PIN SYMBOL	PIN NAME	FUNCTION
1-5, 30-40	A_0-A_{15}	Address Bus	3-State Output, active high. Pins A_0-A_{15} constitute a 16-bit address bus, which provides the address for memory and I/O device data exchanges. Memory capacity 65,536 bytes. A_0-A_7 is also needed as refresh cycle.
7-10, 12-15	D_0-D_7	Data Bus	3-State input/output, active high. Pins D_0-D_7 compose an 8-bit, bidirectional data bus, used for data exchanges with memory and I/O devices.
27	$\overline{M}_1$	Machine Cycle One	Output, active low. $\overline{M}_1$ indicates that the machine cycle in operation is the op code fetch cycle of an instruction execution.
19	$\overline{MREQ}$	Memory Request	3-State output, active low. $\overline{MREQ}$ indicates that a valid address for a memory read or write operation is held in the address.
20	$\overline{IORQ}$	Input/Output Request	3-State output, active low. The I/O request signal indicates that the lower half of the address bus holds a valid address for an I/O read or write operation. The $\overline{IORQ}$ signal is also used to acknowledge an interrupt command, indicating that an interrupt response vector can be placed on the data bus.
21	$\overline{RD}$	Memory Read	3-State output, active low. $\overline{RD}$ indicates that the processor is requesting data from memory or an I/O device. The memory or I/O device being addressed should use this signal to gate data onto the data bus.

PIN IDENTIFICATION (CONT.)

PIN NO.	PIN SYMBOL	PIN NAME	FUNCTION
22	$\overline{WR}$	Memory Write	3-State output, active low. The memory write signal indicates that the processor data bus is holding valid data to be stored in the addressed, memory or I/O device.
28	$\overline{RFSH}$	Refresh	Output, active low. $\overline{RFSH}$ indicates that a refresh address for dynamic memories is being held in the lower 7-bits of the address bus. The $\overline{MREQ}$ signal should be used to implement a refresh read to all dynamic memories.
18	$\overline{HALT}$	Halt State	Output, active low. $\overline{HALT}$ indicates that the processor has executed a HALT software instruction, and will not resume operation until either a non-maskable or a maskable (with mask enabled) interrupt has been implemented. The processor will execute NOP's while halted, to maintain memory refresh activity.
24	$\overline{WAIT}$	Wait	Input, active low. $\overline{WAIT}$ indicates to the processor that the memory or I/O devices being addressed are not ready for a data transfer. As long as this signal is active, the processor will reenter wait states.
16	$\overline{INT}$	Interrupt Request	Input, active low. The $\overline{INT}$ signal is produced by I/O devices. The request will be honored upon completion of the current instruction, if the interrupt enable flip-flop (IFF) is enabled by the internal software. There are three modes of interrupt response. Mode 0 is identical to 8080 interrupt response mode. The Mode 1 response is a restart location at 0038_H. Mode 2 is for simple vectoring to an interrupt service routine anywhere in memory.
17	$\overline{NMI}$	Non-Maskable Interrupt	Input, active low. The non-maskable interrupt has a higher priority than $\overline{INT}$. It is always acknowledged at the end of the current instruction, regardless of the status of the interrupt enable flip-flop. When the $\overline{NMI}$ signal is given, the μPD780 processor automatically restarts to location 0066_H.
26	$\overline{RESET}$	Reset	Input, active low. The $\overline{RESET}$ signal causes the processor to reset the interrupt enable flip-flop (IFF), clear PC and I and R registers, and set interrupt to 8080A mode. During the reset time, the address bus and data bus go to a state of high impedance, and all control output signals become inactive, after which processing continues at 0000H.
25	$\overline{BUSRQ}$	Bus Request	Input, active low. $\overline{BUSRQ}$ has a higher priority than $\overline{NMI}$, and is always honored at the end of the current machine cycle. It is used to allow other devices to take control over the processor address bus, data bus signals; by requesting that they go to a state of high impedance.
23	$\overline{BUSAK}$	Bus Acknowledge	Output, active low. $\overline{BUSAK}$ is used to inform the requesting device that the processor address bus, data bus and 3-state control bus signals have entered a state of high impedance, and the external device can now take control of these signals.

ABSOLUTE MAXIMUM RATINGS*

Operating Temperature	0°C to +70°C
Storage Temperature	-65°C to +150°C
Voltage on any Pin	-0.3 to +7 Volts (1)
Power Dissipation	1.5W

Note: (1) With Respect to Ground.

COMMENT: Stress above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

*T_a = 25°C

DC CHARACTERISTICS

T_a = 0°C to +70°C; V_{CC} = +5V ± 5% unless otherwise specified.

PARAMETER		SYMBOL	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT	TEST CONDITIONS
Clock Input Low Voltage		V_{ILC}	-0.3		0.45	V	
Clock Input High Voltage		V_{IHC}	V_{CC}-0.6		V_{CC}+0.3	V	
Input Low Voltage		V_{IL}	-0.3		0.8	V	
Input High Voltage		V_{IH}	2.0		V_{CC}	V	
Output Low Voltage		V_{OL}			0.4	V	I_{OL} = 1.8 mA
Output High Voltage		V_{OH}	2.4			V	I_{OH} = -250 μA
Power Supply Current	μPD780	I_{CC}			150	mA	t_c = 400 ns
	μPD780-1	I_{CC}		90	200	mA	t_c = 250 ns
Input Leakage Current		I_{LI}			10	μA	V_{IN} = 0 to V_{CC}
Tri-State Output Leakage Current in Float		I_{LOH}			10	μA	V_{OUT} = 2.4 to V_{CC}
Tri-State Output Leakage Current in Float		I_{LOL}			-10	μA	V_{OUT} = 0.4 V
Data Bus Leakage Current in Input Mode		I_{LD}			±10	μA	0 < V_{IN} < V_{CC}

CAPACITANCE

T_a = 25°C

PARAMETER	SYMBOL	LIMITS MIN	LIMITS TYP	LIMITS MAX	UNIT	TEST CONDITIONS
Clock Capacitance	C_ϕ			35	pF	f_c = 1 MHz
Input Capacitance	C_{IN}			5	pF	Unmeasured Pins
Output Capacitance	C_{OUT}			10	pF	Returned to Ground

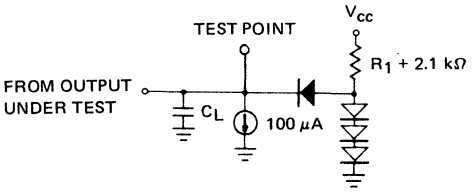
AC CHARACTERISTICS

T_a = 0°C to +70°C; V_{CC} = +5V ± 5%, unless otherwise specified.

PARAMETER	SYMBOL	LIMITS μPD780 MIN	LIMITS μPD780 MAX	LIMITS μPD780-1 MIN	LIMITS μPD780-1 MAX	UNIT	TEST CONDITIONS
Clock Period	t_c	0.4	⑫	0.25	⑫	μs	
Clock Pulse Width, Clock High	$t_{w(\Phi H)}$	180		110		ns	
Clock Pulse Width, Clock Low	$t_{w(\Phi L)}$	180	2000	110	2000	ns	
Clock Rise and Fall Time	$t_{r,f}$		30		30	ns	
Address Output Delay	$t_{D(AD)}$		145		110	ns	
Delay to Float	$t_{F(AD)}$		110		90	ns	
Address Stable Prior to $\overline{MREQ}$ (Memory Cycle)	t_{acm}	①		①		ns	C_L 50 pF
Address Stable Prior to $\overline{IORQ}$, $\overline{RD}$ or $\overline{WR}$ (I/O Cycle)	t_{aci}	②		②		ns	
Address Stable from $\overline{RD}$ or $\overline{WR}$	t_{ca}	③		③		ns	
Address Stable from $\overline{RD}$ or $\overline{WR}$ During Float	t_{caf}	④		④		ns	
Data Output Delay	$t_{D(D)}$		230		150	ns	
Delay to Float During Write Cycle	$t_{F(D)}$		90		90	ns	
Data Setup Time to Rising Edge of Clock During M1 Cycle	$t_{S\Phi(D)}$	50		35		ns	
Data Setup Time to Falling Edge of Clock During M2 to M5 Cycles	$t_{S\bar{\Phi}(D)}$	60		50		ns	C_L = 200 pF
Data Stable Prior to $\overline{WR}$ (Memory Cycle)	t_{dcm}	⑤		⑤		ns	
Data Stable Prior to $\overline{WR}$ (I/O Cycle)	t_{dci}	⑥		⑥		ns	
Data Stable from $\overline{WR}$	t_{cdf}	⑦		⑦		ns	
Any Hold Time for Setup Time	t_H	0			0	ns	
$\overline{MREQ}$ Delay from Falling Edge of Clock to $\overline{MREQ}$ Low	$t_{DL\bar{\Phi}(MR)}$		100		85	ns	
$\overline{MREQ}$ Delay from Rising Edge of Clock to $\overline{MREQ}$ High	$t_{DH\Phi(MR)}$		100		85	ns	
$\overline{MREQ}$ Delay from Falling Edge of Clock to $\overline{MREQ}$ High	$t_{DH\bar{\Phi}(MR)}$		100		85	ns	
Pulse Width, $\overline{MREQ}$ Low	$t_{w(\overline{MRL})}$	⑧		⑧		ns	
Pulse Width, $\overline{MREQ}$ High	$t_{w(\overline{MRH})}$	⑨		⑨		ns	
$\overline{IORQ}$ Delay from Rising Edge of Clock to $\overline{IORQ}$ Low	$t_{DL\Phi(IR)}$		90		75	ns	
$\overline{IORQ}$ Delay from Falling Edge of Clock to $\overline{IORQ}$ Low	$t_{DL\bar{\Phi}(IR)}$		110		85	ns	
$\overline{IORQ}$ Delay from Rising Edge of Clock to $\overline{IORQ}$ High	$t_{DH\Phi(IR)}$		100		85	ns	
$\overline{IORQ}$ Delay from Falling Edge of Clock to $\overline{IORQ}$ High	$t_{DH\bar{\Phi}(IR)}$		110		85	ns	C_L = 50 pF
$\overline{RD}$ Delay from Rising Edge of Clock to $\overline{RD}$ Low	$t_{DL\Phi(RD)}$		100		85	ns	
$\overline{RD}$ Delay from Falling Edge of Clock to $\overline{RD}$ Low	$t_{DL\bar{\Phi}(RD)}$		130		95	ns	
$\overline{RD}$ Delay from Rising Edge of Clock to $\overline{RD}$ High	$t_{DH\Phi(RD)}$		100		85	ns	
$\overline{RD}$ Delay from Falling Edge of Clock to $\overline{RD}$ High	$t_{DH\bar{\Phi}(RD)}$		110		85	ns	
$\overline{WR}$ Delay from Rising Edge of Clock to $\overline{WR}$ Low	$t_{DL\Phi(WR)}$		80		65	ns	
$\overline{WR}$ Delay from Falling Edge of Clock to $\overline{WR}$ Low	$t_{DL\bar{\Phi}(WR)}$		90		80	ns	
$\overline{WR}$ Delay from Falling Edge of Clock to $\overline{WR}$ High	$t_{DH\Phi(WR)}$		100		80	ns	
Pulse Width to $\overline{WR}$ Low	$t_{w(\overline{WRL})}$	⑩		⑩		ns	
$\overline{MI}$ Delay from Rising Edge of Clock to $\overline{MI}$ Low	$t_{DL(MI)}$		120		100	ns	C_L = 30 pF
$\overline{MI}$ Delay from Rising Edge of Clock to $\overline{MI}$ High	$t_{DH(MI)}$		130		100	ns	
$\overline{RFSH}$ Delay from Rising Edge of Clock to $\overline{RFSH}$ Low	$t_{DL(RF)}$		180		130	ns	
$\overline{RFSH}$ Delay from Rising Edge of Clock to $\overline{RFSH}$ High	$t_{DH(RF)}$		150		120	ns	
$\overline{WAIT}$ Setup Time to Falling Edge of Clock	$t_{s(WT)}$	70		70		ns	
$\overline{HALT}$ Delay Time from Falling Edge of Clock	$t_{D(HT)}$		300		300	ns	C_L = 50 pF
$\overline{INT}$ Setup Time to Rising Edge of Clock	$t_{s(IT)}$	80		80		ns	
Pulse Width, $\overline{NMI}$ Low	$t_{w(\overline{NML})}$	80		80		ns	
$\overline{BUSRQ}$ Setup Time to Rising Edge of Clock	$t_{s(BQ)}$	80		50		ns	
$\overline{BUSAK}$ Delay from Rising Edge of Clock to $\overline{BUSAK}$ Low	$t_{DL(BA)}$		120		100	ns	C_L = 50 pF
$\overline{BUSAK}$ Delay from Falling Edge of Clock to $\overline{BUSAK}$ High	$t_{DH(BA)}$		110		100	ns	
$\overline{RESET}$ Setup Time to Rising Edge of Clock	$t_{s(RS)}$	90		60		ns	
Delay to Float ($\overline{MREQ}$, $\overline{IORQ}$, $\overline{RD}$ and $\overline{WR}$)	$t_{F(C)}$		100		80	ns	
$\overline{MI}$ Stable Prior to $\overline{IORQ}$ (Interrupt Ack.)	t_{mr}	⑪		⑪		ns	

Notes:
① $t_{acm} = t_{w(\Phi H)} + t_f - 65$ (75)*
② $t_{aci} = t_c - 70$ (80)*
③ $t_{ca} = t_{w(\Phi L)} + t_r - 50$ (40)*
④ $t_{caf} = t_{w(\Phi L)} + t_r - 45$ (60)*
⑤ $t_{dcm} = t_c - 170$ (210)*
⑥ $t_{dci} = t_{w(\Phi L)} + t_r - 170$ (210)*
⑦ $t_{cdf} = t_{w(\Phi L)} + t_r - 70$ (80)*
⑧ $t_{w(\overline{MRL})} = t_c - 30$ (40)*
⑨ $t_{w(\overline{MRH})} = t_{w(\Phi H)} + t_f - 20$ (30)*
⑩ $t_{w(\overline{WR})} = t_c - 30$ (40)*
⑪ $t_{mr} = 2t_c + t_{w(\Phi H)} + t_f - 65$ (80)*
⑫ $t_c = t_{w(\Phi H)} + t_{w(\Phi L)} + t_r + t_f$

*These values apply to the μPD780.

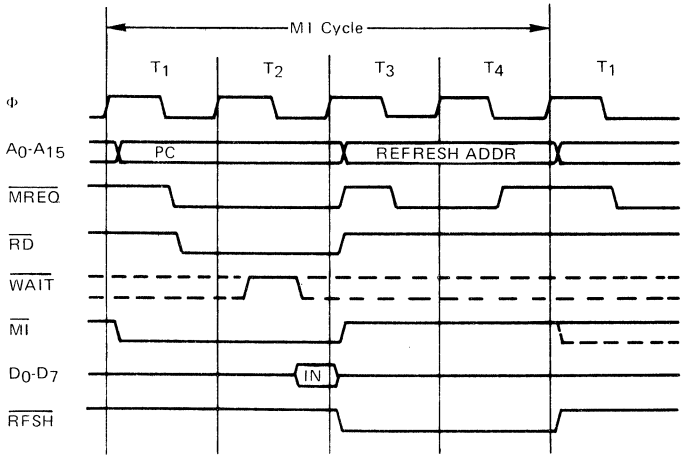


LOAD CIRCUIT FOR OUTPUT

8

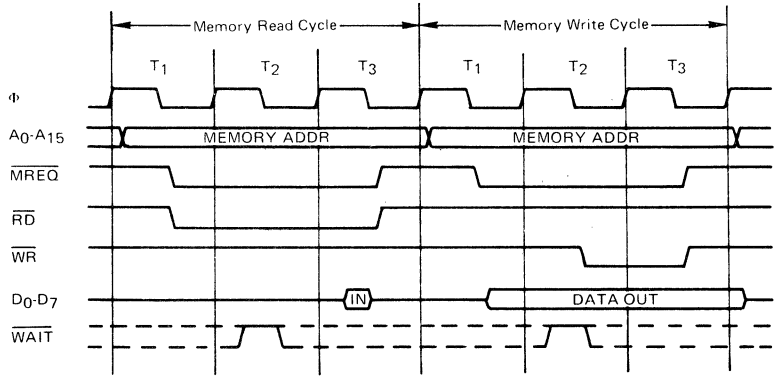
Instruction Op Code Fetch

The contents of the program counter (PC) are placed on the address bus at the start of the cycle. $\overline{MREQ}$ goes active one-half clock cycle later, and the falling edge of this signal can be used directly as a chip enable to dynamic memories. The memory data should be enabled onto the processor data bus when $\overline{RD}$ goes active. The processor takes data with the rising edge of the clock state T_3. The processor internally decodes and executes the instruction, while clock states T_3 and T_4 of the fetch cycle are used to refresh dynamic memories. The refresh control signal $\overline{RFSH}$ indicates that a refresh read should be done to all dynamic memories.



Memory Read or Write Cycles

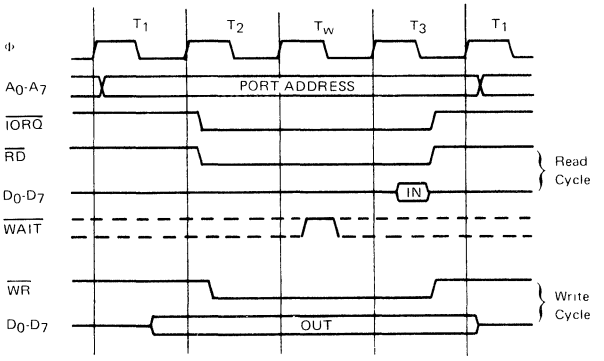
This diagram illustrates the timing of memory read or write cycles other than an op code fetch (M_1 cycle). The function of the $\overline{MREQ}$ and $\overline{RD}$ signals is exactly the same as in the op code fetch cycle. When a memory write cycle is implemented, the $\overline{MREQ}$ becomes active and is used directly as a chip enable for dynamic memories, when the address bus is stable. The $\overline{WR}$ line is used directly as a R/W pulse to any type of semiconductor memory, and is active when data on the data bus is stable.



TIMING WAVEFORMS (CONT.)

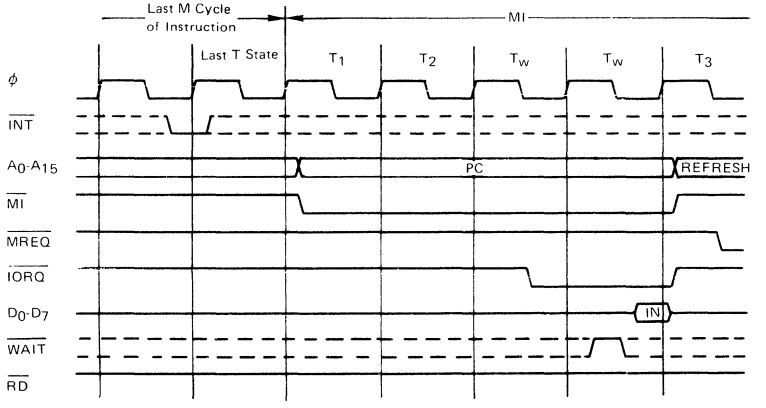
Input or Output Cycles

This illustrates the timing for an I/O read or I/O write operation. A single wait-state (T_W) is automatically inserted in I/O operations to allow sufficient time for an I/O port to decode its address and activate the $\overline{WAIT}$ line, if necessary.



Interrupt Request/Acknowledge Cycle

The processor samples the interrupt signal with the rising edge of the last clock at the end of any instruction. A special M_1 cycle is started when an interrupt is accepted. During the M_1 cycle, the $\overline{IORQ}$ (instead of $\overline{MREQ}$) signal becomes active, indicating that the interrupting device can put an 8-bit vector on the data bus. Two wait states (T_W) are automatically added to this cycle. This makes it easy to implement a ripple priority interrupt scheme.



INSTRUCTION SET

The following summary shows the assembly language mnemonic and the symbolic operation performed by the instructions of the μPD780 and μPD780-1 processors. The instructions are divided into 16 categories:

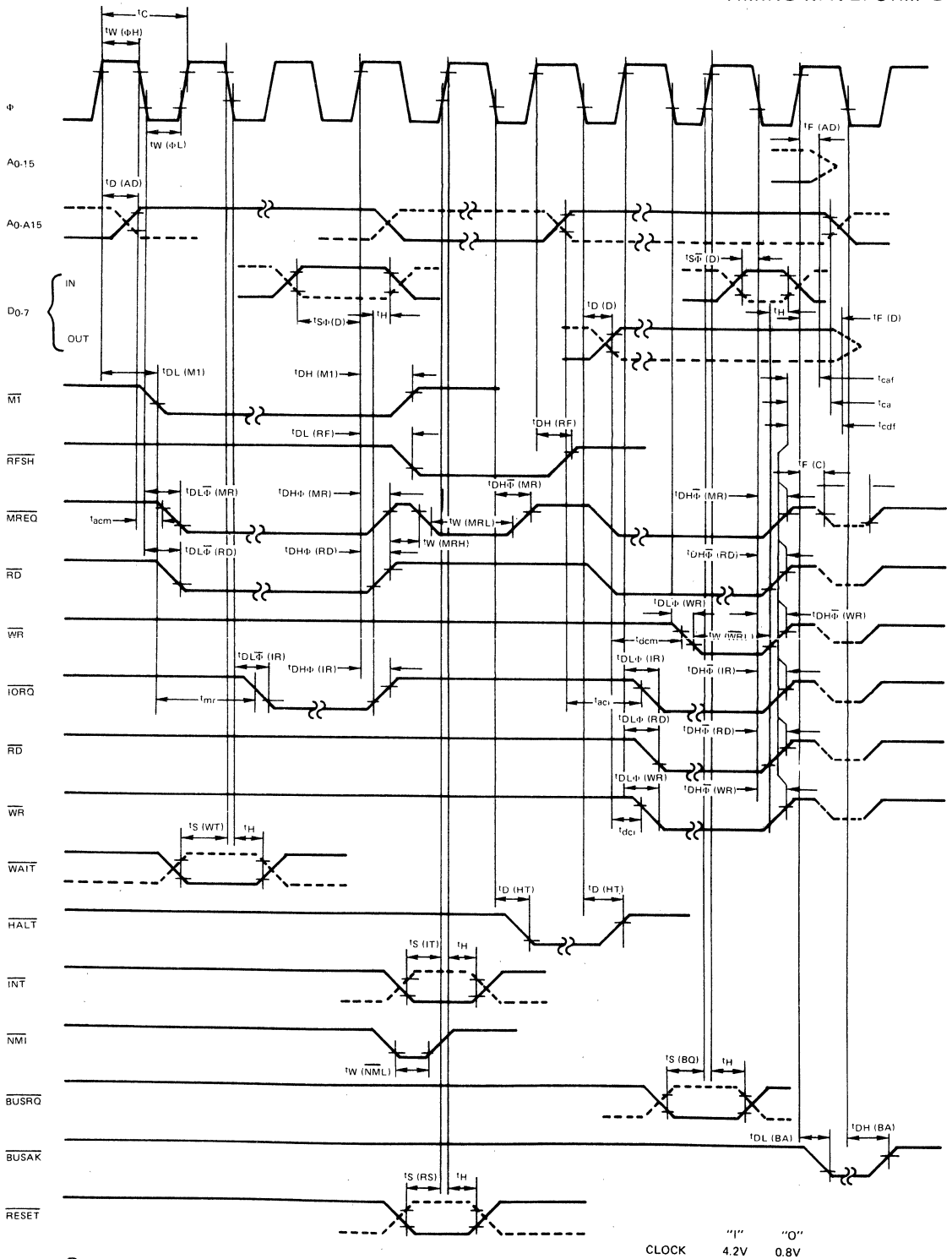
Miscellaneous Group	8-Bit Loads
Rotates and Shifts	16-Bit Loads
Bit Set, Reset and Test	Exchanges
Input and Output	Memory Block Moves
Jumps	Memory Block Searches
Calls	8-Bit Arithmetic and Logic
Restarts	16-Bit Arithmetic
Returns	General Purpose Accumulator and Flag Operations

The addressing Modes include combinations of the following:

Indexed	Immediate
Register	Immediate Extended
Implied	Modified Page Zero
Register Indirect	Relative
Bit	Extended

8

TIMING WAVEFORM ①



	"I"	"O"
CLOCK	4.2V	0.8V
OUTPUT	2.0V	0.8V
INPUT	2.0V	0.8V
FLOAT	ΔV	±0.5V

Note: ① Timing measurements are made at the following voltages unless otherwise specified:

INSTRUCTION SET TABLE

MNEMONIC	SYMBOLIC OPERATION	DESCRIPTION	NO. BYTES	NO. T STATES	FLAGS C	Z	P/V	S	N	H	OP CODE 76 543 210
ADC HL, ss	HL ← HL + ss + CY	Add with carry reg. pair ss to HL	1	11	↕	↕	V	↕	0	X	11 101 101 Ⓐ 01 ss1 010
ADC A, r	A ← A + r + CY	Add with carry Reg. r to ACC	1	4	↕	↕	V	↕	0	↕	10 001 rrr Ⓑ
ADC A,n	A ← A + n + CY	Add with carry value n to ACC		7	↕	↕	V	↕	0	↕	11 001 110 nn nnn nnn
ADC A, (HL)	A ← A + (HL) + CY	Add with carry loc. (HL) to ACC		7	↕	↕	V	↕	0	↕	10 001 110
ADC A, (IX + d)	A ← A + (IX + d) + CY	Add with carry loc. (IX + d) to ACC		19	↕	↕	V	↕	0	↕	11 011 101 10 001 110 dd ddd ddd
ADC A, (IY + d)	A ← A + (IY + d) + CY	Add with carry loc. (IY + d) to ACC		19	↕	↕	V	↕	0	↕	11 111 101 10 001 110 dd ddd ddd
ADD A, n	A ← A + n	Add value n to ACC	2	7	↕	↕	V	↕	0	↕	11 000 110 nn nnn nnn
ADD A, r	A ← A + r	Add Reg. r to ACC	1	4	↕	↕	V	↕	0	↕	10 000 rrr Ⓑ
ADD A, (HL)	A ← A + (HL)	Add location (HL) to ACC	1	7	↕	↕	V	↕	0	↕	10 000 110
ADD A, (IX + d)	A ← A + (IX + d)	Add location (IX + d) to ACC	3	19	↕	↕	V	↕	0	↕	11 011 101 10 000 110 dd ddd ddd
ADD A, (IY + d)	A ← A + (IY + d)	Add location (IY + d) to ACC	3	19	↕	↕	V	↕	0	↕	11 111 101 10 000 110 dd ddd ddd
ADD HL, ss	HL ← HL + ss	Add Reg. pair ss to HL	1	11	↕	•	•	•	0	X	00 ss1 001 Ⓐ
ADD IX, pp	IX ← IX + pp	Add Reg. pair pp to IX	2	15	↕	•	•	•	0	X	11 011 101 Ⓒ 00 pp1 001
ADD IY, rr	IY ← IY + rr	Add Reg. pair rr to IY	2	15	↕	•	•	•	0	X	11 111 101 Ⓓ 00 rr1 001
AND r	A ← A∧r	Logical 'AND' of Reg. r ∧ ACC		4	0	↕	P	↕	0	↕	10 100 rrr Ⓑ
AND n	A ← A∧n	Logical 'AND' of value n ∧ ACC		7	0	↕	P	↕	0	↕	11 100 110 nn nnn nnn
AND (HL)	A ← A∧(HL)	Logical 'AND' of loc. (HL) ∧ ACC		7	0	↕	P	↕	0	↕	10 100 110
AND (IX + d)	A ← A∧(IX + d)	Logical 'AND' of loc. (IX + d) ∧ ACC		19	0	↕	P	↕	0	↕	11 011 101 10 100 110 dd ddd ddd
AND (IY + d)	A ← A∧(IY + d)	Logical 'AND' of loc. (IY + d) ∧ ACC		19	0	↕	P	↕	0	↕	11 111 101 10 100 110 dd ddd ddd
BIT b, (HL)	Z ← $\overline{(HL)}_b$	Test BIT b of location (HL)	2	12	•	↕	X	X	0	1	11 001 011 Ⓔ 01 bbb 110
BIT b, (IX + d)	Z ← $\overline{(IX + d)}_b$	Test BIT b at location (IX + d)	4	20	•	↕	X	X	0	1	11 011 101 Ⓔ 11 001 011 dd ddd ddd 01 bbb 110
BIT b, (IY + d)	Z ← $\overline{(IY + d)}_b$	Test BIT b at location (IY + d)	4	20	•	↕	X	X	0	1	11 111 101 Ⓔ 11 001 011 dd ddd ddd 01 bbb 110
BIT b, r	Z ← $\overline{r}_b$	Test BIT of Reg. r	2	8	•	↕	X	X	0	1	11 001 011 Ⓑ Ⓔ 01 bbb rrr
CALL cc, nn	If condition cc false continue, else same as CALL nn	Call subroutine at location nn if condition cc is true	3	10	•	•	•	•	•	•	11 ←cc→ 100 Ⓗ nn nnn nnn nn nnn nnn
CALL nn	(SP − 1) ← PC_H (SP − 2) ← PC_L PC ← nn	Unconditional call subroutine at location nn	3	17	•	•	•	•	•	•	11 001 101 nn nnn nnn nn nnn nnn
CCF	CY ← $\overline{CY}$	Complement carry flag	1	4	↕	•	•	•	0	X	00 111 111
CP r	A − r	Compare Reg. r with ACC		4	↕	↕	V	↕	1	↕	10 111 rrr Ⓑ
CP n	A − n	Compare value n with ACC		7	↕	↕	V	↕	1	↕	11 111 110 nn nnn nnn
CP (HL)	A − (HL)	Compare loc. (HL) with ACC		7	↕	↕	V	↕	1	↕	10 111 110
CP (IX + d)	A − (IX + d)	Compare loc. (IX + d) with ACC		19	↕	↕	V	↕	1	↕	11 011 101 10 111 110 dd ddd ddd
CP (IY + d)		Compare loc. (IY + d) with ACC		19	↕	↕	V	↕	1	↕	11 111 101 10 111 110 dd ddd ddd
CPD	A − (HL) HL ← HL − 1 BC ← BC − 1	Compare location (HL) and ACC, decrement HL and BC	2	16	•	↕②	↕①	↕	1	↕	11 101 101 10 101 001
CPDR	A − (HL) HL ← HL − 1 BC ← BC − 1 until A = (HL) or BC = 0	Compare location (HL) and ACC, decrement HL and BC, repeat until BC = 0	2	21 if BC = 0 and A ≠ (HL) 16 if BC = 0 or A = (HL)	•	↕②	↕①	↕	1	↕	11 101 101 10 111 001

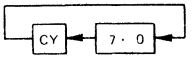
8

MNEMONIC	SYMBOLIC OPERATION	DESCRIPTION	NO. BYTES	NO. T STATES	FLAGS C	Z	P/V	S	N	H	OP CODE 76 543 210
CPI	A – (HL) HL ← HL + 1 BC ← BC – 1	Compare location (HL) and ACC, increment HL and decrement BC	2	16	•	↕②	↕①	↕	1	↕	11 101 101 10 100 001
CPIR	A – (HL) HL ← HL + 1 BC ← BC – 1 until A = (HL) or BC = 0	Compare location (HL) and ACC, increment HL, decrement BC Repeat until BC = C	2	21 if BC = 0 and A ≠ (HL) 16 if BC = 0 or A = (HL)	•	↕②	↕①	↕	1	↕	11 101 101 10 110 001
CPL	A ← A	Complement ACC (1's comp.)	1	4	•	•	•	•	1	1	00 101 111
DAA		Decimal adjust ACC	1	4	↕	↕	P	↕	•	↕	00 100 111
DEC r	r ← r – 1	Decrement Reg. r		4	•	↕	V	↕	1	↕	00 rrr 101Ⓑ
DEC (HL)	(HL) ← (HL) – 1	Decrement loc. (HL)		11	•	↕	V	↕	1	↕	00 110 101
DEC (IX + d)	(IX + d) ← (IX + d) – 1	Decrement loc. (IX + d)		23	•	↕	V	↕	1	↕	11 011 101 00 110 101 dd ddd ddd
DEC (IY + d)	(IY + d) ← (IY + d) – 1	Decrement loc. (IY + d)		23	•	↕	V	↕	1	↕	11 111 101 00 110 101 dd ddd ddd
DEC IX	IX ← IX – 1	Decrement IX	2	10	•	•	•	•	•	•	11 011 101 00 101 011
DEC IY	IY ← IY – 1	Decrement IY	2	10	•	•	•	•	•	•	11 111 101 00 101 011
DEC ss	ss ← ss – 1	Decrement Reg. pair ss	1	6	•	•	•	•	•	•	00 ss1 011Ⓐ
DI	IFF ← 0	Disable interrupts	1	4	•	•	•	•	•	•	11 110 011
DJNZ, e	B ← B – 1 if B = 0 continue if B ≠ 0 PC ← PC + e	Decrement B and jump relative if B = 0	2	8	•	•	•	•	•	•	00 010 000 ◄—e-2—►
EI	IFF ← 1	Enable interrupts	1	4	•	•	•	•	•	•	11 111 011
EX (SP), HL	H ↔ (SP + 1) L ↔ (SP)	Exchange the location (SP) and HL	1	19	•	•	•	•	•	•	11 100 011
EX (SP), IX	IX_H ↔ (SP + 1) IX_L ↔ (SP)	Exchange the location (SP) and IX	2	23	•	•	•	•	•	•	11 011 101 11 100 011
EX (SP), IY	IY_H ↔ (SP + 1) IY_L ↔ (SP)	Exchange the location (SP) and IY	2	23	•	•	•	•	•	•	11 111 101 11 100 011
EX AF, AF'	AF ↔ AF'	Exchange the contents of AF, AF'	1	4	•	•	•	•	•	•	00 001 000
EX DE, HL	DE ↔ HL	Exchange the contents of DE and HL	1	4	•	•	•	•	•	•	11 101 011
EXX	BC ↔ BC' DE ↔ DE' HL ↔ HL'	Exchange the contents of BC, DE, HL with contents of BC', DE', HL', respectively	1	4	•	•	•	•	•	•	11 011 001
HALT	Processor Halted	HALT (wait for interrupt or reset)	1	4	•	•	•	•	•	•	01 110 110
IM 0		Set Interrupt mode 0	2	8	•	•	•	•	•	•	11 101 101 01 000 110
IM 1		Set Interrupt mode 1	2	8	•	•	•	•	•	•	11 101 101 01 010 110
IM 2		Set Interrupt mode 2	2	8	•	•	•	•	•	•	11 101 101 01 011 110
IN A, (n)	A ← (n)	Load ACC with input from device n	2	11	•	•	•	•	•	•	11 011 011 nn nnn nnn
IN r, (C)	r ← (C)	Load Reg. r with input from device (C)	2	12	•	↕	P	↕	0	↕	11 101 101Ⓘ 01 rrr 000
INC (HL)	(HL) ← (HL) + 1	Increment location (HL)	1	11	•	↕	V	↕	0	↕	00 110 100
INC IX	IX ← IX + 1	Increment IX	2	10	•	•	•	•	•	•	11 011 101 00 100 011
INC (IX + d)	(IX + d) ← (IX + d) + 1	Increment location (IX + d)	3	23	•	↕	V	↕	0	↕	11 011 101 00 110 100 dd ddd ddd
INC IY	IY ← IY + 1	Increment IY	2	10	•	•	•	•	•	•	11 111 101 00 100 011
INC (IY + d)	(IY + d) ← (IY + d) + 1	Increment location (IY + d)	3	23	•	↕	V	↕	0	↕	11 111 101 00 110 100 dd ddd ddd
INC r	r ← r + 1	Increment Reg. r	1	4	•	↕	V	↕	0	↕	00 rrr 100Ⓑ
INC ss	ss ← ss + 1	Increment Reg. pair ss	1	6	•	•	•	•	•	•	00 ss0 011Ⓐ
IND	(HL) ← (C) B ← B – 1 HL ← HL – 1	Load location (HL) with input from port (C), decrement HL and B	2	16	•	↕③	X	X	1	X	11 101 101 10 101 010

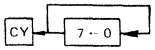
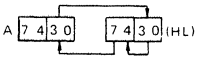
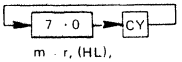
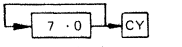
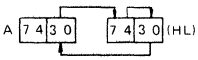
MNEMONIC	SYMBOLIC OPERATION	DESCRIPTION	NO. BYTES	NO. T STATES	FLAGS C	Z	P/V	S	N	H	OP CODE 76 543 210
INDR	(HL) ← (C) B ← B – 1 HL ← HL – 1 until B = 0	Load location (HL) with input from port (C), decrement HL and decrement B, repeat until B = 0	2	21	•	1	X	X	1	X	11 101 101 10 111 010
INI	(HL) ← (C) B ← B – 1 HL ← HL + 1	Load location (HL) with input from port (C); and increment HL and decrement B	2	16	•	↕③	X	X	1	X	11 101 101 10 100 010
INIR	(HL) ← (C) B ← B – 1 HL ← HL + 1 until B = 0	Load location (HL) with input from port (C), increment HL and decrement B, repeat until B = 0	2	21	•	1	X	X	1	X	11 101 101 10 110 010
JP (HL)	PC ← HL	Unconditional jump to (HL)	1	4	•	•	•	•	•	•	11 101 001
JP (IX)	PC ← IX	Unconditional jump to (IX)	2	8	•	•	•	•	•	•	11 011 101 11 101 001
JP (IY)	PC ← IY	Unconditional jump to (IY)	2	8	•	•	•	•	•	•	11 111 101 11 101 001
JP cc, nn	If cc true PC ← nn else continue	Jump to location nn if condition cc is true	3	10	•	•	•	•	•	•	11 ←cc→ 010Ⓗ nn nnn nnn nn nnn nnn
JP nn	PC ← nn	Unconditional jump to location nn	3	10	•	•	•	•	•	•	11 000 011 nn nnn nnn nn nnn nnn
JR C, e	If C = 0 continue If C = 1 PC ← PC + e	Jump relative to PC + e, if carry = 1	2	7 if condition met. 12, if not	•	•	•	•	•	•	00 111 000 ←e-2→
JR e	PC ← PC + e	Unconditional jump relative to PC + e	2	12	•	•	•	•	•	•	00 011 000 ←e-2→
JR NC, e	If C = 1 continue If C = 0 PC ← PC + e	Jump relative to PC + e if carry = 0	2	7	•	•	•	•	•	•	00 110 000 ←e-2→
JR NZ, e	If Z = 1 continue	Jump relative to PC + e if non-zero (Z = 0)	2	7	•	•	•	•	•	•	00 100 000 ←e-2→
JR Z, e	If Z = 0 continue	Jump relative to PC + e if zero (Z = 1)	2	7	•	•	•	•	•	•	00 101 000 ←e-2→
LD A, (BC)	A ← (BC)	Load ACC with location (BC)	1	7	•	•	•	•	•	•	00 001 010
LD A, (DE)	A ← (DE)	Load ACC with location (DE)	1	7	•	•	•	•	•	•	00 011 010
LD A, I	A ← I	Load ACC with I	2	9	•	↕	IFF	↕	0	0	11 101 101 01 010 111
LD A, (nn)	A ← (nn)	Load ACC with location nn	3	13	•	•	•	•	•	•	00 111 010 nn nnn nnn nn nnn nnn
LD A, R	A ← R	Load ACC with Reg. R	2	9	•	↕	IFF	↕	0	0	11 101 101 01 011 111
LD (BC), A	(BC) ← A	Load location (BC) with ACC	1	7	•	•	•	•	•	•	00 000 010
LD (DE), A	(DE) ← A	Load location (DE) with ACC	1	7	•	•	•	•	•	•	00 010 010
LD (HL), n	(HL) ← n	Load location (HL) with value n	2	10	•	•	•	•	•	•	00 110 110 nn nnn nnn
LD ss, nn	ss ← nn	Load Reg. pair ss with value nn	4	20	•	•	•	•	•	•	00 ss0 001Ⓐ nn nnn nnn nn nnn nnn
LD HL, (nn)	H ← (nn + 1) L ← (nn)	Load HL with location (nn)	3	16	•	•	•	•	•	•	00 101 010 nn nnn nnn nn nnn nnn
LD (HL), r	(HL) ← r	Load location (HL) with Reg. r	1	7	•	•	•	•	•	•	01 110 rrrⒷ
LD I, A	I ← A	Load I with ACC	2	9	•	•	•	•	•	•	11 101 101 01 000 111
LD IX, nn	IX ← nn	Load IX with value nn	4	19	•	•	•	•	•	•	11 011 101 00 100 001 nn nnn nnn nn nnn nnn
LD IX, (nn)	IX_H ← (nn + 1) IX_L ← (nn)	Load IX with location (nn)	4	20	•	•	•	•	•	•	11 011 101 00 101 010 nn nnn nnn nn nnn nnn
LD (IX + d), n	(IX + d) ← n	Load location (IX + d) with value n	4	19	•	•	•	•	•	•	11 011 101 00 110 110 dd ddd ddd nn nnn nnn
LD (IX + d), r	(IX + d) ← r	Load location (IX + d) with Reg. r	3	19	•	•	•	•	•	•	11 011 101Ⓑ 01 110 rrr dd ddd ddd

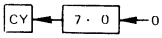
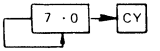
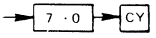
8

MNEMONIC	SYMBOLIC OPERATION	DESCRIPTION	NO. BYTES	NO. T STATES	FLAGS C	Z	P/V	S	N	H	OP CODE 76 543 210
LD IY, nn	IY ← nn	Load IY with value nn	4	14	•	•	•	•	•	•	11 111 101 00 100 001 nn nnn nnn nn nnn nnn
LD IY, (nn)	IY_H ← (nn + 1) IY_L ← (nn)	Load IY with location (nn)	4	20	•	•	•	•	•	•	11 111 101 00 101 010 nn nnn nnn nn nnn nnn
LD ss, (nn)	ss_H ← (nn + 1) ss_L ← (nn)	Load Reg. pair dd with location (nn)	4	20	•	•	•	•	•	•	11 101 101 Ⓐ 01 ss1 011 nn nnn nnn nn nnn nnn
LD (IY + d), n	(IY + d) ← n	Load (IY + d) with value n	4	19	•	•	•	•	•	•	11 111 101 00 110 110 dd ddd ddd nn nnn nnn
LD (IY + d), r	(IY + d) ← r	Load location (IY + d) with Reg r	3	19	•	•	•	•	•	•	11 111 101 Ⓑ 01 110 rrr dd ddd ddd
LD (nn), A	(nn) ← A	Load location (nn) with ACC	3	13	•	•	•	•	•	•	00 110 010 nn nnn nnn nn nnn nnn
LD (nn), ss	(nn + 1) ← ss_H (nn) ← ss_L	Load location (nn) with Reg. pair dd	4	20	•	•	•	•	•	•	11 101 101 Ⓐ 01 ss0 011 nn nnn nnn nn nnn nnn
LD (nn), HL	(nn + 1) ← H (nn) ← L	Load location (nn) with HL	3	16	•	•	•	•	•	•	00 100 010 nn nnn nnn nn nnn nnn
LD (nn), IX	(nn + 1) ← IX_H (nn) ← IX_L	Load location (nn) with IX	4	20	•	•	•	•	•	•	11 011 101 00 100 010 nn nnn nnn nn nnn nnn
LD (nn), IY	(nn + 1) ← IY_H (nn) ← IY_L	Load location (nn) with IY	4	20	•	•	•	•	•	•	11 111 101 00 100 010 nn nnn nnn nn nnn nnn
LD R, A	R ← A	Load R with ACC	2	9	•	•	•	•	•	•	11 101 101 01 001 111
LD r, (HL)	r ← (HL)	Load Reg. r with location (HL)	1	7	•	•	•	•	•	•	01 rrr 110 Ⓑ
LD r, (IX + d)	r ← (IX + d)	Load Reg. r with location (IX + d)	3	19	•	•	•	•	•	•	11 011 101 Ⓑ 01 rrr 110 dd ddd ddd
LD r, (IY + d)	r ← (IY + d)	Load Reg. r with location (IY + d)	3	19	•	•	•	•	•	•	11 111 101 Ⓑ 01 rrr 110 dd ddd ddd
LD r, n	r ← n	Load Reg. r with value n	2	7	•	•	•	•	•	•	00 rrr 110 Ⓑ nn nnn nnn
LD, r, r'	r ← r'	Load Reg. r with Reg. r	1	4	•	•	•	•	•	•	01 rrr rrr Ⓕ
LD SP, HL	SP ← HL	Load SP with HL	1	6	•	•	•	•	•	•	11 111 001
LD SP, IX	SP ← IX	Load SP with IX	2	10	•	•	•	•	•	•	11 011 101 11 111 001
LD SP, IY	SP ← IY	Load SP with IY	2	10	•	•	•	•	•	•	11 111 101 11 111 001
LDD	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1	Load location (DE) with location (HL), decrement DE, HL and BC	2	16	•	•	↕	•	0	0	11 101 101 10 101 000
LDDR	(DE) ← (HL) DE ← DE − 1 HL ← HL − 1 BC ← BC − 1 until BC = 0	Load location (DE) with location (HL)	2	21	•	•	0	•	0	0	11 101 101 10 111 000
LDI	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC − 1	Load location (DE) with location (HL), increment DE, HL, decrement BC	2	16	•	•	↕ Ⓣ	•	0	0	11 101 101 10 100 000
LDIR	(DE) ← (HL) DE ← DE + 1 HL ← HL + 1 BC ← BC − 1 until BC = 0	Load location (DE) with location (HL), increment DE, HL; decrement BC and repeat until BC = 0	2	21 if BC ≠ 0 16 if BC = 0	•	•	0	•	0	0	11 101 101 10 110 000
NEG	A ← 0 − A	Negate ACC (2's complement)	2	8	↕	↕	V	↕	1	↕	11 101 101 01 000 100

MNEMONIC	SYMBOLIC OPERATION	DESCRIPTION	NO. BYTES	NO. T STATES	FLAGS C	Z	P/V	S	N	H	OP CODE 76 543 210
NOP		No operation	1	4	•	•	•	•	•	•	00 000 000
OR r	A · AV r	Logical 'OR' of Reg. r and ACC		4	0	↕	P	↕	0	↕	10 110 rrr (B)
OR n	A · AV n	Logical 'OR' of value n and ACC		7	•	↕	P	↕	0	↕	11 110 110 nn nnn nnn
OR (HL)	A · AV (HL)	Logical 'OR' of loc. (HL) and ACC		7	•	↕	P	↕	0	↕	10 110 110
OR (IX + d)	A · (IX + d)	Logical 'OR' of loc. (IX + d) ∧ ACC		19	•	↕	P	↕	0	↕	11 011 101 10 110 110 dd ddd ddd
OR (IY + d)	A · AV (IY + d)	Logical 'OR' of loc. (IY + d) ∧ ACC		19	•	↕	P	↕	0	↕	11 111 101 10 110 110 dd ddd ddd
OTDR	(C) · (HL) B · B 1 HL · HL 1 until B = 0	Load output port (C) with contents of location (HL), decrement HL and B, repeat until B 0	2	21 if B ≠ 0 16 if B C	•	1	X	X	1	X	11 101 101 10 111 011
OTIR	(C) · (HL) B · B 1 HL · HL + 1 until B 0	Load output port (C) with location (HL), increment HL, decrement B, repeat until B 0	2	21 if B ≠ 0 16 if B C	•	1	X	X	1	X	11 101 101 10 110 011
OUT (C), r	(C) · r	Load output port (C) with Reg. r	2	12	•	•	•	•	•	•	11 101 101 (B) 01 rrr 001
OUT (n), A	(n) · A	Load output port (n) with ACC	2	11	•	•	•	•	•	•	11 010 011 nn nnn nnn
OUTD	(C) · (HL) B · B 1 HL · HL 1	Load output port (C) with location (HL), increment HL and decrement B	2	16	•	↕ (3)	X	X	1	X	11 101 101 10 101 011
OUTI	(C) · (HL) B · B 1 HL · HL 1	Load output port (C) with location (HL), increment HL and decrement B	2	16	•	↕ (3)	X	X	1	X	11 101 101 10 100 011
POP IX	IX_H · (SP + 1) IX_L · (SP)	Load IX with top of stack	2	14	•	•	•	•	•	•	11 011 101 11 100 001
POP IY	IY_H · (SP + 1) IY_L · (SP)	Load IY with top of stack	2	14	•	•	•	•	•	•	11 111 101 11 100 001
POP qq	qq_H · (SP + 1) qq_L · (SP)	Load Reg. pair qq with top of stack	1	10	•	•	•	•	•	•	11 qq0 001 (G)
PUSH IX	(SP 2) · IX_L (SP 1) · IX_H	Load IX onto stack	2	15	•	•	•	•	•	•	11 011 101 11 100 101
PUSH IY	(SP 2) · IY_L (SP 1) · IY_H	Load IY onto stack	2	15	•	•	•	•	•	•	11 111 101 11 100 101
PUSH qq	(SP 2) · qq_L (SP 1) · qq_H	Load Reg. pair qq onto stack	1	11	•	•	•	•	•	•	11 qq0 101 (G)
RES b, r	S_b · 0	Reset Bit b of Reg. r		8	•	•	•	•	•	•	11 001 011 (B) 10 bbb rrr (E)
RES b, (HL)	S_b · 0, (HL)	Reset Bit b of loc. (HL)		15	•	•	•	•	•	•	11 001 011 10 bbb 110
RES b, (IX + d)	S_b · 0, (IX + d)	Reset Bit b of loc. (IX + d)		23	•	•	•	•	•	•	11 011 101 11 001 011 dd ddd ddd 10 bbb 110
RES b, (IY + d)	S_b · 0, (IY + d)	Reset Bit b of loc. (IY + d)		23	•	•	•	•	•	•	11 111 101 11 001 011 dd ddd ddd 10 bbb 110
RET	PC_L · (SP) PC_H · (SP + 1)	Return from subroutine	1	10	•	•	•	•	•	•	11 001 001
RET cc	If condition cc is false cont. else (PC_L · (SP) PC_H · (SP + 1)	Return from subroutine if condition cc is true	1	5 if CC false 11 if CC true	•	•	•	•	•	•	11 cc 000 (H)
RETI		Return from interrupt	2	14	•	•	•	•	•	•	11 101 101 01 001 101
RETN		Return from non-maskable interrupt	2	14	•	•	•	•	•	•	11 101 101 01 000 101
RL r		Rotate left through carry Reg. r		2	↕	↕	P	↕	0	0	11 001 011 (B) 00 010 rrr
RL (HL)		Rotate left through carry loc. (HL)		4	↕	↕	P	↕	0	0	11 001 011 00 010 110
RL (IX + d)		Rotate left through carry loc. (IX + d)		6	↕	↕	P	↕	0	0	11 011 101 11 001 011 dd ddd ddd 00 010 110
RL (IY + d)	m r, (HL), (IX + d), (IY + d), A	Rotate left through carry loc. (IY + d)		6	↕	↕	P	↕	0	0	11 111 101 11 001 011 dd ddd ddd 00 010 110
RLA		Rotate left ACC through carry	1	4	↕	•	•	•	0	0	00 010 111

8

MNEMONIC	SYMBOLIC OPERATION	DESCRIPTION	NO. BYTES	NO. T STATES	FLAGS C	Z	P/V	S	N	H	OP CODE 76 543 210
RLC (HL)		Rotate location (HL) left circular	2	15	↕	↕	P	↕	0	0	11 001 011 00 000 110
RLC (IX + d)		Rotate location (IX + d) left circular	4	23	↕	↕	P	↕	0	0	11 011 101 11 001 011 dd ddd ddd 00 000 110
RLC (IY + d)	 m = r, (HL), (IX + d), (IY + d), A	Rotate location (IY + d) left circular	4	23	↕	↕	P	↕	0	0	11 111 101 11 001 011 dd ddd ddd 00 000 110
RLC r		Rotate Reg. r left circular	2	8	↕	↕	P	↕	0	0	11 001 011 (B) 00 000 rrr
RLCA		Rotate left circular ACC	1	4	↕	•	•	•	0	0	00 000 111
RLD		Rotate digit left and right between ACC and location (HL)	2	18	•	↕	P	↕	0	0	11 101 101 01 101 111
RR r		Rotate right through carry Reg. r		2	↕	↕	P	↕	0	0	11 001 011 (B) 00 011 rrr
RR (HL)		Rotate right through carry loc. (HL)		4	↕	↕	P	↕	0	0	11 001 011 00 011 110
RR (IX + d)		Rotate right through carry loc. (IX + d)		6	↕	↕	P	↕	0	0	11 011 101 11 001 011 dd ddd ddd 00 011 110
RR (IY + d)	 m = r, (HL), (IX + d), (IY + d), A	Rotate right through carry loc. (IY + d)		6	↕	↕	P	↕	0	0	11 111 101 11 001 011 dd ddd ddd 00 011 110
RRA		Rotate right ACC through carry	1	4	↕	•	•	•	0	0	00 011 111
RRC r		Rotate Reg. r right circular		2	↕	↕	P	↕	0	0	11 001 011 (B) 00 001 rrr
RRC (HL)		Rotate loc. (HL) right circular		4	↕	↕	P	↕	0	0	11 001 011 00 001 110
RRC (IX + d)		Rotate loc. (IX + d) right circular		6	↕	↕	P	↕	0	0	11 011 101 11 001 011 dd ddd ddd 00 001 110
RRC (IY + d)	m = r, (HL), (IX + d), (IY + d), A	Rotate loc. (IY + d) right circular		6	↕	↕	P	↕	0	0	11 111 101 11 001 011 dd ddd ddd 00 001 110
RRCA		Rotate right circular ACC	1	4	↕	•	•	•	0	0	00 001 111
RRD		Rotate digit right and left between ACC and location (HL)	2	18	•	↕	P	↕	0	0	11 101 101 01 100 111
RST t	(SP − 1) ← PC_H (SP − 2) ← PC_L PC_H ← 0, PC_L ← T	Restart to location T	1	11	•	•	•	•	•	•	11 ttt 111
SBC A, r	A ← A − r − CY	Subtract Reg. r from ACC w/carry	1	4	↕	↕	V	↕	1	↕	10 011 rrr (B)
SBC A, n	A ← A − n − CY	Subtract value n from ACC with carry		7	↕	↕	V	↕	1	↕	11 011 110 nn nnn nnn
SBC A, (HL)	A ← A − (HL) − CY	Sub. loc. (HL) from ACC w/carry		7	↕	↕	V	↕	1	↕	10 011 110
SBC A, (IX + d)	A ← A − (IX + d) − CY	Subtract loc. (IX + d) from ACC with carry		19	↕	↕	V	↕	1	↕	11 011 101 10 011 110 dd ddd ddd
SBC A, (IY + d)	A ← A − (IY + d) − CY	Subtract loc. (IY + d) from ACC with carry		19	↕	↕	V	↕	1	↕	11 111 101 10 011 110 dd ddd ddd
SBC HL, ss	HL ← HL − ss − CY	Subtract Reg. pair ss from HL with carry	2	15	↕	↕	V	↕	1	X	11 101 101 (A) 01 ss0 010
SCF	CY ← 1	Set carry flag (C = 1)	1	4	1	•	•	•	0	0	00 110 111
SET b, (HL)	$(HL)_b$ ← 1	Set Bit b of location (HL)	2	15	•	•	•	•	•	•	11 001 011 (E) 11 bbb 110
SET b, (IX + d)	$(IX + d)_b$ ← 1	Set Bit b of location (IX + d)	4	23	•	•	•	•	•	•	11 011 101 (E) 11 001 011 dd ddd ddd 11 bbb 110

MNEMONIC	SYMBOLIC OPERATION	DESCRIPTION	NO. BYTES	NO. T STATES	FLAGS C	Z	P/V	S	N	H	OP CODE 76 543 210
SET b, (IY + d)	$(IY + d)_b \leftarrow 1$	Set Bit b of location (IY + d)	4	23	•	•	•	•	•	•	11 111 101 Ⓔ 11 001 011 dd ddd ddd 11 bbb 110
SET b, r	$r_b \leftarrow 1$	Set Bit b of Reg. r	2	8	•	•	•	•	•	•	11 001 011 Ⓑ 11 bbb rrr
SLA r		Shift Reg. r left arithmetic		8	↕	↕	P	↕	0	0	11 001 011 Ⓑ 00 100 rrr
SLA (HL)		Shift loc. (HL) left arithmetic		15	↕	↕	P	↕	0	0	11 001 011 00 100 110
SLA (IX + d)	m ← r, (HL), (IX + d), (IY + d)	Shift loc. (IX + d) left arithmetic		23	↕	↕	P	↕	0	0	11 011 101 11 001 011 dd ddd ddd 00 100 110
SLA (IY + d)		Shift loc. (IY + d) left arithmetic		23	↕	↕	P	↕	0	0	11 111 101 11 001 011 dd ddd ddd 00 100 110
SRA r		Shift Reg. r right arithmetic		8	↕	↕	P	↕	0	0	11 001 011 Ⓑ 00 101 rrr
SRA (HL)		Shift loc. (HL) right arithmetic		15	↕	↕	P	↕	0	0	11 001 011 00 101 110
SRA (IX + d)	m ← r, (HL), (IX + d), (IY + d)	Shift loc. (IX + d) right arithmetic		23	↕	↕	P	↕	0	0	11 011 101 11 001 011 dd ddd ddd 00 101 110
SRA (IY + d)		Shift loc. (IY + d) right arithmetic		23	↕	↕	P	↕	0	0	11 111 101 11 001 011 dd ddd ddd 00 101 110
SRL r		Shift Reg. r right logical		8	↕	↕	P	↕	0	0	11 001 011 Ⓑ 00 111 rrr
SRL (HL)		Shift loc. (HL) right logical		15	↕	↕	P	↕	0	0	11 001 011 00 111 110
SRL (IX + d)	m r, (HL), (IX + d), (IY + d)	Shift loc. (IX + d) right logical		23	↕	↕	P	↕	0	0	11 011 101 11 001 011 dd ddd ddd 00 111 110
SRL (IY + d)		Shift loc. (IY + d) right logical		23	↕	↕	P	↕	0	0	11 111 101 11 001 011 dd ddd ddd 00 111 110
SUB r	A ← A r	Subtract Reg. r from ACC		4	↕	↕	V	↕	1	↕	10 010 rrr Ⓑ
SUB n	A ← A n	Subtract value n from ACC		7	↕	↕	V	↕	1	↕	11 010 110 nn nnn nnn
SUB (HL)	A ← A (HL)	Subtract loc. (HL) from ACC		7	↕	↕	V	↕	1	↕	10 010 110
SUB (IX + d)	A ← A (IX + d)	Subtract loc. (IX + d) from ACC		19	↕	↕	V	↕	1	↕	11 011 101 10 010 110 dd ddd ddd
SUB (IY + d)	A ← A (IY + d)	Subtract loc. (IY + d) from ACC		19	↕	↕	V	↕	1	↕	11 111 101 10 010 110 dd ddd ddd
XOR r	A ← A∀r	Exclusive 'OR' Reg. r and ACC		4	↕	↕	P	↕	1	↕	10 101 rrr Ⓑ
XOR n	A ← A∀n	Exclusive 'OR' value n and ACC		7	↕	↕	P	↕	1	↕	11 101 110 nn nnn nnn
XOR (HL)	A ← A∀ (HL)	Exclusive 'OR' loc. (HL) and ACC		7	↕	↕	P	↕	1	↕	10 101 110
XOR (IX + d)	A ← A∀ (IX + d)	Exclusive 'OR' loc. (IX + d) and ACC		19	↕	↕	P	↕	1	↕	11 011 101 10 101 110 dd ddd ddd
XOR (IY + d)	A ← A∀ (IY + d)	Exclusive 'OR' loc. (IY + d) and ACC		19	↕	↕	P	↕	1	↕	11 111 101 10 101 110 dd ddd ddd

FLAG NOTES:

① P/V flag is 0 if B-1=0, else P/V=1

② Z=1 if A=(HL), else Z=0

③ If B-1=0, Z flag set, else reset

FLAG DEFINITIONS:

• = Flag not affected

0 = Flag reset

1 = Flag set

X = Flag unknown

↕ = Flag affected according to result of operation

V = Overflow set

P = Parity set

IFF = Interrupt flip-flop set

Ⓐ

Reg	ss
BC	00
DE	01
HL	10
SP	11

Ⓑ

Reg	r
A	111
B	000
C	001
D	010
E	011
H	100
L	101

Ⓒ

Reg	pp
BC	00
DE	01
IX	10
SP	11

Ⓓ

Reg	rr
BC	00
DE	01
IY	10
SP	11

Ⓔ

Bit	b
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Ⓕ

Reg	r,r'
A	111
B	000
C	001
D	010
E	011
H	100
L	101

Ⓖ

Reg	qq
BC	00
DE	01
HL	10
AF	11

Ⓗ

CC	Condition		Relevant Flag
000	NZ	Non Zero	Z
001	Z	Zero	Z
010	NC	Non Carry	C
011	C	Carry	C
100	PO	Parity Odd	P/V
101	PE	Parity Even	P/V
110	P	Sign Positive	S
111	M	Sign Negative	S

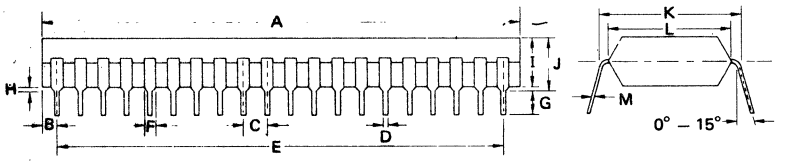
Ⓘ

Reg	r
B	000
C	001
D	010
E	011
H	100
L	101
F	110
A	111

FLAG DESCRIPTION:

C = Carry/Link
Z = Zero
P/V = Parity/Overflow
S = Sign
N = Add/Subtract
H = Half Carry

8



PACKAGE OUTLINE
μPD780C
μPD780-1C

(Plastic)

ITEM	MILLIMETERS	INCHES
A	51.5 MAX	2.028 MAX
B	1.62	0.064
C	2.54 ± 0.1	0.10 ± 0.004
D	0.5 ± 0.1	0.019 ± 0.004
E	48.26	1.9
F	1.2 MIN	0.047 MIN
G	2.54 MIN	0.10 MIN
H	0.5 MIN	0.019 MIN
I	5.22 MAX	0.206 MAX
J	5.72 MAX	0.225 MAX
K	15.24	0.600
L	13.2	0.520
M	$0.25^{+0.1}_{-0.05}$	$0.010^{+0.004}_{-0.002}$